

T5L_DGUSII Application Development Guide

Version 2.2

2022/5/12

Contents

| | |
|--|-----|
| Chapter 1 Quick Start..... | 4 |
| 1.1 Naming Rule..... | 4 |
| 1.2 Interface Definition and Wiring..... | 5 |
| 1.3 Power Supply..... | 8 |
| 1.4 Driver Installation..... | 8 |
| 1.5 Software Installation..... | 10 |
| Chapter 2 Development System..... | 11 |
| 2.1 T5L ASIC..... | 11 |
| 2.2 T5L_DGUS II Development System..... | 14 |
| 2.3 Features..... | 16 |
| 2.4 Software Process..... | 17 |
| 2.5 FLASH Memory Allocation..... | 18 |
| 2.6 RAM Memory Variable Address Space..... | 19 |
| 2.7 RAM Description Pointer Space..... | 21 |
| Chapter 3 SD Card Interface..... | 23 |
| 3.1 MicroSD (TF) Card Format..... | 23 |
| 3.2 SD Card Interface Download Method..... | 24 |
| 3.3 Download File..... | 25 |
| Chapter 4 UART2 Serial Communication Protocol..... | 42 |
| 4.1 Data Frame Structure..... | 42 |
| 4.2 UART2 Debug Interface Instruction..... | 41 |
| Chapter 5 System Variable Interface..... | 46 |
| 5.1 System Variable Interface..... | 46 |
| 5.2 Network Interface..... | 60 |
| Chapter 6 Touch Variable Configuration (13.BIN)..... | 66 |
| 6.1 List of Touch Variables..... | 67 |
| 6.2 Variable Data Input..... | 68 |
| 6.3 Popup Menu..... | 71 |
| 6.4 Incremental Adjustment..... | 73 |
| 6.5 Drag Adjustment..... | 75 |
| 6.6 Return Key Code..... | 77 |
| 6.7 Text Input..... | 79 |
| 6.8 Synchrodata Return..... | 87 |
| 6.9 Rotation Adjustment..... | 90 |
| 6.10 Sliding Adjustment..... | 92 |
| 6.11 Page Sliding..... | 93 |
| 6.12 Slide Icon Selection..... | 95 |
| Chapter 7 Display Variable Configuration..... | 97 |
| 7.1 List of Display Variables..... | 99 |
| 7.2 Variables Icon..... | 102 |
| 7.3 Animation Icon..... | 105 |
| 7.4 Slider Display..... | 108 |
| 7.5 Artistic Variables..... | 110 |
| 7.6 Image Animation..... | 112 |
| 7.7 Icon Rotation..... | 113 |
| 7.8 Bit Icon..... | 116 |

| | |
|--|-----|
| 7.9 Icon Page Tran..... | 119 |
| 7.10 Icon Overlay..... | 120 |
| 7.11 Batch Va Icon..... | 122 |
| 7.12 Data Variables..... | 124 |
| 7.13 Text Display..... | 129 |
| 7.14 RTC Display..... | 134 |
| 7.15 HEX Data..... | 138 |
| 7.16 Text Scroll..... | 140 |
| 7.17 Data Window..... | 142 |
| 7.18 DGUSII Text(Text Display Without Aliasing)..... | 144 |
| 7.19 Roll Character..... | 148 |
| 7.20 GTF Icon Font..... | 143 |
| 7.21 Real-Time Curves..... | 156 |
| 7.23 Zone Scrolling..... | 190 |
| 7.22 Basic Graphics..... | 162 |
| 7.24 QR Code..... | 188 |
| 7.25 Brightness..... | 190 |
| 7.26 Data Transmit..... | 191 |
| 7.27 Camera..... | 192 |
| Chapter 8 DGUS Basic Functions..... | 197 |
| 8.1 Import..... | 197 |
| 8.2 Edit tool..... | 197 |
| Chapter 9 Revision Record..... | 197 |
| Appendix 1:FAQs of T5L_DGUSII Screen..... | 199 |

Chapter 1 Quick Start

1.1 Naming Rule

| Product Line Code | DM | DWIN Smart LCM Product Line | | | | |
|-------------------------------|----------------------|--|--------------------------|----------------------------|----------------|--------------|
| Color Code | 1 Character | T=65K color(16-bit) G=16.7M/262K color(18-bit/24-bit) | | | | |
| | | 24240=240*240 | 32240=320*240 | 48270=480*272 | 48320=480*320 | |
| Resolution | 5 Digits | 64360=640*360 | 64480=640*480 | 80480=800*480 | 85480=854*480 | |
| | | 80600=800*600 | 10600=1024*600 | 10768=1024*768 | 12480=1280*480 | |
| Classification | 1Character | 12720=1280*720 | 12800=1280*800 | 13768=1364*768/1366*768 | | |
| | | 19108=1920*1080 | 19480=1920*480 | | | |
| Size | 3 Digits | F=COF structure | Y=Beauty grade | L=Simple application grade | | |
| | | C=Commerce grade | T=Industrial grade | K=Medical grade | | |
| Attribute Code | 1 Digit or character | S=Harsh environment grade | Q=Automotive grade | | | |
| | | | | | | |
| Hardware Serial Number | — | 020=2.2inch | 024=2.4inch | 028=2.8inch | 030=3.0inch | 035=3.5inch |
| | | 040=4.0inch | 041=4.1inch | 043=4.3inch | 050=5.0inch | 056=5.6inch |
| LCD Temperature Level | — | 070=7.0inch | 080=8.0inch | 088=8.8inch | 097=9.7inch | 101=10.1inch |
| | | 104=10.4inch | 116=11.6inch | 121=12.1inch | 140=14.0inch | 150=15.0inch |
| TP Category | — | 156=15.6inch | 17.=17.3inch | 185=18.5inch | 215=21.5inch | |
| | | | | | | |
| Custom Tag | 1 Digit or character | Separator | | | | |
| | | | | | | |
| Extended Memory Tag | Z+Number | 0=Basic type | | | | |
| | | 1=Basic type with shell | | | | |
| | F+1 Number | 2=Analogue video processing platform | | | | |
| | | 3=Android platform | | | | |
| | F+1 Number | 4=Digital video processing platform products | | | | |
| | | A=DGUSII kernel product | | | | |
| | 1 Digit | 0-9 to distinguish between different hardware versions | | | | |
| | | | | | | |
| | 1 Character | N=Normal temperature | W=Wide temperature | | | |
| | | | | | | |
| | 1 or 2 Character | N=Without touch panel | TR=Resistive touch panel | TC=Capacitive touch panel | | |
| | | | | | | |
| | Z+Number | Z01-Z99, Standard is blank | | | | |
| | | | | | | |
| | F+1 Number | F0=512MB F1=1GB F2=2GB F3=3GB, Standard is blank | | | | |
| | | | | | | |

1.2 Interface Definition and Wiring

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

① I: INPUT, O: OUTPUT, P: POWER

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

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

CN2: Use 10 Pin 1.0 mm spacing FPC mount;

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

④ UART2 is the communication protocol developed by DGUSII, please refer to the detailed description in Chapter 4;

T5L DGUS products are equipped with extended UART4, and its configuration and communication need to be completed through DWIN OS program. TA mode is only available for UART2 communications at the moment.

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

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

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

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



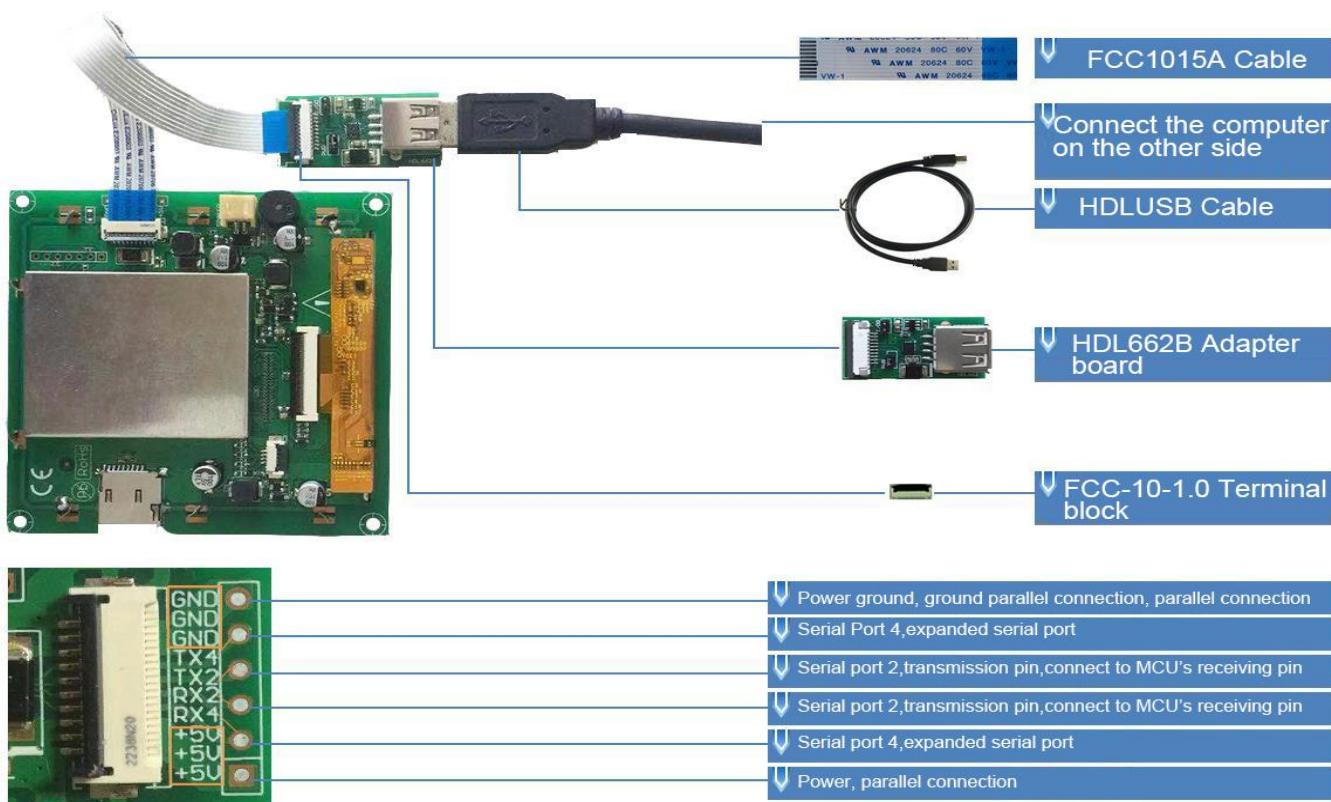
Level selection

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

(1) 10 Pin Interface

One-end of the different side FCC row line is connected to terminal block of the screen, and the other end is connected to the DWIN HDL662B adapter board, both blue side face up.

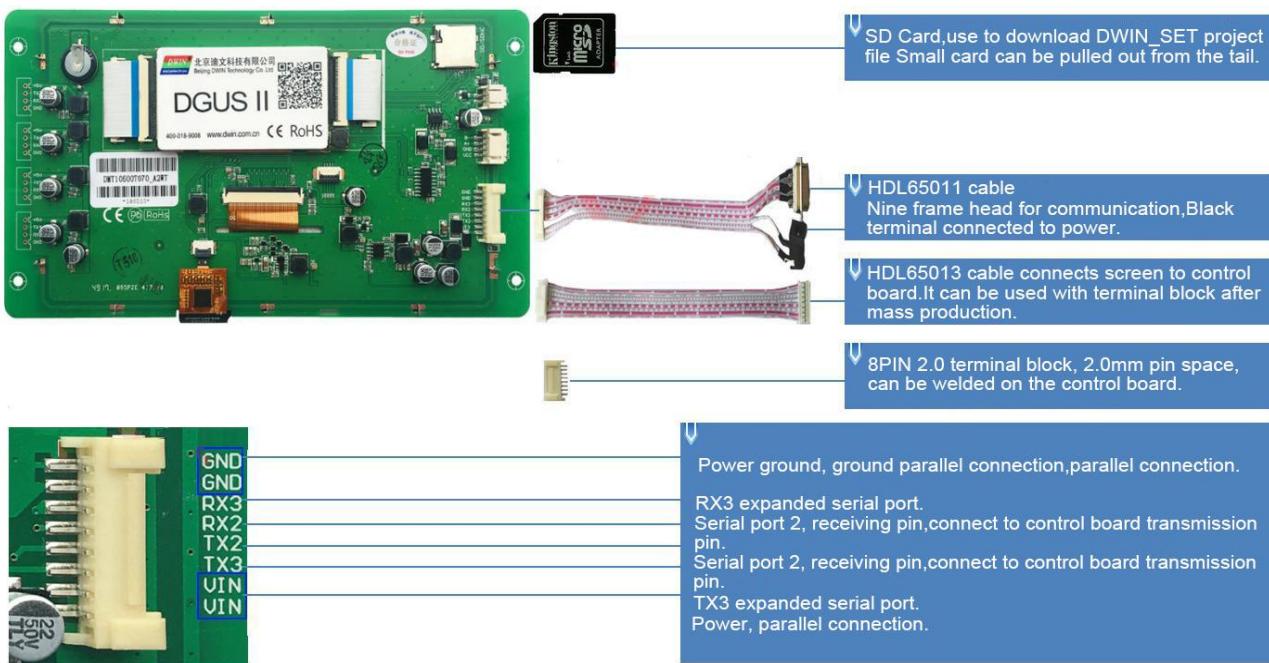
One-end of the double-male USB cable is connected to the adapter board, and the other end is connected to the computer for communication.



10pin interface definition and wiring

(2) 8 Pin Interface

The 8 PIN interface is spaced by 2.0mm.



8pin2.0 interface definition and wiring

1.3 Power Supply

Confirm the voltage and power consumption corresponding to the screen according to the catalogue or datasheet.

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

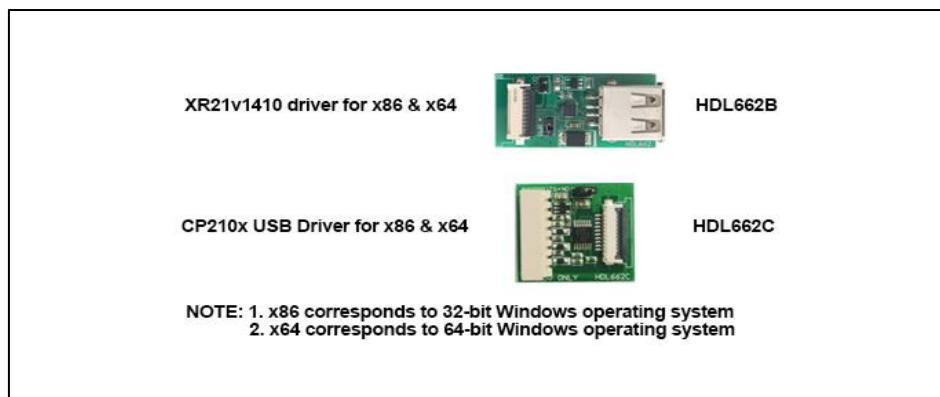
The power supply plays a very important role in the normal display of the screen. If the voltage is too small, the current is unstable, and the power is too low, it may cause abnormal display such as flickering and black screen.

1.4 Driver Installation

1.4.1 Serial Driver

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

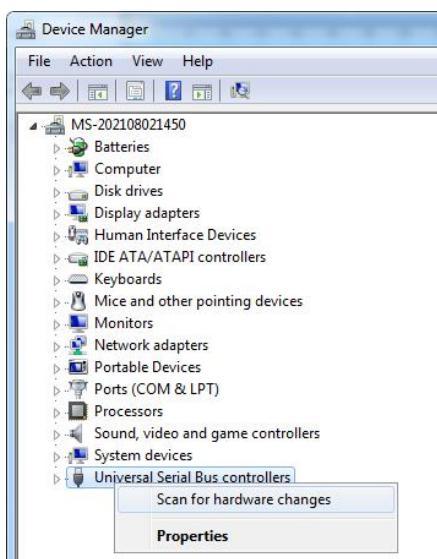
According to the chip type, you can download from DWIN official website or consult customer service to obtain and install the corresponding driver for DGUS screen communication.



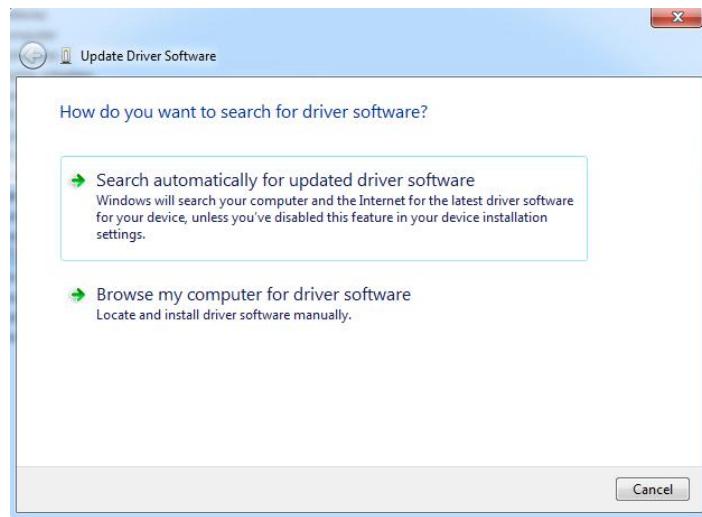
Serial Port Driver

Steps are as follows.

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



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



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



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

1.4.2 Software Operating Environment Driver

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

1.5 Software Installation

Download DGUS V7.624 from our official website <https://www.dwin-global.com/> or get from customer service.

1.5.1 New Project

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

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

Step 2: Download and open DGUS V7.624.

Step 3: Import the image.

Step 4: Configure the Interface.

Step 5: Click save and generate.

Step 6: Project preview.

Chapter 2 Development System

2.1 T5L ASIC

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

(1) Adopts widely-used, mature, and stable 8051 core. The maximum operating frequency of T5L is up to 250MHz, 1T (single instruction cycle) high speed operation.

(2) Separated GUI CPU Core running DGUSII System:

- High-speed display memory, 2.4 GB/S bandwidth, 24bit color display resolution supporting 800*600 (T5L1) or 1366*768 (T5L2).
- 2D hardware acceleration, the decompression speed of JPEG is up to 200fps@1280*800, the UI interface with animation and icons as its main feature is extremely cool and smooth.
- JPEG stores images and icons in the compressed mode, external memory is greatly reduced to low-cost 16Mbytes SPI Flash.
- With a max 400Hz touch point speed, the touch panel supports resistance or capacitance, and its sensitivity can be adjusted.
- High-quality speech compression storage and playback.
- 128KBytes variable storage space, exchanging data with OS CPU core using memory interface, extremely simple to apply.
- 2-way 10bit, 800KHz, DC/DC controller, simplify LED backlight, analog power supply design, and save cost and space.
- 1-way 15bit 32Ksps PWM digital power amplifier driver loudspeaker, save power amplifier cost and achieve high signal-to-noise ratio and sound quality restoration.
- Supporting PC configuration development and simulation, supporting background remote upgrade.

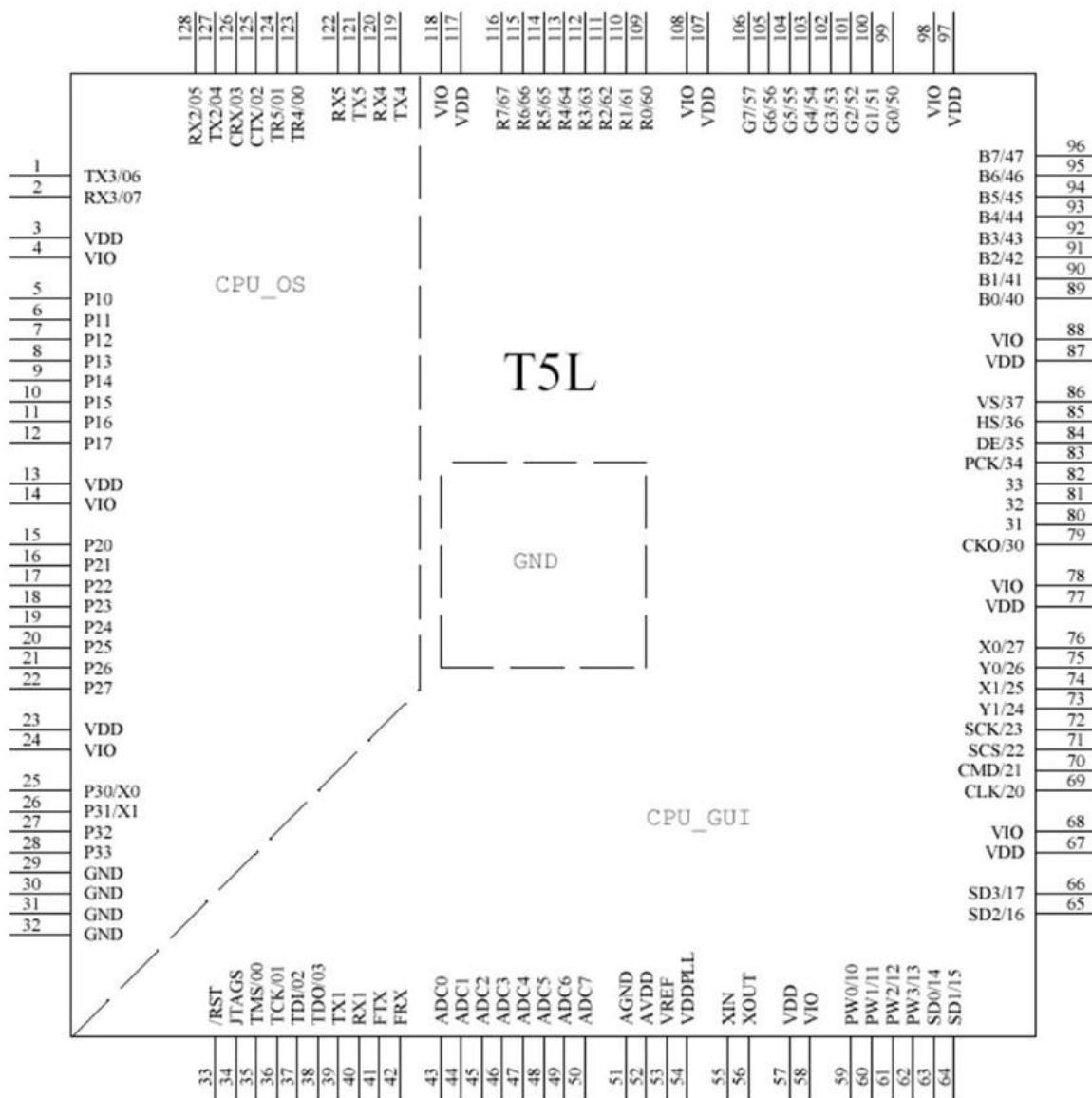
(3) Separated OS CPU core runs user 8051 code or DWIN OS system, user CPU is omitted in practical application.

- Standard 8051 architecture and instruction set, 64Kbytes code space, 32Kbytes on-chip RAM.
- 64-bit integer mathematical operation unit (MDU), including 64 bit MAC and 64 bit divider.
- Built-in software WDT, three 16-bit Timers, 12 interrupt signals with the highest four interrupt nesting. 22IO, 4I UARTs, 1 CAN, 7 12-bit A/Ds, 2 16-bitadjustable-resolution PWM.
- Support IAP online simulation and debugging, unlimited number of breakpoints.
- Upgrade code online through the DGUS system.

(4) 1Mbytes on-chip Flash with DWIN patent encryption technology ensure code and data security, eliminate copycat and cloning.

(5) Reducing crystal requirements and PCB design challenges for a variety of inexpensive wide-range tuned

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



The arrangement of T5L pin

2.2 T5L_DGUS II Development System

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

DGUS is the DWIN Graphic Utilized Software.

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

(1) Variable programming

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

(2) Interface design

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

(3) Interface configuration

The interface is configured by DGUS software on PC.

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

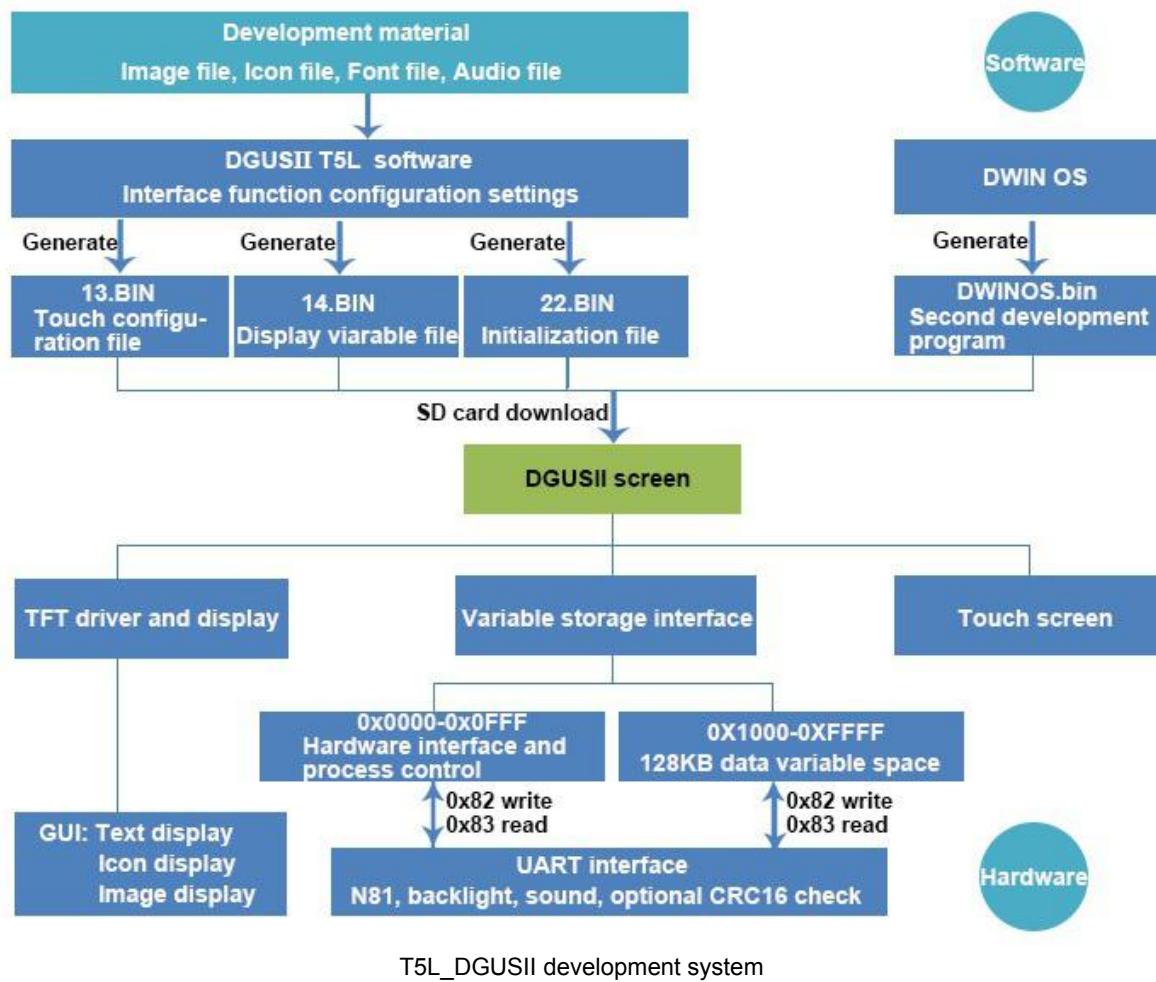
(4) Debug

Put the download files into the DWIN_SET folder of SD card and download them to the screen through the SD card.

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

(5) Mass production according to template

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



2.3 Features

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

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

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

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

The T5L_DGUS II platform specifies a refresh cycle of 20ms, which means that the contents of 13.bin and 14.bin are read once in every 20ms.

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

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

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

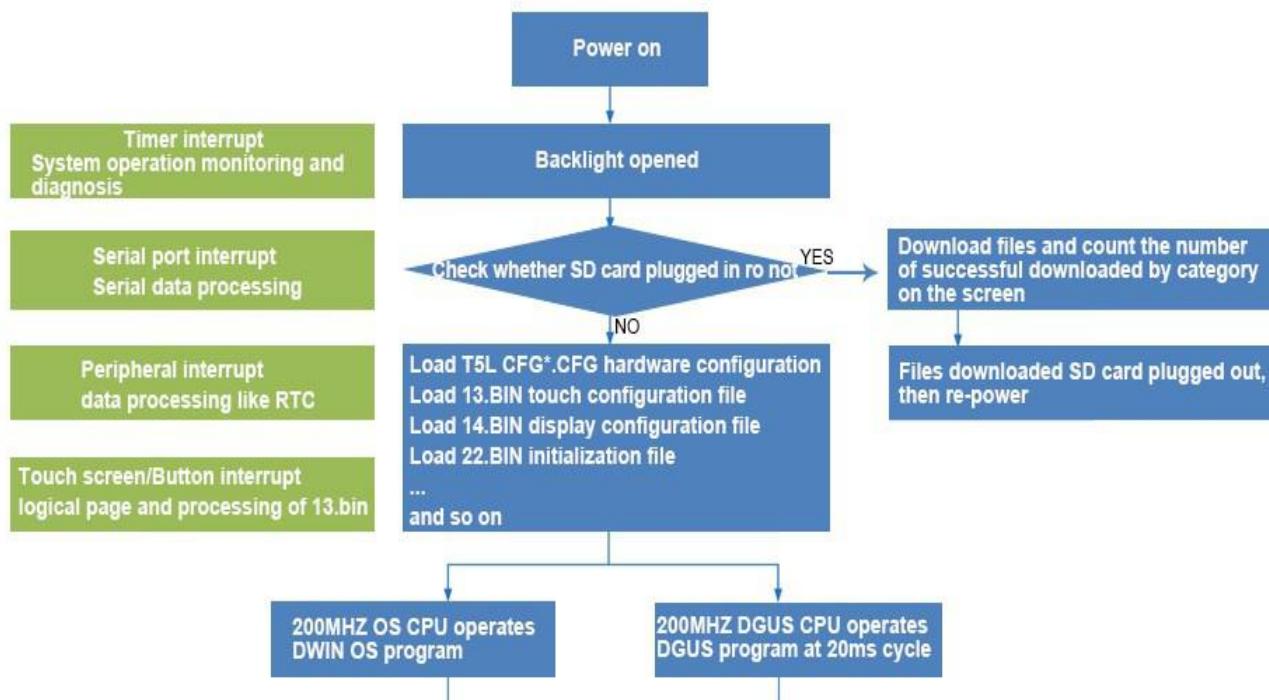
- (1) Based on T5L dual-core ASIC, GUI and OS cores both run at 200MHz main frequency (350MHz main frequency version can be customized), with extremely low power consumption.
- (2) 16Mbytes low cost SPI Flash(can be extended to 64Mbytes NOR Flash or 48Mbytes NOR Flash+512Mbytes NAND Flash), JPEG image, icon compression storage, You can specify the size of background image storage space.
- (3) 512Kbytes Nor Flash in-chip user database.
- (4) 128Kbytes data variable space.
- (5) Up to 255 display variables per page.
- (6) Support standard T5 DWIN OS platform or 8051 OS CPU core development: hardware CAN lead to 20 IO, 4 UARTs, 1 CAN, multiple ADs, customized services available.
- (7) 20mS DGUS cycle, extremely smooth UI.
- (8) Display variables can be opened, closed or modified in the application to achieve complex display combination functions.
- (9) Touch instructions can be opened, closed or modified in the application to realize complex touch combination functions.
- (10) Support SD interface download and configuration, download file statistics display.

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

2.4 Software Process

In the T5L_DGUS II development platform, the configuration of the touch operation of touch panel is also stored in 13.Bin file in the same way.

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



2.5 FLASH Memory Allocation

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

(1) 4-12MB font space can save BIN, HZK, DZK format files, the file ID ranges from 00 to 47

(2) 4-12MB picture space can save ICL files of background picture library and ICL files of icon library. The file ID ranges from 16 to 63.

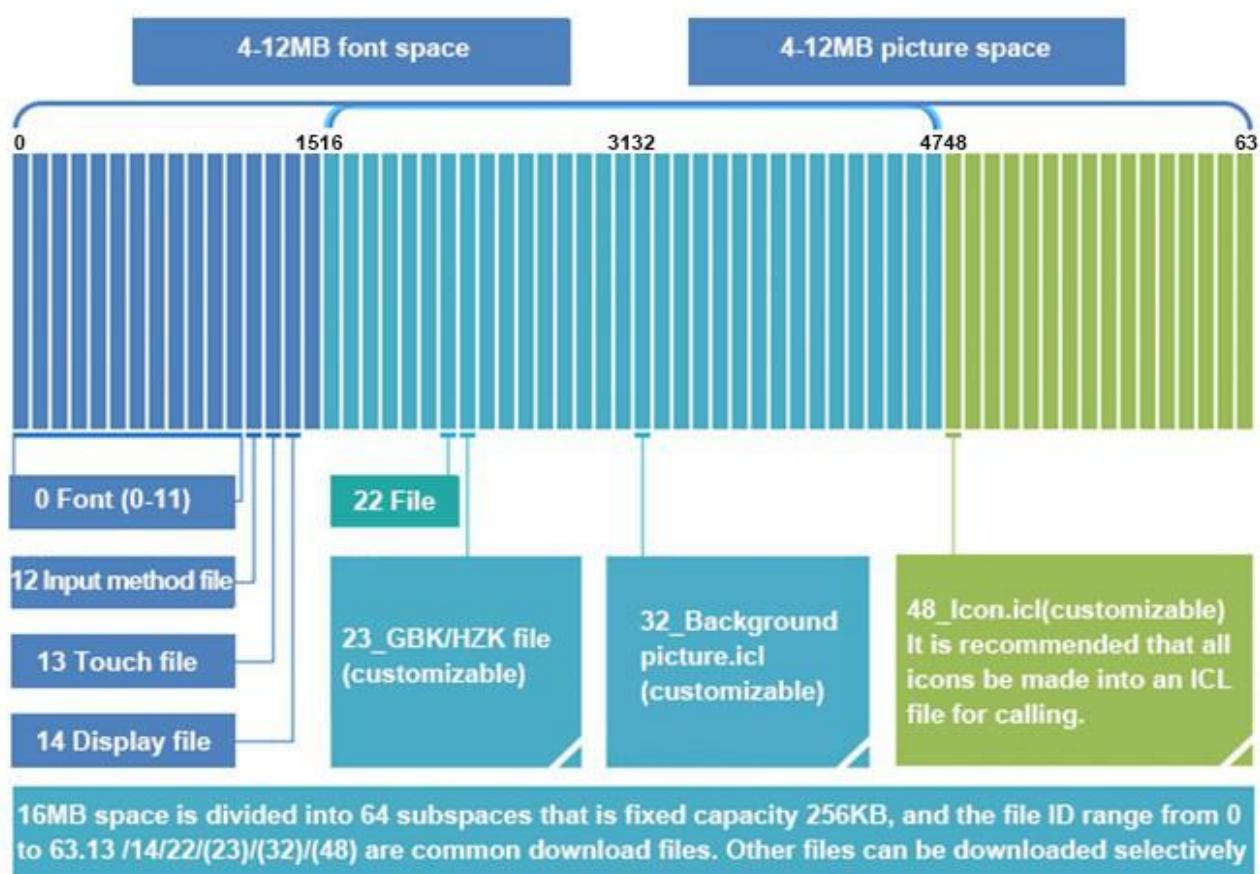
INFO:

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

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

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

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



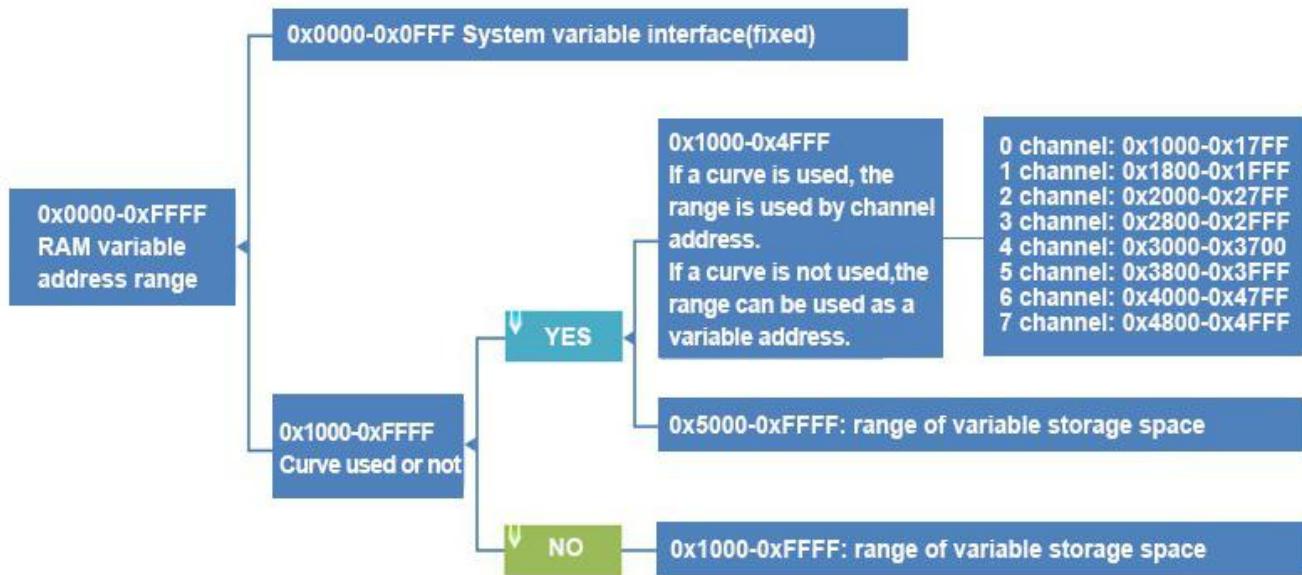
2.6 RAM Memory Variable Address Space

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

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

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

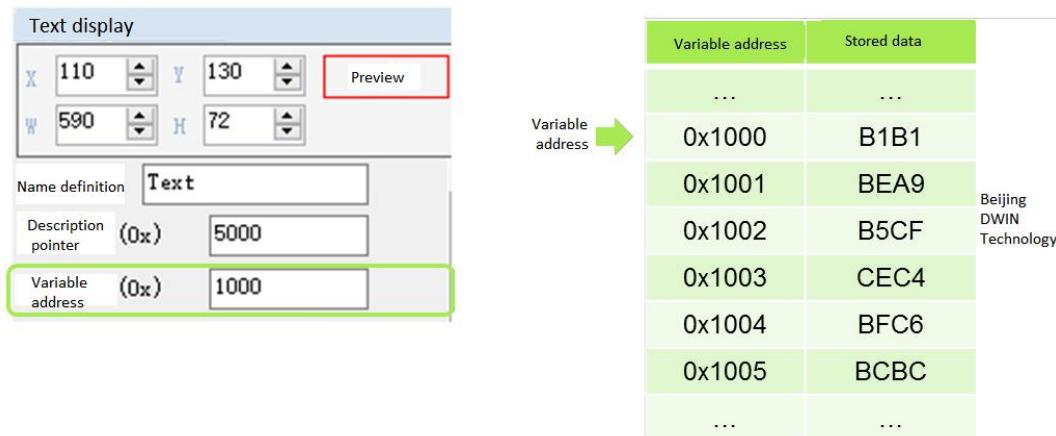
Then the user available address range is: 0x5000-0xFFFF.



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

For example, set the variable address of a text display control to 0x1000, and the text content displayed in the control is "Beijing DWIN Technology", then the storage method in the RAM space is shown in the following figure.

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





DWIN Technology

Professional, Creditable, Successful

T5L_DGUSII Application Development Guide

Storage method of data in variable address

2.7 RAM Description Pointer Space

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

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

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

As following, the text display control instruction storage format, the space pointed to by the text control description pointer stores the variable address, the upper left corner coordinates of the control, the text color, the upper left corner and the lower right corner coordinates of the text box in order. etc.

In the figure, 0x5000_H represents the high byte, and 0x5000_L is the low byte.

| Text Display Control Instruction Storage Format | | | |
|---|-------------|-------------|------------|
| Address | Definition | Data length | illustrate |
| 0x00 | 0x5A11 | 2 | |
| 0x02 | *SP | 2 | |
| 0x04 | 0x000D | 2 | |
| 0x06 | 0x00 *VP | 2 | . |
| 0x08 | X, Y | 4 | |
| 0x0C | Color | 2 | |
| 0x0E | Xs Ys Xe Ye | 8 | |

| | | | | | | |
|-----------|----------|----------|----|-----|---|---|
| SP 0x5000 | 0xFFFF_L | 0x5000_H | 10 | ... | Variable address VP | Text display |
| | 0x5000_L | 00 | 00 | | | X 110 Y 130 预览 |
| | 0x5001_H | 00 | 6E | | | W 590 H 72 |
| | 0x5001_L | 6E | 00 | | The starting display position, the coordinates of the upper left corner | Text |
| | 0x5002_H | 00 | 82 | | | name definition 5000 |
| | 0x5002_L | 82 | 84 | | | description pointer 1000 |
| | 0x5003_H | 84 | 00 | | | variable address 8400 |
| | 0x5003_L | 00 | 00 | | | display color |
| | 0x5004_H | | | | | Encoding 0x02=GBK |
| | | | | | | Character spacing is not automatically adjusted 200 |
| | | | | | | text length FONTO_ID 0 |

[Example]

Send instruction: 5A A5 05 82 5003 F800 (Effect: The character color changes from olive green to red.)

5003: The storage address of the text color.

F800: Red code.

Examples of other description pointer applications are as follows.

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

SP setting problems:

(1) In case of no curve display, 0X1000-0X7FFF can be used for VP, while 0X8000-0xFFFF can be used for SP. The distinction is to avoid conflicts caused by pointer address confusion.

(2) In case of curve display, and all 8 channels are used, 0X1000-0X4FFF will be used as the curve buffer address. Users should not use this part of the address as the address of other variables.

0X5000-0X7FFF can be used as the variable address range; 0X8000-0xFFFF can be used as the description pointer range. In the application, the user can plan the address range of VP and SP according to actual needs.

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

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

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

Chapter 3 SD Card Interface

3.1 MicroSD(TF) Card Format

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

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

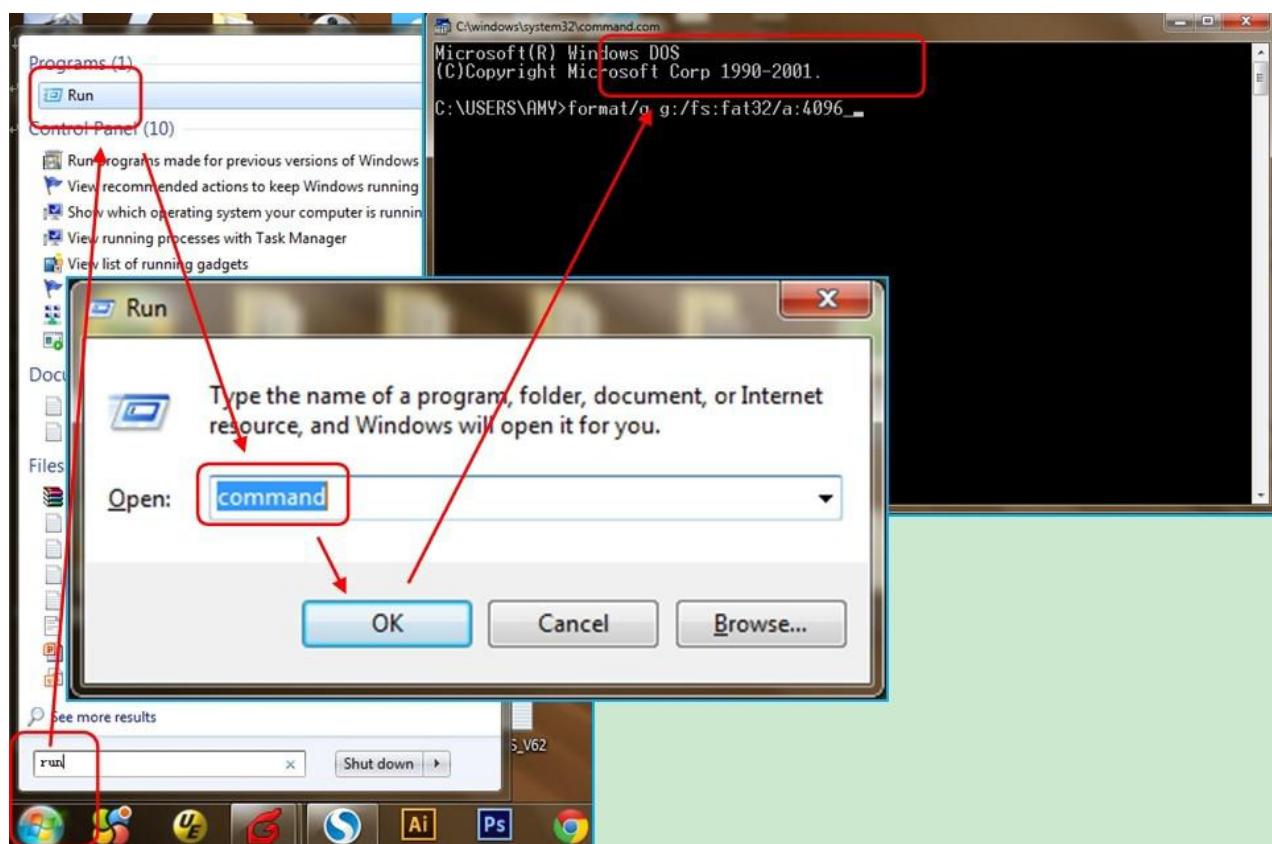
Format operations are as follows.

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

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

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

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



3.2 SD Card Interface Download Method

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

T5L_DGUSII is different from T5_DGUSI. In practical applications, the background images of the T5L screen needs to be downloaded in. ICL file, which can be generated by DGUS. JPEG pictures cannot be recognized directly.

The generation of download files is as follows.

- ① Create a DWIN_SET folder in the SD card root directory.
- ② Put the ICL file, font library, configuration files, etc. to the DWIN_SET folder.
- ③ When the screen is powered on, it will detect that the SD card and recognize whether there is a folder named DWIN_SET in the SD card root directory.
If the folder is successfully detected, the screen will download the relevant files to FLASH. Folders under the DWIN_SET folder cannot be read .
- ④ When the blue screen indicates that the download has completed, power off the screen, pull out SD card, and power on again. Then the project has been downloaded in the screen successfully.

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

3.3 Download File

3.3.1 Download File Type

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

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

| File Type | Naming Rule | Description |
|------------------------------------|--|--|
| Program file | T5L_UI*.BIN T5L_OS*.BIN | Underlying kernel firmware program |
| DWIN OS program | DWINOS*.BIN | DWIN OS program, the code must start from 0x1000. |
| OS CPU 8051 program | T5L51*.BIN | Program based on the standard 8051 platform. |
| NOR Flash data base | ID+(optional) file name.LIB | Each ID corresponds to 2KWords memory with ID range of 0 - 79. The database is located in on-chip NOR FLASH of 160KWords. It can be used to save user data or DWIN OS program library files. |
| Font file (4~12MB) | Font ID+(optional)file name.BIN/DZK/HZK | Font ID: 00-47; Pre-install standard ASCII 0#font library. |
| DGUS input method file | 12*.BIN | Fixed storage in 12 font location |
| DGUS touch control file | 13*.BIN | Fixed storage in 13 font location. Size limit: 256KB |
| DGUS display control file | 14*.BIN | Fixed storage in 14 font location. Size limit: 256KB, in DGUS II format. |
| DGUS variables initialization file | 22*.BIN | Fixed storage in 22 font location. Load 0x2000-0x1FFF address content and initialize 0x1000-0xFFFF variable space. |
| JPEG image and icon file | Font ID+ (optional) filename. ICL | Must be a ICL file in DGUSIII format. For multi FLASH expansion, one ICL data is saved in one Flash. |
| Music file | Font ID+ (optional) filename. WAE | Must be in DGUSII format, generated by DWIN tools. |
| UI configuration file | Configuration module ID+(optional) filename. UIC | The configuration module ID is from 0x0000 to 0xFFFF, and each ID corresponds to 32KB memory space. For multi FLASH expansion, one configuration module data is saved in one Flash. |
| Hardware configuration file | T5LCFG*.CFG | Configure CRC status, baud rate modification, touch panel sound control, touch upload mode, display direction, etc. |
| CRC check file | T5*.CRC | Perform CRC32 check after SD card download. |

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

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

(2) 4-12MB storage space for background images (.ICL files).

For T5L1 screens, the size of a single JPG file should not exceed 248KB, and for T5L2 CPUS, the size of a single JPG file should not exceed 760KB.

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

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

3.3.2 Generation of Download Files

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

DGUSII_T5L calls configuration files by the file name number. Therefore, the files should be named by Alabaster numbers in order.

For example: 32_Background_Image.ICL and 48_Icon_Library.ICL.

3.3.2.1 ICL Background Image Library File

1. Background images should be in JPEG, BMP or PNG formats.
2. The image resolution should be the same as the screen in 24-bit color.
3. By default, image 0 in the ICL file will be displayed at booting. The images are named according to the Arabic numbers, e.g. 00_boot, 01_function, 02_menu.
4. You need to convert images into .ICL file to download it to the screen. That is, the images are packaged and downloaded in the form of a library file.
5. Background image ICL files are used to display background images. If you name background image ICL as 32.ICL, you need to write 0x20 in the 0x08 address of the CFG file so that the underlying programs can correctly read and identify the ICL file.

If the CFG file is empty, the screen will be black (with a weak backlight). In this case, you should check the 0x08 address in the CFG file.

6. For T5L1 screen, the single image size in the ICL file should not exceed 256KB; For T5L2 screen, that should be less than 768KB. There is no limit to the size of the original image materials.

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

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

3.3.2.2 ICL Icon Library File

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

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

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

3.3.2.3 Font File

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

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

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

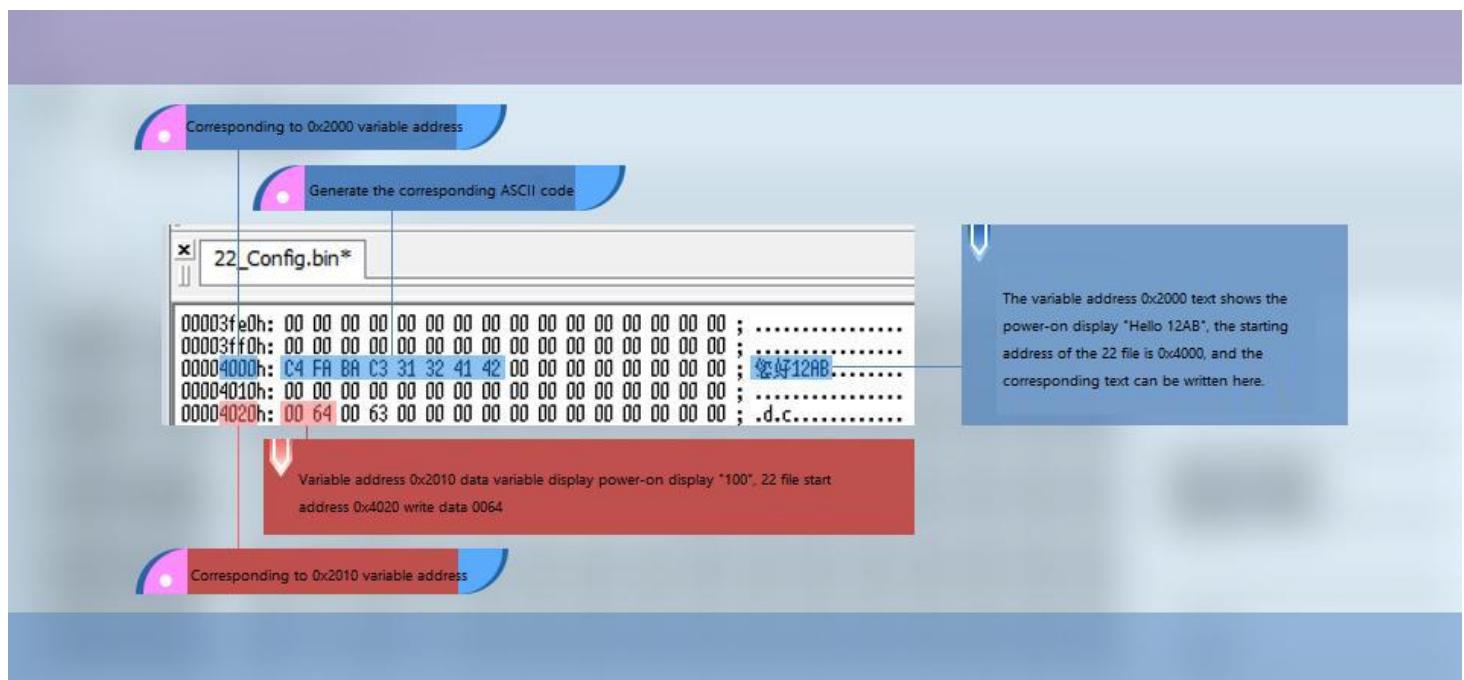
3.3.2.4 22.Bin Initialization Variable File

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

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

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

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



22 file initial value modification

3.3.2.5 System Configuration

T5L CFG *.CFG hardware configuration file is in binary data format and can be edited by UltraEdit.

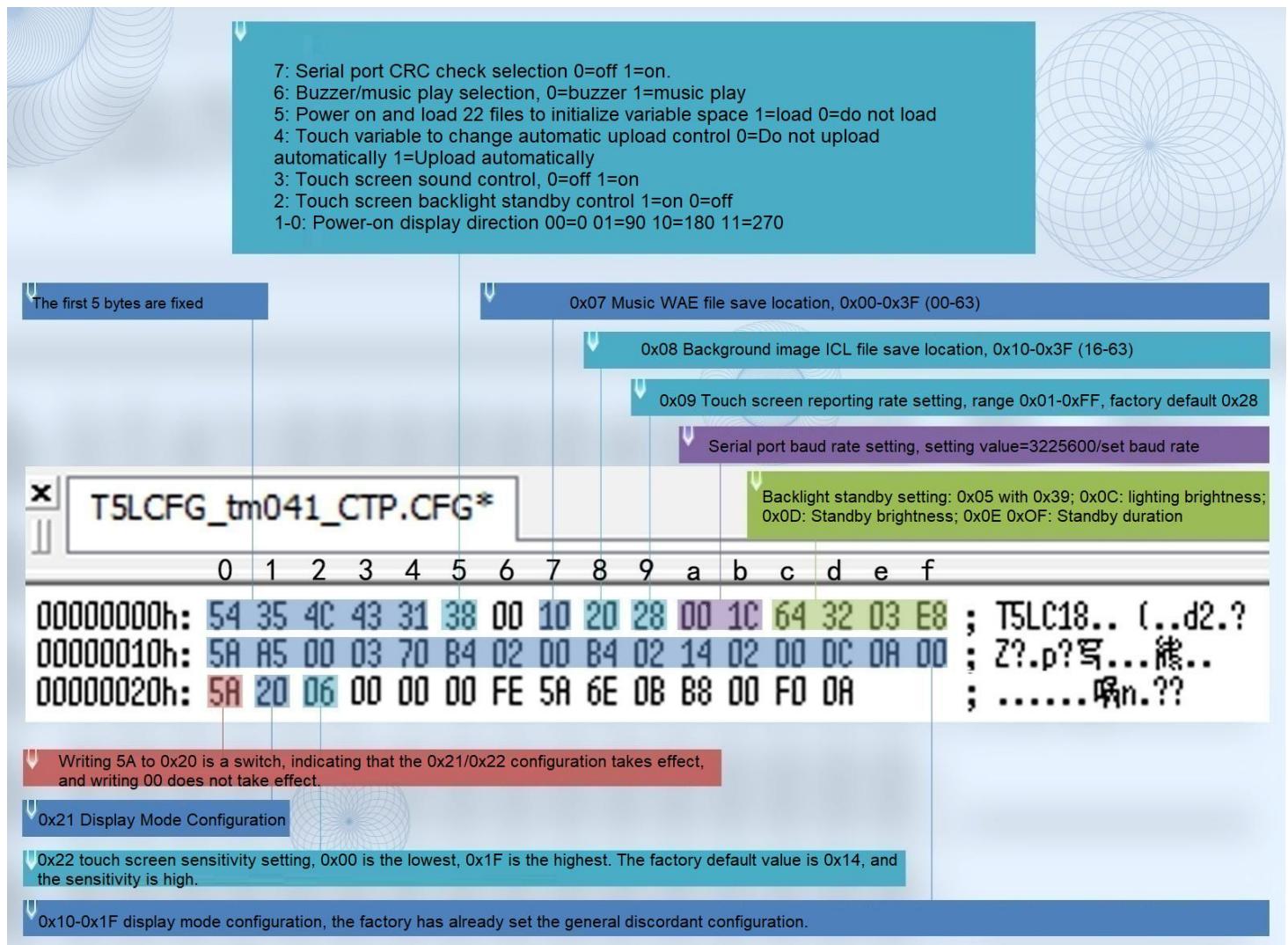
The description is as follows :

| Category | Address | Length | Definition | Description |
|---------------------------------|---------|--------|-----------------------------|--|
| Recognition configuration | 0x00 | 5 | 0x54 0x35 0x4C 0x43 0x31 | Fixed content. |
| System configuration | 0x05 | 1 | Parameter configuration | .7: UART CRC check 0=close 1=open .6: Buzzer / music playback selection. 0=Buzzer 1=music playback .5: Load 22 file initialization VP at power-on 1= Load 0= No Load .4: Touch variable automatic upload 1=On, 0=Off .3:Touch panel audio 1= Open 0= Close .2: Touch panel standby backlight 1= Open 0=Close .1-0: Display direction 00 = 0°, 01 = 90°, 10 =180° ,11 = 270 ° |
| | | | | .7: PWM output 0=user control; 1= PMW backlight brightness control, 1KHz frequency. .6: Expand SPI NAND Flash. 0=not expanded 1=expanded. .5: Set 1 to start SPI NAND Flash format (data will be cleared). .4: SPI NAND Flash memory. 0=1Gbits; 1=4Gbits. .3: Set to 1 to erase all off-chip Flash contents. .2: AD return value resolution selection. 0=12bit; 1=16bit. .1:OS CPU user 8051 program download encryption. 0=unencrypted ; 1=encrypted. (In case of encryption, users should use DWIN tool to encrypt before downloading the 8051 program.) 0: Reserved, write 0 . |
| | | | | 0x07 1 Music WAE file location 0x00-0x3F(00-63) |
| | | | | 0x08 1 Background image ICL file location 0x10-0x3F(16-63), correspond to 12MB-4MB background image size. |
| | | | | 0x09 1 Touch panel reporting point rate 0x01-0xFF. By default 0x28, reporting point rate=400Hz/set value |
| | | | | 0x0A 2 UART baud rate Baud rate =3225600/set baud rate (max of 0x03FF) For 115200bps, set value=0x001C. |
| | | | | 0x0C 1 Normal and startup brightness 0x00-0x64, unit 1% |
| Backlight standby configuration | 0x0D | 1 | Standby brightness | 0x00-0x64, unit 1% |
| | | | | 0x0E 2 Wake up time after standby 0x0001-0xFFFF, unit 10 ms |

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

| | | | | |
|---|------|----|--|---|
| | 0x22 | 1 | TP_Sense | TP sensitivity setting: 0x00-0x1F. 0x00 lowest sensitivity; 0x1F highest sensitivity. Default is 0x14(ILI9881 is 0x01-0x06). 0xFF = no touch (using hardware initials). |
| | 0x23 | 1 | TP_Freq | TP frequency selection, suitable for ILI9881H. Fixed frequency = 0x01 - 0x14, and 0x00 represents frequency hopping. |
| Clock output configuration | 0x24 | 1 | CKO_Set_En | 0x5A= Enable the configuration. |
| | 0x25 | 1 | CKO_En | 0x5A= Enable the output function of CKO (P3.0) |
| | 0x26 | 1 | CKO_DIV | CKO output clock setting. Frequency = 825.7536/CKO_DIV MHz. |
| Buzzer configuration | 0x27 | 1 | BUZZ_Set_En | 0x5A= Enable the configuration. |
| | 0x28 | 1 | BUZZ_Freq_DIV1 | Buzzer frequency=BUZZ_Freq_DIV1) KHz.825753.6 BUZZ_Freq_DIV1 |
| | 0x29 | 2 | BUZZ_Freq_DIV2 | Factory settings: DIV1=0x6E, DIV2=0x0BB8, correspond to the 2.5KHz frequency. |
| | 0x2B | 2 | BUZZ_Freq_Duty | Buzzer duty ratio settings: High level duty ratio=BUZZ_Freq_Duty / BUZZ_Freq_DIV2. Factory settings: 0x00F0 corresponds to 8% high level duty ratio. |
| | 0x2D | 1 | BUZZ_Time | Buzzing time after touch, unit 10 ms; Factory setting: 0x0A |
| Power-on initialization file ID configuration | 0x2E | 1 | Init_File_Set_En | 0x5A = Start configuring power-on initialization variable file ID. |
| | 0x2F | 1 | Init_File_ID | The configured power-on initialization variable file ID, the factory defaults 0x16. |
| Reserved | 0x2E | 18 | Reserved | Write 0x00. |
| SD download file folder name setting | 0x40 | 2 | SD_Set_En | 0x5AA5 means setting the name of SD download folder once, which are saved in the screen Flash. It will not be lost after power off. |
| | 0x42 | 1 | Character length of download folder name | 0x01-0x08 |
| | 0x43 | 8 | Folder name | Up to 8 ASCII characters (only 0-9, a-z, A-Z, -, _). Invalid character setting will use "DWIN_SET" as the default value, which is effective after power off and restart. |
| | 0x4B | 37 | Reserved | Write 0x00. |

Attention: green background parameters in the table must be configured.



General diagram of CFG file configuration

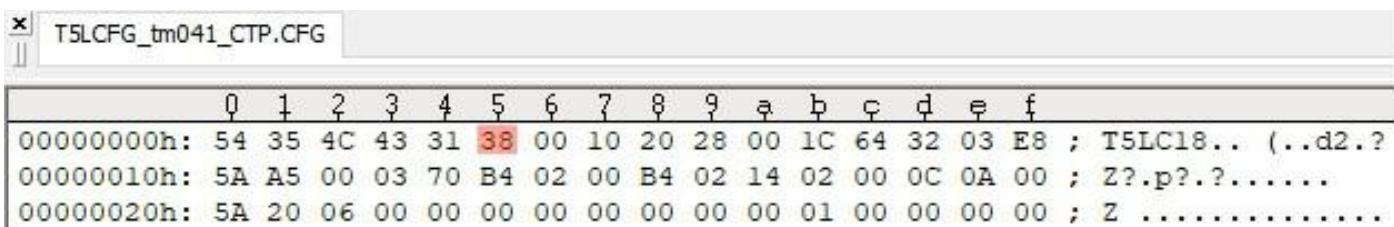
For example, CFG file 0x05 parameter configuration:

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

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

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

Convert the binary 0011 1000 to hexadecimal 0x38, write 0x38 to 0x05 address, and download the CFG file to complete the corresponding configuration.



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

② Enable CRC check

The 7th bit of address 0x05 is configured as 1, which means CRC verification is enabled.

③ Different display direction

Setting value of DMG10600C070-03W (Horizontal screen at factory) in CFG file at different display direction

| Mode | Image Resolution | ICL | Configuration Value |
|----------------|------------------|----------------------------------|---------------------|
| 0 ° lateral | 1024 * 600 | ICL file made by 1024*600 images | 0x38 |
| 90 ° vertical | 600 * 1024 | ICL file made by 600*1024 images | 0x39 |
| 180 ° lateral | 1024 * 600 | ICL file made by 1024*600 images | 0x3A |
| 270 ° vertical | 600 * 1024 | ICL file made by 600*1024 images | 0x3B |

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

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

④ Voice playback

Method 1: Press to play voice

For example, instead of buzzer sound, use a short "tap" sound to replace buzzer.

Use touch controls such as basic touch or return key value.

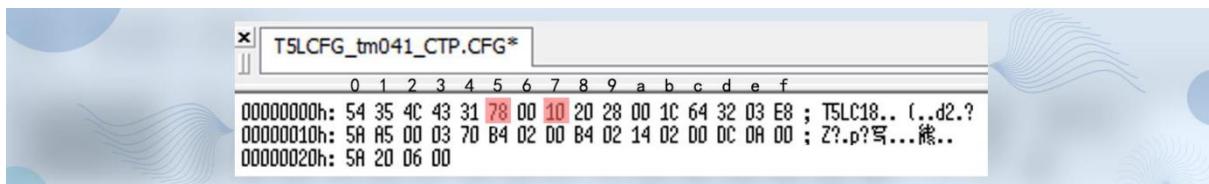
Check the voice ID, set the voice name ID and click to play. (DMT10600C101-07W factory DEMO has a section of 05 water drop sound can be called to use. T5 DEMO can be obtained from our customer service).

Step 1: Convert the voice file to WAE format and name it as 16_voice file.

Step 2: Set the voice ID required by the touch control. In this example, set the ID to 0x10, that is, file 16.

Step 3: Configure the CFG file.

CFG file 0x05 address is written as 0x78, that is, 0111 1000, to specify the music file as file 16.



Step 4: Add "16_ music file. WAE "and T5LCFG.CFG configuration file to the DWIN_SET folder, download them to the screen by SD card, and play the sound by touch.

Method 2: Use instruction to play a piece of speech (music)

Step 1: Convert the voice file to WAE format and name the 16_ voice file.

Use the music file in DMT32240C028-06W DEMO.

Step 2: Configure the CFG file.

CFG file 0x05 address is written as 0x78, that is, 0111 1000, to specify the music file as file 16.

Step 3: Add "16_ music file. WAE "and T5LCFG.CFG configuration file to the DWIN_SET folder, download them to the screen by SD card.

This music file is 7.64MB, $7.64 * 1024 / 256 = 30.56$, occupying 31 intervals, that is, the range from 16 to 46 is occupied. Thus the next file should be named from 47.

If 32 is specified as the background image location, the image will be overwritten.

You can re-specify the background location to display the image. You can refer to Chapter 2.5, for detailed description of file naming and sorting.

Step 4: Send instructions to play the sound: 5AA5 07 82 00A0 1001 4000

0x5AA5: Frame header;

0x07: Data length;

0x82: Write instruction;

0x00A0 System variable interface audio playback address.

0x10 Position of audio file, range 0x00-0xFF;

0x01 Fixed to 0x01 and the whole music is played;

0x40 Volume, unit 1/64. Speaker 3-5W, impedance 8 Ohms, and buzzer interface can be connected to the speaker.

0x00 Playback progress. Read only.

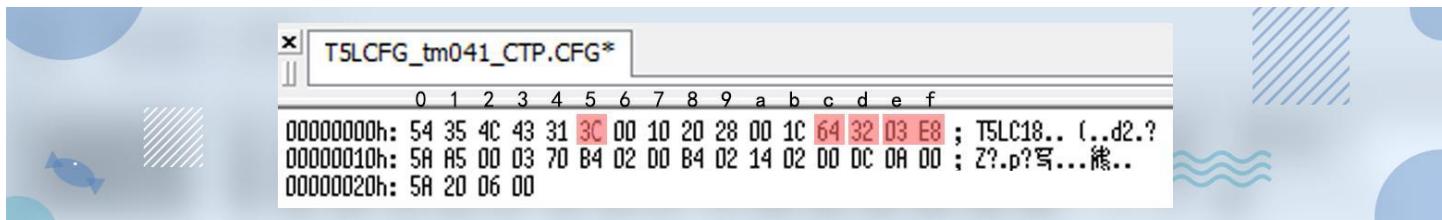
Method 3: Play voice (music) by Synchrodata return control

Control 0x01 mode, data from VP1S will be transmitted to VP1T. Set first press setting as music playback, release as animation playback. This control is explained in chapter 6.8.

(2) CFG 0x0C backlight screensaver configuration

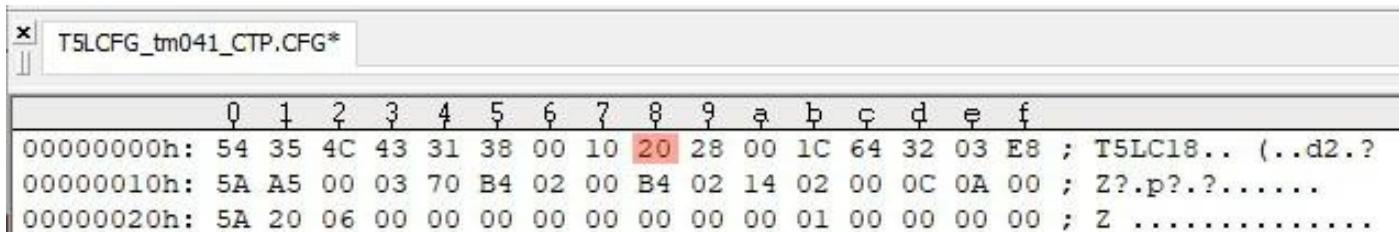
| CFG Address | Description | Configuration Value |
|-------------|--------------------------|-----------------------------------|
| 0x05 | Turn on backlight | 0x3C |
| 0x0C | Brightness value | Range: 0x00-0x64 |
| 0x0D | Standby brightness value | Range: 0x00-0x64 |
| 0x0E 0x0F | Backlight delay |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

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



(3)Background image location

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



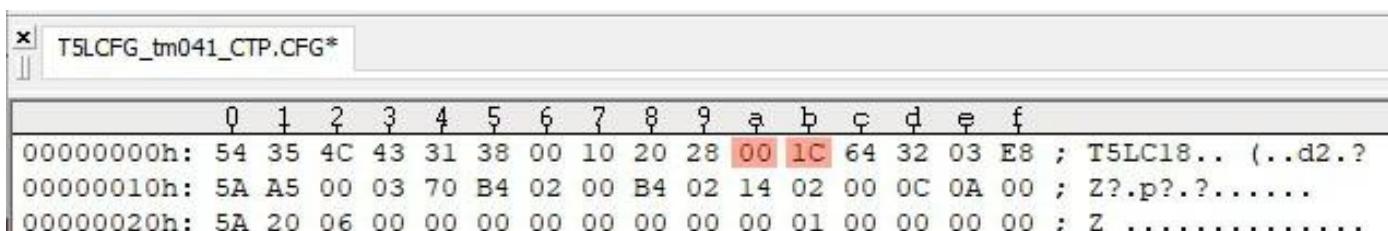
(4) Baud Rate

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

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

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

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



(5) Display

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

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

| Hex | | | | | | | | | | | | | | | |
|------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | a | b | c | d | e | f |
| 00000000h: | 54 | 35 | 4C | 43 | 31 | 38 | 00 | 10 | 20 | 28 | 00 | 1C | 64 | 32 | 03 |
| 00000010h: | 5A | A5 | 00 | 03 | 70 | B4 | 02 | 00 | B4 | 02 | 14 | 02 | 00 | 0C | 0A |
| 00000020h: | 5A | 20 | 06 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 01 | 00 | 00 | 00 | 00 |

0x12-0x1F of CFG file

(6) Sensitivity

Sensitivity address is 0x22, by default the value is set to 0x14. The range is 0x00- 0x1F.

0x5A at 0x20 indicates the configuration will take effect.

| Hex | | | | | | | | | | | | | | | |
|------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | a | b | c | d | e | f |
| 00000000h: | 54 | 35 | 4C | 43 | 31 | 38 | 00 | 10 | 20 | 28 | 00 | 1C | 64 | 32 | 03 |
| 00000010h: | 5A | A5 | 00 | 03 | 70 | B4 | 02 | 00 | B4 | 02 | 14 | 02 | 00 | 0C | 0A |
| 00000020h: | 5A | 20 | 06 | 00 | 00 | 00 | FE | 5A | 6E | 0B | B8 | 00 | F0 | 0A | 00 |

(7) Buzzer

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

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

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

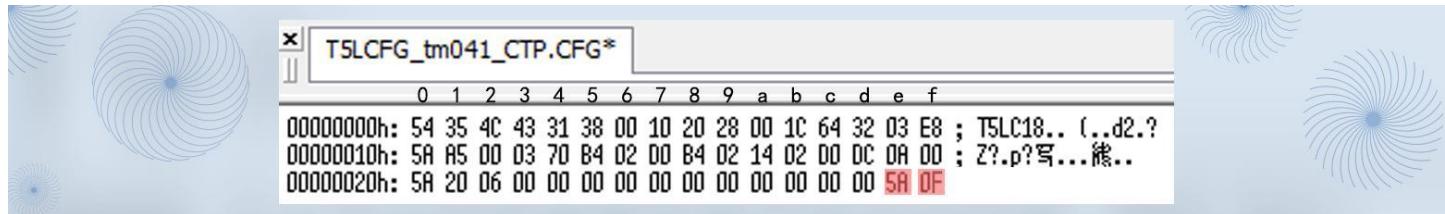
| Hex | | | | | | | | | | | | | | | |
|------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | a | b | c | d | e | f |
| 00000000h: | 54 | 35 | 4C | 43 | 31 | 38 | 00 | 10 | 20 | 28 | 00 | 1C | 64 | 32 | 03 |
| 00000010h: | 5A | A5 | 00 | 03 | 70 | B4 | 02 | 00 | B4 | 02 | 14 | 02 | 00 | 0C | 0A |
| 00000020h: | 5A | 20 | 06 | 00 | 00 | 00 | FE | 5A | 6E | 0B | B8 | 00 | F0 | 0A | 00 |

(8) Power on Initialization variable file ID

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

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

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



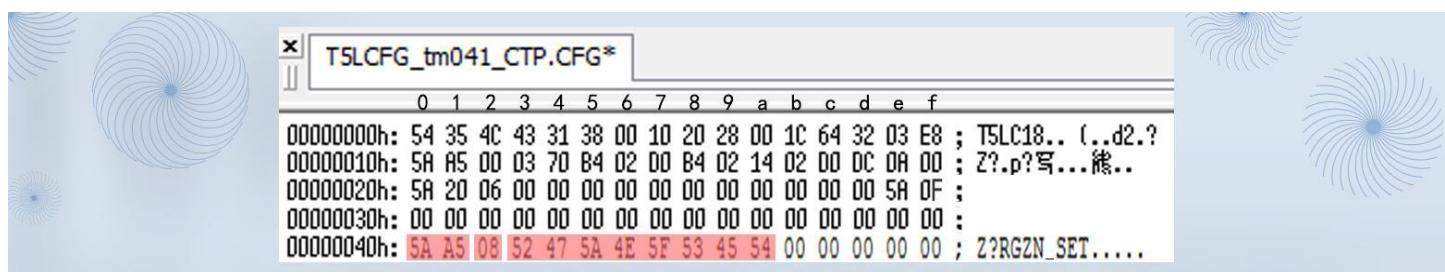
(9) SD download folder name

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

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

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

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



(10) T5L DGUSII touch screen calibration

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

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

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

Step 1: Download the CFG File

Take DMG48480C040_03WTR as an example.

Write 0x8B at CFG 0x21 to turn on the touch panel calibration mode, (set 0x21.7 and 0x21.3 to 1) and download CFG file to screen to achieve touch calibration.

| | | | |
|--|--|--|---|
| | | | Touch screen mode configuration. .7~4 (high 4bit) type selection. .7=1, indicate that start touch calibration. 0x0= Resistance touch screen. 0x1= Capacitance touch screen derived by GT911, GT9271 or GT9110 IC. 0x2=ILI9881H In-cell CTP. 0x21 1 TP_Mode .3 Resistance touch screen calibration: 0=off 1=on, enable when SD card downloading is in process .2~0 (low 3bit) Coordinate setting: .2 X coordinate selection: 0=from 0 to <u>Xmax</u> , 1=from <u>Xmax</u> to 0; .1 Y coordinate selection: 0=from 0 to <u>Ymax</u> , 1=from <u>Ymax</u> to 0; .0 XY exchange: 0=XY 1=YX (3inch RTP is 0x06; 7inch RTP is 0x07; 8inch RTP is 0x05; 10.4inch RTP is 0x03) |
|--|--|--|---|

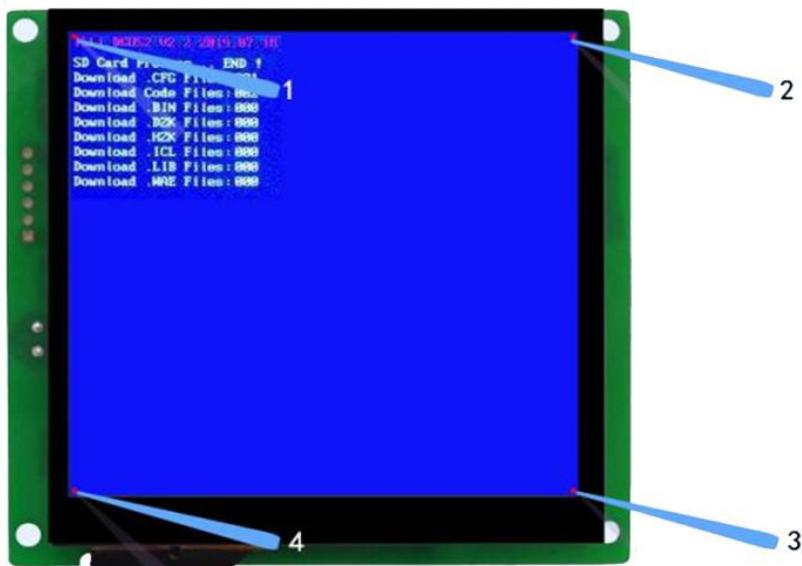
TSLCFG.CFG

| | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | a | b | c | d | e | f |
|------------|-----------|-----------|----|----|----|----|----|----|----|----|----|----|----|----|----|----------------------|
| 00000000h: | 54 | 35 | 4C | 43 | 31 | 38 | 00 | 10 | 20 | 28 | 00 | 1C | 64 | 32 | 03 | E8 ; TSLC18..(..d2.? |
| 00000010h: | 00 | 00 | 00 | 03 | 70 | 84 | 02 | 00 | 84 | 02 | 14 | 02 | 00 | 0C | 0A | 00 ; Z?p?写...然.. |
| 00000020h: | SH | 83 | 14 | 00 | 00 | 00 | FE | 5A | 6E | 08 | 88 | 00 | F0 | 08 | | ;哪.?? |

Step 2: Touch calibration

This step should be performed with continuous power supply.

Click the "top left corner, top right corner, bottom right corner and bottom left corner" of the touch panel successively with sharp objects, and some screens will display red pixel points to indicate that the calibration is completed.



Except DMG80480L070_01 WTR and DMG80480L080_01WTR factory kernel is TA instruction set development mode, the other models are all DGUSII development mode.

Besides, DMG12720C050-03WTC and DMG72720C041-03WTC are Incell capacitive touch, LCD IPS integrated screen.

Table 3.3 parameter configuration of T5L touch panel 0x21

| Size | Resolution | Model | 0x21 Standard Configuration | |
|------|------------|--------------------|-----------------------------|--------------------------|
| | | | Resistive Screen (RTP) | Capacitance Screen (CTP) |
| 3.5 | 480*320 | DMG48320C035_03W | 03 | 10 |
| 4 | 800*480 | DMG80480C040-03W | 03 | 16 |
| 4 | 480*480 | DMG48480C040-03W | 05 | 16 |
| 4.1 | 720*720 | DMG72720C041-03WTC | 20 (Incell touch panel) | |
| 4.3 | 480*272 | DMG48270C043-03W | 03 | 10 |
| 4.3 | 800*480 | DMG80480C043-03W | 05 | 16 |
| 5 | 854*480 | DMG85480C050-03W | 06 | 18 |
| 5 | 800*480 | DMG80480C050-03W | 03 | 10 |
| 5 | 1280*720 | DMG12720C050-03WTC | 20 (Incell touch panel) | |
| 5.6 | 640*480 | DMG64480C056-03W | 00 | 12 |
| 7 | 800*480 | DMG80480C070-03W | 06 | 10 |
| 7 | 800*480 | DMG80480L070_01WTR | 07 | - |
| 7 | 1024*600 | DMG10600C070-03W | 06 | 10 |
| 8 | 800*600 | DMG80600L080_01WTR | 05 | - |
| 8 | 800*600 | DMG80600C080-03W | 05 | 10 |
| 8 | 1024*768 | DMG10768C080-03W | 05 | 10 |
| 9.7 | 1024*768 | DMG10768C097-03W | 06 | 19 |
| 10.1 | 1024*600 | DMG10600C101-03W | 06 | 10 |

(1) CFG file generation by DGUS

Generally, users only need to set system settings and backlight standby to meet the general requirements.

Table 3.4 reference list of display configuration

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

3.3.2.6 T5*.CRC File Format

| Category | Address | Length | Definition |
|---------------------------|---------|--------|--|
| File identification | 0x0000 | 4 | Fixed to 0x43 0x52 0x43 0x10 . |
| CRC result | 0x0004 | 4 | Specify the check position data, and perform the final result of CRC32 check in sequence. |
| Reserve | 0x0008 | 7 | Write 0x00. |
| OS core code verification | 0x000F | 1 | 0x5A means to perform the CRC check of the OS core code, and the rest means skip the check. |
| LIB file check | 0x0010 | 2 | 0x10=ID of LIB file to be checked; 0x11=Number of files to be checked at the beginning of this ID, 0x00 means the end of LIB file check. |
| | 0x0012 | 62 | The remaining 31 defined positions. |
| Font file verification | 0x0050 | 2 | 0x50=the ID of the font file to be checked; 0x51=the number of files that need to be checked at the beginning of this ID, 0x00 means the end of the check of the font file. For off-chip NOR Flash, the font size is 256KB. For off-chip NAND Flash, the font file size is 8MB; and due to bad blocks, the last file should not be used or verified. |
| | 0x0052 | 254 | The remaining 127 defined positions. |
| Reserve | 0x0150 | 176 | Write 0x00. |

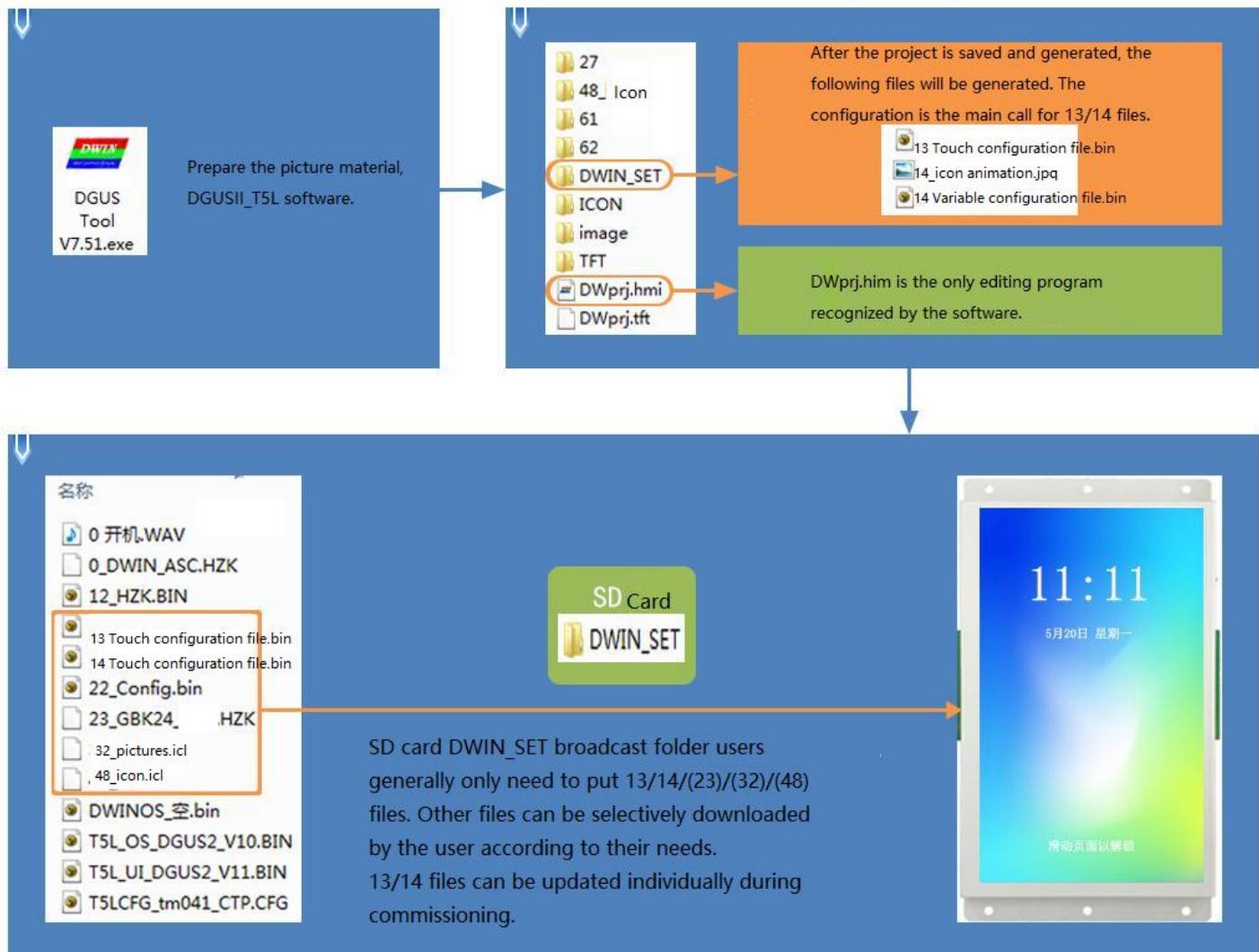
The polynomial for CRC32 calculation is X32+X26+X23+X22+X16+X12+X11+X10+X8+X7+X5+X4+X2+X+1 .

3.3.3 File Download

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

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

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



Download files generation diagram

Chapter 4 UART2 Serial Communication Protocol

DGUS screen adopts asynchronous, full-duplex serial port (UART). The UART mode is 8n1, that is, each data transmitted by 10 bits, including 1 start bit, 8 data bits, and 1 stop bit.

232/TTL communication and main board T/R input and output signal cross wiring, the ground wire must be connected; 485+ connects to A+ and 485- connects to B-.

All instructions or data of UART are in HEX. For word (2 bytes) data, the MSB method is adopted. E.g. for 0x1234, 0x12 is first to transmit.

The maximum data length in a DGUS cycle depends on the complexity of the user interface; it is recommended that customers do not send more than 4KB (approximately equal to 230400-691200bps baud rate continuous transmission) data to the DGUS screen in a DGUS cycle.

4.1 Data Frame Structure

UART 2 can be used for system debug, whose mode is fixed at 8N1.

Baud rate can be set, and the data frame consists of 5 parts.

| Data Block | 1 | 2 | 3 | 4 | 5 |
|-------------|--------|----------------------|----------------------|--------------|----------------------|
| Definition | Frame | Data length | Instruction | Data | CRC (optional) |
| Data Length | 2 | 1 | 1 | Max249 Bytes | |
| Description | 0x5AA5 | Instruction+data+CRC | 0x82Wite 0x83Read | | CRC-16(x16+x15+x2+1) |

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

| Instruction Example | CRC Check Off | CRC Check On |
|----------------------|-------------------------------|-------------------------------------|
| 83 Read instruction | Tx:5A A5 04 83 000F 01 | Tx:5A A5 06 83 000F 01ED 90 |
| 83 Instruction reply | Rx:5A A5 06 83 00 0F 01 14 10 | Rx:5A A5 08 83 00 0F 01 14 10 43 F0 |
| 82 Write instruction | Tx:5A A5 05 82 10 00 31 32 | Tx:5A A5 07 82 10 00 31 32 CC 9B |
| 82 Instruction reply | Rx:5A A5 03 82 4F 4B | Rx:5A A5 05 82 4F 4B A5 EF |
| 83 Touch upload | Rx:5A A5 06 83 10 01 01 00 5A | Rx:5A A5 08 83 10 01 01 00 5A 0E 2C |

4.2 UART2 Debug Interface Instruction

The DGUS screen works in variable-driven mode, and the operating mode of the screen and the status of the GUI are completely controlled by the data variables.

Therefore, the UART instructions only need to read and write the variables.

The instruction set is very simple and the user only needs to use 82/83 instructions.

| Function | Instruction | Data | Description |
|---------------------------------------|--|--|--|
| Visit send register | 0x80 | Send: Register Page (0x00-0x08) + Register Address (0x00-0xFF) + Data to write | Write data string into register. |
| | | Answer: 0x4F 0xA B | Answer of write instruction. |
| | 0x81 | Send: Register Page(0x00-0x08) + Register Address (0x00-0xFF) + Data Length to Read (0x01-0xFB) | Read data from register. |
| | | Answer: Register Page (0x00-0x08) + Register Address (0x00-0xFF) + Data Length + Data | Answer of read instruction. |
| | Instruction example: 5A A5 04 81 00 0A 04 Read 04 bytes of data in the os registers R10~R13 of the 00 register page. The 80 81 instruction is generally not required by users, and is usually used for debugging access to OS functions. You can refer to the register table of the OS application guide for viewing. OS register page ID (0x00-0x07) and interface register page ID (0x08), please refer to Section 3.4 of "T5-based DWIN OS Development Guide" for details. | | |
| Access variable memory (RAM) | 0x82 | Send: Start Address of SRAM (0x0000-0xFFFF) + Data to Write | Write data in to SRAM. Do not write data into space reserved for system. |
| | | Answer: 0x4F 0xA B | Answer of write instruction. |
| | 0x83 | Send: Start Address of SRAM (0x0000-0xFFFF) + Data Length to Read (0x01-0x7D) | Read data form SRAM. Appoint length data. |
| | | Answer: Start Address of SRAM + Data Length (word) + Data | Answer of read instruction. |
| | DGUS II screen have 128KB variable storage, mainly for variable data storage, operating by word addressing. | | |

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

Table 4.2 Definition of register page

| Register Page ID | Definition | Description |
|------------------|---------------|---|
| 0x00-0x07 | Data Register | 256 per group, R0-R255 |
| 0x08 | Port Register | DR0-DR255 See 3.4 of DWIN OS development guide based on T5 for more details. |

4.2.1 Write VPS (0x82)

Write the value 2 in VP 0x1000: **5A A5 05 82 1000 0002**

5A A5: Frame header

05: Data length

82: Write VPS

1000 : RAM address (2bytes)

0002: Data(2bytes)

4.2.2 Read VPS(0x83)

Read the value in VP 0x1000: **5A A5 04 83 1000 01**

5A A5: Frame header

04: Data length

83: Read VPS

1000: RAM address

01: Number of data (words) to read.

Answer from LCM:5AA5 0683 1000 01 **0002**

0002: Value in RMA address 1000

4.2.3 Touch to Return the UART Data (0x83)

Return to the VP 0x1001, key value 0x0002:

5A A5 06 83 1001 01 0002

5A A5: Frame header

04: Data length

83: Read VPS

1001: RAM address(2 bytes)

01: Data length is 1 word

00 02: Key value 0002

4.2.4 Instruction Enable Check

Take 5A A5 05 82 5420 0064 as an example.

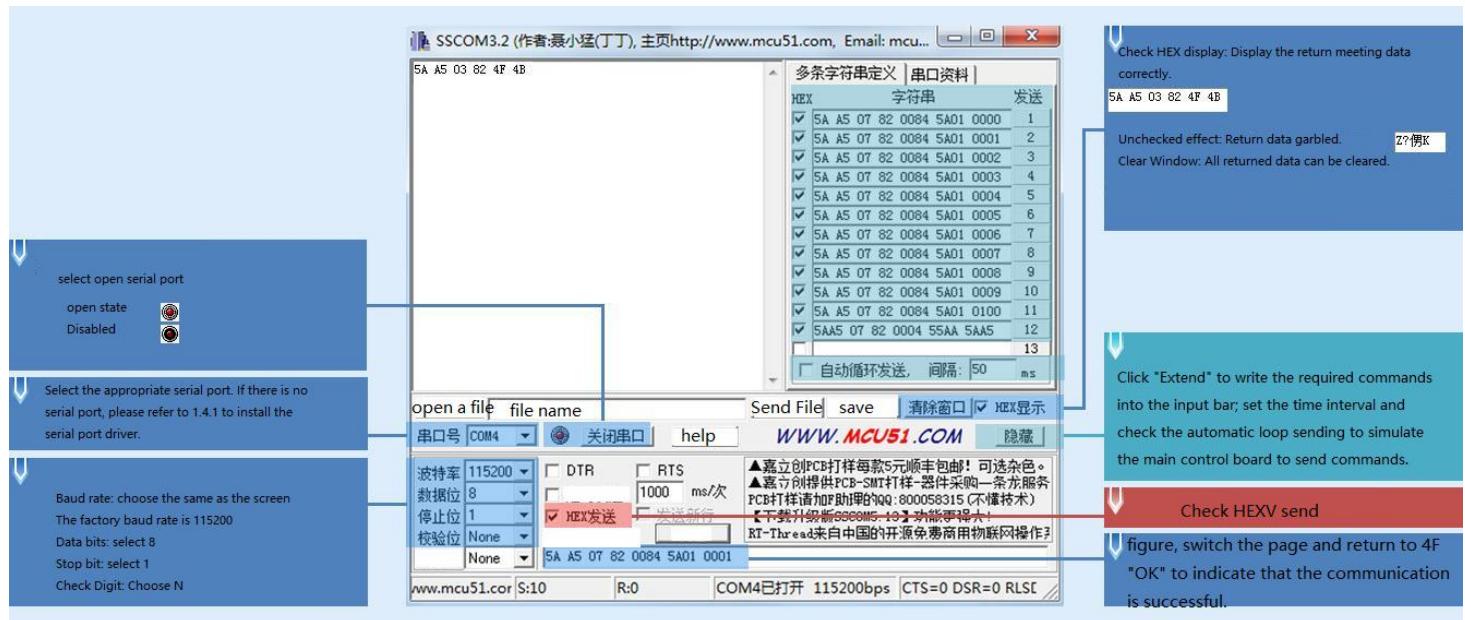
Check format: CRC-16

Check content: 82 5420 0064

Check result: 0x4C0F

The high and low bits of some verification tools are opposite. In that case, if the results is 0x0F4C, then the check value in the instruction should be 0x4C0F.

0x2C.6 at CFG file should be set to enable check in advance.



Chapter 5 System Variable Interface

5.1 System Variable Interface

System variable address range: 0x0000-0xFFFF.

| VP | Definition | R/W | Length (word) | Description |
|-------------|------------------|-----|---------------|--|
| 0x00 | Reserved | | 4 | Undefined |
| 0x04 | System_Reset | W | 2 | Write 0x55AA 5AA5 to reset the T5L CPU once. E.g. 5A A5 07 82 00 04 55 AA 5A A5 Reset T5L, clear all data, that is equivalent to power off. |
| | | | | |
| 0x06 | OS_Update_CMD | W | 2 | D3: write 0x5A to enable DWIN OS program once (write into on-chip Nor Flash). clear after CPU operation. D2: File type 0x10: must be DWIN OS program from 0x1000, update 28KB at every time. 0xA5: 8051 code, update 64KB at every time. D1:0: Start address of SRAM to save/update program to. It must be even. E.g. This instruction is used for downloading and updating the user OS program. "DWIN OS Builder" PC software has integrated the function of downloading the os.bin file into the screen through UART 2 after compilation. |
| | | | | |
| 0x08 | NOR_FLASH_RW_CMD | W | 4 | D7: Operating mode: 0x5A =read 0xA5 =write. Clear after operation. D6:4 : head address of Nor Flash. It must be even. 0x000000-0x03:FFFE, 256KWords. D3:2: head address of data variable space. It must be even. D1:0: Data length to read/write. Must be even. E.g. (1) Write the data 0x12345678 to the variable storage space 0x1002: 5A A5 07 82 10 02 12 34 56 78 (2) Store 2-word data of variable storage space 0x1002 address into Nor Flash database header address 0x000002: 5A A5 0B 82 0008 A5 00 0002 1002 0002 (3) Read data from Nor Flash database 0x00 0002 to variable storage space address 0x2000: 5A A5 0B 82 0008 5A 00 0002 2000 0002. After reading and storing the database with OS, it is necessary to judge whether the D7 address has been automatically cleared. If the D7 address has not been cleared, no operation will be carried out until it is cleared for other operations. Otherwise, logic disorder will occur. If the serial port is used for instruction reading and storage operation, other instructions can be sent after appropriate delay. |
| | | | | |

| | | | | |
|------|------------|-----|---|--|
| 0x0C | Reserved | | 3 | |
| 0x0F | Ver | R | 1 | <p>Application software version. D1 stands for GUI version, D0 stands for DWIN OS version.</p> <p>E.g.</p> <p>5A A5 04 83 000F 01 return: 5A A5 06 83 000F 01 1110 (11 is GUI software version, 10 is DWIN OS software version)</p> <p>Application :</p> <ul style="list-style-type: none"> (1) When using a display terminal, due to the control system, display terminal start time inconsistent (depending on the user's power supply capacity and electricity rate), to ensure that the control system of a start sending data display terminal received correctly and carry out the corresponding functions and instructions, so need to confirm whether the display terminal has the normal operation, the user can be identified by this directive; (2) In the process of debugging need to know whether the system version, configuration and so on is correct, you can also send this instruction to read. (3)The transmission and response of this instruction is often used to verify that the communication between the computer serial port and the serial screen is normal. |
| 0x10 | RTC | R/W | 4 | <p>D7=Year (0-0x63) D6=Month(0-0x0C), D5=Day(0-0x1F), D4=Week(0-0x6), D3=Hour(0-0x17), D2=Minute(0-0x3B), D1=Second(0-0x3B), D0 Undefined. Data format is HEX.</p> <p>Without RTC on hardware, user can write RTC data.</p> <p>E.g.</p> <p>Write: 5A A5 0B 82 0010 13 0A 01 00 0B 0C 0D 00</p> <p>Text RTC display 2019-10-0111:12:13SUN,</p> <p>The first two digits of the year are automatically identified;</p> <p>The week is displayed in English, and the system will convert automatically.</p> <p>Read: 5A A5 04 83 00 1004</p> <p>Answer: 5AA5 0C 83 00 10 04 13 0A 01 00 0B 0C 0D 00</p> |
| 0x14 | PIC_Now | R | 1 | <p>Display current page ID. Read only.</p> <p>E.g.</p> <p>Read: 5A A5 04 83 0014 01</p> <p>Answer: 5AA5 06 83 00 14 01 0007 (0007 is page 07)</p> |
| 0x15 | GUI_Status | R | 1 | <p>GUI status feedback: 0x0000=free, 0x0001=processing 13.bin and 14.bin.</p> <p>The operation state feedback is in millisecond unit level, which is generally applied in special cases. The user can judge whether the GUI kernel is occupied by the DWIN-OS program.</p> |
| 0x16 | TP_Status | | 4 | <p>D7: 0x5A=touch panel data is updated. Others=touch panel coordinates not updated</p> <p>After reading the data, the user can write ≠ 0x5A as a mark. As long as the user does not touch, the mark will not be updated</p> |

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

| | | | | |
|------|------------|-----|---|--|
| | | | | <p>write.</p> <p>1. 0: display direction 00 = 0 °, 01 = 90 °, 10 =180 °,11 = 270 °, read and write.</p> <p>E.g.</p> <p>Read: 5A A5 04 83 0080 02</p> <p>Answer: 5A A5 08 83 00 80 02 00 14 1038</p> <p>Read instructions are typically used to view the screen .CFG configuration. D0 can change the configuration. CFG file 0x05 address, for example, the next two instruction are on the corresponding.4= 1.5 =1, TP sound control off or on, can be applied to the touch button of the buzzer "mute" function.</p> <p>Note: write instruction is not saved after power off.</p> <p>Write: 5A A5 07 82 0080 5A 00 00 30</p> <p>Write: 5A A5 07 82 0080 5A 00 00 38</p> |
| 0x82 | LED_Config | 2 | | <p>Set standby backlight.</p> <p>D3= Turn on brightness, 0x00-0x64; When backlight standby control is off, D3 is the brightness adjustment interface.</p> <p>D2= Turn off brightness, 0x00-0x64; D1:0=open time /10 ms.</p> <p>E.g.</p> <p>1. When CFG address 0x05 is configured to turn on backlight standby control of touch panel:</p> <p>5A A5 07 82 0082 6432 03E8, 10 seconds after the screen backlight automatically reduced to 0x32 brightness.</p> <p>5A A5 07 82 0082 2020 ****, direct instruction control backlight brightness 0x20 brightness, and backlight screen saver brightness is consistent with standby.</p> <p>Note: the first physical touch on the touch panel after a certain period of time is to wake up the backlight function of the screen saver. Even if the brightness is the same as the setting value of standby brightness, additional "first awakening touch and click operation" is still needed after a certain period of time.</p> <p>2. When CFG address 0x05 is configured with backlight standby control over touch panel:</p> <p>5A A5 05 82 0082 0A**, direct instruction control backlight brightness is 0x0A brightness.</p> |
| 0x84 | PIC_Set | R/W | 2 | <p>D3: 0x5A = enable page operation once, The CPU is cleared after processing.</p> <p>D2: Processing Mode.</p> <p>0x01=page switch (Display the image specified in the image storage area to the current background page).</p> <p>0x02=This mode is reserved and not supported.</p> <p>Instruction example:</p> <p>5A A5 07 82 0084 5A01 0001, display the 01 serial number picture under the background picture ICL file (if the serial number picture ID of the sent instruction does not exist, the instruction operation judgment will not take effect.)</p> |

| | | | | |
|------------------|----------|-----|----|---|
| 0x86 | PWM0_Set | R/W | 2 | D3=write 0x5A to enable PWM0 setting once, The CPU is cleared after processing. |
| | | | | <p>D2= frequency division coefficient D1:D0=PWM0 precision PWM0 carrier frequency = 825.7536MHz /(frequency division coefficient * PWM0 precision).</p> <p>E.g.</p> <p>1. Write 0x0086 variable configuration 100KHz The division factor is set to 1, upper limit value=825.7536MHz/0.1MHz=8258 (0x2042). Write data 5A 01 20 42 to the variable of 0x0088, output 13 bit resolution, 100 KHz carrier PWM.</p> <p>2. Write 0x0093 variable to control the output duty ratio of PWM0 Write value = high level ratio * upper limit value For example, 10%, write value is 10% * 8258 = 826 (0x033A)</p> |
| 0x88 | PWM1_Set | R/W | 2 | <p>D3=0x5A starts a PWM1 setting, and the CPU clears it after processing. D2=frequency division coefficient D1:D0=PWM1 precision PWM1 carrier frequency=825.7536MHz/(frequency division coefficient*PWM1 precision).</p> <p>1. Write 0x0088 variable to configure 100KHz. The frequency division coefficient is 1, the upper limit=825.7536MHz/0.1MHz=8258 (0x2042) So, write data 5A 01 20 42 to 0x0088 variable, output PWM with 13bit resolution and 100KHz carrier.</p> <p>2. Write the 0x0093 variable to control the output duty cycle of PWM1 as needed. Write value = high level ratio * upper limit value. For example, 10%, write 10%*8258=826 (0x033A).</p> |
| 0x8A-0x91 | Reserved | | 10 | Undefined |
| 0x92 | PWM0_Out | R/W | 1 | D1:D0=PWM0 output high level width, 0x0000-PWM0 precision. |
| 0x93 | PWM1_Out | R/W | 1 | D1:D0=PWM1 output high level width, 0x0000-PWM1 precision. |
| 0x94-0x9B | Reserved | | 9 | Undefined |

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

| | | | | |
|------------------|---|-----|---|---|
| | | | | <p>WAE music playback setting:</p> <p>1.WAE file save location setting</p> <p>①Set the save location of the .wae file by modifying the parameters of the 0x07 address of the CFG file.</p> <p>For example: 26.wae corresponds to the 0x07 position of the CFG file, write 0x1A.</p> <p>②Configure the parameter modification of the 0x05 address of the CFG file.</p> <p>For example: write 0x7C to the 0x05 position, and the .6 bit enables music playback. (.6: Buzzer/music play selection, 0=buzzer 1=music play.)</p> <p>③Instruction format: 5A A5 07 82 00A0 03 01 40 00 Play music with ID 03 in wae file, play segment 1 segment, volume is 100%, D0: feedback is read only, write instruction can write 00 without definition .</p> |
| 0xA4-0xA9 | Reserved | | 8 | Undefined |
| 0xAA | External 16Mbytes FLASH write operation | R/W | 6 | <p>Update 16Mbytes of external memory content based on 32Kbytes blocks.</p> <p>D11: 0x5A= enable external memory operation, clear after operation.</p> <p>D10: Operation mode</p> <p>D10=0x10, read 16MB SPI NOR FLASH data</p> <p>D9:font library ID, 0x10-0x1F, 256Kbytes per font library, maximum 4Mbytes</p> <p>D8:D6: the starting address of data in the font library is defined according to the Word, 0x0000-0x01FFFF</p> <p>D5:D4: The first address of the read data variable space, which must be even.</p> <p>D3:D2: The length of the data read, defined according to the Word, must be even.</p> <p>D1:D0: undefined, write 0x00.</p> <p>For D10=0x02 write 32Kbytes data block into 16MB SPI NOR FLASH.</p> <p>D9:D8: 32Kbytes memory block address, 0x0000-0x01FF, correspond to 16Mbytes memory.</p> <p>D7:D6: The first address of the update data stored in the data variable space must be even.</p> <p>D5:D4: After the completion of this operation, the time of delay waiting for the next write operation, unit: 1ms.</p> <p>DGUS refresh will stop during the delay wait to prevent errors caused by incomplete updates.</p> <p>D3:D0: Undefined, write 0x00.</p> <p>D10=0x03, font copy</p> <p>D9:D8: Source font ID, 0x0000-0x001F, 256Kbytes per font library, max 4Mbytes.</p> <p>D7:D6: Target font ID, 0x0000-0x003F.</p> <p>D5-D4: Number of font to copy, 0x0000-0x0010.</p> <p>D3:D0: Undefined, write 0x00.</p> <p>D10=0x04, fD10=0x04, write multiple 4KB data blocks to NAND FLASH (NAND Flash must be externally expanded).</p> |

| | | | |
|--|--|--|---|
| | | | D9:D6: The starting address of the NAND Flash to be written, the lower 12 bits are 0 (4KB alignment). If the lower 18 bits of the starting address are all 0s, the T5L will first erase the 256KB block before writing. D5:D4: The write data is stored in the first address of the data variable space, which must be an even number. D3: Number of 4KB blocks written, 0x01-0x10. D2:D0: undefined, write 0x00. D10=0x05, 32-bit CRC check of NAND Flash data (NAND Flash must be externally expanded). D9:D6: NAND source address, 0x0000:0000-0x1FFF:F000, the lower 12 bits are 0 (4KB alignment). D5:D4: The number of 4KB data blocks to be checked, 0x0001-0x8000, the maximum is 128Mbytes. D3:D0: 32-bit CRC check return value, the CRC check polynomial is the same as the SD download CRC check. D10=0x06, NAND Flash Copy (NAND Flash must be externally expanded). D9:D6: NAND source address, 0x0000:0000-0x1FFF:F000, the lower 12 bits are 0 (4KB alignment). D5:D4: Target font ID, 0x0000-0x00FF. 0x00-0xBF is SPI NOR Flash, the size of a single font is 256Kbytes; 0x0C0-0xFF are SPI NAND Flash, and the size of a single font is 8Mbytes. D3:D2: The number of fonts to be copied. D1:D0: undefined, write 0x00. If the copy is accidentally interrupted by power failure, it will automatically continue until it is completed when the power is restarted. E.g. The address operation can realize the update function of serial port 2, such as picture library, word library, etc. The 16Mbytes memory can be seen as either 512 32Kbytes space for writing operation data update or 64 single font ID space with fixed capacity of 256KB. The following instruction 00B0 can also be seen as representing update to 22 fontID. |
|--|--|--|---|

| | | | | |
|-------------|-------------------------|---|---|--|
| | | | | <p>0xB0: 0x5AA5= enable accessing touch control interface once. Clear after CPU operation.</p> <p>0xB1: Page ID of touch control.</p> <p>0xB2: High byte: touch control ID (set in DGUS II development software), 0x01-0xFF;</p> <p>Low byte: touch control code, 0x00-0x7F.</p> <p>0xB3: Access mode</p> <p>0xB4-0xD3: data to modify of mode 0x02, 0x03.</p> <p>Mode 0x0000: turn off this touch control.</p> <p>Mode 0x0001: turn on this touch control.</p> <p>Mode 0x0002: Read this touch control and write it to SRAM that 0xB4 pointing to.</p> <p>Mode 0x0003: update current touch control with data that 0xB4 pointing to, the format and data length must be the same.</p> <p>E.g. 5AA5 0B 82 00B0 5AA5 0001 0905 0000 (close the 9th touch file on page 0001. 05 represents the instruction code, 0000 represents to turn off the touch control.) Basic touch control cannot be turned on/off. If it is ranked in PC software, it does not need to be counted as effective touch number.</p> <p>If you need to change the specific properties of the touch, you need to use mode 03 to replace the 0xb4 prewritten touch file in the property configuration of the 13bin file. Search for the key words of 0xB0 in the DWIN forum, with detailed post description.</p> |
| 0xD4 | TP operation simulation | W | 5 | <p>0xD4: 0x5AA5=enable the operation once, clear after operation.</p> <p>0xD5: press mode. 0x0001=press, 0x0002=release, 0x0003=keep pressing, 0x0004=touch (press + release)</p> <p>0xD6: X coordinate of press position.</p> <p>0xD7: Y coordinate of press position.</p> <p>After simulating mode 0x0001 and 0x0003, must simulate 0x0002.</p> <p>When the assembly touch function is running, x= 0xAA:KH y=0xA5:KL will directly return the key values KH and KL to the assembly touch.</p> <p>For example, when variables are input in assembly mode, the coordinates (0xAAF0, 0xA5F0) will cause the input to end immediately.</p> <p>When the 13 touch file is designed with keying function, X coordinate= 0xFF: key code y coordinate = 0x0001 will trigger the corresponding keying function.</p> <p>E.g.</p> <p>5AA5 0B 82 00D4 5AA5 0004 00EE 008F</p> <p>0004 is click, press + lift, 00EE 008F (283,143) coordinates,</p> <p>Press mode 0x0001= press; 0x0002= release; 0x0003= continue pressing</p> <p>0x0004=click</p> <p>After applying the simulated lift modes 0x0001 and 0x0003, there must be a simulated lift mode of 0x0002.</p> <p>(after "touch panel sound control" is enabled, touch panel operation simulation will also trigger the touch panel buzzer)</p> <p>(after "touch panel backlight standby control" is enabled, touch panel operation simulation will also wake up backlight)</p> |

| | | | |
|-------------|------------------------------|-----|---|
| 0xD8 | Pointer icon overlay display | 4 | <p>0xD8_H: 0x5A = enable the overlay display.</p> <p>0xD8_L: the position of ICL file which the pointer icon is saved at.</p> <p>0xD9: the pointer icon ID.</p> <p>0xDA: the X coordinates of pointer icon.</p> <p>0xDB: the Y coordinates of pointer icon.</p> <p>The pointer icon is always displayed in the background filtering mode, and the background filtering intensity is fixed at 0x08.</p> <p>E.g.</p> <p>5AA5 0B 82 00D8 5A 2D 0001 0064 0064 call the 1 Icon of 45.icl icon to display directly on the screen page position (100100).</p> <p>Note: After switching page, icon overlay will not disappear.</p> <p>write 00 to 0xD8_H or call the blank position icon to make the icon disappear.</p> <p>The ID of the icon can be selected by the user to achieve the image display effect of "click" and "wait" effect.</p> |
| 0xDC | Reserved | 4 | Undefined |
| 0xE0 | Memory CRC check | R/W | <p>D3: write 0x5A to enable the CRC checking once, clear after operation.</p> <p>D2: memory type selection</p> <p>0x00= font space (16Mbytes memory)</p> <p>0x02=DWIN OS code</p> <p>0x03=Nor flash database (LIB file).</p> <p>D1:D0: data interface.</p> <p>► Enable the CRC</p> <p>Font check mode: D1= start font ID (256KB for each font); D0= the number of 4KB blocks checked, 0x00-0xFF.</p> <p>OS code check mode: D1:D0= starting at 0x1000, the byte length of OS code to be checked is 0x0001-0x7000.</p> <p>Nor flash database mode: D1:D0=Nor flash database ID, fixed validation of 4KB data each time.</p> <p>► After the check</p> <p>Returned CRC value.</p> |

| | | | | |
|-----------------------|--|---|----|---|
| 0xE2 | Export the monochrome bitmap of the specified area on the screen | R | 6 | 0xE2_H: 0x5A A monochrome bitmap export is started and the CPU is cleared. 0xE2_L: Write 0x00; 0xE3: X coordinates of the upper left corner of the screen area; 0xE4: Y-coordinate of the upper left corner of the screen area; 0xE5: X coordinates in the lower right corner of the screen area; 0xE6: Y coordinate in the lower right corner of the screen area; 0xE7: The start address of variable storage stored in the output bitmap must be an even number. Bitmap data stored in variable storage, aligned to lines according to MSB, double word pattern. |
| 0xE8- 0xFF | Reserved | | 14 | Undefined |
| 0xF0 | Interface of playing music flow data | W | 4 | <p>D7: 0x5A = enable the music playback operation, clear after operation. D6: mode, 0x00 = stop(clear the buffer), 0x01 = suspend(reserve the buffer), 0x02 = start. D5:D4: undefined, write 0x00. D3:D2: Variable memory address for storing music data, even. D1:D0: Music data word length, maximum 8KWords, even; data is 16bit integer format.</p> <p>This instruction is used to play online music files. First, send the musicflow data to a variable address between 0x1000 and 0xFFFF. For example, the buffer address is 0x8000, Instruction sending format: 5A A5 0B 82 00F0 5A 02 00 00 8000 0400 Play music saved start from 0x8000, instruction with 1K musicdata.</p> |

| | | | | | |
|------|--------------------|---|---|--|---|
| | | | | | D15: 0x5A = enable the drawing touch window. D14: operation mode, 0x00 = normal mode, 0x01 = initialization (clear after initialization). D13: parameter configuration. 7-2 reserved, write 0. 1-0 painting buff processing mode after page changing. 0x00 = close, 0x01 = re-initialization, others = unchanged. D12: reserved, write 0x00. D11: line width, from 0x01 - 0x0F. D10:D8: painting color, D6=RED D5=GREEN D4=BLUE D7:D4: the coordinate position of the window upper left corner(x, y). D3:D2: the width of the pixel points window, must be divisible by 4 D1:D0: the height of the pixel points window, must be divisible by 4 |
| 0xF4 | Painting interface | W | 8 | | <p>The variable memory occupied by drawing touch window is 128KB(Double word address range 0x00:8000-0x00: FFFF) which is corresponding to 208*208 pixel.</p> <p>This instruction is used after touching screen. Display the track of coordinates in the designated area of the screen, which is generally used in "handwriting drawing function".</p> <p>E.g.</p> <p>0x00 mode=normal mode</p> <p>1.Send drawing window instruction on a page: Tx:5AA5 13 82 00 F4 5A 00 00 00 02 FF 00 00 00 2C 00 BC 00 D0 00 D0</p> <p>2.Hand touch panel drawing graphics</p> <p>3.Switch page</p> <p>4.Switch back to the previous page, send the drawing window instruction again, and the characters will be displayed again. In the upper left corner of the window (44,188), the coordinate point corresponds to the largest 208*208 pixel drawing window.</p> <p>0x01 mode: initialization mode</p> <p>1. Send a drawing window instruction on a page, Tx: 5A A5 13 82 00 F4 5A 01 00 00 02 FF 00 00 00 2C 00 BC 00 D0 00 D0</p> <p>2. Hand touch screen to draw graphics,</p> <p>3. Switch pages</p> <p>4. Switch back to the previous page, and send the drawing window instruction again, the characters are no longer displayed, and the blank is convenient for redrawing the graphics.</p> |
| 0xFC | DGUS_STOP_EN | W | 2 | | <p>Writing 0x55 AA 5A A5 will stop the DGUS refresh, which is used to avoid the conflict of rewriting fonts when there is no backup area to upgrade.</p> <p>Write 0x55 AA 5A AA will stop DGUS refresh, stop OS core running, used for ED4 download fixed occupancy variable buffer</p> <p>The buffer area is 0x8000-0xFFFF, to prevent DGUS refresh or OS core operation from rewriting variables to cause upgrade data errors.</p> |

| | | | | |
|----------------------|---------------------------------|-----|------|--|
| 0xFE | UART1 high speed download | | 2 | 0x5AA5=enable the high-speed download operation through UART1 |
| 0x100-0x2FF | Reserved | | 512 | Undefined |
| 0x300-0x37F | Dynamic curve interface | R/W | 128 | <p>0x300-0x30F: state feedback for 8 channel curve buffers (read only advised), 2 words per channel, high word is the storage pointer location (0x0000-0x07FF) where curve data is stored, and low word is the effective data length of curve buffer (0x0000-0x0800). Writing 0x0000 to the effective data length of the curve buffer will cause the curve unable display.</p> <p>0x310-0x311: start writing curve bufferdata</p> <p>D3: D2:0x5AA5 enable the writing curve buffer data operation once, and clear after operation.</p> <p>D1: the number of data blocks, 0x01-0x08.</p> <p>D0: undefined, write 0x00.</p> <p>0x312-0x37F: data block written to the curve buffer, which is 16 bits unsigned.</p> <p>Single data block: data channel ID (0x00-0x07) + data word length (0x01-0x6E) + data.</p> <p>With dynamic curve display enabled, start at 0x1000 and create a data buffer for each curve according to 2Kwords perchannel.</p> <p>The CH0 buffer is 0x1000-0x17FF, the CH1 buffer is 0x1800-0x1FFF, and so on, the unused curve buffer zones can be used as user variables. Also users can directly overwrite the curve buffer data and then modify 0x300-0x30F corresponding storage pointer position and data length to ensure the correct display of the curve.</p> <p>Instruction example: See 7.4.1 Real-time Curve (Trend) display.</p> |
| 0x380-0x3FF | Reserved | | 128 | Undefined. Not available for users. |
| 0x400-0x4FF | Network communication interface | R/W | 256 | WiFi and other Internet communication equipment application control interface. |
| 0x500 - 0x5BF | Multi-Media interface | R/W | 192 | Multi-Media application interface, 0x500-0x57F digital Multi-Media interface, 0x580-0x5BF analog Multi-Media interface. |
| 0x5C0-0x5FF | External memory interface | R/W | 64 | External memory interface (such as U disk) read or write interface. |
| 0x600-0xEFF | Reserved | | 2404 | Undefined. Not available for users. |

| | | | | |
|---------------------|----------------------------|---|-----|--|
| | | | | After setting variables to change the automatic upload function, this function is enabled. D3=5A means variable change, D2:D1= variable memory pointer, D0= variable length (word). E.g. It is used to judge whether the touch data is uploaded to the serial port, or whether there is a touch control operation change for uploading. Because the time of this state is very short, it can't be read with the serial port 0x83 instruction. It is recommended to read the value of D3 with DWIN OS and then judge. LDWR R0,0F00H IJNE R10,5AH,TEXT; Judge the value of 0x5A and execute it down, if not jump to TEXT Application example 2: It is used to judge whether the data of different addresses is changed, and OS can also be used for judgment processing. Send: 5A A5 04 83 0F 00 02 Response: 5A A5 08 83 0F 00 02 00 11 0001 means that the variable address 0x1100 has uploaded 1 word of data. Send: 5A A5 04 83 0F 00 02 Response: 5A A5 08 83 0F 00 02 0010 00 02 means that the variable address 0x1000 has uploaded 2 words of data. |
| 0x0F00 | Variable change indication | R | 2 | |
| 0xF02-0xFFFF | Reserved | | 254 | Undefined, User cannot use. |

0x1000-0xFFFF are available for users.

5.2 Network Interface

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

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

DWIN DGUS II platform (including T5 series and T5L series) has open network interface with WIFI module. Through simple DGUS development, you can access DWIN cloud. Based on DWIN's cloud platform, customers can realize remote APP control, data analysis, equipment operation and other functions.

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

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

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

| | | | |
|-------|----|---|--|
| 0x413 | 1 | | The classification of each manufacturer's equipment is according to the DWIN classification standard equipment classification. After the device is added to the DWIN cloud platform, it is automatically generated by the platform. |
| 0x414 | 2 | | The individual device number of each type of device. After the device is added to the DWIN cloud platform, it is automatically generated by the platform. |
| 0x416 | 26 | 44475553325F5 43555494431 (behind all 00) | Device description text, up to 52Bytes. Recommended value description: 44475553325F5435555494431, corresponding to the ASCII code character DGUSII_T5UID1, has been written as a fixed value written in 22 file. The user can also make a text variable address association for display. |

| | | | |
|--|--|--|--|
| | | | |
|--|--|--|--|

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

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

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

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

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

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

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

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

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

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

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

Chapter 6 Touch Variable Configuration (13.BIN)

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

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

| Number | Definition | Length | Definition |
|--------|------------|---------|---|
| 1 | Pic_ID | 2 | Page ID, the high 4bit is the audio ID selection of this button, 0x00 means without touch voice. |
| 2 | TP_Area | 8 | The touch button area coordinates the upper left corner (Xs, Ys) and the lower right corner (Xe, Ye). Setting the button region outside the screen resolution simulates triggering touch-screen buttons in the system variable interface with specific key values. |
| 3 | Pic_Next | 2 | Target page ID after button pressing, 0xFF** indicates without page switching. |
| 4 | Pic_On | 2 | ID of the page where the effect picture is when the button is pressed, 0xFF** indicates without pressing effect. |
| 5 | TP_Code | 2 | Touch control code: 0xFF** indicates an invalid key code. 0xFE** or 0xFD** indicates a touch button. For example, 0xFE00 indicates the touch screen digital input is enabled. 00 is the touch control code value, which can be named by referring to the touch control code number in the 6.1 List of Touch Variables. 0xFE** function key can be set by CFG file 0x05.4 to upload or not. The function button of 0xFD** always prevents automatic uploading after variable changes. Other touch control codes are represented by ASCII. For example, 0x0031 indicates key 1. |
| 6 | TP_FUN | 0/16/32 | Touch control code: When TP_Code=0xFE**, this parameter is used to describe the touch button. |

6.1 List of Touch Variables

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

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

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

6.2 Variable Data Input

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

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

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

6.2.1 Instruction Storage Format

| Address | Definition | Length(byte) | Description |
|---------|-------------|--------------|--|
| 0x00 | Pic_ID | 2 | Page ID |
| 0x02 | TP_Area | 8 | Area of the Control: upper left corner (Xs, Ys), lower right corner(Xe, Ye). |
| 0x0A | Pic_Next | 2 | Target switch page, 0xFF** means no page switch |
| 0x0C | Pic_On | 2 | Button press effect page, 0xFF** means no button press effect. |
| 0x0E | TP_Code | 2 | 0xFE00, variable data input key code. |
| 0x10 | 0xFE | 1 | 0xFE |
| 0x11 | VP | 2 | Variable pointer |
| 0x13 | V_Type | 1 | Return variable type: 0x00=2bytes: Integer: -32768 to 32767; Unsigned integer: 0 to 65535 0x01=4bytes: Long integer: -2147483648 to 2147483647 Unsigned long integer: 0 to 4294967295 0x02=*VP high byte, unsigned number: 0 to 255 0x03=*VP low byte, unsigned number: 0 to 255 0x04=8bytes, extra long integer: -9223372036854775808 to 9223372036854775807 |
| 0x14 | N_Int | 1 | Integer digits. For example, input 1234.56, and N_Int=0x04 |
| 0x15 | N_Dot | 1 | Decimal digits. For example, input 1234.56, and N_Int=0x02 |
| 0x16 | (x, y) | 4 | Display location: right alignment, (x, y) is the upper right coordinate of the string input cursor. Undefined when using configuration keyboard (KB_Source=0x0F) |
| 0x1A | Color | 2 | Display color Undefined when using configuration keyboard (KB_Source=0x0F) |
| 0x1C | Lib_ID | 1 | ASCII Font position, default 0x00 Undefined when using configuration keyboard (KB_Source=0x0F) |
| 0x1D | Font_Hor | 1 | Font size .0x00 means black, otherwise it is white Undefined when using configuration keyboard (KB_Source=0x0F) |
| 0x1E | Cusor_Color | 1 | Cursor color. Undefined when using configuration keyboard (KB_Source=0x0F) |

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

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

6.2.2 Software Settings

Variable Data Input

| | | |
|--|-------|--|
| X 421 | Y 165 | Coordinates of the upper left corner of the touch area |
| W 195 | H 68 | Touch area width and height |
| <input type="checkbox"/> Key value(0x) 0 | | Can be set by default |
| <input type="checkbox"/> Voice ID 0 | | Voice ID number |

Name Data Input
 Automatic data upload

Button effect
 -1 Specify
 No button effect

Page switching
 -1 Specify
 No page switching

Variable address(0x) 5420
Variable type integer (word)
 Integer digits 5
 Decimal places 0
Display position 657 182 Set up
Display color 0000
Font library location 0
Font size 30 4-255
Cursor color Black
Input display mode Direct display
Keyboard location Other pages

Keyboard settings

| | |
|--|---|
| On the page | 30 |
| Keyboard area | 215 158 506 533 |
| Display position | 401 161 Set up |
| Background transparency | 0 |
| Enable range limit | Lower limit -100 Upper limit 100 |
| <input checked="" type="checkbox"/> Load data during entry | Data address 8003 Download Data 2047 |

Checking means that the input is completed and the data is uploaded. At the same time, the 0X05.4 bit of the CFG file needs to be written to enable the data upload, and the 0X05 bit can be written to 0X38, which means that the 22 initial value file is loaded; the data is uploaded, and the touch sound is turned on.

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

Can be set by default

Variable storage space users can use address range arbitrarily: 0x1000-0xFFFF. Variable data entry and data type occupies 1/2/4 addresses. That is, non-associated control keys need to be spaced apart. System variable interface address range: 0x0000-0x0FFF (if the page switching address is 0x0084).

The number of integer digits displayed during entry.
The number of decimal places displayed during entry.

The coordinates of the data display during entry.
Enter the position of the upper right corner of the display area.

The size of font 0 is 4*8-64*128, dot matrix in X direction, the size is appropriate, and the color can be customized.

Direct display is more intuitive, and “***” can be used for password input.

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

To call the keyboard of the current page, you need to select the current page, and to call other keyboards, you need to select other pages.

The coordinates of the upper left corner of the selected keyboard area.
It is recommended to enter the position of the upper right corner of the display area for the coordinates of the data display during the entry process.

The corresponding data keyboard page needs to be defined by the basic touch setting Key value, and the key value is set under the touch attribute.
Only basic touch controls can be placed on the pop-up area, and touch keys such as key value return and sliding page turning cannot be placed.

The coordinates of the lower right corner of the selected keyboard area.

The coordinates of the upper corner of the pop-up keyboard on the current page. Too far to the right or incomplete display of the keyboard in the lower right corner.
The pop-up keyboard is similar to the transparency of frosted glass, with 0 being opaque.

For example, after checking here, the data can only be entered in the range of -100 to 100.

It is mainly used to combine with the SP (description pointer) modification of the variable display to realize the automatic labeling of the multi-parameter input process.
For example, modify the font color, size, start a (bit) variable icon or area inversion.
0x8003 is the data variable display description pointer 0x8000 address offset by 3 bits, used to change the data color.
2047 is the color value 0x07FF (fluorescent green). It needs to be converted into decimal system. If there is no requirement, it can be unchecked by default.
For example, the data value loaded with the data address determines whether the keyboard is in the input state.

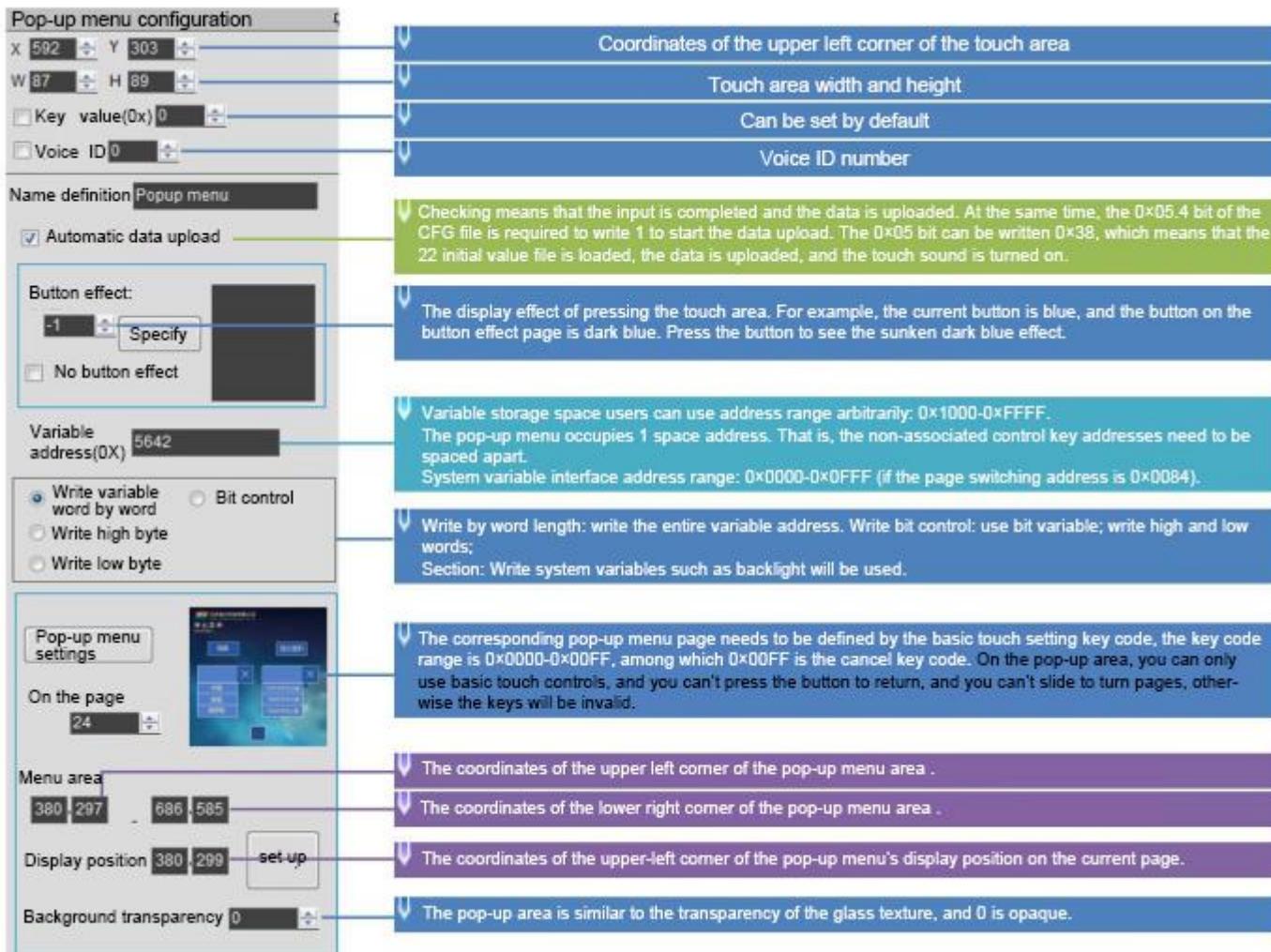
6.3 Popup Menu

6.3.1 Instruction Storage Format

| Address | Definition | Length(byte) | Description |
|-------------|-----------------|--------------|---|
| 0x00 | Pic_ID | 2 | Page ID |
| 0x02 | TP_Area | 8 | Area of the control: upper left corner (Xs, Ys), lower right corner (Xe, Ye). |
| 0x0A | Pic-Next | 2 | Switch target page. 0xFF** means no page switch. |
| 0x0C | Pic_On | 2 | Button press effect page. 0xFF** means no button press effect. |
| 0x0E | TP_Code | 2 | 0xFE01, popup window key code. |
| 0x10 | 0xFE | 1 | 0xFE |
| 0x11 | VP | 2 | Variable pointer |
| 0x13 | VP_Mode | 1 | 0x00: the returned key code is saved to VP (integer) 0x01: the returned key code is saved to high byte of the VP(VP_H) 0x02: the returned key code is saved to low byte of the VP(VP_L) 0x10-0x1F: Write the lowest bit of the returned key code (1bit) to specific bit of VP(0x10 corresponds to VP.0; 0X1F corresponds to VP.F) |
| 0x14 | Pic_Menu | 2 | Page ID of the menu |
| 0x16 | Area_Menu | 8 | Menu area: upper left coordinate (Xs, Ys), lower right coordinate (Xe, Ye). |
| 0x1E | Menu_Position_x | 2 | Upper left corner x-coordinate of the current page |
| 0x20 | 0xFE | 1 | 0xFE |
| 0x21 | Menu_Position_Y | 2 | Upper left corner y-coordinate of the current page |
| 0x23 | Translucent | 1 | The transparency setting. The background transparency can be set within 0x00~0xFF. 0x00 means 100% opaque. |
| 0x24 | NULL | 12 | Write 0x00 |

6.3.2 Software Settings

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



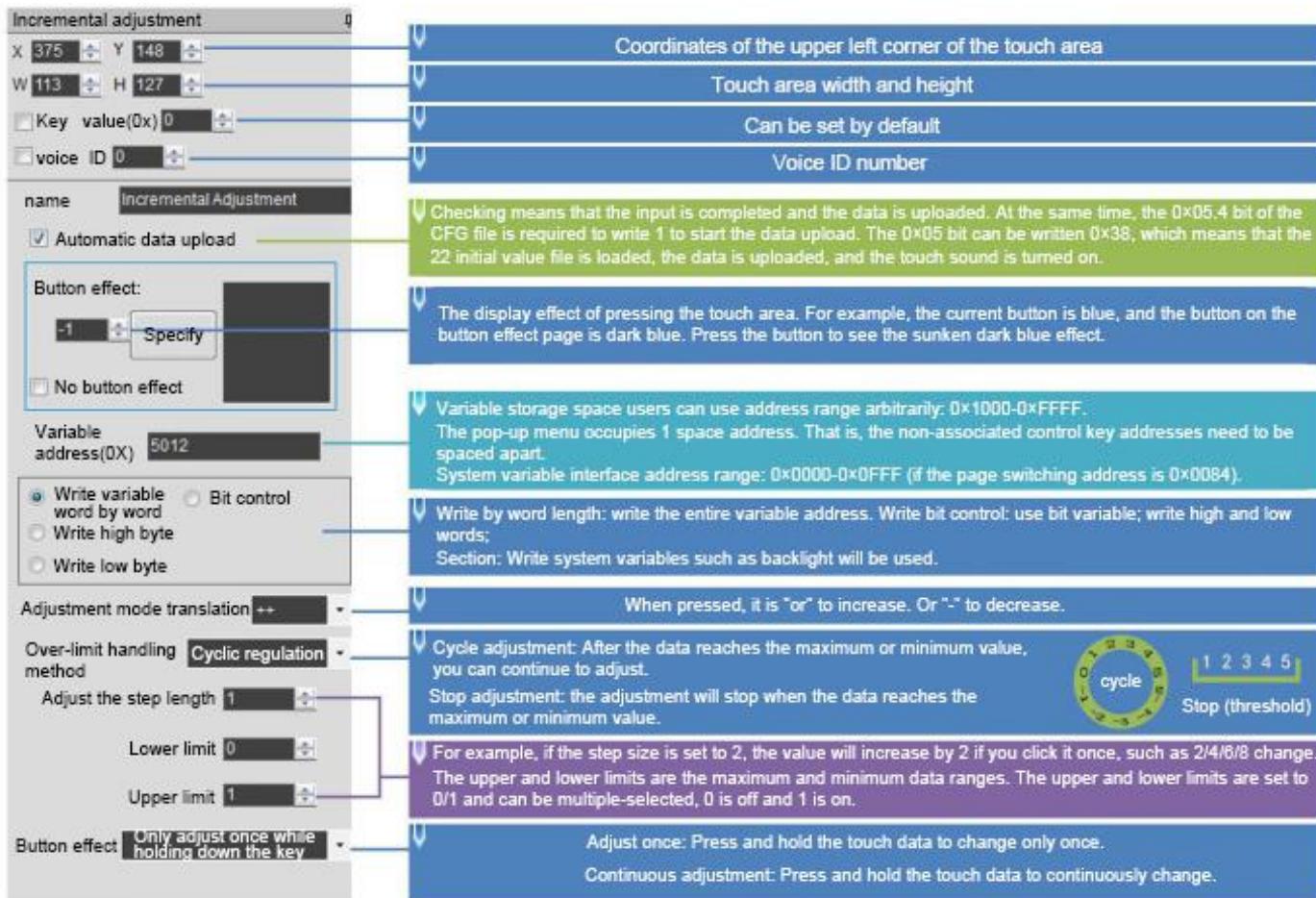
6.4 Incremental Adjustment

6.4.1 Instruction Storage Format

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

6.4.2 Software Settings

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



6.5 Drag Adjustment

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

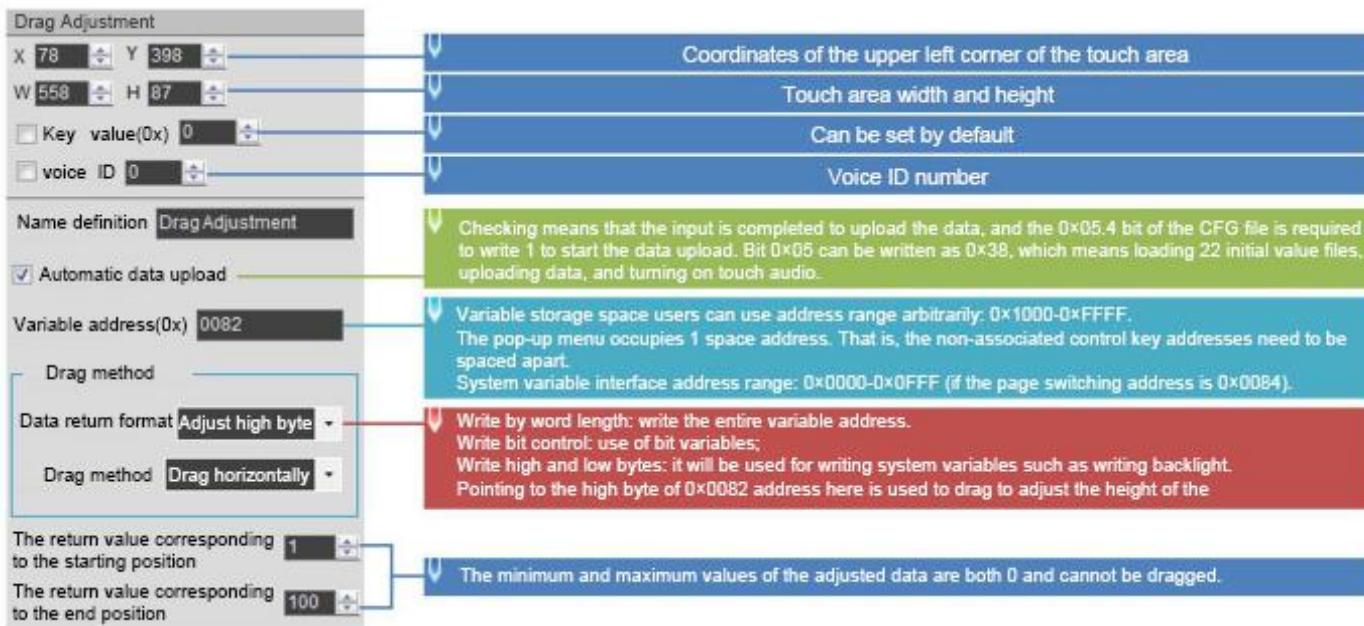
The advantage is that it is intuitive, fast, and the parameters do not cross the boundary.

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

6.5.1 Instruction Storage Format

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

6.5.2 Software Settings



6.6 Return Key Code

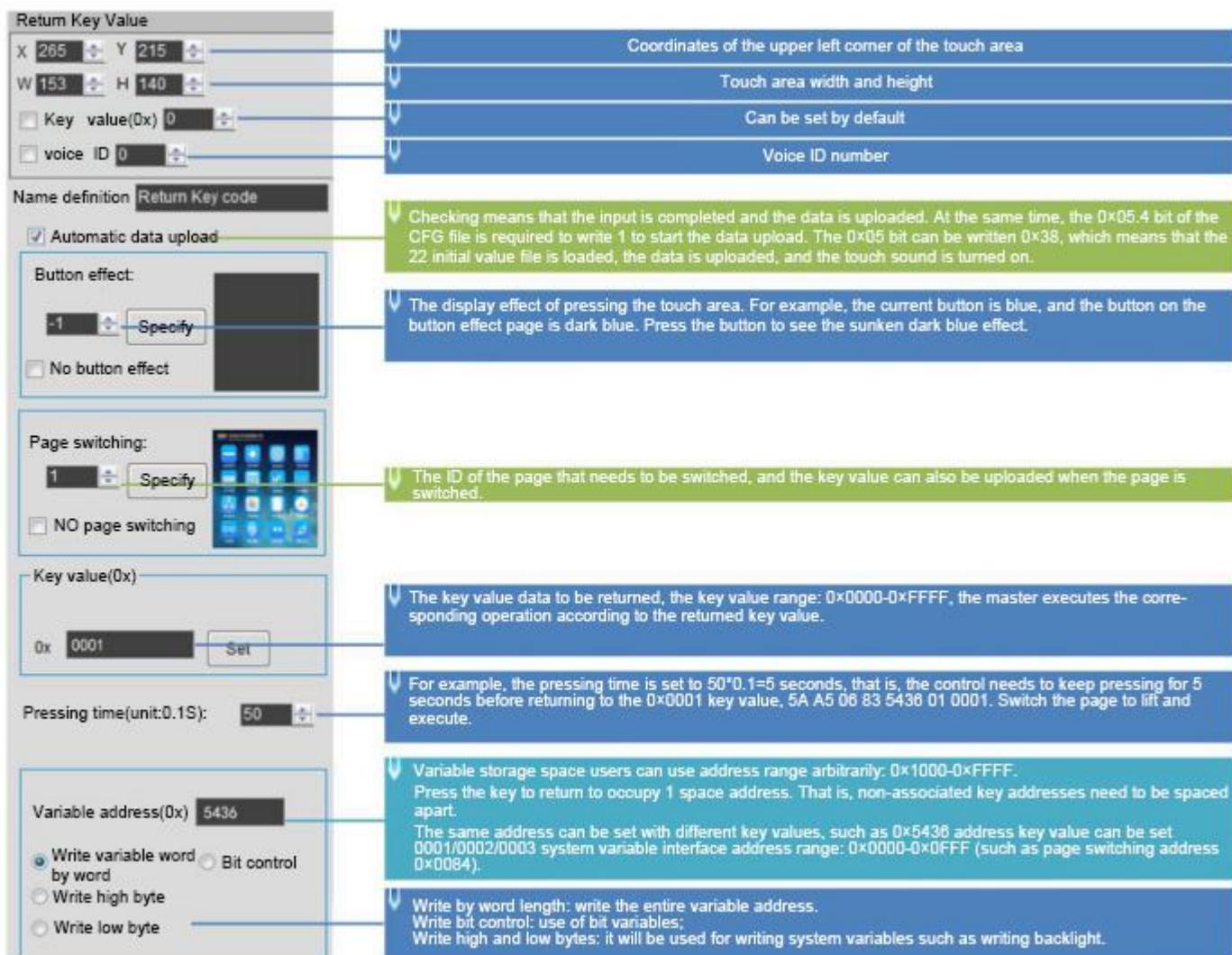
6.6.1 Instruction Storage Format

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

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

6.6.2 Software Settings

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



Return data format:

5A A5 06 83 5436 01 0001

0xA5: 5A Frame header;

0x06: Data length;

0x83: Read instruction;

0x5436: VP;

0x01: word length of the returned data;

0x0001: returned key value of 0001.

6.7 Text Input

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

The transparency of ASCII input keyboard is adjustable.

The keyboard key code is as the table below.

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

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

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

| Key Code | Normal | Capital |
|----------|--------|---------|----------|--------|---------|----------|--------|---------|----------|--------|---------|
| 0x7E60 | ' | ~ | 0x5171 | q | Q | 0x4161 | a | A | 0x5A7A | z | Z |
| 0x2131 | 1 | ! | 0x5777 | w | W | 0x5373 | s | S | 0x5878 | s | S |
| 0x4032 | 2 | @ | 0x4565 | e | E | 0x4464 | d | D | 0x4363 | c | C |
| 0x2333 | 3 | # | 0x5272 | r | R | 0x4666 | f | F | 0x5676 | v | V |
| 0x2434 | 4 | \$ | 0x5474 | t | T | 0x4767 | g | G | 0x4262 | b | B |
| 0x2535 | 5 | % | 0x5979 | y | Y | 0x4868 | h | H | 0x4E6E | n | N |
| 0x5E36 | 6 | ^ | 0x5575 | u | U | 0x4A6A | j | J | 0x4D6D | m | M |
| 0x2637 | 7 | & | 0x4969 | i | I | 0x4B6B | k | K | 0x3C2C | , | < |
| 0x2A38 | 8 | * | 0x4F6F | o | O | 0x4C6C | l | L | 0x3E2E | . | > |
| 0x2839 | 9 | (| 0x5070 | p | P | 0x3A3B | ; | ; | 0x3F2F | / | ? |
| 0x2930 | 0 |) | 0x7B5B | [| { | 0x2227 | ' | " | 0x2020 | SP | SP |
| 0x5F2D | - | _ | 0x7D5D |] | } | 0x0D0D | Enter | Enter | | | |
| 0x2B30 | = | + | 0x7C5C | \ | | | | | | | |

Table 6.7 Keyboard function key code definition

| Key Code | Function | Description |
|----------|-----------|---|
| 0x00F0 | Cancel | Cancel input code return, not affecting the variable data. |
| 0x00F1 | Return | Confirm input code return, and the input text is saved to the specified VP. |
| 0x00F2 | Backspace | Delete one character(backspace). |
| 0x00F3 | Delete | Delete one character backwards. |
| 0x00F4 | CapsLock | Caps lock. If enabled, the corresponding button must define the effect of the button press, i.e. there must be a press effect page in the project . |
| 0x00F7 | Left | Move the cursor forward by one character; it is used to turn pages in GBK Chinese character input. |
| 0x00F8 | Right | Move the cursor back by one character; it is used to turn pages in GBK Chinese character input. |

6.7.1 ASCII Text Input

6.7.1.1 Instruction Storage Format

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

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

6.7.1.2 Software Setting

ASCII input

| | | |
|--|-----------------------|--|
| X 404 | Y 145 | Coordinates of the upper left corner of the touch area |
| W 207 | H 210 | Touch area width and height |
| <input type="checkbox"/> Key value(0x) 0 | Can be set by default | |
| <input type="checkbox"/> voice ID 0 | Voice ID number | |

Name definition ASCII Text Input

Automatic data upload

Button effect:

-1 The display effect of pressing the touch area. For example, the current button is blue, and the button on the button effect page is dark blue. Press the button to see the sunken dark blue effect.

No button effect

Page switching:

-1 The ID of the switched page can be set by default

NO page switching

Variable address(0x) 5240

Text length 50

Entry mode Re-enter

Font library location 0

Number of dots in X direction 24

Number of dots in Y direction 48

Cursor color black

Text color 0000

Input status return

Input display area

56, 191 669, 294

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

Keyboard location Other pages

Keyboard settings

On the page 10

Keyboard area 29, 155 692, 590

The coordinates of the upper left corner can be set consistent

Display position 32, 167

Background transparency 0

Coordinates of the upper left corner of the touch area

Touch area width and height

Can be set by default

Voice ID number

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

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

The ID of the switched page can be set by default

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

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

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

ASCII characters use font size 0 by default, and the X,Y dot matrix double relationship.

Cursor black or white

The display color of the text during the input process.

Pop up keyboard page

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

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

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

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

The called keyboard displays the coordinates in the upper left corner of the current page. Transparency: 0 opacity; similar to the transparency of glass texture.

Keycode

| Keycode | Description |
|-----------|-------------|
| 1 | 1 |
| 2 | 2 |
| 3 | 3 |
| 4 | 4 |
| 5 | 5 |
| 6 | 6 |
| 7 | 7 |
| 8 | 8 |
| 9 | 9 |
| 0 | 0 |
| - | - |
| = | = |
| a | a |
| b | b |
| c | c |
| d | d |
| e | e |
| f | f |
| g | g |
| h | h |
| i | i |
| j | j |
| k | k |
| l | l |
| m | m |
| n | n |
| o | o |
| p | p |
| q | q |
| r | r |
| s | s |
| t | t |
| u | u |
| v | v |
| w | w |
| x | x |
| y | y |
| z | z |
| Space | Space |
| Enter | Enter |
| Cancel | Cancel |
| Return | Return |
| CapsLock | CapsLock |
| Delete | Delete |
| Left | Left |
| Right | Right |
| Backspace | Backspace |

Def Keycode 0000 Add

Description

Select Cancel

Key value(0x) 0x 5070 Set

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

6.7.2 GBK Input

GBK input is used to input Chinese character text.

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

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

6.7.2.1 Instruction Storage Format

| Address | Definition | Length (byte) | Description |
|-------------|--------------|---------------|--|
| 0x00 | Pic_ID | 2 | Page ID |
| 0x02 | TP_Area | 8 | Area of the control: upper left corner (Xs, Ys), lower right corner (Xe, Ye). |
| 0x0A | Pic_Next | 2 | Switch target page. 0xFF** means no page switch. |
| 0x0C | Pic_On | 2 | Button press effect page. 0xFF** means no button press effect |
| 0x0E | TP_Code | 2 | 0xFE06, GBK input key code. |
| 0x10 | 0xFE | 1 | 0xFE |
| 0x11 | VP | 2 | Variable pointer |
| 0x13 | VP_Len_Max | 1 | Max text length, in words (two characters for each word). Range: [0x01,0x7B]. When the text is saved in the specific VP, it will add the 0xFFFF as terminator automatically at the ending of the text. Thus the variable space is actually VP_Len_Max+1 |
| 0x14 | Scan_Mode | 1 | Input mode. 0x00: re-input, 0x01: modify existing text. |
| 0x15 | Lib_GBK1 | 1 | Index in the FLASH memory of the GBK Font to use after entry. 0x00=default. |
| 0x16 | Lib_GBK2 | 1 | Index in the FLASH memory of the GBK Font to use before entry. |
| 0x17 | Font_Scale1 | 1 | Lib_GBK1 font size, lattice number |
| 0x18 | Font_Scale2 | 1 | Lib_GBK2 font size, lattice number |
| 0x19 | Cursor_Color | 1 | Color of the input cursor. 0x00: Black Other Values: White. |
| 0x1A | Color0 | 2 | Text color after entry |
| 0x1C | Color1 | 2 | Text color during entry |
| 0x1E | PY_Disp_Mode | 1 | The display of pinyin hints and corresponding Chinese characters during entry. * 0x00 = pinyin hints are displayed on the top and the corresponding Chinese characters are displayed on a separate line at the bottom. Pinyin hints and Chinese characters are displayed left-aligned, with Scan_Dis line spacing. * 0x01=Pinyin hints are displayed on the left, and the corresponding Chinese characters are displayed on the right. The starting display x position of the Chinese character is: Scan1_Area_Start+3 x Font_Scale2+Scan_Dis. |

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

Note

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

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

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

| Notice | ㄅ | ㄉ | ㄇ | ㄔ | ㄉ | ㄈ | ㄎ | ㄉ | ㄉ | ㄉ | ㄏ | ㄉ |
|-------------------|------|------|------|------|------|------|------|------|------|------|------|------|
| Key Code | 0xc5 | 0xc6 | 0xc7 | 0xc8 | 0xc9 | 0xca | 0xcb | 0xcc | 0xcd | 0xce | 0xcf | 0xd0 |
| Phonetic Key Code | ㄄ | ㄒ | ㄓ | ㄁ | ㄎ | ㄖ | ㄕ | ㄔ | ㄙ | ㄧ | ㄨ | ㄩ |
| | 0xd1 | 0xd2 | 0xd3 | 0xd4 | 0xd5 | 0xd6 | 0xd7 | 0xd8 | 0xd9 | 0xe7 | 0xe8 | 0xe9 |
| Phonetic | ㄚ | ㄛ | ㄜ | ㄝ | ㄞ | ㄟ | ㄢ | ㄡ | ㄩ | ㄣ | ㄤ | ㄥ |
| Key Code | 0xda | 0xdb | 0xdc | 0xdd | 0xde | 0xdf | 0xe0 | 0xe1 | 0xe2 | 0xe3 | 0xe4 | 0xe5 |
| Phonetic | ㄦ | ㄧ | ˊ | ˇ | ˋ | ˊ | ˋ | ˋ | ˋ | ˋ | ˋ | ˋ |
| Key Code | 0xe6 | 0x99 | 0x40 | 0x98 | 0x41 | 0x42 | | | | | | |

6.7.2.2 Software Setting

GBK Input

X 96 Y 153 W 218 H 213 Key value(0x) 0 voice ID 0

Name definition GBK Input

Automatic data upload

Button effect: -1 Specify No button effect

Page switching: -1 Specify NO page switching

Variable address(0x) 5020

Text length 100

Entry mode Re-enter

Display font 23

Dot matrix size 24

Text color 0000

Display font during entry 23

Dot matrix size 24

Text color 0000

Display method Pinyin hints are shown above

Cursor color black

Input status return

Input display area

44 194 668 299 Specify
Pinyin display position 143 269 set up

Display spacing 0

Keyboard settings Other pages

On the page 10

Keyboard area 29 155 - 690 588 Scan0_Area_Start The coordinates of the upper left corner of the called keyboard area: (Xs, Ys)
The coordinates of the upper left corner can be set consistent Scan0_Area_End The coordinates of the lower right corner of the called keyboard area: (Xs, Ys)

Display position 26 171 set up

Coordinates of the upper left corner of the touch area

Touch area width and height

Can be set by default

Voice ID number

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

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

Variable storage space users can use address range arbitrarily: 0x1000-0xFFFF. According to the input character length occupies the corresponding length address, the maximum is 127 words (Word). The VP_Len_Max+1 address is occupied as the 0xFFFF terminator; therefore, one more address needs to be reserved. For example, 5 addresses are actually needed, and because the 0xFFFF terminator occupies one more address, 6 addresses need to be reserved. The VP_1 address is used to save the input tag and effective data length, so the VP_1 address needs to be reserved. System variable interface address range: 0x0000-0x0FFF (if the page switching address is 0x0084).

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

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

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

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

Area_KB Upper left corner Area_KB lower right

Scan0_Area_Start, The coordinates of the upper left corner of the called keyboard area: (Xs, Ys)

guo

Beijing DWIN Technology Co., Ltd.

Pop up keyboard page

Scan0_Area_End, The coordinates of the lower right corner of the called keyboard area: (Xs, Ys)

Keycode

| Keycode | Description | | | | | |
|--------------------|-------------|---|---|---|---|---|
| 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| 8 | 9 | 0 | - | = | b | c |
| d | e | f | g | h | i | j |
| k | l | n | m | p | q | r |
| s | t | u | v | w | x | y |
| z |] | \ | , | . | / | |
| [|] | , | , | , | , | |
| Space | | | | | | |
| Functional Keycode | | | | | | |

Delete Select Delete All Def Keycode 0000 Add Description Save Select Cancel

Key value(0x) 0x E070 Full key definition Set

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

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

6.8 Synchrodata Return

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

6.8.1 Instruction Storage Format

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

The T5L_DGUS II screen has only 01 mode, which can replace the 02 mode of DGUS screen.

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

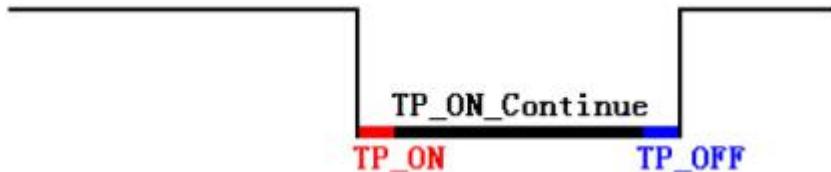
You can set VP1S and VP1T in DGUS.

VP1S can be set as the 22 file address, and VP1T refers to the VP address after the 83 instruction.

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

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

The three states of touch screen pressing are as below.



6.8.2 Software Setting

Synchrodata Return

| | | |
|--|-------|--|
| X 53 | Y 315 | Coordinates of the upper left corner of the touch area |
| W 179 | H 182 | Touch area width and height |
| <input type="checkbox"/> Key value(0x) 0 | | Can be set by default |
| <input type="checkbox"/> voice ID 3F | | Voice ID number |

Button effect:

| | | |
|---|---------|---|
| -1 | Specify | The display effect of pressing the touch area. For example, the current button is blue, and the button on the button effect page is dark blue. Press the button to see the sunken dark blue effect. |
| <input type="checkbox"/> No button effect | | |

Page switching:

| | | |
|--|---------|---|
| -1 | Specify | Page switching ID, can be set by default |
| <input type="checkbox"/> NO page switching | | <ul style="list-style-type: none"> 0x00: No data is returned; 0x01: Read the LEN1 byte length data pointed to by the *VP1S address to *VP1T. At the same time, the data is returned to the serial port. 0x02, 0x03 mode T5/T5L setting is invalid. |

First press

| | |
|--------------------------|---|
| Mode 0x01 | For example, the first time pressing/continuous pressing/releasing the pressing returns to 0×0001/0×0002/0×0003 different data settings; the configuration file needs to be enabled for automatic data upload. The first time the button is pressed, the 2-byte data of the 0×5610 address is given to the 0×5600 address; Finally, write 0×0001 in the starting address of the 22 file 0×ac20 (addressing 0×5610 double address, because the 22 file is in bytes and the variable address is in the unit of word length). In this way, the first press can return to 5A A5 06 83 00 01 00 01. |
| VP1S(0x) 5610 | |
| VP1T(0x) 5600 | |
| Data length 2 Unit: Byte | |

Keep pressing

| | |
|--------------------------|---|
| Mode 0x01 | Press the button continuously to give the 2-byte data of the 0×5620 address to the 0×5600 address; Finally, write 0×0002 in the starting address of the 22 file 0×ac40 (addressing 0×5620 double address, because the 22 file is in bytes, and the variable address is in the unit of word length). In this way, the first press can return to 5A A5 06 83 56 01 00 02. |
| VP2S(0x) 5620 | |
| VP2T(0x) 5600 | |
| Data length 2 Unit: Byte | |

Release the press

| | |
|--------------------------|--|
| Mode 0x01 | Release the button to give the 2-byte data of the 0×5630 address to the 0×5600 address; Finally, write 0×0003 in the starting address of the 22 file 0×ac60 (addressing 0×5630 double address, because the 22 file is in bytes, and the variable address is in the unit of word length). In this way, the first press can return to 5A A5 06 83 56 01 00 03. |
| VP3S(0x) 5630 | |
| VP3T(0x) 5600 | |
| Data length 2 Unit: Byte | |

6.9 Rotation Adjustment

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

6.9.1 Instruction Storage Format

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

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

6.9.2 Software Setting Instructions

Rotation adjustment

| | | |
|--|-------|--|
| X 155 | Y 147 | Coordinates of the upper left corner of the touch area |
| W 417 | H 414 | Touch area width and height |
| <input type="checkbox"/> Key value(0x) 0 | | Can be set by default |
| <input type="checkbox"/> voice ID 3F | | Voice ID number |

Name definition Rotation adjustment

Button effect:
 -1 Specify
 No button effect

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

Page switching:
 -1 Specify
 NO page switching

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

Variable address(0x) 543A

Adjust the data format 0x00 (integer)
 Center coordinates 361, 350
 set up

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

R0(area inner diameter) 100
 R1 (outer diameter of area) 200
 A0 (starting angle) 0
 Corresponding return value 0
 A1 (end angle) 720
 Corresponding return value 720

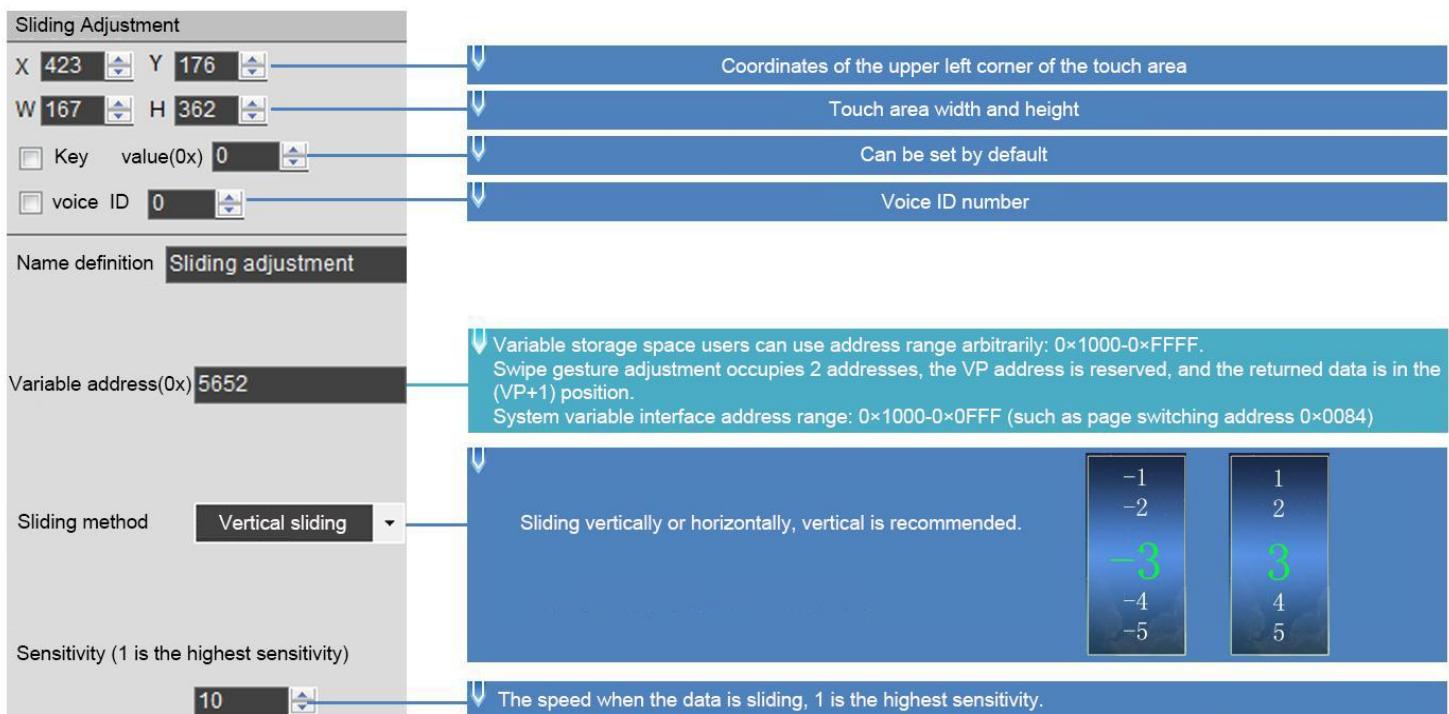
Click to set the coordinates of the center of the circle.
 The touch area can be drawn to be tangent to the circle.
 The rotating icon is made as shown in the figure. After the black background is filtered, only the white pointer area will be displayed.
 As shown:
 R0 inner diameter=453-353
 R1 outer diameter=553-353
 A0: The starting angle of the adjustment area, 0-719, the unit is 0.5
 A1: The end angle of the adjustment area, 1-720, the unit is 0.5

6.10 Sliding Adjustment

6.10.1 Instruction Storage Format

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

6.10.2 Software Setting



6.11 Page Sliding

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

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

6.11.1 Instruction Storage Format

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

6.11.2 Software Setting Instructions

Page Sliding

| | | |
|--|-------|--|
| X 0 | Y 0 | Coordinates of the upper left corner of the touch area |
| W 720 | H 720 | Touch area width and height |
| <input type="checkbox"/> Key value(0x) 0 | | Can be set by default |
| <input type="checkbox"/> voice ID 0 | | Voice ID number |

Name definition Page sliding

previous page:

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

next page:

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

Page turning sensitivity(time/distance)

Time
25

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

distance
50

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

End animation setup

End animation speed
36

illustrate:
0x00 ends without animation; other value unit pixels divided by DGUS period, the recommended value is the horizontal resolution divided by 20

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

6.12 Slide Icon Selection

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

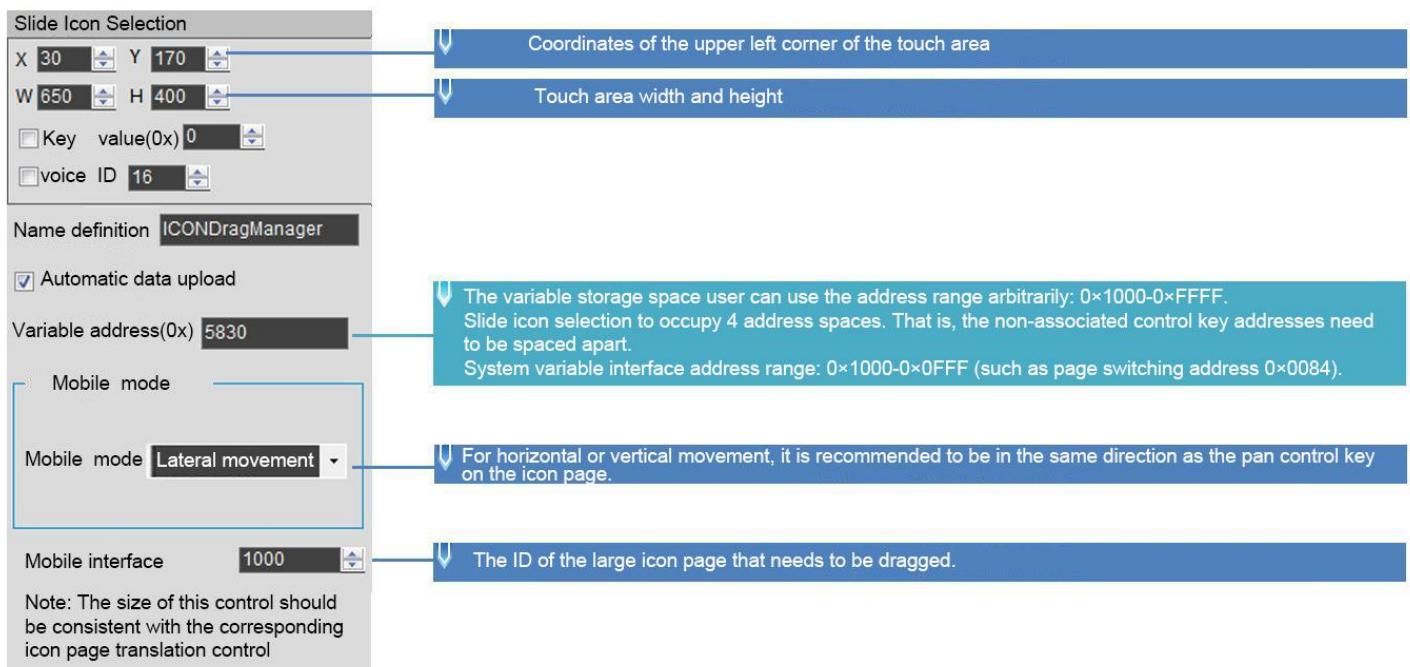
Steps:

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

6.12.1 Instruction Storage Format

| Address | Definition | Length(byte) | Description |
|---------|-----------------|--------------|---|
| 0x00 | Pic_ID | 2 | Page ID |
| 0x02 | TP_Area | 8 | Area of the Control: upper left corner (Xs, Ys), lower right corner (Xe, Ye), only for triggering. It must be consistent with icon display area of the 0x07 displayvariable. |
| 0x0A | Pic_Next | 2 | Undefined, write FFFF. |
| 0x0C | Pic_On | 2 | Undefined, write FFFF. |
| 0x0E | TP_Code | 2 | 0xFE0C, sliding icon selection key code. |
| 0x10 | 0xFE | 1 | 0xFE |
| 0x11 | VP | 2 | Variables pointer |
| 0x13 | Adj_Mode | 1 | 0x00 horizontal sliding; 0x01vertical sliding |
| 0x14 | TP_Page_ID_ICON | 2 | Icon ID 0x0000= undefined |
| 0x16 | reserved | 10 | Write 0x00 |

6.12.2 Software Setting



Chapter 7 Display Variable Configuration

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

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

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

The data variable display space is 128Kbytes.

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

14. BIN can be generated by DGUS and used in applications with touch variables, associated VP, modification of 22 file, SP, access via UART or DWIN OS to achieve complex functions.

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

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



7.1 List of Display Variables

| Number | Code | Variable name | Data Length (Character) | Description |
|--------|------|--------------------|-------------------------|--|
| 01 | 0x00 | Variable Icon | 1 | <p>The change range of a data variable linearly corresponds to a group of icon display; when the variable changes, the icon automatically switches accordingly. It is mostly used for detailed dashboard and progress bar display.</p> <p>Support background overlay and transparency settings. Background filter intensity can be set.</p> |
| 02 | 0x01 | Animation Icon | 2 | <p>A fixed value data variable corresponds to 3 different icon indication states: not display, display fixed icon, and display animation icon. It is mostly used for variable alarm prompts.</p> <p>The variable occupies 2 word positions, and the (VP+1) position is reserved; the icon ID cannot exceed 255 (0xFF).</p> <p>Support background overlay, transparency settings. And animation speed settings. Support single play mode. Background filter intensity can be set.</p> |
| 03 | 0x02 | Slider Display | 1 | <p>The change range of a data variable corresponds to the change of the display position of an icon (slider). It is mostly used for the indication of liquid level, dial and progress meter.</p> <p>Support background overlay and transparency settings. Background filter intensity can be set.</p> |
| 04 | 0x03 | Artistic Variables | 1/2/4 | <p>Use icon to replace font library to display variable data.</p> <p>Support background overlay and transparency settings. Background filter intensity can be set.</p> |
| 05 | 0x04 | Image Animation | Without | Play a group of full-screen pictures at the specified speed. It is mostly used for boot interface or screen saver. |
| 06 | 0x05 | Icon Rotation | 1 | <p>The range of change of a data variable linearly corresponds to the angle data, and then an icon is rotated according to the corresponding angle data.</p> <p>Background filtering intensity can be set.</p> <p>Mostly used for pointer dashboard display.</p> |
| 07 | 0x06 | Bit Icon | 3 | <p>The 0/1 state of each bit of a data variable corresponds to two of the 8 different display schemes, and a icon (or icon animation) is used for corresponding display.</p> <p>Background filter intensity can be set.</p> <p>Mostly used to display the switch status, such as the operation (animation) and stop (stationary icon) of the fan.</p> |
| 08 | 0X07 | Icon Page Tran | 4 | <p>The JPEG icon page that exceeds the screen resolution is displayed by panning up and down or left and right on the screen window.</p> <p>Icon sliding selection can be achieved by combining this control with 0x0C touch variables.</p> <p>Background filter intensity can be set.</p> |
| 09 | 0x08 | Icon Overlay | Max 120KB | <p>The JPEG icon of the variable buffer is superimposed and displayed in the specified area of the current page, and the display brightness and transparency can be set.</p> <p>VP=5AA5: turns on the display; VP+1=JPEG data buffer length; VP+2=JPEG data start.</p> <p>Background filter intensity can be set.</p> |
| 10 | 0x09 | Batch Va Icon | Max 64KB | According to the definition of the variable buffer, the icon is quickly copied from the background or video memory and displayed to the specified location. |

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

| | | | | |
|----|---------|------------------------------------|---------|---|
| 27 | 0x21_09 | Plot_Spectrum Display | | Display spectrum (vertical lines) according to variable data, with controllable line color and position. |
| 28 | 0x21_0D | Adjust the Area Display Brightness | | Adjust the display brightness of the designated display area to highlight or dilute the background display. |
| 29 | 0x21_0E | Graphic_Dual Color Bitmap Display | | Select the seed position and fill the enclosed solid color area with the specified color. |
| 33 | 0x24 | Area Scrolling | 1 | Circulate the contents of the specified area, and the moving direction can be set. It is used to simply realize dynamic running effects such as flowcharts and progress bars on the screen. The variable is occupied by the system and should not be used by the user. |
| 34 | 0x25 | QR Code | Max 259 | Display the QR code graphic on the screen according to the specified content. |
| 35 | 0x26 | Brightness | | Adjusts the display brightness of the specified display area to highlight or fade the background display. |
| 36 | 0x30 | Data Transmit | Max 11 | After the page is switched, the predefined data is transferred to the variable or serial port once. |
| 37 | 0x31 | Camera | 32KB | The current page is superimposed to display the camera monitoring screen. Only supported by T5L2 ASIC. |

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

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

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

7.2 Variables Icon

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

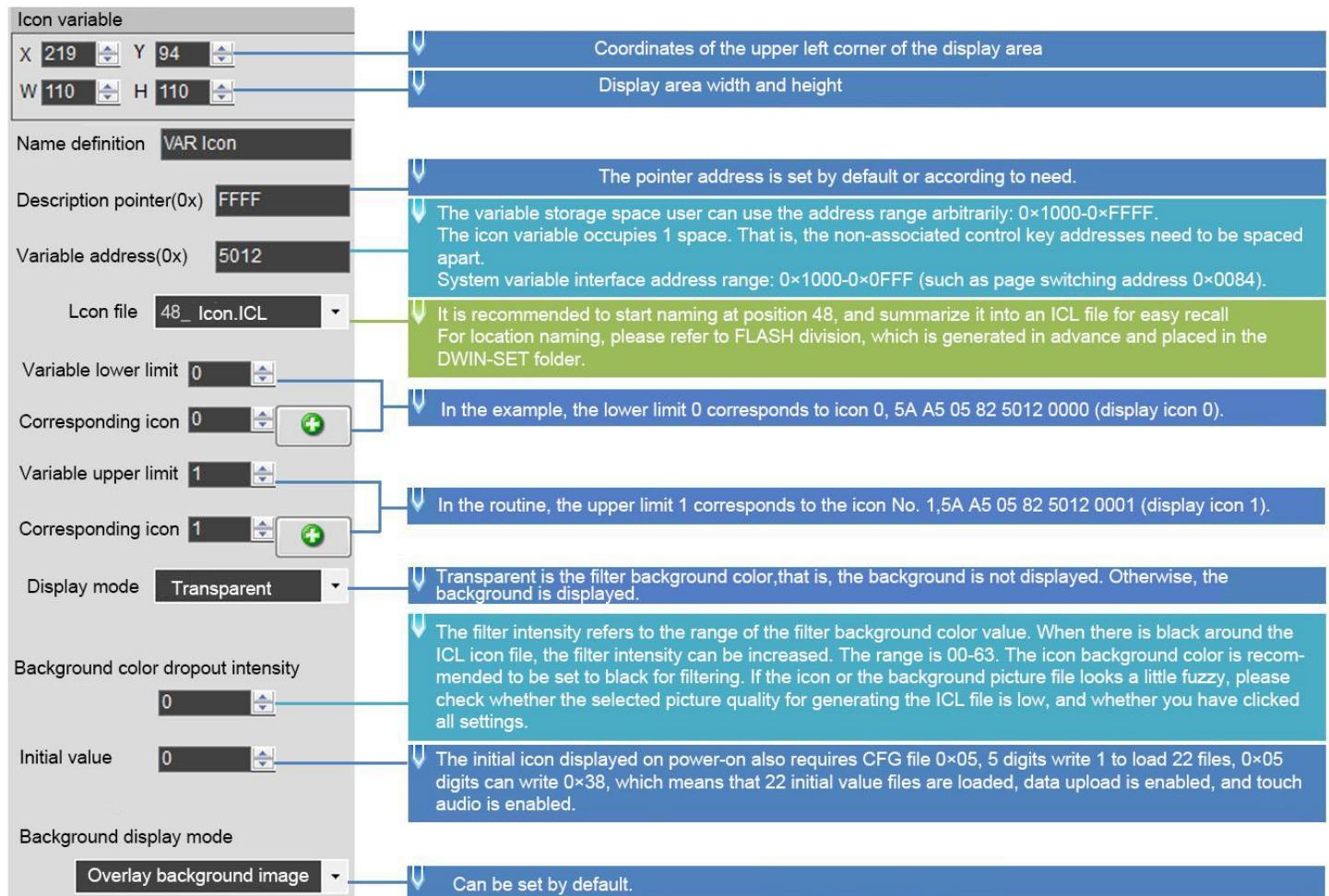
It is mostly used for fine dashboard and progress bar display.

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

7.2.1 Instruction Storage Format

| Address | SP offset | Definition | Length (bytes) | Description |
|---------|-----------|------------|----------------|--|
| 0x00 | | 0x5A00 | 2 | |
| 0x02 | | SP | 2 | Variable description pointer, 0xFFFF means loaded by the configuration file. |
| 0x04 | | 0x000A | 2 | |
| 0x06 | 0x00 | VP | 2 | Variable pointer, integer. |
| 0x08 | 0x01 | (x, y) | 4 | Upper-left coordinates of the icons to display |
| 0x0C | 0x03 | V_Min | 2 | Min value. Values smaller than "V_Min" will show noIcons. |
| 0x0E | 0x04 | V_Max | 2 | Max value. Values greater than "V_Max" will show noIcons. |
| 0x10 | 0x05 | Icon_Min | 2 | Icon associated to the V_Min. |
| 0x12 | 0x06 | Icon_Max | 2 | Icon associated to the V_Max. |
| 0x14 | 0x07:H | Icon_Lib | 1 | Index in the FLASH memory of the icon library to use. |
| 0x15 | 0x07:L | Mode | 1 | Display mode: 0x00= transparency mode Others= background display |
| 0x16 | 0x08:H | Layer_Mode | 1 | 0x00: overlay 0x01: overlay mode 1 0x02: overlay mode 2 |
| 0x17 | 0x08:L | ICON_Gamma | 1 | Icon brightness in overlay mode 2, range 0x00-0xFF, unit 1/256. |
| 0x18 | 0x09:H | PIC_Gamma | 1 | Background brightness in overlay mode 2, range 0x00-0xFF, unit 1/256. |
| 0x19 | 0x09:L | Filter_Set | 1 | The intensity of the background color filtering in transparency mode, range 0x01-0x3F. |

7.2.2 Software Setting



7.2.3 Example of Instruction

① Display the icon corresponding to the lower limit 0

5A A5 05 82 5012 0000

0x5AA5: Frame header;

0x05: Data length;

0x82: Write instruction;

0x5012: VP;

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

② Display the icon corresponding to the upper limit 1

5A A5 05 82 5012 0001

0x5AA5: Frame header;

0x05: Data length;

0x82: Write instruction;

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

③The upper limit and lower limit are not displayed

5A A5 05 82 5010 0002

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

④Describe pointer hiding and moving icons

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

7.3 Animation Icon

7.3.1 Instruction Storage Format

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

7.3.2 Software Setting

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

7.3.3 Example of Instruction

① Turn on animation, loop play mode

5A A5 05 82 5650 0001

5A A5 Frame header;

05 Data length;

82 Write instruction;

6550 VP;

0001 Write 1 to start animation value.

② Stop animation, loop play mode

5A A5 05 82 5650 0000

5A A5 Frame header;

05 Data length;

82 Write instruction;

6550 VP;

0000 write 0 to stop animation value.

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

5A A5 05 82 5650 0001

5A A5 Frame header;

05 Data length;

82 Write instruction;

6550 VP;

0001 Play once.

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

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

5A A5 05 82 5650 0000

5A A5 Frame header;

05 Data length;

82 Write instruction;

6550 VP;

0000 Play in reverse order.

⑤ Hide, move, change the animation icon frame number

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

For the use of the sp, please refer to 7.12 data variables.

7.4 Slider Display

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

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

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

7.4.1 Software Setting

Slider Display

| | | |
|--|---------|--|
| X [84] | Y [335] | Coordinates of the upper left corner of the display area |
| W [271] | H [47] | Display area width and height |
| Name definition Slider Display | | |
| Description pointer(0x) FFFF | | The pointer address is set by default or according to need. |
| Variable address(0x) 00A1 | | The variable storage space user can use the address range arbitrarily: 0x1000-0xFFFF. The icon variable occupies 1 space. That is, the non-associated control key addresses need to be spaced apart. System variable interface address range: 0x1000-0x0FFF (such as page switching address 0x0084). |
| Start variable scale value 0 | | |
| End variable scale value 255 | | The data size range cannot be set to 0 at the same time, otherwise it cannot slide. |
| Scale mode Horizontal scale bar | | Horizontal and vertical sliding mode |
| Icon file 48_Icon.ILC | | It is recommended to start naming at position 48, and summarize it into an ICL file for easy recall For location naming, please refer to FLASH division, which is generated in advance and placed in the DWIN-SET folder. |
| Swipe icon 17 | | Swipe to display the ICO icon |
| Display mode transparent | | Transparent is the filter background color. Show background |
| Icon display (Y) coordinate 503 value | | |
| (X) coordinate forward offset 0 (0-255) | | Set by default or according to needs. |
| Variable type Points to high byte data | | Here the system variable interface 0x00A1 address points to the high byte as drag adjustment to change the volume. |
| Initial value 0 | | |
| The start and end scale coordinates, the system automatically defaults to the area range of the configuration button | | |
| Background display mode | | Overlay background image |
| Overlay background (recommended), mode 1 effect, mode 2 effect.Ipsum | | |

7.4.2 Example of Instructions

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

Display data 10

5A A5 05 82 00A1 000A

5A A5 Frame header;

05 Data length;

82 Write instruction;

00A1 VP; 0

00A Data value.

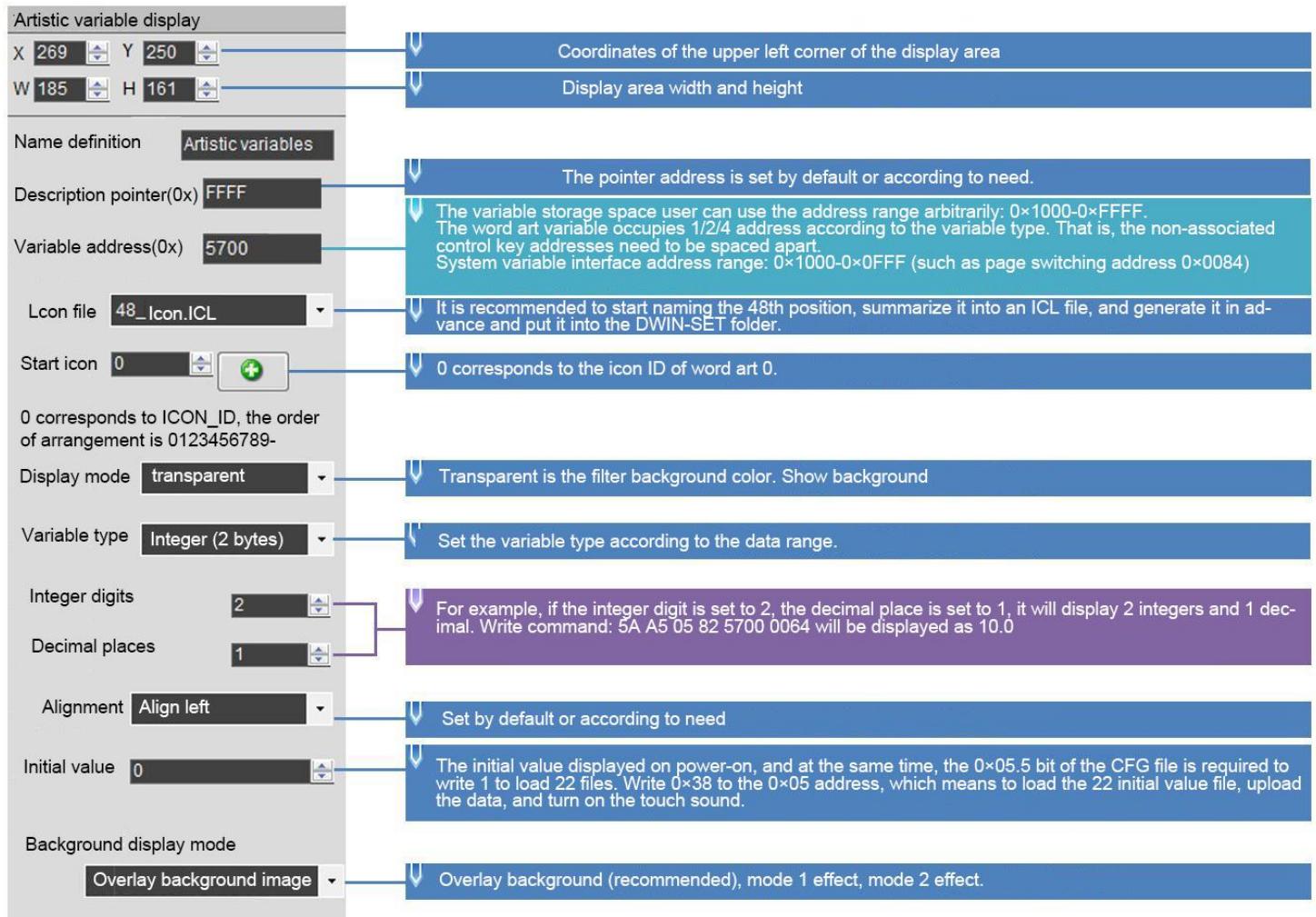
7.5 Artistic Variables

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

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

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

7.5.1 Software Setting



7.5.2 Examples of Instructions

5A A5 05 82 5700 0002

0x5AA5: Frame header;

0x05 Data length;

0x82: Write instruction;

0x 5700: VP;

0x 0002: Display data 2.

7.6 Image Animation

The image animation control is used to create an animation.

It can achieve screen jumps via serial communication.

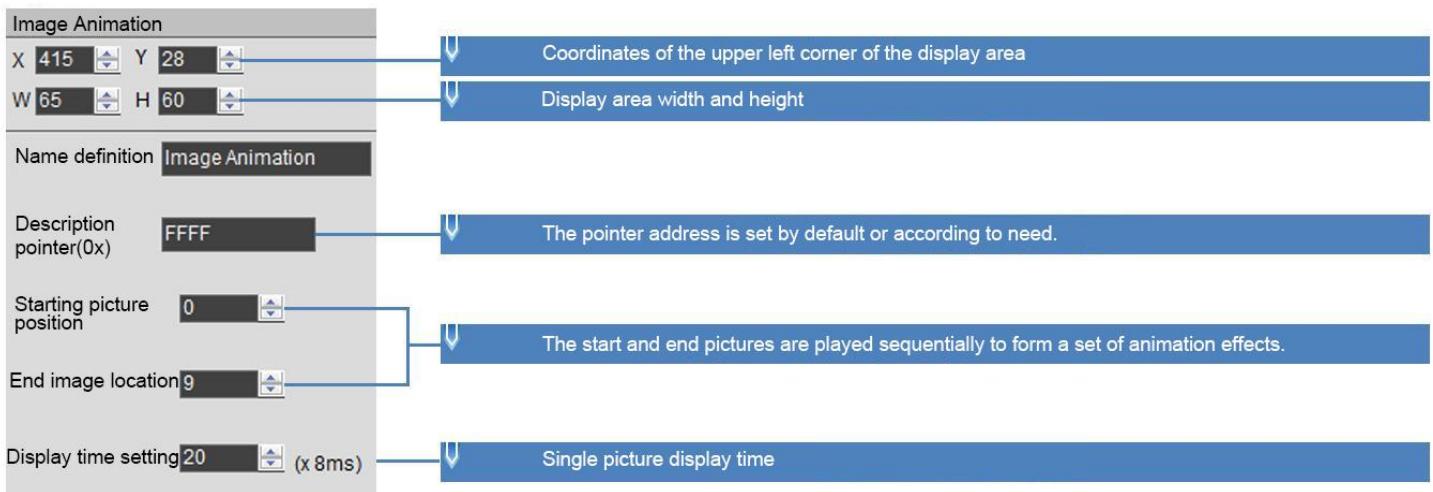
7.6.1 Instruction Storage Format

| Address | SP offset | Definition | Length (bytes) | Description |
|---------|------------|------------|----------------|---|
| 0x00 | 0x0000 | 0x5A04 | 2 | Fixed value 0x0000 |
| 0x01 | Pic_Begain | SP | 2 | Start icon position |
| 0x02 | Pic_End | 0x0005 | 2 | Terminate icon position |
| 0x03:H | Frame_Time | 0x0000 | 1 | The display time of one frame (one picture), the unit is 8ms. |

7.6.2 Software Setting

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

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



7.6.3 Example of Instructions

5A A5 07 82 0084 5A01 0000

0x5AA5: Frame header;

0x07: Data length;

0x82: Write instruction;

0x0084 System variable interface address, which is a fixed switching page address. Refer to the section 7.1 for more details.

0x 5A01: Fixed. The high byte 0x5A means that page processing is started once, and the CPU is cleared after processing; the low byte 0x01 means that the page is switched, and the picture specified in the picture storage area is displayed to the current background page; see the system variable interface list for details.

0x 0000: Page ID, the IF of the page to be switched to.

7.7 Icon Rotation

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

It is mostly used for pointer instrument panel display.

7.7.1 Instruction Storage Format

The background filter intensity can be set.

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

7.7.2 Software Setting

Icon rotation indication

X 240 Y 207
W 27 H 35

Coordinates of the upper left corner of the display area
Display area width and height

Name definition Icon Rotation

Description pointer(0x) FFFF

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

Variable address(0x) 5438

The variable storage space user can use the address range arbitrarily: 0×1000-0×FFFF.
Icon rotation occupies 1 address space. That is, the non-associated control key addresses need to be spaced apart.
System variable interface address range: 0×1000-0×0FFF (such as page switching address 0×0084)

Icon file 48_Icon.ICL

It is recommended to start at position 48 and summarize it into an ICL file. Generate it in advance and put it into the DWIN_SET folder.

Icon ID 70

Pointer icon ID.

Icon rotation center:

X 10 Y 117

Icon rotation center coordinates, which can be specified by loading icons.

The screen rotation center, the system automatically defaults to the upper left coordinate of the configuration button

Variable value of initial rotation angle 0

The start and end values of the data range are displayed.

End rotation angle variable value 200

Starting angle of rotation 495

End rotation angle 225

Rotation angle, 0-720 (0×000-0×200), unit 0.5

Display mode transparent

VP_Mode Points to an integer variable

Initial value 0

495 degrees, 225 degrees calculation:
360 degrees is divided into 4 single sub-areas of 90 degrees,
90 degrees is divided into 4 individual sub-areas of 22.5,
Starting angle of rotation: $720 - (90 + 22.5) \times 2 = 495$ degrees.
End rotation angle: $(90 + 22.5) \times 2 = 225$ degrees

Examples of Instruction:

5A A5 05 82 5438 0000

0x5AA5: Frame header;

0x05: Data length;

0x82: Write instruction;

0x5438 VP;

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

5A A5 05 82 5438 0064

0x5AA5: Frame header;

0x05: Data length;

0x82: Write instruction;

0x5438 VP;

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

5A A5 05 82 5438 00C8

0x5AA5: Frame header;

0x05: Data length;

0x82: Write instruction;

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

7.8 Bit Icon

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

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

7.8.1 Instruction Storage Format

| Address | SP offset | Definition | Length (bytes) | Description | | | | | | | | | | | | | | | | | | | | | | | | | |
|--|--------------------------|--------------------------|----------------|--|------|-----------|--|---|---|------|--------|--------|------|--------|------|------|--------|--------------------------|------|------|--------|------|------|--------------------------|------|--------------------------|--------|------|--------------------------|
| 0x00 | | 0x5A06 | 2 | Bit variable pointer, word variable | | | | | | | | | | | | | | | | | | | | | | | | | |
| 0x01 | | SP | 2 | Auxiliary variable pointer, double word, cannot be accessed by user software | | | | | | | | | | | | | | | | | | | | | | | | | |
| 0x02 | | 0x000C | 2 | The bit position whose value is 1 indicates that the corresponding position of *VP needs to be displayed. | | | | | | | | | | | | | | | | | | | | | | | | | |
| 0x03: H | 0x03:H | Display_Mode | 1 | Define the display mode: | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | | | <table border="1"> <thead> <tr> <th rowspan="2">Mode</th> <th colspan="2">Bit Value</th> </tr> <tr> <th>0</th> <th>1</th> </tr> </thead> <tbody> <tr> <td>0x00</td> <td>ICON0S</td> <td>ICON1S</td> </tr> <tr> <td>0x01</td> <td>ICON0S</td> <td>None</td> </tr> <tr> <td>0x02</td> <td>ICON0S</td> <td>Animation: ICON1S-ICON1E</td> </tr> <tr> <td>0x03</td> <td>None</td> <td>ICON1S</td> </tr> <tr> <td>0x04</td> <td>None</td> <td>Animation: ICON1S-ICON1E</td> </tr> <tr> <td>0x05</td> <td>Animation: ICON0S-ICON0E</td> <td>ICON1S</td> </tr> <tr> <td>0x06</td> <td>Animation: ICON0S-ICON0E</td> <td>None</td> </tr> <tr> <td>0x07</td> <td>Animation: ICON0S-ICON0E</td> <td>Animation: ICON1S-ICON1E</td> </tr> </tbody> </table> | Mode | Bit Value | | 0 | 1 | 0x00 | ICON0S | ICON1S | 0x01 | ICON0S | None | 0x02 | ICON0S | Animation: ICON1S-ICON1E | 0x03 | None | ICON1S | 0x04 | None | Animation: ICON1S-ICON1E | 0x05 | Animation: ICON0S-ICON0E | ICON1S | 0x06 | Animation: ICON0S-ICON0E |
| Mode | Bit Value | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | 0 | 1 | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 0x00 | ICON0S | ICON1S | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 0x01 | ICON0S | None | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 0x02 | ICON0S | Animation: ICON1S-ICON1E | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 0x03 | None | ICON1S | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 0x04 | None | Animation: ICON1S-ICON1E | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 0x05 | Animation: ICON0S-ICON0E | ICON1S | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 0x06 | Animation: ICON0S-ICON0E | None | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 0x07 | Animation: ICON0S-ICON0E | Animation: ICON1S-ICON1E | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Bitmap icon arrangement: 0x00=x++, the non-display bit specified by Act_Bit_Set is not reserved; 0x01=Y++, the non-display bit specified by Act_Bit_Set does not retain the position | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 0x02=x++, the non-display bit specified by Act_Bit_Set is reserved for DIS_MOV position 0x03=Y++, the non-display bit specified by Act_Bit_Set is reserved for DIS_MOV position | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ICON display mode: 0x00: Transparent background. 0x01: opaque | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Index in the FLASH memory of the icon library to use. | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Does not show animation mode, bit_0 icon ID Display animation mode, bit_0 icon animation start ID position | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Display animation mode, bit_0 icon animation end ID position | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Does not show animation mode, bit_1 icon ID Display animation mode, bit_1 icon animation start ID position | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Display animation mode, bit_1 icon animation start ID position | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 0x09 | 0x09 | (x, y) | 4 | The starting position variable display position, the coordinate position of the upper left corner of the icon. | | | | | | | | | | | | | | | | | | | | | | | | | |
| 0x0B | 0x0B | DIS_MOV | 2 | Next icon coordinate move coordinate interval. | | | | | | | | | | | | | | | | | | | | | | | | | |
| 0x0C | 0x0C | Reserved | 2 | Write 0x00 | | | | | | | | | | | | | | | | | | | | | | | | | |

7.8.2 Software Setting

Bit icon

| | | |
|---|-------|---|
| X 41 | Y 183 | Coordinates of the upper left corner of the display area |
| W 50 | H 33 | Display area width and height |
| Name definition Bit icon | | |
| Description pointer(0x) FFFF | | The pointer address is set by default or according to need. |
| Variable address(0x) 6000 | | Variable storage space users can use address range arbitrarily: 0x1000-0xFFFF. The bit variable occupies a maximum of 3 space addresses. That is, the non-associated control key addresses need to be spaced apart. System variable interface address range: 0x0000-0x0FFF (such as page switching address 0x0084). |
| Secondary address(0x) 8020 | | If the animation mode is used, the auxiliary address must be separated by 2 addresses, and the address cannot be repeated. You don't need to set the animation if you don't use it. |
| Bit Icon ON (1) / OFF (0): 00000000000000001 | | The 0th bit is enabled here, a total of 16 bits. |
| Set up | | 0x01-0x07, define the wind table Display_Mode. In 0x03 mode, writing 0 to the corresponding bit does not display the icon, writing 1 to display the icon. |
| Display mode 0x03 | | 0x00-0x03, define the wind table Move_Mode. |
| Mobile mode 0x00 | | 0x00-0x03, define the wind table Move_Mode. |
| Moving interval 0 | | The pixel interval to move the next icon |
| Icon file 48_Icon.ILC | | It is recommended to start at position 48 and summarize them into an ICL file. Generate it in advance and put it into the DWIN_SET folder. |
| ICON0S 0 | | The meaning of icon mode and non-display mode settings: ICON1S: Write 1 to the corresponding bit to display the icon ID, or not display it. ICON1E: No need to set at this time |
| ICON0E 0 | | The meaning of the animation mode setting: ICON1S; Write 1 icon animation start ID in the corresponding bit. ICON1E; Write 0 icon animation end ID |
| ICON1S 67 | | The meaning of icon mode and non-display mode settings: ICON1S: Write 1 to the corresponding bit to display the icon ID, or not display it. ICON1E: No need to set at this time |
| ICON1E 0 | | The meaning of the animation mode setting: ICON1S; Write 1 icon animation start ID in the corresponding bit. ICON1E; Write 1 icon animation end ID |
| Display mode transparent | | |
| Initial value 0 | | |

7.8.3 Example of Instructions

①16 bits fully open

5A A5 05 82 6000 FFFF

0x5A 0xA5: Frame header;

0x 05: Data length;

0x 82: Write instruction;

0x 6000: VP;

0x FFFF: 1111 1111 1111 1111 Write 1 to all 16 bits, which is FFFF.

②16 bits all closed

5A A5 05 826000 0000

0x5A 0xA5: Frame header;

0x 05: Data length;

0x 82: Write instruction;

0x 6000: VP;

0x0000: All 16 bits are written with 0.

③Turn on bit 0 and bit 3

5A A5 05 82 6000 0009

0x5A 0xA5: Frame header;

0x 05: Data length;

0x 82: Write instruction;

0x 6000: VP;

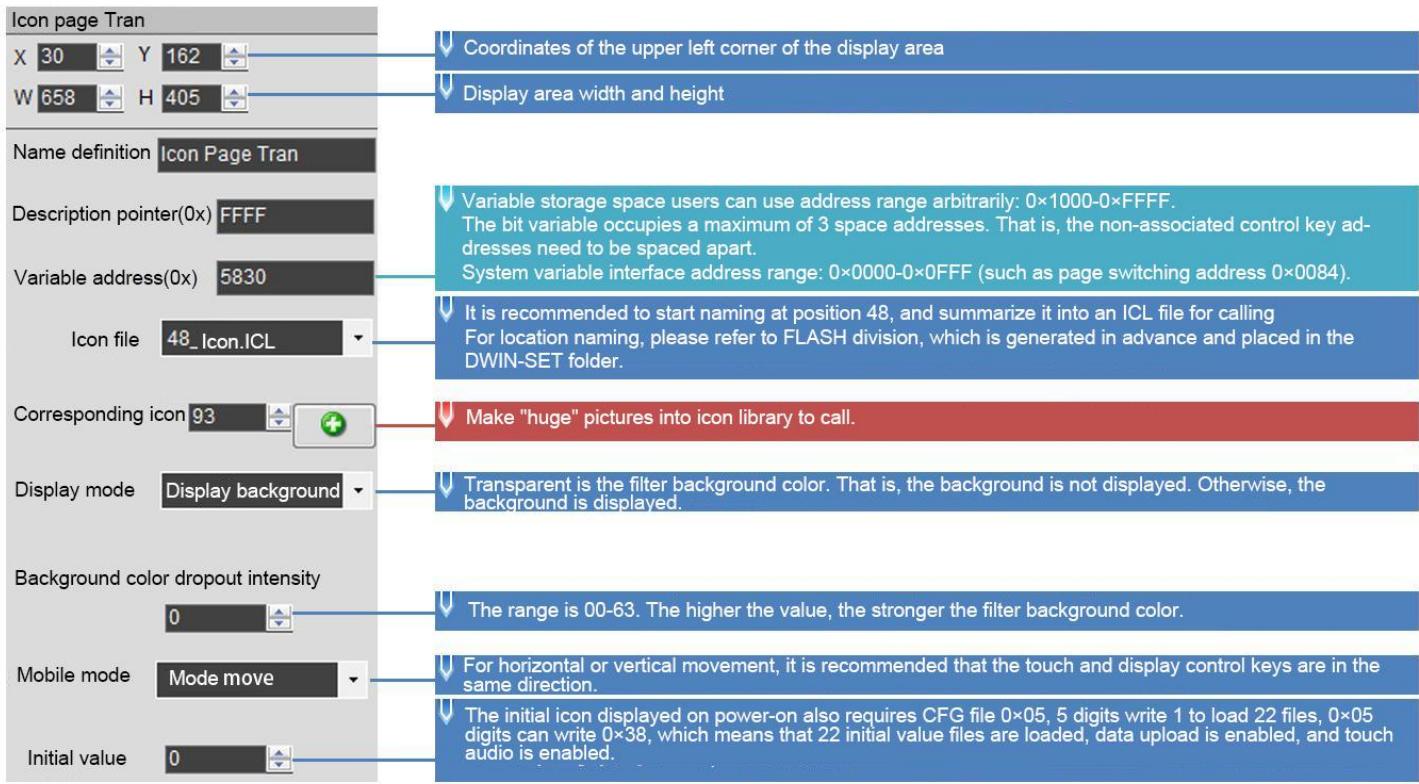
0x 0009: Write 1 to bit 0 and bit 3, 0000 0000 0000 1001 is 0x0009.

7.9 Icon Page Tran

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

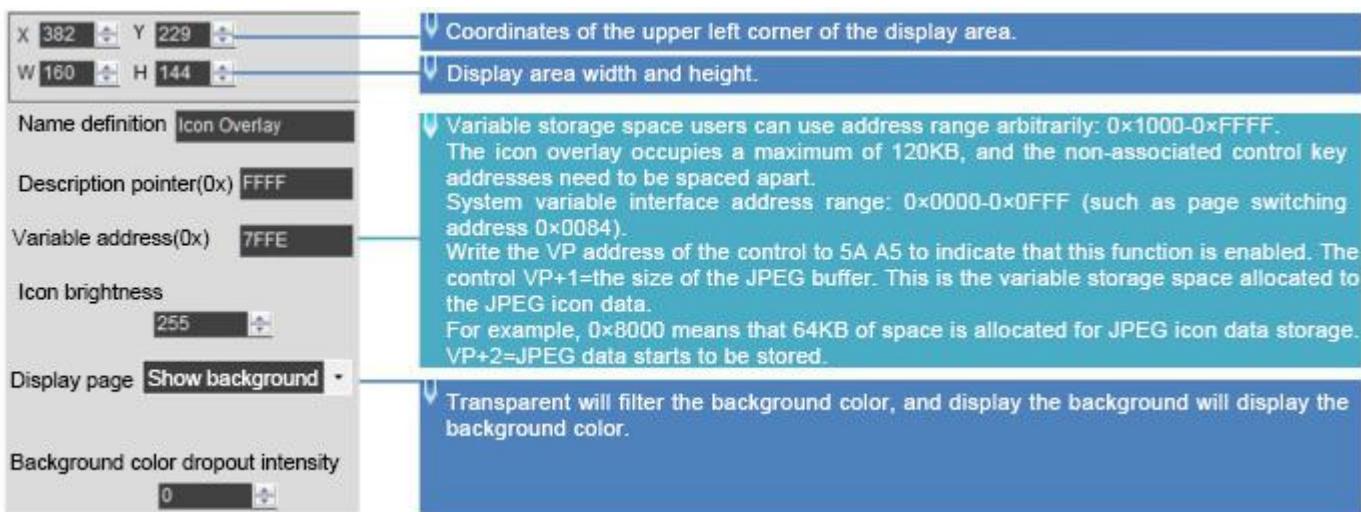
The background filter strength can be set.

7.9.1 Software Setting



7.10 Icon Overlay

7.10.1 Software Setting



(1) Function Overview

Overlay the JPEG icon of the variable buffer to the specified area of the current page. Display brightness and transparency can be set. The background filter strength control can be set.

Write 5A A5 to the VP address of the control (must be an even number) to enable this function.

Control VP+1=size of JPEG buffer, this is the size of variable storage space allocated to JPEG icon data, for example 0x8000 means allocation.

64KB of space for JPEG icon data storage.

VP+2=JPEG data starts to be stored.

(2) Functional Application

For application scenarios that need to dynamically display pictures, such as the function of staff photo replacement, pop-up prompt boxes, etc.

(3) Functional Operation Instructions

Planning JPEG image data needs to be cached to the address on the screen. General applications such as small-sized avatar photos of users take up very little, and the space from 0x8000-0xFFFF is used to store JPEG image (\leq 64KB) data.

Step 1: Set the control, for example, set the address to 0x7FFE

Step 2: Picture data (the data stream of JPEG pictures: data at the beginning of FFD8, and at the end of FFD9) are sent sequentially from the starting address of 0x8000.

7.10.2 Example of Instructions

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

```
5A A5 F3 82 80 00 FF D8 FF E0 00 10 4A 46 49 96 00 01 01 01 00 60 00 60 00 00 FF DB 00 43 00 08 06 06 07 06  
05  
08 07 07 07 09 09 08 0A 0C 14 0D 0C 0B 0B 0C 19 12 13 0F 14 1D 1A 1F 1E 1D 1A 1C 1C 20 24 2E 27 20 22 2C  
23 1C  
1C 28 37 29 2C 30 31 34 34 34 1F 27 39 3D 38 32 3C 2E 33 34 34 32 FF DB 00 43 01 09 09 09 0C 0B 0C 18 0D 0D 18  
32 21  
1C 21 32 32 32 32 .....  
(The DGUSII screen will have a response "5A A5 03 82 4F 4B", and the next one can be sent immediately after receiving the response. In fact, the T5L uses a 20ms refresh rate mechanism. In the case of using a baud rate of ≤ 115200, sending data continuously will not cause serial port busy processing)
```

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

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

Send several packets of data in turn...

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

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

```
1.jpg  
0 1 2 3 4 5 6 7 8 9 a b c d e f  
00000000h: FF DB FF E0 00 10 4A 46 49 96 00 01 01 01 00 60 ; ??JFIF.....`  
00000010h: 00 60 00 00 FF DB 00 43 00 08 06 06 07 06 05 08 ; .`... ?C.....  
00000020h: 07 07 07 09 09 08 0A 0C 14 0D 0C 0B 0B 0C 19 12 ; .....  
00000030h: 13 0F 14 1D 1A 1F 1E 1D 1A 1C 1C 20 24 2E 27 20 ; ..... $.'  
00000040h: 22 2C 23 1C 1C 28 37 29 2C 30 31 34 34 34 1F 27 ; ",#..(7),01444.'  
00000050h: 39 3D 38 32 3C 2E 33 34 32 FF DB 00 43 01 09 09 ; 9=82<.342 ?C...  
00000060h: 09 0C 0B 0C 18 0D 0D 18 32 21 1C 21 32 32 32 32 ; .....!2222  
00000070h: 32 32 32 32 32 32 32 32 32 32 32 32 32 32 32 32 ; 2222222222222222  
00000080h: 32 32 32 32 32 32 32 32 32 32 32 32 32 32 32 32 ; 2222222222222222  
00000090h: 32 32 32 32 32 32 32 32 32 32 32 32 32 32 32 32 FF C0 ; 22222222222222 ?  
000000a0h: 00 11 08 00 AA 01 1B 03 01 22 00 02 11 01 03 11 ; ....?...".  
.....
```

Step 3: Send the display instruction.

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

In fact, "0x80 00" refers to the length of the buffer, JPEG data stream (0xFFD8 ... 0xFFD9) file format, the bottom layer will judge and process automatically. User does not need to worry about the redundant data in the cache address later.



7.11 Batch Va Icon

7.11.1 Instruction Storage Format

| Address | SP offset | Length (bytes) | Description |
|---------|-------------|----------------|--|
| 0x00 | | 2 | |
| 0x02 | | 2 | Parameter pointer. 0xFFFF: Disables SP (no run-time modification). |
| 0x06 | 0x00 | 4 | Variable pointer, must be even. D3: 0x5A = enable the display, others = disable the display. D2: displayed ICON ID, range 0-N. D1: display mode, only works for the display of background copy. 7 filter switch 0=open filter,1=display background. 6 reserved,0. 5..0 filter set value 0x01-0x3F. D0: brightness of ICON, only works for the display of background copy. If the brightness of ICON is not set to 0xFF, the icon will display together with background, and the display speed will be slower about 30%. |
| 0x08 | 0x01 | 4 | The position of the icon to display, must be even. D3:D2: X coordinate of the upper-left ICON corner. D1:D0: Y coordinate of the upper-left ICON corner. The data is described through the number sequence of the iconID(0-N). |
| 0x0A | VP2 | 8 | Parameter pointer of the source ICON, must be even. Every icon occupy 4 bytes. D7:D6: X coordinate of the upper-left ICON corner. D5:D4: Y coordinate of the upper-left ICON corner. D3:D2: Width pixels of the icon. D1:D0: Height pixels of the icon. The data is described through the number sequence of the iconID(0-N). |
| 0x0C | Data_Num | 2 | Number of data items corresponding to *VP, 0x0000-0x1000, up to 4096 data items |
| 0x0E | Icon_Source | 1 | 0x00: current page. Others: icon background image. |
| 0x0F | Icon_Lib | 1 | Index in the FLASH memory of the icon library to use. Only works in icon background image mode. |
| 0x10 | Icon_ID | 2 | ID of the icon. Only works in icon background image mode. |
| 0x12 | Reserved | 14 | 0x00 |

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

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

Example:

Assuming that each data has 8 different icons to choose from. A single icon is 64*64 pixels, spliced into a 560*64 background image, and there are 100 data items to be displayed. Then the processing time is $105 * 560 * 64 / 400 + 100 * 64 * 64 / 200 = 11.456 \text{ mS}$, In the video memory mode, it is 5.12 mS.

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

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

Example:

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

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

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

7.12 Data Variables

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

7.12.1 Instruction Storage Format

and the instruction storage format is shown in the table.

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

7.11.2 Example of Instructions

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

(1) Display integer data 100.

5AA5058254200064

0x5AA5A5: Frame header;

0x05: Data length, which is the 5 bytes of 8254 2000 64;

0x82: Write instruction;

0x5420: VP;

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

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

5A A5 09 82 53000064 0063 0062

0x5AA5A5: Frame header;

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

0x82: Write instruction;

0x5300: VP;

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

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

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

(3) Display long integer data 100.

5A A5 07 82 54220000 0064

0xA5: Frame header;

0x07: Data length;

0x82: Write instruction;

0x5422: VP;

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

(4) Display ultra-long integer data 100

5A A5 0B 82 54240000 0000 0000 0064

0x5AA5: Frame header;

0x0B: Data length;

0x82: Write instruction;

0x5424: VP;

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

(5) Display single-precision floating-point number data 99

5A A5 07 82 542842C6 0000

0x5AA5: Frame header;

0x07: Data length;

0x82: Write instruction;

0x5428: VP;

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

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

5A A5 0B 82 542B 4058C000 0000 0000

0x5AA5: Frame header;

0x0B: Data length;

0x82: Write instruction;

0x542B: VP;

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

(7) Display integer type data -100 5A A5 05 82 5300FF9C

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

0x5300: VP;

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

(8) Display 2 decimal places

Sending 5A A5 05 82 5300 0064 will display 1.00.

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

(9) Change the data color

5A A5 05 82 9003F800

0x5AA5: Frame header;

0x05: Data length;

0x82: Write instruction;

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

0xF800: Color code.

(10) Change the data coordinates

5A A5 07 82 900 10064 0064

0x5AA5: Frame header;

0x07: Data length;

0x82: Write instruction;

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

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

(11) Change the font size

5A A5 05 82 90040028

0x5AA5: Frame header;

0x05: Data length;

0x82: Write instruction;

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

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

(12) Change the integer digits to 3 digits

5A A5 05 82 90050003

0x5AA5: Frame header;

0x05: Data length;

0x82: Write instruction;

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

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

(13) Change the number of decimal places to 2

5A A5 05 82 90060200

0x5AA5: Frame header;

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

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

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

(14) No display data 0, hidden data 0 after power-on

Set the variable address to 0x5300 and the description pointer address to 0x9000. Method one: hide

Hidden data variable display: 5A A5 05 82 9000FF00

0x5AA5: Frame header; 0x05:

Data length;

0x82: Write instruction;

0x9000 SP;

0xFF00: Hide the data. Display data after hiding: 5A A5 05 82 90005300

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

0x9000: Represents the SP;

0x5300: is the VP.

Change the data value again: 5A A5 05 82 5300 0062 0x5AA5:

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

0x5300 VP;

0x0062 Display data 98.

Method 2: Use the description pointer to change the coordinate position

5A A5 07 82 900102D0 02D0

0x5AA5: Frame header;

0x07: Data length;

0x82: Write instruction;

0x9001: The description pointer is offset by 1 bit based on 0x9000;

0x02D0: (720,720) coordinate point.

With the help of cross-border is not displayed. Such as 720*720 resolution screen, change the upper left display coordinate to (720*720) cross-border hidden, the normal display will move back to the original coordinate.

7.13 Text Display

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

This control is usually used with the text input control.

7.13.1 Instruction Storage Format

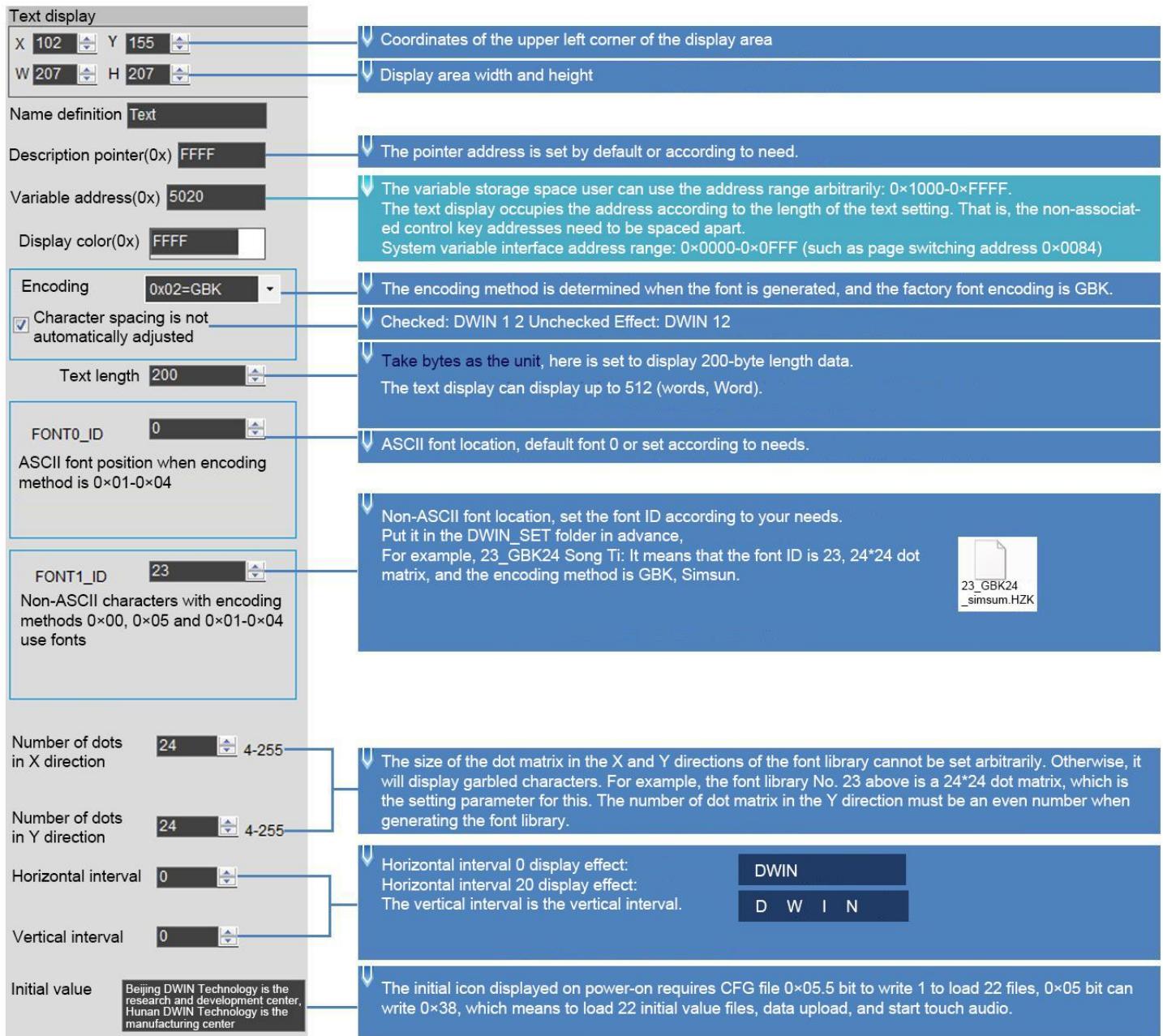
| Address | Definition | Length (bytes) | Description |
|---------|------------------|----------------|--|
| 0x00 | *VP | 2 | Text pointer. |
| 0x01 | (X, Y) | 4 | The starting display position, which displays the coordinates of the upper left corner of the string. |
| 0x03 | Color | 2 | Display text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

7.13.2 Software Setting

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

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

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



7.13.3 Example of Instructions

① Display "North 12AB"

5AA5 0B82 5020 B1B1 3132 4142 FFFF

0x5AA5: Frame header; 0x0B: Data length;

0x82: Write instruction;

0x5020: VP;

0xB1B1: ASCII code of "North";

0x3132: ASCII code of "12";

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

② New line display.

5A A5 05 82 5023 0D0A

0x5AA5: Frame header;

0x05: Data length;

0x82: Write instruction;

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

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

③ Clear the text display.

5A A5 07 82 5020 2020

0x5AA5: Frame header;

0x07: Data length;

0x82: Write instruction;

0x5020: VP;

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

④ Change the text color

5A A5 05 82 8803F800

0x5AA5: Frame header;

0x05: Data length;

0x82: Write instruction;

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

0xF800: Color value.

⑤ Hide text

Method 1: Hide

5A A5 05 82 8800FF00

0x5AA5: Frame header;

0x05: Data length;

0x82: Write instruction;

0x8800: SP;

0xFF00: Hidden text.

Display after hiding:

5A A5 05 82 88008070

0x5AA5: Frame header;

0x05: Data length;

0x82: Write instruction;

0x8800: SP;

0x8070: VP.

Method 2: Modify the coordinates

5A A5 07 82 8801 02D0 02D0

0x5AA5: Frame header;

0x05: Data length;

0x82: Write instruction;

0x8801: SP;

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

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

Display after hiding:

5A A5 07 82 8801 0064 0064

0x5AA5: Frame header;

0x05: Data length;

0x82: Write instruction;

0x8801: SP;

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

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

Method 3: Change the display length to 0000

5A A5 05 82 8808 0000

0x5AA5: Frame header;

0x05: Data length;

0x82: Write instruction;

0x8808: SP;

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

0x0000 displays 0 bytes of data.

Display after hiding:

5A A5 05 82 8808 0064

0x5AA5: Frame header;

0x05: Data length;

0x82: Write instruction;

0x8808: SP;

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

0x0064 displays 100 bytes of data.

7.14 RTC Display

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

7.14.1 Instruction Storage Format

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

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

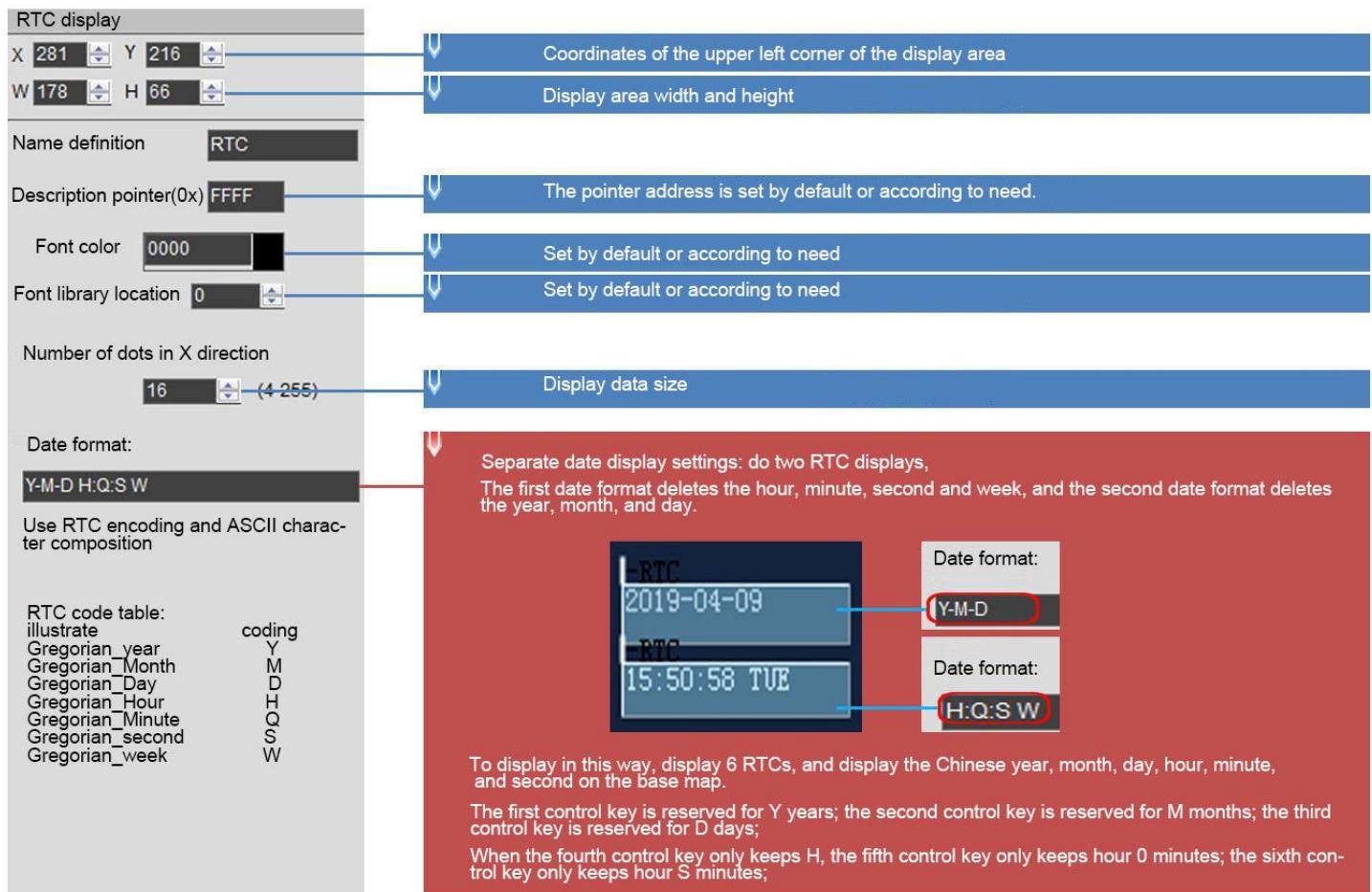
(2) RTC code.

| Explanation | Code | Display Format |
|------------------|------|-----------------------------|
| Gregorian_Year | Y | 2000-2099 |
| Gregorian_Month | M | 01-12 |
| Gregorian_Day | D | 01-31 |
| Gregorian_Hour | H | 00-23 |
| Gregorian_Minute | Q | 00-59 |
| Gregorian_Second | S | 00-59 |
| Gregorian_Week | W | SUN MON TUE WED THU FRI SAT |
| End of encode | 0x00 | |

7.14.2 Software Setting

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

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



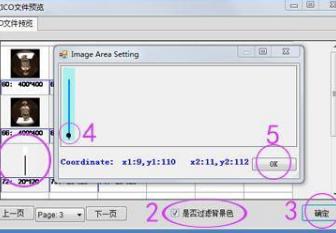
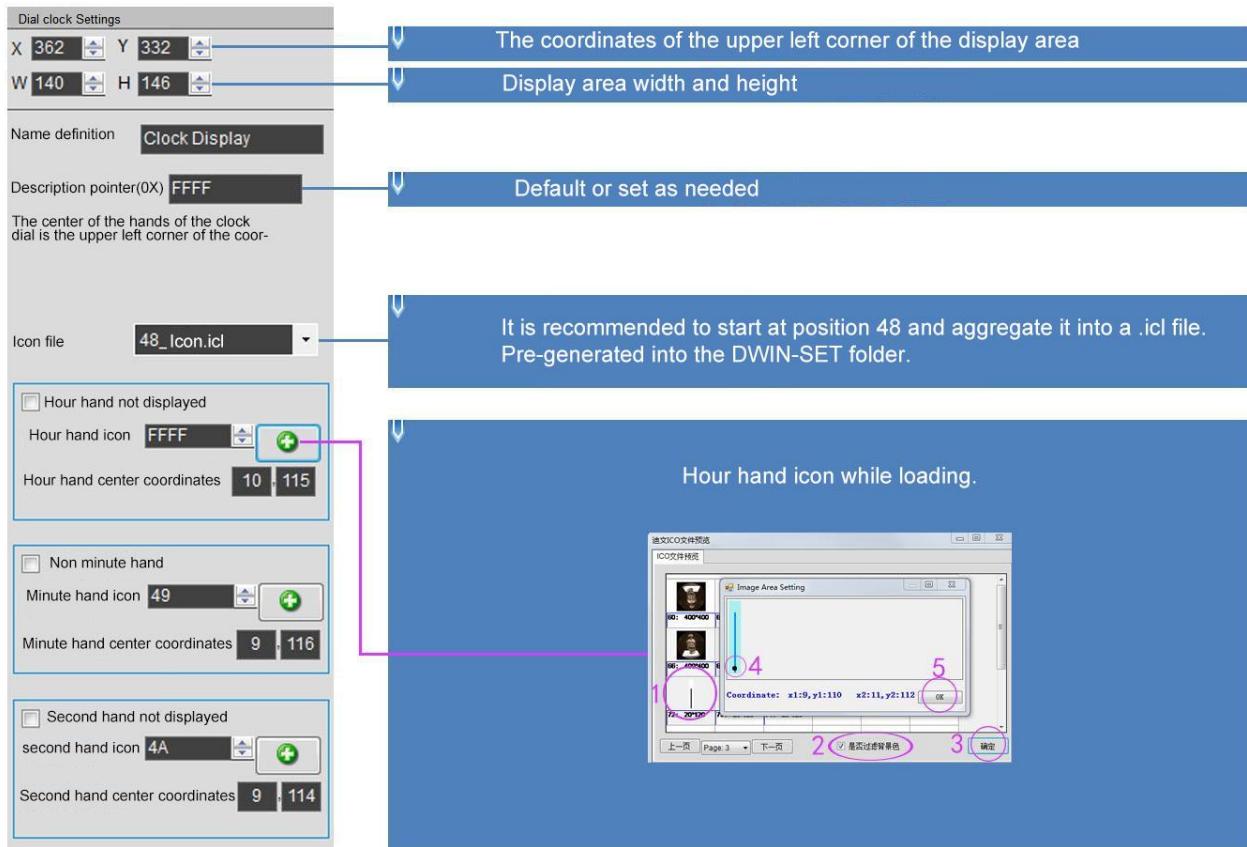
7.14.3 RTC Display(watch)

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

7.14.3.1 Instruction Storage Format

| SP offset | Definition | Length (bytes) | Description |
|-----------|---------------------|----------------|--|
| 0x00 | 0x0001 | 2 | Write 0x0001. |
| 0x01 | (X, Y) | 4 | The center of a clock dial. |
| 0x03 | Icon_Hour | 2 | ID of pointer ICON, 0xFFFF indicates that the hour hand is not displayed. |
| 0x04 | Icon_Hour_Central | 4 | Position of the center of rotation of the clockwise icon. |
| 0x06 | Icon_minute | 2 | ID of the minute hand ICON. 0xFFFF indicates that the minute hand is not displayed. |
| 0x07 | Icon_Minute_Central | 4 | Rotation center position of minute hand ICON. |
| 0x09 | Icon_Second | 2 | The ID of the second hand icon; 0xFFFF indicates that the second hand is not displayed. |
| 0x0A | Icon_Second_Central | 4 | Position of rotation center of second hand ICON. |
| 0x0C:H | Icon_Lib | 1 | ID of the icon library file where the pointer icon resides |
| 0x0C:L | Undefined | 1 | Write 0x00 |

7.14.3.2 Software Setting



7.14.3.3 Example of Instruction

Display time October 1, 2050 11:12:13

5A A5 0B 82 009C 5A A5 32 0A 01 0B 0C 0D

0x5AA5: Frame header;

0x0B Data length;

0x82: Write instruction;

0x009C: VP; this address is fixed and cannot be customized. For details, please refer to the section 7.1 0x9C address definition;

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

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

7.15 HEX Data

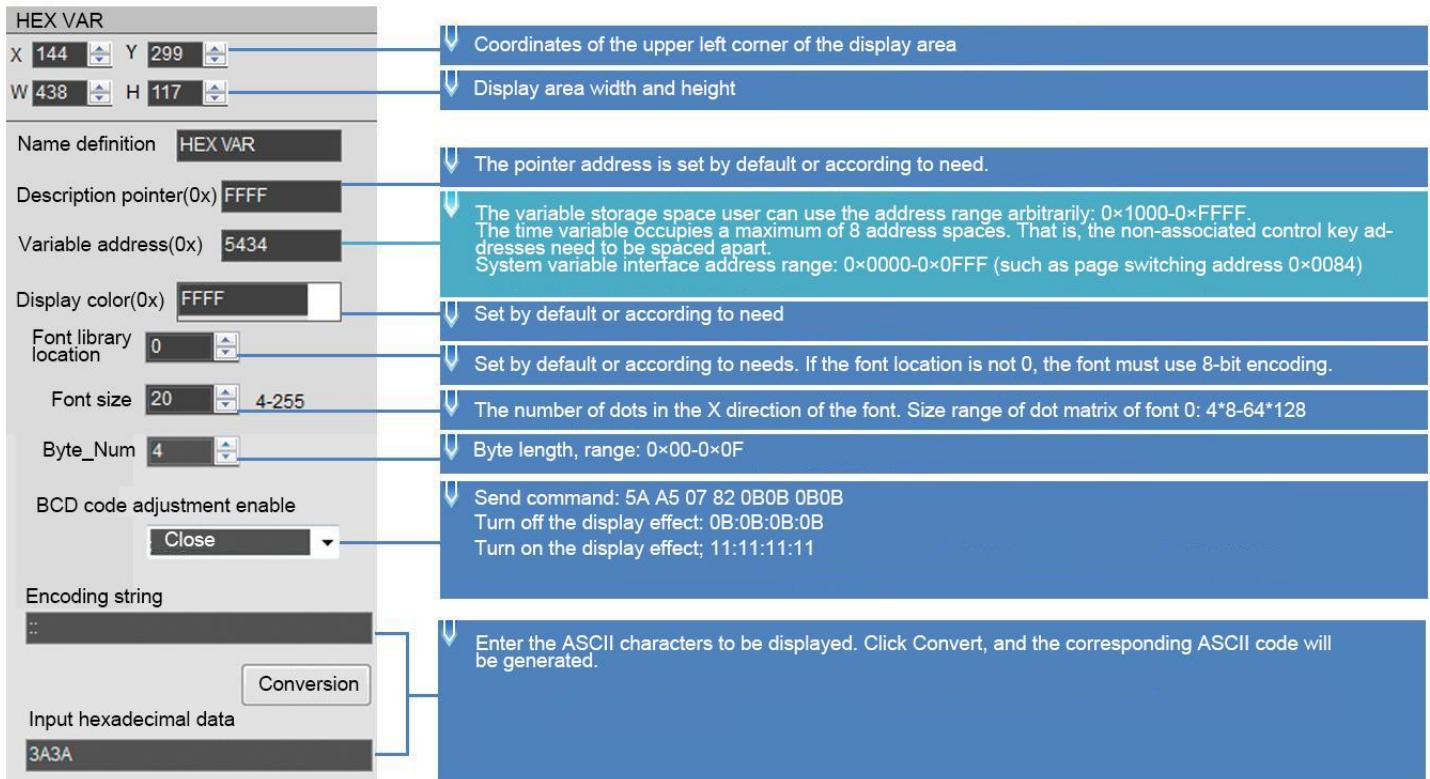
The HEX data control displays variable data in byte HEX intervals with user-specified ASCII characters. It is mostly used for timing display.

For example, 1234 is displayed as 12:34.

7.15.1 Instruction Storage Format

| Address | Definition | Length (bytes) | Description |
|---------|---------------------|----------------|---|
| 0x00 | 0x0001 | 2 | Write 0x0001 |
| 0x01 | SP | 4 | The center of the hands of the clock dial. |
| 0x03 | Icon_Hour | 2 | The ID of the pointer icon. 0xFFFF means that the hour hand is not displayed. |
| 0x04 | Icon_Hour_Central | 4 | The position of the rotation center of the hour hand icon. |
| 0x06 | Icon_Minute | 2 | ID of the minute hand icon. 0xFFFF means the minute hand is not displayed. |
| 0x07 | Icon_Minute_Central | 4 | The position of the center of rotation of the minute hand icon. |
| 0x09 | Icon_Second | 2 | The ID of the second hand icon, 0xFFFF means that the second hand is not displayed. |
| 0x0A | Icon_Second_Central | 4 | The position of the center of rotation of the second hand icon. |
| 0x0C:H | Icon_Lib | 1 | Icon library file ID where the pointer icon is located |
| 0x0C:L | Undefined | 1 | write 0x00 |

7.15.2 Software Setting



7.15.3 Example of Instruction

5A A5 07 82 5434 10 11 12 13

0x5AA5: Frame header;

0x07: Data length;

0x82: Write instruction;

0x5434: VP;

0x10 11 12 13: BCD code.

7.16 Text Scroll

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

7.16.1 Instruction Storage Format

| SP offset | Definition | Length (bytes) | Description |
|-----------|--------------|----------------|---|
| | 0x5A14 | 2 | Fixed value 0x5A14. |
| | SP | 2 | The variable describes the pointer. |
| | 0x000B | 2 | Fixed value 0x000B. |
| 0x00 | VP | 2 | Text Variable pointer. First three VP must be reserved. Text is saved after the 3 rd VP and ended with 0x00 or 0x0F. |
| 0x01:H | Rolling_Mode | 1 | Scroll 0x00: from right to left. |
| 0x01:L | Rolling_Dis | 1 | Rolling space in a DGUS cycle, in pixel. |
| 0x02:H | Adjust_Mode | 1 | 0x00: Left-alignment. 0x01: Right-alignment. 0x02: Center-alignment. Rolling will stop when the text length smaller than textbox. |
| 0x02:L | Undefined | 1 | Write 0x00 |
| 0x03 | Color | 2 | Text color. |
| 0x04 | Xs Ys Xe Ye | 8 | Text box area. |
| 0x08:H | Font0_ID | 1 | The location of ASCII characters in the font library when the encoding mode is 0x01-0x04. If the encoding mode is 0x00 or 0x05, do not set this parameter and write 0x00. |
| 0x08:L | Font1_ID | 1 | The font location of non-ASCII characters when the encoding mode is 0x01-0x04. When the encoding mode is 0x00 or 0x05, the location of the character library used by the character is displayed. |
| 0x09:H | Font_X_Dots | 1 | Font width, in pixels. Must be even. In encoding modes 0x01 -0x04, must be twice the Width. |
| 0x09:L | Font_Y_Dots | 1 | Number of font lattice in Y direction. |
| 0xA:H | Encode_Mode | 1 | Character spacing mode is defined by bit7: 0: Automatic spacing. 1: Fixed spacing. Font encoding is defined by bits 6 to 0: 0x00: 8 bit. 0x01: GB2312. 0x02: GBK. 0x03: BIG5 0x04: SJIS 0x05: UNICODE. |
| 0xA:L | Text_Dis | 1 | Spacing between characters, in pixels. |
| 0xB:H | Reserved | 4 | 0x00 |

7.16.2 Software setting

| | | |
|---|---|--|
| Roll Text | X 533 Y 94 W 157 H 48 | Coordinates of the upper left corner of the display area Display area width and height |
| Name definition | RollText | The variable storage space user can use the address range arbitrarily: 0x1000-0xFFFF. The text display occupies the address according to the length of the text setting. That is, the non-associated control key addresses need to be spaced apart. |
| Description pointer(0x) | FFFF | System variable interface address range: 0x1000-0x0FFF (such as page switching address 0x0084). The first 3 words of the text pointer must be reserved. The text content displayed by the user is stored from (VP+3). The text must end with 0xFF or 0x00. |
| Variable address(0x) | 6010 | |
| Display color(0x) | 0000 | |
| Encoding | 0x02=GBK | The encoding method is determined when the font is generated, and the factory font encoding is GBK. |
| <input checked="" type="checkbox"/> Character spacing is not automatically adjusted | | Unchecked effect: DWIN 12 Checked effect: DWIN 12 |
| Scrolling mode | 0x00= | The text is displayed from right to left |
| Scroll pitch | 2 | The number of pixel dots for text scrolling in each DGUS cycle, 0 to stop scrolling |
| Alignment | 0x00= Align left | The length of the set text content is greater than the length of the display box, otherwise the scrolling will stop. The alignment is valid when the text content is less than the length of the display box. |
| FONT0_ID | 0 | The position of the ASCII font library can be the default font size 0. |
| ASCII font position when encoding method is 0x01-0x04 | | |
| FONT1_ID | 23 | Non-ASCII font location, set the font ID according to your needs. Put it in the DWIN_SET folder in advance, For example, 23_GBK24 Song Ti: It means that the font ID is 23, 24*24 dot matrix, and the encoding method is GBK, SIMSUM. |
|  23_GBK24_simsum.HZK | | |
| X direction array | 24 4-255 | The size of the dot matrix in the X and Y directions of the font library cannot be set arbitrarily. Otherwise, it will display garbled characters. For example, the font library No. 23 above is a 24*24 dot matrix, which is the setting parameter for this. The number of dot matrix in the Y direction must be an even number when generating the font library. |
| Number of dots in Y direction | 24 4-255 | |
| Character interval | 0 | Character interval 0 display effect: DWIN Character interval 20 display effect: DWIN |
| Initial value | Set the length of the text content to be greater than the length of the display box, otherwise it will stop scrolling | The initial text displayed when the power is on, 0xB8 can be written in the x08 position of the CFG file to enable data upload. |

7.16.3 Examples of Instructions

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

0x5AA5: Frame header:

0x13: Data length;

0x82: Write instruction

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

0xBBB6 D3AD C0B4 B5BD B5CF CEC4 BFC6 BCBC, the ASCII code of "Welcome to DWIN Technology".

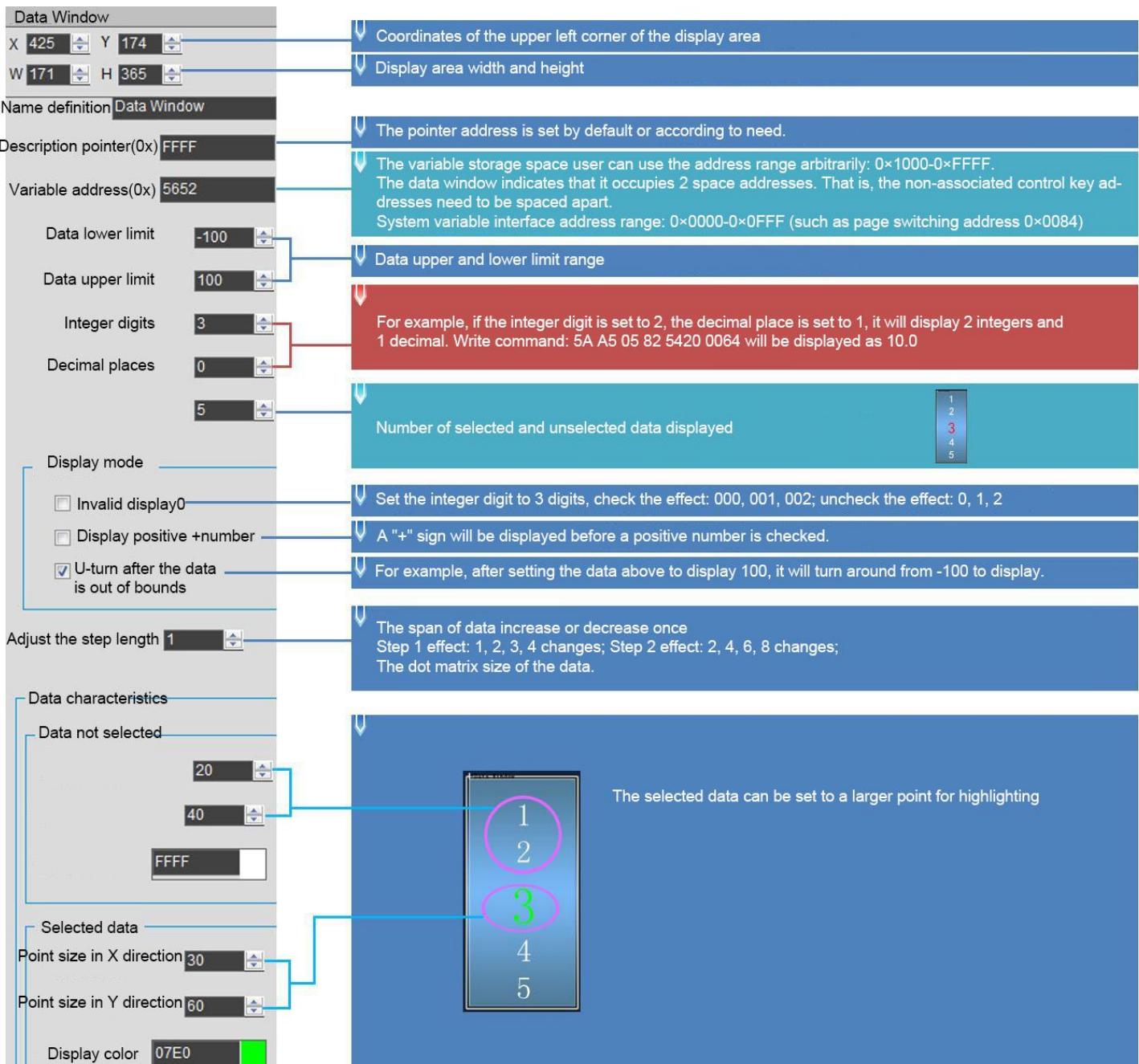
7.17 Data Window

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

7.17.1 Instruction Storage Format

| SP offset | Definition | Length (bytes) | Description |
|-----------|-------------------------|----------------|--|
| 0x00 | *VP | 2 | Variable pointer, variable is a double word, low - value word reserved, high - value word integer data -32768 to +32767. |
| 0x01 | Lower limit of the data | 2 | Lower limit of the data |
| 0x02 | Upper limit of the data | 2 | Upper limit of the data |
| 0x03:H | Integer number | 1 | Integer number |
| 0x03:L | Decimal digits | 1 | Min value. |
| 0x04:H | The number of data | 1 | Display the total number of data |
| 0x04:L | Display mode | 1 | 0x01= Display invalid 0 0x02= display positive number + 0x04= Turn around after out of bounds |
| 0x05 | V_MAX | 4 | The center point coordinate position of the variable displayed. |
| 0x07 | Integer_Digits | 2 | Number of digits to the left of the decimal separator. |
| 0x08:H | Decimal_Digits | 1 | Unchecked data x-direction point size |
| 0x08:L | Data_Num | 1 | Unchecked data y-direction point size |
| 0x09 | Mode | 2 | Data color not selected |
| 0x0A:H | Font size | 1 | Selected data x-direction point size |
| 0x0A:L | Font size | 1 | Selected data Y-direction point size |
| 0x0B | Font color | 2 | Selected data color |
| 0x0C | 0000 | | Reserved write 0000 |

7.17.2 Software Setting



7.17.3 Examples of Instructions

5A A5 05 82 5652 0064

0x5AA5: Frame header;

0x05 Data length;

0x82: Write instruction;

0x5652: VP;

0x0064: Write data 100 to VP

7.18 DGUSII Text(Text Display Without Aliasing)

7.18.1 Instruction Storage Format

| Address | Definition | Length (bytes) | Description |
|---------|---------------|----------------|---|
| 0x00 | *VP | 2 | Text pointer. the max text length is 4Kbytes. |
| 0x01 | (X, Y) | 4 | Text display position: left-aligned mode, the coordinates of the upper left corner of the string. |
| 0x03 | Color | 2 | Display text color. |
| 0x04 | Undefined | 2 | Write 0x00 |
| 0x05 | Xs, Ys Xe, Ye | 8 | Text box. |
| 0x09 | Text_Length | 2 | Display the number of bytes, when it encounters 0xFFFF, 0x0000 or displays to the end of the text box, it will no longer be displayed. |
| 0x0A | LIB_ID | 2 | Font location for non-ASCII characters when the encoding is 0x00, 0x05, and 0x01-0x04. |
| 0x0B:H | Display_Mode | 1 | .7 defines whether the character spacing of the text display is automatically adjusted. .7=0 Character spacing is automatically adjusted. .7=1 Character spacing is not automatically adjusted, character width is fixed. .6 - .0 defines the scale of the character display, 0x00-0x7F, unit 1/16. In fact, DGUS divide 155 steps according to 0.25-8.0 in 0.05 steps, and then converted to 1/16 data. The data is converted to 1/16 corresponding to 0x04-0x7F. |
| 0x0B:L | HOR_Dis | 1 | Characters are spaced horizontally. |
| 0x0C:H | VER_Dis | 1 | Characters are spaced vertically. |
| 0x0C:H | Undefined | 1 | Write 0x00. |

7.18.2 Software Setting

DGUSII text display

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



7.18.3 Example of Instructions

① Display "North 12AB"

5AA5 0B 82 6050 B1B1 3132 4142 FFFF

0x5AA5: Frame header;

0x0B: Data length;

0x82: Write instruction;

0x6050: VP;

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

0x3132: ASCII code of "12";

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

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

7.18.4 Gray Word Library

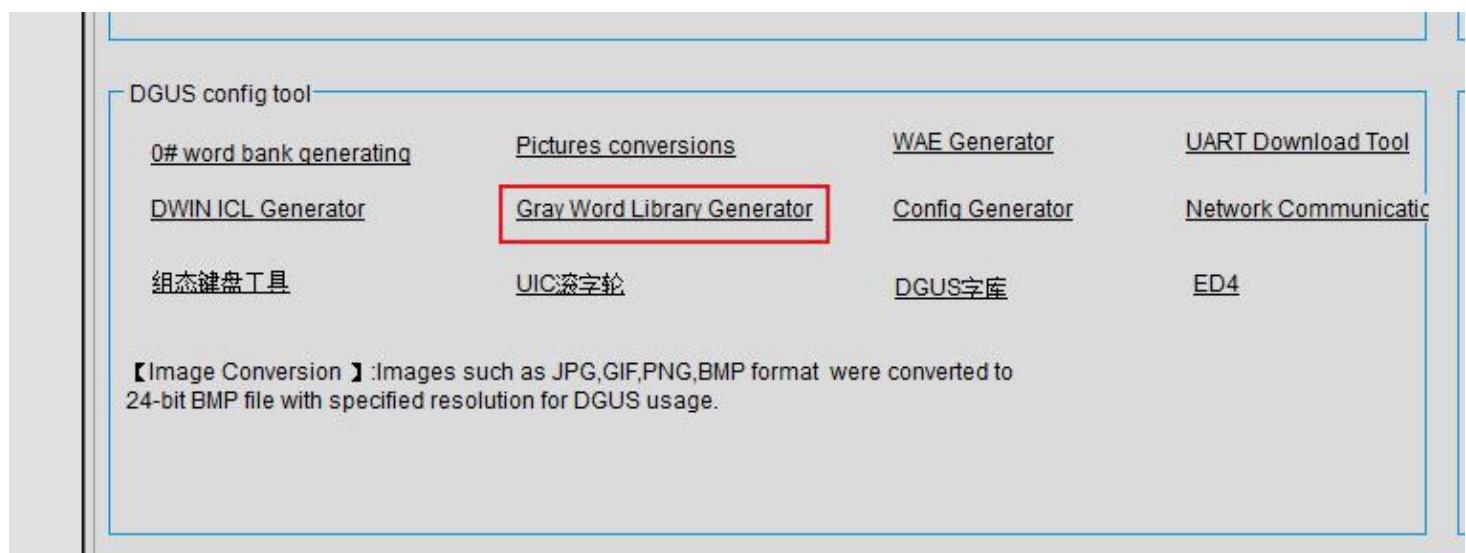
You can use DGUS to generate text display dedicated gray word library.

Features:

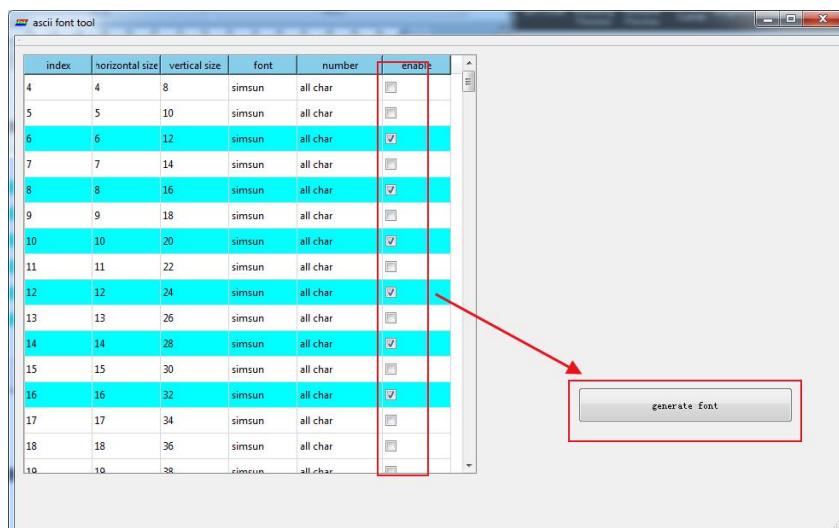
- (1) It can be used to remove the jaggies from the edges of characters, and the edge is smoothly displayed.
 - (2) T5L V45 and above kernel products support this display control.
 - (3) 20*20 dot matrix font size is 5.85MB. You should choose the appropriate dot matrix size accordingly, and arrange reasonably the storage space in advance.
 - (4) When using a large font size such as 24*24 dot matrix (12MB) or more, you need choose/customize products with expanded Flash(>16MB).

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

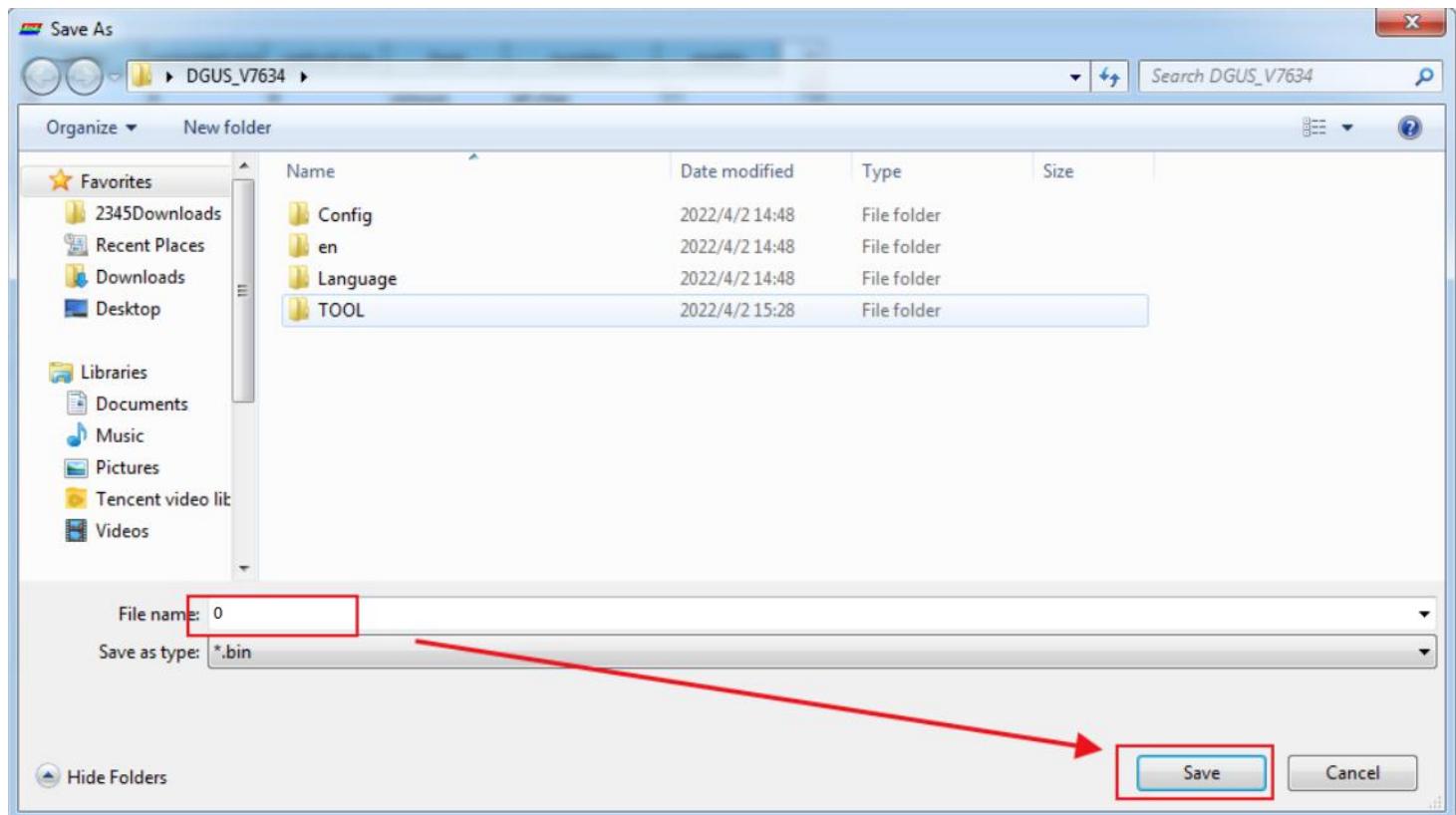
- (1) Open DGUS and click Gray Word Library Generator.



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



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



7.19 Roll Character

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

7.19.1 Instruction Storage Format

| Address | SP Offset | Definition | Data length (byte) | Description |
|---------|-----------|--------------|--------------------|---|
| 0x00 | | 0x5A17 | 2 | |
| 0x02 | | SP | 2 | Variable description pointer, 0xFFFF means loaded by configuration file. |
| 0x04 | | 0x000D | 2 | |
| 0x06 | 0x00 | VP | 2 | Data pointer. Each data occupies four words of storage space, defined as follows: VP=storage address of the selected data, VP+1=adjustment parameter, corresponding to gesture adjustment VP+1 or incremental adjustment VP. VP+2=System reserved, display offset, integer. |
| 0x08 | 0x01:H | Adj_Mod | 1 | The high 4bit is the data type: 0x0=Integer (2 bytes), -32768 to 32767 0x1=*VP high byte, unsigned number 0-255 0x2=*VP low byte, unsigned number 0-255 0xE=*VP The data directly corresponds to the font ID 0-255, which is suitable for small screen input keyboard. 0xF=*VP data is an ASCII string pointer, and each line can contain up to 256 characters. The low 4bit is (number of character lines-1)/2, 0x0-0x4, up to 9 lines. |
| 0x09 | 0x01:L | Data_Mod | 1 | Data mode: The upper 4bit is the number of integers, 0x00-0x05. The lower 4bit is the number of decimal places, 0x00-0x05. String mode: character pointer interval (word length), 0x01-0xFF. |
| 0x0A | 0x02 | VP_String | 2 | Data mode: interval step length of data variable, 0x0001-0x7FFF. String pattern: The string variable storage pointer corresponding to the starting value (0x00), 0xFF represents the end of data. |
| 0x0C | 0x03 | V_Min | 2 | The lower limit of the data, fixed-point integer. |
| 0x0E | 0x04 | V_Max | 2 | The lower limit of the data, fixed-point integer. |
| 0x10 | 0x05:H | Display_Mode | 1 | Display mode: .7 1=invalid 0 displayed; 0=invalid 0 not displayed. .6-.4 1=The selected line displays the Font0; 0x0-0x7. .3 1=The positive "+" is displayed; 0=The positive "+" is not displayed. .2 1=Character background is not filtered out; 0=Character background is filtered out. .1-.0 Alignment mode: 00=center, 01=left alignment, 02=right alignment. |
| 0x11 | 0x05:L | Speed_Set | 1 | .7-.6 Data change speed, 0x00-0x03, 0x00 is the slowest. .5-.0 Scroll speed (pixels scrolled in each DGUS cycle), 0x01-0x3F. The larger the value, the faster the scrolling. The recommended value is 1/16 of the 0 line spacing. |
| 0x12 | 0x06 | Font_ID | 2 | Select the configuration icon font number, 0x0000-0xFFFF. |
| 0x14 | 0x07 | (X, Y) | 4 | Show the display coordinates of the selected row. Center mode: the center coordinates of the selected line; left alignment mode: the left midpoint coordinates of the first character of the selected line; right alignment mode: the right midpoint coordinates of the last character of the selected line; |

| | | | | |
|------|--------|---------------|---|---|
| 0x18 | 0x09:H | Line_Height0 | 1 | Line spacing 0 (the spacing between the selected line and the top 1 line; the spacing between the selected line and the bottom 1 line is also this value, treated symmetrically; the same below.) Height (pixels in the Y direction). |
| 0x19 | 0x09:L | Line_Height1 | 1 | Line space 1 (the space between the top 1 line and the top 2 lines) height (pixels in the Y direction). |
| 0x1A | 0x0A:H | Line_Height2 | 1 | Line space 1 (the space between the top 2 lines and the top 3 lines) height (pixels in the Y direction). |
| 0x1B | 0x0A:L | Line_Height3 | 1 | Line space 1 (the space between the top 3 lines and the top 4 lines) height (pixels in the Y direction). |
| 0x1C | 0x0B:H | DIM_No_Select | 1 | The window brightness is not selected, 0x00-0xFF. 0x00 is the darkest, 0xFF is the brightest; combined with the background. |
| 0x1D | 0x0B:L | Height_Sel | 1 | The height of the display area of the selected line must be higher than the character height of Font0. |
| 0x1E | 0x0C:H | Font1:2 | 1 | Higher 4bit, the font of 1 line above, 0x00-0x07; Lower 4bit, the font of 2 lines above, 0x00-0x07; |
| 0x1F | 0x0C:L | Font3:4 | 1 | Higher 4bit , the font of 3 lines above, 0x00-0x07; Lower 4bit, the font of 4 lines above, 0x00-0x07; |

7.19.2 Software Setting

Roller Character

| | | |
|-------|-------|--|
| X 411 | Y 194 | Coordinates of the upper left corner of the display area |
| W 133 | H 358 | Display area width and height |

Name definition **Roller character**

Description pointer(0x) **FFFF**

Variable address(0x) **5840**

type of data **Integer (2 bytes)**

Number of character lines **7**

Integer digits **5**

Decimal places **0**

Word length **0**

String address (0x) **0000**

Data lower limit **0**

Data upper limit **100**

Display mode

- Invalid display0
- Display positive + sign
- Character background is not filtered

Alignment mode **Centered**

Scroll speed **5**

Icon library number **140**

Selected window height **80**

Unchecked window brightness **80**

Select the first row up

Spacing from the previous line **58**

Line font number **1**

Select the second row from the top

Spacing from the previous line **48**

Line font number **2**

Select the third row up

Spacing from the previous line **40**

Line font number **3**

Row middle row up fourth row

Spacing from the previous line **0**

Line font number **0**

Coordinates of the upper left corner of the display area

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

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

The number of lines of character display, for example, the intuitive effect of setting 7 lines is shown on the right. For example, the integer digit is set to 3 digits, and the decimal digit is set to 2 digits. The intuitive effect of displaying data 512.512

Data upper and lower limit range

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

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

Alignment mode: center, left, right, center alignment is recommended

Scrolling speed (pixels rolled in each DGUS cycle), 0x01-0xFF. The larger the value, the faster the scrolling. The recommended value is /16 of the line spacing 0. The height of the selected window is 80/16=5(d setting value on the left)

Line spacing

Line spacing

Line spacing

Line spacing

Configuration icon. UIC file naming number. Naming ID = number/8 (ID does not conflict with other words)

For example, 140/8=17.5 (the occupied position does not conflict with fonts such as 13, 14), 256/8=32KB, a single IC file block occupies the 32KB interval.

The height of the display area of the selected line must be higher than the character height of Font0.

The selected font is 0#, the height on the software is 46, and the 80 display will be looser, and it can be set according to actual needs.

Shows the brightness of the selected window, 0x00-0xFF. 0x00 is the darkest, 0xFF is the brightest; combined with the background. The scope of the unselected window area and the intuitive effect are shown in the right figure. The function is to change the font brightness of the selected area and highlight the selected font.

Here, the font numbers 0, 1, 2, and 3 correspond to the numbers defined in the font settings of the configuration icon font library software welcome interface.

The distance between the previous line and the previous line is set according to the height of the font.

For example, the line font number is 1, the software is 30 occupies the array height, the setting 58 is to make the interval look a little looser, setting 34/38/42 can be displayed.

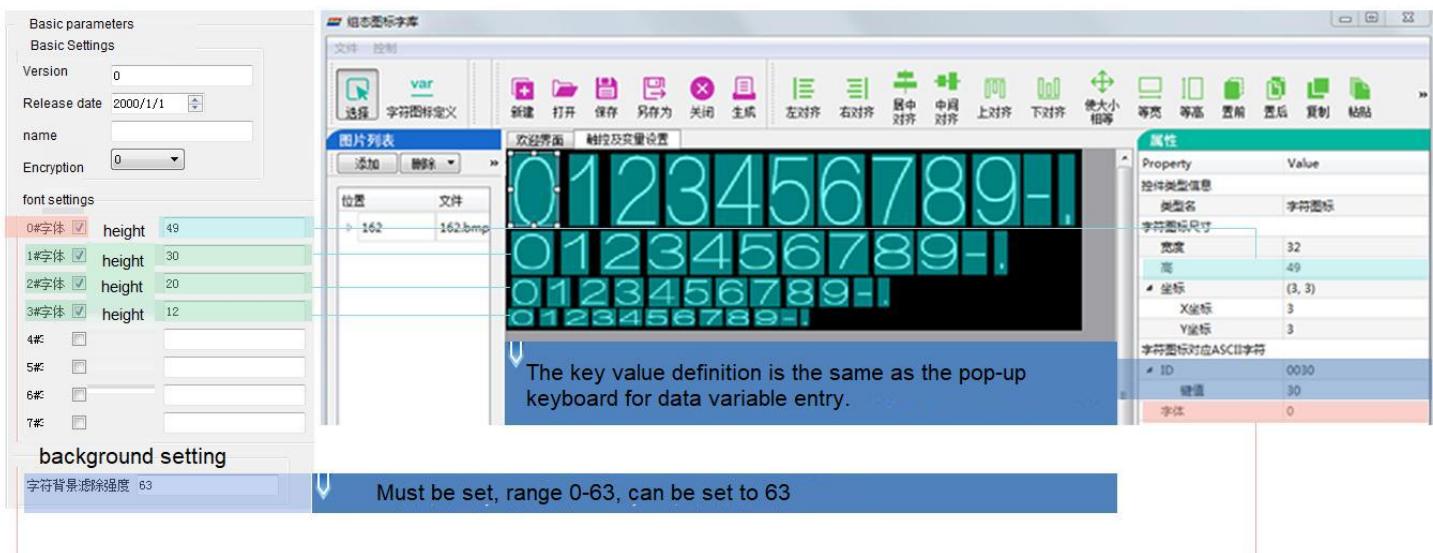
Font settings

| | |
|--------------------|---|
| 0#Font | Font height: 46 |
| 1#Font | Font height: 30 |
| 2#Font | Font height: 18 |
| 3#Font | Font height: 12 |
| 4#Font | Font height: |
| 5#Font | Font height: |
| 6#Font | Font height: |
| 7#Font | Font height: |
| Background setting | Character background filtering strength |

Step 1: Create a new 477*124 pixel canvas, arrange the numbers in order, and set the canvas size according to actual needs.



Step 2: Open the configuration icon font library software - click the new project - the resolution is modified to be the same as the canvas -----Select the saved file path-----OK- -----Add canvas picture-----click the character icon definition, select the range of the intercepted icon, the height of each row should be the same, the width can be inconsistent, the same row, you can use the array----- - After configuration, click Save, name it such as 140, and put it in the DWIN-SET folder to call it.



7.19.3 Example of Instructions

5AA5 04 82 5840 0001

0x5AA5: Frame header;

0x04: Data length;

0x82: Write instruction;

0x5840: VP;

0x0001: Data value 1.

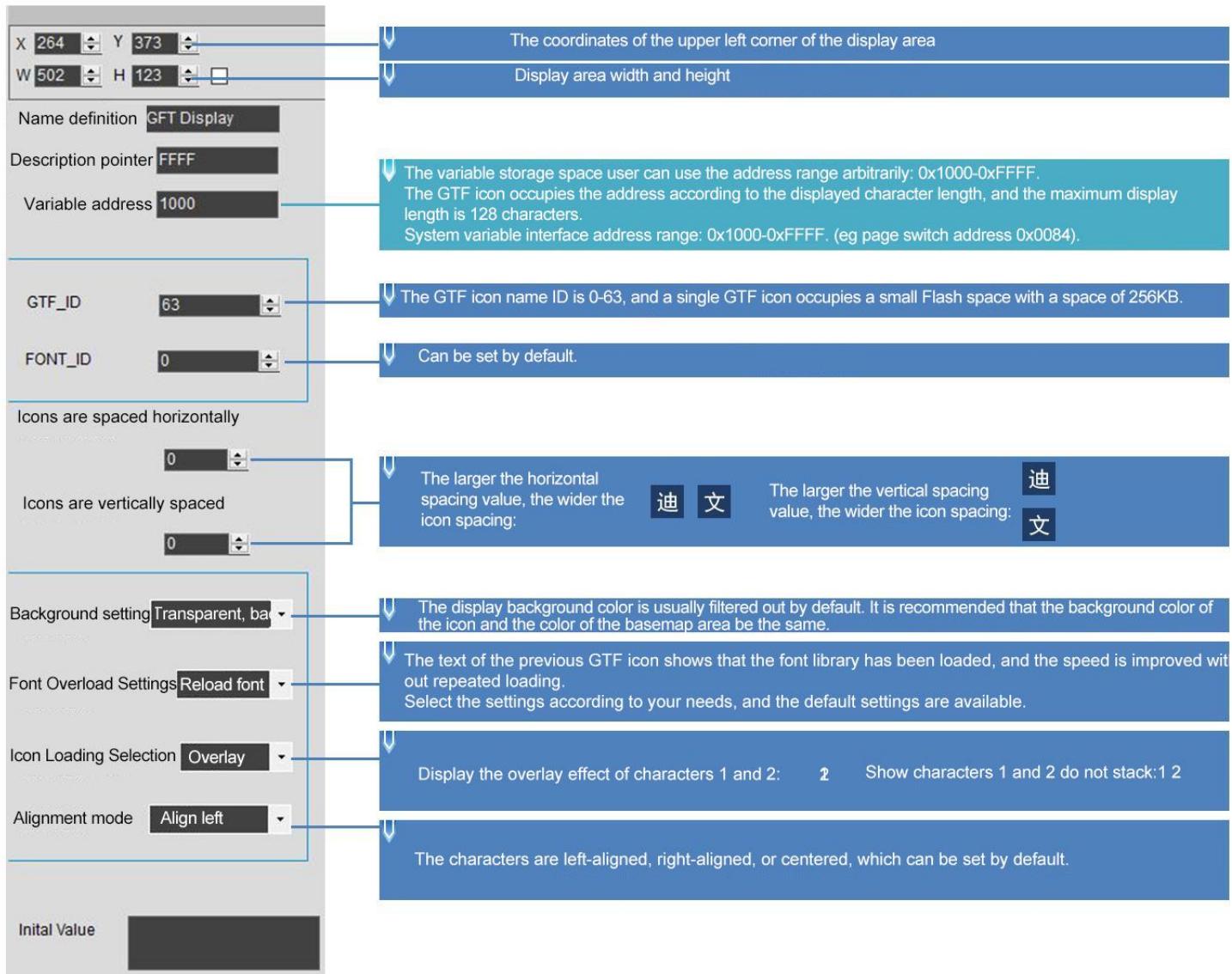
7.20 GTF Icon Font

The GTF icon control supports efficient display of icon characters.

7.20.1 Instruction Storage Format

| Address | SP offset | Definition | Length (bytes) | Description |
|-------------|-----------|----------------------|----------------|---|
| 0x00 | | 0x5A18 | 2 | |
| 0x02 | | SP | 2 | Variable description pointer, 0xFFFF means loaded by the configuration file. |
| 0x04 | | 0x000B | 2 | |
| 0x06 | 0x00:H | *VP | 1 | Text pointer, must be an even address. The maximum size of text variable data is 255bytes. 0xFFFF, 0x0000 data or display to the end of the text box will no longer be displayed. |
| 0x08 | 0x00:L | GTF_ID | 1 | Use GTF font encoding, 0x0000-0xFFFF. |
| 0x0A | 0x01 | Font_ID | 8 | Use the font code ID in the GTF font library, 0x0000-0x03FB. |
| 0x0C | 0x05 | (X, Y) | 2 | Start display position. Left alignment mode: the coordinates of the upper left corner displayed on the first line; Right-aligned mode: the coordinates of the upper right corner displayed in the first line; Centering mode: undefined, any value can be used. |
| 0x10 | 0x06 | (Xs, Ys) (Xe ,Ye) | 8 | The coordinates of the upper left and lower right corners of the text box. |
| 0x18 | 0x07 | HOR_Dis | 1 | Icons are spaced horizontally. |
| 0x19 | 0x08 | VER_Dis | 1 | Icons are vertically spaced. |
| 0x1A | 0x0A:H | Display_Mode | 1 | .7 Background transparency options: 0 is transparent, the background is not displayed; 1 Opaque, background displayed. .6 font reload. 0 Reload font library (the first GTF icon text display of the page must be loaded); 1 The previous GTF icon text shows that the font library has been loaded, and the speed is improved without repeated loading. .5 Icon Overlay Selection 0 is not superimposed, and the coordinate position automatically moves backward after an icon is displayed; 1 Icons are displayed superimposed. .4-.2 Reserved, write 0. .1-0 alignment mode: 0=left-aligned 1=centered 2=right-aligned |
| 0x1B | 0x0A:L | Undefined | 1 | Write 0x00 |

7.20.2 Software Setting



7.20.3 GTF TOOL Software Setup Instruction

(1) Software tool: GTF Tool

(2) Setting

- ① Click the "+" in the lower left corner to add a picture;
- ② Use the "rectangle" control to define key values;
- ③ Box-select the character icon area;
- ④ Double-click the control to define the key value;

Key value range 0x01-0xFF;

0-9 key value: 0x30-0x39;

A-Z key value: 0x41-0x5A;

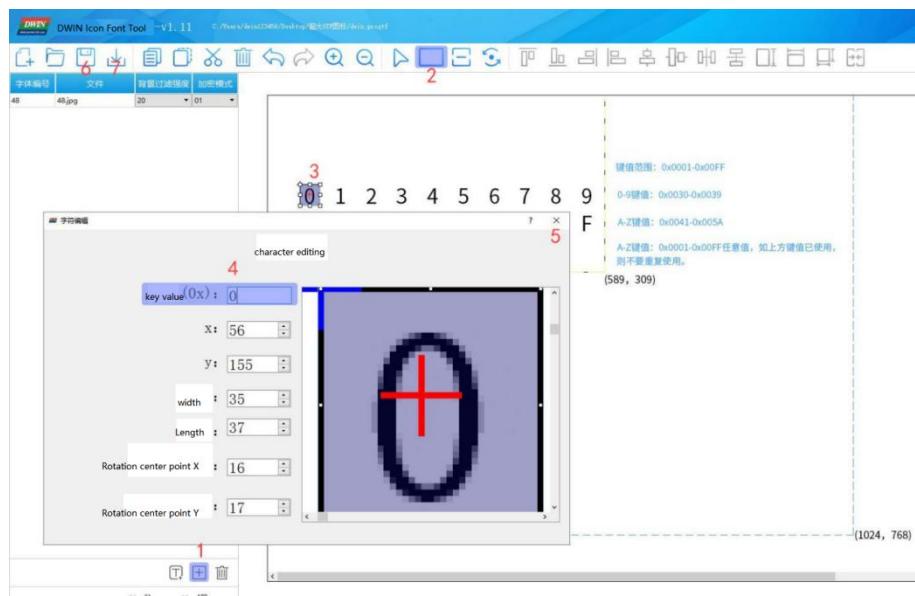
Other key values can be used arbitrarily as needed.

- ⑤ Click "x" to save and close;
- ⑥ Click "Save" to generate the configuration file;
- ⑦ Click "Generate" to save the GTF icon library file.

Video tutorial: <http://inforum.dwin.com.cn:20080/forum.php?mod=viewthread&tid=5241&dsign=51ea05f0>

(3) Precautions

- It is recommended that the GTF icon should not exceed 1920*1080, and the icon will not be displayed if it is too large;
- If the original image of GTF does not display the bmp and jpg images output by the bitmap software, you can use the bmp and jpg images of the vector graphics software;
- The FTG icon name ID is 0-63, and a single GTF icon occupies the Flash space with a subspace size of 256KB.



7.20.4 Example of Instructions

The max text variable data is 255 bytes, and the variable address is reserved as needed.

① Display character 0

Tx: 5A A5 04 82 1000 30

② Display character 12

Tx: 5A A5 05 82 1000 3132

③ Display character 0123456789

Tx: 5A A5 0D 82 1000 3031 3233 3435 3637 3839

④ Display character A

Tx: 5A A5 04 82 1000 41

⑤ Display characters ABCD, remove redundant characters and add terminator FFFF

Tx: 5A A5 09 82 1000 4142 4344 FFFF

⑥ "China" on the display icon

Tx: 5A A5 07 82 1000 0102 FFFF

⑦ Display icon

Tx: 5A A5 05 82 1000 0304

⑧ "Weather" character on the display icon

Tx: 5A A5 07 82 1000 0506 FFFF

⑨ Clear all characters

Tx: 5A A5 05 82 1000 FFFF

7.21 Real-Time Curves

The real-time curves control is to automatically match and display the real-time curve (trend graph) by writing the curve buffer data with the 0x84 instruction.

The display area, the coordinates of the central axis and the display scale (enlargement/reduction) can be set by users.

7.21.1 Instruction Storage Format

| SP offset | Definition | Length (bytes) | Description |
|---------------|---------------|----------------|--|
| 0x00:H | 0x5A20 | 1 | 0x00=The latest data is on the far right, and the curve moves from right to left. Other = The latest data is on the far left, the curve moves from left to right. |
| 0x00:L | SP | 1 | Undefined |
| 0x01 | Xs, Ys Xe, Ye | 8 | The coordinates of the upper left corner of the curve window (Xs, Ys) and the coordinates of the lower right corner (Xe, Ye) Curve out of bounds will not be displayed. |
| 0x05 | Y_Central | 2 | The position of the center axis of the curve |
| 0x06 | VD_Central | 2 | The curve data value corresponding to the central axis generally takes half of the sum of the maximum and minimum values of the data. |
| 0x07 | Color | 2 | Curve color |
| 0x08 | MUL_Y | 2 | The magnification of the vertical axis, the unit is 1/256, 0x0000-0x7FFF. |
| 0x09:H | CHANNEL | 1 | Data source channel, 0x00-0x07 |
| 0x09:L | Dis_HOR | 1 | Horizontal axis interval, 0x01-0xFF. |
| 0x0A:H | Pixel_Scale | 1 | Curve pixel lattice size (curve line width), 0x00-0x07 corresponds to 1*1 to 8*8. |
| 0x0A:L | Reserved | 1 | Write 0x00 |

7.21.2 Software Setting

Open DGUS, click display control – real-time curves, then set the control display area, and configure the curve color, data source channel, etc. in the setting menu on the right.

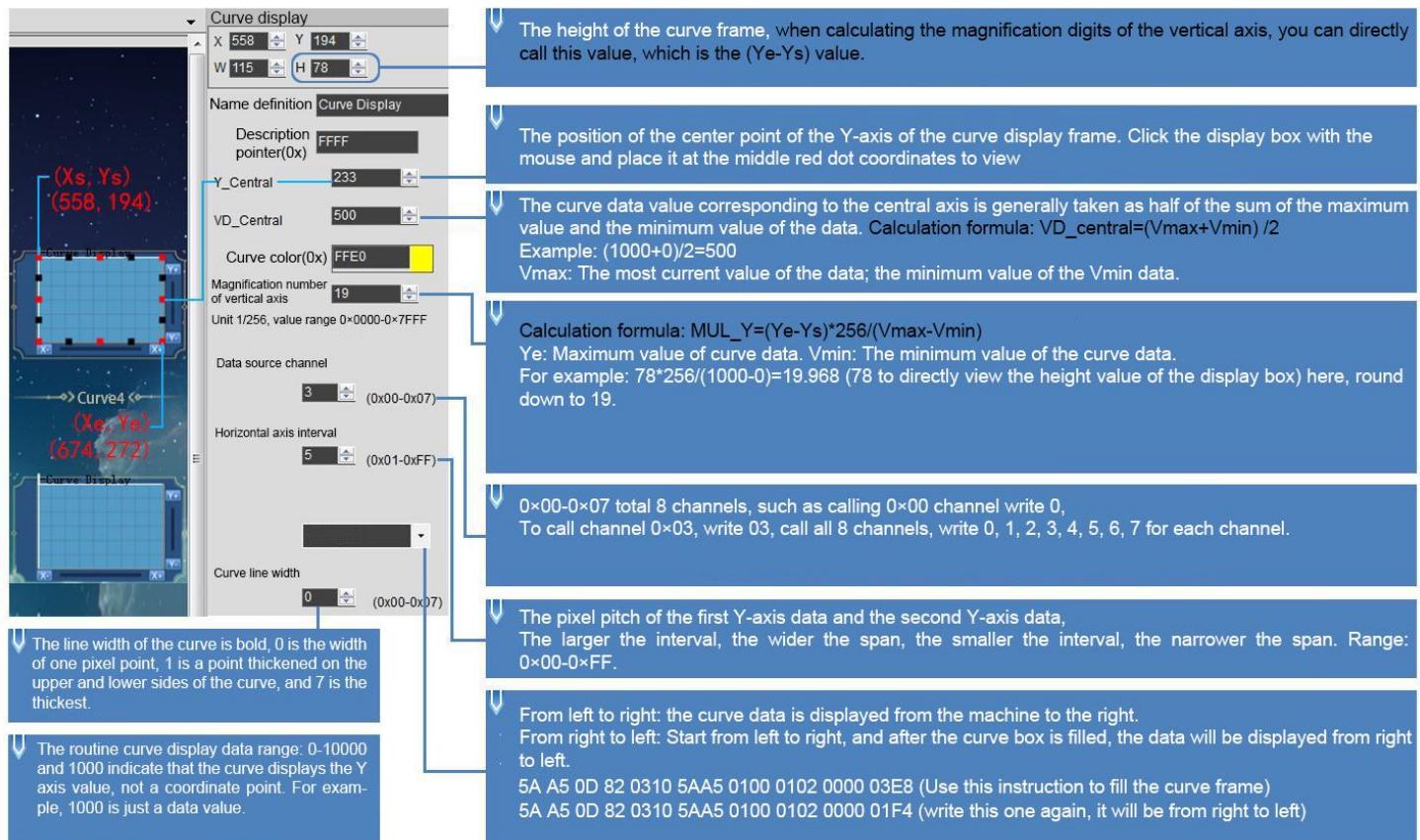
If you need to display a thicker curve line, you can place multiple up and down (Y-axis) translation curve variables in the adjacent position and refer to the same data source to achieve this. That is, press Ctrl+C, Ctrl+V to copy a set of curve controls.

Click the copied curve box and press up and down keys of the keyboard. The more control you copy, the thicker the curve line.

[Note] The configuration and usage examples with touch controls are as follows.

If the content of vp is stored in sp, combined with incremental adjustment control, adjusting the +0x09 low byte access address, you can realize automatic scaling of the curve without code.

Also, if combined with drag adjustment control, adjusting +0x05 to modify the Y_Central value, the curve can be moved up and down without code.



7.21.3 Example of Instructions

| VP | Explanation |
|-------------|---|
| 0x300-0x30F | <p>Status feedback of 8 channel curve buffers (Read-only for users). Each channel occupies 2 words.</p> <p>The storage pointer address of the high byte storage curve data (0x0000-0x07FF), The storage pointer address of the low byte storage curve data (0x0000-0x0800). Write 0x0000 to the effective Data length of the curve buffer will cause the curve not to be displayed.</p> <p>Example: 5A A5 05 82 0301 0000 The curve 0 channel will be cleared, and it will be restored if the value is rewritten; 5A A5 05 82 0303 0000 Curve 1 channel will be cleared, and it will be restored if the value is rewritten; 5A A5 05 82 0305 0000 Curve 2 channel will be cleared, and it will be restored if the value is rewritten; 5A A5 05 82 0307 0000 Curve 3 channel will be cleared, and it will be restored if the value is rewritten; 5A A5 05 82 0309 0000 Curve 4 channel will be cleared, and it will be restored if the value is rewritten; 5A A5 05 82 030B 0000 Curve 5 channel will be cleared, and it will be restored if the value is rewritten; 5A A5 05 82 030D 0000 Curve 6 channel will be cleared, and it will be restored if the value is rewritten; 5A A5 05 82 030F 0000 Curve 7 channel will be cleared, and it will be restored if the value is rewritten;</p> |
| 0x310-0x311 | <p>Write start to curve buffer. D3: D2: 0x5AA5 Start a curve buffer data write operation, and the CPU will clear it after the operation. D1: The number of data blocks, 0x01-0x08. D0: undefined, write 0x00.</p> |
| 0x312-0x37F | <p>The data block that needs to be written into the curve buffer, the data is a 16bit unsigned number. A single block of data is defined as a data channel ID (0x00-0x07) + data word length (0x01-0x6E) + data.</p> |

① 0x00 channel display data 0 and 1000

5A A5 0D 82 0310 5AA5 0100 0002 0000 03E8

0x5AA5: Frame header;

0x0D: Data length;

0x82: Write instruction;

0x0310: The curve buffer data writing is started; it is the system variable hardware interface curve fixed address;

0x5AA5: Start a curve buffer data writing operation, and clear it after the CPU operation;

0x0100: The number of high byte 0x01 curve data blocks, that is, the number of channels occupied; write 0x00 in case of low byte undefined;

0x0002: High byte 0x00 represents channel 0, low byte 0x02 represents the number of data word lengths. In this example, it is a two-word data, namely 0x0000, 0x03E8;

0x0000: The data to be displayed 0;

0x03E8: The data to be displayed 1000.

② 0x06 channel display data 0 and 1000

5A A5 0D 82 0310 5AA5 0100 0602 03E8 0000

③ 0x07 channel display data 0 and 1000:

5A A5 0D 82 0310 5AA5 0100 0702 0000 03E8

④ 0x00 and 0x07 channel display data 0 and 1000

5A A5 13 82 0310 5AA5 0200 0002 0000 03E8 0702 0000 03E8

0x5AA5: Frame header;

0x0D: Data length;

0x82: Write instruction;

0x0310: The curve buffer data writing is started; it is the system variable hardware interface curve fixed address;

0x5AA5: Start a curve buffer data writing operation, and clear it after the CPU operation;

0x0200: The number of high byte 0x02 curve data blocks, that is, the number of channels occupied; If an instruction writes 8 channels, this byte needs to be changed to 08; write 0x00 in case of low byte undefined;

0x0002: High byte 0x00 represents channel 0, low byte 0x02 represents the number of data word lengths. In this example, it is a two-word data, namely 0x0000, 0x03E8;

0x0000: The data to be displayed 0;

0x03E8: The data to be displayed 1000;

0x0702: High byte 0x07 represents channel 7, low byte 0x02 represents the number of data word lengths. In this example, it is a two-word data, namely 0x0000, 0x03E8;

0x0000: The data to be displayed 0;

0x03E8: The data to be displayed 1000

The channel and data word lengths of the two curves are written separately.

⑤ Channels 0x06 and 0x07 display two cross curves and display from right to left

Instruction 1: 5A A5 13 82 0310 5AA5 0200 0602 0000 03E8 0702 03E8 0000

Instruction 1 is to form a cross curve, where the maximum and minimum values of the two curves are set opposite, so that the middle of the curves can be crossed.

The cross curve is actually two different channel curves, so you need to make two curve controls and set different channels.

Instruction 2: 5A A5 13 82 0310 5AA5 0200 0602 0064 01F4 0702 01F4 0064

On the right-to-left setting screen of the software, it is displayed from left to right at first. When the curve box is filled, write instruction 2, and then you can see that the curve is displayed from right to left.

⑥ 0x00-0x078 channels are all displayed

5A A5 37 82 0310 5AA5 0800 0002 0000 03E8 0102 0000 03E8 0202 0000 03E8 0302 0000 03E8 0402 0000
03E8 0502 0000 03E8 0602 0000 03E8 0702 0000 03E8

0x00-0x07 Channels Each channel can be refreshed separately in turn to achieve the effect of refreshing different channel curves at the same time.

T5L DGUS curve 8-channel display, a total of 8 channels, each channel can store 2048 words.

After enabling dynamic curve display, starting from 0x1000, a data buffer is established for each curve according to 2Kwords per channel.

CH0 buffer is 0x1000-0x17FF, CH1 buffer is 0x1800-0x1FFF,

CH2 buffer is 0x2000-0x27FF, CH3 buffer is 0x2800-0x2FFF,

CH4 buffer is 0x3000-0x37FF, CH5 buffer is 0x3800-0x3FFF,

CH6 buffer is 0x4000-0x47FF, CH7 buffer is 0x4800-0x4FFF,

Unused curve buffers can be used as user vp. User can also directly rewrite the curve buffer,

For example, directly rewrite the variable address 5A A5 05 82 1000 0065...

The value of the corresponding channel of the curve will change accordingly.

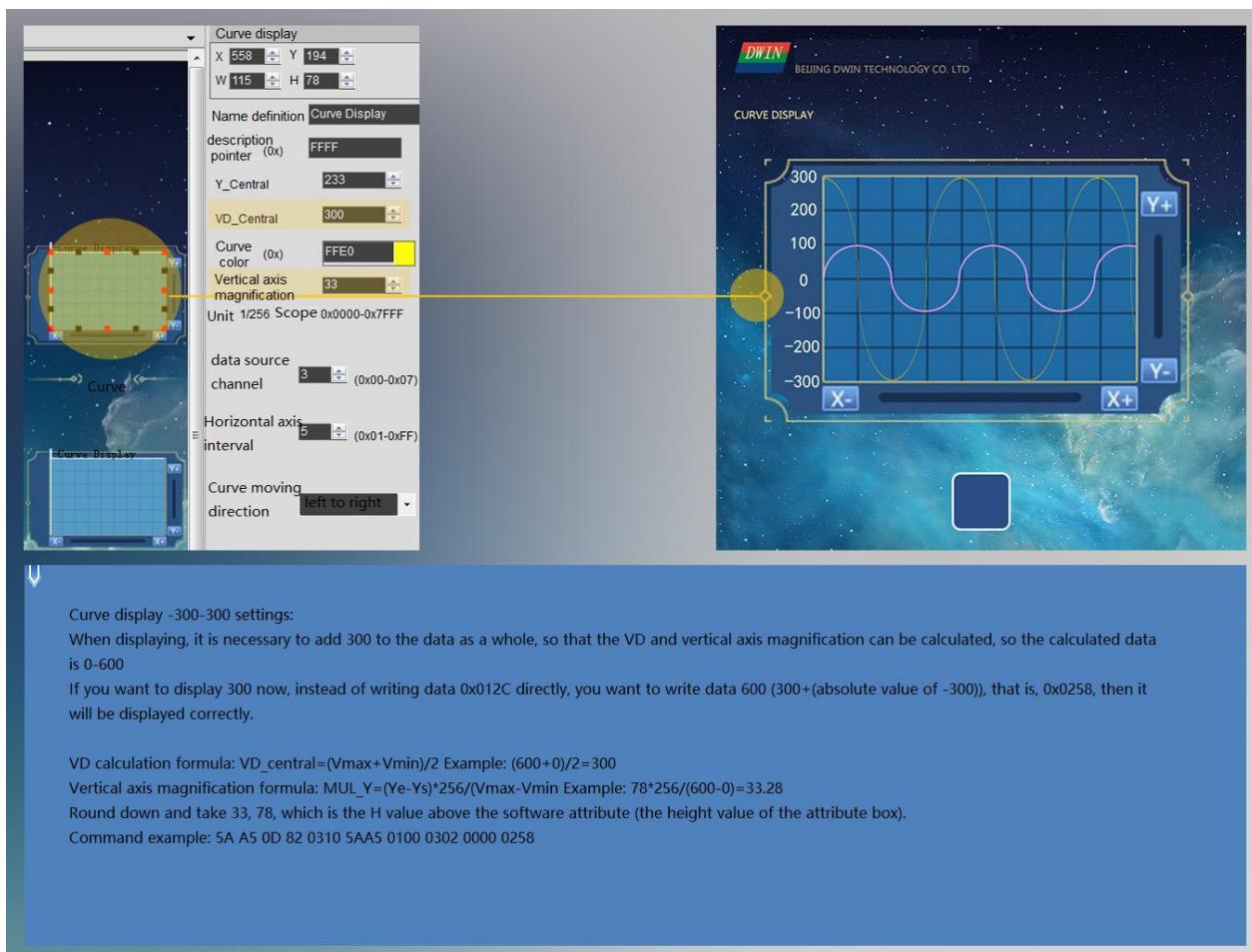
If the user needs to use 8 channels, the variable addresses and description pointers of other display controls are used from the range of 0x5000~0xFFFF.

⑦ Real-time curves SP usage and example

The SP storage address format of this function and some examples of commonly used access instructions are as below.

| SP | Instruction (SP of the example is set to 0x8000) | Instruction Effect and Application |
|--------|--|---|
| 0x05 | 5A A5 05 82 8005 0064 | Change the position of the center axis of the curve (the curve moves up and down): move it up and down to the y axis100 position. |
| 0x06 | 5A A5 05 82 8006 0064 | Change the range of the curve (vertical scaling of the curve): the curve data value corresponding to the central axis, generally half of the sum of the maximum and minimum data. For example, here the instruction is changed to 0x200 range, (corresponding magnification needs to be modified in the 0x08 pointer address) |
| 0x07 | 5A A5 05 82 8007 F800 | The color of the curve display changes to red |
| 0x08 | 5A A5 05 82 8008 00F0 | Modify the magnification of the vertical axis, the unit is 1/256, 0x0000-0x7FFF. (Need to be used with 0x06 pointer) |
| 0x09:H | | Data source channel, 0x00-0x07 |
| 0x09:L | 5A A5 05 82 8009 0014 | Change the horizontal axis interval (curve horizontal scaling): Display the horizontal axis interval of the 00 channel curve to 0x14 (range 0x01-0xFF). |

⑧ Negative number display



7.22 Basic Graphics

The basic graphics control is to define a "drawing board" function in the display configuration file 14.BIN, and the specific drawing operation is determined by the content of the variable memory pointed to by *VP.

Users can realize different drawing functions by changing the data in the variable memory.

7.22.1 Instruction Storage Format.

| SP offset | Definition | Length (bytes) | Description |
|-----------|----------------|----------------|---|
| 0x00 | *VP | 2 | Variable data pointer |
| 0x01 | Area | 8 | The coordinates of the upper left corner and the lower right corner of the drawing display area; if the drawing exceeds the boundary, it will not be displayed. only for 0x0001-0x0005, 0x0009, 0x000A, 0x000B instructions are valid. |
| 0x05:H | Dashed_Line_En | 1 | 0x5A: Drawing instructions using line segments (0x02, 0x03, 0x09, 0x0A instructions) will use dotted lines or dotted lines to display line segments; Miscellaneous: Drawing instructions that use line segments display line segments with solid lines. |
| 0x05:L | Dash_Set | 4 | The 4 bytes are sequentially formatted with dotted lines (dot-dash): The number of solid lines in the first segment, the number of dotted lines in the first segment, the number of solid lines in the second segment, and the number of dotted lines in the second segment. For example, setting 0x10 0x04 0x10 0x04 will display dashed lines; setting 0x10 0x04 0x02 0x04 will display dotted lines. |
| 0x07 | Undefined | 13 | Reserved 0x00 |

The description of the variable data format pointed to by the variable data pointer (variable storage space) is shown in the table.

| Address | Definition | Description |
|---------|-------------------|---|
| VP | CMD | Drawing instructions |
| VP+1 | Data_Pack_Num_Max | Max data packet: connection (0x0002), defined as the number of connection lines (the number of vertices-1); |
| VP+2 | DATA_Pack | Data |

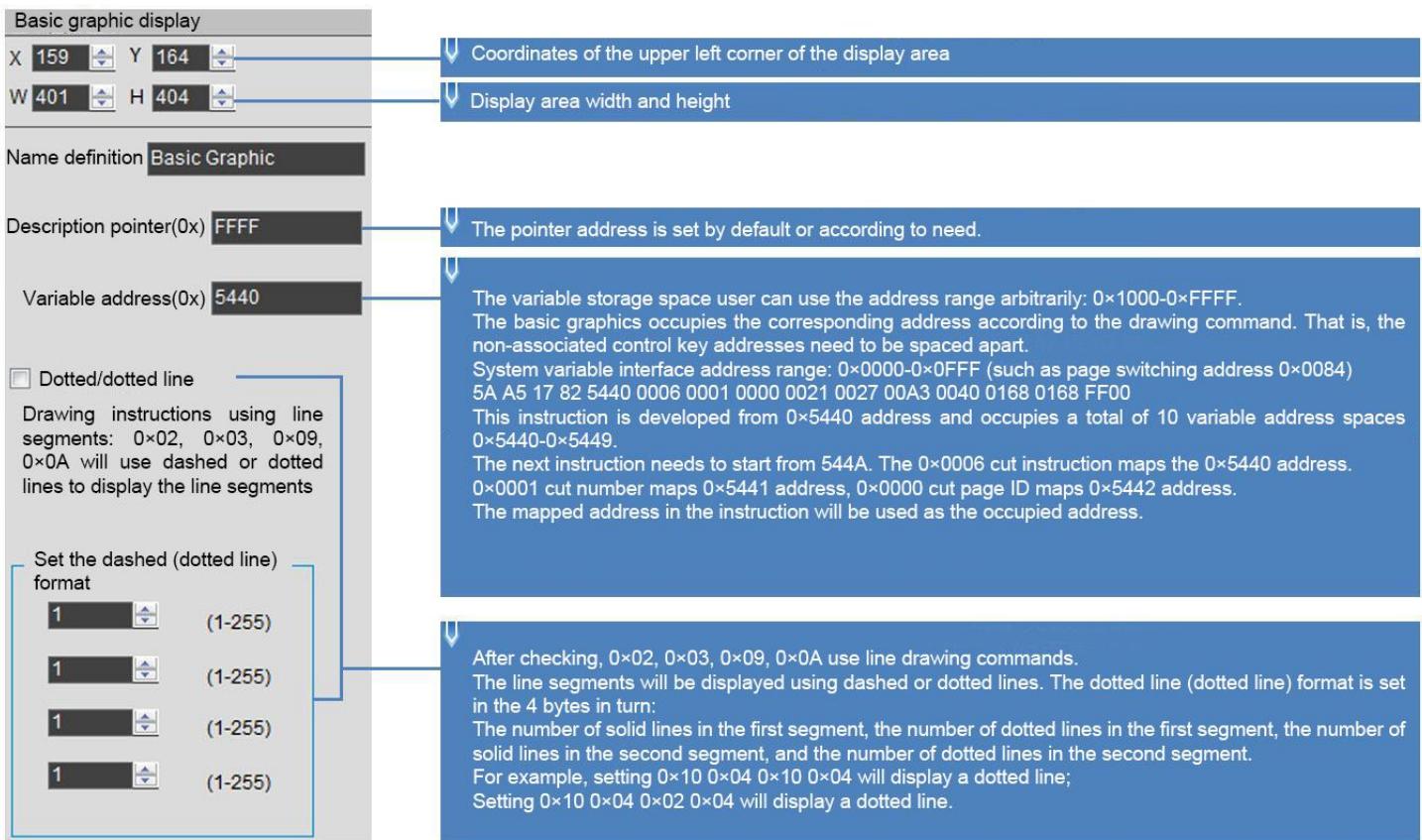
| CMD | Function | Address | Length | Description | |
|---------------|------------------------------------|----------|--------|-------------|---|
| 0x0001 | Dot | 0x00 | 2 | (x, y) | The coordinate position of dot. The high byte of coordinate x is the judgment condition. |
| | | 0x02 | 1 | Color | Dot color |
| 0x0002 | Line | 0x00 | 1 | Color | Line color |
| | | 0x01 | 2 | (x, y)0 | The coordinate of the line vertex0, and the high byte of the x coordinate are the judgment conditions. |
| | | 0x03 | 2 | (x, y)1 | The coordinate of the line vertex1, and the high byte of the x coordinate are the judgment conditions. |
| | | 0x01+2*n | 2 | (x, y)n | The coordinate of the line vertex n, and the high byte of the x coordinate are the judgment conditions. |
| 0x0003 | Rectangular | 0x00 | 2 | (x, y)s | The coordinates of the upper left corner of the rectangle and the high byte of the x coordinate are the judgment conditions. |
| | | 0x02 | 2 | (x, y)e | The coordinates of the bottom right corner of the rectangle. |
| | | 0x04 | 1 | Color | Rectangular color |
| 0x0004 | Rectangle filling | 0x00 | 2 | (x, y)s | The coordinates of the upper left corner of the rectangle and the high byte of the x coordinate are the judgment conditions. |
| | | 0x02 | 2 | (x, y)e | The coordinates of the bottom right corner of the rectangle. |
| | | 0x04 | 1 | Color | Fill color of the rectangle field. |
| 0x0005 | Circle | 0x00 | 2 | (x, y)s | Center coordinates. |
| | | 0x02 | 1 | R | The radius of the circle. |
| | | 0x03 | 1 | Color | The arc color. |
| 0x0006 | Picture Copy&Paste | 0x00 | 1 | Pic_ID | The ID of the page where the cut image area is located; the high byte is the judgment condition |
| | | 0x01 | 2 | (x, y)s | The coordinates of the upper left corner of the image cutting area. |
| | | 0x03 | 2 | (x, y)e | The coordinates of the lower right corner of the image cutting area. |
| | | 0x05 | 2 | (x, y) | The coordinates of the upper left corner of the area in the current page that it is cut and pasted to |
| 0x**07 | Icon Display | 0x00 | 2 | (x, y) | The coordinate position of display area, and the high byte of the x coordinate is the judgment condition. |
| | | 0x02 | 1 | ICON_ID | Icon ID, the location of the icon library is specified by the high byte of instruction. The icon is fixed to not display the background color. |
| 0x0008 | Closed Region Filling | 0x00 | 2 | (x, y) | The coordinates of point. |
| | | 0x02 | 1 | Color | The color of filling the rectangle field. |
| 0x0009 | Frequency Spectrum (Vertical line) | 0x00 | 1 | Color0 | Connect (x0, Y0s) (x0, Y0e) with Color0 color, and the high byte of x0 is the judgment condition. |
| | | 0x01 | 3 | X0,Y0s,Y0e | |

| | | | | | |
|---------------|--------------------------|------|---|-----------|---|
| 0x000A | Line segment display | 0x00 | 1 | color | Connect (Xs, Ys) (Xe, Ye) with Color, and the high byte of Xs is the judgment condition. |
| | | 0x00 | 1 | color | |
| | | 0x00 | 1 | color | |
| 0x000D | xOR | 0x00 | 2 | (x, y)s | The coordinates of the upper left corner of the rectangular area, and the high byte of the x coordinate are the judgment conditions. |
| | | 0x02 | 2 | (x, y)e | The coordinates of the lower right corner of the rectangular field. |
| | | 0x04 | 1 | Color | The color of xOR of the rectangular field, and 0xFFFF will be inverted. |
| 0x000E | Two-color bitmap display | 0x00 | 2 | (x, y)s | The bitmap displays the coordinates of the upper left corner of the rectangle, and the high byte of the x-coordinate is the judgment condition. |
| | | 0x02 | 1 | X_Dots | The number of dots in the x direction of the bitmap |
| | | 0x03 | 1 | Y_Dots | The number of dots in the y direction of the bitmap |
| | | 0x04 | 1 | Color1 | "1" bit corresponding display color |
| | | 0x05 | 1 | Color0 | The display color corresponding to "0" bit; if you set Color0 and Color1 to be the same, Indicates that "0" bit does not need to be displayed, just skip it. |
| | | 0x06 | N | Date_Pack | Display data, MSB mode; in order to facilitate users to read and write data, each line of data must be aligned to a word, that is, the next line of data always starts from a new data word (Word). |
| | | 0x00 | 2 | (x, y) | The coordinates of the center of the ellipse, and the high byte of the x coordinate is the judgment condition. |
| 0x0011 | Ellipse display | 0x02 | 1 | A | Long axis length. |
| | | 0x03 | 1 | B | Short axis length. |
| | | 0x04 | 1 | color | Elliptical arc color. |

The value 0xFF indicates that the drawing operation is over, and 0xFE indicates that the operation is skipped (ignored).

7.22.2 Software Setting

Open DGUS, click display control – basic graphics, then set the control display area, and configure parameters in the right menu, and finally use instructions to realize drawing.



7.22.3 Example of Instructions

The following instructions are required to cover the whole area of the page with the basic graphics box.

The following instructions should be tested on the resolution of 720*720. If the resolution is small, it is necessary to pay attention to whether the coordinate point (360,360) is within the display range, so as to avoid not being displayed beyond the range.

7.22.4 Set point

Show a red dot

5A A5 0F 82 5440 0001 0001 0168 0168 F800 FF00

0x5AA5: Frame header;

0x0D Data length;

0x82: Write instruction;

0x5440: VP;

0x0001: Draw point;

0x0001: The number of points;

0x0168:0168 (360,360) coordinates;

0xF800: Color value;

0xFF00 End of drawing operation.

Show two red dots

5A A5 15 82 5440 000100020168 0168 F8000169 0169 F800 FF00

0x5AA5: Frame header;

0x15 Data length;

0x82: Write instruction;

0x5440: VP;

0x0001: Draw point;

0x0001: The number of points;

0x0168 0168: (360,360)coordinates, 0xF800 color value;

0x0169 0169: (361,361)coordinates, 0xF800 color value;

0xFF00: End of drawing operation.

Show three red dots

5A A5 19 82 5440 000100030168 0168 F8000169 0169 F800016A 016A F800 FF00

0x5AA5: Frame header;

0x19 Data length;

0x82: Write instruction;

0x5440: VP;0x0001: Draw point;

0x0003: the number of points;

0x0168 0168: (360,360)coordinates;

0xF800 color value;

0x0169 0169: (361,361)coordinates,

0xF800 color value;

0x016A 016A: (362,362)coordinates,

0xF800 color value;

0xFF00: End of drawing operation.

7.22.5 End Point Connection

Two points are connected to form a line

5A A5 13 82 5440 00020001F80000FC 0168015E 0168 FF00

0x5AA5: Frame header;

0x13 Data length;

0x82: Write instruction;

0x5440: VP;

0x0002: Draw line segment;

0x0001: The number of connections;

0xF800: Color value;

0x00FC 0168: (252,360)coordinates;

0x015E 0168: (350,360)coordinates;

0xFF00: End of drawing operation.

Three points are connected to form lines

5A A5 17 82 5440 00020002F80000FC 0168015E 0168015E 0136 FF00

0x5AA5: Frame header;

0x17 Data length;

0x82: Write instruction;

0x5440: VP;

0x0002: Draw line segment;

0x0002: The number of connections;

0xF800: Color value;

0x00FC 0168: (252,360)coordinates;

0x015E 0168: (350,360)coordinates;

0x015E 0136: (350,310)coordinates;

0xFF00 End of drawing operation.

Four points are connected to form lines

5A A5 1B 82 5440 00020003F80000FC 0168015E 0168015E 01360190 0136 FF00

0x5AA5: Frame header;

0x1B Data length;

0x82: Write instruction;

0x5440: VP;

0x0002: Draw line segment;

0x0003: The number of connections;

0xF800: Color value;

0x00FC 0168: (252,360)coordinates;

0x015E 0168: (350,360)coordinates;

0x015E 0136: (350,310)coordinates;

0x0190 0136: (350,310)coordinates;

0xFF00: End of drawing operation.

Five points are connected to form lines

5A A5 1F 82 5440 00020004F80000FC 0168015E 0168015E 01360190 01360190 0168 FF00

0x5AA5: Frame header;

0x1F Data length;

0x82: Write instruction;

0x5440: VP;

0x0002: Draw line segment;

0x0004: The number of connections;

0xF800: Color value;

0x00FC 0168: (252,360)coordinates;

0x015E 0168: (350,360)coordinates;

0x015E 0136: (350,310)coordinates;

0x0190 0136: (350,310)coordinates;

0x0190 0168: (400,360)coordinates;

0xFF00: End of drawing operation.

Six points are connected to form lines

5A A5 23 82 5440 00020005F80000FC 0168015E 0168015E 01360190 01360190 016801C2 0168 FF00

0x5AA5: Frame header;

0x23 Data length;

0x82: Write instruction;

0x5440: VP;

0x0002: Draw line segment;

0x0005: The number of connections;

0xF800: Color value;

0x00FC 0168: (252,360)coordinates;

0x015E 0168: (350,360)coordinates;

0x015E 0136: (350,310)coordinates;

0x0190 0136: (400,310)coordinates;

0x0190 0168: (400,360)coordinates;

0x01C2 0168: (450,360)coordinates;

0xFF00: End of drawing operation.

Seven points connected to form lines

5A A5 27 82 5440 00020006F80000FC 0168015E 0168015E 01360190 01360190 016801C2 016801C2 0136

FF00

0x5AA5: Frame header;

0x27 Data length;

0x82: Write instruction;

0x5440: VP;

0x0002: Draw line segment;

0x0006: The number of connections;

0xF800: Color value;

0x00FC 0168: (252,360)coordinates;

0x015E 0168: (350,360)coordinates;

0x015E 0136: (350,310)coordinates;

0x0190 0136: (400,310)coordinates;

0x0190 0168: (400,360)coordinates;

0x01C2 0168: (450,360)coordinates;

0x01C2 0136: (450,310)coordinates;

0xFF00: End of drawing operation.

Eight points are connected to form lines

5A A5 2B 82 5440 0002 0007 F800 00FC 0168 015E 0168 015E 0136 0190 0136 0190 0168 01C2 0168 01C2

0136 01F4 0136 FF00

0x5AA5: Frame header;

0x2B Data length;

0x82: Write instruction;

0x5440: VP;

0x0002: draw a line segment;
0x0007: The number of connections;
0xF800: Color value;
0x00FC 0168: (252,360)coordinates;
0x015E 0168: (350,360)coordinates;
0x015E 0136: (350,310)coordinates;
0x0190 0136: (400,310)coordinates;
0x0190 0168: (400,360)coordinates;
0x01C2 0168: (450,360)coordinates;
0x01C2 0136: (450,310)coordinates;
0x01F4 0136: (500,310)coordinates;
0xFF00: End of drawing operation.

Nine points connected to form lines

5A A5 2F 82 5440 0002 0008 F800 00FC 0168 015E 0168 015E 0136 0190 0136 0190 0168 01C2 0168 01C2 0136 01F4

013601F4 0168 FF00

0x5AA5: Frame header;
0x2F Data length;
0x82: Write instruction;
0x5440: VP;
0x0002: Draw line segment;
0x0008: The number of connections;
0xF800: Color value;
0x00FC 0168: (252,360)coordinates;
0x015E 0168: (350,360)coordinates;
0x015E 0136: (350,310)coordinates;
0x0190 0136: (400,310)coordinates;
0x0190 0168: (400,360)coordinates;
0x01C2 0168: (450,360)coordinates;
0x01C2 0136: (450,310)coordinates;
0x01F4 0136: (500,310)coordinates;
0x01F4 0168: (500,360)coordinates;
0xFF00: End of drawing operation.

Ten points connected to form lines

5A A5 33 82 5440 0002 0009 F800 00FC 0168 015E 0168 015E 0136 0190 0136 0190 0168 01C2 0168 01C2 0136

01F4 0168 0226 0168 FF00

0x5AA5: Frame header;

0x33 Data length;

0x82: Write instruction;

0x5440: VP;

0x0002: Draw line segment;

0x0009: The number of connections;

0xF800: Color value;

0x00FC 0168: (252,360)coordinates;

0x015E 0168: (350,360)coordinates;

0x015E 0136: (350,310)coordinates;

0x0190 0136: (400,310)coordinates;

0x0190 0168: (400,360)coordinates;

0x01C2 0168: (450,360)coordinates;

0x01C2 0136: (450,310)coordinates;

0x01F4 0136: (500,310)coordinates;

0x01F4 0168: (500,360)coordinates;

0x0226 0168: (550,360)coordinates;

0xFF00: End of drawing operation.

Eleven points are connected to form lines

5A A5 37 82 5440 0002 000A F800 00FC 0168 015E 0168 015E 0136 0190 0136 0190 0168 01C2 0168 01C2

01F4 0168 0226 0168 0226 0136 FF00

0x5AA5: Frame header;

0x37 Data length;

0x82: Write instruction;

0x5440: VP;

0x0002: Draw line segment;

0x000A: The number of connections;

0xF800: Color value;

0x00FC 0168: (252,360)coordinates;

0x015E 0168: (350,360)coordinates;

0x015E 0136: (350,310)coordinates;

0x0190 0136: (400,310)coordinates;

0x0190 0168: (400,360)coordinates;

0x01C2 0168: (450,360)coordinates;

0x01C2 0136: (450,310)coordinates;

0x01F4 0136: (500,310)coordinates;

0x01F4 0168: (500,360)coordinates;

0x0226 0168: (550,360)coordinates;

0x0226 0136: (550,310)coordinates;

0xFF00: End of drawing operation.

7.22.6 Rectangle

Show a rectangle

5A A5 13 82 5440 0003 0001 011E 012C 01AA 018C F800 FF00

0x5AA5: Frame header;

0x13 Data length;

0x82: Write instruction;

0x5440: VP;

0x0003: Draw rectangle;

0x0001: draw one rectangle;

0x011E 012C: (286, 300) upper left coordinate;

0x01AA 018C: (426, 396) lower right coordinate;

0xF800: Color;

0xFF00: The drawing operation has ended.

Show two rectangles

5A A5 1D 82 5440 0003 0002 011E 012C 01AA 018C F800 01AB 018D 01E6 01E6 F800 FF00

0x5AA5: Frame header;

0x1D Data length;

0x82: Write instruction;

0x5440: VP;

0x0003: Draw rectangle;

0x0002: Draw two rectangles;

0x011E 012C: (286, 300) upper left coordinate;

0x01AA 018C: (426, 396) lower right coordinates;

0xF800: Color;

0x01AB 018D: (427, 397) upper left coordinate;

0x01E6 01E6: (486,486) lower right coordinate;

0xF800: Color;

0xFF00: The drawing operation has ended.

Show three rectangles

5A A5 27 82 5440 0003 0003 011E 012C 01AA 018C F800 01AB 018D 01E6 01E6 F800 01E6 01E6 0162 022C

F800

FF00

0x5AA5: Frame header;

0x27 Data length;

0x82: Write instruction;

0x5440: VP;

0x0003: Draw rectangle;

0x0003: Draw three rectangles;
0x011E 012C: (286, 300) upper left coordinate;
0x01AA 018C: (426,96) lower right coordinate;
0xF800: Color;
0x01AB 018D: (426,96) upper left coordinate;
0x01E6 01E6: (486,486) upper right coordinate;
0xF800: Color;
0x01E6 01E6: (486,486) upper left coordinate;
0x0162 022C: (354,556) lower right coordinate;
0xF800: Color;
FF00: The drawing operation ends.

7.22.7 Rectangle Fill

Fill a rectangle

5A A5 13 82 5440 0004 0001 011E 012C 01AA 018C F800 FF00

0x5AA5: Frame header;
0x13 Data length;
0x82: Write instruction;
0x5440: VP;
0x0004: Fill rectangle;
0x0001: Fill one rectangle;
0x011E 012C: (286, 300) upper left coordinate;
0x01AA 018C: (426, 396) lower right coordinate;
0xF800: Color;
0xFF00: The drawing operation has ended.

Fill two rectangles

5A A5 1D 82 5440 0004 0002 011E 012C 01AA 018C F800 01AB 018D 01E6 01E6 F800 FF00

0x5AA5: Frame header;
0x1D Data length;
0x82: Write instruction;
0x5440: VP;
0x0004: Fill rectangle;
0x0002: Fill two rectangles;
0x011E 012C (286,00) upper left coordinate;
0x01AA 018C (426, 96) lower right coordinate;
0xF800: Color;
0x01AB 018D (427,97) upper left coordinate;

0x01E6 01E6 (486,486) lower right coordinate;

0xF800: Color;

0xFF00: The drawing operation ends.

Fill three rectangles

5A A5 27 82 5440 0004 0003 011E 012C 01AA 018C F800 01AB 018D 01E6 01E6 F800 01E6 01E6 0221 022C

F800 FF00

0x5AA5: Frame header;

0x27 Data length;

0x82: Write instruction;

0x5440: VP;

0x0004: Fill rectangle;

0x0003: Fill three rectangles;

0x011E 012C (286,00) upper left coordinate;

0x01AA 018C (426,396) lower right coordinate;

0xF800: Color;

0x01AB 018D (427,97) upper left coordinate;

0x01E6 01E6 (486,486) lower right coordinate,

0xF800: Color;

0x01E6 01E6 (486,486) upper left coordinate;

0x0221 022C (545,556) lower right coordinate;

0xF800: Color;

0xFF00: The drawing operation has ended.

When filling, the coordinates of the lower right corner are larger than the coordinates of the upper left corner.

7.22.8 Image Copy and Paste

This instruction is frequently used and can be used for the clipping display of the progress bar.

5A A5 17 82 5440 0006 0001 0000 0021 002700A3 0040 0168 0168 FF00

0x5AA5: Frame header;

0x17 Data length;

0x82: Write instruction;

0x5440: VP;

0x0006: image copy and paste;

0x0001: Cut a region;

0x0021 0027 (33, 39): Top left coordinates of page 0,

0x00A3 0040 (163,64): Bottom right coordinates of page 0;

0x0168 0168 (360,360): Paste to the current page coordinates;

0xFF00 Drawing operation ended.

7.22.9 Icon Display

Displays icon No. 01 in the No. 48 icon library.

5A A5 0F 82 5440 3007 0001 0168 0168 0001 FF00

0x5AA5: Frame header;

0x0F Data length;

0x82: Write instruction;

0x5440: VP;

0x30: Icon library No. 48;

0x07: 07 icon Write instruction;

0x0001: display an icon;

0x0168 0x0168: The starting display coordinates of the upper left corner of the icon (360, 360);

0x0001: Icon No. 1 in the No. 48 icon library.

0xFF00: Terminator

Displays icon No. 02 from the icon library No. 48.

5A A5 0F 82 5440 3007 0001 0168 0168 0002 FF00

0x5AA5: Frame header;

0x0F Data length;

0x82: Write instruction;

0x5440: VP;

0x30: Icon library No. 48;

0x07: 07 icon Write instruction;

0x0001: Display an icon;

0x0168 0x0168: The starting display coordinates of the upper left corner of the icon (360, 360);

0x0002: Icon No. 2 in the No. 48 icon library.

0xFF00: Terminator

7.22.10 Enclosed Area Filling

Fill a circle

It is necessary to make two basic graphic display boxes and set two different variable addresses, such as one variable address is set to 0x5440, the other is set to 0x5540, and the background image should be solid color, otherwise it cannot be filled normally.

Step 1: The 0x5440 address displays a circle:

5A A5 11 82 5440 0005 0001 0168 0168 0040 F800 FF00

0x5AA5: Frame header;

0x11 Data length;

0x82: Write instruction;

0x5440: VP;

0x0005 Draw circle;

0x0001 Draws one circle;

0x0168, 0x0168: Circular coordinates (360,360);

0x0040 radius, in pixels;

0xF800 red;

0xFF00 terminator.

Step 2: 5A A5 0F 82 5540 0008 0001 0168 0168 F800 FF00

7.22.11 Spectrum Display

Show a spectrum

5A A5 11 82 5440 0009 0001 F800 0168 0168 0190 FF00

0x5AA5: Frame header;

0x11 Data length;

0x82: Write instruction;

0x5440: VP;

0x0009 Spectrum display;

0x0001 shows a spectrum;

0xF800: Color;

0x0168x0 coordinates,

0x0168 Y0s start coordinate;

0x0190 Y0e end coordinate;

0xFF00 Drawing operation ended.

5A A5 19 82 5440 0009 0002 F800 0168 0168 0190 F800 0190 0168 01C2 FF00

0x5AA5: Frame header;

0x11 Data length;

0x82: Write instruction;

0x5440: VP;

0x0009: Spectrum display;

0x0002: Shows two spectrums;

0xF800: Color; 0x0168 (360) x0 coordinates,

0x0168 (360) Y0s start coordinates;

0x0190 (400) Y0e end coordinates;

0xF800: Color; 0x0190 (400) x1 coordinates;

0x0168 (360) Y1s start coordinates;

0x01C2 (450) Y1e end coordinates;

0xFF00 Drawing operation ended.

Display three spectrums

5A A5 21 82 5440 0009 0003 F800 0168 0168 0190 F800 0190 0168 01C2 F800 01C2 0168 01F4 FF00

0x5AA5: Frame header;

0x11 Data length;

0x82: Write instruction;

0x5440: VP;

0x0009: Spectrum display;

0x0003: Shows three spectrums;

0xF800: Color;

0x0168 (360) x0 coordinates,

0x0168 (360) Y0s start coordinates;

0x0190 (400) Y0e end coordinates;

0xF800: Color;

0x0190 (400) x1 coordinates;

0x0168 (360) Y1s start coordinates;

0x01C2 (450) Y1e end coordinates;

0xF800: Color;

0x01C2 (450) x2 coordinates;

0x0168 (360) Y2s start coordinates;

0x01F4 (500) Y2e end coordinates;

0xFF00 Drawing operation ended.

Displays four spectrums

5A A5 29 82 5440 0009 0004 F800 0168 0168 0190 F800 0190 0168 01C2 F800 01C2 0168 01F4 F800 01F4 0168
020D FF00
5A A5 Frame header;
0x11 Data length;
0x82: Write instruction;
0x5440: VP;
0x0009 spectrum display;
0x0004 displays four spectrums;
0xF800: Color;
0x0168 (360) x0 coordinates,
0x0168 (360) Y0s start coordinates;
0x0190 (400) Y0e end coordinates;
0xF800: Color;
0x0190 (400) x1 coordinates;
0x0168 (360) Y1s start coordinates;
0x01C2 (450) Y1e end coordinates;
0xF800: Color;
0x01C2 (450) x2 coordinates;
0x0168 (360) Y2s start coordinates;
0x01F4 (500) Y2e end coordinates;
0xF800: Color;
0x01F4 (500) x3 coordinates;
0x0168 (360) Y3s start coordinates;
0x020D (525) Y3e end coordinates;
0xFF00 Drawing operation ended.

Display five spectrums

5A A5 31 82 5440 0009 0005 F800 0168 0168 0190 F800 0190 0168 01C2 F800 01C2 0168 01F4 F800 01F4 0168
020D
F800 0226 0168 0226 FF00
0x5AA5: Frame header;
0x11 Data length;
0x82: Write instruction;
0x5440: VP;
0x0009: Spectrum display;
0x0005: Displays five spectrums;
0xF800: Color;

0x0168 (360) x0 coordinates;
0x0168 (360) Y0s start coordinates;
0x0190 (400) Y0e end coordinates;
0xF800: Color;
0x0190 (400) x1 coordinates;
0x0168 (360) Y1s start coordinates;
0x01C2 (450) Y1e end coordinates;
0xF800: Color;
0x01C2 (450) x2 coordinates;
0x0168 (360) Y2s start coordinates;
0x01F4 (500) Y2e end coordinates;
0xF800: Color;
0x01F4 (500) x3 coordinates;
0x0168 (360) Y3s start coordinates;
0x020D (525) Y3e end coordinates;
0xF800: Color;
0x0226 (550) x4 coordinates;
0x0168 (360) Y4s start coordinates;
0x0226 (550) Y4e end coordinates;
0xFF00 End drawing operation
Y0s Y0e settings cannot be the same.

7.22.12 Line Segment Display

Show a line segment

5A A5 13 82 5440 000A 0001 F800 0168 0168 0190 0168 FF00

0x5AA5: Frame header;
0x11 Data length;
0x82: Write instruction;
0x5440: VP;
0x000A Draw line segment;
0x0001 Draws a line segment;
0xF800: Color;
0x0168 0168 (360,360) coordinates;
0x0190 0168 (400,360) coordinates;
0xFF00 End drawing operation

Show two line segments

5A A5 1D 82 5440 000A 0002 F800 0168 0168 0190 0168 F800 0168 0190 01C2 0190 FF00

0x5AA5: Frame header;

0x11 Data length;

0x82: Write instruction;

0x5440: VP;

0x000A Draw line segment;

0x0002 Draws two line segments;

0xF800: Color;

0x0168 0168 (360,360) coordinates;

0x0190 0168 (400,360) coordinates;

0xF800: Color;

0x0168 0190 (360,400) coordinates;

0x01C2 0190 (450,400) coordinates;

0xFF00 End drawing operation

Show three line segments

5A A5 27 82 5440 000A 0003 F800 0168 0168 0190 0168 F800 0168 0190 01C2 0190 F800 0168 01C2 01F4

01C2 FF00

0x5AA5: Frame header;

0x11 Data length;

0x82: Write instruction;

0x5440: VP;

0x000A Draw line segment;

0x0003 Draw three line segments;

0xF800: Color;

0x0168 0168 (360,360) coordinates;

0x0190 0168 (400,360) coordinates;

0xF800: Color;

0x0168 0190 (360,400) coordinates;

0x01C2 0190 (450,400) coordinates;

0xF800: Color;

0x0168 01C2 (360,450) coordinates;

0x01F4 01C2 (500,450) coordinates;

0xFF00 End drawing operation

Show four line segments

5A A5 31 82 5440 000A 0004 F800 0168 0168 0190 0168 F800 0168 0190 01C2 0190 F800 0168 01C2 01F4

01C2 F800

0168 01F4 0226 01F4 FF00

0x5AA5: Frame header;

0x11 Data length;

0x82: Write instruction;

0x5440: VP;

0x000A Draw line segment;

0x0004 Draw four line segments;

0xF800: Color;

0x0168 0168 (360,360) coordinates;

0x0190 0168 (400,360) coordinates;

0xF800: Color;

0x0168 0190 (360,400) coordinates;

0x01C2 0190 (450,400) coordinates;

0xF800: Color;

0x0168 01C2 (360,450) coordinates;

0x01F4 01C2 (500,450) coordinates;

0xF800: Color;

0x0168 01F4 (360,500) coordinates;

0x0226 01F4 (550,500) coordinates;

0xFF00 End drawing operation

Step 2: Based on the 0x5540 address, fill the circular enclosed area with the center of the circle as the seed point:

5A A5 0F 82 5540 0008 0001 0168 0168 F800 FF00

0x5AA5: Frame header;

0x0F Data length;

0x82: Write instruction;

0x5440: VP;

0x0008 Fill instruction;

0x0001 Fill the enclosed area;

0x0168, 0x0168 The coordinates of the seed point of the circular enclosed area (360,360);

0xF800 Red;

0xFF00 Terminator.

7.22.13 Rectangular Field XOR Highlighting Inverse Color Display

Rectangular field XOR, highlight and inverse color display once

5A A5 13 82 5440 000D 0001 0168 0168 0190 0190 F800 FF00

0x5AA5: Frame header;

0x13 Data length;

0x82: Write instruction;

0x5440: VP;

0x000D Rectangle XOR;

0x0001 XOR the rectangular field once;

0x0168 0168 (360,360) coordinates;

0x0190 0190 (400,400) coordinates;

0xF800: Color;

0xFF00 Drawing operation ended.

Rectangular field XOR, highlight and invert twice

5A A5 1D 82 5440 000D 0002 0168 0168 0190 0190 F800 0190 0190 01C2 01C2 F800 FF00

0x5AA5: Frame header;

0x1D Data length;

0x82: Write instruction;

0x5440: VP;

0x000D Rectangle XOR;

0x0002Rectangular XOR twice;

0x0168 0168 (360, 360) coordinates;

0x0190 0190 (400, 400) coordinates;

0xF800 colors;

0x0190 0190 (400, 400) coordinates;

0x01C2 01C2 (450, 450) coordinates;

0xF800 colors;

0xFF00 Drawing operation ended.

Rectangular field XOR, highlight and inverse color display three times

5A A5 27 82 5440 000D 0003 0168 0168 0190 0190 F800 0190 0190 01C2 01C2 F800 01C2 01C2 01F4 01F4

F800 FF00

0x5AA5: Frame header;

0x1D Data length;

0x82: Write instruction;

0x9010 VP;

0x000D rectangle XOR;

0x0003 rectangular field XOR three times;

0x0168 0168 (360, 360) coordinates;

0x0190 0190 (400, 400) coordinates;

0xF800 colors;

0x0190 0190 (400, 400) coordinates;

0x01C2 01C2 (450, 450) coordinates;

0xF800 colors;

0x01C2 01C2 (450, 450) coordinates;

0x01F4 01F4 (500, 500) coordinates;

0xF800 colors;

0xFF00 Drawing operation ended.

7.22.14 Two-Color Bitmap Display

(1) Function

Two-color bitmap can be used for special characters or graphics display.

Generally, special characters or graphics usually use icon or the background image ,here mainly for your understanding of the function.

(2) Explanation

(1) X, Y: the bitmap displays the coordinates of the upper left corner of the rectangular field. The height of the X coordinate is the judgment condition, and the starting position of the character area is displayed.

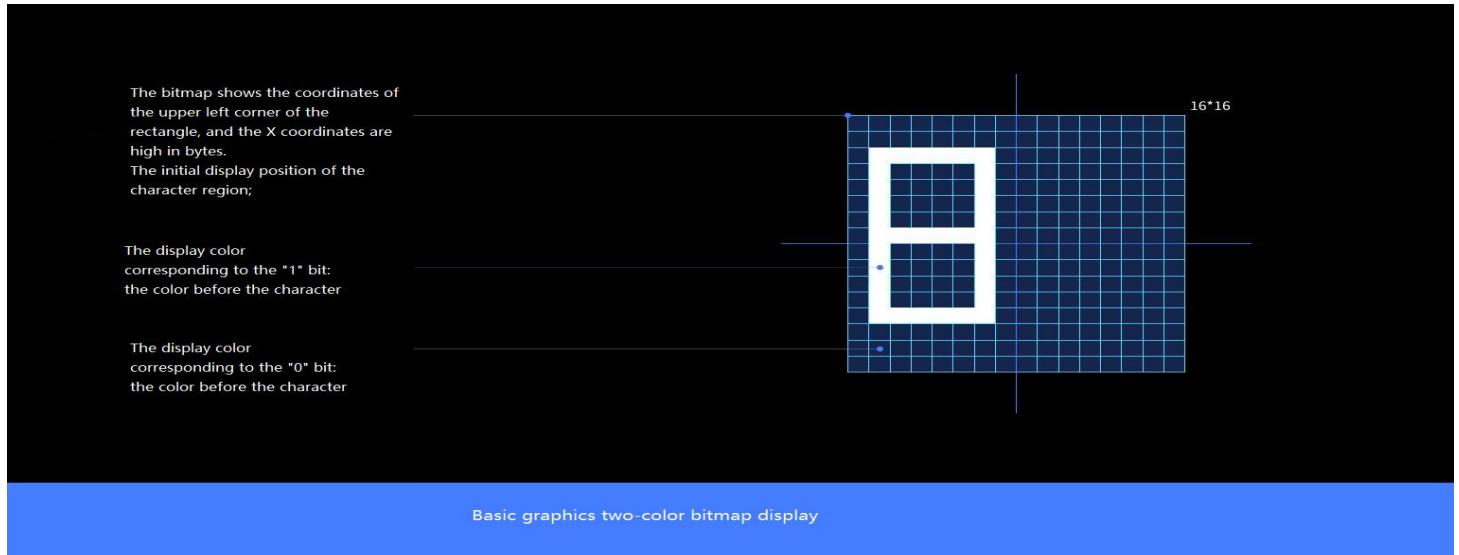
(2) X_Dots, Y_Dots: the number of lattice arrays in X and Y directions;

(3) Display color corresponding to "1" bit: foreground color;

(4) Display color corresponding to "0" bit: background color;

(5) Display data: directly use modulus data.

| | | | |
|--|------|-----------|--|
| 0x00E Tw o-color bitmap display | 0x00 | (x, y)s | The bitmap shows the coordinates of the upper left corner of the rectangular field. The height of the x coordinate is byte as the judgment condition. |
| | 0x02 | X_Dots | Number of bitmap X- direction lattice |
| | 0x03 | Y_Dots | Number of bitmap Y- direction lattice |
| | 0x04 | Color1 | Display color corresponding to "1" bit |
| | 0x05 | Color0 | The display color corresponding to "0" bit; If Color0 is set to the same as Color1, "0" bit No |
| | 0x06 | Date_Pack | Display data, MSB mode; To make it easy for users to read and write data, each line of data must be aligned to a Word, that is, the next line of data always starts with a new data Word (Word). |



Bitmap Character Description

(3) the character "8" dot matrix module program

Website: <https://www.zhetao.com/fontarray.html>

(4) Instruction examples

① Display character graphic "8" in the range of 15*15 dot matrix

Tx: 5A A5 35 82 5440 000E 0001 0168 0168 000F 000F F800 07E0 0000 0000 0000 3C00 4200 4200 4200 2400

1800 2400

4200 4200 4200 3C00 0000 0000 FF00

0x5AA5: Frame header;

0x35: Data length 53 bytes.

0x82: Write instruction;

0x5440: VP;

0x000E: Two-color bitmap instruction;

0x0001: Write a character;

0x0168, 0x0168: The upper left coordinates of the character;

0x000F, 0x000F: The number of dots in the character display area;

0xF800: Color1 character color;

0x07E0: Color0 character background color;

0x0000 0000 0000 3C00 4200 4200 2400 1800 2400 4200 4200 4200 4200 4200 3C00 0000 0000:

0xFF00: End character.

② Display character graphic "°C" within the range of 15*15 dot matrix

Tx: 5A A5 35 82 5440 000E 0001 0168 0168 000F 000F F800 07E0 6000 91F4 960C 6C04 0804 1800 1800 1800

1800 1800 1800 1800 1800 1800 0800 0C04 0608 01F0 0000 FF00

7.22.15 Ellipse Display

Ellipse display + fill (the ellipse function needs to use the UI kernel of V41_Increase Ellipse Display_20210802).
For ellipse display example, set VP of basic graphic display control to 0x5440. It is the same for the ellipse filling.
Filling closed area requires the background image to be a solid color, otherwise the filling will display abnormally.

Step 1: Display an Ellipse

5A A5 13 82 5440 0011 0001 0168 0168 0032 0016 F800 FF00

0x5AA5: Frame header;

0x13: Data length;

0x82: Write instruction;

0x5440: VP;

0x0011: Draw ellipse instruction;

0x0001: Draw an ellipse;

0x0168 0168: The coordinates of the ellipse center;

0x0032: The diameter of the major axis of the ellipse;

0x0016: The diameter of the minor axis of the ellipse;

0xF800: Arc color;

0xFF00: Terminator.

Step 2: Ellipse Fill

5A A5 0F 82 5540 0008 0001 0168 0168 07FF FF00

0x5AA5: Frame header;

0x0F: Data length;

0x82: Write instruction;

0x5440: VP;

0x0008: Fill instruction;

0x0001: Fill the closed area;

0x0168 0168: The coordinates of the ellipse center;

0xF800: Fill color;

0xFF00: Terminator.

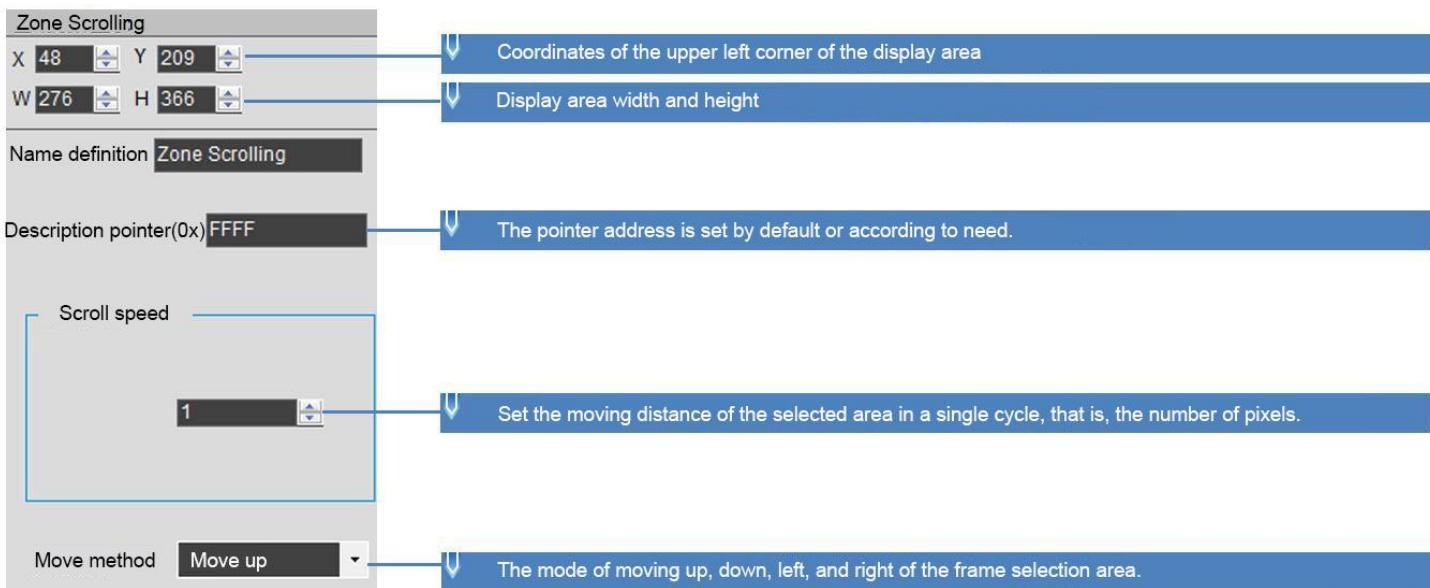
7.23 Zone Scrolling

The zone scrolling control is to move the contents of the specified area around, and the moving direction can be set. It can be used to simply realize dynamic running effects such as flow charts and progress bars on the screen. The VP is handled by the underlying system, and the user does not need to set it.

7.23.1 Instruction Storage Format

| SP offset | Definition | Length (bytes) | Description |
|-----------|------------|----------------|--|
| 0x00 | VP | 2 | A word variable to hold the panning data. Not available to the user. |
| 0x01 | (x, y) s | 4 | The coordinates of the upper left corner of the moving area. |
| 0x02 | (x, y) e | 2 | The coordinates of the upper right corner of the moving area. |
| 0x05 | Dis_Move | 2 | Move speed. |
| 0x06_H | Mode_Move | 2 | Shift left 0x01=Shift right 0x02=Shift up 0x03=Shift down. |

7.23.2 Software Setting



The variable is occupied by the system and should not be used by the user.

7.24 QR Code

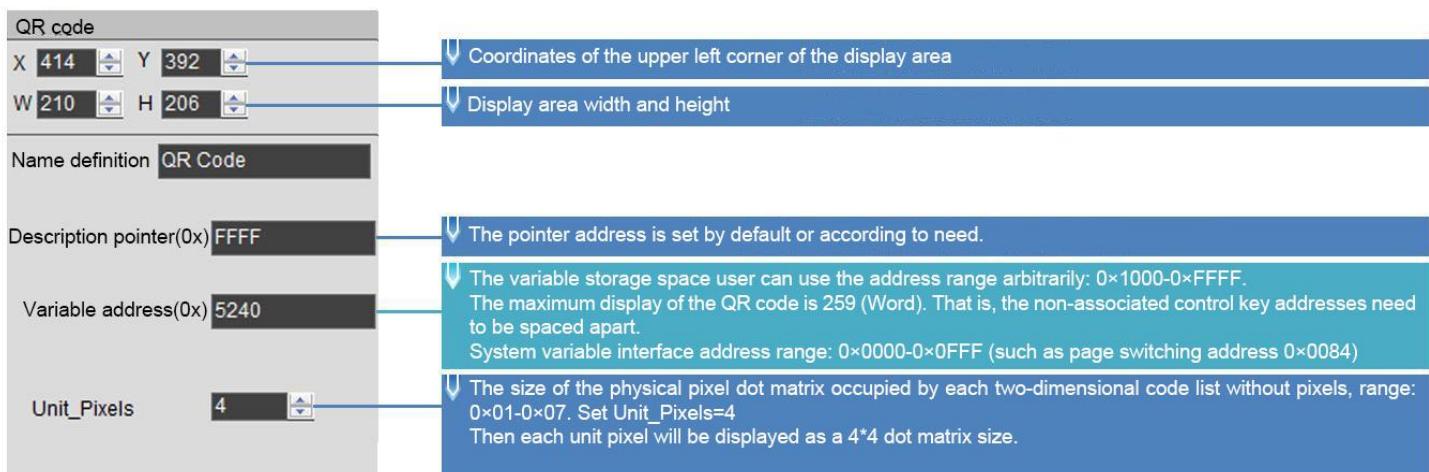
The QR code control is to display the designated QR code on the screen according to the designated content.

7.24.1 Instruction Storage Format

| SP offset | Definition | Length (bytes) | Description |
|-----------|-------------|----------------|--|
| 0x00 | VP | 2 | Variable pointer. The content of the QR code is up to 458Bytes, and 0x0000 or 0xFFFF is the terminator. |
| 0x01 | (x, y) | 4 | The coordinate position of the upper left corner displayed by the QR code. There are two types of QR code graphics: 45*45 unit pixels (data less than 155 bytes) and 73*73 unit pixels (data less than 459Bytes). |
| 0x03 | Unit_Pixels | 2 | The physical pixel lattice size occupied by each QR code unit pixel, 0x01-0x07. Set Unit_Pixels=4, then each unit pixel will be displayed as 4*4 lattice size. |
| 0x05:H | Fix_Mode | 1 | 0x01: Fixed as 73*73 unit pixel format QR code. The rest: automatically match the QR code size according to the Data length. |
| | Reserved | 1 | Undefined, write 0x00. |

7.24.2 Software Setting

Open DGUS, click display control – QR code, then set the control display area, and configure VP, the physical pixel size occupied by each QR code unit pixel, etc.



7.24.3 Example of Instructions

Send an instruction to display the URL <http://www.dwin.com.cn/> QR code.

5A A5 1C 82 5240 68 74 74 70 3A 2F 2F 77 77 77 2E 64 77 69 6E 2E 63 6F 6D 2E 63 6E 2F FFFF

0x5AA5: Frame header;

0x1C: Data length;

0x82: Write instruction;

0x5240: VP;

0x68 74 74 70 3A 2F 2F 77 77 77 2E 64 77 69 6E 2E 63 6F 6D 2E 63 6E 2F : the ASCII code of the URL;

0xFFFF: Terminator.

If the size of the QR code needs to be fixed, You can set SP to 0x8000, offset 5 bits from SP.

Then write 0x01 to 0.8005 to fix the size of QR code.

7.25 Brightness

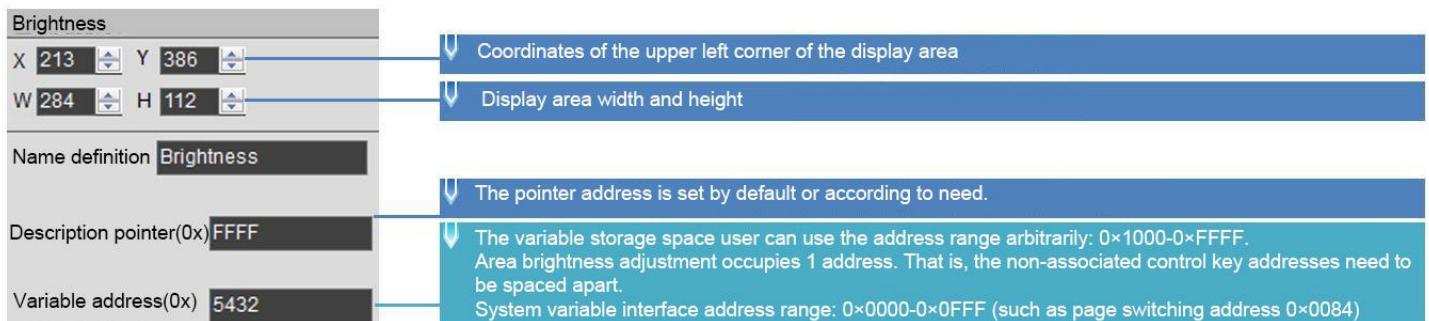
The brightness control is used to adjust the display brightness of the specified display area, and is used to highlight or fade the background display.

The difference between this control and the backlight adjustment system variable interface is that the backlight is to adjust the brightness of the whole screen.

7.25.1 Instruction Storage Format

| SP Offset | Definition | Length (bytes) | Description |
|-----------|------------|----------------|--|
| 0x00 | *VP | 2 | Upper-left coordinates of the Icons to display. |
| 0x01 | (Xs, Ys) | 4 | Specifies the coordinates of the upper left corner of the area. |
| 0x03 | (Xe, Ye) | 4 | Specifies the coordinates of the upper right corner of the area. |
| | Reserved | 18 | Undefined, write 0x00 . |

7.25.2 Software Setting



7.25.3 Examples of Instruction

5A A5 05 82 5432 0032

0x5AA5: Frame header;

0x05: Data length;

0x82: Write instruction;

0x5432 VP;

0x0032 Brightness value. (Range 0x00-0x64, 100-level brightness adjustment)

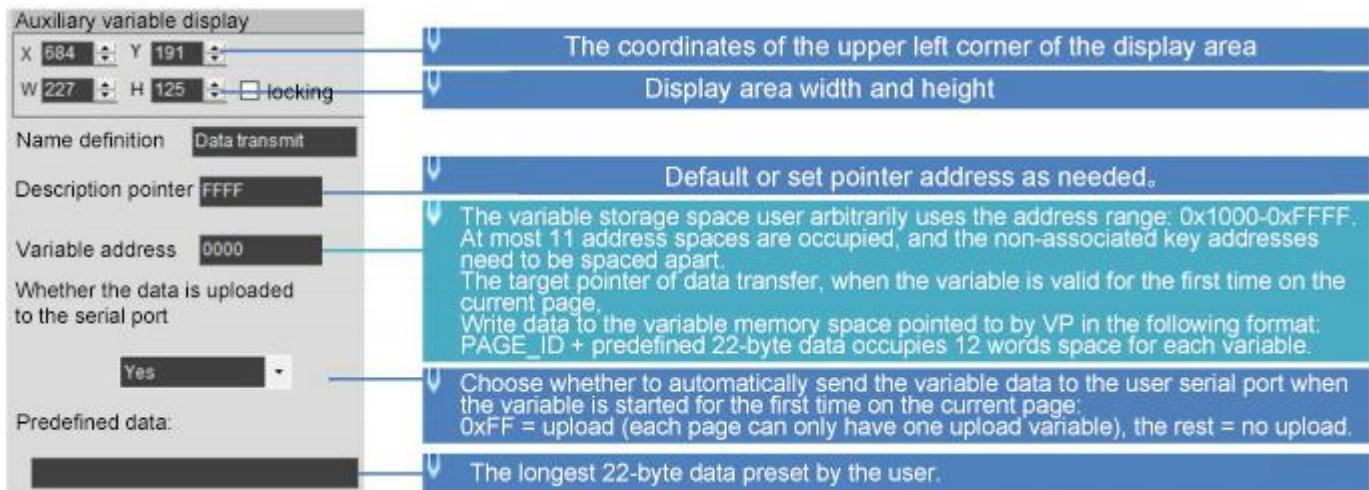
7.26 Data Transmit

After the page is switched, the predefined data is transferred to the variable or UART once.

7.26.1 Instruction Storage Format

| Address | SP offset | Definition | Data length (byte) | Description |
|-------------|---------------|-----------------|--------------------|---|
| 0x00 | | 0x5A30 | 2 | |
| 0x02 | | *SP | 2 | Variable description pointer, 0xFFFF means loaded by the configuration file. |
| 0x04 | | 0x000D | 2 | |
| 0x06 | 0x00 | *VP | 2 | The target pointer for data transfer. When the variable is valid for the first time on the current page, the data is written to the pointer pointed to by VP according to the following format. Variable memory space: PAGE_ID + predefined 22 bytes data Each variable occupies 12 words of space. |
| 0x08 | 0x01:H | AUTO_COM _En | 1 | Select whether to actively upload the variable data to the user when the variable is first started on the current page Serial port: 0xFF = upload (each page can only have 1 upload variable), the rest = no upload. |
| 0x09 | 0x01:L | Reserved | 1 | Write 0x00 |
| 0x0A | 0x06 | Predefined data | 22 | The max length of data preset by the user is 22 bytes. |

7.26.2 Software Setting



7.27 Camera

The camera control is only available for DGUS platform T5L2 products. The standard products need to be customized to use this control.

This control uses DWIN two-wire power carrier camera, which can be connected to the FSK carrier bus interface of DGUS screen (the connection cable length should not exceed 300 meters).

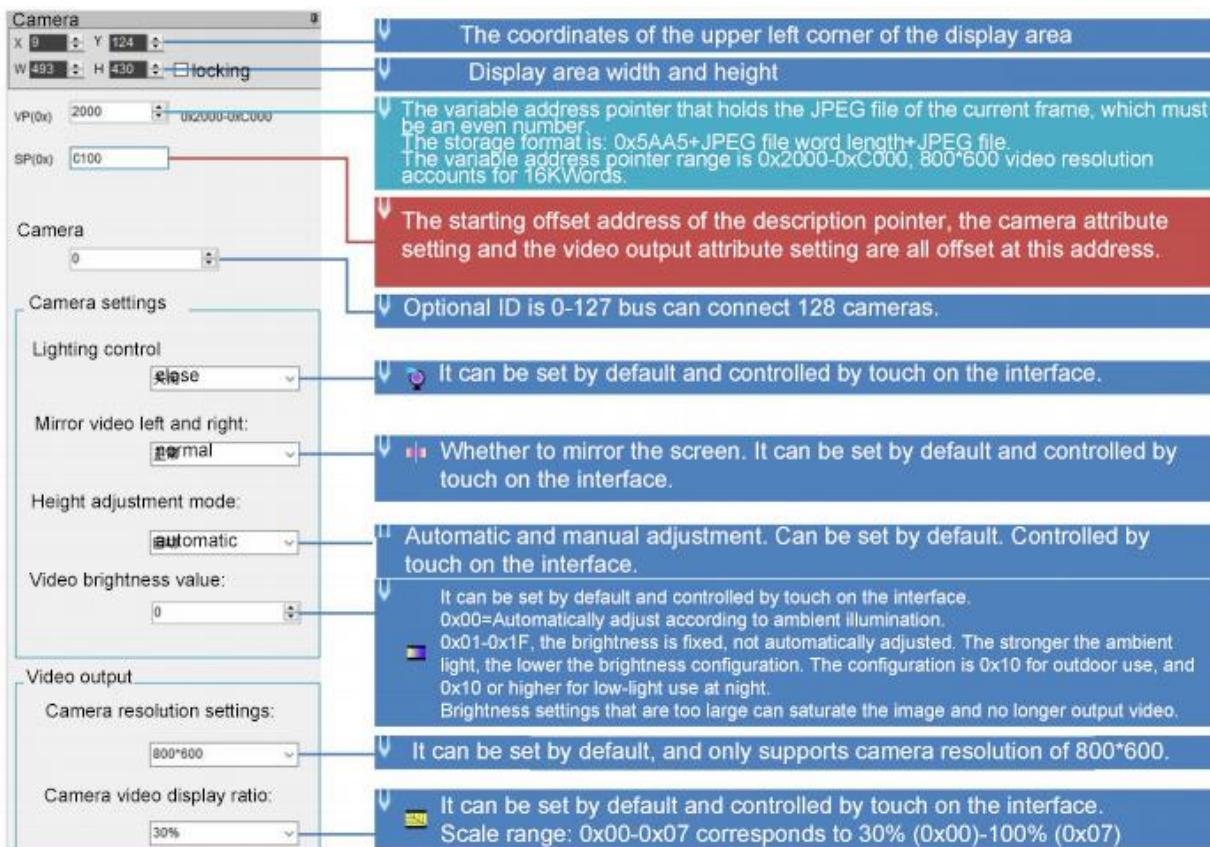
It support FSK bus connection for 128 cameras, and functions of camera lighting control, left and right mirroring of video images, video screen freeze and pause, video screen brightness adjustment, video screen ratio setting.

7.27.1 Instruction Storage Format

| Address | SP offset | Definition | Data length (byte) | Description |
|---------|-----------|------------|--------------------|--|
| 0x00 | | 0x5A31 | 2 | |
| 0x02 | | *SP | 2 | Parameter pointer. 0xFFFF means loaded by the configuration file. |
| 0x04 | | 0x000A | 2 | |
| 0x06 | 0x00 | *VP_JPEG | 2 | <p>VP that saves the JPEG file of the current frame of the video, which must be an even number.</p> <p>The storage format:</p> <p>0x5AA5+JPEG file word length+JPEG file .</p> <p>VP range 0x2000-0xC000.</p> <p>800*600 video resolution accounts for a maximum of 16KWords.</p> |
| 0x08 | 0x01 | (x, y)s | 4 | The coordinates of the upper left corner of the video display window on the screen. |
| 0x0C | 0x03 | Wide | 1 | The width of the video display window on the screen. |
| 0x0E | 0x04 | Height | 2 | The height of the video display window on the screen. |
| 0x10 | 0x05 | Reserved | 4 | Fixed write 0xFFFF |
| 0x14 | 0x07_H | Display_EN | 1 | Camera display control, 0x5A=on, rest=off. |
| 0x15 | 0x07_L | Camera_ID | 1 | Camera selection, 0x00-0x7F corresponds to a maximum of 128 cameras on the bus. |
| 0x16 | 0x08 | Camera_Set | 4 | <p>0x08_H, camera working mode configuration:</p> <p>.7 Lighting control, 0=off, 1=on.</p> <p>.6 Video left and right mirror control, 0=normal, 1=mirror.</p> <p>.5 Video Freeze 0=Normal, 1=Freeze.</p> <p>When frozen, the current JPEG data in the variable memory pointed to by *VP_JPEG is no longer updated.</p> <p>.4-.0 Reserved, write 0.</p> <p>0x08_L, video brightness adjustment:</p> <p>0x00=Automatically adjust according to ambient illumination.</p> |

| | | | | |
|------|------|------------|---|--|
| | | | | 0x01-0x1F, the brightness is fixed, not automatically adjusted. The stronger the ambient light, the lower the brightness configuration. The configuration is 0x01 for outdoor use, and 0x10 or above for low-light use at night. Brightness settings that are too large can saturate the image and no longer output video. 0x09_H, camera video resolution setting: 0x00=800*600 (the output JPEG file size should not exceed 32KB). 0x09_L, camera video display ratio setting: 0x00-0x07 corresponds to 30% (0x00)-100% (0x07). |
| 0x16 | 0x08 | Camera_Set | 4 | Write 0x00. |

7.27.2 Software Setting



7.27.3 Example of Instructions

The camera and video output properties can be changed by instructions. Usually, the return key code control + the incremental adjustment control + the bit icon control are used to set the variable.

You can also use the return key code control, the sliding adjustment control, the pop-up menu control and the incremental adjustment control to adjust the high and low bytes of VP.

The use of SP is described in detail in 7.12 Data Variables.

The following instructions take 0xC100 as SP.

Video tutorial: http://inforum.dwin.com.cn:20080/forum.php?mod=viewthread&tid=5252&_dsign=f77d4401

① Enable video screen mirror display: 5A A5 05 82 C108 4000

0xC108: Indicate that the camera data attribute is changed by 8 addresses offset from the 0xC100 SP;

0x4000: Indicate to enable the mirror display, which is converted from binary 0100 0000 0000 0000. Write 1 to the E-th bit of the 0xC108 address to enable the mirror display, and write 0 for normal display.

② Open screen freeze pause: 5A A5 05 82 C108 2000

0xC108: Indicate that the camera data attribute is changed by 8 addresses offset from the 0xC100 SP;

0x2000: Indicate that the video screen freezes and pauses, which is converted from binary 0010 0000 0000 0000. Write 1 to the D-th bit of the 0xC108 address to enable freezing, and write 0 for normal display.

③ Video screen display ratio 100%: 5A A5 05 82 C109 0007

0xC109: Indicate that the camera video attribute is changed by 9 addresses offset from the 0xC100 SP;

0x0007: Indicate that the video aspect is displayed at 100%, and 0x00-0x07 corresponds to 30% (0x00)-100% (0x07).

④ Video screen display ratio 30%: 5A A5 05 82 C109 0000

0xC109: Indicate that the camera video attribute is changed by 9 addresses offset from the 0xC100 SP;

0x0000: Indicate that the video aspect is displayed at 30%, and 0x00-0x07 corresponds to 30% (0x00)-100% (0x07).

7.27.4 Effect

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

The supporting camera module finished products have included OV2640 camera and die sinking.

FDV200-1: no infrared.



FDV200-1

FDV200-2: with infrared.

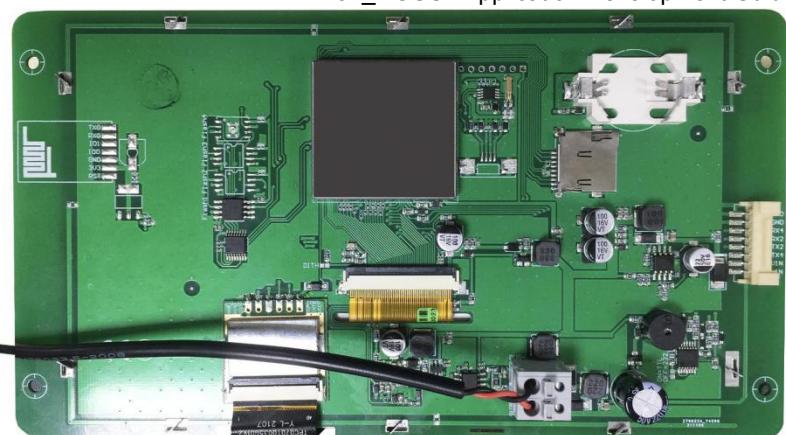


FDV200-2

The camera in the following figures is FDV200-1.



Front



Back

Chapter 8 DGUS Basic Functions

8.1 Import

If you only have a DWIN_SET folder, you can restore the project using the import function. Export is guide out of the project 13.bin, 14.bin pointer address EXCEL table.

Step 1: Open DGUS and create a project. The resolution should be consistent with the picture.

Step 2: Add all the images in DWIN_SET to the new project.

Step 3: Select the original DWIN_SET folder, click import, you can see the project control has been restored.

Step 4: Import successful, the original DWIN_SET inside the icon library, character library and other files are copied to the new project DWIN_SET. You just need to adjust the control position.

8.2 Edit tool

User can easily align the touch and display controls by this function, making the controls neat, beautiful and orderly.

Step 1: Selects the control s that need to be aligned;

Step 2: Click Edit tool-Up;

Step 3: Click the standard control, and then other control s will be aligned with it.

Left-aligned, right-aligned, same size of controls can be set in the same way.

8.3 Front and Back

When the page sliding control and sliding adjustment control are superimposed, the former will be invalid if the former is under the latter. In this case, select the page sliding control and click Edit tool-front.

It is mostly used to add variable overlay and other functions to the icon.

8.4 Viewing the Usage of VP

In the process of development, it is necessary to check whether the VP or SP of the current project conflicts. Click Display-Global check.

8.5 Preview

During the development, you can click Display_Preview from the current page to preview from the current position to check whether the project is set correctly.

Double-click the black border of the preview to exit.

8.6 New Resolution

When creating a new project, manually input the required resolution (W*H) value.

8.7 Converting Resolution

For example, change the resolution of 720*720 to 480*480. In order to avoid the conversion damage of the original project, please make a copy of the original project as a backup.

Step 1: Click Setting_Set resolution, select the desired resolution, and click OK;

Step 2: Click File_Save As, select "Yes";

Step 3: Then click File_Save and Generate.

If the user copies controls of the larger resolution project to the smaller resolution project, you may find that the control disappears.

In this case, press Ctrl+A to select all, and press "↑" or "←" on the keyboard. Then you will see the corresponding controls.

If icon or text display control are used, the size of them generally needs to be re-adjusted.

8.8 Replace T5 project with T5L project

If the resolution is the same, you can open the T5 DEMO by DGUS, make the background image 32_background image.ICL file, and make the icon 48_Icon .ICL file.

That is, use 32.ICL and 48.ICL to replace the original .ICO and BMP format images. This part is detailed in chapter 3.

Since the DGUSI V5.04 13/14 files are not compatible with DGUS V7.6, the DGUSI screen cannot be modified directly. If you need to use T5L, you need to redo the DEMO.

Chapter 9 Revision Record

| Rev | Revise Date | Content | Editor |
|-----|-------------|--|--------|
| 2.0 | 2022-04-12 | First Edition | FKX |
| 2.1 | 2022-05-09 | Update 7.18.4 | FKX |
| 2.2 | 2022-05-12 | Update 7.27.4 | FKX |
| 2.3 | 2022-05-20 | Update instructions when CRC check is on | FKX |

Appendix1:FAQs of T5L_DGUSII Screen

1. Only one picture is displayed as a black screen.

A: For the T5L1 platform, the size of a single JPG image in the ICL format file package should not exceed 256Kbytes. On the T5L2 platform, a single JPG image cannot exceed 768Kbytes. Due to the inconsistent display complexity of images such as image gradients, especially with higher resolutions, the final compressed .JPG size may exceed 256KB, and users need to slightly lower the JPG image quality percentage. You can use the latest PC version tool software to convert, there will be a prompt when the size exceeds.

If you have any questions during the use of this document or DWIN products, or want to know more about the latest information of DWIN products, please contact us.

Hotline: 400 018 9008

Corporate QQ and WeChat: 400 018 9008

DWIN website: <https://www.dwin-global.com/>

Thank you for your continuous support to DWIN, your support is the driving force for our progress!