



Professional, Creditable, Successful



DWIN's High Quality HMI, The Best Solution for Industrial Field Applications!

DGUS Command Demonstration

Revision on May,2013

Beijing DWIN Technology Co., Ltd.

Content

1 Data Frame	2
2 Command Set.....	2
3 VP & SP	3
3.1 VP (Variable Pointer)	3
3.1.1 About VP.....	3
3.1.2 How to use VP?	4
3.2 SP (Stack Pointer)	6
3.2.1 About SP	6
3.2.2 How to use SP?	7
4 Examples	9
4.1 Access Register of DGUS.....	9
4.1.1 Write Data into Register	9
4.1.2 Read Data from Register.....	10
4.1.3 Response from the DGUS module	10
4.2 Access Variable SRAM.....	11
4.2.1 Write Data into variable SRAM.....	11
4.2.2 Read Data from variable SRAM	11
4.2.3 Response from the DGUS module	11
4.3 Dynamic Trend Curve Display	12
4.4 Basic Graphic Display	13

1 Data Frame

Data	1	2	3	4	5
Definition	Frame Header	Data Length	Command	Data	CRC checksum of the command and data
Data Length	2	1	1	N	2
Description	Defined by R3 & RA in CONFIG.TXT	Data length, include command, data and checksum	0x80-0x84		Defined by R2 in CONFIG.TXT

Data frame is made up by 4 parts, shown as below.

2 Command Set

Function	CMD	Data	Description
Access Register	0x80	ADR(0x00-0xFF)+Data_Pack	Write data into the specified register
	0x81	ADR(0x00-0xFF)+RD_LEN(0x00-0xFF)	Read data from the specified register
		ADR(0x00-0xFF)+RD_LEN+Data_Pack	Response from the DGUS module
Access Variable SRAM	0x82	ADR_H:L(0x0000-0x6FFF)+Data0...DataN	Write data into the specified variable SRAM
	0x83	ADR_H:L(0x0000-0x6FFF)+RD_LEN(0x00-0x7F)	Read data from the specified variable SRAM
		ADR_H:L+RD_LEN+Data0...DataN	Response from the DGUS module
Trend Curve Buffer	0x84	CH_Mode(Byte)+Data0(Word)+...+DataN	Write data into trend curve buffer.

DGUS Register Space: 0x00H-0xFFH, is written / read by byte.

DGUS Variable Data Memory Space: 0x0000H-0x6FFFH, is written / read by word.

Data in Curve buffer is written / read by word.

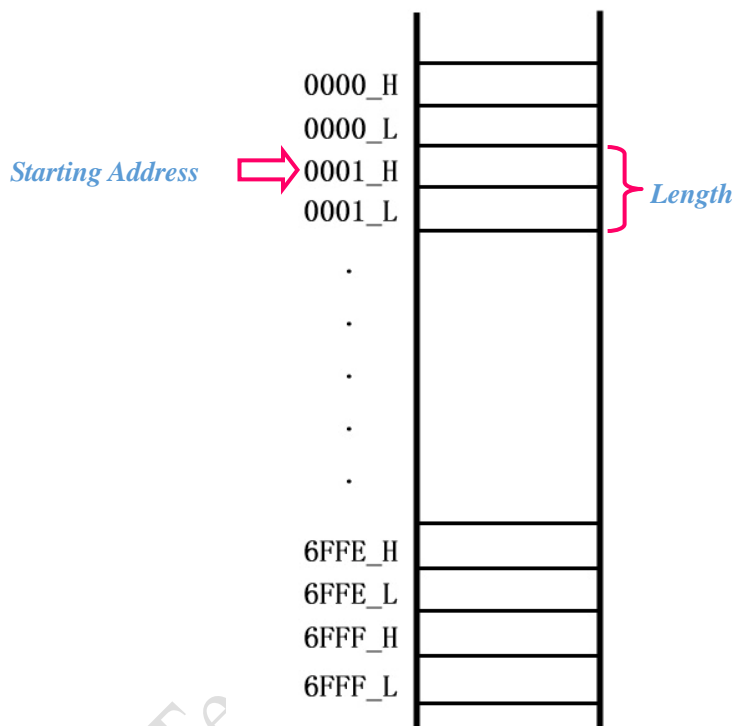
The communication between DGUS LCMs & Controllers (MCU) are driven by Variables that you may read and write in corresponding address.

3 VP & SP

3.1 VP (Variable Pointer)

3.1.1 About VP

VP is the initial address of variables in variable SRAM (56KB) which separated into 28672 pcs of memory spaces from 0x0000 to 0x6FFF occupying two bytes for each below. (High bytes and Low bytes)

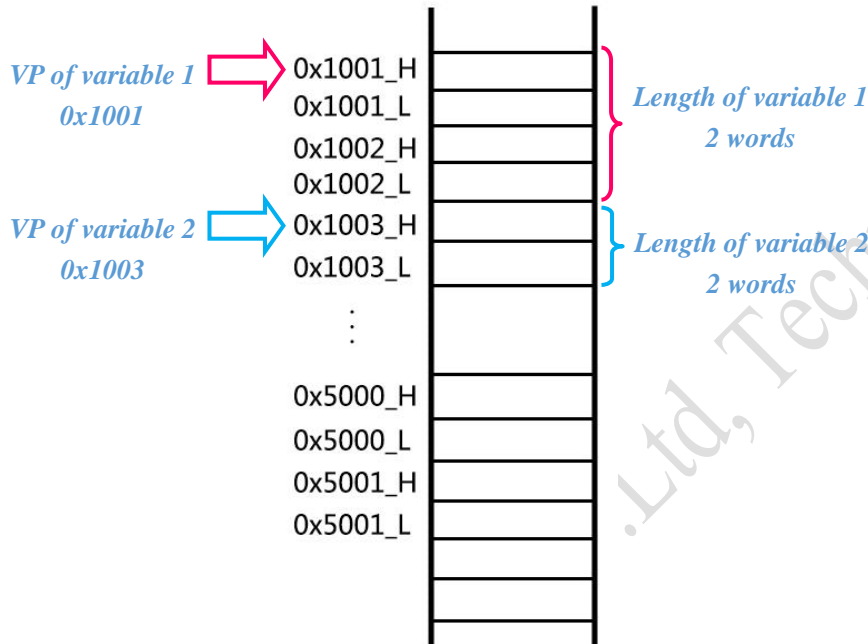


The communication between DGUS LCMs and host is oriented by variables that you may read or write in corresponding address. Reading & Writing of initial address and variable length could be changed via Command 0x82&0X83 if known in advance.

3.1.2 How to use VP?

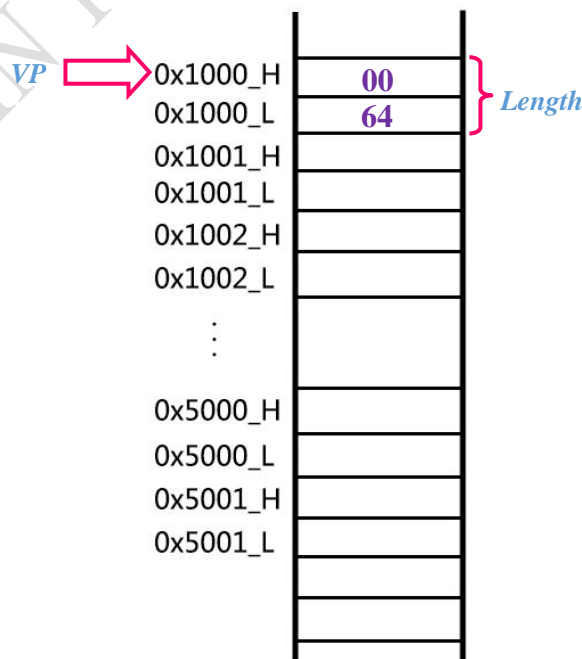
➤ **Assign VP for each variables**

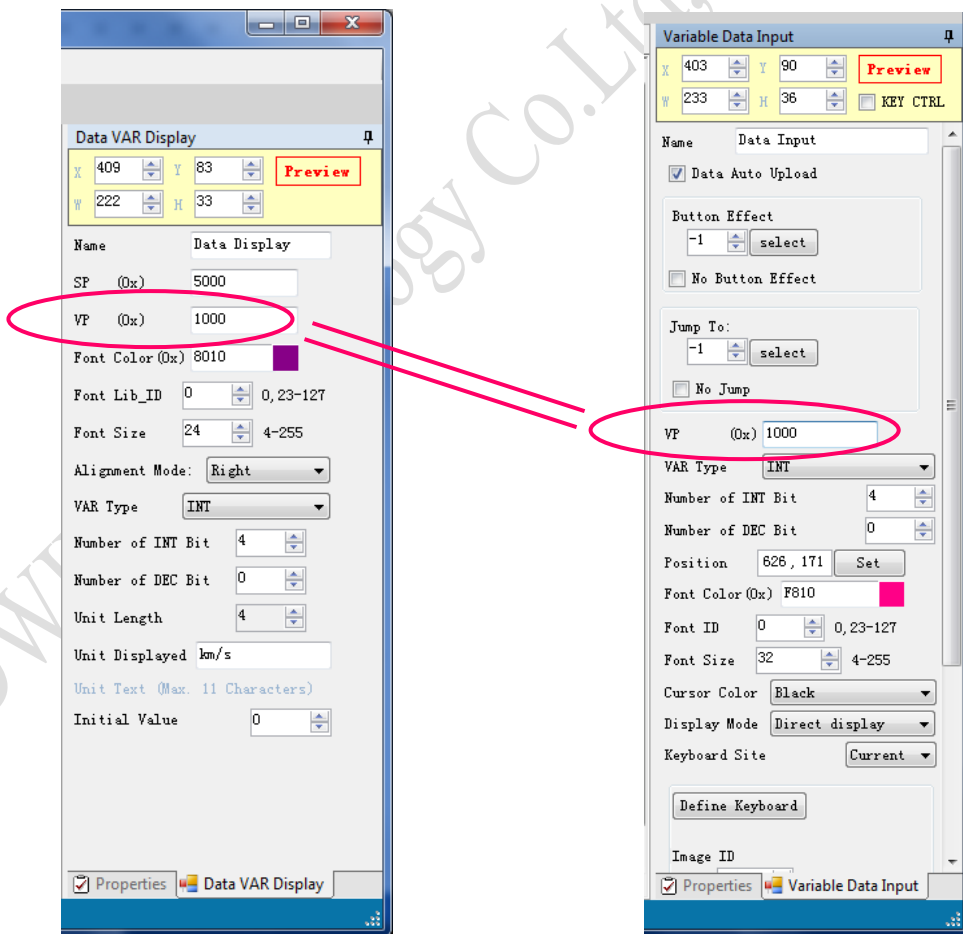
Each variable should be assigned a VP, yet overlap is not allowed. The value of the variable will be saved from the VP. For example, variable 1 as a long integer saved in 0x1001 while variable 2 as an integer have to be stored on another address preventing 0x1001 and 0x1002 from conflicts with variable 1.



➤ **Set VP via DGUS_SDK**

For example, display and input a variable (No. 100) which is an integer. The VP of this variable is 0x1000.





Supposing that reading & writing to the same one, touch and input parameters of VP should be the same accordingly.

3.2 SP (Stack Pointer)

3.2.1 About SP

SP is the initial address for saving variable attributes which described features of the variables, such as font color, font size and unit of variables, etc. The variable attributes refers to the Chapter Five of DGUS Dev. Guide.

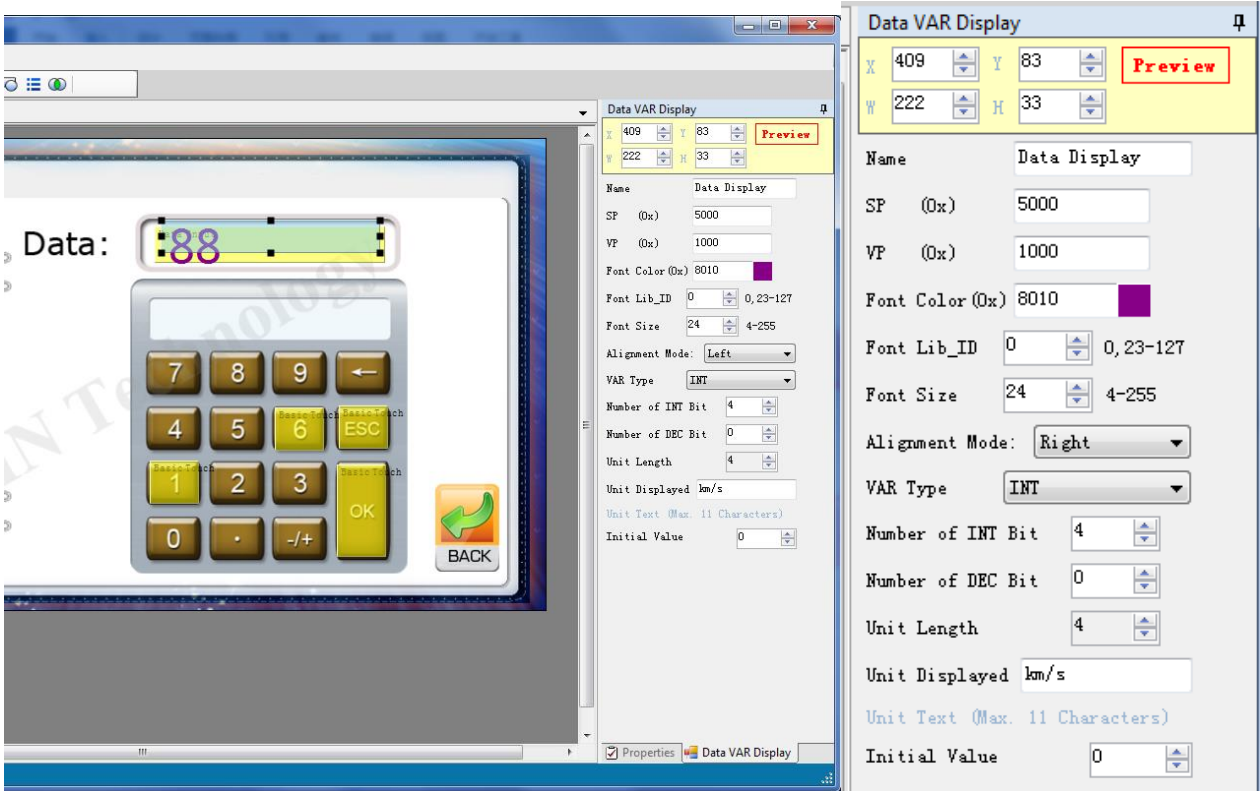
Address	Definition	Data Length	Description
0x00	0x5A10	2	
0x02	*SP	2	Stack pointer, default setting is 0xFFFF (set by Config. file).
0x04	0x000D	2	The whole process length (in terms of words).
0x06	0x00 *VP	2	Variable pointer.
0x08	0x01 X, Y	4	Top-left coordinate of text string.
0x0C	0x03 COLOR	2	Text color.
0x0E	0x04:H Lib_ID	1	Address of font file.
0x0F	0x04:L Font_X_Dots	1	Horizontal pixel numbers.
0x10	0x05:H ALI	1	0x00: right-aligned, 0x01: left-aligned, 0x02: centered.
0x11	0x05:L Int_Num	1	Length of integer digits.
0x12	0x06:H Dec_Num	1	Length of decimal digits.
			The sum should be less than 20.
0x13	0x06:L VP_Data_Mode	1	VP mode. 0x00: integer (2 bytes). -32768 – 32767 0x01: long integer (4 bytes). -2147483648 – 2147483647 0x02: high byte in VP address. 0 – 255 0x03: low byte in VP address. 0 – 255 0x04: double long integer (8 bytes). -9223372036854775808 – 9223372036854775807 0x05: unsigned integer (2 bytes). 0 – 65535 0x06: unsigned long integer (4 bytes). 0 – 4294967295
0x14	0x07:H Len_unit	1	Length of unit. 0x00: without unit.
0x15	0x07:L String_Unit	Max11	Unit data, by ASCII code.

SP shares the same SRAM with VP from 0x0000 to 0x6FFF. It is only used for variable display but overlap exception.

3.2.2 How to use SP?

➤ *Set SP via DGUS_SDK*

Take "Data Display" as example:



	0x4FFF_L		
<i>SP</i> <i>0x5000</i>	0x5000_H	10	} <i>VP</i>
	0x5000_L	00	
	0x5001_H	01	} <i>X, Y</i>
	0x5001_L	99	
	0x5002_H	00	} <i>Color</i>
	0x5002_L	53	
	0x5003_H	80	} <i>Lib_ID</i>
	0x5003_L	10	
	0x5004_H	00	} <i>Font_X_Dots</i>
	0x5004_L	18	
	0x5005_H	01	} <i>Alignment Mode</i>
	0x5005_L	04	
	0x5006_H	02	} .
	0x5006_L	00	

➤ **Change variable attributes via SP**

Same sample as above:

In operation, if user attempts to do the changes on data color to red, just need able to write down a new value into corresponding address.

5.2.1 Data Variable (0x10)

Address	Definition	Data Length	Description
0x00	0x5A10	2	
0x02	*SP	2	Stack pointer, default setting is 0xFFFF (set by Config. file).
0x04	0x000D	2	The whole process length (in terms of words).
0x06	0x00 *VP	2	Variable pointer.
0x08	0x01 X, Y	4	Top-left coordinate of text string.
0x0C	0x03 COLOR	1	Text color.
0x0E	0x04:H Lib_ID	1	Address of font file.
0x0F	0x04:L Font_X_Dots	1	Horizontal pixel numbers.
0x10	0x05:H ALI	1	0x00: right-aligned, 0x01: left-aligned, 0x02: centered.
0x11	0x05:L Int_Num	1	Length of integer digits.
0x12	0x06:H Dec_Num	1	Length of decimal digits.
0x13	0x06:L VP_Data_Mode	1	VP mode. 0x00: integer (2 bytes). -32768 – 32767 0x01: long integer (4 bytes). -2147483648 – 2147483647 0x02: high byte in VP address. 0 – 255 0x03: low byte in VP address. 0 – 255 0x04: double long integer (8 bytes). -9223372036854775808 – 9223372036854775807 0x05: unsigned integer (2 bytes). 0 – 65535 0x06: unsigned long integer (4 bytes). 0 – 4294967295
0x14	0x07:H Len_unit	1	Length of unit. 0x00: without unit.
0x15	0x07:L String_Unit	Max11	Unit data, by ASCII code.

SP=0x5000

The color is saved in 0x5003

Frame header (2 Bytes)+Data length (1 Byte)+Command (1 Byte)+Data (N Byte):
 ADR+data/LEN)+CRC (2 Bytes, optional)

Send: 5A A5 05 82 50 03 F8 00

Description: 5003: the address of color

F800: the value of red color

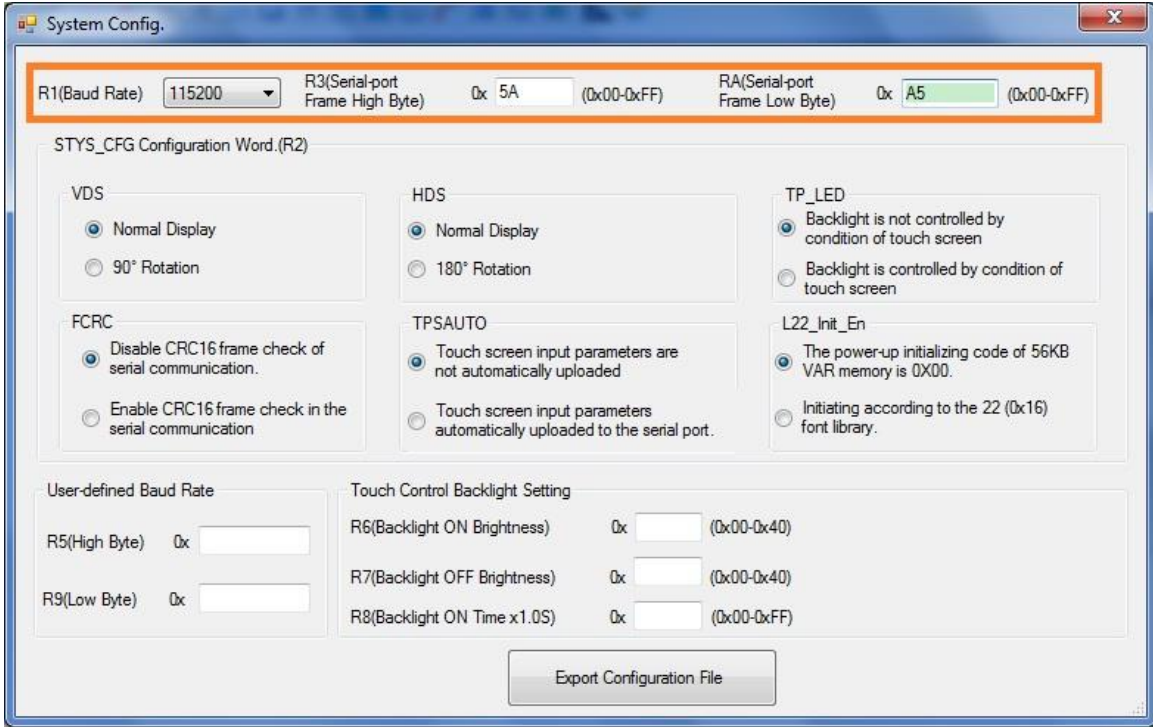
4 Examples

Configure frame header and baud rate in CONFIG.TXT via DGUS SDK as below:

R1=07

R3=5A

RA=A5

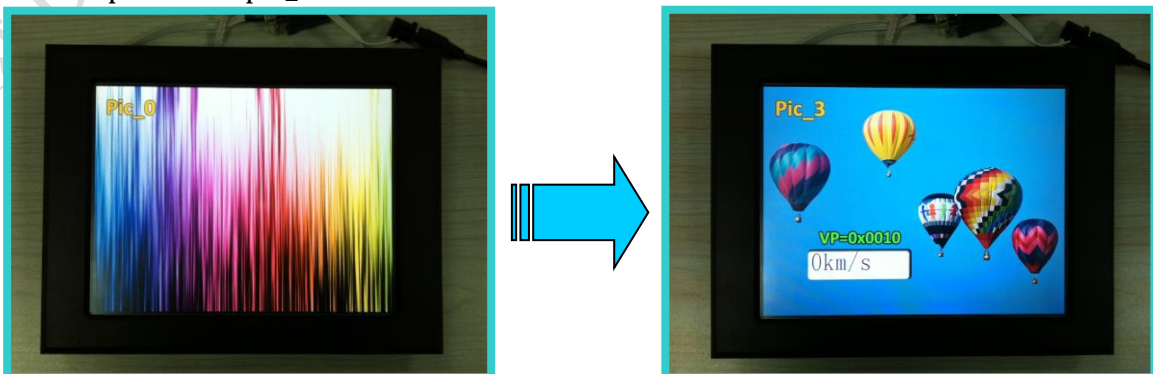


Frame header (2 Bytes)+Data length (1 Byte)+Command (1 Byte)+Data (N Byte: ADDR+data/LEN)+CRC (2 Bytes)

4.1 Access Register of DGUS

4.1.1 Write Data into Register

Switch current picture to pic_3:



> Register

A 256B register is designed for hardware setting and process control. Refer to the table below:

Register Address	Definition	Length (Byte)	Description
0x00	Version	1	DGUS version number, BCD code, 0x10 indicates V1.0.
0x01	LED_NOW	1	LED brightness, 0x00-0x40.
0x02	BZ_TIME	1	Buzzer beeping time, by every 10ms.
0x03	PIC_ID	2	Read: read current picture ID. Write: jump to appointed picture ID. 0x5A: there is update of touching coordinates.
0x05	TP_Flag	1	Others= no updating. Touchpanel data is no longer updated if user did not clear the flag after data reading. 0x01: first click. 0x03: pressing down.

Send: 5A A5 04 80 03 00 03

Description: 03: Register for Picture ID
00 03: Appoint picture ID

4.1.2 Read Data from Register

Handshaking: Read DGUS version (V 5.7):

> Register

A 256B register is designed for hardware setting and process control. Refer to the table below:

Register Address	Definition	Length (Byte)	Description
0x00	Version	1	DGUS version number, BCD code, 0x10 indicates V1.0.
0x01	LED_NOW	1	LED brightness, 0x00-0x40.
0x02	BZ_TIME	1	Buzzer beeping time, by every 10ms.
0x03	PIC_ID	2	Read: read current picture ID. Write: jump to appointed picture ID. 0x5A: there is update of touching coordinates. Others= no updating.

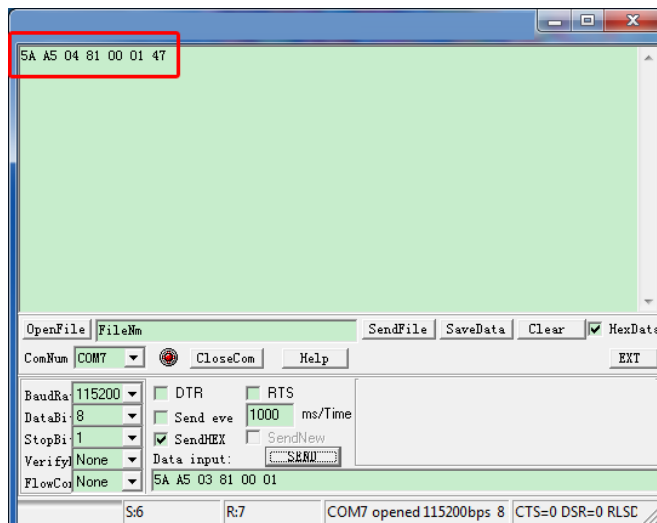
Send: 5A A5 03 81 00 01

Description: 00: Register for DGUS Version
01: Read data by a byte

4.1.3 Response from the DGUS module

Receive: 5A A5 04 81 00 01 47

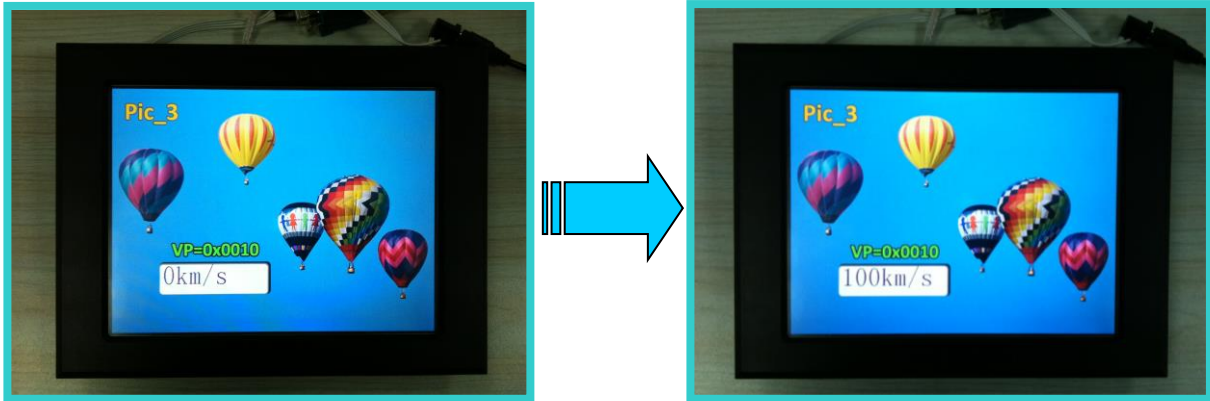
Description: 47: The version is 47 (V 4.7)



4.2 Access Variable SRAM

4.2.1 Write Data into variable SRAM

Write Data 100 to VP=0x0010



Send: 5A A5 05 82 00 10 00 64

Description: 0010: Variable pointer
0064: Data 100 in hexadecimal format

4.2.2 Read Data from variable SRAM

Send: 5A A5 04 83 00 10 01

Description: 0010: Variable pointer
01: Read data by a word

4.2.3 Response from the DGUS module

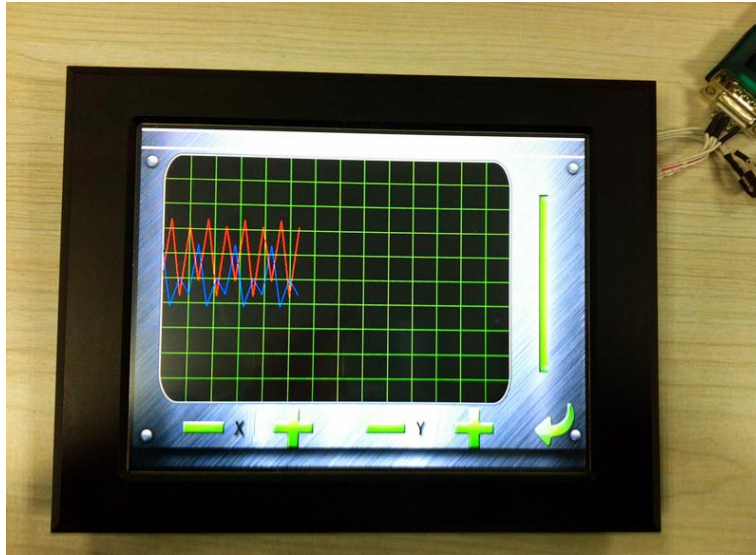
Receive: 5A A5 05 83 00 10 01 00 20

Description: 0020: Data 32 in hexadecimal format

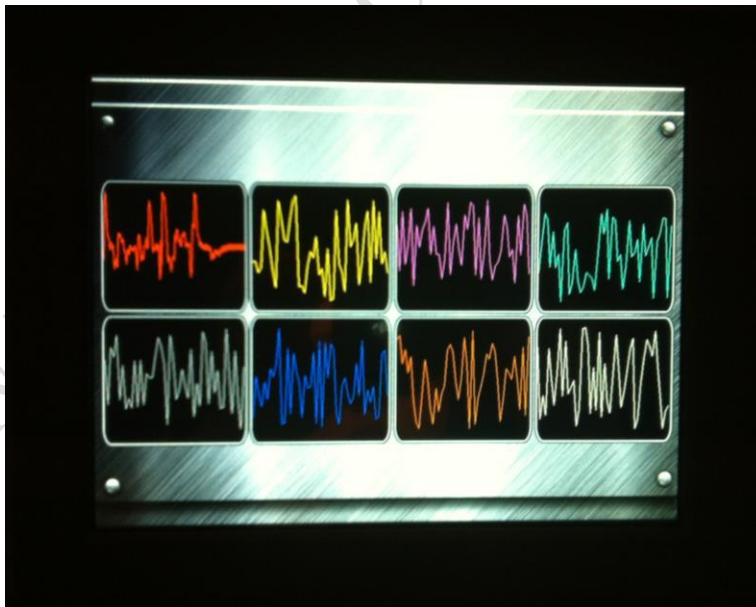
4.3 Dynamic Trend Curve Display

Send: 5A A5 12 84 12 00 32 00 72 00 9F 00 04 00 17 00 36 00 93 00 1A

Description: 12: Channel 4 & Channel 1, 8bit(0001 0010)



Display the trend curve in the same area



Display the trend curve in the different area simultaneously

4.4 Basic Graphic Display

String Format

Address	Definition	Description
VP	CMD	Command.
VP+1	Data_Pack_Num_Max	Max number of data packs. For command 0x0002, it's number of graph.
VP+2	DATA_Pack	

Date Pack For Basic Graphic

CMD	Function	Description of Data Format, by word			
		Relative Address	Data Length	Definition	Description
0x0001	Dot	0x00	2	(x, y)	Coordinate of dot.
		0x02	1	Color	Color of dot.
0x0002	Line	0x00	1	Color	Color of line.
		0x01	2	(x, y)0	Coordinate of vertex 0.
		0x03	2	(x, y)1	Coordinate of vertex 1.
		0x01+2*n	2	(x, y)n	Coordinate of vertex n.
0x0003	Rectangle	0x00	2	(x, y)s	Coordinate of top-left.
		0x02	2	(x, y)e	Coordinate of bottom-right.
		0x04	1	Color	Color of rectangle.
0x0004	Rectangle Area Fill	0x00	2	(x, y)s	Coordinate of top-left.
		0x02	2	(x, y)e	Coordinate of bottom-right.
		0x04	1	Color	Filled color.
0x0005	Circle	0x00	2	(x, y)	Coordinate of center of circle.
		0x02	1	Rad	Radius of circle.
		0x03	1	Color	Color of circle.
0x0006	Picture Cut/Paste	0x00	1	Pic_ID	Image ID of cutting area.
		0x01	2	(x, y)s	Coordinate of top-left of cutting area.
		0x03	2	(x, y)e	Coordinate of bottom-right of cutting area.
		0x05	2	(x, y)	Paste position on current screen.
0x**07	Icon Display	0x00	2	(x, y)	Coordinate of top-left of icon.
		0x02	1	ICON_ID	Icon ID in icon file, high byte of command specifies address of icon file, display mode is transparent.
0x0008	Area Fill	0x00	2	(x, y)	Coordinate of sampling dot.
		0x02	1	Color	Filled color.
0x0009	Vertical Line	0x00	1	Color0	Connect (X0, Y0s), (X0, Y0e) with color0.
		0x01	1	X0	
		0x02	1	Y0s	
		0x03	1	Y0e	

Status Flag:

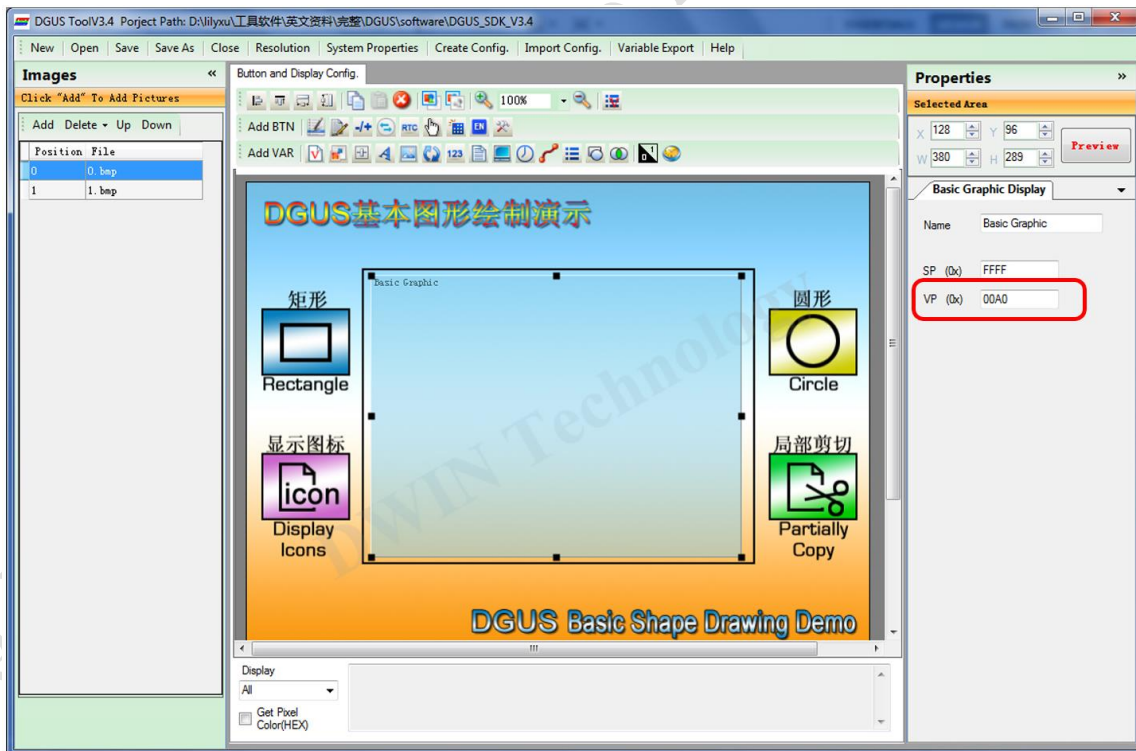
0xFF Current drawing operation finished.

0xFE This operation will be skipped (ignored).

Take Circle Drawing for example



1) Add a basic graphic display via DGUS SDK software and set vp to 0x00A0 as below:



2) Send commands by serial port

0x0005	Circle	0x00	2	(x, y)	Circle center coordinates.
		0x02	1	Rad	Radius of circle.
		0x03	1	Color	Circle color.

Send: 5A A5 11 82 00 A0 00 05 00 01 01 64 00 EF 00 64 F8 00 FF 00

Hex	Description
5A A5	Frame header
11	Data length
82	Command
00 A0	VP
00 05	Circle command
00 01	One circle
01 64	X coordinate of center of circle:356
00 EF	Y coordinate of center of circle:239
00 64	Radius of circle:100
F8 00	Color: red
FF 00	Drawing operation finished