



DWIN DGUS

USER MANUAL

v2.0

Revision at July,2012

Index

INTRODUCTION.....	- 2 -
1 Hardware Configuration.....	- 4 -
1.1 SD Card Configuration Interface	- 4 -
1.2 CONFIG.TXT Configuration File Description.....	- 5 -
1.3 Memory Space Description	- 6 -
1.4 Software upgrade procedure	- 8 -
1.5 Touch Screen Calibration.....	- 9 -
2 Serial Port Operations	- 10 -
2.1 Data Framework	- 10 -
2.2 Command Set.....	- 10 -
3 Data Format.....	- 11 -
4 Description of Touch Config. File	- 12 -
4.1 Variable data input (0x00).....	- 12 -
4.2 Select the Pop-up menu (0x01).....	- 14 -
4.3 Incremental Adjustment (0x02)	- 15 -
4.4 Slider Adjustment (0x03).....	- 16 -
4.5 The RTC Set (0x04).....	- 17 -
4.6 Key value for return(0x05)	- 18 -
4.7 ASCII Text entry(0x06)	- 18 -
4.8 Hardware parameter configuration(0x07).....	- 20 -
5 Variable Configuration File Descriptions	- 21 -
5.1 Icon variable	- 21 -
5.1.1 Variable icon to display (0x00)	- 21 -
5.1.2 Animated icons show(0x01)	- 22 -
5.1.3 Slider scale indicator(0x02)	- 23 -
5.1.4 WordArt variable display(0x03)	- 24 -
5.1.5 Picture Animation(0x04).....	- 25 -
5.1.6 ICON Rotation Instructions(0x05).....	- 26 -
5.2 Text Variables.....	- 27 -
5.2.1 Data variables show(0x10)	- 27 -
5.2.2Text Display (0x11)	- 28 -
5.2.3 The RTC Display(0x12).....	- 29 -
5.3 Graphics variable	- 30 -
5.3.1 Curve Display (0x20).....	- 30 -
5.3.2 Basic graphical display(0x21).....	- 31 -
5.3.3 Form Display(0x22).....	- 36 -
5.3.4 Exclusive functions for special applications	- 37 -
6 Development Steps	- 38 -

Appendix 1 DGUS key functions preview.....	- 42 -
Appendix 2 Application Examples	- 44 -
(1) Display, entry and adjustment of parameters	- 44 -
(2)Display and adjust the curve	- 45 -
(3)Time display and set.....	- 46 -
(4)Progress bar display	- 47 -
(5)Control switch, display animation of operating status	- 48 -
(6)Dashboard display.....	- 49 -
(7)Display and print a form	- 50 -
(8)Text display and input	- 51 -
(9)Electronic Albums (boot animation)	- 52 -
(10) Multi-channel switch display	- 53 -

INTRODUCTION

DGUS V2.0 is brand-new GUI software from **DWIN Technology** based on DWIN K600+ products, which was developed in the way of configuration software under new beginning pattern, via exchanging data between memorizer port and users to make ZERO CODE for HMI development coming true.

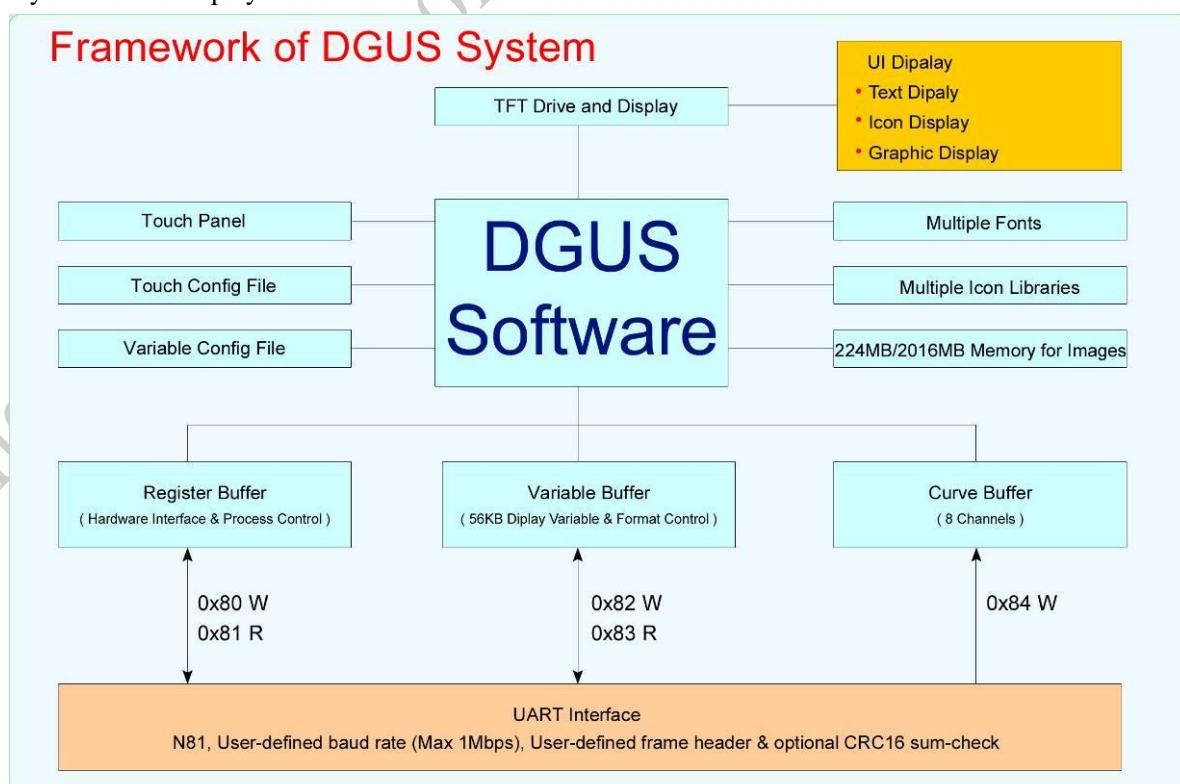
DGUS NEW FEATURES

DGUS configuration development mode is different from the traditional development method of DWIN HMI with following new features:

- a. **Without** code programming for realizing GUI effects, users may read and rewrite the code that is stored in the variable memory by serial ports, which greatly reduces the code amount of users' microcontroller, and the touching logics as well as displayed contents are defined by configuration files.
- b. Config.File can be rapidly integrated and developed by DWIN PC Software.
- c. Graphic database is defined by user themselves (256 MB Memory).
- d. Variable graphic layers were made by user-defined priority.
- e. User can modify the display format in operation process.
- f. Parameter input of the touch screen is performed by DWIN display.

Flexible design of DGUS

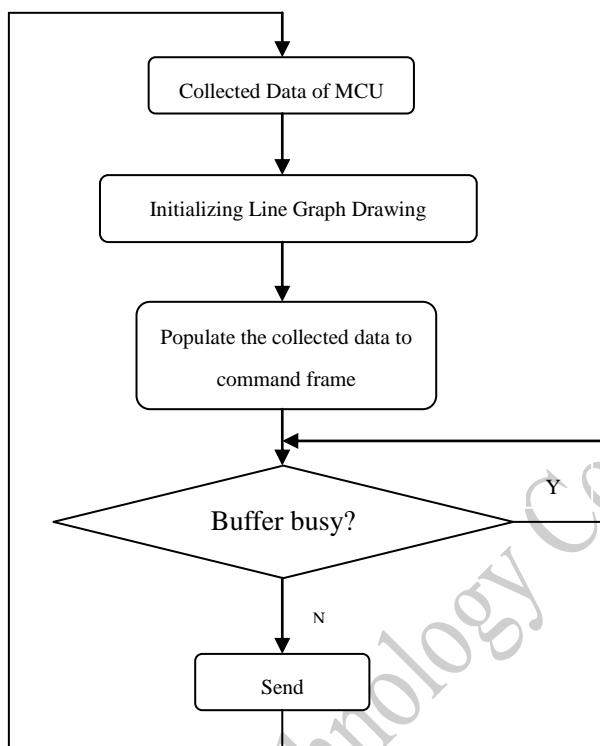
- a. It is only need that SD card have DWIN_SET folder which contained touch config. File, Image&Icon file for achieving interface effects without PC terminal assistant operation.
- b. DGUS v2.0 functions as converses the image format, supports animation effects and images. There are also fonts, Icon library, image library (224MB/2016MB) available for user selection.
- c. Extreme response speed of refresh time (≤ 200 ms) to meet users' highly real-time requirements; Commands won't overflow from the buffer, so that no necessity for users to judge the condition of serial port.
- d. Highly abstract variables (64 variables/page), the number of touch buttons has no limits.
- e. DGUS have function of achieving RTC display, up/down and zoom-in/out of curve as well as icon/images recycles automatic play with no involvement of MCU code.



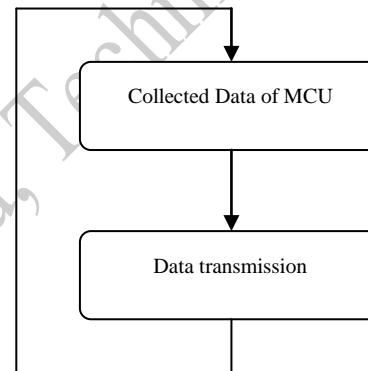
Also, DGUS v2.0 designed for slider operation, and synchronic cutting-effects of keyboard, as well as touch control can be achieved without complex programming and MCU code writing. (It has to achieve by repeated crossing operations of Command 0x71 and Command 0x98 in older Terminal Assistant operation). It helps a lot in R&D cycle simplifying and making mass productions quick comes into real application.

Take the oscilloscope development as example: if use DGUS v2.0 to develop the oscilloscope, the only thing that user needs to do is to transmit A/D sampling data to DWIN display through the serial port. The following is a comparison of the DWIN Command Set v2.4 and DGUS v2.0 operation:

DRWING PROCESS of DWIN Command Set v2.4



DRAWING PROCESS OF DGUS v2.0



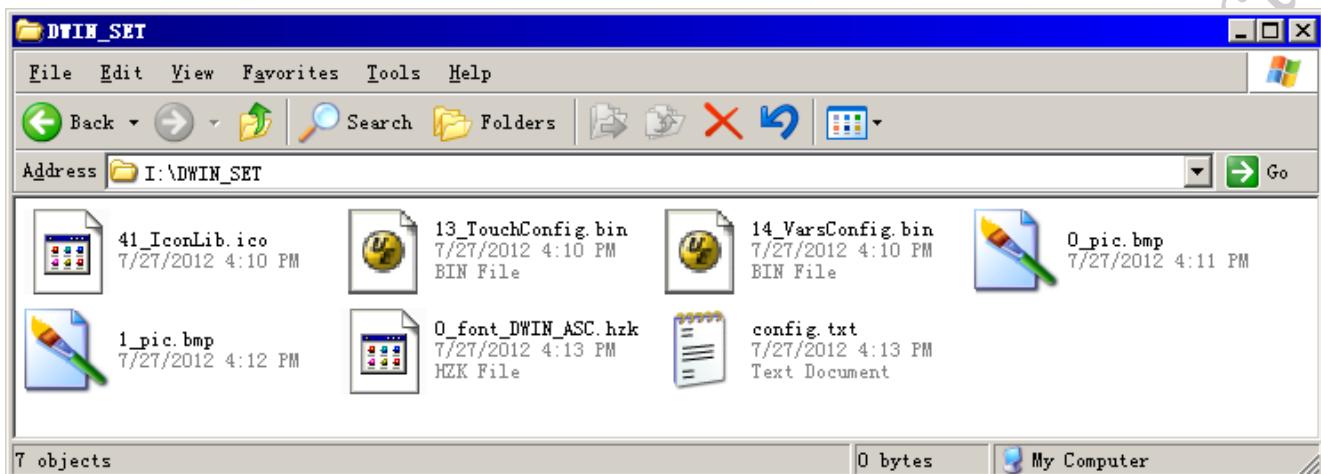
1 Hardware Configuration

1.1 SD Card Configuration Interface

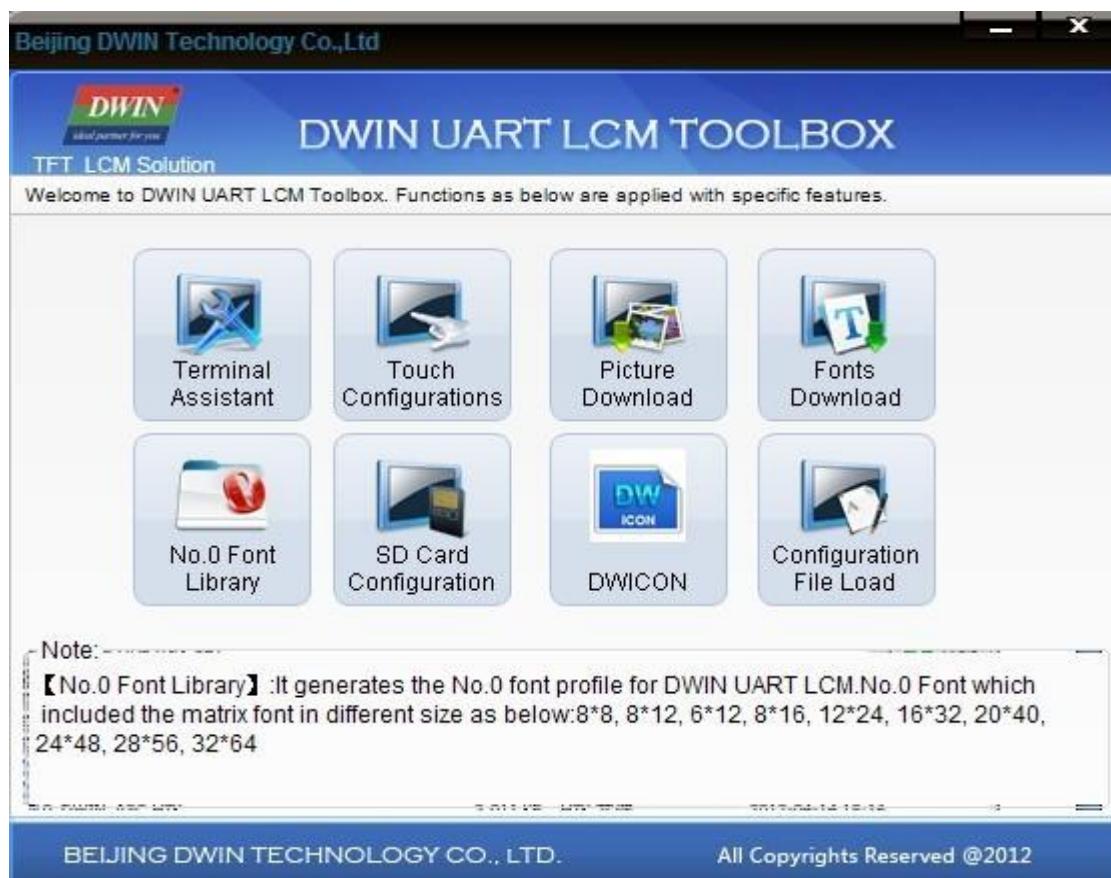
All parameter settings and data download for DWIN DGUSV2.0 are only finished via the SD card interface which supports SD or SDHC card format with FAT32 file system required.

File Format Description:

- Establish a folder named DWIN_SET in the SD card root directory;
- Images, fonts and configuration files are placed in the DWIN_SET folder as shown below



SD Card File Format Specification			
File Type	Naming rules	Example	Description
Picture file	Image storage position+(optional) file name.BMP	00_starting page.BMP	Must be identical to the DWIN module resolution that 24-bit color BMP files.
Font file	Font storage position+(optional)file name.BIN/HZK/DZK	32_GBK12 Chinese. DZK	Be created by the Font Generator
Icon library	Storage position+ (optional) file name. ICO	41_iconlibrary. ICO	Generated by DWIN Toolbox "DWICON"
Specified character font	0*.HZK	0_DWIN_ASC.HZK	Generated by DWIN Toolbox "No.0 font library"
Input method font	12*.BIN	12_GBKPY.BIN	DWIN pre-installed
Touch configuration	13*.BIN	13_touch configuration file.BIN	Created by DWIN DGUS software
Variable configuration	14*.BIN	14_variables configuration file. BIN	Created by DWIN DGUS software
Hardware settings	CONFIG.TXT	CONFIG.TXT	



1.2 CONFIG.TXT Configuration File Description

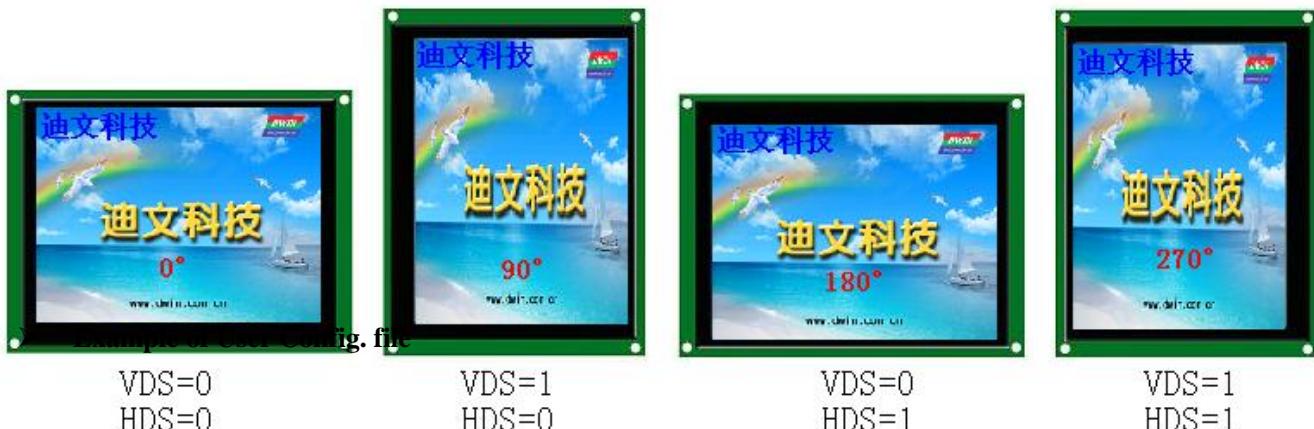
CONFIG.TXT file introduce sort of scripting language to describe the parameter registers that one parameter described by each line. Meanwhile, unused parameters are not necessary to write. See below instructions:

Name of parameter register	Range	Description																																								
R0	Depends on the DWIN screen	DWIN Module Driver Mode, no need for user to set.																																								
R1	0x00-0x11	Baud rate setting, 0x00-0x10 matchup with 1200bps—921600bps <table border="1" style="margin-left: auto; margin-right: auto;"> <tr><td>R1</td><td>0x00</td><td>0x01</td><td>0x02</td><td>0x03</td><td>0x04</td><td>0x05</td><td>0x06</td><td>0x07</td><td>0x08</td></tr> <tr><td>Baudrate</td><td>1.2K</td><td>2.4K</td><td>4.8K</td><td>9.6K</td><td>19.2K</td><td>38.4K</td><td>57.6K</td><td>115.2K</td><td>28.8K</td></tr> <tr><td>R1</td><td>0x09</td><td>0x0A</td><td>0x0B</td><td>0x0C</td><td>0x0D</td><td>0x0E</td><td>0x0F</td><td>0x10</td><td>0x11</td></tr> <tr><td>Baudrate</td><td>76.8K</td><td>62.5K</td><td>125K</td><td>250K</td><td>230.4K</td><td>345.6K</td><td>691.2K</td><td>921.6K</td><td>Defined</td></tr> </table>	R1	0x00	0x01	0x02	0x03	0x04	0x05	0x06	0x07	0x08	Baudrate	1.2K	2.4K	4.8K	9.6K	19.2K	38.4K	57.6K	115.2K	28.8K	R1	0x09	0x0A	0x0B	0x0C	0x0D	0x0E	0x0F	0x10	0x11	Baudrate	76.8K	62.5K	125K	250K	230.4K	345.6K	691.2K	921.6K	Defined
R1	0x00	0x01	0x02	0x03	0x04	0x05	0x06	0x07	0x08																																	
Baudrate	1.2K	2.4K	4.8K	9.6K	19.2K	38.4K	57.6K	115.2K	28.8K																																	
R1	0x09	0x0A	0x0B	0x0C	0x0D	0x0E	0x0F	0x10	0x11																																	
Baudrate	76.8K	62.5K	125K	250K	230.4K	345.6K	691.2K	921.6K	Defined																																	
R2	0x00-0xFF	SYS_CFG configuration was defined by bit position. See further more as below.																																								
R3	0x00-0xFF	UART_SYNC_H High byte of the serial port header																																								
R4	Depends on the DWIN screen	DWIN Module Driver mode, no need for user to set.																																								
R5	In Flux	When R1 =0x11, high byte of baudrate configuration. R5:R9=625000/user-defined baudrate. E.g.:setting10000bpsbaud rate, R5:R9=6250000/10000=625=0x0271 R5=02 R9=71																																								
R6	0x00-0x40	The brightness that backlight on when touch the screen, if backlight control enabled																																								
R7	0x00-0x40	The brightness that backlight off when no touch for a while, if backlight control enabled																																								
R8	0x01-0xFF	Backlight lighting-up time with 1.0s unit, if backlight control enabled.																																								
R9	In Flux	When R1 =0x11, baudrate config low byte of word																																								
RA	0x00-0xFF	UART_SYNC_L Low byte of the serial port header																																								
Parameter of configuration file is one byte HEX-mode, for example:0A means 10 in hexadecimal number																																										
Parameter of configuration file must be two-bit, for example: 00 should not be written as 0.																																										

➤ R2(SYS_CFG configuration Byte)

Position	Ratio	Definition	Description
.7	0x80	VDS	0=Normal display 1=90 °Rotation
.6	0x40	HDS	0=Normal Display 1=180 °Rotation(anti-visual-angel)
.5	0x20	TP_LED	0=Backlight is not controlled by condition of touch screen 1=Backlight is controlled by condition of touch screen, set by R6, R7, R8 in config.txt.
.4	0x10	FCRC	0=Unable CRC16frame check in the serial communication 1=Enable CRC16frame check in the serial communication
.3	0x08	TPSAUTO	0=Touch screen input parameters are not automatically uploaded(user query); 1=Touch screen input parameters automatically uploaded to the serial port.
.2	0x04	L22_Init_En	0=56KB Variable Storage initiated by 0x00 1=56KB Variable Storage initiated by the values defined by 22 (0x16) font library file
.1	0x02	FreshTime_Sel	0=Refresh cycle of variable is 200mS; 1= Refresh cycle of variable is 250mS. Recommended to write 0.
.0	0x01	RnumMode_Sel	0=Normal mode; 1=High-speed mode (Power consumption will be higher than normal mode about 100mW). Recommended to write 0.

➤ VDS and HDS Control Panel Display Mode



*R1=07 ;Baudrate,07=115200bps
R2=20 ; SYS_CFG*

R6=40; brightness of backlight-on in the mode of automatic control,40 =100%brightness (the brightest selection)

R7=10 ; brightness of backlight-off in the mode of automatic control, 10 =25%brightness

R8=14 ;Light-up time, units: 0.5 seconds, 14=10 seconds

R3=A5 ; high-byte of frame header:0xA55A

RA=5A ; low-byte of frame ending:0xA55A

Note: Besides of modification via SD card, R0-RA Register can be modified by using Command 0XFE07.

1.3 Memory Space Description

➤ Font Library

DWIN DGUS v2.0 software uses 32MB FLASH Memory as a font memory which is divided by 128 units of 256KB space. The correspondence font ID is 0-127. See below explanation:

Font ID	size	Explanation	Note
0	3072KB	0#ASCII Font	0_DWIN_ASC.HZK
12	256KB	Pinyin input method font	12_PYK.BIN
13	256KB	Touch configuration file	13_touch.BIN
14	2048KB	Variable configuration files(up to 1024pages with max.64variables per page)	14_variable.BIN
22	256KB	Variable initializing file (56KB Default value of variable storage that loading depends on SYS_CFG.2)	22_variable initializing.BIN
23-127	26.3MB	Font, icon library. Icon 64-127 can be used as user database.	User defined

➤ Images Space

DWIN DGUS v2.0 software use 224MB (256MB K600+) or 2016MB (2GB K600+) to save the images, corresponding to different resolution, can save a number of pictures as the table below.

Screen Resolution	256MB K600+	2GB K600+
320×240	869	7807
480×272	869	7807
640×480	290	2602
800×480	290	2602
800×600	217	1952
1024×600	174	1561
1024×768	145	1301

➤ Register Space

Dwin DGUS v2.0 software provides one register interface (256 Bytes), used for processing hardware or image display etc. The register space is defined as follows:

Register Address	Definition	Byte length	Description
0x00	Version	1	DGUS version number, BCD code, 0x20 indicates V2.0
0x01	LED_NOW	1	LED brightness control register, 0x00-0x40
0x02	BZ_TIME	1	The buzzer control register: Unit is 10mS
0x03	PIC_ID	2	Reading: the current display page ID Writing: switch to a specified page(process control)
0x05	TP_Flag	1	0x5A=there are updating news for touch panel; Others= no updating. Touch panel data is no longer updated if user did not clear the flag after data retrieving
0x06	TP_Status	1	0x01=press in first time; 0x03=keep pressing; 0x02=uplift pressing; Others=invalid
0x07	TP_Position	4	Coordinate of touching position: X_H:Y_H:L
0x0B	TPC_Enable	1	0x00=Disable the Touch panel. Others=Touch panel is enabled.(The default is 0xFF)
0x0C-0x0F	Retained	4	Undefined
0x10-0x1A	R0-RA	11	Mapping of SD card config. register, only read for serial port, valid for writing
0x1F	RTC_COM_ADJ	1	0x5A=RTC data is rewritten through serial port. Cleared when RTC is modified by DGUS.
0x20	RTC_NOW	16	YY:MM:DD:WW:HH:MM:SS
0x30-0x3E	Retention	16	Undefined
0x40	En_Lib_OP	1	0x5A indicates user apply for doing operation in Font Storage, value will be cleared after the operation is done.
0x41	Lib_OP_Mode	1	0x50: Transfer the specified data in the variable storage space to specified font storage. 0xA0: Transfer the specified data in the font storage to variable storage.
0x42	Lib_ID	1	Specified font storage space, 0x40-0x7F, 128K Words each font library file, maximum capacity is 8M Words (16MB).
0x43	Lib_Address	3	Specified the first (Word) address for data operation in Font storage, 0x00:00:00-0x01:FF:FF
0x46	VP	2	Specified the first (Word) address for data operation in Variables storage, 0x00:00-0x6F:FF
0x48	OP_Length	2	Word Length of data operation, 0x00:01-0x6F:FF
0x4A-0xFF	Retention	182	Undefined

Register space accessed by Command 0x80/0x81

➤ Variable data memory space

DWIN DGUS v2.0 provides 28KW (56KB) SRAM as data memory for GUI variables data storage, and being as

expanded reference memory access when reading and writing real-time (mS level) were low required.

The data memory spaces can only by word to read and write, with the corresponding address space 0x0000-0x6FFF. The data memory space accessed by Command 0x82/0x83

➤ Curve data buffer

In order to simplify the display of real-time curve, DWIN DGUS v2.0 designed a curve data buffer that is used to cache the data of the user's curve, which will not occupy that data space while 8 units of curves are working at the same time.

Curve data buffer can only by word to read. Each data on the point of the curve is predicted by a 2-Byte unsigned integer.

Curve data buffer can be accessed by Command 0x84 (only write).

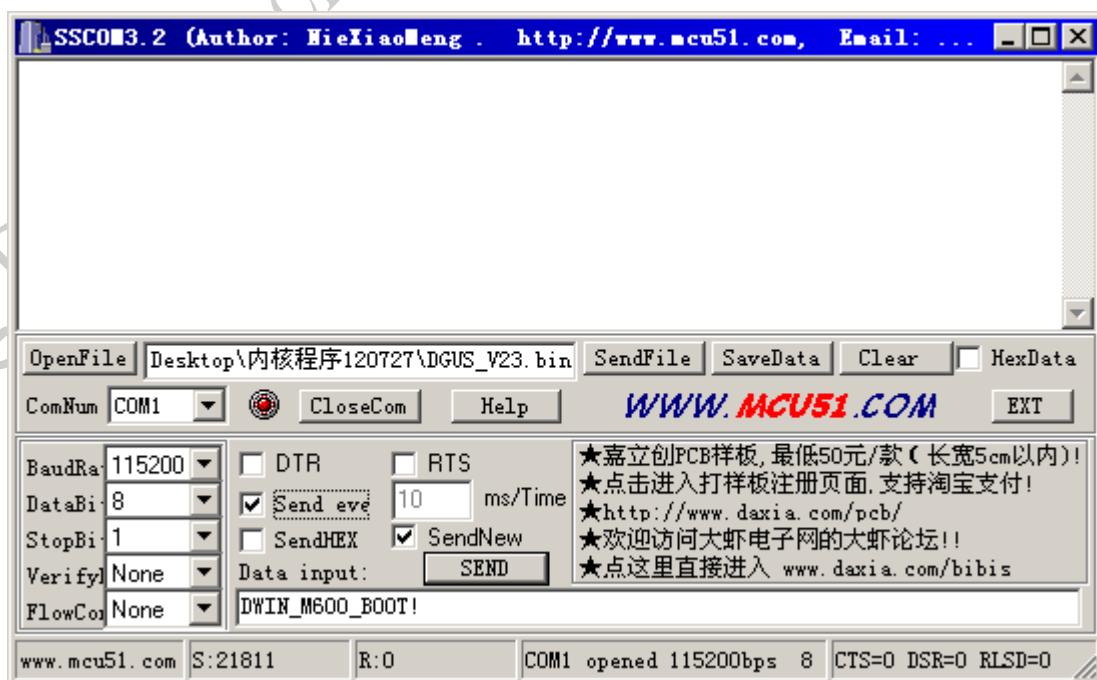
1.4 Software upgrade procedure

Tools:

- A DC regulated power supply that adjust the output voltage to the appropriate value;
- A serial port cable;
- A computer which has a hardware serial port and serial debugging assistant SSCOM3.2 software or **any similar software under different OS**.

The Steps instruction of Update:

- a. **Turn off the DWIN Module** and make sure serial port is connected to the computer's COM port.(eg:com1)
- b. Open the SSCOM3.2 software and click "Open File" to **select DGUS_V2.0.BIN**;
- c. Write "**DWIN_M600_BOOT!**" in data input column and set **send-eve** interval to 10ms/Time;
- d. Select "**Send New**" and "**Send-eve**", then power DWIN module again.
- e. The serial port will receive "Erase". If no response, please check the connection.
- f. It will receive "Please Tx File!" after one second, then hook off "Send New" and click "Send File".
- g. Wait another 3 to 10 seconds. If the serial port receives "*****END*****", it means the download is finished.
- h. Restart the module. Update finished.



Update your kernel software by using SD card is STRONGLY RECOMMENDED for version above V2.0 by saving the DGUS_V*.bin file in the DWIN_Set folder.

The Steps instruction of checking kernel software version:

- Physically, connect DGUS HMI serial port with COM port on PC, and connect power properly.
- Open the SSCOM3.2 software and make sure that "*SendHEX*" and "*HexData*" are checked.
- Then write command in data input column: "**FRAMEHEAD(2 BYTES) 03 81 00 01**" and press "SEND" button.
- Your computer will receive a feedback message like "**FRAMEHEAD(2 BYTES) 03 81 00 XX**". The number XX is the version number in BCD format. If it's 10, that means v1.0; if 23, means 2.3.



1.5 Touch Screen Calibration

When DWIN LCM is running with DGUS v2.0 working, it will enter the calibration mode when fast clicking screen over 20 times within 4 second. Here is the instruction:

- Quickly tap the touch screen more than 20 times in 4 seconds.
- Notice: do not click button area.
- Click until a long beep from the buzzer.
- Enter the calibration mode, click specify location of the touch screen that cross line indicates to calibrate the touch screen
- Calibration finished and back to starting page.

2 Serial Port Operations

All UART LCM standard products of DWIN technology are asynchronous, full-duplex serial port (UART) mode. Serial port mode is 8n1; i.e. each data transmission has 10bits: one initial bit, eight data bits (Least Significant Bit, LSB) and one stop bit.

The Serial port baudrate is configured by SD card. All serial commands or data are in hexadecimal (HEX) format; for word data (2 bytes), high byte will be adopted for transmission firstly. For example, 0x12 will be sent firstly when 0x1234 in transmission.

If module installed DGUS, BUSY Signal in previous version will be invalid, please leave it in floating.

2.1 Data Framework

The serial data frame is composed of four blocks of data as described in the following table

Data Block	1	2	3	4	5
Definitions	Frame Head	Data length	Command	Data	CRC checksum of the command and data
Data length	2	1	1	N	2
Description	CONFIG.TXT configuration filesR3: RA definition.	Data length, including command, data and checksum	0x80-0x84		CONFIG.TXT profileR2 decide whether to enable

2.2 Command Set

Function	Command	Data	Description
Access control register	0x80	ADR(0x00-0xFF)+Data_Pack	Specified address to write register data
	0x81	ADR(0x00-0xFF)+RD_LEN(0x00-0xFF)	Specified address to read RD_LEN bytes of register data
		ADR(0x00-0xFF)+RD_LEN+Data_Pack	Response of DWIN Display to read register
Access data memory	0x82	ADR_H:L(0x0000-0x6FFF)+DATA0...DATAn	Specified address to write data string(word data) in the variable storage area
	0x83	ADR_H:L(0x0000-0x6FFF)+RD_LEN(0x00-0x7F)	Read into RD_LEN length word data start from the specified address of the variable storage area
		ADR_H:L+RD_LEN+DATA0.....DATAn	Response of DWIN Display to read data storage
Write the curve buffer	0x84	CH_Mode(Byte)+DATA0(Word)+...+DATAn	Write the curve of the buffer data: CH_Mode defined channel of follow-up data order: <ul style="list-style-type: none"> ➤ Each bit of CH_Mode corresponds to one channel; e.g.: CH_Mode.0 corresponds channel 0, .7 corresponds channel7 ➤ The position1 indicates the presence of the corresponding channel data; ➤ Data of lower channel is arranged more forward; e.g.: CH_Mode = 0x83 (10000011B), indicates a follow-up data format :(channel0+channel 1+ channel7) +...+ (channel 0 +channel 1+ channel7).

Access to the register with Byte as data unit, while Word in Data Storage, Curve Buffer as data unit.

For further information, please reference Chapter 1.3.

3 Data Format

Due to the aim of the embedded system application, considering conveniences of user proceed, DWIN DGUS v2.0 adopted integer (word), unsigned integer (word) and long integer (double word) for indicating the data as follows:

Integer: -32768 (0x8000) to +32767 (0x7FFF)

Unsigned integer: 0 (0x0000) to 65535 (0xFFFF)

Long integer: -2147483648 (0x80000000) to +2147483647 (0x7FFFFFFF)

Decimal indicated by fixed-point decimal representation that defined by user self, for example: 0x4D2(1234) indicates 12.34 if there are two decimal digits.

DWIN DGUS v2.0 adopted 65K-color systems as following color plate definition:

65K-color Definition																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Define	R4	R3	R2	R1	R0	G5	G4	G3	G2	G1	G0	B4	B3	B2	B1	B0
	Red 0xF800					Green 0x07E0					Blue 0x001F					

4 Description of Touch Config. File.

The touch Config. File is consisted of several commands that are configured according to different pages. Each touch command takes up to 16bytes, 32bytes or 48bytes memory space and constructed by following six parts,

Serial Number	Definition	Data Length	Description
1	Pic_ID	2	Picture ID
2	TP_Area	8	Touch button area:(X _s ,Y _s)(X _e ,Y _e)
3	Pic_Next	2	Go_to Page. 0xFF** means no page switches.
4	Pic_On	2	Picture ID that shows the effect animation when button pressed, 0xFF** means that there is no button pressing effect.
5	TP_Code	2	Touch key code: 0xFF**: invalid key code; 0xFE**: touching function on/off. E.g.: 0xFE00 means starts to variables data input by touch screen. 0x00**: touch key code that ASCII representation; E.g.:0x0031means button "1".
6	TP_FUN	32	When TP_Code=0xFE**, it used to describe the touch function.

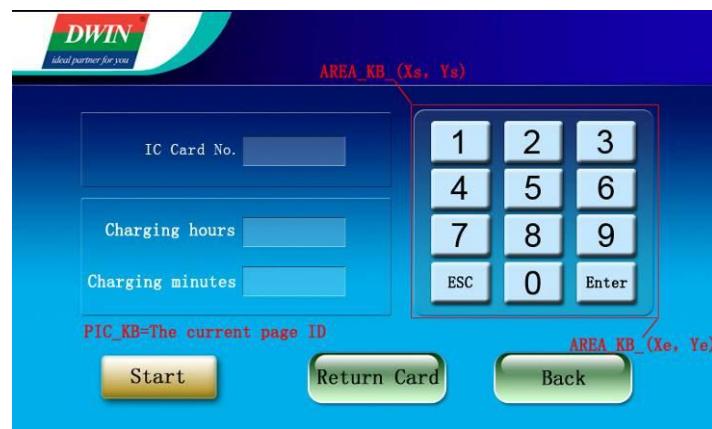
4.1 Variable Data Input (0x00)

Address	Definition	Data Length	Description
0x00	Pic_ID	2	Page ID
0x02	TP_Area	8	Touch button areas: (X _s , Y _s) (X _e , Y _e)
0x0A	Pic_Next	2	Page ID that will switch to, 0xFF** means no page switches
0x0C	Pic_On	2	Page ID that shows the effect animation when button pressed,0xFF** means that there is no button pressing effect.
0x0E	TP_Code	2	0xFE00
0x10	0xFE	1	0xFE
0x11	*VP	2	Variable address pointer that corresponded to data input
0x13	V_Type	1	Variable type that returns : 0x00=integer (word), 0x01=long integer (double word) 0x02=Unsigned Byte Parameter(High Byte of *VP), 0x03=Unsigned Byte Parameter(Low byte of *VP)
0x14	N_Int	1	Integer digits. E.g.: input 1234.56, so N_Int=0x04;
0x15	N_Dot	1	Decimal digits. E.g.:input1234.56, so N_Dot=0x02.
0x16	(x,y)	4	Display position in input process: right-alignment. (x, y) is upper-right coordinates of input cursor.
0x1A	Color	2	Font color.
0x1C	Lib_ID	1	Font Library position of ASCII, 0x00=default font lib.
0x1D	Font_Hor	1	Font size, the dot matrix number in X-direction
0x1E	Cusor_Color	1	Cursor color, 0x00 =black; others= white
0x1F	Hide_En	1	0x00=input cover, displayed as "*", other value is directly displayed.
0x20	0xFE	1	0xFE
0x21	KB_Source	1	0x00 = keyboard is on the current page; Others=keyboard is not on the current page.
0x22	PIC_KB	2	Keyboard page ID, only valid when KB_Source is not equal to 0x00.
0x24	AREA_KB	8	The area of the keyboard (X _s , Y _s , X _e , Y _e), only valid if KB_Source is not equal to 0x00.
0x2C	AREA_KB_Position	4	The location that keyboard displayed on the current page, upper-left coordinates, only valid if KB_Source is not equal to 0x00.
0x30	0xFE	1	0xFE
0x31	Limits_En	1	0xFF=Limit the input range, input beyond the boundary is invalid(Equivalent to cancel); Others=no range limit.
0x32	V_min	4	The lower limit of enter, long integer (4 byte)
0x36	V_max	4	The upper limit of enter, long integer (4 byte)
0x3A	Retention	6	Write 0x00

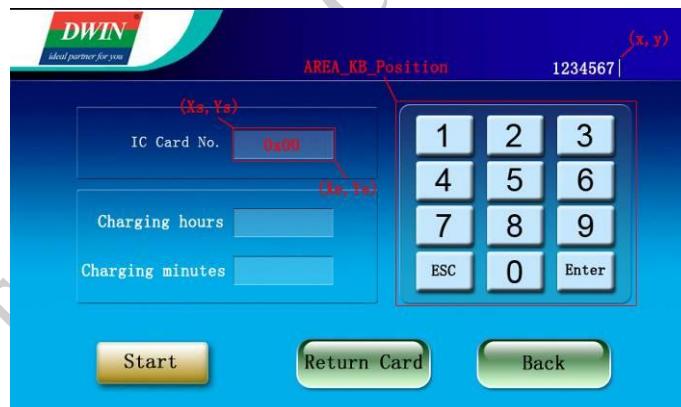
Valid keynote in input process: 0x0030-0x0039(Number 0-9), 0x002E (.), 0x002D (+/-), 0xF0 (Cancel), 0x00F1 (Confirm), 0xF2(Backspace)



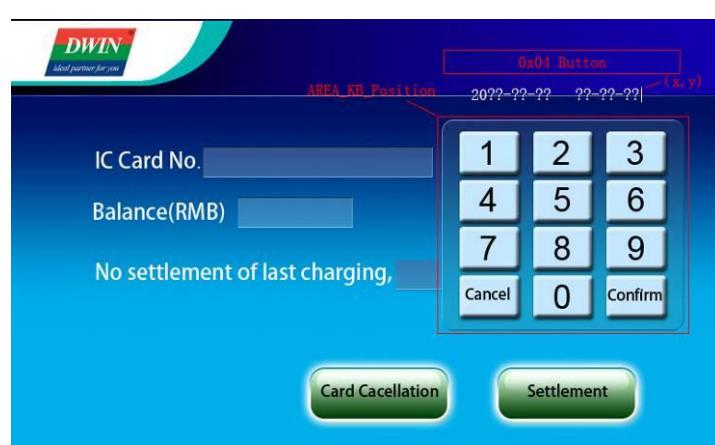
Keyboard and input start button are on a same page(KB_Source=0x00)



Keyboard is not on the current page (KB_Source=0x01): input was not triggered.



Keyboard is not on the current page (KB_Source=0x01) after trigging the keyboard

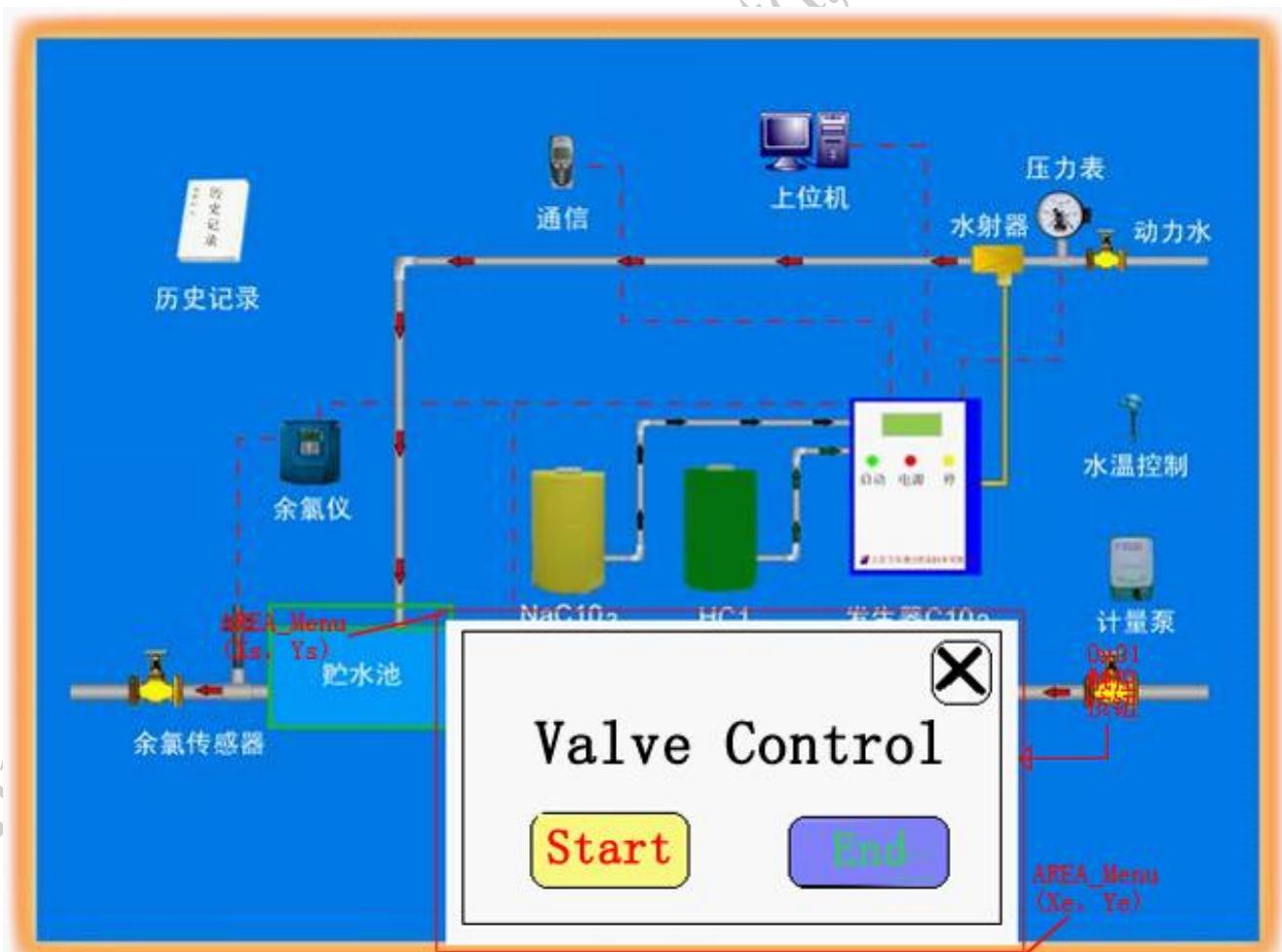


Keyboard is not on the current page (KB_Source=0x01): keyboard page

4.2 Pop-up Menu Selection (0x01)

Address	Definition	Data Length	Description
0x00	Pic_ID	2	Page ID
0x02	TP_Area	8	Touch button area: (X _s , Y _s) (X _e , Y _e)
0x0A	Pic_Next	2	Page ID which will switch to, 0xFF** means no page switches
0x0C	Pic_On	2	Page ID that shows the effect animation when button pressed, 0xFF** means that there is no button pressing effect.
0x0E	TP_Code	2	0xFE01
0x10	0xFE	1	0xFE
0x11	*VP	2	Variable address pointer, variable format is decided by VP_Mode
0x13	VP_Mode	1	0x00 = 0x00**key written to the *VP address (integer); 0x01 = **key written to the high byte address of VP position (VP_L) 0x02 = **key written to the low byte address of VP position (VP_H)
0x14	Pic_Menu	2	The picture position of the pop-up menu
0x16	AREA_Menu	8	Menu area: (X _s , Y _s)and (X _e , Y _e)
0x1E	Menu_Position_X	2	Position where pop-up menu display in current page: X of the upper-left coordinate
0x20	0xFE	1	Fixed
0x21	Menu_Position_Y	2	Position where pop-up menu display in current page: Y of the upper-left coordinates
0x23	NUL	13	Write 0x00

Notes: Valid keynote in input process: 0x0000-0x00FF, 0x00FF as cancellation (directly return without parameter selection).



In above example, pop-up menu is located in another page. Both key code button of "Start" and "End" (0x0000-0x00FE) will returned to the variable that 0x01 touch button pointed to; The key note of cancellation is 0x00FF, it will not get return values at clicking time.

4.3 Incremental Adjustment (0x02)

Address	Definition	Data Length	Description
0x00	Pic_ID	2	ID Page ID
0x02	TP_Area	8	Touch button configuration areas: (X _s , Y _s) (X _e , Y _e)
0x0A	Pic_Next	2	Page ID that will switch to, 0xFF** means no page switches. Must be 0xFF**
0x0C	Pic_On	2	Page ID that shows the effect animation when button pressed, 0xFF** means that there is no button pressing effect.
0x0E	TP_Code	2	0xFE02
0x10	0xFE	1	0xFE
0x11	*VP	2	Variable address pointer, and variable format is decided by VP_Mode
0x13	VP_Mode	1	0x00 = adjustment of VP address(integer); 0x01 = adjustment of the high-byte address of VP position (1 byte unsigned number, VP_H); 0x02 = adjustment of the low byte address of the VP address (1 byte unsigned number, VP_L).
0x14	Adj_Mode	1	Adjustment method:0x00=-- Others=++
0x15	Return_Mode	1	Treatment when over the limit: 0x00=Stop (equal to the threshold) others=cycle regulation
0x16	Adj_Step	2	Adjusted step size: 0x0000-0x7FFF
0x18	V_Min	2	Lower limit: 2-byte integer (VP_Mode=0x01or0x02, only valid in low byte)
0x1A	V_Max	2	Upper limit: 2-byte integer (VP_Mode=0x01or0x02,only valid in low byte)
0x1C	NULL	4	Write:0x00

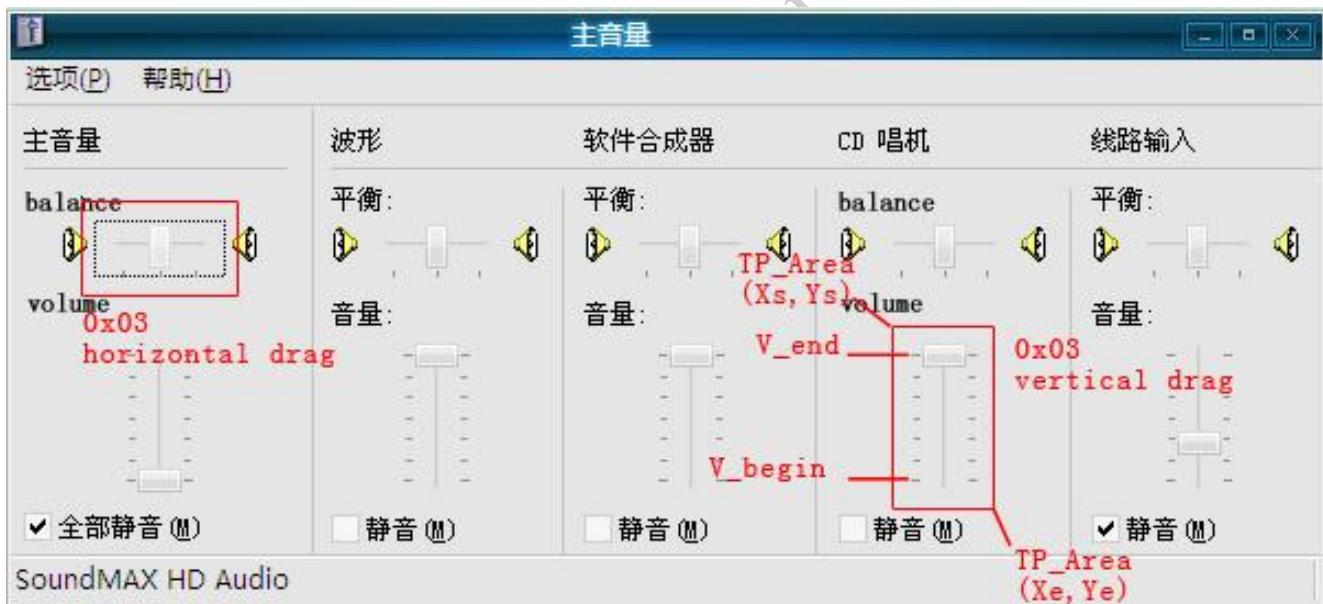


Notes: In above interface, “+” and “-” are configured to ++ increment (Adj_mode=0x01) and -- increment (Adj_mode=0x00)

4.4 Slider Adjustment (0x03)

Add.	Defination	Data Length	Description
0x00	Pic_ID	2	Page ID
0x02	TP_Area	8	Touch button areas: (X _s , Y _s) (X _e , Y _e)
0x0A	Pic_Next	2	Page ID that will switch to, 0xFF** means no page switches. Must be 0xFF**
0x0C	Pic_On	2	Page ID that shows the effect animation when button pressed, 0xFF** means that there is no button pressing effect. Must be 0xFF**
0x0E	TP_Code	2	0xFE03
0x10	0xFE	1	0xFE
0x11	*VP	2	Variable address pointer
0x13	Adj_Mode	1	➤ High 4 bit defined the format of return data: 0x0*= adjust VP word address (integer) 0x1*=adjust the high byte of word data in VP word(1 byte unsigned number, VP_H) 0x2*= adjust the low byte of word data in VP word(1 byte unsigned number, VP_L) ➤ Low 4bit defined the slider method: 0x*0 =horizontal drag, 0x*1=vertical drag.
0x14	Area_Adj	8	Working effective Region: X _s , Y _s , X _e , Y _e ; and MUST BE identical to TP_Area (touch region).
0x1C	V_begin	2	The return value corresponding to starting position, integral number.
0x1E	V_end	2	The return value corresponding to end position, integral number.

Notes: In case of any operation mistakes, pressing and drag begin to work 0.5s later.



As shown above, scale function indications is actualized by Variable 0x02 (Please see Chapter 5.1.3)

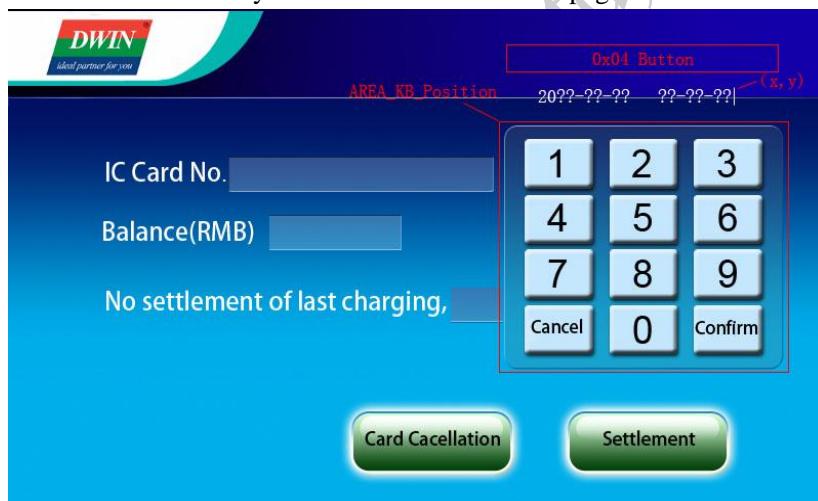
The advantage of slider input is visible, easy and convenient to operate, and no possible for over the range. If the higher accuracy of the variable input is required, use the Data Variable Show (Variable 0x10) to show the slider values in the display.



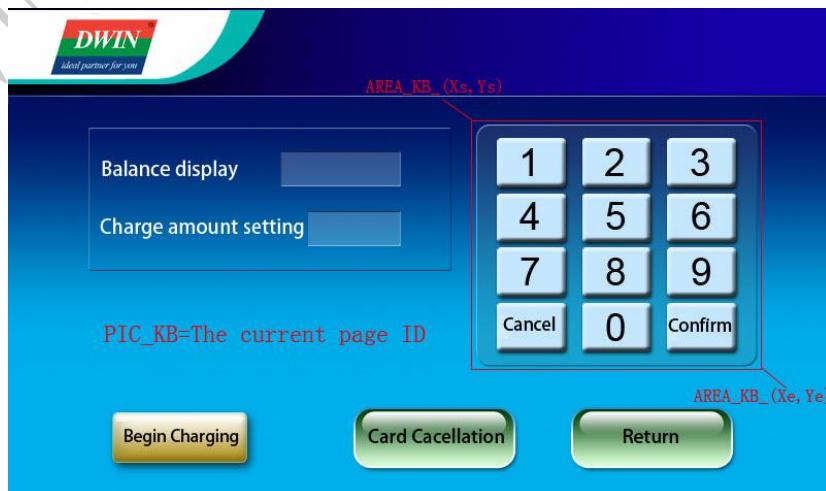
4.5 The RTC Settings (0x04)

Add.	Definition	Data Length	Description
0x00	Pic_ID	2	Page ID
0x02	TP_Area	8	Touch button areas: (X _s , Y _s) (X _e , Y _e)
0x0A	Pic_Next	2	Page ID that will switch to, 0xFF** indicates there is no switching.
0x0C	Pic_On	2	Page ID that shows the effect animation when button pressed, 0xFF** means that there is no button pressing effect.
0x0E	TP_Code	2	0xFE04
0x10	0xFE	1	0xFE
0x11	0x00 00 00	3	0x00 00 00
0x14	(x,y)	4	Text position in input process, right alignment, (x,y) is the coordinates in upper-right corner of string
0x18	Color	2	Texted font color.
0x1A	Lib_ID	1	ASCII font position, 0x00 =default font
0x1B	Font_Hor	1	Font size, the amount of dot-matrix in the X-direction
0x1C	Cursor_Color	1	Cursor color, 0x00 =black, others = white
0x1D	KB_Source	1	0x00 = keyboard is on the current page; Others=keyboard is not the current page
0x1E	PIC_KB	2	Page ID of keyboard, only valid when KB_Source is not equal to 0x00
0x20	0xFE	1	0xFE
0x21	AREA_KB	8	Keyboard area: upper left (X _s , Y _s), lower right (X _e , Y _e), only valid when KB_Source is not equal to 0x00.
0x29	AREA_KB_Position	4	The location that keyboard displayed on the current page, only valid when KB_Source is not equal to 0x00.
0x2D	NULL	3	Write 0x00

Design method is as the same as 0x00: Keyboard is not on the current page.



Keyboard is not on the current page (KB_Source=0x01) after trigging the keyboard



Keyboard is not on the current page (KB_Source=0x01): keyboard page

4.6 Button Result Value for Return(0x05)

Add.	Definition	Data length	Description
0x00	Pic_ID	2	Page ID
0x02	TP_Area	8	Touch button areas: (X _s , Y _s)(X _e , Y _e)
0x0A	Pic_Next	2	Page ID that will switch to, 0xFF** indicates there is no switching.
0x0C	Pic_On	2	Page ID that shows the effect animation when button pressed, 0xFF** means that there is no button pressing effect.
0x0E	TP_Code	2	0xFE05
0x10	0xFE	1	0xFE
0x11	*VP	2	Variable address pointer
0x13	VP_Mode	1	0x00=Returned key value is saved in VP word address (integer) 0x01=Low byte of returned key value is saved in high byte address of VP word address(VP_H); 0x02=High byte of returned key value is saved in low byte address of VP word address (VP_L)
0x14	Key_Code	2	Return Key
0x16	NULL	10	Write 0x00

4.7 ASCII Text Entry(0x06)

Add.	Definition	Data length	Description
0x00	Pic_ID	2	Page ID
0x02	TP_Area	8	Touch button areas: (X _s , Y _s)(X _e , Y _e)
0x0A	Pic_Next	2	Page ID that will switch to, 0xFF** indicates there is no switching.
0x0C	Pic_On	2	The ID number of picture of pressing effects animation, 0xFF** means that there is no button pressing effect.
0x0E	TP_Code	2	0xFE06
0x10	0xFE	1	0xFE
0x11	*VP	2	Variable address pointer
0x13	VP_Len_Max	1	Max length of the text variable, Word number, 0x01-0x7B; When the text is saved to the specified address , 0xFFFF will automatically be the terminator of the text; Text variable input maybe occupy the max variable space =VP_Len_Max +1
0x14	Scan_Mode	1	Entry mode control: 0x00=Re-entry 0x01=Open original text and modify
0x15	Lib_ID	1	The position of ASCII font which is used to display, 0x00=The default font
0x16	Font_Hor	1	Font size, amount of dot matrix in X-direction
0x17	Font_Ver	1	Font size, amount of dot matrix in Y-direction (When Lib_ID=0x00, amount of dot matrix in Y-direction must be 2*X)
0x18	Cusor_Color	1	Cursor color, 0x00=Black others=White
0x19	Color	2	Text display color
0x1B	Scan_Area_Start	4	Top left coordinate of text input display area(X _s , Y _s)
0x1F	0x00	1	
0x20	0xFE	1	
0x21	Scan_Area_End	4	Bottom right coordinate of text input display area(X _e , Y _e)
0x25	KB_Source	1	Select the keyboard page: 0x00= Keyboard is on the current page; Others=Keyboard is not on the current page
0x26	PIC_KB	2	The following data is valid when KB_Source isn't 0x00; Keyboard page ID
0x28	AREA_KB	8	The keyboard area coordinates on the keyboard page: Upper left corner(X _s , Y _s)Lower right corner(X _e , Y _e)
0x30	0xFE	1	
0x31	AREA_KB_Position	4	Display position to which the keyboard pasted (Top left corner coordinate)
0x35	NULL	11	Write 0x00

Keyboard code table for text entry

In config.file, low byte means ordinary key and high byte means capital key in two-byte key code

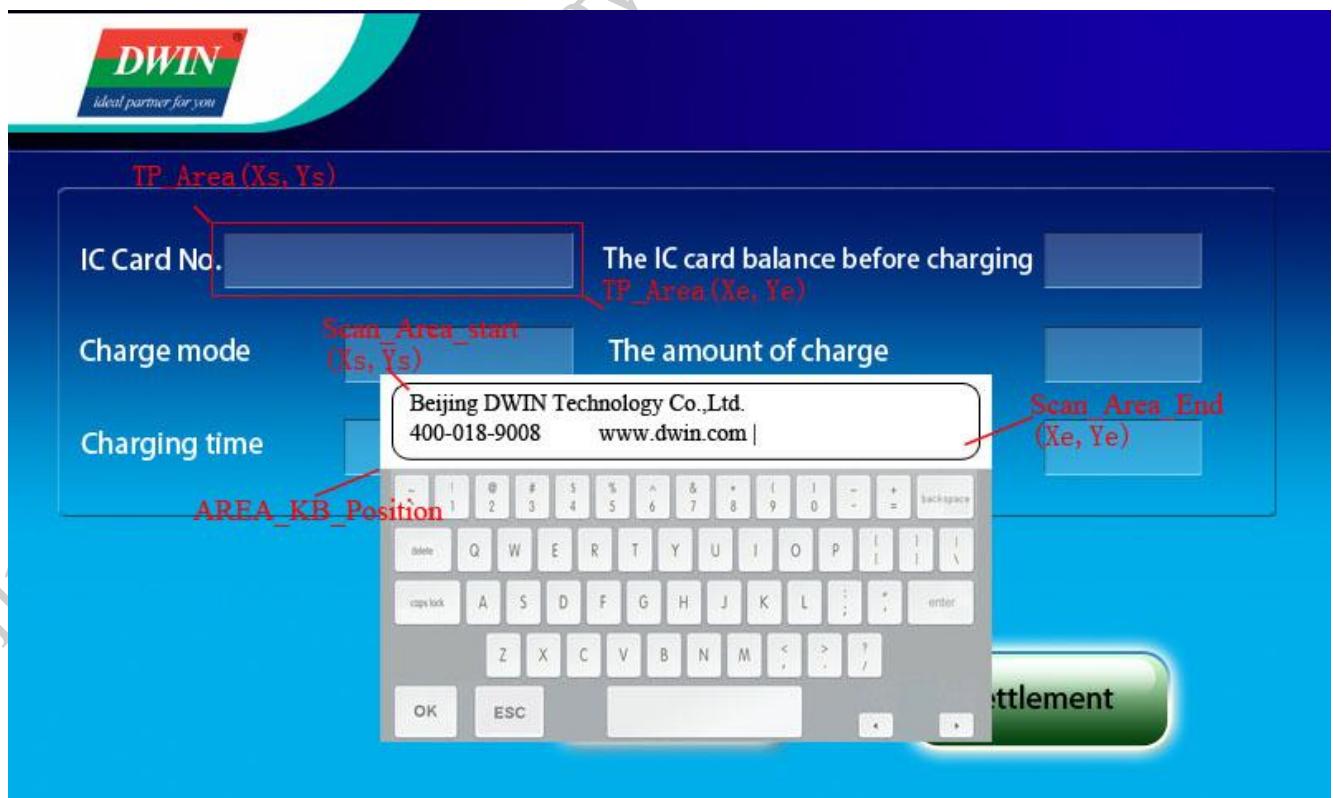
Typical English ASCII input keyboard defined as shown in the table below:

Key	Ordinary	Capital									
0x7E60	`	~	0x5171	q	Q	0x4161	a	A	0x5A7A	z	Z
0x2131	1	!	0x5777	w	W	0x5373	s	S	0x5878	x	X
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			
0x2B3D	=	+	0x7C5C	\							

Note: Key code must be less than 0x80(ASCII); 0x0D key entry will automatically transform to 0x0D 0x0A; 0x00 and 0xFF key are prohibited.

Keyboard function and key definition

Key	Definition	Description
0x00F0	Cancel	Cancel the entry to return, will not affect the variable data
0x00F1	Return	Confirm the entry to return, input the text and save it to the specified variable position
0x00F2	Backspace	Forward(Backspace) to delete a character
0x00F3	Delete	Delete a character backward
0x00F4	CapsLock	Caps lock, if enabled, corresponding button should be defined with the pressed effect
0x00F7	Left	The cursor moves one position forward
0x00F8	Right	The cursor moves one position backward



4.8 Firmware Parameter Configuration(0x07)

Add.	Definition	Data length	Description
0x00	Pic_ID	2	Page ID
0x02	TP_Area	8	Touch button areas: (X _s ,Y _s)(X _e ,Y _e)
0x0A	Pic_Next	2	Page ID that will switch to, 0xFF**indicates there is no switching.
0x0C	Pic_On	2	The ID number of picture of pressing effects animation, 0xFF** means that there is no button pressing effect.
0x0E	TP_Code	2	0xFE07
0x10	0xFE	1	0xFE
0x11	Mode	1	Operation mode selection, see the description of the ‘Operating Mode Table’
0x12	DATA_PACK	14	Data packet of operation mode, see the description of the ‘Operating Mode Table’

Operation Mode Table

Mode	Data_Pack	Notes for Data_Pack	Function																																																							
0x00	No	No	Load data in register variable area to 0x6F00-0x6FFF variable memory space(occupied low byte)																																																							
0x01	No	No	Load 0x6F00-0x6FFF variable memory space (occupying a low byte) data to the register variable area; meantime rewrite the corresponding R1-R3,R5-RA of SD card interface configuration variables																																																							
0x02	Tran_Area	Area coordinates need to be converted: Top left, bottom right corner	Convert the designated area content into a monochrome bitmap (vertical scale, print bitmap format) and save it to the data memory pointed by VP pointer. 1.Area Width(Xe-Xs+1)must be the even number; 2.Area Height(Ye-Ys+1)must be a multiple of 8; 3.*VP Pointer data format is saved as follows: *VP: Status bits, after processing, set it to be 0x5555; *VP+1: The horizontal word length=(Xe-Xs+1)&0xFFFFE/2; *VP+2: Data segment number=(Ye-Ys+1)&0xFFFF8/8; *VP+3: Bitmap data start in MSB method If the parameter automatically upload feature is enabled (R2.3=1), message will be uploaded automatically after conversion according to the revised 0x5555 instead of original *VP content.																																																							
	*VP	The 1 st address of the data buffer where saved the converted bitmap																																																								
		<table border="1"> <tr> <td></td><td>X=0</td><td>X=1</td><td>X=2</td><td>X=3</td><td>...</td><td>X=126</td><td>X=127</td></tr> <tr> <td>Y=0</td><td>D0.15</td><td>D0.7</td><td>D1.15</td><td>D1.7</td><td></td><td>D63.15</td><td>D63.7</td></tr> <tr> <td>...</td><td>...</td><td></td><td></td><td></td><td></td><td></td><td></td></tr> <tr> <td>Y=7</td><td>D0.8</td><td>D0.0</td><td>D1.8</td><td>D1.0</td><td></td><td>D63.8</td><td>D63.0</td></tr> <tr> <td>Y=8</td><td>D64.15</td><td></td><td></td><td></td><td></td><td></td><td></td></tr> <tr> <td>...</td><td>...</td><td></td><td></td><td></td><td></td><td></td><td></td></tr> <tr> <td>Y=15</td><td>D64.8</td><td></td><td></td><td></td><td></td><td></td><td></td></tr> </table>		X=0	X=1	X=2	X=3	...	X=126	X=127	Y=0	D0.15	D0.7	D1.15	D1.7		D63.15	D63.7							Y=7	D0.8	D0.0	D1.8	D1.0		D63.8	D63.0	Y=8	D64.15													Y=15	D64.8						
	X=0	X=1	X=2	X=3	...	X=126	X=127																																																			
Y=0	D0.15	D0.7	D1.15	D1.7		D63.15	D63.7																																																			
...	...																																																									
Y=7	D0.8	D0.0	D1.8	D1.0		D63.8	D63.0																																																			
Y=8	D64.15																																																									
...	...																																																									
Y=15	D64.8																																																									
0x03	*VP	Data pointer	Send the Tx_Len data seated from *VP pointer position																																																							
	Tx_LEN	Data length for sending	Tx_Len is a word variable, with length from 0x0001 to 0xFFFF																																																							
0x04	The functionality is the same as 0x03, data is sent to COM2(the serial is reserved by system)																																																									
0x05	Tran_Area	Area coordinates need to be converted: Top left, bottom right corner	Converse the designated area content into a monochrome bitmap (horizontal scale, print bitmap format) and save it to the data memory pointed by VP pointer. 1.Area Width(Xe-Xs+1)must be a multiple of 16; 2.*VP Pointer data format is saved as follows: *VP: Status bits, after processing, set it to be 0x5555; *VP+1: The horizontal word length=(Xe-Xs+1)&0xFFFF0/16; *VP+2: Data segment number=(Ye-Ys+1); *VP+3: Bitmap data start in MSB method If the parameter automatically upload feature is enabled (R2.3=1), message will be uploaded automatically after conversion according to the revised 0x5555 instead of original *VP content.																																																							
	*VP	The 1 st address of the data buffer where saved the converted bitmap	The command is mainly for printing and output of the screen contents.																																																							
0x06	Frame_Head	Frame header (2byte)	Send the current touched position to COM2 (System reserved com port), the format is:																																																							
	Frame_End	Frame end (2byte)	Frame_Head+X+Y+Check(The cumulative Sum for 1 byte of X,Y) + Frame_end.																																																							

5 Variable Configuration File Descriptions

Variable configuration file is composed of several pieces of variable commands that is configurated by pages. Each of variable command occupies 32 bytes storage in regular.

Fixed allocation of 2KB (0x0800) for each page of variable storage space can be set to 64 variables per page.

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

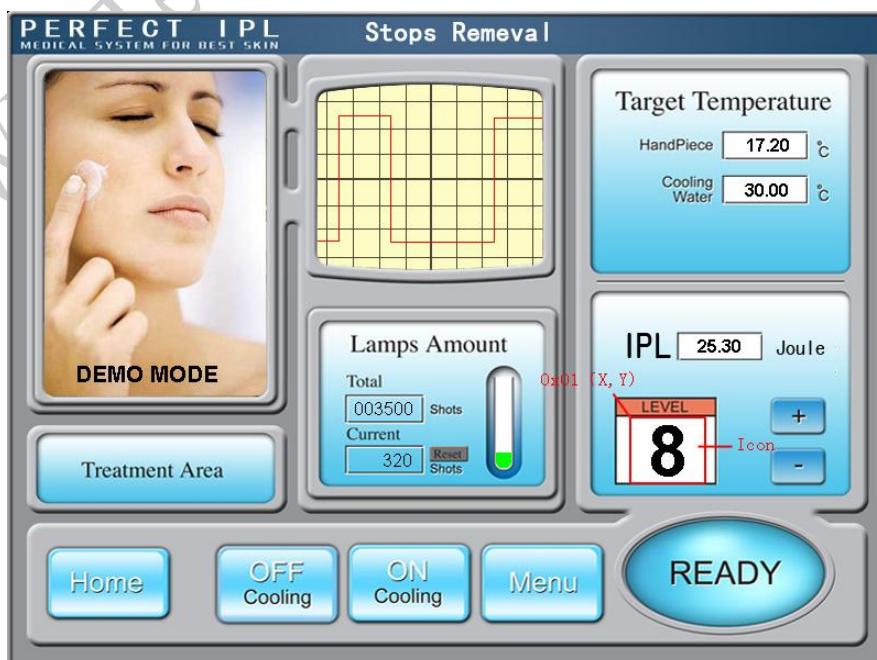
The touch control commands consist the following five parts:

Serial No.	Definition	Data Length	Description
1	0x5A	1	Fixed
2	Type	1	Variable category
3	*SP	2	After loading from Flash, the variable descriptions files saved in the address pointer in data storage area, the 0xFFFF means it will not save to the RAM
4	Len_Dsc	2	Word length of variable description
5	Description	N	Variable description

5.1 Icon variable

5.1.1 Variable icon display (0x00)

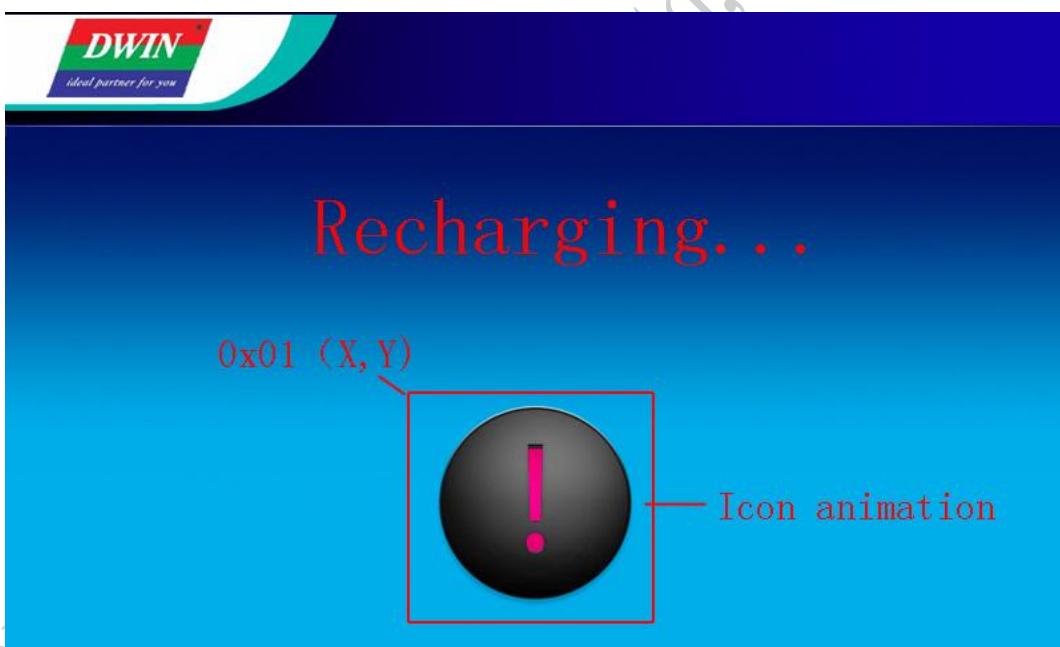
Add.	Definition	Data Length	Description
0x00	0x5A00	2	
0x02	*SP	2	Variable description pointer, 0xFFFF indicates it loaded from the configuration file.
0x04	0x0008	2	
0x06	0x00	*VP	Variable pointer, variable is integer
0x08	0x01	(x,y)	Variable display position, coordinates of upper-left corner of the icon
0x0C	0x03	V_Min	Lower limit of variables, will not be displayed if over range
0x0E	0x04	V_Max	Upper limit of variables, will not be displayed if over range
0x10	0x05	Icon_Min	Icon ID corresponding from V_Min
0x12	0x06	Icon_Max	Icon ID corresponding from V_Max
0x14	0x07:H	Icon_Lib	Storage location of the icon library
0x15	0x07:L	Mode	ICON display mode, 0x00 =transparent(does not display background) Others= display the icon background



5.1.2 Animation icons display(0x01)

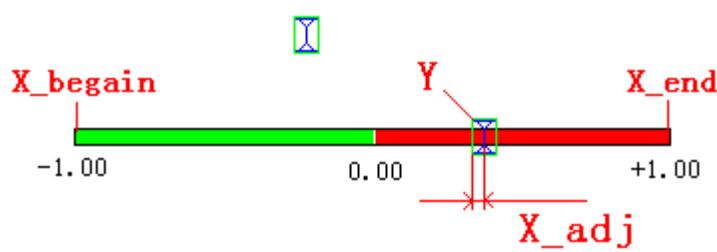
Add.	Definition	Data length	Description
0x00	0x5A01	2	
0x02	*SP	2	Variable description pointer, 0xFFFF indicates it loaded from the configuration file.
0x04	0x000A	2	
0x06	0x00 *VP	2	The initial icon variable pointer, double-word variable, low word retention, high word of unsigned (0x0000-0xFFFF) to control the animation mode.
0x08	0x01 (x,y)	4	Variable display position, coordinates of upper-left corner of the icon
0x0C	0x03 0x0000	2	FIXED
0x0C	0x03 0x0000	2	Fixed
0x0E	0x04 V_Stop	2	When variable is this value, will display fixed icon
0x10	0x05 V_Start	2	When variable is this value, will automatically display animation icon
0x12	0x06 Icon_Stop	2	Icon fixed displayed when variable is the value of V_STOP
0x14	0x07 Icon_Start	2	Displaying icons from Icon_Start to Icon_End when variable is V_Start
0x16	0x08 Icon_End	2	
0x17	0x09:H Icon_Lib	1	Storage location of the icon library
0x18	0x09:L Mode	1	ICON display mode, 0x00 =transparent(does not display background) Others= display the icon background

When the variable is not equal to V_Stop or V_Start, it will not display the icon or animation.



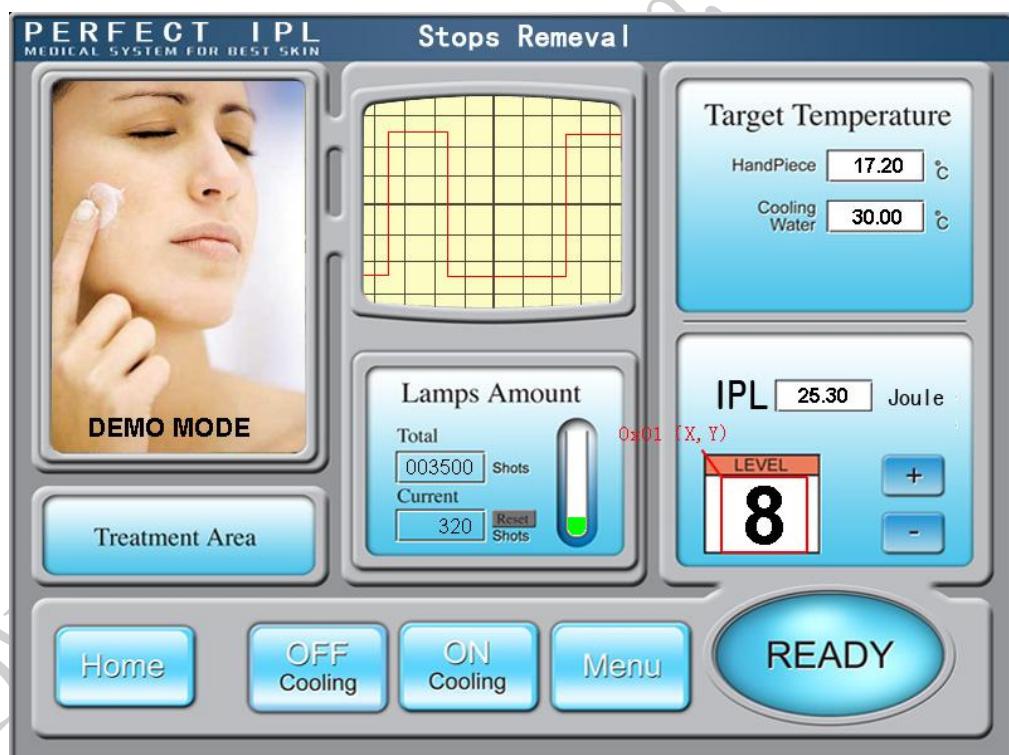
5.1.3 Slider scale indicator(0x02)

Add.		Definition	Data Length	Description
0x00		0x5A02	2	
0x02		*SP	2	Variables description pointer, 0xFFFF indicates it loaded by config. file
0x04		0x0009	2	
0x06	0x00	*VP	2	Variable pointer, variable format is decided by VP_DATA_Mode
0x08	0x01	V_begin	2	Variable value corresponding to the starting scale
0x0A	0x02	V_end	2	Variable value corresponding to the end scale
0x0C	0x03	X_begin	2	The starting scale coordinates (Y coordinates vertical)
0x0E	0x04	X_end	2	The termination of scale coordinates (Y coordinates for vertical)
0x10	0x05	Icon_ID	2	Icon ID of scale slider
0x12	0x06	Y	2	Indicator shows the Y coordinate position (vertical for the X coordinate)
0x14	0x07:H	X_adj	1	0x00-0xFF indicates the X coordinate of the icon displayed forward offset (Y for vertical), 0x00-0xFF
0x15	0x07:L	Mode	1	Scale mode: 0x00 = horizontal scale bar; 0x01 = vertical scale bar
0x16	0x08:H	Icon_Lib	1	icon library storage location
0x17	0x08:L	Icon_mode	1	ICON display mode, 0x00 =transparent(does not display background) Others = display the icon background
0x18	0x09:H	VP_DATA_Mode	1	0x00: *VP points to an integer variable 0x01: *VP points to the high byte data in the integer variable 0x02: *VP points to the low byte data in the integer variable



5.1.4 WordArt variable display(0x03)

Add.	Definition	Data Length	Description
0x00	0x5A03	2	
0x02	*SP	2	Variables description pointer, 0xFFFF indicates it loaded by config. file
0x04	0x0007	2	
0x06	0x00	2	Variables description pointer
0x08	0x01	X,Y	Starting position, left-alignment mode, The coordinate is upper-left point of displayed string.
0x0C	0x03	Icon0	ICON_ID corresponding to 0, the sequence is 0123456789-
0x0E	0x04:H	Icon_Lib	Icon library location
0x0F	0x04:L	Icon_Mode	The display mode of the ICON, 0x00 =transparent(does not display background) Others =display the icon background
0x10	0x05:H	Number of integer digits	Integer digits to display
0x11	0x05:L	Number of decimal digits	Decimal digits to display
0x12	0x06:H	Variable data type	0x00=integer(2 bytes) 0x01 =long integer(4 bytes)



5.1.5 Images Animation(0x04)

Add.	Definition	Data Length	Description
0x00		0x5A04	2
0x02		*SP	2 Variables description pointer, 0xFFFF indicates it loaded by config. file
0x04		0x0004	2
0x06	0x00	0x0000	2 Fixed
0x08	0x01	Pic_Begin	2 Initial image location
0x0A	0x02	Pic_End	2 End image location
0x0C	0x03:H	Frame_Time	1 Time to show one picture, the unit is 8ms

The index for starting frame must be smaller than the index for last frame;

If animation variable is set in Pic_End, it will achieve multiple plays;

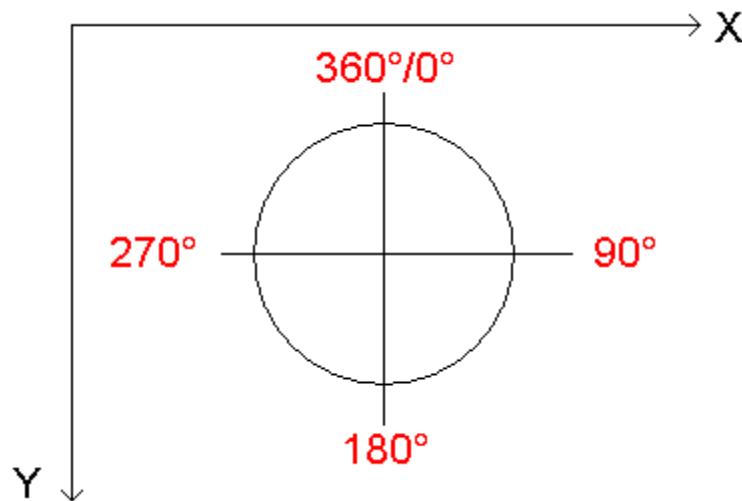
Multiple plays could be ended by serial port command or touch control operation for images switches.



5.1.6 Icon Rotation Instructions(0x05)

Add.		Definition	Data Length	Description
0x00		0x5A05	2	
0x02		*SP	2	Variables description pointer, 0xFFFF indicates it loaded by config. file
0x04		0x000C	2	
0x06	0x00	*VP	2	Variable pointer, variable format is decided by VP_Mode
0x08	0x01	Icon_ID	2	Specified ICON ID。
0x0A	0x02	Icon_Xc	2	Rotation center position of icon : X coordinate
0x0C	0x03	Icon_Yc	2	Rotation center position of icon : Y coordinate
0x0E	0x04	Xc	2	Rotation center position for ICON display on current screen: Xcoordinate
0x10	0x05	Yc	2	Rotation center position for ICON display on current screen: Ycoordinate
0x12	0x06	V_Begin	2	The variable value corresponding to the starting angle of rotation, integer, no display if cross border
0x14	0x07	V_End	2	The variable value corresponding to the end angle of rotation, integer, no display if cross border
0x16	0x08	AL_Begin	2	Starting angle of rotation, 0-720(0x000-0x2D0), unit is 0.5 °.
0x18	0x09	AL_End	2	End angle of rotation, 0-720(0x000-0x2D0), unit is 0.5 °.
0x1A	0x0A:H	VP_Mode	1	0x00: *VP points to an integer variable 0x01: *VP points to the high byte data of an integer variable 0x02: *VP points to the low byte data of an integer variable
0x1B	0x0A:L	Lib_ID	1	ICON library ID.
0x1C	0x0B	Mode	1	ICON display mode, 0x00=Transparent(does not display the background)Others =display the icon background

The rotation is always assumed as *clockwise* rotation, namely AL_End must be greater than AL_Begin (if AL_End is less than AL_Begin, system processing will automatically add 360 °)

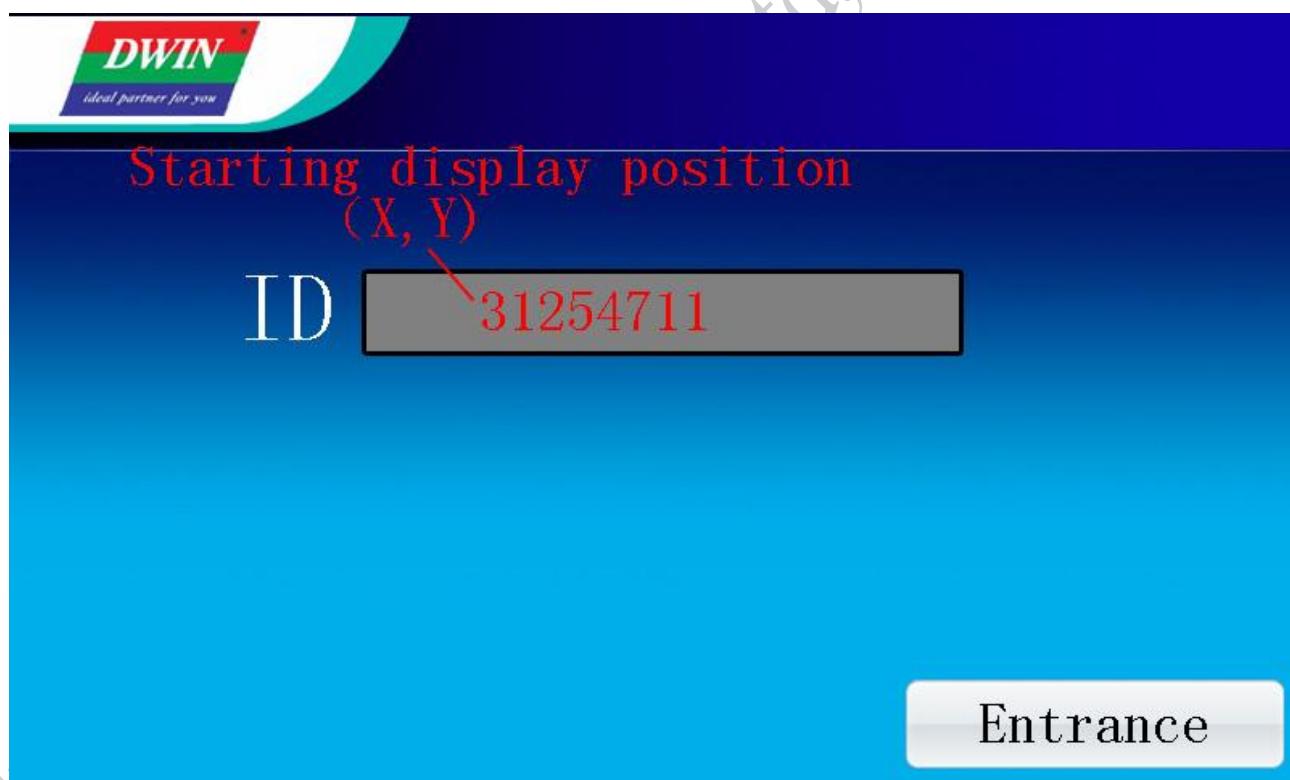


The command is mainly used for pointer of instrument dials

5.2 Text Variables

5.2.1 Data variables display(0x10)

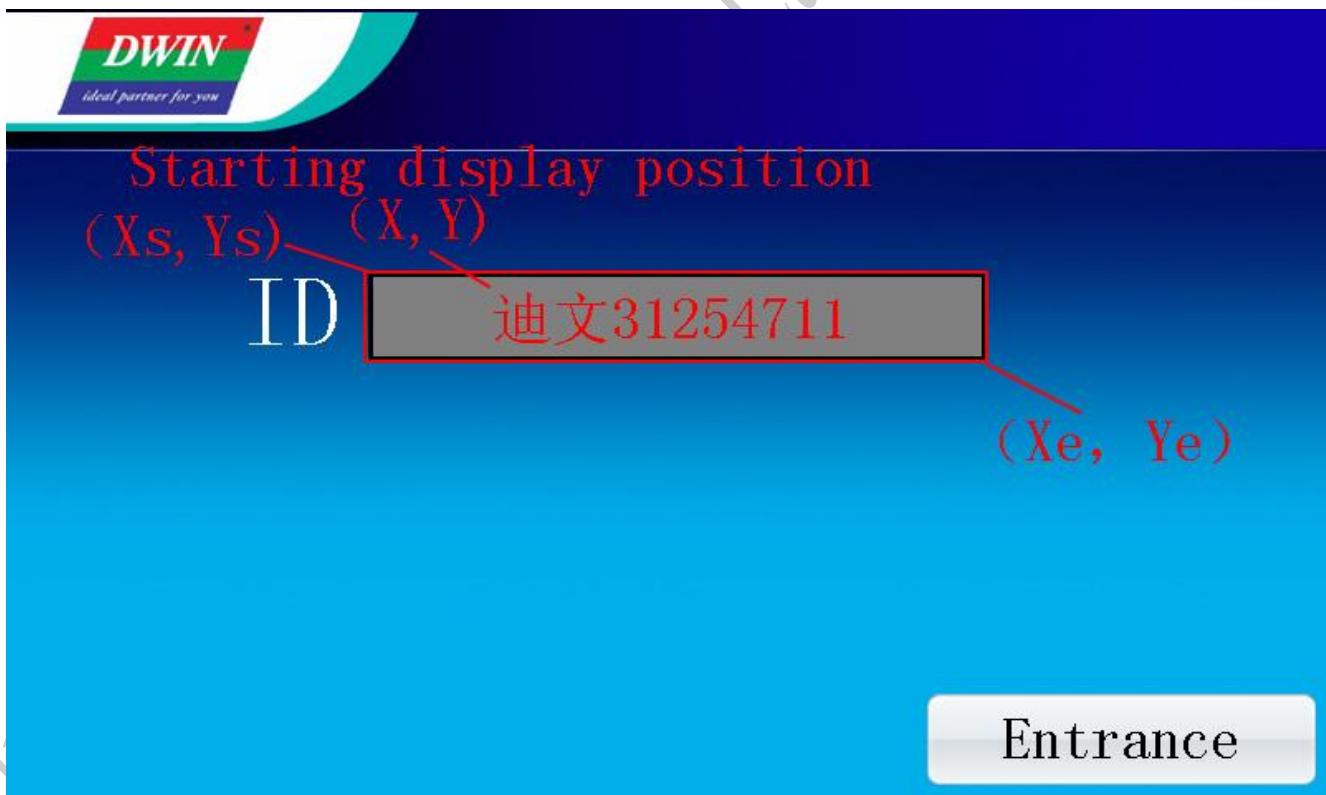
Add.	Defined	Data Length	Description
0x00		0x5A10	2
0x02		*SP	2 Variable description pointer, 0xFFFF indicates it loaded by config. file
0x04		0x000D	2
0x06	0x00	*VP	2 variable pointer
0x08	0x01	X,Y	4 Starting display position. The coordinate is upper-left point of displayed string.
0x0C	0x03	COLOR	2 Text color
0x0E	0x04:H	Lib_ID	1 ASCII character position
0x0F	0x04:L	Font size	1 Dot matrix of the font X-Direction
0x10	0x05:H	Alignment	1 0x00 = right-aligned 0x01 = left-align 0x02 = center
0x11	0x05:L	Integer digit	1 Integer display
0x12	0x06:H	Decimal digit	1 Decimal display
0x13	0x06:L	Variable data type	1 0x00 = integer (2 bytes) 0x01 = long integer (4 bytes) 0x02 = high-byte address of *VP 0x03 = low-byte address of the *VP
0x14	0x07:H	Len_unit	1 Length of the variable unit(fixed string) to display, 0x00 indicates no display
0x15	0x07:L	String_Unit	Max11 Unit string, an ASCII coding



5.2.2 Text Display (0x11)

Address	Definition	Data Length	Description
0x00	0x5A11	2	
0x02	*SP	2	Variable description pointer, 0xFFFF indicates it loaded by config. file
0x04	0x000D	2	
0x06	0x00	2	*VP Text pointer
0x08	0x01	X,Y	4 Starting display position. The coordinate is upper left point of displayed string.
0x0C	0x03	Color	2 Text color
0x0E	0x04	Xs Ys Xe Ye	8 Text box
0x16	0x08	Text_length	2 Amount of displaying bytes. It will be shown unless 0xFFFF or texting towards the end of the textbox.
0x18	0x09:H	Font0_ID	1 Font location of the ASCII in coding 0x01-0x04
0x19	0x09:L	Font1_ID	1 Font location of coding 0x00&0x05 and non-ASCII in coding 0x01-0x04
0x1A	0x0A:H	Font_X_Dots	1 Dot matrix of the font X-Direction(0x01-0x04 mode,ASCII character X is accounted by X/2)
0x1B	0x0A:L	Font_Y_Dots	1 Dot matrix of the font Y-Direction
0x1C	0x0B:H	Encode_Mode	1 Coding method: 0=8bit coding 1=GB2312 2=GBK 3=BIG5 4=SJIS 5=UNICODE
0x1D	0x0B:L	HOR_Dis	1 Character horizontal interval
0x1E	0x0C:H	VER_Dis	1 Character vertical interval

Note: dot matrix of Y-Direction should be even number when text displayed.



5.2.3 The RTC Display(0x12)

➤ Text RTC Display

Add.	Definition	Data Length	Description
0x00	0x5A12	2	
0x02	*SP	2	Variable description pointer, 0xFFFF indicates loaded by config. file
0x04	0x000D	2	
0x06	0x00	2	
0x08	0x01	X,Y	4 Display position. The coordinate is upper left point of displayed string.
0x0C	0x03	Color	2 Font color
0x0E	0x04:H	Lib_ID	1 Character position
0x0F	0x04:L	Font size	1 Number of X-direction dot matrix
0x10	0x05	String_Code	MAX16 Encoded string, using the RTC code tables and ASCII characters. Suppose the current time is 2012-05-02 12:00:00 Wednesday. ➤ Y-M-D H: Q: S 0x00 will be displayed as 2012-05-02 12:00:00 ➤ M-D W H:Q 0x00 will be displayed as 05-02 WED 12:00

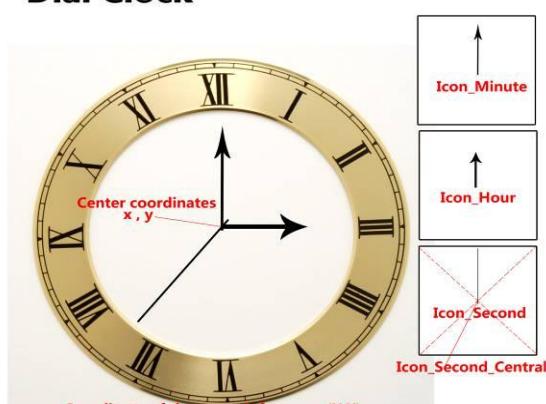
RTC coding table

Description	Coding	Display Format
Calendar year	Y	2000-2099
Calendar month	M	01-12
Calendar day	D	01-31
Calendar hour	H	00-23
Minute	Q	00-59
Second	S	00-59
Date	W	SUN MON TUE WED THU FRI SAT
Coding end	0x00	

➤ Dial Clock Display

Add.	Definition	Data Length	Description
0x00	0x5A12	2	
0x02	*SP	2	Variable description pointer, 0xFFFF indicates loaded by the config. file;
0x04	0x000D	2	
0x06	0x00	2	
0x08	0x01	X,Y	4 Pointer center of Dial clock
0x0C	0x03	Icon_Hour	2 ID of Hour hand ICON, 0xFFFF indicates no display
0x0E	0x04	Icon_Hour_Central	4 Rotation center of Hour hand ICON
0x12	0x06	Icon_Minute	2 ID of Minute hand ICON, 0xFFFF indicates no display
0x14	0x07	Icon_Minute_Central	4 Rotation center of Minute hand ICON
0x18	0x09	Icon_Second	2 ID of Second hand ICON, 0xFFFF indicates no display
0x1A	0x0A	Icon_Second_Central	4 Rotation center of Second hand ICON
0x1E	0x0C:H	ICON_Lib	1 ICON Library ID which includes the pointer icons
0x1F		No defintion	1 Write 0x00

Dial Clock



2012-08-01 15:00:36

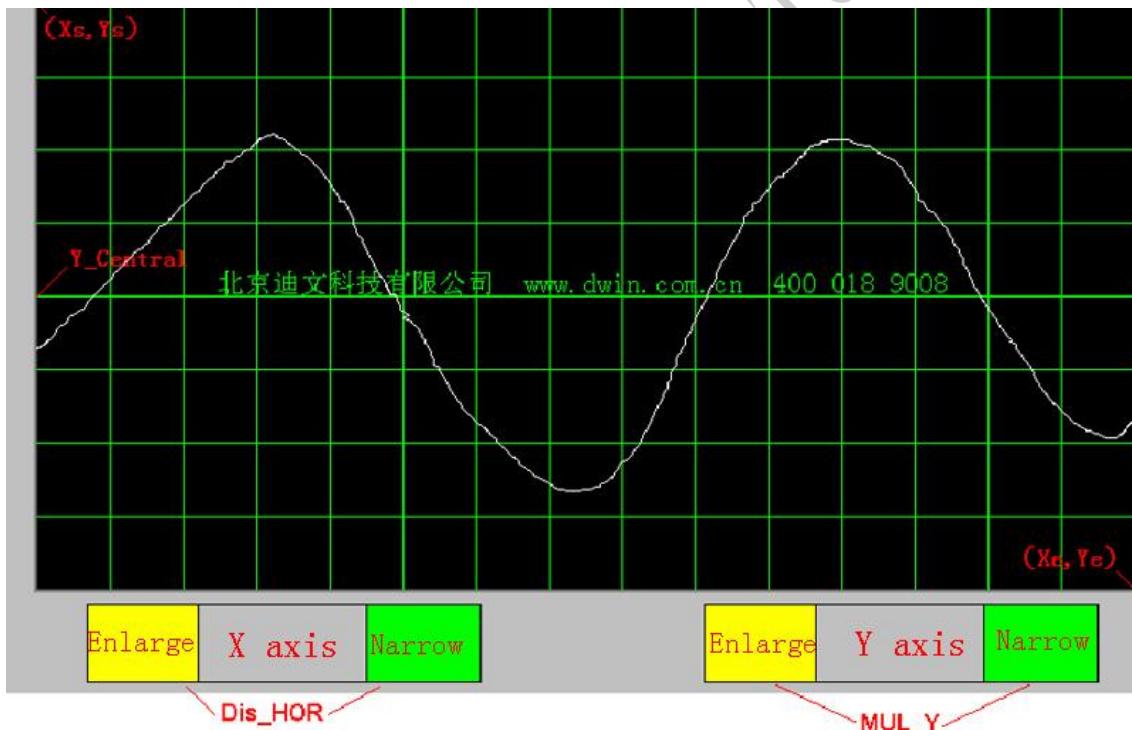
5.3 Graphics variable

5.3.1 Curve Display (0x20)

Add.	Definition	Data Length	Description
0x00		0x5A20	2
0x02	*SP	2	Variables description pointer, 0xFFFF indicates loaded by config. file
0x04		0x000A	2
0x06	0x00	0x0000	2 Not defined
0x08	0x01	X _s :Y _s ;X _e :Y _e	8 Curve window, over range of curve will not displayed Upper left: (X _s , Y _s); Bottom right(X _e , Y _e)
0x10	0x05	Y_Central	2 Curve central axis position
0x12	0x06	VD_Central	2 Curve data value corresponding to the central axis which should generally take the value of 50% of sum of max. and min.
0x14	0x07	Color	2 Curve color
0x16	0x08	MUL_Y	2 Magnification in vertical, the unit is 1/256, 0x0000-0x7FFF
0x18	0x09:H	CHANEL	1 Data source channel, 0x00-0x07
0x19	0x09:L	Dis_HOR	1 Interval in horizontal, 0x01-0xFF

Curve data sent by Command 0x84.

For further information, please reference Chapter 2.2 Command Set.



If the content of variable description stored in the data storage space (*SP specify the storage position) :

- With 0x02 incremental touch commands, automatic curve scaling could be achieved with no need of the user code.
- With 0x03 drag touch command to revise Y_Central value, curve moving up and down could be achieved with no need of the user code.

Vertical axis magnification calculation of full scale curve:

$MUL_Y = (Ye - Ys) * 256 / (Vmax - Vmin)$ Ye, Ys is the Y coordinate of the curve window,

V_{max}, V_{min} is the maximum and the minimum of the curve data, for example,

Should a 12bit A/D data acquisition ($V_{max} = 4095$ $V_{min} = 0$) be full-scale displayed in the $Ys=50, Ye=430$ area correspondingly that $MUL_Y = (430 - 50) * 256 / (4095 - 0) = 23.7$, we round down to get 23.

5.3.2 Basic Graphic Display(0x21)

Add.		Definition	Data Length	Description	
0x00		0x5A21	2		
0x02		*SP	2	Variables description pointer, 0xFFFF indicates it loaded by config. file	
0x04		0x0005	2		
0x06	0x00	*VP	2	Variable data pointer;	
0x08	0x01	Area	8	Definition of drawing display area: the upper left corner of the specified area and the lower right corner coordinates; No display if drawing cross the border; Only be valid for 0x0001-0x0005 graphic command.	

Format description of Variable data

Add.	Definition	Description
VP	CMD	Graphic command
VP+1	Data_Pack_Num_Max	The max data packet number; Drawing will automatic terminate when get the closing conditions; Line command (0x0002), defined as line number (Vertices-1);
VP+2	DATA_Pack	

Date packet Description of Graphic drawing command

Command	Operation	Format description of Graphic data packet (Unit of the comparative addressand data length is word)			
		Comparative Add.	Data length	Definition	Description
0x0001	Point set	0x00	2	(x,y)	Coordinate of the set point, use the high byte of X as judging factor.
		0x02	1	Color	Color of the set point
0x0002	Line	0x00	1	Color	Line color
		0x01	2	(x,y)0	Coordinate of line vertex0, use the high byte of X as judging factor.
		0x03	2	(x,y)1	Coordinate of line vertex1, use the high byte of X as judging factor.
		0x01+2*n	2	(x,y)n	Coordinate of line vertexn, use the high byte of X as judging factor.
0x0003	Rectangle	0x00	2	(x,y)s	Top left corner coordinate of rectangle, use the high byte of X as judging factor.
		0x02	2	(x,y)e	Bottom right corner coordinate of the rectangle
		0x04	1	Color	Color of rectangle
0x0004	Rectangle area filling	0x00	2	(x,y)s	Top left corner coordinate of rectangle, use the high byte of X as judging factor.
		0x02	2	(x,y)e	Bottom right corner coordinate of the rectangle
		0x04	1	Color	Filling color of rectangle
0x0005	Round	0x00	2	(x,y)	Center coordinate, use the high byte of X as judging factor.
		0x02	1	Rad	Radius
		0x03	1	Color	Round color
0x0006	Cut and paste of image area	0x00	1	Pic_ID	ID of the page in which to shear the image; use the high byte as judging factor.
		0x01	2	(x,y)s	Top left coordinate of the cut area of image
		0x03	2	(x,y)e	Bottom right coordinate of the cut area of image
		0x05	2	(x,y)	The coordinate position of current page which the cut image pasted to, top left corner
0x**07	ICON display	0x00	2	(x,y)	Coordinate position to display, use the high byte of X as judging factor.

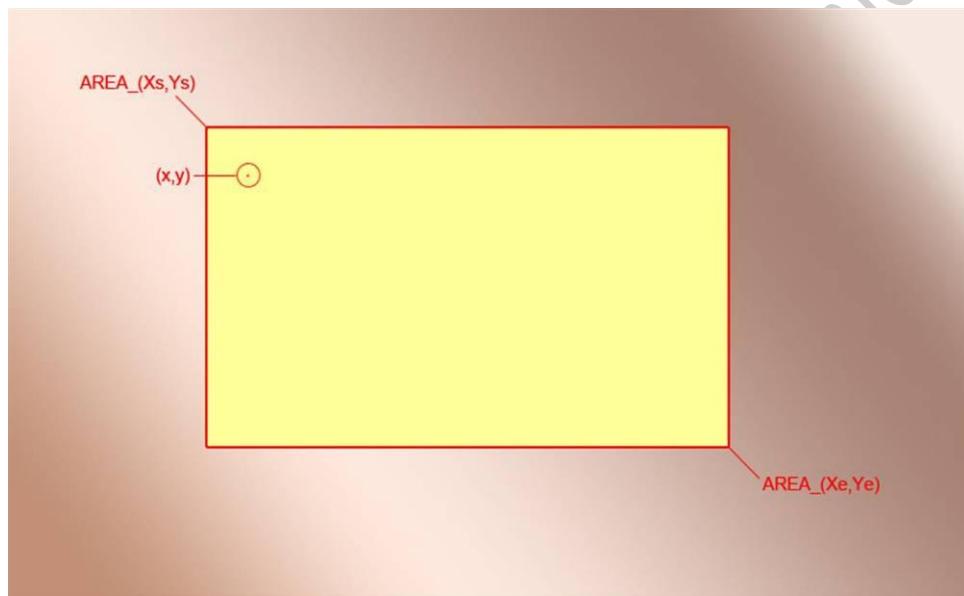
		0x02	1	ICON_ID	ICON ID, ICON library position and display mode are specified by the high byte of command. ICON is fixed to display without the background color
0x0008	Area filling	0x00	2	(x,y)	Coordinates for color seed, use the high byte of X as judging factor.
		0x02	1	Color	Fill with single color
0x0009	Vertical line	0x00	1	Color0	Connect 2 coordinates (X0,Y0s),(X0,Y0e) with line in Color0, use the high byte of X0 as judging factor.
		0x01	1	X0	
		0x02	1	Y0s	
		0x03	1	Y0e	

Judging factor:

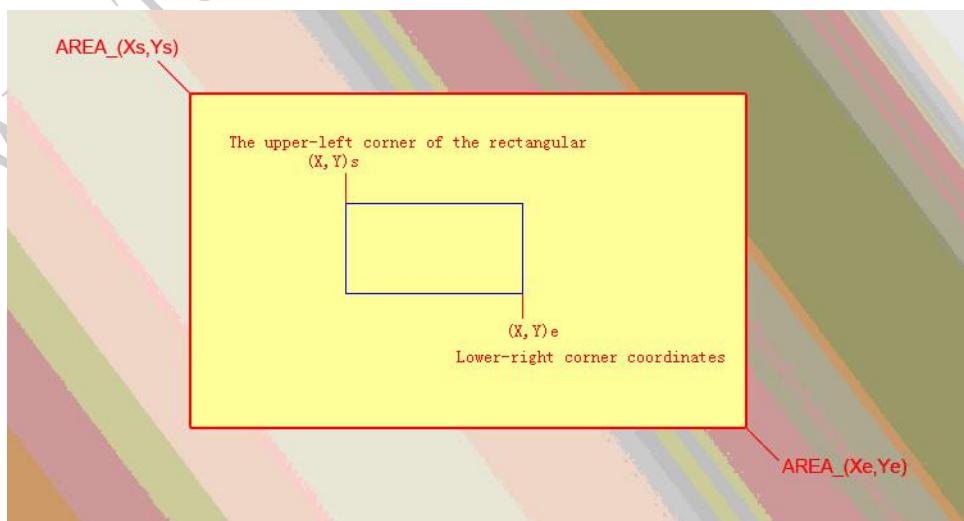
0xFF Current drawing operation finished.

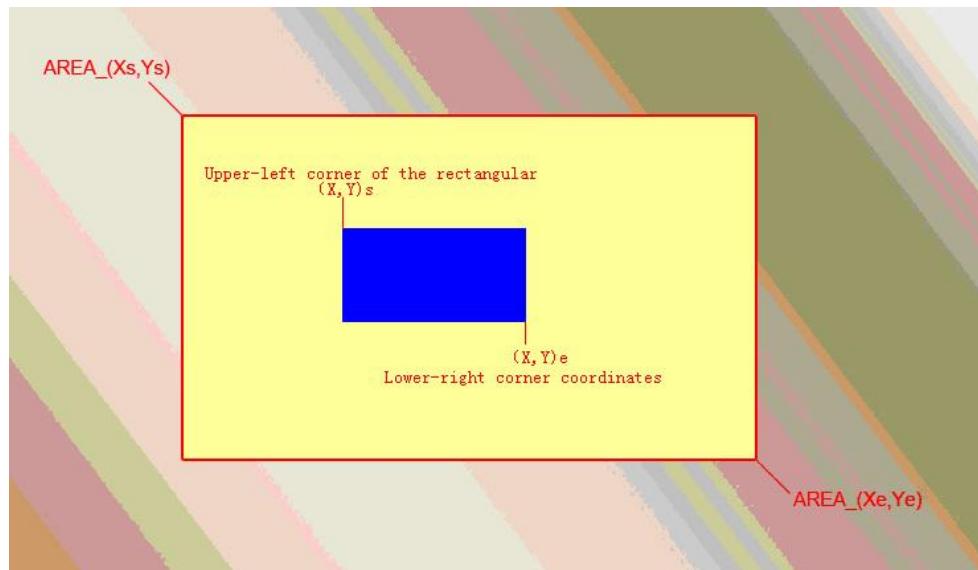
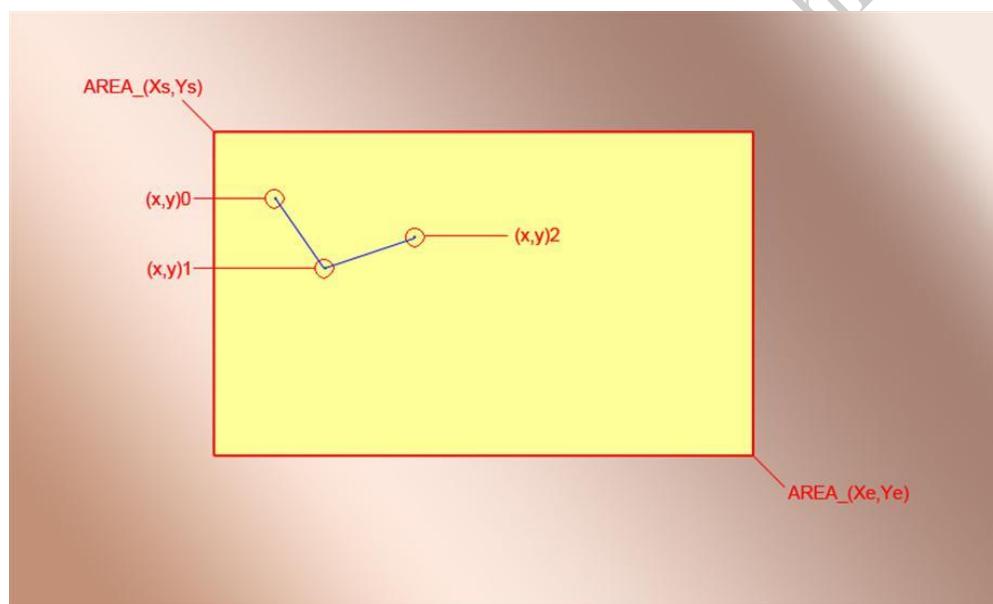
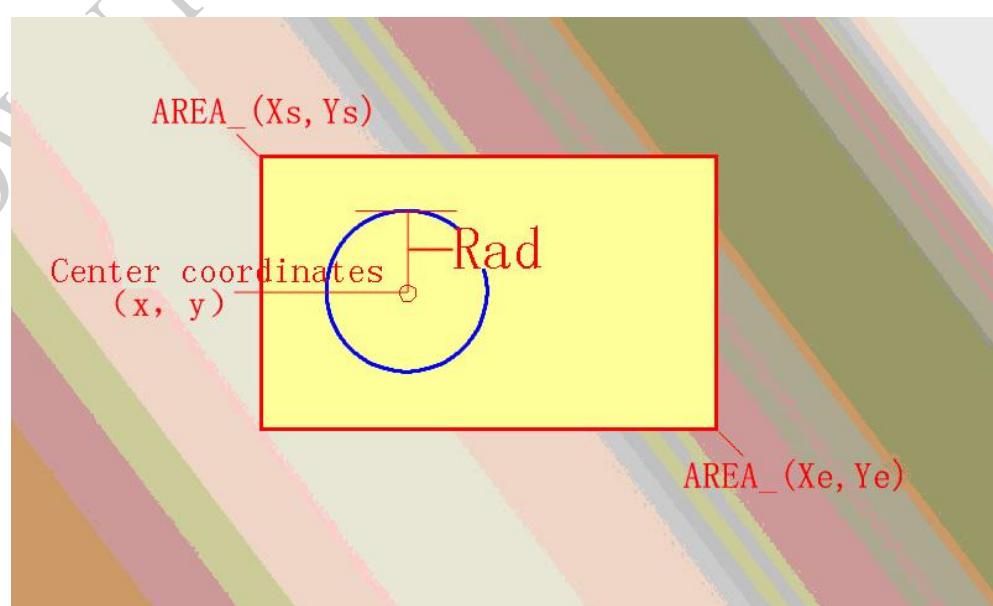
0xFE This operation will be skipped (ignored).

0x0001 Set point

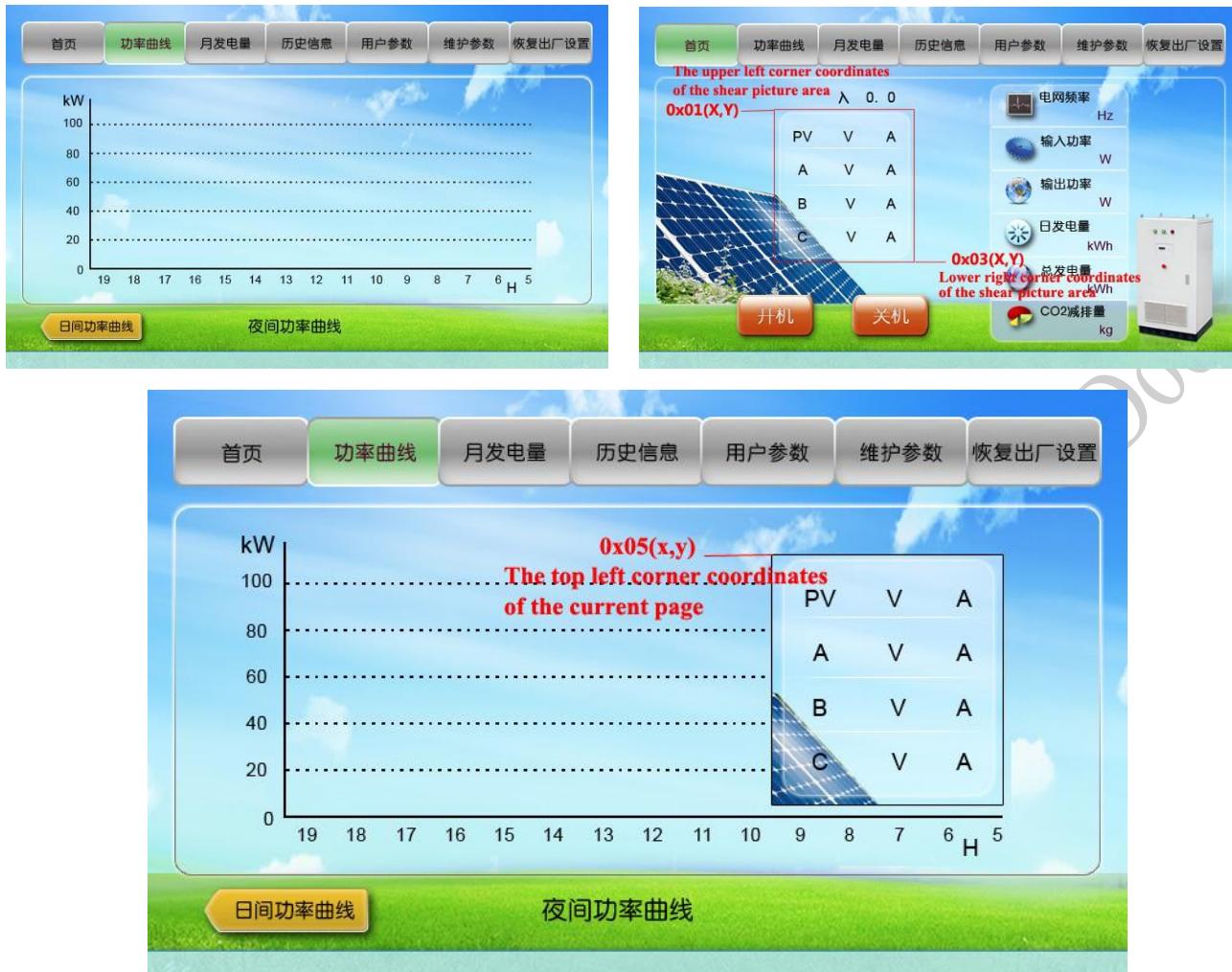


0x0002 Rectangular box

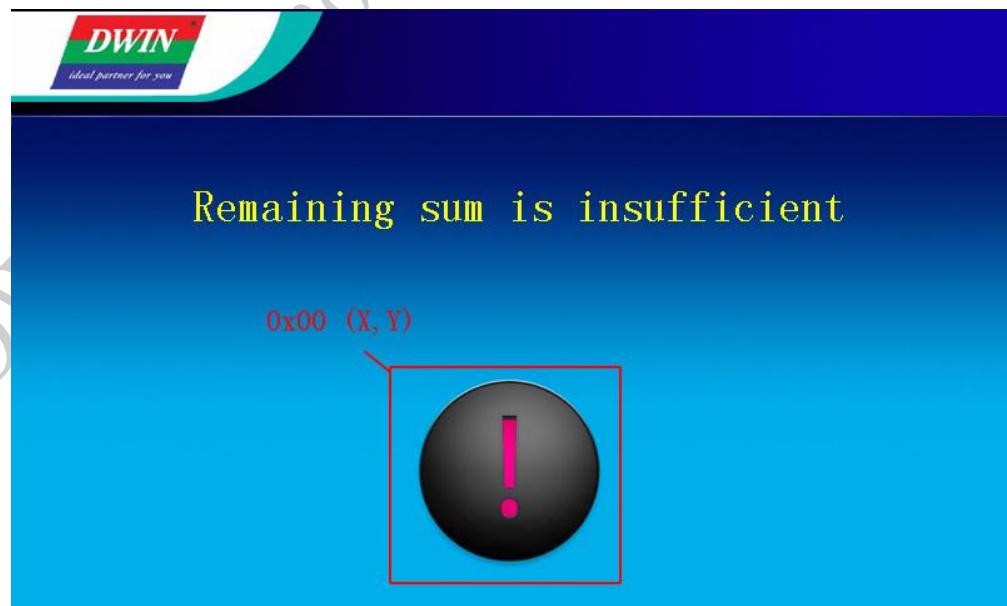


0x0003 Rectangle**0x0004 Rectangle area filling****0x0005 Round**

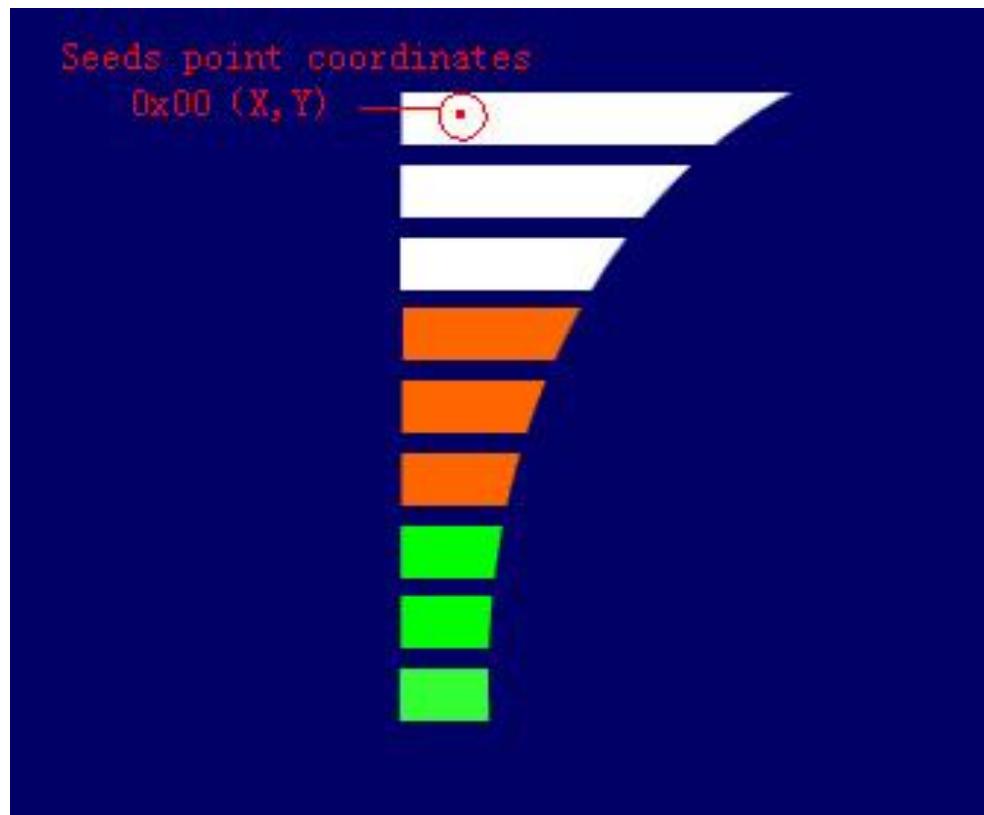
0x0006 Cut and paste of image area



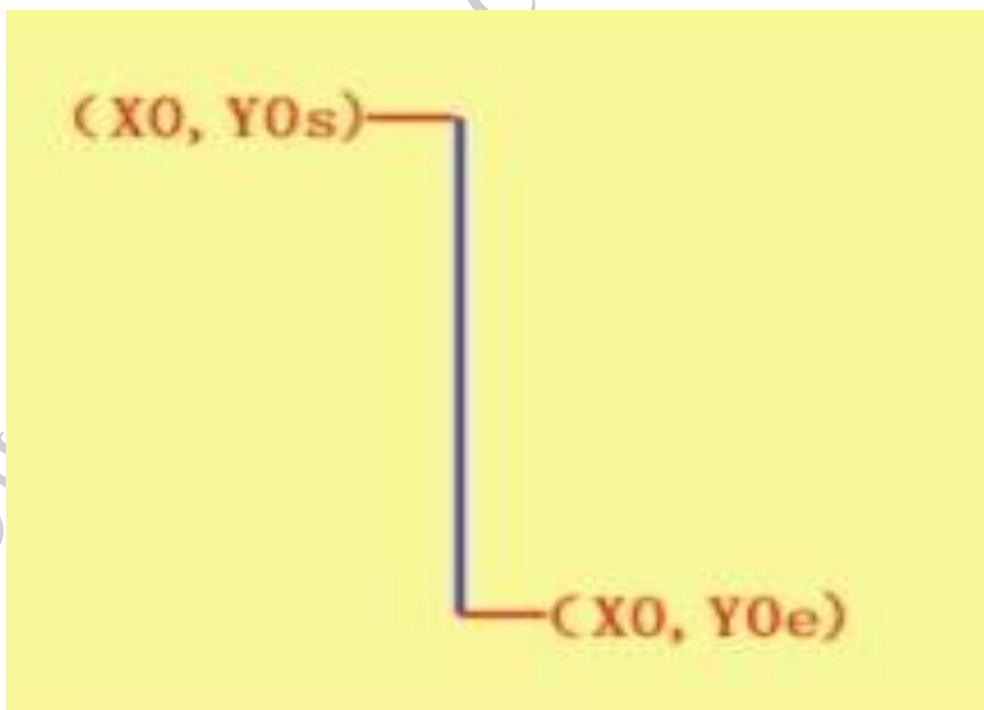
0x0007 ICON display



0x0008 Area filling



0x0009 Vertical line



5.3.3 Tabulate Display(0x22)

Add.		Definition	Data Length	Description
0x00		0x5A22	2	
0x02		*SP	2	Variable description pointer, 0xFFFF indicates loaded by the config. file
0x04		0x000C	2	
0x06	0x00	*VP	2	Data content pointer of form, which is the 1 st data array address of TAB[Row_Num][Col_Num][Unit_Data_Num]
0x08	0x01:H	TAB_X_Num	1	Column Number, 0x01-0xFF
0x09	0x01:L	TAB_Y_Num	1	Row Number, 0x01-0xFF
0x0A	0x02:H	TAB_X_Start	1	Starting column of form, 0x00-0xFF
0x0B	0x02:L	TAB_Y_Start	1	Starting row of form, 0x00-0xFF
0x0C	0x03:H	Unit_Data_Num	1	<ul style="list-style-type: none"> ➤ 0x01-0x7f All cells share the same data length ➤ Data length for single cell (Counted in WORD) ➤ 0x00 The data length for cells in different columns determined by the variable storage space which is pointed by *VP <p>When Unit_Data_Num=0x00, the address for the content of the table will be shifted (TAB_X_Num /2) bytes (Upward round) 0x1000-0x1003 sequentially stored data length information in Row 1-6, therein the low byte of 0x1003 is not occupied, therefore the content of the table will be stored in 0x1004 and thereafter</p>
0x0D	0x03:L	Encode_Mode	1	Text encoding: 0=8bit coding 1=GB2312 internal code 2=GBK 3=BIG5 4=SJIS 5=UNICODE
0x0E	0x04	Xs Ys Xe Ye	8	Definition of form display area(Top left corner coordinate to bottom left corner coordinate) Form display will always be from the upper left corner; The cross- border will end the display
0x16	0x08	Color_line	2	Form boarder line color
0x18	0x09	Color_text	2	Form text color
0x1A	0x0A:H	Font0_ID	1	Font location of the ASCII in coding 0x01-0x04
0x1B	0x0A:L	Font1_ID	1	Font location of coding 0x00&0x05, and non-ASCII in coding 0x01-0x04
0x1C	0x0B:H	Font_X_Dots	1	Dot-matrix of font in X direction (0x01-0x04 mode, ASCII character X is by X/2 calculation)
0x1D	0x0B:L	Font_Y_Dots	1	Dot number in Y direction of font
0x1E	0x0C:H	TAB_X_Adj_Mod	1	When TAB_X_Start is NOT 0, table title can be hidden: 0x00= Hide the first column of the table 0x01= Show the first column
0x1F	0x0C:L	TAB_Y_Adj_Mod	1	When TAB_X_Start is NOT 0, table title can be hidden: 0x00= Hide the first row of the table 0x01= Show the first row

When the actual content of form is shorter than the prescribed length by Unit_Data_Num, write 0xFFFF as the terminator for the cell text.

For special large form, the positioning and drag of the form could be achieved by modifying TAB_X_Num, TAB_Y_Start value through the touch screen.



5.3.4 Exclusive functions for special applications

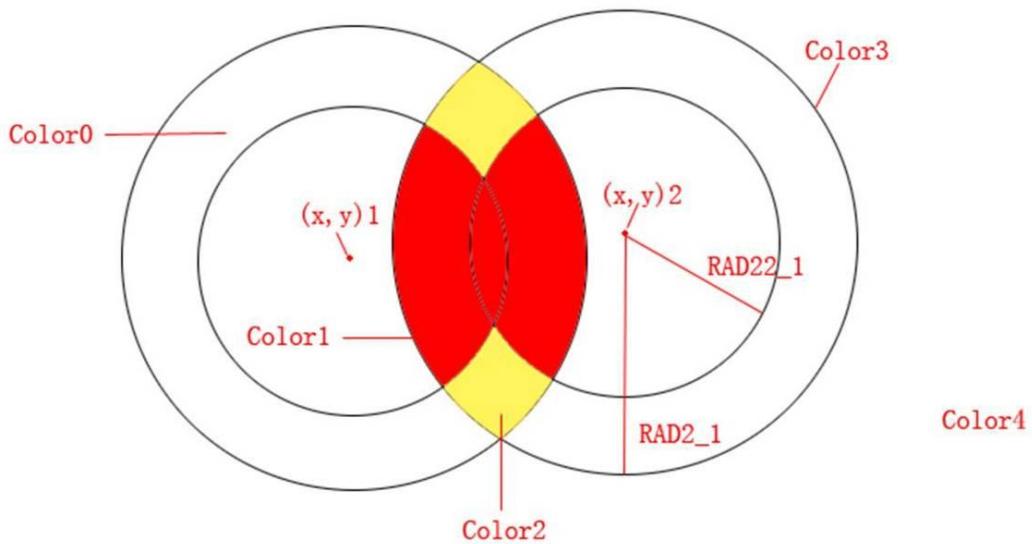
Add.	Definition	Data Length	Description
0x00		0x5A23	2
0x02		*SP	2 Variables description pointer, 0xFFFF indicates it loaded by config. file
0x04		0x0001	2
0x06	0x00	*VP	2 Variable data pointer
0x08			24 Write 0x00

Data format for Variables

Add.	Definition	Description
VP	CMD	Command
VP+1	Data_Pack_Num_Max	Maximum data; automatically terminated when ending condition is met
VP+2	DATA_Pack	

Data format for Commands

Command (CMD)	Operation	Definition of data format for drawing basic shapes (Relative address and length are counted by WORD)			
		Comparative Add.	Length	Definition	Description
0x0001	Fill the overlapped area for multiple circle area	0x00	1	Color0	Color for "Safe Zone"
		0x01	1	Color1	Color for normal overlapped area (Overlapped once)
		0x02	1	Color2	Color for High-Risk overlapped area (Overlapped twice or more)
		0x03	1	Color3	Color for the outline of the circle area
		0x04	1	Color4	Color for evasion (e.g filtered background color for text or Grid)
		0x05	4	Disp_Area	Display area, no display if element is out of boundary
		0x09+4*n	2	(x,y)n	If the high byte for x coordinate is 0xFF indicates neglected for the multiple circle area at the same center.
		0x0B+4*n	1	RADn_1	The radius of No. N bigger circle area
		0x0C+4*n	1	RAD2n_2	The radius of No. N smaller circle area.



6 Development Steps

Step 1: Variable Scheme

Variable Scheme basically follows two basic principles:

- Data variables should be arranged in continuous addresses for convenience to write/read;
- Address of parameter description variables and data variables should be separate and no cross-connect.

Data length of variable data in DWIN DGUS V2.0				
Category	Command	Function	Length of the variable data(Word)	
			Data variables	Description
Touch variables	0x00	Variable data input	int(1)/long int(2)	No require
	0x01	Pop-up menu selection	int(1)	No require
	0x02	Incremental variables adjustment	int(1)/unsigned char(0.5)	No require
	0x03	Slider adjustment	int(1)/unsigned char(0.5)	No require
	0x04	RTC setting	No require	No require
	0x05	Button result value for return	int(1)/unsigned char(0.5)	No require
	0x06	Text entry	Variable	No require
	0x07	Firmware parameter configuration	Variable	No require
Data variables	0x00	Variable icon display	int(1)	8
	0x01	Animation icons display	long int(2)	10
	0x02	Scale indicator	int(1)	9
	0x03	WordArt variable display	int(1)/long int(2)	7
	0x04	Images animation	No Require	4
	0x05	Icon rotation instructions	int(1)/unsigned char(0.5)	12
	0x10	Data variable display	int(1)/long int(2)	13
	0x11	Text display	Variable, user-defined	13
	0x12	RTC display	No require	13
	0x20	Curve display	No require	9
	0x21	Basic graphic display	Variable, user-defined	7
	0x22	Tabulate display	Variable, user-defined	12
	0x23	Exclusive functions for special applications	Variable, user-defined	1

Int(1) indicates that variable is integer that sized in one word;
Long int (2) indicates that variable is a long integer that sized in two words.



Variable behavior	Memory address	Length(byte)
Voltage	0000	2
Current	0001	2
Power	0002	2
Operating power	0003	2
Operating speed	0004	2
Output torque	0005	2

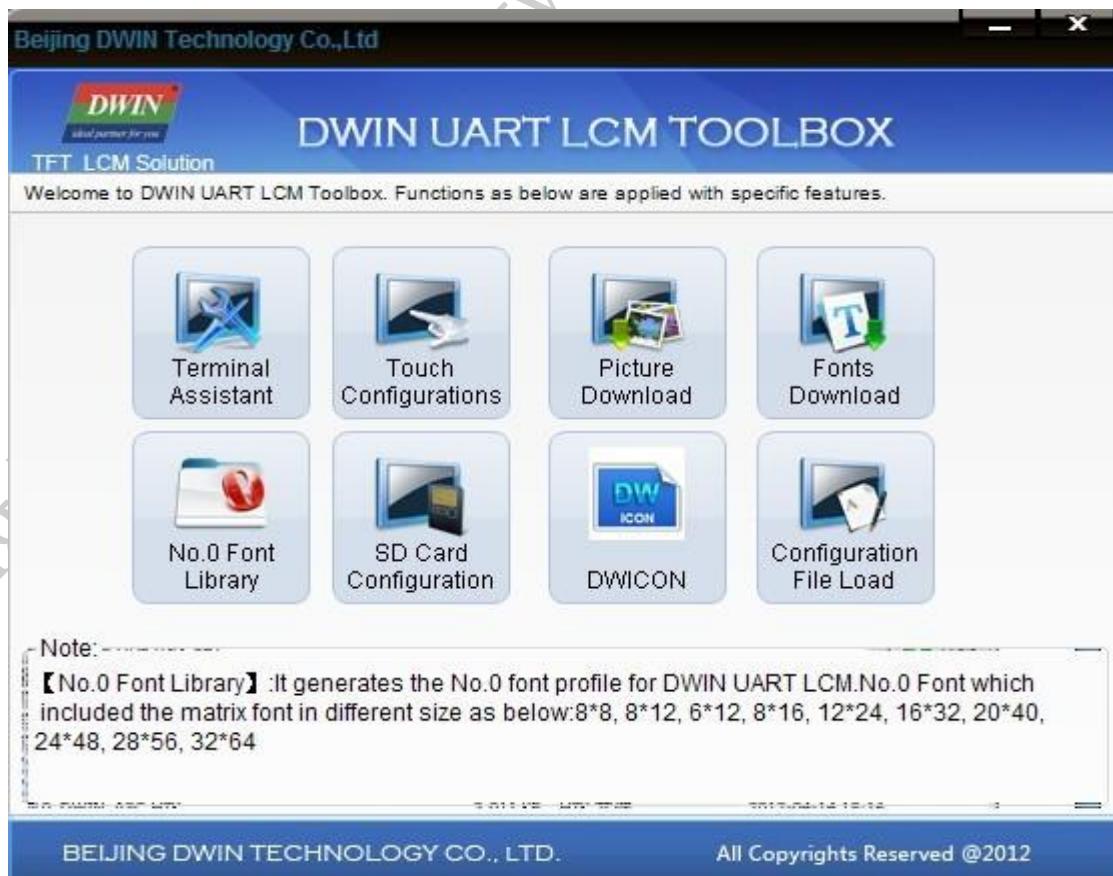
Step 2: Interface Design

Use Photoshop (or other softwares) to design the interface and related items (icon, font etc.). In process, please select 65K color in palette system for ensuring the identical with final display effects.



Step 3: Interface Configuration

Use DWIN software to configure the interface to generate the touch configuration file and variable configuration file.



Step 4: Debug Test

Download the config. file, images, font, icon library etc. to DWIN module via SD card for debugging and adjustment (Step 2&3); Connecting with user's host by serial port to process up data transmission.



Step5: Archive Version

Config. file, images, font, icons library etc. that involved with module works were saved in an SD card for mass production.

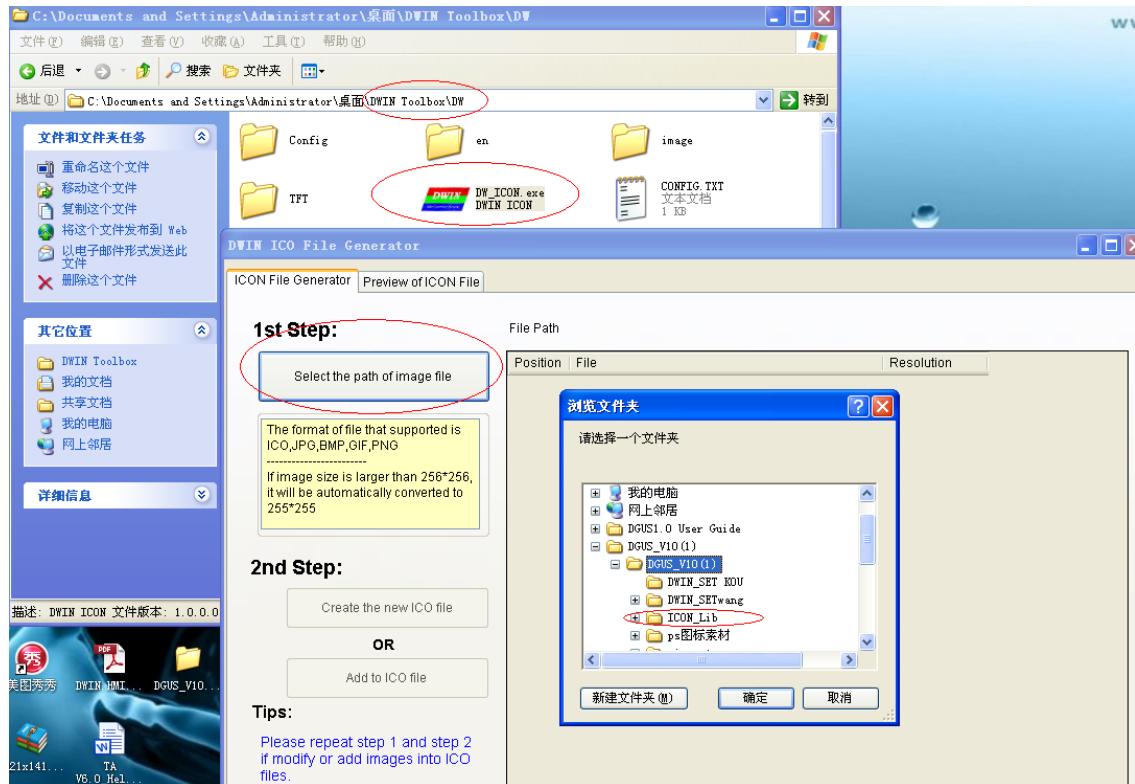


Please see DGUS operation details as below:

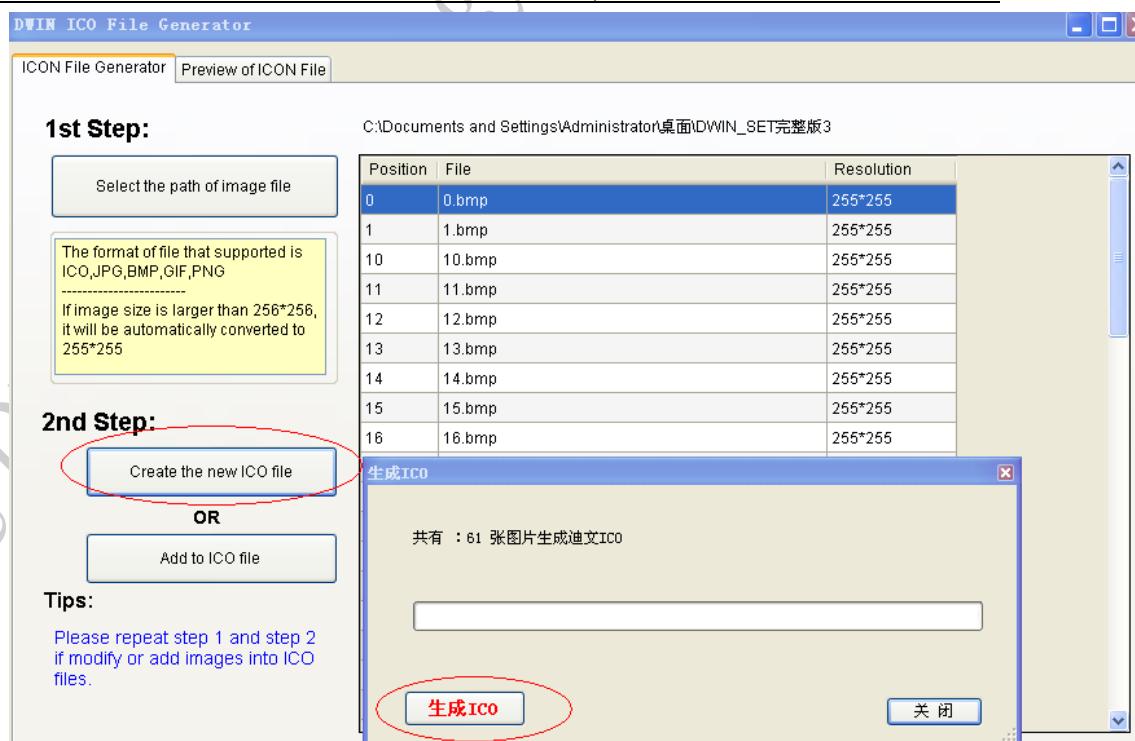
The method of creating icon library:

- Download the pictures and cut to the appropriate icon size by PS or other drawing software, such as 80 * 80. Save it to a folder and naming it, such as "ICON"
- Open DWIN Toolbox-ICON Library,

Step 1: Select the picture path

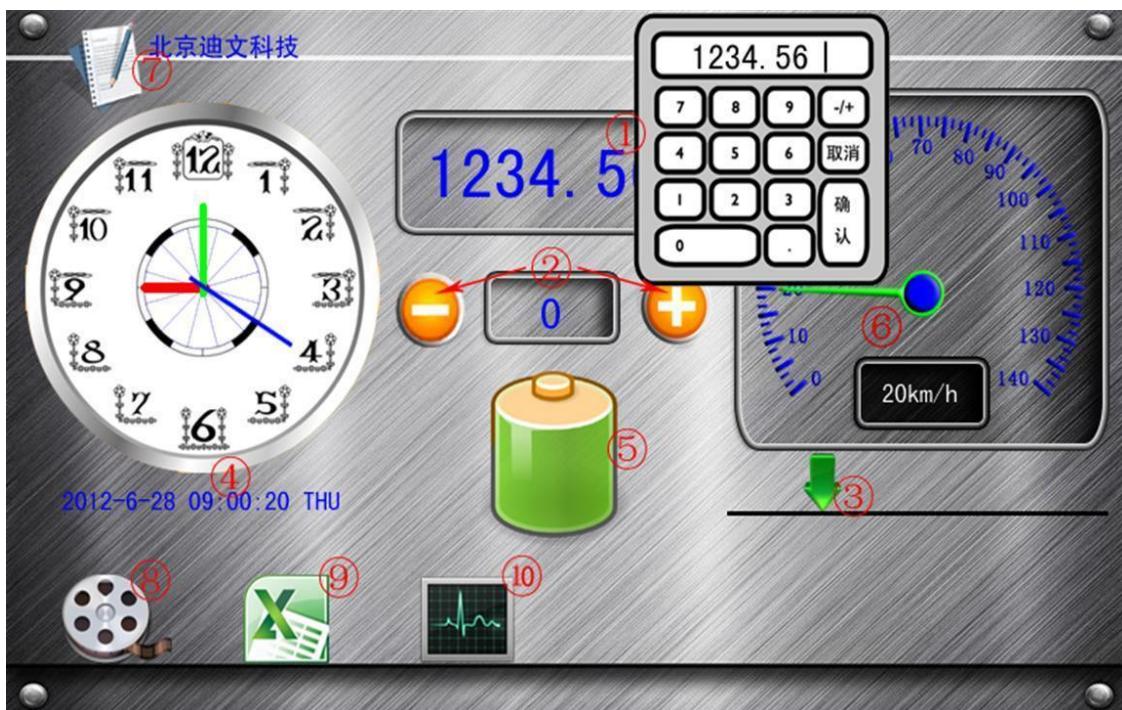


Step 2: Create a new ICO file which MUST BE named 41, saved it to a folder such as ICO.1.



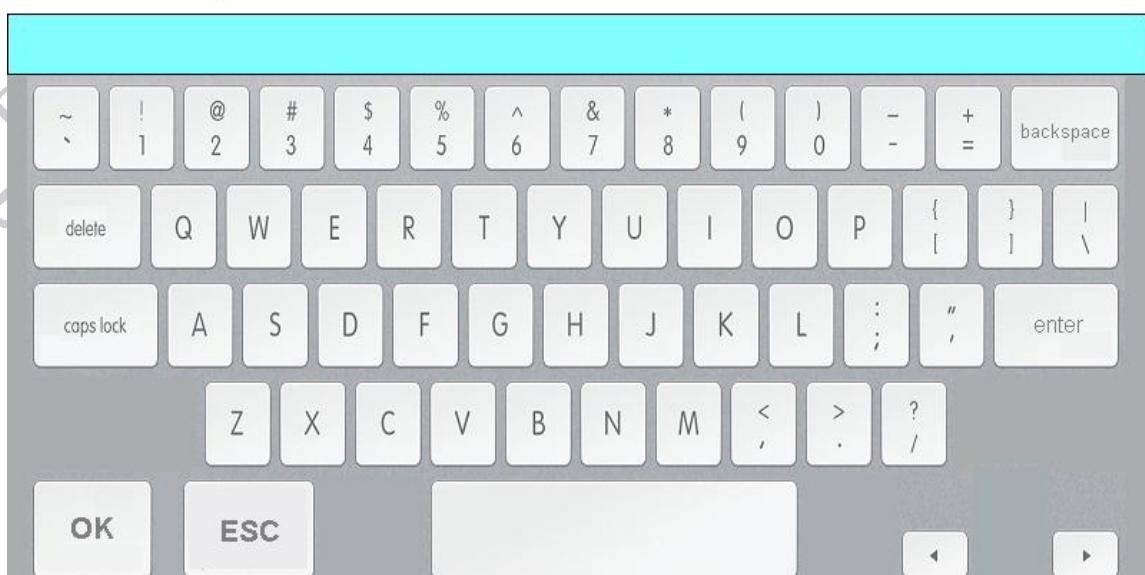
(Rename the icon library from 41. The second icon library named 42 or 43. The size of each icon library is 256K, if exceed 256K, the result in an icon library 42 can be displayed but the 41 icon library cannot displayed).
Display text using the 0_DWIN_ASC.HZK font library in the DGUS tool.

Appendix 1 DGUS key functions preview



- ① **Variable input and display:** 0xFE00 Variable Data Input(pop-up keyboard which is not on the current page); 0x5A10 Data Variable Display.
- ② **Variable adjustment("+" ,"-") and display:** 0xFE02 Adjustment of the Variable Value; 0x5A10 Data Variable Display.
- ③ **Variable adjustment (drag with following slider) :** 0xFE03 Slider Adjustment ; 0x5A02 Slider Scale Indicator.
- ④ **Time set and display: 0xFE04 RTC Set(similar to 0xFE00 Variable Data Input) ; 0x5A12 RTC Display (Two kinds of time display: dial and text).**
- ⑤ **Scale bar, status bar :0x5A00 Variable Icon Display (Display different icons corresponding to the different variable values); 0x5A01 Animation Icon Display(When the variable is a specified value, display multiple icons circulation animation in proper order).**
- ⑥ **Dashboard :0x5A05 Icon Rotation Instructions.**
- ⑦ **Text input and display: 0xFE06 Text Entry ;0x5A11 Text display.**

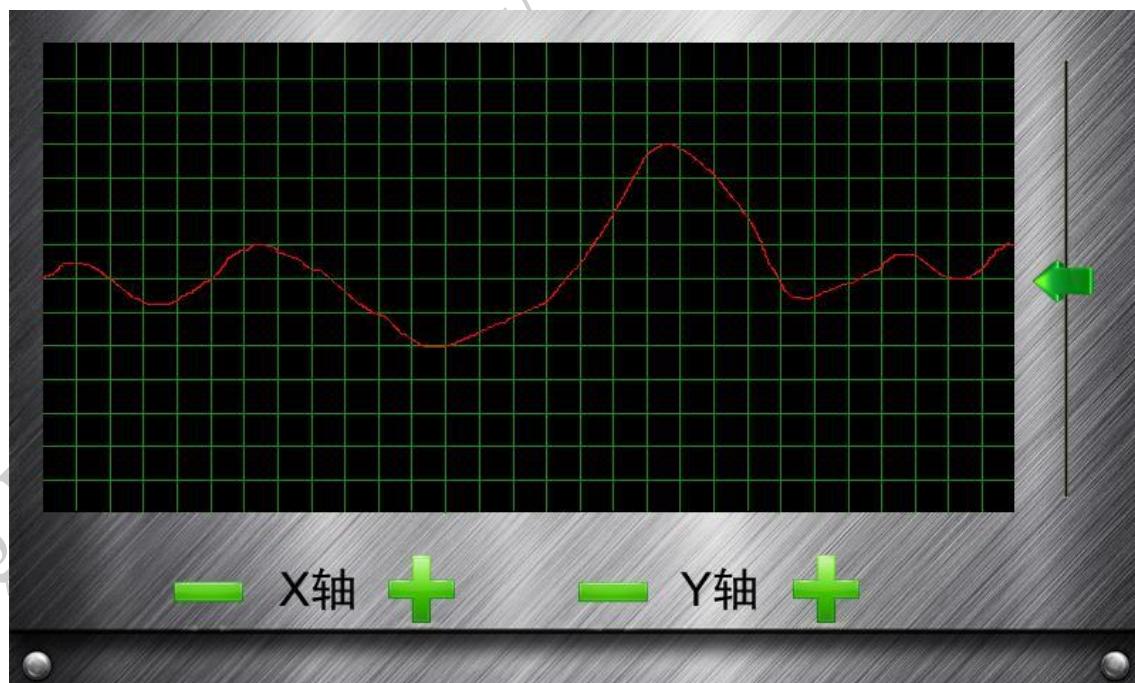
Beijing DWIN Technology Co., Ltd. is a leading global company with cutting-edge technology mainly focuses on HMI (TFT Serial LCD Module) researching, manufacturing and sales. |



- ⑧ Boot animation, electronic album: 0x5A04 Images Animation.
⑨ Tabulate display, drag and print: 0x5A22 Tabulate Display ;0xFE03 Slider Adjustment ; 0x5A02 Slider Scale Indicator ;0xFE07 Firmware parameter Configuration.

尺寸	比例	型号
7.0	16:9	DMT80480C070_02W
7.0	16:9	DMT80480T070_06W
7.0	16:9	DMT80480S070_06W
7.0	16:9	DMT10600T070_01W
7.0	16:9	DMT80480T070_18WT

- ⑩ Display and adjust the curve: 0x5A20 Curve Display (Supports up to eight channels simultaneously receive data and show the real-time curve); 0xFE03 Slider Adjustment ;0x5A02 Slider scale indicator; 0xFE02 Incremental Adjustment. Adjustment of the curve scaling and central axis position can be achieved without code interference.



Appendix 2 Application Examples

(1) Display, entry and adjustment of parameters

0x5A10 Data Variable Display , optional font, font size, font color, and data unit.



0xFE00 Variable Data Input, click on the touch region, display the pop-up keyboard and the input process with flashing cursor.



0xFE02 Incremental Adjustment, click on the buttons to adjust the data in the way of "--" & "++".

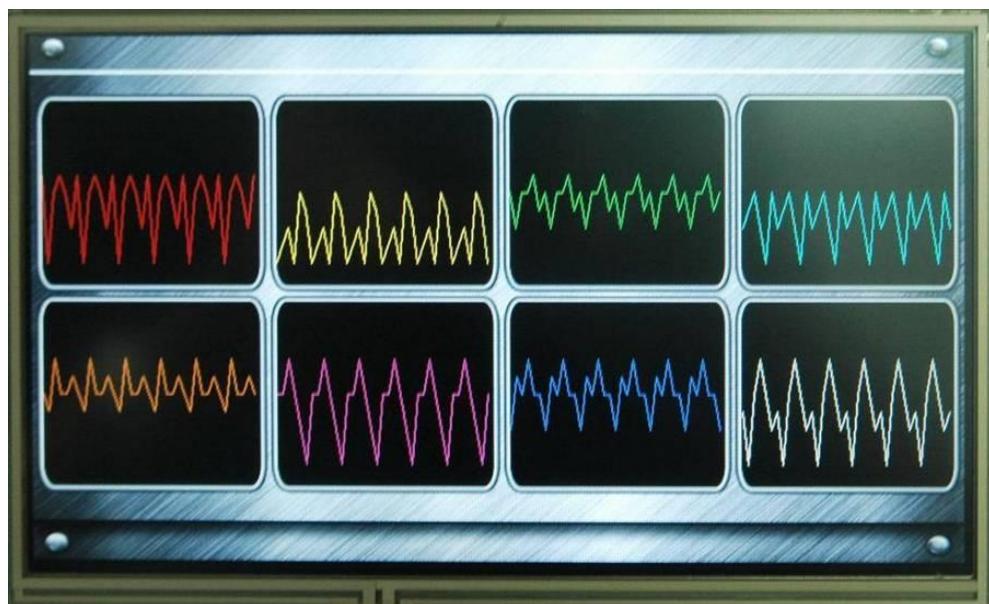


0xFE03 Slider Adjustment, press the touch region 0.5 second and then drag to adjust the data.

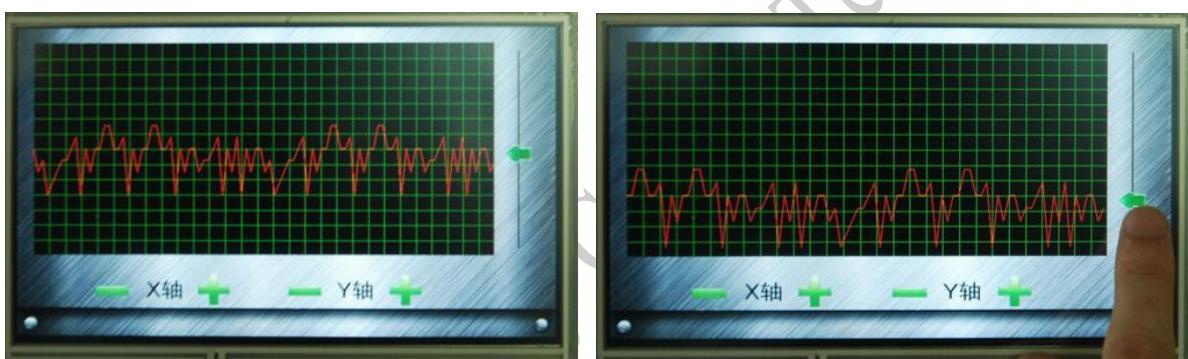


(2)Display and adjust the curve

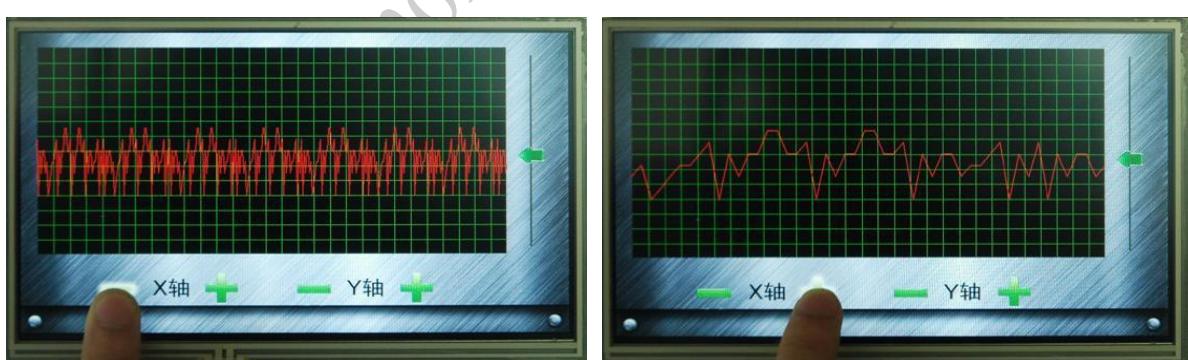
0x5A20 Curve Display, support up to 8 channels to receive data and real-time display the curve at the same time.



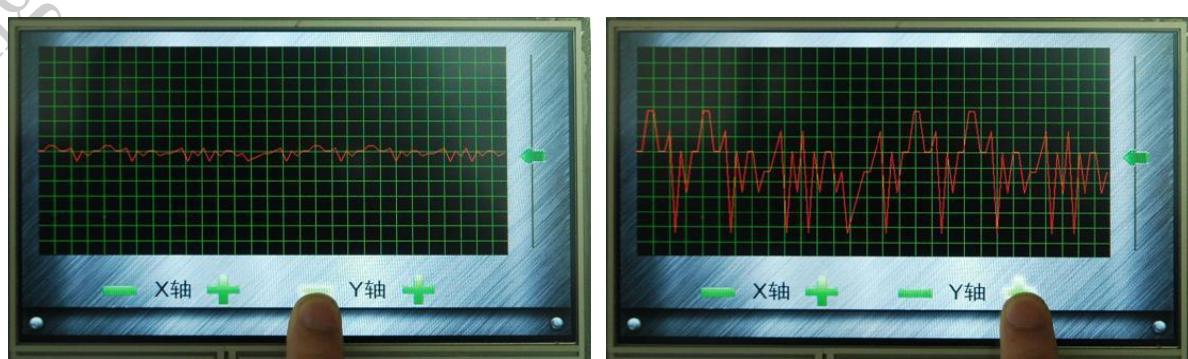
0xFE03 Slider Adjustment with 0x5A02 Slider Scale Indicator, control the position of the curve center axis.



0xFE02 Incremental Adjustment, adjust X-axis scaling of the curve.



0xFE02 Incremental Adjustment , adjust Y-axis scaling of the curve.



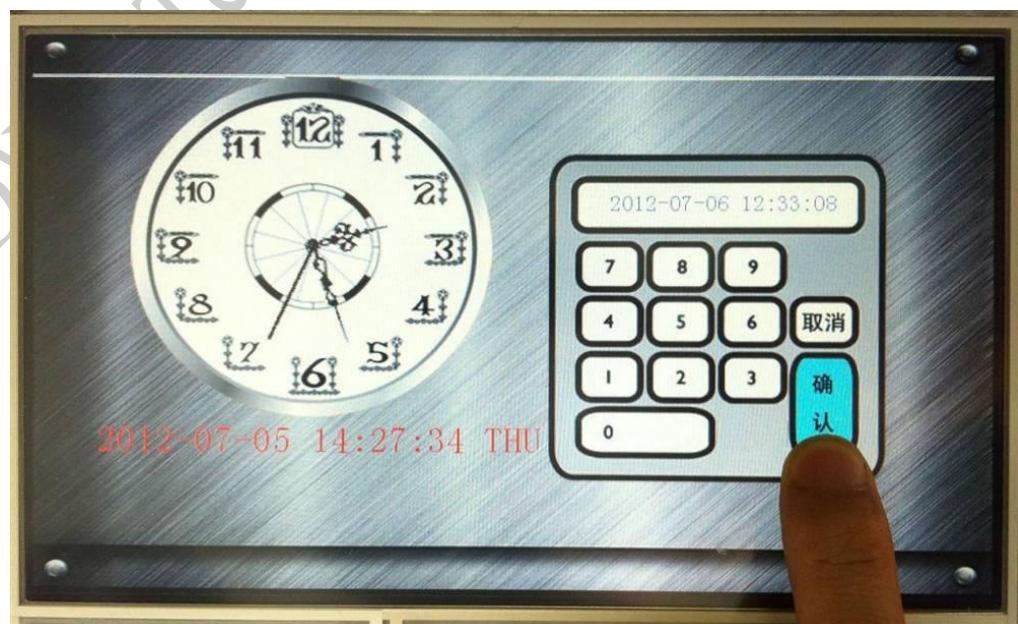
(3)Time display and set

0x5A12 RTC display, tow kinds of time display: text and dial.

Text time display with user-defined display format, dial time display with user-designed dial and pointers.



0xFE04 RTC Setting, click on the touch area, display the pop-up setting keyboard and the input process with the flashing cursor.



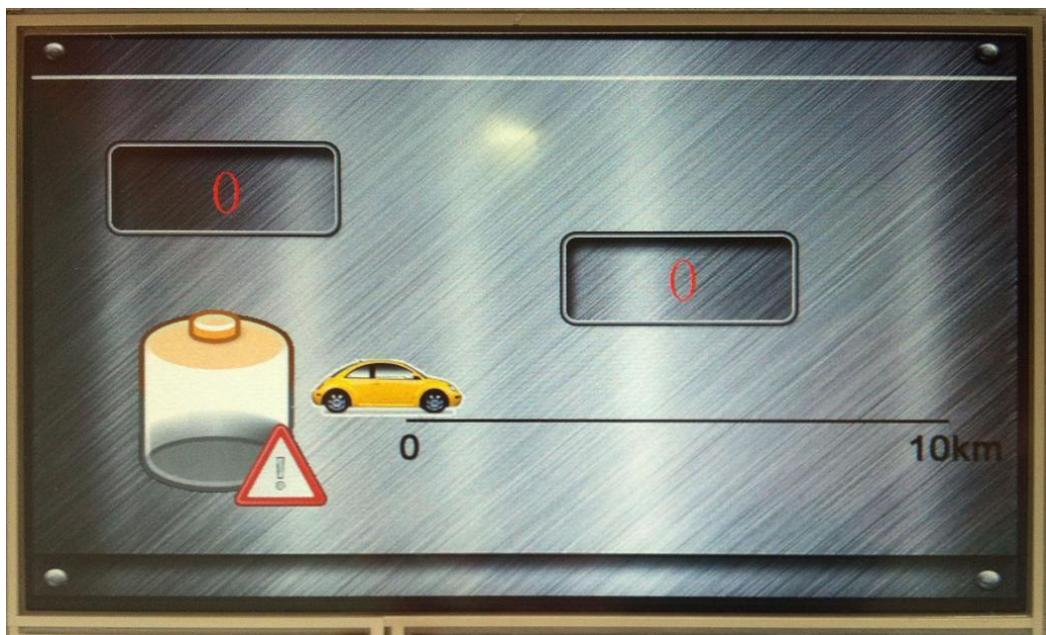
(4) Progress bar display

The progress bar is achieved by mainly two ways: variable icon to display and slider scale to indicate.

1) **0x5A00 Variable Icon Display:** Display different icons corresponding to different variable values.

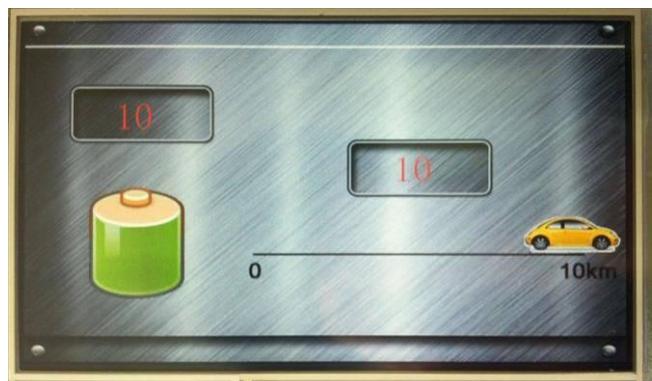
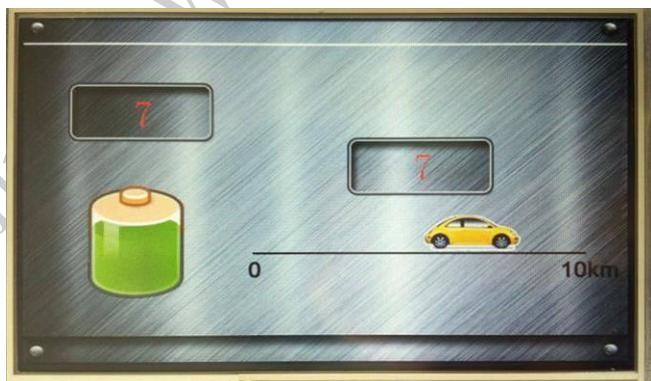
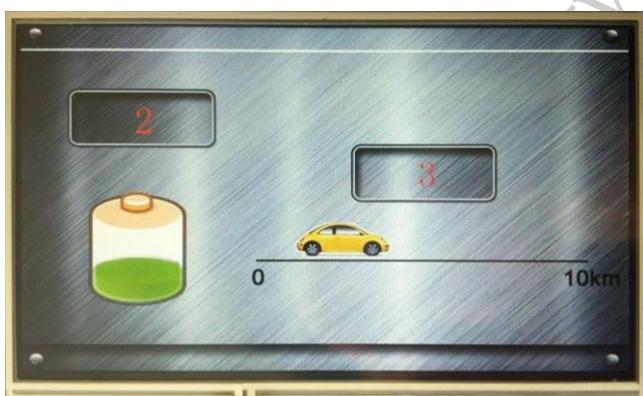
2) **0x5A02 Slider Scale Indicator:** The slider shows in different locations corresponding to different variable values.

In following case the battery indicator is variable icon to display, mileage display is slider scale. Both variable initial values are 0.



Battery indicator: display different battery icons corresponding to different power values.

The mileage display: The car icon is displayed in different locations corresponding to different mileage values.



(5)Control switch, display animation of operating status

Set **0xFE02 Incremental Adjustment** button, adjustment in step of 1, range 0-1, cycle when over the range. The variable value switches between 0 and 1 by pressing the button. Control the equipment operating status by querying the value of the variable.

0x5A01 Animation icon Display, When the variable is 0, fixed display the no-electricity icon.



Touch the switch, the variable value changes to 1 and the device start running. The following electricity icons display in turns and achieve the circulation charging animation.



Touch the switch again, the variable value changes to 0 and the device stop with fixed no-electricity icon.



(6) Dashboard display

0x5A05 ICON Rotation Instructions.

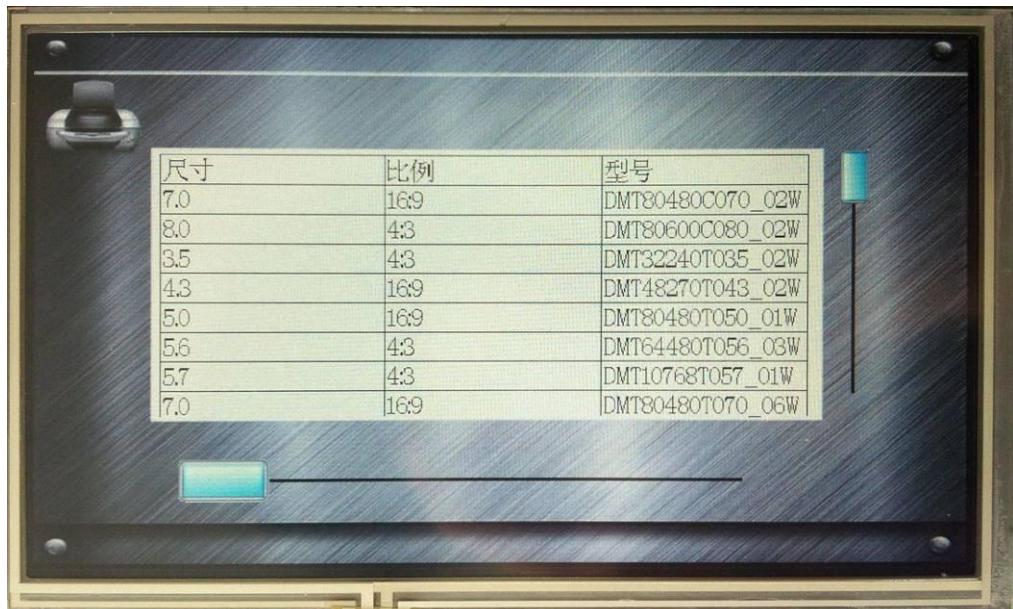


The dial pointer icon displays with different angles of rotation corresponding to different values of a variable.

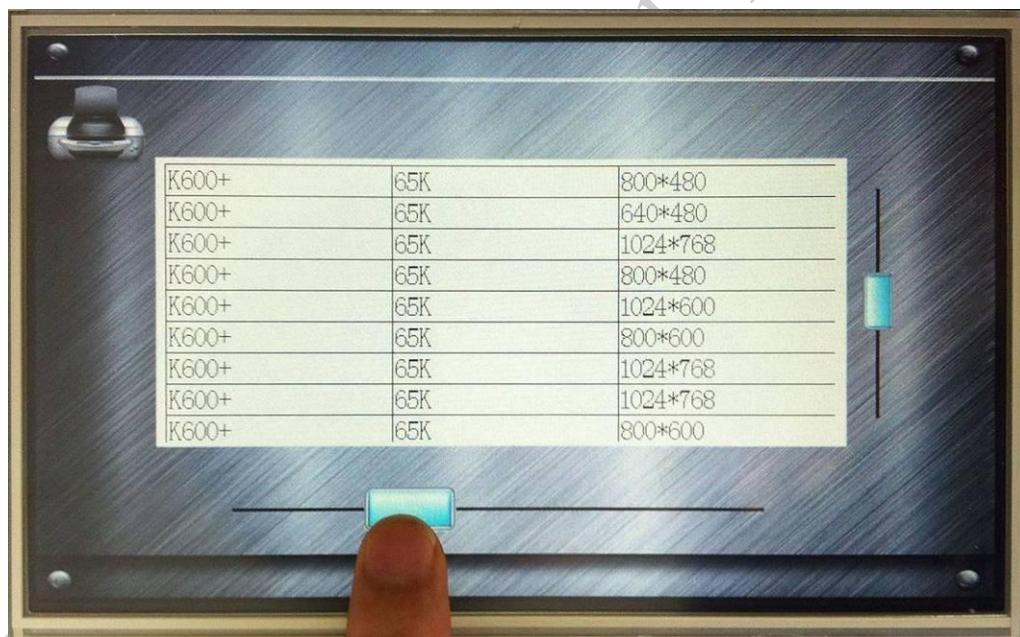


(7)Display and print a form

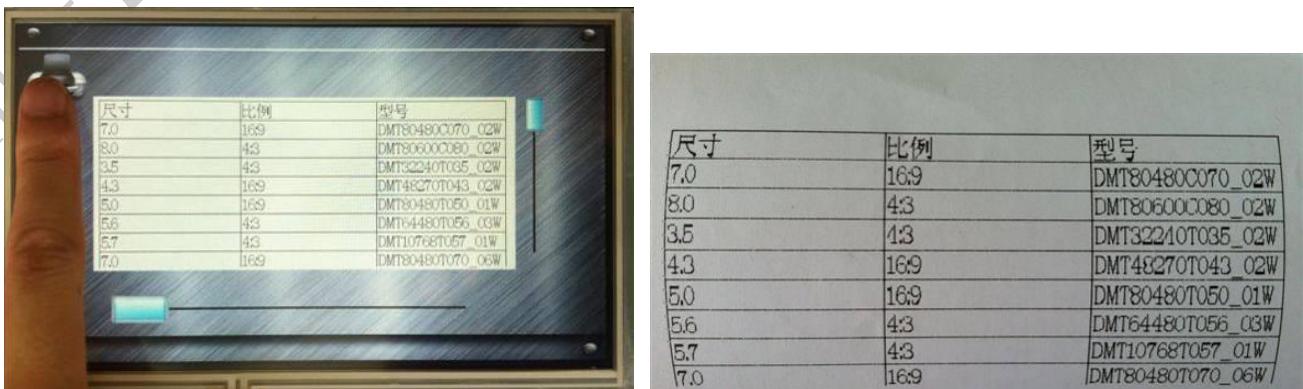
0x5A22 Tabulate Display, content to display can be generated by the variable initializing configuration file.



If the form is too large to display completely, change the form display area by **0xFE03 Slider Adjustment** with **0x5A02 Slider Scale Indicator**.



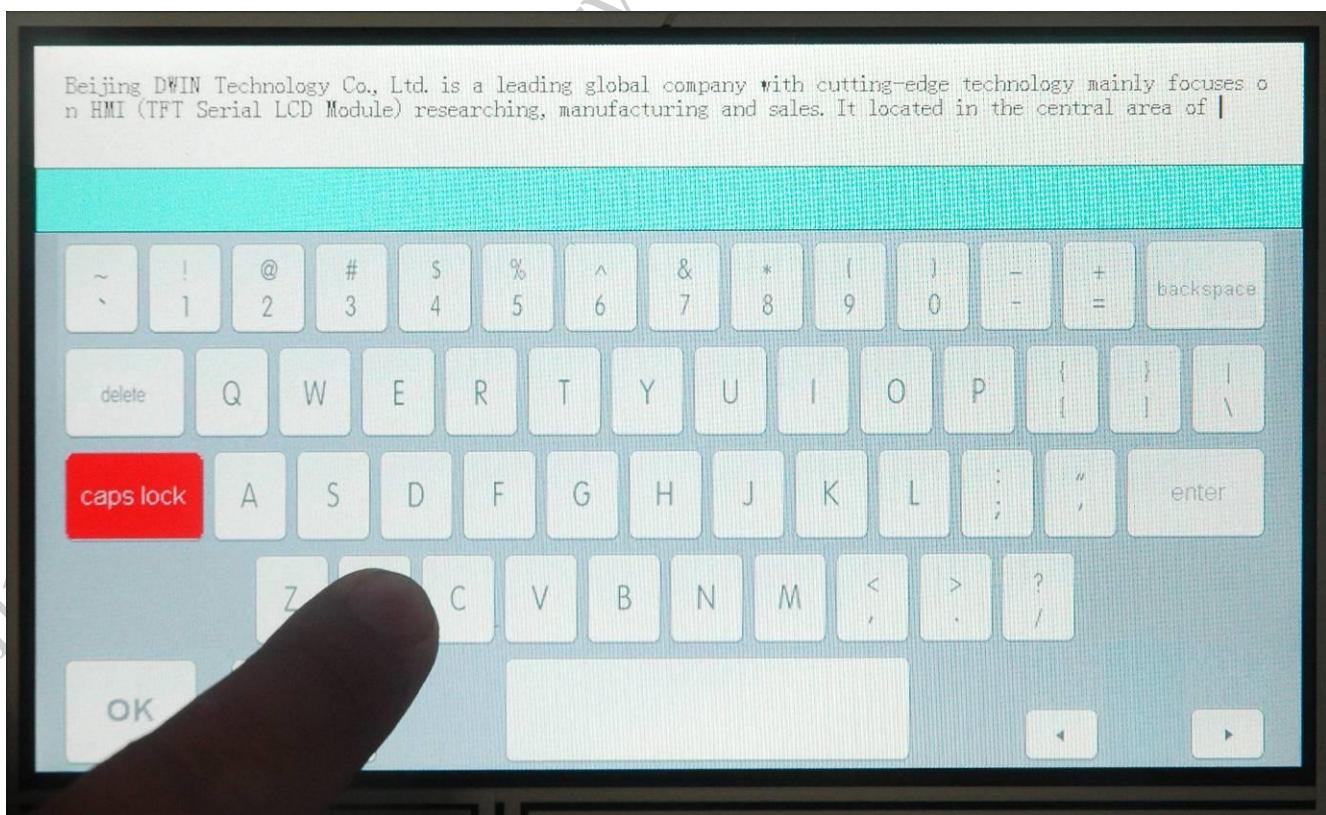
Print specified area displaying on the screen by **0xFE07 Firmware Parameter Configuration** touch button, with the user's MCU connecting to a serial printer.



(8)Text display and input

0x5A11Text Display, 0xFE06 ASCII Text Entry, support ASCII text and GBK Chinese character text display and input.

Click on the touch area, pop-up text entry keyboard.



(9) Electronic Album (boot animation)

0x5A04 Images Animation, multiple full-screen pictures switch in order, with user-defined time interval of the switching.

By software for video capture, with this function, you can make video for the boot animation.



(10) Multi-channel switch display

Four-channel switch control as example, set four **0xFE02 Incremental Adjustment** button and four **0x5A00 Variable Icon display** pointing to four different variables, adjustment in step of 1, range 0-1, cycle when over the range. The red icon corresponds to value 0, the green icon corresponds to value 1.



Click on the switch button each time, the corresponding variable value switches between 0 and 1, the corresponding variable icon switches between the red and the green. Control the state of the switches by querying the values of the corresponding variables.

