

# T5L\_DGUSII Application Development Guide

Version 2.8



Contents

|  |     |
|--|-----|
| <b>Chapter 1 Quick Start</b> .....                           | 4   |
| 1.1 Naming Rule .....  | 4   |
| 1.2 Interface Definition and Wiring .....                    | 5   |
| 1.3 Power Supply .....                                       | 8   |
| 1.4 Driver Installation .....                                | 8   |
| 1.5 Software Installation .....                              | 9   |
| <b>Chapter 2 Development System</b> .....                    | 10  |
| 2.1 T5L ASIC .....   | 10  |
| 2.2 T5L_DGUS II Development System .....                     | 12  |
| 2.3 Features .....   | 14  |
| 2.4 Software Process .....                                   | 15  |
| 2.5 FLASH Memory Allocation .....                            | 16  |
| 2.6 RAM Memory Variable Address Space .....                  | 17  |
| 2.7 RAM Description Pointer Space .....                      | 18  |
| <b>Chapter 3 SD Card Interface</b> .....                     | 21  |
| 3.1 MicroSD(TF) Card Format .....                            | 21  |
| 3.2 SD Card Interface Download Method .....                  | 22  |
| 3.3 Download File .....                                      | 23  |
| <b>Chapter 4 Communication Protocol</b> .....                | 45  |
| 4.1 UART(UART2) .....  | 45  |
| 4.2 UART2 Debug Interface Instructions .....                 | 45  |
| <b>Chapter 5 System Variable Interface</b> .....             | 49  |
| 5.1 System Variable Interface .....                          | 49  |
| 5.2 FSK bus interface .....                                  | 77  |
| 5.3 Network Interface .....                                  | 79  |
| <b>Chapter 6 Touch Variable Configuration (13.BIN)</b> ..... | 85  |
| 6.1 List of Touch Variables .....                            | 86  |
| 6.2 Variable Data Input .....                                | 87  |
| 6.3 Popup Menu .....   | 90  |
| 6.4 Incremental Adjustment .....                             | 92  |
| 6.5 Drag Adjustment .....                                    | 94  |
| 6.6 Return Key Code .....                                    | 96  |
| 6.7 Text Input .....   | 98  |
| 6.8 Synchrodata Return .....                                 | 106 |
| 6.9 Rotation Adjustment .....                                | 109 |
| 6.11 Page Sliding .....                                      | 112 |
| 6.12 Slide Icon Selection .....                              | 114 |
| 6.13 Bit button .....  | 116 |
| <b>Chapter 7 Display Variable Configuration</b> .....        | 119 |
| 7.1 List of Display Variables .....                          | 120 |
| 7.2 Variables Icon .....                                     | 123 |
| 7.3 Animation Icon .....                                     | 126 |
| 7.4 Slider Display .....                                     | 130 |
| 7.5 Artistic Variables .....                                 | 132 |
| 7.6 Image Animation .....                                    | 134 |
| 7.7 Icon Rotation .....                                      | 136 |
| 7.8 Bit Icon .....   | 139 |
| 7.9 Icon Page Tran .....                                     | 143 |
| 7.10 Icon Overlay .....                                      | 146 |
| 7.11 Batch Variable Icon .....                               | 150 |
| 7.12 Data Variables .....                                    | 152 |
| 7.12.3 Example of Instructions .....                         | 154 |
| 7.13 Text Display .....                                      | 157 |
| 7.14 RTC Display .....                                       | 162 |
| 7.15 HEX Data .....  | 166 |
| 7.16 Text Scroll .....                                       | 168 |
| 7.17 Data Window .....                                       | 171 |
| 7.19 Roll Character .....                                    | 179 |
| 7.20 GTF Icon Font .....                                     | 183 |



|   |            |
|---|------------|
| 7.21 Real-Time Curves .....                   | 188        |
| 7.22 Basic Graphics .....                     | 198        |
| 7.23 Process Bar .....                        | 227        |
| 7.24 Zone Scrolling .....                     | 229        |
| 7.25 QR Code .....                            | 230        |
| 7.26 Brightness .....                         | 233        |
| 7.27 Data Transmit .....                      | 234        |
| 7.28 Video .....                              | 235        |
| 7.28.3 ICL&WAE file .....                     | 236        |
| <b>Chapter 8 DGUS Basic Functions</b> .....   | <b>242</b> |
| 8.1 Import .....                              | 242        |
| 8.2 Edit tool .....                           | 242        |
| 8.3 Front and Back .....                      | 242        |
| 8.4 Viewing the Usage of VP .....             | 242        |
| 8.5 Preview .....                             | 242        |
| 8.6 New Resolution .....                      | 243        |
| 8.7 Converting Resolution .....               | 243        |
| 8.8 Replace T5 project with T5L project ..... | 243        |



## Chapter 1 Quick Start

### 1.1 Naming Rule

|                               |                      |  |  |   |   |   |
|-------------------------------|----------------------|--|--|---|---|---|
| <b>Product Line Code</b>      | DM                   | DWIN Smart LCM Product Line  |  |   |   |   |
| <b>Color Code</b>             | 1 Character          | T=65K color(16-bit) G=16.7M/262K color(18-bit/24-bit)  |  |   |   |   |
| <b>Resolution</b>             | 5 Digits             | 24240=240*240<br>64360=640*360<br>80600=800*600<br>12720=1280*720<br>19108=1920*1080   | 32240=320*240<br>64480=640*480<br>10600=1024*600<br>12800=1280*800<br>19480=1920*480 | 48270=480*272<br>80480=800*480<br>10768=1024*768<br>13768=1364*768/1366*768 | 48320=480*320<br>85480=854*480<br>12480=1280*480                          |   |
| <b>Classification</b>         | 1 Character          | F=COF structure<br>C=Commerce grade<br>S= Harsh environment grade  | Y=Beauty grade<br>T=Industrial grade<br>Q= Automotive grade                          | L=Simple application grade<br>K=Medical grade                               |   |   |
| <b>Size</b>                   | 3 Digits             | 020=2.2inch<br>040=4.0inch<br>070=7.0inch<br>104=10.4inch<br>156=15.6inch  | 024=2.4inch<br>041=4.1inch<br>080=8.0inch<br>116=11.6inch<br>173 =17. 3 inch         | 028=2.8inch<br>043=4.3inch<br>088=8.8inch<br>121=12.1inch<br>185=18.5inch   | 030=3.0inch<br>050=5.0inch<br>097=9.7inch<br>140=14.0inch<br>215=21.5inch | 035=3.5inch<br>056 =5. 6 inch<br>101=10.1inch<br>150=15.0inch |
| -                             | -                    | Separator  |  |   |   |   |
| <b>Attribute Code</b>         | 1 Digit or character | 0= Basic type<br>1= Basic type with shell<br>2= Analogue video processing platform<br>3= Android platform<br>4= Digital video processing platform products<br>A= DGUSII kernel product |  |   |   |   |
| <b>Hardware Serial Number</b> | 1 Digit              | 0 - 9 to distinguish between different hardware versions   |  |   |   |   |
| <b>LCD Temperature Level</b>  | 1 Character          | N= Normal temperature W= Wide temperature  |  |   |   |   |
| <b>TP Category</b>            | 1 or 2 Character     | N= Without touch panel TR= Resistive touch panel TC= Capacitive touch panel  |  |   |   |   |
| <b>Custom Tag</b>             | Z+ Number            | Z01- Z99, standard is blank  |  |   |   |   |
| <b>Extended Memory Tag</b>    | F+1 Number           | F0= 512MB F1=1GB F2= 2GB F3= 3GB, standard is blank  |  |   |   |   |



## 1.2 Interface Definition and Wiring

| Pin Class |          |                    | Socket Type |        |     |
|-----------|----------|--------------------|-------------|--------|-----|
| PIN Name  | Pin Type | Pin Description    | CN1         | CN2    | CN3 |
| VCC       | P        | Power supply input | 1,2         | 1,2,3  | 1   |
| TX4       | O        | UART4 output       | 3           | 4      | 2   |
| TX2       | O        | UART2 output       | 4           | 5      | 3   |
| RX2       | I        | UART2 input        | 5           | 6      | 4   |
| RX4       | I        | UART4 input        | 6           | 7      | 5   |
| GND       | P        | Power supply input | 7,8         | 8,9,10 | 6   |

① I: INPUT, O: OUTPUT, P: POWER

② PCB pins of the same defined pins have been connected in parallel.

③ CN1: Use 8 Pin 2.0 mm spacing patch socket;

CN2: Use 10 Pin 1.0 mm spacing FPC mount;

CN3: Additional 6 Pin 2.54 mm spacing through hole pad is provided.

④ UART2 is the communication protocol developed by DGUSII, please refer to the detailed description in Chapter 4; T5L DGUS products are equipped with extended UART4, and its configuration and communication need to be completed through DWIN OS program. TA mode is only available for UART2 communications at the moment.

⑤ DWIN adopt TTL/RS232 compatible interface. Users can use 0 Ω resistor or solder to short connect directly.

R232=0 (short circuit) selects TTL level input; R232=1(disconnect) selects RS232 level.

As shown below, the place is marked by white silk screen.

By default, the factory disconnects the 232 level, and selects the TTL level. That is OFF=232, ON=TTL.



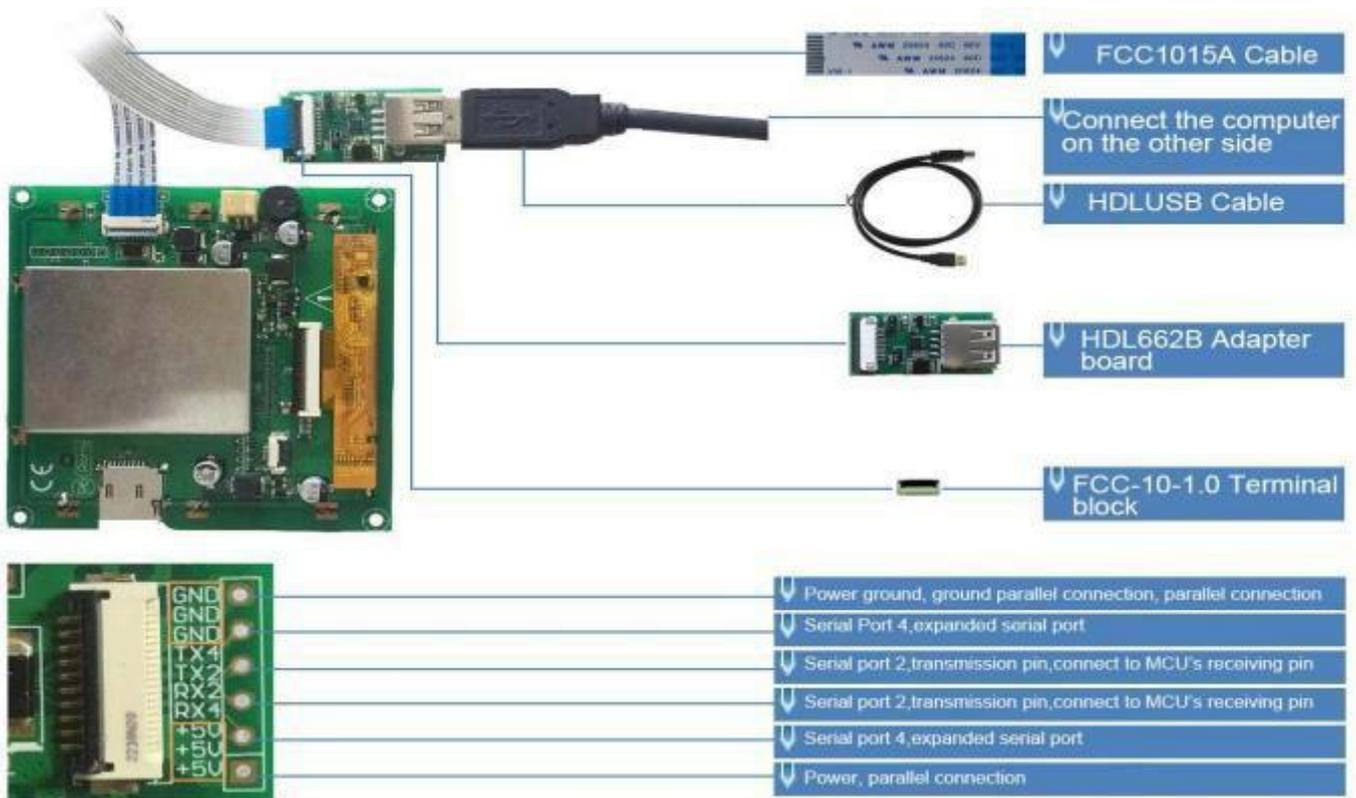
ON: short circuit, TTL level  
OFF: disconnect, 232 level  
232 level is not connected by default

Level selection

### (1) 10 Pin Interface

One- end of the different side FCC row line is connected to terminal block of the screen, and the other end is connected to the DWIN HDL6 6 2 B adapter board, both blue side face up.

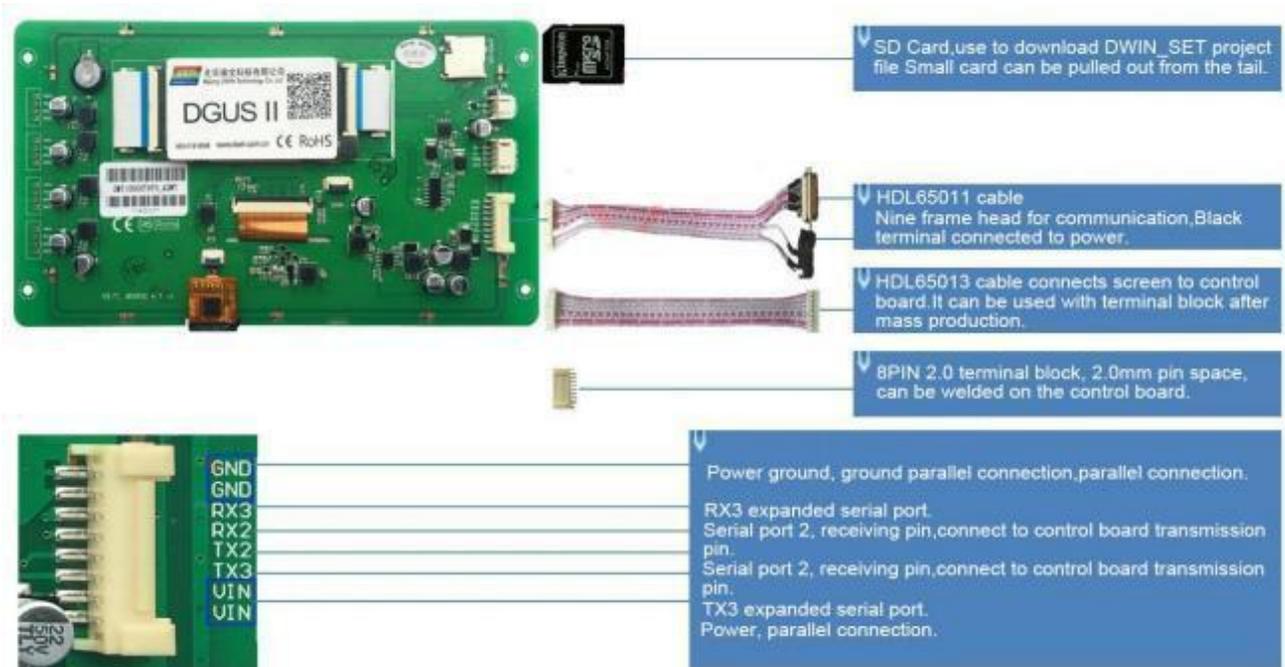
One-end of the double-male USB cable is connected to the adapter board, and the other end is connected to the computer for communication.



10 pin interface definition and wiring

## (2) 8 Pin Interface

The 8 PIN interface is spaced by 2.0 mm.



8 pin 2.0 interface definition and wiring



### 1.3 Power Supply

Confirm the voltage and power consumption corresponding to the screen according to the catalogue or data-sheet.

Power on the screen with the DC regulated power supply 5V/12V/24V.

The power supply plays a very important role in the normal display of the screen. If the voltage is too small, the current is unstable, and the power is too low, it may cause abnormal display such as flickering and black screen.

### 1.4 Driver Installation

#### 1.4.1 Serial Driver

The USB-to-UART chips of DWIN serial adapter boards are XR21V1410 chip and CP2102 chip.

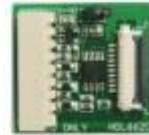
According to the chip type, you can download from DWIN official website or consult customer service to obtain and install the corresponding driver for DGUS screen communication.

XR21v1410 Driver for x86 & x64



HDL662B

CP210x USB Driver for x86 & x64



HDL662C

**NOTE:**

- 1. x86 corresponds to 32-bit Windows operating system
- 2. x64 corresponds to 64-bit Windows operating system

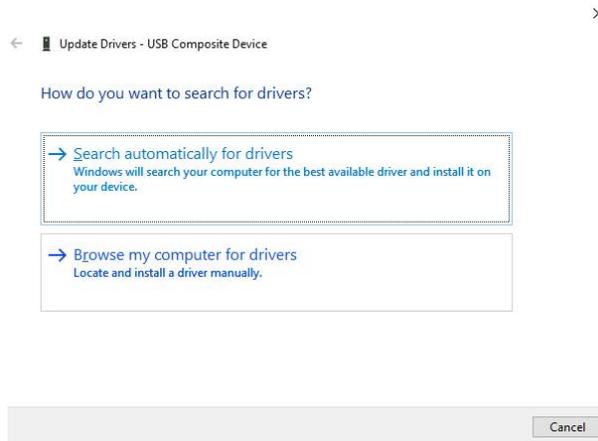
Serial Port Driver

Steps are as follows .

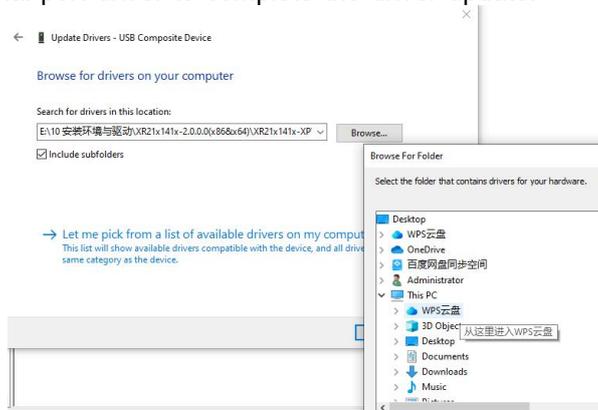
Step 1: Right-click "Computer"; click "Properties" at the bottom; click "Device Manager" at the top left, right-click "USB Device" on the "Device Manager" operation box, and select "Update Driver".



Step 2: Select "Next" in the pop-up window.



Step 3: Select the path of the serial port driver to complete the driver update.



Step 4 : After the installation is complete, the port number corresponding to the driver can be viewed in the device manager.

## 1.4.2 Software Operating Environment Driver

Obtain the software operating environment and double-click to install it.

## 1.5 Software Installation

Download DGUS V7.624 from our official website [https:// www. dwin-global.com](https://www.dwin-global.com) or get from customer service.

### 1.5.1 New Project

Taking the variable data input control and preview function an example, the steps as follows .

Step 1 : Prepare the background image of the same resolution as the screen, in jpg /bmp format, 24 - bit color. The pictures name starts from 00 .

Step 2: Download and open DGUS V7.626.

Step 3: Import the image.

Step 4: Configure the Interface.

Step 5: Click save and generate.

Step 6: Project preview.



## Chapter 2 Development System

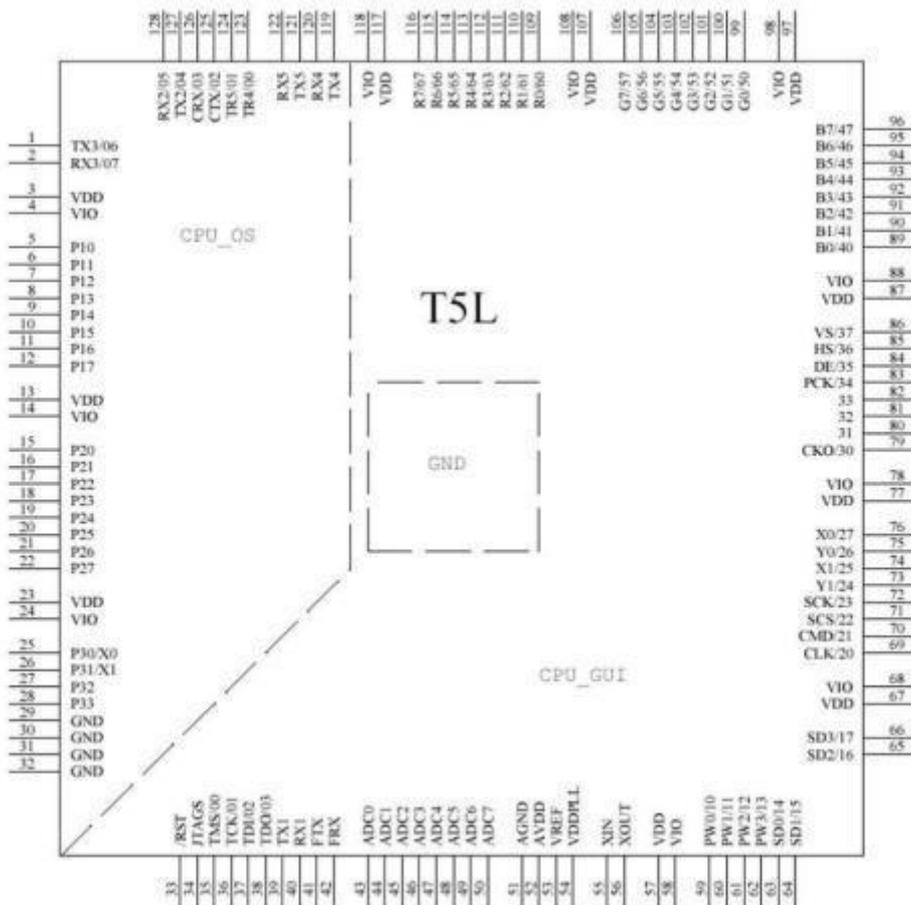
### 2.1 T5L ASIC

The T5L series ASIC is a single-chip and dual-core ASIC designed by DWIN Technology for AIoT applications with low power consumption, high cost performance, GUI and highly-integrated application, including T5L1 (low resolution) and T5L2 (high resolution). The main features are as below.

- (1) Adopts widely-used, mature, and stable 8051 core. The maximum operating frequency of T5L is up to 250 MHz, 1T (single instruction cycle) high speed operation.
- (2) Separated GUI CPU Core running DGUSII System:
  - High-speed display memory, 2.4 GB/S bandwidth, 24bit color display resolution supporting 800\*600 (T5L1) or 1366\*768 (T5L2).
  - 2D hardware acceleration, the decompression speed of JPEG is up to 200fps@ 180 \* 800 , the UI interface with animation and icons as its main feature is extremely cool and smooth.
  - JPEG stores images and icons in the compressed mode, external memory is greatly reduced to low-cost 16 Mbytes SPI Flash.
  - With a max 400Hz touch point speed, the touch panel supports resistance or capacitance, and its sensitivity can be adjusted.
  - High-quality speech compression storage and playback.
  - 128K bytes variable storage space, exchanging data with OS CPU core using memory interface, extremely simple to apply.
  - 2-way 10 bit, 800KHz, DC/DC controller, simplify LED backlight, analog power supply design, and save cost and space.
  - 1 - way 15 bit 32Ksps PWM digital power amplifier driver loudspeaker, save power amplifier cost and achieve high signal-to-noise ratio and sound quality restoration.
  - Supporting PC configuration development and simulation, supporting background remote upgrade.
- (3) Separated OS CPU core runs user 8051 code or DWIN OS system, user CPU is omitted in practical application.
  - Standard 8051 architecture and instruction set, 64K bytes code space, 32K bytes on-chip RAM.
  - 64-bit integer mathematical operation unit (MDU), including 64 bit MAC and 64 bit divider.
  - Built-in software WDT, three 16-bit Timers, 12 interrupt signals with the highest four interrupt nesting.
  - 22 IO, 4UARTs, 1 CAN, 7 12-bit A/Ds, 2 16-bit adjustable-resolution PWM.
  - Support IAP online simulation and debugging, unlimited number of breakpoints.
  - Upgrade code online through the DGUS system .
- (4) 1Mbytes on-chip Flash with DWIN patent encryption technology ensure code and data security, eliminate copycat and cloning.



- (5) Reducing crystal requirements and PCB design challenges for a variety of inexpensive wide- range tuned.
- (6) Impedance crystal oscillators and PLL.
- (7) 3.3V IO voltage, can adapt to 1.8/2.5/3.3 various levels.
- (8) Supporting SD interface download and configuration, supporting SD card file reading and rewriting.
- (9) Supporting DWIN WiFi module to access to DWIN cloud directly, and easily developing various cloud platform applications.
- (10) Working temperature ranges from - 40 °C to 85 °C (Customizable IC for - 55 °C to 105 °C operating temperature range).
- (11) With low power consumption and strong anti- interference ability, it can work steadily on the double- sided PCB design, and is easy to pass EMC/ EMI test.
- (12) Using 0.4 mm ELQFP128 packaging, low manufacturing difficulty and low cost.
- (13) Providing **T5L IC+ LCD+ touch panel** cost-effective supporting scheme and comprehensive technical service support for industry customers.



The arrangement of T5L pins



## 2.2 T5L\_DGUS II Development System

DGUS development system is composed of DGUS software and DGUS screen.

DGUS is the DWIN Graphic Utilized Software.

DGUS screen works based on configuration files, so the development process is that users complete variable configuration files by DGUS software.

### (1) Variable programming

Users can make a table in advance to plan the required variable address and facilitate the modification and maintenance of subsequent projects.

### (2) Interface design

The simple interface can be made by yourself; the complex or beautiful interface can be made by UI or DM. Icon-making is the same as image.

### (3) Interface configuration

The interface is configured by DGUS software on PC.

After completing the project, click " save" and " generate" to generate 13.bin touch configuration file, 14.bin display configuration file and 22.bin initialization file.

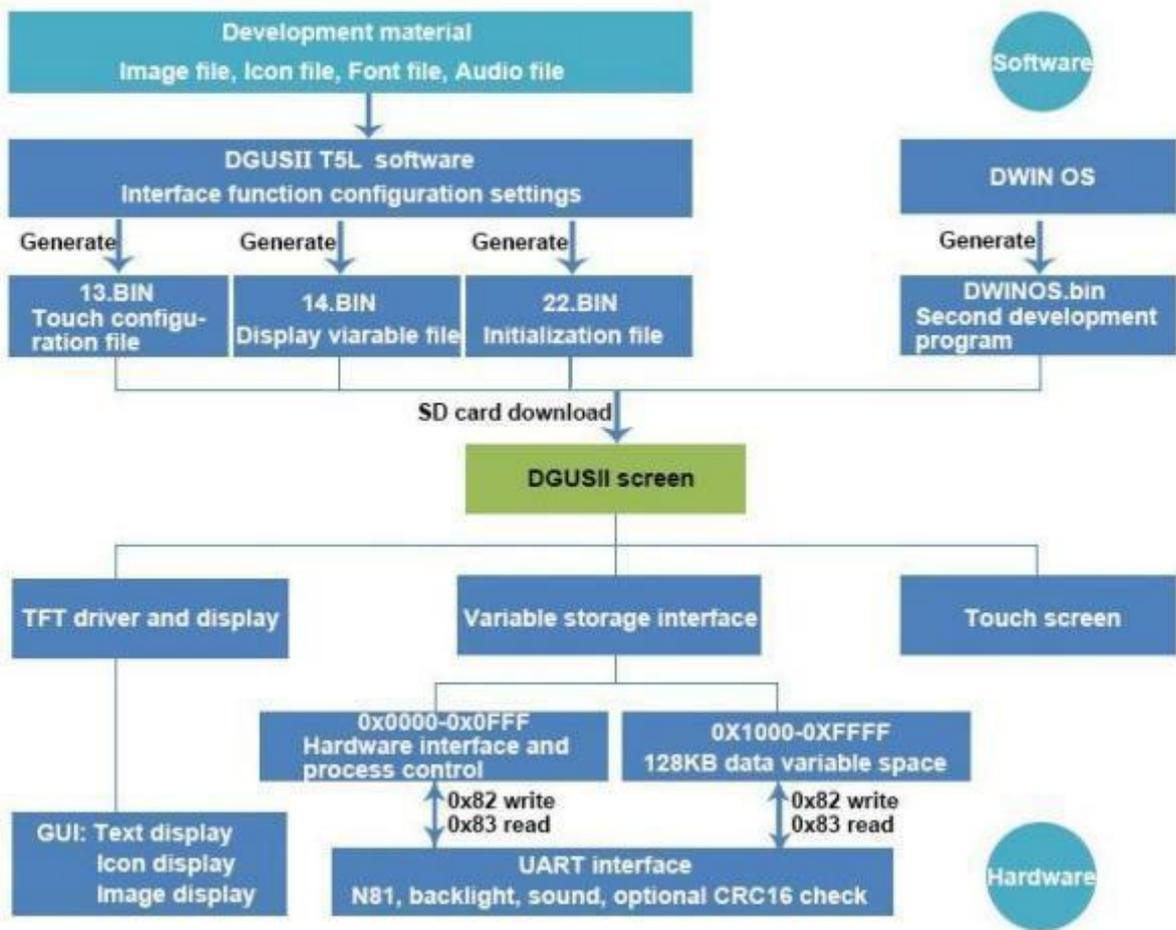
### (4) Debug

Put the download files into the DWIN\_SET folder of SD card and download them to the screen through the SD card.

The sequence is power off - insert SD card - power on - blue screen; download completed: " SD card process END! " - power off; pull out SD card - power on.

### (5) Mass production according to template

After finalizing the template, put the download files, picture files, icons, fonts, etc. into the DWIN\_SET folder, and then mass production download can be realized through SD card.



T5 L\_ DGUSII development system



## 2.3 Features

Different from the traditional LCM control display through timing or instruction, DGUS adopts the direct variable-driven display mode. Displays and operations are based on the pre-set variable configuration files. Thus, the software architecture is simpler and secondary development is easier.

In the DGUS development platform, it is developed through the use of controls. The attributes and methods of the defined data are encapsulated in the control, and the user only needs to configure and use the control on PC.

DGUS has defined the controls required for common HMI, which is easy to use and efficient.

In some medium industrial and automation project applications, when the entire system consists of relatively independent and fully functional components (such as supporting Modbus). The user can directly use the DGUS as the host, use the 485 to form a network between the device and the DWIN screen, and directly develop users based on the secondary development platform carried on the screen. The main control software runs directly on the DGUS screen, replacing the user's CPU.

The T5L\_DGUS II platform specifies a refresh cycle of 20 ms, which means that the contents of 13.bin and 14.bin are read once in every 20ms.

The DGUS II development platform supports up to 256 variables per page, so the processing capability of DGUS II is very powerful.

OS operating mode: real-time operation (1 8051 core runs GUI and 1 8051 core runs user code on DWIN OS platform, which are completely independent). Thus users do not have to worry about whether DGUS and DWIN OS will affect each other when they run on the same system.

The main features of DGUS II software platform based on T5L ASIC CPU are as follows .

- (1) Based on T5 L dual-core ASIC, GUI and OS cores both run at 200 MHz main frequency (350 MHz main frequency version can be customized) , with extremely low power consumption.
- (2) 16 Mbytes low cost SPI Flash(can be extended to 64 Mbytes NOR Flash or 48 Mbytes NOR Flash+ 512 Mbytes NAND Flash), JPEG image, icon compression storage, You can specify the size of background image storage space.
- (3) 512 Kbytes Nor Flash in- chip user database.
- (4) 256Kbytes data variable space.
- (5) Up to 255 display variables per page.
- (6) Support standard T5 DWIN OS platform or 8051 OS CPU core development: hardware CAN lead to 20 IO, 4 UARTs, 1 CAN, multiple ADs, customized services available.
- (7) 20mS DGUS cycle, extremely smooth UI.
- (8) Display variables can be opened, closed or modified in the application to achieve complex display combination functions.
- (9) Touch instructions can be opened, closed or modified in the application to realize complex touch combination functions.
- (10) Support SD interface download and configuration, download file statistics display.

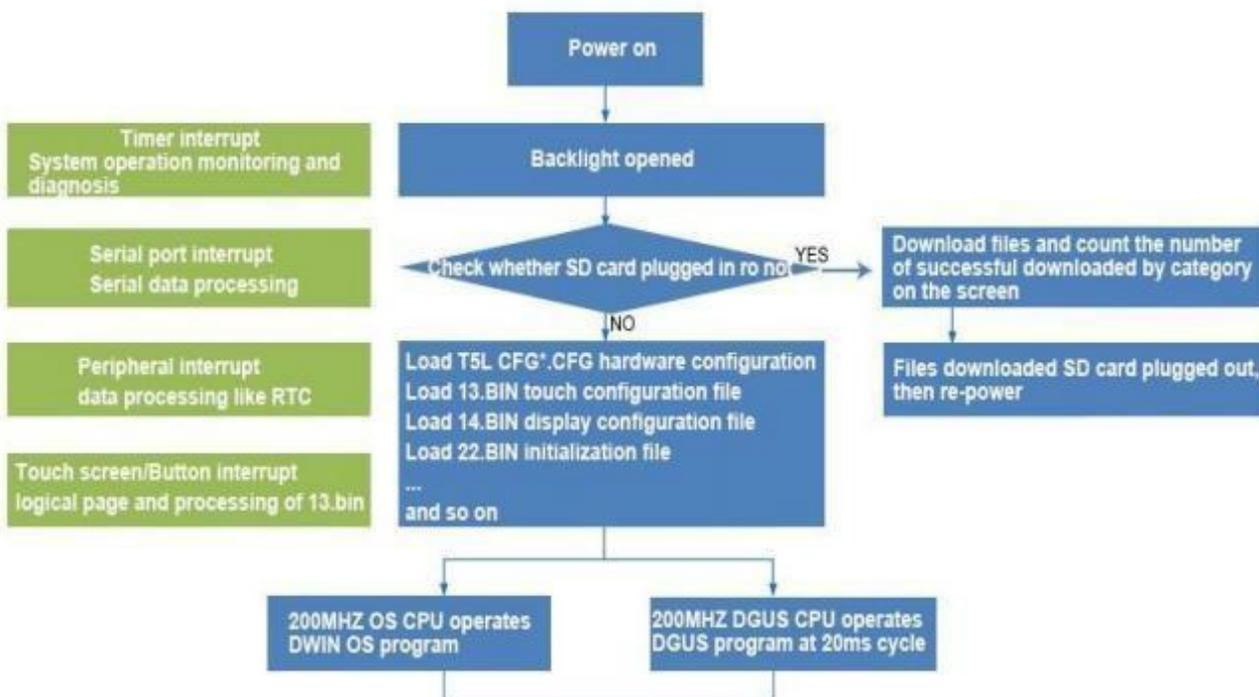


- (11) Support CTP sensitivity adjustment, convenient for users to install the panel ( the thickest to 6 mm toughened glass) application .
- (12) Data variable display, automatically recognize and support the zigzag optimization of the 8 bit encoding font library character display, to support non- standard fonts (such as digital tube font), non- half corner characters display; 8 bit coded word library using DWIN tool software generation.
- (13) Support pre-installed configuration module development mode, greatly improve the speed and quality of UI development.
- (14) Support UART1 online update software and font library, convenient debugging.
- (15) Support TP handwriting input method (need to customize hardware).
- (16) Support digital camera real- time monitoring display (need to customize hardware).

### 2.4 Software Process

In the T5L\_DGUSII development platform, the configuration of the touch operation of touch panel is also stored in 13.bin file in the same way.

The characteristic attributes of variables (characters, data), various animations and other functions to be displayed are represented by codes and stored in the corresponding address of 14.bin file. The operation in the development software of DGUSII is to configure the control of display function or touch function. When the system is powered to run, the system will call 13.bin file and 14.b in file, and then the HIMI can run normally.



## 2.5 FLASH Memory Allocation

The 16 MB Flash memory is divided into 64 fixed 256 KB subspaces, and the file ID ranges from 0 to 63. According to the different contents of the stored files, the memory is mainly divided into two parts:

- (1) 4-12MB font space can save BIN, HZK, DZK format files, the file ID ranges from 00 to 47
- (2) 4-12MB picture space can save ICL files of background picture library and ICL files of icon library. The file ID ranges from 16 to 63.

### INFO:

Note that there are overlaps between the font space and the image space, therefore you should avoid conflicts when naming ID.

For the T5L1 CPU platform, the size of a single picture file in the packed ICL file should not exceed 248 KB, and that in the T5L2 CPU platform should not exceed 760 KB. The download file must be placed in the DWIN\_SET folder of the SD card root directory, which must be a 4 KB sector, FAT32 format SD or SDHC card.

After familiarizing with the division of the shared interval range, you can set the allocation by yourself.

For example, the font file at the user's 23 position occupies 10 intervals. At this time, the background image position cannot start from 32, but needs to start from the 33 position. If the user has a background image that is displayed as a black screen, while other images are displayed normally, check whether the page size exceeds 256KB.



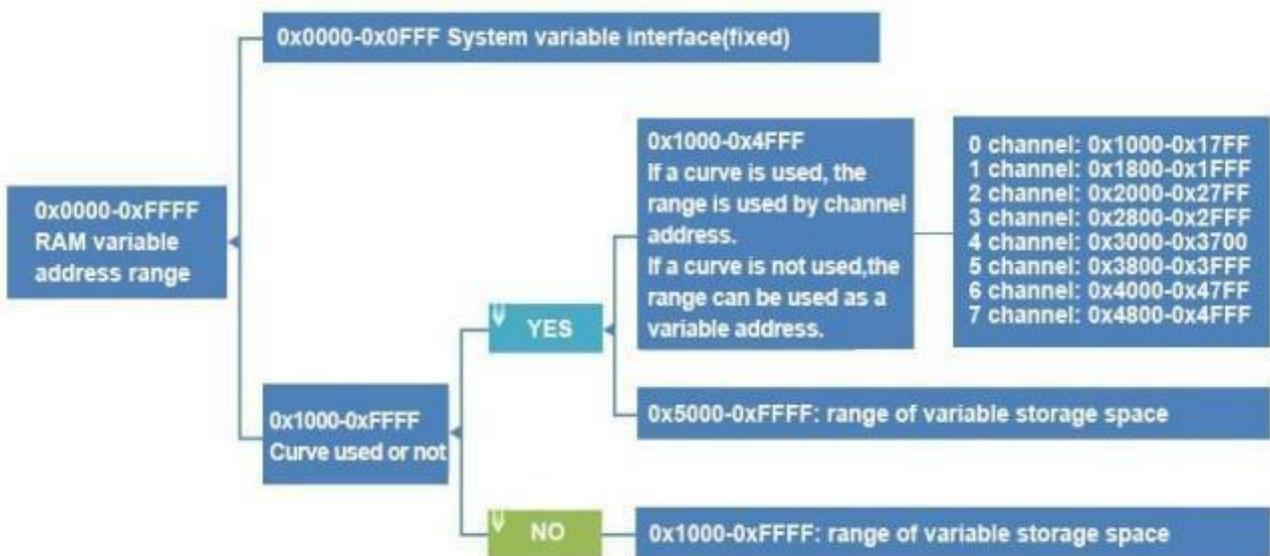
## 2.6 RAM Memory Variable Address Space

The RAM space is fixed at 128KB, which is divided into 0x0000-0xFFFF. Each variable address corresponds to 2 bytes of the corresponding space. A byte corresponds to 8 bits of the corresponding space.

Among them, 0x0000-0x0FFF is the system variable interface address space, which cannot be customized by the user; 0x1000-0xFFFF is available for users.

If 8-channel curves are used at the same time, 0x1000-0x4FFF will be used as the curve buffer address, at this time this part is occupied and cannot be used by other controls.

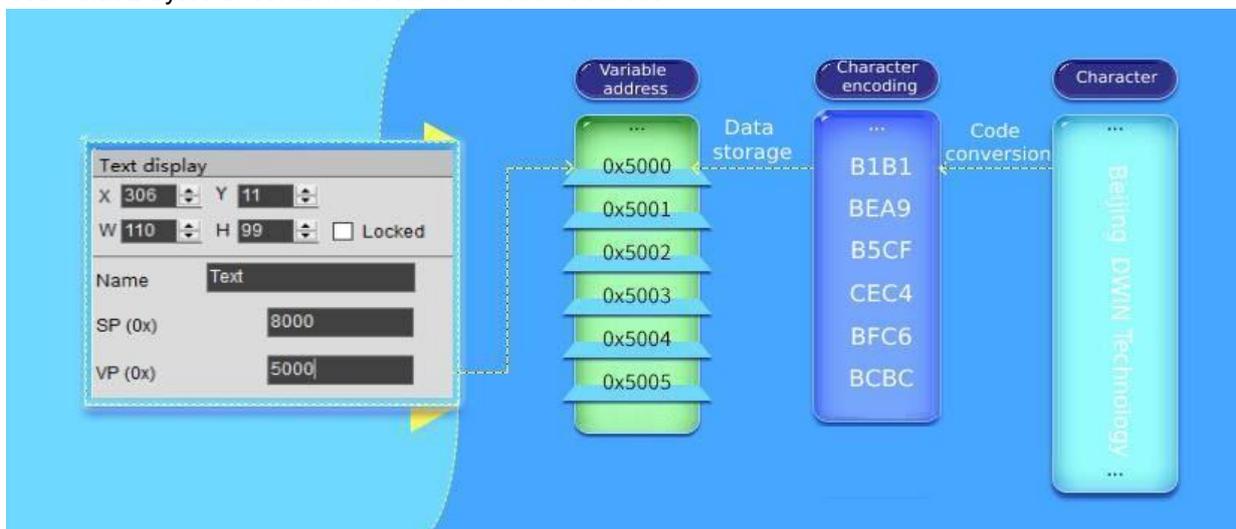
Then the user available address range is: 0x5000-0xFFFF.



The variable address is the first address of the subspace in which one or more variables are stored in the RAM space, and the code of the display variable or the value of the state variable is stored in the space pointed to by this address.

For example, set the variable address of a text display control to 0x5000 and the text content displayed in the control is "Beijing DWIN Technology", then the storage method in the RAM space is shown in the following figure.

It can be seen that 2 bytes of content are stored in each address.



Storage method of data in variable address



If you need to change the characters displayed in the text display control, you only need to change the character encoding stored in the corresponding variable address. The data stored in the variable address can be modified by sending instructions and typing on the touch screen. For example, modifying the text display control can be realized through the text input control. It is only necessary to set the same variable address for the two controls. At the same time, the value stored in the variable address can also be modified by sending instructions.

For detailed instructions, please refer to Variable Display Control.

### 2.7 RAM Description Pointer Space

The description pointer(SP) is the first address of the subspace in the RAM space that stores the attributes describing a variable. The space pointed to by this address stores the attribute values of the display variable, such as display coordinates, color, and font size.

It should be pointed out that the SP and the variable pointer(VP) share the RAM space, and the overlapping of the space should be avoided when assigning variable addresses.

The explanation will be combined with the text display control. (For more instructions on the specific usage of SP, please refer to chapter 7 of this document).

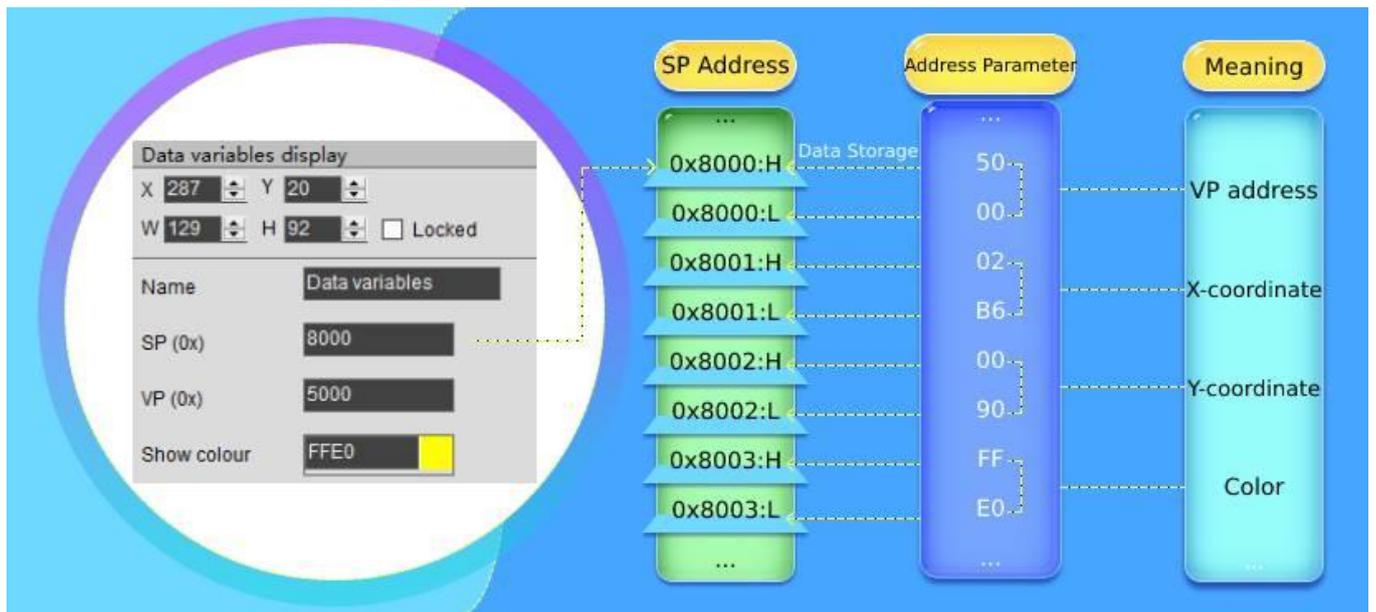
As shown in the table below, "SP offset" indicates the address length that needs to be offset (increased) on the basis of the description pointer(SP) address, and its unit is the word length of "Word". "Definition" explains the attributes represented by this offset address, the common attributes are: coordinate position, color, character size, alignment, etc. This is a common attribute of software functions, but other attributes according to different types. It should be noted that some attributes occupy a word-length address or multiple addresses, some attributes only occupy one byte address, as shown in the table below, 0x04:H high byte indicates the font location, 0x04:L low byte indicates the font size, understand the description pointer usage after understanding the high and low byte writing.

| SP Offset | Definition         | Length (bytes) | Description   |
|-----------|--------------------|----------------|---|
| 0x00      | *VP                | 2              | Variable pointer  |
| 0x01      | X, Y               | 4              | The starting display position, which displays the coordinates of the upper left corner of the string.   |
| 0x03      | COLOR              | 2              | Display Color   |
| 0x04:H    | Lib_ID             | 1              | ASCII font location   |
| 0x04:L    | Font size          | 1              | The number of dots in the x-direction of the character.   |
| 0x05:H    | Alignment          | 1              | Text color.   |
| 0x05:L    | Integer digits     | 1              | 0x00=left-aligned 0x01=right-aligned 0x02=centered  |
| 0x06:H    | Decimal places     | 1              | Index in the FLASH memory of the ASCII Font touse.  |
| 0x06:L    | Variable data type | 1              | 0x00: Integer (2 bytes). Range: -32768 ~ 32767<br>0x01: 32-bit Integer (4 bytes). Range: -2147483648 ~ 2147483647<br>0x01 = long integer (4 bytes) in the range -2147483648 to 2147483647<br>0x02=*VP high byte, unsigned, range 0 to 255<br>0x03=*VP low byte, unsigned, range 0 to 255<br>0x04= very long integer (8 bytes), the range is -9223372036854775808 to 9223372036854775807<br>0x05=unsigned integer (2 bytes), range 0 to 65535<br>0x06=unsigned long integer (4 bytes) in the range 0 to 4294967295 |
| 0x07:H    | Len_unit           | 1              | Variable unit (fixed string) display length, 0x00 means no unit display   |
| 0x07:L    | String_Unit        | Max11          | Unit string, ASCII encoding.  |

The corresponding relationship between the control properties and the instruction storage format table is shown in the figure below, and other properties can be deduced by analogy.



The storage relationship of pointer address parameters as shown in the figure below





Examples of other description pointer applications are as follows .

| Function                            | Send Instruction            | Effect   |
|-------------------------------------|-----------------------------|--|
| Change Data Display Position        | 5A A5 07 82 5001 0000 0000  | The coordinates of the upper left corner of the character display position become (0,0).   |
| Change ASCII Character Lattice Size | 5A A5 05 82 500A 30 60      | Change the character lattice to 48 * 96 . Note: The lattice values in the X and Y directions need to be modified.                        |
| Hidden Characters                   | 5A A5 05 82 5008 0000       | Change the character length to 0 to hide characters.   |
| Change Font File                    | 5A A5 07 82 5009 003C 10 10 | Call size 60 font, the font size is 16 *16. That is, after changing the font file, the character size will also change at the same time. |

SP setting problems:

(1) In case of no curve display, 0x1000-0x7FFF can be used for VP, while 0x8000-0xFFFF can be used for SP. The distinction is to avoid conflicts caused by pointer address confusion.

(2) In case of curve display, and all 8 channels are used, 0x1000- 0x4FFF will be used as the curve buffer address . Users should not use this part of the address as the address of other variables.

0x5000-0x7FFF can be used as the variable address range; 0x8000-0xFFFF can be used as the description pointer range. In the application, the user can plan the address range of VP and SP according to actual needs.

The division of VP and SP here is only for your reference.

(3) The same display controls SP can be set to 10H addresses, and different types of display controls can be set to 30H addresses. The distinction is to avoid display conflicts caused by pointer address confusion.

(4) If the SP of the same type of display controls is set to be the same, it can only be realized by copying to other pages. The font size and other attributes shall not be changed and the controls shall not be moved, otherwise It will cause abnormal display.

## Chapter 3 SD Card Interface

### 3.1 MicroSD(TF) Card Format

All hardware parameters and data of T5L screen can be set by the SD/TF card interface, and the configuration files must be in FAT32 format.

Non-DWIN-official SD cards need to be formatted under the DOS system. Otherwise, the number of downloaded files is 0 during the blue screen downloading, or the screen fails to recognize the card to enter the download interface.

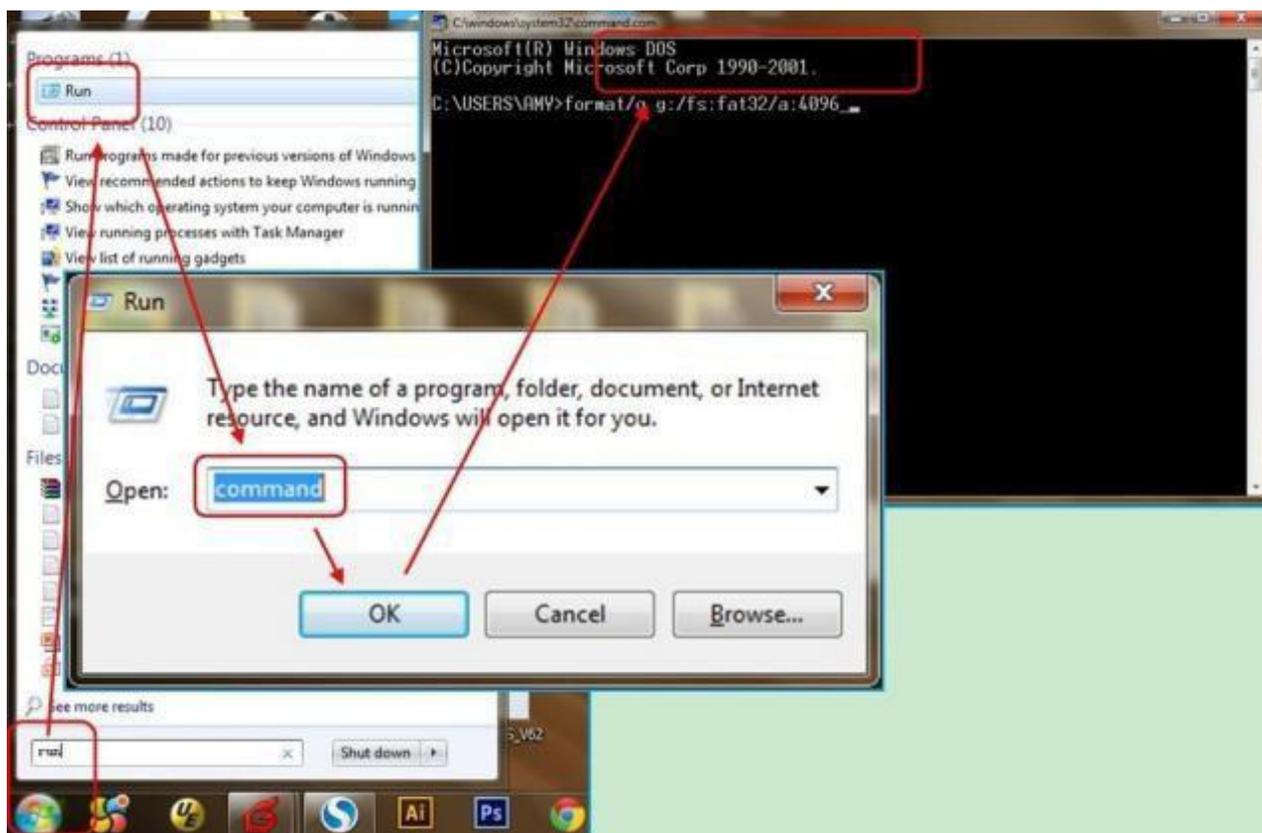
Format operations are as follows .

Step 1: Start-run-input “command” (“CMD” for win7) to enter DOS system;

Step 2: Input command: `format/q g:/fs:fat32/a:4096` (Note that q is followed by a space). Click enter after input.

Where g is the disk number of SD card displayed on PC, and the corresponding disk number of different users is different (for example, it can be “ h” or “ i” ).

Note: The SD card cannot be completely formatted to FAT32 by right clicking formatting. Generally, it is only for the SD card size of 1-16G.





### 3.2 SD Card Interface Download Method

SD card does not support hot swap updates. In order to prevent damage to FLASH, you must power off the screen before insert the SD card, and then power on to download. During the downloading, please be sure to keep the power supply. Power failure in the middle may cause a black screen abnormality.

T5L\_DGUSII is different from T5\_DGUSI. In practical applications, the background images of the T5L screen needs to be downloaded in ICL file, which can be generated by DGUS. JPEG pictures cannot be recognized directly.

The generation of download files is as follows .

- ① Create a DWIN\_SET folder in the SD card root directory.
- ② Put the ICL file, font library, configuration files, etc. to the DWIN\_SET folder.
- ③ When the screen is powered on, it will detect that the SD card and recognize whether there is a folder named DWIN\_SET in the SD card root directory.

If the folder is successfully detected, the screen will download the relevant files to FLASH. Folders under the DWIN\_SET folder cannot be read .

- ④ When the blue screen indicates that the download has completed, power off the screen, pull out SD card, and power on again. Then the project has been downloaded in the screen successfully.

DWIN standard screen, will only recognize folder named DWIN\_SET if not specially customized or with download encryption. Other folders can not be recognized.



### 3.3 Download File

#### 3.3.1 Download File Type

The SD/SDHC interface of T5L screen supports download and update of the following files.

**SD card does not support hot swap update. You must power off the screen before insert the SD card, and then power on to download .**

| File Type                               | Naming Rule   | Description   |
|---|---|---|
| Program file                            | T5L_UI*.BIN<br>T5L_OS*.BIN                                    | Underlying kernel firmware program  |
| DWIN OS program                         | DWINOS*.BIN   | DWIN OS program, the code must start from 0x1000.   |
| OS CPU 8051 program                     | T5L51*.BIN  | Program based on the standard 8051 platform. You can choose to encrypt or not to encrypt when downloading.  |
| Program configuration or initialization | T5L*.BIN  | LCD screen TCON configuration or initialization procedure.  |
| NOR Flash data base                     | ID+(optional) file name. LIB                                  | Each ID corresponds to 2 KWords memory with ID range of 0-79 . The database is located in on- chip NOR FLASH of 160 KWords. It can be used to save user data or DWIN OS program library files.  |
| Font file(4~ 12MB)                      | Font ID+(optional)file name. BIN/ DZK/HZK/GTF, font ID 00-47. | Font ID: 00-47; Pre-install standard ASCII 0# font library.   |
| DGUS input method file                  | 12*.BIN   | Fixed storage in 12 font location   |
| DGUS touch control file                 | 13*.BIN   | Fixed storage in 13 font location. Size limit: 256 KB   |
| DGUS display control file               | 14*.BIN   | Fixed storage in 14 font location. Size limit: 256 KB, in DGUS II format.   |
| DGUS variables initialization file      | 22*.BIN   | Fixed storage in 22 font location. Load 0x2000 - 0xFFFF address content and initialize 0x1000-0xFFFF variable space.  |
| JPEG image and icon file                | Font ID+ (optional) filename. ICL                             | Must be a ICL file in DGUSIII format. For multi FLASH expansion, one ICL data is saved in one Flash.<br><b>For product kernel of DGUS2 V4.0 and above, you must use DGUS software V7.623 and above to generate ICL files.</b>   |
| Music file                              | Font ID+ (optional) filename. WAE                             | Must be in DGUSII format, generated by DWIN tools.  |
| UI configuration file                   | Configuration module ID+ (optional) filename. UIC             | The configuration module ID is from 0x0000 to 0xFFFF, and each ID corresponds to 32 KB memory space. For multi FLASH expansion, one configuration module data is saved in one Flash.  |
| Hardware configuration file             | T5LCFG*.CFG   | Configure CRC status, baud rate modification, touch panel sound control, touch upload mode, display direction, etc.<br><b>Don' t download incompatible CFG files that may cause screen malfunction. Users can contact customer service to get the CFG files needed.</b> |
| CRC check file                          | T5*.CRC   | Perform CRC32 check after SD card download.   |

16 MB Flash memory ( max expandable to 64 Mbytes Nor Flash or 48 MB Nor + 512 MB NAND Flash) is divided into two parts:

- (1) 4-12MB font library space. Single space is 256Kbytes for font library, icon libraries, configuration files.
- (2) 4-12MB storage space for background images (.ICL files).

For T5L1 screens, the size of a single JPG file should not exceed 248 KB, and for T5L2 CPUS, the size of a single JPG



file should not exceed 760 KB.

When using NAND Flash expansion, it must be soldered in the position of expanded Flash 3. Corresponding font library ID=0xC0-0xFF, each of 8MB.

The download file must be placed in the SD card root directory DWIN\_SET folder. And the SD card must be a 4KB -sector, FAT32-formatted SD or SDHC card.

### 3.3.2 Generation of Download Files

From Chapter 2 , we know that when configuring GUI for DGUSII screens, we need to prepare material such as background images, icons, font libraries, etc.

DGUSII\_ T5L calls configuration files by the file name number. Therefore, the files should be named by Alabaster numbers in order.

For example: 32\_Background\_Image.ICL and 48\_Icon\_Library.ICL.

#### 3.3.2.1 ICL Background Image Library File

1. Background images should be in JPEG, BMP or PNG formats.
2. The image resolution should be the same as the screen in 24-bit color.
3. By default, image 0 in the ICL file will be displayed at booting. The images are named according to the Arabic numbers, e.g. 00\_boot, 01\_function, 02\_menu.
4. You need to convert images into .ICL file to download it to the screen. That is, the images are packaged and downloaded in the form of a library file.
5. Background image ICL files are used to display background images. If you name background image ICL as 32 .ICL, you need to write 0x20 in the 0x08 address of the CFG file so that the underlying programs can correctly read and identify the ICL file.  
If the CFG file is empty, the screen will be black (with a weak backlight). In this case, you should check the 0x08 address in the CFG file.
6. For T5L1 screen, the single image size in the ICL file should not exceed 256KB; For T5L2 screen, that should be less than 768 KB. There is no limit to the size of the original image materials.

When using DGUS to generate ICL files, if a image in the generated ICL file is larger than the limit, you should reduce the quality percentage of images according to the prompt.

You can also view size information of the ICL file by DGUS ICL generator. If scattered dots are displayed in background images, you can select the image type 4:4:4.

#### 3.3.2.2 ICL Icon Library File

1. The max resolution of images in the icon library should not exceed 1024\*1024, or DGUS will automatically compress them to 1024\*1024;
2. The images should be in JPEG, BMP or PNG formats in 24-bit color;
3. The images should be named by Alabaster numbers in order.

In the FLASH storage space, a single font space is 256 KB in size.



When a single icon library file exceeds 256 KB, it is necessary to strictly calculate the interval occupied by the icon library, otherwise it may cause a conflict in the position of the icon interval and cause display abnormalities.

### 3.3.2.3 Font File

DGUS support multiple international font codes:8-bit, ASCII, GBK, GB2312 and UNICODE.

DGUS screen has been pre-installed with ASCII code 0# font library at the factory, which contains all ASCII characters with dot matrix size of 4\*8 ~ 64\*128. Users can directly call the 0 # font library to realize the display of numbers, letters and symbols.

When you need to use other coded font libraries, you need to generate them through the font generator. DGUS support BIN, DZK, HZK font files.

### 3.3.2.4 22.Bin Initialization Variable File

The initial value of the variable address is 0 by default when it is powered on. If you need to set an initial value other than 0, you need to turn on the function of loading 22 file at power-on by configuring the CFG file 0x05.5 address.

Generally, 22.bin can be directly generated by DGUS, and the initial value is automatically written when the configuration files are generated. Users can also edit values in the 22 file by UltraEdit.

Note that DGUS variable address is in word, while in UltraEdit 22. bin is edited by byte . Therefore, when modifying the VP 0x2000 word initial value, you should modify 0x4000 byte address in UltraEdit.

When editing 22.bin, you need to copy it on the desktop. Before downloading, put them back to the DWIN\_ SET folder. Otherwise, modifications will overwritten when the project is saved and generated.

The screenshot shows a hex editor window titled "22\_Config.bin" with the following data:

|            |   |       |
|------------|---|-------|
| 00003fe0h: | 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 | ..... |
| 00003ff0h: | 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 | ..... |
| 00004000h: | C4 FA BA C3 31 32 41 42 00 00 00 00 00 00 00 00 | ..... |
| 00004010h: | 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 | ..... |
| 00004020h: | 00 64 00 63 00 00 00 00 00 00 00 00 00 00 00 00 | ..... |

Annotations in the image include:

- "Corresponding to 0x2000 variable address" pointing to the top of the hex editor.
- "Generates the corresponding ASCII code" pointing to the hex data.
- A red box at the bottom: "Variable address 0x2010 data variable display power-on display '100'; 22 file start address 0x4020 write data 0064".
- "Corresponding to 0x2010 variable address" pointing to the bottom of the hex editor.
- A blue box on the right: "The variable address 0x2000 text shows the power-on display 'Hello 12AB'; the starting address of the 22 file is 0x4000, and the corresponding text can be written here." with an arrow pointing to the hex data at 0x4000.

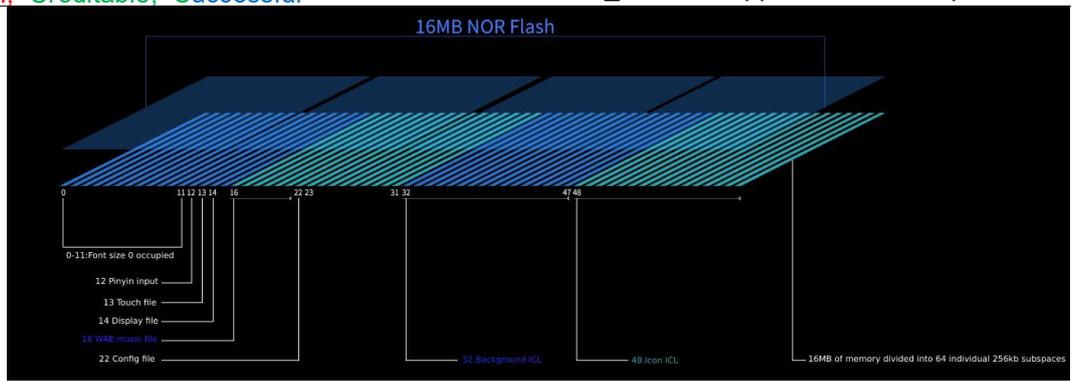
22 file initial value modification



3.3.2.5 System Configuration

T5LCFG\*. CFG hardware configuration file is in binary data format and can be edited by UltraEdit. The description is as follows :

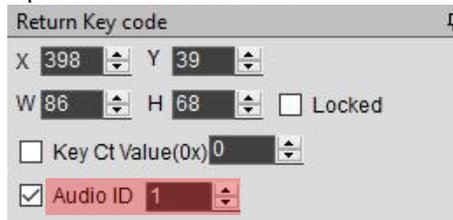
| Category   | Address | Length                  | Definition   | Description   |      |   |                         |                  |
|--|---------|-------------------------|--|---|------|---|-------------------------|------------------|
| Recognition configuration  | 0x00    | 5                       | 0x54 0x35 0x4C<br>0x43 0x31                                    | Fixed content.  |      |   |                         |                  |
| System configuration   | 0x05    | 1                       | Parameter configuration  | .7: UART CRC check 0=close 1=open   |      |   |                         |                  |
|  |         |                         |  | . 6: Buzzer / music playback selection.<br>0=Buzzer 1= music playback   |      |   |                         |                  |
|  |         |                         |  | . 5: Load 22 file initialization VP at power-on 1= Load 0= No Load  |      |   |                         |                  |
|  |         |                         |  | .4: Touch variable automatic upload 1= On, 0= Off   |      |   |                         |                  |
|  |         |                         |  | . 3:Touch panel audio 1= Open 0= Close  |      |   |                         |                  |
|  |         |                         |  | .2: Touch panel standby backlight 1= Open 0=Close   |      |   |                         |                  |
|  |         |                         |  | . 1-0: Display direction 00 = 0° , 01 = 90° , 10 =180° ,11 = 270°   |      |   |                         |                  |
|  | 0x06    | 1                       | Parameter configuration 1                                      | .7: PWM output 0= user control; 1= PMW backlight brightness control, 1 KHz frequency.   |      |   |                         |                  |
|  |         |                         |  | .6: Expand SPI NAND Flash. 0=not expanded 1=expanded.   |      |   |                         |                  |
|  |         |                         |  | . 5: Set 1 to start SPI NAND Flash format (data will be cleared).   |      |   |                         |                  |
| 0x07   | 1       | Music WAE file location | . 4: SPI NAND Flash memory. 0=1Gbits; 1=4 Gbits.               |   |      |   |                         |                  |
|  |         |                         | . 3: Set to 1 to erase all off-chip Flash contents.            |   |      |   |                         |                  |
|  |         |                         | . 2: AD return value resolution selection. 0=12 bit; 1=16 bit. |   |      |   |                         |                  |
|  |         |                         |  | . 1: OS CPU user 8051 program download encryption. 0=unencrypted; 1=encrypted.<br>(In case of encryption, users should use DWIN tool to encrypt before downloading the 8051 program.) |      |   |                         |                  |
|  |         |                         |  | .0: OS core running selection during SD download, 0=not allowed, 1=running.   |      |   |                         |                  |
| <p>Example: press button to music playback</p> <p>1. Content understanding</p> <p>(1) The relationship between .wae and .wav files</p> <p>The voice ID of the touch button refers to calling the .wav voice entry in the .wae voice package, and the .wae voice package contains N .wav voice entries.</p> <p><b>.wae voice package naming range: 0-63.</b></p> <table border="1"> <tr> <td>0x07</td> <td>1</td> <td>Music WAE file location</td> <td>0x00-0x3F(00-63)</td> </tr> </table> <p>Naming range of .wav voice: 1-63, 0 means no touch voice and does not support playback. The touch voice ID is only up to 0x3F, that is, 63, and the setting of a larger value is invalid, and it will jump back to 0x3F. (When using the 0xA0 command to play, the range of .wav is 0-255).</p> <p>(2) .wae file size calculation</p> <p>The .wae voice package occupies 16MB, and the calculation method of Nor Flash is the same as other font files.</p> <p>"x" file size*1024/256=N (occupies several 256KB subspaces), such as "x" file 1.23MB, 1.28*1024/256=5.12, rounded up, 0.12 also occupies a part of 256KB, need to count one , that is, the "x" file occupies 6 spaces, and 16, 32, and 48 custom files are calculated according to this.</p> <p>For example, the number 16 music file is 1.23MB, that is, the file occupies 6 spaces of 16, 17, 18, 19, 20, and 21. At this time, if you put a Chinese character font and name it 17, the name is wrong, and you need to put it in another location according to the above calculation.</p> <p>Adjacent .wav voice materials only need to have different names and IDs, and the program will index the .wav serial number to play, such as one voice 1S, one voice 3S, two voices can be named consecutively, and the longest voice duration of one voice is 1024S.</p> |         |                         |  |   | 0x07 | 1 | Music WAE file location | 0x00-0x3F(00-63) |
| 0x07   | 1       | Music WAE file location | 0x00-0x3F(00-63)   |   |      |   |                         |                  |



## 2. .wae voice pack generation

## 3. Touch ID Settings

The range is 1-63, and the software input box ID is hexadecimal.



## 4. CFG file hardware configuration

(1) Turn on the hardware function:

CFG file 0x05 address write 0x78, 0x05 address 8 bits 0111 1000, 0x05.6 bit write 1 to start music playback.

(2) Specify the .wae voice package:

Write 0x10 in CFG file 0x07 address, specify the 16th .wae voice package, and the attached DEMO configuration can be used directly.

```

00000000h: 54 35 4C 43 31 78 00 10 20 28 00 1C 64 32 00 64 ; T5LClx.. (.d2.d
00000010h: 5A A5 00 08 04 0C 01 E0 08 04 13 03 20 08 0E 00 ; Z?....?....
00000020h: 5A 16 FF 00 00 00 FE 00 6E 0B B8 00 F0 0A ; Z. ...?n??
  
```

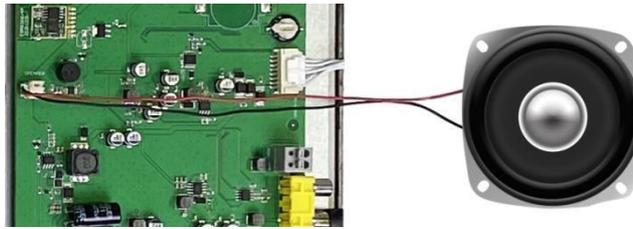
## 5. File download

Put "16.WAE" and "T5LCFG.cfg" configuration files, 13, 14, 22, (32\_background picture), (48\_icon) and other files into the DWIN\_SET folder, download them to the screen through the SD card, click the touch ready to play.

## 6. speaker

(1) With speaker interface

The power of the speaker is 3-5W, the impedance is 8Ω, the 2Pin seat is connected to the speaker interface, regardless of the positive and negative poles



(2) Without speaker interface

Write 0x78 in the CFG file 0x05 address to start music playback, and the buzzer can also play music. Because the circuit design of the speaker and the buzzer are different, it will cause the buzzer to heat up. This method can be used as a test, and it is not recommended for long-term use.

7. 0xA0 command play

Voice file making and downloading is the same as above.

5AA5 07 82 00A0 0001 4000

Meaning: 0x5AA5 frame header; 0x07 data length; 0x82 write command;

0x00A0 system variable interface playback voice address;

0x00 Music segment naming serial number position, range 0x00-0xFF, ID number of .wav file;

0x01 is fixed to 0x01, the whole piece of music is played;

0x40 is the volume, the unit is 1/64.

0x00 is the playback progress, you can write 0x00, read-only.

Download 480\*480DEMO to the screen.

Play the first .wav voice command: 5AA5 07 82 00A0 0001 4000

Play the first .wav voice command: 5AA5 07 82 00A0 0101 4000

Play the second .wav voice command: 5AA5 07 82 00A0 0201 4000

Play the 32nd .wav voice command: 5AA5 07 82 00A0 2001 4000

Play the 63rd .wav voice command: 5AA5 07 82 00A0 3F01 4000

|                                 |      |   |                                    |  |
|---------------------------------|------|---|------------------------------------|--|
|                                 | 0x08 | 1 | Background image ICL file location | 0 x10-0x3F( 16-63), correspond to 12MB-4MB background image size.                    |
|                                 | 0x09 | 1 | Touch panel reporting point rate   | 0x01-0xFF.<br>By default 0x28, reporting point rate=400 Hz/setvalue                  |
|                                 | 0x0A | 2 | UART baud rate                     | Baud rate =3225600/set baud rate (max of 0x03FF)<br>For 115200bps, set value=0x001C. |
| Backlight standby configuration | 0x0C | 1 | Normal and startup brightness      | 0x00-0x64, unit 1%   |
|                                 | 0x0D | 1 | Standby brightness                 | 0x00-0x64, unit 1%   |
|                                 | 0x0E | 2 | Wake up time after standby         | 0x0001-0xFFFF, unit 10 ms  |



|                   |      |   |                   |   |
|-------------------|------|---|-------------------|---|
| LCD configuration | 0x10 | 2 | Display_Config_En | 0x5AA5 = Enable the configuration.<br>It has been configured at the factory. Users do not need to configure it.   |
|                   | 0x12 | 1 | PCLK_PHS          | Data latch phase setting.<br>0x00=PCLK up 0x01=PCLK down  |
|                   | 0x13 | 1 | PCLK_DIV          | PCLK frequency(MHz)=206.4/PCLK_DIV  |
|                   | 0x14 | 1 | H_W               |   |
|                   | 0x15 | 1 | H_S               |   |
|                   | 0x16 | 2 | H_D               | Horizontal ( X axis) resolution   |
|                   | 0x18 | 1 | H_E               |   |
|                   | 0x19 | 1 | V_W               |   |
|                   | 0x1A | 1 | V_S               |   |
|                   | 0x1B | 2 | V_D               | Vertical ( Y axis) resolution   |
|                   | 0x1D | 1 | V_E               |   |
|                   | 0x1E | 1 | TCON_SEL          | 0x00= do not need to configure TCON   |
| TP configuration  | 0x1F |   | Reserved          | Write 0x00  |
|                   | 0x20 |   | TP_Set_En         | 0x5A= Enable the configuration. It has been configured at the factory. Users do not need to configure it again  |
|                   | 0x21 | 1 | TP_Mode           | <p>Touch panel mode configuration</p> <p><b>. 7-.4(high 4bit). Select touch panel type:</b></p> <p>0x0*= 4- wire resistive touch panel</p> <p>0x1*= GT911, GT9271 or GT9110 Driver IC capacitive touch panel</p> <p>0x2*= ILI9881H Incell CTP</p> <p>0x3*= ILI driver IC capacitive touch panel such as ILI2117</p> <p>0x4*= ILI driver IC capacitive touch panel such as ILI2130</p> <p>0x5*= Betterlife driver IC capacitive touch panel such as BL8825</p> <p>0x6 *= Sili micro driver IC capacitive touch screen such as GSL1680.</p> <p>0x7*= Hynitron driver IC capacitive touch screen such as CST340</p> <p>0x8*= ILI driver IC capacitive touch screen such as ILI231*/251*</p> <p>0xF*= 5- wire resistive touch panel</p> <p><b>. 3 Resistive touch panel calibration: 0= off; 1= on, enabled for SD card downloading</b></p> <p><b>.2-.0 ( low 3bit) Touch panel setting: (Only CTP works)</b></p> <p>.2 X coordinate selection: 0= from 0 to Xmax; 1= from Xmax to 0;</p> <p>. 1 Y coordinate selection: 0= from 0 to Ymax; 1=from Ymax to 0;</p> <p>.0 X, Y exchange: 0=XY; 1=YX</p> |



|   | 0x22                | 1                       | TP_Sense           | <p>CTP sensitivity setting:<br/>0x00-0 x1F. 0x00 lowest sensitivity; 0x1F highest sensitivity.<br/>Default is 0x14(ILI9881 is 0x01-0x06).<br/><b>0xFF = no touch ( using hardware initials).</b></p> <p>RTP sensitivity setting:<br/>0x10-0xFF, the smaller the value, the higher the precision but the sensitivity lower, the larger the value, the lower the precision and the higher the sensitivity, the factory configuration value is 0x40.</p>  |                     |                         |                         |                    |      |          |          |         |      |          |          |         |      |          |          |         |      |          |          |         |      |          |          |         |
|---|---------------------|-------------------------|--------------------|--|---------------------|-------------------------|-------------------------|--------------------|------|----------|----------|---------|------|----------|----------|---------|------|----------|----------|---------|------|----------|----------|---------|------|----------|----------|---------|
|   | 0x23                | 1                       | TP_Freq            | <p>TP frequency selection, suitable for ILI9881 H.<br/>Fixed frequency = 0x01 - 0x14 , and 0x00 represents frequency hopping.</p>  |                     |                         |                         |                    |      |          |          |         |      |          |          |         |      |          |          |         |      |          |          |         |      |          |          |         |
| Clock output configuration                    | 0x24                | 1                       | CKO_Set_En         | 0x5A= Enable the configuration.  |                     |                         |                         |                    |      |          |          |         |      |          |          |         |      |          |          |         |      |          |          |         |      |          |          |         |
|   | 0x25                | 1                       | CKO_En             | 0x5A= Enable the output function of CKO (P3.0)   |                     |                         |                         |                    |      |          |          |         |      |          |          |         |      |          |          |         |      |          |          |         |      |          |          |         |
|   | 0x26                | 1                       | CKO_DIV            | CKO output clock setting. Frequency = 825.7536 / CKO_DIV MHz.  |                     |                         |                         |                    |      |          |          |         |      |          |          |         |      |          |          |         |      |          |          |         |      |          |          |         |
| Buzzer configuration                          | 0x27                | 1                       | BUZZ_Set_En        | 0x5A= Enable the configuration.  |                     |                         |                         |                    |      |          |          |         |      |          |          |         |      |          |          |         |      |          |          |         |      |          |          |         |
|   | 0x28                | 1                       | BUZZ_Freq_DIV1     | Buzzer frequency= BUZZ_Freq_DIV1) KHz.825753.6<br>BUZZ_Freq_DIV1   |                     |                         |                         |                    |      |          |          |         |      |          |          |         |      |          |          |         |      |          |          |         |      |          |          |         |
|   | 0x29                | 2                       | BUZZ_Freq_DIV2     | Factory settings: DIV1=0x6E, DIV2=0x0BB8, correspond to the 2.5KHz frequency.  |                     |                         |                         |                    |      |          |          |         |      |          |          |         |      |          |          |         |      |          |          |         |      |          |          |         |
|   | 0x2B                | 2                       | BUZZ_Freq_Duty     | Buzzer duty ratio settings:<br>High level duty ratio=BUZZ_Freq_Duty/ BUZZ_Freq_DIV2. Factory settings: 0x00F0 corresponds to 8% high level duty ratio.   |                     |                         |                         |                    |      |          |          |         |      |          |          |         |      |          |          |         |      |          |          |         |      |          |          |         |
|   | 0x2D                | 1                       | BUZZ_Time          | Buzzing time after touch, unit 10 ms; Factory setting: 0x0A  |                     |                         |                         |                    |      |          |          |         |      |          |          |         |      |          |          |         |      |          |          |         |      |          |          |         |
| Power-on initialization file ID configuration | 0x2E                | 1                       | Init_File_Set_En   | 0x5A = Start configuring power- on initialization variable file ID.  |                     |                         |                         |                    |      |          |          |         |      |          |          |         |      |          |          |         |      |          |          |         |      |          |          |         |
|   | 0x2F                | 1                       | Init_File_ID       | The configured power- on initialization variable file ID, the factory defaults 0x16 .  |                     |                         |                         |                    |      |          |          |         |      |          |          |         |      |          |          |         |      |          |          |         |      |          |          |         |
| System clock adjustment                       | 0x30                | 1                       | Sysclk_set_EN      | 0x5A means this configuration is valid.  |                     |                         |                         |                    |      |          |          |         |      |          |          |         |      |          |          |         |      |          |          |         |      |          |          |         |
|   | 0x31                | 1                       | Ssclk_Adj_Set      | <p>System clock adjustment, -2 (0xFE) to +2 (0x02), the unit is 1/224.<br/>The factory configuration value is 0x00.<br/>Different configuration values correspond to PLL main frequency, CPU main frequency, and baud rate constants as follows:</p> <table border="1"> <thead> <tr> <th>Configuration Value</th> <th>PLL Main Frequency(MHz)</th> <th>CPU Main Frequency(MHz)</th> <th>Baud Rate Constant</th> </tr> </thead> <tbody> <tr> <td>0xFE</td> <td>818.3808</td> <td>204.5952</td> <td>6393600</td> </tr> <tr> <td>0xFF</td> <td>822.0672</td> <td>205.5168</td> <td>6422400</td> </tr> <tr> <td>0x00</td> <td>825.7536</td> <td>206.4384</td> <td>5451200</td> </tr> <tr> <td>0x01</td> <td>829.4400</td> <td>207.3600</td> <td>6480000</td> </tr> <tr> <td>0x02</td> <td>833.1364</td> <td>208.2816</td> <td>6508800</td> </tr> </tbody> </table> | Configuration Value | PLL Main Frequency(MHz) | CPU Main Frequency(MHz) | Baud Rate Constant | 0xFE | 818.3808 | 204.5952 | 6393600 | 0xFF | 822.0672 | 205.5168 | 6422400 | 0x00 | 825.7536 | 206.4384 | 5451200 | 0x01 | 829.4400 | 207.3600 | 6480000 | 0x02 | 833.1364 | 208.2816 | 6508800 |
|   | Configuration Value | PLL Main Frequency(MHz) |                    | CPU Main Frequency(MHz)  | Baud Rate Constant  |                         |                         |                    |      |          |          |         |      |          |          |         |      |          |          |         |      |          |          |         |      |          |          |         |
|   | 0xFE                | 818.3808                |                    | 204.5952   | 6393600             |                         |                         |                    |      |          |          |         |      |          |          |         |      |          |          |         |      |          |          |         |      |          |          |         |
|   | 0xFF                | 822.0672                |                    | 205.5168   | 6422400             |                         |                         |                    |      |          |          |         |      |          |          |         |      |          |          |         |      |          |          |         |      |          |          |         |
|   | 0x00                | 825.7536                |                    | 206.4384   | 5451200             |                         |                         |                    |      |          |          |         |      |          |          |         |      |          |          |         |      |          |          |         |      |          |          |         |
|   | 0x01                | 829.4400                |                    | 207.3600   | 6480000             |                         |                         |                    |      |          |          |         |      |          |          |         |      |          |          |         |      |          |          |         |      |          |          |         |
|   | 0x02                | 833.1364                |                    | 208.2816   | 6508800             |                         |                         |                    |      |          |          |         |      |          |          |         |      |          |          |         |      |          |          |         |      |          |          |         |
|   |                     |                         |                    |  |                     |                         |                         |                    |      |          |          |         |      |          |          |         |      |          |          |         |      |          |          |         |      |          |          |         |
|   |                     |                         |                    |  |                     |                         |                         |                    |      |          |          |         |      |          |          |         |      |          |          |         |      |          |          |         |      |          |          |         |
|   |                     |                         |                    |  |                     |                         |                         |                    |      |          |          |         |      |          |          |         |      |          |          |         |      |          |          |         |      |          |          |         |
|   |                     |                         |                    |  |                     |                         |                         |                    |      |          |          |         |      |          |          |         |      |          |          |         |      |          |          |         |      |          |          |         |
|   |                     |                         |                    |  |                     |                         |                         |                    |      |          |          |         |      |          |          |         |      |          |          |         |      |          |          |         |      |          |          |         |
|   |                     |                         |                    |  |                     |                         |                         |                    |      |          |          |         |      |          |          |         |      |          |          |         |      |          |          |         |      |          |          |         |
| Picture Replace Configuration                 | 0x32                | 1                       | Picture_Replace_En | <p><b>0x5A</b> means <b>enable the picture replacement</b>. If picture replacement is enabled, when the system executes the command related to image display, it will first check whether there is a corresponding picture in the ICL file where the replacement picture is stored, and if there is, it will display it, if not, it will display the corresponding picture in the ICL file specified in location 0x08 of the CFG file.</p>   |                     |                         |                         |                    |      |          |          |         |      |          |          |         |      |          |          |         |      |          |          |         |      |          |          |         |

|      |   |        |   |
|------|---|--------|---|
| 0x33 | 1 | ICL_ID | The ID of the ICL file that saves the replaced picture, 0x00 means no replacement (closed). |
|------|---|--------|---|

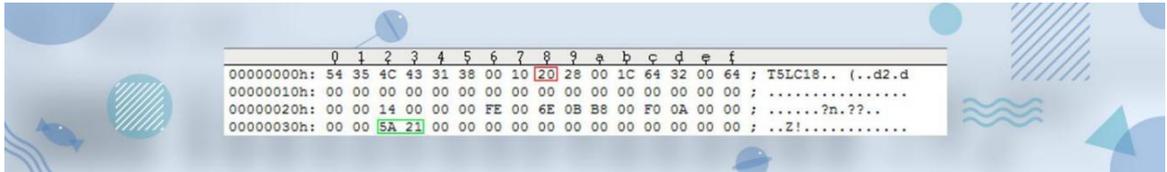
1. Application instructions

It is used to replace single or multiple pictures, which is convenient for users to customize the application.

2. Function setting

According to the function description of 0x32 and 0x33, the settings are as follows, write 0x5A to 0x32 to enable, write 0x21 to specify the location at 0x33, and specify the call through the CFG hardware parameter configuration file. Cancel replace 0x33 write 0x00.

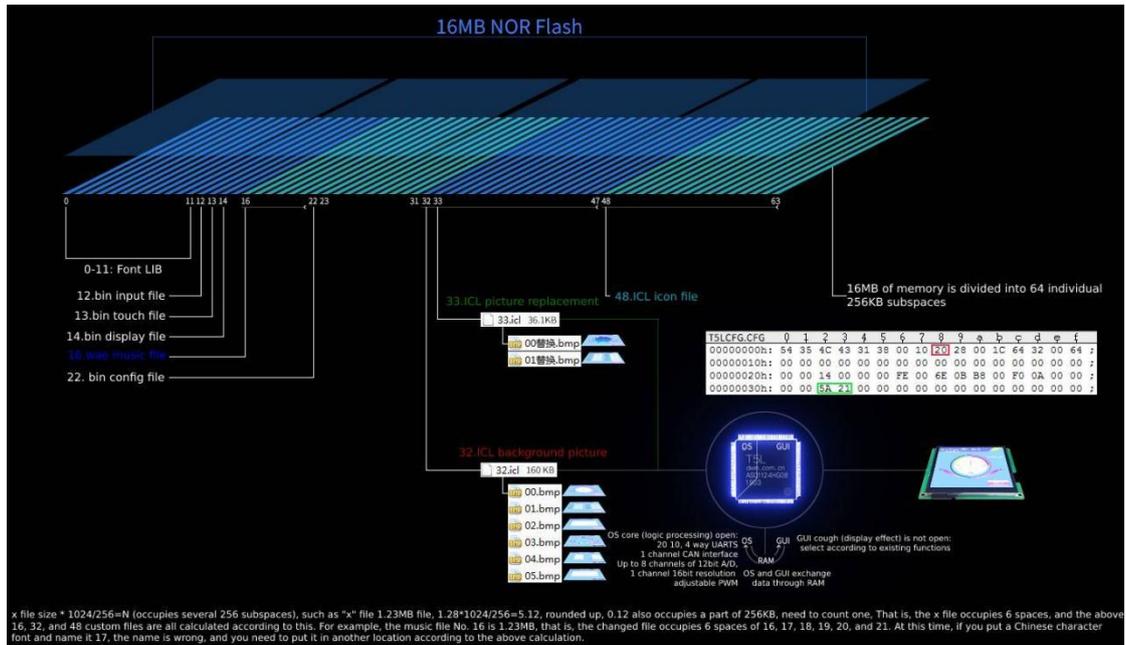
The background picture file of routine No. 32 is 160KB, and only occupies one 256KB Flash subspace. The spare replacement file is placed at 0x21. If the No. 32 file is 300KB, the spare replacement file needs to be moved back.



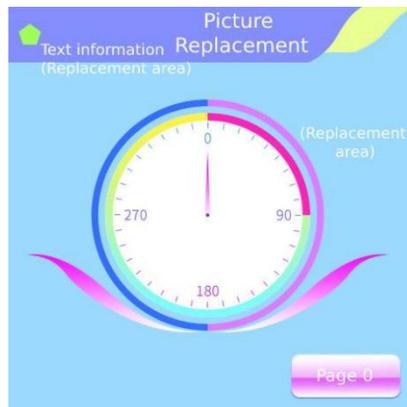
3. Backup picture

(1) The ID of the backup picture needs to be the same as that of the original picture. For example, if the ID of the original image is 00, the backup picture can be named "00" or "00\_Replacement".

(2) Replacement means that the buffer loads the ID data of Flash file No. 33 to the LCD display. The data calling position is different, and the data of No. 32 file is not covered by No. 33 file.



4. Display effect



Original



Replace



|   |      |    |  |  |
|---|------|----|--|--|
| Reserved  | 0x30 | 16 | Reserved                                 | Write 0x00   |
| SD download file folder name setting            | 0x40 | 2  | SD_Set_En                                | 0x5AA5 means setting the name of SD download folder once, which are saved in the screen Flash. It will not be lost after power off.  |
|   | 0x42 | 1  | Character length of download folder name | 0x01-0x08  |
|   | 0x43 | 8  | Folder name                              | Up to 8 ASCII characters (only 0-9, a-z, A-Z, -, _ ). Invalid character setting will use " DWIN_SET" as the default value, which is effective after power off and restart. |
|   | 0x4B | 37 | Reserved                                 | Write 0x00.  |
| Power-on backlight delay lighting configuration | 0x70 | 1  | Configuration enable                     | 0x5A Start a power-on backlight delay lighting time configuration.   |
|   | 0x71 | 1  | Delay enabled                            | Power-on backlight delay start time, the unit is 10ms, the factory configuration is 0x0A.  |
| Undefined                                       | 0x72 | 14 | Reserved                                 | 0x00   |

**Attention: Users can configure the parameters in the green background of the CFG file as needed, but if you feel uncertain or need to modify the parameters in the white background, never hesitate to contact customer service. We are always willing to offer assistance.**

(1) CFG file description

① Naming rule: T5LCFG.CFG

② Meaning: T5LCFG is the identification mark, these 6 characters cannot be modified, .CFG is the suffix indicating the file format.

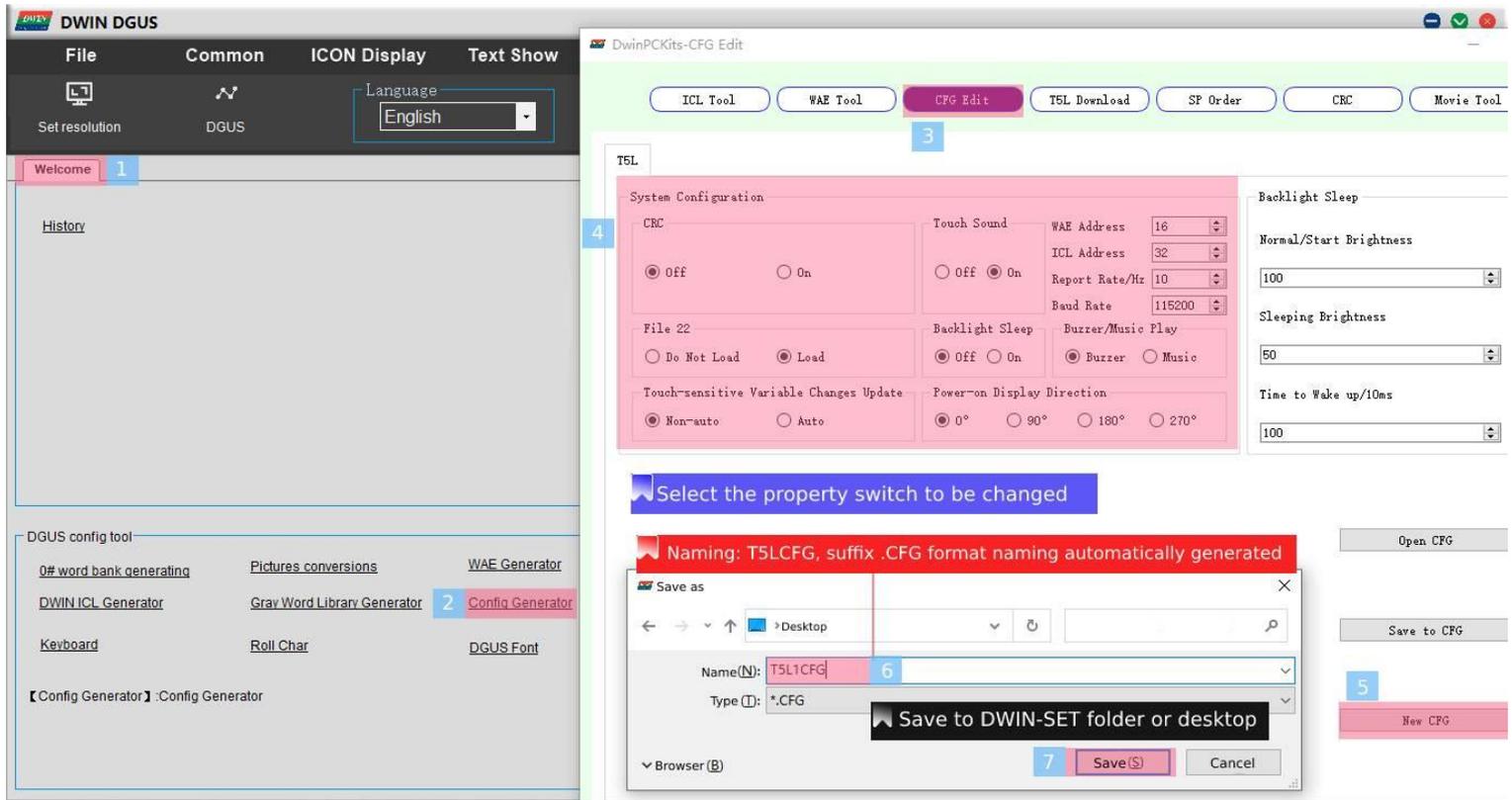
③ Common naming: T5LCFG landscape.CFG; T5LCFG vertical.CFG; T5LCFG\_800480.CFG is the correct naming. T5L.CFG is the wrong naming.

④ Rationale: 1 word (word) = 2 bytes (byte), 1 byte (byte) = 8 bits (bit). 1 bit bit corresponds to a function, and the function has 2 states: off and on.

⑤ Byte configuration:

| Function   | On | OFF | Such as the result of choosing | Convert binary to hexadecimal   |
|------------|----|-----|--------------------------------|---|
| Function 7 | 1  | 0   | 0                              | 0011 1000 binary conversion is 0x38 in hexadecimal, this value is the parameter configuration written to the corresponding address, and the file will take effect after downloading |
| Function 6 | 1  | 0   | 0                              |   |
| Function 5 | 1  | 0   | 1                              |   |
| Function 4 | 1  | 0   | 1                              |   |
| Function 3 | 1  | 0   | 1                              |   |
| Function 2 | 1  | 0   | 0                              |   |
| Function 1 | 1  | 0   | 0                              |   |
| Function 0 | 1  | 0   | 0                              |   |

⑥ Software generation: CFG files can be modified manually or generated by software, DGUS software generation is done in the following 7 steps.



⑦CFG configuration diagram: the first row is the common configuration, the second row is the LCD configuration, the third row is the touch configuration, etc. The first line 0x05 address is the most commonly used address.

7: Serial port CRC check selection 0=off 1=on.  
 6: Buzzer/music play selection, 0=buzzer 1=music play  
 5: Power on and load 22 files to initialize variable space 1=load 0=do not load  
 4: Touch variable to change automatic upload control 0=Do not upload automatically 1=Upload automatically  
 3: Touch screen sound control, 0=off 1=on  
 2: Touch screen backlight standby control 1=on 0=off  
 1-0: Power-on display direction 00=0 01=90 10=180 11=270

The first 5 bytes are fixed

0x07 Music WAE file save location, 0x00-0x3F (00-63)

0x08 Background image ICL file save location, 0x10-0x3F (16-63)

0x09 Touch screen reporting rate setting, range 0x01-0xFF, factory default 0x28

Serial port baud rate setting, setting value=3225600/set baud rate

Backlight standby setting: 0x05 with 0x39; 0x0C: lighting brightness; 0x0D: Standby brightness; 0x0E 0x0F: Standby duration

| Address    | 0  | 1  | 2  | 3  | 4  | 5  | 6  | 7  | 8  | 9  | a  | b  | c  | d  | e  | f  | Description         |
|------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|---------------------|
| 00000000h: | 54 | 35 | 4C | 43 | 31 | 38 | 00 | 10 | 20 | 28 | 00 | 1C | 64 | 32 | 03 | E8 | ; T5LC18.. (...d2.? |
| 00000010h: | 5A | A5 | 00 | 03 | 70 | B4 | 02 | 00 | B4 | 02 | 14 | 02 | 00 | DC | 0A | 00 | ; Z?.p?写...總..      |
| 00000020h: | 5A | 20 | 06 | 00 | 00 | 00 | FE | 5A | 6E | 0B | B8 | 00 | F0 | 0A |    |    | ; .....喚n.??        |

Writing 5A to 0x20 is a switch, indicating that the 0x21/0x22 configuration takes effect, and writing 00 does not take effect.

0x21 Display Mode Configuration

0x22 touch screen sensitivity setting. 0x00 is the lowest, 0x1F is the highest. The factory default value is 0x14, and the sensitivity is high.

0x10-0x1F display mode configuration, the factory has already set the general discordant configuration.

For example, CFG file 0x05 parameter configuration:

①For example: set the initial value at power on, automatic upload of touch data; turn on the touch panel audio, turn off the backlight, and configure the display direction at 0°.

| Bit | Parameter         | Value | Description                  |
|-----|-------------------|-------|------------------------------|
| 7   | CRC check         | 0     | 0= close 1=open              |
| 6   | Buzzer            | 0     | 0= buzzer 1= musicplayback   |
| 5   | Initial value     | 1     | 0= close 1=open              |
| 4   | Data upload       | 1     | 0= close 1=open              |
| 3   | Sound             | 1     | 0= close 1=open              |
| 2   | Backlight         | 0     | 0= close 1=open              |
| 1   | Display direction | 0     | 00=0° 01=90° 10=180° 11=270° |
| 0   |                   | 0     |                              |

1 byte corresponds to 8 bits, and each bit corresponds to two states: on and off. Write 1 to set the corresponding bit on and 0 to off.

Convert the binary 00111000 to hexadecimal 0x38, write 0x38 to 0x05 address, and download the CFG file to complete the corresponding configuration.

```

x T5LCFG_tm041_CTP.CFG
  0 1 2 3 4 5 6 7 8 9 a b c d e f
00000000h: 54 35 4C 43 31 38 00 10 20 28 00 1C 64 32 03 E8 ; T5LC18.. (...d2.?
00000010h: 5A A5 00 03 70 B4 02 00 B4 02 14 02 00 0C 0A 00 ; Z?.p?.?.....
00000020h: 5A 20 06 00 00 00 00 00 00 00 00 01 00 00 00 00 ; Z .....
  
```

② Enable CRC check

CFG file 0x05 address can write 0xB8, 8 bits are configured as 1011 1000, 0x05 position. 7 bits are written 1.

Take reading the version number as an example:

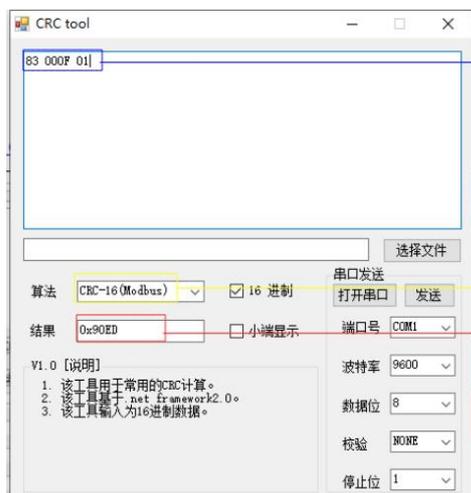
No check Tx (send): 5A A5 04 83 000F 01

Add check Tx (send): 5A A5 06 83 000F 01 ED90

Meaning: 0x5A A5 06: The frame header and data length do not need to participate in the verification;

0x83 000F 01: The part of the instruction that needs to participate in the verification;

0xED90: check value, check format CRC-16



Manual input command, 82/83 write/read command + data part

Set reference

The high and low bits of some verification tools are reversed. Here, when writing instructions, write the verification value to 0xED90

It's same for instructions, taking controlling the brightness of the backlight as an example:

No checksum Tx (send): 5A A5 04 82 0082 0A

Add check Tx (send): 5A A5 06 82 0082 0A C8FB

Meaning: 0x5A A5 06: The frame header and data length do not need to participate in the verification;

0x82 0082 0A: The instruction part that needs to participate in the verification; the first 0x82 is the write instruction, and the second 0x0082 is the backlight control address, do not confuse it.

Backlight brightness range 0x00-0x64.

0xC8FB: check value, check format CRC-16.

Setting value of DMG85480C050-03W (Vertical screen at factory) in CFG file at different display direction.

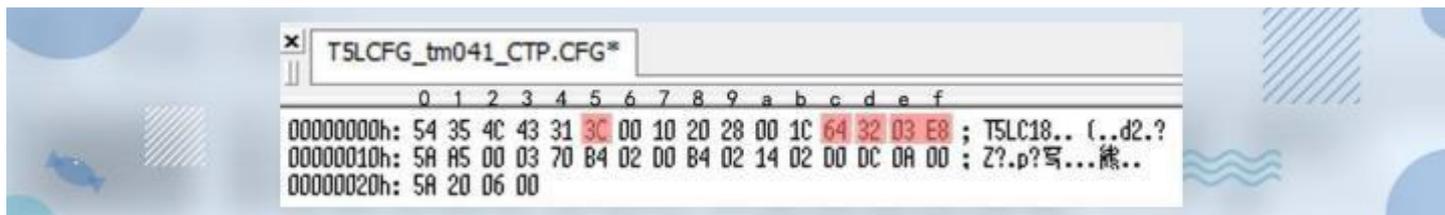
| Mode          | Image Resolution | ICL                             | Configuration Value |
|---------------|------------------|---------------------------------|---------------------|
| 0° lateral    | 854*480          | ICL file made by 854*480 image  | 0x38                |
| 90° vertical  | 480*854          | ICL file made by 480*854 images | 0x39                |
| 180° lateral  | 854*480          | ICL file made by 854*480 images | 0x3A                |
| 270° vertical | 480*854          | ICL file made by 480*854 images | 0x3B                |



(2) CFG 0x0C backlight screensaver configuration

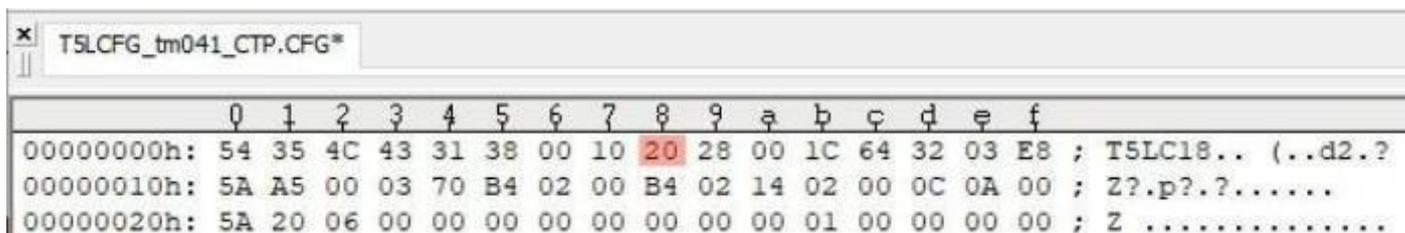
| CFG Address | Description              | Configuration Value               |
|-------------|--------------------------|-----------------------------------|
| 0x05        | Turn on backlight        | 0x3C                              |
| 0x0C        | Brightness value         | Range: 0x00-0x64                  |
| 0x0D        | Standby brightness value | Range: 0x00-0x64                  |
| 0x0E 0x0F   | Backlight delay          | Range: 0x0001-0xFFFF, unit 10 ms. |

After the backlight standby screen protection is turned on, the first touch on the touch panel will wake up the backlight, and the second touch will trigger the touch control.



(3) Background image location

If the background image file is set to 32. ICL, 0x20 needs to be written at CFG 0x08.



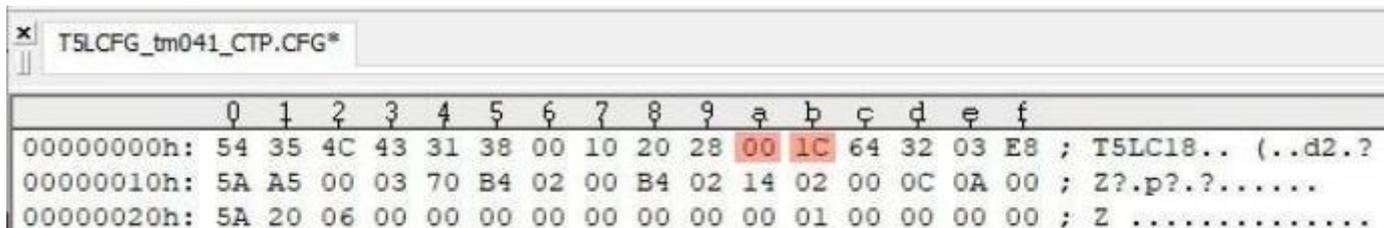
(4) Baud Rate

Baud rate setting value=3225600/baud rate. The maximum value is 0x03FF.

E.g. factory default=115200, baud rate setting value=3225600/ 115200=28=0x1C.

Because the baud rate accounts for 0x0A 0x0B 2 bytes, CFG file 0x0A address should be started as 0x001C.

Accordingly, for 9600 baud rate, the settings value is 0x0150.





(5) Display

Display configuration mode and touch configuration mode have been configured at factory. In case of reverse touch/ page sliding or white screen or screen flicker during configuration, you can try to download the factory CFG file.

Write 0x5A at 0x10, 0x11. A5 indicates that 0x12-0x1F parameters will take effect, and 0x00 indicate that they will not take effect.

```

x T5LCFG_tm041_CTP.CFG*
||
  0 1 2 3 4 5 6 7 8 9 a b c d e f
00000000h: 54 35 4C 43 31 38 00 10 20 28 00 1C 64 32 03 E8 ; T5LC18.. (..d2.?
00000010h: 5A A5 00 03 70 B4 02 00 B4 02 14 02 00 0C 0A 00 ; Z?.p?.?.....
00000020h: 5A 20 06 00 00 00 00 00 00 00 00 01 00 00 00 00 ; Z .....

```

0x12-0x1F of CFG file

(6) CTP Sensitivity Setting

Only for capacitive screens to increase the sensitivity of the PMMA or tempered glass modification, 0x20 write 0x5A, that the configuration of the 0x22 position takes effect, 0x22 with the sensitivity value, the factory default is 0x14, the range 0x0000-0x1F, generally the greater the thickness of the PMMA or tempered glass, the corresponding modified value is also greater.

```

x T5LCFG_tm041_CTP.CFG*
||
  0 1 2 3 4 5 6 7 8 9 a b c d e f
00000000h: 54 35 4C 43 31 38 00 10 20 28 00 1C 64 32 03 E8 ; T5LC18.. (..d2.?
00000010h: 5A A5 00 03 70 B4 02 00 B4 02 14 02 00 0C 0A 00 ; Z?.p?.?.....
00000020h: 5A 20 06 00 00 00 FE 5A 6E 0B B8 00 F0 0A 00 00 ; Z .... 06n.??..

```

(7) RTP Sensitivity Setting

The RTP sensitivity setting needs to be upgraded to V60 and above, and the touch screen needs to be calibrated once. If it has been upgraded, this step can be skipped.

Minimal sensitivity: 0x20 writes 0x5A, 0x21 writes 0x00, 0x22 writes 0x10; the touch feels relatively dull and a little scratchy.

Maximum sensitivity: 0x20 writes 0x5A, 0x21 writes 0x00, 0x22 writes 0xFF; the feeling of touch is relatively fast and gravity-free.

The feeling is only noticeable when using sliding controls to make touch control more obvious, such as dragging adjustment, sliding adjustment, sliding gesture to turn the page. The effect of button controls such as basic touch is not obvious.

The factory value is 0x40, which is the recommended default setting, but can also be set according to actual needs.

```

  0 1 2 3 4 5 6 7 8 9 a b c d e f
00000000h: 54 35 4C 43 31 38 00 00 20 28 00 08 64 64 FF FF ; T5LC18.. (..dd
00000010h: 00 00 01 16 29 02 01 E0 02 0A 02 01 10 02 00 00 ; .....)?.....
00000020h: 5A 00 40 00 00 00 00 00 00 00 00 00 00 00 00 00 ; Z.θ.....

```

### (8) Buzzer

Buzzer can be configured in the CFG file to adjust and modify the buzzer sound volume and duration.

0x2B in the CFG file determines the sound volume of the buzzer. A larger duty cycle means a louder sound.

0x2D in the CFG file determines touch button buzzer time, unit: 10mS. Factory default is 0x0A.

After configuration, the key accompaniment sound of the touch screen and the buzzer sound controlled by the system variable interface buzzer serial port command will change accordingly.

Attachment: Description of system variable interface 0xA0:

On-chip Flash saves WAE music playback settings:

D7: The segment ID of this playback, 0x00-0xFF; the location of the WAE file is specified by the CFG configuration file or D2:D1.

D6: The number of playing segments this time, fixed at 0x01, and cleared to zero after DGUS processing; in buzzer mode, it is the beeping time, and the unit is 8ms.

D5: playback volume, the unit is 1/64; the initial value after power-on is 0x40(100%).

D4: Play status feedback, 0x00=stop, 0x01=pause, 0x02=playing.

D3: 0x5A indicates that the WAE file is specified by D2:D1, otherwise it is specified by the CFG configuration file.

D2:D1: WAE file location, only valid when D3=0x5A.

D0: Undefined, write 0x00.

Command example:

5A A5 05 82 00A0 007D; The buzzer sounds for 1 second, 1000ms/8ms=125=007Dh.

5A A5 05 82 00A0 003E; The buzzer sounds for 0.5 seconds, 500ms/8ms=62.5=003Eh.

```

x T5LCFG_tm041_CTP.CFG*
0 1 2 3 4 5 6 7 8 9 a b c d e f
00000000h: 54 35 4C 43 31 38 00 10 20 28 00 1C 64 32 03 E8 ; T5LC18.. (...d2.?
00000010h: 5A A5 00 03 70 B4 02 00 B4 02 14 02 00 0C 0A 00 ; Z?.p?.?.?.....
00000020h: 5A 20 06 00 00 00 FE 5A 6E 0B B9 00 F0 0A 00 00 ; Z .... 唎n.??..
    
```

### (9) Power on Initialization variable file ID

0x5A at CFG 0x2E indicates that the function is enabled.

0x2F at CFG 0x0F indicates that the ID of the initialization file is 15, instead of 22.

This feature is useful when you need to make full use of the font space.

```

x T5LCFG_tm041_CTP.CFG*
0 1 2 3 4 5 6 7 8 9 a b c d e f
00000000h: 54 35 4C 43 31 38 00 10 20 28 00 1C 64 32 03 E8 ; T5LC18.. (...d2.?
00000010h: 5A A5 00 03 70 B4 02 00 B4 02 14 02 00 0C 0A 00 ; Z?.p?写...總..
00000020h: 5A 20 06 00 00 00 00 00 00 00 00 00 00 00 00 00 ; Z .... 唎n.??..
    
```

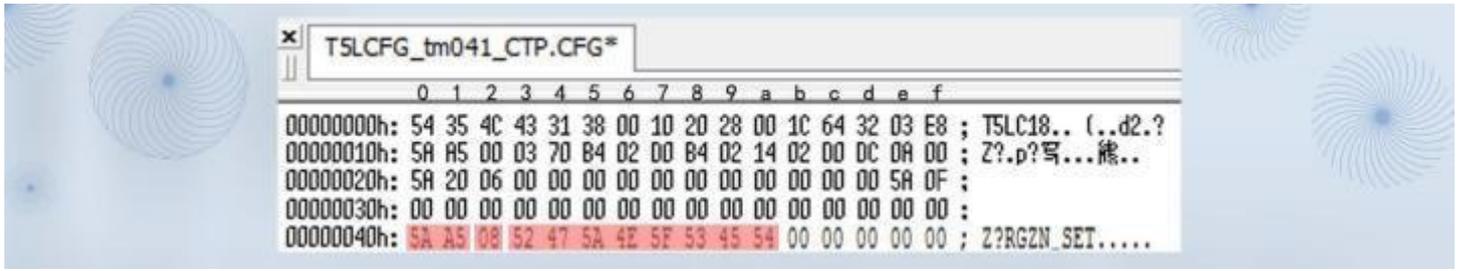
### (10) SD download folder name

0x5AA5 at CFG 0x40 and 0x41 enables file name rewriting.

0x08 at CFG 0x42 indicates that the folder ID overwritten is 8 bytes in length and ranges from 0x01 to 0x08.

0x4a at CFG 0x43 indicates the overwritten folder new name, such as RGZN\_SET.

After the configuration file is downloaded, it takes effect after re-power.



(11) T5L DGUSII touch screen calibration

T5L screen with RTP are designed with automatic error identification and dynamic calibration algorithm, which can solve the problem of touch offset.

When products are in a strong interference environment, touch drift may occur occasionally. In this case, you can use the following methods to restore the touch screen to normal functions.

The product calibration methods of T5L TA and DGUS are the same. During the switch between TA instruction set development mode and DGUS development mode, touch drift may occur occasionally. At this time, it can also be calibrated as follows .

|                  |      |   |           |   |
|------------------|------|---|-----------|---|
| TP configuration | 0x1F |   | Reserved  | Write 0x00  |
|                  | 0x20 |   | TP_Set_En | 0 x5 A= Enable the configuration. It has been configured at the factory. Users do not need to configure it again  |
|                  | 0x21 | 1 | TP_Mode   | <p>Touch panel mode configuration</p> <p>. 7-. 4(<b>high 4bit</b>). <b>Select touch panel type:</b></p> <p>0x0*= 4- wire resistive touch panel</p> <p>0x1*= GT911, GT9271 or GT9110 Driver IC <u>capacitive</u> touch panel</p> <p>0x2*= ILI9881H <u>Incell</u> CTP</p> <p>0 x3*= ILI driver IC <u>capacitive</u> touch panel such as ILI2117</p> <p>0 x4*= ILI driver IC <u>capacitive</u> touch panel such as ILI2130</p> <p>0 x5*= <u>Betterlife</u> driver IC <u>capacitive</u> touch panel such as BL8825</p> <p>0 x6 *= <u>Silij</u> micro driver IC <u>capacitive</u> touch screen such as GSL1680.</p> <p>0 x7*= <u>Hynitron</u> driver IC <u>capacitive</u> touch screen such as CST340</p> <p>0x8* = ILI driver IC <u>capacitive</u> touch screen such as ILI231*/251*</p> <p>0xF*= 5- wire resistive touch panel</p> <p>. 3 <b>Resistive touch panel calibration: 0= off; 1= on, enabled for SD card downloading</b></p> <p>.2-.0 ( <b>low 3bit</b>) <b>Touch panel setting: (Only CTP works)</b></p> <p>. 2 X coordinate selection: 0= from 0 to Xmax; 1= from Xmax to 0;</p> <p>. 1 Y coordinate selection: 0= from 0 to Ymax; 1=from Ymax to 0;</p> <p>.0 X, Y exchange: 0=XY; 1=YX</p> |

Taking the DMG48320C035\_03WTR four-wire resistive screen as an example, the first upgrade of the V60 and above kernel version calibration, the steps are as follows, the V45-V55 version does not upgrade the calibration and skip the first step:

- ① Download the V60 kernel program first—and then download the .INI file (the screen will be black if you don't download this file for V60 and above, just download it once), and skip this step if you have upgraded V60.
- ② CFG file 0x21 writes 0x08.

|                              |                 |                 |
|------------------------------|-----------------|-----------------|
| 4-wire resistive touch panel | Calibration     | 0x21 write 0x08 |
|                              | Non calibration | 0x21 write 0x00 |
| 5-wire resistive touch panel | Calibration     | 0x21 write 0xF8 |
|                              | Non calibration | 0x21 write 0xF0 |



0x08 meaning: the upper touch screen configuration table .7-.4 (high 4bit) indicates the touch screen type, DMG48320C035\_03WTR is a 4-wire resistive screen, so the high 4bit corresponds to choose to write 0. In the same way, if the 5-wire resistive screen is 4 bits high, choose to write F. Writing 1 to bit3 in the lower 4 bits means turning on resistive screen touch calibration, and bit2/bit1/bit0 are only valid for capacitive screens, that is, writing 0/1 is OK, and the lower 4 bits of 4 bits binary 1000 is converted to hexadecimal 8. Therefore, writing 0x08 at address 0x21 is the standard value, and it is also possible to write a value within the range of 0x08-0x0F at the same time, as long as bit3 writes 1.

You can write 0x00-0x07 without calibration, the default is 0x00, as long as bit3 writes 0 lines.

If you want to calibrate 0x21 for 5-wire resistive screen, you can write 0xF8-0xFF, and you can not calibrate 0xF0-0xF7.

|            | 0  | 1  | 2  | 3  | 4  | 5  | 6  | 7  | 8  | 9  | a  | b  | c  | d  | e  | f                  |
|------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|--------------------|
| 00000000h: | 54 | 35 | 4C | 43 | 31 | 38 | 00 | 00 | 20 | 28 | 00 | 08 | 64 | 64 | FF | FF ; T5LC18.. (.dd |
| 00000010h: | 00 | 00 | 01 | 16 | 29 | 02 | 01 | E0 | 02 | 0A | 02 | 01 | 10 | 02 | 00 | 00 ; ....)?.....   |
| 00000020h: | 5A | 08 | 10 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 ; Z.....        |

③Download the configuration file, 5-point calibration on the screen.

Use the tip of the pen to apply a little pressure in turn (resistive touch screen pressure sensing, to ensure effective operation of the operating contacts), click the center of the "ten" cursor on the touch screen in turn, the order is "upper left corner, upper right corner, lower right corner, lower left corner, and middle". After power on, the calibration is complete.

(12) 0x21 capacitive touch screen configuration mode modification

Capacitive touch screens have no calibration, only touch mode configuration. Misconfiguration can result in no touch or inaccurate touch.

According to the touch screen configuration table above, there are many different types of capacitive touch screens, that is, the configuration parameters of different types of screens are different. If you download the configuration file by mistake and cause no touch or reverse touch, you can download the corresponding model configuration file from the link below to update.

Take the DMG48480C040\_03WTC capacitive screen as an example, writing 0x16 at address 0x21 will result in invalid touch reverse (download a wrong one), find the factory configuration file of the corresponding model and write 0x10 at position 0x21 to update (download a correct one). At the same time, the type of touch screen used can be known according to the high 4bit value.

|            | 0  | 1  | 2  | 3  | 4  | 5  | 6  | 7  | 8  | 9  | a  | b  | c  | d  | e  | f                    |
|------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----------------------|
| 00000000h: | 54 | 35 | 4C | 43 | 31 | 38 | 00 | 00 | 20 | 28 | 00 | 1C | 64 | 00 | 00 | 00 ; T5LC1(.. (.d... |
| 00000010h: | 5A | A5 | 00 | 0E | 08 | 08 | 01 | E0 | 08 | 02 | 0C | 01 | E0 | 06 | 08 | 00 ; Z?....?..?      |
| 00000020h: | 5A | 10 | 14 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 5A | 16 ; Z.....Z.        |



(13)0x71 Power-on backlight delay lighting configuration

|   |      |   |                      |  |
|---|------|---|----------------------|--|
| Power-on backlight delay lighting configuration | 0x70 | 1 | Configuration enable | 0x5A Start a power-on backlight delay lighting time configuration                        |
|   | 0x71 | 1 | Delay enabled        | Power-on backlight delay start time, the unit is 10ms, the factory configuration is 0x0A |

0x70 write 0x5A to enable the function,

Delay 0.1s to light up the LCD screen: 0x71 factory setting 0x0A, 0x0A=10, 10x10ms=100ms=0.1s; **recommended factory default value 0.1s.**

Delay 2.55S to light up the LCD screen: 0x71 is set to 0xFF, 0xFF=255, 255x10ms=2550ms=2.55s, 0x71 only occupies 1 byte, and the delay range is 0x00-0xFF.

|            | 0  | 1  | 2  | 3  | 4  | 5  | 6  | 7  | 8  | 9  | a  | b  | c  | d  | e  | f  |                   |
|------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|-------------------|
| 00000000h: | 54 | 35 | 4C | 43 | 31 | 28 | 00 | 10 | 20 | 28 | 00 | 1C | 64 | 32 | 00 | 64 | ; T5LC1(.. ..d2.d |
| 00000010h: | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | ; .....           |
| 00000020h: | 00 | 00 | 14 | 00 | 00 | 00 | FE | 00 | 6E | 0B | B8 | 00 | F0 | 0A | 00 | 00 | ; .....?n??..     |
| 00000030h: | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | ; .....           |
| 00000040h: | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | ; .....           |
| 00000050h: | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | ; .....           |
| 00000060h: | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | ; .....           |
| 00000070h: | 5A | FF | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | ; Z .....         |

(14)Display configuration reference

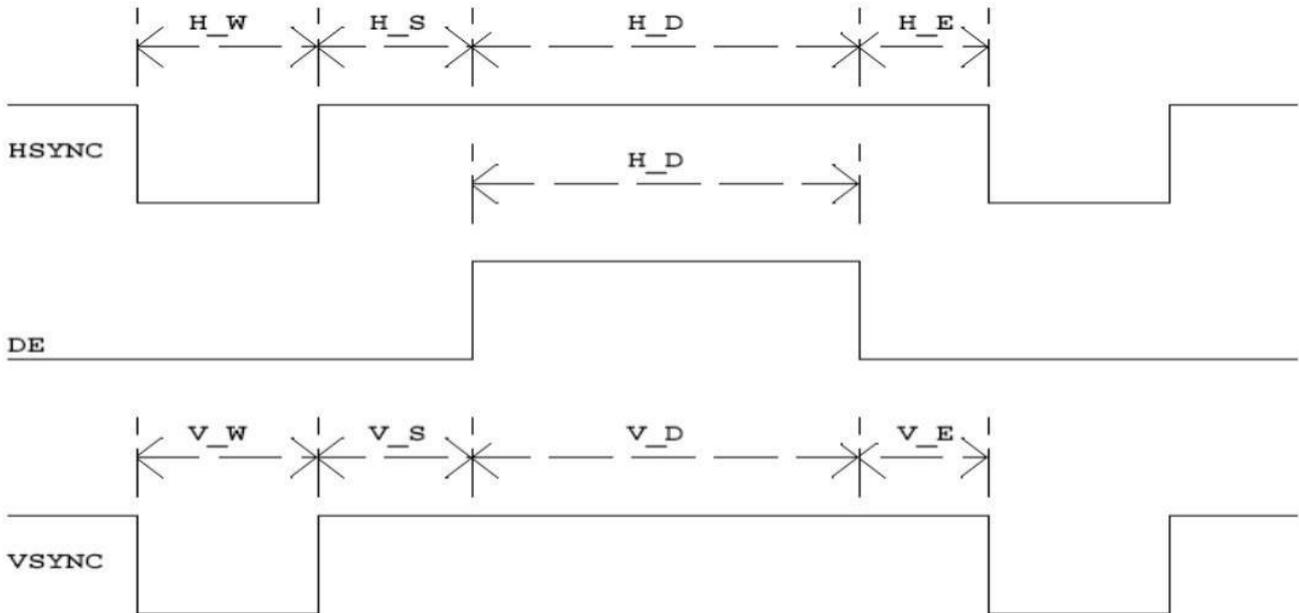




Table 3. 4 reference list of display configuration

| Size/ Resolution              | Display Screen Configuration Value (HEX Format) |      |      |      |      |      |      |      |      |      |      |      |      |      |
|-------------------------------|---|------|------|------|------|------|------|------|------|------|------|------|------|------|
|                               | 0x12  | 0x13 | 0x14 | 0x15 | 0x16 | 0x17 | 0x18 | 0x19 | 0x1A | 0x1B | 0x1C | 0x1D | 0x1E | 0x1F |
|                               | PHS   | PHS  | PHS  | PHS  | PHS  | PHS  | PHS  | PHS  | PHS  | PHS  | PHS  | PHS  | PHS  | PHS  |
| 2.0_240*320IPS                | 01  | 26   | 0A   | 14   | 00   | F0   | 0A   | 02   | 02   | 01   | 40   | 02   | 11   | 00   |
| 2.4_240*320IPS                | 01  | 26   | 0A   | 14   | 00   | F0   | 0A   | 02   | 02   | 01   | 40   | 02   | 0D   | 00   |
| 2.4_320*240                   | 01  | 26   | 0A   | 14   | 00   | F0   | 0A   | 02   | 02   | 01   | 40   | 02   | 05   | 00   |
| 2.8_240*320 (ST7789)          | 01  | 26   | 0A   | 14   | 00   | F0   | 0A   | 02   | 02   | 01   | 40   | 02   | 03   | 00   |
| 2.8_240*320B                  | 01  | 26   | 0A   | 14   | 00   | F0   | 0A   | 02   | 02   | 01   | 40   | 02   | 01   | 00   |
| 3.5_320*240                   | 01  | 26   | 0A   | 14   | 00   | F0   | 0A   | 02   | 02   | 01   | 40   | 02   | 02   | 00   |
| 3.5_320*480                   | 01  | 14   | 0A   | 04   | 01   | 40   | 0A   | 02   | 02   | 01   | E0   | 02   | 04   | 00   |
| 3.5_480*640 (ST7796S)         | 01  | 0A   | 10   | 20   | 01   | 40   | 20   | 02   | 03   | 02   | 80   | 02   | 06   | 00   |
| 3.5_320*480 IPS (IL9488)      | 01  | 14   | 0A   | 04   | 01   | 40   | 20   | 02   | 03   | 02   | 80   | 02   | 13   | 00   |
| 3.5_480*640                   | 01  | 0A   | 10   | 20   | 01   | E0   | 20   | 02   | 03   | 02   | 80   | 02   | 13   | 00   |
| 3.5_480*800 IPS(ST7701S)      | 00  | 08   | 04   | 0C   | 01   | E0   | 08   | 04   | 13   | 03   | 20   | 80   | 17   | 00   |
| 4.0_480*480 IPS(ST7701S)      | 00  | 0E   | 08   | 08   | 0E   | E0   | 08   | 02   | 0C   | 01   | E0   | 06   | 08   | 00   |
| 4.0_480*800 IPS(NT35512)      | 00  | 08   | 08   | 08   | 01   | E0   | 08   | 04   | 0A   | 03   | 20   | 0A   | 09   | 00   |
| 4.0_480*800 IPS(ST7701S)      | 00  | 08   | 08   | 08   | 01   | E0   | 08   | 04   | 0A   | 03   | 20   | 0A   | 16   | 00   |
| 4.0_720*720 (IPS Incell)      | 00  | 03   | 70   | BE   | 02   | D0   | B4   | 02   | 14   | 02   | D0   | DC   | 0A   | 00   |
| 5.0_720*1280(IPS Incell)      | 00  | 03   | 04   | 14   | 02   | D0   | 14   | 02   | 12   | 05   | 00   | C8   | 08   | 00   |
| 5.0_480*854 IPS (HSD ST7701S) | 00  | 08   | 08   | 08   | 01   | E0   | 08   | 02   | 0C   | 03   | 56   | 06   | 0C   | 00   |
| 5.0_480*854 IPS (BOE ST7701S) | 00  | 06   | 20   | 40   | 0E   | 10   | 40   | 02   | 06   | 03   | 56   | 08   | 14   | 00   |
| 4.3_480*800 IPS (9701)        | 00  | 08   | 04   | 0C   | 01   | E0   | 08   | 04   | 13   | 03   | 20   | 08   | 0E   | 00   |
| 4.3_480*800 IPS (ST7701S)     | 00  | 08   | 04   | 0C   | 01   | E0   | 08   | 04   | 13   | 03   | 20   | 0A   | 15   | 00   |
| 3.0_360*640 IPS               | 00  | 0A   | 20   | 3C   | 01   | 68   | 20   | 06   | 36   | 02   | 80   | 08   | 0F   | 00   |
| 480*272                       | 01  | 16   | 29   | 02   | 01   | E0   | 02   | 0A   | 02   | 01   | 10   | 02   | 00   | 00   |
| 640*480                       | 01  | 08   | 1E   | 72   | 02   | 58   | 10   | 03   | 20   | 01   | E0   | 0A   | 00   | 00   |
| 800*480                       | 01  | 06   | 1E   | 10   | 03   | 20   | D2   | 03   | 14   | 01   | E0   | 0C   | 00   | 00   |
| 800*600                       | 01  | 05   | 1E   | 10   | 03   | 20   | D2   | 03   | 14   | 02   | 58   | 0C   | 00   | 00   |
| 1024*600                      | 01  | 04   | A0   | 88   | 04   | 00   | 18   | 06   | 1D   | 02   | 58   | 03   | 00   | 00   |
| 1024*768                      | 01  | 04   | 10   | 40   | 04   | 00   | 20   | 04   | 08   | 03   | 00   | 04   | 00   | 00   |
| 1280*720                      | 01  | 03   | 10   | 40   | 05   | 00   | 20   | 08   | 20   | 02   | D0   | 20   | 00   | 00   |
| 1280*800                      | 01  | 03   | 10   | 1C   | 05   | 00   | 10   | 08   | 10   | 03   | 20   | 10   | 00   | 00   |
| 1366*768                      | 01  | 03   | 10   | 20   | 05   | 54   | 20   | 06   | 10   | 03   | 00   | 08   | 00   | 00   |
| 136*768 eDP                   | 00  | 03   | 10   | 20   | 05   | 54   | 20   | 06   | 10   | 03   | 00   | 08   | 10   | 00   |
| 1024*768 VGA                  | 00  | 03   | 88   | A0   | 04   | 00   | 18   | 06   | 1D   | 03   | 00   | 03   | 00   | 00   |
| 1280*800 VGA                  | 00  | 03   | 80   | C8   | 05   | 00   | 48   | 06   | 16   | 03   | 20   | 03   | 00   | 00   |



### 3.3.2.6 T5\*.CRC File Format

| Category                  | Address | Length | Definition  |
|---------------------------|---------|--------|---|
| File identification       | 0x0000  | 4      | Fixed to 0x43 0x52 0x43 0x10 .  |
| CRC result                | 0x0004  | 4      | Specify the check position data, and perform the final result of CRC32 check in sequence.   |
| Reserve                   | 0x0008  | 7      | Write 0x00.   |
| OS core code verification | 0x000F  | 1      | 0x5A means to perform the CRC check of the OS core code, and the rest means skip the check.   |
| LIB file check            | 0x0010  | 2      | 0x10 = ID of LIB file to be checked; 0x11= Number of files to be checked at the beginning of this ID, 0x00 means the end of LIB file check.   |
|                           | 0x0012  | 62     | The remaining 31 defined positions.   |
| Font file verification    | 0x0050  | 2      | 0 x50 = the ID of the font file to be checked; 0x51 = the number of files that need to be checked at the beginning of this ID, 0x00 means the end of the check of the font file.<br>For off-chip NOR Flash, the font size is 2 5 6 KB.<br>For off-chip NAND Flash, the font file size is 8 MB; and due to bad blocks, the last file should not be used or verified. |
|                           | 0x0052  | 254    | The remaining 127 defined positions.  |
| Reserve                   | 0x0150  | 176    | Write 0x00.   |

The polynomial for CRC32 calculation is  $X^{32}+X^{26}+X^{23}+X^{22}+X^{16}+X^{12}+X^{11}+X^{10}+X^8+X^7+X^5+X^4+X^2+X+1$  .

### 3.3.2.6 CTPCFG\*.CFG capacitive touch screen configuration burning file format

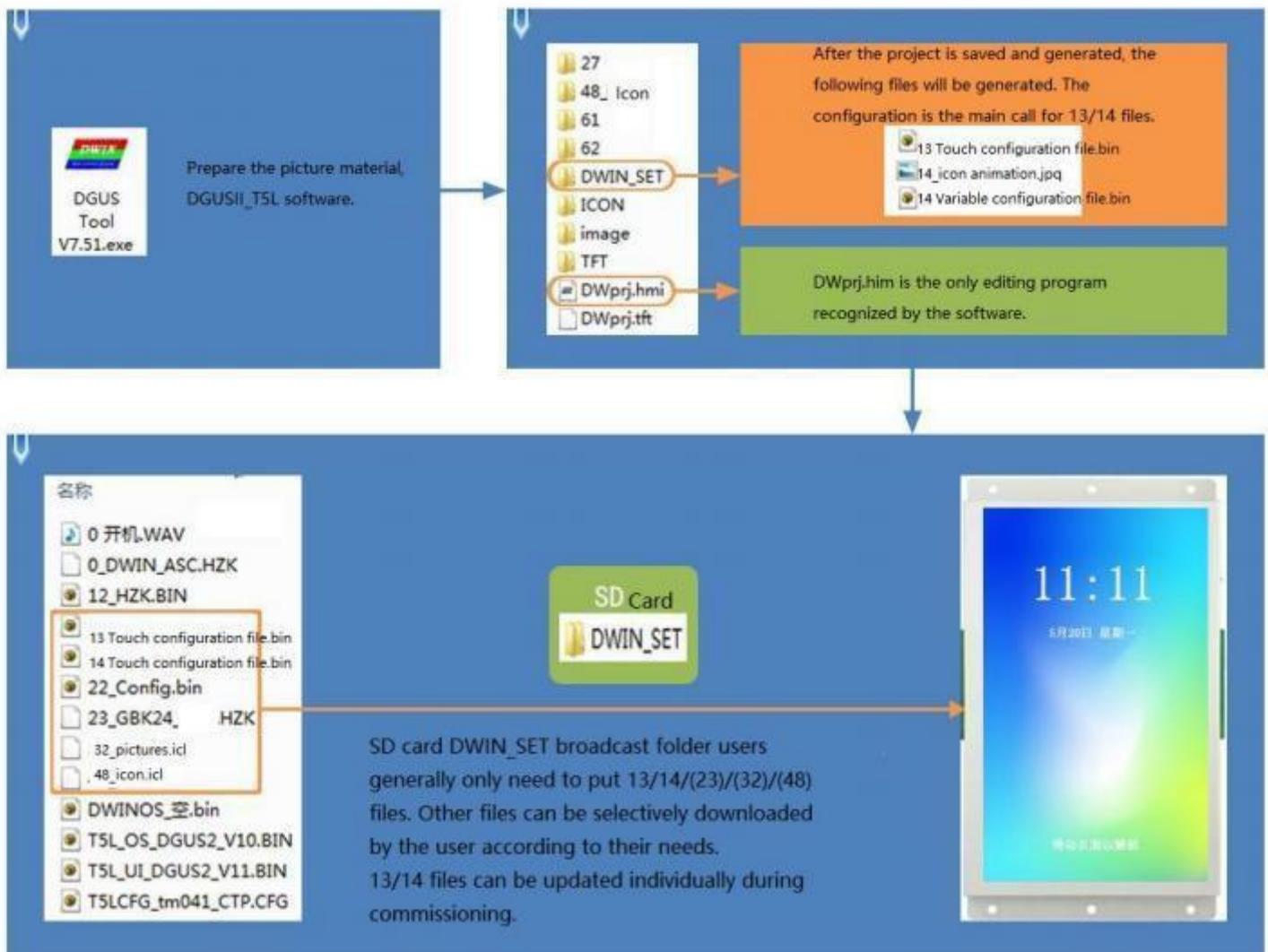
| Category                           | Address | Length | Definition   |
|------------------------------------|---------|--------|--|
| File recognition                   | 0x0000  | 4      | Fixed as 0x43 0x54 0x50 0x53   |
| CTP IC                             | 0x0004  | 2      | 0x0001-GT911/GT9110H   |
| PACK_NUM                           | 0x0006  | 1      | Configure the number of packets, 0x01-0x08   |
| PACK_Delay                         | 0x0007  | 1      | Configure the delay time after 1 packet (then configure the next packet), 0x01-0xFF, the unit is 1ms |
| Reserved                           | 0x0008  | 24     | Reserved, write 0x00   |
| Configuration package 1 definition | 0x0020  | 2      | The number of registers that need to be configured in this package                                   |
|                                    | 0x0022  | 2      | The starting register address configured by this package   |
|                                    | 0x0024  | 476    | Register data  |
| Configuration package 2 definition | 0x0200  | 2      | The number of registers that need to be configured in this package                                   |
|                                    | 0x0202  | 2      | The starting register address configured by this package   |
|                                    | 0x0204  | 508    | Register data  |
| .....                              |         |        |  |
| Configuration package 8 definition | 0x0E00  | 2      | The number of registers that need to be configured in this package                                   |
|                                    | 0x0E02  | 2      | The starting register address configured by this package   |
|                                    | 0x0E04  | 508    | Register data  |

### 3.3.3 File Download

Only the folder named as DWIN\_SET can be identified by screens, which should contains all files that need to be downloaded, such as "13TouchFile", "14ShowFile", "22\_Config.bin", etc.

When creating a new project, the software will automatically generate a series of files, among which "DWprj.hmi" is the only editing program that the DGUS software can recognize. This file cannot be renamed or deleted.

Since the T5L screen cannot read JPEG pictures directly, it is necessary to make pictures and icons into ICL files. The generation process of download files is as follows .



Download files generation diagram

## Chapter 4 Communication Protocol

### 4.1 UART(UART2)

The system debugging UART2 mode is fixed to 8N1, and the baud rate can be set. The data frame consists of 5 data blocks.

| Data block  | 1            | 2   | 3                       | 4       | 5                    |
|-------------|--------------|---|-------------------------|---------|----------------------|
| Definition  | Frame Header | Length                                    | Instruction             | Data    | CRC Check( optional) |
| Length      | 2            | 1   | 1                       | Max.249 | 2                    |
| Description | 0x5AA5       | Including instruction, data and CRC check | 0x82 Write<br>0x83 Read |         | CRC-16(x16+x15+x2+1) |

The CRC check on/off is controlled by bit 0x05.7 of the configuration file.

| Instruction               | Unable CRC Check              | CRC Check                           |
|---------------------------|-------------------------------|-------------------------------------|
| 0x83 Read Instruction     | Tx: 5AA5 04 83 000F 01        | Tx: 5AA5 06 83 000F 01ED 90         |
| 0x83 Instruction Response | Rx: 5AA5 06 83 00 0F 01 14 10 | Rx: 5AA5 08 83 00 0F 01 14 10 43 F0 |
| 0x82 Write Instruction    | Tx: 5AA5 05 82 10 00 31 32    | Tx: 5AA5 07 82 10 00 31 32 CC 9B    |
| 0x82 Instruction Response | Rx: 5AA5 03 82 4F 4B          | Rx: 5AA5 05 82 4F 4B A5 EF          |
| 0x83 Touch Upload         | Rx: 5AA5 06 83 10 01 01 00 5A | Rx: 5AA5 08 83 10 01 01 00 5A 0E 2C |

### 4.2 UART2 Debug Interface Instructions

The DGUS screen works in a variable-driven mode, and the working mode of the screen and the state of the GUI are completely controlled by data variables. Therefore, serial port instructions only need to read and write variables, and the instruction set is very simple, users only need to use 82/83 instructions.

| Instruction   | Data  | Description  |
|---|---|--|
| 0x80  | Send:<br>Register page (0x00-0x08)+ register address (0x00-0xFF) + data written                             | Specify the address to start writing the data string to the register.  |
|   | Answer:<br>0x4F 0x4B.   | The write instruction is answered.   |
| 0x81  | Send:<br>Register page (0x00-0x08) + register address (0x00-0xFF) + byte length of the read data(0x01-0xFB) | Start reading data from the specified register.  |
|   | Answer:<br>Register page (0x00-0x08) + register address (0x00-0xFF) + data length + data                    | The data is answered.  |
| <p><b>Instruction example: 5A A5 04 81 00 0A 04 Read 04 bytes of data from os registers R10~R13 in register page 00. 80 81 Instructions are generally not used by users, and are usually used for debugging access to OS functions. You can refer to the register table in the OS application guide. OS Registers Register Page IDs (0x00-0x07) and Interface register page ID (x08).</b></p> |   |  |
| 0x82  | Send:<br>Variable space first address (0x0000-0xFFFF) + written data  | Write data string ( word data) to variable space starting from the specified address. Do not write the space reserved by the system. |
|   | Answer:<br>0x4F 0x4B.   | Write instruction answer.  |



|      |  |  |
|------|--|--|
| 0x83 | Send:<br>Variable space first address (0x0000- 0xFFFF) + byte length of the read data(0x01-0x7D)               | Read word data of the specified length from the specified address of the variable space.   |
|      | Answer:<br>Variable space first address + byte length of the variable data + the read variable data            | The data is answered.  |
| 0x84 | Send:<br>CH_Mode (Byte) +DATA0 (Word) + ...+DATAn  | Write curve buffer data.<br>CH_Mode defines the order in which the channels of subsequent data are arranged.<br><br>Each bit ( bit) of CH_Mode corresponds to 1 channel.<br>CH_Mode.0 corresponds to channel 0, .7 to channel 7.<br>The corresponding position 1 indicates the presence of the corresponding channel data.<br>The corresponding position 0 indicates that the corresponding channel data does not exist. |
|      | Answer:<br>0x4F 0x4B   | The low channel data comes first.<br>For example, CH_Mode=0x83 (10000011B) means that the subsequent data (channel 0 + channel 1 + channel 7) + . . . + ( channel 0+channel 1+ channel 7).<br>( channel 0 + channel 1 + channel 7 ) .<br>The curve buffer is defined in the 0x1000 - 0x4FFF variable storage space, 2K Words per curve.  |
| 0x86 | Send:<br>Variable space double word first address (0x000000-0x00FFFF) + data written                           | Write data to the variable space from the specified double word address.<br>The accessible variable space is 256 Kbytes .<br>Do not write the system reserved space.   |
|      | Answer:<br>0x4F 0x4B   | Write instruction answer.  |
| 0x87 | Send:<br>Variable space double word first address (0x000000-0x00FFFF)+read data double word length             | Read specified length double- word data from the variable space specified double word address.<br>The accessible variable space is 256 Kbytes.   |
|      | Answer:<br>Variable space double word first address+ variable data double- word length+ the read variable data | The data is answered.  |

Note: DGUS II has replaced common hardware-related operating registers with system variable address interface access instructions. The space reserved by the system does not need to be accessed.

The register pages are defined as follows.

| Register Page ID | Definition         | Description  |
|------------------|--------------------|--|
| 0x00-0x07        | Data register      | 256 per group, R0-R255   |
| 0x08             | Interface register | DR0 -DR255. For details, please refer to "DWIN OS Development Guide based on T5 " 3.4 Interface register definition description. |

T5L supports ED4 USB downloader to communicate with DGUS screen through SD card interface. The virtual serial port mode of the downloader to communicate on the PC side is 8N1, and the baud rate is fixed at 8Mbps. When the PC-side software is processing, it must receive the ED4 response before sending a new data frame. Each data frame can read and write up to 128KB of data.



### 4.2.1 Write variable memory instruction instruction (0x82)

Here, take writing the value 2 to the variable address of 1000 as an example:

**5A A5 05 82 1000 0002**

5A A5 means: frame header

05 means: data length

82 means: write variable memory instruction

1000 means: variable address (two bytes)

0002 means: data 2 (two bytes)

Explanation: assign a value of 2 to address 0000 through the command, and the display on the screen shows that the data variable is of integer type 2

Note: The address of the DGUS screen **reading and writing** variables can be regarded as the starting address: for example, the effect of the following a and b commands is equal to c

a: 5AA5 05 82 1001 000A

b: 5AA5 05 82 1002 000B

c: 5AA5 07 82 1001 000A 000B

### 4.2.2 Read variable memory command command (0x83)

Here is an example of reading the value in the variable address of 1000 (assuming the current value is 2):

**5A A5 04 83 1000 01**

5A A5 means: frame header

04 means: data length

83 means: read data storage area command

1000 means: variable address (two bytes)

01 means: read 1 word length from address 1000, the maximum allowable length of data instruction is 0x7c

After reading, the screen will return a read response to the serial port:

**5A A5 06 83 1000 01 0002**

5A A5 means: frame header

06 means: data length

83 means: read variable memory instruction instruction

1000 means: variable address (two bytes)

01 means: read 1 word length data from address 1000

00 02 means: the data value in address 1000 is 2

### 4.2.3 Touch the key to return to the serial port data (0x83)

Here, take the return variable address 0x1001 and key value 0x0002 as an example:

**5A A5 06 83 1001 01 0002**

5A A5 means: frame header

06 means: data length



83 means: read variable memory instruction instruction

1001 means: variable address (two bytes)

01 means: 1 word length data

00 02 means: key value 0002

Button return (non-basic touch) can be sent through the serial port after the system configuration CFG file configures data upload.

Upload protocol format: (key return address 1001 key value 000A) 5A A5 06 83 1001 01 000A

After pressing the return button, the value screen in this address will not be automatically cleared.

For RS485 half-duplex communication, you can use the 5A A5 04 83 10 01 01 read command to read the key value, and the response format of the read command is consistent with the touch upload format.

RS485 adopts half-duplex communication mode, so do not receive data and send data at the same time.

### 4.2.4 Virtual UART

(ED4 USB downloader, communicating with DGUS screen via SD card interface)

The PC side ED4 USB downloader communication mode is 8 N1, with a fixed baud rate of 8 Mbps, and each data frame consists of 4 data blocks.

Interface table:

| Data block        | 1            | 2                               | 3           | 4        |
|-------------------|--------------|---------------------------------|-------------|----------|
| Definition        | Frame Header | Length                          | instruction | Data     |
| Length            | 2            | 3                               | 1           | N        |
| Description       | 0x5AA5       | Including instruction and, data | 0x82/0x83   |          |
| Sample(no parity) | 5A A5        | 00 00 05                        | 83          | 00100004 |

| Instruction | Data  | Description  |
|-------------|---|--|
| 0x82        | Send:<br>Variable space first address (0x0000- 0xFFFF) + written data                               | Write data string ( word data) to variable space starting from the specified address. Do not write the space reserved by the system. |
|             | Answer:<br>0x4F 0x4B.   | Write instruction answer.  |
| 0x83        | Send:<br>Variable space first address (0x0000 - 0xFFFF) + byte length of the read data(0x01-0x7D)   | Read word data of the specified length from the specified address of the variable space.   |
|             | Answer:<br>Variable space first address + byte length of the variable data + the read variable data | The data is answered.  |

The PC software must receive an ED4 response before sending a new data frame, and each data frame can read and write up to 128 KB of data.



## Chapter 5 System Variable Interface

### 5.1 System Variable Interface

System variable address range: 0x0000-0x0FFF.

| VP   | Definition       | R/W | Length (word) | Description  |
|------|------------------|-----|---------------|--|
| 0x00 | Reserved         |     | 4             | Undefined  |
| 0x04 | System_Reset     | W   | 2             | Write 0x55 AA 5AA5 to reset the T5L CPU once.<br>E.g. 5A A5 07 82 00 04 55 AA 5A A5<br>Reset T5L, clear all data, that is equivalent to power off.   |
| 0x06 | OS_Update_CMD    | W   | 2             | D3 : write 0x5A to enable DWIN OS program once (write into on-chip Nor Flash). clear after CPU operation.<br>D2: File type<br>0x10: must be DWIN OS program from 0x1000, update 28KB at every time.<br>0xA5: 8051 code, update 64KB at every time.<br>D1: 0 : Start address of SRAM to save/ update program to. It must be even.<br>E.g. This instruction is used for downloading and updating the user OS program. " DWIN OS Builder" PC software has integrated the function of downloading the OS. bin file into the screen through UART 2 after compilation.   |
| 0x08 | NOR_FLASH_RW_CMD | W   | 4             | D7: Operating mode: 0x5A=read 0xA5=write. Clear after operation.<br>D6:4: head address of Nor Flash. It must be even. 0x000000-0x03:FFFE, 256 KWords.<br>D3:2: head address of data variable space. It must be even.<br>D1:0: Data length to read/write. Must be even.<br>E.g.<br>( 1) Write the data 0x12345678 to the variable storage space 0x1002: 5A A5 07 82 10 02 12 34 56 78<br>(2) Store 2-word data of variable storage space 0x1002 address into Nor Flash database header address 0x000002:<br>5A A5 0B 82 0008 A5 00 0002 1002 0002<br>(3) Read data from Nor Flash database 0x00 0002 to variable storage space address 0x2000:<br>5A A5 0B 82 0008 5A 00 0002 2000 0002.<br>After reading and storing the database with OS, it is necessary to judge whether the D7 address has been automatically cleared. If the D7 address has not been cleared, no operation will be carried out until it is cleared for other operations. Otherwise, logic disorder will occur.<br>If the serial port is used for instruction reading and storage operation, other instructions can be sent after appropriate delay. |



|      |                     |   |   |   |
|------|---------------------|---|---|---|
| 0x0C | UART2 configuration | W | 2 | <p>D3: Write 0x5A to configure the UART2 baud rate once; change it to 0xA5 after OS core processing.<br/> D2: serial port mode, bit definition.<br/> .7 CRC check, 1=on, 0=off.<br/> .6-.0 Reserved, write 0.<br/> D1:D0: Baud rate configuration value, D1:D0=CPU main frequency/(64*set baud rate).</p> <p>1. Application instructions<br/> Baud rate and frame CRC switches for real-time modification of UART2.</p> <p>2. Instruction example<br/> Configuration value:<br/> 206.4384MHz*1000000=206438400Hz<br/> 9600 baud rate configuration value: 206438400/(64*9600)=336=0x0150<br/> 115200 Baud rate configuration value: 206438400/(64*115200)=28=0x001C</p> <p>(1) The baud rate is modified, and the CRC is not modified (closed)<br/> ① Change the baud rate from 115200 to 9600<br/> Send: 5AA5 07 82 000C 5A00 0150<br/> After the modification, you can send a command to read the version number. If the correct version number is returned, the modification is successful. After the baud rate is modified on the screen,<br/> The baud rate of the sending end also needs to be modified.<br/> Read version number: 5AA5 04 83 000F 01 (0x01 refers to reading the interface address of 0x000F, and returns the address in 0x000F<br/> 1 word length version number data stored on the face. )<br/> Return: 5AA5 06 83 000F 01 63 22 (response to the command above, the high byte 0x63 is the GUI version number,<br/> The low byte 0x22 is the DWIN OS software version number, and the program version ID will be updated. The data returned here is for reference only. )</p> <p>② Change the 9600 baud rate above to 115200<br/> Send: 5AA5 07 82 000C 5A00 001C (the above has been modified, the baud rate of the sending end needs to be 9600)<br/> Read version number: 5AA5 04 83 000F 01 (the baud rate of the sending end must be 115200)<br/> Return: 5AA5 06 83 000F 01 63 22</p> <p>(2) The baud rate is not modified, but the CRC is modified<br/> ① Enable CRC: 5AA5 05 82 000C 5A80<br/> Read version number: 5AA5 06 83 000F 01 ED90 (0xED90 is CRC16 check, see section 4.2.4)<br/> Return: 5AA5 08 83 000F 01 63 22 E5D5<br/> ② Close CRC: 5AA5 07 82 000C 5A00 A6BD<br/> Read version number: 5AA5 04 83 000F 01<br/> Return: 5AA5 06 83 000F 01 63 22</p> <p>(3) Baud rate modification + CRC modification<br/> ① Change the baud rate from 115200 to 9600, and enable CRC:<br/> Send: 5AA5 07 82 000C 5A80 0150<br/> Read version number: 5AA5 06 83 000F 01 ED90 (the baud rate of the sending end must be 9600)<br/> Return: 5AA5 08 83 000F 01 63 22 E5D5<br/> ② Change the 9600 baud rate above to 115200, close CRC<br/> Send: 5AA5 09 82 000C 5A00 001C BA18 (the above has been modified, the baud rate of the sending end needs to be 9600)<br/> Read version number: 5AA5 04 83 000F 01 (the baud rate of the sending end must be 115200)<br/> Return: 5AA5 06 83 000F 01 63 22</p> <p>3. Function setting<br/> (1) CRC check<br/> Set the factory baud rate to 115200, CRC is off, and the return value of the above version number is tested under the V63 GUI version and V22 DWIN OS version. After the program is updated, the return version number data is modified, and the check value will also be modified.<br/> (2) Touch settings<br/> It can be set through the pop-up menu, data entry, and key value return. After setting the value, you need to write 0x5A to the high byte of 0x000C to write the address.<br/> When using the key value return to check the high byte, the low byte of the return key value is stored in the high byte address (VP_H) of the VP address.<br/> (3) The set value will not be saved when power off.</p> |
|------|---------------------|---|---|---|



|      |            |     |   |  |
|------|------------|-----|---|--|
| 0x0F | Ver        | R   | 1 | <p>Application software version. D3=0x00, D2=CTP driver software version, D1 means GUI software version, D0 means DWIN OS software version.</p> <p>E.g.</p> <p>Send: 5A A5 04 83 000F 01</p> <p>Return: 5A A5 06 83 000F 01 6121 (61 is GUI software version, 21 is DWIN OS software version)</p> <p>Send: 5A A5 04 43 000E 02</p> <p>Return: 5A A5 08 83 000E 02 00 41 61 21</p> <p>Application :</p> <p>(1) When using a display terminal, due to the control system, display terminal start time inconsistent (depending on the user' s power supply capacity and electricity rate) , to ensure that the control system of a start sending data display terminal received correctly and carry out the corresponding functions and instructions, so need to confirm whether the display terminal has the normal operation, the user can be identified by this directive;</p> <p>(2) In the process of debugging need to know whether the system version, configuration and so on is correct, you can also send this instruction to read.</p> <p>(3) The transmission and response of this instruction is often used to verify that the communication between the computer serial port and the serial screen is normal.</p> |
| 0x10 | RTC        | R/W | 4 | <p>D7=Year (0-0x63) D6=Month(0-0x0C), D5=Day(0-0x1F), D4=Week(0-0x6), D3 =Hour(0-0x17), D2=Minute(0-0x3B), D1=Second(0-0x3B), D0 Undefined. Data format is HEX.</p> <p>Without RTC on hardware, user can write RTC data.</p> <p>E.g.</p> <p>Write: 5A A5 0B 82 0010 13 0A 01 00 0B 0C 0D 00</p> <p>Text RTC display 2019-10-01 11:12:13 SUN,</p> <p>The first two digits of the year are automatically identified;</p> <p>The week is displayed in English, and the system will convert automatically.</p> <p>Read: 5A A5 04 83 00 1004</p> <p>Answer: 5A A5 0C 83 00 10 04 13 0A 01 00 0B 0C 0D 00</p>  |
| 0x14 | PIC_Now    | R   | 1 | <p>Display current page ID. Read only.</p> <p>E.g.</p> <p>Read: 5A A5 04 83 0014 01</p> <p>Answer: 5A A5 06 83 00 14 01 0007 (0007 is page 07)</p>   |
| 0x15 | GUI_Status | R   | 1 | <p>GUI status feedback: 0x0000=free, 0x0001=processing 13.bin and 14.bin.</p> <p>The operation state feedback is in millisecond unit level, which is generally applied in special cases. The user can judge whether the GUI kernel is occupied by the DWIN - OS program.</p>   |
| 0x16 | TP_Status  |     | 4 | <p>D7: 0x5A= touch panel data is updated. Others= touch panel coordinates not updated</p> <p>After reading the data, the user can write ≠ 0x5A as a mark. As long as the user does not touch, the mark will not be updated</p>   |



|                  |                             |     |    |  |
|------------------|-----------------------------|-----|----|--|
|                  |                             |     |    | <p>D6: touch panel status. 0x00=release, 0x01=first press, 0x02=lift, 0x03=pressing</p> <p>D5: D4=X coordinate</p> <p>D3: D2=Y coordinate</p> <p>D1: D0=0x0000.</p> <p>E.g.</p> <p>Read: 5A A5 04 83 001603</p> <p>Answer: 5A A5 0A 83 00 16 03 5A 02 02 55 00 E5</p> <p>Read the coordinates of the touch</p> <p>Write: 5A A5 05 82 0016 0000 reset touch panel data update mark.</p> <p>Usually, DWIN OS program can be used to deal with the judgment of touch, long press, short press and other operations.</p>   |
| <b>0x1A-0x30</b> | Reserved                    |     | 23 | Undefined  |
| <b>0x31</b>      | LED_Now                     | R   | 1  | <p>D1: 0x5A : backlight brightness value, AD0- AD7 instant value is updated.</p> <p>D0: current backlight brightness value, 0x00-0x64.</p> <p>E.g.</p> <p>Read: 5A A5 04 83 003101</p> <p>Answer: 5A A5 06 83 00 31 01 5A 64</p>   |
| <b>0x32</b>      | AD0-AD7 Instantaneous value | R   | 8  | <p>Instantaneous value of AD0-AD7, 1 word per AD. The resolution is set by CFG file 0x06.2.</p> <p>12bit mode: voltage=AD value*3300/4095mV.</p> <p>16bit mode: voltage=AD value*3300/65520mV.</p>   |
| <b>0x3A-0x79</b> | Reserved                    |     | 64 | Undefined  |
| <b>0x7A</b>      | LCD_HOR                     | R   | 1  | <p>Horizontal resolution of the screen</p> <p>E.g.</p> <p>Read: 5AA5 04 83 007A 01</p> <p>Answer: 5A A5 06 83 00 7A 01 01 E0 (0x01E0 is horizontal resolution)</p>   |
| <b>0x7B</b>      | LCD_VER                     | R   | 1  | <p>Vertical resolution of the screen</p> <p>E.g.</p> <p>Read:5AA5 04 83 007B 01</p> <p>Answer:5A A5 06 83 00 7B 01 01 10 (0x0110 is vertical resolution)</p>   |
| <b>0x7C-0x7F</b> | Reserved                    |     | 4  | Undefined  |
| <b>0x80</b>      | System_Config               | R/W | 2  | <p>D3: 0x5A = enable the parameter configuration once, clear after operation.</p> <p>D2: touch panel sensitivity configuration value, read only.</p> <p>D1: touch panel mode configuration value, read only.</p> <p>D0: system status setting.</p> <p>7: Serial port CRC check 0=off 1=on, read only.</p> <p>6: Reserved, write 0.</p> <p>5: Power on load 22 file to initialize variable space. 1=load 0=unload, read only.</p> <p>4 : Variable automatic upload setting 1= on, 0= off, read and write.</p> <p>3: Touch panel audio control 1= on 0= off, read and write.</p> <p>2: Touch panel backlight standby control 1= on 0= off, read and write.</p> |



|      |             |     |   |   |
|------|-------------|-----|---|---|
|      |             |     |   | <p>1. 0: display direction 00 = 0 ° , 01 = 90 ° , 10 =180 ° ,11 = 270 ° , read and write.</p> <p>E.g.<br/>Read: 5A A5 04 83 0080 02<br/>Answer: 5A A5 08 83 00 80 02 00 14 1038<br/>Read instructions are typically used to view the screen .CFG configuration . D0 can change the configuration. CFG file 0 x05 address, for example, the next two instruction are on the corresponding .4 = 1 .5 =1, TP sound control off or on, can be applied to the touch button of the buzzer “ mute” function.<br/>Note: write instruction is not saved after power off.<br/>Write: 5A A5 07 82 0080 5A 00 00 30<br/>Write: 5A A5 07 82 0080 5A 00 00 38</p>   |
| 0x82 | LED_ Config |     | 2 | <p>Set standby backlight.<br/>D3 = Turn on brightness, 0x00-0x64; When backlight standby control is off, D3 is the brightness adjustment interface.<br/>D2= Turn off brightness, 0x00-0x64; D1:0=open time /10 ms.</p> <p>E.g.<br/>1. When CFG address 0x05 is configured to turn on backlight standby control of touch panel:<br/>5A A5 07 82 0082 6432 03E8, 10 seconds after the screen backlight automatically reduced to 0x32 brightness.<br/>5 A A5 07 82 0082 2020 ****, direct instruction control backlight brightness 0x20 brightness, and backlight screen saver brightness is consistent with standby.<br/>Note: the first physical touch on the touch panel after a certain period of time is to wake up the backlight function of the screen saver. Even if the brightness is the same as the setting value of standby brightness, additional first awakening touch and click operation" is still needed after a certain period of time.<br/>2. When CFG address 0x05 is configured with backlight standby control over touch panel:<br/>5A A5 05 82 0082 0A**, direct instruction control backlight brightness is 0x0A brightness.</p> |
| 0x84 | PIC_Set     | R/W | 2 | <p>D3:0x5A= enable page operation once, The CPU is cleared after processing.<br/>D2: Processing Mode.<br/>0x01=page switch ( Display the image specified in the image storage area to the current background page).<br/>0x02=This mode is reserved and not supported.</p> <p>Instruction example:<br/>5A A5 07 82 0084 5A01 0001, display the 01 serial number picture under the background picture ICL file (if the serial number picture ID of the sent instruction does not exist, the instruction operation judgment will not take effect.)</p>   |



|           |          |     |    |  |
|-----------|----------|-----|----|--|
| 0x86      | PWM0_Set | R/W | 2  | <p>D3 =write 0x5A to enable PWM0 setting once, The CPU is cleared after processing.</p> <p>D2 = frequency division coefficient</p> <p>D1:D0= PWM0 precision</p> <p>PWM0 carrier frequency=825.7536 MHz/(frequency division coefficient PWM0 precision).</p> <p>E.g.</p> <p>1. .Write 0x0086 variable configuration 100 KHz<br/>The division factor is set to 1, upper limit value=825.7536MHz/0. 1MHz=8258 (0x2042).</p> <p>Write data 5A 01 20 42 to the variable of 0x0088, output 13 bit resolution, 100 KHz carrier PWM.</p> <p>2. Write 0 x0093 variable to control the output duty ratio of PWM0 Write value = high level ratio * upper limit value<br/>For example, 10%, write value is 10% * 8258= 826(0x033A)</p> |
| 0x88      | PWM1_Set | R/W | 2  | <p>D3= 0 x5 A starts a PWM1 setting, and the CPU clears it after processing. D2 = frequency division coefficient D1:D0 = PWM1 precision PWM1 carrier frequency= 825.7536 MHz/( frequency division coefficient*PWM1 precision).</p> <p>1. Write 0x0088 variable to configure 100 KHz.<br/>The frequency division coefficient is 1 , the upper limit=825.7536MHz/0. 1MHz=8258 (0x2042)<br/>So, write data 5A 01 20 42 to 0x0088 variable, output PWM with 13bit resolution and 100 KHz carrier.</p> <p>2. Write the 0x0093 variable to control the output duty cycle of PWM1 as needed.</p> <p>Write value = high level ratio * upper limit value.<br/>For example, 10%, write 10%*8258=826 (0x033A).</p>                    |
| 0x8A-0x91 | Reserved |     | 10 | Undefined  |
| 0x92      | PWM0_Out | R/W | 1  | D1: D0= PWM0 output high level width, 0x0000- PWM0 precision.  |
| 0x93      | PWM1_Out | R/W | 1  | D1: D0= PWM1 output high level width, 0x0000- PWM1 precision.  |
| 0x94-0x9B | Reserved |     | 9  | Undefined  |



|      |                       |     |   |   |
|------|-----------------------|-----|---|---|
| 0x9C | RTC_Set               | W   | 4 | <p>D7:D6=write 0x5AA5 to enable RTC setting once;<br/>D5:D0=year, month, day, hour, minute, second, all in HEX format.<br/>Need hardware RTC support.</p> <p>0X9C T5 RTC entry: RTC rewriting occupies 4 addresses in total: 009C, 009D, 009E, and 009F. 009C write 5AA5</p> <p>Start the RTC setting, the high and low bytes of 009D correspond to the year and month, the high and low bytes of 009 E correspond to the date and time, and the high and low bytes of 009F are paired<br/>Should be seconds.</p> <p>① The key value is returned, the address is set to 009C, and the key value is set to 5AA5. Press the "Save Settings" button.</p> <p>② Data variable entry, address 009D, variable type set the high byte of variable pointer, make a quantity variable display,<br/>Address 009D, variable type set VP high byte; high byte changes year.<br/>Data variable entry, address 009D, variable type set variable pointer low byte, make a quantity variable display, ground<br/>Address 009D, variable type VP low byte; low byte changes month.<br/>The day, hour, minute, second, and repeat are divided into high and low bytes and are copied and arranged in sequence, with a total of 6 input keys and 6 display controls.<br/>After a single input operation, click the ① button "Save Settings" 009C button value to change the time.</p> |
| 0xA0 | WAE<br>Music_Play_Set | R/W | 2 | <p>The on-chip Flash saves WAE music playback settings:</p> <p>D7: indicates the segment ID of this playback, 0x00 - 0xFF. The WAE file location is specified by the CFG configuration file or D2:D1.</p> <p>D6: Number of playing segments, fixed at 0x01, cleared after DGUS processing; In buzzer mode, the unit is 8ms.</p> <p>D5: Playback volume, unit:1/64;The initial power-on value is 0x40 (100%).</p> <p>D4: Playback status feedback, 0x00= stop, 0x01=pause, 0x02=playback.</p> <p>D3 : Segment ID of this playback, 0x00-0xFF. WAE file location is specified by CFG configuration file.</p> <p>D2: Number of segments, fixed 0x01 . Clear after DGUS operation. Under buzzer mode, it is buzz time, unit: 8ms.</p> <p>D1: Volume, unit: 1/64; Initial value is 0x40(100%).</p> <p>D0: Feedback, 0x00= stop, 0x01= pause, 0x02= playing.</p> <p>Example buzzer control instructions:<br/>5 A A5 05 82 00A0 007D The buzzer beeps for 1 second.</p>  |



|                  |  |     |   |   |
|------------------|--|-----|---|---|
|                  |  |     |   | <p>WAE music playback setting:</p> <p>1 . WAE file save location setting</p> <p>① Set the save location of the .wae file by modifying the parameters of the 0x07 address of the CFG file.</p> <p>For example: 26 .wae corresponds to the 0x07 position of the CFG file, write 0x1A.</p> <p>② Configure the parameter modification of the 0 x05 address of the CFG file.</p> <p>For example: write 0x7C to the 0x05 position, and the .6 bit enables music playback. (.6: Buzzer/ music play selection, 0= buzzer 1= music play.)</p> <p>③ Instruction format: 5A A5 07 82 00A0 03 01 40 00 Play music with ID 03 in wae file, play segment 1 segment, volume is 100%, D0: feedback is read only, write instruction can write 00 without definition .</p>  |
| <b>0xA4-0xA9</b> | Reserved                                 |     | 8 | Undefined   |
| <b>0xAA</b>      | External 16 Mbytes FLASH write operation | R/W | 6 | <p>Update 16Mbytes of external memory content based on 32Kbytes blocks.</p> <p>D11: 0x5A= enable external memory operation, clear after operation.</p> <p>D10: Operation mode</p> <p>D10=0x10, read 16MB SPI NOR FLASH data</p> <p>D9: font library ID, 0x10-0x1F, 256Kbytes per font library, maximum 4Mbytes</p> <p>D8:D6: the starting address of data in the font library is defined according to the Word, 0x0000-0x0FFFF</p> <p>D5: D4: The first address of the read data variable space, which must be even.</p> <p>D3: D2: The length of the data read, defined according to the Word, must be even.</p> <p>D1:D0: undefined, write 0x00.</p> <p><b>For D10=0x02 write 32 Kbytes data block into 16 MB SPI NOR FLASH.</b></p> <p>D9: D8: 32Kbytes memory block address, 0x0000-0x01FF, correspond to 16 Mbytes memory.</p> <p>D7: D6: The first address of the update data stored in the data variable space must be even.</p> <p>D5: D4: After the completion of this operation, the time of delay waiting for the next write operation, unit:1 ms.</p> <p>DGUS refresh will stop during the delay wait to prevent errors caused by incomplete updates.</p> <p>D3:D0: Undefined, write 0x00 .</p> <p><b>D10=0x03, font copy</b></p> <p>D9:D8: Source font ID, 0x0000-0x001F, 256Kbytes per font library, max 4 Mbytes.</p> <p>D7:D6: Target font ID, 0x0000-0x003F.</p> <p>D5-D4: Number of font to copy, 0x0000-0x0010.</p> <p>D3:D0: Undefined, write 0x00 .</p> <p><b>D10=0x04, D10=0x04, write multiple 4KB data blocks to NAND FLASH ( NAND Flash must be externally expanded) .</b></p> |

D9: D6: The starting address of the NAND Flash to be written, the lower 12 bits are 0 (4KB alignment).

If the lower 18 bits of the starting address are all 0s, the T5L will first erase the 256 KB block before writing.

D5: D4: The write data is stored in the first address of the data variable space, which must be an even number.

D3: Number of 4KB blocks written, 0x01-0x10.

D2: D0: undefined, write 0x00 .

**D10=0x05, 32-bit CRC check of NAND Flash data ( NAND Flash must be externally expanded) .**

D9: D6: NAND source address, 0x0000:0000-0x1FFF:F000, the lower 12 bits are 0 (4 KB alignment).

D5: D4: The number of 4KB data blocks to be checked, 0x0001-0 x8000 , the maximum is 128 Mbytes.

D3: D0: 32-bit CRC check return value, the CRC check polynomial is the same as the SD download CRC check.

**D10=0x06 , NAND Flash Copy ( NAND Flash must be externally expanded).**

D9:D6: NAND source address, 0x0000:0000-0x1FFF:F000, the lower 12 bits are 0 (4KB alignment).

D5:D4: Target font ID, 0x0000-0x00FF.

0 x00-0xBF is SPI NOR Flash, the size of a single font is 256 Kbytes;

0 x0C0-0xFF are SPI NAND Flash, and the size of a single font is 8Mbytes.

D3: D2: The number of fonts to be copied.

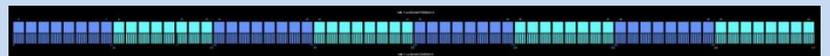
D1:D0: undefined, write 0x00.

If the copy is accidentally interrupted by power failure, it will automatically continue until it is completed when the power is restarted.

**Example 1: 0xAA 02 mode serial port update 32.ICL file**

Based on the standard 16MB Flash, update 32.ICL with a size of 4.17KB and the first address of the data variable space (RAM) 0x8000

1. Based on the 32Kbytes data block, calculate the start position of the ICL data packet.



The 16MB memory can be divided into 64 256KB memories, that is, the range of 0-63 (32.ICL is divided according to this); it can also be divided into 512 32KB memories.

One 16MB Flash block:  $16 * 1024 / 32 = 512$  (unit conversion, 1MB=1024KB), that is, the range is: 0x0000-0x01FF (0-511);

Four 16MB Flash blocks:  $16 * 4 * 1024 / 32 = 2048$ , that is, the range is: 0x0000-0x07FF (0-2047).

Update 32.ICL to calculate the start ID according to the 32KB data block sent:  $32 * 256 / 32 = 256 = 0x0100$ , that is, the first data packet of 32.ICL is placed at 0x0100.

2. Write the 32.ICL data into the first address of the variable space (RAM) 0x8000 for temporary storage.

One 32KB occupied RAM address calculation:  $32 \times 1024 / 2 = 16384$  (word) = 0x4000 (word); (1kB=1024bytes, 2bytes=1 word, conversion between decimal and hexadecimal).

$0x8000 + 0x4000 = 0xC000$ ; (RAM range is 0x0000-0xFFFF)

The above is a 32KB data block occupying the address range of 0x8000-0xC000 (the temporary storage address should not be used for other purposes, otherwise the data will be modified).

Since an instruction does not exceed 255 bytes at most, and the data part does not exceed 249 bytes, 240 bytes is used as the length of the data part (easy to calculate).

The 32KB data packet is divided into N instructions:  $32 \times 1024 / 240 = 136.5$  (32KB data conversion, a total of 137 instructions)

The 4.17KB data packet is divided into N instructions:  $4.17 \times 1024 / 240 = 17.8$  (4.17KB data conversion, a total of 18 instructions)

Interval address calculation for each instruction:  $240 / 2 = 120 = 0x78$

(1 variable address=2 bytes, add 0x78 to the start address of each instruction)

3. The starting address of 18 instructions in a 17KB packet is as follows:

0x8000 0x8078

0x80F0 0x8168

0x81E0 0x8258

0x82D0 0x8348

0x83C0 0x8438

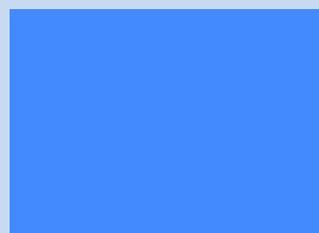
0x84B0 0x8528

0x85A0 0x8618

0x8690 0x8708

0x8780 0x87F8

Attached picture 32. The ICL command is as follows (choose a solid color image, the amount of data will be small, and the same part of the data can be copied):













|      |                                    |   |   |
|------|------------------------------------|---|---|
|      |                                    |   | <p>5. Caution</p> <p>1. Update the black screen:</p> <p>(1) Data sub-package error.</p> <p>(2) Data update location is not correct, such as the 14.bin file updated to 32.icl, it will empty the image data resulting in a black screen: 5A A5 0F 82 00 AA 5A 02 01 00 80 00 00 14 00 00 00 00 (e.g. RAM writes all 14 file data and writes to 32. (e.g. RAM writes all 14 files, and writing to position 32 will clear the picture data in Flash). 5A A5 0F 82 00 AA 5A 02 00 70 80 00 0014 00 00 00 00</p> <p>2. Update the splash screen:</p> <p>(1) A part of the image display screen: the last instruction data length check is 0xF3, if the data is less than 240 bytes, then the data length needs to be modified;</p> <p>(2) The data content part needs to be correct.</p>  |
| 0xB0 | Touch instruction access interface | W | <p>36</p> <p>0xB0: 0 x5 AA5 = enable accessing touch control interface once. <b>Clear after CPU operation.</b></p> <p>0xB1: Page ID of touch control.</p> <p>0xB2: <b>High byte:</b> touch control ID (set in DGUS II development software), 0x01-0xFF;</p> <p><b>Low byte:</b> touch control code, 0x00-0x7F.</p> <p>0xB3: Access mode</p> <p>0xB4-0xD3: data to modify of mode 0x02, 0x03.</p> <p>Mode 0x0000: <b>turn off</b> this touch control.</p> <p>Mode 0x0001: <b>turn on</b> this touch control.</p> <p>Mode 0x0002 : Read this touch control and write it to SRAM that 0xB4 pointing to.</p> <p>Mode 0 x0003 : update current touch control with data that 0 xB4 pointing to, the format and data length must be the same.</p> <p>Example command:</p> <p>Closes the 4 popup menu touch control in the routine, closing only part of the page. See 0xFC address for closing all touches.</p> <p>(1) Close command</p> <p>Close the popup menu on page 0, sequence 2: 5A A5 0B 82 00B0 5AA5 0000<br/>02 01 0000</p> <p>Close the pop-up menu on page 0, sequence 3: 5A A5 0B 82 00B0 5AA5 0000<br/>03 01 0000</p> <p>Close the pop-up menu on page 0, sequence 4: 5A A5 0B 82 00B0 5AA5 0000<br/>04 01 0000</p> <p>Close the pop-up menu on page 0, sequence 5: 5A A5 0B 82 00B0 5AA5 0000<br/>05 01 0000</p> <p>(2) Open after the above is closed</p> <p>Open the pop-up menu on page 0, sequence 2: 5A A5 0B 82 00B0 5AA5 0000<br/>02 01 0001</p> <p>Open the pop-up menu on page 0, sequence 3: 5A A5 0B 82 00B0 5AA5 0000<br/>03 01 0001</p> <p>Open the pop-up menu on page 0, sequence 4: 5A A5 0B 82 00B0 5AA5 0000<br/>04 01 0001</p> |



Open the pop-up menu on page 0, sequence 5: 5A A5 0B 82 00B0 5AA5 0000  
05 01 0001

(3)Note on "Clear after CPU operation"

The T5L refresh cycle is 20ms. When multiple instructions need to be closed, it takes one cycle for the CPU to clear and refresh before executing the next close instruction. That is, after the first instruction is sent, you can read the 0xB0 address to judge whether it is 0, if it is 0, execute the next instruction, and if it is not 0, do not execute the next instruction. You can use the 20ms cycle as the standard, **delay 20ms before sending the next shutdown command, otherwise there will be a situation where only one shutdown is disabled, and other commands are invalid**, for example, only a 5ms delay is required. At this time, one cycle is not refreshed, and the 0xB0 address is not cleared. If it is zero, if you continue to refresh the closing command in this cycle, other commands cannot be closed.

An example of the "clear after CPU operation" instruction is as follows:

Close popup menu for page 0 order number 2: 5A A5 0B 82 00B0 5AA5 0000  
02 01 0000

Read 0xB0 address: 5A A5 04 83 00B0 01;

Return: 5A A5 06 83 00B0 01 00 00; it has been cleared, and the next closing command can be executed at this time;

Read 0xB0 address: 5A A5 04 83 00B0 01;

Return: 5A A5 06 83 00B0 01 5A A5; not cleared, at this time, wait for the next close instruction;

(4)Command meaning

Close popup menu for page 0 order number 2: 5A A5 0B 82 00B0 5AA5 0000  
02 01 0000

0x5AA5: start command access;

0x0000: Turn off the touch on page 0;

0x02 : Turn off the No. 2 sorting touch on page 0;

0x01: No. 2 sorting touch on page 0 is a pop-up menu, the function key code is 0x01, and the key code value is shown in the list of touch functions;

0x0000: Disable command.

Precautions:

**The basic touch cannot be controlled to turn on/off. If there is a ranking number in the PC software, it does not need to be counted in the number of valid touches**, that is, the sorting ID needs to be subtracted from the number occupied by the front basic touch, such as 0000 page touch sequence 06 is the basic touch, 07 is the pop-up menu, then the command to close the pop-up menu is:

Correct: 5A A5 0B 82 00B0 0000 06 01 0000; touch sequence ID minus 1 invalid base touch.

Error: 5A A5 0B 82 00B0 0000 07 01 0000



If you need to change the specific attributes of the touch, you need to use mode 03 to replace the pre-written touch file starting with 0xb4 in the attribute configuration of the 13bin file.

(5) Touch ID + touch key code search

Touch ID:

Control view in the lower left corner - pull down and select touch control - sequence ID

**Bottom left control view - drop down to select touch control - order ID**

6.1 List of Touch Variables

| Number | Touch control code | Variable name          | User variable length (Word) | Description   |
|--------|--------------------|------------------------|-----------------------------|---|
| 01     | 00                 | Variable Data Input    | 1/2/4                       | Input integer, fixed decimal and other data into the specified variable storage space. Pop-up keyboard transparency can be set. Support configuration touch control.                      |
| 02     | 01                 | Popup Menu             | 1                           | Click to trigger a pop-up menu that returns the key code of the menu item. Popup menu transparency can be set.  |
| 03     | 02                 | Incremental Adjustment | 1                           | Click the button to perform +/- operation on the specified variable, and set the step size and upper and lower limits. Setting the 0- 1 range loop enables the column check box function. |
| 04     | 03                 | Drag Adjustment        | 1                           | Drag the slider to achieve variable data entry. The range can be set.   |
| 05     | 05                 | Return Key Code        | 1                           | Click the button to directly return the key value to the variable. Support bit variable return. Support press time threshold setting.   |

**Touch key code:**

6.1 List of Touch Variables

| Number | Touch control code | Variable name          | User variable length (Word) | Description   |
|--------|--------------------|------------------------|-----------------------------|---|
| 01     | 00                 | Variable Data Input    | 1/2/4                       | Input integer, fixed decimal and other data into the specified variable storage space. Pop-up keyboard transparency can be set. Support configuration touch control.                      |
| 02     | 01                 | Popup Menu             | 1                           | Click to trigger a pop-up menu that returns the key code of the menu item. Popup menu transparency can be set.  |
| 03     | 02                 | Incremental Adjustment | 1                           | Click the button to perform +/- operation on the specified variable, and set the step size and upper and lower limits. Setting the 0- 1 range loop enables the column check box function. |
| 04     | 03                 | Drag Adjustment        | 1                           | Drag the slider to achieve variable data entry. The range can be set.   |
| 05     | 05                 | Return Key Code        | 1                           | Click the button to directly return the key value to the variable. Support bit variable return. Support press time threshold setting.   |

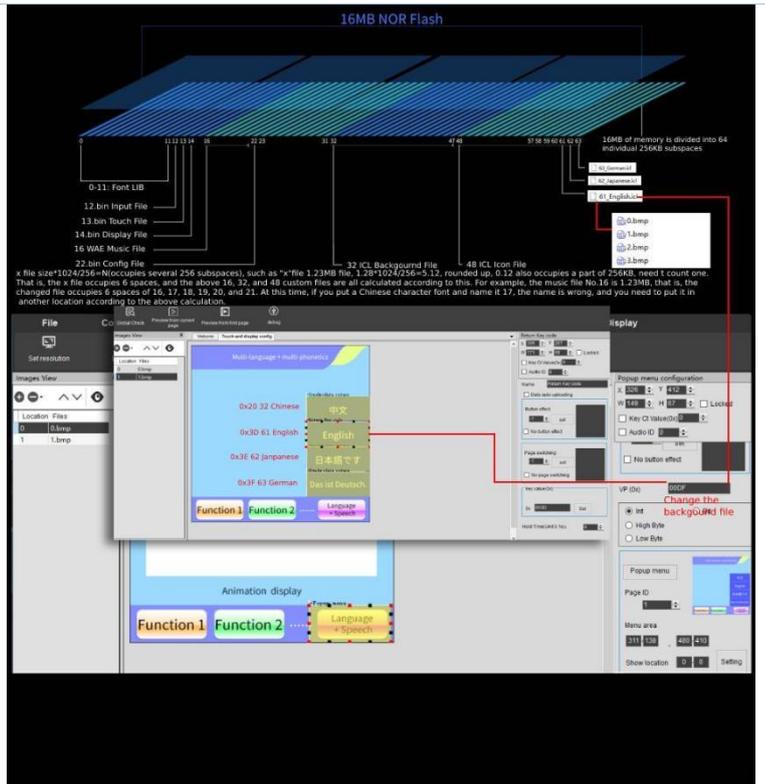


|      |                              |   |   |  |
|------|------------------------------|---|---|--|
| 0xD4 | TP operation simulation      | W | 5 | <p>0xD4: 0x5 AA5= enable the operation once, clear after operation.</p> <p>0xD5: press mode. 0x0001=press, 0x0002=release, 0x0003=keep pressing, 0x0004=touch (press + release)</p> <p>0xD6: X coordinate of press position.</p> <p>0xD7: Y coordinate of press position.</p> <p>After simulating mode 0x0001 and 0x0003, must simulate 0x0002.</p> <p>When the assembly touch function is running, x= 0xAA:KH y=0xA5: KL will directly return the key values KH and KL to the assembly touch.</p> <p>For example, when variables are input in assembly mode, the coordinates ( 0xAAF0, 0xA5 F0) will cause the input to end immediately.</p> <p>When the 13 touch file is designed with keying function, X coordinate = 0xFF: key code y coordinate = 0x0001 will trigger the corresponding keying function.</p> <p>E.g.</p> <p>5AA5 0B 82 00D4 5AA5 0004 00EE 008F</p> <p>0004 is click, press + lift, 00EE 008F (283,143) coordinates,</p> <p>Press mode 0x0001=press; 0x0002=release; 0x0003=continue pressing 0x0004=click</p> <p>After applying the simulated lift modes 0x0001 and 0x0003, there must be a simulated lift mode of 0x0002.</p> <p>( after " touch panel sound control" is enabled, touch panel operation simulation will also trigger the touch panel buzzer)</p> <p>(after " touch panel backlight standby control" is enabled, touch panel operation simulation will also wake up backlight)</p> |
| 0xD8 | Pointer icon overlay display |   | 4 | <p>0xD8_ H: 0x5A = enable the overlay display.</p> <p>0xD8_ L: the position of ICL file which the pointer icon is saved at.</p> <p>0xD9: the pointer icon ID.</p> <p>0xDA: the X coordinates of pointer icon.</p> <p>0xDB: the Y coordinates of pointer icon.</p> <p>The pointer icon is always displayed in the background filtering mode, and the background filtering intensity is fixed at 0x08.</p> <p>E.g.</p> <p>5A A5 0B 82 00D8 5A 2D 0001 0064 0064 call the 1 Icon of 45.ICL icon to display directly on the screen page position (100100).</p> <p>Note: After switching page, icon overlay will not disappear.</p> <p>write 00 to 0xD8_ H or call the blank position icon to make the icon disappear.</p> <p>The ID of the icon can be selected by the user to achieve the image display effect of "click" and "wait" effect.</p>  |
| 0xDC | Music WAE file switch        | W | 2 | <p>D3: 0x5A indicates that the music WAE file switching function is turned on, and it is initialized to 0x00 (off) after power-on.</p> <p>D2: Undefined, write 0x00.</p> <p>D1:D0: music WAE file ID.</p>  |





|      |                       |   |   |   |
|------|-----------------------|---|---|---|
| 0xDE | Background ICL Switch | W | 2 | <p><b>0xDE multi-language background picture ICL file switching:</b></p> <p>1. Application instructions</p> <p>By switching the ICL background pictures of different IDs, one-click switching of multiple languages can be realized. For example, to make 4 sets of pictures in different languages, the remaining 3 sets only need to generate ICL background images, called through the pop-up menu, 13/14 files share one project.</p> <p>2. Function setting</p> <p>(1) Write 0x5A into the high byte of address 0x00DE to enable the function.</p> <p>CFG file 0x08 address power-on specified background picture ICL position 0x20, 0x00DE power-on initialization no longer supports specification.</p> <p>①Touch on: You can use the key value to return, the address is set to 0x00DE, the key value is written to 0x5A, and the high byte is checked. When the high byte is selected, the low byte of the return key value is stored in the high byte address (VP_H) of the VP word address.</p> <p>②Command opening: 5A A5 05 82 00DE 5A00.</p> <p>(2) 0x00DF address is written to the background picture ICL file ID.</p> <p>Use the pop-up menu to select the switch, the address is set to 0x00DF, the basic touch key value on the menu corresponds to the ICL file ID of the picture in different languages, and the key value is a hexadecimal parameter.</p> <p>3. Background image</p> <p>(1) Name the picture material</p> <p>For example: Chinese image naming IDs are 0, 1, 2, 3; other language naming IDs also need to be 0, 1, 2, 3 for parallel replacement and calling.</p> <p>(2) ICL background image naming</p> <p>①The ICL file is stored in 16MB NORFLASH, and is called according to the ID, and the ID position is calculated as follows.</p> <p>Font position calculation: "x" file size*1024/256=N (occupies several 256KB subspaces), such as "x" file 1.28MB, <math>1.28*1024/256=5.12</math>, rounded up, 0.12 also occupies a part of 256KB, one needs to be counted, that is, the "x" file occupies 6 spaces, and 16, 32, and 48 custom files are calculated according to this.</p> <p>For example, the number 16 music file is 1.28MB, that is, the file occupies 6 spaces of 16, 17, 18, 19, 20, and 21. At this time, if you put a Chinese character font and name it 17, the name is wrong, and you need to put it in another location according to the above calculation.</p> <p>②ID+Chinese remarks, such as 61_English.ICL, add remarks for easy identification and setting.</p> |
|------|-----------------------|---|---|---|



**0xDE multi-language background image + ICL file switching of different language icons**

DEMO production is the same as above.

1. Application instructions

When DEMO is in multiple languages, it is used to switch control icons with language characters.

2. Function setting

DEMO production is the same as above. Different language folders have added control icons with language characters.

3. Control icons with language characters

(1) Icon distinction

This refers specifically to control icons with language characters, such as Chinese "on, off" 2 icons, English "on, off" 2 icons, Japanese "on, off" 2 icons, German "on, off" 2 icons, 4 groups of 8 are required, and they are respectively placed in the background picture folder of the same language.

The general icon is the same as the standard DEMO setting, just put it in the 48.ICL position.

(2) Icon naming

Control icons with language characters can be named starting from 100, IDs 0-99 are reserved for background pictures, so that background pictures can be added when making DEMO.

(3) Icon call

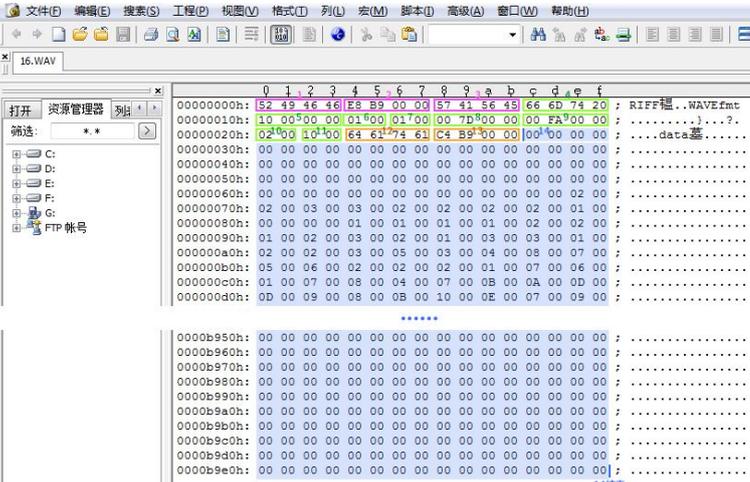
For the "Variable Icon Display" control, select icon library No. 0 as the icon file, which means to call N small icons (small pictures) with different IDs contained in the background icon library. At this time, it points to the 32.ICL background image library. Icon file "0" is just a trigger flag, not an icon library number 0

|      |                       |   |   |
|------|-----------------------|---|---|
| 0xDE | Background ICL Switch | W | 2 |
| 0xDC | Reserved              |   | 4 |

Undefined.



|           |  |     |    |   |
|-----------|--|-----|----|---|
| 0xE0      | Memory CRC check   | R/W | 2  | <p>D3: write 0 x5 A to enable the CRC checking once, clear after operation.</p> <p>D2: memory type selection</p> <p>0x00= font space ( 16Mbytes memory)</p> <p>0x02=DWIN OS code</p> <p>0x03= Nor flash database (LIB file).</p> <p>D1:D0: data interface.</p> <p>► Enable the CRC</p> <p>Font check mode: D1= start font ID (256 KB for each font); D0= the number of 4KB blocks checked, 0x00-0xFF.</p> <p>OS code check mode: D1:D0 = starting at 0x1000 , the byte length of OS code to be checked is 0x0001-0x7000.</p> <p>Nor flash database mode: D1:D0 = Nor flash database ID, fixed validation of 4 KB data each time.</p> <p>► After the check</p> <p>Returned CRC value.</p>  |
| 0xE2      | Export the monochrome bitmap of the specified area on the screen | R   | 6  | <p>0xE2_H: 0x5A A monochrome bitmap export is started and the CPU is cleared.</p> <p>0xE2_L: Brightness threshold when exporting bitmaps, 0x20-0xDF (0x3F recommended)</p> <p>0xE3: X coordinates of the upper left corner of the screen area;</p> <p>0xE4: Y coordinate of the upper left corner of the screen area;</p> <p>0xE5: X coordinates in the lower right corner of the screen area;</p> <p>0xE6: Y coordinate in the lower right corner of the screen area;</p> <p>0xE7: The start address of variable storage stored in the output bitmap must be an even number.</p> <p>Bitmap data stored in variable storage, aligned to lines according to MSB, double word pattern.</p>  |
| 0xE8-0xEF | Reserved   |     | 14 | Undefined   |
| 0xF0      | Interface of playing music flow data                             | W   | 4  | <p>D7: 0x5A= enable the music playback operation, clear after operation.</p> <p>D6: mode, 0x00=stop(clear the buffer), 0x01=suspend(reserve the buffer), 0x02=start.</p> <p>D5:D4: undefined, write 0x00.</p> <p>D3:D2: Variable memory address for storing music data, even.</p> <p>D1:D0: Music data word length, maximum 8KWords, even; data is 16bit integer format.</p> <p>This instruction is used to play online music files. First, send the music flow data to a variable address between 0x1000 and 0xFFFF.</p> <p>For example, the buffer address is 0x8000.</p> <p>1.WAV conversion</p> <p>Convert the stereo WAV file with the "To wav" tool to a single WAV file.</p> <p>Such as the three -dimensional WAV file format of the routine:</p> <p>Sampling rate: 44100Hz      Voice: stereo</p> <p>Bit depth: 16 bits</p> <p>Format: 16-bit integer of waveform audio.</p> |

|                    |   |          |          |  |
|--------------------|---|----------|----------|--|
| <p><b>0xF0</b></p> | <p>Interface of playing<br/>music flow data</p> | <p>W</p> | <p>4</p> | <p>The conversion single -channel WAV file format:</p> <p>Sampling rate: 32000Hz</p> <p>Voice: Mono</p> <p>Bit depth: 16 bits</p> <p>Format: 16 -bit integer of waveform audio.</p> <p>Mono is a left channel or right channel voice, which can be used in the files required by the above format.</p> <p>2. Copy WAV audio data to the initialization of file 22</p> <p>Set the start address of the VP variable of DGUS screen to 0x8000, copy an empty 22 file to the DWIN_SET folder, and copy the WAV data to the start address of the 22 file at 0x1000. The DGUS screen is in MSB mode, and the WAV and WAE files are in LSB mode, so when copying the data, the first byte of the data portion will not be copied, and the whole byte of the data will be shifted.</p>  <p>3. Download the 22 file to the DGUS screen</p> <p>Download online or SD card.</p> <p>4. Instruction playback</p> <p>Each time you send 8KWords data, 8KWords is 8*1024, that is, the variable address interval is 0x2000. 1 thousand bytes (KB) = 1024 bytes (bytes).</p> <p>5A A5 0B 82 00F0 5A02 0000 8000 2000</p> <p>5A A5 0B 82 00F0 5A02 0000 A000 2000</p> <p>5A A5 0B 82 00F0 5A02 0000 C000 2000</p> <p>5A A5 0B 82 00F0 5A02 0000 8000 2000</p> <p>Meaning: 0x5A A5: Frame head;</p> <p>0x0B: Data length, 0x82 starts, without 0x0B bytes;</p> <p>0x82: Write instructions;</p> <p>0x00F0: Music flow interface address;</p> <p>0x5A02: Write 0x5a to start a music playback, and the CPU operation is cleared.</p> |
|--------------------|---|----------|----------|--|



|                    |   |          |          |  |
|--------------------|---|----------|----------|--|
| <p><b>0xF0</b></p> | <p>Interface of playing music flow data</p>                                   | <p>W</p> | <p>4</p> | <p>0x02 = Play</p> <p>0x0000: Not defined, write 0x00.</p> <p>0x8000: Variable memory address to save music data, even.</p> <p>0x2000: The length of the music data that needs to be played, the maximum 8KWords, the even number; the audio data is 16bit integer format.</p> <p>5. Explanation</p> <p>(1) Effective voice data and all voice data</p> <p>0x2000, 10-proof is 8192, <math>47556/8192 \approx 5.81</math>, that is, 6 instructions are written with complete data. When playing, the first two instructions can be played with valid audio for 16kwords data. The latter part is noise and invalid data, which is related to voice pronunciation and ending time.</p> <p>(2) Playing is a click noise.</p> <p>DGUS MSB mode, WAV/WAE file is LSB mode, the first byte of the audio data is removed, and the data is staggered as a whole.</p>                      |
| <p><b>0xF4</b></p> | <p>Painting interface(The function has been removed after the V45 update)</p> | <p>W</p> | <p>8</p> | <p>D15: 0x5A = enable the drawing touch window.</p> <p>D14: operation mode, 0x00 = normal mode, 0x01 = initialization (clear after initialization).</p> <p>D13: parameter configuration.</p> <p>7-2 reserved, write 0.</p> <p>1-0 painting buff processing mode after page changing. 0x00 = close, 0x01= re- initialization, others = unchanged. D12: reserved, write 0x00.</p> <p>D11: line width, from 0x01 - 0x0F.</p> <p>D10:D8: painting color, D6=RED D5=GREEN D4=BLUE</p> <p>D7:D4 : the coordinate position of the window upper left corner(x,y).</p> <p>D3:D2 : the width of the pixel points window, must be divisible by4</p> <p>D1:D0 : the height of the pixel points window, must be divisible by 4</p> <p>The variable memory occupied by drawing touch window is 128KB(Double word address range 0x00:8000-0x00:FFFF) which is corresponding to 208*208 pixel.</p> |



|      |              |   |   |  |
|------|--------------|---|---|--|
|      |              |   |   | <p>This instruction is used after touching screen. Display the track of coordinates in the designated area of the screen, which is generally used in "handwriting drawing function".</p> <p>E.g.</p> <p>0x00 mode= normal mode</p> <ol style="list-style-type: none"> <li>1. Send drawing window instruction on a page:<br/>Tx:5A A5 13 82 00 F4 5A 00 00 00 02 FF 00 00 00 2C 00 BC 00 D0 00 D0</li> <li>2. Hand touch panel drawing graphics</li> <li>3. Switch page</li> <li>4. Switch back to the previous page, send the drawing window instruction again, and the characters will be displayed again. In the upper left corner of the window (44, 188), the coordinate point corresponds to the largest 208*</li> </ol> <p>208 pixel drawing window.</p> <p>0x01 mode: initialization mode</p> <ol style="list-style-type: none"> <li>1. Send a drawing window instruction on a page,<br/>Tx: 5A A5 13 82 00 F4 5A 01 00 00 02 FF 00 00 00 2C 00 BC 00 D0 00 D0</li> <li>2. Hand touch screen to draw graphics,</li> <li>3. Switch pages</li> <li>4. Switch back to the previous page, and send the drawing window instruction again, the characters are no longer displayed, and the blank is convenient for redrawing the graphics.</li> </ol> |
| 0xFC | DGUS_STOP_EN | W | 2 | <p>Writing 0x55 AA 5A A5 will stop the DGUS refresh, which is used to avoid the conflict of rewriting fonts when there is no backup area to upgrade.</p> <p>Write 0 x55 AA 5A AA will stop DGUS refresh, stop OS core running, used for ED4 download fixed occupancy variable buffer</p> <p>The buffer area is 0x8000 - 0xFFFF, to prevent DGUS refresh or OS core operation from rewriting variables to cause upgrade data errors. Writing <b>0x55 AA 5A 5A</b> will <b>stop the touch processing</b> and no longer respond to touch screen operation (0x0016 variable coordinates normal update).</p> <p>Instruction example:</p> <p>Close all touch: 5A A5 07 82 00FC <b>55AA 5A5A</b></p> <p>Open all touch: 5A A5 07 82 00FC <b>0000 0000</b></p> <p>Close part of the touch can refer to the 0xB0 instruction.</p>   |



|             |  |     |     |  |
|-------------|--|-----|-----|--|
| 0x FE       | UART1 high speed download                                    |     | 2   | 0x5AA5 = enable the high- speed download operation through UART1   |
| 0x100       | FSK bus interface  | R/W | 512 | FSK bus interface, supports up to 31 bus devices.  |
| 0x300-0x37F | Dynamic curve interface                                      | R/W | 128 | <p>0x300- 0x30F: state feedback for 8 channel curve buffers (read only advised), 2 words per channel, high word is the storage pointer location (0x000 - 0x07FF) where curve data is stored, and low word is the effective data length of curve buffer (0x0000-0x0800). Writing 0x0000 to the effective data length of the curve buffer will cause the curve unable display.</p> <p>0x310-0x311: start writing curve buffer data</p> <p>D3: D2: 0x5AA5 enable the writing curve buffer data operation once, and clear after operation.</p> <p>D1: the number of data blocks, 0x01-0x08.</p> <p>D0: undefined, write 0x00.</p> <p>0x312-0x37F: data block written to the curve buffer, which is 16 bits unsigned.</p> <p>Single data block: data channel ID (0x00-0x07) + data word length (0x01-0x6E) + data.</p> <p>With dynamic curve display enabled, start at 0x1000 and create a data buffer for each curve according to 2K words per channel.</p> <p>The CH0 buffer is 0x1000-0x17FF, the CH1 buffer is 0x1800-0x1FFF, and so on, the unused curve buffer zones can be used as user variables. Also users can directly overwrite the curve buffer data and then modify 0x300 - 0x30F corresponding storage pointer position and data length to ensure the correct display of the curve.</p> <p>Instruction example: See 7.4.1 Real-time Curve (Trend) display.</p> |
| 0x380       | The first channel curve data automatic reading configuration | W   | 2   | <p><b>D3:</b> 0x5A enables the curve data of the first channel to be automatically read from the variable space;</p> <p><b>D2:</b> The time interval of automatic reading, the unit is 10ms;</p> <p><b>D1:D0:</b> variable space address.</p> <p>(1) Function description: The data displayed by the curve can be configured to be read from the specified variable address when the OS is approved, which is convenient for the display of the curve of the variable (such as temperature) with a low refresh rate. It is necessary to upgrade the OS kernel program to V21 and above.</p> <p>The latest kernel link is in the compressed package of DGUS V60 software:<br/><a href="https://forums.dwin-global.com/index.php/forums/topic/notes-on-dwin-dgus-t5l-kernel-firmwarev60-upgrade/">https://forums.dwin-global.com/index.php/forums/topic/notes-on-dwin-dgus-t5l-kernel-firmwarev60-upgrade/</a></p> <p>(2) For details on the setting, negative number setting and curve address range on the curve DGUS software, see section 7.4.1 Real-time curve trend graph.</p> <p>(3) Correspondence between the 8 channels and the starting address of the system variable interface:</p> <p>Channel 0: 0x0380<br/>Channel 1: 0x0382<br/>Channel 2: 0x0384<br/>Channel 3: 0x0386<br/>Channel 4: 0x0388<br/>Channel 5: 0x038A<br/>Channel 6: 0x038C</p>              |

#### Channel 7: 0x038E

#### (4) Command example:

① Use channel 0 of the curve, the data range is 0 to 100, the variable address is 0x6000, and it is read once every 100ms.

The command is as follows:

Variable address write data 50: 5A A5 05 82 6000 0032

Read to channel 0 of curve: 5A A5 07 82 0380 5A0A 6000

There is only one data, and the curve will be displayed in a straight line.

Variable address write data 00: 5AA5 05 82 6000 0000

Variable address write data 50: 5AA5 05 82 6000 0032

Variable address write data 100: 5AA5 05 82 6000 0064

Read to channel 0 of curve: 5A A5 07 82 0380 5A0A 6000

The more data, the larger the range of the curve display.

② Use channel 1 of the curve, the data range is 0 to 100, the variable address is 0x6001, and it is read once every 100ms.

The command is as follows:

Variable address write data 50: 5AA5 05 82 6001 0032

Read curve to channel 1: 5AA 07 82 0382 5A0A 6001

③ Use the curve 7 channels, the data range is 0 to 100, the variable address is 0x6007, and it is read once every 100ms.

The command is as follows:

Variable address write data 50: 5AA5 05 82 6007 0032

Read curve to channel 7: 5AA 07 82 038C 5A0A 6007

④ Use channel 0 and channel 1 to read data of address 0x6000 and 0x6001 at the same time. The data range is 0 to 100, and the data is read once every 100ms. The default variable address has the correct value. The command is as follows:

5AA5 0B 82 0380 5A0A 6000 5A0A 6001 ( separate from the start channel address, continuous write operation can be completed by one instruction)

⑤ Use channel 0, channel 1, channel 2, channel 3, channel 4, channel 5, channel 6, channel 7, and read data of address 0x6000, 0x6001, 0x6002, 0x6003, 0x6004, 0x6005, 0x6006, 0x6007 at the same time, the data range is 0 To 100, read once every 100ms, the default variable address has the correct value, the instruction is as follows:

5AA5 23 82 0380 5A0A 6000 5A0A 6001 5A0A 6002 5A0A 6003 5A0A 6004 5A0A 6005 5A0A 6006 5A0A 6007

Write multiple consecutive addresses to change data from the initial variable address, and rewrite addresses 0x6000, 0x6001, 0x6002, 0x6003, 0x6004, 0x6005, 0x6006, and 0x6007 at the same time.

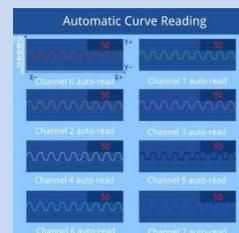
Write data 0, the instruction is as follows: 5AA5 13 82 6000 0000 0000 0000 0000 0000 0000 0000 0000

Write data 50, the instruction is as follows: 5AA5 13 82 6000 0032 0032 0032 0032 0032 0032 0032 0032

Write data 100, the instruction is as follows: 5AA5 13 82 6000 0064 0064 0064 0064 0064 0064 0064 0064

#### (5) Effect diagram

The above command is 8 horizontal lines when the data is not refreshed.



|               |                                 |     |      |  |
|---------------|---------------------------------|-----|------|--|
| 0x382         | Channel 2-8 configuration       | W   | 14   | The definition is the same as channel 1.   |
| 0x390         | Reserved                        | -   | 112  | Undefined, not available for user.   |
| 0x400-0x4FF   | Network communication interface | R/W | 256  | WiFi and other Internet communication equipment application control interface.   |
| 0x500 - 0x5BF | Multi-Media interface           | R/W | 192  | Multi-Media application interface, 0x500-0x57F digital Multi-Media interface, 0x580-0x5BF analog Multi-Media interface.  |
| 0x5C0-0x5FF   | External memory interface       | R/W | 64   | External memory interface ( such as U disk) read or write interface.   |
| 0x600-0xEFF   | Reserved                        |     | 2404 | Undefined. Not available for users.  |
| 0x0F00        | Variable change indication      | R   | 2    | After setting variables to change the automatic upload function, this function is enabled.<br>D3 =5A means variable change,<br>D2:D1= variable memory pointer,<br>D0 = variable length ( word).  |
|               |                                 |     |      | <p>E.g.</p> <p>It is used to judge whether the touch data is uploaded to the serial port, or whether there is a touch control operation change for uploading.</p> <p>Because the time of this state is very short, it can't be read with the serial port 0x83 instruction. It is recommended to read the value of D3 with DWIN OS and then judge.</p> <p>LDWR R0,0 F00H<br/>IJNE R10, 5 AH, TEXT; Judge the value of 0x5 A and execute it down, if not jump to TEXT</p> <p>Application example 2: It is used to judge whether the data of different addresses is changed, and OS can also be used for judgment processing.</p> <p>Send: 5A A5 04 83 0F 00 02<br/>Response: 5A A5 08 83 0F 00 02 00 11 00 01 means that the variable address 0x1100 has uploaded 1 word of data.</p> <p>Send: 5A A5 04 83 0F 00 02<br/>Response: 5A A5 08 83 0F 00 02 0010 00 02 means that the variable address 0x1000 has uploaded 2 words of data.</p> <p>Application example 3: 0x0F00 variable space. After writing 0x5a + VP + Len, the bottom layer will directly send the VP address. The data of the length of the Len word is sent from UART2 to the serial port in the format of the 0x83 instruction.</p> <p>Instruction format:</p> <p>Send: 5A A5 07 82 0F00 5A 10 02 03<br/>Answer: 5A A5 03 82 4F 4B</p> <p>5A A5 0A 83 1002 03 0000 0000</p> |
| 0xF02-0xFFFF  | Reserved                        |     | 254  | Undefined, User cannot use.  |

0x 1000-0xFFFF are available for users.

## 5.2 FSK bus interface

(0x0100-0x02FF variable address space)

| Definition                 | Address | Length | Description   |
|----------------------------|---------|--------|---|
| FSK interface control      | 0x100   | 2      | <p>D3: FSK bus control;<br/>0 x00 = Bus off, 0x5A= bus on, 0xA5 = reconfigure the bus once ( it will be changed to 0x5A automatically after the configuration is finished).</p> <p>D2: Bus configuration;<br/>.7 Broadcast instruction send1 = Start broadcast instruction transmit once and clear automatically after the transmit.<br/>.6-. 4 Double word length of broadcast instruction data, 0x01-0x07 .<br/>.3 -.0 Bus speed configuration, the lower the speed the longer the communication distance.<br/>0x00=100Kbytes/S; 0x01=200Kbytes/S; 0x02=400Kbytes/S; 0x03=600Kbytes/S.<br/>0x04=1 Mbytes/S; 0x05-0x0F: undefined.<br/>D1: Bus answer wait time, 0x01-0xFF,unit of 0.125mS.<br/>The DGUS screen acts as the master and polls the bus devices according to the configuration of the device interface. If no device answer data is received within the answer wait time, the next device will be polled.<br/>D0: Reserved, write 0x00.</p> |
| Broadcast instruction data | 0x101   | 14     | D27- D0: Broadcast instruction data, the broadcast instruction transmit has the highest priority and is delayed by a maximum of 1 DGUS cycle.   |
| Device interface 1         | 0x110   | 16     | <p>The 1st FSK device control interface, each device takes up fixed 16 words(32 bytes).<br/>Device interfaces can be dynamically assigned to different devices.<br/>D31, device switch.<br/>.7=1 device on, .7=0 device off.<br/>.6-.5 Reserved, write 0.<br/>.4-.0 indicates device ID, 0x00-0x1E.<br/>D30 - D29 : DGUS screen reads the double word address of the bus device data stored into the variable space, 0x0800-0xBFFF, 184 KB in size.<br/>The format of the data stored to the variable space is 0x5 AA5 + data word length + data.<br/>D28: Double word length of the data sent to the device, 0x01-0x07.<br/>D27- D0: Data sent to the device.</p>  |
| Device interface 2         | 0x120   | 16     | The 2rd FSK device control interface,   |
|                            | .....   |        |   |
| Device interface 63        | 0x2F0   | 16     | The 31th FSK device control interface,  |

### 5.2.1 FSK bus camera application

- Display

The camera video display is implemented by using the 0x5A08 variable data JPEG icon overlay display control, which matches the control's VP address (word address) with the corresponding camera's device interface storage address and reserves enough space.

For example, the video of bus camera 0# (ID=0x00) is displayed at (16,16), the display window size is 640\*480 pixels. The variable space used starts from 0x01:0000 and 40 KB of storage space is reserved.

```

0 1 2 3 4 5 6 7 8 9 a b c d e f
00004080h: 5A 08 FF FF 00 07 00 00 00 10 00 10 02 80 01 E0 ; Z. ....e.?
00004090h: FF 01 00 81 00 00 00 00 00 00 00 00 00 00 00 00 ; ..?.....

```

Corresponding display control configuration data



Use device interface 2 for 0# camera, storage location from address 0x010000, corresponding configuration instruction. 5A A5 06 82 0120 (device interface 2) 80(0# device on) 8000(double word address for data storage)

The storage space is calculated according to 1bit per pixel, e.g., 640\*480 resolution = 640\*480/8192= 37.5KB, which can be reserved as 40 KB.

● Configuration

This is achieved by defining instructions at locations D27-D0 in the device interface ( or broadcast instruction data), with D27 being the instruction and D26-D0 being the data. The instruction set is as follows .

| Instruction | Data  | Description  |
|-------------|---|--|
| 0x51        | D26: Camera configuration value.<br>.7 Left and right mirroring control, 0= normal<br>1=mirroring.<br>.6 IR illumination switch, 0=off 1=on.<br>.5-.0 Reserved, write 0.<br>D25:D24: Camera horizontal resolution, must be a multiple of 4.<br>D23: D22: Camera portrait resolution, must be a multiple of 4. | Camera configuration value, not saved after power down.<br>Resolution cannot exceed 64 KB memory space (800*600 is slightly larger).   |
| 0x52        | D26: Camera configuration value, same as 0x51 instruction.<br>D25 : D24: Camera horizontal resolution, same as 0 x51 instruction.<br>D2 3: D22: Camera portrait resolution, same as 0 x51 instruction.<br>D21: FSK bus interface speed (0x00-0x04)  | Camera power on parameters configuration, saved after power down.<br>The camera will reboot once after configuration.<br>The factory initial configuration is 00 02 80 01 E0 04. |
| 0x5D        | D26 : The number of register data groups to be modified, 0x01-0x0D.<br>D25 : D0: Register data, two bytes per group ( address:data), up to 13 groups.   | Camera debug instruction ( user should not use).   |
| 0x5F        | D26: Configured camera bus address, 0x00-0x1E.<br>Default 0x00.   | Camera bus address configuration, the camera will reboot once after configuration.   |

The data not used in the configuration is not written or not sent.

Configure the 0# camera resolution of device interface 2 to 640\*480, left and right mirroring.

The corresponding configuration instruction is as follow.

5AA5 0D 82 0120 80 8000 02 (transmit only 2 double words) 51(camera configuration instruction) 80(left and right mirroring) 02 80 01 E0



### 5.3 Network Interface

DWIN has developed a WiFi module "WiFi-10" and a cloud platform "DWIN Cloud" for AIOT applications.

WiFi-10 is specially designed for the DGUS development platform, and the data interfaces have been defined, which greatly reduces the development difficulty and shortens the development time for users.

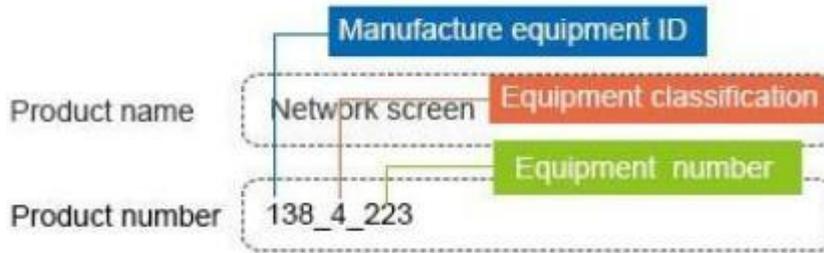
DWIN DGUS II platform (including T5 series and T5 L series) has open network interface with WiFi module. Through simple DGUS development, you can access DWIN cloud. Based on DWIN's cloud platform, customers can realize remote APP control, data analysis, equipment operation and other functions.

Note: The green filled part of the table indicates that the user can choose to modify it according to the actual situation.

For other parts, it is generally recommended to keep the default value and configure the 22.bin file corresponding to the double byte address.

| Definition               | Address | Length (Word) | Recommended value(hex) | Instructions  |
|--------------------------|---------|---------------|------------------------|---|
| Network Switch Interface | 0x400   | 1             | 5AA5                   | 0x5AA5 indicates that the network communication interface is enabled  |
|                          |         |               |                        | Recommended value description: Generally 0x5AA5 is written as a fixed value to 22 file. It is also possible to use the configuration button to return or incremental adjustment and other touch controls to write the trigger key value 0x5AA5 to address 0x400 to achieve the opening of the network interface.  |
| RAM ALARM                | 0x401   | 3             | 0000<br>0000<br>0000   | D5-D4: 0x5AA5 enable the RMA spatial data uploading to the server, clear after operation.<br>D3 - D2: RMA variable memory address to be uploaded.<br>D1-D0: the word length of the RMA variable memory to be uploaded. (Currently up to 4KByte)   |
|                          |         |               |                        | Recommended value description: It is used to transmit the data of the variable address on the screen to the Cloud server, which is usually applied to the Cloud call view of the alarm history information and other parameters. If you do not need to use this function, you can fill in 0000 by default in 22 file.   |
| Reserve                  | 0x404   | 12            | All 00                 | Reserved.   |
| Device Description       | 0x410   | 1             | 5A45                   | High byte: 0x5A indicates that the device description is valid.<br>Low byte: the encoding method and length of the device description text.<br>7-.6: encoding 0x00= UNICODE 0x01= GBK, GBK is recommended.<br>5-.0: describes the text length 0x00-0x34.  |
|                          | 0x411   | 2             |                        | Recommended value description: Write the configuration at a fixed value of the 22 file corresponding address, the encoding method GBK written here, the text length is 05.<br>Device manufacturer ID, assigned by DWIN factory, 0xFFFF: 0000-0xFFFF: FFFF segment is reserved. After the device is added to the DWIN cloud platform, it is automatically generated by the platform. |

|       |    |   |   |
|-------|----|---|---|
| 0x413 | 1  |   | <p>The classification of each manufacturer's equipment is according to the DWIN classification standard equipment classification.</p> <p>After the device is added to the DWIN cloud platform, it is automatically generated by the platform.</p>   |
| 0x414 | 2  |   | <p>The individual device number of each type of device.</p> <p>After the device is added to the DWIN cloud platform, it is automatically generated by the platform.</p>   |
| 0x416 | 26 | 44475553325F5<br>43555494431<br>(behind all 00) | <p>Device description text, up to 52 Bytes.</p> <p>Recommended value description:<br/>44475553325 F543555494431, corresponding to the ASCII code character DGUSII_T5UID1, has been written as a fixed value written in 22 file. The user can also make a text variable address association for display.</p> |



|                                     |       |   |        |   |
|-------------------------------------|-------|---|--------|---|
| Device Description<br>(RMA Mapping) | 0x430 | 1 | 012C   | <p>RMA automatically refreshes the server interval 0x0000-0xFFFF in 0.1 seconds.</p> <p>Recommended value description: 0x0000 indicates that automatic refresh is not required. 0x012C indicates that the automatic refresh interval is 30 seconds.</p> |
|                                     | 0x431 | 1 | 1000   | <p>RMA (mapped to the server's variable memory) read space start address, word address, out of bounds can not be read.</p> <p>Recommended value description: 0x1000 indicates that start reading space from 0x1000 address.</p>                         |
|                                     | 0x432 | 1 | 0004   | <p>RMA read space size in 128 Words with a maximum size of 2KBytes and 0x0000 indicates forbidden reading.</p> <p>Recommended value description: 0x0004 indicates that the read space size is 512 Words, which can be set to 0x01F4 at most.</p>        |
|                                     | 0x433 | 1 | 2000   | <p>RMA write space start address, can overlap with the read space, can not write out of bounds.</p> <p>Recommended value description: 0x2000 indicates that the space is read from the address of 0x2000.</p>   |
|                                     | 0x434 | 1 | 0004   | <p>RMA write space size in 128Words with a maximum size of 2KWords, 0x0000 indicates write forbidden.</p> <p>Recommended value description: 0x0004 indicates that the read space size is 512 Words, which can be set to 0x01 F4 at most.</p>            |
|                                     | 0x435 | 3 | All 00 | Reserved  |



|                                     |       |   |                     |   |
|-------------------------------------|-------|---|---------------------|---|
| Device Description (Remote Upgrade) | 0x438 | 1 | 5AA5                | <p>The remote upgrade interface is enabled. 0x5AA5 indicates that the device is enabled with the remote upgrade interface. The system will automatically detect the Buffer.</p> <p>Recommended value description: Generally 0x5AA5 is written as a fixed value to 22 file. It is also possible to use the configuration button return or incremental regulation and other touch controls to write the trigger key value 0x5AA5 to address 0x438 to achieve the switch.</p>  |
|                                     | 0x439 | 1 | 0064                | Remote upgrade packet timeout timer configuration in 0.1 second.  |
|                                     | 0x43A | 4 | 5A 00010 00000 0F00 | <p>The first remote upgrade space for the device ( aligned to 4 KB) is defined:</p> <p>D7 : 0x5A indicates that the remote upgrade space is enabled.</p> <p>D6 - D3 : 32 - bit start address of the upgradeable space ( lower 12 bits is 0), up to 4 GB.</p>  |
|                                     | 0x43E | 4 | 5A00 0000 0000 0000 | The second remote upgradeable space definition for the device   |
|                                     | 0x442 | 4 | All 00              | The third remote upgradeable space definition for the device  |
|                                     | 0x446 | 4 | All 00              | The fourth remote upgrade space definition for the device.  |
|                                     | 0x44A | 2 | 5A 00 08 B0 00      | <p>Remote upgrade Buffer interface definition:</p> <p>D3: 0x5 A indicates that the remote upgrade Buffer is valid.</p> <p>D2: upgrade mode 0x00=the communication side is responsible for verifying the data CRC, and the error frame informs the host to resend.</p> <p>D1: the number of buffers available for remote upgrades, 0x01-0x10 , up to 16.</p> <p>D0: the starting address of Buffer0 is high byte ( word address) and low address is 8bit 0x00.</p> <p>Each Buffer is fixed in 2304 words ( 0x900) space, which is arranged in the back row.</p> <p>A single Buffer definition ( the first 512 bytes is the control interface, the next 4KB is the data):</p> <p>D0:0x5A means to start the remote upgrade of this buffer once, and clear it after CPU processing.</p> <p>D1: Remote upgrade space selection, 0x00-0x03 , one of 4 remote upgrade spaces.</p> <p>D2-D5 : Write the destination address of the remote upgrade space. The lower 12 bits are 0 (aligned to 4KB).</p> <p>D6-D7: Data byte length, 0x0001-0x0FFF. D8-D9: Data CRC checksum.</p> <p>D10-D511: Reserved.</p> <p>D512: Data starts, up to 4096 bytes.</p> |
|                                     | 0x44C | 4 | All 00              | Reserved  |



|                                  |       |    |               |   |
|----------------------------------|-------|----|---------------|---|
| Device Description QR Code       | 0x450 | 48 |               | <p>Device QR code.</p> <p>The QR code is automatically generated according to the device description data of 0x411, 0x413, and 0x414, and the function of downloading the APP, binding the device, and paying attention to the public number can be implemented with the mobile phone.</p>  |
| Communication Device Description | 0x480 | 16 | User- defined | <p>D31: 0x5A indicates that the communication device description data is valid.</p> <p>D30 : The encoding method and length of the device description text.<br/>7-.6 Encode mode 0x00= UNICODE 0x01= GBK, GBK is recommended.</p> <p>5-.0 Description: The length of the text is 0x00-0x14 .</p> <p>D29: Communication device category 0x01= WiFi D28 :<br/>Communication device status, bit definition<br/>7 Equipment working status 0 = Configuration 1 =Normal;<br/>6-.3 undefined, write 0;</p> <p>2-.0 signal quality, 0x00-0x07 a total of 8 levels, 0x00 means the worst, 0x07 means the best.</p> <p>D27-D20: 8 Bytes communication device MAC address, low (D20) alignment.</p> <p>D19-D0: Description of the communication device in text format, up to 20 Bytes. Use hexadecimal to describe the information as the corresponding ASCII character.</p> <p>The user can refer to the DWIN example' s configuration at 22 to set 0x480 to write 0x5A 47 01 00<br/>0 x0482 can display the 8 Bytes communication device MAC address value through the HEX variable display control</p> |
| Communication Device             | 0x490 | 8  | User- defined | <p>Restart WiFi module is valid</p> <p>D15-D14 : The baud rate setting is reserved. Currently fixed at 921600 bps, it cannot be set.</p> <p>D13-D9: Reserved.</p> <p>D8: WiFi module switching Debug firmware 5A means valid.</p> <p>D7: D0: WiFi module type identification information Default "DWD100".</p>  |



|                |       |   |            |  |
|----------------|-------|---|------------|--|
| WiFi_ Config   | 0x498 | 4 |            | <p>D7: Writing 5A means starting the WiFi configure network, and the configure network is completed. The value is cleared to 0.</p> <p>D6: 0x5A means to start the network name and password to connect to the router (the network name and password are saved at 0x4B0)</p> <p>D7-D6, as long as it is started once, D7-D6 will be cleared at the same time after completion. The two Internet access modes cannot be started at the same time. If they are all 5A, the one-click configure network takes precedence.</p> <p>D5: Network time 5A means the screen comes with RTC (0x9C); 5B means RTC library RTC (0xF430)</p> <p>D4- D2: Reserved.</p> <p>D1: The WiFi module is automatically upgraded. 5A means enable.</p> <p>D0: Soft reset. 5A initiates reset and reset is cleared to 0.</p> <p>The user can design the configure button in the UI.<br/>Start manual configure network touch button (press to return 0x498 address write 0x005A)<br/>Start one-key configure network touch button (press to return 0x498 address write 0x5A00)</p> |
| Reserved       | 0x49C | 4 | All are 00 | Reserved   |
| Network Status | 0x4A0 | 8 |            | <p>D15-D14: WiFi version number.</p> <p>D13-D12: configure network status feedback</p> <p>00: unmatched network</p> <p>01: start configure network</p> <p>02: in the configure network</p> <p>03: configure network success</p> <p>04: configure network failure</p> <p>D11-D10: network connection status</p> <p>00: user name and password not obtained</p> <p>01: WiFi router connection is successful</p> <p>02: WiFi module self-upgrade</p>  |
|                |       |   |            | <p>03: connect to the server</p> <p>04: logged in to the server</p> <p>05: connected to the Cloud</p> <p>The user can directly display the interface variable, use the 0x4A0 address to display the WiFi version number with the data variable display control, and the 0x4A1 address uses the icon variable to make 5 small icons for displaying the communication status. The 0x4A2 address uses 6 small icons to display the network connection status.</p>   |
| State Machine  | 0x4A8 | 4 |            | <p>D7: state machine.</p> <p>D6: UART state machine.</p> <p>D3-D0: remain stacking space.</p>  |
| RTC            | 0x4AC | 4 |            | <p>D7: 5A means the time is valid.</p> <p>D6-D0: yy:mm:dd:ww (0-6) hh:mm:ss</p>  |



|                     |       |    |  |
|---------------------|-------|----|--|
| Network information | 0x4B0 | 32 | 0x4B0: SSID, the end must be 0xFF.                                       |
|                     |       |    | 0x4C0: SN, the end must be 0xFF.   |
|                     |       |    | Text ASCII input and display controls can be used for input and display. |

WiFi-10 actively reads 0x490 and other registers every 0.1s- 0.2s ( the register that DGUS issues instructions to WiFi-10).

WiFi-10 will write its status to the DGUS register every 3-5 seconds (the status of the WiFi module:0x480, 0x4A0, etc.).

After completing the configure operation on the DGUS screen or mobile app, WiFi-10 will automatically access the Internet and start running data synchronization.

The device description information is configured in the 22 initialization file 0x0800-0x09FF byte addresses corresponding contents. (The underlying program will automatically handle regardless of the 22 file initialization variable buffer function enabled or not).

Cloud platform website: <http://merchant.dwinhmi.com.cn/>

Server mapped memory debug interface website: <http://tools.dwinhmi.com.cn/>



## Chapter 6 Touch Variable Configuration (13.BIN)

The touch variable configuration is stored in the 13.BIN, which consists of touch instructions according to the control configurations.

Each touch instruction occupies a fixed storage space of 16, 32 or 48 bytes and consists of 6 parts, as shown below.

| Number | Definition | Length  | Definition  |
|--------|------------|---------|---|
| 1      | Pic_ID     | 2       | Page ID, the high 4 bit is the audio ID selection of this button, 0 x00 means without touch voice.  |
| 2      | TP_Area    | 8       | The touch button area coordinates the upper left corner (Xs, Ys) and the lower right corner (Xe, Ye).<br>Setting the button region outside the screen resolution simulates triggering touch-screen buttons in the system variable interface with specific key values.   |
| 3      | Pic_Next   | 2       | Target page ID after button pressing, 0xFF** indicates without page switching.  |
| 4      | Pic_On     | 2       | ID of the page where the effect picture is when the button is pressed, 0xFF* * indicates without pressing effect.   |
| 5      | TP_Code    | 2       | Touch control code:<br>0xFF** indicates an invalid key code.<br>0xFE** or 0xFD** indicates a touch button.<br>For example, 0 xFE00 indicates the touch screen digital input is enabled.<br>00 is the touch control code value, which can be named by referring to the touch control code number in the 6. 1 List of Touch Variables.<br>0xFE** function key can be set by CFG file 0x05.4 to upload or not.<br>The function button of 0xFD** always prevents automatic uploading after variable changes.<br>Other touch control codes are represented by ASCII.<br>For example, 0x0031 indicates key 1. |
| 6      | TP_FUN     | 0/16/32 | Touch control code:<br>When TP_Code= 0xFE**, this parameter is used to describe the touch button.   |



## 6.1 List of Touch Variables

| Number | Touch control code | Variable name          | User variable length (Word) | Description   |
|--------|--------------------|------------------------|-----------------------------|---|
| 01     | 00                 | Variable Data Input    | 1/2/4                       | Input integer, fixed decimal and other data into the specified variable storage space.<br>Pop- up keyboard transparency can be set.<br>Support configuration touch control.   |
| 02     | 01                 | Popup Menu             | 1                           | Click to trigger a pop-up menu that returns the key code of the menu item.<br>Popup menu transparency can be set.   |
| 03     | 02                 | Incremental Adjustment | 1                           | Click the button to perform +/- operation on the specified variable, and set the step size and upper and lower limits.<br>Setting the 0- 1 range loop enables the column check box function.  |
| 04     | 03                 | Drag Adjustment        | 1                           | Drag the slider to achieve variable data entry. The range can be set.   |
| 05     | 05                 | Return Key Code        | 1                           | Click the button to directly return the key value to the variable. Support bit variable return.<br>Support press time threshold setting.  |
| 06     | 06                 | ASCII Text Input       | Max 127                     | Input text characters in ASCII or GBK Chinese characters. Cursor movement and editing are supported.<br>You can set the (VP-1) position to store the input status and input length.<br>ASCII input mode pop- up keyboard transparency can be set.<br>The pop- up keyboard can be dynamically switched.  |
| 07     | 08                 | Synchrodata Return     | User defined                | Click the screen to return the data to the variable as specified.   |
| 08     | 09                 | Rotation Adjustment    | 1                           | Arc type drag adjustment.   |
| 09     | 0A                 | Sliding Adjustment     | 2                           | Slide the touch screen along the X or Y axis of the specified area to return the relative adjustment value in real time.<br>With the data window indicating the display of variables, you can achieve dynamic roll- character adjustment.<br>VP is reserved and returns data at (VP+1) position.  |
| 10     | 0B                 | Page Sliding           | None                        | Slide the touch screen in the X-axis direction of the specified area to achieve dynamic page dragging.<br>You can set the page switch target, area, the current page variable display will follow the drag.<br>If there are other touch controls on the page at the same time, and the whole page ( including drag and drop touch buttons) needs to be switched by gesture, you must set the touch priority of the page sliding to the highest. |
| 11     | 0C                 | Slide Icon Sel         | 4                           | Realize icon page sliding selection with 0x07 display variable (JPEG icon panning display).   |
| 12     | 0D                 | Bit Button             | 1                           | Click button to specify bit adjustment for specified variable.  |

The touch configuration file ( 13\*.bin) cannot exceed 256Kbytes.

You can dynamically turn on, off, or modify the specified touch instruction by UART or DWIN OS access to the touch instruction access interface defined in 0x00B0 variable space to achieve complex touch functions.



## 6.2 Variable Data Input

The variable data input control can change and adjust data by pressing the control area and popping up an input keyboard.

You can achieve touch screen simulation operation by this control with 0xD4 instruction. That means when the operation is not completed and the pop-up keyboard is in non-operation state, you can use 0xD4 instruction to exit the pop-up keyboard state after a specified time.

You can also use it to input integers, fixed decimals and other data into the specified variable storage space on the current page or the pop-up keyboard. And the pop-up keyboard transparency can be set.

### 6.2.1 Instruction Storage Format

| Address | Definition  | Length(byte) | Description  |
|---------|-------------|--------------|--|
| 0x00    | Pic_ID      | 2            | Page ID  |
| 0x02    | TP_Area     | 8            | Area of the Control: upper left corner ( Xs, Ys), lower right corner( Xe, Ye).   |
| 0x0A    | Pic_Next    | 2            | Target switch page, 0xFF** means no page switch  |
| 0x0C    | Pic_On      | 2            | Button press effect page, 0xFF** means no button press effect.   |
| 0x0E    | TP_Code     | 2            | 0 xFE00 , variable data input key code.  |
| 0x10    | 0x FE       | 1            | 0xFE   |
| 0x11    | VP          | 2            | Variable pointer   |
| 0x13    | V_Type      | 1            | Return variable type :<br>0x00=2bytes:<br>Integer: -32768 to 32767;<br>Unsigned integer: 0 to 65535<br>0x01=4bytes:<br>Long integer: -2147483648 to 2147483647<br>Unsigned long integer: 0 to 4294967295<br>0x02=* VP high byte, unsigned number: 0 to255<br>0x03=* VP low byte, unsigned number: 0 to255<br>0x04=8bytes, extra long integer: -9223372036854775808 to 9223372036854775807<br>0x05=Single-precision floating-point number (4 bytes) |
| 0x14    | N_Int       | 1            | Integer digits. For example, input 1234.56, and N_Int=0x04   |
| 0x15    | N_Dot       | 1            | Decimal digits. For example, input 1234.56, and N_Int=0x02   |
| 0x16    | (x, y)      | 4            | Display location: right alignment, (x, y) is the upper right coordinate of the string input cursor.<br>Undefined when using configurationkeyboard (KB_Source=0x0F)   |
| 0x1A    | Color       | 2            | Display color<br>Undefinedwhen using configurationkeyboard (KB_Source=0x0F)  |
| 0x1C    | Lib_ID      | 1            | ASCII Font position, default 0x00<br>Undefinedwhen using configuration keyboard (KB_Source=0x0 F)  |
| 0x1D    | Font_Hor    | 1            | Font size 0x00 means black, otherwise it is white<br>Undefinedwhen using configuration keyboard (KB_Source=0x0F)   |
| 0x1E    | Cusor_Color | 1            | Cursor color.<br>Undefinedwhen using configuration keyboard (KB_Source=0x0 F)  |



|      |                  |   |   |
|------|------------------|---|---|
| 0x1F | Hide_En          | 1 | 0x00 : The entered text is displayed as *; other values are displayed according to the entered content.   |
| 0x20 | 0xFE             | 1 | 0xFE  |
| 0x21 | KB_Source        | 1 | 0x00=Current page;<br>0x01=Other page;<br>0x0F= Assembly keyboard   |
| 0x22 | PIC_KB           | 2 | The page ID where the keyboard is located is valid only if KB_Source is not equal to 0x00.<br>Assembly keyboard (KB_Source=0x0F)=assembly function file   |
| 0x24 | AREA_KB          | 8 | Only the pop-up keyboard (KB_Source=0x01) mode is valid;<br>Keyboard area: upper left coordinate (Xs, Ys), lower right coordinate (Xe, Ye).   |
| 0x2C | AREA_KB_Position | 4 | Valid for pop-up keyboard or number entry configuration keyboard.<br>The upper-left coordinate of the keyboard display position on the current page.  |
| 0x30 | 0xFE             | 1 | Fixed value 0xFE.   |
| 0x31 | Limite_En        | 1 | 0xFF: Indicates that the input range limit is enabled, and the input out-of-bounds is invalid (equivalent to cancellation);   |
| 0x32 | V_min            | 4 | Input lower limit, 4 bytes (long or unsigned long).   |
| 0x36 | V_max            | 4 | Input upper limit, 4 bytes (long or unsigned long).   |
| 0x3A | Return_Set       | 1 | 0x5A: During input, load Return_Data to the Return_VP address to end the automatic recovery.<br>0x00: No data is loaded during input.<br>Load data function: mainly used for multi-parameter entry with display variable SP( description pointer)<br>The process is automatically marked, such as changing font color, size, enable a (bit) variable icon or invert an area.<br>It can also be used as a marker bit of the input process to meet special requirements with DWIN_OS development. |
| 0x3B | Return_VP        | 2 | The VP address where the data is loaded during entry.   |
| 0x3D | Return_Data      | 2 | Data loaded into Return_VP during entry.  |
| 0x3F | Layer_Gama       | 1 | When the keyboard or the numeric input configuration keyboard pops up, the background transparency can be set within 0x00~0xFF. 0x00 means 100% opaque.   |

**Note:** The valid key codes for digital input are 0x0030-0x0039, 0x002E(.), 0x002D(+/-), 0x00F0(Cancel), 0x00F1(OK), 0x00F2(Backspace).

### 6.2.2 Software Settings

**Variable Data Input**

- X 421, Y 165: Coordinates of the upper left corner of the touch area
- W 195, H 68: Touch area width and height
- Key value(0x) 0: Can be set by default
- Voice ID 0: Voice ID number

**Name** Data Input

- Automatic data upload: Checking means that the input is completed and the data is uploaded. At the same time, the 0X05.4 bit of the CFG file needs to be written to enable the data upload, and the 0XD5 bit can be written to 0X38, which means that the 22 initial value file is loaded, the data is uploaded, and the touch sound is turned on.
- Button effect: The display effect of pressing the touch area. For example, the current button is blue, and the button on the button effect page is dark blue. Press the button to see the sunken dark blue effect.
- Page switching: Can be set by default
- No page switching:

**Variable address(0x)** 5420

- Variable type: integer (word)
- Integer digits: 5: The number of integer digits displayed during entry.
- Decimal places: 0: The number of decimal places displayed during entry.
- Display position: 657, 182: The coordinates of the data display during entry. Enter the position of the upper right corner of the display area.
- Display color: 0000
- Font library location: 0: The size of font 0 is 4\*8-64\*128, dot matrix in X direction, the size is appropriate, and the color can be customized.
- Font size: 30: Direct display is more intuitive, and "" can be used for password input.
- Cursor color: Black
- Input display mode: Direct display
- Keyboard location: Other pages: To call the keyboard of the current page, you need to select the current page, and to call other keyboards, you need to select other pages.

**Keyboard settings**

- On the page: 30
- Keyboard area: 215, 158, 506, 533
- Display position: 401, 161: The coordinates of the upper left corner of the selected keyboard area.
- Background transparency: 0: It is recommended to enter the position of the upper right corner of the display area for the coordinates of the data display during the entry process.
- Enable range limit: Lower limit -100, Upper limit 100: The corresponding data keyboard page needs to be defined by the basic touch setting key value, and the key value is set under the touch attribute. Only basic touch controls can be placed on the pop-up area, and touch keys such as key value return and sliding page turning cannot be placed.
- Load data during entry: The coordinates of the lower right corner of the selected keyboard area.
- Data address: 8003: The coordinates of the upper corner of the pop-up keyboard on the current page. Too far to the right or incomplete display of the keyboard in the lower right corner.
- Download Data: 2047: The pop-up keyboard is similar to the transparency of frosted glass, with 0 being opaque.

**Integer (2 bytes)**

- Integer (2 bytes): occupy 1 address
- Long integer (4 bytes): occupy 2 addresses
- \*VP high byte: occupy 1 byte
- \*VP low byte: occupy 1 byte
- Long integer (8 bytes): occupy 4 addresses

**Additional Notes:**

- Variable storage space users can use address range arbitrarily: 0x1000-0xFFFF. Variable data entry and data type occupies 1/24 addresses. That is, non-associated control keys need to be spaced apart. System variable interface address range: 0x0000-0x0FFF (if the page switching address is 0x0084).
- It is mainly used to combine with the SP (description pointer) modification of the variable display to realize the automatic labeling of the multi-parameter input process. For example, modify the font color, size, start a (bit) variable icon or area inversion. 0x8003 is the data variable display description pointer 0x8000 address offset by 3 bits, used to change the data color. 2047 is the color value 0x07FF (fluorescent green). It needs to be converted into decimal system. If there is no requirement, it can be unchecked by default. For example, the data value loaded with the data address determines whether the keyboard is in the input state.



## 6.3 Popup Menu

### 6.3.1 Instruction Storage Format

| Address | Definition      | Length(byte) | Description  |
|---------|-----------------|--------------|--|
| 0x00    | Pic_ID          | 2            | Page ID  |
| 0x02    | TP_Area         | 8            | Area of the control: upper left corner ( Xs, Ys), lower right corner (Xe, Ye).   |
| 0x0A    | Pic-Next        | 2            | Switch target page. 0xFF** means no page switch.   |
| 0x0C    | Pic_On          | 2            | Button press effect page. 0xFF** means no button press effect.   |
| 0x0E    | TP_Code         | 2            | 0 xFE01 , popup window key code.   |
| 0x10    | 0xFE            | 1            | 0x FE  |
| 0x11    | VP              | 2            | Variable pointer   |
| 0x13    | VP_Mode         | 1            | 0 x00 : the returned key code is saved to VP ( integer)<br>0x01: the returned key code is saved to high byte of the VP(VP_H)<br>0x02: the returned key code is saved to low byte of the VP(VP_L)<br>0x10-0 x1F: Write the lowest bit of the returned key code ( 1bit) to specific bit of VP(0 x10 corresponds to VP.0; 0X1 F corresponds to VP.F ) |
| 0x14    | Pic_Menu        | 2            | Page ID of the menu  |
| 0x16    | Area_Menu       | 8            | Menu area: upper left coordinate (Xs, Ys), lower right coordinate (Xe, Ye).  |
| 0x1E    | Menu_Position_x | 2            | Upper left corner x- coordinate of the current page  |
| 0x20    | 0xFE            | 1            | 0x FE  |
| 0x21    | Menu_Position_Y | 2            | Upper left corner y- coordinate of the current page  |
| 0x23    | Translucent     | 1            | The transparency setting.<br>The background transparency can be set within 0x00~ 0xFF. 0x00 means 100% opaque.   |
| 0x24    | NULL            | 12           | Write 0x00   |

### 6.3.2 Software Settings

Open DGUS, and click touch control-popup menu. Set the control touch area, and configure functions in the menu on the right.

The screenshot shows the 'Pop-up menu configuration' interface with several sections and their corresponding settings:

- Pop-up menu configuration:**
  - X: 592, Y: 303 (Coordinates of the upper left corner of the touch area)
  - W: 87, H: 89 (Touch area width and height)
  - Key value(0x): 0 (Can be set by default)
  - Voice ID: 0 (Voice ID number)
- Name definition:**
  - Automatic data upload:  (Checking means that the input is completed and the data is uploaded. At the same time, the 0x05.4 bit of the CFG file is required to write 1 to start the data upload. The 0x05 bit can be written 0x38, which means that the 22 initial value file is loaded, the data is uploaded, and the touch sound is turned on.)
  - Button effect: -1 (Specify) (The display effect of pressing the touch area. For example, the current button is blue, and the button on the button effect page is dark blue. Press the button to see the sunken dark blue effect.)
  - No button effect:
  - Variable address(0X): 5642 (Variable storage space users can use address range arbitrarily: 0x1000-0xFFFF. The pop-up menu occupies 1 space address. That is, the non-associated control key addresses need to be spaced apart. System variable interface address range: 0x0000-0x0FFF (if the page switching address is 0x0084).)
  - Write variable word by word:  (Write by word length: write the entire variable address. Write bit control: use bit variable; write high and low words; Section: Write system variables such as backlight will be used.)
  - Write high byte:
  - Write low byte:
- Pop-up menu settings:**
  - On the page: 24 (The corresponding pop-up menu page needs to be defined by the basic touch setting key code, the key code range is 0x0000-0x00FF, among which 0x00FF is the cancel key code. On the pop-up area, you can only use basic touch controls, and you can't press the button to return, and you can't slide to turn pages, otherwise the keys will be invalid.)
- Menu area:**
  - 380 | 297 | 686 | 585 (The coordinates of the upper left corner of the pop-up menu area. / The coordinates of the lower right corner of the pop-up menu area.)
  - Display position: 380 | 299 (set-up) (The coordinates of the upper-left corner of the pop-up menu's display position on the current page.)
  - Background transparency: 0 (The pop-up area is similar to the transparency of the glass texture, and 0 is opaque.)



## 6.4 Incremental Adjustment

### 6.4.1 Instruction Storage Format

| Address | Definition  | Length(byte) | Description  |
|---------|-------------|--------------|--|
| 0x00    | Pic_ID      | 2            | Page ID  |
| 0x02    | TP_Area     | 8            | Area of the control: upper left corner ( Xs, Ys), lower right corner ( Xe, Ye).  |
| 0x0A    | Pic_Next    | 2            | Switch target page. 0xFF** means no page switch.   |
| 0x0C    | Pic_On      | 2            | Button press effect page. 0xFF** means no button press effect  |
| 0x0E    | TP_Code     | 2            | 0 xFE02, incremental adjustment key code   |
| 0x10    | 0xFE        | 1            | 0xFE   |
| 0x11    | VP          | 2            | Variable pointer, returned data is defined by VP_Mode  |
| 0x13    | VP_Mode     | 1            | 0x00: Adjust the VP(integer)<br>0x01: Adjust high byte of the VP( 1- byte unsigned integer, VP_H) 0x02: Adjust low byte of the VP( 1- byte unsigned integer, VP_L)<br>0x10-0 x1F: Adjust the specific bit of VP(0x10 corresponds to VP.0, 0x1F corresponds to VP.F), range:0- 1. |
| 0x14    | Adj_Mode    | 1            | 0x00= - - others=+   |
| 0x15    | Return_Mode | 1            | 0 x00 : Disabled. The value stops changing when max. or min. values are reached.<br>Other Values: Enabled. The value loops around the range when it reaches max.   |
| 0x16    | Adj_Step    | 2            | 0x0000-0x7FFF  |
| 0x18    | V_Min       | 2            | Min value accepted by the control:<br>2 -bytes integer(When VP_Mode is 0x01 or 0x02, only low byte is effective)   |
| 0x1A    | V_Max       | 2            | Max value accepted by the control:<br>2 -bytes integer(When VP_Mode is 0x01 or 0x02, only low byte is effective)   |
| 0x1C    | Key_Mode    | 1            | 0 x00: Continuous. The value is changed while the user hold sit.<br>0 x01: The value is changed once per touch.  |
| 0x1D    | NULL        | 3            | Write 0x00   |

## 6.4.2 Software Settings

Open DGUS, click touch control - incremental adjustment. Then set the control touch area, and configure the functions in the menu on the right.

**Incremental adjustment**

X 375 Y 148  
W 113 H 127

Key value(0x) 0  
 voice ID 0

name Incremental Adjustment

Automatic data upload

Button effect:  
-1 Specify  
 No button effect

Variable address(0X) 5012

Write variable word by word  Bit control  
 Write high byte  
 Write low byte

Adjustment mode translation ++

Over-limit handling method Cyclic regulation  
Adjust the step length 1  
Lower limit 0  
Upper limit 1

Button effect Only adjust once while holding down the key

Coordinates of the upper left corner of the touch area

Touch area width and height

Can be set by default

Voice ID number

Checking means that the input is completed and the data is uploaded. At the same time, the 0x05.4 bit of the CFG file is required to write 1 to start the data upload. The 0x05 bit can be written 0x38, which means that the 22 initial value file is loaded, the data is uploaded, and the touch sound is turned on.

The display effect of pressing the touch area. For example, the current button is blue, and the button on the button effect page is dark blue. Press the button to see the sunken dark blue effect.

Variable storage space users can use address range arbitrarily: 0x1000-0xFFFF. The pop-up menu occupies 1 space address. That is, the non-associated control key addresses need to be spaced apart. System variable interface address range: 0x0000-0x0FFF (if the page switching address is 0x0084).

Write by word length: write the entire variable address. Write bit control: use bit variable; write high and low words; Section: Write system variables such as backlight will be used.

When pressed, it is "+" to increase. Or "-" to decrease.

Cycle adjustment: After the data reaches the maximum or minimum value, you can continue to adjust.  
Stop adjustment: the adjustment will stop when the data reaches the maximum or minimum value.

For example, if the step size is set to 2, the value will increase by 2 if you click it once, such as 2/4/6/8 change. The upper and lower limits are the maximum and minimum data ranges. The upper and lower limits are set to 0/1 and can be multiple-selected, 0 is off and 1 is on.

Adjust once: Press and hold the touch data to change only once.  
Continuous adjustment: Press and hold the touch data to continuously change.

## 6.5 Drag Adjustment

Drag adjustment is to change and adjust the data pattern by dragging the slider.

The advantage is that it is intuitive, fast, and the parameters do not cross the boundary.

In combination with the "Data Variables" control, the data can be dragged for precise display and the scale range can be set.

### 6.5.1 Instruction Storage Format

| Address | Definition | Length | Description   |
|---------|------------|--------|---|
| 0x00    | Pic_ID     | 2      | Page ID.  |
| 0x02    | TP_Area    | 8      | Area of the control: upper left corner ( Xs, Ys), lower right corner (Xe, Ye).  |
| 0x0A    | Pic_Next   | 2      | Switch target page. 0xFF** means no page switch.  |
| 0x0C    | Pic_On     | 2      | Button press effect page. 0xFF** means no button press effect   |
| 0x0E    | TP_Code    | 2      | 0xFE03  |
| 0x10    | 0xFE       | 1      | 0x FE   |
| 0x11    | VP         | 2      | Variable pointer.   |
| 0x13    | Adj_Mode   | 1      | The high 4 bit defines the data return format: 0<br>x0*= Adjust VP word address ( integer number);<br>0x1 *= Adjust the high byte address of the VP word address (1byte unsigned number, VP_H);<br>0x2 * = Adjust the low byte address of the VP word address (1byte unsigned number, VP_L).<br>The low 4 bit defines the drag mode:<br>0x0=horizontal drag; 0x1=longitudinal drag. |
| 0x14    | Area_Adj   | 8      | Effective adjustment area: ( Xs, Ys) (Xe, Ye), which must be the same as the touch area.  |
| 0x1C    | V_Begin    | 2      | The return value corresponding to the starting position, an integer.  |
| 0x1E    | V_End      | 2      | The return value corresponding to the end position, an integer.   |

## 6.5.2 Software Settings

|   |   |
|---|---|
| <p><b>Drag Adjustment</b></p> <p>X 78 Y 398</p> <p>W 558 H 87</p> <p><input type="checkbox"/> Key value(0x) 0</p> <p><input type="checkbox"/> voice ID 0</p> <p>Name definition Drag Adjustment</p> <p><input checked="" type="checkbox"/> Automatic data upload</p> <p>Variable address(0x) 0082</p> <p>Drag method</p> <p>Data return format Adjust high byte</p> <p>Drag method Drag horizontally</p> <p>The return value corresponding to the starting position 1</p> <p>The return value corresponding to the end position 100</p> | <p>Coordinates of the upper left corner of the touch area</p> <p>Touch area width and height</p> <p>Can be set by default</p> <p>Voice ID number</p> <p>Checking means that the input is completed to upload the data, and the 0x05.4 bit of the CFG file is required to write 1 to start the data upload. Bit 0x05 can be written as 0x39, which means loading 22 initial value files, uploading data, and turning on touch audio.</p> <p>Variable storage space users can use address range arbitrarily: 0x1000-0xFFFF. The pop-up menu occupies 1 space address. That is, the non-associated control key addresses need to be spaced apart. System variable interface address range: 0x0000-0x0FFF (if the page switching address is 0x0084).</p> <p>Write by word length: write the entire variable address.<br/>Write bit control: use of bit variables;<br/>Write high and low bytes: it will be used for writing system variables such as writing backlight. Pointing to the high byte of 0x0082 address here is used to drag to adjust the height of the</p> <p>The minimum and maximum values of the adjusted data are both 0 and cannot be dragged.</p> |
|---|---|



## 6.6 Return Key Code

### 6.6.1 Instruction Storage Format

The return key code control is to return the key value to the variable by pressing the control area, and supports the return of bit variables. Users can perform corresponding logical operations based on the returned key values.

| Address | Definition | Length(byte) | Description  |
|---------|------------|--------------|--|
| 0x00    | Pic_ID     | 2            | Page ID  |
| 0x02    | TP_Area    | 8            | Area of the control: upper left corner (Xs, Ys), lower right corner (Xe, Ye).  |
| 0x0A    | Pic_Next   | 2            | Switch target page. 0xFF** means no page switch.   |
| 0x0C    | Pic_On     | 2            | Button press effect page. 0xFF** means no button press effect.   |
| 0x0E    | TP_Code    | 2            | 0xFE05   |
| 0x10    | 0xFE       | 1            | 0x FE  |
| 0x11    | VP         | 2            | Variable pointer   |
| 0x13    | TP_Mode    | 1            | 0x00: the returned key code is saved to VP (integer)<br>0x01: the returned key code is saved to high byte of the VP(VP_H)<br>0x02: the returned key code is saved to low byte of the VP(VP_L)<br>0x10-0 x1F: Write the lowest bit of the returned key code (1bit) to specific bit of VP(0x10 corresponds to VP.0 ; 0X1F corresponds to VP.F) |
| 0x14    | Key_Code   | 2            | Returned key code  |
| 0x16    | Hold_Time  | 1            | Unit of 0.1s. There will be a response when the press time exceed the Hold_Time.   |
| 0x17    | NULL       | 10           | Write 0x00   |

### 6.6.2 Software Settings

Open DGUS, and click touch control – return key code. Then set the control area, and configure the button effect, page switching, key value, variable address, etc.

The screenshot shows the 'Return Key Value' configuration window in DGUS. It includes sections for 'Return Key Value' (with X, Y, W, H coordinates and key/voice ID), 'Name definition' (Return Key code), 'Automatic data upload' (checked), 'Button effect' (set to 1), 'Page switching' (set to 1), 'Key value(0x)' (set to 0001), 'Pressing time' (set to 50), and 'Variable address(0x)' (set to 5436). Callouts explain the meaning of these settings: coordinates, dimensions, default values, voice ID, data upload requirements, button effects, page switching IDs, key value ranges, pressing time, variable address ranges, and write control options.

Return data format:

**5AA5 06 83 5436 01 0001**

0x5A: 5A Frame header;

0x06: Data length;

0x83: Read instruction;

0x5436: VP;

0x01: word length of the returned data;

0x0001: returned key value of 0001.



### 6.7 Text Input

Text Input includes ASCII and GBK Chinese characters. You can move the cursor and edit during input. The input status and input length saved at (VP-1) address can be set.

The transparency of ASCII input keyboard is adjustable.

The keyboard key code is as the table below.

Key codes are the key values defined for basic keyboard operations. The lower byte of the two-byte key code indicates the normal key code, and the higher byte indicates the uppercase key code.

For example, 0x61 corresponds to a, 0x41 to A, and 0x31 to 1.

Besides, the key code must be smaller than 0x80 (ASCII code). 0x0D key code input will be automatically converted to 0x0D 0x0A; 0x00 and 0xFF key codes are disabled.

| Key Code | Normal | Capital |
|----------|--------|---------|----------|--------|---------|----------|--------|---------|----------|--------|---------|
| 0x7E60   | `      | ~       | 0x5171   | q      | Q       | 0x4161   | a      | A       | 0x5A7A   | z      | Z       |
| 0x2131   | 1      | !       | 0x5777   | w      | W       | 0x5373   | s      | S       | 0x5878   | s      | S       |
| 0x4032   | 2      | @       | 0x4565   | e      | E       | 0x4464   | d      | D       | 0x4363   | c      | C       |
| 0x2333   | 3      | #       | 0x5272   | r      | R       | 0x4666   | f      | F       | 0x5676   | v      | V       |
| 0x2434   | 4      | \$      | 0x5474   | t      | T       | 0x4767   | g      | G       | 0x4262   | b      | B       |
| 0x2535   | 5      | %       | 0x5979   | y      | Y       | 0x4868   | h      | H       | 0x4E6E   | n      | N       |
| 0x5E36   | 6      | ^       | 0x5575   | u      | U       | 0x4A6A   | j      | J       | 0x4D6D   | m      | M       |
| 0x2637   | 7      | &       | 0x4969   | i      | I       | 0x4B6B   | k      | K       | 0x3C2C   | ,      | <       |
| 0x2A38   | 8      | *       | 0x4F6F   | o      | O       | 0x4C6C   | l      | L       | 0x3E2E   | .      | >       |
| 0x2839   | 9      | (       | 0x5070   | p      | P       | 0x3A3B   | ;      | :       | 0x3F2F   | /      | ?       |
| 0x2930   | 0      | )       | 0x7B5B   | [      | {       | 0x2227   | ,      | “       | 0x2020   | SP     | SP      |
| 0x5F2D   | -      | _       | 0x7D5D   | ]      | }       | 0x0D0D   | Enter  | Enter   |          |        |         |
| 0x2B30   | =      | +       | 0x7C5C   | \      |         |          |        |         |          |        |         |

Table 6.7 Keyboard function key code definition



| Key Code | Function          | Description   |
|----------|-------------------|---|
| 0x00F0   | Cancel            | Cancel input code return, not affecting the variable data.  |
| 0x00F1   | Return            | Confirm input code return, and the input text is saved to the specified VP.   |
| 0x00F2   | Backspace         | Delete one character( backspace) .  |
| 0x00F3   | Delete            | Delete one character backwards.   |
| 0x00F4   | CapsLock          | Caps lock. <b>If enabled, the corresponding button must define the effect of the button press, i. e. there must be a press effect page in the project .</b> |
| 0x00F7   | Left              | Move the cursor forward by one character; it is used to turn pages in GBK Chinese character input.  |
| 0x00F8   | Right             | Move the cursor back by one character; it is used to turn pages in GBK Chinese character input.   |
| 0x00F9   | Picture_KB_Change | Used to switch between different keyboard backgrounds in order when they are not on the current page.   |
| 0x00FA   |                   | 0 x00F9: 2 background pages, PIC_KB and PIC_KB+ 1.<br>0 x00FA: 3 background pages, PIC_KB, PIC_KB+ 1, PIC_KB+2.   |

In case of using the keyboard (keycode in the 0x4 F register) for text input, if you use the CapsLock key, please define the button animation in the area where you want to indicate "CapsLock"; after this, when you press the CapsLock key, the area icon of " CapsLock" will be displayed on the screen automatically.



## 6.7.1 ASCII Text Input

### 6.7.1.1 Instruction Storage Format

| Address | Definition       | Length(byte) | Description  |
|---------|------------------|--------------|--|
| 0x00    | Pic_ID           | 2            | Page ID  |
| 0x02    | TP_Area          | 8            | Control area: upper left corner ( Xs, Ys), lower right corner (Xe, Ye).  |
| 0x0A    | Pic_Next         | 2            | Switch target page. 0xFF** means no page switch.   |
| 0x0C    | Pic_On           | 2            | Button press effect page. 0xFF** means no button press effect.   |
| 0x0E    | TP_Code          | 2            | 0xFE06   |
| 0x10    | 0xFE             | 1            | 0x FE  |
| 0x11    | VP               | 2            | Variable pointer   |
| 0x13    | VP_Len_Max       | 1            | Max text length, in words (two characters for each word) . Range: [0x01,0x7B].<br>When the text is saved in the specific VP, 0xFFFF will be added as terminator at the ending of the text. Thus the max variable space is actually VP_Len_Max+1.   |
| 0x14    | Scan_Mode        | 1            | Input mode.<br>0 x00: re-input; 0x01: modify existing text.  |
| 0x15    | Lib_ID           | 1            | Index in the FLASH memory of the ASCII Font to use.<br>0x00=default.   |
| 0x16    | Font_Hor         | 1            | Font width, in pixels.   |
| 0x17    | Font_Ver         | 1            | Font height, in pixels. When "Lib_ID" = 0x00 , it must be twice the width.   |
| 0x18    | Cursor_Color     | 1            | Color of the input cursor. 0x00: Black<br>Other Values: White.   |
| 0x19    | Color            | 2            | Text color.  |
| 0x1B    | Scan_Area_Start  | 4            | Upper-left coordinates of the area where the text will be displayed: (Xs, Ys).   |
| 0x1F    | Scan_Return_Mode | 1            | 0 x55= save the input end tag and valid data length at the * (vp-1) position.<br>* (vp-1) high byte, input end mark: 0x5A means input end, 0x00 means idle or input state.<br>* (vp-1) low byte, effective input data length, in byte.<br>0x00= does not return the input end tag and data length. |
| 0x20    | 0xFE             | 1            | 0x FE  |
| 0x21    | Scan_Area_End    | 4            | Lower-right coordinates of the area where the text will be displayed: (Xe, Ye).  |
| 0x25    | KB_Source        | 1            | Keyboard image page settings.<br>0x00: Keyboard on current page.<br>Other Values: Keyboard on another page.  |



|             |                  |   |  |
|-------------|------------------|---|--|
| <b>0x26</b> | PIC_KB           | 2 | The page where the keyboard is located ( valid when the keyboard is not on the current page)   |
| <b>0x28</b> | Area_KB          | 8 | Keyboard area coordinates: upper left corner ( Xs, Ys), lower right corner( Xe, Ye).           |
| <b>0x30</b> | 0xFE             | 1 | 0xFE   |
| <b>0x31</b> | AREA_KB_Position | 4 | Upper- left coordinates of the keyboard. ( valid when the keyboard is not on the current page) |
| <b>0x35</b> | Display_EN       | 1 | 0x00= Normal display;<br>0x01=Characters displayed as asterisks(*)                             |
| <b>0x36</b> | Layer_Gama       | 1 | The background transparency can be set within 0 x00 ~ 0xFF.<br>0 x00 means 100% opaque.        |
| <b>0x37</b> | NULL             | 9 | Write 0x00   |

### 6.7.1.2 Software Setting

**ASCII input**

- X: 404, Y: 145 → Coordinates of the upper left corner of the touch area
- W: 207, H: 210 → Touch area width and height
- Key value(0x): 0 → Can be set by default
- voice ID: 0 → Voice ID number

**Name definition** ASCII Text Input

- Automatic data upload → Checking means that the input is completed and the data is uploaded. At the same time, the 0×05.4 bit of the CFG file is required to write 1 to start the data upload. The 0×05 bit can be written 0×38, which means that the 22 initial value file is loaded, the data is uploaded, and the touch sound is turned on.
- Button effect: -1 → The display effect of pressing the touch area. For example, the current button is blue, and the button on the button effect page is dark blue. Press the button to see the sunken dark blue effect.
- No button effect
- Page switching: -1 → The ID of the switched page can be set by default
- NO page switching

**Variable address(0x)** 5240 → Variable storage space users can use address range arbitrarily: 0×1000-0×FFFF. According to the input character length occupies the corresponding length address, the maximum is 127 words (Word). The VP\_Len\_Max+1 address is occupied as the 0×FFFF terminator, therefore, one more address needs to be reserved. For example, 5 addresses are actually needed, and because the 0×FFFF terminator occupies one more address, 6 addresses need to be reserved. System variable interface address range: 0×0000-0×0FFF (if the page switching address is 0×0084) For example, 5 addresses are actually needed, and because the 0×FFFF terminator occupies one more address, 6 addresses need to be reserved.

**Text length** 50 → Maximum length of text variable, number of words (not bytes), range 0×01-0×7F

**Entry mode** Re-enter → Re-enter: When entering again, the previously entered characters will be cleared. Modify the text: When entering again, the previously entered characters still exist, and the previously entered characters need to be deleted manually.

**Font library location** 0 → ASCII characters use font size 0 by default, and the X,Y dot matrix double relationship.

**Number of dots in X direction** 24 (range 4-255)

**Number of dots in Y direction** 48 (range 4-255)

**Cursor color** black → Cursor black or white

**Text color** 0000 → The display color of the text during the input process.

Input status return

**Input display area** 56, 191, 669, 294 → set up

**Input process display mode** Normal display → \*\* Can be used for password input.

**Keyboard location** Other pages

**Keyboard settings**

- On the page: 10
- Keyboard area: 29, 155, 692, 590 → The coordinates of the upper left corner can be set consistent
- Display position: 32, 167 → set-up
- Background transparency: 0 → The called keyboard displays the coordinates in the upper left corner of the current page. Transparency: 0 opacity; similar to the transparency of glass texture.

**Pop up keyboard page**

The coordinates of the upper left corner of the so-called keyboard are (Xs, Ys)

Enter the coordinates of the upper left corner of the display area (Xs, Ys)

Enter the coordinates of the lower right corner of the display area (Xs, Ys)

The coordinates of the lower right corner of the called keyboard are (Xs, Ys)

**Keypads**

Keypads interface showing key definitions and a virtual keyboard.

Key value(0x): Full key definition: 0x0100 → Set

The called keyboard page uses basic touch control to define the key value. Note that the key value of data such as 1 should be set to 0×2131. You can check the keyboard, refer to the routine, or refer to the ASCII key code table setting



## 6.7.2 GBK Input

GBK input is used to input Chinese character text.

Open DGUS, and click touch control - GBK input. Then set the control area, and configure functions in the menu on the right.

GBK input can be combined with the text display control to display entered characters.

### 6.7.2.1 Instruction Storage Format

| Address | Definition      | Length (byte) | Description   |
|---------|-----------------|---------------|---|
| 0x00    | Pic_ID          | 2             | Page ID   |
| 0x02    | TP_Area         | 8             | Area of the control: upper left corner ( Xs, Ys), lower right corner ( Xe, Ye).   |
| 0x0A    | Pic_Next        | 2             | Switch target page. 0xFF** means no page switch.  |
| 0x0C    | Pic_On          | 2             | Button press effect page. 0xFF** means no button press effect   |
| 0x0E    | TP_Code         | 2             | 0xFE06 , GBK input key code.  |
| 0x10    | 0xFE            | 1             | 0xFE  |
| 0x11    | VP              | 2             | Variable pointer  |
| 0x13    | VP_Len_Max      | 1             | Max text length, in words (two characters for each word). Range: [0x01,0x7B].<br>When the text is saved in the specific VP, it will add the 0xFFFF as terminator automatically at the ending of the text. Thus the variable space is actually VP_Len_Max+1.   |
| 0x14    | Scan_Mode       | 1             | Input mode.<br>0x00: re-input, 0x01: modify existing text.  |
| 0x15    | Lib_GBK1        | 1             | Index in the FLASH memory of the GBK Font to use after entry. 0x00= default.  |
| 0x16    | Lib_GBK2        | 1             | Index in the FLASH memory of the GBK Font to use before entry.  |
| 0x17    | Font_Scale1     | 1             | Lib_GBK1 font size, lattice number  |
| 0x18    | Font_Scale2     | 1             | Lib_GBK2 font size, lattice number  |
| 0x19    | Cursor_Color    | 1             | Color of the input cursor.<br>0x00: Black<br>Other Values: White  |
| 0x1A    | Color0          | 2             | Text color after entry  |
| 0x1C    | Color1          | 2             | Text color during entry   |
| 0x1E    | PY_Display_Mode | 1             | The display of pinyin hints and corresponding Chinese characters during entry.<br>* 0x00 = pinyin hints are displayed on the top and the corresponding Chinese characters are displayed on a separate line at the bottom.<br>Pinyin hints and Chinese characters are displayed left- aligned, with Scan_Dis line spacing.<br>* 0x01 = Pinyin hints are displayed on the left, and the corresponding Chinese characters are displayed on the right.<br>The starting display x position of the Chinese character is: Scan1 _Area_Start+ 3 x Font_Scale2+Scan_Dis. |



|             |                  |   |   |
|-------------|------------------|---|---|
| <b>0x1F</b> | Scan_Return_Mode | 1 | 0xAA=Save the input end mark and valid data length at *(VP-1).<br>*(VP-1) high byte, end of input mark: 0x5A means end of input, 0x00 means input is still in progress.<br>*(VP-1) low byte, valid input data length in byte.<br>* 0xFF= Not return the input end mark and data length. |
| <b>0x20</b> | 0xFE             | 1 | 0xFE  |
| <b>0x21</b> | Scan0_Area_Start | 4 | Upper- left coordinates of the area where the text will be displayed: ( Xs, Ys).  |
| <b>0x25</b> | Scan0_Area_End   | 4 | Lower- right coordinates of the area where the text will be displayed: ( Xe, Ye).   |
| <b>0x29</b> | Scan1_Area_Start | 4 | Coordinates of the upper left corner of the text display area of the pinyin prompt during entry.  |
| <b>0x2D</b> | Scan_Dis         | 1 | The spacing of each character displayed during entry. A maximum of 8 characters per line can be displayed.  |
| <b>0x2E</b> | 0x00             | 1 | 0x00  |
| <b>0x2F</b> | KB_Source        | 1 | Keyboard image page settings.<br>0 x00 : Keyboard on current page.<br>Other Values: Keyboard on another page.   |
| <b>0x30</b> | 0xFE             | 1 | 0xFE  |
| <b>0x31</b> | PIC_KB           | 2 | The page where the keyboard is located ( valid when the keyboard is not on the current page) .  |
| <b>0x33</b> | Area_KB          | 8 | Keyboard area coordinates: upper left corner ( Xs, Ys), lower right corner( Xe, Ye).  |
| <b>0x3B</b> | Area_KB_Position | 4 | Upper- left coordinates of the keyboard. ( valid when the keyboard is not on the current page)  |
| <b>0x3F</b> | Scan_Mode        | 1 | 0x02= pinyin input method<br>0x03= phonetic input method.   |

#### Note

Pinyin " bd" corresponds to all GBK- coded full- width punctuation entry;

DWIN' s pre-installed 0# font library contains all ASCII characters of 4\*8--64\*128dot matrix.

The key code (low byte) of the phonetic input method is defined as below (the phonetic input method is mainly used in Taiwan).

6.7.2.2 Software Setting

**GBK Input**

X 96 Y 153

W 218 H 213

Key value(0x) 0

voice ID 0

---

**Name definition** GBK Input

Automatic data upload

Button effect: -1 Specify

No button effect

Page switching: -1 Specify

NO page switching

Variable address(0x) 5020

Text length 100

Entry mode Re-enter

Display font 23

Dot matrix size 24

Text color 0000

Display font during entry 23

Dot matrix size 24

Text color 0000

Display method Pinyin hints are shown above

Cursor color black

Input status return

Input display area 44 194 668 299 Specify

Pinyin display position 143 269 set up

Display spacing 0

Keyboard settings Other pages

---

Keyboard settings

On the page 10

Keyboard area 29 155 690 588

Display position 26 171 set up

Coordinates of the upper left corner of the touch area

Touch area width and height

Can be set by default

Voice ID number

Checking means that the input is completed and the data is uploaded. At the same time, the 0x05.4 bit of the CFG file is required to write 1 to start the data upload. The 0x05 bit can be written 0x38, which means that the 22 initial value file is loaded, the data is uploaded, and the touch sound is turned on.

The display effect of pressing the touch area. For example, the current button is blue, and the button on the button effect page is dark blue. Press the button to see the sunken dark blue effect.

Variable storage space users can use address range arbitrarily: 0x1000-0xFFFF. According to the input character length occupies the corresponding length address, the maximum is 127 words (Word). The VP\_Len\_Max+1 address is occupied as the 0xFFFF terminator; therefore, one more address needs to be reserved. For example, 5 addresses are actually needed, and because the 0xFFFF terminator occupies one more address, 6 addresses need to be reserved. The VP\_1 address is used to save the input tag and effective data length, so the VP\_1 address needs to be reserved. System variable interface address range: 0x0000-0x0FFF (if the page switching address is 0x0084).

Maximum length of text variable, number of words (not bytes), range 0x01-0x7F;

Re-enter: When entering again, the previously entered characters will be cleared  
Modify the text: When entering again, the previously entered characters still exist, and the previously entered characters need to be deleted manually.

Chinese character library ID, set according to the dot matrix size at the time of generation. For example, here is the 23rd font, 24\*24 dot matrix is fixed size, other dot matrix cannot be set, otherwise the code is garbled.

Chinese character library ID, set according to the dot matrix size at the time of generation. For example, here is the 23rd font, 24\*24 dot matrix is fixed size, other dot matrix cannot be set, otherwise the code is garbled.

The basic touch of the called keyboard page is to define the key value. Note that the key value of the data such as 1 should be set to 0x2131. You can check the keyboard, refer to the routine, or refer to the ASCII key code table setting.

The coordinates of the upper left corner of the called keyboard displayed on the current page.



## 6.8 Synchrodata Return

There are three states of synchrodata return control: first pressing, continuous pressing, and releasing. Click the touch screen to return the data to the variable according to the regulations.

### 6.8.1 Instruction Storage Format

The synchrodata return control is to click the screen and return data to the variable or UART.

The T5L\_DGUS II screen has only 01 mode, which can replace the 02 mode of DGUS screen .

New functions: frame header + data length + 83 and VP address + length + DATA.

You can set VP1 S and VP1 T in DGUS.

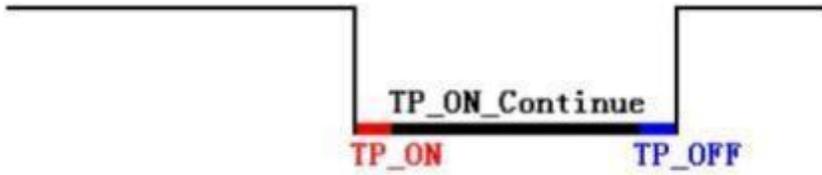
VP1 S can be set as the 22 file address, and VP1T refers to the VP address after the 83 instruction.

| Address | Definition          | Length(byte) | Description  |
|---------|---------------------|--------------|--|
| 0x00    | Pic_ID              | 2            | Page ID  |
| 0x02    | TP_Area             | 8            | Area of the control: upper left corner ( Xs, Ys), lower right corner ( Xe, Ye).  |
| 0x0A    | Pic_Next            | 2            | Switch target page. 0xFF** means no page switch.   |
| 0x0C    | Pic_On              | 2            | Button press effect page. 0xFF** means no button press effect  |
| 0x0E    | TP_Code             | 2            | 0 xFE08 , synchrodata return key code.   |
| 0x10    | 0xFE                | 1            | 0xFE   |
| 0x11    | TP_On_Mode          | 1            | Data return mode for the first touch.<br>0x00= no data is returned<br>0x01= read *VP2S pointing to LEN2 length data sent in DGUS serial protocol format to serial 2.                                     |
| 0x12    | VP1S                | 2            | Read the data in the address when first pressing.  |
| 0x14    | VP1T                | 2            | Write the data to the address when first pressing.   |
| 0x16    | 0x00                | 1            | 0x00   |
| 0x17    | LEN1                | 1            | Return data length in bytes. When TP_On_Mode=0x01, LEN1 must be even.  |
| 0x18    | 0xFE                | 1            | 0xFE   |
| 0x19    | TP_On_Continue_Mode | 1            | When the screen is continuously pressed, data return mode:<br>0x00= no data return<br>0 x01 = read the LEN2 length data pointed to by * VP2 S and send it to UART 2 in DGUS serial port protocol format. |
| 0x1A    | VP2S                | 2            | Read the data in the address when continuous pressing.   |
| 0x1C    | VP2T                | 2            | Write the data in the address when continuous pressing.  |
| 0x1E    | 0x00                | 1            | 0x00   |
| 0x1F    | LEN2                | 1            | Return data length in bytes. When TP_On_Continuous_Mode=0 x01,LEN2 must be even.   |
| 0x20    | 0xFE                | 1            | 0xFE   |
| 0x21    | TP_OFF_Mode         | 1            | When the screen is released, data return mode:<br>0x00= no data return<br>0x01= read the LEN2 length data pointed to by *VP2S and send it to UART2 in DGUS serial port protocol format.                  |
| 0x22    | VP3S                | 2            | Read the data in the address when the screen is released.  |
| 0x24    | VP3T                | 2            | Write the data in the address when the screen is released.   |



|      |      |   |  |
|------|------|---|--|
| 0x26 | 0x00 | 1 | 0x00   |
| 0x27 | LEN3 | 1 | Returned data length in bytes. When TP_OFF_Mode= 0x01 , LEN3 must be even. |
| 0x28 | 0x00 | 8 | Reserved. Write 0x00   |

The three states of touch screen pressing are as below.



### 6.8.2 Software Setting

|   |       |
|---|-------|
| Synchrodata Return  |       |
| X 53  | Y 315 |
| W 179   | H 182 |
| <input type="checkbox"/> Key value(0x) 0  |       |
| <input type="checkbox"/> voice ID 3F  |       |
| Button effect:<br>-1 Specify<br><input type="checkbox"/> No button effect                       |       |
| Page switching:<br>-1 Specify<br><input type="checkbox"/> NO page switching                     |       |
| First press<br>Mode 0x01<br>VP1S(0x) 5610<br>VP1T(0x) 5600<br>Data length 2<br>Unit: Byte       |       |
| Keep pressing<br>Mode 0x01<br>VP2S(0x) 5620<br>VP2T(0x) 5600<br>Data length 2<br>Unit: Byte     |       |
| Release the press<br>Mode 0x01<br>VP3S(0x) 5630<br>VP3T(0x) 5600<br>Data length 2<br>Unit: Byte |       |

Coordinates of the upper left corner of the touch area

Touch area width and height

Can be set by default

Voice ID number

The display effect of pressing the touch area. For example, the current button is blue, and the button on the button effect page is dark blue. Press the button to see the sunken dark blue effect.

Page switching ID, can be set by default

0×00: No data is returned;  
 0×01: Read the LEN1 byte length data pointed to by the \*VP1S address to \*VP1T. At the same time, the data is returned to the serial port.  
 0×02, 0×03 mode T5/T5L setting is invalid.

For example, the first time pressing/continuous pressing/releasing the pressing returns to 0×0001/0×0002/0×0003 different data settings; the configuration file needs to be enabled for automatic data upload.  
 The first time the button is pressed, the 2-byte data of the 0×5610 address is given to the 0×5600 address;  
 Finally, write 0×0001 in the starting address of the 22 file 0×ac20 (addressing 0×5610 double address, because the 22 file is in bytes and the variable address is in the unit of word length).  
 In this way, the first press can return to 5A A5 06 83 00 01 00 01.

Press the button continuously to give the 2-byte data of the 0×5620 address to the 0×5600 address;  
 Finally, write 0×0002 in the starting address of the 22 file 0×ac40 (addressing 0×5620 double address, because the 22 file is in bytes, and the variable address is in the unit of word length).  
 In this way, the first press can return to 5A A5 06 83 56 01 00 02.

Release the button to give the 2-byte data of the 0×5630 address to the 0×5600 address;  
 Finally, write 0×0003 in the starting address of the 22 file 0×ac60 (addressing 0×5630 double address, because the 22 file is in bytes, and the variable address is in the unit of word length).  
 In this way, the first press can return to 5A A5 06 83 56 01 00 03.



## 6.9 Rotation Adjustment

The rotation adjustment realizes variable data input by turning the knob, and can realize arc type drag adjustment.

### 6.9.1 Instruction Storage Format

| Address | Definition  | Length (byte) | Description  |
|---------|-------------|---------------|--|
| 0x00    | Pic_ID      | 2             | Page ID  |
| 0x02    | TP_Area     | 8             | Area of the control: upper left corner ( Xs, Ys), lower right corner ( Xe, Ye).  |
| 0x0A    | Pic_Next    | 2             | Switch target page. 0xFF** means no page switch.   |
| 0x0C    | Pic_On      | 2             | Button press effect page. 0xFF** means no button press effect.   |
| 0x0E    | TP_Code     | 2             | 0xFE09 , rotation adjustment key code.   |
| 0x10    | 0xFE        | 1             | 0xFE   |
| 0x11    | VP          | 2             | Variable pointer   |
| 0x13    | Data_Format | 1             | 0x00 : Adjust the VP(integer)<br>0x01: Adjust high byte of the VP( 1-byte unsigned integer, VP_H)<br>0x02: Adjust low byte of the VP( 1-byte unsigned integer, VP_L) |
| 0x14    | (X, Y)      | 4             | Center coordinate  |
| 0x18    | R0          | 2             | Inner diameter   |
| 0x1A    | R1          | 2             | Outer diameter   |
| 0x1C    | A0          | 2             | Start angle, range from 0 to 719, unit 0.5° .  |
| 0x1E    | V_Begin     | 2             | Value for start angle, integer.  |
| 0x20    | 0xFE        | 1             | 0xFE   |
| 0x21    | A1          | 2             | End angle, range from 0 to 719,unit 0.5°   |
| 0x23    | V_End       | 2             | Value for end angle, integer.  |

[ Note] The rotation adjustment needs to be used with the icon rotation control", and it is always assumed to be clockwise.

### 6.10.2 Software Setting Instructions

**Rotation adjustment**

X 155 Y 147

W 417 H 414

Key value(0x) 0

voice ID 3F

---

**Name definition** Rotation adjustment

Button effect:

-1 Specify

No button effect

Page switching:

-1 Specify

NO page switching

---

Variable address(0x) 543A

Adjust the data format 0x00 (integer)

Center coordinates 361, 350

set up

R0(area inner diameter) 100

R1 (outer diameter of area) 200

A0 (starting angle) 0

Corresponding return value 0

A1 (end angle) 720

Corresponding return value 720

Coordinates of the upper left corner of the touch area

Touch area width and height

Can be set by default

Voice ID number

The display effect of pressing the touch area. For example, the current button is blue, and the button on the button effect page is dark blue. Press the button to see the sunken dark blue effect.

Variable storage space users can use address arbitrarily: 0x1000-0xFFFF.  
The pop-up menu occupies 1 space address. That is, the non-associated control key addresses need to be spaced apart.  
System variable interface address range: 0x0000-0x0FFF (if the page switching address is 0x0084).

Integer: write the entire variable address.  
Write high and low bytes: it will be used for writing system variables such as writing backlight.

Click to set the coordinates of the center of the circle.  
The touch area can be drawn to be tangent to the circle.

The rotating icon is made as shown in the figure. After the black background is filtered, only the white pointer area will be displayed.

As shown:  
R0 inner diameter=453-353  
R1 outer diameter=553-353  
A0: The starting angle of the adjustment area, 0-719, the unit is 0.5  
A1: The end angle of the adjustment area, 1-720, the unit is 0.5

## 6.10 Sliding Adjustment

### 6.10.1 Instruction Storage Format

| Address | Definition | Length (byte) | Description  |
|---------|------------|---------------|--|
| 0x00    | Pic_ID     | 2             | Page ID  |
| 0x02    | TP_Area    | 8             | Area of the control: upper left corner ( Xs, Ys), lower right corner (Xe, Ye).   |
| 0x0A    | Pic_Next   | 2             | Switch target page. 0xFF** means no page switch.   |
| 0x0C    | Pic_On     | 2             | Button press effect page. 0xFF** means no button press effect.   |
| 0x0E    | TP_Code    | 2             | 0xFE0A, sliding adjustment.  |
| 0x10    | 0xFE       | 1             | 0xFE   |
| 0x11    | *VP        | 2             | Variable pointer, return adjustment data.<br>* VP reserved:<br>*VP+ 1 return data:<br>High byte: adjustment direction. 0x00= increase; 0xFF=decrease<br>Low byte: adjustment size. |
| 0x13    | Adj_Mode   | 1             | 0x00 horizontal; 0x01 vertical.  |
| 0x14    | Step_Dis   | 1             | Sensitivity; range: 0x01-0xFF.   |

### 6.10.2 Software Setting

The screenshot shows the 'Sliding Adjustment' configuration interface. The settings are as follows:

- X:** 423, **Y:** 176 (Coordinates of the upper left corner of the touch area)
- W:** 167, **H:** 362 (Touch area width and height)
- Key value(0x):** 0 (Can be set by default)
- voice ID:** 0 (Voice ID number)
- Name definition:** Sliding adjustment
- Variable address(0x):** 5652 (Variable storage space users can use address range arbitrarily: 0x1000-0xFFFF. Swipe gesture adjustment occupies 2 addresses, the VP address is reserved, and the returned data is in the (VP+1) position. System variable interface address range: 0x1000-0x0FFF (such as page switching address 0x0084))
- Sliding method:** Vertical sliding (Sliding vertically or horizontally, vertical is recommended. Includes visual feedback for values -1 to -5 and 1 to 5.)
- Sensitivity (1 is the highest sensitivity):** 10 (The speed when the data is sliding, 1 is the highest sensitivity.)



## 6.11 Page Sliding

Slide the touch screen on the X- axis direction of the specified area to realize the dynamic dragging of the page. You can set the target and area of page sliding, and the variable display of the current page will follow and drag.

If there are other touch buttons on the page at the same time, and the whole page ( including the touch buttons) needs to be switched by gesture, the touch priority of the page sliding must be set to the highest.

### 6.11.1 Instruction Storage Format

| Address | Definition       | Length (byte) | Description  |
|---------|------------------|---------------|--|
| 0x00    | Pic_ID           | 2             | Page ID  |
| 0x02    | TP_Area          | 8             | Area of the control: upper left corner ( Xs, Ys), lower right corner (Xe, Ye).   |
| 0x0A    | Pic_Next         | 2             | Switch target page. 0xFF** means no page switch.   |
| 0x0C    | Pic_On           | 2             | Button press effect page. 0xFF** means no button press effect.   |
| 0x0E    | TP_Code          | 2             | 0xFE0B, page sliding key code.   |
| 0x10    | 0xFE             | 1             | 0xFE   |
| 0x11    | Pic_Front        | 2             | Last page, 0xFF**= none.   |
| 0x13    | Pic_Next         | 2             | Next page, 0xFF**= none.   |
| 0x15    | Pic_Area         | 4             | ( Ys, Ye) defines the area valid for page sliding in the Y- axis direction.  |
| 0x19    | Push_Speed_Set   | 1             | Response time, range :0 x01-0xFF in 40 ms.   |
| 0x1A    | Push_Dis_Set     | 2             | Response sliding minimum space in X coordinates.   |
| 0x1C    | FB_En            | 1             | 0x00=no upload ;<br>Others= upload page ID.  |
| 0x1D    | End_Carton_Speed | 1             | 0x00= end without animation ;<br>Animation speed at the end of the slide, in pixels/DGUS cycle.<br>Recommended value is 1/20 of the horizontal resolution. |

### 6.11.2 Software Setting Instructions

**Page Sliding**

X 0 Y 0  
W 720 H 720  
 Key value(0x) 0  
 voice ID 0

Name definition Page sliding

previous page:  
4 Specify  
 No previous page

next page:  
6 Specify  
 No next page

Page turning sensitivity(time/distance)

Time 25  
distance 50

End animation setup  
End animation speed 36

illustrate:  
0×00 ends without animation; other value unit pixels divided by DGUS period, the recommended value is the horizontal resolution divided by 20

Coordinates of the upper left corner of the touch area

Touch area width and height

Can be set by default

Voice ID number

When swiping to the right, a new page appears on the left, for example, the middle page is page 5, and the previous page is page 4

When swiping to the left, a new page appears on the left, for example, the middle page is page 5, and the next page is page 6.

It is judged that the page is turned after pressing the finger for a certain period of time. Setting value\*40ms, the highest sensitivity is 100ms.

When the lateral sliding distance of the finger exceeds a certain value, it is judged to be a page turning. The maximum sensitivity is 50 pixels.

When you slide to turn the page, there will be an end ease-out effect, and the animation speed is controllable. The unit is pixel/DGUS period, and the recommended value is horizontal resolution/20. The setting value 00 means that no animation ends, compatible with the previous mode.



### 6.12 Slide Icon Selection

Sliding icon selection cooperates with icon page trans (JPEG icon panning display) to realize icon page sliding selection.

Steps:

- (1) Arrange the icons (program or menu index) to be used on the JPEG images with a resolution less than 4079\*4079 ( no more than 256KB after T5L1 compression, 768KB for T5L2).
- (2) Define a touch of a virtual page for this image (similar to the pop- up keyboard).
- (3) Place a window on the page to be used, and then you can (or write the VP+2 position integer value to UART to achieve movement) to slide the screen and choose icon.

#### 6.12.1 Instruction Storage Format

| Address | Definition      | Length(byte) | Description   |
|---------|-----------------|--------------|---|
| 0x00    | Pic_ID          | 2            | Page ID   |
| 0x02    | TP_Area         | 8            | Area of the Control: upper left corner (Xs, Ys), lower right corner (Xe,Ye), only for triggering.<br>It must be consistent with icon display area of the 0x07 display variable. |
| 0x0A    | Pic_Next        | 2            | Undefined, write FFFF.  |
| 0x0C    | Pic_On          | 2            | Undefined, write FFFF.  |
| 0x0E    | TP_Code         | 2            | 0xFE0C, sliding icon selection key code.  |
| 0x10    | 0xFE            | 1            | 0xFE  |
| 0x11    | VP              | 2            | Variable pointer  |
| 0x13    | Adj_Mode        | 1            | 0x00 horizontal sliding; 0x01 vertical sliding.   |
| 0x14    | TP_Page_ID_ICON | 2            | Icon ID<br>0x0000= undefined  |
| 0x16    | reserved        | 10           | Write 0x00  |

### 6.12.2 Software Setting

The screenshot shows the 'Slide Icon Selection' configuration window. It includes fields for X (30), Y (170), W (650), and H (400). There are checkboxes for 'Key value(0x) 0' and 'voice ID 16'. The 'Name definition' is 'ICONDragManager'. 'Automatic data upload' is checked. 'Variable address(0x)' is set to '5830'. 'Mobile mode' is set to 'Lateral movement'. 'Mobile interface' is set to '1000'. A note at the bottom states: 'Note: The size of this control should be consistent with the corresponding icon page translation control'. Five callout boxes provide additional details:

- Coordinates of the upper left corner of the touch area**: Points to the X and Y input fields.
- Touch area width and height**: Points to the W and H input fields.
- The variable storage space user can use the address range arbitrarily: 0x1000-0xFFFF. Slide icon selection to occupy 4 address spaces. That is, the non-associated control key addresses need to be spaced apart. System variable interface address range: 0x1000-0x0FFF (such as page switching address 0x0084).**: Points to the 'Variable address(0x)' field.
- For horizontal or vertical movement, it is recommended to be in the same direction as the pan control key on the icon page.**: Points to the 'Mobile mode' dropdown menu.
- The ID of the large icon page that needs to be dragged.**: Points to the 'Mobile interface' field.

### 6.13 Bit button

Click the button to specify the bit adjustment for the specified variable. It can be used in combination with the bit variable icon display control, for example, 1 group of 16 devices corresponding to 16 bit addresses, you can set the variable address for each bit to start and stop the device operation.

#### 6.13.1 Instruction Storage Format

| Address | Definition | Length(byte) | Description  |
|---------|------------|--------------|--|
| 0x00    | Pic_ID     | 2            | Page ID  |
| 0x02    | TP_Area    | 8            | Area of the Control: (Xs, Ys), (Xe, Ye)  |
| 0x0A    | Pic_Next   | 2            | The target page to switch to, 0xFF** means no page switch. <b>Must be 0xFF**.</b>  |
| 0x0C    | Pic_On     | 2            | The page where the button press effect image is located, 0xFF** means there is no button press effect.   |
| 0x0E    | TP_Code    | 2            | 0xFE0D   |
| 0x10    | 0xFE       | 1            | 0xFE   |
| 0x11    | *VP        | 2            | Variable pointer   |
| 0x13    | Bit_Pos    | 1            | Adjusted bit variable position, 0x00-0x0F.   |
| 0x14    | Adj_Mode   | 1            | Adjustment mode.<br>0x00=returns 0.<br>0x01=returns 1.<br>0x02= inversion.<br>0 x03 = 1 when the button is pressed, 0 when the button is released. |
| 0x15    | NULL       | 11           | Write 0x00   |

#### 6.13.2 Software Setting

The screenshot shows the Bit Button configuration window with several key settings and annotations:

- Upper left coordinates of the control:** X: 489, Y: 140
- Width and height of the control:** W: 133, H: 165
- Variable storage space range available to users:** 0X1000-0XFFFF. Bit button occupies 1 address, i.e. the address of non-associated control need to be spaced by 1. System variable interface address range: 0X0000-0X0FFF (e.g. page switch address 0X84)
- 1word=2byte=16bit.** This refers to a specific bit out of the 16 bits from 0 to 15.
- Adjust bit variable positions, 0x00-0x0F.** Each bit combined with the variable display can be controlled individually. The sequence of high to low bits is from bit15 to bit0. Each bit has two states, 0 and 1, and the corresponding ID icon display state prompt is invoked by writing 0 or 1 to the specific bit position.

**Bit icon control interface:** Shows a bit control settings dialog with a bit mask (1111111111111111) and a settings window for bit 15.

**Bit icon instruction format:**

| Address  | SP offset              | Definition   | Length (bytes) | Description              |      |           |      |       |      |       |      |       |      |      |      |      |      |                        |      |                        |      |                        |
|--|------------------------|--------------|----------------|--------------------------|------|-----------|------|-------|------|-------|------|-------|------|------|------|------|------|------------------------|------|------------------------|------|------------------------|
| 0x0: H   | 0x0: H                 | Display_Mode | 1              | Define the display mode: |      |           |      |       |      |       |      |       |      |      |      |      |      |                        |      |                        |      |                        |
| <table border="1"> <thead> <tr> <th>Mode</th> <th>Bit Value</th> </tr> </thead> <tbody> <tr> <td>0x00</td> <td>ICONS</td> </tr> <tr> <td>0x01</td> <td>ICONS</td> </tr> <tr> <td>0x02</td> <td>ICONS</td> </tr> <tr> <td>0x03</td> <td>None</td> </tr> <tr> <td>0x04</td> <td>None</td> </tr> <tr> <td>0x05</td> <td>Animation: ICONS-ICONS</td> </tr> <tr> <td>0x06</td> <td>Animation: ICONS-ICONS</td> </tr> <tr> <td>0x07</td> <td>Animation: ICONS-ICONS</td> </tr> </tbody> </table> |                        |              |                |                          | Mode | Bit Value | 0x00 | ICONS | 0x01 | ICONS | 0x02 | ICONS | 0x03 | None | 0x04 | None | 0x05 | Animation: ICONS-ICONS | 0x06 | Animation: ICONS-ICONS | 0x07 | Animation: ICONS-ICONS |
| Mode   | Bit Value              |              |                |                          |      |           |      |       |      |       |      |       |      |      |      |      |      |                        |      |                        |      |                        |
| 0x00   | ICONS                  |              |                |                          |      |           |      |       |      |       |      |       |      |      |      |      |      |                        |      |                        |      |                        |
| 0x01   | ICONS                  |              |                |                          |      |           |      |       |      |       |      |       |      |      |      |      |      |                        |      |                        |      |                        |
| 0x02   | ICONS                  |              |                |                          |      |           |      |       |      |       |      |       |      |      |      |      |      |                        |      |                        |      |                        |
| 0x03   | None                   |              |                |                          |      |           |      |       |      |       |      |       |      |      |      |      |      |                        |      |                        |      |                        |
| 0x04   | None                   |              |                |                          |      |           |      |       |      |       |      |       |      |      |      |      |      |                        |      |                        |      |                        |
| 0x05   | Animation: ICONS-ICONS |              |                |                          |      |           |      |       |      |       |      |       |      |      |      |      |      |                        |      |                        |      |                        |
| 0x06   | Animation: ICONS-ICONS |              |                |                          |      |           |      |       |      |       |      |       |      |      |      |      |      |                        |      |                        |      |                        |
| 0x07   | Animation: ICONS-ICONS |              |                |                          |      |           |      |       |      |       |      |       |      |      |      |      |      |                        |      |                        |      |                        |



### 6.13.3 Examples of Instructions

Take enabling the highest bit" bit15" of bit15 to bit0 as an example, 1000 0000 0000 0000 upload instructions in set 0, set 1, INV, and Inching as follows .

(1) **Set 0**. Write 0 to bit15 of the 16 bits.

Rx: 5AA5 06 83 5000 01 0000; 0x0000 is obtained by converting the binary data 0000 0000 0000 0000 into hexadecimal data.

(2) **Set 1**. Write 1 to bit15 of the 16 bits.

Rx: 5AA5 06 83 5000 01 8000; 0x8000 is obtained by converting the binary data 8000 0000 0000 0000 into hexadecimal data.

(3) **Inversion**, single bit inversion, 1 inversion for 0; 0 inversion for 1.

Set the current state of bit15 to 1, take the inverse 0 and upload it.

Rx: 5AA5 06 83 5000 01 0000; 0x0000 is obtained by converting binary data 0000 0000 0000 0000 into hexadecimal data.

Set the current state of bit15 to 0, take the inverse 1 and upload it: Rx: 5AA5 06 83 83

Rx: 5AA5 06 83 5000 01 8000; 0x8000 is obtained by converting binary data 1000 0000 0000 0000 into hexadecimal data.

(4) **Inching**, press once to write 1, release once to write 0.

Rx1: 5AA5 06 83 5000 01 8000; 0x8000 is obtained by converting binary data 1000 0000 0000 0000 into hexadecimal data.

Rx2: 5AA5 06 83 5000 01 0000; 0x0000 is obtained by converting binary data 0000 0000 0000 0000 into hexadecimal data.

(5) Write instruction, **bit0** on (write 1)

Tx: 5AA5 05 82 5000 0001; 0x0001 obtained by converting the binary data 0000 0000 0000 0001 to hexadecimal data.

(6) Write instruction, **bit15** on (write 1)

Tx: 5AA5 05 82 5000 8000; 0x8000 is obtained by converting binary data 1000 0000 0000 0000 into hexadecimal data.

(7) Write instruction, **high 8 bits** on (write 1)

Tx: 5AA5 05 82 5000 FF00; 0xFF00 is obtained by converting binary data 1111 1111 0000 0000 to hexadecimal data.

(8) ) Write instruction, **low 8 bits** on (write 1)

Tx: 5AA5 05 82 5000 00FF; 0x00FF is obtained by converting the binary data 0000 0000 1111 1111 to hexadecimal data.

(9) Write instruction, bit0, bit2, bit4, bit6, bit8, bit10, bit12, bit14 on

Tx: 5AA5 05 82 5000 **5555**; 0x5555 is obtained by converting binary data 0 **10 1 010 1 0 101 0 10 1 0 1** into hexadecimal data.

(10) Write instruction, bit1, bit3, bit5, bit7, bit9, bit11, bit13, bit15 on

Tx: 5AA5 05 82 5000 **AAAA**; 0xAAAA is obtained by converting the binary data **1010 1010 10 10 10 10** into hexadecimal data.

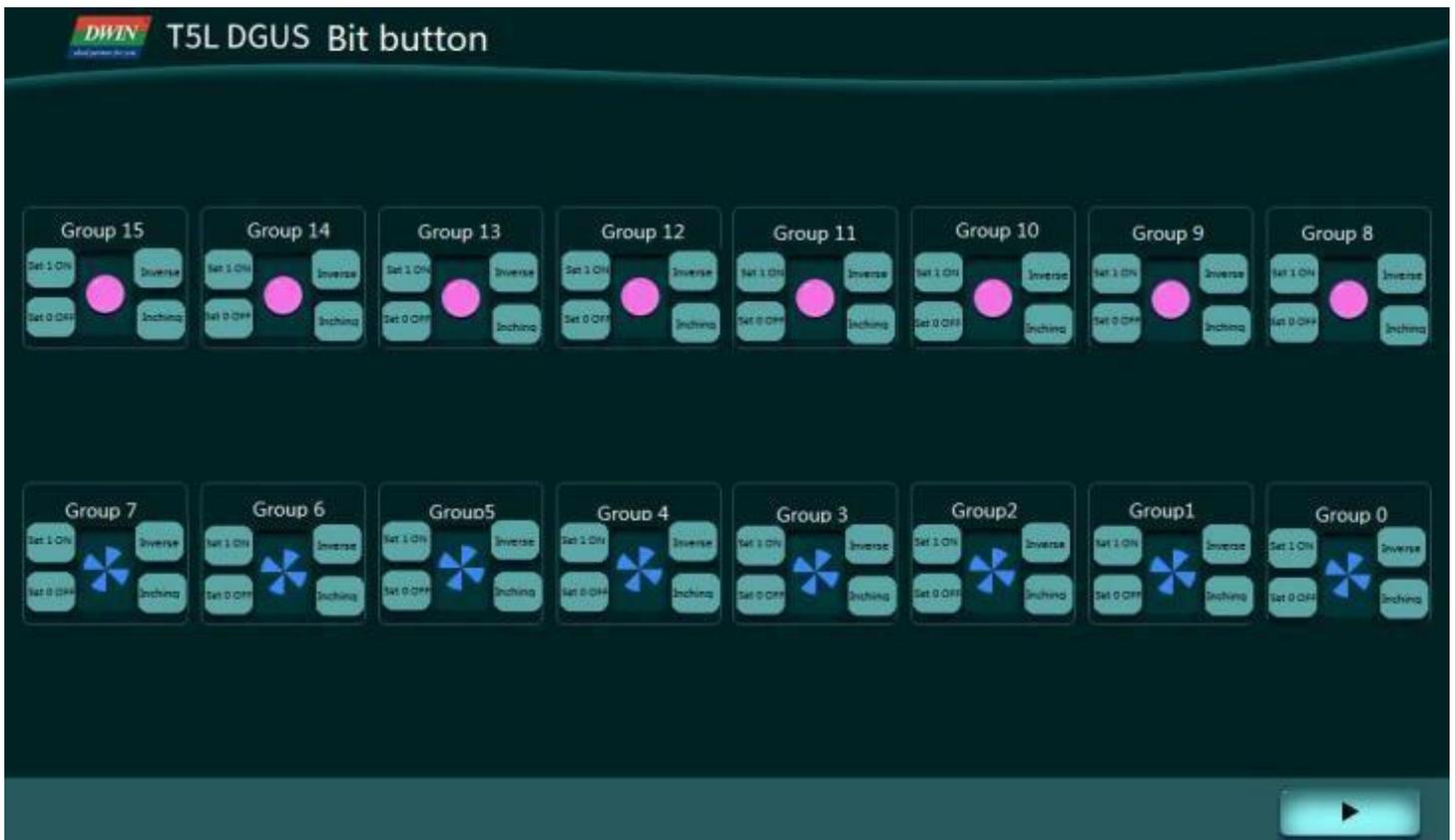
(11) Write instruction, **all 16 bits** on (all write 1)

Tx: 5AA5 05 82 5000 **FFFF**; 0xFFFF is obtained by converting binary data **11 11 11 11 11 11 11 11** to hexadecimal data.

(12) Write instruction, **all 16 bits** off (all write 0)

Tx: 5AA5 05 82 5000 **0000**; 0x0000 is obtained by converting the binary data 0000 0000 0000 0000 into hexadecimal data.

#### 6.13.4 Effect





## Chapter 7 Display Variable Configuration

The display variable configuration is stored in the 14. BIN, which consists of display instructions according to the control configurations.

Each display instruction occupies a fixed storage space of 32 bytes. Each page has a fixed allocation of 2KB, 4KB or 8KB (0x0800, 0x1000 or 0x2000) of variable storage space, i.e. up to 64, 128 or 255 variables per page. The maximum size of the 14. BIN file is 2 MB, which means up to 1024 pages can be configured (512 pages in 128-variable mode and 255 pages in 255-variable mode).

For 16 MBFLASH products, one 256 KB subspace in FLASH is fixed as 14. BIN configuration file space, that is, under the fixed allocation of 2KB, 4KB or 8KB (64, 128 or 255 variables) variable storage space per page, up to 128 pages (64 pages in 128-variable mode, 32 pages in 255-variable mode).

The data variable display space is 128 Kbytes.

For the same type of variables, the higher the storage location, the higher the display priority.

14. BIN can be generated by DGUS and used in applications with touch variables, associated VP, modification of 2.2 file, SP, access via UART or DWIN OS to achieve complex functions.

A display variable configuration instruction consists of the following six parts.

| Number | Definition  | Length | Definition   |
|--------|-------------|--------|--|
| 1      | 0x5A        | 1      | Fixed  |
| 2      | Type        | 1      | Variable type  |
| 3      | *SP         | 2      | The variable description file is loaded from Flash and stored to the address pointer of the data storage area. 0xFFFF means not dumped to the data storage area.               |
| 4      | Len_Dsc     | 2      | The word length of the variable description content.   |
| 5      | *VP         | 2      | The variable address is 0x0000 - 0xFFFF. For some variables that do not need to specify VP, write 0x0000. When the high byte of VP is 0xFF, this instruction will be canceled. |
| 6      | Description | N      | The variable description content.  |



### 7.1 List of Display Variables

| Number | Code | Variable name      | Data Length<br>(Character) | Description   |
|--------|------|--------------------|----------------------------|---|
| 01     | 0x00 | Variable Icon      | 1                          | The change range of a data variable linearly corresponds to a group of icon display; when the variable changes, the icon automatically switches accordingly. It is mostly used for detailed dashboard and progress bar display.<br>Support background overlay and transparency settings. Background filter intensity can be set.  |
| 02     | 0x01 | Animation Icon     | 2                          | A fixed value data variable corresponds to 3 different icon indication states: not display, display fixed icon, and display animation icon. It is mostly used for variable alarm prompts.<br>The variable occupies 2 word positions, and the (VP+1) position is reserved; the icon ID cannot exceed 255 (0xFF).<br>Support background overlay, transparency settings. And animation speed settings. Support single play mode. Background filter intensity can be set. |
| 03     | 0x02 | Slider Display     | 1                          | The change range of a data variable corresponds to the change of the display position of an icon (slider). It is mostly used for the indication of liquid level, dial and progress meter.<br>Support background overlay and transparency settings. Background filter intensity can be set.  |
| 04     | 0x03 | Artistic Variables | 1/2/4                      | Use icon to replace font library to display variable data.<br>Support background overlay and transparency settings. Background filter intensity can be set.   |
| 05     | 0x04 | Image Animation    | Without                    | Play a group of full-screen pictures at the specified speed. It is mostly used for boot interface or screen saver   |
| 06     | 0x05 | Icon Rotation      | 1                          | The range of change of a data variable linearly corresponds to the angle data, and then an icon is rotated according to the corresponding angle data.<br>Background filtering intensity can be set.<br>Mostly used for pointer dashboard display.   |
| 07     | 0x06 | Bit Icon           | 3                          | The 0/1 state of each bit of a data variable corresponds to two of the 8 different display schemes, and a icon (or icon animation) is used for corresponding display.<br>Background filter intensity can be set.<br>Mostly used to display the switch status, such as the operation (animation) and stop (stationary icon) of the fan.  |
| 08     | 0x07 | Icon Page Tran     | 4                          | The JPEG icon page that exceeds the screen resolution is displayed by panning up and down or left and right on the screen window.<br>Icon sliding selection can be achieved by combining this control with 0x0C touch variables.<br>Background filter intensity can be set.   |
| 09     | 0x08 | Icon Overlay       | Max 120KB                  | The JPEG icon of the variable buffer is superimposed and displayed in the specified area of the current page, and the display brightness and transparency can be set.<br>VP= 5AA5: turns on the display; VP+1=JPEG data buffer length; VP+2=JPEG data start.<br>Background filter intensity can be set.   |
| 10     | 0x09 | Batch Va Icon      | Max 64KB                   | According to the definition of the variable buffer, the icon is quickly copied from the background or video memory and displayed to the specified location.   |



|    |         |                             |              |   |
|----|---------|-----------------------------|--------------|---|
| 11 | 0x10    | Data Variables              | 1/2/4        | Display a data variable according to the specified format (integer, decimal with unit or not) with the specified font and size of Arabic numerals. Support character spacing adjustment/ non- adjustment selection; support integer bit invalid zero display/ non- display.<br>Supports jagged optimized 8 bit encoding font library.   |
| 12 | 0x11    | Text Display                | Max 2K       | Display the character string in the specified text box display area in the specified format ( decided by the selected font library) .<br>Supports jagged optimized 8 bit encoding font library.   |
| 13 | 0x12_00 | RTC Display(Text)           | None         | Display RTC in text according to user edit format.<br>Supports jagged optimized 8 bit encoding font library.  |
| 14 | 0x12_01 | RTC Display(Watch)          | None         | The calendar RTC is displayed with a pointer dial using the icon rotation.  |
| 15 | 0x13    | HEX Data                    | Max 8        | The variable data is displayed at intervals of ASCII characters specified by the user in byte HEX mode. It is mostly used for timing display, such as displaying 0x1234 as 12:34.<br>Supports jagged optimized 8 bit encoding font library.<br>Support the conversion of HEX data into BCD code display, for example, 0x0C is converted to 0x12 and displayed as 12.                                      |
| 16 | 0x14    | Text Scroll                 | User-defined | Scroll the text stored in VP in the designated area of the screen.  |
| 17 | 0x15    | Data Window                 | 2            | Display the data variable in a designated display window, and highlight the selected value. Combined with the touch screen sliding or incremental adjustment, the data can be scrolled and displayed. It can also be controlled by DWIN OS to adjust the speed. Supports saw tooth optimized 8 bit encoding font library.<br>The variable occupies 2 word positions, and the (VP+1) position is reserved. |
| 18 | 0x16    | DGUSII Text                 | Max 2K       | Based on DGUS II font, display the string in the specified text box display area, and scaling is not supported.<br>Compared to 0x11 text display, 0x16 mainly displays text without jaggies in multiple language from direct font libraries.  |
| 19 | 0x17    | Roll Character              | Max 2        | The basic functions are the same as the data window instructions. Increase the process animation display, increase the input method keyboard and text scrolling selection.  |
| 20 | 0x18    | GTF Icon                    | Max 128      | Display icons with high efficiency.   |
| 21 | 0x20    | Real- Time Curves           | 2K/channel   | Based on the curve buffer data to automatically match and display the real- time curve (trend graph). You can specify the display area, center axis coordinates, display scale (enlarge/ reduce) , and set the direction of the curve.  |
| 22 | 0x21_01 | Graphic_ Placement          | User-defined | Set point (x, y, color).  |
| 23 | 0x21_02 | Graphic_ End Connection     |              | End point connection (color, (x0, y0), ... (xn, yn).  |
| 24 | 0x21_03 | Graphic_ Rectangle          |              | Display rectangle, color, position and size are controllable.   |
| 25 | 0x21_04 | Graphic_ Rectangle Fill     |              | Fill the specified rectangular area with controllable fill color, position and size.  |
| 26 | 0x21_05 | Graphic_ Circle Display     |              | Displays the circle as center and radius. The color and thickness of the circle can be set.   |
| 27 | 0x21_06 | Graphic_ Picture Copy paste |              | Copy an area from the specified image and paste it on the currently displayed page.   |
| 28 | 0x21_07 | Graphic_ ICON Display       |              | The ICON is displayed, and the icon library can be selected.  |
| 29 | 0x21_08 | Graphic_ Enclosed Area Fill |              | Select the seed position and fill the enclosed solid color area with the specified color.   |



|    |         |                                    |         |  |
|----|---------|------------------------------------|---------|--|
| 30 | 0x21_09 | Plot_Spectrum Display              |         | Display spectrum (vertical lines) according to variable data, with controllable line color and position.   |
| 31 | 0x21_11 | Drawing_Ellipse arc display        |         | Display the arc of the ellipse according to the center of the ellipse, the major axis, the minor axis, and the display color.  |
| 32 | 0x21_0D | Adjust the Area Display Brightness |         | Adjust the display brightness of the designated display area to highlight or dilute the background display.  |
| 33 | 0x21_0E | Graphic_Dual Color Bitmap Display  |         | Select the seed position and fill the enclosed solid color area with the specified color.  |
| 34 | 0x21_12 | Drawing_four-color bitmap display  |         | Every 2 bits represents 1 point, and the four-color bitmap display is performed quickly in the specified area.<br>Multiple four-color bitmap display areas can be defined, and multiple layers can be used for more color display.   |
| 35 | 0x23    | Process Bar                        | 1       | Display a process bar at the specified position.   |
| 36 | 0x24    | Area Scrolling                     | 1       | Move the content of the specified area in a circular manner, and the direction of movement can be set.<br>Used to simply implement dynamic running effects such as flowcharts and progress bars. The variable address is occupied by the system and should not be used by users.   |
| 37 | 0x25    | QR Code                            | Max 259 | Display the QR code graphic on the screen according to the specified content.  |
| 38 | 0x26    | Brightness                         |         | Adjusts the display brightness of the specified display area to highlight or fade the background display.  |
| 39 | 0x30    | Data Transmit                      | Max 11  | After the page is switched, the predefined data is transferred to the variable or serial port once.  |
| 40 | 0x31    | Video                              | 6       | Simultaneous playback of ICL files and WAE files converted by MJPEG to form digital video.<br>VP and VP+ 1 positions (user control interface, double word).<br>D3: 0 x5 A means digital video playback is on, otherwise it's off.<br>D2: Playback status control, DGUS will clear it after processing.<br>0x01 = Stop, the screen stays at the first frame.<br>0x02 = Pause/resume playback.<br>0x03 = Playback from the specified position (position is determined by D1:D0).<br>D1: D0: Playback start position in seconds, valid only when D2=0x03.<br>VP+ 2 to VP+ 5 positions (status feedback interface, two double words, user can only read).<br>D7: Current playback status feedback, 0x00= Stop, 0x01= Playing.<br>D6:D4: Undefined.<br>D3:D2 Total video length, 0x0000-0xFFFF, in seconds.<br>D1:D0 Current playback video position, 0x0000-0xFFFF in seconds. |

Note: VP refers to the storage location (pointer) of the user variable storage space.

Set the variable SP during the development by DGUS, and store the display variable configuration information in the user variable space pointed to by the variable SP.

In the application, it can be accessed through UART or DWIN OS, dynamically open, close or modify the variable configuration information to realize the complex display variable combination function.



## 7.2 Variables Icon

The function of variable icon is to display the variation range of a data variable linearly corresponding to a group of icons. When the variable changes, the icons are automatically switched accordingly.

It is mostly used for fine dashboard and progress bar display.

This display function is usually used with incremental adjustment and text input.

### 7.2.1 Instruction Storage Format

| Address | SP offset | Definition | Length (bytes) | Description  |
|---------|-----------|------------|----------------|--|
| 0x00    |           | 0x5A00     | 2              |  |
| 0x02    |           | *SP        | 2              | Variable description pointer, 0xFFFF means loaded by the configuration file.           |
| 0x04    |           | 0x000A     | 2              |  |
| 0x06    | 0x00      | VP         | 2              | Variable pointer, integer.   |
| 0x08    | 0x01      | (x, y)     | 4              | Upper- left coordinates of the icons to display  |
| 0x0C    | 0x03      | V_Min      | 2              | Min value. Values smaller than " V_Min" will show no Icons.                            |
| 0x0E    | 0x04      | V_Max      | 2              | Max value. Values greater than " V_Max" will show no Icons.                            |
| 0x10    | 0x05      | Icon_Min   | 2              | Icon associated to the V_Min.  |
| 0x12    | 0x06      | Icon_Max   | 2              | Icon associated to the V_Max.  |
| 0x14    | 0x07:H    | Icon_Lib   | 1              | Index in the FLASH memory of the icon library to use.                                  |
| 0x15    | 0x07:L    | Mode       | 1              | Display mode:<br>0 x00 = transparency mode<br>Others= background display               |
| 0x16    | 0x08:H    | Layer_Mode | 1              | 0x00: overlay<br>0x01: overlay mode 1<br>0x02: overlay mode 2                          |
| 0x17    | 0x08:L    | ICON_Gamma | 1              | Icon brightness in overlay mode 2, range 0x00-0xFF, unit 1/256.                        |
| 0x18    | 0x09:H    | PIC_Gamma  | 1              | Background brightness in overlay mode 2, range 0 x00 - 0xFF, unit 1/256.               |
| 0x19    | 0x09:L    | Filter_Set | 1              | The intensity of the background color filtering in transparency mode, range 0x01-0x3F. |

## 7.2.2 Software Setting

|                                    |                          |  |
|------------------------------------|--------------------------|--|
| Icon variable                      |                          |  |
| X 219                              | Y 94                     | Coordinates of the upper left corner of the display area   |
| W 110                              | H 110                    | Display area width and height  |
| Name definition                    | VAR Icon                 | The pointer address is set by default or according to need.  |
| Description pointer(0x)            | FFFF                     | The variable storage space user can use the address range arbitrarily: 0×1000-0×FFFF. The icon variable occupies 1 space. That is, the non-associated control key addresses need to be spaced apart. System variable interface address range: 0×1000-0×0FFF (such as page switching address 0×0084).   |
| Variable address(0x)               | 5012                     | It is recommended to start naming at position 48, and summarize it into an ICL file for easy recall. For location naming, please refer to FLASH division, which is generated in advance and placed in the DWIN-SET folder.   |
| Lcon file                          | 48_Icon.ICL              |  |
| Variable lower limit               | 0                        | In the example, the lower limit 0 corresponds to icon 0, 5A A5 05 82 5012 0000 (display icon 0).   |
| Corresponding icon                 | 0                        |  |
| Variable upper limit               | 1                        | In the routine, the upper limit 1 corresponds to the icon No. 1, 5A A5 05 82 5012 0001 (display icon 1).   |
| Corresponding icon                 | 1                        |  |
| Display mode                       | Transparent              | Transparent is the filter background color, that is, the background is not displayed. Otherwise, the background is displayed.  |
| Background color dropout intensity | 0                        | The filter intensity refers to the range of the filter background color value. When there is black around the ICL icon file, the filter intensity can be increased. The range is 00-63. The icon background color is recommended to be set to black for filtering. If the icon or the background picture file looks a little fuzzy, please check whether the selected picture quality for generating the ICL file is low, and whether you have clicked all settings. |
| Initial value                      | 0                        | The initial icon displayed on power-on also requires CFG file 0×05, 5 digits write 1 to load 22 files, 0×05 digits can write 0×38, which means that 22 initial value files are loaded, data upload is enabled, and touch audio is enabled.   |
| Background display mode            | Overlay background image | Can be set by default.   |

### 7.2.3 Example of Instruction

① Display the icon corresponding to the lower limit 0

5A A5 05 82 5012 0000

0x5AA5: Frame header;

0x05: Data length;

0x82: Write instruction;

0x5012: VP;

0 x0000: Display the icon corresponding to the lower limit of 0.

② Display the icon corresponding to the upper limit 1

5A A5 05 82 5012 0001

0x5AA5: Frame header;

0x05: Data length;

0x82: Write instruction;

0x5012: VP;

0 x0001: display the icon corresponding to the upper limit 1.

③ The upper limit and lower limit are not displayed

5A A5 05 82 5010 0002

This instruction can call the icon ID that does not exist, it will not be displayed, and 0x0002 is the limit value.

④ Describe pointer hiding and moving icons

Hide and move the position need to use the description pointer. For the detailed explanation of the description pointer, see the data variable display description pointer instruction section.





## 7.3 Animation Icon

### 7.3.1 Instruction Storage Format

| Address | SP offset | Definition     | Length (bytes) | Description   |
|---------|-----------|----------------|----------------|---|
| 0x00    |           | 0x5A01         | 2              |   |
| 0x02    |           | SP             | 2              | Parameter pointer.<br>0xFFFF: Disables SP ( no run- time modification).   |
| 0x04    |           | 0x000D         | 2              |   |
| 0x06    | 0x00      | VP             | 2              | Variable pointer.<br>High word: Unsigned Integer(0x0000- 0xFFFF). stores animation status.<br>Low word: Reserved  |
| 0x08    | 0x01      | (x, y)         | 4              | Upper- left coordinates of the icons to display   |
| 0x0C    | 0x03      | Reset_ ICON_En | 2              | Indicates whether the animation should always start from the first frame when restarting.<br>0x0000: Animation continues from the last shown frame when reset.<br>0x0001: Animation starts from the first frame ("Icon_Start" ) when reset.   |
| 0x0E    | 0x04      | V_ Stop        | 2              | Value that stops the animation.   |
| 0x10    | 0x05      | V_ Start       | 2              | Value that starts the animation.  |
| 0x12    | 0x06      | ICON_ Stop     | 2              | Icon displayed when the animation is stopped.<br>Range : 0x0000-0x00FF  |
| 0x14    | 0x07      | ICON_ Start    | 2              | Icon displayed at the first frame of the animation.   |
| 0x16    | 0x08      | ICON_ End      | 2              | Icon displayed at the last frame of the animation.  |
| 0x18    | 0x09:H    | ICON_ Lib      | 1              | Index in the FLASH memory of the icon library to use.   |
| 0x19    | 0x09:L    | Mode           | 1              | ICON display mode:<br>0 x00 : Transparent background.<br>Other Values: Opaque   |
| 0x1A    | 0x0A:H    | Layer_ Mode    | 1              | 0x00: overlay<br>0x01: overlay mode 1 0x02: overlay mode 2  |
| 0x1B    | 0x0A:L    | ICON_ Gamma    | 1              | ICON brightness in overlay mode 2, range 0x00-0xFF, unit 1/256.   |
| 0x1C    | 0x0B:H    | PIC_ Gamma     | 1              | Background brightness in overlay mode 2, range 0x00 - 0xFF, 1/256.  |
| 0x1D    | 0x0B:L    | Time           | 1              | Time of single ICON, unit DGUS cycle, range 0x01-0xFF.  |
| 0x1E    | 0x0C:H    | Display mode   | 1              | 0x00: loop mode. 0x01: single mode.<br>When the variable is VP_ Stop, it play an animation once from Icon_ End to Icon_ Start.<br>When the variable is VP_ Start, it play an animation once from Icon_ Start to Icon_ End.<br>When the variable are other values, it display stop icon. |
| 0x1F    | 0x0C:L    | Filter_ Set    | 1              | Filter set value in transparency mode, range 0x00-0x3F.   |

### 7.3.2 Software Setting

|   |  |
|---|--|
| Animation icon  |  |
| X 106   | Coordinates of the upper left corner of the display area   |
| Y 82  |  |
| W 267   | Display area width and height  |
| H 267   |  |
| Name definition Animation icon  |  |
| Description pointer(0x) FFFF  | The pointer address is set by default or according to need.  |
| Variable address(0x) 5650   | The variable storage space user can use the address range arbitrarily: 0x1000-0xFFFF. The icon variable occupies 1 space. That is, the non-associated control key addresses need to be spaced apart. System variable interface address range: 0x1000-0x0FFF (such as page switching address 0x0084). |
| Stop value 0  | Stop value 0 (5A A5 05 82 5650 0000)   |
| Start value 1   | Start value 1 (5A A5 05 82 5650 0001)  |
| Lcon file 48_Icon.ICL   | It is recommended to start naming at position 48, and summarize it into an ICL file for easy recall. For location naming, please refer to FLASH division, which is generated in advance and placed in the DWIN-SET folder.   |
| Stop icon ID 66   | Write 0000 to the variable address, the corresponding icon ID when the animation stops.  |
| Start icon ID 54  | Write 0001 to the variable address to play a set of icon IDs corresponding to the start and end of the animation.  |
| End icon ID 66  |  |
| Display mode transparent  |  |
| Initial value 0   | The initial icon displayed on power-on also requires CFG file 0x05, 5 digits write 1 to load 22 files, 0x05 digits can write 0x38, which means that 22 initial value files are loaded, data upload is enabled, and touch audio is enabled.   |
| Single icon shows time 10   | Display time: 10*20ms (T5L cycle) 200ms  |
| Animation display mode Play repeatedly  |  |
| Background display mode Overlay background image  |  |
| For example: Loop playback: 13 pictures in total from 54-66. When the 66th batch is played, it will return to the 54th to repeat the playback. Single play: write 0000 value to the variable address, only play icon 54-66 once. Write the value 0001 to the variable address, and icon No. 66-54 will be played only once. |  |
| Overlay background (recommended), mode 1 effect, mode 2 effect.   |  |



### 7.3.3 Example of Instruction

① Turn on animation, loop play mode

5A A5 05 82 5650 0001

5A A5 Frame header;

05 Data length;

82 Write instruction;

6550 VP;

0001 Write 1 to start animation value.

② Stop animation, loop play mode

5A A5 05 82 5650 0000

5A A5 Frame header;

05 Data length;

82 Write instruction;

6550 VP;

0000 write 0 to stop animation value.

③ Sequential animation, single play mode, animation display mode needs to be set to single play.

5A A5 05 82 5650 0001

5A A5 Frame header;

05 Data length;

82 Write instruction;

6550 VP;

0001 Play once.

After playing in sequence, it needs to be played in reverse order, not in sequence all the time.

④ Reverse order animation, single play mode, animation display mode needs to be set to single play.

5A A5 05 82 5650 0000

5A A5 Frame header;

05 Data length;

82 Write instruction;

6550 VP;

0000 Play in reverse order.

⑤ Hide, move, change the animation icon frame number

Hide, move the position, etc. need to use the description pointer.

For the use of the sp, please refer to 7. 12 data variables.



### 7.3.4 Effect

A set of icons for animation icon.





## 7.4 Slider Display

The slider display control is to correspond the change range of a data variable to the change of the display position of an icon ( slider). It is mostly used for the indication of liquid level, dial, and progress meter.

Background overlay and transparency settings are supported, and the background filtering strength can be set.

| Address | SP offset | Definition   | Length (bytes) | Description   |
|---------|-----------|--------------|----------------|---|
| 0x00    |           | 0x5A02       | 2              |   |
| 0x02    |           | *SP          | 2              | Parameter pointer.<br>0xFFFF: Disables SP ( no run- time modification)  |
| 0x04    |           | 0x000C       | 2              |   |
| 0x06    | 0x00      | VP           | 2              | Variable pointer.   |
| 0x08    | 0x01      | V_Begin      | 2              | Min value accepted by the control.  |
| 0x0A    | 0x02      | V_End        | 2              | Max value accepted by the control.  |
| 0x0C    | 0x03      | x_Begin      | 2              | Min position of the slider, when its value equals"V_Begin".<br>X coordinate for horizontal sliders, Y coordinate for vertical sliders.  |
| 0x0E    | 0x04      | x_End        | 2              | Max position of the slider, when its value equals "V_Begin". X coordinate for horizontal sliders, Y coordinate for vertical sliders.  |
| 0x10    | 0x05      | ICON_ID      | 2              | Index of the icon in the icon library that will be used as the Slider handle.   |
| 0x12    | 0x06      | Y            | 2              | Position of slider icon in the secondary axis.<br>Y coordinate for horizontal sliders, X coordinate for vertical sliders.   |
| 0x14    | 0x07:H    | x_adj        | 1              | Icon position offset adjustment on the primary axis, in pixels. Offset in the X axis for horizontal sliders, and in the Y axis for vertical sliders.                          |
| 0x15    | 0x07:L    | Mode         | 1              | Slider mode. 0x00: Horizontal<br>0x01: Vertical.  |
| 0x16    | 0x08:H    | ICON_Lib     | 1              | Index in the FLASH memory of the icon library to use.   |
| 0x17    | 0x08:L    | ICON_Mode    | 1              | ICON display mode:<br>0 x00: Transparent background.<br>Other Values: Opaque background   |
| 0x18    | 0x09:H    | VP_Data_Mode | 1              | Value Memory Size.<br>0 x00: *VP points to the VP(integer).<br>0 x01: * VP points to the High Byte of the VP( VP_ H).<br>0x02: * VP points to the Low Byte of the VP( VP_ L). |
| 0x19    | 0x09:L    | Layer_Mode   | 1              | 0x00: overlay<br>0x01: overlay mode1<br>0x02: overlay mode 2  |
| 0x1A    | 0x0A:H    | ICON_Gamma   | 1              | ICON brightness in overlay mode 2, range 0x00-0xFF, unit 1/256.   |
| 0x1B    | 0x0A:L    | PIC_Gamma    | 1              | Background brightness in overlay mode 2, range0x00-0xFF, 1/256 .  |
| 0x1C    | 0x0B:H    | Filter_Set   | 1              | Filter set value in transparency mode, range0x00-0x3F.  |

### 7.4.1 Software Setting

|   |  |
|---|--|
| <p>Slider Display</p> <p>X 84 Y 335</p> <p>W 271 H 47</p> <p>Name definition Slider Display</p> <p>Description pointer(0x) FFFF</p> <p>Variable address(0x) 00A1</p> <p>Start variable scale value 0</p> <p>End variable scale value 255</p> <p>Scale mode Horizontal scale bar</p> <p>Lcon file 48_icon.ICL</p> <p>Swipe icon 17</p> <p>Display mode transparent</p> <p>Icon display (Y) coordinate value 503</p> <p>(X) coordinate forward offset 0<br/>(0-255)</p> <p>Variable type Points to high byte data</p> <p>Initial value 0</p> <p>The start and end scale coordinates, the system automatically defaults to the area range of the configuration button</p> <p>Background display mode</p> <p>Overlay background image</p> | <p>Coordinates of the upper left corner of the display area</p> <p>Display area width and height</p> <p>The pointer address is set by default or according to need.</p> <p>The variable storage space user can use the address range arbitrarily: 0×1000-0×FFFF. The icon variable occupies 1 space. That is, the non-associated control key addresses need to be spaced apart. System variable interface address range: 0×1000-0×0FFF (such as page switching address 0×0084).</p> <p>The data size range cannot be set to 0 at the same time, otherwise it cannot slide.</p> <p>Horizontal and vertical sliding mode</p> <p>It is recommended to start naming at position 48, and summarize it into an ICL file for easy recall! For location naming, please refer to FLASH division, which is generated in advance and placed in the DWIN-SET folder.</p> <p>Swipe to display the ICO icon</p> <p>Transparent is the filter background color. Show background</p> <p>Set by default or according to needs.</p> <p>Here the system variable interface 0=00A1 address points to the high byte as drag adjustment to change the volume.</p> <p>Overlay background (recommended), mode 1 effect, mode 2 effect.Ipsium</p> |
|---|--|

The diagram shows a horizontal slider bar with a blue knob. The bar is divided into three segments: a green segment on the left labeled 'X\_begin' with a value of -1.00, a red segment on the right labeled 'X\_end' with a value of +1.00, and a central white segment labeled 'Y' with a value of 0.00. A vertical line with a double-headed arrow indicates the current position of the knob, labeled 'X\_adj'.

### 7.4.2 Example of Instructions

Use 0x82 instruction to write the data within the setting range to VP.

Display data 10

5A A5 05 82 00A1 000A

5A A5 Frame header;

05 Data length;

82 Write instruction;

00A1 VP; 0

00A Data value.

### 7.4.4 Effect





### 7.5 Artistic Variables

The artistic variables control is to use a icon to replace the font library for the variable data display . Its function is similar to the variables icon control.

It supports background overlay and transparency settings, and the background filtering strength can be set.

| Address | SP offset | Definition      | Length (bytes) | Description   |
|---------|-----------|-----------------|----------------|---|
| 0x00    |           | 0x5A03          | 2              |   |
| 0x02    |           | *SP             | 2              | Parameter pointer.<br>0xFFFF: disables SP ( no run- time modification).   |
| 0x04    |           | 0x0009          | 2              |   |
| 0x06    | 0x00      | VP              | 2              | Variable pointer.   |
| 0x08    | 0x01      | (X, Y)          | 4              | Start display position:<br>Upper- left coordinates of the icons to display in Left- alignment mode or Upper- right coordinates of the icons to display in Right- alignment mode.  |
| 0x0C    | 0x03      | ICON0           | 2              | Icon corresponding to digit 0. The Icon Library must follow this indexing order: [0123456789-].   |
| 0x0E    | 0x04:H    | ICON Lib        | 1              | Index in the FLASH memory of the icon library to use.   |
| 0x0F    | 0x04:L    | ICON_ Mode      | 1              | 0 x00 : Transparent background.<br>Other Values: Opaque background  |
| 0x10    | 0x05:H    | Integer_ Digits | 1              | Number of digits to the left of the decimal separator.  |
| 0x11    | 0x05:L    | Decimal_ Digits | 1              | Number of digits to the right of the decimal separator.   |
| 0x12    | 0x06:H    | VP_ Mode        | 1              | Value Memory Size.<br>0x00: 16-bit Integer(2 bytes). Range: -32768 ~ 32767<br>0x01: 32-bit Integer(4 bytes). Range: -2147483648 ~ 2147483647<br>0x02: 8- bit Unsigned Integer in High Byte of the VP. Range: 0 ~ 255<br>0x03: 8-bit Unsigned Integer in Low Byte of the VP. Range: 0 ~ 255<br>0x04: 64-bit Integer(8 bytes). Range: -9223372036854775808 ~ 9223372036854775807<br>0x05: 16-bit Unsigned Integer(2 bytes). Range: 0 ~ 65535<br>0x06: 32-bit Unsigned Integer(4 bytes). Range: 0 ~ 4294967295<br>0x07: float point number(4 bytes)<br>0 x08: double float point number(8 bytes) |
| 0x13    | 0x06:L    | Alignment       | 1              | 0 x00: Left- alignment.<br>0x01: Right- alignment.  |
| 0x14    | 0x07:H    | Layer_ Mode     | 1              | 0x00: overlay<br>0x01: overlay mode 1<br>0x02: overlay mode 2   |
| 0x15    | 0x07:L    | ICON Gamma      | 1              | Icon brightness in overlay mode 2, range 0x00-0xFF, unit 1/256.   |
| 0x16    | 0x08:H    | PIC_ Gamma      | 1              | Background brightness in overlay mode 2, range 0x00 - 0xFF, unit 1/256.   |
| 0x17    | 0x08:L    | Filter_ Set     | 1              | Filter set value in transparency mode, range 0x00-0x3F.   |

### 7.5.1 Software Setting

|   |  |
|---|--|
| Artistic variable display   |  |
| X 269 Y 250   | Coordinates of the upper left corner of the display area   |
| W 185 H 161   | Display area width and height  |
| Name definition Artistic variables                                |  |
| Description pointer(0x) FFFF                                      | The pointer address is set by default or according to need.  |
| Variable address(0x) 5700   | The variable storage space user can use the address range arbitrarily: 0x1000-0xFFFF. The word art variable occupies 1/2/4 address according to the variable type. That is, the non-associated control key addresses need to be spaced apart. System variable interface address range: 0x1000-0x0FFF (such as page switching address 0x0084) |
| Lcon file 48_Icon.ICL   | It is recommended to start naming the 48th position, summarize it into an ICL file, and generate it in advance and put it into the DWIN-SET folder.  |
| Start icon 0  | 0 corresponds to the icon ID of word art 0.  |
| 0 corresponds to ICON_ID, the order of arrangement is 0123456789- |  |
| Display mode transparent  | Transparent is the filter background color. Show background  |
| Variable type Integer (2 bytes)                                   | Set the variable type according to the data range.   |
| Integer digits 2  | For example, if the integer digit is set to 2, the decimal place is set to 1, it will display 2 integers and 1 decimal. Write command: 5A A5 05 82 5700 0064 will be displayed as 10.0   |
| Decimal places 1  |  |
| Alignment Align left  | Set by default or according to need  |
| Initial value 0   | The initial value displayed on power-on, and at the same time, the 0x05.5 bit of the CFG file is required to write 1 to load 22 files. Write 0x38 to the 0x05 address, which means to load the 22 initial value file, upload the data, and turn on the touch sound.  |
| Background display mode Overlay background image                  | Overlay background (recommended), mode 1 effect, mode 2 effect.  |

### 7.5.2 Examples of Instructions

- 5A A5 05 82 5700 0002
- 0 x5AA5: Frame header;
- 0x05 Data length;
- 0x82: Write instruction;
- 0x 5700: VP;
- 0x 0002: Display data 2.

### 7.5.4 Effect



## 7.6 Image Animation

The image animation control is used to create an animation.

It can achieve screen jumps via serial communication.

### 7.6.1 Instruction Storage Format

| Address | SP offset | Definition  | Length (bytes) | Description  |
|---------|-----------|-------------|----------------|--|
| 0x00    |           | 0x5A04      | 2              |  |
| 0x02    |           | *SP         | 2              |  |
| 0x04    |           | 0x0005      | 2              |  |
| 0x06    | 0x00      | 0x0000      | 2              | Fixed value 0x0000   |
| 0x08    | 0x01      | Pic_Begain  | 2              | Start icon position  |
| 0x0A    | 0x02      | Pic_End     | 2              | Terminate icon position  |
| 0x0C    | 0x03:H    | Frame_Time  | 1              | The display time of one frame (one picture), the unit is 8ms.  |
| 0x0D    | 0x03:L    | ICL_LIB_ID  | 1              | Picture library file location, 0x00-0xff, 0x00 indicates the use of the background icon library.                             |
| 0x0E    | 0x04      | Pic_End_Exp | 2              | When using NAND FLASH to extend the animation stored in it, the page position automatically returned after the play is over. |
| 0x10    | 0x05      | Reserved    | 16             | Write 0x00   |

The starting picture position must be less than the terminal location.

If pictures animation variables are also set on the PIC\_END page, it can be continuously replayed.

Candidate instruction Switching pictures or touch instruction Switching pictures can end the replay.

### 7.6.2 Software Setting

Open DGUS, click display control - image animation, then set the control display area, and configure the functions.

This function does not need to be triggered by a button, and the area can be arbitrarily selected to ensure that it is on the specified page.

The screenshot shows the 'Image Animation' configuration window with the following settings and callouts:

- X 415 Y 28**: Coordinates of the upper left corner of the display area
- W 65 H 60**: Display area width and height
- Name definition Image Animation**: Name of the animation
- Description pointer(0x) FFFF**: The pointer address is set by default or according to need.
- Starting picture position 0** and **End image location 9**: The start and end pictures are played sequentially to form a set of animation effects.
- Display time setting 20 (x 8ms)**: Single picture display time

### 7.6.3 Example of Instructions

```
5A A5 07 82 0084 5A01 0000 0
```

x5AA5: Frame header;



0x07: Data length;

0x82: Write instruction;

0x0084 System variable interface address, which is a fixed switching page address . Refer to the section 7. 1 for more details.

0x5A01 : Fixed. The high byte 0x5A means that page processing is started once, and the CPU is cleared after processing; the low byte 0x01 means that the page is switched, and the picture specified in the picture storage area is displayed to the current background page; see the system variable interface list for details.

0x0000: Page ID, the IF of the page to be switched to.

### 7.6.4 Effect

0.bmp    1.bmp    2.bmp    3.bmp    4.bmp    5.bmp    6.bmp    7.bmp    8.bmp    9.bmp

Start image position.  
 Select an image animation control on page 0 of the project.  
 Start picture page ID: 0  
 End picture page ID: 9  
 The screen starts from page 0 by default, so a set of boot animation is made.

End picture position  
 Image animation end page ID: 0-9  
 It can realize replay and use as screensaver.



## 7.7 Icon Rotation

The icon rotation control linearly corresponds the change range of a data variable to the angle data, and then rotate an icon according to the corresponding angle data and display it.

It is mostly used for pointer instrument panel display.

### 7.7.1 Instruction Storage Format

The background filter intensity can be set

| Address | SP offset | Definition | Length (bytes) | Description  |
|---------|-----------|------------|----------------|--|
| 0x00    |           | 0x5A05     |                |  |
| 0x02    |           | SP         | 2              | Parameter pointer.<br>0xFFFF: Disables SP (Non run-time modification).   |
| 0x04    |           | 0x000C     | 2              | 0x000C.  |
| 0x06    | 0x00      | VP         | 2              | Variable pointer.  |
| 0x08    | 0x01      | ICON_ID    | 2              | Index of the icon in the icon library that will be rotated (usually a dial pointer).   |
| 0x0A    | 0x02      | ICON_xc    | 2              | Center of rotation on the icon. X coordinate.  |
| 0x0C    | 0x03      | ICON_Yc    | 2              | Center of rotation on the icon. Y coordinate.  |
| 0x0E    | 0x04      | xc         | 2              | The x coordinates of the icon rotation center on the current page.   |
| 0x10    | 0x05      | Yc         | 2              | The y coordinates of the icon rotation center on the current page.   |
| 0x12    | 0x06      | V_Begin    | 2              | Min value.   |
| 0x14    | 0x07      | V_End      | 2              | Max value.   |
| 0x16    | 0x08      | AL_Begin   | 2              | Starting rotation angle, 0-720 (0x000-0x2D0), unit 0.5° .  |
| 0x18    | 0x09      | AL_End     | 2              | End rotation angle, 0-720 (0x000-0x2D0), unit 0.5° .   |
| 0x1A    | 0x0A:H    | VP_Mode    | 1              | Value Memory Size.<br>0x00: *VP points to the VP (integer).<br>0x01: *VP points to the High Byte of the VP (VP_H).<br>0x02: *VP points to the Low Byte of the VP (VP_L). |
| 0x1B    | 0x0A:L    | Lib_ID     | 1              | Index in the FLASH memory of the icon library to use.  |
| 0x1C    | 0x0B:H    | Mode       | 1              | 0x00 : Transparent background.<br>Other Values: Opaque background  |
| 0x1D    | 0x0B:L    | Filter_Set | 1              | When the icon is transparent, the strength of the background color filtering is 0x01-0x3F.   |

### 7.7.2 Software Setting

|  |                               |  |
|--|-------------------------------|--|
| <b>Icon rotation indication</b>  |                               |  |
| X 240  | Y 207                         | Coordinates of the upper left corner of the display area   |
| W 27   | H 35                          | Display area width and height  |
| <b>Name definition</b> Icon Rotation   |                               |  |
| Description pointer(0x) FFFF   |                               | The pointer address is set by default or according to need.  |
| Variable address(0x) 5438  |                               | The variable storage space user can use the address range arbitrarily: 0×1000-0×FFFF. Icon rotation occupies 1 address space. That is, the non-associated control key addresses need to be spaced apart.<br>System variable interface address range: 0×1000-0×0FFF (such as page switching address 0×0084) |
| Icon file 48_Icon.ICL  |                               | It is recommended to start at position 48 and summarize it into an ICL file. Generate it in advance and put it into the DWIN_SET folder.   |
| Icon ID 70   |                               | Pointer icon ID.   |
| <b>Icon rotation center:</b>   |                               |  |
| X 10   | Y 117                         | Icon rotation center coordinates, which can be specified by loading icons.   |
| The screen rotation center, the system automatically defaults to the upper left coordinate of the configuration button |                               |  |
| Variable value of initial rotation angle   | 0                             | The start and end values of the data range are displayed.  |
| End rotation angle variable value  | 200                           |  |
| Starting angle of rotation   | 495                           | 495 degrees, 225 degrees calculation:<br>360 degrees is divided into 4 single sub-areas of 90 degrees,<br>90 degrees is divided into 4 individual sub-areas of 22.5,<br>Starting angle of rotation: $720 - (90 + 22.5) * 2 = 495$ degrees.<br>End rotation angle: $(90 + 22.5) * 2 = 225$ degrees          |
| End rotation angle   | 225                           |  |
| Rotation angle, 0-720 (0×000-0×200), unit 0.5  |                               |  |
| Display mode   | transparent                   |  |
| VP_Mode  | Points to an integer variable |  |
| Initial value  | 0                             |  |

### 7.7.3 Examples of Instructions

5A A5 05 82 5438 0000

0x5AA5: Frame header;

0x05: Data length; 0x82:

Write instruction;

0x5438 VP;

0x0000 data 0; the icon pointer points to the dial 0 degree.

5A A5 05 82 5438 0064

0x5AA5: Frame header;

0x05: Data length; 0x82:

Write instruction;

0x5438 VP;

0x0064 data 100; the icon pointer points to 100 degrees.

5A A5 05 82 5438 00C8

0x5AA5: Frame header;

0x05: Data length; 0x82:

Write instruction;

0x5438 VP; 0x00C8 data 200; the icon pointer points to 200 degrees.

### 7.7.4 Effect



## 7.8 Bit Icon

The bit icon control is used to show fixed and/or animated Icons, according to a bit flag value on the VP. The value of each bit represents the state of a single Icon, and many icons can be shown in different states, based on the VPC.

It is often used to display several alarms at once, or to implement bar graphs.

### 7.8.1 Instruction Storage Format

| Address | SP offset                | Definition               | Length (bytes) | Description  |      |           |  |  |   |   |      |        |        |      |        |      |      |        |                          |      |      |        |      |      |                          |      |                          |        |      |                          |      |      |                          |                          |
|---------|--------------------------|--------------------------|----------------|--|------|-----------|--|--|---|---|------|--------|--------|------|--------|------|------|--------|--------------------------|------|------|--------|------|------|--------------------------|------|--------------------------|--------|------|--------------------------|------|------|--------------------------|--------------------------|
| 0x00    |                          | 0x5A06                   | 2              |  |      |           |  |  |   |   |      |        |        |      |        |      |      |        |                          |      |      |        |      |      |                          |      |                          |        |      |                          |      |      |                          |                          |
| 0x02    |                          | *SP                      | 2              | Variable description pointer, 0xFFFF means loaded by configuration file.   |      |           |  |  |   |   |      |        |        |      |        |      |      |        |                          |      |      |        |      |      |                          |      |                          |        |      |                          |      |      |                          |                          |
| 0x04    |                          | 0x000C                   | 2              |  |      |           |  |  |   |   |      |        |        |      |        |      |      |        |                          |      |      |        |      |      |                          |      |                          |        |      |                          |      |      |                          |                          |
| 0x06    | 0x00                     | *VP                      | 2              | Bit variable pointer, word variable  |      |           |  |  |   |   |      |        |        |      |        |      |      |        |                          |      |      |        |      |      |                          |      |                          |        |      |                          |      |      |                          |                          |
| 0x08    | 0x01                     | *VP_AUx                  | 2              | Auxiliary variable pointer, double word, cannot be accessed by user software   |      |           |  |  |   |   |      |        |        |      |        |      |      |        |                          |      |      |        |      |      |                          |      |                          |        |      |                          |      |      |                          |                          |
| 0x0A    | 0x02                     | Act_Bit_Set              | 2              | The bit position whose value is 1 indicates that the corresponding position of *VP needs to be displayed.  |      |           |  |  |   |   |      |        |        |      |        |      |      |        |                          |      |      |        |      |      |                          |      |                          |        |      |                          |      |      |                          |                          |
| 0x0C    | 0x03:H                   | Display_mode             | 1              | Define the display mode: <table border="1" data-bbox="643 1048 1477 1386"> <thead> <tr> <th>Mode</th> <th colspan="2">Bit value</th> </tr> <tr> <td></td> <th>0</th> <th>1</th> </tr> </thead> <tbody> <tr> <td>0x00</td> <td>ICON0S</td> <td>ICON1S</td> </tr> <tr> <td>0x01</td> <td>ICON0S</td> <td>None</td> </tr> <tr> <td>0x02</td> <td>ICON0S</td> <td>Animation: ICON1S-ICON1E</td> </tr> <tr> <td>0x03</td> <td>None</td> <td>ICON1S</td> </tr> <tr> <td>0x04</td> <td>None</td> <td>Animation: ICON1S-ICON1E</td> </tr> <tr> <td>0x05</td> <td>Animation: ICON0S-ICON0E</td> <td>ICON1S</td> </tr> <tr> <td>0x06</td> <td>Animation: ICON0S-ICON0E</td> <td>None</td> </tr> <tr> <td>0x07</td> <td>Animation: ICON0S-ICON0E</td> <td>Animation: ICON1S-ICON1E</td> </tr> </tbody> </table> | Mode | Bit value |  |  | 0 | 1 | 0x00 | ICON0S | ICON1S | 0x01 | ICON0S | None | 0x02 | ICON0S | Animation: ICON1S-ICON1E | 0x03 | None | ICON1S | 0x04 | None | Animation: ICON1S-ICON1E | 0x05 | Animation: ICON0S-ICON0E | ICON1S | 0x06 | Animation: ICON0S-ICON0E | None | 0x07 | Animation: ICON0S-ICON0E | Animation: ICON1S-ICON1E |
| Mode    | Bit value                |                          |                |  |      |           |  |  |   |   |      |        |        |      |        |      |      |        |                          |      |      |        |      |      |                          |      |                          |        |      |                          |      |      |                          |                          |
|         | 0                        | 1                        |                |  |      |           |  |  |   |   |      |        |        |      |        |      |      |        |                          |      |      |        |      |      |                          |      |                          |        |      |                          |      |      |                          |                          |
| 0x00    | ICON0S                   | ICON1S                   |                |  |      |           |  |  |   |   |      |        |        |      |        |      |      |        |                          |      |      |        |      |      |                          |      |                          |        |      |                          |      |      |                          |                          |
| 0x01    | ICON0S                   | None                     |                |  |      |           |  |  |   |   |      |        |        |      |        |      |      |        |                          |      |      |        |      |      |                          |      |                          |        |      |                          |      |      |                          |                          |
| 0x02    | ICON0S                   | Animation: ICON1S-ICON1E |                |  |      |           |  |  |   |   |      |        |        |      |        |      |      |        |                          |      |      |        |      |      |                          |      |                          |        |      |                          |      |      |                          |                          |
| 0x03    | None                     | ICON1S                   |                |  |      |           |  |  |   |   |      |        |        |      |        |      |      |        |                          |      |      |        |      |      |                          |      |                          |        |      |                          |      |      |                          |                          |
| 0x04    | None                     | Animation: ICON1S-ICON1E |                |  |      |           |  |  |   |   |      |        |        |      |        |      |      |        |                          |      |      |        |      |      |                          |      |                          |        |      |                          |      |      |                          |                          |
| 0x05    | Animation: ICON0S-ICON0E | ICON1S                   |                |  |      |           |  |  |   |   |      |        |        |      |        |      |      |        |                          |      |      |        |      |      |                          |      |                          |        |      |                          |      |      |                          |                          |
| 0x06    | Animation: ICON0S-ICON0E | None                     |                |  |      |           |  |  |   |   |      |        |        |      |        |      |      |        |                          |      |      |        |      |      |                          |      |                          |        |      |                          |      |      |                          |                          |
| 0x07    | Animation: ICON0S-ICON0E | Animation: ICON1S-ICON1E |                |  |      |           |  |  |   |   |      |        |        |      |        |      |      |        |                          |      |      |        |      |      |                          |      |                          |        |      |                          |      |      |                          |                          |
| 0x0D    | 0x03:L                   | Move_Mode                | 1              | Bitmap icon arrangement:<br>0x00= x++, the non-display bit specified by Act_Bit_Set is not reserved; 0x01 =Y++, the non-display bit specified by Act_Bit_Set does not retain the position<br>0x02=x++, the non-display bit specified by Act_Bit_Set is reserved for DIS_MOV position<br>0x03=Y++, the non-display bit specified by Act_Bit_Set is reserved for DIS_MOV position  |      |           |  |  |   |   |      |        |        |      |        |      |      |        |                          |      |      |        |      |      |                          |      |                          |        |      |                          |      |      |                          |                          |
| 0x0E    | 0x04:H                   | Icon_Mode                | 1              | ICON display mode:<br>0x00: Transparent background<br>0x01: opaque   |      |           |  |  |   |   |      |        |        |      |        |      |      |        |                          |      |      |        |      |      |                          |      |                          |        |      |                          |      |      |                          |                          |
| 0x0F    | 0x04:L                   | Icon_Lib                 | 1              | Index in the FLASH memory of the icon library to use.  |      |           |  |  |   |   |      |        |        |      |        |      |      |        |                          |      |      |        |      |      |                          |      |                          |        |      |                          |      |      |                          |                          |
| 0x10    | 0x05                     | ICON0S                   | 2              | Dose not show animation mode, bit_0 icon ID<br>Display animation mode, bit_0 icon animation start ID position  |      |           |  |  |   |   |      |        |        |      |        |      |      |        |                          |      |      |        |      |      |                          |      |                          |        |      |                          |      |      |                          |                          |
| 0x12    | 0x06                     | ICON0E                   | 2              | Display animation mode, bit_0 icon animation end ID position   |      |           |  |  |   |   |      |        |        |      |        |      |      |        |                          |      |      |        |      |      |                          |      |                          |        |      |                          |      |      |                          |                          |
| 0x14    | 0x07                     | ICON1S                   | 2              | Dose not show animation mode, bit_1 icon ID<br>Display animation mode, bit_1 icon animation start ID position  |      |           |  |  |   |   |      |        |        |      |        |      |      |        |                          |      |      |        |      |      |                          |      |                          |        |      |                          |      |      |                          |                          |
| 0x16    | 0x08                     | ICON1E                   | 2              | Display animation mode, bit_1 icon animation end ID position   |      |           |  |  |   |   |      |        |        |      |        |      |      |        |                          |      |      |        |      |      |                          |      |                          |        |      |                          |      |      |                          |                          |



|             |               |            |   |  |
|-------------|---------------|------------|---|--|
| <b>0x18</b> | <b>0x09</b>   | (x, y)     | 4 | The starting position variable display position, the coordinate position of the upper left corner of the icon. |
| <b>0x1C</b> | <b>0x0B</b>   | DIS_MOV    | 2 | Next icon coordinate move coordinate interval.   |
| <b>0c1E</b> | <b>0x0C:H</b> | Filter_Set | 1 | Write 0x00   |

## 7.8.2 Software Setting

|   |  |  |
|---|--|--|
| <b>Bit icon</b><br>X 41 Y 183<br>W 50 H 33  |  | Coordinates of the upper left corner of the display area<br>Display area width and height.   |
| Name definition Bit icon<br>Description pointer(0x) FFFF<br>Variable address(0x) 6000<br>Secondary address(0x) 8020 |  | The pointer address is set by default or according to need.<br>Variable storage space users can use address range arbitrarily: 0x1000-0xFFFF. The bit variable occupies a maximum of 3 space addresses. That is, the non-associated control key addresses need to be spaced apart. System variable interface address range: 0x0000-0x0FFF (such as page switching address 0x0084).<br>If the animation mode is used, the auxiliary address must be separated by 2 addresses, and the address cannot be repeated. You don't need to set the animation if you don't use it.  |
| Bit Icon ON (1) / OFF (0):<br>000000000000000001<br>Set up  |  | The 0th bit is enabled here, a total of 16 bits.   |
| Display mode 0x03<br>Mobile mode 0x00<br>Moving interval 0<br>Icon file 48_Icon.ICL                                 |  | 0x01-0x07, define the wind table Display_Mode. In 0x03 mode, writing 0 to the corresponding bit does not display the icon, writing 1 to display the icon.<br>0x00-0x03, define the wind table Move_Mode.<br>The pixel interval to move the next icon.<br>It is recommended to start at position 48 and summarize them into an ICL file. Generate it in advance and put it into the DWIN_SET folder.  |
| ICON0S 0<br>ICON0E 0<br>ICON1S 67<br>ICON1E 0   |  | The meaning of icon mode and non-display mode settings:<br>ICON1S: Write 1 to the corresponding bit to display the icon ID, or not display it.<br>ICON1E: No need to set at this time<br>The meaning of the animation mode setting:<br>ICON1S; Write 1 icon animation start ID in the corresponding bit. ICON1E; Write 0 icon animation end ID<br>The meaning of icon mode and non-display mode settings:<br>ICON1S: Write 1 to the corresponding bit to display the icon ID, or not display it.<br>ICON1E: No need to set at this time<br>The meaning of the animation mode setting:<br>ICON1S; Write 1 icon animation start ID in the corresponding bit. ICON1E; Write 1 icon animation end ID |
| Display mode transparent<br>Initial value 0   |  |  |

### 7.8.3 Example of Instructions

① 16 bits fully open

5A A5 05 82 6000 FFFF

0x5A 0xA5: Frame header;

0x05: Data length;

0x82: Write instruction;

0x6000: VP;

0x FFFF: 11 11 11 11 11 11 11 11 11 11 Write 1 to all 16 bits, which is FFFF.

② 16 bits all closed

5A A5 05 82 6000 0000

0x5A 0xA5: Frame header;

0x05: Data length;

0x82: Write instruction;

0x6000: VP;

0x0000: All 16 bits are written with 0.

③ Turn on bit 0 and bit 3

5A A5 05 82 6000 0009

0x5A 0xA5: Frame header;

0x05: Data length;

0x82: Write instruction;

0x6000: VP;

0x0009: Write 1 to bit 0 and bit 3, 0000 0000 0000 1001 is 0x0009.

### 7.8.4 Effect





## 7.9 Icon Page Tran

Display icons larger than the screen resolution by panning up and down or left and right in the screen viewport. Icon sliding selection can be realized by this control and sliding icon control.

The background filter strength can be set.

### 7.9.1 Instruction Storage Format

| Address | SP offset | Definition    | Length (bytes) | Description  |
|---------|-----------|---------------|----------------|--|
| 0x00    |           | 0x5A07        | 2              |  |
| 0x02    |           | *SP           | 2              | Variable description pointer, 0xFFFF means loaded by configuration file.   |
| 0x04    |           | 0x0009        | 2              |  |
| 0x06    | 0x00      | *VP           | 2              | Variable pointer, each variable occupies 4 words.<br>VP=The display starting position of the current display window on the icon page, (x, y) 2 characters;<br>VP+2=moving distance, 16bit signed number;<br>Negative numbers move right (down), positive numbers move left (up);<br>VP+3 Reserved.   |
| 0x08    | 0x01:H    | ICON_Lib      | 1              | Icon library ID, 0x00 means use the background icon library.   |
| 0x09    | 0x01:L    | Disp_Mode     | 1              | ICON display mode: 0x00=transparent (do not display icon background); others=display background.   |
| 0x0A    | 0x02:H    | Filter_Set    | 1              | When the icon is displayed transparently, the intensity of background color filtering is 0x01-0x3F.  |
| 0x0B    | 0x02:L    | Move_Mode     | 1              | Move mode:<br>0x00=horizontal movement, the X coordinate of the ICON page can be very large.<br>Others = vertical movement, the Y coordinate of the ICON page can be very large.<br>The JPEG file size of the ICON icon page cannot exceed the hardware limit:<br>252KB for T5L1 and 764KB for T5L2. |
| 0x0C    | 0x03      | ICON_ID       | 2              | Icon (page) ID.  |
| 0x0E    | 0x04      | (Xs,Yx,Xe,Ye) | 8              | On the current page, the icon page displays the area.  |
| 0x16    | 0x08      | Reserved      | 9              |  |



### 7.9.2 Software Setting

The screenshot shows the configuration interface for 'Icon page Tran'. The settings include:

- X 30, Y 162:** Coordinates of the upper left corner of the display area.
- W 658, H 405:** Display area width and height.
- Name definition:** Icon Page Tran
- Description pointer(0x):** FFFF
- Variable address(0x):** 5830
- Icon file:** 48\_Icon.ICL
- Corresponding icon:** 93
- Display mode:** Display background
- Background color dropout intensity:** 0
- Mobile mode:** Mode move
- Initial value:** 0

Callout boxes provide additional details:

- Variable storage space users can use address range arbitrarily: 0x1000-0xFFFF. The bit variable occupies a maximum of 3 space addresses. That is, the non-associated control key addresses need to be spaced apart. System variable interface address range: 0x0000-0x0FFF (such as page switching address 0x0084).
- It is recommended to start naming at position 48, and summarize it into an ICL file for calling. For location naming, please refer to FLASH division, which is generated in advance and placed in the DWIN-SET folder.
- Make "huge" pictures into icon library to call.
- Transparent is the filter background color. That is, the background is not displayed. Otherwise, the background is displayed.
- The range is 00-63. The higher the value, the stronger the filter background color.
- For horizontal or vertical movement, it is recommended that the touch and display control keys are in the same direction.
- The initial icon displayed on power-on also requires CFG file 0x05, 5 digits write 1 to load 22 files, 0x05 digits can write 0x38, which means that 22 initial value files are loaded, data upload is enabled, and touch audio is enabled.

### 7.9.3 JPEG icon page pan setting considerations

(1) There can be multiple long-sliding slide pictures, ID range: 1000-1023. For example, there are three strip icons of 1000/1001/1002, the three icons can be generated as one ICL file, or three can be generated separately. When generating the ICL file, the JPEG file size of the DGUS software compressed ICON icon page cannot exceed the hardware limit: 252KB for T5L1 and 764KB for T5L2.



(2) The maximum size of a long bar icon is 4079\*4079, and it is recommended to be within 4000\*4000. The larger the area, the larger the volume. Sliding only supports vertical or horizontal sliding, and does not support 45-degree side sliding. Do the chart when the height of the long bar icon and the height of the sliding box can be made consistent, a little higher can be, try not to be too high.



- 3) The sliding icon selection touch control does not need to be exactly the same size as the sliding page translation, that is, the coordinate points do not need to be completely aligned.
- (4) Only the touch control is supported on the long bar sliding page, and the display control is not supported, and the button effect is not supported, because the No. 32 background image does not contain the 1000 long bar icon page.
- (5) On the long sliding page, the button value is usually used to return to the menu selection, and the key value can be uploaded. Combined with variable icons, it is used in such scenarios as menu selection and function selection. You can also use basic touch to make sliding selection to turn pages, but the number of sliding selection items is unpredictable when you use it for the first time, which will affect the selection to a certain extent, and you can choose it yourself when you use it.

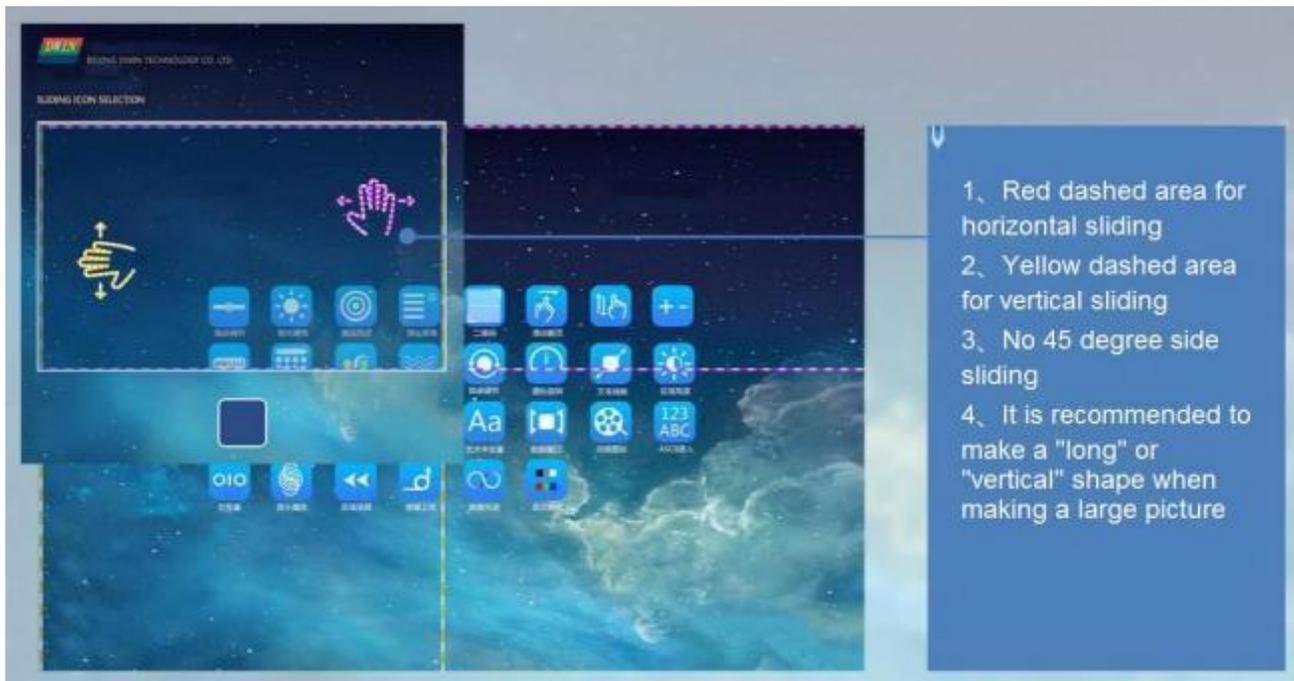
### 7.9.4 Example of Instructions



Set the variable address to 0x5830 and use the Return Key Code Control on the long bar icon.

- 5A A5 06 83 5830 01 0001; return key value 1
- 5A A5 06 83 5830 01 0002; return key value 2
- 5A A5 06 83 5830 01 0003; return key value 3

### 7.9.5 Effect



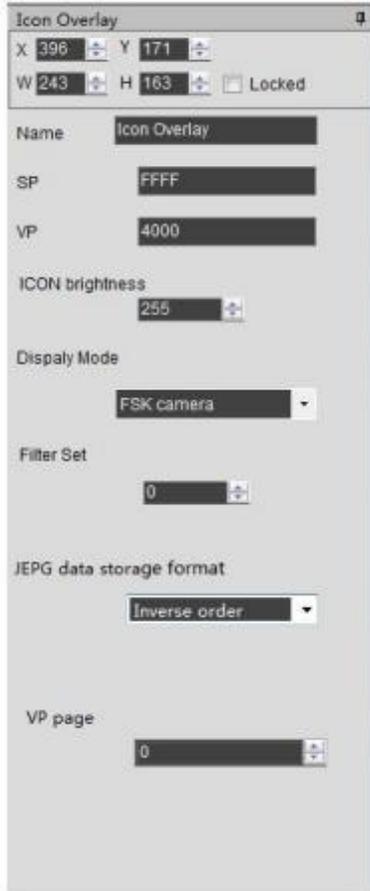


## 7.10 Icon Overlay

### 7.10.1 Instruction Storage Format

| Address | SP offset | Definition | Length (bytes) | Description   |
|---------|-----------|------------|----------------|---|
| 0x00    |           | 0x5A08     | 2              |   |
| 0x02    |           | *SP        | 2              | Variable description pointer. 0xFFFF means loaded by the configuration file.  |
| 0x04    |           | 0x0009     | 2              |   |
| 0x06    | 0x00      | *VP        | 2              | JPEG variable data pointer, must be even.<br>VP: write 0 x5 AA5 means JPEG data is valid and display is on. VP+ 1: JPEG data buffer word (Words) length, must be even. VP+ 2: JPEG data start.  |
| 0x08    | 0x01      | (x, y)     | 4              | The display position of the JPEG icon on the current page.  |
| 0x0C    | 0x03      | Wide_X     | 2              | The width of the display window of the JPEG icon on the current page.   |
| 0x0E    | 0x04      | Wide_Y     | 2              | The height of the display window of the JPEG icon on the current page.  |
| 0x10    | 0x05:H    | Dim_Set    | 1              | Display brightness of the overlaid JPEG icons, 0x00 (darkest) - 0xFF (brightest).   |
| 0x11    | 0x05:L    | Disp_Mode  | 1              | Display modes.<br>0 x00 = JPEG icon transparently displayed ( icon background is not displayed).<br>Other = JPEG icon background is displayed. <b>FSK bus camera display is 0x01.</b>   |
| 0x12    | 0x06:H    | Filter_Set | 1              | The intensity of the background color filtering is 0x01- 0x3 F when the JPEG icon is displayed transparently.   |
| 0x13    | 0x06:L    | *VP_Page   | 1              | <b>. 7 JPEG data storage format setting.</b><br><b>0=sequential (D3:D0=0xFFD8FFD0), 1=inverse order (D0:D3=0xFFD8FFE0).</b><br><b>FSK bus camera data is selected in inverse order.</b><br><b>.6-.4 reserved, write 0.</b><br><b>. 3-.0 variable memory page address, 0x00-0 x0F, which forms a 2 0 bit variable pointer together with * VP, corresponding to 2 MBbytes of variable memory space.</b> |
| 0x14    |           | Reserved   | 12             |   |

### 7.10.2 Software Setting



Coordinates of the control upper left corner

Control width and height

Set SP by default or actual needs.

Variable storage space range available to user: 0x1000-0xFFFF. Icon overlay occupies 120KB at most, and not relevant controls should be spaced out. System variable interface address range: 0x0000-0X0FFF.(e.g. page switch address 0x0084)  
Display the overlaid JPEG icon from the variable buffer to the specified area of the current page, and display brightness and transparency can be set. VP (must be even) = 5AA5 to turn on the display, vp+1 = JPEG data buffer word length (even), vp+2 = JPEG data start storage.

JPEG icon brightness: 0x00(darkest)-0xFF(brightest)

0x00 = JPEG icon transparently displayed (icon background is not displayed).  
Other = JPEG icon background is displayed.  
FSK bus camera display is 0x01.

The background color filtering is 0x01-0x3F when the JPEG icon is displayed transparently.

.7 JPEG data storage format setting.  
0=sequential (D3:D0=0xFFD8FFD0), 1=inverse order (D0:D3=0xFFD8FFE0).  
**FSK bus camera data is selected in inverse order.**  
.6-.4 reserved, write 0.  
.3-.0 variable memory page address, 0x00-0x0F, which forms a 20bit variable pointer together with \*VP, corresponding to 2Mbytes of variable memory space.

### 7.10.3 Effect

Camera module consists of OV2640 camera + T5L0 driver board + FSK bus.

The supporting camera module finished products have included OV2640 camera and die sinking. FDV200-1: no infrared. The left camera below is the actual object  
FDV200-2: with infrared.



Camera + smart screen physical front view



Camera + smart screen physical back view



### 7.10.2 Example of Instructions

Example1: Each instruction can be sent as 240- byte length data.

```
5A A5 F3 82 80 00 FF D8 FF E0 00 10 4A 46 49 96 00 01 01 01 00 60 00 60 00 00 FF DB 00 43 00 08 06 06 07 06
05 08 07 07 07 09 09 08 0A 0C 14 0D 0C 0B 0B 0C 19 12 13 0F 14 1D 1A 1F 1E 1D 1A 1C 1C 20 24 2E 27 20 22
2C 23 1C 1C 28 37 29 2C 30 31 34 34 34 1F 27 39 3D 38 32 3C 2E 33 34 32 FF DB 00 43 01 09 09 09 0C 0B 0C 18
0D 0D 18 32 21 1C 21 32 32 32 32 .....
```

( The DGUSII screen will have a response "5 A A5 03 82 4 F 4 B", and the next one can be sent immediately after receiving the response. In fact, the T5L uses a 20 ms refresh rate mechanism . In the case of using a baud rate of ≤ 115200, sending data continuously will not cause serial port busy processing)

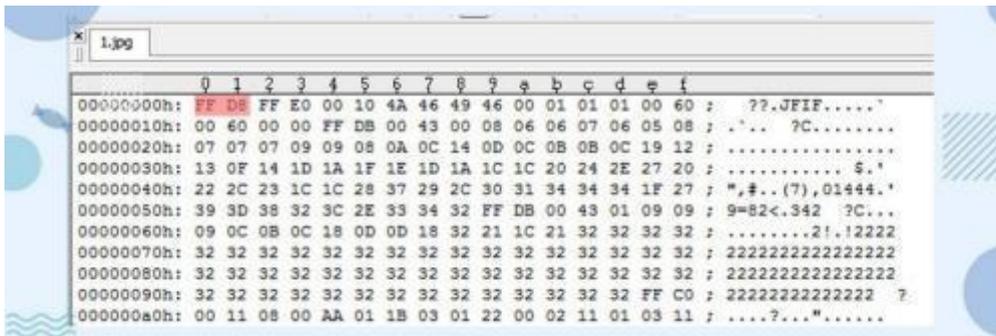
Example2: The address is incremented in sequence according to the word length of 120 (78 h).

```
5A A5 F3 82 80 78.....
```

Send several packets of data in turn...

The last line less than 240 bytes can be complemented with "00 00" at the end

The figure below is the specific data of the JPEG image viewed with " UltraEdit-32 " software.



Step 3: Send the display instruction.

```
5A A5 07 82 7F FE 5A A5 80 00
```

In fact, " 0x8000 " refers to the length of the buffer, JPEG data stream ( 0xFFD8 . . . 0xFFD9 ) file format, the bottom layer will judge and process automatically. User does not need to worry about the redundant data in the cache address later.

### 7.10.3 Effect





## 7.11 Batch Variable Icon

### 7.11.1 Instruction Storage Format

| Address | SP offset | Definition  | Length (Word) | Description  |
|---------|-----------|-------------|---------------|--|
| 0x00    |           | 0x5A09      | 2             |  |
| 0x02    |           | *SP         | 2             | Parameter pointer.<br>0xFFFF: Disables SP ( no run- time modification).  |
| 0x04    |           | 0x0006      | 2             |  |
| 0x06    | 0x00      | *VP         | 2             | Variable pointer, must be even.<br>D3 : 0x5A = enable the display, others = disable the display.<br>D2: displayed ICON ID, range 0-N.<br>D1 : display mode, only works for the display of background copy<br>. 7 filter switch 0= open filter, 1= display background.<br>6 reserved,0.<br>5-.0 filter set value 0x01-0x3F.<br>D0: brightness of ICON, only works for the display of background copy.<br>If the brightness of ICON is not set to 0xFF, the icon will display together with background, and the display speed will be slower about 30% . |
| 0x08    | 0x01      | *VP1        | 2             | The data corresponds to the icon display position description pointer, which must be an even number, and each icon has 2 addresses: D3:D0: coordinates (x, y) of the upper left corner of the icon display position;<br>The data is described in sequence according to the data source number (0-(Data_Num-1)).  |
| 0x0A    | 0x02      | *VP2        | 2             | Each icon data description pointer to be copied must be an even number, and each icon has 4 addresses:<br>D7:D4: coordinates of the upper left corner of the icon (x, y);<br>D3:D2: icon width pixel value;<br>D1:D0: icon height pixel value;<br>Data are described sequentially by icon number (0-N).  |
| 0x0C    | 0x03      | Data_Num    | 2             | *The number of data items corresponding to VP, 0x0000-0x1000, up to 4096 data items.   |
| 0x0E    | 0x04:H    | Icon_Source | 1             | 0x00: current display page (video memory mode).<br>Other: icon background image (Flash read).  |
| 0x0F    | 0x04:L    | Icon_Lib    | 1             | Icon background image mode only: The ICL file number of the icon background image.   |
| 0x10    | 0x05      | Icon_ID     | 2             | Icon background image mode only: ID number of the icon background image in the ICL file.   |
| 0x12    | 0x06      | Reserved    | 14            |  |

**Estimated processing time in icon background image mode** (Copy the specified area from the icon patch background image saved in Flash and display it to the current page):

$T = (\text{number of valid data items} + 5) * \text{image background image pixels} / 400 + \text{icon pixels corresponding to valid data items} / 200 \text{ uS.}$

Example:

Assuming that each data has 8 different icons to choose from. A single icon is 64 \* 64 pixels, spliced into a 560 \* 64 background image, and there are 100 data items to be displayed. Then the processing time is  $105 * 560 * 64 / 400 + 100$



$64 \times 64 / 200 = 11.456 \text{ mS}$ , In the video memory mode, it is 5.12 mS.

**Estimated processing time in video memory mode** (copy the specified area from the currently displayed page to the current page):

$T = \text{icon pixels corresponding to valid data items} / 80 \text{ uS}$ .

Example:

Assuming that each data has 8 different icons to choose from, a single icon is  $32 \times 32$  pixels, has been displayed on the current page, and there are 1000 data items to be displayed, then the processing time is  $1000 \times 32 \times 32 / 80 = 12.80 \text{ Ms}$ . In the icon back ground image mode, it is 30.848 mS.

In the video memory mode, the speed is fast, and it needs to occupy a certain area of the display area to display the icons to be copied, and restore the background after use.

In the icon background image mode, it is suitable when there are not too many data items and the resolution of the icon background image is not high.



## 7.12 Data Variables

The data variables control is to display a data variable according to the specified format (integer, decimal, whether with AICII unit or not) with Arabic numerals with specified font and size.

### 7.12.1 Instruction Storage Format

The instruction storage format is shown in the table.

| Address | SP offset | Definition         | Length (Word) | Description  |
|---------|-----------|--------------------|---------------|--|
| 0x00    |           | 0x5A10             | 2             |  |
| 0x02    |           | *SP4               | 2             | Variable description pointer. 0xFFFF means loaded by the configuration file.   |
| 0x04    |           | 0x000D             | 2             |  |
| 0x06    | 0x00      | *VP                | 2             | Variable pointer.  |
| 0x08    | 0x01      | X, Y               | 4             | The starting display position, which displays the coordinates of the upper left corner of the string.  |
| 0x0C    | 0x03      | COLOR              | 2             | Display Color  |
| 0x0E    | 0x04:H    | Lib_ID             | 1             | ASCII font location  |
| 0x0F    | 0x04: L   | Font size          | 1             | The number of dots in the x- direction of the character.   |
| 0x10    | 0x05:H    | Alignment          | 1             | 0x00=left-aligned 0x01=right-aligned 0x02=centered   |
| 0x11    | 0x05:L    | Integer digits     | 1             | Display integer digits. The sum of integer digits and decimal digits cannot exceed 20.   |
| 0x12    | 0x06:H    | Decimal places     | 1             | Display integer digits. The sum of integer digits and decimal digits cannot exceed 20.   |
| 0x13    | 0x06:L    | Variable data type | 1             | 0x00: Integer (2 bytes). Range: -32768 ~ 32767<br>0x01: 32-bit Integer (4 bytes). Range: -2147483648 ~ 2147483647<br>0x01 = long integer (4 bytes) in the range -2147483648 to 2147483647<br>0x02=* VP high byte, unsigned, range 0 to 255<br>0x03=*VP low byte, unsigned, range 0 to 255<br>0x04= very long integer (8 bytes), the range is -9223372036854775808 to 9223372036854775807<br>0 x05=unsigned integer (2 bytes), range 0 to 65535<br>0 x06=unsigned long integer (4 bytes) in the range 0 to 4294967295 |
| 0x14    | 0x07:H    | Len_unit           | 1             | Variable unit ( fixed string) display length, 0x00 means no unit display   |
| 0x15    | 0x07:L    | String_Unit        | Max11         | Unit string, ASCII encoding.   |

### 7.12.2 Software settings

The screenshot shows the 'Data variables display' configuration window. The settings are as follows:

- X: 434, Y: 173, W: 195, H: 82, Locked:
- Name: Data variables
- SP (0x): FFFF
- VP (0x): 5420
- Show colour: FFFF
- Word bank ID: 0
- Font size: 20 (range 4-255)
- Alignment: Left
- Automatic Character Spacing Adjustment
- Display Invalid Zero
- Variable type: Int (2 bytes)
- Integer digits: 9
- Decimal digits: 0
- Variable unit length: 0
- Show unit:
- Unit character string, ASCII code:
- Initial value: 0

Callout boxes provide the following explanations:

- Coordinates of the control upper left corner** (points to X and Y)
- Control width and height** (points to W and H)
- Set SP by default or actual needs** (points to SP)
- Variable storage space range available to user: 0X1000-0XFFFF. Data variables occupy 1/2/4 addresses depending on the variable type. That is, the addresses of non-associated controls need to be spaced apart. System variable interface address range: 0X0000-0X0FFF (e.g. page switch address 0X0084)** (points to VP)
- Set by default or actual needs** (points to Show colour)
- Set by default or actual needs** (points to Word bank ID)
- No.0 font library size: 4\*8-64\*128** (points to Font size)
- Set by default or actual needs** (points to Alignment)
- If checked, the character spacing will be reduced** (points to Automatic Character Spacing Adjustment)
- If checked, the instruction 5AA5 05 82 5420 0008 is displayed as 0008, otherwise it is 8** (points to Display Invalid Zero)
- Select the appropriate variable type according to the data range** (points to Variable type)
- If integer digit is 2 and decimal digits is 1, then the instruction 5AA5 05 82 0064 will displays 10.0.** (points to Integer and Decimal digits)
- Set by default or actual needs** (points to Variable unit length)
- The initial value displayed at power-up. Requires CFG file 0X05 .5 bit being 1 to load 22 file. For example, writes 0X38 in 0X05 to indicate loading 22 files, turning on data upload and touch .** (points to Initial value)



### 7.12.3 Example of Instructions

The variable address range is 0x1000-0xFFFF, and the instruction data is in hexadecimal format.

(1) Display integer data 100.

5AA5058254200064

0x5AA5A5: Frame header;

0x05: Data length, which is the 5 bytes of 82 54 20 00 64;

0x82: Write instruction;

0x5420: VP;

0x0064: Data 100. Integer range: -32768 to 32767, occupying 2 bytes.

(2) Display 3 data by one instruction. (The addresses must be continuous)

5A A5 09 82 53 00 00 64 00 63 00 62

0x5AA5A5: Frame header;

0x09: Data length, indicating the number of bytes containing data after 0x09; 0x82:

Write instruction;

0x5300: VP;

0x0064: Data 100 (to 0x5300 address);

0x0063: Data 99 (to 0x5301 address);

0x0062: Data 98 (to 0x5302 address).

(3) Display long integer data 100 .

5A A5 07 82 54 22 00 00 00 64

0xA5: Frame header;

0x07: Data length;

0x82: Write instruction;

0x5422: VP;

0x0000 0064: Long integer data 100. The long integer ranges from -2147473648 to 2147483647, which occupies 4 bytes, and the bits that are not enough are filled with 00.

(4) Display ultra-long integer data 100

5A A5 0B 82 54 24 00 00 00 00 00 00 00 64

0x5AA5: Frame header;

0x0B: Data length;

0x82: Write instruction;

0x5424: VP;

0x0000 0000 0000 0064: Data 100. The ultra-long integer range ranges from -9223372036854775808 to 9223372036854775807, which occupies bytes, and the bits that are not enough are filled with 00.



(5) Display single- precision floating- point number data

99 5A A5 07 82 54 28 42 C6 00 00

0x5AA5: Frame header;

0x07: Data length;

0x82: Write instruction;

0x5428: VP;

0x42C6 0000: Single-precision floating-point number 99, which can be converted by floating- point number conversion tool.

(6) Display double- precision floating- point number data 99

5A A5 0B 82 54 2B 40 58 C0 00 00 00 00 00

0x5AA5: Frame header;

0x0B: Data length;

0x82: Write instruction;

0x542B: VP;

0x4058: C000 0000 0000 Double-precision floating-point number 99, which can be converted by floating-point conversion tool.

(7) Display integer type data - 100 5A A5 05 82 53 00 FF 9C

0x5AA5: Frame header; 0x05: Data length; 0x82 instruction;

0x5300: VP;

0xFF9C: negative number 100 . The complement code of a negative number: The sign bit is 1, and the remaining bits are the bit wise NOT of the original code absolute value. Then the entire number is incremented by 1. If entering -200 , it will return 5A A5 06 83 68 20 01 FF 38, and the returned data 0xFF38 is the entered data -200.

(8) Display 2 decimal places

Sending 5A A5 05 82 5300 0064 will display 100.

Sets the integer digit of the data variables control to 1 and the decimal digit to 2.

(9) Change the data color

5A A5 05 82 9003 F800

0x5AA5: Frame header;

0x05: Data length;

0x82: Write instruction;

0x9003: means the SP is offset by 3 bits;

0xF800: Color code.

(10) Change the data coordinates

5A A5 07 82 900 10064 0064

0x5AA5: Frame header;

0x07: Data length;

0x82: Write instruction;

0x9001: Indicates that the SP is offset by 1 bit based on 0x9000. Refer to the 7.12.1 Instruction Storage Format for more details;

0x0064: The data is offset to the (100, 100) coordinates.

(11) Change the font size

5A A5 05 82 90040028

0x5AA5: Frame header;

0x05: Data length;

0x82: Write instruction;

0x9004: Indicates that the description pointer is offset by 4 bits based on 0x9000;

0x0028: The high byte 0x00 is the position of font 0, just write 0x00, and the low byte 0x28 is the font size.

(12) Change the integer digits to 3 digits

5A A5 05 82 90050003

0x5AA5: Frame header;

0x05: Data length;

0x82: Write instruction;

0x9005: Indicates that the description pointer 0x9000 is offset by 5 bits;

0x0003: The high byte 0x00 is left-justified, and the low byte 0x03 is an integer number of bits.

(13) Change the number of decimal places to 2

5A A5 05 82 9006 0200

0x5AA5: Frame header;

0x05: Data length; 0x82: Write instruction;

0x9006: Indicates that the description pointer is offset by 6 bits based on 0x9000;

0x0200: The high byte 0x02 is the number of decimal places, and the low byte 0x00 indicates that the data type is an integer.



## 7.13 Text Display

The text display control is to display the character string in the specified text box display area according to the specified format (determined by selecting the font library).

This control is usually used with the text input control.

### 7.13.1 Instruction Storage Format

| Address | SP offset | Definition       | Length (Word) | Description   |
|---------|-----------|------------------|---------------|---|
| 0x00    |           | 0x5A11           |               |   |
| 0x02    |           | *SP              |               | Variable description pointer. 0xFFFF means loaded by the configuration file.  |
| 0x04    |           | 0x000D           |               |   |
| 0x06    | 0x00      | *VP              | 2             | Text pointer.   |
| 0x08    | 0x01      | (X, Y)           | 4             | The starting display position, which displays the coordinates of the upper left corner of the string.   |
| 0x0C    | 0x03      |                  |               | Display text color  |
| 0x0E    | 0x04      | (Xs, Ys)(Xe, Ye) | 8             | Text box.   |
| 0x16    | 0x08      | Text_Length      | 2             | Display the number of bytes, when it encounters 0xFFFF, 0x0000 or displays to the end of the text box, it will no longer be displayed.  |
| 0x18    | 0x09:H    | Color            | 2             | The font location used by ASCII characters when the encoding method is 0x01-0x04.   |
| 0x19    | 0x09:L    | (Xs, Ys)(Xe, Ye) | 8             | Font location for non-ASCII characters when the encoding is 0x00, 0x05, and 0x01-0x04.  |
| 0x1A    | 0x0A:H    | Text_Length      | 2             | The number of dots in the x direction of the font (0x01- 0x04 mode, the number of dots in the x direction of ASCII characters is calculated as x/2).  |
| 0x1B    | 0x0A:L    | Font0_ID         | 1             | The number of dots in the Y direction of the font.  |
| 0x1C    | 0x0B:H    | Encode_Mode      | 1             | Character spacing mode is defined by bit<br>.7 = 0 character spacing is automatically adjusted;<br>.7 = 1 The character spacing is not automatically adjusted, and the character width is fixed to the set number of dots.<br>.6-.0 defines how the text is encoded:<br>0 = 8 bit code 1=GB2312 Inner code 2=GBK 3=BIG5 4=SJIS 5= UNICODE |
| 0x1D    | 0x0B:L    | HOR_Dis          | 1             | Characters are spaced horizontally.   |
| 0x1E    | 0x0C:H    | VER_Dis          | 1             | Characters are spaced vertically.   |
| 0x1F    | 0x0C:H    | Undefined        | 1             | Write 0x00.   |

### 7.13.2 Software Setting

Open DGUS, click display control - text display, then set the control display area, and configure parameters in the setting menu on the right.

The number of dots in the Y direction of the font must be even.

DGUS screen is pre-installed with 0# font library, including all ASCII characters of 4\*8~64\*128 dot matrix.

**Text display**

- X 102 Y 155**: Coordinates of the upper left corner of the display area
- W 207 H 207**: Display area width and height
- Name definition Text**: Name of the text display
- Description pointer(0x) FFFF**: The pointer address is set by default or according to need.
- Variable address(0x) 5020**: The variable storage space user can use the address range arbitrarily: 0x1000-0xFFFF. The text display occupies the address according to the length of the text setting. That is, the non-associated control key addresses need to be spaced apart. System variable interface address range: 0x0000-0x0FFF (such as page switching address 0x0084)
- Display color(0x) FFFF**: Color of the text display
- Encoding 0x02=GBK**: The encoding method is determined when the font is generated, and the factory font encoding is GBK.
- Character spacing is not automatically adjusted**: Checked: DWIN 1 2 Unchecked Effect: DWIN 12
- Text length 200**: Take bytes as the unit, here is set to display 200-byte length data. The text display can display up to 512 (words, Word).
- FONT0\_ID 0**: ASCII font location, default font 0 or set according to needs.
- FONT1\_ID 23**: Non-ASCII font location, set the font ID according to your needs. Put it in the DWIN\_SET folder in advance. For example, 23\_GBK24 Song Ti: It means that the font ID is 23, 24\*24 dot matrix, and the encoding method is GBK, Simsun.
- Number of dots in X direction 24**: The size of the dot matrix in the X and Y directions of the font library cannot be set arbitrarily. Otherwise, it will display garbled characters. For example, the font library No. 23 above is a 24\*24 dot matrix, which is the setting parameter for this. The number of dot matrix in the Y direction must be an even number when generating the font library.
- Number of dots in Y direction 24**: Same as X direction.
- Horizontal interval 0**: Horizontal interval 0 display effect: Horizontal interval 20 display effect: The vertical interval is the vertical interval.
- Vertical interval 0**: Same as horizontal interval.
- Initial value**: The initial icon displayed on power-on requires CFG file 0x05.5 bit to write 1 to load 22 files, 0x05 bit can write 0x38, which means to load 22 initial value files, data upload, and start touch audio.

### 7.13.3 Example of Instructions

① Display "North 12 AB"

5AA5 0B82 5020 B1B1 3132 4142 FFFF

0x5AA5: Frame header;

0x0B: Data length;

0x82: Write instruction;

0x5020: VP;

0xB1B1: ASCII code of "North";

0x3132: ASCII code of "12";

0x4142 "AB" ASCII code. The ASCII code can be converted by an ASCII conversion tool. 0xFFFF is the end character, and the data after the end character is no longer displayed.

② New line display.

5A A5 05 82 5023 0D0A

0x5AA5: Frame header;

0x05: Data length;

0x82: Write instruction;

0x5023 start from the address and display in a new line;

The ASCII code of 0x0D0A line feed can be converted by an ASCII conversion tool.

③ Clear the text display.

5A A5 07 82 5020 2020 2020

0x5AA5: Frame header;

0x07: Data length;

0x82: Write instruction;

0x5020: VP;

0x20: ASCII code of space, 2020 clears the data of an address. Clear the data of 2 addresses starting from address 5020.

④ Change the text color

5A A5 05 82 8803 F800

0x5AA5: Frame header;

0x05: Data length;

0x82: Write instruction;

0x8803: description pointer 0x8800 base address offset by 3 bits to change the color; for the offset address, please refer to the SP description pointer offset in the text display instruction storage format table; for the description of the pointer usage, please refer to the section 7.12 Data Variables.

0xF800: Color value.



⑤Hide text

Method 1: Hide

5A A5 05 82 8800 FF00

0x5AA5: Frame header;

0x05: Data length;

0x82: Write instruction;

0x8800: SP;

0xFF00: Hidden text.

Display after hiding:

5A A5 05 82 8800 8070

0x5AA5: Frame header;

0x05: Data length;

0x82: Write instruction;

0x8800: SP;

0x8070: VP.

Method 2 : Modify the coordinates

5A A5 07 82 8801 02D0 02D0

0x5AA5: Frame header;

0x05: Data length;

0x82: Write instruction;

0x8801: SP;

0x8800: SP 0x8000 base address offset by 1 bit to change the display coordinates;

0x02D0: (720,720) coordinate point, moved beyond the boundary to a invisible place.

Display after hiding:

5A A5 07 82 8801 0064 0064

0x5AA5: Frame header;

0x05: Data length;

0x82: Write instruction;

0x8801: SP;

0x8000 base address offset by 1 bit to change the display coordinates;

0x0064: (100, 100) coordinate point, move beyond the boundary to a visible place.



Method 3: Change the display length to 0000

5A A5 05 82 8808 0000

0x5AA5: Frame header;

0x05: Data length;

0x82: Write instruction;

0x8808: SP;

0x8000 base address offset by 8 bit to change the display coordinates;

0x0000 displays 0 bytes of data.

Display after hiding:

5A A5 05 82 8808 0064

0x5AA5: Frame header;

0x05: Data length;

0x82: Write instruction;

0x8808: SP;

0x8000 base address offset by 8 bit to change the display coordinates;

0x0064 displays 100 bytes of data.

#### 7.13.4 Effect





## 7.14 RTC Display

The RTC display control is to display the Gregorian calendar RTC in text according to the format edited by the user.

### 7.14.1 Text RTC Instruction Storage Format

(1) Display current date and/or time, in digital format by the internal RTC.

| SP offset | Definition  | Length (bytes) | Description   |
|-----------|-------------|----------------|---|
| 0x00      | 0x0000      | 2              | 0x0000  |
| 0x01      | (X, Y)      | 4              | Display position, display the coordinates of the upper left corner of the string.   |
| 0x03      | Color       | 2              | Text color.   |
| 0x04:H    | Lib_ID      | 1              | Index in the FLASH memory of the ASCII font library.  |
| 0x04:L    | Font_Width  | 1              | Font width, in pixels.  |
| 0x05      | String_Code | MAx16          | Display format string. Use ASCII characters and the Field Codes on the following table.<br>E. g. If current time =2012-05-02 12:00:00 Wednesday, then in Y-M-D H: Q: S 0x00 mode, it will display "2012-05-02 12:00:00" .<br>And in M-D W H: Q 0x00 mode, it will display "05-02 WED 12:00" . |

(2) RTC code.

| Explanation       | Code | Display Format              |
|-------------------|------|-----------------------------|
| Gregorian_ Year   | Y    | 2000-2099                   |
| Gregorian_ Month  | M    | 01-12                       |
| Gregorian_ Day    | D    | 01-31                       |
| Gregorian_ Hour   | H    | 00-23                       |
| Gregorian_ Minute | Q    | 00-59                       |
| Gregorian_ Second | S    | 00-59                       |
| Gregorian_ Week   | W    | SUN MON TUE WED THU FRI SAT |
| End of encode     | 0x00 |                             |

### 7.14.2 Text RTC Software Setting

Open DGUS, click display control – RTC display, then set the control display area, and configure the font library and time format in the setting menu on the right.

The time can be modified through the RTC setting or using UART instructions.

RTC display

X 281 Y 216

W 178 H 66

Name definition **RTC**

Description pointer(0x) **FFFF**

Font color **0000**

Font library location **0**

Number of dots in X direction **16** (4-255)

Date format:  
**Y-M-D H:Q:S W**

Use RTC encoding and ASCII character composition

RTC code table:  
illustrate coding  
Gregorian\_year Y  
Gregorian\_Month M  
Gregorian\_Day D  
Gregorian\_Hour H  
Gregorian\_Minute Q  
Gregorian\_second S  
Gregorian\_week W

Coordinates of the upper left corner of the display area

Display area width and height

The pointer address is set by default or according to need.

Set by default or according to need

Set by default or according to need

Display data size

Separate date display settings: do two RTC displays,  
The first date format deletes the hour, minute, second and week, and the second date format deletes the year, month, and day.

To display in this way, display 6 RTCs, and display the Chinese year, month, day, hour, minute, and second on the base map.

The first control key is reserved for Y years; the second control key is reserved for M months; the third control key is reserved for D days;  
When the fourth control key only keeps H, the fifth control key only keeps hour 0 minutes; the sixth control key only keeps hour S minutes;

### 7.14.3 Text RTC Effect



### 7.14.4 RTC Display(watch)

The RTC display (watch) uses icon rotation to display the RTC of the Gregorian calendar.

#### 7.14.4.1 Instruction Storage Format

| Address | SP offset | Definition          | Length (Word) | Description  |
|---------|-----------|---------------------|---------------|--|
| 0x00    |           | 0x5A12              |               |  |
| 0x02    |           | *SP                 |               | Variable description pointer. 0xFFFF means loaded by the configuration file.               |
| 0x04    |           | 0x000D              |               |  |
| 0x06    | 0x00      | 0x0001              | 2             |  |
| 0x08    | 0x01      | (X, Y)              | 4             | The center of a clock dial.  |
| 0x0C    | 0x03      | Icon_Hour           | 2             | ID of pointer ICON, 0xFFFF indicates that the hour hand is not displayed.                  |
| 0x0E    | 0x04      | Icon_Hour_Central   | 4             | Position of the center of rotation of the clockwise icon.                                  |
| 0x12    | 0x06      | Icon_minute         | 2             | ID of the minute hand ICON. 0xFFFF indicates that the minute hand is not displayed.        |
| 0x14    | 0x07      | Icon_Minute_Central | 4             | Rotation center position of minute hand ICON.  |
| 0x18    | 0x09      | Icon_Second         | 2             | The ID of the second hand icon;<br>0xFFFF indicates that the second hand is not displayed. |
| 0x1A    | 0x0A      | Icon_Second_Central | 4             | Position of rotation center of second hand ICON.   |
| 0x1E    | 0x0C:H    | Icon_Lib            | 1             | ID of the icon library file where the pointer icon resides                                 |
| 0x1F    | 0x0C:L    | Undefined           | 1             | Write 0x00   |

#### 7.14.4.2 Software Setting

The screenshot shows the 'Dial clock Settings' window with the following annotations:

- X 382 Y 332**: The coordinates of the upper left corner of the display area
- W 140 H 148**: Display area width and height
- Name definition: Clock Display**: Default or set as needed
- Description pointer(0X) FFFF**: Default or set as needed
- Icon file: 48\_icon.icl**: It is recommended to start at position 48 and aggregate it into a .icl file. Pre-generated into the DWIN-SET folder.
- Hour hand icon: FFFF**: Hour hand icon while loading.
- Hour hand center coordinates: 10 115**: Hour hand icon while loading.
- Minute hand icon: 49**: Hour hand icon while loading.
- Minute hand center coordinates: 9 116**: Hour hand icon while loading.
- Second hand icon: 4A**: Hour hand icon while loading.
- Second hand center coordinates: 9 114**: Hour hand icon while loading.

The 'Hour hand icon while loading' section includes a sub-screenshot showing a clock dial with numbered points 1 through 5, where 1 is the center, 2 is the rotation center, 3 is the icon, 4 is the center of rotation, and 5 is the icon.



### 7.14.4.3 Example of Instruction

Display time October 1, 2050 11: 12: 13

5A A5 0B 82 009C 5A A532 0A01 0B0C0D

0 x5AA5: Frame header;

0x0B Data length;

0x82: Write instruction;

0x009C: VP; this address is fixed and cannot be customized. For details, please refer to the section 7. 0x9C address definition;

0x5AA5: Start RTC setting once. For details, please refer to the section 7.1 0x9C address definition

0x32 year, 0x0A month, 0x01 day, 0x0B hour, 0x0C minute, 0x0D second.

### 7.14.4.4 Effect





## 7.15 HEX Data

The HEX data control displays variable data in byte HEX intervals with user-specified ASCII characters. It is mostly used for timing display.

For example, 1234 is displayed as 12:34.

### 7.15.1 Instruction Storage Format

| Address | SP offset | Definition  | Length (Word) | Description   |
|---------|-----------|-------------|---------------|---|
| 0x00    |           | 0x5A13      | 2             |   |
| 0x02    |           | *SP         | 2             | Variable description pointer. 0xFFFF means loaded by the configuration file.  |
| 0x04    |           | 0x000D      | 2             |   |
| 0x06    | 0x00      | *VP         | 2             | Variable data string pointer.   |
| 0x08    | 0x01      | X,Y         | 4             | Display the starting position, and display the coordinates of the upper left corner of the string.  |
| 0x0C    | 0x03      | Color       | 2             | Font color.   |
| 0x0E    | 0x04:H    | Mode        | 1             | .7: Data BCD code adjustment enable, 0=adjustment off, 1=adjustment on.<br>When the BCD code adjustment is enabled, 0x0A will be adjusted to 0x10 and displayed as 10.<br>.6-.4: Reserved, write 0.<br>.3-.0: The number of bytes displayed from the high byte of *VP pointer, 0x01-0x0F  |
| 0x0F    | 0x04:L    | Lib_ID      | 1             | Font location; the font must be in half-width format.<br>If Lib_ID is not 0, the font must use 8bit encoding.   |
| 0x10    | 0x05:H    | Font_x      | 1             | The number of dots in the X direction.  |
| 0x11    | 0x05:L    | String_Code | MAX15         | The encoding string is used to combine with the time variable to produce the display format required by the client.<br>After each BCD time code is displayed, an ASCII character will be taken out in sequence from the encoded string characters to display at intervals.<br>In encoded strings, special characters are defined as follows:<br>0x00: invalid, this character will not be displayed, and the two BCD time codes will be displayed together;<br>0x0D: Newline display, that is, X=Xs, Y=Y+Font_X*2 |

### 7.15.2 Software Setting

|                                     |   |
|-------------------------------------|---|
| HEX VAR                             |   |
| X 144 Y 299                         | Coordinates of the upper left corner of the display area  |
| W 438 H 117                         | Display area width and height   |
| Name definition HEX VAR             | The pointer address is set by default or according to need.   |
| Description pointer(0x) FFFF        | The variable storage space user can use the address range arbitrarily; 0×1000-0×FFFF. The time variable occupies a maximum of 8 address spaces. That is, the non-associated control key addresses need to be spaced apart. System variable interface address range: 0×0000-0×0FFF (such as page switching address 0×0084) |
| Variable address(0x) 5434           |   |
| Display color(0x) FFFF              | Set by default or according to need   |
| Font library location 0             | Set by default or according to needs. If the font location is not 0, the font must use 8-bit encoding.  |
| Font size 20 4-255                  | The number of dots in the X direction of the font. Size range of dot matrix of font 0: 4*8-64*128   |
| Byte_Num 4                          | Byte length, range: 0×00-0×0F   |
| BCD code adjustment enable<br>Close | Send command: 5A A5 07 82 0B0B 0B0B<br>Turn off the display effect: 0B:0B:0B:0B<br>Turn on the display effect: 11:11:11:11  |
| Encoding string<br>:<br>Conversion  | Enter the ASCII characters to be displayed. Click Convert, and the corresponding ASCII code will be generated.  |
| Input hexadecimal data<br>3A3A      |   |

### 7.15.3 Example of Instruction

5A A5 07 82 5434 10 11 12 13

0x5AA5: Frame header;

0x07: Data length;

0x82: Write instruction;

0x5434: VP;

0x10 11 12 13: BCD code.

### 7.15.4 Effect





## 7.16 Text Scroll

Text scroll function is the variable data scroll display in the specified area with the specified direction .

### 7.16.1 Instruction Storage Format

| SP offset | Definition   | Length (bytes) | Description  |
|-----------|--------------|----------------|--|
|           | 0x5A14       | 2              |  |
|           | SP           | 2              | The variable describes the pointer.  |
|           | 0x000C       | 2              |  |
| 0x00      | VP           | 2              | Text pointer, <b>must be an even address.</b><br><b>The first 3 words of the text pointer must be reserved, and the user display text content is stored from (VP+3). Text must end with 0xFFFF or 0x0000.</b>  |
| 0x01:H    | Rolling_Mode | 1              | Scroll 0x00: from right to left. 0x02: from bottom to top, <b>each line must end with 0x0D0A</b> as line division.   |
| 0x01:L    | Rolling_Dis  | 1              | Rolling space in a DGUS cycle, in pixel.   |
| 0x02:H    | Adjust_Mode  | 1              | 0x00: Left-alignment. 0x01: Right-alignment. 0x02: Center-alignment.<br>Rolling will stop when the text length smaller than text box.<br>For scroll mode 0x02:<br>The lower 4 bits are horizontal alignment mode: 0x0=left alignment<br>0x1=right alignment 0x2=center.<br>The high 4bit is the vertical alignment mode when stopped: 0x0=top alignment 0x1=bottom alignment 0x2=center. |
| 0x02:L    | Run_Control  | 1              | Run-control:<br>0x00=Normal scrolling 0x01=Pause 0x02=Off 0x03=Initialization (static display)<br>0x04=Continuous scrolling, no matter whether the content length is shorter than the text box or not.   |
| 0x03      | Color        | 2              | Text color.  |
| 0x04      | Xs Ys Xe Ye  | 8              | Text box.  |
| 0x08:H    | Font0_ID     | 1              | The location of ASCII characters in the font library when the encoding mode is 0x01-0x04.<br>If the encoding mode is 0x00 or 0x05, do not set this parameter and write 0x00.   |
| 0x08:L    | Font1_ID     | 1              | The font location of non-ASCII characters when the encoding mode is 0x01 - 0x04.<br>When the encoding mode is 0x00 or 0x05 , the location of the character library used by the character is displayed.   |
| 0x09:H    | Font_X_Dots  | 1              | Font width, in pixels. Must be even. In encoding modes 0x01-0x04, must be twice the Width.   |
| 0x09:L    | Font_Y_Dots  | 1              | Number of font lattice in Y direction.   |
| 0x0A:H    | Encode_Mode  | 1              | Character spacing mode is defined by bit<br>7: 0: Automatic spacing.<br>7.1: Fixed spacing.<br>Font encoding is defined by bits 6 to 0<br>0x00:8 bit. 0x01: GB2312. 0x02: GBK. 0x03: BIG5 0x04: SJIS 0x05: UNICODE.  |
| 0x0A:L    | Text_Dis     | 1              | Spacing between characters, in pixels.   |
| 0x0A:H    | Speed_Set    | 1              | Scroll speed setting, 0x00-0x0F, 0x00 is the fastest, 0x0F is the slowest.   |
| 0x0B:H    | Undefined    | 3              | 0x00   |

### 7.16.2 Software setting

|   |  |
|---|--|
| Roll Text   |  |
| X 533 Y 94  | Coordinates of the upper left corner of the display area   |
| W 157 H 48  | Display area width and height  |
| Name definition RollText  |  |
| Description pointer(0x) FFFF  |  |
| Variable address(0x) 6010   | The variable storage space user can use the address range arbitrarily: 0x1000-0xFFFF. The text display occupies the address according to the length of the text setting. That is, the non-associated control key addresses need to be spaced apart. System variable interface address range: 0x1000-0xFFFF (such as page switching address 0x0084). The first 3 words of the text pointer must be reserved. The text content displayed by the user is stored from (VP+3). The text must end with 0xFF or 0x00. |
| Display color(0x) 0000  |  |
| Encoding 0x02=GBK   | The encoding method is determined when the font is generated, and the factory font encoding is GBK.  |
| <input checked="" type="checkbox"/> Character spacing is not automatically adjusted | Unchecked effect: DWIN 12    Checked effect: DWIN 12   |
| Scrolling mode 0x00=  | The text is displayed from right to left   |
| Scroll pitch 2  | The number of pixel dots for text scrolling in each DGUS cycle, 0 to stop scrolling  |
| Alignment 0x00= Align left  | The length of the set text content is greater than the length of the display box, otherwise the scrolling will stop. The alignment is valid when the text content is less than the length of the display box.  |
| FONT0_ID 0  | The position of the ASCII font library can be the default font size 0.   |
| ASCII font position when encoding method is 0x01-0x04                               |  |
| FONT1_ID 23   | Non-ASCII font location, set the font ID according to your needs. Put it in the DWIN_SET folder in advance, For example, 23_GBK24 Song Ti: It means that the font ID is 23, 24*24 dot matrix, and the encoding method is GBK, SIMSUM.  |
| Non-ASCII characters with encoding methods 0x00, 0x05 and 0x01-0x04 use fonts       |  |
| X direction array 24  | The size of the dot matrix in the X and Y directions of the font library cannot be set arbitrarily. Otherwise, it will display garbled characters. For example, the font library No. 23 above is a 24*24 dot matrix, which is the setting parameter for this. The number of dot matrix in the Y direction must be an even number when generating the font library.   |
| Number of dots in Y direction 24  |  |
| Character interval 0  | Character interval 0 display effect: <b>DWIN</b><br>Character interval 20 display effect: <b>D W I N</b>   |
| Initial value   | The initial text displayed when the power is on, 0xB8 can be written in the *08 position of the CFG file to enable data upload.  |

### 7.16.3 Examples of Instructions

5A A5 13 B2 6013 BBB6 D3AD C0B4 B5BD B5CF CEC4 BFC6 BCBC

0x5AA5: Frame header:

0x 13: Data length;

0x82: Write instruction

0x6013: VP +3, address 0x6010 cannot be used directly,

0xBBB6 D3AD C0 B4 B5 BD B5 CF CEC4 BFC6 BCBC, the ASCII code of "Welcome to DWIN Technology".



### 7.16.4 Effect





## 7.17 Data Window

The data window control displays the data variables in a specified display window, highlighting the selected values.

### 7.17.1 Instruction Storage Format

| Address | SP offset | Definition   | Length (Word) | Description   |
|---------|-----------|--------------|---------------|---|
| 0x00    |           | 0x5A15       | 2             |   |
| 0x02    |           | *SP          | 2             | Variable description pointer. 0xFFFF means loaded by the configuration file.  |
| 0x04    |           | 0x000D       | 2             |   |
| 0x06    | 0x00      | *VP          | 2             | Data pointer, each data occupies 2 words storage space, defined as follows:<br>VP=the storage address of the selected data, and the stored data is a fixed-point integer.   |
| 0x08    | 0x01      | V_Min        | 2             | Data lower limit, fixed-point integer.  |
| 0x0A    | 0x02      | V_Max        | 2             | Data upper limit, fixed-point integer.  |
| 0x0C    | 0x03:H    | NUM_I        | 1             | Display integer digits, 0x00-0x05.  |
| 0x0D    | 0x03L     | NUM_F        | 1             | Display decimal places, 0x00-0x05.  |
| 0x0E    | 0x04:H    | NUM_Digit    | 1             | Data window height: the number of all data groups displayed, must be an odd number, 0x03-0x07.  |
| 0x0F    | 0x04:L    | Display_mode | 1             | Display mode:<br>.0 1=Invalid 0 is displayed; 0=Invalid 0 is not displayed.<br>.1 1=Positive "+" display; 0=Positive "+" not displayed.<br>.2 1=Turn around after the data crosses the border (cycle) 0=Stop when the data crosses the border<br>.3 1=automatic adjustment of display spacing 0=no adjustment of display spacing<br>.4-.7 Undefined, write 0. |
| 0x10    | 0x05      | (X,Y)        | 4             | Displays the center coordinates of the selected value.  |
| 0x14    | 0x07      | Step_Adj     | 2             | Adjustment step size, positive integer, 0x0001-0x7FFF.  |
| 0x16    | 0x08:H    | Font0_X_Dots | 1             | Font size of unselected data: the number of dots in the X direction, 0x04-0x40.   |
| 0x17    | 0x08:L    | Font0_Y_Dots | 1             | The number of lattice points occupied by the Y direction of the unselected data.<br>0x08-0xFF, not less than 2*Font0_X_Dots.  |
| 0x18    | 0x09      | Font0_Color  | 2             | Unchecked data display color.   |
| 0x1A    | 0x0A:H    | Font1_X_Dots | 1             | The font size of the selected data: the number of dots in the X direction, 0x04-0x40.   |
| 0x1B    | 0x0A:L    | Font1_Y_Dots | 1             | The number of lattice points occupied by the Y direction of the selected data.<br>0x08-0xff, not less than 2*Font1_X_Dots.  |
| 0x1C    | 0x0B      | Font1_Color  | 2             | The selected data display color.  |
| 0x1E    | 0x0C      | Font_Lib     | 1             | Font selection, the default is 0x00.  |
| 0x1F    | 0x0D      | Undefined    | 1             | Write 0x00.   |

### 7.17.2 Software Setting

**Data Window**  
 X 425 Y 174  
 W 171 H 365

**Name definition** Data Window

**Description pointer(0x)** FFFF

**Variable address(0x)** 5852

**Data lower limit** -100

**Data upper limit** 100

**Integer digits** 3

**Decimal places** 0

**Display mode**  
 Invalid display0  
 Display positive +number  
 U-turn after the data is out of bounds

**Adjust the step length** 1

**Data characteristics**  
 Data not selected: 20, 40, FFFF  
 Selected data: Point size in X direction 30, Point size in Y direction 60, Display color 07E0

**Callouts:**  
 - Coordinates of the upper left corner of the display area  
 - Display area width and height  
 - The pointer address is set by default or according to need.  
 - The variable storage space user can use the address range arbitrarily: 0x1000-0xFFFF. The data window indicates that it occupies 2 space addresses. That is, the non-associated control key addresses need to be spaced apart. System variable interface address range: 0x0000-0xFFFF (such as page switching address 0x0084)  
 - Data upper and lower limit range  
 - For example, if the integer digit is set to 2, the decimal place is set to 1, it will display 2 integers and 1 decimal. Write command: 5A A5 05 82 5420 0064 will be displayed as 10.0  
 - Number of selected and unselected data displayed  
 - Set the integer digit to 3 digits, check the effect: 000, 001, 002; uncheck the effect: 0, 1, 2  
 - A "+" sign will be displayed before a positive number is checked.  
 - For example, after setting the data above to display 100, it will turn around from -100 to display.  
 - The span of data increase or decrease once  
 Step 1 effect: 1, 2, 3, 4 changes; Step 2 effect: 2, 4, 6, 8 changes;  
 The dot matrix size of the data.  
 - The selected data can be set to a larger point for highlighting



### 7.17.3 Examples of Instructions

5A A5 05 82 5652 0064

0x5AA5: Frame header

0x05 Data length

0x82: Write instruction

0x5652: VP

0x0064: Write data 100 to VP

### 7.17.4 Effect





## 7.18 DGUSII Text(Text Display Without Aliasing)

### 7.18.1 Instruction Storage Format

| Address | SP offset | Definition    | Length (Word) | Description  |
|---------|-----------|---------------|---------------|--|
| 0x00    |           | 0x5A16        | 2             |  |
| 0x02    |           | *SP           | 2             | Variable description pointer. 0xFFFF means loaded by the configuration file.   |
| 0x04    |           | 0x000D        | 2             |  |
| 0x06    | 0x00      | *VP           | 2             | Text pointer, the maximum text length is 4Kbytes.  |
| 0x08    | 0x01      | (X, Y)        | 4             | Text display position: left- aligned mode, the coordinates of the upper left corner of the string.   |
| 0x0C    | 0x03      | Color         | 2             | Display text color.  |
| 0x0F    | 0x04      | Undefined     | 2             | Write 0x00   |
| 0x10    | 0x05      | Xs, Ys Xe, Ye | 8             | Text box.  |
| 0x18    | 0x09      | Text_Length   | 2             | Display the number of bytes, when it encounters 0xFFFF, 0x0000 or displays to the end of the text box, it will no longer be displayed.   |
| 0x1A    | 0x0A      | LIB_ID        | 2             | Font location for non- ASCII characters when the encoding is 0x00, 0x05, and 0x01 - 0x04.  |
| 0x1C    | 0x0B:H    | Display_Mode  | 1             | .7 defines whether the character spacing of the text display is automatically adjusted.<br>.7 = 0 Character spacing is automatically adjusted.<br>.7 = 1 Character spacing is not automatically adjusted, character width is fixed.<br>.6 - .0 defines the scale of the character display, 0x00-0x7F, unit 1/ 16.<br>In fact, DGUS divide 155 steps according to 0.25-8.0 in 0.05 steps, and then converted to 1/ 16 data.<br>The data is converted to 1/ 16 corresponding to 0x04-0x7F. |
| 0x1D    | 0x0B:L    | HOR_Dis       | 1             | Characters are spaced horizontally.  |
| 0x1E    | 0x0C:H    | VER_Dis       | 1             | Characters are spaced vertically.  |
| 0x1F    | 0x0C:L    | Align_Mode    | 1             | Alignment mode: 0x00=left alignment 0x01=right alignment 0x02=center<br>Right-aligned and centered modes are only valid for single-line display.   |

### 7.18.2 Software Setting

The screenshot shows the 'DGUSII text display' configuration window. The settings are as follows:

- X:** 138, **Y:** 154 (Coordinates of the upper left corner of the display area)
- W:** 278, **H:** 155 (Display area width and height)
- Name definition:** Text (Default or set as needed)
- Description pointer:** FFFF (Default or set as needed)
- Variable address:** 6050 (The variable storage space user can use the address range arbitrarily: 0x1000-0xFFFF. The text display occupies the address according to the high length of the text. That is, the addresses of non-associated keys need to be spaced apart. System variable interface address range: 0x0000-0x0FFF (such as page switching address 0x0084).)
- Display color:** FFFF
- Character spacing:** is not automatically adjusted (Unchecked the effect: DWIN 12; Checked the effect: DWIN)
- Text length:** 100 (The number of bytes displayed, the maximum is 0x1000. Encountered 0xFFFF, 0x0000 data or displayed to the end of the text box will no longer be displayed.)
- FONT\_ID:** 34 (Grayscale font name ID, put it in the DWIN-SET folder in advance, such as the font library on the right: the font ID is 34, 20\*20 dot matrix, the encoding method is GBK, Song type. Because there is only one page of pictures, and it is close to the 32.ICL file, if there are many pictures, it is necessary to calculate the interval naming in a single font space of 256KB to avoid conflicts between fonts.)
- Text scaling:** 100% (100% is the lattice at the time of generation. If the upper part is 20\*20, T5L does not support scaling.)
- Horizontal interval:** 0 (Horizontal interval 0 display effect: DWIN Technology; Horizontal interval 20 display effect: DWIN Technology)
- Vertical spacing:** 0 (Vertical spacing is the spacing between verticals.)
- Initial value:** DWIN Technology (The initial value displayed by power-on, the 0x05 position of the CFG file can be set to 0x38 to open the initial value display.)

### 7.18.3 Example of Instructions

① Display "North 12 AB"

5AA5 0B 82 6050 B1B1 3132 4142 FFFF

0 x5AA5: Frame header;

0x0B: Data length;

0x82: Write instruction;

0x6050: VP;

0 xB1B1: ASCII code of the word "North";

0x3132: ASCII code of "12";

0x4142: ASCII code of "AB". ASCII code can be converted by ASCII conversion tool, which can be obtained from customer service.

0xFFFF: Terminator. Adding 0xFFFF to the end of the character data will not display the following characters.

### 7.18.4 Effect

Zoom ratio 100%, 200%, 300%, 400% effect.



### 7.18.4 Gray Word Library

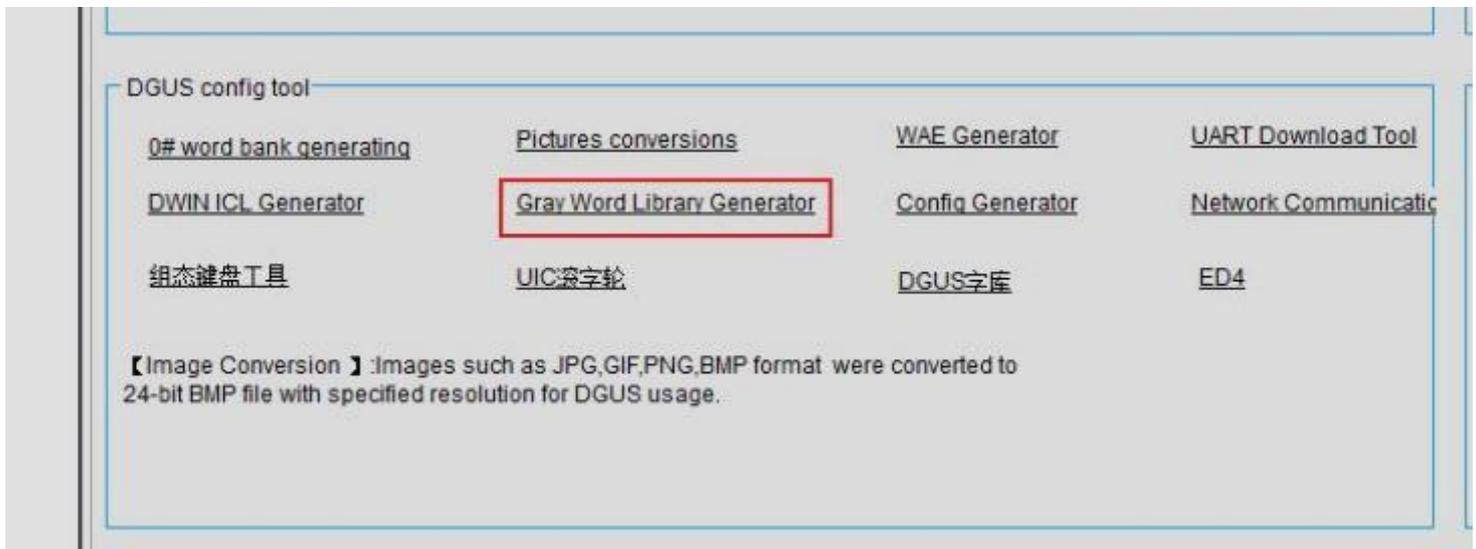
You can use DGUS to generate text display dedicated gray word library.

Features:

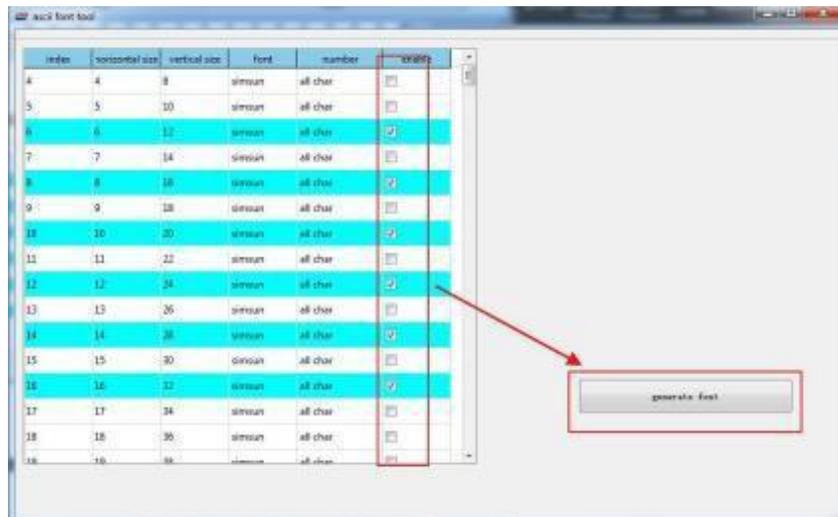
- (1) It can be used to remove the jaggies from the edges of characters, and the edge is smoothly displayed.
- (2) T5L V45 and above kernel products support this display control.
- (3) 20 \* 20 dot matrix font size is 5. 85 MB. You should choose the appropriate dot matrix size accordingly, and arrange reasonably the storage space in advance.
- (4) When using a large font size such as 24\*24 dot matrix (12 MB) or more, you need choose/customize products with expanded Flash(>16MB).

If users want to generate a gray word library, please follow the steps below.

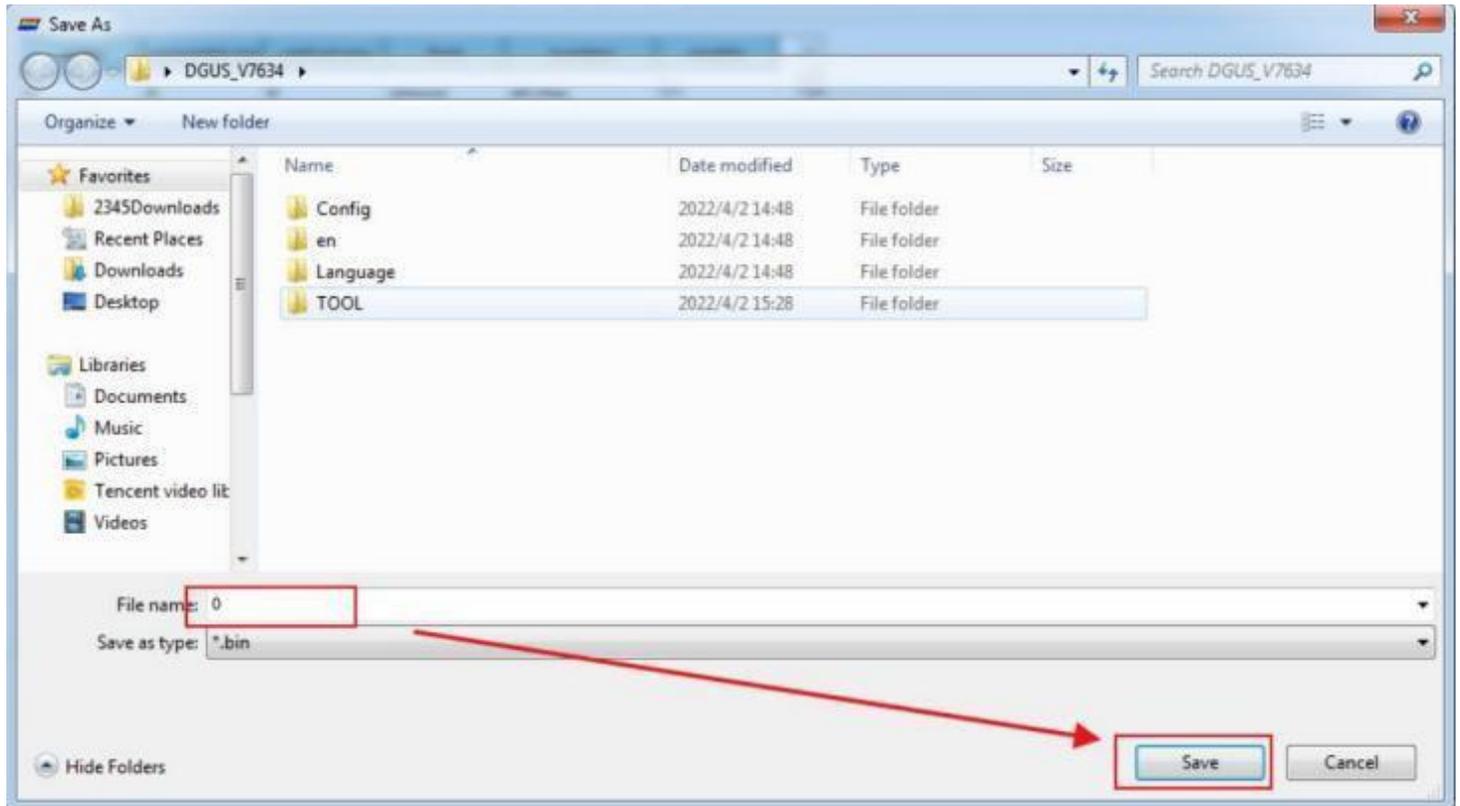
- (1) Open DGUS and click Gray Word Library Generator.



- (2) Select font size as you want and click generate.



(3) Select the saving path, name it accordingly and click save.





## 7.19 Roll Character

The characters are displayed in the form of a scroll, and the data can be changed in combination with sliding adjustment.

### 7.19.1 Instruction Storage Format

| Address | SP Offset | Definition   | Data length (byte) | Description  |
|---------|-----------|--------------|--------------------|--|
| 0x00    |           | 0x5A17       | 2                  | Variable pointer, the variable is a double word, the low word is reserved, and the high word is integer data -32768 to +32767.   |
| 0x02    |           | SP           | 2                  | Lower limited.   |
| 0x04    |           | 0x000D       | 2                  | Upper limited.   |
| 0x06    | 0x00      | VP           | 2                  | Data pointer. Each data occupies four words of storage space, defined as follows: VP= storage address of the selected data, VP+ 1 = adjustment parameter, corresponding to gesture adjustment VP+ 1 or incremental adjustment VP. VP+ 2 = System reserved, display offset, integer   |
| 0x08    | 0x01:H    | Adj_Mod      | 1                  | The high 4 bit is the data type:<br>0x0=Integer (2 bytes), -32768 to 32767<br>0 x1=* VP high byte, unsigned number 0-255<br>0 x2=* VP low byte, unsigned number 0-255<br>0xE=* VP The data directly corresponds to the font ID 0 - 255 , which is suitable for small screen input keyboard.<br>0xF=* VP data is an ASCII string pointer, and each line can contain up to 256 characters. The low 4 bit is (number of character lines-1)/2, 0x0-0x4, up to 9 lines. |
| 0x09    | 0x01:L    | Data_Mod     | 1                  | Data mode:<br>The upper 4bit is the number of integers, 0x00-0 x05. The lower 4bit is the number of decimal places, 0x00-0x05.<br>String mode: character pointer interval ( word length), 0 x01 - 0xFF.  |
| 0x0A    | 0x02      | VP_String    | 2                  | Data mode: interval step length of data variable, 0 x00 0 1 - 0 x7FFF.<br>String pattern:<br>The string variable storage pointer corresponding to the starting value (0x00), 0xFF represents the end of data.  |
| 0x0C    | 0x03      | V_Min        | 2                  | The lower limit of the data, fixed- point integer.   |
| 0x0E    | 0x04      | V_Max        | 2                  | The lower limit of the data, fixed- point integer.   |
| 0x10    | 0x05:H    | Display_Mode | 1                  | Display mode:<br>.7 1= invalid 0 displayed; 0= invalid 0 not displayed.<br>. 6-4 1=The selected line displays the Font0; 0x0-0x7.<br>.3 1=The positive "+" is displayed; 0=The positive "+"is not displayed.<br>.2 1= Character background is not filtered out; 0=Character background is filtered out.<br>.1-0 Alignment mode: 00=center, 01=left alignment, 02=right alignment.  |
| 0x11    | 0x05:L    | Speed_Set    | 1                  | .7 -.6 Data change speed, 0x00-0x03, 0x00 is the slowest.<br>. 5-.0 Scroll speed (pixels scrolled in each DGUS cycle), 0x01-0x3F.<br>The larger the value, the faster the scrolling. The recommended value is 1/ 16 of the 0 line spacing.   |
| 0x12    | 0x06      | Font_ID      | 2                  | Select the configuration icon font number, 0x0000 - 0xFFFF.  |
| 0x14    | 0x07      | (X, Y)       | 4                  | Show the display coordinates of the selected row. Center mode: the center coordinates of the selected line; left alignment mode: the left midpoint coordinates of the first character of the selected line; right alignment mode: the right midpoint coordinates of the last character of the selected line.   |



|      |        |               |   |   |
|------|--------|---------------|---|---|
| 0x18 | 0x09:H | Line_Height0  | 1 | Line spacing 0 ( the spacing between the selected line and the top 1 line; the spacing between the selected line and the bottom 1 line is also this value, treated symmetrically; the same below.) Height ( pixels in the Y direction). |
| 0x19 | 0x09:L | Line_Height1  | 1 | Line space 1 ( the space between the top 1 line and the top 2 lines) height ( pixels in the Y direction).   |
| 0x1A | 0x0A:H | Line_Height2  | 1 | Line space 1 (the space between the top 2 lines and the top 3 lines) height ( pixels in the Y direction).   |
| 0x1B | 0x0A:L | Line_Height3  | 1 | Line space 1 (the space between the top 3 lines and the top 4 lines) height ( pixels in the Y direction).   |
| 0x1C | 0x0B:H | DIM_No_Select | 1 | The window brightness is not selected, 0x00-0xFF. 0x00 is the darkest, 0xFF is the brightest; combined with the background.   |
| 0x1D | 0x0B:L | Height_Sel    | 1 | The height of the display area of the selected line must be higher than the character height of Font 0.   |
| 0x1E | 0x0C:H | Font1:2       | 1 | Higher 4 bit, the font of 1 line above, 0x00-0x07;<br>Lower 4bit, the font of 2 lines above, 0x00-0x07;   |
| 0x1F | 0x0C:L | Font3:4       | 1 | Higher 4bit , the font of 3 lines above, 0x00-0x07;<br>Lower 4bit, the font of 4 lines above, 0x00-0x07;  |

### 7.19.2 Software Setting

**Roller Character**

X 411 Y 194  
W 133 H 358

Name definition Roller character

Description pointer(0x) FFFF

Variable address(0x) 5840

type of data Integer (2 bytes)

Number of character lines 7

Integer digits 5

Decimal places 0

Word length 0

String address (0x) 0000

Data lower limit 0

Data upper limit 100

Display mode

Invalid display0

Display positive + sign

Character background is not filtered

Alignment mode Centered

Scroll speed 5

Icon library number 140

Selected window height 80

Unchecked window brightness 80

Select the first row up

Spacing from the previous line 58

Line font number 1

Select the second row from the top

Spacing from the previous line 48

Line font number 2

Select the third row up

Spacing from the previous line 40

Line font number 3

Row middle row up fourth row

Spacing from the previous line 0

Line font number 0

Coordinates of the upper left corner of the display area

Display area width and height

The variable storage space user can use the address range arbitrarily: 0x1000-0xFFFF. The scroll wheel display of the configuration icon font library occupies 4 addresses. That is, the non-associated control key addresses need to be spaced apart. System variable interface address range: 0x0000-0x0FFF (such as page switching address 0x0084)

When you need the data to show the scrolling effect, you can choose integer data;

The number of lines of character display, for example, the intuitive effect of setting 7 lines is shown on the right.

For example, the integer digit is set to 3 digits, and the decimal digit is set to 2 digits. The intuitive effect of displaying data 512 5.12

Data upper and lower limit range

The effect of displaying invalid 0: 01; the effect of unchecking: 1

Character background does not filter effect:  Character background filter effect:

Alignment mode: center, left, right, center alignment is recommended

Scrolling speed (pixels rolled in each DGUS cycle), 0x01-0xFF. The larger the value, the faster the scrolling. The recommended value is /16 of the line spacing 0. The height of the selected window is 80/16=5 (d setting value on the left)

Line spacing 0123456789-  
Line spacing 0123456789-  
Line spacing 0123456789-  
Line spacing 0123456789-

Configuration icon. UIC file naming number. Naming ID = number/8 (ID does not conflict with other words). For example, 140/8=17.5 (the occupied position does not conflict with fonts such as 13, 14), 256/8=32KB, a single IC file block occupies the 32KB interval.

The height of the display area of the selected line must be higher than the character height of Font0.

The selected font is 0#, the height on the software is 46, and the 80 display will be looser, and it can be set according to actual needs.

Shows the brightness of the selected window, 0x00-0xFF. 0x00 is the darkest, 0xFF is the brightest; combined with the background.

The scope of the unselected window area and the intuitive effect are shown in the right figure. The function is to change the font brightness of the selected area and highlight the selected font.

Here, the font numbers 0, 1, 2, and 3 correspond to the numbers defined in the font settings of the configuration icon font library software welcome interface.

The distance between the previous line and the previous line is set according to the height of the font.

For example, the line font number is 1, the software is 30 occupies the array height, the setting 58 is to make the interval look a little looser, setting 34/38/42 can be displayed.

| Font settings                           |    |
|---|----|
| 1#Font/Font height                      | 44 |
| 2#Font/Font height                      | 38 |
| 3#Font/Font height                      | 32 |
| 4#Font/Font height                      | 22 |
| 5#Font/Font height                      | 18 |
| 6#Font/Font height                      | 14 |
| 7#Font/Font height                      | 12 |
| 8#Font/Font height                      | 10 |
| 9#Font/Font height                      | 8  |
| 10#Font/Font height                     | 6  |
| 11#Font/Font height                     | 4  |
| 12#Font/Font height                     | 3  |
| 13#Font/Font height                     | 2  |
| 14#Font/Font height                     | 1  |
| Background setting                      | 63 |
| Character background filtering strength | 63 |

Step 1: Create a new 477\*124 pixel canvas, arrange the numbers in order, and set the canvas size according to actual needs.



Step 2: Open the configuration icon font library software - click the new project - the resolution is modified to be the same as the canvas -----Select the saved file path-----OK. -----Add canvas picture-----click the character icon definition, select the range of the intercepted icon, the height of each row should be the same, the width can be inconsistent, the same row, you can use the array----- After configuration, click Save, name it such as 140, and put it in the DWIN-SET folder to call it.

| Property      | Value |
|---------------|-------|
| 图标类型信息        | 字符图标  |
| 字符图标尺寸        |       |
| 高度            | 32    |
| 宽             | 48    |
| 坐标 (X, Y)     |       |
| X坐标           | 3     |
| Y坐标           | 3     |
| 字符图标对应ASCII字符 |       |
| ID            | 0030  |
| 键值            | 90    |
| 字符            | 0     |

The key value definition is the same as the pop-up keyboard for data variable entry.

Must be set, range 0-63, can be set to 63

### 7.19.3 Example of Instructions

- 5AA5 04 82 5840 0001
- 0x5AA5: Frame header;
- 0x04: Data length;
- 0x82: Write instruction;
- 0x5840: VP;
- 0x0001: Data value 1.

### 7.19.4 Effect





## 7.20 GTF Icon Font

The GTF icon control supports efficient display of icon characters.

### 7.20.1 Instruction Storage Format

| Address | SP Offset | Definition           | Length (byte) | Description  |
|---------|-----------|----------------------|---------------|--|
| 0x00    |           | 0x5A18               | 2             |  |
| 0x02    |           | SP                   | 2             | Variable description pointer, 0xFFFF means loaded by the configuration file.   |
| 0x04    |           | 0x000B               | 2             |  |
| 0x06    | 0x00:H    | *VP                  | 1             | Text pointer, must be an even address.<br>The maximum size of text variable data is 255 bytes.<br>0xFFFF, 0x0000 data or display to the end of the text box will no longer be displayed.   |
| 0x08    | 0x00:L    | GTF_ID               | 1             | Use GTF font encoding, 0x0000-0xFFFF.  |
| 0x0A    | 0x01      | Font_ID              | 8             | Use the font code ID in the GTF font library, 0x0000-0x03FB.   |
| 0x0C    | 0x05      | (X, Y)               | 2             | Start display position.<br>Left alignment mode: the coordinates of the upper left corner displayed on the first line;<br>Right-aligned mode: the coordinates of the upper right corner displayed in the first line;<br>Centering mode: undefined, any value can be used.   |
| 0x10    | 0x06      | (Xs, Ys)<br>(Xe ,Ye) | 8             | The coordinates of the upper left and lower right corners of the text box.   |
| 0x18    | 0x07      | HOR_Dis              | 1             | Icons are spaced horizontally.   |
| 0x19    | 0x08      | VER_Dis              | 1             | Icons are vertically spaced.   |
| 0x1A    | 0x0A:H    | Display_Mode         | 1             | .7 Background transparency options:<br>0 is transparent, the background is not displayed;<br>1 Opaque, background displayed.<br>.6 font reload.<br>0 Reload font library (the first GTF icon text display of the page must be loaded);<br>1 The previous GTF icon text shows that the font library has been loaded, and the speed is improved without repeated loading.<br>.5 Icon Overlay Selection<br>0 is not superimposed, and the coordinate position automatically moves backward after an icon is displayed;<br>1 Icons are displayed superimposed.<br>.4-.2 Reserved, write 0.<br>.1-.0 alignment mode:<br>0=left-aligned 1=centered 2=right-aligned |
| 0x1B    | 0x0A:L    | Undefined            | 1             | Write 0x00   |

### 7.20.2 Software Setting

|   |  |
|---|--|
| <p>X <input type="text" value="264"/> Y <input type="text" value="373"/></p> <p>W <input type="text" value="502"/> H <input type="text" value="123"/></p> <p>Name definition <input type="text" value="GFT Display"/></p> <p>Description pointer <input type="text" value="FFFF"/></p> <p>Variable address <input type="text" value="1000"/></p> <hr/> <p>GTF_ID <input type="text" value="63"/></p> <p>FONT_ID <input type="text" value="0"/></p> <p>Icons are spaced horizontally <input type="text" value="0"/></p> <p>Icons are vertically spaced <input type="text" value="0"/></p> <hr/> <p>Background setting <input type="text" value="Transparent, ba"/></p> <p>Font Overload Settings <input type="text" value="Reload font"/></p> <p>Icon Loading Selection <input type="text" value="Overlay"/></p> <p>Alignment mode <input type="text" value="Align left"/></p> <hr/> <p>Initial Value <input type="text"/></p> | <p>The coordinates of the upper left corner of the display area</p> <p>Display area width and height</p> <p>The variable storage space user can use the address range arbitrarily: 0x1000-0xFFFF. The GTF icon occupies the address according to the displayed character length, and the maximum display length is 128 characters. System variable interface address range: 0x1000-0xFFFF. (eg page switch address 0x0084).</p> <p>The GTF icon name ID is 0-63, and a single GTF icon occupies a small Flash space with a space of 256KB.</p> <p>Can be set by default.</p> <p>The larger the horizontal spacing value, the wider the icon spacing: <b>迪 文</b></p> <p>The larger the vertical spacing value, the wider the icon spacing: <b>迪</b><br/><b>文</b></p> <p>The display background color is usually filtered out by default. It is recommended that the background color of the icon and the color of the basemap area be the same.</p> <p>The text of the previous GTF icon shows that the font library has been loaded, and the speed is improved without repeated loading. Select the settings according to your needs, and the default settings are available.</p> <p>Display the overlay effect of characters 1 and 2: <b>2</b> Show characters 1 and 2 do not stack: 1 2</p> <p>The characters are left-aligned, right-aligned, or centered, which can be set by default.</p> |
|---|--|

### 7.20.3 GTF TOOL Software Setup Instruction

(1) Software tool: GTF Tool

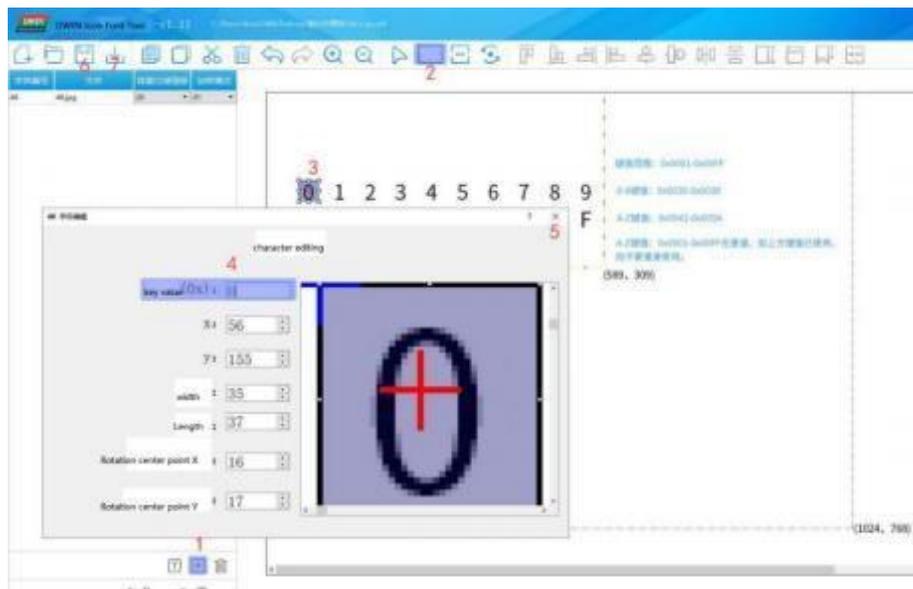


(2) Setting

- ① Click the "+" in the lower left corner to add a picture;
  - ② Use the "rectangle" control to define key values;
  - ③ Box-select the character icon area;
  - ④ Double-click the control to define the key value;
- Key value range 0x01-0xFF;  
0-9 key value: 0x30-0x39;  
A-Z key value: 0x41-0x5A;  
Other key values can be used arbitrarily as needed.
- ⑤ Click "x" to save and close;
  - ⑥ Click " Save" to generate the configuration file;
  - ⑦ Click " Generate" to save the GTF icon library file.

(3) Precautions

- It is recommended that the GTF icon should not exceed 1920\*1080, and the icon will not be displayed if it is too large;
- If the original image of GTF does not display the bmp and jpg images output by the bitmap software, you can use the bmp and jpg images of the vector graphics software;
- The FTG icon name ID is 0-63, and a single GTF icon occupies the Flash space with a subspace size of 256 KB.



#### 7.20.4 Example of Instructions

The max text variable data is 255 bytes, and the variable address is reserved as needed.

① Display character 0

Tx: 5A A5 04 82 1000 30

② Display character 12

Tx: 5A A5 05 82 1000 3132

③ Display character 0123456789

Tx: 5A A5 0D 82 1000 3031 3233 3435 3637 3839

④ Display character A

Tx: 5A A5 04 82 1000 41

⑤ Display characters ABCD, remove redundant characters and add terminator FFFF

Tx: 5A A5 09 82 1000 4142 4344 FFFF

⑥ "China" on the display icon

Tx: 5A A5 07 82 1000 0102 FFFF

⑦ Display icon

Tx: 5A A5 05 82 1000 0304

⑧ "Weather" character on the display icon

Tx: 5A A5 07 82 1000 0506 FFFF

⑨ Clear all characters

Tx: 5A A5 05 82 1000 FFFF

### 7.20.5 Effect





## 7.21 Real-Time Curves

The real-time curves control is to automatically match and display the real-time curve (trend graph) by writing the curve buffer data with the 0 x84 instruction.

The display area, the coordinates of the central axis and the display scale (enlargement/reduction) can be set by users.

### 7.21.1 Instruction Storage Format

| Address | SP offset | Definition    | Length (bytes) | Description   |
|---------|-----------|---------------|----------------|---|
| 0x00    |           | 0x5A20        | 2              |   |
| 0x02    |           | SP            | 2              | SP, 0xFFFF means loaded by configuration file.  |
| 0x04    |           | 0x000B        | 2              |   |
| 0x06    | 0x00:H    | Mode          | 1              | 0x00 = The latest data is on the far right, and the curve moves from right to left. Other = The latest data is on the far left, the curve moves from left to right.   |
| 0x07    | 0x00:L    | 0x00          | 1              | Display mode, bit definition.<br>.7 0=Curve is displayed with data point connection; 1=Curve is displayed with data point.<br>.6 Curve color selection:<br>0 = Curve has only one color, defined by Color.<br>1=The curve has 16 colors, Color is a variable pointer, which saves 16 colors (65K colors); at this time, the high 4bit of the curve data is the color index.<br>.5-.0 is undefined, write 0. |
| 0x08    | 0x01      | Xs, Ys Xe, Ye | 8              | The coordinates of the upper left corner of the curve window ( Xs, Ys) and the coordinates of the lower right corner ( Xe, Ye)<br>Curve out of bounds will not be displayed.  |
| 0x10    | 0x05      | Y_Central     | 2              | The position of the center axis of the curve  |
| 0x12    | 0x06      | VD_Central    | 2              | The curve data value corresponding to the central axis generally takes half of the sum of the maximum and minimum values of the data.   |
| 0x14    | 0x07      | Color         | 2              | Curve color   |
| 0x16    | 0x08      | MUL_Y         | 2              | The magnification of the vertical axis, the unit is 1/256, 0x0000-0x7FFF.   |
| 0x18    | 0x09:H    | CHANEL        | 1              | Data source channel, 0x00-0x07  |
| 0x19    | 0x09:L    | Dis_HOR       | 1              | Horizontal axis interval, 0x01 -0xFF.   |
| 0x1A    | 0x0A:H    | Pixel_Scale   | 1              | Curve pixel lattice size (curve line width), 0x00-0x07 corresponds to 1*1 to 8*8.   |
| 0x1B    | 0x0A:L    | Reserved      | 1              | Write 0x00  |

### 7.21.2 Software Setting

Open DGUS, click display control – real- time curves, then set the control display area, and configure the curve color, data source channel, etc. in the setting menu on the right.

If you need to display a thicker curve line, you can place multiple up and down (Y- axis) translation curve variables in the adjacent position and refer to the same data source to achieve this. That is, press Ctrl+ C, Ctrl+ V to copy a set of curve controls.

Click the copied curve box and press up and down keys of the keyboard The more control you copy, the thicker the curve line.

[ Note] The configuration and usage examples with touch controls are as follows .

If the content of vp is stored in sp, combined with incremental adjustment control, adjusting the + 0 x0 9 low byte access address, you can realize automatic scaling of the curve without code .

Also, if combined with drag adjustment control, adjusting + 0x0 5 to modify the Y\_ Central value, the curve can be moved up and down without code.

The screenshot shows the 'Curve display' configuration window. On the left, a diagram illustrates a curve on a screen with a bounding box. The top-left corner of the box is labeled (Xs, Ys) (558, 194) and the top-right corner is (Xe, Ye) (674, 277). The settings on the right are as follows:

- X: 558, Y: 194
- W: 115, H: 78
- Name definition: Curve Display
- Description pointer(0x): FFFF
- Y\_Central: 233
- VD\_Central: 500
- Curve color(0x): FFE0
- Magnification number of vertical axis: 19
- Unit: 1/256, value range: 0x0000-0x7FFF
- Data source channel: 3 (0x00-0x07)
- Horizontal axis interval: 5 (0x01-0xFF)
- Curve line width: 0 (0x00-0x07)

Callout boxes provide the following details:

- Y-axis height:** The height of the curve frame, when calculating the magnification digits of the vertical axis, you can directly call this value, which is the (Ye-Ys) value.
- Y-axis center:** The position of the center point of the Y-axis of the curve display frame. Click the display box with the mouse and place it at the middle red dot coordinates to view.
- VD\_Central:** The curve data value corresponding to the central axis is generally taken as half of the sum of the maximum value and the minimum value of the data. Calculation formula:  $VD\_central = (Vmax + Vmin) / 2$ . Example:  $(1000 + 0) / 2 = 500$ . Vmax: The most current value of the data; the minimum value of the Vmin data.
- Magnification:** Calculation formula:  $MUL\_Y = (Ye - Ys) * 256 / (Vmax - Vmin)$ . Ye: Maximum value of curve data, Vmin: The minimum value of the curve data. For example:  $78 * 256 / (1000 - 0) = 19.968$  (78 to directly view the height value of the display box) here, round down to 19.
- Data source channel:** 0x00-0x07 total 8 channels; such as calling 0x00 channel write 0, To call channel 0x03, write 03, call all 8 channels, write 0, 1, 2, 3, 4, 5, 6, 7 for each channel.
- Horizontal axis interval:** The pixel pitch of the first Y-axis data and the second Y-axis data, The larger the interval, the wider the span, the smaller the interval, the narrower the span. Range: 0x00-0xFF.
- Curve line width:** The line width of the curve is bold, 0 is the width of one pixel point, 1 is a point thickened on the upper and lower sides of the curve, and 7 is the thickest.
- Data range:** The routine curve display data range: 0-10000 and 1000 indicate that the curve displays the Y axis value, not a coordinate point. For example, 1000 is just a data value.
- Data direction:** From left to right: the curve data is displayed from the machine to the right. From right to left: Start from left to right, and after the curve box is filled, the data will be displayed from right to left. Example code: `5A A5 0D 82 0310 5AA5 0100 0102 0000 03E8` (Use this instruction to fill the curve frame) and `5A A5 0D 82 0310 5AA5 0100 0102 0000 01F4` (write this one again, it will be from right to left).



### 7.21.3 Example of Instructions

| VP                        | Explanation   |
|---------------------------|---|
| <p><b>0x300-0x30F</b></p> | <p>Status feedback of 8 channel curve buffers (Read-only for users).<br/>Each channel occupies 2 words.</p> <p>The storage pointer address of the high byte storage curve data (0 x0000 - 0 x07 FF),<br/>The storage pointer address of the low byte storage curve data (0x0000-0x0800).<br/>Write 0x0000 to the effective Data length of the curve buffer will cause the curve not to be displayed.</p> <p>Example:<br/>5 A A5 05 82 0301 0000 The curve 0 channel will be cleared, and it will be restored if the value is rewritten;<br/>5 A A5 05 82 03 0 3 00 0 0 Curve 1 channel will be cleared, and it will be restored if the value is rewritten;<br/>5 A A5 0 5 8 2 03 0 5 0 0 0 0 Curve 2 channel will be cleared, and it will be restored if the value is rewritten;<br/>5 A A5 0 5 8 2 03 0 7 0 0 0 0 Curve 3 channel will be cleared, and it will be restored if the value is rewritten;<br/>5 A A5 0 5 8 2 03 0 9 0 0 0 0 Curve 4 channel will be cleared, and it will be restored if the value is rewritten;<br/>5 A A5 05 82 030B 0000 Curve 5 channel will be cleared, and it will be restored if the value is rewritten;<br/>5 A A5 0 5 8 2 0 3 0 D 0 0 0 0 Curve 6 channel will be cleared, and it will be restored if the value is rewritten;<br/>5 A A5 05 82 030F 0000 Curve 7 channel will be cleared, and it will be restored if the value is rewritten;</p> |
| <p><b>0x310-0x311</b></p> | <p>Write start to curve buffer.<br/>D3:D2: 0x5AA5 Start a curve buffer data write operation, and the CPU will clear it after the operation.<br/>D1: The number of data blocks, 0x01-0x08.<br/>D0: undefined, write 0x00.</p>  |
| <p><b>0x312-0x37F</b></p> | <p>The data block that needs to be written into the curve buffer, the data is a 16 bit unsigned number.<br/>A single block of data is defined as a data channel ID (0x00-0x07) + data word length (0x01- 0x6 E) + data.</p>   |



## ① 0x00 channel display data 0 and 1000

```
5A A5 0D 82 0310 5AA5 0100 0002 0000 03E8
```

0x5AA5: Frame header;

0x0D: Data length;

0x82: Write instruction;

0 x0310 : The curve buffer data writing is started; it is the system variable hardware interface curve fixed address;

0 x5 AA5: Start a curve buffer data writing operation, and clear it after the CPU operation;

0x0100: The number of high byte 0x01 curve data blocks, that is, the number of channels occupied; write 0x00 in case of low byte undefined;

0x0002: High byte 0x00 represents channel 0, low byte 0x02 represents the number of data word lengths. In this example, it is a two-word data, namely 0x0000, 0x03E8;

0 x0000: The data to be displayed 0;

0x03E8: The data to be displayed 1000.

## ②0x06 channel display data 0 and 1000

```
5A A5 0D 82 0310 5AA5 0100 0602 03E8 0000
```

## ③0x07 channel display data 0 and 1000:

```
5A A5 0D 82 0310 5AA5 0100 0702 0000 03E8
```

## ④ 0x00 and 0x07 channel display data 0 and 1000

```
5A A5 13 82 0310 5AA5 0200 0002 0000 03E8 0702 0000 03E8
```

0x5AA5: Frame header;

0x0D: Data length;

0x82: Write instruction;

0 x0310 : The curve buffer data writing is started; it is the system variable hardware interface curve fixed address;

0 x5 AA5: Start a curve buffer data writing operation, and clear it after the CPU operation;

0x0200: The number of high byte 0x02 curve data blocks, that is, the number of channels occupied; If an instruction writes 8 channels, this byte needs to be changed to 08; write 0x00 in case of low byte undefined;

0x0002: High byte 0x00 represents channel 0, low byte 0x02 represents the number of data word lengths. In this example, it is a two-word data, namely 0x0000, 0x03E8;

0 x0000: The data to be displayed 0;

0x03E8: The data to be displayed 1000;

0x0702: High byte 0x07 represents channel 7, low byte 0x02 represents the number of data word lengths. In this example, it is a two-word data, namely 0x0000, 0x03E8;

0 x0000: The data to be displayed 0;

0x03E8: The data to be displayed 1000

The channel and data word lengths of the two curves are written separately.



⑤ Channels 0x06 and 0x07 display two cross curves and display from right to left

Instruction 1: 5A A5 13 82 0310 5AA5 0200 0602 0000 03E8 0702 03E8 0000

Instruction 1 is to form a cross curve, where the maximum and minimum values of the two curves are set opposite, so that the middle of the curves can be crossed.

The cross curve is actually two different channel curves, so you need to make two curve controls and set different channels.

Instruction 2: 5A A5 13 82 0310 5AA5 0200 0602 0064 01F4 0702 01F4 0064

On the right-to-left setting screen of the software, it is displayed from left to right at first. When the curve box is filled, write instruction 2, and then you can see that the curve is displayed from right to left.

⑥ 0x00-0x078 channels are all displayed

5A A5 37 82 0310 5AA5 0800 0002 0000 03E8 0102 0000 03E8 0202 0000 03E8 0302 0000 03E8 0402 0000 03E8 0502 0000 03E8 0602 0000 03E8 0702 0000 03E8

0x00 - 0x07 Channels Each channel can be refreshed separately in turn to achieve the effect of refreshing different channel curves at the same time.

T5L DGUS curve 8-channel display, a total of 8 channels, each channel can store 2048 words .

After enabling dynamic curve display, starting from 0 x 1000 , a data buffer is established for each curve according to 2Kwords per channel.

CH0 buffer is 0x1000-0x17FF, CH1 buffer is 0x1800-0x1FFF,

CH2 buffer is 0x2000-0x27FF, CH3 buffer is 0x2800-0x2FFF,

CH4 buffer is 0x3000-0x37FF, CH5 buffer is 0x3800-0x3FFF,

CH6 buffer is 0x4000-0x47FF, CH7 buffer is 0x4800-0x4FFF,

Unused curve buffers can be used as user vp. User can also directly rewrite the curve buffer,

For example, directly rewrite the variable address 5A A5 05 82 1000 0065...

The value of the corresponding channel of the curve will change accordingly.

If the user needs to use 8 channels, the variable addresses and description pointers of other display controls are used from the range of 0x5000~0xFFFF.

⑦ Real- time curves SP usage and example

The SP storage address format of this function and some examples of commonly used access instructions are as below.

| SP     | Instruction ( SP of the example is set to 0x8000) | Instruction Effect and Application   |
|--------|---|--|
| 0x05   | 5A A5 05 82 8005 0064                             | Change the position of the center axis of the curve (the curve moves up and down): move it up and down to the y axis 100 position.   |
| 0x06   | 5A A5 05 82 8006 0064                             | Change the range of the curve (vertical scaling of the curve): the curve data value corresponding to the central axis, generally half of the sum of the maximum and minimum data. For example, here the instruction is changed to 0 x2 0 0 range, ( corresponding magnification needs to be modified in the 0 x08 pointer address) |
| 0x07   | 5A A5 05 8 08007 F800                             | The color of the curve display changes to red  |
| 0x08   | 5A A5 05 82 8008 00F0                             | Modify the magnification of the vertical axis, the unit is 1/256, 0x0000-0x7FFF. (Need to be used with 0x06 pointer)   |
| 0x09:H |   | Data source channel, 0x00-0x07   |
| 0x09:L | 5A A5005 82 8007 0014                             | Change the horizontal axis interval (curve horizontal scaling): display the horizontal axis interval of the 00 channel curve to 0x14 (range 0x01-0xFF)   |

⑧ Negative number display

Curve display -300-300 settings:  
 When displaying, it is necessary to add 300 to the data as a whole, so that the VD and vertical axis magnification can be calculated, so the calculated data is 0-600  
 If you want to display 300 now, instead of writing data 0x012C directly, you want to write data 600 (300+(absolute value of -300)), that is, 0x0258, then it will be displayed correctly.

VD calculation formula:  $VD\_central = (Vmax + Vmin) / 2$  Example:  $(600 + 0) / 2 = 300$   
 Vertical axis magnification formula:  $MUL\_Y = (Ye - Ys) * 256 / (Vmax - Vmin)$  Example:  $78 * 256 / (600 - 0) = 33.28$   
 Round down and take 33, 78, which is the H value above the software attribute (the height value of the attribute box).  
 Command example: 5A A5 0D 82 0310 5A A5 0100 0302 0000 0258



Real-time curve data point refresh, displayed in point form

Set the VP address 0x8000, + 0x0000 attribute address low byte. 7 bits write 1 to open the display data point, write 0 to connect the data point,

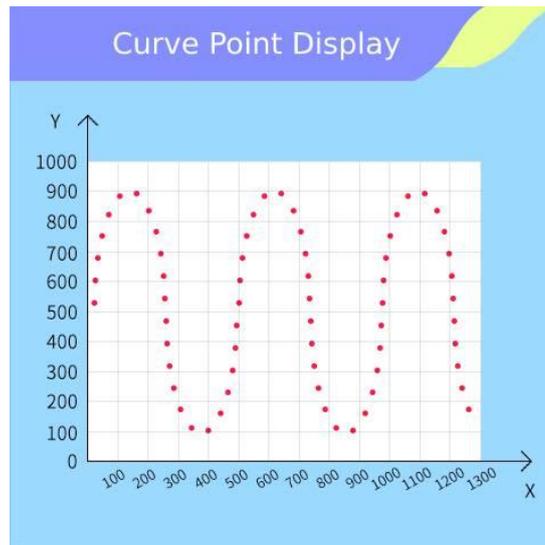
Convert binary 1000 0000 into hexadecimal 0x80, the command example is as follows:

Turn on the data in dot form: 5A A5 05 82 8000 00 80

Curve connection display : 5A A5 05 82 8000 00 00

Display data 1000: 5A A5 0B 82 0310 5AA5 0100 0001 03E8

Display data 500: 5A A5 0B 82 0310 5AA5 0100 0001 01F4



Example of 16 colors display command for the same curve

**3 colors:** After the DGUS software enables multiple color display functions and specifies the color address, write the color value (red: 0xF800, yellow: FFE0, blue: 0x001F) to the color variable address.

① Define the color position first:

5A A5 09 82 9000 F800 FFE0 001F

Meaning: 0x9000 is the starting address of the specified color value, up to 16 color values, 16 color values can be written together, here is an example to write 3 color values, red: 0xF800, yellow: FFE0, blue: 0x001F.

② Re-index call color:

5A A5 06 84 01 0010 03E8;

meaning:

0x5A A5: frame header;

0x06: data length;

0x84: write curve buffer data;

0x01: curve channel 0;

0x0010: curve data of the high 4bit is the colour index ID, 1 word = 2 bytes, 1byte = 8bit, such as: 1111 0000, the high 4bit is 1111, the low 4bit is 0000; high 4bits are written 1 is the binary 1111, converted into



hexadecimal is **F**, that is, there are a total of 16 kinds of combinations ID (0-15), corresponding to 16 kinds of colour can be set; when using a certain ID, the program will call its corresponding colour, this colour value is the value written to the 0x9000 address above; low 12bit represents the curve data, 0x9000 address, this colour value is the value written to the upper. There are 16 kinds of ID (0-15), which can be set to 16 kinds of colors; when using a certain ID, the program will call its corresponding colour, and this colour value is the value written to the 0x9000 address from above; **the low 12bit indicates the curve data, 0x010 is data 16, 0x3E8 is data 1000.**

**③ Different IDs of other data indexes call different colors**

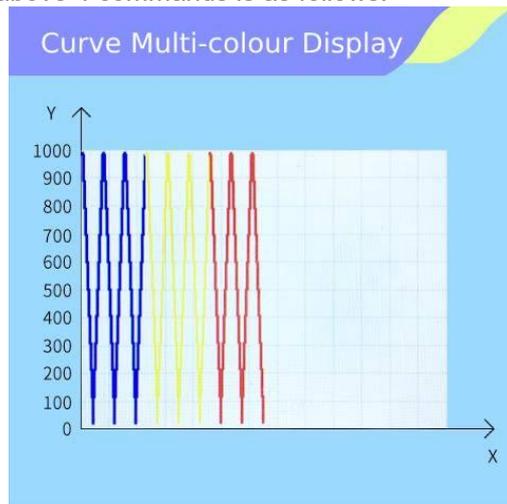
5A A5 06 84 01 **1010 13E8**;

The high 4bit of the curve data is the color index ID; **the low 12bit represents the curve data, 0x010 is data 16, 0x3E8 is data 1000**

5A A5 06 84 01 **2010 23E8**;

The high 4bit of the curve data is the color index ID; **the low 12bit represents the curve data, 0x010 is data 16, 0x3E8 is data 1000**

The actual display of the effect of the above 4 commands is as follows:



**16 colors:** After the DGUS software enables multiple color display functions and specifies the color address, write the color value to the color variable address

ID0: Red: 0xF800

ID1: Blue: 0x001F

ID2: Green: 0x07E0

ID3: Orange: 0xFC00

ID4: Purple: 0x801F

ID5: Turquoise: 0x07FF

ID 6: Brown: 0x4000

ID7: Pink: 0xFC1F

ID8: Dark green: 0x0208

ID9: Yellow-green: 0x8400

ID10: Rose red: 0xF810



ID11: Deep Purple: 0x4010

ID12: Sky blue: 0x041F

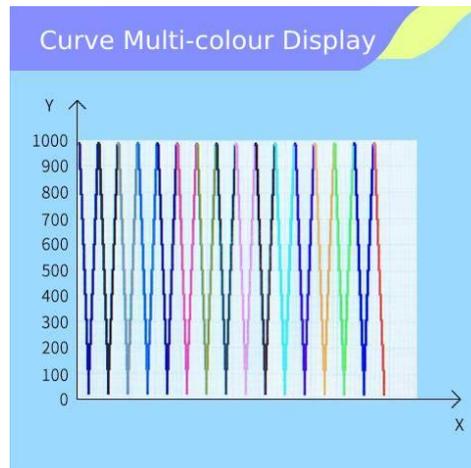
ID13: Neutral gray: 0x8410

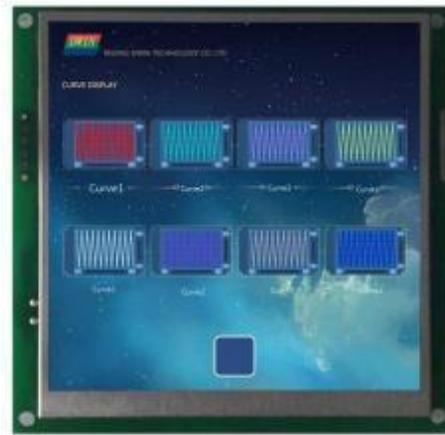
ID14: Black: 0x0000

ID15: Dark blue: 0x0010

Define color position: 5A A5 23 82 9000 F800 001F 07E0 FC00 801F 07FF 4000 FC1F 0208 8400 F810 4010 041F 8410 0000 0010

- Index ID0 data display red: 5A A5 06 84 01 0010 03E8
- Index ID1 data display blue: 5A A5 06 84 01 1010 13E8
- Index ID2 data display green: 5A A5 06 84 01 2010 23E8
- Index ID3 data is displayed in orange: 5A A5 06 84 01 3010 33E8
- Index ID4 data display purple: 5A A5 06 84 01 4010 43E8
- Index ID5 data showing turquoise: 5A A5 06 84 01 5010 53E8
- Index ID6 data display brown: 5A A5 06 84 01 6010 63E8
- Index ID7 data shows pink: 5A A5 06 84 01 7010 73E8
- Index ID8 data shows dark green: 5A A5 06 84 01 8010 83E8
- Index ID9 data display yellow-green: 5A A5 06 84 01 9010 93E8
- Index ID10 data showing rose red: 5A A5 06 84 01 A010 A3E8
- Index ID11 data shows dark purple: 5A A5 06 84 01 B010 B3E8
- Index ID12 data display sky blue: 5A A5 06 84 01 C010 C3E8
- Index ID13 data display neutral gray: 5A A5 06 84 01 D010 D3E8
- Index ID14 data display black: 5A A5 06 84 01 E010 E3E8
- Index ID15 data display dark blue: 5A A5 06 84 01 F010 F3E8







## 7.22 Basic Graphics

The basic graphics control is to define a "drawing board" function in the display configuration file 1 4 . BIN, and the specific drawing operation is determined by the content of the variable memory pointed to by \*VP.

Users can realize different drawing functions by changing the data in the variable memory.

### 7.22.1 Instruction Storage Format

| Address | SP Offset | Definition     | Length (byte) | Description   |
|---------|-----------|----------------|---------------|---|
| 0x00    |           | 0x5A21         |               |   |
| 0x02    |           | *SP            |               |   |
| 0x04    |           | 0x0008         |               |   |
| 0x06    | 0x00      | *VP            | 2             | Variable data pointer   |
| 0x08    | 0x01      | Area           | 8             | The coordinates of the upper left corner and the lower right corner of the drawing display area; if the drawing exceeds the boundary, it will not be displayed. only for 0x0001-0x0005, 0x0009, 0x000A, 0x000 Binstructions are valid.  |
| 0x10    | 0x05:H    | Dashed_Line_En | 1             | 0x5 A: Drawing instructions using line segments ( 0x02 , 0x03 , 0x09 , 0x0A instructions) will use dotted lines or dotted lines to display line segments; Miscellaneous: Drawing instructions that use line segments display line segments with solid lines.  |
| 0x11    | 0x05:L    | Dash_Set       | 4             | The 4 bytes are sequentially formatted with dotted lines (dot-dash): The number of solid lines in the first segment, the number of dotted lines in the first segment, the number of solid lines in the second segment, and the number of dotted lines in the second segment. For example, setting 0x10 0x04 0x10 0x04 will display dashed lines; setting 0x10 0x04 0x02 0x04 will display dotted lines. |
| 0x15    | 0x07:L    | Pixel_Scale    |               |   |
| 0x16    | 0x07      | Undefined      | 13            | Reserved 0x00   |

The description of the variable data format pointed to by the variable data pointer (variable storage space) is shown in the table.

| Address | Definition        | Description  |
|---------|-------------------|--|
| VP      | CMD               | Drawing instruction  |
| VP+1    | Data_Pack_Num_Max | Max data packet: connection (0x0002) , defined as the number of connection lines (the number of vertices-1); |
| VP+2    | DATA_Pack         | Data   |



| CMD    | Function                           | Address  | Length | Definition   | Description   |
|--------|------------------------------------|----------|--------|--|---|
| 0x0001 | Dot                                | 0x00     | 2      | (x, y)   | The coordinate position of dot. The high byte of coordinate x is the judgment condition.  |
|        |                                    | 0x02     | 1      | Color  | Dot color   |
| 0x0002 | Line                               | 0x00     | 1      | Color  | Line color  |
|        |                                    | 0x01     | 2      | (x, y) <sub>0</sub>                                | The coordinate of the line vertex <sub>0</sub> , and the high byte of the x coordinate are the judgment conditions.   |
|        |                                    | 0x03     | 2      | (x, y) <sub>1</sub>                                | The coordinate of the line vertex <sub>1</sub> , and the high byte of the x coordinate are the judgment conditions.   |
|        |                                    | 0x01+2*n | 2      | (x, y) <sub>n</sub>                                | The coordinate of the line vertex <sub>n</sub> , and the high byte of the x coordinate are the judgment conditions.   |
| 0x0003 | Rectangular                        | 0x00     | 2      | (x, y) <sub>s</sub>                                | The coordinates of the upper left corner of the rectangle and the high byte of the x coordinate are the judgment conditions.  |
|        |                                    | 0x02     | 2      | (x, y) <sub>e</sub>                                | The coordinates of the bottom right corner of the rectangle.  |
|        |                                    | 0x04     | 1      | Color  | Rectangular color   |
| 0x0004 | Rectangle filling                  | 0x00     | 2      | (x, y) <sub>s</sub>                                | The coordinates of the upper left corner of the rectangle and the high byte of the x coordinate are the judgment conditions.  |
|        |                                    | 0x02     | 2      | (x, y) <sub>e</sub>                                | The coordinates of the bottom right corner of the rectangle.  |
|        |                                    | 0x04     | 1      | Color  | Fill color of the rectangle field.  |
| 0x0005 | Circle                             | 0x00     | 2      | (x, y) <sub>s</sub>                                | Center coordinates.   |
|        |                                    | 0x02     | 1      | R  | The radius of the circle.   |
|        |                                    | 0x03     | 1      | Color  | The arc color.  |
| 0x0006 | Picture Copy& Paste                | 0x00     | 1      | Pic_ID   | The ID of the page where the cut image area is located; the high byte is the judgment condition   |
|        |                                    | 0x01     | 2      | (x, y) <sub>s</sub>                                | The coordinates of the upper left corner of the image cutting area.   |
|        |                                    | 0x03     | 2      | (x, y) <sub>e</sub>                                | The coordinates of the lower right corner of the image cutting area.  |
|        |                                    | 0x05     | 2      | (x, y)   | The coordinates of the upper left corner of the area in the current page that it is cut and pasted to.  |
| 0x**07 | Icon Display                       | 0x00     | 2      | (x, y)   | The coordinate position of display area, and the high byte of the x coordinate is the judgment condition.   |
|        |                                    | 0x02     | 1      | ICON_ID  | Icon ID, the location of the icon library is specified by the high byte of instruction.<br>The icon is fixed to not display the background color.                             |
| 0x0008 | Closed Region Filling              | 0x00     | 2      | (x, y)   | The coordinates of point.   |
|        |                                    | 0x02     | 1      | Color  | The color of filling the rectangle field.   |
| 0x0009 | Frequency Spectrum (Vertical line) | 0x00     | 1      | Color <sub>0</sub>                                 | Connect (x <sub>0</sub> , Y <sub>0s</sub> ) (x <sub>0</sub> , Y <sub>0e</sub> ) with Color <sub>0</sub> color, and the high byte of x <sub>0</sub> is the judgment condition. |
|        |                                    | 0x01     | 3      | X <sub>0</sub> , Y <sub>0s</sub> , Y <sub>0e</sub> |   |



|        |                          |      |   |           |   |
|--------|--------------------------|------|---|-----------|---|
| 0x000A | Line segment display     | 0x00 | 1 | color     | Connect (Xs, Ys) (Xe, Ye) with Color, and the high byte of Xs is the judgment condition.  |
|        |                          | 0x00 | 1 | color     |   |
|        |                          | 0x00 | 1 | color     |   |
| 0x000D | xOR                      | 0x00 | 2 | (x, y)s   | The coordinates of the upper left corner of the rectangular area, and the high byte of the x coordinate are the judgment conditions.  |
|        |                          | 0x02 | 2 | (x, y)e   | The coordinates of the lower right corner of the rectangular field.   |
|        |                          | 0x04 | 1 | Color     | The color of xOR of the rectangular field, and 0xFFFF will be inverted.   |
| 0x000E | Two-color bitmap display | 0x00 | 2 | (x, y)s   | The bitmap displays the coordinates of the upper left corner of the rectangle, and the high byte of the x-coordinate is the judgment condition.   |
|        |                          | 0x02 | 1 | X_Dots    | The number of dots in the x direction of the bitmap   |
|        |                          | 0x03 | 1 | Y_Dots    | The number of dots in the y direction of the bitmap   |
|        |                          | 0x04 | 1 | Color1    | " 1 " bit corresponding display color   |
|        |                          | 0x05 | 1 | Color0    | The display color corresponding to "0 " bit; if you set Color0 and Color1 to be the same, Indicates that "0 " bit does not need to be displayed, just skip it.                                      |
|        |                          | 0x06 | N | Data_Pack | Display data, MSB mode; in order to facilitate users to read and write data, each line of data must be aligned to a word, that is, the next line of data always starts from a new data word (Word). |
| 0x0011 | Ellipse display          | 0x00 | 2 | (x, y)    | The coordinates of the center of the ellipse, and the high byte of the x coordinate is the judgment condition.  |
|        |                          | 0x02 | 1 | A         | Long axis length.   |
|        |                          | 0x03 | 1 | B         | Short axis length.  |
|        |                          | 0x04 | 1 | color     | Elliptical arc color.   |
| 0x0012 | Four-color bitmap        | 0x00 | 2 | ( x, y) s | The upper left coordinates of the displayed bitmap, and the high byte of the x- coordinate is the judgment condition.<br><b>The starting VP address must be even ( double- word aligned) .</b>      |
|        |                          | 0x02 | 1 | X_Dots    | The number of bitmap X-direction pixels, which must be divisible by 16 .  |
|        |                          | 0x03 | 1 | Y_Dots    | The number of bitmap Y- direction pixels.   |
|        |                          | 0x04 | 1 | Color0    | The display color corresponding to the "00" bit.  |
|        |                          | 0x05 | 1 | Color1    | The display color corresponding to the "01" bit.  |
|        |                          | 0x06 | 1 | Color2    | The display color corresponding to the "10" bit.  |
|        |                          | 0x07 | 1 | Color3    | The display color corresponding to the "11" bit.  |
|        |                          | 0x08 | N | Data_Pack | Display data in MSB.<br>Each line of data must be aligned to a double word ( 16 pixels) for users to read and write the data.   |

The value 0xFF indicates that the drawing operation is over, and 0xFE indicates that the operation is skipped (ignored).

## 7.22.2 Software Setting

Open DGUS, click display control – basic graphics, then set the control display area, and configure parameters in the right menu, and finally use instructions to realize drawing.

**Basic graphic display**

X 159 Y 164  
W 401 H 404

Coordinates of the upper left corner of the display area  
Display area width and height

Name definition Basic Graphic

Description pointer(0x) FFFF  
The pointer address is set by default or according to need.

Variable address(0x) 5440  
The variable storage space user can use the address range arbitrarily: 0×1000-0×FFFF.  
The basic graphics occupies the corresponding address according to the drawing command. That is, the non-associated control key addresses need to be spaced apart.  
System variable interface address range: 0×0000-0×0FFF (such as page switching address 0×0084)  
5A A5 17 82 5440 0006 0001 0000 0021 0027 00A3 0040 0168 0168 FF00  
This instruction is developed from 0×5440 address and occupies a total of 10 variable address spaces 0×5440-0×5449.  
The next instruction needs to start from 544A. The 0×0006 cut instruction maps the 0×5440 address.  
0×0001 cut number maps 0×5441 address, 0×0000 cut page ID maps 0×5442 address.  
The mapped address in the instruction will be used as the occupied address.

Dotted/dotted line  
Drawing instructions using line segments: 0×02, 0×03, 0×09, 0×0A will use dashed or dotted lines to display the line segments

Set the dashed (dotted) line format

1 (1-255)  
1 (1-255)  
1 (1-255)  
1 (1-255)

After checking, 0×02, 0×03, 0×09, 0×0A use line drawing commands.  
The line segments will be displayed using dashed or dotted lines. The dotted line (dotted line) format is set in the 4 bytes in turn:  
The number of solid lines in the first segment, the number of dotted lines in the first segment, the number of solid lines in the second segment, and the number of dotted lines in the second segment.  
For example, setting 0×10 0×04 0×10 0×04 will display a dotted line;  
Setting 0×10 0×04 0×02 0×04 will display a dotted line.



### 7.22.3 Example of Instructions

The following instructions are required to cover the whole area of the page with the basic graphics box.

The following instructions should be tested on the resolution of 720\*720. If the resolution is small, it is necessary to pay attention to whether the coordinate point (360,360) is within the display range, so as to avoid not being displayed beyond the range.

### 7.22.4 Set point

Show a red dot

5A A5 0F 82 5440 0001 0001 0168 0168 F800 FF00

0x5AA5: Frame header;

0x0D Data length;

0x82: Write instruction;

0x5440: VP;

0x0001: Draw point;

0x0001: The number of points;

0x0168:0168 (360,360) coordinates;

0xF800: Color value;

0xFF00 End of drawing operation.

Show two red dots

5A A5 15 82 5440 0001 0002 0168 0168 F800 0169 0169 F800 FF00

0x5AA5: Frame header;

0x15 Data length;

0x82: Write instruction;

0x5440: VP;

0x0001: Draw point;

0x0001: The number of points;

0x0168 0168: (360,360)coordinates, 0xF800 color value;

0x0169 0169: (361,361)coordinates, 0xF800 color value;

0xFF00: End of drawing operation.



Show three red dots

5A A5 19 82 5440 0001 0003 0168 0168 F800 0169 0169 F800 016A 016A F800 FF00

0x5AA5: Frame header;

0x 19 Data length;

0x82: Write instruction;

0x5440: VP;0x0001: Draw point;

0x0003: the number of points;

0x0168 0168: (360,360)coordinates;

0xF800 color value;

0x0169 0169: (361,361)coordinates,

0xF800 color value;

0x016A 016A: (362,362)coordinates,

0xF800 color value;

0 xFF00: End of drawing operation.

### 7.22.5 End Point Connection

Two points are connected to form a line

5A A5 13 82 5440 0002 0001 F800 00FC 0168015E 0168 FF00

0x5AA5: Frame header;

0x 13 Data length;

0x82: Write instruction;

0x5440: VP;

0x0002: Draw line segment;

0x0001: The number of connections;

0xF800: Color value;

0x00FC 0168: (252,360)coordinates;

0x015E 0168: (350,360)coordinates;

0 xFF00: End of drawing operation.

Three points are connected to form lines

5A A5 17 82 5440 0002 0002 F800 00FC 0168015E 0168 015E 0136 FF00

0x5AA5: Frame header;

0x 17 Data length;

0x82: Write instruction;

0x5440: VP;

0x0002: Draw line segment;

0x0002: The number of connections;

0xF800: Color value;



0x00FC 0168: (252,360)coordinates;  
0x015E 0168: (350,360)coordinates;  
0x015E 0136: (350,310)coordinates;  
0 xFF00 End of drawing operation.

Four points are connected to form lines

5A A5 1B 82 5440 0002 0003 F800 00FC 0168 015E 0168 015E 01360190 0136 FF00

0x5AA5: Frame header;  
0x 1B Data length;  
0x82: Write instruction;  
0x5440: VP;  
0x0002: Draw line segment;  
0x0003: The number of connections;  
0xF800: Color value;  
0x00FC 0168: (252,360)coordinates;  
0x015E 0168: (350,360)coordinates;  
0x015E 0136: (350,310)coordinates;  
0x0190 0136: (350,310)coordinates;  
0 xFF00: End of drawing operation.

Five points are connected to form lines

5A A5 1F 82 5440 0002 0004 F800 00FC 0168 015E 0168 015E 0136 0190 0136 0190 0168 FF00

0x5AA5: Frame header;  
0x 1F Data length;  
0x82: Write instruction;  
0x5440: VP;  
0x0002: Draw line segment;  
0 x0004: The number of connections;  
0xF800: Color value;  
0x00FC 0168: (252,360)coordinates;  
0x015E 0168: (350,360)coordinates;  
0x015E 0136: (350,310)coordinates;  
0x0190 0136: (350,310)coordinates;  
0x0190 0168: (400,360)coordinates;  
0 xFF00: End of drawing operation.

Six points are connected to form lines

5A A5 23 82 5440 0002 0005 F800 00FC 0168 015E 0168 015E 0136 0190 0136 0190 0168 01C2 0168 FF00

0x5AA5: Frame header;

0x23 Data length;  
0x82: Write instruction;  
0x5440: VP;  
0x0002: Draw line segment;  
0x0005: The number of connections;

0xF800: Color value;  
0x00FC 0168: (252,360)coordinates;  
0x015E 0168: (350,360)coordinates;  
0x015E 0136: (350,310)coordinates;  
0x0190 0136: (400,310)coordinates;  
0x0190 0168: (400,360)coordinates;  
0x01C2 0168: (450 360)coordinates;  
0xFF00: End of drawing operation.

Seven points connected to form lines

5A A5 27 82 5440 0002 0006 F800 00FC 0168 015E 0168 015E 0136 0190 0136 0190 0168 01C2 0168 01C2  
0136 FF00

0x5AA5: Frame header;  
0x27 Data length;  
0x82: Write instruction;  
0x5440: VP;  
0x0002: Draw line segment;  
0x0006: The number of connections;  
0xF800: Color value;  
0x00FC 0168: (252,360)coordinates;  
0x015E 0168: (350,360)coordinates;  
0x015E 0136: (350,310)coordinates;  
0x0190 0136: (400,310)coordinates;  
0x0190 0168: (400,360)coordinates;  
0x01C2 0168: (450 360)coordinates;  
0x01C2 0136: (450,310)coordinates;  
0xFF00: End of drawing operation.

Eight points are connected to form lines

5A A5 2B 82 5440 0002 0007 F800 00FC 0168 015E 0168 015E 0136 0190 0136 0190 0168 01C2 0168 01C2  
0136 01F4 0136 FF00

0x5AA5: Frame header;  
0x2B Data length;  
0x82: Write instruction;



0x5440: VP;  
 0x0002: Draw a line segment;  
 0 x0007: The number of  
 connections; 0xF800: Color value;  
 0x00FC 0168: (252,360)coordinates;  
 0x015E 0168: (350,360)coordinates;  
 0x015E 0136: (350,310)coordinates;  
 0x0190 0136: (400,310)coordinates;  
 0x0190 0168: (400,360)coordinates;  
 0x01C2 0168: (450 360)coordinates;  
 0x01C2 0136: (450,310)coordinates;  
 0x01F4 0136:  
 (500,310)coordinates; 0xFF00:  
 End of drawing operation.

Nine points connected to form lines

5A A5 2F 82 5440 0002 0008 F800 00FC 0168 015E 0168 015E 0136 0190 0136 0190 0168 01C2 0168 01C2  
 0136 01F4

013601F4 0168 FF00

0x5AA5: Frame

header; 0x2F Data

length;

0x82: Write instruction;

0x5440: VP;

0x0002: Draw line segment;

0 x0008: The number of connections;

0xF800: Color value;

0x00FC 0168: (252,360)coordinates;

0x015E 0168: (350,360)coordinates;

0x015E 0136: (350,310)coordinates;

0x0190 0136: (400,310)coordinates;

0x0190 0168: (400,360)coordinates;

0x01C2 0168: (450 360)coordinates;

0x01C2 0136: (450,310)coordinates;

0x01F4 0136: (500,310)coordinates;

0x01F4 0168:

(500,360)coordinates; 0xFF00:

End of drawing operation. Ten

points connected to form lines



5A A5 33 82 5440 0002 0009 F800 00FC 0168 015E 0168 015E 0136 0190 0136 0190 0168 01C2 0168 01C2 0136  
01F4 0168 0226 0168 FF00

- 0x5AA5: Frame header;
- 0x33 Data length;
- 0x82: Write instruction;
- 0x5440: VP;
- 0x0002: Draw line segment;
- 0x0009: The number of connections;
- 0xF800: Color value;
- 0x00FC 0168: (252,360)coordinates;
- 0x015E 0168: (350,360)coordinates;
- 0x015E 0136: (350,310)coordinates;
- 0x0190 0136: (400,310)coordinates;
- 0x0190 0168: (400,360)coordinates;
- 0x01C2 0168: (450 360)coordinates;
- 0x01C2 0136: (450,310)coordinates;
- 0x01F4 0136: (500,310)coordinates;
- 0x01F4 0168: (500,360)coordinates;
- 0x0226 0168: (550,360)coordinates;
- 0xFF00: End of drawing operation.

Eleven points are connected to form lines

5A A5 37 82 5440 0002 000A F800 00FC 0168 015E 0168 015E 0136 0190 0136 0190 0168 01C2 0168 01C2  
01F4 0168 0226 0168 0226 0136 FF00

- 0x5AA5: Frame header;
- 0x37 Data length;
- 0x82: Write instruction;
- 0x5440: VP;
- 0x0002: Draw line segment;
- 0x000A: The number of connections;
- 0xF800: Color value;
- 0x00FC 0168: (252,360)coordinates;
- 0x015E 0168: (350,360)coordinates;
- 0x015E 0136: (350,310)coordinates;
- 0x0190 0136: (400,310)coordinates;
- 0x0190 0168: (400,360)coordinates;
- 0x01C2 0168: (450 360)coordinates;
- 0x01C2 0136: (450,310)coordinates;
- 0x01F4 0136: (500,310)coordinates;
- 0x01F4 0168: (500,360)coordinates;



0x0226 0168: (550,360)coordinates;

0x0226 0136: (550,310)coordinates;

0xFF00: End of drawing operation.

### 7.22.6 Rectangle

Show a rectangle

5A A5 13 82 5440 0003 0001 011E 012C 01AA 018C F800 FF00

0x5AA5: Frame header;

0x13 Data length;

0x82: Write instruction;

0x5440: VP;

0x0003: Draw rectangle;

0x0001: draw one rectangle;

0x011E 012C: (286, 300) upper left coordinate;

0x01AA 018C: (426, 396) lower right coordinate;

0xF800: Color;

0xFF00: The drawing operation has ended.

Show two rectangles

5A A5 1D 82 5440 0003 0002 011E 012C 01AA 018C F800 01AB 018D 01E6 01E6 F800 FF00

0x5AA5: Frame header;

0x 1D Data length;

0x82: Write instruction;

0x5440: VP;

0x0003: Draw rectangle;

0x0002: Draw two rectangles;

0x011E 012C: (286, 300) upper left coordinate;

0x01AA 018C: (426, 396) lower right coordinates;

0xF800: Color;

0x01AB 018D: (427, 397) upper left coordinate;

0x01E6 01E6: (486,486) lower right coordinate;

0xF800: Color;

0xFF00: The drawing operation has ended.

Show three rectangles

5A A5 27 82 5440 0003 0003 011E 012C 01AA 018C F800 01AB 018D 01E6 01E6 F800 01E6 01E6 0162 022C F800 FF00

0x5AA5: Frame header;

0x27: Data length;

0x82: Write instruction;



0x5440: VP;  
0x0003: Draw rectangle;  
0x0003: Draw three rectangles;  
0x011E 012C: (286, 300) upper left coordinate;  
0x01AA 018C: (426,96) lower right coordinate;  
0xF800: Color;  
0x01AB 018D: (426,96) upper left coordinate;  
0x01E6 01E6: (486,486) upper right coordinate;  
0xF800: Color;  
0x01E6 01E6: (486,486) upper left coordinate;  
0x0162 022C: (354,556) lower right coordinate;  
0xF800: Color;  
FF00: The drawing operation ends.

### 7.22.7 Rectangle Fill

Fill a rectangle

5A A5 13 82 5440 0004 0001 011E 012C 01AA 018C F800 FF00

0x5AA5: Frame header;  
0x 13 Data length;  
0x82: Write instruction;  
0x5440: VP;  
0x0004: Fill rectangle;  
0x0001: Fill one rectangle;  
0x011E 012C: (286, 300) upper left coordinate;  
0x01AA 018C: (426, 396) lower right coordinate;  
0xF800: Color;  
0xFF00: The drawing operation has ended.

Fill two rectangles

5A A5 1D 82 5440 0004 0002 011E 012C 01AA 018C F800 01AB 018D 01E6 01E6 F800 FF00

0x5AA5: Frame header;  
0x1D Data length;  
0x82: Write instruction;  
0x5440: VP;  
0x0004: Fill rectangle;  
  
0x0002: Fill two rectangles;  
0x011E 012C (286,00) upper left coordinate;  
0x01AA 018C (426, 96) lower right coordinate;



0xF800: Color;  
0x01AB 018D (427,97) upper left coordinate;  
0x01E6 01E6 (486,486) lower right coordinate;  
0xF800: Color;  
0xFF00: The drawing operation ends.

Fill three rectangles

5A A5 27 82 5440 0004 0003 011E 012C 01AA 018C F800 01AB 018D 01E6 01E6 F800 01E6 01E6 0221 022C

F800 FF00

0x5AA5: Frame header;  
0x27: Data length;  
0x82: Write instruction;  
0x5440: VP;  
0x0004: Fill rectangle;  
0x0003: Fill three rectangles;  
0x011E 012C (286,00) upper left coordinate;  
0x01AA 018C (426,396) lower right coordinate;  
0xF800: Color;  
0x01AB 018D (427,97) upper left coordinate;  
0x01E6 01E6 (486,486) lower right coordinate,  
0xF800: Color;  
0x01E6 01E6 (486,486) upper left coordinate;  
0x0221 022C (545,556) lower right coordinate;  
0xF800: Color;  
0xFF00: The drawing operation has ended.

When filling, the coordinates of the lower right corner are larger than the coordinates of the upper left corner.



### 7.22.8 Image Copy and Paste

This instruction is frequently used and can be used for the clipping display of the progress bar.

5A A5 17 82 5440 0006 0001 0000 0021 002700A3 0040 0168 0168 FF00  
0x5AA5: Frame header;  
0x17: Data length;  
0x82: Write instruction;  
0x5440: VP;  
0x0006: Image copy and paste;  
0x0001: Cut a region;  
0x0021 0027 (33, 39): Top left coordinates of page 0,  
0x00A3 0040 ( 163,64): Bottom right coordinates of page 0;  
0x0168 0168 (360,360): Paste to the current page coordinates;  
0xFF00 Drawing operation ended.

### 7.22.9 Icon Display

Displays icon No. 01 in the No. 48 icon library.

5A A5 0F 82 5440 3007 0001 0168 0168 0001 FF00  
0x5AA5: Frame header;  
0x0F: Data length;  
0x82: Write instruction;  
0x5440: VP;  
0x30: Icon library No.48;  
0x07: 07 icon Write instruction;  
0x0001: display an icon;  
0x0168 0x0168: The starting display coordinates of the upper left corner of the icon (360, 360);  
0x0001: Icon No. 1 in the No. 48 icon library.  
0xFF00: Terminator

Displays icon No. 02 from the icon library No. 48.

5A A5 0F 82 5440 3007 0001 0168 0168 0002 FF00  
0x5AA5: Frame header;  
0x0F: Data length;  
0x82: Write instruction;  
0x5440: VP;  
0x30: Icon library No.48;  
0x07: 07 icon Write instruction;  
0x0001: Display an icon;  
0x0168 0x0168: The starting display coordinates of the upper left corner of the icon (360,360);



0x0002: Icon No. 2 in the No. 48 icon library.

0xFF00: Terminator

### 7.22.10 Enclosed Area Filling

#### Fill a circle

It is necessary to make two basic graphic display boxes and set two different variable addresses, such as one variable address is set to 0x5440, the other is set to 0x5540, and the background image should be solid color, otherwise it cannot be filled normally.

Step 1: The 0x5440 address displays a circle:

5A A5 11 82 5440 0005 0001 0168 0168 0040 F800 FF00

0x5AA5: Frame header;

0x11: Datalength;

0x82: Write instruction;

0x5440: VP;

0x0005: Draw circle;

0x0001: Draws one circle;

0x0168, 0x0168: Circular coordinates (360,360);

0x0040: radius, in pixels;

0xF800: red;

0xFF00: terminator.

Step 2: 5A A5 0F 82 5540 0008 0001 0168 0168 F800 FF00

### 7.22.11 Spectrum Display

Show a spectrum

5A A5 11 82 5440 0009 0001 F800 0168 0168 0190 FF00

0x5AA5: Frame header;

0x11: Datalength;

0x82: Write instruction;

0x5440: VP;

0x0009: Spectrum display;

0x0001: shows a spectrum;

0xF800: Color;

0x0168: x0 coordinates,

0x0168: Y0s start coordinate;

0x0190: Y0e end coordinate;

0xFF00: Drawing operation ended.



5A A5 19 82 5440 0009 0002 F800 0168 0168 0190 F800 0190 0168 01C2 FF00

0x5AA5: Frame header;

0x11: Datalength;

0x82: Write instruction;

0x5440: VP;

0x0009: Spectrum display;

0x0002: Shows two spectrums;

0xF800: Color;

0x0168 (360): x0 coordinates,

0x0168 (360): Y0s start coordinates;

0x0190 (400): Y0e end coordinates;

0xF800: Color;

0x0190 (400) x1 coordinates;

0x0168 (360): Y1s start coordinates;

0x01C2 (450): Y1e end coordinates;

0xFF00: Drawing operation ended.

Display three spectrums

5A A5 21 82 5440 0009 0003 F800 0168 0168 0190 F800 0190 0168 01C2 F800 01C2 0168 01F4 FF00

0x5AA5: Frame header;

0x11: Datalength;

0x82: Write instruction;

0x5440: VP;

0x0009: Spectrum display;

0 x0003: Shows three spectrums;

0xF800: Color;

0x0168 (360): x0 coordinates,

0x0168 (360): Y0s start coordinates;

0x0190 (400): Y0e end coordinates;

0xF800: Color;

0x0190 (400) x1: coordinates;

0x0168 (360): Y1s start coordinates;

0x01C2 (450): Y1e end coordinates;

0xF800: Color;

0x01C2 (450): x2 coordinates;

0x0168 (360): Y2s start coordinates;

0x01F4 (500): Y2e end coordinates;

0 xFF00: Drawing operation ended.



Displays four spectrums

5A A5 29 82 5440 0009 0004 F800 0168 0168 0190 F800 0190 0168 01C2 F800 01C2 0168 01F4 F800 01F4 0168 020D FF00

5A A5 Frame header;

0x11: Data length;

0x82: Write instruction;

0x5440: VP;

0x0009: spectrum display;

0x0004: displays four spectrums;

0xF800: Color;

0x0168 (360): x0 coordinates;

0x0168 (360): Y0s start coordinates;

0x0190 (400): Y0e end coordinates;

0xF800: Color;

0x0190 (400): x1 coordinates;

0x0168 (360): Y1s start coordinates;

0x01C2 (450): Y1e end coordinates;

0xF800: Color;

0x01C2 (450): x2 coordinates;

0x0168 (360): Y2s start coordinates;

0x01F4 (500): Y2e end coordinates;

0xF800: Color;

0x01F4 (500): x3 coordinates;

0x0168 (360): Y3s start coordinates;

0x020D (525): Y3e end coordinates;

0xFF00: Drawing operation ended.

Display five spectrums

5A A5 31 82 5440 0009 0005 F800 0168 0168 0190 F800 0190 0168 01C2 F800 01C2 0168 01F4 F800 01F4 0168 020D

F800 0226 0168 0226 FF00

0x5AA5: Frame header;

0x11: Datalength;

0x82: Write instruction;

0x5440: VP;

0x0009: Spectrum display;

0 x0005: Displays five spectrums;

0xF800: Color;

0x0168 (360) x0 coordinates,  
0x0168 (360) Y0s start coordinates;  
0x0190 (400) Y0e end coordinates;  
0xF800: Color;  
0x0190 (400) x1 coordinates;  
0x0168 (360) Y1s start coordinates;  
0x01C2 (450) Y1e end coordinates;  
0xF800: Color;  
0x01C2 (450): x2 coordinates;  
0x0168 (360): Y2s start coordinates;  
0x01F4 (500): Y2e end coordinates;  
0xF800: Color;  
0x01F4 (500): x3 coordinates;  
0x0168 (360): Y3s start coordinates;  
0x020D (525): Y3e end coordinates;  
0xF800: Color;  
0x0226 (550): x4 coordinates;  
0x0168 (360): Y4s start coordinates;  
0x0226 (550): Y4e end coordinates;  
0xFF00: End drawing operation  
Y0s Y0e settings cannot be the same.

### 7.22.12 Line Segment Display

Show a line segment

5A A5 13 82 5440 000A 0001 F800 0168 0168 0190 0168 FF00  
0x5AA5: Frame header;  
0x11:Data length;  
0x82: Write instruction;  
0x5440: VP;  
0x000Z: Draw line segment;  
0x0001: Draws a line segment;  
0xF800: Color;  
0x0168 0168 (360,360) coordinates;  
0x0190 0168 (400,360) coordinates;  
0xFF00: End drawing operation



Show two line segments

5A A5 1D 82 5440 000A 0002 F800 0168 0168 0190 0168 F800 0168 0190 01C2 0190 FF00

0x5AA5: Frame header;

0x11: Datalength;

0x82: Write instruction;

0x5440: VP;

0x000A: Draw line segment;

0x0002: Draws two line segments;

0xF800: Color;

0x0168 0168 (360,360) coordinates;

0x0190 0168 (400,360) coordinates;

0xF800: Color;

0x0168 0190 (360,400) coordinates;

0x01C2 0190 (450,400) coordinates;

0xFF00 : End drawing operation

Show three line segments

5A A5 27 82 5440 000A 0003 F800 0168 0168 0190 0168 F800 0168 0190 01C2 0190 F800 0168 01C2 01F4  
01C2 FF00

0x5AA5: Frame header;

0x1:1 Datalength;

0x82: Write instruction;

0x5440: VP;

0x000A: Draw line segment;

0x0003:Draw three line segments;

0xF800: Color;

0x0168 0168 (360,360) coordinates;

0x0190 0168 (400,360) coordinates;

0xF800: Color;

0x0168 0190 (360,400) coordinates;

0x01C2 0190 (450,400) coordinates;

0xF800: Color;

0x0168 01C2 (360,450) coordinates;

0x01F4 01C2 (500,450) coordinates;

0xFF00 End drawing operation

Show four line segments

5A A5 31 82 5440 000A 0004 F800 0168 0168 0190 0168 F800 0168 0190 01C2 0190 F800 0168 01C2 01F4  
01C2 F800

0168 01F4 0226 01F4 FF00  
0x5AA5: Frame header;  
0x11: Datalength;  
0x82: Write instruction;  
0x5440: VP;  
0x000A: Draw line segment;  
0x0004: Draw four line segments;  
0xF800: Color;  
0x0168 0168 (360,360) coordinates;  
0x0190 0168 (400,360) coordinates;  
0xF800: Color;  
0x0168 0190 (360,400) coordinates;  
0x01C2 0190 (450,400) coordinates;  
0xF800: Color;  
0x0168 01C2 (360,450) coordinates;  
0x01F4 01C2 (500,450) coordinates;  
0xF800: Color;  
0x0168 01F4 (360,500) coordinates;  
0x0226 01F4 (550,500) coordinates;  
0xFF00 End drawing operation

Step 2: Based on the 0x5540 address, fill the circular enclosed area with the center of the circle as the seed point:

5A A5 0F 82 5540 0008 0001 0168 0168 F800 FF00  
0x5AA5: Frame header;  
0x0F: Data length;  
0x82: Write instruction;  
0x5440: VP;  
0x0008 Fill instruction;  
0x0001 Fill the enclosed area;  
0x0168, 0x0168 The coordinates of the seed point of the circular enclosed area (360,360);  
0xF800 Red;  
0xFF00 Terminator.



### 7.22.13 Rectangular Field XOR Highlighting Inverse Color Display

Rectangular field XOR, highlight and inverse color display once

5A A5 13 82 5440 000D 0001 0168 0168 0190 0190 F800 FF00

0x5AA5: Frame header;

0x13 Data length;

0x82: Write instruction;

0x5440: VP;

0x000D: Rectangle XOR;

0x0001: XOR the rectangular field once;

0x0168 0168 (360,360) coordinates;

0x0190 0190 (400,400) coordinates;

0xF800: Color;

0xFF00 Drawing operation ended.

Rectangular field XOR, highlight and invert twice

5A A5 1D 82 5440 000D 0002 0168 0168 0190 0190 F800 0190 0190 01C2 01C2 F800 FF00

0x5AA5: Frame header;

0x1D: Data length;

0x82: Write instruction;

0x5440: VP;

0x000D: Rectangle XOR;

0x0002: Rectangular XOR twice;

0x0168 0168 (360, 360) coordinates;

0x0190 0190 (400, 400) coordinates;

0xF800: Colors;

0x0190 0190 (400, 400) coordinates;

0x01C2 01C2 (450, 450) coordinates;

0xF800: Colors;

0xFF00: Drawing operation ended.

Rectangular field XOR, highlight and inverse color display three times

5A A5 27 82 5440 000D 0003 0168 0168 0190 0190 F800 0190 0190 01C2 01C2 F800 01C2 01C2 01F4 01F4

F800 FF00

0x5AA5: Frame header;

0x1D: Data length;

0x82: Write instruction;

0x9010: VP;

0x000D: Rectangle XOR;

0x0003: Rectangular field XOR three times;



0x0168 0168 (360, 360) coordinates;

0x0190 0190 (400, 400) coordinates;

0xF800: Colors;

0x0190 0190 (400, 400) coordinates;

0x01C2 01C2 (450, 450) coordinates;

0xF800: colors;

0x01C2 01C2 (450,450) coordinates;

0x01F4 01F4 (500,500) coordinates;

0xF800: Colors;

0xFF00: Drawing operation ended.

### 7.22.14 Two-Color Bitmap Display

#### (1)Function

Two- color bitmap can be used for special characters or graphics display .

Generally, special characters or graphics usually use icon or the background image ,here mainly for your understanding of the function.

#### (2) Explanation

( 1) X, Y: the bitmap displays the coordinates of the upper left corner of the rectangular field. The height of the X coordinate is the judgment condition, and the starting position of the character area is displayed.

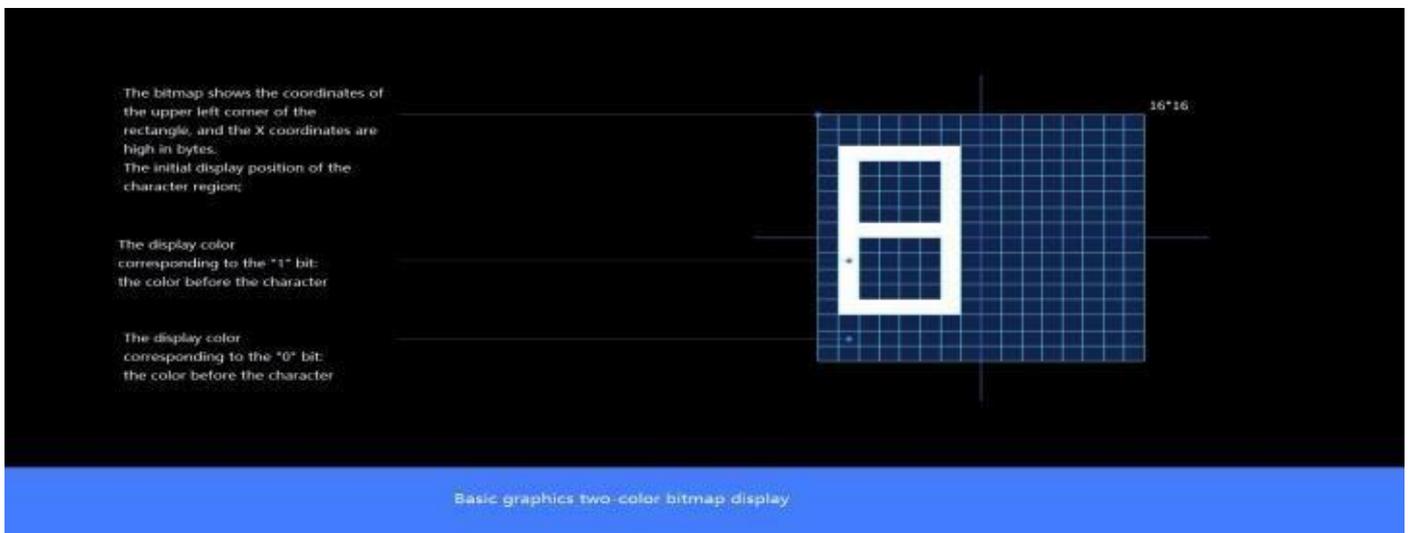
②X\_Dots, Y\_ Dots: the number of lattice arrays in X and Y directions;

③ Display color corresponding to "1 " bit: foreground color;

④ Display color corresponding to "0" bit: background color;

⑤ Display data: directly use modulus data.

|        |                                |      |           |   |
|--------|--------------------------------|------|-----------|---|
| 0x000E | Two-color<br>bitmap<br>display | 0x00 | (x,y)s    | The bitmap shows the coordinates of the upper left corner of the rectangular filed. The height of the X coordinate is byte as the judgment condition.   |
|        |                                | 0x02 | X_Dots    | Number of bitmap X_direction lattice.   |
|        |                                | 0x03 | Y_Dots    | Number of bitmap Y_ direction lattice.  |
|        |                                | 0x04 | Color1    | Display color corresponding to "1"bit   |
|        |                                | 0x05 | Color0    | The display color corresponding to the "0" bit; if Color0 and Color1 are set to be the same, it means that the "0" bit does not need to be displayed and skipped directly.                          |
|        |                                | 0x06 | Data Pack | Display data, MSB mode; in order to facilitate users to read and write data, each line of data must be aligned to a word, that is, the next line of data always starts from a new data word (Word). |



Bitmap Character Description



(3) the character "8" dot matrix module program

Website: <https://www.zhetao.com/fontarray.html>

(4) Instruction examples

① Display character graphic "8" in the range of 15\*15 dot matrix

Tx: 5A A5 35 82 5440 000E 0001 0168 0168 000F 000F F800 07E0 0000 0000 0000 3C00 4200 4200 4200 2400  
1800 2400

4200 4200 4200 3C00 0000 0000 FF00

0x5AA5: Frame header;

0x35: Data length 53 bytes.

0x82: Write instruction;

0x5440: VP;

0x000E: Two-color bitmap instruction;

0x0001: Write a character;

0x0168, 0x0168: The upper left coordinates of the character;

0x000F, 0x000F: The number of dots in the character display area;

0xF800: Color1 character color;

0x07E0: Color0 character background color;

0x0000 0000 0000 3C00 4200 4200 2400 1800 2400 4200 4200 4200 4200 4200 4200 3C00 0000 0000:

0xFF00: End character.

② Display character graphic "C" within the range of 15 \* 15 dot matrix

Tx: 5A A5 35 82 5440 000E 0001 0168 0168 000F 000F F800 07E0 6000 91F4 960C 6C04 0804 1800 1800 1800  
1800 1800 1800 1800 1800 1800 0800 0C04 0608 01F0 0000 FF00



### 7.22.15 Four-Color Bitmap Display

Each pixel point is represented by 2 bits of a color, 00/ 1 1 combined into a total of 4 numbers 00, 01, 10, 11, which can correspond to 4 colors respectively, and then convert the 32 bit data of the double word into hexadecimal data, which is the data to be sent.

The number of pixel matrix in the bitmap area must be a multiple of 16 , otherwise it will not be displayed.

Application data can be refreshed to show different graphics as needed. Users can also use multiple display controls of different vp to make all kinds of effects.

#### (1) Instruction storage format

|        |                   |      |   |            |   |
|--------|-------------------|------|---|------------|---|
| 0x0012 | Four-color bitmap | 0x00 | 2 | ( x, y ) s | The upper left coordinates of the displayed bitmap, and the high byte of the x- coordinate is the judgment condition.<br><i>The starting VP address must be even ( double-word aligned) .</i> |
|        |                   | 0x02 | 1 | X_Dots     | The number of bitmap X-direction pixels, which must be divisible by 16 .  |
|        |                   | 0x03 | 1 | Y_Dots     | The number of bitmap Y- direction pixels.   |
|        |                   | 0x04 | 1 | Color0     | The display color corresponding to the "00" bit.  |
|        |                   | 0x05 | 1 | Color1     | The display color corresponding to the "01" bit.  |
|        |                   | 0x06 | 1 | Color2     | The display color corresponding to the "10" bit.  |
|        |                   | 0x07 | 1 | Color3     | The display color corresponding to the "11" bit.  |
|        |                   | 0x08 | N | Data_Pack  | Display data in MSB.<br>Each line of data must be aligned to a double word ( 16 pixels) for users to read and write the data.   |

#### (2) Instruction examples

Take 16 \* 16 pixel matrix as an example to display 4- color bitmap basic graphics, the graphic pattern to be displayed is as follows .

Tx: 5A A5 59 82 5440 0012 0001 0168 0168 0010 0010 FFFF F800 07E0 0000 0000 0000 0000 0000 0000 0000 0000

00FF AA00 00FF AA00 00FF AA00 00FF AA00 0055 FF00 0055 FF00 0055 FF00 0055 FF00 0000 0000 0000 0000 0000 0000 0000

0000 FF00

Meaning:

0x5A A5: frame header.

0x59: 89 bytes of data length.

0x82: write instruction.

0x5440: vp.

0x0012: four-color bitmap.

0x0001: number of graphics.

0x0168 0x0168: the upper left coordinates of the displayed bitmap

0 x0010 0x0010: the number of bitmap X- and Y-direction pixels, must be divisible by 16, e.g. 0x 10 for 16 pixels, 0 x20 for 32 pixels, otherwise it will not be displayed.

0xFFFF: white, color 0; set the display color corresponding to the "00" bit.

0xF800: red, color 1; set the display color corresponding to "01" bit.



0x07 E0: green, color 2; set the display color corresponding to "10" bit.

0x0000: black, color 3; set the display color corresponding to the "11" bit.

0x0000 0000: line 1 double word data.

0x0000 0000: line 2 double-word data.

0x0000 0000: line 3 double-word data.

0x0000 0000: line 4 double-word data.

0x00FF AA00: line 5 double-word data.

0x00FF AA00: line 6 double-word data.

0x00FF AA00: line 7 double-word data.

0x00FF AA00: line 8 double-word data.

0x0055 FF00: line 9 double-word data.

0x0055 FF00: line 10 double-word data .

0 x0055 FF00: line 11 double-word data.

0 x0055 FF00: line 12 double-word data.

0x0000 0000: line 13 double-word data.

0x0000 0000: line 14 double-word data.

0x0000 0000: line 15 double-word data.

0x0000 0000: line 16 double-word data.

0xFF00: terminator.



**(3)** Data description

|        |  |
|--------|--|
| Color0 | The display color corresponding to the "00" bit. |
| Color1 | The display color corresponding to the "01" bit. |
| Color2 | The display color corresponding to the "10" bit. |
| Color3 | The display color corresponding to the "11" bit. |

Each pixel point is represented by 2 bits of a color, 00/11 combined into a total of 4 numbers 00, 01, 10, 11, which can correspond to 4 colors respectively, and then convert the 32 bit data of the double word into hexadecimal data, which is the data to be sent.





## 7.22.16 Ellipse Display

Ellipse display + fill (the ellipse function needs to use the UI kernel of V41\_Increase Ellipse Display\_20210802). For ellipse display example, set VP of basic graphic display control to 0x5440. It is the same for the ellipse filling. Filling closed area requires the background image to be a solid color, otherwise the filling will display abnormally.

Step 1: Display an Ellipse

```
5A A5 13 82 5440 0011 0001 0168 0168 0032 0016 F800 FF00
```

0x5AA5: Frame header;

0x13: Data length; 0x82:

Write instruction;

0x5440: VP;

0x0011: Draw ellipse instruction;

0x0001: Draw an ellipse;

0x0168 0168: The coordinates of the ellipse center;

0x0032: The diameter of the major axis of the ellipse;

0x0016: The diameter of the minor axis of the ellipse;

0xF800: Arc color;

0xFF00: Terminator.

Step 2: Ellipse Fill

```
5A A5 0F 82 5540 0008 0001 0168 0168 07FF FF00
```

0x5AA5: Frame header;

0x0F: Data length; 0x82:

Write instruction;

0x5440: VP;

0x0008: Fill instruction;

0x0001: Fill the closed area;

0x0168 0168: The coordinates of the ellipse center;

0xF800: Fill color;

0xFF00: Terminator.



## 7.23 Process Bar

### 7.23.1 Instruction Storage Format

| SP offset | Definition       | Length (bytes) | Description  |
|-----------|------------------|----------------|--|
|           | 0x5A23           | 2              |  |
|           | *SP              | 2              | Variable SP, 0xFFFF means loaded by configuration file.  |
|           | 0x000D           | 2              |  |
| 0x00      | *VP              | 2              | Process bar data display pointer.  |
| 0x01      | (Xs, Ys)         | 4              | The coordinates of the upper left corner of the process bar display area.  |
| 0x03      | (Xe, Ye)         | 4              | The coordinates of the lower right corner of the process bar display area.   |
| 0x05      | Border color     | 2              |  |
| 0x06      | Foreground color | 2              |  |
| 0x07      | Background color | 2              |  |
| 0x08      | Variable Max.    | 2              | Corresponding to 100% progress, integer, -32768 to 32767.  |
| 0x09      | Variable Min.    | 2              | Corresponding to 0% progress, integer, -32768 to 32767.  |
| 0x0A_H    | Mode             | 1              | .7 Return the percentage data of the progress bar to the specified variable control, 0=do not return, 1=return.<br>.1-.0 Border display mode<br>0x00: Display the outer border and fill the background.<br>0x01: Do not display the outer border, fill the background.<br>0x02: Display the outer border without filling the background.<br>0x03: Do not display the outer border, do not fill the background. |
| 0x0A_L    | Direction        |                | 0x00=Right 0x01=Left 0x02=Up 0x03=Down   |
| 0x0B_H    | Type of data     |                | 0x00=integer 0x01=variable high byte 0x02=variable low byte  |
| 0x0B_L    | Reserve          | 1              | Reserved, write 0x00.  |
| 0x0C      | *VP_RT           | 2              | The calculated progress bar percentage (unit 1%) returns the pointer address, and the returned data is an integer, 0x0000-0x0064.  |

## 7.23.2 Software Setting

The screenshot shows the 'progress bar' configuration window with the following settings and annotations:

- Coordinates of the upper left corner, defining the position of the graphic:** X: 570, Y: 36
- Width and height, defining the touch range area:** W: 270, H: 370
- Dynamically modify attributes in combination with SP offsets:** SP(0x): FFFF, VP(0x): 2000. Note: The variable storage space 0x1000-0xFFFF can be freely used by users. The progress bar display occupies 1 address, that is, the addresses of non-associated controls need to be spaced apart. System variable interface address range: 0x0000-0x0FFF (such as page switching address 0x0084)
- Color settings:** border color: F800 (red), forecolor: 07E0 (green), backcolor: FFE0 (yellow). A diagram shows a bar with a green foreground, a red border, and a yellow background.
- Data range:** max: 100, min: 0. Note: The maximum and minimum values are the data range, integer type -32768 to 32768. Such as setting 0~100, -100~100.
- rate addr:** Unchecked. Note: You can associate a data variable display control, the address of the variable is the same as the address of the process bar, and it will be displayed in the form of data and process bar after adjusting the parameters. Check: If you set the percentage address 0x6050, set the data variable display address 0x6050, it is recommended to distinguish the percentage address from the variable address. After adjusting the parameters, the data is displayed in the form of percentage and process bar.
- border and background:** Checked. Note: Check Show Border and Fill Background. Uncheck Show Border and Fill Background Effects. It is recommended to check it so that the graph range can be displayed intuitively. Visual examples show 'Right', 'Left', 'UP', and 'Down' indicators.
- orientation:** right. Note: Integer type -32768 to 32767; When setting the high/low byte, please pay attention to the same display variables to avoid data display errors.
- type:** int. Note: The data value displayed on power-on also needs to write 0x38 at address 0x05 of the CFG file.
- ini:** 0. Note: Indicates that data uploading is enabled, 22 initial value files are loaded, and touch sound is enabled.

## 7.23.3 Example of Instructions

① For example, if the data range is 0-100, the data display is consistent with the percentage.

The data shows 0, the percentage shows 0%, and the process does not show: 5A A5 05 82 6000 0000

The data shows 50, the percentage shows 50%, and the process does not show half: 5A A5 05 82 6000 0032

The data shows 100, the percentage shows 100%, and the process shows: 5A A5 05 82 6000 0064

② For example, if the data range is -32768 to 32767, the percentage will be displayed in equal parts according to the average, and the data variables will be displayed normally.

The data display is -32768, the percentage display is 0%: 5A A5 05 82 6000 8000 (the complement of the negative number sent by the data)

Data display 0, percentage display 50%: 5A A5 05 82 6000 0000

The data shows 327, the percentage shows 50%: 5A A5 05 82 6000 0147

The data shows 328, and the percentage shows 51%: 5A A5 05 82 6000 0148

Display data 982, percentage display 51%: 5A A5 05 82 6000 03D6

Display data 983, percentage display 52%: 5A A5 05 82 6000 03D7

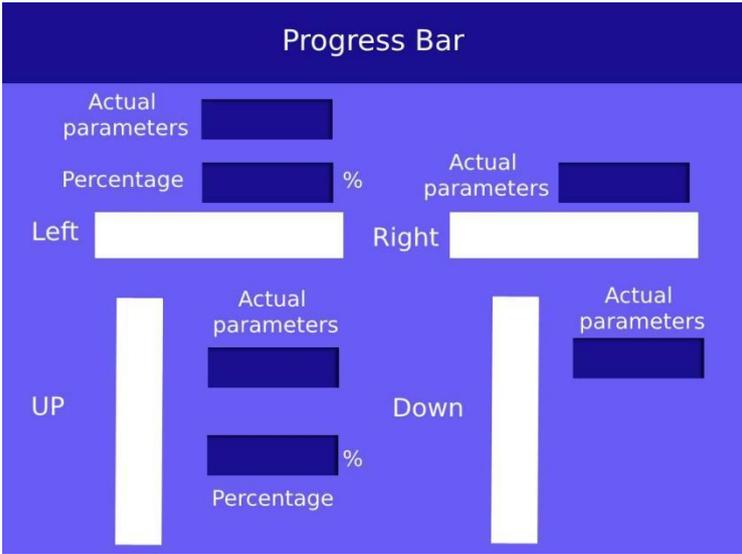
The data shows 32767, and the percentage shows 100%: 5A A5 05 82 6000 7FFF

③ For example, if the data shows -100 to 100, the percentage will be displayed in equal parts according to the average, and the data variables will be displayed normally.

Display data -100, percentage display 0%: 5A A5 05 82 6030 FF9C (0x FF9C negative number 100. Complement code of negative number: the sign bit is 1, and the remaining bits are the original code of the absolute value of the number; then the whole number Add 1. If you enter -200, it will return 5A A5 06 83 68 20 01 FF 38, and the returned data 0xFF38 is the entered data -200.)

- Display data 0, percentage display 50%: 5A A5 05 82 6030 0000
- Display data 1, percentage display 51%: 5A A5 05 82 6030 0001
- Display data 2, percentage display 51%: 5A A5 05 82 6030 0002
- Display data 3, percentage display 52%: 5A A5 05 82 6030 0003

**7.23.4 Effect**



**7.24 Zone Scrolling**

The zone scrolling control is to move the contents of the specified area around, and the moving direction can be set. It can be used to simply realize dynamic running effects such as flow charts and progress bars on the screen. The VP is handled by the underlying system, and the user does not need to set it.

**7.24.1 Instruction Storage Format**

| SP offset | Definition | Length (bytes) | Description  |
|-----------|------------|----------------|--|
| 0x00      | VP         | 2              | A word variable to hold the panning data. Not available to the user. |
| 0x01      | (x, y) s   | 4              | The coordinates of the upper left corner of the moving area.         |
| 0x02      | (x, y) e   | 2              | The coordinates of the upper right corner of the moving area.        |
| 0x05      | Dis_Move   | 2              | Move speed.  |
| 0x06_H    | Mode_Move  | 2              | Shift left 0x01= Shift right 0x02= Shift up 0x03= Shift down.        |

## 7.24.2 Software Setting

The screenshot shows the 'Zone Scrolling' configuration window. It includes fields for X (48), Y (209), W (276), and H (366). Below these are 'Name definition' (Zone Scrolling), 'Description pointer(0x)' (FFFF), a 'Scroll speed' section with a value of 1, and a 'Move method' dropdown set to 'Move up'. Blue callout boxes provide the following descriptions:

- Coordinates of the upper left corner of the display area (pointing to X and Y).
- Display area width and height (pointing to W and H).
- The pointer address is set by default or according to need. (pointing to Description pointer(0x)).
- Set the moving distance of the selected area in a single cycle, that is, the number of pixels. (pointing to Scroll speed).
- The mode of moving up, down, left, and right of the frame selection area. (pointing to Move method).

The variable is occupied by the system and should not be used by the user.

## 7.25 QR Code

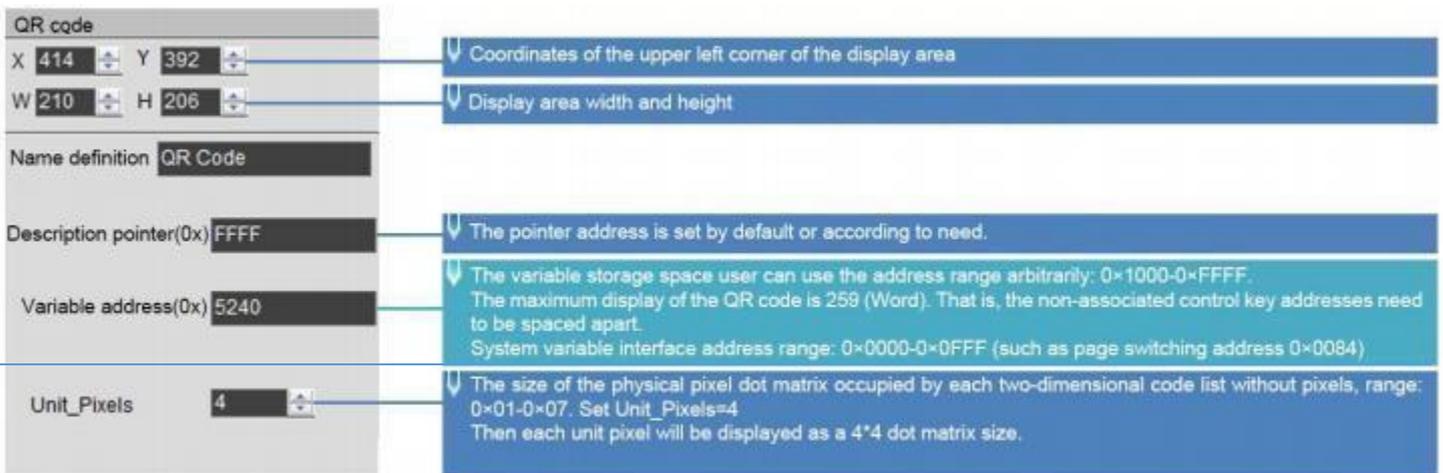
The QR code control is to display the designated QR code on the screen according to the designated content.

### 7.25.1 Instruction Storage Format

| SP offset | Definition  | Length (bytes) | Description  |
|-----------|-------------|----------------|--|
| 0x00      | VP          | 2              | Variable pointer.<br>The content of the QR code is up to 45 8 Bytes, and 0 x0000 or 0xFFFF is the terminator.  |
| 0x01      | (x, y)      | 4              | The coordinate position of the upper left corner displayed by the QR code.<br>There are two types of QR code graphics: 45 * 45 unit pixels ( data less than 155 bytes) and 73*73 unit pixels (data less than 459 Bytes). |
| 0x03      | Unit_Pixels | 2              | The physical pixel lattice size occupied by each QR code unit pixel, 0 x01- 0 x07 .<br>Set Unit_Pixels=4 , then each unit pixel will be displayed as 4* 4 lattice size.  |
| 0x05:H    | Fix_Mode    | 1              | 0 x01: Fixed as 73*73 unit pixel format QR code.<br>The rest: automatically match the QR code size according to the Data length.   |
|           | Reserved    | 1              | Undefined, write 0x00.   |

### 7.25.2 Software Setting

Open DGUS, click display control – QR code, then set the control display area, and configure VP, the physical pixel size occupied by each QR code unit pixel, etc.



### 7.25.3 Example of Instructions

1) Send an instruction to display the URL `http:// www.dwin.com .cn/` QR code.

5A A5 1C 82 5240 68 74 74 70 3A 2F 2F 77 77 77 2E 64 77 69 6E 2E 63 6F 6D 2E 63 6E 2F FFFF

0x5AA5: Frame header;

0x 1C: Data length;

0x82: Write instruction;

0x5240: VP;

0x68 74 74 70 3A 2F 2F 77 77 77 2E 64 77 69 6E 2E 63 6F 6D 2E 63 6E 2F : the ASCII code of the URL;

0xFFFF: Terminator.

If the size of the QR code needs to be fixed, You can set SP to 0x8000, offset 5 bits from SP.

Then write 0x01 to 0.8005 to fix the size of QR code.

2) QR code background and foreground color modification.

Set the QR code display control VP address 0x8000 with the following example command:

Turn off default black and white, 0x04L .7 bits write 0:

5A A5 0582 8004 0000

On to change the color, 0x04L .7 bits write 1: 5A A5 0582 8004 0000

5A A5 0582 8004 0080; 1000 0000 binary converted to hexadecimal as 0x80

Background color change to **blue**, VP address 0x8000 + 0x05 change background color

5A A5 0582 8005 1800; 0x1800 blue value

Background color change to **green**, VP address 0x8000 + 0x05 change background color

5A A5 0582 8005 07E0; 0x07E0 green value

Foreground color change to **red**, VP address 0x8000 + 0x06 change foreground color

5A A5 0582 8006 C800; 0xC800 red value

Foreground color change to purple, VP address 0x8000 + 0x06 change foreground color

5A A5 0582 8006 B900; 0xB900 purple value

#### 7.25.4 QR code instruction display effect



## 7.26 Brightness

The brightness control is used to adjust the display brightness of the specified display area, and is used to highlight or fade the background display.

The difference between this control and the backlight adjustment system variable interface is that the backlight is to adjust the brightness of the whole screen.

### 7.26.1 Instruction Storage Format

| SP Offset | Definition | Length (bytes) | Description  |
|-----------|------------|----------------|--|
| 0x00      | *VP        | 2              | Upper- left coordinates of the Icons to display.                 |
| 0x01      | (Xs, Ys)   | 4              | Specifies the coordinates of the upper left corner of the area.  |
| 0x03      | (Xe, Ye)   | 4              | Specifies the coordinates of the upper right corner of the area. |
|           | Reserved   | 18             | Undefined, write 0x00 .  |

### 7.26.2 Software Setting

**Brightness**

X 213 Y 386

W 284 H 112

Name definition **Brightness**

Description pointer(0x) **FFFF**

Variable address(0x) **5432**

- Coordinates of the upper left corner of the display area
- Display area width and height
- The pointer address is set by default or according to need.
- The variable storage space user can use the address range arbitrarily: 0x1000-0xFFFF. Area brightness adjustment occupies 1 address. That is, the non-associated control key addresses need to be spaced apart. System variable interface address range: 0x0000-0x0FFF (such as page switching address 0x0084)

### 7.26.3 Examples of Instruction

5A A5 05 82 5432 0032

0x5AA5: Frame header;

0x05: Data length;

0x82: Write instruction;

0x5432 VP;

0x0032 Brightness value. (Range 0x00-0x64, 100-level brightness adjustment)

## 7.27 Data Transmit

After the page is switched, the predefined data is transferred to the variable or UART once.

### 7.27.1 Instruction Storage Format

|      |        |                 |    |   |
|------|--------|-----------------|----|---|
| 0x00 |        | 0x5A30          | 2  |   |
| 0x02 |        | *SP             | 2  | Variable description pointer, 0xFFFF means loaded by the configuration file.  |
| 0x04 |        | 0x000D          | 2  |   |
| 0x06 | 0x00   | *VP             | 2  | The target pointer for data transfer.<br>When the variable is valid for the first time on the current page, the data is written to the pointer pointed to by VP according to the following format.<br>Variable memory space:<br>PAGE_ID + predefined 22 bytes data<br>Each variable occupies 12 words of space. |
| 0x08 | 0x01:H | AUTO_COM<br>En  | 1  | Select whether to actively upload the variable data to the user when the variable is first started on the current page<br>Serial port:<br>0xFF=upload (each page can only have 1 upload variable), the rest =no upload.   |
| 0x09 | 0x01:L | Reserved        | 1  | Write 0x00  |
| 0x0A | 0x06   | Predefined data | 22 | The max length of data preset by the user is 22 bytes.  |

### 7.27.2 Software Setting

The screenshot shows the 'Auxiliary variable display' configuration window. The settings are as follows:

- X:** 684, **Y:** 191 (Coordinates of the upper left corner of the display area)
- W:** 227, **H:** 125 (Display area width and height)
- Name definition:** Data transmit
- Description pointer:** FFFF (Default or set pointer address as needed)
- Variable address:** 0000 (The variable storage space user arbitrarily uses the address range: 0x1000-0xFFFF. At most 11 address spaces are occupied, and the non-associated key addresses need to be spaced apart. The target pointer of data transfer, when the variable is valid for the first time on the current page. Write data to the variable memory space pointed to by VP in the following format: PAGE\_ID + predefined 22-byte data occupies 12 words space for each variable.)
- Whether the data is uploaded to the serial port:** Yes (Choose whether to automatically send the variable data to the user serial port when the variable is started for the first time on the current page. 0xFF = upload (each page can only have one upload variable), the rest = no upload.)
- Predefined data:** (The longest 22-byte data preset by the user.)

## 7.28 Video

### 7.28.1 Instruction Storage Format

| Address | SP offset | Definition  | Data length (byte) | Description   |
|---------|-----------|-------------|--------------------|---|
| 0x00    |           | 0x5A31      | 2                  |   |
| 0x02    |           | *SP         | 2                  | Parameter pointer. 0xFFFF means loaded by the configuration file.   |
| 0x04    |           | 0x0008      | 2                  |   |
| 0x06    | 0x00      | *VP         | 12                 | <p>Variable data pointer, must be even, occupies 12 bytes.</p> <p>User control interface, double word, VP and VP+ 1 position are defined as follows.</p> <p>D3 : 0 x5A means digital video playback is on, otherwise it is off.</p> <p>D2 : Playback status control, DGUS will clear after processing.</p> <p>0 x01= Stop, the screen stays at the first frame.</p> <p>0x02 = Pause/resume playback.</p> <p>0 x0 3 = Playback from the specified position ( position is determined by D1:D0).</p> <p>D1 : D0: Playback start position in seconds, valid only when D2= 0 x03 .</p> <p>Status feedback interface, two double words, VP+ 2 to VP+ 5 positions, user can only read.</p> <p>D7: Current playback status feedback, 0x00= Stop, 0x01= Playing. D6 :D4: Undefined.</p> <p>D3 :D2 Total video length, 0x0000-0xFFFF, in seconds.</p> <p>D1 :D0 Current playing video position, 0x0000-0xFFFF in seconds.</p> |
| 0x08    | 0x01      | (x, y)      | 4                  | The coordinates of the upper left corner of the video display window on the screen.   |
| 0x0C    | 0x03      | Wide_X      | 2                  | The width of the video display window on the screen.  |
| 0x0E    | 0x04      | Wide_Y      | 2                  | The height of the video display window on the screen.   |
| 0x10    | 0x05:H    | FPS_Video   | 1                  | Frame rate of video playback(frames/sec)  |
| 0x11    | 0x05:L    | Type_Video  | 1                  | <p>Digital video category.</p> <p>0x00 : Digital video for the combination of ICL files and WAE files.</p> <p>Other: undefined.</p>   |
| 0x12    | 0x06      | File_ID_ICL | 2                  | ID of the ICL file where the video picture is stored, 0x0000-0xFFFF.  |
| 0x14    | 0x07      | File_ID_WAE | 2                  | <p>The WAE file ID where the audio is stored, 0x0000- 0xFFFF.</p> <p>The audio is in 32KHz sampled WAV format and is saved in 0x00 of the file.</p>   |
| 0x16    | 0x08:H    | Reserved    | 10                 | Undefined   |

## 7.28.2 Software Setting

**数字视频**

X: 461 Y: 239  
W: 157 H: 103  锁定

VP(0x): 5000  
SP(0x): FFFF  
帧率: 30

数字视频类型:  
ICL, WAE组合

ICL文件ID: 36  
WAE文件ID: 55

Upper left coordinates of the control

Width and height of the control

Variable storage space range available to users: 0X1000-0XFFFF. Digital video playback occupies 6 addresses, i.e. the addresses of non-associated controls need to be spaced 6. System variable interface address range: 0X0000-0X0FFF (e.g. page switch address 0X84)

SP start address, combined. Combining the offset address can dynamically modify the control properties, which can generally be set by default.

ICL video file frame rate, e.g. 24 fps, 30 fps, determined according to the source video.

ICL is the video file and WAE is the audio file. Generally, set by default.

The ICL file and the WAE file together occupy 16MB of memory space. E.g., the size of 36.ICL file in DEMO is 4.28MB, it takes up  $4.28 \times 1024 / 256 = 17.12$ , which requires 18 256KB subspaces,  $36 + 18 = 54$ , the next file needs to be named from 55.

## 7.28.3 ICL&WAE file

ICL file making procedures:

DWIN DGUS

File Common ICON Display Text Show Graph Show Touch Control Edit Tool Setting Display

New Open Save SaveAs Close Generate Export Import

1. Welcome  
Click "welcome"  
Engineering management

New Open file

History

- C:\Users\Administrator\Desktop\DGUS\_V76414\DGUS\_V7641
- F:\test02
- C:\Users\Administrator\Desktop\item\md\自定义宏宏新格式数据\07\保存实例
- F:\test01

DGUS config tool

DEWAVE Generator Pictures conversion **2. WAE Generator** USB

DWIN ICL Generator Gray Word Library Generator Click "WAE generator" Int

Keyboard Role Char DGUS Font ED

WAE Generator Functions: 1. Converting music files to wav; 2. Converting music files to Devion custom audio format WAE.

DwinPCKits-Movie Tool

ICL Test WAE Test CPU Edit TTL Download SP Order CMC **3. Movie Test**

4. open video Choose the video file you have prepared.

set resolution ratio

5. X1: 0 Y1: 0 Set the video resolution as you want as long as within the screen resolution range.

Picture saving accuracy Frame:

6. Ultra-low precision It is the image quality and ultra low is available.

7. create wav file The audio is in 32KHZ, sampled WAV format and saved in the 0X00 segment of the file. At this point the ID must be named 00, otherwise the video can not be played. Then save and convert it to the WAE format.

8. create icl file Save the ICL video (image) files in the DWIN\_SET folder of the project. Note that the naming should not conflict. The naming can be done as ID+length+frame rate format for easy file identification, e.g. example: 36\_65 length 30 frames per second. The ICL file and the WAE file together occupy 16MB of the memory space. E.g., the size of 36.ICL file in DEMO is 4.28MB, and it takes up  $4.28 \times 1024 / 256 = 17.12$ , which requires 18 256KB subspaces,  $36 + 18 = 54$ , the next file needs to be named from 55.

WAE file making procedures:

The screenshot displays the DWIN DGUS software interface. The main window is titled "DWIN DGUS" and features a menu bar with options: File, Common, ICON Display, Text Show, Graph Show, Touch Control, Edit Tool, Setting, and Display. Below the menu bar is a toolbar with icons for New, Open, Save, Save As, Close, Generate, Export, and Import. The main workspace is divided into several panels:

- 1. Welcome:** Contains a "Click 'welcome'" button and "Engineering management" options: "New" and "Open file".
- History:** Lists recent actions:
  1. C:\Users\Administrator\Desktop\DGUS\_V76414\DGUS\_V76414
  2. F:test02
  3. C:\Users\Administrator\Desktop\me\me\中文安装指南\新式管理\0\中文安装新式
  4. F:test01
- DGUS config tool:** A grid of tool buttons including "WAE Generator", "Click 'WAE generator'", "Keyboard", "Bale Char", and "DGUS Font".

The "DwinPCKis-WAE Tool" dialog box is open, showing the following steps and instructions:

- 3. WAE Tool:** A blue box says "Click 'WAE Tool'".
- 4. Select Files:** A blue box says "Click 'Select Files' and choose the 0.WAV file you have prepared." The dialog shows "Select Audio" and "Select Files" buttons.
- 5. Data Format:** A blue box says "Choose '32kHz 16bit WAV'". The dialog shows "Data Format" set to "32kHz 16bit WAV".
- 6. Generate WAE:** A blue box says "According to the video file size, calculate the next file naming ID, which can be ID + duration + file format to facilitate file identification, such as the example: 55 - 32kHz 16bit WAV\_65 duration. The files are put into the project DWIN\_SET folder, which is convenient to download and find." The dialog shows a "Generate WAE" button.

The dialog also includes a "File Name" field, a "Progress" bar, and an "OK" button.

## 7.28.4 Instruction examples

| Address | SP offset | Definition | Data length( byte) | Description  |
|---------|-----------|------------|--------------------|--|
| 0x06    | 0x00      | *VP        | 12                 | <p>Variable data pointer, must be even, occupies 12 bytes.</p> <p>User control interface, double word, VP and VP+1 position are defined as follows.</p> <p>D3 : 0x5A means digital video playback is on, otherwise it is off.</p> <p>D2 : Playback status control, DGUS will clear after processing.</p> <p>0x01 = Stop, the screen stays at the firstframe.</p> <p>0x02 =Pause/resume playback.</p> <p>0x03 = Playback from the specified position ( position is determined by D1:D0).</p> <p>D1:D0: Playback start position in seconds, valid only when D2=0x03.</p> <p>Status feedback interface, two double words, VP+ 2 to VP+ 5 positions, user can only read.</p> <p>D7:Current playback status feedback, 0x00=Stop, 0x01=Playing.</p> <p>D6:D4: Undefined.</p> <p>D3:D2 Total video length, 0x0000-0xFFFF, in seconds.</p> <p>D1:D0 Current playing video position, 0x000-0xFFFF in seconds.</p> |

Set the project display curves using 8 channels, i.e. 0x1000-0x4FFF address is occupied, here the address is used arbitrarily from 0x5000, the video duration is 10S, the instructions are as follows .

### ( 1)Play

Start playback can be written to the address 0x5A03 , from the beginning of the second or the specified time to start playback, the instruction examples are as follows .

- ① Default playback from the 0th second: 5A A5 05 82 5000 5A 03
- ② Specify playback from the 0th second: 5A A5 05 82 5000 5A 03 0000
- ③ Specify playback from the 3rd second: 5A A5 05 82 5000 5A 03 0003
- ④ Specify playback from the 10th second: 5A A5 05 82 5000 5A 03 000A

### (2) Pause/re-play

Whether to send a instruction or touch, the first time is to play, the second time is to pause, and so on:

5A A5 05 82 5000 5A 02

### (3) Stop

When stopping, the screen stays at the first frame.

5A A5 05 82 5000 5A 01



#### (4) Read playback status

High byte of VP+2 address, current playback status feedback, 0x00=stop, 0x01=play.

##### ① Playing

Tx send: 5A A5 04 83 5002 01; status is located in the high byte of VP+2 address, read 1 word length data to determine the high byte.

Rx return: 5A A5 06 83 5002 01 0100; the high byte of VP+2 address is 0x01, which means playing.

##### ② Stop status

Tx send: 5A A5 04 83 5002 01; status is located in VP+2 address high byte, read 1 word length data to judge high byte.

Rx return: 5A A5 06 83 5002 01 0000; the high byte of VP+2 address is 0x00, indicating the stop state.

#### (5) Total video length

VP+ 4 address, total video length, 0x0000-0xFFFF, in second.

Tx send: 5A A5 04 83 5004 01; attribute is located at VP+4 address, read 1 word data.

Rx return: 5A A5 06 83 5004 01 0006; 0x0006 indicates the total video length of 6S, which can be used as the total playback length.

#### (6) The current playback video position

VP+ 5 address, the current playback video position, 0x0000-0xFFFF, in seconds

Tx send: 5A A5 04 83 5005 01; attribute is located at VP+5 address, read 1 word length data.

Rx return: 5A A5 06 83 5005 01 0001; 0x0001 indicates that the 1st second video is being played, which can be used as the current playback length position display.

Rx return: 5A A5 06 83 5005 01 0002; 0x0002 means in playing the 2nd second video, can be used as the current playback duration position display.

.....

Rx returns: 5A A5 06 83 5005 01 0006; 0x0006 means the 6th second of video is playing and can be used as the current playback position.

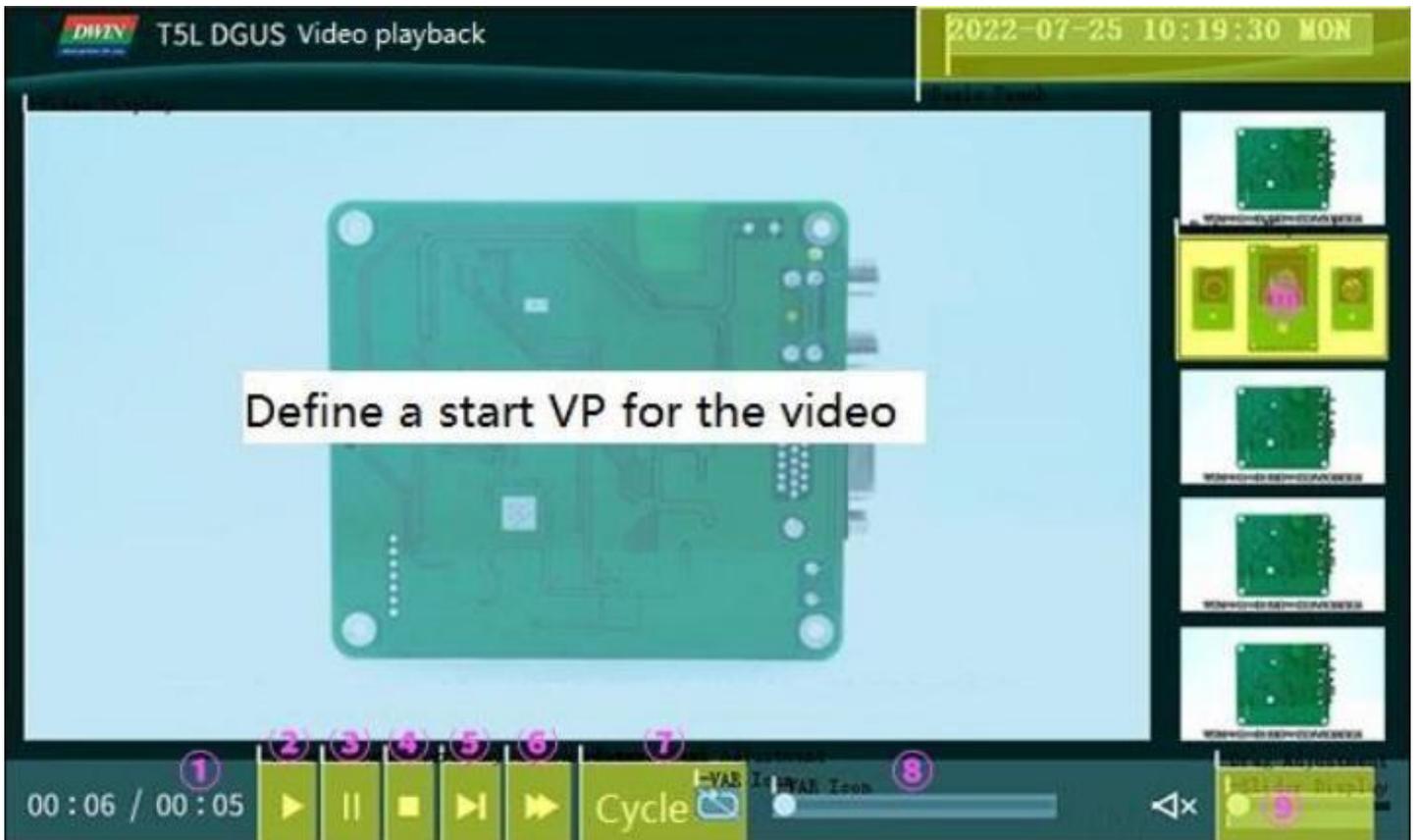
### 7.28.5 DEMO

Resolution : DEMO 800\*480, video 640\*360.

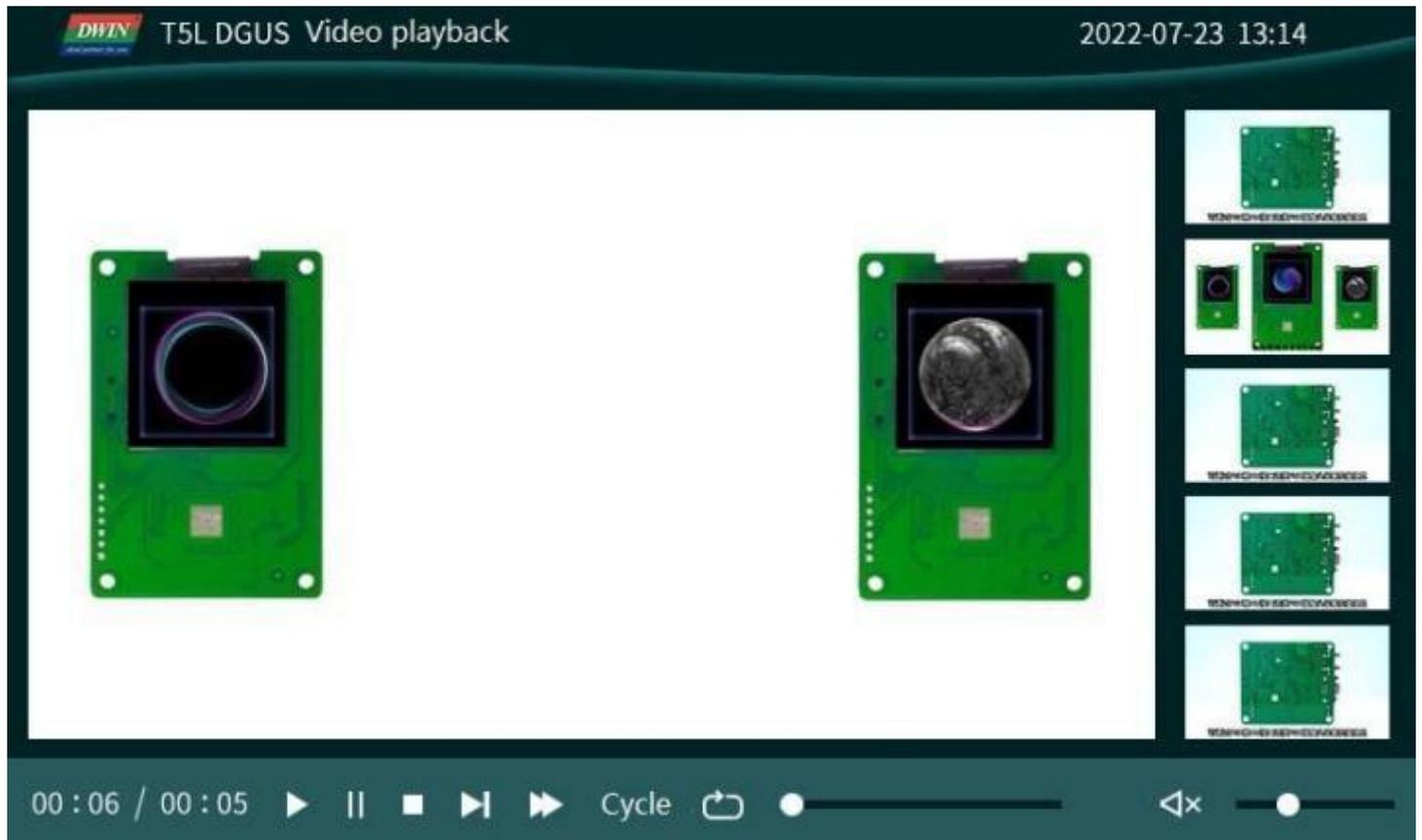
DEMO link: <http://inforum.dwin.com.cn:20080/forum.php?mod=viewthread&tid=6404>

DEMO function description.

- ① Use data variable to display, read VP+5 video position, and write to variable address to display.
  - ② Use return keycode, key value 0x5A03 to play.
  - ③ Use return keycode, key value 0x5A02 to pause.
  - ④ Use return keycode, key value 0x5A01 to stop.
  - ⑤ ⑥ Use synchrodata return to play from the specified position.
  - ⑦ Use incremental adjustment, cycle between lower limit 0 and upper limit 1, detect the variable address to upload key value 1, then refresh a play instruction at regular intervals, upload key value 0 to close the cycle play.
  - ⑧ use variable icon display, refresh icon position according to video duration, using 100 progress icons, or modify variable icon position with description pointer.
- For example, 100S video just corresponds to 100 icon state; 107S video corresponds to switching icon every 1.07 S.
- ⑨ use drag adjustment, system variable interface 0x00 A1 high byte to adjust the volume.
  - ⑩ For multiple page playback, use return keycode to switch page, set key value to 0x5A01 for pause or 0x5A02 for stop, to close the previous video and audio after page switching.



### 7.28.6 Effect



### 7.28.7 Video tutorial

Link: <http://inforum.dwin.com.cn:20080/forum.php?mod=viewthread&tid=6404>

You can also refer to the customer service to get the tutorial.

## Chapter 8 DGUS Basic Functions

### 8.1 Import

If you only have a DWIN\_SET folder, you can restore the project using the import function. Export is guide out of the project 13. bin, 14. bin pointer address EXCEL table.

Step 1: Open DGUS and create a project. The resolution should be consistent with the picture. Step 2: Add all the images in DWIN\_SET to the new project.

Step 3: Select the original DWIN\_SET folder, click import, you can see the project control has been restored.

Step 4: Import successful, the original DWIN\_SET inside the icon library, character library and other files are copied to the new project DWIN\_SET. You just need to adjust the control position.

### 8.2 Edit tool

User can easily align the touch and display controls by this function, making the controls neat, beautiful and orderly. Step 1: Selects the controls that need to be aligned;

Step 2: Click Edit tool-Up;

Step 3: Click the standard control, and then other controls will be aligned with it. Left-aligned, right-aligned, same size of controls can be set in the same way.

### 8.3 Front and Back

When the page sliding control and sliding adjustment control are superimposed, the former will be invalid if the former is under the latter. In this case, select the page sliding control and click Edit tool-front.

It is mostly used to add variable overlay and other functions to the icon.

### 8.4 Viewing the Usage of VP

In the process of development, it is necessary to check whether the VP or SP of the current project conflicts.

Click Display-Global check.

### 8.5 Preview

During the development, you can click Display\_Preview from the current page to preview from the current position to check whether the project is set correctly.

Double-click the black border of the preview to exit.



## 8.6 New Resolution

When creating a new project, manually input the required resolution ( W\* H) value.

## 8.7 Converting Resolution

For example, change the resolution of 720\* 720 to 480\*480. In order to avoid the conversion damage of the original project, please make a copy of the original project as a backup.

Step 1: Click Setting\_ Set resolution, select the desired resolution, and click OK; Step 2: Click File\_ Save As, select "Yes";

Step 3: Then click File\_ Save and Generate.

If the user copies controls of the larger resolution project to the smaller resolution project, you may find that the control disappears.

In this case, press Ctrl+ A to select all, and press " ↑" or " ←" on the keyboard . Then you will see the corresponding controls.

If icon or text display control are used, the size of them generally needs to be re- adjusted.

## 8.8 Replace T5 project with T5L project

If the resolution is the same, you can open the T5 DEMO by DGUS, make the background image 3 2 \_ background image. ICL file, and make the icon 48\_ Icon .ICL file.

That is, use 32 . ICL and 48 . ICL to replace the original . ICO and BMP format images. This part is detailed in chapter 3.

Since the DGUSI V5.04 13/ 14 files are not compatible with DGUS V7. 6 , the DGUSI screen cannot be modified

directly. If you need to use T5L, you need to redo the DEMO.

## Chapter 9 Revision Record

| Rev | Revise Date | Content  | Editor |
|-----|-------------|--|--------|
| 2.0 | 2022-04-12  | First Edition  | Rosy   |
| 2.1 | 2022-05-09  | Update 7. 18.4   | Rosy   |
| 2.2 | 2022-05-12  | Update 7.27.4  | Rosy   |
| 2.3 | 2022-05-20  | Update instructions when CRC check is on   | Rosy   |
| 2.4 | 2022-06-09  | Update 3.3 CFG configuration attention   | Rosy   |
| 2.5 | 2022-08-03  | <ol style="list-style-type: none"> <li>1. Update 6.7 Text input 0x00F9, 0x00FA description</li> <li>2. Update 5. 1 System variable interface 0xB0 description</li> <li>3. Update 2.3 Features 256KB data variable space</li> <li>4. Add 7. 1 Four-color bitmap display and video control function</li> <li>5. Update 7.22. 1 basic graphics control</li> <li>6. Add 7.28 Video control function</li> <li>7. Added the option of downloading files with encryption.</li> <li>8 . Update 0x00E2 system variable interface, add bitmap export brightness threshold setting.</li> <li>9. Update 6. 1 0x0D bit button function.</li> <li>10. Update 4. 1 UART2 0x84(curve buffer write instruction), 0x86, 0x87(double-word read/write mode) instruction(OS core code should at least be V2.0) .</li> <li>11. Add 4.2 Virtual UART.</li> <li>12. Add 4.3 MODBUS development.</li> <li>13. Optimize the algorithm of 4-wire resistive touch screen, the touch screen is more accurate (you need to re-calibrate the touch screen once after upgrading).</li> <li>14. Add 6. 13 Bit button.</li> <li>15. 6.7.2. 1 Remove the note input method</li> <li>16. 7.5.3 Remove FSK camera subsection</li> <li>17. Update 7. 10 Icon overlay</li> <li>18. Update 7.9 Icon Page Tran</li> <li>19. Update 7.8 Bit icon</li> <li>20. Add 5. 1 FSK bus interface</li> <li>21. Add 3.3.1 GTF format and JPEG file description</li> <li>22. Add 3.3.2.5 0x21 0x8* = ILI231*/251* etc. ILI driver IC capacitive touch screen.</li> <li>23. Add effect image for touch controls.</li> <li>24. Add 5. 1 0x100 address description; 0x08 description.</li> </ol> | Rosy   |

|            |                   |   |                |
|------------|-------------------|---|----------------|
| <p>2.6</p> | <p>2023-03-27</p> | <ol style="list-style-type: none"> <li>1. Update LCD screen TCON configuration or initialization procedure.</li> <li>2. CFG 0x70-0x71 position, backlight power-on delay</li> <li>3. CFG 0x06.0 position, whether OS core is running when the SD interface is downloaded</li> <li>4. Resistive touch screen sensitivity (.CFG file location 0x20-0x22 configuration)</li> <li>5. Variable Data Input adds support for returning floating point numbers in single precision format</li> <li>6. 0x21.11 Drawing_Ellipse arc display and 0x23 Process Bar function.</li> <li>7. 0x380 System configuration - Curve display</li> <li>8. 0x100 System configuration - FSK Bus interface</li> </ol> | <p>Allison</p> |
| <p>2.7</p> | <p>2023-06-02</p> | <ol style="list-style-type: none"> <li>1. Update QR code foreground colour and background colour change</li> <li>2. Example of icon page tran instruction</li> <li>3. Real-time curve data point refresh, displayed as a point</li> </ol>   | <p>Allison</p> |

## Appendix1:FAQs of T5L\_DGUSII Screen

1. Only one picture is displayed as a black screen.

A: For the T5L1 platform, the size of a single JPG image in the ICL format file package should not exceed 256 Kbytes. On the T5L2 platform, a single JPG image cannot exceed 768 Kbytes. Due to the inconsistent display complexity of images such as image gradients, especially with higher resolutions, the final compressed .JPG size may exceed 256KB, and users need to slightly lower the JPG image quality percentage. You can use the latest PC version tool software to convert, there will be a prompt when the size exceeds .

If you have any questions during the use of this document or DWIN products, or want to know more about the latest information of DWIN products, please contact us.

Hotline: 400 018 9008

Corporate QQ and WeChat: 400 018 9008

DWIN website: <https://www.dwin-global.com/>

Thank you for your continuous support to DWIN, your support is the driving force for our progress!