DGUS Development Guide

Beijing Devine Technology Co., Ltd.

July 21, 2016

table of Contents

Symbols and terminology explained

display method	Explanation	
0x with H	Plus before the number 0x After adding or digital H Hexadecimal number, such as 0x10 = 10H = 16	
Variable address	RAM First address space of a certain space, the storage space for the variable.	
Description Pointer	RAM First address space of a certain space, the space for storing the value of the variable property description.	
High / low byte	All commands and data are serial 16 Hexadecimal format for fonts (2 Bytes) of data is always transmitted byte first (MSB)the way. such as 0x1234 When the first transfer transfer 0x12 , which is 0x12 High byte, 0x34 Low byte.	
System Configuration Register	Especially for the screen configuration of the system register, by writing CONFIG.txt Configuration file to achieve. its , Each register in capital letters R Plus the beginning of hexadecimal digits to represent, such as R2 , RC Wait.	
Variable storage	RAM Space, and storing variable address pointer to the data described. Stored data is not saved when power.	
Font space	Store configuration files, font files, icon libraries.	Are FLASH Part of the storage space, the data is not lost when power-down store.
Image Hosting	Storage interface picture.	
Database space	Store user database.	
Register space	Especially read by the register space communication, wherein each register address are represented by hexadecimal digits, such as 0x01 , 0x4F Wait.	

The first chapter Quick Start

1.1 Connection with the power on

Confirmation screen voltage and power consumption, power supply by the switch to illuminate the screen. Switching power supply to the normal screen display a very important role, the voltage is too small, the current instability, power is too low may result in abnormal status display a black screen, splash screen.

Map 1.1 Voltage instructions on the back of the screen

DGUS Three typical screen interface connection mode, according to the product type to select the appropriate cable to the screen for the power supply. Interface follows:

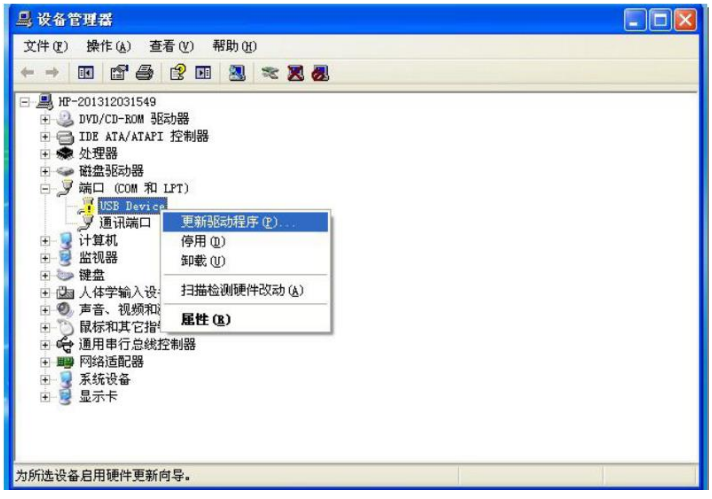
A. 10pin interface

The end of the flexible flat cable and DGUS Screen is connected to the terminal block, and the other end of the belt USB Interface adapter plate HDL662 Connection, then by a double male connector USB Line can be connected to a computer serial communications.

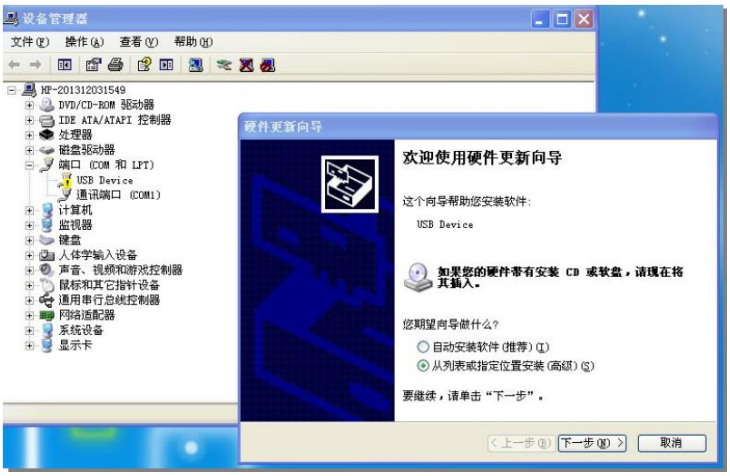
Map 1.2 10pin Interface wiring instructions

There are two USB-to-UART chip, XR21V1410 Chips and CP2102 chip. The chip type, Di Wenguan network to download and install the appropriate driver to ensure DGUS Screen can communicate properly with the computer. To ensure smooth communication, perform the following steps:

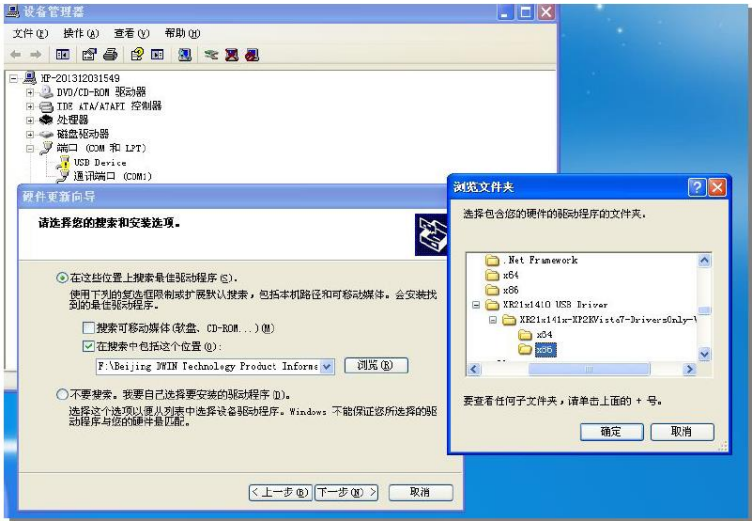
Step1 : Open the Device Manager, right-click " USB Device "And select Update Driver.



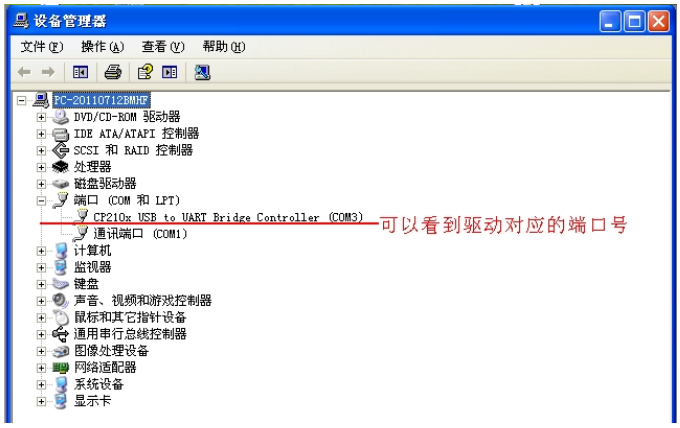
Step2 : In the pop-up window, select "Next."



Step3 : Select the path of the driver software to complete the updated driver.



After installation, the drive can view the corresponding port number in the Device Manager.



B. 8pin interface

8pin Interface points 2.0mm Spacing and 2.54mm Two kinds of pitches, corresponding to different types of connecting lines. As shown below, DGUS Screen and electricity

Brain, microcontroller connection lines used are not the same.

Map 1.3 8pin Interface wiring instructions

table 1.1 8pin Interface power supply accessories

Accessories Model	Applicable Interface	Explanation
HDL65011	8pin_2.0mm band DB9 And Power Interface 8Pin line. Cable length 20cm (Excluding DB9 head).	
HDL65012	8pin_2.0mm A connection DGUS Screen, a user side welded. Cable length 30cm .	
HDL65013	8pin_2.0mm double 8Pin_2.0mm Cable. Cable length 20cm .	
HDL65014	8pin_2.0mm double 8Pin_2.0mm Cable. Cable length 50cm .	
CI0608	8pin_2.0mm	8Pin 2.0mm SMT socket.
HDL65001	8Pin_2.54mm band DB9 And Power Interface 8Pin line. Cable length 20cm (Excluding DB9 Joints).	
HDL65002	8Pin_2.54mm band DB9 And Power Interface 8Pin line. Cable length 50cm (Excluding DB9 Joints).	
HDL65003	8Pin_2.54mm band DB9 And Power Interface 8Pin line. Cable length 100cm (Excluding DB9 Joints).	
HDL65020	8Pin_2.54mm double 8Pin_2.54mm Cable. Cable length 20cm .	
HDL65050	8Pin_2.54mm double 8Pin_2.54mm Cable. Cable length 50cm .	
HDL65100	8Pin_2.54mm double 8Pin_2.54mm Cable. Cable length 100cm .	

Precautions :

- In making the connection line, note DOUT Pin is the sender of the screen, DIN Pin receiving end panel;
- When the screen has two-level signal optional, defaults RS232 Signal, when the pad is shorted TTL signal.

Map 1.4 DGUS Level instructions on the back of the screen

C. 6pin interface

6pin Phoenix terminal cable required to produce a user, the terminal specified as follows:

table 1.2 RS232 / 485 Wiring interface description

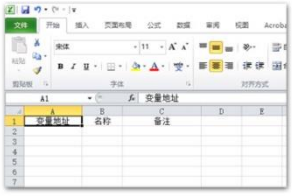
RS232 interface		RS485 interface	
GND	Power supply, the serial line to	485 +	The converter A (Positive)
VIN	power input	485--	The converter B (Negative)
232_Rx	Serial line output	VIN	power input
232_Tx	Serial input line	GND	Power Ground

[Note] use 485 Interface connected to your computer, you need a 232 turn 485 Interface to 232 Signal is converted to 485 And then signal DGUS Panels Communication.

Map 1.5 6pin Meet RS485 Wiring Instructions

1.2 Installation and use DGUS

1. 变量规划

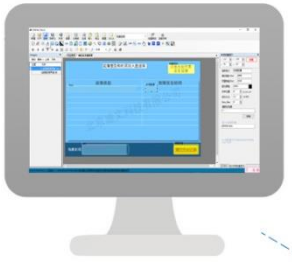


- 用户需要对所需变量进行定义并对数量进行规划。例如，需要监测两个温度值则需要两个变量。
- 推荐用户在整个开发过程中用Excel表格来记录整理变量的分配记录，便于后期维护。

2. 界面设计

- 利用绘图软件对界面及图标、字体库、按钮样式等进行设计，确保最终的显示效果与设计的效果一致。
- DGUS提供256MB/1GB/2GB的FLASH空间，可储存界面图片、图标、字体库等素材文件。

3. 界面配置



- 通过PC端DGUS开发软件对页面上的控件进行配置，并生成触控配置文件（13.BIN）和变量配置文件（14.BIN）。
- 在DGUS中可快速配置数据和文字的显示及输入，动画显示，图标显示，RTC，曲线显示等多种功能。

通过SD卡下载工程

4. 测试修改

- 界面的测试和修改（step2-3）。
- 通过串口连接用户的MCU系统，进行数据联调。

5条指令
实现数据读写

5. 定版归档

- 定版后，把配置文件、图片、字库、图标库等DGUS屏涉及的文件保存在一张SD卡就可以进行量产。

You need to install the production of works DGUS Software, then you need to install. Net Framework2.0 . . Net Framework2.0 Yes DGUS software

Operating environment driven. Proceed as follows:

Step1 : Right-click on "My Computer" and select "Properties", check the computer system 32 bit still is 64 bit ;

Step2 : According to the computer system digits to Di Wenguan website to download. Net Framework2.0 (x86) or. Net Framework2.0 (x64) Installed.

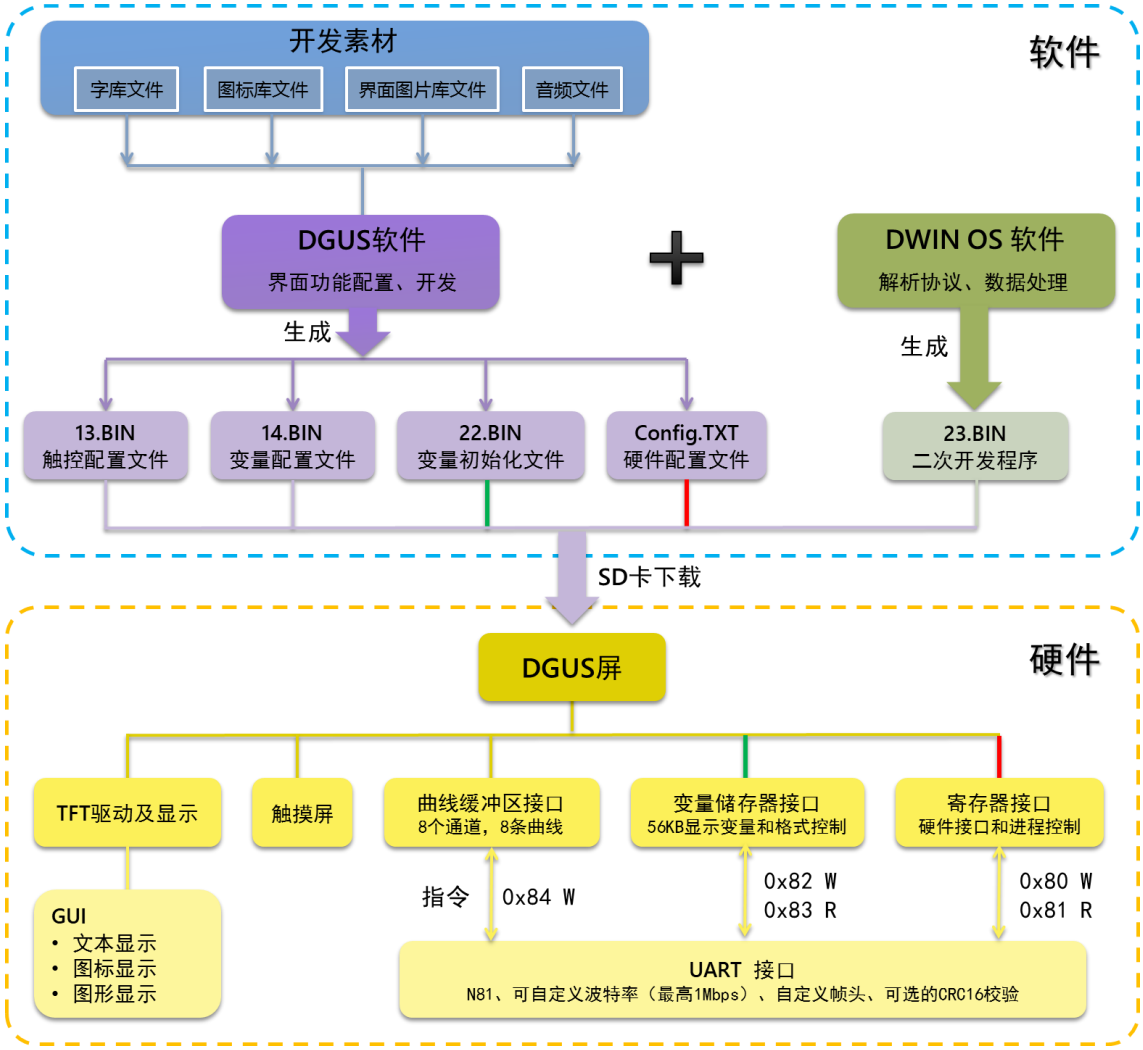
DGUS Screen is based on the configuration file to work, the entire development process is the use of user PC end DGUS Development of software-aided design complete change

The amount of configuration files, the basic development process shown in FIG.

Chapter DGUS Development System

2.1 DGUS Development System Introduction

DGUS System developed by DGUS Screen and DGUS Development software configuration, as shown below. DGUS Yes D WIN G raphic U tilized S oftware abbreviation of.



Map 2.1 DGUS Graphical development system

among them, DGUS Software is Beijing Devine Technology Co., Ltd. independent research and development of intelligent, graphical interface, man-machine system development software. The developi Based on pieces of supporting K600 + Kernel DGUS Screen use. The user through the computer terminal DGUS Development of software DGUS Screen development and design capabilities, which greatly reduces the difficulty of the user's development also reduces the user's development costs.

DGUS Developing software download address:

http://www.dwin.com.cn/supports/doc-download.html

DGUS Screen (hardware) from Beijing Devine Technology Co., Ltd. independent research and development factory-installed DGUS Based on software and K600 + Kernel driver smart screen, with reliable, stable, functional, good ease of use and many other advantages.

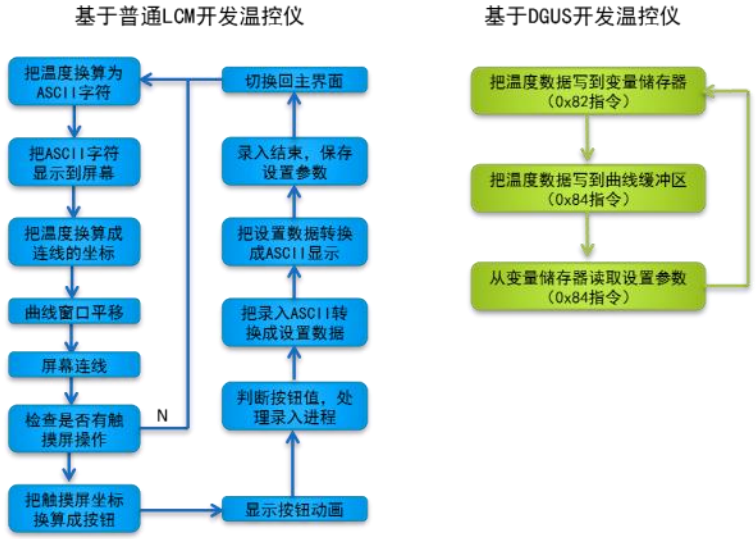
Two typical DGUS Screen(DMT80480T070_18WT with DMT80480T070_07WT) FIG appearance 2.2 And 2.3 Fig.

Map 2.2 DGUS Screen(DMT80480T070_18WT) Appearance

Map 2.3 DGUS Screen(DMT80480T070_07WT) Appearance

2.2 DGUS Development system advantages

And traditional LCM By timing different display or the command control, DGUS Direct drive variable-screen display, all the display and operation are based on pre-set variable configuration file to work, two different ways of working cause secondary open software architecture and development of applications when the user Send difficulty completely different.



Map 2.4 Development temperature controller flow chart

Suppose the touch screen to make a simple temperature controller, the temperature required to achieve real-time measurement of the current page in the display, and can switch to **set the parameters for the page by clicking on the touch screen. Tradition LCM with DGUS Comparative development mode in FIG. 2.4 Fig.**

D 2.4 In contrast, it is easy to see DGUS Screen development approach greatly reduces the user's development effort. Examples of the above relates to only a few parameters and page

Simple GUI If a complex contains dozens of pages and parameters, taking into account animations, icons and so eye-catching GUI And other needs, just 3-4 Engineer (can work together) spend 2-3 Days to complete.

In some small-scale projects in industrial automation applications, when the whole system is composed of a number of relatively independent, fully functional components (such as support Modbus The temperature controller protocol or other secondary instrument) is configured, the user can directly DGUS Used as the master, with 485 And the network equipment DGUS Screen to form a network, based on DGUS Mounted on the screen DWIN OS Secondary development platform to direct the development of soft and directly in the user master DGUS Screen On the run, instead of the user CPU work.

DGUS Development System **Advantage** They are listed below:

1. DGUS The GUI (Graphical user interface) of each page into multiple controls, users need to implement only need to use a feature PC End development software to add functionality controls on the corresponding page can be realized.

2. DGUS There are a variety of controls to choose from, can achieve rich functionality (such as data display, touch input, play a sound, etc.).

3. have 56K of RAM Variable space, 8 Trend graph channel memory, fast (fastest 80ms) Variable display response speed.

4. have 256 Read control byte configuration registers, instruction and support serial, hardware control and easy operation.

5. Each page can be set up to 128 A display control (supports overlay display controls) and any multi-touch control.

6. have SD / SDHC Interface, you can use SD Card to achieve DGUS Screen hardware parameters, picture and data downloading software upgrades, and mass production is extremely convenient, and easy to manage production files.

7. Integrated RTC (Calendar / lunar), integrated adjusting backlight brightness and the touch buzzer sound function.

8. Support voice playback, support for capacitive touch screen, the user can construct highly reliable database in the picture memory space.

9. Integrated DWIN OS Platform, which allows the user to put a portion of the code DGUS Running on the screen, allowing the user's secondary development easier, users can take advantage of rich instruction to achieve more complex functions. It also enables DGUS Screen as the master system as possible.

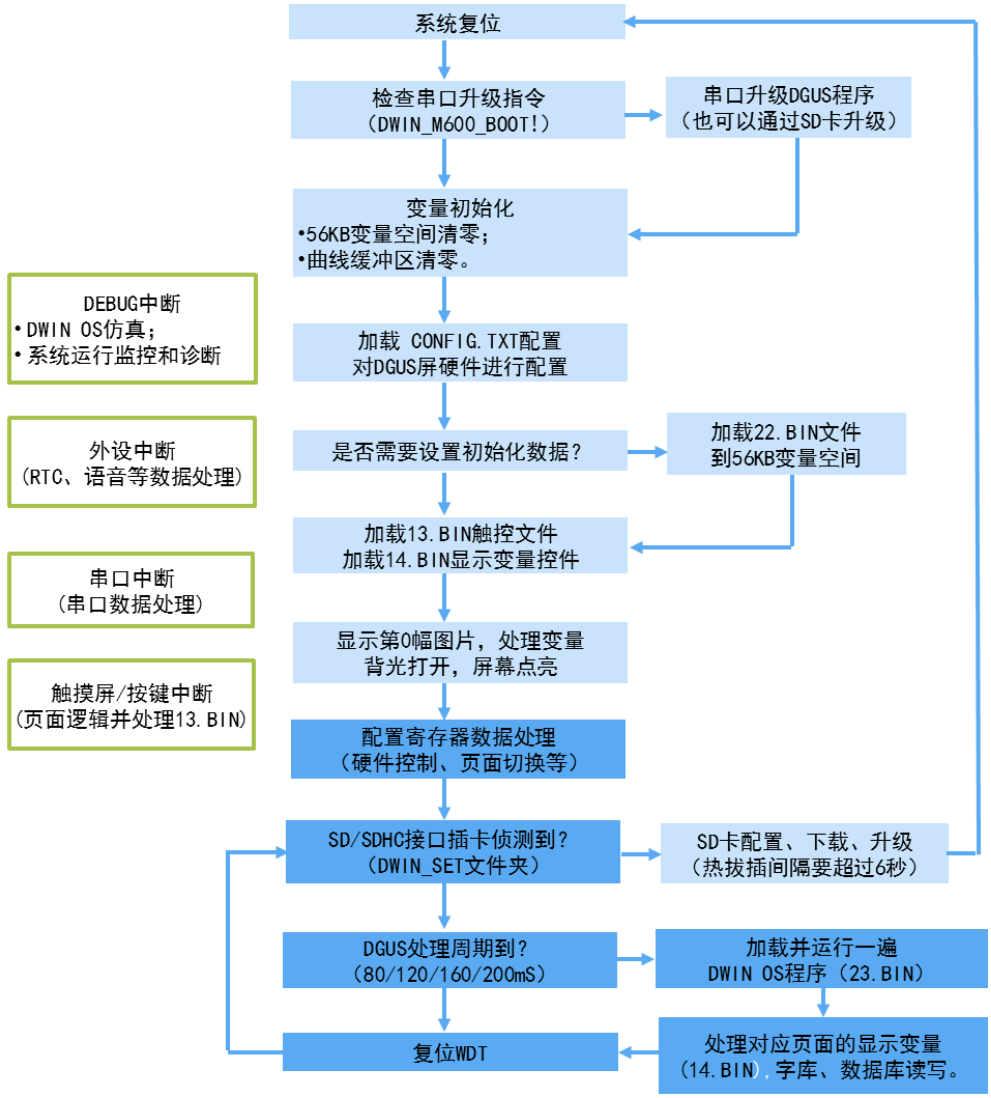
10. DWIN OS Platform integrates mathematical operations (including MAC , CRC), The data (including Flash Read database), serial communication, commonly used communications protocol processing (such as Modbus protocol, DT / T465 Electricity meter reading protocol, etc.), serial peripheral (e.g. a printer) driver, DGUS Process control instructions Typical applications include case Modbus Bus management, power meter reading, bill printing, POS Equipment.

11. It provides a reliable hardware platform (based Devine ASIC of HMI Platform architecture has experienced nearly 10 In industrial applications test), Devin proprietary software design (DGUS Software design using assembly code, code for a total of about 50KB), Making DGUS Screen is not only superior performance, but also extremely stable and reliable operation.

12. By product TUV , CE with RoHS Certification.

2.3 DGUS Software Processing Flow

When a user with PC end DGUS And using software to generate profiles SD Card to download DGUS Screen time, DGUS Software Processing Flow screen is shown 2.5 Fig. **Note: DWIN OS In each program DGUS cycle(80/120/160 / 200ms) Are complete running again, so DWIN OS Program in an infinite loop or delay by the instruction cycle can not occur.**



Map 2.5 DGUS Software processing flow chart

Chapter III DGUS screen configuration

3.1 Source file format requirements

From DGUS Introduction Development System, we learned that by DGUS When the screen development engineering, material documents need to be prepared with pictures, icon libraries, fonts, audio, and so on.

DGUS When calling material file is carried out by file number. therefore, **When the material file name, shall start with Arabic numerals ,**

Achieve numbered by Arabic numerals file. E.g, 0_ Boot. bmp , 01.bmp , 30 switch. ico , 48 Times New Roman GBK24.HZK , 0_ Please credit card. wav

Wait. After writing Arabic numerals can not write. Note that, the same material in the file does not allow the same number appears, otherwise the call will result in an error file.

A. Image files

Image files can be used as a user interface, the picture is stored in a dedicated storage space. In order to achieve a good display of the image file

Format, there are strict requirements, and image files must be DGUS Screen with the same resolution twenty four Bit color BMP Format files.

Note that, the picture is not only the background of the user interface, but also contains an icon appears in all of this interface buttons, menu options, in DGUS Software development can only define the functionality of a region of the picture, you can no longer add new buttons or menu icons in a certain area,

So please plan at the time of making the picture a good location and icon for the current page of all the buttons and menu options.

B. Icon library files

Icon file is created by drawing software picture format, using the Arabic numerals on the picture icon are numbered, by the need DGUS Icon icon library generation software to generate files. Each icon pixel size should be less than 255 * 255 When outside of this range, the icon generating software will automatically FIG.

Marked down to 255 .

Note that the library file is usually larger than an icon 256KB The need to occupy more than one space (FLASH Partition space available at the relevant 3.3 Section). Therefore, the number of icon libraries can not be consecutive numbers. For example, when each icon library files take up two spaces, document preparation

Numbers should be separated 1 : 30 remind. ico , 32 Lights. ico . If each of the three icon library files take up space, file number should interval 2 : 31 malfunction. ico ,

34 hint. ico .

C. Font file

DGUS Support a variety of general international character encoding: 8-bit , ASCII , GBK , GB2312 , UNICODE . DGUS It has been pre-installed

ASCII Character encoding, which includes a dot size 64 ~ 4 * 8 * 128 All ASCII character. The font number 0 It can be called directly

Digital, letters, etc. is displayed.

When the need to use other coded character, generated by the character generator needs. DGUS stand by BIN , DZK , HZK These three font file.

D. Audio files

section DGUS Screen (see specific corresponding model DGUS Description screen hardware specifications of the book) support 128 Segment voice playback, you need to download the sound files stored in advance DGUS Screen. Number range for the audio file 0-127 Containing 0 with 127 . Sound file format wav ,use 32KHz

Sampling frequency, 16bit Mono wav File format. Sound files using additional Flash Storage, do not take DGUS of FLASH Space, download speed is about 32KB / s .

In order to ensure the correct format audio files can be generated by Towav.EXE Format conversion software. The software can be obtained from Devine engineer.

DGUS Screen data formats and color systems as follows:

A. DGUS Screen data format

due to DGUS The main screen for MCU And other embedded system applications, in order to deal with the user's convenience, DGUS Screen data used in integer (word), unsigned integer (word), long integer (double word), long integer (4 Words) that represents the range in relation to the following:

Integer	- 32768 (0x8000) To + 32767 (0x7FFF)
Unsigned integer 0 (0x0000) To 65535 (0xFFFF)	
Long integer	- 2147483648 (0x80000000) To + 2147483647 (0x7FFFFFFF)
Long integer	- 9223372036854775808 To + 9223372036854775807

Decimal fixed-point decimal representation of user-defined number of decimal places. Such as 0x4D2 (1234) Provisions for the decimal 2 When, indicates 12.34 .

B. DGUS Screen color system

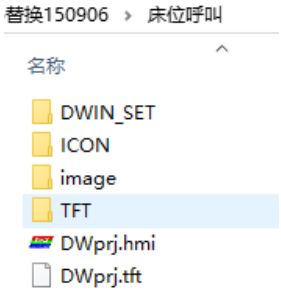
DGUS Use screen 65K Color system defined color palette Table 3.1 Fig.

table 3 .1 D G US Screen Tone color board of set Righteousness

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Define R4	R3	R2	R1	R0	G5	G4	G3	G2	G1	G0	B4	B3	B2	B1	B0	
	red 0xF800						green 0x07E0									blue 0x001F

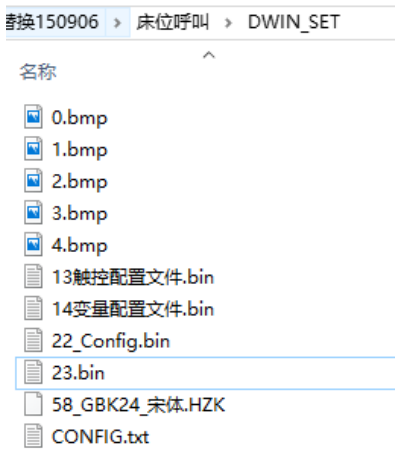
3.2 Up the configuration file

When a new project, will automatically generate a series of documents, as shown in FIG. among them, DWprj.hmi Yes DGUS Software uniquely identifiable The editing program, the file can not be renamed, not deleted. DWIN_SET Folder contains the final will be downloaded to DGUS All files in the monitor.



Map 3.1 DGUS Project file in the root directory

As shown below, DWIN_SET Folder contains the interface picture, touch the configuration file, the configuration file variable, variable initialization file (22_Config.bin), OS Code (23_DIV.bin), Icon library (30.ico , 35. ico), The screen parameter configuration file (CONFIG.txt). Among them, number 13 , 14 , twenty two , twenty three of bin Documents and CONFIG.txt File is DGUS Software and DWIN OS Generated by all profiles.



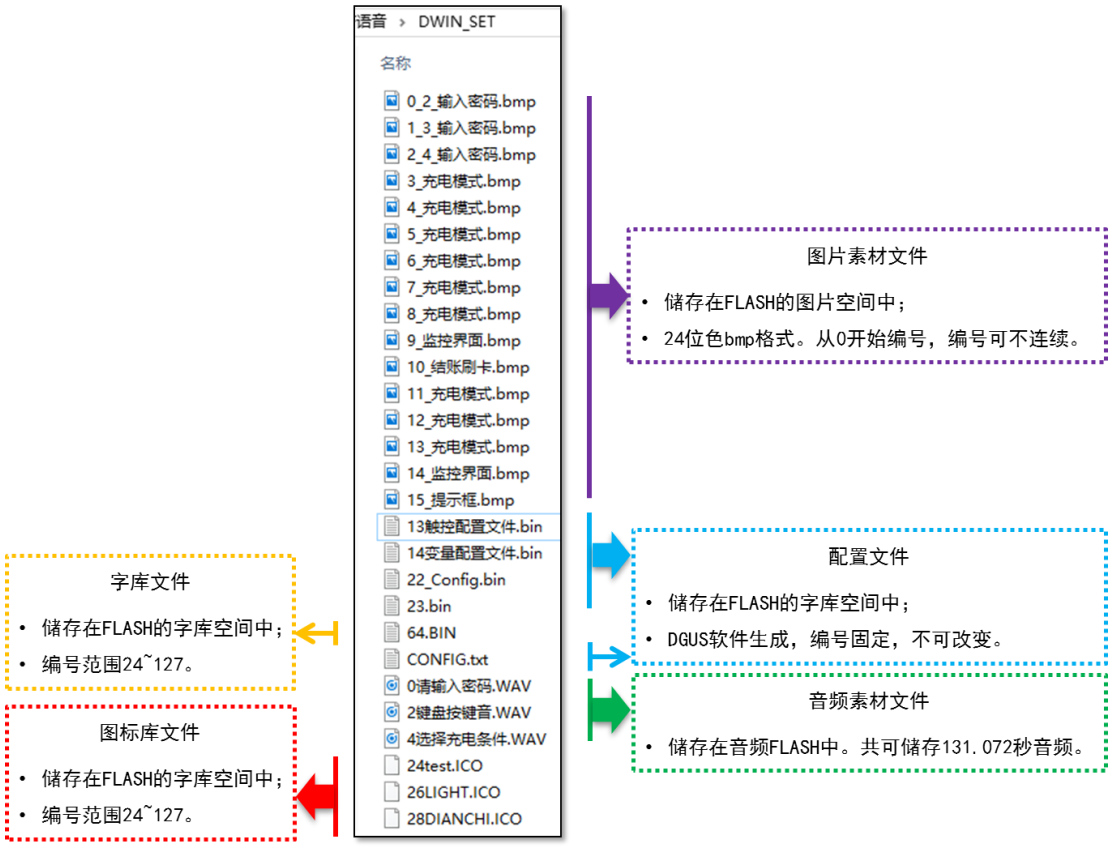
Map 3.2 DWIN_SET Folder root

3.3 Storing material files, configuration files and data

DGUS Screen offers FLASH (256MB / 1GB / 2GB)storage space, RAM (56KB) Storage, configuration register (256Byte) Space, and a buffer curve (16KB). Support for playback of voice DGUS Screen, provides additional audio FLASH space.

The following figure shows the way to store content files and configuration files, subsequent chapters will explain how to allocate space to store all types of files. As can be seen from the figure, DGUS Files are numbered. Number of the file stored in the same space, can not be repeated, for example, can not have two 0 No. Fig. sheet. The file number of different spaces do not interfere with each other, for example, 13.bmp with 13_ Touch configuration. bin It can exist at the same time.

At the same time, we also found some numbers icon library files, audio files are not continuous, which DGUS of FLASH Partitioning and storage related manner, in the following detailed description will do.



Map 3.3 File storage location Overview

3.3.1 FLASH storage space

FLASH The main storage for storing the profile parameters (CONFIG.txt), Image files, sound files, font files, icon libraries, touch configuration files, variables profile, OS Code, the user data and the like. Store in FLASH Is not lost when power-down data and files. No matter how much capacity is FLASH Space, both fixed separation 32 MB Space (hereinafter referred to as character space) to store DGUS Stored files, some

Configuration files, fonts and icons as well as user-defined libraries.

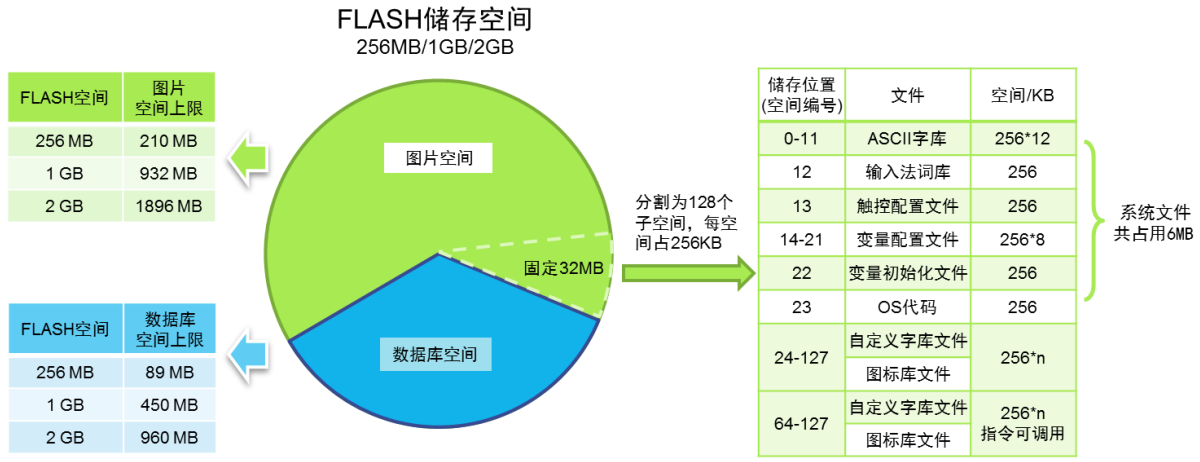
FLASH According to the contents of the storage space is divided into picture storage space and database space. Among them, the picture space, fixed separation 32 MB Font space. Font space is divided into 128 Parts, the size of each space is 256 KB . According to the configuration file to the corresponding file number storage space

Position, as shown in FIG. 0-11 No space to store the DGUS Pre-numbered 0 of ASCII Font file, the file is large, it takes up from 0 No space begins 12 Spaces. 13 No space stored by DGUS Software-generated number 13 Touch configuration file. As can be seen from the figure, 32MB Font space, 0-23 No space (total occupancy 6MB) Stored in

the file type is fixed and can not be used to store other files.

Therefore, user-defined font file number, file number should be in the library icon 24-127 The number of taken (including twenty four with 127), Which only 64-127 No space can be invoked

through the command.



Map 3.4 FLASH storage space

Stored in the font file space character should be represented by a storage location (0-127) At the beginning of Arabic numerals to name. If the user needs to

Font files are stored to DGUS The first screen 35 A font space, font file name format is " 35 * .HKZ "(Where * can be any input may not enter), such as" 35 Chinese character library. HKZ "" 35.HKZ "or" 035 test. HKZ "Etc., can not be named" font 35.HKZ . "

table 3.2 Content font storage space

storage location	file type	Naming Specification	For example	Explanation
0-11	dedicated ASCII Font	0 * .HZK	0_ASC Library. HZK	Devine pre-installed
12	Input method thesaurus	12 * .BIN	12_PY Library. BIN	
13	Touch Configuration	13 * .BIN	13 Touch files. BIN	DGUS Development of software to generate
14-21	Variable configuration	14 * .BIN	14 Variable file. BIN	
twenty two	Variable initialization	22 * .BIN	twenty two initial. BIN	
twenty three	OS Code	23 * .BIN	twenty three Water treatment. BIN	DWIN OS generate
24-127	Font file * Font storage location. bin / HKZ / DZK 35 Chinese character library. DZK TS3 Font extraction software generated icon library			
	* Font storage location. ICO		51 Icon library. ICO	Devine Toolbox generation

Except 32MB Outer space character, the rest of the space for storing pictures, and user interface data. For different FLASH Space capacity, and the capacity of the database of images have an upper limit, as shown in the following table.

table 3.3 The upper limit of the picture space and database space

Flash Picture maximum maximum database space space space		
256 MB	210 MB	89 MB
1 GB	932 MB	450 MB
2 GB	1896 MB	960 MB

In the case of the largest space of the picture, the number of images of different resolutions can be stored as table 3.4 In FIG.

table 3.4 Picture storage space and the relationship between the number of

FLASH space	Different resolutions DGUS The maximum number of stored images screen						
	320 * 240 480 * 272 640 * 480 800 * 480 800 * 600 * 768 1024 * 6001024						
256MB	836	836	278	278	209	167	139
1GB	3728	3728	1242	1242	932	745	621
2GB	7584	7584	2528	2528	1896	1516	1264

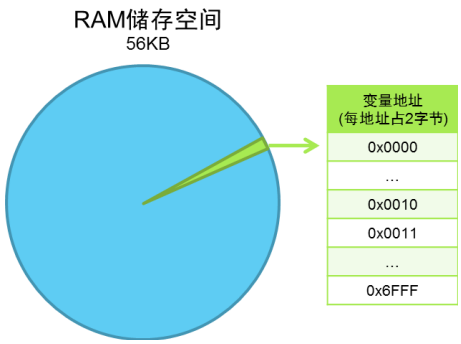
3.3.2 RAM storage space

RAM Space is fixed at 56KB , Divided into address 0x0000 ~ 0x6FFF Subspace (Mini DGUS for 0x0000 ~ 0x07FF). Each address space corresponding accounting 2 byte. in DGUS When used in the description or the variable address pointer is set to address the first address space of the data storage, i.e. the address data (first address) begins sequentially arranged sequentially stored. Each variable address (first address) point of space is not fixed,

Thus, in DGUS When the variable address assigned to each variable software should calculate the amount of data to be stored is good, otherwise it will allocate space possible overlap and cause an error.

General description pointer is set to recommend 0x4000 ~ 0x6F00 Between the variable address set 0x0000 ~ 0x4000 So it will not conflict.

【Note】 6F00 To 6FFF Variable address portion between the hardware parameters is used, it should be avoided.

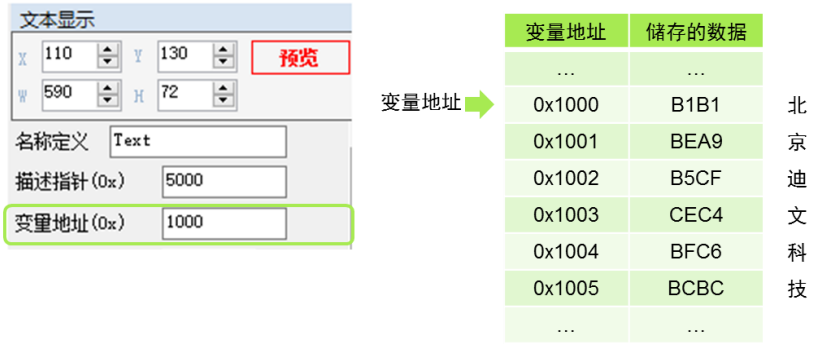


Map 3.5 RAM storage space

A. Variable address

Variable address RAM Stored in the first address space subspaces one or more variables, values stored code or state variables in the variable display address space of the **pointed, will be exemplified below. For example, a text display variable control is set to address 0x1000 , The text displayed in the control as "Beijing Devine Technology", then RAM Storage** space in a manner as shown in FIG. As can be seen, each address can be

Store 2 Byte Content.



Map 3.6 Variables stored in the address data in a manner

If you need to change the text display character displayed in the control, only you need to change the character encoding to address stored in the corresponding variable. Variable address can modify the data stored by the transmission instruction input and the touch screen. For example, modifying a text display control may be implemented by a text entry control, just two controls set to the same variable address, details, refer to the subsequent section; the same time, can change the contents stored in the address by sending a command.

【example】 Send command: **5A A5 05 82 1001 0031**

1001 : Variable address the Chinese character "jing" storage location

0031 : Numeric character " 1 "Encoding

Effect instructions: the character " is replaced by a display position of a digital " 1 "

Similarly, by modifying the values of the variables also modify the address stored in the display data, activation / stop animation, icon or the like to switch various effects. Such as:

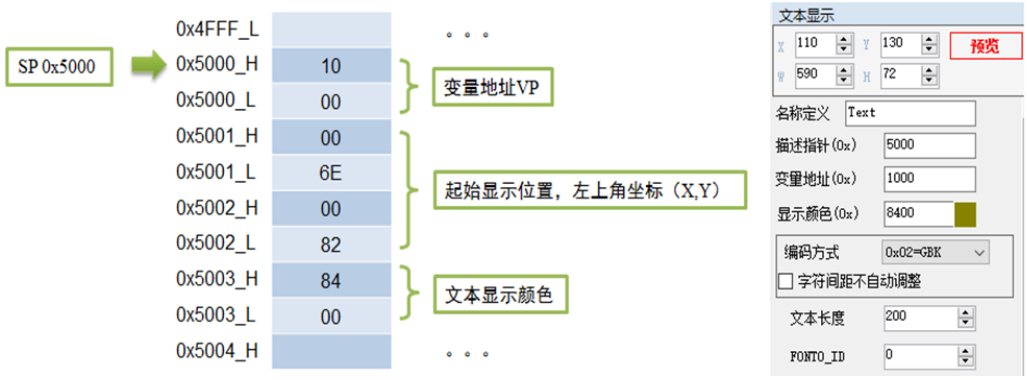
Features	Send instructions	effect
Icon display	**5A A5 05 82 0001 0001**	**In the address 0x0001 Controls at 1 Icon**
Real-time data	**5A A5 05 82 000A 000B**	**in 0x000A Address in written 11**

B. Description Pointer

Pointers are described RAM First address space stored in the subspace of a variable of descriptive attributes stored in the display space of the address pointed to by the The property values of the variables, such as display coordinates, color, font size. It should be noted that the description of the common pointer and pointer variable RAM Space, allocating variables should avoid overlap address space.

Let's continue in conjunction with the text display controls to explain. As shown below, according to a text display control instructions stored in a table format can be seen, the text control described Space pointer to said sequentially successively stored in the variable address, the character display position coordinates of the upper left corner, a character color, a text box the upper left **corner and lower-right coordinates data. Figure 0x5000_H It represents the high byte, 0x5000_L Low byte.**

文本显示控件指令储存格式				
地址		定义	数据长度	说 明
0x00		0x5A11	2	
0x02		*SP	2	变量描述指针，0xFFFF 表示由配置文件加载
0x04		0x000D	2	
0x06	0x00	*VP	2	文本指针
0x08	0x01	X, Y	4	起始显示位置，显示字符串左上角坐标。
0x0C	0x03	Color	2	显示文本颜色
0x0E	0x04	Xs Ys Xe Ye	8	文本框



Map 3.7 Data storage methods described pointer

Typically, we describe variables to change the properties described change the value stored in the pointer by sending a command.

【example】 Send command: 5A A5 05 82 5003 F800 (Effect: character color from yellow to red soil.)

5003 : Store Address text color.

F800 : Red code.

Other applications describe pointer example as follows:

Features	Send instructions	effect
Changing the display position data	5A A5 07 82 5001 0000 0000 Character display left corner position becomes (0,0).	
change ASCII Character dot size	5A A5 05 82 500A 3060	The character dot matrix changed 48 * 96 ,Note: X with Y Lattice direction need to be modified value.
Hidden characters	5A A5 05 82 5008 0000	The character length value becomes 0 In order to achieve a hidden character.
Replace the font file	5A A5 07 82 5009 003C 1010 transfer 60 No. font, font size dot matrix 16 * 16 . That is, after replacing the font file Character size also changes.	

3.3.3 Configuration register space

DGUS Register space capacity 256 Byte . Different from the memory registers, which is used to store register state, such as RTC Real-time status (real time), the backlight brightness. by DGUS Changing the value of each serial instruction register, and the host computer may be DGUS Screen information transmission and control.

3.3.4 Buffer curve

DGUS provide 16 KB Buffer curves, can be stored in the buffer 8 Trend curves. According to the data DGUS Instruction format is sent to the buffer, the graph can be implemented quickly. Curve data buffer is 16 Bit unsigned numbers.

3.4 Profiles

This section describes DGUS Screen parameter configuration file CONFIG.txt And variable initialization file 22.bin Touch Profiles 13.bin And variables

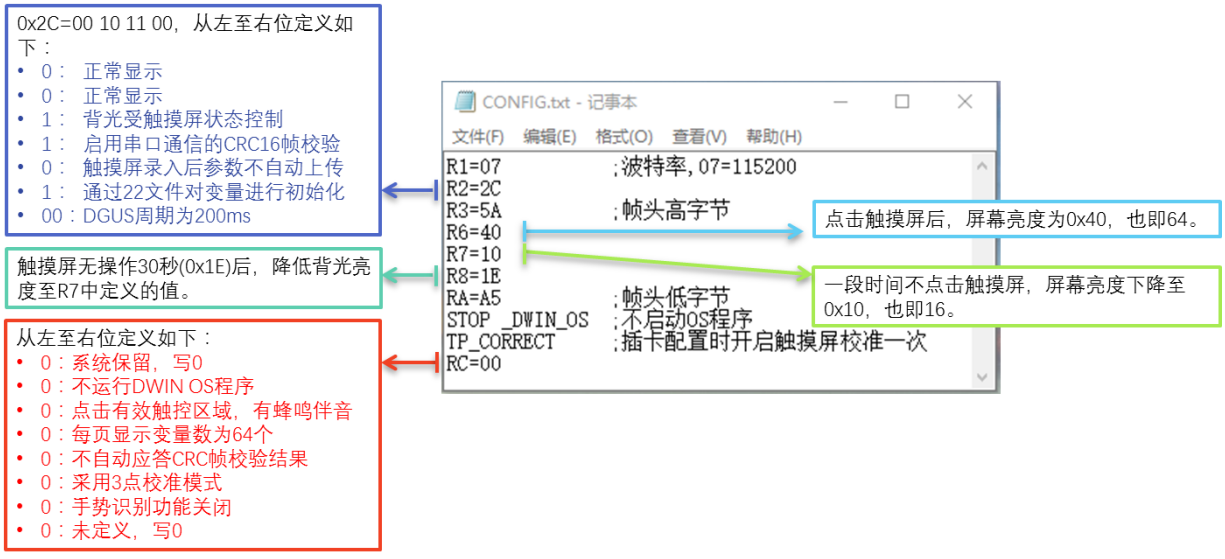
Profiles 14.bin See relevant content to explain the subsequent chapters.

3.4.1. System configuration files CONFIG.txt

DGUS Parameter configuration screen is by CONFIG.txt Profile write register references, and with SD Card Download the file to DGUS Screen implemented. CONFIG.txt File is windows The text document format, use a scripting language similar way to describe the parameter register, each line describes a parameter, without modifying the parameters can not write. The file can also DGUS Development of software to generate. in DGUS Development software, click " Configuration " Button in the pop-up menu for DGUS Screen baud rate, frame header, after setting the backlight control point hit " Output Profiles " To generate CONFIG.txt Profile. System configuration window as shown in FIG. The window settings for each item can be edited by CONFIG.txt Documents to complete, only some of the features by editing CONFIG.txt File to achieve, details can be found in the table below for each register.

Map 3.8 DGUS Software configuration window

CONFIG.txt File format must be R? = HH Where? Is the number of registers, HH It is the configuration register 16 Hex (HEX) value, The letters must be capitalized . The figure shows a typical CONFIG.txt File and its significance.



Map 3.9 CONFIG.txt Standard Configuration Example

Typically, DGUS Software generates CONFIG When the files, only displays R1 And other items different from the factory settings. From the figure above

As can be seen, the parameter registers may be implemented just a few lines of DGUS Dozen system configuration, system configuration register may be implemented by the rich control.

Note that, the system configuration register, there can not be modified (e.g., resolution), and some can be directly used alone, some need

Used together with other registers, we will do explain.

1. Physical resolution setting screen (R0)

Display by a physical resolution R0 Register set, as shown in Table 3.5 Fig. **[Note] that has been set up at the factory, users do not need to configure.**

table 3.5 Physical resolution of the display

R0 Setting value	Resolution setting	typical DGUS Screen models
00	640 * 480	DMT64480T056_03W
01	640 * 480	DMT64480T057_01W
02	800 * 480	DMT80480T070_07W
03	800 * 600	DMT80600T080_07W
04	1024 * 768	Special customized screen
05	1024 * 768	DMT10768T057_01W
06	800 * 600	Special customized screen
07	800 * 600	Special customized screen
08	800 * 600	MGA01 , MDVI01
09	1024 * 768	DMT10768T150_02W
0A	1280 * 800	Unused
0B	1024 * 600	DMT10600T102_02W
0C	1366 * 768	Unused
0D	240 * 320	Special customized screen
0E	320 * 240	Special customized screen
0F	480 * 272	DMT48270T043_03W
10	480 * 272	Special customized screen
11	800 * 480	Special customized screen
12	320 * 240	DMT32240T035_02W

2. Bit clock phase setting display (R4)

DGUS Since the LCD screen used TCON Different, which shows the phase relationship data and the display clock, there are two bits, the R4 Register set, as shown in Table 3.6 Fig. **[Note]** that have been set up at the factory, the user no configuration, if the configuration error will result in the display screen is flickering or appear burrs.

table 3.6 Bit clock phase relationship display

R4 Settings	Bit clock phase relationship display
00	Display data in the bit clock falling edge latches
FF	Display data in the bit clock rising latch

3. Set the serial port baud rate (R1 , R5 , R9)

DGUS Serial communication baud rate by a screen R1 , R5 , R9 Register settings. **DGUS When screen factory R1 = 07 That the baud rate is 115.2Kbps .**

when R1 The value of 00-10 When between, R5 with R9 Does not work, at 17 A gear shift select baud rate, each file corresponding values shown in the following table.

table 3.7 Serial port baud rate setting

R1 Settings 00		01	02	03	04	05	06	07	08	09	0A	0B	0C		0D	0E	0F	10
Baud Rate / Kbps	1.2	2.4	4.8	9.6	19.2	38.4	57.6	115.2	28.8	76.8	62.5	125	250				230.4	345.6	6912	921.6	

when R1 Value of 11 When, at this time by the baud rate R5 versus R9 Decision, is calculated as follows:

R5 : R9 = 6250000 / Baud rate, which R5 : R9 A parameter indicative of double-byte, R5 High byte, R9 Low byte.

【example】 To set the baud rate is 10Kbps , R5 : R9 = 6250000/10000 = 625 = 0x0271 , Is set R5 = 02 , R9 = 71

4. Set serial communication frame header (R3 , RA)

A communication frame header has two functions: identifying a synchronous serial data for one frame, the second is a plurality of DGUS When the screen in parallel, as the header For the address of the device to be distinguished. DGUS Serial communication through the screen header R3 , RA Register settings.

【example】 Set up R3 = AA , RA = BB , The command must be sent 0xAA 0xBB Beginning to be DGUS Identifying and receiving screen.

【Note】 DGUS The factory default screen for the communication header R3 = 5A , RA = A5 That is the header 0x5AA5 .

5. Set software operating mode (R2 , RC)

R2 , RC Register in binary bits to represent, for configuring DGUS Software operating mode screen. R2 Register defined in Table 3.8 (The factory default values are 0).

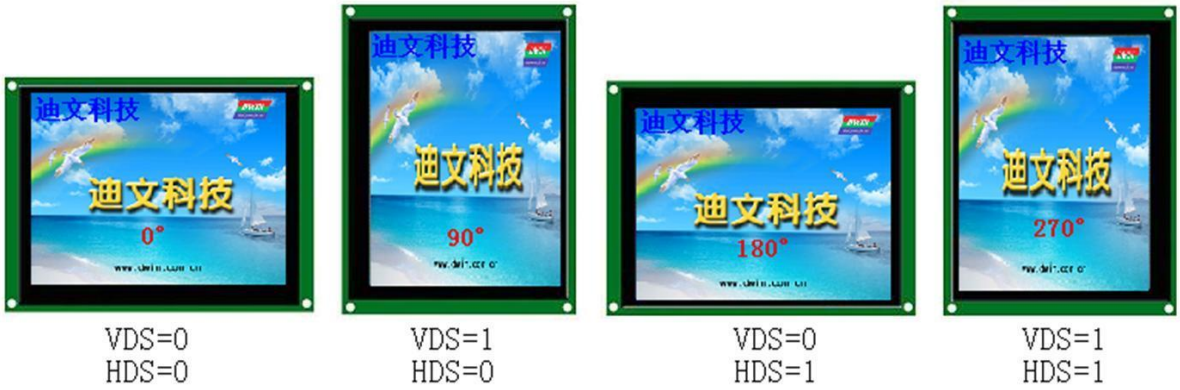
table 3.8 R2 Defined register

Place	Weights	definition	Explanation
7	0x80	VDS	0 = normal display 1 = deflection 90 ° display
6	0x60	HDS	0 = normal display 1 = deflection 180 ° (Anti-viewing angle) display
5	0x20	TP_LED	0 = No state of the touch screen backlight control 1 = The backlight control state by the touch screen, control parameters R6 , R7 , R8 Register Set
4	0x10	FCRC	0 = Do not enable serial communication CRC16 Frame Check 1 = Enabling serial communication CRC16 Frame Check
3	0x08	TPSAUTO	0 = The touch screen does not automatically uploaded (user query) after logging parameters 1 = The touch screen is automatically uploaded to the serial input parameters (determined by the configuration variable corresponding touch)
. 2	0x04	L22_Init_En	0 = 56KB Variable memory (RAM) To power up initialization data 0x00 1 = 56KB Variable memory (RAM) By a power-on initialization data twenty two Font file is loaded
. 1	0x02	FRS1	Set up DGUS cycle, DGUS The smaller the cycle is variable in response more sensitive, but the lower the ability of the process variable. 【Note】 for 1024 * 768 Resolution, it is recommended DGUS Period of more than 120ms . DGUS week Period will affect the animation Map Standard display of Animation speed degree.
0	0x01	FRS0	

DGUS cycle	80ms	120ms	160ms	200ms
FRS1	1	1	0	0
FRS0	1	0	1	0

【example】 To achieve the input parameters automatically uploaded (Weights 0x08), DGUS cycle 120ms (R2.1 Weights 0x02 ; R2.0 Bit 0 , Not counting the weight), The other parameters are still factory value. Non 0 Do the right project and add weight to calculate R2 ,then R2 = 0x08 + 0x02 = 0A (= 00,001,010)

R2.7 (VDS) with R2.6 (HDS) FIG four combinations Effect 3.5 Fig.



Map 3.10 R2.7 (VDS) with R2.6 (HDS) Four combinations effect

RC (AUX_CFG Define the configuration word) registers as shown below (**The factory default values are 0**).

table 3.9 RC (AUX_CFG Define the configuration word) register

Place Weights		definition	Explanation
7	0x80	Write reservations system 0	
6	0x60	RUN_OS_EN 0 = Do not run DWIN OS program 1 = run DWIN OS program	
5	0x20	TP_BUZZ_EN	0 = Click on a buzzer sound when the effective area of the touch screen 1 = No buzzer sound when tapping on the touch screen active area, but still be able to 0x20 Writing register data to control the buzzer beeps.
4	0x10	PAGE128_EN 0 = The number of variables per page 64 A need in DGUS Select the corresponding software development 64 Variable mode 1 = The number of variables per page 128 A need in DGUS Select the corresponding software development 128 Variable mode	
3	0x08	CRC_ACK_EN 0 = start up CRC After the frame check, acknowledgment frames does not automatically check result 1 = start up CRC After the frame check, an automatic response frame check result	
. 2	0x04	TP_CAL_MOD 0 = Touch screen with 3 Point calibration mode 1 = Touch screen with 5 Point calibration mode	
. 1	0x02	Gesture_EN	0 = Touch screen gesture recognition function off 1 = Touch screen gesture recognition feature is turned on
0	0x01	Undefined	write 0

【example】 For variables per page 128 One (**Weights 0x10**),start up CRC Check autoresponder results (**Weights 0x08**), Turned gesture recognition (**Weights 0x02**), The other to keep the factory preset values. Then do these three weights and added, RC = 0x10 + 0x08 + 0x02 = 0x1A (**= 00,011,010**).

6. Standby and wake-up the backlight (R2.5 , R6 , R7 , R8)

When set R2.5 = 1 , The operation can be achieved without quenching effect and touch screen standby wake of the screen. The effect is to be combined R6 , R7 , R8 These three Register together to complete. R6 Configuration of backlight luminance when the wake-up screen (backlight brightness range 0 ~ 64 , which is 0x00 ~ 0x40); R7 Backlight brightness (backlight brightness range supra) Once the quench standby screen; R8 Configuration start time of no quenching standby screen, that is how long the screen put out when no operation panel. Configuration shown in the following table. **[Note] backlight standby after the first pressing of the touch screen is not touch-trigger control.**

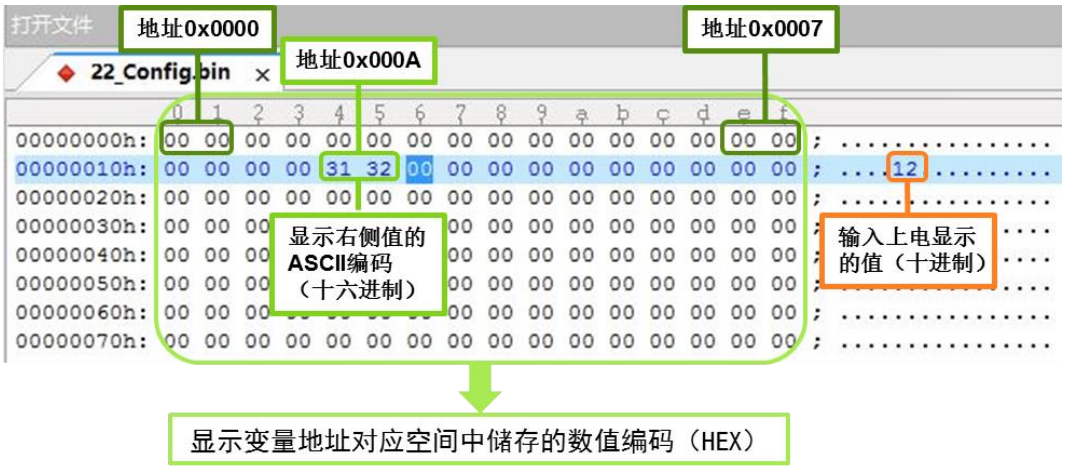
table 3.10 Touch screen backlight control

Register range	Explanation
R6	0x00-0x40 R2.5 = 1 When, after the wake touch screen backlight brightness.
R7	0x00-0x40 R2.5 = 1 When a period of time not tap the touch screen, the standby screen backlight brightness during quenching.
R8	0x01-0xFF R2.5 = 1 When the touch screen backlight off time, in seconds.

3.4.2 Variable initialization file 22.bin

22.bin File can be used to display the variables are initialized. When enabled twenty two When files, variables display controls at power displays 22.bin File defined values, rather than 0 .

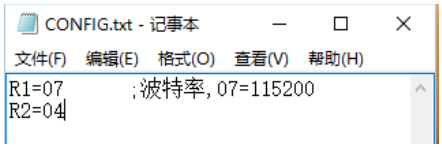
22.bin File by DGUS Software generation, but also through new programming software, simply file eventually named 22 * .bin (* Represents any non Numeric characters, can not write). Editing the file is shown below in FIG. DGUS Variables corresponding to each two-byte address space, therefore, the first file 0 On line 0 Column and 1 Columns are variable address 0x0000 correspond. In Fig. 3.11 , The variable address 0x000A Stored value is modified to 12 That after power-on all the variables address 0x000A The controls are digital display 12 .



Map 3.11 22.bin File editing method icon

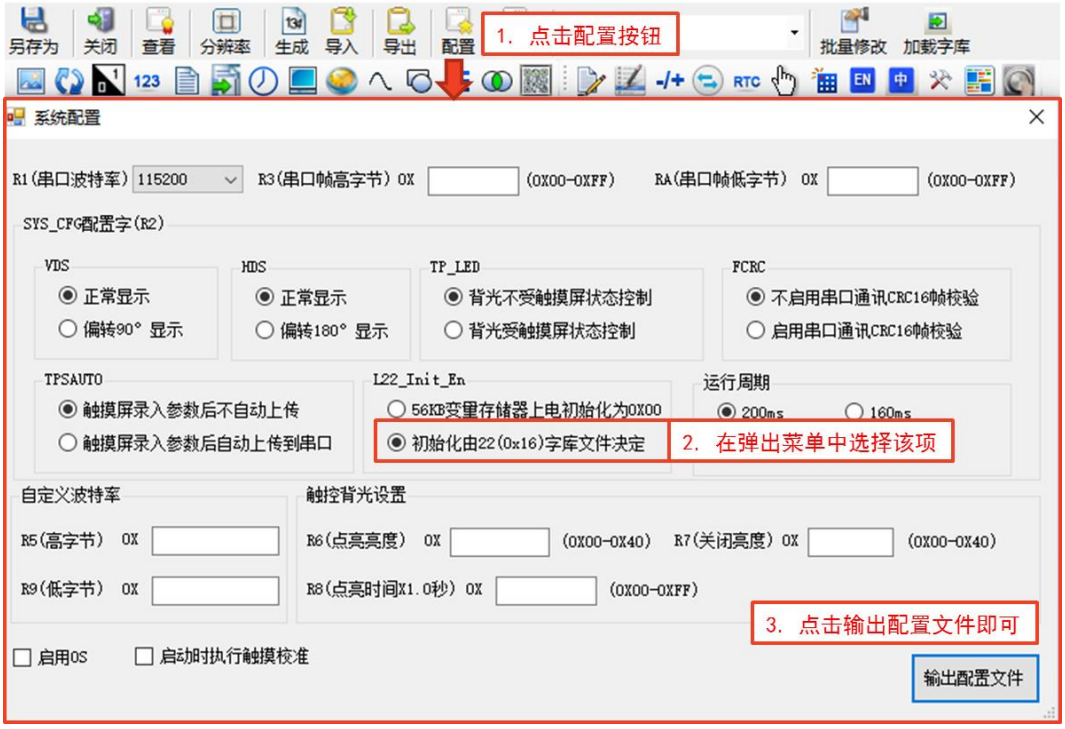
have to be aware of is, DGUS Software is not enabled by default 22.bin Initialization files, you need to manually enable the initialization function. 22.bin Enable file is included in the DGUS Parameter configuration, which is explained in the previous section CONFIG.txt file. according to CONFIG Modify the way files can be enabled in two ways twenty two Initialization file.

A first, manually CONFIG.txt Add a row R2 = 04 It can be. (register R2 See the definition of a.)



Map 3.12 Enable twenty two File initialization mode of operation

The second, in DGUS Click Software " Configuration " Button in the pop-up interface, select " Initialized by the twenty two Font file determines " The output configuration File to update CONFIG Files.



Map 3.13 In DGUS Enable Software twenty two file

Finally, the new 22.bin File and CONFIG.txt Files in DWIN_SET Folder, by SD Card to download DGUS Screen

It can be.

3.5 DGUS Screen debugging

3.5.1 Screen Calibration

There are three calibration methods:

1) Calibration methods 1

Power on state, 4 Within seconds faster continuous click non-touch area of the touch screen (i.e., a region not set the touch function) exceeds 20 Times, the process proceeds to the calibration mode of the touch screen, comprises the following steps:

Step1 : 4 Quick click within seconds the touch screen area than non-touch 20 Times.

Step2 : Buzzer sounding 1 Seconds, stop tap the screen when you hear the buzzer.

Step3 : Calibration mode, the specified position of the touch screen prompts click cross lines to calibrate the touch screen.

Step4 : End of calibration, return to the screen before the automatic calibration.

2) Calibration methods 2

in CONFIG.txt File, writes a line of special text "TP_CORRECT" Will initiate a touch screen calibration function.

3) Calibration methods 3

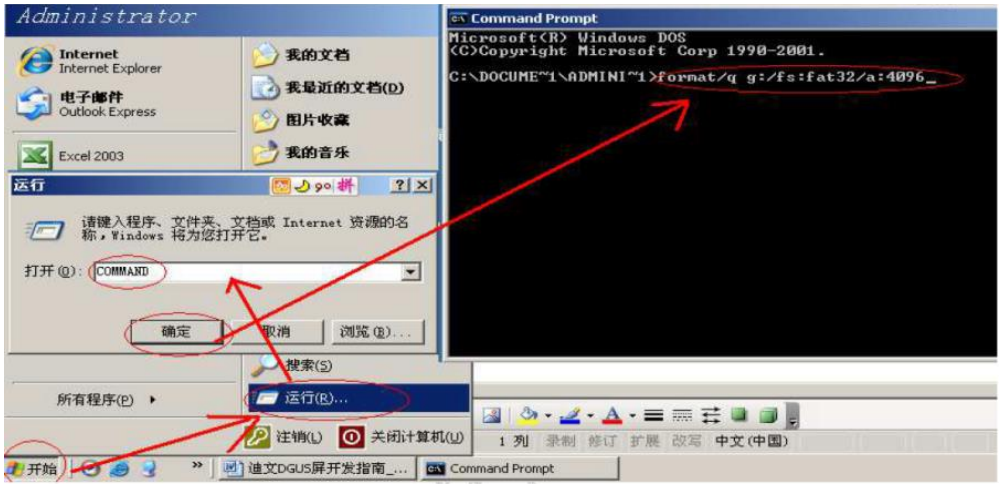
Through the serial port to 0xEA Write register 0x5A Instruction will start the calibration process once the touch screen.

3.5.2 download tool SD Use the card

DGUS All hardware parameter setting screen, as well as download DGUS Software upgrades through on screen SD / SDHC Interface to complete. For the first time SD When the first card is recommended for users SD Card format, and the file system is set to FAT32 Format, the following steps:

Step1 :in Windows Click "Start" - "Run", type " COMMAND "run DOS system;

Step2 : Enter **format / q g: / Fs: fat32 / a: 4096** . among them **g:** for SD Letter cards, can be replaced according to the actual situation.

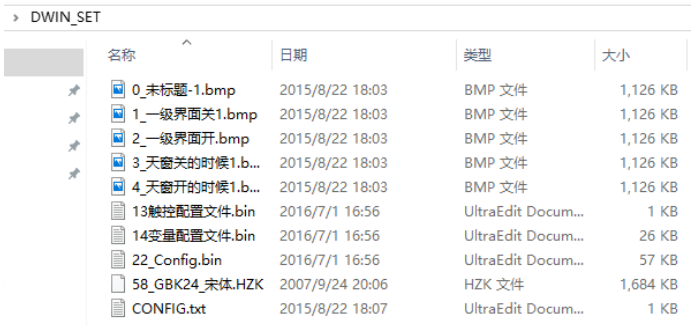


Map 3.14 SD Card Format Description

1. by SD Download the project card

Will contain the image files, font files, 13.bin , 14.bin , Sound files, CONFIG.txt And other material files and configuration files DWIN_SET Folder to SD Card; then, the SD Card into DGUS Screen SD Card interface, DGUS Screen detected SD After the card will display a blue screen prompts the user to detect SD Card, then according to SD Card file to automate parameter configuration screen or download the data to DGUS Screen FLASH in. SD After the download is complete card, DGUS Screen will automatically reset once the user pull out SD Card, the download is complete.

[Note] twice SD At least the Interval between hot-swappable cards 6 Seconds or DGUS Screen will be considered with a SD Card without data download operation.



Map 3.15 DWIN_SET folder

2. use SD Card Export User Database

in 3.3 Section, and a storage position explained capacity of the user database. So how to export user database? User database write process, DGUS Data will be encrypted and forward error correction (FEC) Operative to ensure the reliability of data storage. Export user database method is as follows:

Step1 : Export the database to calculate the page position revelation suppose you want to export the database first address ADR , The maximum of the two byte address + 256 Is the corresponding SD Card Export page ID .

Step2 : Transcend derived database name and a size equal to a start page position (aligned to 128KB)of DAT file;

The file naming convention " * Font Home location. DAT "

Step3 : take this DAT File into SD Card DWIN_SET Folder, inserted into DGUS Screen SD Card interface, the DGUS

The screen will automatically read out the contents of the specified database covering SD On the card DAT file.

【example】 Suppose you want to export the database space 0x0010 0000 To 0x0017 FFFF Altogether 1M Data, the corresponding SD Card Export page ID for 0x00

10 + 16 + 256 = 256 = 272 . in SD Card DWIN_SET Placing a directory size 1MB The file is named "272 * .DAT" , The SD

Card is inserted into DGUS Screen SD / SDHC Interface can export the contents of the database.

[Note] database export rate of about 180KB / s , So for very large databases, it is recommended into several files to deal with.

3. by SD Card upgrade DGUS version

DGUS Screen by SD Its built-in card DGUS Software liters (lower) level. The DGUS Update (DGUS_V * .BIN) Into SD card DWIN_SET Next to the

directory.

4. SD / SDHC Interface Block and unblock

- **SD / SDHC Interface ban**

After the completion of customer testing mass production, in order to prevent end users SD Wrong card upgrade or download operation resulting in abnormal operation can be achieved

by CONFIG.txt File, add a line of text to prohibit special SD Card interface, those listed in Table 3.11 Fig.

table 3.11 CONFIG.txt Document ban SD Card Interface description text

The first 1 section SD_LOCK_ The first fixed 2 section	
1000	To restart SD Password interface in the variable memory space, the range 0000-6FF8
The first 3 section ABCD1234 Restart SD Card interface 8 Digit password	

【Note】 SD After the card is prohibited, DGUS Screen will not update the data, Information and calibrate the touch screen. Make sure to protect the user SD Card unlock password, because once SD

Interface card lock and unlock password not only by replacing the kernel Depot CPU In order to enable it again.

【example】 Assume ban SD / SDHC After re-enable password interface 12345678 The password stored in the variable storage space 0x6000 Location is prohibited SD / SDHC Step

interfaces are as follows:

Step1 : in CONFIG.txt Increase instruction document SD_LOCK_6000_12345678

Step2 : The CONFIG.txt use SD Card to download DGUS Screen

Step3 : DGUS Automatically disabled SD / SDHC interface

- **SD / SDHC Unlock Interface**

There are three ways to unlock SD / SDHC Interface, still above examples will be described:

method 1 : Send the correct password to the storage location via the serial port, SD The card will be activated once. Transmission command as follows: 5A A5 0B 82 60 00 **3,132,333,435,363,738**

(After which 8 Is a set of numbers corresponding to the password ASCII Code, such as

1 Corresponding 16 Ary ASCII Code 31)

method 2 : Using the touch screen ASCII Text entry function to set a " Unlock " Actions menu can also be activated once SD card.

method 3 : CONFIG.txt Document written cancellation SD Card ban text commands "SD_UNLOCK_ password " , Deposit SD Card to re-activate

SD / SDHC interface. In the above example, the write command for the text SD_UNLOCK_12345678 .

3.5.3 Debugging Tools ED-2 usage of

ED2 Downloader is the essence SD To USB Converter, users can solve the frequent plugging in the development process SD Card download files inconvenient problem. At the same time, it can also be combined DWIN OS Breakpointing simulation. ED2 Download circuit board shown in FIG kind photos 3.16 Fig.

Map 3.16 ED2 Downloader-kind photo

ED2 Download main functions are: 6.25Mbps High-speed downloading fonts and image files; real-time read and write DGUS Register screen; real-time read DGUS The variable screen memory; DWIN OS Online Programming (IAP) And break simulation. **ED2 Not a substitute RS232 or RS485 Carried out a string Communication port, it is only used in place of SD Card to unilaterally DGUS Memory transfer data screen. Its baud rate is fixed 6250000bps With users DGUS Screen-Independent definition of the serial communication baud rate.**

use ED2 Note that the header and baud rate should be subject to the following requirements:

- Serial port baud rate: Fixed 6250000bps , n81 ,no CRC ;
- Fixed header 0x5A A5 .

【example】 Suppose you want to pass ED2 to DGUS Screen 0 Image resolution picture storage space transmission, the following steps:

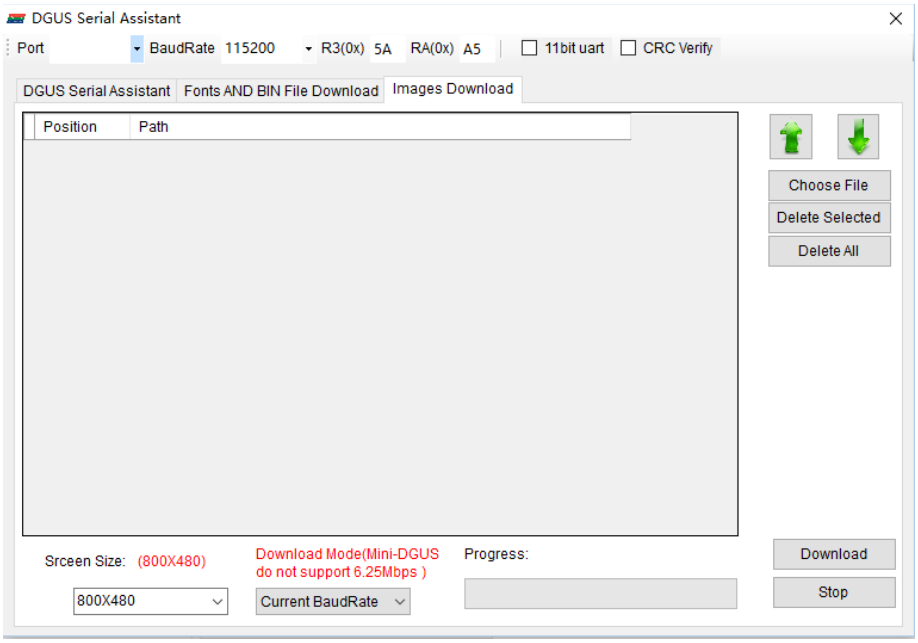
Step1 : Firstly Devine toolbox image conversion tool to convert the picture 0 Beginning with resolution DGUS The same screen bmp File, in this case, the picture is named "0_ Colorful jellyfish. bmp " .

Step2 :use USB Conversion line and by ED2 Computer and will download DGUS Screen connected to ensure ED2 The red indicator light (on behalf of the normal connection) on the downloader.

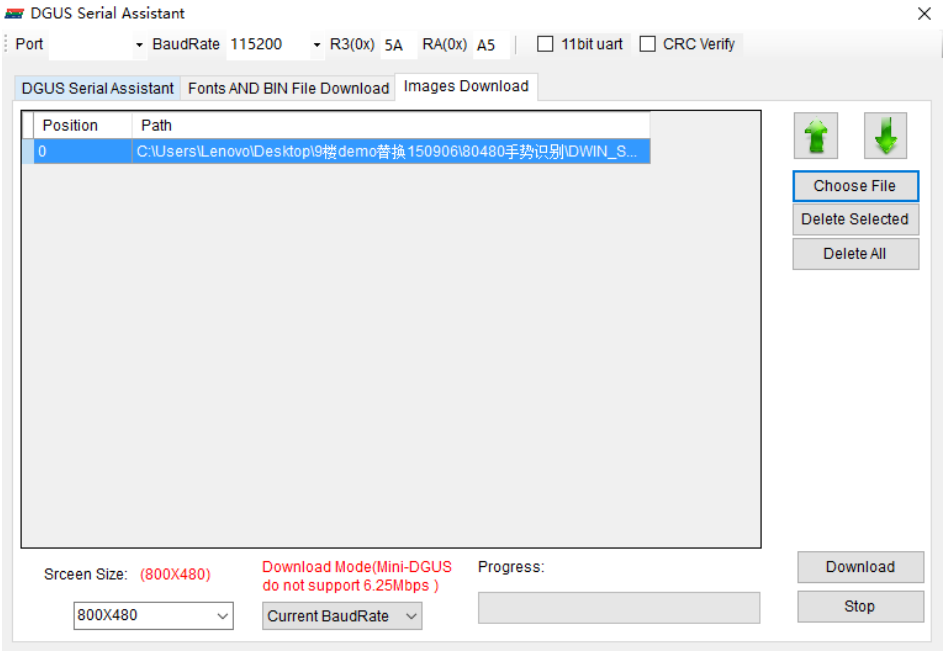
Step3 :turn on DGUS Development software, and click on the toolbar at the top DGUS Serial tool icon, change the parameters as shown 3.17 Shown (which in Port Is determined according to the actual port connected to a computer) and select Images Download Tab.

Step4 : After clicking Choose File And select the converted image, as shown below. among them Position middle 0 Is the storage location of the image, input in the present example 0 .

Step5 : Click the lower right corner Download Buttons, images will be downloaded to DGUS Screen picture memory 0 position.



Map 3.17 Change parameters



Map 3.18 Select the converted picture

Serial communication Chapter IV DGUS screen

DGUS Screen using asynchronous, full-duplex serial port (UART), Serial mode 8n1 , I.e., using ten bits for each data transfer, comprising 1 Start bit, 8 Data bits, 1 Stop bit.

Serial communication may be employed RS232 or RS485 Two means of communication. Serial port baud rate by CONFIG File to configure. All instructions or data are serial 16 Hex (HEX) Format; for font (2 Bytes) of data is always transmitted byte first (MSB) Way, as 0x1234 First transfer 0x12 .

DGUS Screen serial reception FIFO for 4KB , That is 80/120/160 / 200ms (1 More DGUS Period) can be transmitted at least 4KB Data (approximately equal to 230400-691200bps Continuous transmission baud rate); a DGUS The maximum length of the data period can be transmitted depends on the complexity of the user interface; recommend a customer DGUS Do not send more than in the period 4KB Data to DGUS Screen.

4.1 Serial communication condition detecting

When the user to ensure that the wiring is correct after the serial port, via the serial port debugging assistant SSCOM3.2 Performs tests to verify the communication port is normal.

Serial assistant Download:

http://www.dwin.com.cn/supports/doc-download.html

Test methods are as follows:

Step1 : Correct connection as described above, is opened SSCOM3.2 Serial assistants, set the baud rate and other parameters in FIG. 4.1 As shown in (serial port number where your selected according to the actual port connected to the computer).

Map 4.1 Serial communication detection

Step2 : In character string transmission instruction input box 5A A5 03 81 00 01 (This command is used to query DGUS Screen software version number), if received in the receiving window to the top 5A A5 04 81 00 01 ** Response instruction, the communication is normal. (Which is ** DGUS Screen software version number, in this case the software is 15 year 12 Released in February V71 version).

4.2 Communication Instructions

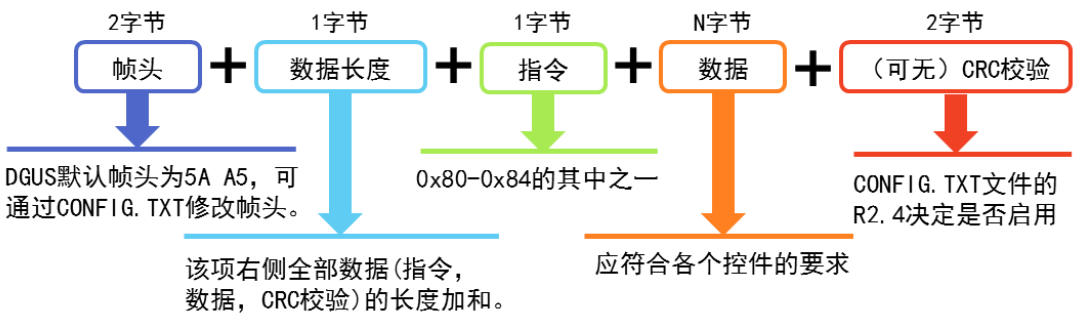
Introduced in the previous chapter DGUS The GUI (Graphical user interface) of each page into a plurality of control variables, i.e. DGUS Screen

Operating mode using variable drive mode operation, and screen GUI Entirely controlled by the state variable data. Therefore, serial command only needs to read the variables, you can write, the instruction set is very simple, a total of only 5 Instructions. Table write instruction set 4.1 Fig.

table 4.1 DGUS Instructions

Features instruction	data		Explanation
Access Storage Device	0x80 Issued: register address (0x00-0xFF) + data input To the specified register address Write data		
	0x81	Issued: register address (0x00-0xFF) + Read byte length(0x00-0xFF)	Starting from the specified register address Read Specifies the length in bytes of the data
		Answer: Register Address (0x00-0xFF) + data byte The read register data length +	to DGUS After sending a read register instruction screen, DGUS Screen response
	DGUS There are screen 256Byte Registers, mainly for the software interface hardware operations, in accordance with byte(Byte) Addressing operation		
Variable access memory (RAM)	0x82 Issued: variable memory address (0x0000-0x6FFF) + write The variable data		Starting from the specified variable memory address Write Data string (word data) to the variable storage area
	0x83	Issued: variable memory address (0x0000-0x6FFF) + Read variable data word length(0x00-0x7F)	Starting at the specified address variable storage area Read Designation word The length of the data word
		Answer: variable Variable data memory address + word The read variable data length +	to DGUS Command variable transmission screen memory after reading, DGUS Screen response
	DGUS There are screen 56KB Variable memory, mainly used for storing variable data, in accordance with word(Word) Addressing operation		
Write slow curve Chong District	0x84 CH_Mode (Byte) + DATA0 (Word) + ... + DATAn		Write buffer data curve: ◇ CH_Mode Each bit (bit)correspond 1 Channels; CH _Mode.0 correspond 0 aisle,. 7 correspond 7 Channel; position corresponding to the channel 1 It represents the channel data is present; is 0 It represents the channel data does not exist. ◇ data arranged in a lower front channel data. Such as: CH_Mode = 0x83 (1 00000 1 1 B) Representing subsequent data format :(aisle 0 + aisle 1 + aisle 7) + ... + (aisle 0+ aisle 1+ aisle 7).
	DGUS There is a screen 16KB Can store 8 Buffer curve trend curve for users to simply, quickly graph. Curve data buffer is 16 Bit unsigned numbers.		

A complete serial command structure as shown in FIG.



Map 4.2 Serial command structure

among them, CRC Checking and data length does not include the header, checksum only for instructions and data. CRC Check uses ANSI CRC-16 (X16 + X15 + X2 + 1)format. When enabled CRC After the frame check and turn on the automatic answering function (R2.4 = 1 , RC.3 = 1), DGUS Screen will CRC

After the automatic response checks of verification is completed, a return instruction is structured as follows:

Header + 02+ (DGUS Receiving screen) + instruction data (0xFF Show CRC Check correctly, 0x00 Show CRC Parity error) + CRC .

【example】 PC via the serial port to DGUS Screen issued 5A A5 03 81 20 07 Instruction, wherein 5A A5 For the header, 03 It represents byte length instruction data, 81 Data read instruction register, 20 First address register is read, 07 Byte length data is read. Function of this instruction is read 0x20 Currently stored in the register RTC Value (year, month, date, day, hour, minute, second). Serial data is returned: 5A A5 0A

81 2007+ current RTC value.

4.3 Troubleshooting common serial communication

When a communication failure, should first determine whether the wiring is correct, especially with RS232 When the serial communication module, make sure the module TXD Mouth and DGUS Screen 232_RX Port is connected, the module RXD Mouth and DGUS Screen 232_TX Port connection, must not be reversed.

4.3.1 DGUS Screen and computer communication failure

Fault 1: Use 10pin Interface, the serial port is not recognized.

10pin Interface with a computer communication requires use of an adapter plate (TTL turn USB) Level conversion is required to install the driver. Download and install the following software Di Wenguan online:

- .Net Framework 2.0

  The software USB Drive operating environment software, select the appropriate version is installed according to the computer system. How to check the computer using 32bit Or 64bit It? On the "My Computer," right-click and select "Properties" in the pop-up window to view.

- USB-to-UART Interface conversion chip driver (XR21V1410 or CP2102)

  Devine There are two converter chip, product selection according to the appropriate driver installed.

If you are unsure of the type of chip models use the purchased product, the software can be installed over all three.

Failure II: PC's serial port is occupied.

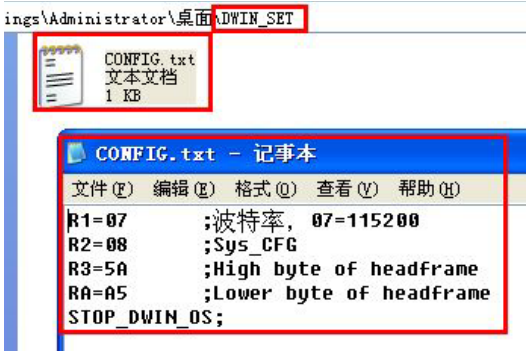
Close other programs and software running on the computer to run the software by terminating the serial port.

Failure three: for while providing TTL with 232 Serial screen, how to switch serial mode?

Short to take over the position of FIG.

Fault IV: connected properly and the computer screen, debug terminal settings are correct, send commands no return value?

This may be due to inconsistencies caused by the baud rate. by CONFIG.TXT The header file baud rate and re-configured factory defaults:



Map 4.3 Set up CONFIG The screen file restore factory settings

will DWIN_SET Folder into SD Under card root directory, to be screen " Down - Card - Power-on " Operation, blue screen 1 Or so seconds, reading SD Cary CONFIG.txt Profile. Then check the communication situation by providing a debug terminal and sends the instructions.

Failure Friday: connected properly and the computer screen, debug terminal is also set correctly, CONFIG Files have been downloaded, still no return value?

This is usually the user to update the kernel unsuccessful due. Please Di Wenguan network to download the latest kernel with SD Card update the kernel. The kernel on DWIN_SET Folder, Copy to SD Card root directory, be on the screen " Down - Card - Power-on " Operation, blue screen 3 Seconds left Right, reading SD Cary kernel. And reconfiguring CONFIG.txt File, again check the communication situation.

Map 4.4 DGUS Upgrade website

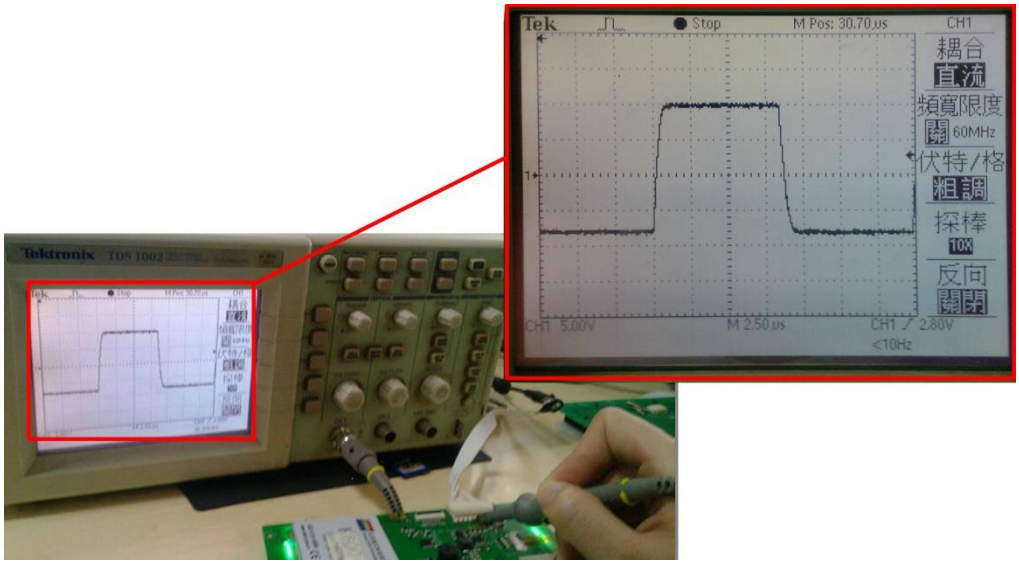
4.3.2 DGUS Screen and microcontroller communication failure

Fault 1: MCU and DGUS Screen baud rate is inconsistent.

First, we should current baud rate detection by the oscilloscope. The screen connected to the computer, in the case of a computer screen and can communicate through debugging **assistant sscom32 each 100ms Sent once send 0xAA . Measured with an oscilloscope needle DIN Pin, square-wave capture, measurement 1 More bit The time value T , 1 / T At this time is** the real baud rate.

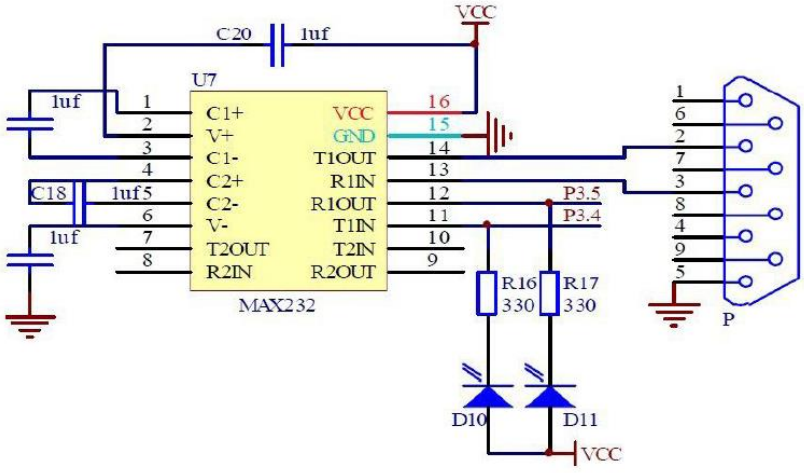
According to FIG. 4.5 The test results can be calculated, 1 More bit The time value T for 2.5μs * 3.5 = 8.75μs , 1 / T * 10^3 for 114 285 Baud rate error for(115200-114285) /115200=0.79% , 0.79% <± 2% And therefore does not affect communication.

If the screen is detected and the microcontroller baud rate is inconsistent, a method described in the employed, by modifying the CONFIG File Quick reset DGUS Screen baud rate.



Map 4.5 Measuring an actual oscilloscope baud

On the other hand, due to the RS232 / TTL Chip (such as MAX232) Bypass capacitor matching, lack of storage capacity, the baud rate may result in an error Large waveform distortion. Recommended 4 Stars 105 Capacitance to solve this problem.



Map 4.6 RS232 / TTL chip

Failure II: SCM and DGUS Screen connection exception.

Check if the wiring is incorrect. This is OK for the MCU TXD connection DGUS Screen DIN , RXD versus DOUT Connection between the two GND Connected to each other.

Failure three: Use 232 turn USB Passive module.

in DGUS And between screens using passive SCM 232 Module, properly set the serial port of the microcontroller to communicate to an end.

Chapter V DGUS screen configuration register

Configuration register space is used to store instruction status, such as RTC Real-time status (real time), the backlight brightness. Understand the function of each register and the address register can be realized with the host computer through the serial port command DGUS Screen information transmission and control. This chapter will help you understand DGUS Screen and the function corresponding to the respective address registers and operation instructions to guide the user through a series of detailed examples, but will

More complex database operations such as a more detailed explanation.

【review】 DGUS provide 256 Byte Configuration register space; 0x80 Instructions for writing to the register, 81 Instructions for reading the register.

5.1 Configuration Register Function Summary

Firstly DGUS Address register and the screen definition table described with reference to an example of a user may attempt to write.

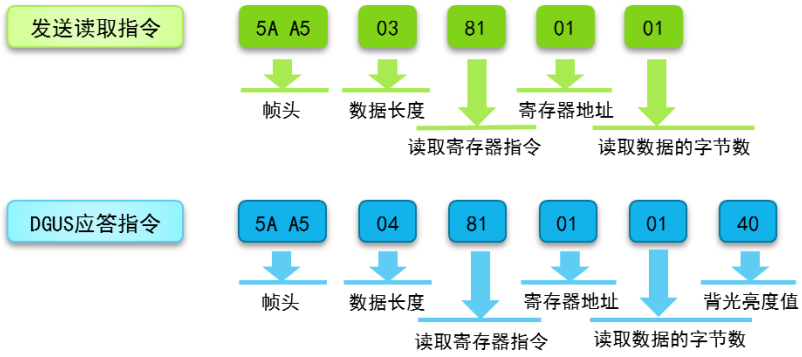
table 5.1 List of Registers

Register Address	definition	R / W Byte	length	Explanation
0x00	Version	R	1	DGUS The version number to BCD Code representation, 0x12 Show V1.2
0x01	LED_SET	W	1	Backlight brightness control memory, range 0x00-0x40
0x02	BZ_TIME	W	1	Buzzer control register unit 10ms
0x03	PIC_ID	R / W	2	Read: the currently displayed page ID Write: switch to the specified ID Page
0x05	TP_Flag	R / W	1	0x5A = Other touch-screen coordinate = touch-screen coordinate update is not updated when the flag is not cleared after reading the user data, the touch screen data is no longer updated.
0x06	TP_Status	R	1	0x01 = First press 0x03 = Press 0x02 = Lift the other invalid =
0x07	TP_Posion	R	4	Touch screen presses coordinate position: X_H: L Y_H: L
0x0B	TPC_Enable	R / W	1	0x00 = Touch does not start Other touch = default power-on startup 0xFF
0x0C-0x0F	RUN_TIME	R	4	Running time of power, BCD Code indicates minutes and seconds, where h is two bytes, the maximum 9999 : 59 : 59
0x10-0x1C	RO-RC	R	13	SD Card configuration mapping registers, read-only serial port, serial port can not be modified
0x1D	Retention		1	Undefined
0x1E	LED_STA	R	1	Return Value backlight brightness
0x1F	RTC_COM_ADJ	W	1	0x5A It indicates that the user has rewritten the application through the serial port RTC data, DGUS modify RTC Cleared upon completion.
0x20	RTC_NOW	R / W	16	Read and write RTC : YY: MM: DD: WW: HH: MM: SS + Lunar calendar YY: MM: DD + Attribution Zodiac +
0x30-0x3F	Retention		16	Undefined
0x40	En_Lib_OP	R / W	1	0x5A Indicates that the user apply font memory read operation, when DGUS After clearing operation is complete. Each DGUS Perform a read cycle.
0x41	Lib_OP_Mode	W	1	0xA0 It represents the read data into the variable space specified font memory space.
0x42	Lib_ID	W	1	Specify the font space, range 0x40-0x7F, Each font 256KB , Corresponding to the maximum Flash space 16MB.
0x43	Lib_Address	W	3	The first data operation specified character space (word) address, range 0x00: 00: 00-0x01: FF: FF
0x46	VP	W	2	The first operation data memory space specified variable (word) address, range 0x00: 00-0x6F: FF
0x48	OP_Length	W	2	Operation data (word) length, range 0x00: 01-0x6F: FF
0x4A	Timer0	R / W	2	16bit Software timer, the unit is 4ms Naturally reduced 0 stop. Between design and actual operating values There +/- 4ms Error, it is better to set 2 Actually run 4-12ms between.
0x4C	Timer1	R / W	1	8bit Software timer, the unit is 4ms Naturally reduced 0 stop.
0x4D	Timer2	R / W	1	8bit Software timer, the unit is 4ms Naturally reduced 0 stop.
0x4E	Timer3	R / W	1	8bit Software timer, the unit is 4ms Naturally reduced 0 stop.
0x4F	Key_code	W	1	User key code, used to trigger 0x13 Touch File: Range 0x01-0xFF ,among them 0x00 It indicates an invalid; DGUS Key code register will be automatically cleared after the key code process.
0x50	Play_Music_Set	W	3	The format 0x5A: Play_Start: Play_Num , Designated to play music. Play_Start Play is the initial segment, Play_Num Play is continuous number of segments (0x00 Will stop playing).
0x53	Volume_Adj	W	2	The format 0x5A: VOL , Adjust the playback volume, volume = VOL / 64 , Power-on default 0x40 .
0x55	Retention		1	Undefined
0x56	En_DBL_OP	R / W	1	0x5A Represents the user application database memory operation, DGUS After clearing operation is complete. Each DGUS Cycles to execute a read or write operation.

0x57	OP_Mode	W	1	0x50: The variable space data to a database memory space. 0xA0: The read data from the database into the variable spatial memory space.
0x58	DBL_Address	W	4	Database space word address range 0x00: 00: 00: 00-1D: FF: FF: FF, maximum 480MW (960MB) Data space. The first database storage space from physical space 64MB Begin storing, and picture memory space there is an overlap, each 1Byte Database occupy memory 2Byte Physical memory. use SD When the card export database, each font size 128KB , Numbered from 236 Start, 960MB When reading and writing, DGUS Automatically handles inter-character situation.
0x5C	VP	W	2	The first operation data memory space specified variable (word) address, range 0x00: 00-0x6F: FF
0x5E	OP_Length	W	2	(Word) length of database operations, range 0x00: 01-0x6F: FF
0x60-0xE8	Retention		137	Undefined
0xE9	Scan_Status	R	1	0x01 = Touch-screen input in state 0x00 = The touch screen is not in a state entry
0xEA	TPCal_Triger	W	1	Write 0x5A After starting a touch screen calibration, the calibration is complete DGUS Cleared.
0xEB	Trendline_Clear	W	1	Defined value is written to clear the specific data buffer corresponding to the curve. 0x55: Clear all 8 Of buffer data curves. 0x56-0x5d: Were cleared CH0-CH7 Curve buffer data channel. After completion of the data buffer cleared curve, DGUS This register will be cleared.
0xEC-0xED	Retention		2	Retention
0xEE-0xEF	Reset_Triger	W	2	Write 0x5AA5 Make DGUS Screen software reset once
0xF0-0xFF	Retention		16	Retention

【example 1] Inquiries and adjusting backlight brightness value.

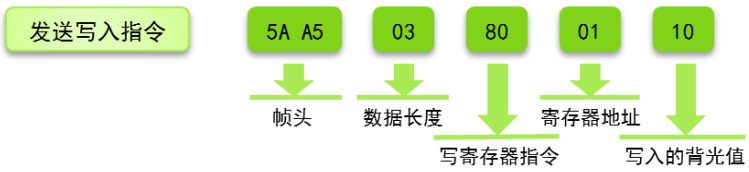
Table 5.1 Found backlight luminance is stored in the address register 0x01 Space. Backlight brightness value by sending the read command, as shown in FIG.



Map 5.1 Sends an instruction to read the screen backlight brightness

Transmitting a query command backlight brightness, received DGUS Screen response is 5A A5 04 81 01 01 40 Most right 0x40 To read data, i.e. the back brightness. DGUS Screen brightness range 0x00-0x40 , You can find that the screen is the brightest state.

When the backlight brightness to be adjusted, a write register instruction may be sent, as shown in FIG. Instruction rightmost 0x10 Backlight brightness is adjusted, instructions may be transmitted to the screen was observed to darken. Can send a query command again, the screen response 5A A5 04 81 01 01 10 Further evidence of brightness modified successfully.

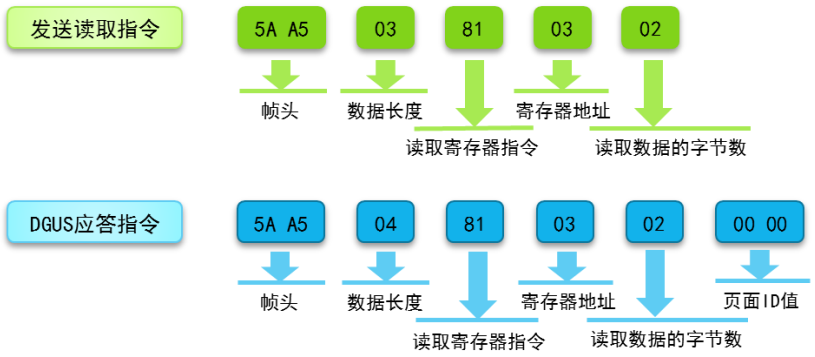


Map 5.2 Adjust the backlight brightness transmission command

[Note] If the system configuration register R2.5 = 1 (On standby and wake on touch screen quench screen function), then adjust the backlight when required by the command 5A A5 03 80 01 10 , 5A A5 03 80 01 3F Transmitted together two instructions.

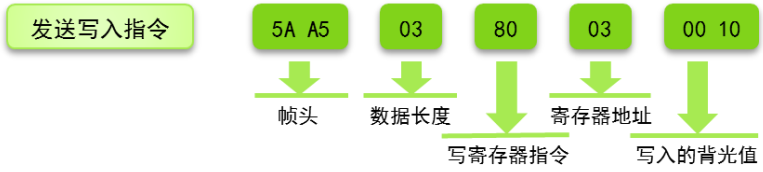
【example 2 Read page] ID And switch pages.

page ID save at 0x03 Register space. Transmission instruction by reading the current page number, as shown below, the currently displayed page ID for 0 .



Map 5.3 Send page to read instructions ID

It may also be implemented by sending an instruction to switch the page, as shown, the switching instruction to the effect of ID for 0x10 Page, that number is 16 Page surface. If not download ID for 16 Page to DGUS Screen, black screen is displayed.



Map 5.4 Page transmission command switch

By reading and writing configuration registers, may also perform various functions:

Started buzzing 5A A5 03 80 02 20 unit 10ms , The longest ringing time FF

Reset 5A A5 04 80 EE 5A A5 It corresponds to the power-down / power-up operation

Touch screen calibration 5A A5 03 80 EA 5A Mini DGUS The command is not supported

Off the touch function 5A A5 03 80 0B 00

Touch function is turned on 5A A5 03 80 0B 01

5.2 Application Examples register

5.2.1 RTC Read and write

1. read RTC

RTC Related Registers Table 5.2 Fig.

table 5.2 RTC Related register

0x1F	RTC_COM_ADJ	W	1	0x5A It indicates that the user has rewritten the application through the serial port RTC data, DGUS modify RTC Cleared upon completion.
0x20	RTC_NOW	R / W	16	Read and write RTC : YY: MM: DD: WW: HH: MM: SS + Lunar calendar YY: MM: DD + Attribution Zodiac +

It is seen from the table, to 0x20 Led addresses 16 Bytes of register space to store the current time, the calendar time stored previous 7 Words Section. To read the Gregorian calendar (month, day, week, minutes and seconds), the assistant may send commands through the serial port: 5A A5 03 81 20 07 , As 5.3 Fig. Screen response is 5A A5 0A 81 20 07 16012501164649 That time is now 16 year 1 month 25 Monday 16:46:49

If the long read time (minutes and seconds), from RTC Storage format learned from twenty four To 26 It is stored in three bytes HH: MM: SS Therefore hair Send instructions 5A A5 03 81 24 03 It can be.

Map 5.5 Sends an instruction to read the current time

2. write RTC

If you find DGUS When the screen is allowed to be calibrated clock, to be RTC Data register write time.

Step1 : Need to send rewrite RTC Application, namely 0x1F Write data register 0x5A , Instructions 5A A5 03 80 1F 5A .

Step2 : To order 0x20 Led addresses RTC Data is written to the register you want to modify. For example, the time is set to 2016 year 1 month

25 day 17 : 08 : 20 Is sent 5A A5 09 80 20 16 01 25 **00** 170 820 .

The two instructions above can also be combined to simplify the operation, i.e., from 1F Write address 5A + Time to be modified, transmitted instructions 5A A5 0A

80 1F 5A 16 01 25 **00** 170 820 .

[Note] write RTC When, only rewritten the date when the minutes and seconds to the Gregorian calendar, DGUS Screen will automatically convert the week and lunar calendar. In the example above, the position data rewrite week casually write to 00 .

5.2.2. Read font

In front of us the space character, the first 24-127 No space character can store user-defined font file and icon library files, which **Only the first 64 to 127 Font memory can be read through the serial port to the command variable (RAM)In** . Read data variable reservoir by 0x82 Read the instructions again.

table 5.3 Font-related registers

0x40	En_Lib_OP R / W 1 Write 0x5A Start reading represents the font data manipulation. when DGUS After clearing operation is complete. Each DGUS Perform a read cycle.	
0x41 Lib_OP_Mode W 1 Write 0xA0 It represents the read data into the variable space specified font memory space.		
0x42	Lib_ID	W 1 Specify the font space, range 0x40-0x7F, Each font 256KB , Corresponding to the maximum Flash space 16MB.
0x43	Lib_Address W 3 The first data operation specified character space (word) address, range 0x00: 00: 00-0x01: FF: FF	
0x46	VP	W 2 The first operation data memory space specified variable (word) address, range 0x00: 00-0x6F: FF
0x48 OP_Length W 2 Operation data (word) length, range 0x00: 01-0x6F: FF		

【example】 From 82 number(0x52) Font 0x000010 Address to start reading 8 KB (8 * 1024/2 = 4096 Word , which is 0x1000 Word) To the first address data 0x1000 of RAM Space.

When the transmission instruction, sequential data to be written into the instruction register can be transmitted. **Data read can not exceed the space character, i.e., Lib_Address + OP_Length <256KB which is 0x02 00 00).**

Send command: **5A A5 0C 80 40 5A A0 52 000 010 1000 1000**

40 : First address register

5A : 0x40 Written 5A Read font represents the start operation

A0 : 0x41 Written A0 It represents the read font data into the RAM space

52 : Font file number

000 010 : From the font file 0x000010 Address to start reading data

1000 : The data read write head address 0x1000 of RAM Space

1000 : Total length of data to be read (unit: word)

5.2.3 Audio Player

section DGUS Screen support for audio playback. By user SD Card music files (32K Sampling frequency, 16bit Mono WAV)download

To the screen, DGUS Automatically store the audio file segment, each maximum 1.024 Second, you can store a total of 128 segment. Support for audio playback

DGUS Screen buttons can achieve sound, voice reminders and other effects. This effect may be achieved through the touch and send instructions in two ways.

If you want to realize key sound, can be DGUS Check Voice software option basic touch controls, and set the audio file ID And that is the number of segments

It may be, as shown in FIG. At this time, the buzzer sound will stop automatically. **Do not play the voice when the number of segments is zero. In audio playback, voice unsaved blank section will directly be skipped.**

Map 5.6 DGUS Software set to play a key speech

by 0x80 The instructions may control audio playback and adjust the volume. Related Registers Table 5.4 Fig.

table 5.4 Audio playback relevant register

0x50 Play_Music_Set W 3 Play the specified music format **0x5A: Play_Start: Play_Num** . **Play_Start** Play is the starting position, **Play_Num** Play is continuous number of segments (0x00 Will stop playing)	
0x53 Volume_Adj	**W 2 Adjust the playback volume, format 0x5A: VOL , Volume = VOL / 64 , Power-on default 0x40 .**

【example】 An audio (such as " Welcome to Beijing Devine ") Length 3.5 Second, from 6 No voice began to save space, the space occupied by voice 6-9 No space for a total of 4 Audio space.

To order 50% Volume to play this music, send commands **5A A5 07 80 50 5A 06 04 5A 20**

50 : Audio playback register 0x50 .

5A : Format requirements, write 5A It can be.

06 : Storage space of the audio starting number.

04 : The number of total occupied space of the audio.

5A : Format requirements, write 5A It can be.

20 :will VOL Value was adjusted to 20 , which is 40 Of the half.

To stop voice playback, just the number of players in the segment play instruction set 0x00 To, i.e., send instructions 5A A5 05 80 50 5A **0600**

To increase the volume to 150% (64 * 1.5 = 96 ,correspond 0x60), Sends an instruction 5A A5 04 80 53 5A 60

5.2.4 Reading and writing database

table 5.5 Register Database related

0x56 En_DBL_OP R / W 1 0x5A Indicating that the startup database read and write operations, DGUS After clearing operation is complete. Each DGUS Cycles to execute a read or write operation.		
0x57	OP_Mode	W 1 0x50 : The variable space data to a database memory space. 0xA0 : The data in the database is read into the variable space memory space.
0x58 DBL_Address W 4 Spatial Database starting (word) address, range 0x0000 0000 ~ 0x1DFF FFFF ,maximum 960MB Data space. each 1Byte Database occupy memory 2Byte FLASH .		
0x5C	VP	W 2 The first operation data memory space specified variable (word) address, range 0x0000-0x6FFF
0x5E OP_Length W 2 (Word) length of database operations, range 0x0001-0x6FFF		

When the read-write database, you first need to know the database FLASH Storage position in space, that is, the first address database. From the explanation FLASH Chapter storage space, we learned that the space can be allocated to the database is capped (see 3.3 Section). Maximum storage allocated to the database When space, corresponding to a minimum of pictures, tables, 5.6 Given parameter table address calculation required for the first database space.

table 5.6 Image ID And storage coefficient K1 Chart

Resolution	320 * 240	480 * 272	640 * 480	800 * 480	800 * 600	768 1024 *	6001024
K1	1	1	3	3	4	5	6
PIC_ID	128	128	42,43	42,43	32	25,26	21,22

PIC_ID : Space allocated to the database maximum number of pictures that can store up to, but also the largest number where images can be used. There are two tables in the partial resolution PIC_ID ,Such as 640 * 480 Resolutions PIC_ID for 42 with 43 , Means that the maximum picture ID for 42 , But because of a picture takes up more space, it should be from the original database 44 No picture storage location to start storing.

Space is provided starting address calculation database N Pictures need to be stored, the database of the smallest first address

= ((N * K1) - 128) * 64 * 1024

Note: N more than the PIC_ID ; 128 , 64 , 1024 Are constants, can be directly into the calculation.

【example】 480 * 272 Under the resolution, set aside 200 Space pictures, then the starting address of the database Adr_Min for:

Adr_Min = ((200 * 1) - 128) * 64 * 1024 = 4718592 = 0x0048 0000 【each 1Byte Database occupy memory 2Byte FLASH 】

use SD When the card export database, each font size 128KB , Numbered from 236 Start. DGUS Automatically handles inter-character situation. Reading and writing process in the user database, DGUS Data will be encrypted and error correction operations, to ensure the reliability of data storage. User Database in FLASH It is the size of a number of 128KB Database structure, each page is written life 10 Million times (start 1 Write operations decreased 1 Times write life), but the address of read and write operations are continuous, not affected by paging, DGUS It will automatically handle the problem paging.

5.2.5 External keyboard control

DGUS No interface screen keyboard, but many applications need to use a keyboard or button operation. DGUS Screen 0x4F Register with Offering users to use the keyboard to control DGUS The touch screen interface file (13 Touch files. BIN) Interface, the user only needs to send commands to write key 0x4F Registers, it may be formed DGUS Accordance with the rules and the touch screen description files to handle the relevant pages. Table registers associated key 5.7 Fig.

table 5.7 Button associated registers

0x4F Key_code W 1 User key code, used to trigger 0x13 Touch File: Range 0x01-0xFF, among them 0x00 Indicate invalid DGUS Key code register will be automatically cleared after the key code process.				

【example】 When the need to implement an external keyboard control, first need to DGUS Attribute set in the software "substantially touch" or "return value key" and other touch keys as a "key " ,As shown below. Can 01 ~ FF Disposed between, may also be assigned 255 Keys. Once set up, control color

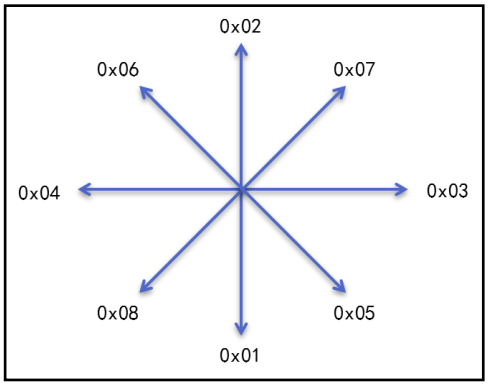
From yellow to gray. Set up keying controls on the screen can not be controlled through the touch of a finger . The need to touch and to achieve key, two controls may be provided in the same place, a set key, a key is not provided, it does not affect the use of two overlapping controls.

Can be achieved by key button command transmission control effect: setting the control key to 0A , Then sends an instruction 5A A5 03 80 4F 0A To real

Now touch the effect of the control. If the current control is switched to the touch effect pages 10 When the command is sent, the page is switched to page 10 .

Map 5.6 in DGUS External buttons control software settings

It should be noted that the key functions 01 to 08 Fixed to gesture recognition key key code, as shown, the screen automatically recognizing a gesture write key 0x4F Register, then if RC.1 = 1 (I.e., open gesture recognition) can be achieved gesture page switching effect.



Map 5.7 Gesture key code map of FIG.

Chapter VI DWIN OS programming

DWIN OS Development platform allows the user to code in part DGUS Running on the screen, allowing the user's secondary development easier, users can take advantage of rich instruction to achieve more complex functions. DWIN OS Platform uses a similar assembler of the preparation of specifications, users can fast and reliable secondary development for their own particular needs.

DWIN OS The common application is to use DWINOS Process to resolve a user-defined data and protocol data, in order to replace registration HMI Or IPC. This not only reduces the cost (DGUS Only standard screen prices HMI Or one-tenth of IPC), but also greatly enhance the reliability of the (standard system HMI Mostly based PC Or industrial architecture, the software is in Windows CE Under a common operating system platform like development).

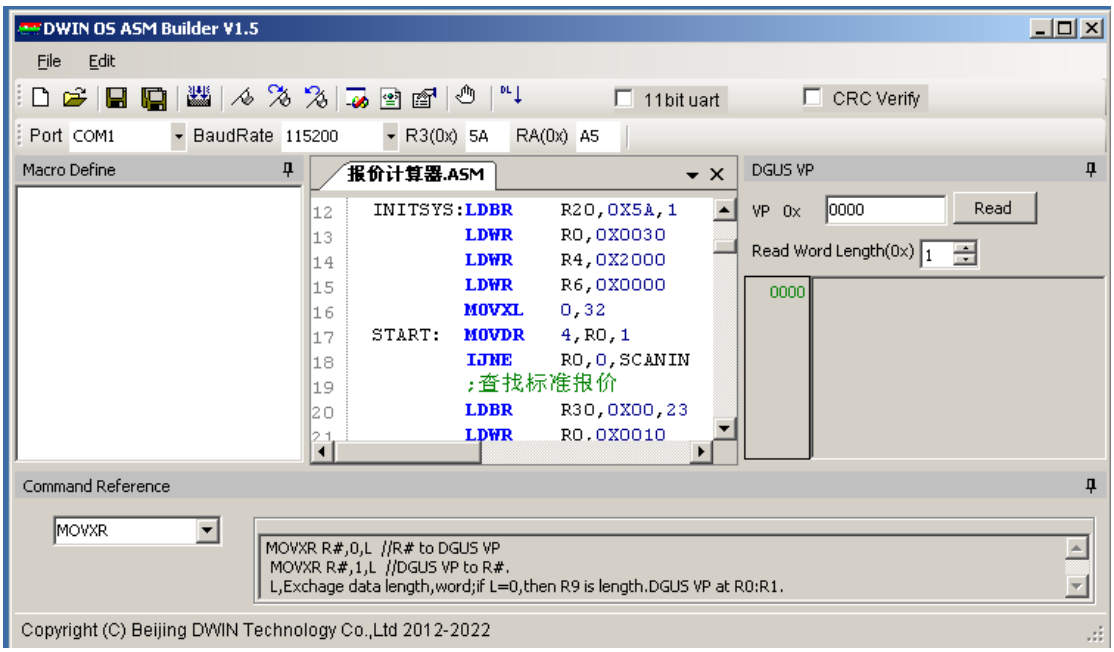
This chapter will introduce users DWIN OS Programming software and the basic programming conventions of the platform and a compilation instruction set, and performed by some examples on in detail.

6.1 DWIN OS Programming Software

DWIN OS Programming Software Download:

http://www.dwin.com.cn/supports/doc-download.html

Compiled interface shown in Figure 6.1 Fig.



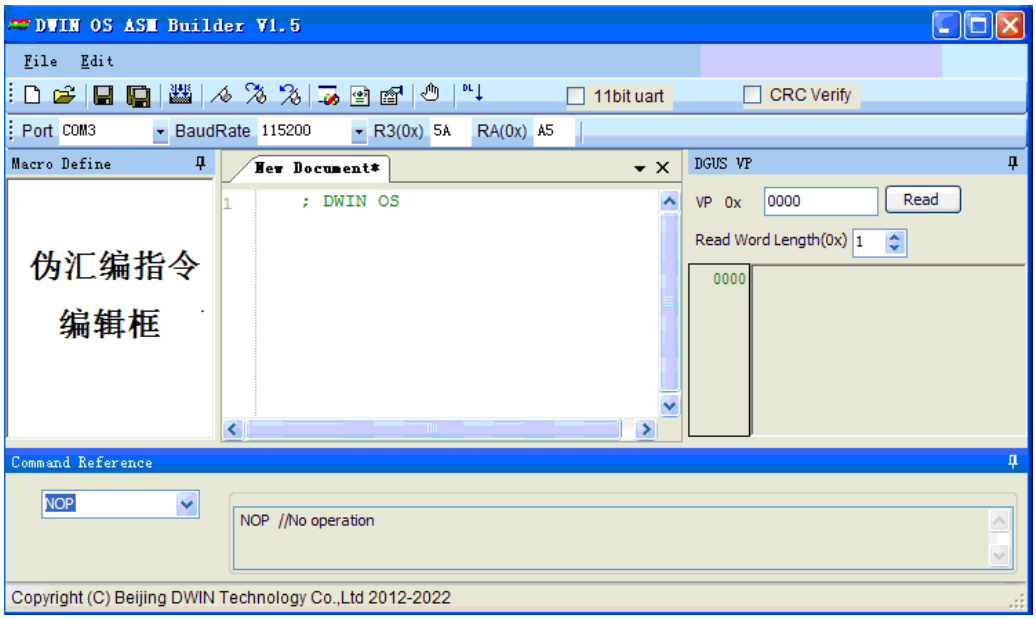
Map 6.1 DWIN OS Compiler Interface

Using the software programming code and click the button on the interface subsequent to the preparation program (Build Button), the software will automatically generate 23 * .BIN document. You need to use the file SD Card to download DGUS Screen, and CONFIG.txt Profile added RUN_DWIN_OS Statement can be realized DWIN OS Program running as 6.2 Fig.

Map 6.2 CONFIG.txt Profile added RUN_DWIN_OS

6.2 The basic agreement

- in DGUS of DWIN OS Platform, code space is the biggest user 256KB (and 32764 Lines of code). DWIN OS In each program DGUS cycle(80/120/160 / 200ms) Are fully operational again, so DWIN OS Program can not be an infinite loop.
- DWIN OS Register variables: R0-R255 A total of 256 A, each register variable 1byte (8bit)
- DGUS Register: corresponds DGUS 0x80 / 0x81 Instruction register space accessed (0x00-0xFF)
- DGUS Variables: correspondence DGUS 0x82 / 0x83 Variable memory space access instruction (0x0000-0x6FFF), Each variable 2byte (16bit)
- Font space: correspondence 32-127 (0x20-0x7F No.) Store, a total of 24MB
- Pseudo assembler instructions: DWIN OS Supporting pseudo assembly instructions, assembly instructions only pseudo DWIN OS of PC Editing software pseudo assembler instruction edit box, as shown in 6.3 Fig.



Map 6.3 Pseudo assembler instruction edit box

The following are some examples concerning pseudo assembler instructions.

【example 1】 EQU The function is to replace, direct replacement when compiling instructions. such as

```
PICID EQU 3

WORD EQU 2

MOVDR PICID, R10, WORD; Equivalent to MOVDR 3, R10,2
```

Note: Comments invalid mark use, use ; .

[Example 2] DB The function is defined 1 Bytes or words (less than the defined data 255 The automatically defined as bytes) ROM data. LDADR TAB1 Function is to TAB1 of 24bit Save to Address R5: R6: R7 Address pointer register, such as:

```
TAB1: DB 1,2,3,4

      DB 1000,2000,3000,4000 -100

      DB " Beijing Devine DGUS "
```

6.3 DWIN OS Assembly instruction set

This section will provide users DWIN OS Assembler command table, in the table, conventions:

- R # Show DWIN OS of 256 Registers in any one or a group R0-R255
- <> It represents an immediate, assembly code, 100 , 0x64 , 64H , 064H They are represented 10 Binary data 100
- COM2 Related instructions and features only serial DGUS + Core hardware platform support.

DWIN OS Table assembler instruction list 6.1 (Table green Examples instruction).

table 6.1 DWIN OS Assembly instruction sheet

Command functions	Opcode	Operand	Explanation
No operation	NOP	Null	Do nothing 【example】 NOP
DGUS Variable and OS Register data exchange	MOVXR R #, <MOD>, <NUM>		R # : OS First address register or register set <MOD> : 0 = Register to variable 1 = Variables to register <NUM> : Exchanging data words (Word) length, 0x00-0x80 When < NUM> for 0x00 When the data length of the R9 Decision. DGUS Variable pointer by the R0: R1 Defined register. 【example】 MOVXR R20,0,2
load 1 More 8bit Immediate to OS Register set	LDBR	R #, <DATA>, <NUM>	R # : OS First address register or register set <DATA> : The data to be loaded <NUM> : Number of registers to be loaded, 0x00 Show 256 More 【example】 LDBR R8,0x82,3
load 1 More 16bit Immediate to OS Register set	LDWR	R #, <DATA>	R # : OS First address register group <DATA> : The data to be loaded 【example】 LDWR R8,1000
Look-up table program space (program space to data register) MOVC		R #, <NUM>	R # : OS First address register or register set <NUM> : Return data byte length lookup table address pointer by the R5: R6: R7 Register Definition 【example】 MOVC R20,10
Load address	LDADR	<Address>	The < Address> Loaded into R5: R6: R7 【example】 LDADR TAB LADAR 0x123456
OS Data exchange between registers	MOV	R # S, R # T, <NUM>	R # S : OS First address register or register set data source R # T : To be passed OS The first group of address register or register <NUM> : The byte length of data to be transmitted 【example】 MOV R10, R20
OS To register DGUS register	MOVRD	R #, D #, <NUM>	R # : OS The first group of address register or register D # : DGUS The first group of address register or register <NUM> Byte length of data to be exchanged: 【example】 MOV R10,2,1
DGUS To register OS register	MOVDR	D #, R #, <NUM>	D # : DGUS The first group of address register or register R # : OS The first group of address register or register <NUM> Byte length of data to be exchanged: 【example】 MOV 3, R10,2
DGUS And font data variables, database exchange	MOVXL	<MOD>, <NUM>	<MOD> : 0 = Font data read DGUS variable 1 = DGUS Variable data is written font 2 = Read user database DGUS variable 3 = DGUS Variable data written to the user database <NUM> : Exchanging data word length; NUM = 0 By R9: R10 definition DGUS By the variable address R0: R1 Read-write mode register defines the font (MOD = 0,1) : Number of fonts R4 (0x20-0x7F) Register designated, R5: R6: R7 Operation of header data in the font address, bounds user database read-write mode is canceled (MOD = 2,3) : The first address by a user database R4: R5: R6: R7 Register Definition 【example】 MOVXL 0,300

DGUS The exchange of data between variables	MOVXX	<NUM>	<NUM> : Word data to be exchanged (Word) length <NUM> for 0 Represented by the length of R8: R9 Register Definition DGUS Variable by the source address R0: R1 Register Definition DGUS Address by the target variable R2: R3 Register Definition 【example】 MOVXX 100
Register Indexed	MOVA		R2 The predetermined source register (s) Address R3 It specifies the destination register (s) Address R9 It specifies the number of data bytes to be transferred 【example】 MOVA
32bit Plastic additions	ADD	R # A, R # B, R # C	C = A + B , A with B for 32bit Integer, C for 64bit Integer 【example】 ADD R10, R20, R30
32bit Integers subtraction	SUB	R # A, R # B, R # C	C = AB , A with B for 32bit Integer, C for 64bit Integer 【example】 SUB R10, R20, R30
64bit Long integer MAC	MAC	R # A, R # B, R # C	C = (A * B + C) , A with B Yes 32bit Integer, C Yes 64bit Integer 【example】 MAC R10, R20.R30
64bit Long integer division	DIV	R # A, R # B, <MOD>	A / B The calculated results into commercial A , The remainder is stored BA with B They are 64bit register <MOD> : 0 = Suppliers not be rounded up 1 = Business rounding 【example】 DIV R10, R20,1
Square root calculation	SQRT	R # A, R # B	A calculation 64bit Unsigned R # A And save it to the square root R # B in R # A : Saved 8 Byte unsigned number R # B : Saved 4 Byte unsigned results 【example】 SQRT R80, R90

Variable extended to 32bit	EXP	R # S, R # T, <MOD>	The R # S Converted into data points 32bit And save it to an integer R # TR # S :source
			OS Register (group)
			R # T : 32bit aims OS Register set
			<MOD> : R # S type of data, 0 = 8bit Unsigned 1 = 8bit Signed
			2 = 16bit Unsigned 3 = 16bit Signed
			【example】 EXP R10, R20,2

32bit Unsigned MAC	SMAC	R # A, R # B, R # C	C = (A * B + C) , A with B Yes 16bit Unsigned, C Yes 32bit Unsigned 【example】 SMAC R10, R20.R30
OS Since the increment register	INC	R #, <MOD>, <NUM>	R # = R # + NUM , Unsigned incremented calculated < NUM> for 0x00-0xFF <MOD> : R # Data types 0 = 8bit 1 = 16bit 【example】 INC R10,1,5
OS Register from reductions	DEC	R #, <MOD>, <NUM>	R # = R # -NUM , Unsigned decrement calculation, < NUM> for 0x00-0xFF <MOD> : R # Data types 0 = 8bit 1 = 16bit 【example】 DEC R10,0,1
AND logic operation	AND	R # A, R # B, <NUM>	R # A, R #B : OS The first group of address register or register NUM : Calculating the byte length of data A = A AND B , And the logic operation sequence 【example】 AND R10, R20,1
Or logic operation	OR	R # A, R # B, <NUM>	R # A, R #B : OS The first group of address register or register NUM : Calculating the byte length of data A = A OR B , Sequence or logic operation 【example】 OR R10, R20,1
XOR operation	XOR	R # A, R # B, <NUM>	R # A, R #B : OS The first group of address register or register NUM : Calculating the byte length of data A = A XOR B , XOR operation sequence 【example】 XOR R10, R20,1
Solutions integer linear equation	ROOTLE		by 16bit Integer (X0, Y0)with(X1, Y1) On a line determined by two points X To calculate the corresponding Y Input value: X = R16 , X0 = R20 , Y0 = R22 , X1 = R24 , Y1 = R26 Output: Y = R18 【example】 ROOTLE

ANSI CRC-16 Compute	CRCA	R # S, R # T, R # N	Calculation of sequence data ANSI CRC-16 (X16 + X15 + X2 + 1) R # S: To calculate CRC The data in OS The first group of address register **R # T: Calculated CRC result(16bit) , LSB (LSB first) save** R # N: Involved in the calculation of data bytes in the specified register (8bit) **【example】 CRCA R10, R80, R9**
CCITT CRC-16 Compute	CRCC	R # S, R # T, R # N	Calculation of sequence data CCITT CRC-16 (X16 + X12 + X5 + 1) R # S: To calculate CRC The data in OS The first group of address register **R # T: Calculated CRC result(16bit) , MSB (The previous high) save** R # N: Involved in the calculation of data bytes in the specified register (8bit) **【example】 CRCC R10, R80, R9**
Cumulative byte checksum calculation SUMADD		R # S, R # T, R # N	And a check byte calculating cumulative data sequence **R # S :Input OS Register set** **R # T : Output 1 And byte accumulation result, 8bit R # N :** The sequence length register, 8bit **【example】 SUMADD R10, R80, R9**
Accumulation into byte checksum calculation	SUMADDC	R # S, R # T, R # N	Computing sequence of bytes into the data bits and parity accumulation **R # S :Input OS Register set** **R # T : Output 1 And byte accumulation result, 8bit R # N :** The sequence length register, 8bit **【example】 SUMADDC R10, R80, R9**
XOR checksum calculation	SUMXOR	R # S, R # T, R # N	Sequence data byte XOR checksum calculation **R # S :Input OS Register set** **R # T : Output 1 Byte XOR result, 8bit R # N : The** sequence length register, 8bit **【example】 SUMADDC R10, R80, R9**
Bit resolution	BITS	R #, <VP>	**The R # of 8 Decomposed into bits VP Points 8 More DGUS Word variable, MSB (High byte first) embodiment. bit 0 Decomposed into 0x0000 , bit 1 Decomposed into 0x0001 <VP> : DGUS Variable address** **【example】 BITS R10,0x2000**
Bit combinations	BITI	R #, <VP>	**The VP Points 8 More DGUS Word variables into 1 Variable bit bytes, MSB (High byte first) embodiment. 0x0000 for bit 0 , 0x0001 for bit 1 <VP> : DGUS Variable address** **【example】 BITI R10,0x2000**
Sequence comparison	TESTS	R # A, R # B, <NUM>	**Compare turn A versus B Two OS Register value is not the same sequence, return A At this time, the sequence of addresses R0 If the register A versus B The same is returned 0x00 To R0 register** **R # A : A Sequence OS register** **R # B : B Sequence OS register** <NUM> : Comparison of the maximum number of data bytes **【example】 TESTS R10, R20,16**
Bit test and jump	JB	R #, <Bit>, <TAB>	**test R # The first register < Bit> Position, 1 Jump, 0 Executes the next instruction, the jump range of +/- 127 Instructions** **R # : Bit Test OS register, 16bit <Bit> : Bit test position, 0x00-0x0F , MSB (High** byte first) embodiment <TAB> : Jump position **【example】 JB R10,15, TEST1** **NOP TEST1: ADD R8, R12, R16**
Compare variables are not equal jump	CJNE	R # A, R # B, <TAB>	**Compare A with B Two 8bit Contents of the register, the next instruction is equal, not equal, Branch range of +/- 127** Instructions **R # A , R # B : To be compared OS register** <TAB> : Jump position **【example】 TEST1: NOP** **INC R10,0,1 CJNE R10, R11,** **TEST1**

Comparative integers, smaller than the jump	JS	R # A, R # B, <TAB>	Compare A with B Two 16bit Integer size, A> B Next instruction is executed, A <B Jump, jump range of +/- 127 Instructions R # A , R # B : To be compared OS First address register group <TAB> : Jump position 【example】 JS R10, R12, TEST1 NOP TEST1: NOP
Variable and immediate comparison is not equal jump	IJNE	R #, <INST>, <TAB>	Compare 8bit Register, immediate < INST> Content equal to the next instruction is executed, jump if not equal, a jump range of +/- 127 Instructions 【example】 IJNE R10,25, TEST1 NOP TEST1: NOP
Jump directly	GOTO	<PC>	Program jump 【example】 GOTO TEST1 NOP TEST1: NOP
Force the end of the current input method	EXIT	R # A, R # B	Forced to end the current input method R # A : Control whether to switch pages, 0x00 = Not switch 0x01 = Switch R # B : To switch: To switch the page ID (16bit) 【example】 LDBR R10,1 LDWR R11,110 EXIT R10, R11
The current input to the read content OS register	SCAN	R #, <NUM>	The current input has been entered up to < NUM> Characters loaded into R # + 1 Started OS Register, R # Store data length, the number of characters from the current cursor position before bound calculation input 【example】 SCAN R20,6
Input buffer added	SCANADD	R # A, R # B	The text input state, the R # A Pointed to by a byte length R # B Defined character string to the input buffer from the current cursor position, the string 0xFF It will be ignored. This command is valid only when the text input state. 【example】 SCANADD R80, R90
Subroutine call	CALL	<PC>	Subroutine calls, supports up to 32 Nesting level program <PC> : To be called subroutines 【example】 CALL TEST
Subroutine call return	RET		CALL Call return instruction 【example】 RET
COM Configuration	COMSET	<MODE>, <BS>	<MODE> : Serial mode set high 4bit Select the serial port to be configured, 0 = COM1 1 = COM2 low 4bit Select mode, 0x * 0 = N81 mode 0x * 1 = E81 mode 0x * 2 = 081 mode 0x * 3 = N82 mode <BS> : For baud rate setting COM1 Set value = 6250000 / The baud rate set for COM2 , Baud rate set in the high byte selection factor, 00 = 5.208mBPS 01 = 15.625Mbps 02 = 1.302Mbps Low byte = baud rate set value factor / baud rate to be set each time the setting is automatically cleared serial receive FIFO once 【example】 COMSET 0,54
Serial port	COMTXD	<COM>, R # S, R # N	Send data to the specified serial port <COM> : Select the serial port, 0 = COM1 (DGUS Users serial port) 1 = COM2 (DGUS Reserved Serial) R # S : Data to be sent OS Register set R # N : Byte count register to be transmitted, 8bit , Register data 0x00 Send representation 256 Byte data 【example】 COMTXD 0, R10, R9
Serial Print	CPRTS	<COM>, <VP>	an examination VP Points DGUS There are no print variable address information, it is printed to the serial port. VP for DGUS of 0xFE07_05 Printing instruction corresponding to the variable VP Value, after printing cleared VP Print tag address <COM> : Select the serial port, 0 = COM1 (DGUS Users serial port) 1 = COM2 (DGUS Reserved Serial) 【example】 CPRTS 0,0x2000

Direct serial port	COMTXI	<COM>, R #, <NUM>	The R # Directed < NUM> More OS Sent to the contents of the register COM <COM> : Select the serial port, 0 = COM1 (DGUS Users serial port) 1 = COM2 (DGUS Reserved Serial) 【example】 COMTXI 0, R10,16
an examination COM_Rx_FIFO	RDXLEN	<COM>, R #	return COM Receive Buffer (FIFO) Receiving of data bytes (0-253) To R # register, 0x00 No data is <COM> : Select the serial port, 0 = COM1 (DGUS Users serial port) 1 = COM2 (DGUS Reserved Serial) 【example】 RDXLEN 0, R10
Read COM_Rx_FIFO	RDXDAT	<COM>, R # A, R # B	From COM Receive Buffer (FIFO) Read R # B Bytes (1-253) Of data to R # A Register set, read the FIFO Automatically adjusting the length of the <COM> : Select the serial port, 0 = COM1 (DGUS Users serial port) 1 = COM2 (DGUS Reserved Serial) 【example】 RDXDAT 0, R11, R10
To COM1_Rx_FIFO Read MODBUS Data Frame	RMODBUS	R # A, R # T, R # C	an examination COM1 receive FIFO Is there to meet the requirements of MODBUS Data frame, if data is read OS And flushes the receive register FIFO . R # A :save MODBUS Receiving a packet data before 3 Byte (Address: Instruction: length) OS Register set. If the length of 0x00 , It indicates no match length, followed by data (4 Bytes) indicates the data length of the address, command, checksum and outside. R # C : Return status OS Register, which holds the returned data, 0x00 It represents a match is not received MODBUS Data Frame, 0xFF Indicates that the received matching MODBUS And the data frame to the data read R # T Register set. R # T : After the match, save MODBUS data OS Register set. 【example】 RMODBUS R10, R20, R13
To COM1_Rx_FIFO Read DL / T645 Data Frame RD645		R # A, R # T, R # C	an examination COM1 receive FIFO Is there to meet the requirements of DL / T645 Data frame, if data is read OS And flushes the receive register FIFO . R # A :save 6 Byte address (LSB Endian order, compression BCD Codes OS Register set. R # C : Return status OS Register, which holds the returned data, 0x00 It represents a match is not received MODBUS Data Frame, 0xFF Indicates that the received matching DL / T645 And the data frame to the data read R # T Register set. R # T : Results register, matching, stored DL / T645 data OS Register group, the data format is: control code data + data length + 【example】 RD645 R10, R20, R16
Dynamic curve designated write channel buffer	WRLINE	R # S, R # I, <CH>	The R # S Points N More 16bit Unsigned integer plus 16bit Unsigned integer V_BIAS After wrote < CH> (0x00-0x07) Dynamic curve designated buffer R # I = NR # I + 1: R # I + 2 = V_BIAS 【example】 LDBR R12,24,1 LDWR R13,100 WRLINE R20, R12,1
HEX turn ASC	HEXASC	R # S, R # T, <MOD>	R # S : Need to convert 32bit Integer R # T : Converted ASCII String register set <MOD> : Conversion mode control, high 4bit It is an integer bit length, low 4bit Conversion of the number of decimal places ASCII Rope symbols, right-aligned, with gap 0x20 For filling data 0x12345678 , <MOD> = 0x62 Converts the result to + 054198.96 <MOD> = 0xF2 Converts the result to + 3,054,198.96 【example】 HEXASC R20, R30,0X62
HEX Converted into a compressed BCD code	HEXBCD	R # S, R # T, <MOD>	The HEX Compressed data into BCD Code, such as data 1000 Will be converted into 0x10,0x00 R # S : Enter HEX data OS First address register group R # T : Compressed output BCD Codes OS First address register group <MOD> :high 4bit It represents the input HEX The number of bytes, 0x01-0x08 low 4bit It represents the output BCD Code number of bytes, 0x01-0x0A 【example】 HEXBCD R10, R80,0x23

compression BCD Code into HEX	BCDHEX	R # S, R # T, <MOD>	The compressed BCD Code into HEX Data, such as data 0x1000 Will be converted into 0x3E8 R # S : Enter Compression BCD Codes OS First address register group R # T : Output HEX data OS First address register group <MOD> :high 4bit It represents the input BCD Code number of bytes, 0x01-0x08 low 4bit It represents the output HEX The number of bytes, 0x01-0x0A 【example】 BCDHEX R10, R80,0x32
ASCII String turn HEX ASCHEX		R # S, R # T, <LEN>	The ASCII String to convert 64bit Signed HEX data R # S : Enter ASCII Strings OS First address register R # T : Output HEX data, 64bit of OS register <LEN> : ASCII The length of the string data, comprising a sign bit and a decimal point, 0x01-0x15 【example】 ASCHEX R10, R80,0x05
Chinese character library search match	LIBSCH	R # A, R # B, R # C	Chinese Character string searching for matching to the specified data R # A Started 2 Matching a predetermined string register format R # A : String length (0x00-0x1F) , 0x00 Represented by a string 0x00 or 0xFF end R # A + 1 : String data, most 31 Characters R # B Started 11 Register specifies the database (DATA [M] [N] Word Array) format and a searchable format R # B : Font ID (0x20-0x7F, 0x00 Representation without reloading), each DGUS After the cycle is called once, as long as no other character base for operations done it without reloading R # B + 1 : Row dimension two-dimensional array M , 0x0001-0x0FFFF R # B + 3 : Two-dimensional array of column dimensions N , 0x01-0x80 The number of words R # B + 4 : Pattern matching search, 0x00 = Left 0x01 = Anywhere matches R # B + 5 : Search start line, 0x0000-0xFFFF R # B + 7 : Searching each row start columns, 0x00-0xFF , A byte position in the R # B + 8 : Searching each row of termination columns, 0x00-0xFF Only when the search pattern matching is 0x01 (When an arbitrary position matching) effective R # B + 9 : After the match search, the search returns the starting position of data columns, 0x00-0xFF R # B + 10 : After a number of data bytes match, return, 0x01-0xFF R # C Started 4 Register specifies the variable return R # C : Search tags, 0x00 = Unmatched 0xFF = match R # C + 1 : (Row dimension in matching + 1) value R # C + 3 : Match, return the saved data first address register 【example】 LIBSCH R10, R12, R23
Erase the specified font	ERASE	<L_ID>	<L_ID> : To erase the character base ID , 0x20-0x7F If < L_ID> for 0x00 , Represented by the character position R9 Register specification 【example】 ERASE 40
FEC coding	FECEN	R # A, R # BR # C	Correct R # A Byte length is directed R # C The data string FEC Coding, coding output saved R # B The pointer position. FEC Coding will 1 Bytes of raw data is coded 2 Byte code data. 【example】 FECEN R80, R100, R10
FEC decoding	FECDE	R # A, R # BR # C	Correct R # A Word length is directed R # C The data string FEC Decoding, decoded output is stored in R # B The pointer position. 【example】 FECDE R80, R100, R10

Time calculation function	TIME	R # A, R # B, <MOD>	R # A with R # B :save 6 Byte time variable OS Register, the time variable BCD format
			MOD = 0 Calculated A = AB Calculate the relative value between the two times. Claim A> B , when A <B Not calculated and returned R # A The first byte 0xFF MOD = 1 Calculated A = B-RTC MOD = 2 Calculated A = RTC-B
			【example】 TIME R0, R10,0

Increased variable display	ADDL14	R # A, R # B, <MOD>	R # A :save 1 Article display variables (32 Bytes) OS register R # B : The Add position variable, 0x00-0x1F , Can add up to 32 A display variable <MOD> : 0x5A = Adding to the specified location = delete other designated location at this time R # A Undefined 【example】 ADDL14 R80, R79,0x5A
End of program	END		DWIN OS Program run end command 【example】 END

6.4 DWIN OS Programming example

This section will explain commonly used instructions over a few examples.

【example 1】 Suppose you want a variable address 0x0010 The data is read into R10 Started 4 A register code

```
LDWR R0 , 0x0001          ; After execution R0 = 0x00 , R1 = 0x01 Specify DGUS Variable starting address 0x0001
MOVXR R10 , 1 , 2         ; After the implementation of the DGUS From the variable address 0x0001 Since the length of 2 Words ( 4 Byte) data read
                                R10 : R11 : R12 : R13
```

To R10 Started 4 A transmission data register to the address 0x0010,0x0011 of DGUS Variables, code

```
LDWR R0 , 0x0010          ; Designation DGUS Variable starting address 0x0010
LDWR R10 , 0x0001         ; will 0x0001 Storage to R10: R11 in
LDWR R12 , 0x0002         ; will 0x0002 Storage to R12: R13 in
MOVXR R10 , 0 , 2         ; will R10: R11: R12: R13 Of length 2 Words of data
                                (In this case 0x00010002 ) Sent to the address 0x0010 with 0x0011 of DGUS Variable.
```

[Example 2] Suppose you want to pass MOVRD Instructions implement buzzer, in the 4 chapter 4.1 Section learned 0x02 As the buzzer control register. The code

```
LDBR R10 , 10 , 1
MOVRD R10 , 2 , 1         ; After performing buzzer 100ms
```

To pass MOVDR Instruction reads the current page number, 0x03 For the page ID register. The code

```
MOVDR 3 , R10 , 2         ; After execution R10: R11 Is the current page number. If the current is the first 2 Page, R10 = 0x00 , R11 = 0x02
```

[Example 3] Suppose you want to calculate 35 + 11 Value, we need to use ADD instruction. 35 of 16 Hex is 0x00000023 , 11 of 16 Hex is 0x0000000B The symbol

```
LDBR R10 , 0 , 16         ; After execution R10 ~ R25 = 0
LDBR R13 , 35 , 1         ; After execution R13 = 0x23
LDWR R17 , 11             ; After execution R17 = 0x0B
ADD R10 , R14 , R18       ; After execution R18 ~ R24 = 0 , R25 = 0x2E , R18: R19: R20: R21: R22: R23: R24: R25 = 46 To calculate the result
```

To calculate 257-122 Values, needs SUB instruction, 257 of 16 Hex is 0x00000101 , 122 of 16 Hex is 0x0000007A The symbol

```
LDBR R10 , 0 , 16         ; After execution R10 ~ R25 = 0
LDBR R12 , 1 , 2          ; After execution R12 = 0x01, R13 = 0x01
LDBR R17 , 122 , 1        ; After execution R17 = 0x7A
SUB R10 , R14 , R18       ; After the execution, R18 ~ R24 = 0 , R25 = 0x87 , R18: R19: R20: R21: R22: R23: R24: R25 = 135
                                To calculate the result
```

To calculate 7/3 = 2 ... .1, Need to use DIV Instruction code

```
LDBR R10 , 0 , 16         ; After execution R10 ~ R25 = 0
```

```
LDBR R17 , 7 , 1                ; After execution R17 = 0x07
LDBR R25 , 3 , 1                ; After execution R25 = 0x03
DIV R10 , R18 , 0               ; After execution R10 ~ R16 = 0 , R17 = 0x02 ; R18 ~ 24 = 0 , R25 = 0x01
```

[Example 4] Assume use DB Pseudo-command defines two groups ROM Data are as follows

```
TAB1: DB 0xFF , 0xFE
TAB2: DB 0xFFFE , 0xFDFC
```

by MOVC with LDADR Instruction load can be achieved and address look-up table function. Instructions

```
LDADR TAB1                      ; The TAB1 Defined in the first byte data is loaded into the address R5: R6: R7
MOVC R10 , 2                    ; After execution R10 = 0xFF , R11 = 0xFE
LDADR TAB2                      ; The TAB2 As defined in the first address of word data is loaded into R5: R6: R7
INC R7 , 0 , 2                  ; Adding an offset
MOVC R10 , 1                    ; After execution R10 = 0xFD
```

[Example 5] Suppose you want 0x5A Bitwise decomposition, the need to use BITS instruction. Code

```
LDBR R10 , 0x5A , 1             ; R10 = 0101 1010
BITS    R10 , 0x0010            ; After the execution, 0010 = 0x0000,0011 = 0x0001,0012 = 0x0000,0013 = 0x0001,0014 = 0x0001,
                                0015 = 0x0000, 0016 = 0x0001 , 0017 = 0x0000
```

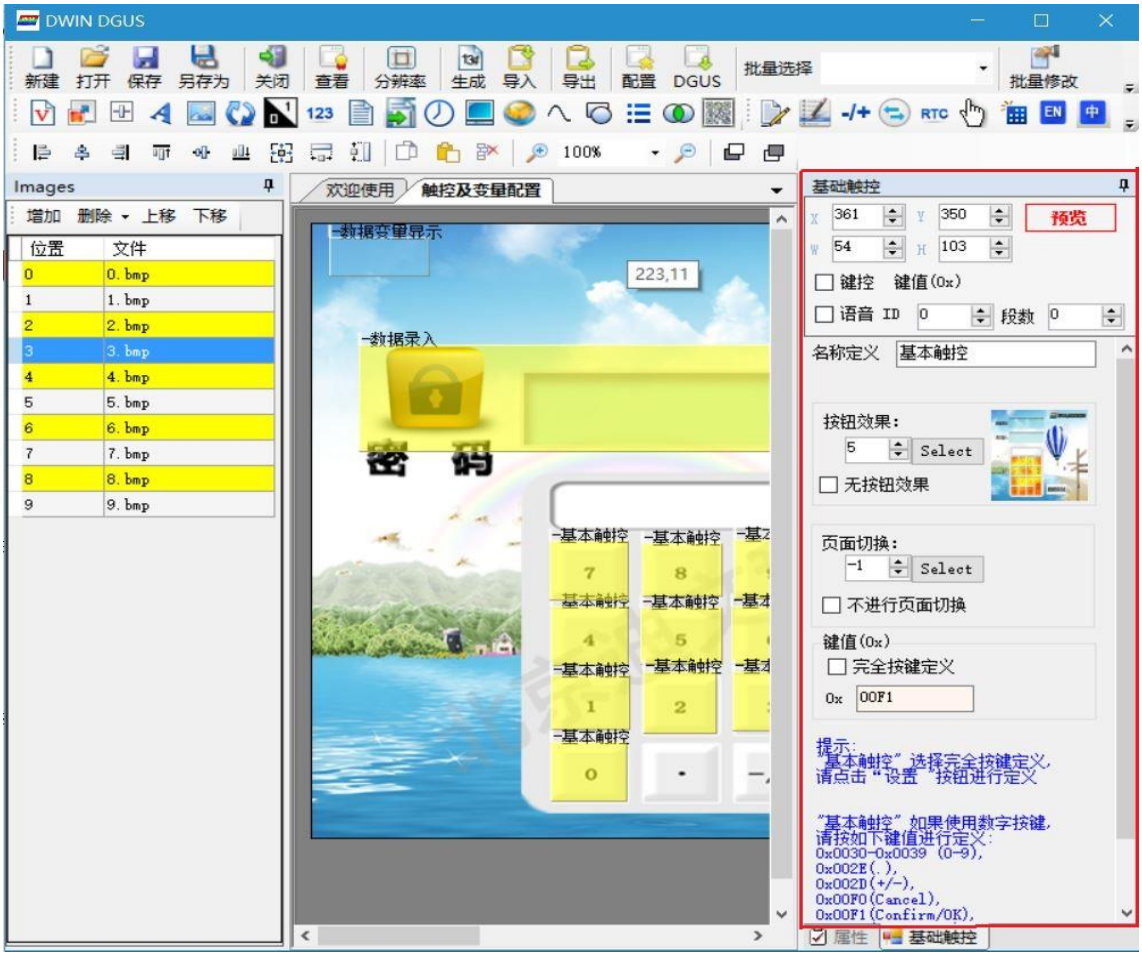
[Example 6] Suppose you want to calculate 2016 year 2 month 13 day 9:13:21 versus 2016 year 2 month 14 day 10:20:22 The time difference between the need to use TIME instruction. Code

```
LDBR R10 , 0x16 , 1             ; Time A Save to R10 Started 6 More OS register
LDBR R11 , 0x02 , 1
LDBR R12 , 0x14 , 1
LDBR R13 , 0x10 , 1
LDBR R14 , 0x20 , 1
LDBR R15 , 0x22 , 1
LDBR R20 , 0x16 , 1             ; Time B Save to R20 Started 6 More OS register
LDBR R21 , 0x02 , 1
LDBR R22 , 0x13 , 1
LDBR R23 , 0x09 , 1
LDBR R24 , 0x13 , 1
LDBR R25 , 0x21 , 1
TIME R10 , R20                  ; A> B After execution R10 Started 6 More OS Time difference stored in the register
```

Chapter VII Touch / Push Profile Description

in DGUS Development and design screen, touch / keyboard and display Configuration variables are key to the design. This chapter will touch / keyboard configuration file (13.BIN) A detailed description and explanation.

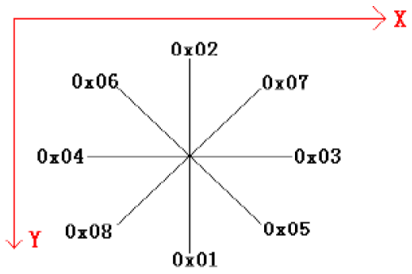
Touch / keyboard configuration file can be used PC End of DGUS Very easy set up software development and production. For example, to add a touch basic functions at a position on the page, just click on the icon, and select a trigger range in the designated area of the picture frame, and then the right side of the pop-up interface touch base setup menu (Fig. 7.1 Red box) on the touch function is provided can be completed simply as 7.1 Fig.



Map 7.1 Touch base settings

In order to facilitate a better understanding of the user, the way of thinking of this chapter will be instructions on the touch / keyboard configuration file more in-depth explanation. Touch with Set font file is stored in the space, it is N Article in accordance with the configuration of the touch instruction page composition, each occupying a fixed touch-control commands 16 , 32 or 48 Bytes of storage space. An instruction by a touch 6 Components, as shown in Table 7.1 Fig.

table 7.1 Composition of a touch-control commands

No. definition	Data bytes	length	Explanation
1	Pic_ID	2	page ID
2	TP_Area	8	 The touch button area may be its upper-left corner coordinates (Xs, Ys) And lower right coordinates (Xe, Ye)to make sure. when Xs = 0xFFFF (Start button voice sound is 0x5FFF When), represented by the trigger controller 0x4F Trigger keycode register, at this time Ys_H (which is Ys High byte address) is set trigger key code values (Ys_L , Xe , Ye The value can be written); When triggered by key code values, the effect is set to an invalid button press. Touchscreen gestures fixed identification key code 0x01-0x08 (Correspondence relationship between the direction and key code as shown above), is automatically written to the identified gesture 0x4F Register is enabled by the register RC.1 To control. (RC.1 = 1 When turned on the touch screen gesture recognition). when Xs = 0x5 *** , It indicates that the key audio using voice prompts instead. One 127 Start of section respectively in the sound Xe , Ye height of 4 Bit player in several segments saved Ys height of 4 Bit (0x00-0x0F Show 0x01-0x10 , When the number of segments to play 0x00 When expressed stop the current voice playback). Once the voice sound button is enabled, the key buzzer will sound off.
3	Pic_Next	2	After pressing the button to switch the operation target page, 0xFF ** It means no page switch.
4	Pic_On	2	Button presses the effect of where the page, 0xFF ** That there is no button press effect.
5	TP_Code	2	Touch key code: 0xFF ** It indicates an invalid key code. 0xFE ** or 0xFD ** Represent the touch function keys, such as 0xFE 00 Indicating that the startup digital touch screen input. among them 00 A touch key code value, a reference list may touch / keying features. 0xFE ** The function keys may be made R2.3 Whether to automatically upload and set it to change variables (R2.3 = 0 When the touch screen does not automatically uploaded after logging parameters, R2.3 = 1 Whether automatically upload determined by the configuration of the respective touch variable), 0xFD Automatically upload the variable function keys always change prohibited. Other key code represent the touch, with ASCII Representation; for example, 0x0031 Indicates key "1" .
6	TP_FUN	32	when TP_Code = 0xFE ** When, for function keys on the touch will be described.

The following describes the various functions for the user touch / keying, and some of the commonly used functions divided explain in detail.

7.1 Touch / keying List of Functions

Touch / keyboard configuration file that allows DGUS Screen enables several functions, such as table function 7.2 Fig.

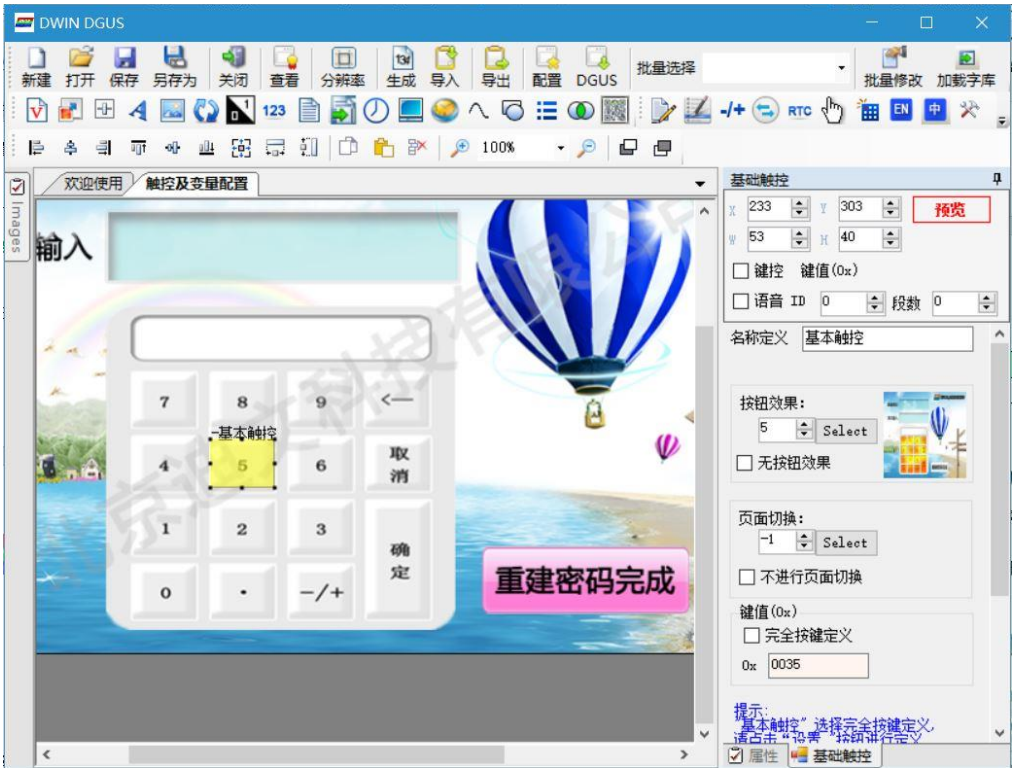
table 7.2 DGUS Screen functions implemented

No. Touch key code		Features	Explanation
01	00	Digital Input	Input integer, decimal point, and other data to the specified variable storage space.
02	01	Pop-up menu	Click trigger a pop-up menu, back key code menu item.
03	02	Upregulation	Click, +/- variables specified operation, and upper and lower limit setting step. Set up 0-1 Cycle range adjustment, variable display can be achieved with the icon column box features such as one click " Select " Click again " cancel " .
04	03	Drag adjustment	Drag the slider to achieve variable data entry, the scale range can be provided.
05	04	RTC Set up	DGUS Touch screen keyboard settings RTC We need to complete entry Gregorian calendar (year, month, day, hour, minute, second)
06	05	Back key value	Click the button to return directly to the key values to variables support level variable returns.
07	06	Text entry	Use a variety of characters in text entry, entry process support for cursor movement and editing. stand by ASCII character, GBK Traditional Chinese phonetic input method and input; and modify fonts 0 # Font can support all similar ASCII Characters 8bit Coded text entry; complex DWIN OS can be realised Unicode Or mixed language input.
08	07_00	Register writes variable space	The method of rewriting the register space provided by a touch screen, the indirect control hardware.
09	07_01	Wrote register variable space	The read register variables such as a backlight, and written back to the manipulated variable to adjust the backlight brightness.
10	07_02	Images into monochrome bitmap (longitudinal)	Converting color bitmap into a specified area stored in the monochrome bitmap and VP Specified variable area.
11	07_05	Images into monochrome bitmap (lateral)	The main screen shows the printout for the current content.
12	07_03	Send data to COM1	Click on the touch screen, the designated VP Transmitting serial data to the user area (COM1).
13	07_04	Send data to COM2	Click on the touch screen, the designated VP Transmitting serial data to the user area (COM2). COM2 For DGUS Screen extensions, and not drawn. .
14	07_06	Send touch screen coordinates to COM2	Click on the touch screen, will send the click position coordinates to the Extended SIO (COM2).
15	08	Touch screen presses status data is returned	Click on the touch screen to return to variable data or serial port as required.
16	09	Turn the adjustment	Turn the knob to realize variable data entry, the scale range can be provided.

7.2 Touch screen data entry

Data entry, i.e., the touch screen input data to the specified variable storage space. Input data consists of numbers, characters and Chinese characters. These data obviously can not be stored directly in the variable storage space (such as character e Need to be converted to the corresponding ASCII code 0x45), And thus needs to be mapped to a key value corresponding with the contents of the touch keypad input.

【example】 A numeric keypad are mapped to each numeric key corresponding to ASCII Code, first with PC end DGUS Add to develop software in the key area of the keyboard page basic touch that box is checked and click on figure 5 Key area in the right menu key (0x) Input 0035 (will 5 defined as ASCII code 0x35). Figure 7.2 Fig.



Map 7.2 Based touch key code provided

Define the remaining keys Similarly, to complete the definition of the keyboard.

7.2.1. Digital Input

That input digital input integer, decimal point, and other data to the specified variable storage space. Storing the instruction format shown in Table 7-3 Fig.

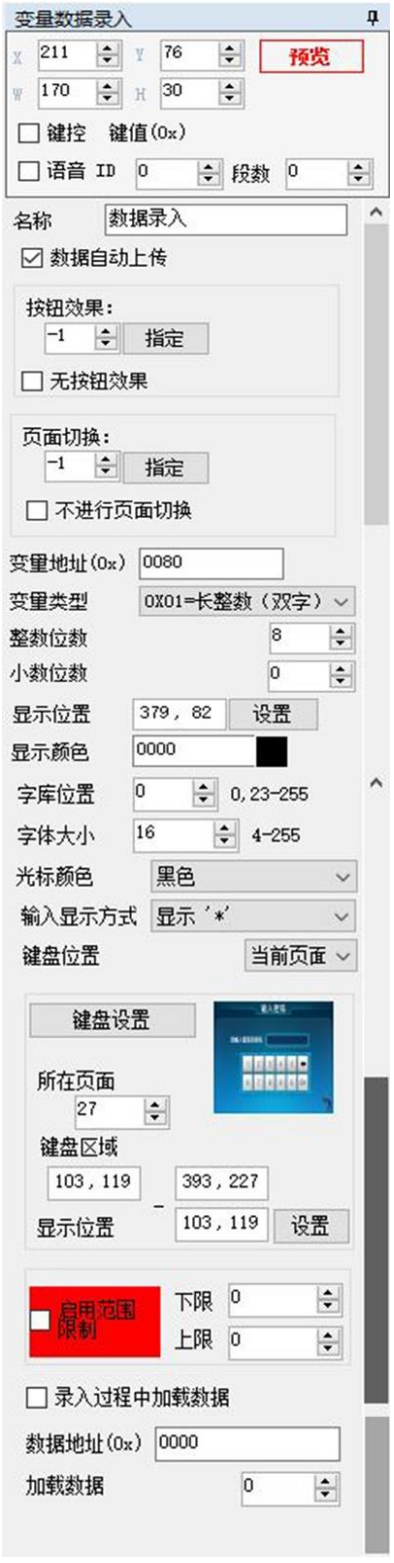
table 7.3 Input instructions stored digital format

address	definition	Data length	Explanation
0x00	Pic_ID	2	page ID
0x02	TP_Area	8	Touch button area: the upper left corner coordinates (Xs, Ys), The lower right corner coordinates (Xe, Ye).
0x0A	Pic_Next	2	Target switch pages, 0xFF ** It means no page switch.
0x0C	Pic_On	2	Button press renderings where the page, 0xFF ** That there is no button press effect.
0x0E	TP_Code	2	The touch key code value 0xFE00 , Indicating that the startup digital touch-screen input.
0x10	0xFE	1	Fixed value 0xFE
0x11	* VP	2	Variable input data corresponding to the address pointer
0x13	V_Type	1	Return types with arguments: 0x00 Show 2 Byte variables: Integer: - 32768 To 32767 ; Unsigned integer: 0 To 65536 0x01 Show 4 Byte variable:

			Long: - 2147483648 To 2147483647 Unsigned long integer: 0 To 4294967295 0x02 * Denote VP High byte, unsigned: 0 To 255 0x03 * Denote VP Low byte unsigned number: 0 To 255 0x04 Show 8 Byte long integer: - 9223372036854775808 To 9223372036854775807
0x14	N_Int	1	Entry integer digits. As entry 1234.56 ,then N_Int = 0x04 .
0x15	N_Dot	1	Entry of decimal places. As entry 1234.56 ,then N_Int = 0x02 .
0x16	(x , y)	4	The position of the input process: right-alignment, (x , y) Coordinates of the top right corner of the input string cursor.
0x1A	Color	2	Enter the display font color.
0x1C	Lib_ID	1	The display used ASCII Font position, 0x00 The default font.
0x1D	Font_Hor	1	font size, x The number of lattice direction
0x1E	Cusor_Color	1	Cursor color, 0x00 He represents black, or white.
0x1F	Hide_En	1	0x00 It represents the text entry process is not directly displayed, is displayed as "*"; directly process the input display contents to any value.
0x20	0xFE	1	Fixed value 0xFE
0x21	KB_Source	1	0x00 It indicates that the keyboard in the current page; other value indicates that the keyboard is no longer the current page.
0x22	PIC_KB	2	Keyboard page where ID Only when KB_Source not equal to 0x00 Valid.
0x24	AREA_KB	8	The keyboard area coordinates: the coordinates of the upper left corner (Xs, Ys) , The lower right corner coordinates (Xe, Ye) Valid only in the keyboard when not in the current page, that is, KB_Source not equal to 0x00 .
0x2C AREA_KB_Position		4	Keyboard display left corner position of the current page, is effective only when the keyboard is not in the current page.
0x30	0XFE	1	Fixed value 0XFE
0x31	Limite_En	1	0xFF : Indicates Enable input range limits, bounds an invalid input (canceled equivalents); denotes an input range no limit to other values.
0x32	V_min	4	Enter the lower limit, 4 Byte (long or unsigned long).
0x36	V_max	4	Enter the upper limit, 4 Byte (long or unsigned long).
0x3A	Return_Set	1	0x5A : Entry process to Return_VP address(0x3B)load Return_Data (address 0x3D Content), automatic recovery after the end. 0x00: Not load the data entry process. Load data features: mainly used for variables and display SP (Description Pointer) modify binding, multi-parameter automatic indication of the entry process, such as changing font color, size, start a (bit) variable region reverse color or icon. It can be used as a marker bit entry process, with DWIN_OS Development to achieve specific needs.
0x3B	Return_VP	2	Loading the data entry process VP address.
0x3D	Return_Data	2	Entry process to load Return_VP The data.
0x3F	Retention	1	write 0x00

[Note] Digital input key code is valid 0x0030-0x0039 , 0x002E (.) , 0x002D (+/-) , 0x00F0 (cancel), 0x00F1 (determine), 0x00F2 (backspace)

Digital input function simply by PC end DGUS Development of software. turn on DGUS Software, and click the button on the page. Next, a marquee area with the mouse, the function can be provided in the menu on the right. Figure 7.3 Fig.



Map 7.3 Digital input function set

Range setting region : Set the touch button area;

Preview : View the touch of a button effect;

Name Definition : Set a name for the button, easy access in the "Variable View.";

Data is automatically uploaded : When checked, the data will be uploaded to the serial port;

Page switching : Specifies the handover to the target picture;

Button effect : Button press renderings page where (- 1 Default is no pressing effect).

Variable address : Definition data storage address;

Variable Types : 0x00 = Integer (word);

0x01 = Integer (Gemini);

0x02 = Unsigned byte parameters (variable address high byte);

0x03 = Unsigned byte parameters (variable address low byte);

0x04 = Long integers (8 byte);

Integer digits : Data entry integer digits;

Decimal places : Data entry of decimal places;

Show location : Display position data input process;

Color display : Enter the font color process, you can manually fill in;

Font position :to show ASC II font position, "0" Show 0 Font.

font size : X The number of lattice direction;

The cursor color :black White;

Input Display : Direct display / display is "*" ;

Keyboard position : Other pages / current page.

Keyboard Settings : Set the keyboard and the keyboard area where the page;

Page where : Keyboard page where;

Keyboard area : Keyboard area of the page on the keyboard;

Show location : Keyboard at the current position of the page (when the keyboard when this page is not);

Enable range limit : When checked prescribed upper and lower limits of the digital input (out of range can not be entered). (**Range limit Bit integer value plus decimal places in the range, for example, set up 3 integer bits, 2 Decimal places, the The value for the upper limit 10000 instead of 100 .**)

[Note] data can also be entered through the "data variables, word variables Art" and the displayed data.

7.2.2 Text entry

Text entry contains ASCII Code and GBK Chinese character input, characters and key code in accordance with the standard ASCII Code definitions, ASCII Table code list 7.4 Fig.

table 7.4 AS CII Code list

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

[Note] Low table lowercase key code byte, high-byte code key caps. Such as 0x61 correspond a , 0x41 correspond A . Another example 0x31 correspond 1 , 0x21 correspond!.

Text keyboard key code is defined to be less than 0x80 (ASCII code). 0x0D Input key code is automatically converted to 0x0D 0x0A ; 0x00 with 0xFF key Code is disabled. In addition, some keyboard function keys, key code as defined in Table 7.5 Fig.

table 7.5 Function keys, key code definition

Keycode	definition	Explanation
0x00F0	Cancel	Cancel entry to return, does not affect the variable data.
0x00F1	Return	Confirm entry returned, save input variable to specify the location of text.
0x00F2	Backspace	Forward (Backspace) to delete a character.
0x00F3	Delete	To delete a character backward.
0x00F4	CapsLock	caps Lock. If enabled, the corresponding button is pressed the button of the effect must be defined.
0x00F7	Left	Cursor forward one character; GBK Chinese character input for the flip.
0x00F8	Right	After the cursor one character; GBK Chinese character input for the flip.

Following are descriptions of ASCII Code entry and GBK Chinese character input.

1) ASCII Code text entry

ASCII Code instructions stored entry format as shown in Table 7.6 Fig.

And similar digital input, ASCII The same code text entry by simply PC end DGUS Development of software. turn on DGUS Software, and click the button on the page. Next, a marquee area with the mouse, the function can be provided in the menu on the right. A method is also provided similar to the digital input, not repeat it here. **After entering the text by "text display" function displays the data.**

[Note] Devine preinstalled 0 # Font contains 4 --- 64 * 128 * 8 All dot matrix ASCII Code characters.

table 7.6 ASCII Input instruction code storage format

address	definition	Data length	Explanation
0x00	Pic_ID	2	page ID
0x02	TP_Area	8	Touch button area :(Xs, Ys), (Xe, Ye)
0x0A	Pic_Next	2	Target switch pages, 0xFF ** That there is no button press effect.
0X0C	Pic_On	2	Button press renderings where the page, 0xFF ** That there is no pressing effect.
0x0E	TP_Code	2	0xFE06 (I.e., text entry touch key code).
0x10	0xFE	1	0xFE
0x11	* VP	2	Variable address pointer
0x13	VP_Len_Max	1	The maximum length of variable text, word (Word) Number, in the range of 0x01-0x7B; When you save text to the specified address will be automatically added at the end of the text 0xFFFF As a terminator; entry of text variables may actually occupy the largest space variable = VP_Len_Max + 1 .
0x14	Scan_Mode	1	Input Mode Control: 0x00 = Re-entry, 0x01 = Open to amend the original text.
0x15	Lib_ID	1	Display to be used ASCII Font position, 0x00 = The default font.
0x16	Font_Hor	1	font size, X The number of lattice direction
0x17	Font_Ver	1	font size, Y The number of lattice direction (when using the default font, Y The number of lattice direction must be X Twice the number of lattice direction).
0x18	Cusor_Color	1	Cursor color, 0x00 = Black, the other white =
0x19	Color	2	Text color display
0x1B	Scan_Area_Start	4	Input text display area of the upper left corner coordinates (Xs, Ys)
0x1F	Scan_Return_Mode	1	
0x20	0xFE	1	0xFE
0x21	Scan_Area_End	4	Text entry area the lower right corner coordinates (Xe, Ye)
0x25	KB_Source	1	Keyboard page location options: 0x00 = Keyboard on the current page; other = keyboard is not in the current page.
0x26	PIC_KB	2	The page where the keyboard (the keyboard is not effective when the current page)
0x28	Area_KB	8	:(keyboard area coordinates when the keyboard where the page Xs, Ys), (Xe, Ye) (When the keyboard is not currently active page)
0x30	0xFE	1	0xFE
0x31	AREA_KB_Position	4	When the keyboard is not in the current page, the coordinates of the location of the upper left corner of the keyboard display.
0x35	Display_EN	1	0x00 = During normal input display; 0x01 = Input process is displayed as "*" for password entry.
0x36	NULL	10	write 0x00

2) GBK Chinese text entry

GBK Characters entered instructions stored format shown in Table 7.7 Fig. After the text entry required by the "text display" function displays the data.

【Note】

- pinyin "Bd" All correspondence GBK Encoding full-width punctuation entry
- Devine preinstalled 0 # Font contains 4 * 128 * 8--64 All dot matrix ASCII Character code
- Not using the touch screen while using the keyboard (0x4F Key code storage register, see section 4 chapter 4.6 Section) do GBK When entry, must be 0x01--0x08 Key code to select the corresponding characters.
- Key (low byte) as shown in Table phonetic input method 7.8 Define (phonetic input method is mainly used in Taiwan).

table 7.7 GBK Character input instruction storage format

address	definition	Data length	Explanation
0x00	Pic_ID	2	page ID
0x02	TP_Area	8	Touch button area :(Xs, Ys), (Xe, Ye)
0x0A	Pix_Next	2	Target switch pages, 0xFF ** It means no page switch.
0x0C	Pic_On	2	In which the page button press renderings, 0xFF ** That there is no button press effect.
0x0E	TP_Code	2	0xFE06 (I.e., text entry touch key code).
0x10	0xFE	1	0xFE
0x11	* VP	2	Variable address pointer
0x13	VP_Len_Max	1	The maximum length of variable text, word (word)number, 0x01-0x7B; When you save text to the pointer address, will be automatically added at the end of the text 0xFFFF As a terminator;

			The maximum variable space entry of text variables may actually occupied as follows: VP_Len_Max + 1 .
0x14	Scan_Mode	1	Input Mode Control: 0x00 = Re-entry; 0x01 = Open Text has been modified again.
0x15	Lib_GBK1	1	Kanji character display used GBK Font ID , ASCII Character is used by default 0x00 Font.
0x16	Lib_GBK2	1	Entry process used by the Chinese character display GBK Font ID .
0x17	Font_Scale1	1	Lib_GBK1 Font size, the number of dot-matrix
0x18	Font_Scale2	1	Lib_GBK2 Font size, the number of dot-matrix
0x19	Cusor_Color	1	Cursor color, 0x00 = Black, the other white =
0x1A	Color0	2	Display color of the text entry.
0x1C	Color1	2	Display color entry process of the Chinese version.
0x1E	PY_Disp_Mode	1	Entry process, and a corresponding Chinese character alphabet tips display mode: * 0x00 = Pinyin hints are displayed at the top, corresponding kanji separate row below; pinyin character display prompts and left-aligned, line spacing Scan_Dis . * 0x01 = Pinyin prompt displayed on the left, to the right of the corresponding character display; start display character tips x Location: Scan1_Area_Start + 3 X Font_Scale2 + Scan_Dis .
0x1F	Scan_Return_Mode	1	0xAA = in*(VP-1) Location of the saved data input and an effective length of the end marker. * (VP-1) High byte input end flag: 0x5A It represents the input end, 0x00 Still represents input. * (VP-1) Low byte, the effective length of the input data, in units of bytes. 0xFF = And not return to the input end of the data length mark.
0x20	0xFE	1	0xFE
0x21	Scan0_Area_Start	4	Input text display area of the upper left corner coordinates (Xs, Ys).
0x25	Scan0_Area_End	4	Input text display area of the lower right corner coordinates (Xe, Ye).
0x29	Scan1_Area_Start	4	Pinyin text entry process prompted the upper left corner coordinates of the area.
0x2D	Scan_Dis	1	Entry process, the pitch of each character displayed. Per line up to display 8 Chinese characters.
0x2E	0x00	1	0x00
0x2F	KB_Source	1	Keyboard page location options: 0x00 = Keyboard on the current page; other = keyboard is not in the current page.
0x30	0xFE	1	0xFE
0x31	PIC_KB	2	Keyboard page where ID . (Only when the keyboard is not valid when the current page)
0x33	Area_KB	8	Keyboard Keyboard area where the coordinates of the page :(Xs, Ys); (Xe, Ye).
0x3B	Area_KB_Position	4	When the keyboard is not in the current page, in the upper left corner of the keyboard coordinate the currently displayed page.
0x3F	Scan_Mode	1	0x02 = Pinyin input method 0x03 = Phonetic input method (Taiwan Traditional entry).

table 7.8 Phonetic input method key code

注音	ㄅ	ㄆ	ㄇ	ㄈ	ㄉ	ㄊ	ㄋ	ㄌ	ㄍ	ㄎ	ㄏ	ㄐ
键码	0xc5	0xc6	0xc7	0xc8	0xc9	0xca	0xcb	0xcc	0xcd	0xce	0xcf	0xd0
注音	ㄑ	ㄒ	ㄓ	ㄔ	ㄕ	ㄖ	ㄗ	ㄘ	ㄙ	ㄧ	ㄨ	ㄩ
键码	0xd1	0xd2	0xd3	0xd4	0xd5	0xd6	0xd7	0xd8	0xd9	0xe7	0xe8	0xe9
注音	ㄚ	ㄛ	ㄜ	ㄝ	ㄞ	ㄟ	ㄠ	ㄡ	ㄢ	ㄣ	ㄤ	ㄥ
键码	0xda	0xdb	0xdc	0xdd	0xde	0xdf	0xe0	0xe1	0xe2	0xe3	0xe4	0xe5
注音	ㄦ	一	ˊ	ˇ	ˋ	˙						
键码	0xe6	0x99	0x40	0x98	0x41	0x42						

And similar digital input, GBK Chinese text entry can also be simply by PC end DGUS Development of software. turn on DGUS Software, and click the button on the page.

Next, a marquee area with the mouse, the function can be provided in the menu on the right. The method of the digital input provided similar, not repeat it here.

7.3 Pop-up menu

Click the pop-up menu to select the function that is to trigger a pop-up menu and back key code menu item. Storing the instruction format shown in Table 7.9 Fig.

table 7.9 Pop-up menu function instruction storage format

address	definition	Data length	Explanation
0x00	Pic_ID	2	page ID
0x02	TP_Area	8	Touch button area :(Xs, Ys), (Xe, Ye)
0x0A	Pic_Next	2	Target switch pages, 0xFF ** It means no page switch.
0x0C	Pic_On	2	In which the page button press renderings, 0xFF ** That there is no button press effect.
0x0E	TP_Code	2	0xFE01 (Click trigger a pop-up menu, back key code menu item.)
0x10	0xFE	1	0xFE
ox11	* VP	2	Variable address pointer return data has VP_Mode Decision.
0x13	VP_Mode	1	0x00 = The 0x00 ** Key code is written VP Word address (integer type); 0x01 = The 0x ** Key code is written VP High byte address of the word (VP_H); 0x02 = The 0x ** Key code is written VP Low address byte from the address (VP_L); 0x10-0x1F : ** key code to the least significant bit (1bit) Variable and write VP The specified bit word address (0x10 modify VP.0 , 0x1F modify VP.F)
0x14	Pic_Menu	2	Photos Location pop-up menu.
0x16	Area_Menu	8	Pop-up menu areas: the upper left corner coordinates (Xs, Ys) , Lower right coordinates (Xe, Ye).
0x1E	Menu_Position_X	2	Menu page is displayed in the upper left corner of the current position X coordinate.
0x20	0xFE	1	0xFE
0x21	Menu_Position_Y	2	Menu page is displayed in the upper left corner of the current position Y coordinate.
0x23	NULL	13	write 0x00

[Note] valid key code input process: 0x0000-0x00FF ,among them 0x00FF is canceled (ie no parameters are directly returned).

【example】 Suppose you want to design a syringe selection system, when you click the drop-down menu arrow will pop up menu, as 7.4 Fig.

Map 7.4 Syringe selection pop-up menu

Map 7.4 The pop-up menu on the other interface, the key range of configuration options button menu is 0x0000-0x00FF , Wherein when a button is clicked, the corresponding key code will be returned to the variable pointed to the pop-up menu. After the selection, you can set the selected item is displayed in the text box.

usable PC end DGUS Development software simply implement the functionality. turn on DGUS Software, and click the button on the page. Next, a marquee area with the mouse, the function can be provided in the menu on the right. Figure 7.5 Fig.

Map 7.5 Syringe pop-up menu selection system configuration

Gamut setting: the touch button area;

Preview: View performance touch of a button;

Name Definition: set a name for the button, " Variables View " Conveniently query;

Automatically uploaded data: When checked, the data will be uploaded to the serial port;

Button effect: a button press effect diagram where the page (- 1 Default is no animation).

Variable Address: definition data storage address;

Variable type: 0x00 = Key code is written VP Word address;

0x01 = Key code is written VP The high byte of the address (VP_H);

0x02 = Key code is written VP The low byte of the address (VP_L);

0x10-0x1F = Key code least significant bit (1bit) Variable Write VP Word address

The specified bit (0x01 modify VP.0 , 0x1F modify VP.F);

Pop-up menu: Set pop-up menu where the page;

Page: to be pop-up menu in the area of the page;

Show Location: Location pop-up menu to be displayed in the current page.

[Note] on the pop-up menu can only do basic touch keys and values returned.

7.4 Upregulation

Upregulated function instructions stored format shown in Table 7.10 Fig.

table 7.10 Upregulated instructions stored function format

address	definition	Data length	Explanation
0x00	Pic_ID	2	page ID
0x02	TP_Area	8	Touch button area :(Xs, Ys), (Xe, Ye)
0x0A	Pic_Next	2	Target switch pages, you must 0xFF ** , It means no page switch.
0x0C	Pic_On	2	In which the page button press renderings, 0xFF ** That there is no button press effect.
0x0E	TP_Code	2	0xFE02
0x10	0xFE	1	0xFE
0x11	* VP	2	Variable address pointer returned by the data VP_Mode Decision.
0x13	VP_Mode	1	0x00 = Adjust VP Word address (integer); 0x01 = Adjust VP High byte address of the word (1 Byte unsigned number, VP_H); 0x02 = Adjust VP Low byte address of the word (1 Byte unsigned number, VP_L); 0x10-0x1F :Correct VP The specified bit word address (0x10 correspond VP.0 , 0x1F correspond VP.F) Is adjusted, the adjustment must be set to range 0-1 .
0x14	Adj_Mode	1	Adjustment: 0x00 = - Other = ++
0x15	Return_Mode	1	Jabber approach: 0x00 = Stop (equal to the threshold) the other loop regulation =
0x16	Adj_Step	2	Adjustment step, 0x0000-0x7FFF
0x18	V_Min	2	Lower limit: 2 Byte integer (VP_Mode = 0x01 or 0x02 , Only the active low byte)
0x1A	V_Max	2	Limit: 2 Byte integer (VP_Mode = 0x01 or 0x02 , Only the active low byte)
0x1C	Key_Mode	1	0x00 : When the key is held down continuously adjustable; 0x01 : Just hold down the button to adjust the time 1 Times.
0x1D	NULL	3	write 0x00

[Note] variables can be adjusted through the "variable data, icons, WordArt variables" and other data will be displayed.

Available PC end DGUS Incremental software development regulation. in DGUS Software, click the button, then use the mouse to a marquee

Regions, and the functions provided in the menu on the right, and as the upper and lower adjustment step, and selecting a corresponding adjustment mode (+ / -) and the like.

【example】 He produced a series of incremental control button in a page. Wherein the upper and lower triangular triangular two touch buttons are configured ++ Incremental (Adj_Mode = 0x01) And - incremental (Adj_Mode = 0x00). Can hold the button, the data provided in the right of the menu implemented by Set in steps of continued addition and subtraction. Results shown in Figure 7.6 Fig.

Map 7.6 Incremental control buttons

7.5 Drag adjustment

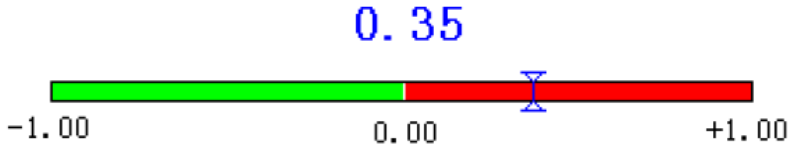
Drag regulation instruction storage format as Table 7.11 Fig.

table 7.11 Drag adjustment function instruction storage format

address	definition	Data length	Explanation
0x00	Pic_ID	2	page ID
0x02	TP_Area	8	Touch button area :(Xs, Ys), (Xe, Ye)
0x0A	Pic_Next	2	Target switch pages, you must 0xFF ** , It means no page switch.
0x0C	Pic_On	2	Page button press renderings are located, must be 0xFF ** , Indicating no effect of a button press.
0x0E	TP_Code	2	0xFE03
0x10	0xFE	1	0xFE
0x11	* VP	2	Variable address pointer.
0x13	Adj_Mode	1	☆ high 4bit It defines the format of data returned: 0x0 * = Adjust VP Word address (integer); 0x1 * = Adjust VP High byte address of the word (1 Byte unsigned number, VP_H); 0x2 * = Adjust VP Low byte address of the word (1 Byte unsigned number, VP_L). Low ☆ 4bit It defines the drag way: 0x * 0 = Lateral drag; 0x * 1 = Longitudinal drag.
0x14	Area_Adj	8	Effective regulatory regions :(Xs, Ys) (Xe, Ye), Must be consistent with the touch area.
0x1C	V_Begain	2	Start position corresponding to the return value, an integer.
0x1E	V_End	2	End position corresponding to the return value, an integer.

[Note] need to adjust the drag with a "sliding scale indicating" to achieve, relevant content sliding scale instructions, refer to Section 8 chapter. Drag adjusting key is not supported (i.e. 0x4F Key code stored in the register) control. The adjustment slide data can be displayed through the "data variable, the variable icons" and other functions.

Can use PC end DGUS Development of software to drag regulation. in DGUS Software, click on the button, the mouse followed by a marquee area, and the functions provided in the menu on the right, such as dragging and the start position corresponding to the return value to terminate the like. After configuration can Achieve a similar map 7.7 Effect.



Map 7.7 Drag the effect of regulatory function

Drag the regulation has the advantage of intuitive, efficient, and parameters are not out of range. When more precise drag input, drag can be adjusted by the variable data points "Variable Data Display" function is displayed (FIG. 7.7 middle 0.35).

7.6 RTC Set up

In the project, the time display and is provided with a very wide range of applications, in order to facilitate the user to directly set the time, joined the RTC Setting function, that is, DGUS Screen set directly by touching the keyboard RTC Need to complete entry Gregorian calendar (year, month, day, hour, minute, second). The function instruction storage format as Table 7.12 Fig.

table 7.12 RTC Setting function storage format

address	definition	Data length	Explanation
0x00	Pic_ID	2	page ID
0x02	TP_Area	8	Touch button area :(Xs, Ys) (Xe, Ye)
0x0A	Pic_Next	2	Target switch pages, 0xFF ** It means no page switch.
0x0C	Pic_On	2	In which the page button press renderings, 0xFF ** That there is no button press effect.
0x0E	TP_Code	2	0xFE04 (Ie RTC Set)
0x10	0xFE	1	0xFE
0x11	0x00 00 00	3	0x00 00 00
0x14	(X, y)	4	Process input display position, the right alignment, (x, y) String top right corner coordinates.
0x18	Color	2	Enter the display font color.
0x1A	Lib_ID	1	The display used ASCII Font position, 0x00 = The default font.
0x1B	Font_Hor	1	font size, X The number of lattice direction.
0x1C	Cusor_Color	1	Cursor color, 0x00 = Black White Other =
0x1D	KB_Source	1	0x00 = Keyboard = keyboard is not in other current page in the current page
0x1E	PIC_KB	2	Keyboard page where ID , Valid only when the keyboard is not in the current page.
0x20	0xFE	1	0xFE
0x21	Area_KB	8	Keyboard area :(Xs, Ys) (Xe, Ye). Valid only when the keyboard is not in the current page.
0x29	Area_KB_Position	4	Keyboard display coordinate position in the upper left corner of the current page; valid only when the keyboard is not in the current page.
0x2D	NULL	3	write 0x00

【Note】 RTC Making and keyboard configuration variables used in the same entry, are in accordance ASCII Code key defined. Show time by " RTC Display, clock display "function is implemented, the section on display will be the first 8 Chapter to explain.

in DGUS Development software, click the RTC icon, and set the button area marquee effect is shown 7.8 Fig.

Map 7.8 RTC The configuration

After the program configured, when the need to modify the time, just click the button and define the pop-up keyboard, enter the complete calendar (year, month, day, hour, minute, second) can easily modify the time.

7.7 Return value key

That is the key return value of the function Click the button to return directly to the value of the keys to the specified variable (support level variable returns).

table 7.13 The return value of the key format instructions stored

address	definition	Data length	Explanation
0x00	Pic_ID	2	page ID
0x02	TP_Area	8	Touch button area :(Xs, Ys) (Xe, Ye)
0x0A	Pic_Next	2	Target switch pages, 0xFF ** It means no page switch
0x0C	pic_On	2	In which the page button press renderings, 0xFF ** That there is no button press effect.
0x0E	TP_Code	2	0xFE05
0x10	0xFE	1	0xFE
0x11	* VP	2	Variable address pointer
0x13	TP_Mode	1	0x00 = Returns the key value stored in VP Word address (integer); 0x01 = Returns the key value stored in the low byte VP High byte address of the word (VP_H); 0x02 = Returns the key value stored in the low byte VP Low byte address of the word (VP_L); 0x10-0x1F : The return of keys the least significant bit (1bit) Write VP The specified bit word address (0x10 modify VP.0 , 0x1F modify VP.F).
0x14	Key_Code	2	Return key.
0x16	NULL	10	write 0x00 .

in DGUS Development software, click the icon, marquee button area and button effects, page switching, the return key, the variable address, etc.

To complete the configuration of the design.

If the variable is defined address 000C , The key is returned 0005 And turn on automatic data upload function. When you click the button, the serial format received command 5A A5 06 83 00 0C 00 05 .

7.8 Hardware configuration parameters

"Hardware Configuration" enables a plurality of functions, including data transfer between registers and variables, the image is converted to monochrome FIG transmission data to the serial port. Storing the instruction format shown in Table 7.14 Fig.

table 7.14 Hardware configuration instruction storage format parameter

address	definition Data length		Explanation
0x00	Pic_ID	2	page ID
0x02	TP_Area	8	Touch button area :(Xs, Ys) (Xe, Ye)
0x0A	Pic_Next	2	Target switch pages, 0xFF ** It means no page switch.
0x0C	Pic_On	2	In which the page button press renderings, 0xFF ** That there is no button press effect.
0x0E	TP_Code	2	0xFE07
0x10	0xFE	1	0xFE
0x11	Mode	1	Operation mode selection, see "operating mode table" will be described.
0x12	Data_Pack	14	Packet mode of operation, see "operating mode table" will be described.

Hardware configuration Table operation mode parameters 7.15 Fig. DGUS Development software, add the function button icon.

table 7.15 Hardware configuration parameters operating mode

<table>
<tr><th>Mode</th><th>Data_Pack</th><th>Data_Pack Explanation</th><th>Features</th></tr>
<tr><td>0x00</td><td>no</td><td>no</td><td>Loading data into a register variable region 0x6F00-0x6FFF Variable memory space (occupied low byte).</td></tr>
<tr><td>0x01</td><td>no</td><td>no</td><td>load 0x6F00-0x6FFF While the corresponding rewriting; variable memory space (occupied low byte) data into the register variable region R1-R3 , R5-RA SD / SDHC Interface configuration variables.</td></tr>
<tr><td rowspan="3">0x02</td><td>Tran_Area</td><td>The area to be converted coordinates:
(Xs, Ys) (Xe, Ye)</td><td rowspan="3">The contents of the specified area is converted into a monochrome bitmap (longitudinal modulus printing bitmap format), and to save VP A data memory pointer.
1 , The width of the region (Xe-Xs + 1) Must be an even number;
2 Regional height (Ye-Ys + 1)must be 8 Multiples;
3 * VP Pointer data stored in the following format:
* VP : Status bits arranged dealt 0x5555 ;
* VP + 1 : Lateral word length = (Xe-Xs + 1) & 0xFFFE / 2 ;
* VP + 2 : = Number of data segments (Ye-Ys + 1) & 0xFFF8 / 8 ;
* VP + 3 : Bitmap data began, MSB the way. If parameters are automatically upload function (enabled R2.3 = 1 After), then the conversion is complete, according to * VP Content is modified to 0x5555 Automatically post a message. This instruction is mainly used for the printout of the screen contents.</td></tr>
<tr><td>* VP</td><td>Save the converted bitmap data buffer first address</td></tr>
<tr><td colspan="2"><table>
<tr><td></td><td>X=0</td><td>X=1</td><td>X=2</td><td>X=3</td><td>…</td><td>X=126</td><td>X=127</td></tr>
<tr><td>Y=0</td><td>D0.15</td><td>D0.7</td><td>D1.15</td><td>D1.7</td><td></td><td>D63.15</td><td>D63.7</td></tr>
<tr><td>…</td><td>…</td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td>Y=7</td><td>D0.8</td><td>D0.0</td><td>D1.8</td><td>D1.0</td><td></td><td>D63.8</td><td>D63.0</td></tr>
<tr><td>Y=8</td><td>D64.15</td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td>…</td><td>…</td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td>Y=15</td><td>D64.8</td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
</table></td></tr>
<tr><td rowspan="2">0x03</td><td>* VP</td><td>Data Pointer</td><td rowspan="2">The * VP Starting position of the pointer, Tx_Len Transmitting user data to serial byte length.
Tx_Len Word is a variable length from 0x0001-0xFFFF .</td></tr>
<tr><td>Tx_Len</td><td>Length of data to be transmitted</td></tr>
<tr><td colspan="4">0x04 The same function 0x03 , Except that the data is sent to COM2 (Reserved by the system serial port).</td></tr>
<tr><td rowspan="2">0x05</td><td>Tran_Area</td><td>The area to be converted coordinates:
(Xs, Ys) (Xe, Ye)</td><td rowspan="2">The contents of the specified area is converted into a monochrome bitmap (modulo lateral print bitmap format), and to save VP A data memory pointer.
1 , The width of the region (Xe-Xs + 1)must be 16 Multiples;
2 * VP Pointer data stored in the following format:
* VP : Status bits arranged dealt 0x5555 ;
* VP + 1 : Lateral word length = (Xe-Xs + 1) & 0xFFF0 / 16 ;
* VP + 2 : = Number of data segments (Ye-Ys + 1);
* VP + 3 : Bitmap data began, MSB the way. If parameters are automatically upload function (enabled R2.3 = 1 After), then the conversion is complete, according to * VP Content is modified to 0x5555 Automatically post a message. This instruction is mainly used for the printout of the screen contents.</td></tr>
<tr><td>* VP</td><td>Save the converted bitmap data buffer first address</td></tr>
<tr><td rowspan="2">0x06</td><td>Frame_Head</td><td>Header (2 byte)</td><td rowspan="2">Send touch-screen coordinates of the current position of the click COM2 (Retention serial system), the following format:
Frame_Head + X + Y + Check (X, Y of 1 And cumulative byte) + Frame_end.</td></tr>
<tr><td>Frame_End</td><td>End of frame (2 byte)</td></tr>
</table>

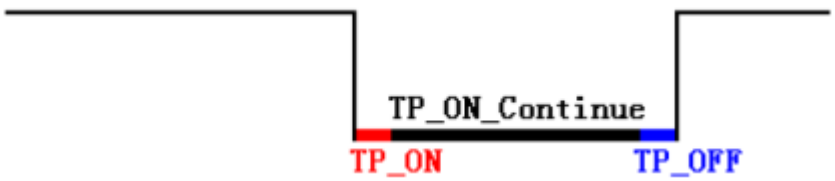
7.9 Touch screen presses state synchronization data is returned

Touch screen presses state synchronization function returns the data that is tapping on the touch screen to return to variable data or serial port as required.

table 7.16 The touch screen presses status return command synchronization data storage format

address	definition	Data length	Explanation
0x00	Pic_ID	2	page ID
0x02	TP_Area	8	Touch button area :(Xs, Ys) (Xe, Ye)
0x0A	Pic_Next	2	Target switch pages, 0xFF ** It means no page switch.
0x0C	Pic_On	2	In which the page button press renderings, 0xFF ** That there is no button press effect.
0x0E	TP_Code	2	0xFE08
0x10	0xFE	1	0xFE
0x11	TP_On_Mode	1	When the first touch-screen is pressed down, the data return mode: 0x00 = Does not return data 0x01 = Read * VP1S Points LEN1 * Length of the data to VP1T Point of storage space. 0x02 = Read * VP1S Points LEN1 Length data sent to the serial port. 0x03 = Read * VP1S Points LEN1 * Length of the data to VP1T Point register space.
0x12	VP1S	2	The first touch-screen is pressed, the read address data.
0x14	VP1T	2	The first touch-screen is pressed, the write address data.
0x16	0x00	1	0x00
0x17	LEN1	1	Return data length, the number of bytes. TP_On_Mode = 0x01 Time, LEN1 It must be even.
0x18	0xFE	1	0xFE
0x19	TP_On_Continue_Mode	1	After the first touch-screen is pressed down, when continuously pressed, the data return mode: 0x00 = Does not return data 0x01 = Read * VP2S Points LEN2 * Length of the data to VP2T Point of storage space. 0x02 = Read * VP2S Points LEN2 Length data sent to the serial port. 0x03 = Read * VP2S Points LEN2 * Length of the data to VP2T Point register space.
0x1A	VP2S	2	When continuously pressed touch screen, read the address data.
0x1C	VP2T	2	When continuously pressed touch screen, write address data.
0x1E	0x00	1	0x00
0x1F	LEN2	1	Return data length, the number of bytes. TP_On_Continue_Mode = 0x01 Time, LEN2 It must be even.
0x20	0xFE	1	0xFE
0x21	TP_OFF_Mode	1	Touch-screen release, data return mode: 0x00 = Does not return data 0x01 = Read * VP3S Points LEN3 * Length of the data to VP3T Point of storage space. 0x02 = Read * VP3S Points LEN3 Length data sent to the serial port. 0x03 = Read * VP3S Points LEN3 * Length of the data to VP3T Point register space.
0x22	VP3S	2	Touch-screen release, read the address data.
0x24	VP3T	2	Touch-screen release, the write address data.
0x26	0x00	1	0x00
0x27	LEN3	1	Return data length, the number of bytes. TP_OFF_Mode = 0x01 Time, LEN3 It must be even.
0x28	0x00	8	Reservations, write 0x00

The touch screen presses 3 States in FIG. 7.9 Fig. in DGUS Development software, add the function button icon.



Map 7.9 The touch screen presses 3 States

7.10 Turn the adjustment

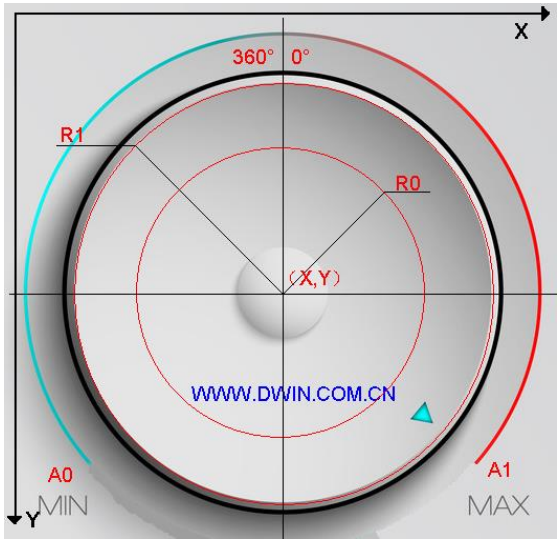
Rotation regulating function is realized by turning the knob variable data entry, such as the instruction table storage format 7.17 Fig.

table 7.17 Turn the adjustment instructions stored format

address	definition	Data length	Explanation
0x00	Pic_ID	2	page ID
0x02	TP_Area	8	Touch button area :(Xs , Ys) (Xe , Ye) As the outer frame region domain regulating circle.
0x0A	Pic_Next	2	Target switch pages, you must 0xFF ** , It means no page switch.
0x0C	Pic_On	2	Button press renderings which pages, must be 0xFF ** , Indicating no effect of a button press.
0x0E	TP_Code	2	0xFE09
0x10	0xFE	1	0xFE
0x11	* VP	2	Variable address pointer.
0x13	Data_Format	1	Adjust data format: 0x00 = Adjust VP Word address (integer); 0x01 = Adjust VP High byte address of the word (1 Byte unsigned number, VP_H); 0x02 = Adjust VP Low byte address of the word (1 Byte unsigned number, VP_L);
0x14	(X , Y)	4	Regulatory region center coordinates.
0x18	R0	2	Adjusting an inner diameter region.
0x1A	R1	2	Regulatory region outer diameter.
0x1C	A0	2	Regulatory region starting angle, 0-719 ,the unit is 0.5 °.
0x1E	V_Begain	2	Starting angle corresponding to the return value, an integer.
0x20	0xFE	1	0xFE
0x21	A1	2	Adjusting the angle of the termination region, 1-720 ,the unit is 0.5 °.
0x23	V_End	2	End angle corresponding to the return value, an integer.

[Note] turning the adjustment must comply with the "icon indicating rotation" to achieve, always assumed to be "clockwise" rotation. Turn the adjustment control button is not supported, (i.e. 0x4F Key code stored in the register).

in DGUS Development software, add the function button icon. Map 7.10 A rotatable adjustment button making good.



Map 7.10 Rotation adjustment button

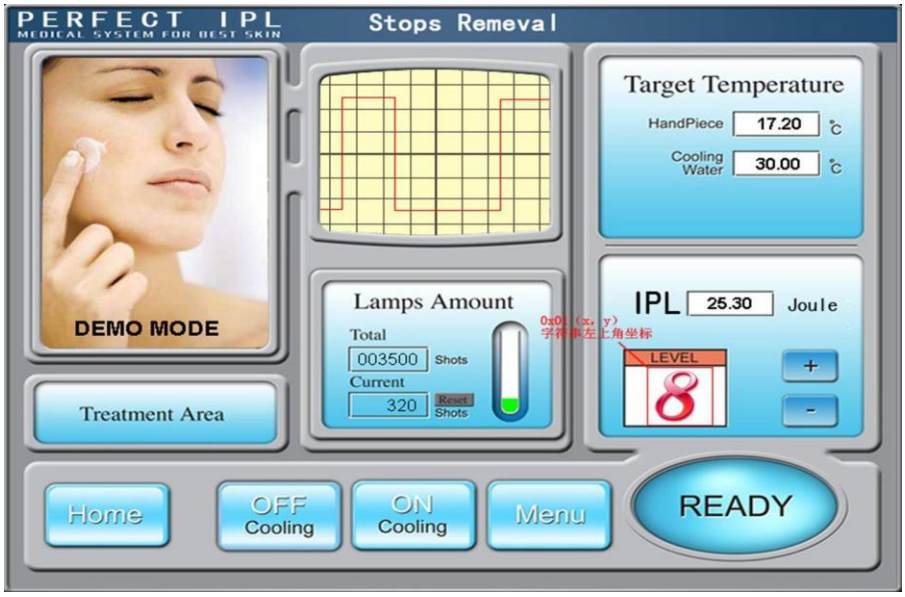
Chapter VIII display variable configuration file description

The first 7 Chapter describes the function and configuration of touch / keyboard configuration file. Many features such as drag adjustment, etc. need to fit the display rotation regulating variable to achieve. This chapter will show variable configuration file (14.BIN) A detailed description and explanation.

Like the touch / keyboard configuration file, display variable configuration file can also use PC End of DGUS Development of software is very easy to generate and settings. In order to facilitate a better understanding of the user, the way of thinking of this chapter will be displayed on the instructions of the configuration file variable more in-depth explanation. Display character profiles are stored in the variable space, which is N Article in accordance with the variable command page configuration composition, each occupying a fixed variable command 32 Byte storage space.

Each page fixed allocation 2KB or 4KB (0x0800 or 0x1000) Variable storage space , So each page can be set up 64 or 128 Variables. Up to display variable configuration file 2MB Up, it is possible to configure 1024 Pages (128 Variable mode 512 Pages). for

The same types of variables, the more the position on the storage, which shows the higher the priority. Map 8.1 It is shown comprising a variable icon display, display data variables, slider scale indication, the variable display WordArt 17 FIG variable display interface.



Map 17 8.1 FIG variable display interface

An instruction by a touch 6 Components, as shown in Table 8.1 Fig.

table 8.1 A touch command 6 Parts

No.	definition	Of data bytes	Explanation
1	0x5A	1	fixed
2	Type	1	Variable Types
3	* SP	2	Variable description file from Flash After loading the address pointer stored in the data storage area, 0xFFFF They said they did not dump the data store.
4	Len_Dsc	2	Description of variable word length
5	* VP	2	Variable address, 0x0000-0xFFFF Some variables need to specify the address to write 0x0000 When the address of a variable to the high byte 0xFF When, this directive will be canceled
6	Description	N	Variable Description Nate capacity

The following describes the display will provide users with a variety of functions and variables of some commonly used functions divided explain in detail.

8.1 Variable list display function

Display variable configuration file that allows DGUS Screen enables several functions, such as table function 8.2 Fig.

table 8.2 Display variable profile feature

No.	Features Code	Features	Explanation
01	00	Variables icon display	The linear range of the variable data corresponding to a set of ICON Icon display; when a variable changes, the icon automatically phase You should switch. Used for fine dashboard, the progress bar display.
02	01	Movie icon display	A variable value corresponding to the data 3 Different kinds of icons indicate the status: no display, showing fixed icon display animation icon. Used for alarm variables.
03	02	Slider scale indication	The variation range of a variable data corresponding to an icon (slider) changes a display position. For multi-level, dials, Indication schedule.
04	03	WordArt Variables, ICON	Replace the font icon to display the variable data.
05	04	Image animation	A set of full-screen images are played in a specified speed. For multi-boot interface or screensaver.
06	05	Rotation Indicator icon	The variation range of a linear variable data corresponding to angle data, and then a ICON Corresponding to the icon number in accordance with an angle According to the rotation is displayed. Multi pointer for instrument panel display.
07	06	Bit variable icon display	Each bit of data to a variable (bit)of 0/1 State corresponds 8 Two different types of display schemes, with ICON Map Mark (or icon animation) to the corresponding display.
08	10	Variable data show	The variable in accordance with a specified data format (integer, decimal, with or without units) with a number of Arab specify the font and size Word is displayed.
09	11	Text Display	The string according to the specified format (font selection decision) is displayed in the display area designated text box.
10	12_00	text format RTC display	Edited by the user according to the Gregorian calendar format RTC Text displayed.
11	12_01	Dial format RTC Display	uses ICON Rotation icon, the calendar mode dial pointer RTC show.
12	13	HEX Data Display	The variable data in bytes HEX User-specified intervals way ASCII Character is displayed. Used for time display, For example, the 1234 shown as 12:34
13	14	Scrolling text display	The variable data in the designated area by the designated direction scroll.
14	20	Real-time graph (trend) binding	0x84 Serial write data buffers to automatically match the curve displayed in real-time graph (trend). You can specify the display area, Central axis coordinates, the display ratio (enlargement / reduction) controllable.
15	21_01	_ Drawing setpoint	Set point (x, y, color)
16	21_02	Drawing _ endpoint connection	Endpoint connection (color, (x0, y0), ... (xn, yn))
17	21_03	_ Drawing a rectangle	A rectangular display, color, position and size control.
18	21_04	_ Drawing a rectangle filled	Fill the specified rectangular area, fill color, position and size control.
19	21_05	Drawing Circle _	Display full radius, color, position and size control.
20	21_06	Drawing cut and paste pictures _	Cut and paste a picture from the specified area to the currently displayed page.
twenty one	21_07	Drawing _ ICON Icon display	ICON Icons, icon libraries to choose from.
twenty two	21_08	_ Drawing enclosed area filled	Enclosed area is filled, the seed point coordinates, fill color controllable.
twenty three	21_09	Drawing _ spectrum display	Spectrum (vertical lines), the line color, according to the position of the controllable variable data.
twenty four	21_0A	_ Drawing a line segment display	The variable data segment connecting the endpoints, a controllable color.
25	21_0B	_ Drawing a circular arc display	Display arc, the radius, color and controlled starting and ending angles.
26	21_0C	Drawing _ character display	The single character data variable.
27	21_0D	Rectangular drawing _ XOR	Specified rectangular domain bitmap data with the specified color XOR Operation, used for highlighting.
28	21_0E	Plot bitmap display color _	Variables as color bitmap data memory, 0/1 You can specify the corresponding color, used for the custom cursor.
29	21_0F	Plot bitmap display _	Variable bit memory data 5K Color bitmap data used for real-time icon (picture) show download.
30	21_10	_ Drawing area enlargement	Paste Amplifying the specified region 1 Paste location specified times, used with 21_0F Photo instruction to achieve real-time display.
31	twenty two	List shows	The columns are displayed in tables according to defined two-dimensional array of data.
32	25	Two-dimensional code display	In the two-dimensional code graphic display screen specified by the specified content.

8.2 Icon display

To use icon-related functions, first of all you want to load the icon library files to PC end DGUS Development software. Methods as below:

Step1 : First, open the Toolbox Devine Devine ICO File generator, FIG. 8.2 Fig.

Step2 : in ICO Select the desired file generator is converted to ICO Icon image file (image format, such as the figure above description), and click on Health Into a new ICO Files can be generated ICO Icon library files (for example, 30_ Icon library test. ICO).

Step3 : The icon library files by copying to SD Kagan directory DWIN_SET And downloaded to the folder DGUS Screen can be.

Map 8.2 Open the toolbox Devine Devine ICO File Generator

8.2.1 Variables icon display

Variable function icons displayed is a linear variation range of the variable data corresponding to a set of ICON Icon display, when a variable changes, the icon is automatically switched as appropriate. Used for fine dashboard, the progress bar display. The display typically used with digital input. As the instruction storage format table 8.3 Fig.

table 8.3 Variables icon display instruction storage format

address		definition	Data length	Explanation
0x00		0x5A00	2	Fixed value, wherein the two 00 Is a function code, details of the first parameter list
0x02		* SP	2	Description pointer variable, 0xFFFF Represented by a configuration file is loaded
0x04		0x0008	2	Fixed value 0x0008
0x06	0x00	* VP	2	Pointer variable, the variable is the integer format
0x08	0x01	(X, y)	4	Variable upper left corner of the display, the icon coordinate position
0x0C	0x03	V_Min	2	The lower limit of variable cross-border are not displayed
0x0E	0x04	V_Max	2	Variable limit, does not show cross-border
0x10	0x05	Icon_Min	2	V_Min The corresponding icon ID
0x12	0x06	Icon_Max	2	V_Max The corresponding icon ID
0x14	0x07_H	Icon_Lib	1	Icon library storage location
0x15	0x07_L	Mode	1	ICON Display mode, 0x00 = Other = Show icon transparent background

[Note] two addresses correspond to the storage format 64/128 Variable mode. The following instructions and the same, Is omitted.

Displaying the icon with a variable number typically entry, etc. upregulated use, simply by PC end DGUS Development of software. turn on DGUS Software, and click the button on the page. Followed by a region of the mouse marquee, the feed can in the menu on the right

Line settings. Figure 8.3 Fig.

Regional settings: (X, Y) for ICON Upper left corner of the icon display coordinate position on the current page.

Name Definition: set a name for the button in the "variable view" in convenient query;

Automatically uploaded data: When checked, the data will be uploaded to the serial port;

Page switch: switch to specify the target picture;

Button effect: a button press effect diagram where the page (- 1 Default is no pressing effect).

Variable Address: definition of variable data storage address;

File icon: To call icon library files (You need to first file into the icon library DGUS Screen can be read correctly

Take, this chapter began icon library files Add Method)

Variable upper limit, the lower limit of the variables: variable display of the specified range, out of range are not displayed.

The corresponding icon: the variable lower limit corresponding icon;

Display modes: a transparent display (the background color display was filtered off) / display background.

Map 8.3 Icon Function variable settings menu

After the configurations may be implemented similar to FIG. 8.4 The effect shown in both figures.

Map 8.4 Variable function icon display

By modifying the pointer data variable, to achieve switching of different icons, as shown in 8.4 Icons corresponding to the display on and off, respectively.

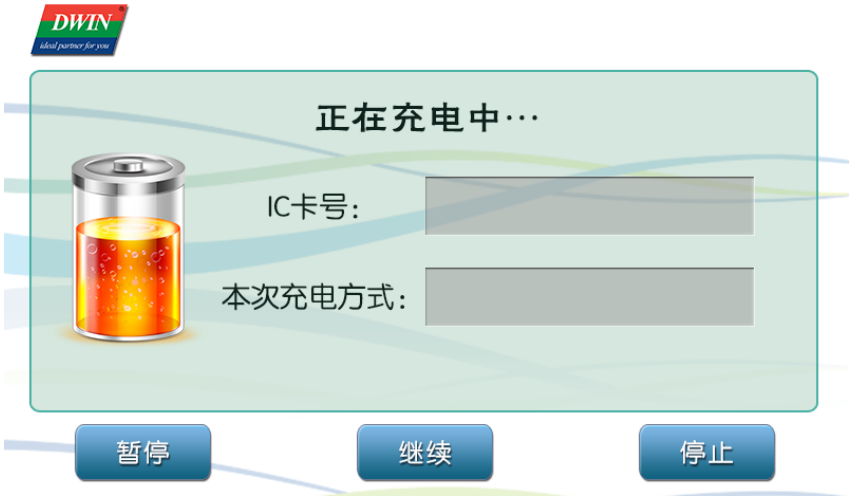
8.2.2 Movie icon display

Movie icon display is a value corresponding to the variable data 3 Icons indicate the status of different species: no display, showing fixed icon display animated icons. Used for alarm variables. Storing the instruction format shown in Table 8.4 Fig.

table 8.4 Movie icon display instruction storage format

address		definition	Data length	Explanation
0x00		0x5A01	2	Fixed value 0x5A01
0x02		* SP	2	Description pointer variable, 0 * FFFF It will load the configuration file.
0x04		0x000A	2	Fixed value 0x000A
0x06	0x00	* VP	2	Initial icon pointer variable, the variable double word, low word reserved, unsigned high word (0x0000-0xFFFF) Data user control the animation icon.
0x08	0x01	(x, y)	4	Variable display position coordinates of the top left corner position.
0x0C	0x03	Reset_ICON_En	2	0x0000: When stopped, the start value is not reset animated icons (animated icons from the display ICON_Start To ICON_End Anywhere between the start of a display). 0x0001: Stop, reset animated icons start value (animated icon will be displayed from fixed ICON_Start Begin to show).
0x0E	0x04	V_Stop	2	Fixed icon display variable to this value.
0x10	0x05	V_Start	2	Movie icon is automatically displayed when a variable to this value.
0x12	0x06	ICON_Stop	2	Variable V_Stop Fixed display the icon.
0x14	0x07	ICON_Start	2	Variable V_Start Value, automatically from ICON_Start To ICON_End Display icon form animation.
0x16	0x08	ICON_End	2	
0x18	0x09: H	ICON_Lib	1	Icon library storage location
0x19	0x09: L	Mode	1	ICON Display mode, 0x00 = Transparent

in DGUS Development software, click the icon, and after the marquee display area to complete the configuration of this feature. And wherein the stop value represents a fixed stop icon when the icon displayed for the variable value. The start value indicating the start of an animated icon shows the trigger value, start, end icon is in the range of variable values at the start of the animation icon cycle. The effect is similar to FIG. 8.5 Fig.



Map 8.5 Animated icons function display

8.2.3 Slider scale indication

In the first 7 chapter 7.5 Drag adjustment described in the section, this section describes the slider scale with an indication of its use. Slider scale indicator is a Range data corresponding to a variable icon (slide) a display position change. Used for indicating the level, scale, schedule. Storing the instruction format shown in Table 8.5 Fig.

table 8.5 Slider scale indicating instructions stored format

address		definition	Data length	Explanation
0x00		0x5A02	2	Fixed value 0x5A02
0x02		* SP	2	Description pointer variable, 0xFFFF Represented by a configuration file is loaded
0x04		0x000A	2	Fixed value 0x000A
0x06	0x00	* VP	2	Pointer variable, the variable format VP_Data_Mode Decision.
0x08	0x01	V_Begain	2	Variable corresponding to the initial scale value
0x0A	0x02	V_End	2	Terminating the corresponding variable value scale.
0x0C	0x03	X_Begain	2	Scale start coordinate (longitudinal direction Y coordinate)
0x0E	0x04	X_End	2	Scale termination coordinate (longitudinal direction Y coordinate)
0x10	0x05	ICON_ID	2	Icon Scale sliding block ID
0x12	0x06	Y	2	This indicator shows the scale of Y A coordinate position (longitudinal direction X coordinate)
0x14	0x07: H	X_adj	1	This indicator shows the scale of X Forward offset coordinate (longitudinal direction Y coordinate), 0x00-0xFF
0x15	0x07: L	Mode	1	Scale mode: 0x00 = Lateral scale bar 0x01 = Longitudinal scale bar
0x16	0x08: H	ICON_Lib	1	Icon library storage location
0x17	0x08: L	ICON_Mode	1	ICON Display mode, 0x00 = Transparent (the background is not displayed), other background display icon =
0x18	0x09: H VP_Data_Mode		1	0x00 : * VP Points to an integer variable 0x01 : * VP Points to an integer variable high byte address 0x02 : * VP Points to an integer variable low byte address

in DGUS Development software, click the icon, and after the marquee area to complete the configuration of this feature. As mentioned earlier, the slider is a function of scale indication, the drag regulation control function, which can be realized together with the two drag the slider icon to change the function variable value. Additionally, it can also be used separately displaying the progress bar, with the case without the drag adjustment function.

When [Note] with drag regulate the use, drag adjustment range to be consistent with the scope marquee indicates a slider scale, so as to achieve the effect of dragging a finger along the slider.

The functional effects similar to FIG. 8.6 (The user can drag the slider icon with your finger to change the percentages to the right).

Map 8.6 Slider icon display function

8.2.4 WordArt variable display

WordArt variable display is ICON Replace the font icon to display the variable data. This function icons variables show similar features. Storing the instruction format shown in Table 8.6 Fig.

table 8.6 Art display instruction word variable storage format

address		definition	Data Size	Description
0x00		0x5A03	2	Fixed value: 0x5A03
0x02		* SP	2	Description pointer variable, 0xFFFF Represented by a configuration file is loaded
0x04		0x0007	2	Fixed value: 0x0007
0x06	0x00	* VP	2	Variable pointer
0x08	0x01	(X , Y)	4	Starting position is displayed: Left mode, the coordinates of the upper left corner coordinates of the character string; right alignment mode, the coordinates of the upper right corner of the display character string coordinates.
0x0C	0x03	ICON0	2	0 corresponding ICON_ID , Of the order 0123456789 ....
0x0E	0x04: H	ICON_Lib	1	ICON Library Location
0x0F	0x04: L	ICON_Mode	1	ICON Display mode, 0x00 = Other transparent background display =
0x10	0x05: H	Integer digits	1	Integer digits displayed
0x11	0x05: L	Decimal places	1	Number of decimal places displayed
0x12	0x06: H Variable data types		1	0x00 = Integer (2 Byte), the range - 32768 To 32767 0x01 = Long integers (4 Byte), the range - 2147483648 To 2147483647 0x02 = * VP High byte, unsigned number, scope 0 To 255 0x03 = * VP Low byte, unsigned number, scope 0 To 255 0x04 = Long integers (8 Byte), the range - 9223372036854775808 To 9223372036854775807 0x05 = Unsigned integer (2 Byte), range 0 To 65535 0x06 = Unsigned long (4 Byte), range 0 To 4294967295
0x13	0x06: L	Align mode	1	0x00 = Left 0x01 = Align Right

in DGUS Development software, click the icon, and after the marquee area to complete the configuration of this feature. The effect is similar to FIG. 8.7 Fig.

Map 8.7 WordArt variable function display

8.2.5 Image animation

Image animation display is a set of full-screen images at the specified speed playback, multi-function for the boot interface or screensaver. Storing the instruction format shown in Table 8.7 Fig.

table 8.7 Image animation display instruction storage format

address		definition	Data length	Explanation
0x00		0x5A04	2	Fixed value 0x5A04
0x02		* SP	2	Description pointer variable, 0xFFFF It will load the configuration file.
0x04		0x0004	2	Fixed value 0x0004
0x06	0x00	0x0000	2	Fixed value 0x0000
0x08	0x01	Pic_Begain	2	Starting position icon
0x0A	0x02	Pic_End	2	Termination icon positions
0x0C	0x03: H	Frame_Time	1	A (picture) of the time display units 8ms .

[Note] position must be less than the starting picture terminate picture position. If the page is also set to terminate the image animation variables, users can achieve replay.

in DGUS Development software, click the icon, marquee area after the specified page and complete the configuration to the function. This feature eliminates the need trigger button, so the area can be any marquee, as long as you can in the specified page.

【example】 Assuming 12 Add the picture number, and set the start and end positions are pictures 0 with 9 , 0-9 As shown in the picture 8.8 Fig.

When opening DGUS Screen and switch to 12 When the picture, this feature is automatically activated, according to the picture 0-9 The final stop to show the order 9 No picture.

Map 8.8 Image animation display example

8.2.6 Rotation Indicator icon

In the first 7 chapter 7.10 Turning the adjustment described in the section, this section describes the rotation icon with an indication of its use. Icons indicating function is to change the rotation range of a linear variable data corresponding to the angle data, and then a ICON Icons according to the corresponding rotation angle data is displayed. This function is used for a pointer instrument panel display. Storing the instruction format shown in Table 8.8 Fig.

table 8.8 Rotation icon indicating instructions stored format

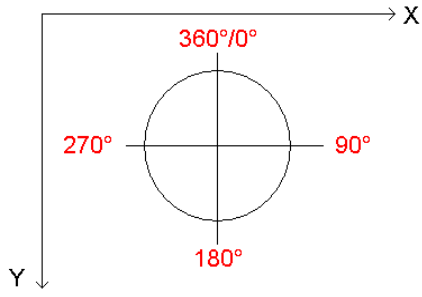
address		definition	Data length	Explanation
0x00		0x5A05	2	Fixed value 0x5A05
0x02		* SP	2	Description pointer variable, 0xFFFF Represented by a configuration file is loaded
0x04		0x000C	2	Fixed value 0x000C
0x06	0x00	* VP	2	Pointer variable, the variable mode VP_Mode Decision.
0x08	0x01	ICON_ID	2	Designated icon ID .
0x0A	0x02	ICON_Xc	2	ICON Rotational center of the icon: X coordinate
0x0C	0x03	ICON_Yc	2	ICON Rotational center of the icon: Y coordinate
0x0E	0x04	Xc	2	ICON Display current position to the center of rotation of the screen: X coordinate
0x10	0x05	Yc	2	ICON Display current position to the center of rotation of the screen: Y coordinate
0x12	0x06	V_Begain	2	Variable corresponding to the initial rotation angle value, an integer, does not show cross-border
0x14	0x07	V_End	2	Rotation angle corresponding to the variable termination value, an integer, does not show cross-border
0x16	0x08	AL_Begain	2	Starting angle of rotation, 0-720 (0x000-0x2D0),unit 0.5 °.
0x18	0x09	AL_End	2	Termination of the rotation angle, 0-720 (0x000-0x2D0),unit 0.5 °.
0x1A	0x0A: H VP_Mode		1	0x00 : * VP Points to an integer variable 0x01 : * VP It points to an integer variable of the high byte 0x02 : * VP Points to a data low byte integer variable
0x1B	0x0A: L	Lib_ID	1	ICON Icon Library ID .
0x1C	0x0B	Mode	1	ICON Display mode, 0x00 = Transparent (not displayed graph background) other background display icon =

[Note] rotation is always assumed to be "clockwise", that is, AL_End It must be greater than AL_Begin If the AL_End Less than AL_Begin The system will automatically add 360 °

in DGUS Development software, click the icon, and after the marquee area to complete the configuration of this feature. As previously described, the icon rotating means It is a functional diagram, the rotation control function is adjusted, so that both can be achieved together with the rotary slider icon to change the function value of the variable. Additionally, it may be a separate instrument panel display, with the case without rotation regulating function.

When [Note] with the rotation adjustment used to adjust the rotation to be consistent with the scope and range of marquee icon indicating the rotation, so as to achieve the effect of dragging a finger along the

The angle defining the functions of FIG. 8.9 Fig.



Map 8.9 The angle of rotation of the custom icon

8.2.7 Bit variable icon display

Bit variable icon display function is to each bit of a data variable (bit)of 0/1 State corresponds 8 Two different types of display schemes, with ICON Icon (icon or animation) to the corresponding display. Storing the instruction format shown in Table 8.9 Fig.

table 8.9 Icon display instruction bit variable storage format

address		definition	Data length	Explanation
0x00		0x5A06	2	Fixed value 0x5A06
0x02		* SP	2	Description pointer variable, 0xFFFF Represented by a configuration file is loaded
0x04		0x000C	2	Fixed value 0x000C
0x06	0x00	* VP	2	Bit pointer variable, a variable word
0x08	0x01	* VP_AUX	2	Auxiliary variable pointer, double word, the user can not access software
0x0A	0x02	Act_Bit_Set	2	Value 1 of bit Location Description * VP Corresponding to the position to be displayed.
0x0C	0x03: H Display_Mode		1	Define the display mode formula: Such as setting Display_Mode = 2 , Then: * VP Corresponding to a variable bit 0 When displayed ICONS icon
0x0D	0x03: L Move_Mode		1	Bitmap icon arrangement: 0x00 = X ++ , Act_Bit_Set Specified does not show bit Not reserved bits; 0x01 = Y ++ , Act_Bit_Set Specified does not show bit Does not retain position 0x02 = X ++ , Act_Bit_Set Specified does not show bit Retention DIS_MOV position 0x03 = Y ++ , Act_Bit_Set Specified does not show bit Retention DIS_MOV position
0x0E	0x04: H	Icon_Mode	1	ICON Display Mode: 0x00 = Transparent 0x01 = opaque
0x0F	0x04: L	Icon_Lib	1	Icon library storage location
0x10	0x05	ICON0S	2	It does not display the animation mode, bit_0 icon ID Animation display mode, bit_0 Icon Animation start ID position
0x12	0x06	ICON0E	2	Animation display mode, bit_0 Icon Animation ended ID position
0x14	0x07	ICON1S	2	It does not display the animation mode, bit_1 icon ID Animation display mode, bit_1 Icon Animation start ID position
0x16	0x08	ICON1E	2	Animation display mode, bit_1 Icon Animation ended ID position
0x18	0x09	(x , y)	4	Start bit variable upper left corner of the display, the icon coordinate position.
0x1C	0x0B	DIS_MOV	2	The next icon coordinate moving coordinate interval
0x1E	0x0C	Retention	2	write 0x00

Display_ Mode	Bit variables (bit) value	
	0	1
0x00	ICON0S	ICON1S
0x01	ICON0S	Do not show
0x02	ICON0S	ICON1S-ICON1E Animation
0x03	Do not show	ICON1S
0x04	Do not show	ICON1S-ICON1E Animation
0x05	ICON0S-ICON0E Animation	ICON1S
0x06	ICON0S-ICON0E Animation does not show	
0x07	ICON0S-ICON0E Animation	ICON1S-ICON1E Animation

in DGUS Development software, click the button on the page. Followed by a region of the mouse marquee, this function can be provided in the menu on the right. Figure 8.10 Fig.

Map 8.10 Bit variable icon display menu settings

Upper corner coordinate position.

Name Definition : Set a name for the button, easy access in the "Variable View.";

Description Pointer : Defines the profile data memory address, default 0xFFFF By Profiles

Load (usually without modification).

Variable address : Bit pointer variables, word variables.

Secondary address : Auxiliary variable pointer, double word; do not conflict with other addresses.

Set up : Bit control settings for 1 of bit Variable address location corresponding to the position to be displayed.

Display Mode : 0x01-0x07 Defined in the table supra Display_Mode

Mobile mode : Bitmap icon arrangement,

0x00 : X ++ , Setting processing is not specified bit Not retained position.

0x01 : Y ++ , Setting processing is not specified bit Not retained position.

0x02 : X ++ , Setting processing is not specified bit Moving the position reserved interval.

0x03 : Y ++ , Setting processing is not specified bit Moving the position reserved interval.

Mobile intervals : Next icon coordinate moving intervals.

Icon file : Icon library file to invoke.

ICON0S : Do not display animation mode, bit0 icon ID ; Display animation mode, bit0 Icon Animation from beginning ID position.

ICON0E : Display animation mode, bit0 Icon Animation ended ID position.

ICON1S : Do not display animation mode, bit1 icon ID ; Display animation mode, bit1 Movie icon from beginning ID position.

ICON1E : Display animation mode, bit1 Movie icon End ID position.

Display mode: transparent display / display background.

Range setting region :(X, Y) The display position of the icon as the starting left variable

8.3 Text display

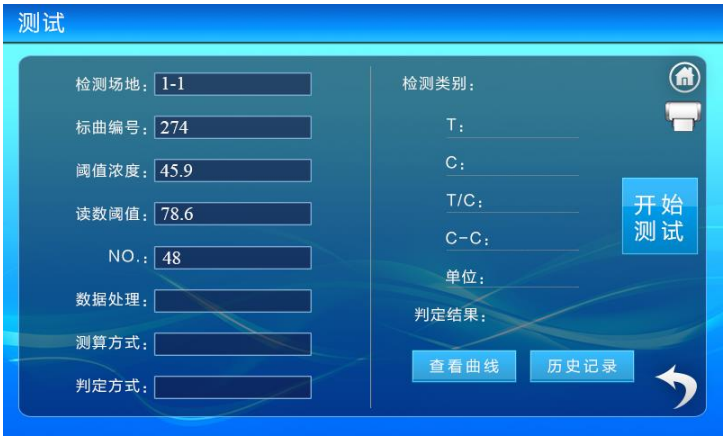
8.3.1 Variable data show

Variable display data is a data variable to be displayed according to the specified format (integer, decimal, whether with units) with the specified font size and Arabic numerals, which instructions are stored as table format 8.10 Fig.

table 8.10 Variable display instruction data storage format

address		definition	Data length	Explanation
0x00		0x5A10	2	Fixed value 0x5A10
0x02		* SP	2	Description pointer variable, 0xFFFF Represented by a configuration file is loaded
0x04		0x000D	2	Fixed value 0x000D
0x06	0x00	* VP	2	Variable pointer
0x08	0x01	X, Y	4	Display start position coordinates of the upper left corner of the display string.
0x0C	0x03	COLOR	2	Color display
0x0E	0x04: H	Lib_ID	1	ASCII Font position
0x0F	0x04: L	font size	1	character X The number of lattice direction
0x10	0x05: H	Alignment	1	0x00 = Left 0x01 = Align Right 0x02 = Center
0x11	0x05: L	Integer digits	1	Display bit integer Integer digits and decimal places can not exceed 20 .
0x12	0x06: H	Decimal places	1	Decimal display
0x13	0x06: L Variable data types		1	0x00 = Integer (2 Byte), the range - 32768 To 32767 0x01 = Long integers (4 Byte), the range - 2147483648 To 2147483647 0x02 = * VP High byte, unsigned number, scope 0 To 255 0x03 = * VP Low byte, unsigned number, scope 0 To 255 0x04 = Long integers (8 Byte), the range - 9223372036854775808 To 9223372036854775807 0x05 = Unsigned integer (2 Byte), the range of 0 To 65535 0x06 = Unsigned long (4 Byte), the range of 0 To 4294967295
0x14	0x07: H	Len_unit	1	Variable Units (fixed character string) display length, 0x00 That there is no unit display
0x15	0x07: L	String_Unit	Max11 Units string, ASCII coding	

in DGUS Development software, click the icon, after the marquee display area, and feature configuration settings menu on the right can be realized in class Like Figure 8.11 The data shown in display.



Map 8.11 Variable data show

8.3.2 Text Display

Display the text string is specified in the format (font selection decision) is displayed in the display area specified text box. This function is fit text entry feature. Storing the instruction format shown in Table 8.11 Fig.

table 8.11 Instructions stored text display format

address		definition	Data length	Explanation
0x00		0x5A11	2	Fixed value 0x5A11
0x02		* SP	2	Description pointer variable, 0xFFFF Represented by a configuration file is loaded
0x04		0x000D	2	Fixed value 0x000D
0x06	0x00	* VP	2	Text pointer
0x08	0x01	(X , Y)	4	Display start position coordinates of the upper left corner of the display string.
0x0C	0x03	Color	2	Display text color
0x0E	0x04	(Xs, Ys) (Xe, Ye)	8	Text Box
0x16	0x08	Text_Length	2	Show the number of bytes, when faced 0xFFFF , 0x0000 Or the end of the text box displayed will not be displayed.
0x18	0x09: H	Font0_ID	1	Encoding to 0x01-0x04 When ASII Position of the character font used.
0x19	0x09: L	Font1_ID	1	Encoding to 0x00 , 0x05 ,as well as 0x01-0x04 When the non ASCII Position of the character font used.
0x1A 0x0A: H		Font_X_Dots	1	Fonts X Direction dot-matrix (0x01-0x04 mode, ASCII Characters X In accordance with the number of lattice direction X / 2 Calculation)
0x1B 0x0A: L		Font_Y_Dots	1	Fonts Y The number of lattice direction
0x1C 0x0B: H		Encode_Mode	1	7 It defines whether text display character spacing automatically adjusting; 7 = 0 Character spacing automatic 7 = 1 Not automatically adjust the character spacing, character width is fixed dot number set. 6-.0 Defined text encoding: 0 = 8bit coding 1 = GB2312 Internal Code 2 = GBK 3 = BIG5 4 = SJIS 5 = UNICODE
0x1D 0x0B: L		HOR_Dis	1	Horizontal spacing of characters
0x1E 0x0C: H		VER_Dis	1	Character vertical spacing
0x1F	0x0C: L	Undefined	1	write 0x00

[Note] font Y You must be an even number of lattice direction. DGUS Pre-ecreen 0 # Font contains 4 * 8-6 * 128 All dot matrix ASCII character.

in DGUS Development of the software, click on the button with the mouse and the marquee display area, and later configures the settings menu function can be realized similar to the right in FIG. 8.12 Display illustrated.

Map 8.12 Text display

8.3.3 RTC display

1) text format RTC display

Clock display text is edited by the user according to the format of the Gregorian RTC Text displayed. Instructions stored in table format as 8.12 Fig.

table 8.12 Clock display format instruction stored in the text

address		definition Data length		Explanation
0x00		0x5A12	2	Fixed value 0x5A12
0x02		* SP	2	Description pointer variable, 0xFFFF Represented by a configuration file is loaded
0x04		0x000D	2	write 0x000D
0x06	0x00	0x0000	2	write 0x0000
0x08	0x01	(X, Y)	4	Display position coordinates of the upper left corner of the display string.
0x0C	0x03	Color	2	Font color
0x0E	0x04: H	Lib_ID	1	Font position
0x0F	0x04: L font size		1	X The number of lattice direction
0x10	0x05	String_Code MAX16		Encoded string, using RTC Coding and ASCII Character constituted. Suppose the current time is 2012-05-02 12:00:00 Wednesday, the YMD H: Q: S 0x00 It will appear as 2012-05-02 12:00:00 MD WH: Q 0x00 It will appear as 05-02 WED 12:00

RTC Table encoding 8.13 Fig.

table 8.13 RCT coding

Explanation	coding	Display Format
In the Gregorian calendar _	Y	2000-2099
Gregorian calendar month _	M	01-12
Day calendar _	D	01-31
Gregorian _ hours	H	00-23
Gregorian _ minutes	Q	00-59
Gregorian _ seconds	S	00-59
_ Calendar week	W	SUN MON TUE WED THU FRI SAT
Coding end	0x00	

in DGUS Text can be realized after the clock developing software, click on the button with the mouse after the marquee display range, and configure font, the time format in the setup menu on the right. able to pass RTC Setting function or use of serial time-modify command.

2) Dial format RTC display

Dial using the clock display ICON Rotation icon, the calendar mode dial pointer RTC show.

table 8.14 Clock dial display format instructions stored

address		definition	Data Size	Description
0x00		0x5A12	2	Fixed value 0x5A12
0x02		* SP	2	Description pointer variable, 0xFFFF Represented by a configuration file is loaded
0x04		0x000D	2	write 0x000D
0x06	0x00	0x0001	2	write 0x0001
0x08	0x01	(X, Y)	4	Central clock dial pointer.
0x0C	0x03	Icon_Hour	2	pointer ICON of ID , 0xFFFF It represents the clock is not displayed.
0x0E	0x04	Icon_Hour_Central	4	Hour ICON The position of the center of rotation.
0x12	0x06	Icon_Minute	2	The minute hand ICON of ID , 0xFFFF It represents the minute hand did not appear.
0x14	0x07	Icon_Minute_Central	4	The minute hand ICON The position of the center of rotation.
0x18	0x09	Icon_Second	2	Second hand ICON of ID , 0xFFFF It represents the second hand is not displayed.
0x1A	0x0A	Icon_Second_Central	4	Second hand ICON The position of the center of rotation.
0x1E	0x0C: H	Icon_Lib	1	Where the pointer icon ICON Library file ID
0x1F	0x0C: L	Undefined	1	write 0x00

in DGUS Development software, just click the button and set the hour, minute, second hand icon in the menu on the right to implement this feature. Clock dial effects similar to FIG. 8.13 Fig.

Map 8.13 Dial clock display

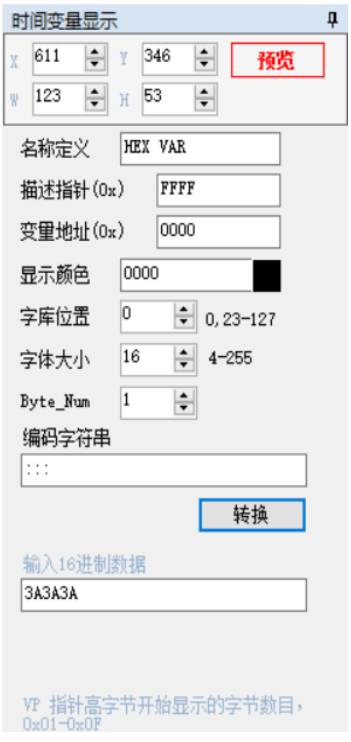
8.3.4 HEX Data Display

HEX The variable display data is data in bytes HEX User-specified intervals way ASCII Character is displayed. Used for time display, such as the 1234 shown as 12:34 , This feature is also known as the time variable display. Storing the instruction format shown in Table 8.15 Fig.

table 8.15 HEX Data stored in the display instruction format

address		definition Data length		Explanation
0x00		0x5A13	2	Fixed value 0x5A13
0x02		* SP	2	Description pointer variable, 0xFFFF Represented by a configuration file is loaded
0x04		0x000D	2	write 0x000D
0x06	0x00	* VP	2	First address data string pointer variable, variable BCD (HEX)coding. For example: data 0x32 shown as 32 ; data 0xBF will 2 shown as BF .
0x08	0x01	(X, Y)	4	String display start position coordinates of the upper left corner.
0x0C	0x03	Color	2	font color
0x0E	0x04: H	Byte_Num	1	* VP High byte pointer byte number display, 0x01-0x0F
0x0F	0x04: L	Lib_ID	1	Font location: half-width font must be a way. in case Lib_ID It is not 0 , You must use the font 8bit coding.
0x10	0x05: H	Font_x	1	X The number of lattice direction.
0x11	0x05: L	String_Code MAX15		Coded string that combinations of variables, and a display time format customer needs. Each displays a BCD After the time code, the order will be taken to a coded string ASCII Character spacing display. Encoded string, the special characters are defined as follows: 0x00 : Invalid, the characters are not displayed, two BCD The time code displayed together; 0x0D : Wrap show that X = Xs , Y = Y + Font_X * 2 .

in DGUS Development of the software, click on the button, to be followed by the scope of the mouse marquee display, and configure the font size, color, and other encoded string in the setup menu on the right side to implement the functionality. FIG configuration menu 8.14 Fig.



Map 8.14 HEX Data display configuration menu

Range setting region :(X, Y) The coordinates of the upper left corner to display a string in the current page position.

Name Definition : Set a name for the button, easy access in the "Variable View."

Description Pointer : Address define the profile data memory. The default is 0xFFFF , Plus the configuration file Load (usually without modification).

Variable address : First address pointer variable data string, Variable BCD coding .

Color display : Color font.

Font position : Call font position, font must be a half-width mode; if the location is not font 0 ,word Libraries must be used 8bit coding. Font size: Fonts X The number of lattice direction.

Byte_Num : VP High byte pointer byte number display, 0x01-0x0F .

Coded Character : Direct Enter the desired final display ASCII character, Character conversion tools .

Entry 16 Binary data : Click Conversion Automatically encoding the string into 16 Binary data, this also It can be entered directly.

[Note] encoded strings and is used in combination a variable data format of a Client to be displayed. Each displays a BCD After the code will be encoded into a sequence taken from a string More ASCII Character spacing display. 0x00 indicates an invalid, the characters are not displayed, two BCD Codes together. 0x0D Newline display.

【example】 As shown in FIG assumed that the input encoded as string ":::" click-conversion, 16 Binary data 3A3A3A . Assume that the variable is 0x01210224 , The final output is displayed 01: 21: 02:24 .

8.3.5 Scrolling text display

It is to scroll the text display variable data specified direction scroll in the designated area. Storing the instruction format shown in Table 8.16 Fig.

table 8.16 Scrolling text display instruction storage format

address		definition	Data length	Explanation
0x00		0x5A14	2	Fixed value 0x5A14
0x02		* SP	2	Variable Description Pointer
0x04		0x000B	2	Fixed value 0x000B
0x06	0x00	* VP	2	Text pointer. Text pointer ago 3 Words must be preserved, from the user to display text content (VP + 3) To start storage. Text must 0xFF or 0x00 end.
0x08	0x01: H	Rolling_Mode	1	Scroll Mode: 0x00 = Scrolling from right to left.
0x09	0x01: L	Rolling_Dis	1	Scroll spacing, each DGUS The number of pixels dot matrix text scrolling cycle.
0x0A	0x02: H	Adjust_Mode	1	0x00 = Left 0x01 = Align Right 0x02 = Center text display content is insufficient to stop scrolling text box is displayed aligned mode has been effective.
0x0B	0x02: L	Undefined	1	write 0x00
0x0C	0x03	Color	2	Display text color
0x0E	0x04	Xs Ys Xe Ye	8	Text box area
0x16	0x08: H	Font0_ID	1	Encoding to 0x01-0x04 Time, ASCII Font character display position. Encoding to 0x00 , 0x05 When, do not set this parameter, write 0x00 It can be.
0x17	0x08: L	Font1_ID	1	Encoding to 0x01-0x04 When non ASCII Font character display position. Encoding to 0x00 , 0x05 When the display position of the character font used.
0x18	0x09: H	Font_X_Dots	1	Fonts X Direction dot-matrix (0x01-0x04 mode, ASCII character X Automatically in X / 2 Calculation).
0x19	0x09: L	Font_Y_Dots	1	Fonts Y The number of lattice direction.
0x1A	0x0A: H	Encode_Mode	1	7 Defines whether to display the character spacing is automatically 7 = 0 Character spacing automatic 7 = 1 Not automatically adjust the character spacing, character width dot number is set. 0 To. 6 Defined text encoding: 0 = 8bit coding 1 = GB2312 Internal Code 2 = GBK 3 = BIG5 4 = SJIS 5 = UNICODE
0x1B	0x0A: L	Text_Dis	1	Character Spacing
0x1C	0x0B: H	Undefined	4	write 0x00

[Note] font Y You must be an even number of lattice direction. DGUS Pre-installed screen 0 # Font contains 4 * 8 - 64 * 128 All dot matrix ASCII character.

in DGUS Developing software, click the button, then use the mouse marquee display area on the right and the settings menu you can achieve this functionality after scrolling mode, scroll, font color configuration. Which show similar configuration text, not repeated here.

8.4 Graphical display of variables

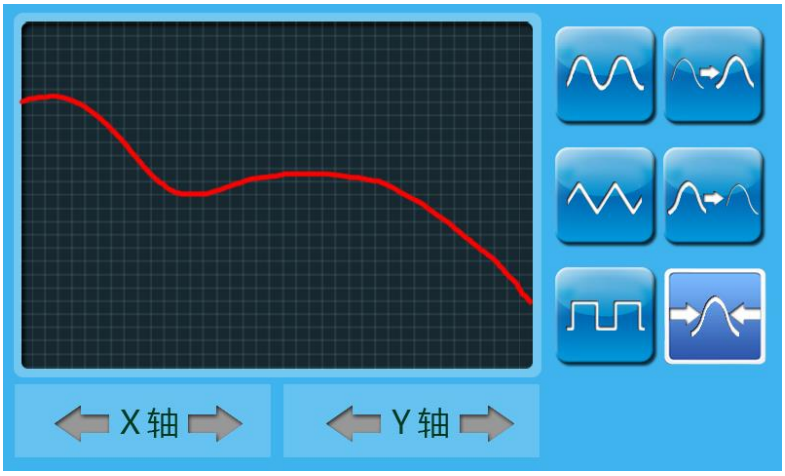
8.4.1 Real-time graph (trend) display

Real-time graph (trend) is a combination display function 0x84 Curve instruction buffer write data to automatically match the display real-time curve (trend). You can specify the display area, the coordinates of the center axis, the display ratio (enlargement / reduction) controllable. The function instruction storage format as Table 8.17 Fig.

table 8.17 Real-time graph (trend) display instruction storage format

address		definition	Data length	Explanation
0x00		0x5A20	2	Fixed value 0x5A20
0x02		* SP	2	Description pointer variable, 0xFFFF Represented by a configuration file is loaded
0x04		0x000A	2	Fixed value 0x000A
0x06	0x00	0x0000	2	Undefined
0x08	0x01	Xs, Ys Xe, Ye	8	Curve window Upper left coordinates (Xs, Ys) And lower right coordinates (Xe, Ye) Bounds the curve will not be displayed.
0x10	0x05	Y_Central	2	The central axis position of the curve
0x12	0x06	VD_Central	2	Curve corresponding to the central axis of the data value, and generally half of the data and the maximum and minimum values.
0x14	0x07	Color	2	Color Curve
0x16	0x08	MUL_Y	2	Longitudinal magnification, the unit is 1/256 , 0x0000-0x7FFF .
0x18	0x09: H	CHANEL	1	A data source channel, 0x00-0x07
0x19	0x09: L	Dis_HOR	1	The horizontal axis spacing, 0x01-0xFF .

in DGUS Development of the software, click on the button with the mouse after the marquee display range, color and configuration of the curve, like the right-hand channel data source is provided to realize the function menu. The effect is similar to FIG. 8.15 Fig.



Map 8.15 Real-time graph (trend) display

[Note] If the variable describing the content stored in the data storage space (* SP Designated storage location), then the combined incremental touch-control commands, the user code can be achieved withou

Curve intervention autoscale; touch drag instruction to modify binding Y_Central Curve values can be achieved without user intervention code is moved up and down.

To display the fairly thick curve line, you can be placed in the same position a plurality of vertically (Y Axis) curve and a reference variable translatable access the same data source can be realized.

Longitudinal magnification full scale curve calculated:

MUL_Y = (Ye-Ys) * 256 / (Vmax-Vmin) ,among them Ye, Ys It is a graph of the window Y coordinate, Vmax, Vmin For the maximum and minimum curve data. For example, a 12bit of A / D Data collection(Vmax = 4095, Vmin = 0) To correspond to the Ys = 50, Ye = 430 The full-scale display screen area, then calculated MUL_Y = 23.7 , Rounded down to take twenty three .

8.4.2 The basic graphic display

The basic function is to display graphical display configuration file 14.BIN Define a "graphics tablet" function, and the particular drawing operations by * VP Variable memory contents pointed to a decision. To achieve different mapping function by changing the user data in variable memory. Storing the instruction format shown in Table 8.18 Fig.

table 8.18 Basic instructions stored graphical display format

address		definition	Data length	Explanation
0x00		0x5A21	2	Fixed value 0x5A21
0x02		* SP	2	Description pointer variable, 0xFFFF Represented by a configuration file is loaded
0x04		0x0008	2	Fixed value 0x0008
0x06	0x00	* VP	2	Variable data pointer
0x08	0x01	Area	8	Drawing display area coordinates of the upper left corner, lower right corner coordinates; drawing out of bounds will not be displayed. Only 0x0001-0x0005 , 0x0009 , 0x000A , 0x000B Command is valid.
0x10	0x05: H Dashed_Line_En		1	0x5A : Use line drawing command (0x02 , 0x03 , 0x09 , 0x0A Instruction) will display a dotted line or dot chain line; Other: using the drawing instruction segments using a solid line segments.
0x11	0x05: L	Dash_Set	4	4 Bytes sequentially provided a broken line (dotted line) format: a first 1 The solid line segment dot number, the dot number of the first paragraph of the broken line, the first 2 The solid line segment dot number, the 2 Number dot broken line segments. For example, setting 0x10 0x04 0x10 0x04 The dotted line is displayed; provided 0x10 0x04 0x02 0x04 The display dot chain line.
0x15	0x07	Undefined	13	Reservations, write 0x00

Variable variable data pointer points to the data format described in Table 8.19 Fig.

table 8.19 Variable data pointed to by variable data format specification

address	definition	Explanation
VP	CMD	Drawing instruction
VP + 1	Data_Pack_Num_Max The maximum data packet: connection command (0x0002), Defined as the number of connection lines (number of vertices - 1);	
VP + 2	DATA_Pack	data

Drawing command packet as described in Table 8.20 Fig.

table 8.20 Drawing instruction described packet

instruction (CMD)	operating	Description drawing data packet format (relative address and the word length of the units are (word))			
		Relative address length		definition	Explanation
0x0001	Set point	0x00	2	(X, y)	Setpoint position coordinates, x High byte of coordinate determination condition.
		0x02	1	Color	Set some color
0x0002	Endpoint Connection	0x00	1	Color	Line Color
		0x01	2	(X, y) 0	Front vertex 0 coordinate, x High byte of coordinate determination condition.
		0x03	2	(X, y) 1	Front vertex 1 coordinate, x High byte of coordinate determination condition.
		0x01 + 2 * n	2	(X, y) n	Front vertex n coordinate, x High byte of coordinate determination condition.
0x0003	rectangle	0x00	2	(X, y) s	Rectangle coordinates of the upper left corner, x High byte of coordinate determination condition.
		0x02	2	(X, y) e	Bottom right corner of the rectangle coordinates.
		0x04	1	Color	Rectangular color.
0x0004 Rectangular filling		0x00	2	(X, y) s	Rectangle coordinates of the upper left corner, x High byte of coordinate determination condition.
		0x02	2	(X, y) e	Bottom right corner of the rectangle coordinates.
		0x04	1	Color	Rectangular fill color.
0x0005 Display full radius		0x00	2	(X, y)	Center coordinates, x High byte of coordinate determination condition.
		0x02	1	Rad	radius
		0x03	1	Color	Arc color
0x0006 Picture area cut Cut and paste		0x00	1	Pic_ID Cut the picture area where the page ID ; High byte of judgment condition	
		0x01	2	(X, y) s	Cut the upper left corner coordinates of the picture area.
		0x03	2	(X, y) e	Cut the bottom right of the picture area coordinates.
		0x05	2	(X, y)	Cut and paste the picture area to the left corner of the current page coordinates of the location.

0x ** 07	ICON icon display	0x00	2	(X, y)	Display coordinate position, x High byte of coordinate determination condition.
		0x02	1	ICON_ID icon ID , Icon library high byte location specified by the instruction. Fixed icon to not display background color.	
0x0008	Area filling	0x00	2	(X, y)	Seed point coordinates, x High byte of coordinate determination condition.
		0x02	1	Color	Fill color.
0x0009	Spectrum Display (Vertical lines)	0x00	1	Color0	The (X0, Y0s) (x0, Y0e)use Color0 Color Connection, X0 High byte determination condition.
		0x01	3	X0, Y0s, Y0e	
0x000A Segment display		0x00	1	Color	The (Xs, Ys) (xe, Ye)use Color Color Connection, Xs High byte determination condition.
		0x01	2	(Xs, Ys)	
		0x03	2	(Xe, Ye)	
0x000B	Arc display	0x00	1	Color0	Arc color display
		0x01	2	(X, Y) 0	Center (X, Y)coordinate, X High byte of coordinate determination condition.
		0x03	1	RAD0	radius
		0x04	1	DEG_S0 Starting point of view, the unit 0.5 °, range 0-720	
		0x05	1	DEG_E0 End angle, the unit 0.5 °, range 0-720	
0x000C	Character display	0x00	1	Color0	Character display color
		0x01	2	(X, Y) 0	Character display top left corner position, character coordinates, X High byte of coordinate determination condition.
		0x03H	0.5	Lib_ID Font position	
		0x03L	0.5 En_Mode Character encoding mode: 0 = 8bit 1 = GB2312 2 = GBK 3 = BIG5 4 = SJIS 5 = UNICODE		
		0x04H	0.5 X_Dots		character X The number of lattice direction
		0x04L	0.5 Y_Dots		character Y The number of lattice direction
		0x05	1	Text0	Character data, 8bit Encoding, only the high byte is valid. When the encoding is 01-04 When, if character data ASCII Character will automatically use 0 # Pre-installed font display.
0x000D Rectangular XOR		0x00	2	(X, y) s	Upper left corner of the rectangular field coordinates, x High byte of coordinate determination condition.
		0x02	2	(X, y) e	Bottom right corner of the rectangular field coordinates.
		0x04	1	Color	Rectangular do XOR s color, 0xFFFF The anti-color operation.
0x000E FIG significant bit color Show		0x00	2	(X, y) s	Rectangular upper left corner of the bitmap display coordinates, x High byte of coordinate determination condition.
		0x02	1	X_Dots	bitmap X The number of lattice direction
		0x03	1	Y_Dots	bitmap Y The number of lattice direction
		0x04	1	Color1	"1" bit Corresponding to the display color
		0x05	1	Color0	"0" bit Corresponding to the display color; if set Color0 with Color1 The same, saying that " 0 "bit Need not be displayed, skip.
		0x06	N	Data_Pack	Display Data, MSB : In order to facilitate the user to read and write data, each row of data must be aligned to a word, i.e. from the next line of data is always a new data word (Word)Start.
0x000F	Bitmap display	0x00	2	(X, y) s	Rectangular upper left corner of the bitmap display coordinates, x High byte of coordinate determination condition.
		0x02	1	X_Dots	bitmap X The number of lattice direction
		0x03	1	Y_Dots	bitmap Y The number of lattice direction
		0x04	N	Data_Pack The display data, one word for each pixel (MSB , 5R6G5B number According format).	
0x0010 Amplifying a region Paste-fold display		0x00	2	(X, y)	Twice enlarged image pasting coordinates at the upper left of the screen, x High byte determination condition.
		0x02	2	(X, y) s	Rectangular coordinate of the upper left corner to be amplified.
		0x04	2	(X, y) e	Rectangular bottom right coordinates to be amplified.

By variable space size limit, the maximum display bitmap 196 * 146 (4: 3) Or 226 * 126 (16: 9)

An enlarged region to be located within the image area, must be aligned with the enlarged lower right corner. Nested zoom magnification can be greater.

[Note] table 8.19 It is determined as adjusted 0xFF Representative drawing operation ends, 0xFE On behalf of this operation is skipped (Ignored).

in DGUS Development software, first click on the button you want to display the drawing board after the display area of the page with the mouse to select box, and on the right

Settings menu address defined variables, can be finally realized using serial drawing instruction.

【example】 By using the above method DGUS Development of software to add a drawing board in a page, followed by the program SD Card to download DGUS Screen, and transmitted through the serial port in Table 8.21 Instruction format. Before drawing of each graphic will be cleared after power. To ensure there every time after power fixed pattern, by twenty two Variable initialization file (22 * .BIN)to fulfill.

table 8.21 Drawing instruction illustrated

Header	Data length	instruction	Address	drawing command	variable (CMD)	Packet number	The top left / bottom right corner coordinates	colour
5A A5	11	82	2000	0003		0001	00 64 00 64 02 BC 01 90	F8 00

The upper face of a drawing instruction described packet, the drawing command 0x0003 To draw a rectangle, so the instruction is variable through the serial port to the memory (address 0x2000) Write data to achieve a red rectangle drawing function. Drawing Other implementations Similarly, the user may attempt to self.

This example red rectangle drawn in FIG. 8.16 Fig.

Map 8.16 Red rectangle display

8.4.3 List shows

List display function is to define the data in the two-dimensional array is displayed in tabular columns. Storing the instruction format shown in Table 8.22 Fig.

table 8.22 Instructions stored list display format

address		definition	Data length	Explanation
0x00		0x5A22	2	Fixed value 0x5A22
0x02		* SP	2	Description pointer variable, 0xFFFF Represented by a configuration file is loaded
0x04		0x000C	2	Fixed value 0x000C
0x06	0x00	* VP	2	Table describes pointers, i.e., TAB [TAB_X_Num] [TAB_Y_Num] The first address of the array.
0x08	0x01: H	TAB_X_Num	1	The number of columns, 0x01-0xFF
0x09	0x01: L	TAB_Y_NUM	1	The number of rows, 0x01-0xFF
0x0A 0x02: H		TAB_X_Start	1	Start display position table column, 0x00-0xFF
0x0B 0x02: L		TAB_Y_Start	1	Table row starting position of the display, 0x00-0xFF
0x0C 0x03: H Unit_Data_Num			1	0x01-0x7F All cells storing a data length of the same data unit lattice space occupied length (Word , Word length). 0x00 by* VP Pointer data variable length storage space defines different columns of cells (Word , Word length). when Unit_Data_Num = 0x00 When, after the contents of table data storage location corresponding extension (TAB_X_NUM / 2) Take up the entire word address. such as,* VP = 0x1000, TAB_X_Num = 0x07 , Then: 0x1000-0x1003 Followed by the first store 0-6 Table data length of the column, wherein 1003 The low byte is unused. 0x1004 Start address table storage content.
0x0D 0x03: L		Encode_Mode	1	7 Defines whether text displayed character spacing is automatically adjusted 7 = 0 Character spacing automatic 7 = 1 Not automatically adjust the character spacing, character width is fixed dot number set. 6 Defines the format of the table of 6 = 0 Table to text format; 6. = 1 Content format table represented by the first two words of the cell data, See Remarks [1] ; 5 Defines the border lines appear: 5 = 0 Show Border; 5 = 1 It does not show borders; 4 Undefined, write 0 . 3-.0 Defined text encoding: 0 = 8bit 1 = GB2312 Internal Code 2 = GBK 3 = BIG5 4 = SJIS 5 = UNICODE
0x0E	0x04	Xs Ys Xe Ye	8	The table shows the top left and bottom-right corner coordinates of the area: the table is always displayed from the top left corner position, out of bounds to end the display.
0x16	0x08	Color_Line	2	Table border line color
0x18	0x09	Color_Text	2	Text display color table
0x1A 0x0A: H		Font0_ID	1	Encoding 0x01-0x04 Time ASCII Font position.
0x1B 0x0A: L		Font1_ID	1	Encoding 0x00 , 0x05 ,as well as 0x01-0x04 Non ASCII Character font used.
0x1C 0x0B: H		Font_X_Dots	1	Fonts X Direction dot-matrix (0x01-0x04 mode, ASCII character X according to X / 2 Calculation)
0x1D 0x0B: L		Font_Y_Dots	1	Fonts Y The number of lattice direction
01E	0x0C: H TAB_X_Adj_Mod		1	When set TAB_X_Start It is not 0 , The display control header: 0x00 = The first column is not displayed 0x01 = The first column shows
0x1F	0x0C: L TAB_Y_Adi_Mod		1	When set TAB_Y_Start It is not 0 , The display control header: 0x00 = The first line is not present 0x02 = The first line shows

Remarks [1] :when Encode_Mode.6 = 1 , The contents of each cell before the data word defines the data format table, which follows: High byte of the first word: Mode Select the type of data

0x00 = Integer (2 Byte), the range - 32768 To 32767

0x01 = Long integers (4 Byte), the range - 2147483648 To 2147483647

0x02 = * VP High byte, unsigned number, scope 0 To 255

0x03 = * VP Low byte, unsigned number, scope 0 To 255

0x04 = Long integers (8 Byte), the range - 9223372036854775808 To 9223372036854775807

0x05 = Unsigned integer (2 Byte), range 0 To 65535

0x06 = Unsigned long (4 Byte), range 0 To 4294967295

0x10 = Time format 1 , 12:34:56 BCD Code string

0x11 = Time format 2 , 12-34-56 BCD Code string

0x12 = Time format 3 , YYYY-MM-DD HH: MM: SS BCD Code string

0xFF = Text low

byte first word:

Mode = 0x00-0x06 It defines the point of variable data display format, high 4bit It represents an integer number of bits, low 4bit Represent decimal places.

Mode = 0x10-0x11 time BCD Byte code string length

Mode = Other undefined

Second word: the cell defines the text color.

[Note] If the table is shorter than the actual content Unit_Data_Num When a predetermined length, using 0xFFFF As cell text terminator. For particularly large tables, through Modified by touching the screen TAB_X_Start with TAB_Y_Start Value can easily achieve positioning and dragging the table.

in DGUS Developing software, click the button, then use the mouse to select the frame to be displayed in the table area, then in the settings menu on the right The number of rows team, number of columns, text color, etc. can be configured to achieve list display function. When configuring pay attention X with Y The direction of the lattice to be consistent with the number to call character dot matrix.

[Note] data in the table will be lost after a power failure, to display fixed data in the table, by twenty two Variable initialization file (22 * .BIN)to fulfill.

Map 8.17 FIG effect is a list display function.

Map 8.17 List display

8.4.4 Two-dimensional code display

Two-dimensional QR Code display is a two-dimensional graphic code specified in the screen according to the specified content. Storing the instruction format shown in Table 8.23 Fig.

table 8.23 Two-dimensional QR Code instructions stored graphical display format

address		definition	Data length	Explanation
0x00		0x5A25	2	Fixed value 0x5A25
0x02		* SP	2	Description pointer variable, 0xFFFF Represented by a configuration file is loaded
0x04		0x0004	2	Fixed value 0x0004
0x06	0x00	* VP	2	Two-dimensional code displays the contents of the pointer. Content longest two-dimensional code 458Bytes, 0x0000 or 0xFFFF As the terminator.
0x08	0x01	(X, y)	4	Left corner position of the two-dimensional code displayed. There are two-dimensional graphic code 45 * 45 Pixel unit (data less than 155 Byte) and 73 * 73 Pixel unit (data less than 459Bytes) Two kinds.
0x0C	0x03 Unit_Pixels		2	Each unit pixel occupied by the two-dimensional code physical pixel matrix size, 0x01-0x07 . Set up Unit_Pixels = 4 , Each unit pixel will appear as 4 * 4 Dot size.
0x0B-0x1F		Retention	18	Undefined, write 0x00 .

in DGUS Development of the software, click on the button with the mouse after the marquee display range, the menu is provided on the right and the variable address, each unit pixel occupied by the two-dimensional code physical pixel size can be configured to achieve a two-dimensional code display. Map 8.18 A two-dimensional code display renderings.

Map 8.18 Two-dimensional code display

Chapter IX DGUS special application software for use

9.1 based on Modbus Host Interface DGUS software application

- Upgrade Program Name: DGUS_V71_MODBUS.BIN ,corresponding DGUS Version is V71 ,stand by K600 + or DGUS Kernel. download link:

http://www.dwin.com.cn/supports/doc-download.html

- Main features: a user interface command set Modbus RTU Host mode, no longer supports standard DGUS Instruction set (including DWIN OS Simulation, serial images, fonts download). DWIN OS No longer supported commands Table 9.1 Fig.

table 9.1 DWIN OS No longer supported instructions

instruction	Explanation
RMODBUS	To COM1_Rx_FIFO Read MODBUS Data Frame
COMSET	COM1 Serial Configuration
COMTXD	Transmitting serial data to COM1
CPRTS	Serial Print
RDXLEN	an examination COM_Rx_FIFO
RDXDAT	Read COM_Rx_FIFO
COMTXI	Direct serial port
RD645	To COM1_Rx_FIFO Read DL / T645 Data Frame

- by 22.BIN Font file 20KW-28KW (16KB Space size, file byte address 0x0A000-0x0DFFF) To define Modbus Operating parameters, DGUS The power will automatically put 8KW Profile decode loaded into the variable buffer final 8KW , As defined in Table 9.2 Fig.

table 9.2 Modbus Operating parameters

DGUS Variable address	definition	Explanation
0x5000	Modbus Enable mark	0x5AA5 Enable representation Modbus Communication.
0x5001: H	Save the configuration file mark 0x5A :save DGUS Variable buffer Modbus Configuration file to twenty two Font	
0x5001: L	Load Profile Tag 0x5A :From twenty two Font loading Modbus Configuration file to DGUS Variable buffer	
0x5002	Baud rate	3.1 Fixed-point decimal format (115200bps = 0x480) ,the unit is kbs ,maximum 999.9kbs
0x5003: H	Serial Port Profile	0x00 = 8N1 (No parity) 0x01 = 8E1 (Even parity, EVEN) 0x02 = 801 (Odd parity, ODD) 0x03 = 8N2 (No parity, 2 Stop bit)
0x5004: H	Variable write flag	0x5A Saves the specified representation DGUS Variable space (PS: 00-PE: 00) To twenty two Font
0x5004: L	Read Mark variables	0x5A Representation from twenty two Reading font data corresponding to the position DGUS Specify the variable space
0x5005	Starting and ending address to read and write variables PS: PE PS with PE Yes DGUS Variable space began, ending address high byte, low byte is set 0	
0x5006-0x5007	Retention	Fixed value 0x0000
0x5008-0x500F The first 1 article MODBUS instruction (16 byte)		0x00 (0x5008H) : 0x5A = This directive effectively other = this directive is invalid 0x01 (0x5008L) : Reading and writing Modbus Device Address 0x02 (0x5009H) : Read / Write use Modbus instruction 0x03 (0x5009L) : Read and write data length, 0x00 Represent this directive is invalid 0x04 (0x500A) : Regular time instruction processing section, comprising a command transmission time, 4 Bit integer units ms ,maximum 9999ms . For read instructions, the timing time is the longest time will respond. 0x06 (0x500B) : 4 Bytes specified Modbus Read instruction transmission scheme 0x0000: **** All pages of instructions are executed 0x0001: Page_ID Only execute instructions at a specified page 0x0002: VP only at VP Variable low-byte buffer pointed to 0x5A When the instruction is executed after the implementation of all relevant instructions automatically cleared VP Pointing to content 0x0A (0x500D) : Read and write data in this directive DGUS Screen start address of the variable storage area. If the address high byte 0xFF Representing the read data is written DGUS Buffer curve, when a curve represented by the low byte address data format. 0x0C (0x500E) : Read and write data in this directive Modbus Starting address on the data device 0x0E (0x500FH) : Bus communication status feedback, write instructions always 0x00 Read instruction return 0x00 (Lose

		defeat)/ 0xFF (success). 0x0F (0x500FL) : Retention, fixed value 0x00.
…	…	…
0x6FF8-0x6FFF	The first 1023 Instructions Supports up 1023 article Modbus instruction	

Modbus Instruction operation corresponding to Table 9.3 Fig.

table 9.3 Modbus Instruction operation correspondence table

Modbus instruction	Features	Read and write data length	Modbus starting address
0x01	Reading the state of the input coil	The number of coils / 8	Starting position of the coil
0x02	Reading the state variable input bits	The number of variable bit / 8	Output start position
0x03	Read the saved data register	Number of registers * 2	Save first address register
0x04	Reads the input data register	Number of registers * 2	Enter the first address register
0x05	Strong single coil set	0x02	Coil addresses
0x06	Preset single register	0x02	Register Address
0x07	Read abnormal state	0x01	Any value
0x0F	Force setting a plurality of coils	Number of coils	Starting position of the coil
0x10	Preset Multiple Registers	Number of registers * 2	First address register
0x11	Read from the machine marked	The number of bytes from the designated machine	Any value

【note】 Modbus Bit (coil) is variable in accordance with LSB Defined, and DGUS in accordance with the MSB Defined.

9.2 based on Modbus Slave interface DGUS software application

- Upgrade Program Name: DGUS_V71_MODBUS_SLAVE.BIN ,correspond DGUS Version is V71 ,stand by K600 + or DGUS Kernel. download link:

http://www.dwin.com.cn/supports/doc-download.html

- **The main function** : A user interface command set Modbus RTU Slave mode, no longer supports standard serial port DGUS Instruction set (including DWIN OS Simulation, serial images, fonts download). DWIN OS No longer supported commands Table 9.4 Fig.

table 9.4 DWIN OS No longer supported instructions

instruction	Explanation
RMODBUS	To COM1_Rx_FIFO Read MODBUS Data Frame
COMSET	COM1 Serial Configuration
COMTXD	Transmitting serial data to COM1
CPRTS	Serial Print
RDXLEN	an examination COM_Rx_FIFO
RDXDAT	Read COM_Rx_FIFO
COMTXI	Direct serial port
RD645	To COM1_Rx_FIFO Read DL / T645 Data Frame

- CONFIG.txt Profiles R3 Register Set Modbus Slave address;

  CONFIG.txt Profiles RA Register Set Modbus Serial format:

  0x00 = 8N1 (No parity), 0x01 = 8E1 (Even parity, EVEN) , 0x02 = 8O1 (Odd parity, ODD) , 0x03 = 8N2 (No parity, two stop Stopping)

  Serial port baud rate setting reference 3.4.1. Section.

table 9.5 Modbus Instruction operation correspondence table

Modbus instruction	Features	Read and write data length	Address range

0x03	Read RAM	Word length	0x0000-0x6FFF
	Tx: 5A 03 00 00 00 02 C9 20 Rx: 5A 03 04 00 00 00 00 10 F6	From RAM of 0x0000 Address to start reading 2 Variable Data	
0x04	Read DGUS register	Number of registers / 2	0x00-0xFF
	Tx: 5A 04 00 00 00 02 7C E0 Rx: 5A 04 04 71 40 00 00 0B A9	From the register area 0x00 Address to start reading 4 Data register	
0x0F	write DGUS register	Byte length	0x00-0xFF
	Tx: 5A 0F 00 03 00 00 02 00 01 A8 20 To register area 0x03 Address to start writing 2 Data Rx: 5A 0F 00 03 00 00 A8 E0		
0x10	write RAM	Word length	0x0000-0x6FFF
	Tx: 5A 10 00 00 00 02 04 31 32 33 34 6D 9E From RAM of 0x0000 Address to start writing 2 Variable Data Rx: 5A 10 00 00 00 02 4C E3		
0x10	write DGUS Buffer curve	Word length	Address High byte 0xFF Curve represents the buffer data is written, Low address byte graph data format.
	Tx: 5A 10 FF 01 00 02 04 3F FE 20 00 27 96 Rx: 5A 10 FF 01 00 02 2D 37	Write to a buffer curve 2 Data	

Chapter X application examples and rapid development guide

In the previous seven chapters in accordance with DGUS Each function screen to divide the chapter, described in detail for the user DGUS Panel parameter configuration, data storage, program download, serial communications, DWIN OS Program development, the touch / display the development and use of the respective functions and configuration files. In order to facilitate The user more convenient and intuitive understanding, in the chapter by several specific examples of application development, along the lines of the program development process DGUS Development of the screen for further explanation.

10.1 Lock system

【example 1】 Design a lock system (supports up to 6 Digit password) to achieve the following functions:

- The boot screen transition effects protection
- Password to unlock the interface have the clock display
- After unsuccessful attempts locks the screen text prompts and sound an alarm buzzer
- Once the password to unlock the lock can be achieved successfully change passwords, modify the system time, change the administrator name of the function.

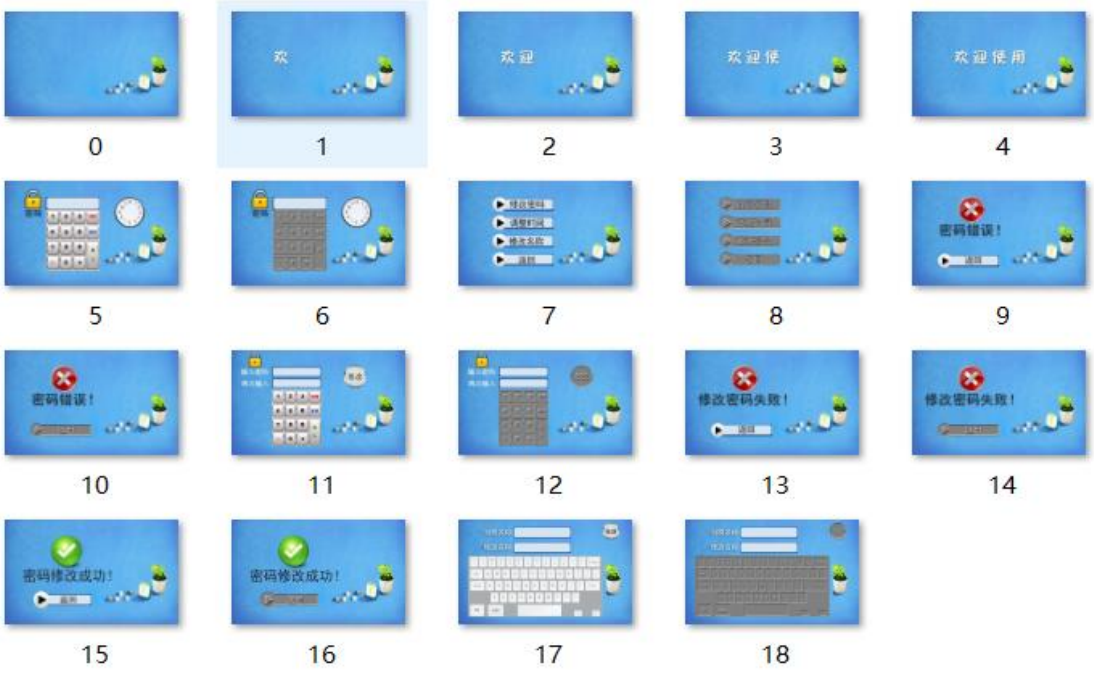
Design steps:

Step1 Variable Plan: The number of variables in some projects more, in order to facilitate program management and follow-up program to upgrade, it is recommended use Excel To organize project variables. Variable storage space, and the space character used in the present project Table 10.1 Fig.

Step2 Interface design: The images used in the project are PS Production. If users have higher requirements for the product interface, the user is recommended commissioned a professional art or design firm to assist in the design. The project is shown in the picture 10.1 Fig.

table 10.1 Variable storage space and character

Variable name	Data word length	storage location
original password	2	20H Font, 000000H address
original password	2	Variable storage space 0005H , 0006H
Verification page (5) enter password	2	Variable storage space 0000H , 0001H
Change Password page (11) enter password	2	Variable storage space 0007H , 0008H
Change Password page (11) re-enter	2	Variable storage space 0009H , 0010H
Modify button	1	Variable storage space 0003H Its value is 0001H
Administrator name	10	Variable storage space 0030H ~ 0039H
Pointer icon library	/	41 # Font
Chinese Character	/	50 # Font



Map 10.1 Pictures of the project

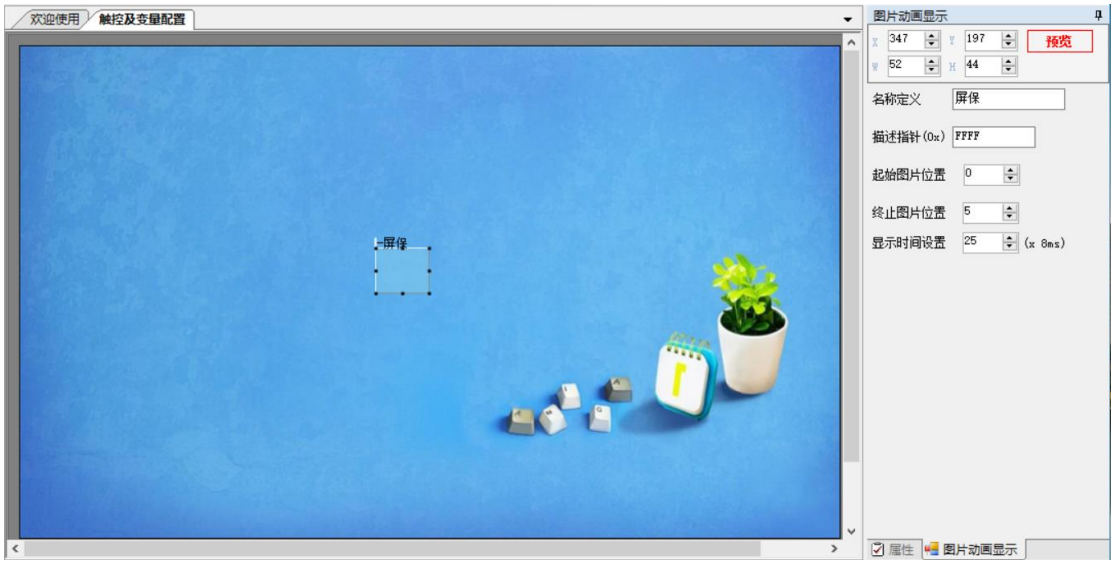
Among them, the picture 0-4 It is a screensaver picture transition, pictures 5 The password verification interface, Pictures 7 Options for the interface, picture 11 To modify the password Interface, Pictures 17 To modify the name of the administrator interface, Pictures 6 , 8 , 10 , 12 , 14 , 16 , 18 Each corresponding interface is a button click picture effect.

Step3 Interface Configuration : use PC End of DGUS Software interface configuration. Each page configuration is as follows:

- image 0-4 Power for the project screensavers transition pictures, so pictures 0 Added "picture animation" feature, set the interval to picture

0.2 Second, as 10.2 Fig.

- After the screensaver is finished, the screen will display pictures 5 . In the picture 5 In, to add a "touch base" function for each key of the keyboard (0-9 correspond 0x0030-0x0039 Determine a corresponding 0x00F1 Cancel the corresponding 0x00F0 , Clear the corresponding 0x00F2), Add "digital input" function for the password input box can(Variable memory address 0000H with 0001H), And add location watch pictures of "dial Time Display" function, as 10.3 Fig.



Map 10.2 Power screensaver settings Figure

Map 10.3 Keyboard clock configuration and FIG.

[Note] dial to the required time display icon library, where use DGUS Toolbox icon to generate a library file that contains the hour, minute and second hands icon (41 pointer. ICO).

- If you enter the correct password, go to the pictures 7 ,image 7 Options for the interface you want to use "basic touch" function for each menu item to set the appropriate picture jump (Change Password jump into the picture 11 , Change the name to jump to pictures 17 , Return jump into the picture 5). In addition, to use " RTC Setting Function "to achieve the adjustment time (Pictures from the keyboard 5 Call), Click on "Adjust Time" will pop up on the right side of the page and when the keyboard Inter box. FIG configuration page 10.4 Fig.

If you enter the wrong password, go to the pictures 9 And a beep alarm sound 1 second(use DWIN OS Program realization). It requires the use of "substantially touch" function is defined in the interface return button (Jump into the picture 5). Figure 10.5 Fig.

Map 10.4 Interface options

Map 10.5 Interface map error

- In the options screen (picture 7), Click on "Change Password", then jump into the picture 11 ,image 11 To modify the password interface. In this interface, to be used "substantially touch" function to define a keyboard (0-9 correspond 0x0030-0x0039 Determine a corresponding 0x00F1 Cancel the corresponding 0x00F0 , Clearance of should 0x00F2), And using the "digital input" function to enter a password box (Variable memory address 0007H with 0008H) And again input box (Variable memory address 0009H with 0010H) Configuration. In addition, the need to change the definition of a button with "key-return" function (Variable memory address 0003H Its value is 0001H). Figure 10.6 Fig.

- If the two passwords are not the same as 1 Then change the password successfully, or failed to change the password. When you click the Modify button will jump to the corresponding results page (This feature uses DWIN OS Program implementation, modification failed jump into the picture 13 , Modified successfully jump into the picture 15). And pictures 9 Like, you need to use the "touch base" function to define the return button (Jump into the picture 5).

Map 10.6 Reset Password screen map

- In the options screen (picture 7), Click on the "Edit Name", then jump into the picture 17 ,image 17 It is to modify the name of the administrator interface. In the page you want to use the "Text display" function to display the current name (Variable memory address 0030H ~ 0039H),use" GBK Chinese character input "function

Can be achieved entry to change the name, but also need to use the "touch base" function to define a complete full keyboard (Key "full button

definition"). The configuration page shown in Figure 10.7 Fig.

Map 10.7 Modify the name of the administrator interface of FIG.

【Note】 GBK Chinese character input and text display function needs to be called Chinese character (font can be used TS3 Character generator to produce, as the project file is the font 50_GBK24_ Times New Roman. HZK), In addition to Chinese character input function needs to use the input method dictionary file (12_PYK_DGUS.BIN).

- After completing the interface configuration, DGUS Development software, click "Generate" button to generate 13 Touch configuration file. BIN with 14 Variable profile. BIN And click "Configure " Button to generate CONFIG.txt Profiles (Note Tick Enable OS).

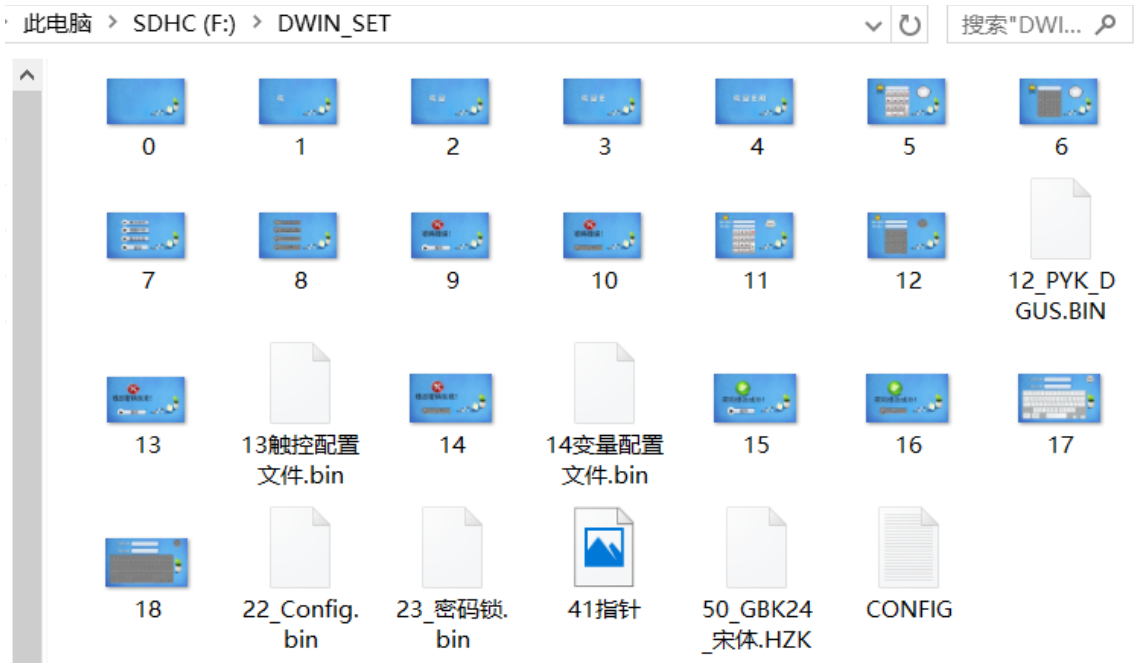
Step4 Test modification : As part of this project required by the function DWIN OS Program is implemented, so as to complete the planning and interface design variables After meter and configuration, need DWIN OS Programming software for programming and an appropriate BIN file(twenty three_ password. BIN). Procedures are as follows:

```
; The program for the DGUS Lock screen programming;
initialization
MOVDR 3 , R30 , 2
IJNE        R31 , 0 , START              ; Screen pictures 0 When the system is initialized
LDWR R30 , 0
LDWR R32 , 1
LDWR R0 , 0000H
MOVXR R30 , 0 , 2                        ; The password stored address 0000H Initialized data 1
;verify password
START :
IJEN        R31 , 5 , BUYANZHENG ;image 5 Is to verify the password page, if the current picture is the 5 Verify the start, otherwise jump
LDWR R4 , 2000H
LDWR R6 , 0000H                          ; Original password is stored in 20H Font, 000000H address
LDWR R0 , 0005H
MOVXL 0 , 2                              ; To read the original password 0005H with 0006H in
LDWR R0 , 0000H                          ; Password stored in the verification interface 0000H with 0001H in
MOVXR R30 , 1 , 2                        ; The password dump R30-R33 in
IJNE        R30 , 0 , YANZHENG
IJNE        R31 , 0 , YANZHENG
IJNE        R32 , 0 , YANZHENG
IJNE        R33 , 1 , YANZHENG ; If the password is 1 , Not be verified
GOTO JIESHU
YANZHENG ;
LDWR R0 , 0005H
MOVXR R35 , 1 , 2                        ; The original password dump R35 ~ R38 in
GOTO BUYANZHENG
CJNE        R30 , R35 , CUOWU
```

```
CJNE        R31 , R36 , CUOWU
CJNE        R32 , R37 , CUOWU
CJNE        R33 , R38 , CUOWU ; Comparing the input password and the original password, if the right to continue, if error jump
LDWR R30 , 7
MOVXR R30 , 3 , 2                   ; Then enter the correct picture 7 Menu options
LDWR R30 , 0
LDWR R32 , 1
LDWR R0 , 0000H
MOVXR R30 , 0 , 2                   ; The password stored address 0000H The data set again 1
GOTO BUYANZHENG CUOWU :

LDWR R30 , 9
MOVRD R30 , 3 , 2                   ; Interface switch to the Picture 9 ,wrong password
LDBR R30 , 100 , 1
MOVRD R30 , 2 , 1                   ; Buzzer 1s
LDWR R30 , 0
LDWR R32 , 1
LDWR R0 , 0000H
MOVXR R30 , 0 , 2                   ; The password stored address 0000H The data set again 1
GOTO JIESHU
BUYANZHENG :
NOP
;change Password
LDWR R0 , 0003H
MOVXR R30 , 1 , 1
IJNE        R31 , 1 , JIESHU ; Analyzing has pressed the "Edit" button, jump if not depressed
LDWR R0 , 0003H
LDWR R30 , 0000H
MOVXR R30 , 0 , 1                   ;will 0003H Data reset to 0
LDWR R0 , 0007H
MOVXR R30 , 1 , 2                   ; Password will be read into the first input R30 ~ R33 in
LDWR R0 , 0009H
MOVXR R35 , 1 , 2                   ; Password entered into the second read R35 ~ R38 in
CJNE        R30 , R35 , ERROR
CJNE        R31 , R36 , ERROR
CJNE        R32 , R37 , ERROR
CJNE        R33 , R38 , ERROR ; Analyzing two password are the same, the same continues, different Jump
IJNE        R30 , 0 , CHONGZHI
IJNE        R31 , 0 , CHONGZHI
IJNE        R32 , 0 , CHONGZHI
IJNE        R33 , 1 , CHONGZHI ; Determining whether the password 1 ,for 1 Proceed, not 1 Jump
ERROR :
LDWR R30 , 13
MOVRD R30 , 3 , 2                   ; Interface switch to the Picture 13 ,fail to edit
GOTO        JIESHU
CHONGZHI :
LDWR R4 , 2000H
LDWR R6 , 0000H
LDWR R0 , 0005H
MOVXR R30 , 0 , 2
MOVXL 1 , 2                         ; Writing the new password 20H Font, 000000H address
LDWR R30 , 15
MOVRD R30 , 3 , 2                   ; Interface switch to the Picture 15 ,Successfully modified
JIESHU :
END
```

After completion of the above work, you can use SD Card will DWIN OS Program and various configuration files to download DGUS Test screen. SD Card file Figure 10-8 Fig.



Map 10.8 DWIN_SET File contents Figure

in DGUS Test the program screen, if there is a problem or not satisfied with the use of display DGUS Development of software to modify and repair File with the change SD Download to card again DGUS Screen to continue testing until each function operates properly, the results were satisfactory so far.

Step5 Version of Archiving : After the fixed version, the configuration files, images, fonts, icons, libraries, etc. DGUS File screen involved (Figure 10.8 As shown) is stored in a SD Card can be mass production.

10.2 Motor control system

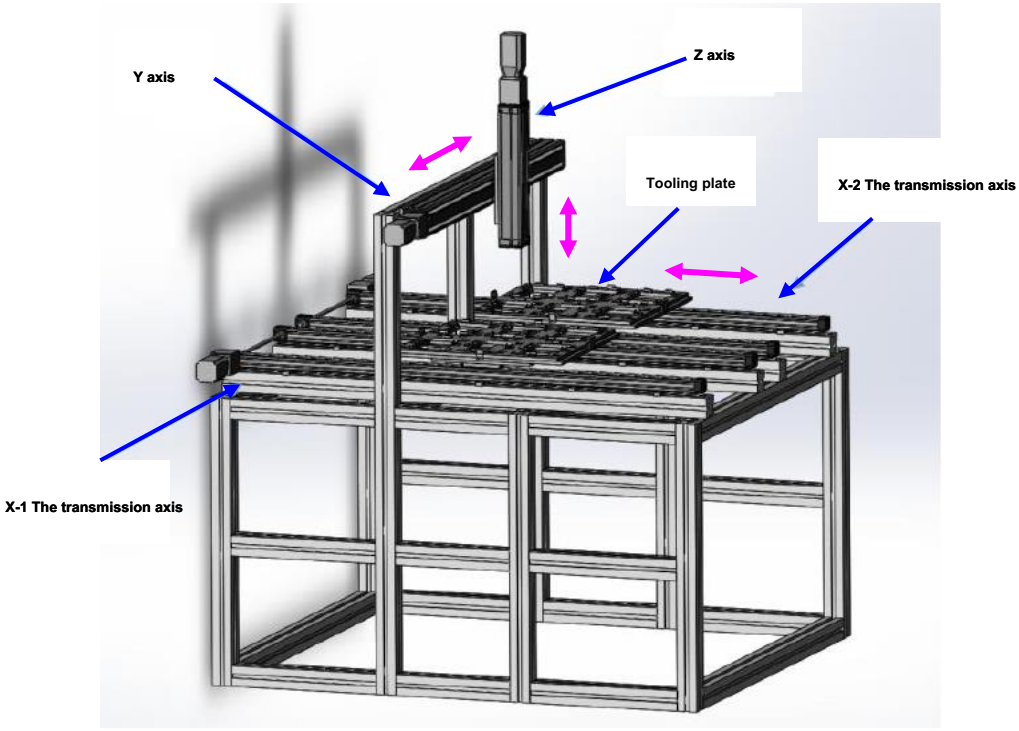
[Example 2] based on Modbus Agreement to DGUS Screen as the main controller, to achieve servo control of a three-axes robot.

- System by 4 Motors with screw realization X , Y , Z Position servo control three degrees of freedom, for Devine DGUS Paste screen panel production, structural principle is shown 10.9 Fig.

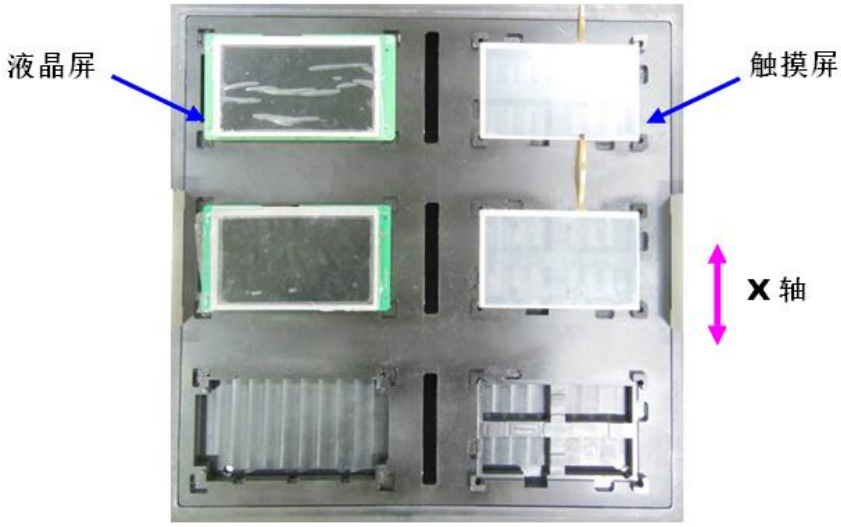
There are two three-axes robot tooling plate, and mounting LCD touch screen (shown in the upper tooling plate 10.10 Shown); X direction

There are two transmission shaft, to realize conveyance tooling plate movement control; Z Direction of the drive shaft is mounted on a vacuum chuck, vacuum chuck driving vertical

movement to achieve gripping of the touch screen; Y The direction of the drive shaft Z Along the axis Y Direction, to achieve a vacuum chuck gripping the touch screen and LCD screen attached to the engagem



Map 10.9 Structure principle a three-axes robot



Map 10.10 LCD and placed on the tooling plate touchscreen

With X-1 Axis, for example, the specific production process are:

Step1 : All axes are at the origin, the origin is determined by the photoelectric switch;

Step2 : X After the tooling plate feed shaft and Y Axis simultaneous movement, the suction position is reached, Z Vacuum generator shaft of a chuck aligned contact

Touch the center of the screen; Z The axial movement to the right height, the vacuum suction operation generator glass screen;

Step3 : Z Back to the axial movement of the origin;

Step4 : Y The axis motion Z Vacuum generator shaft motion to the chuck of a central direction of the liquid crystal panel, and positioned at the center position of the liquid crystal panel;

Step5 : Z Under axial movement to a suitable height, so that the glass screen and LCD screen attached, while the vacuum generator stops suction;

Step6 : Z Back to the axial movement of the origin;

Step7 : X Shaft and Y Axis simultaneous movement, the vacuum chuck aligned with the second block generator touch screen center, the above operation is repeated until X Axis tooling

All three screens fit on the disc is completed, all axes back to square one. in case X-2 Work put into

the feed axis is completed, the screen continues paste production, and so forth.

- System by DGUS Screen human-computer interaction, touch-screen at any time in a manner convenient for parameter setting, command is given. Simultaneously, DGUS

Screen is the master controller, with DWIN OS Programming software program design, and based on Modbus Agreement with DCS005 A touch screen mounted to a dedicated

communication controller controls the motor driver, a servo motor to achieve control.

Design steps:

Step1 Modbus Configuration : This case involved Modbus Communications, in order to simplify the development of, first, Modbus Related configuration. DCS005

Specific touchscreen controller mount read, write register space as defined in Table 10.2 And Table 10.3 Fig.

table 10.2 Read register space (0x03 instruction fetch)

address	data	definition	Explanation
0	0x0602	DCS Controller device ID	
1	IN_STA	16 Way switch input state detecting	IN_STA.0 correspond D11 ..... IN_STA.15 correspond D16
2	RL_STA	4 State relay switch output	RL_STA.0 correspond K1 ..... RL_STA.3 correspond K4
3	P_STA	4 road MOSFET Switch output state	PL_STA.0 correspond P1 ..... PL_STA.3 correspond P4
4	M1_STA	M1 Motor status	high 8bit To execute instructions: 00 = stop/ 01 = Homing / 03 = go ahead/ 04 = Low Back 8bit The implementation of state: 0x00 = End/ 0xFF = Execution
5	M2_STA	M2 Motor status	
6	M3_STA	M3 Motor status	
7	M4_STA	M4 Motor status	
8-127	ND	Undefined	

In the first 9 Chapter has been based on Modbus Interface DGUS Software applications are described. In the present embodiment, in order to facilitate the development of (an important technique to simplify development), by first 22.BIN Font files Modbus Operating parameters are defined, the DCS005 Mounting the touch screen controller dedicated register mapped to the same address space DGUS Screen register space, and parameters shown in 10.11 . In Fig. 10.11 , The red box

Content of this project Modbus Parameter defines the specific meaning can refer to Table 9.2 . 22.BIN Font file can be used hex Editor for editing.

table 10.3 Write register space (0x10 Instruction fetch)

address	data	definition	Explanation
128	RL_SET	4 Relay switch control output	RL_STA.0 correspond K1 ..... RL_STA.3 correspond K4
129	P_SET	4 road MOSFET Switch output state	PL_STA.0 correspond P1 ..... PL_STA.3 correspond P4
130	DIS_M1_SET	M1 control, 48bit Unsigned high 16bit It defines the output mode:	
133	DIS_M2_SET	M2 control, 48bit Unsigned	0x5A00: stop
136	DIS_M3_SET	M3 control, 48bit Unsigned	0x5A01: Homing (for M1 , D = 1 Turn until DIZ = 0) 0x5A02: Homing (for M1 , D = 0 Turn until DIZ = 0 0x5A03: D = 1 ,go ahead N A pulse encoder
139	DIS_M4_SET	M4 control, 48bit Unsigned	0x5A04: D = 0 , Back N Low pulse code disc 32bit: N A plurality of pulse code wheel output frequency 100KHz . Lower actual data only 24bit effective.
142-255	ND	Undefined	

00009FD0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00009FE0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00009FF0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
0000A000	5A	A5	00	00	04	80	00	00	00	00	00	00	00	00	00	00
0000A010	5A	01	03	10	00	20	00	00	00	00	00	00	00	00	00	00
0000A020	5A	01	10	1C	00	20	00	02	08	00	00	80	00	80	00	00
0000A030	0A	01	10	02	00	05	00	02	04	06	00	8A	00	8A	00	00
0000A040	0A	01	10	02	00	05	00	02	04	07	00	8A	00	8B	00	00
0000A050	0A	01	10	02	00	05	00	02	04	08	00	8A	00	8C	00	00
0000A060	0A	01	03	02	00	05	00	00	00	00	FF	01	00	06	00	00
0000A070	0A	01	03	02	00	05	00	00	00	00	FF	02	00	08	00	00
0000A080	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
0000A090	0A	01	10	08	00	05	00	00	00	00	00	81	00	81	00	00
0000A0A0	0A	01	10	02	00	05	00	02	04	09	00	85	00	85	00	00
0000A0B0	0A	01	10	06	00	05	00	00	00	00	00	86	00	86	00	00
0000A0C0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
0000A0D0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00

Map 10.11 Modbus Operating parameter definitions

【Note】 Modbus Bit (coil) is variable in accordance with LSB Defined, and DGUS In accordance with the MSB Defined.

In actual application, many devices use PLC Or microcontroller, which interfaces are Modbus Communication protocol, the user simply DGUS

Screen connected up, and after a simple to develop, can efficiently complete a project.

Step2 Variable Plan : Variable storage space used in the present project Table 10.2 ,table 10.3 And Table 10.4 Fig. Among them, the table 10.2 And Table 10.3

Lists DCS005 Specific touchscreen controller reads the variable placement register space and write register space defined; Table 10.4 To achieve the lists

Motor control, this project custom variables.

table 10.4 The project custom variables

Variable name	Data word length storage location		Explanation
Start / stop / reset select	1	142	3 Keys return type, the same address, return values were 0,1,2 , Corresponding to starting, stopping and resetting.
Platform Choice	1	143	2 Return type keys respectively corresponding platform 1 And platform 2 .
Production processes flag	1	144	The current labeling system processes.
platform 1 Place flag	1	145	Key return type, inform the platform 1 Are in place.
platform 2 Place flag	1	146	Key return type, inform the platform 2 Are in place.
Platform production schedule	1	147	Each platform can produce once 3 Member, the count variable for the production of a number of platform member
M1 To fix 1 The number of pulses	2	148 : 149	Y Axis to X-1 Tooling plate shaft touchscreen Y The direction of the center of the number of pulses
M1 To fix 2 The number of pulses	2	150 : 151	Y Axis to X-2 Tooling plate shaft touchscreen Y The direction of the center of the number of pulses
M1 Carry the number of pulses	2	152 : 153 Tooling plate pair of touch screen and LCD screen Y From the number of pulses direction	
M2 / M3 carry 1 The number of pulses	2	154 : 155	X Starting the first axis aligned with a number of touch screen vacuum chuck pulses from the origin
M2 / M3 carry 2 The number of pulses	2	156 : 157 Tooling plate two touch screen X Axial direction from the number of pulses	
M4 carry 1 The number of pulses	2	158 : 159 Vacuum chuck suction touchscreen decrease the number of pulses	
M4 carry 2 The number of pulses	2	163: 164	Vacuum chuck bonded LCD touch screen and the number of pulses fall
M1 Variable number of pulses transit	2	160: 161	DWIN OS Programming needs
Timer counter	1	162	And a vacuum generator to draw down takes time, which variable is used for timing
Current production quantity	1	165	For counting the current number of production parts, the reset clears

In the Table 10.4 in, M1 Finger Y The direction of the motor, M2 , M3 Finger X Two motors direction, M4 Finger Z Motor direction.

Step3 Interface design : This case DGUS The use of screen interface as shown in the picture 10.12 Shown, or the customer may be designed according to actual demand COMMITTEE Care professional artists. The first character when naming the picture is the interface number that interface in accordance with the order from 0 Start.

Map 10.12 This case involved the interface list of FIG.

Step4 Interface Configuration : use PC End of DGUS Software interface configuration. Each page configuration is as follows:

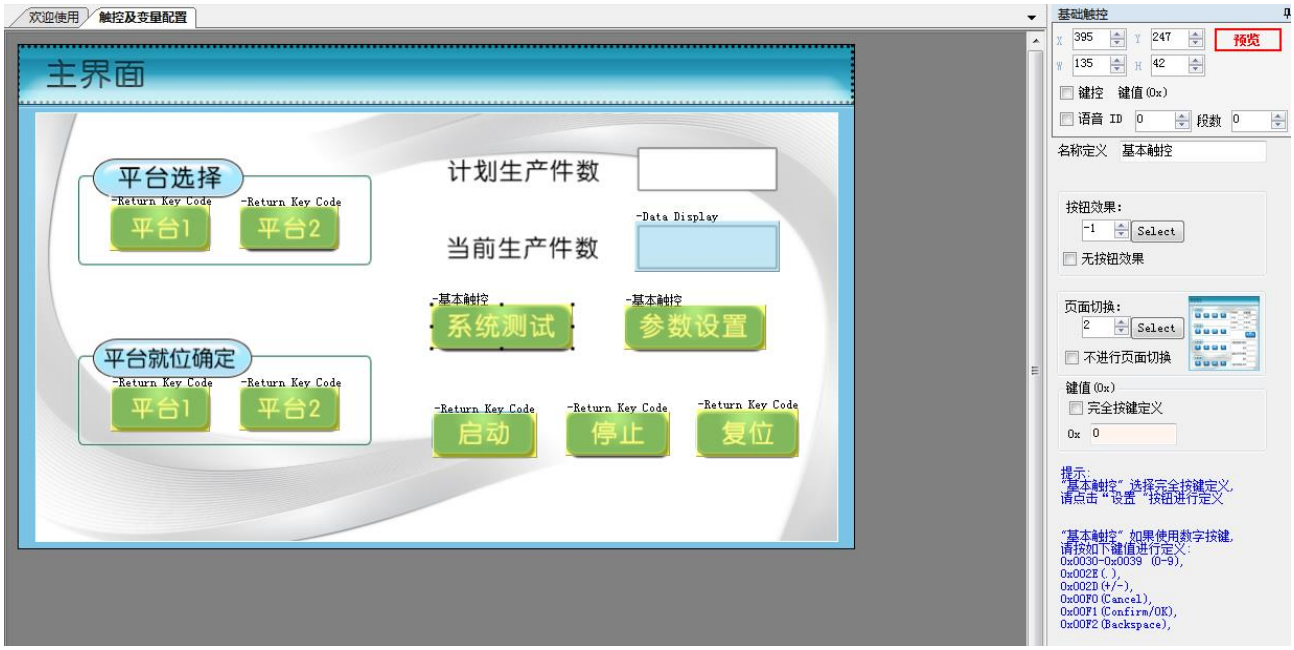
- **Main interface** : The interface controls which use three, respectively, "return button", "substantially touch" and "data variable to display."

"Button back" function that returns the specified value to the specified address after the touch, the example provided in FIG. 10.13 , The input variables and the corresponding key address. For the realization of other functions, follow the instructions to set up.

"Touch Basic" page switching may be achieved, as shown in 10.14 As shown, switch the page to the page you want to switch the option settings.

"Variable Data display" for displaying variable data which is stored in the specified address, as shown in 10.15 As shown in this embodiment need to set the address of a variable, the variable type, and the initial value of the integer number of bits.

Map 10.13 "Return key" Setting Example

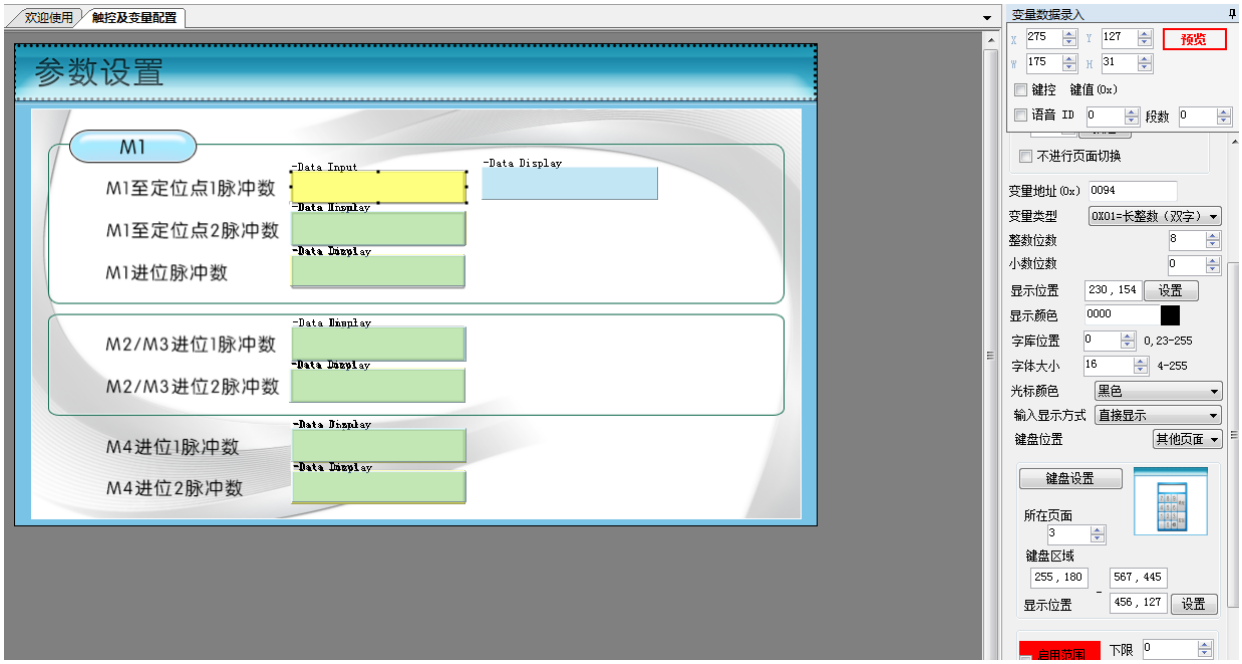


Map 10.14 "Touch Basic" Setting Example

Map 10.15 "Variable Data Display" Setting Example

- **Parameter setting interface** : The interface among controls use of the "Variable data entry" and "data variables are displayed." "Variable data entry" function here is to the specified address DGUS Register input data, as shown in 10.16 , The need to set parameters

There are several variables address, variable type and keyboard location. In this example, the keyboard position in other pages, the page where you can set the keyboard through the keyboard setup options, and select the keyboard area. Meanwhile, the option is provided which can set the display position of the keyboard data entry. After setting, the preview.



10.16 "Variable Data Entry" Setting Example

"Variable Data Display" and can "Variable data entry" controls used in conjunction with data input real-time display, which is set to be variable and address, etc.

"Variable data entry" consistent controls, moreover, in the interface which can also coincide, more in line with user habits.

- **System Test Interface** : The interface control among the "return button", "variable data input" and "Display data variables."

Special Note Is "OK send" button, the control key is used "return button", provided in FIG. 10.17 Fig. The key

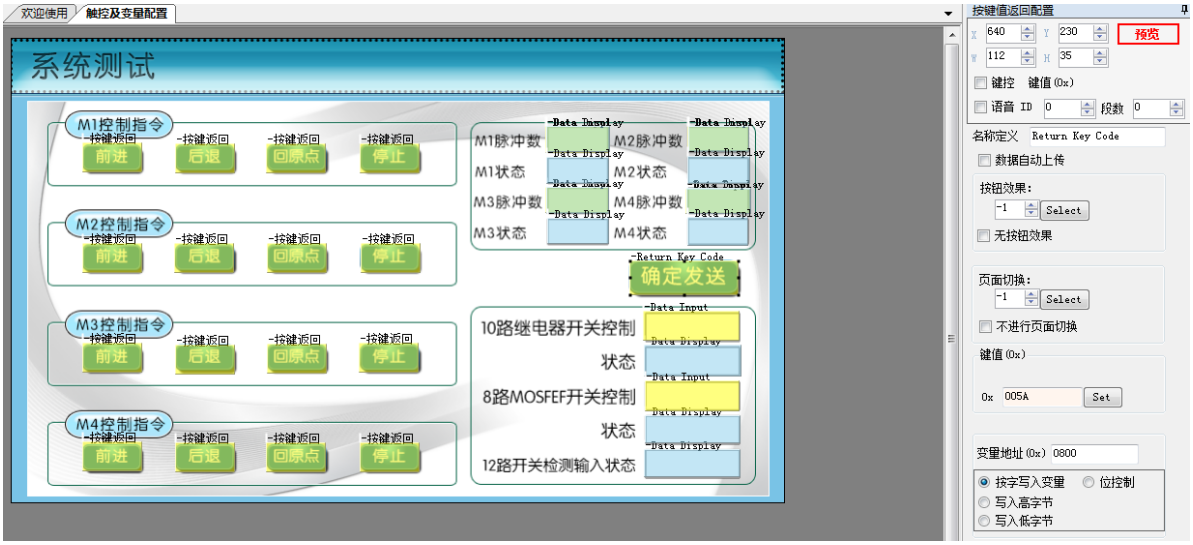
Role is to address 0x0800 of DGUS Write register 0x005A , So that the system tests to determine whether the transmission instruction input. This is because

22.BIN File, define the Modbus When operating parameters, write commands are sent only way VP Variable low-byte buffer pointed to 0x5A

When the instruction is executed after the implementation of all relevant instructions automatically cleared VP Pointing to content. In this example, VP Variable buffer pointed to 0x0800 .

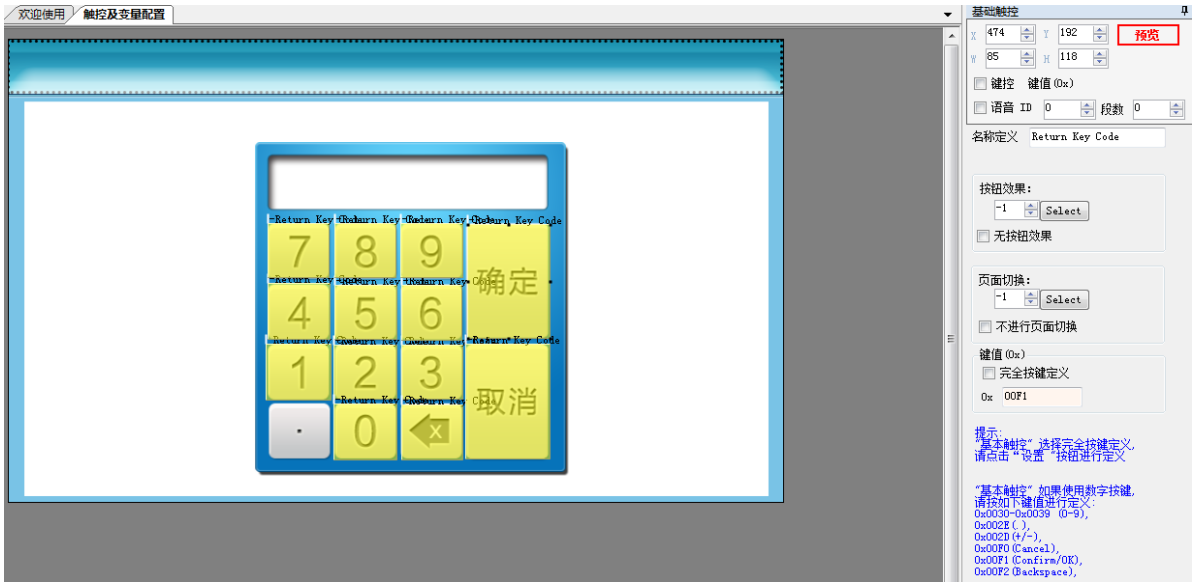
Similarly, in DWIN OS Write programs, but also pay attention to DCS005 Touch-screen controller dedicated mount a write command, you need to 0x0800 Write

0x005A .



Map 10.17 "Send OK" button to set

- **Keyboard Interface** : This interface is used in conjunction with other data entry interface, controls only the "basic control", setting example of FIG. 10.18 Shown (0-9 correspond 0x0030-0x0039 Determine a corresponding 0x00F1 Cancel the corresponding 0x00F0 , Clear the corresponding 0x00F2).



Map 10.18 Button configuration

Step5 Test modification : Since in this case the functions required by the DWIN OS With the program is implemented, so when after the completion of the planning and interface design variables and configuration needed DWIN OS Programming software for programming and an appropriate BIN file(twenty three_ motor control. BIN). Procedures are as follows:

```
; Reset 2/ Power 1/ Shutdown 0
LDWR R0, 142
MOVXR R4, 1 , 1
IJNE          R5, 0 , RS
GOTO OVER
```

```
RS:
IJNE R5, 2 , PLATFORM
; Reset
RESET:
LDWR R0, 147
LDWR R12, 0
MOVXR R12, 0 , 1          ; Platform production schedule cleared
LDWR R0, 144
LDWR R8, 0
MOVXR R8, 0 , 1           ; Production flag is cleared
LDWR        R6, 0
LDWR R0, 143
MOVXR R6, 0 , 1           ; Platform flag is cleared
LDWR R0, 165
LDWR R30, 0
MOVXR R30, 0 , 1          ; The current number of production parts cleared
; M1, M2, M3, M4 Return home
CALL ORIGIN1
CALL ORIGIN4
LDWR R14, 0x5A02
LDWR R0, 133
MOVXR R14, 0 , 1
LDWR R0, 136
MOVXR R14, 0 , 1
LDWR R0, 129
LDWR R14, 0
MOVXR R14, 0 , 1
CALL SEND
LDWR R4, 0
LDWR R0, 142
MOVXR R4, 0 , 1
GOTO OVER
; Platform Choice
PLATFORM:
; platform 1/ platform 2
LDWR R0, 143
MOVXR R6, 1 , 1
IJNE        R7, 0 , P1
; platform 1
P0:
LDWR R0, 148
LDWR R2, 160
MOVXX 2
LDWR R16, 133
LDWR R18, 134
LDWR R20, 5
LDWR R10, 145
GOTO START
; platform 2
P1:
LDWR R0, 150
LDWR R2, 160
MOVXX 2
LDWR R16, 136
LDWR R18, 137
LDWR R20, 6
LDWR R10, 146
; Power
START:
LDWR R0, 144
MOVXR R8, 1 , 1           ; Production Flag
IJNE        R9, 0 , COM1
; Determine whether the platform is in place
MOV R10, R0, 2
MOVXR R22, 1 , 1
```

```
IJNE        R23, 1 , CHANGE
LDWR R22, 0
MOVXR R22, 0 , 1        ; Platform in place flag is cleared
CALL        PLUS        ; Flag change 1
; M1, M2 / M3 To fix 1
LDWR R0, 160
LDWR R2, 131
MOVXX 2
LDWR R0, 130
LDWR R14, 0x5A04
MOVXR R14, 0 , 1
LDWR R0, 154
MOV R18, R2, 2
MOVXX 2
MOV R16, R0, 2
LDWR R14, 0x5A03
MOVXR R14, 0 , 1
CALL WAIT4
CALL SEND
GOTO OVER
; Replace the platform
CHANGE:
IJNE R7, 0 , TO0
LDWR R6, 1
LDWR R0, 143
MOVXR R6, 0 , 1
GOTO OVER TO0:

LDWR R6, 0
LDWR R0, 143
MOVXR R6, 0 , 1
GOTO OVER
COM1:
IJNE R9, 1 , COM2
; Judge M1, M2 / M3 status
LDWR R0, 4
MOVXR R24, 1 , 1
IJNE R25, 0 , OVER
MOV         R20, R0, 2
MOVXR R24, 1 , 1
IJNE R25, 0 , OVER
CALL        PLUS ; Flag change 2; M4 down

LDWR R0, 158
LDWR R2, 140
MOVXX 2
LDWR R0, 139
LDWR R14, 0x5A04
MOVXR R14, 0 , 1
CALL WAIT1
CALL WAIT2
CALL        SEND
GOTO OVER COM2:

IJNE R9, 2 , COM3
; Judge M4 status
LDWR R0, 7
MOVXR R24, 1 , 1
IJNE R25, 0 , OVER
CALL        PLUS ; Flag change 3; Vacuum
generator operation
LDWR R0, 129
LDWR R14, 1
MOVXR R14, 0 , 1
CALL WAIT4
```

```
CALL        SEND
GOTO OVER
COM3:
IJNE R9, 3 , COM4
LDWR R0, 162                    ; Used for timing the operation is completed the vacuum generator
MOVXR R28, 1 , 1
INC         R28, 1 , 1
MOVXR R28, 0 , 1
LDWR R26, 3
JS          R28, R26, OVER
CALL PLUS                       ; Flag change 4
; M4 up
CALL ORIGIN4
CALL SEND
GOTO OVER
COM4:
IJNE R9, 4 , COM5
; Judge M4 status
R0, 7
MOVXR R24, 1 , 1
IJNE        R25, 0 , OVER
CALL        PLUS ; Flag change 5; M1 Retreat

LDWR        R0, 152
LDWR        R2, 131
MOVXX 2
LDWR        R0, 130
LDWR        R14, 0x5A04
MOVXR R14, 0 , 1
CALL        WAIT4
CALL        SEND
GOTO        OVER
COM5:
IJNE        R9, 5 , COM6
; Judge M1 status
LDWR        R0, 4
MOVXR R24, 1 , 1
IJNE        R25, 0 , OVER
CALL        PLUS ; Flag change 6; M4 down

LDWR        R0, 163
LDWR        R2, 140
MOVXX 2
LDWR        R0, 139
LDWR        R14, 0x5A04
MOVXR R14, 0 , 1
CALL        WAIT1
CALL        SEND
GOTO OVER COM6:

IJNE R9, 6 , COM7
; Judge M4 status
LDWR R0, 7
MOVXR R24, 1 , 1
IJNE R25, 0 , OVER
CALL        PLUS ; Flag change 7; Vacuum
generator operation
LDWR R0, 129
LDWR R14, 0
MOVXR R14, 0 , 1
CALL WAIT4
CALL        SEND
GOTO OVER COM7:

IJNE R9, 7 , COM8
```

```
LDWR R0, 162                ; Used for timing the operation is completed the vacuum generator
MOVXR R28, 1 , 1
DEC          R28, 1 , 1
MOVXR R28, 0 , 1
LDWR         R26, 0
JS           R26, R28, OVER
CALL         PLUS ; Flag change 8;
M4 up
CALL ORIGIN4
CALL         SEND
GOTO OVER OVER:

END
COM8:
IJNE R9, 8 , COM9
; Judge M4 status
LDWR R0, 7
MOVXR R24, 1 , 1
IJNE         R25, 0 , OVER
CALL PLUS                   ; Flag change 9
; Record number of current production pieces
LDWR R0, 165
MOVXR R30, 1 , 1
INC          R30, 1 , 1
MOVXR R30, 0 , 1
; Judging platform production schedule, production is less than the number of 3 , Continue the production platform, or platform replacement
LDWR R0, 147
MOVXR R12, 1 , 1
INC          R12, 1 , 1
LDWR         R26, 2
JS           R26, R12, BACK
MOVXR R12, 0 , 1
; M1 To fix 1, M2 / M3 Retreat
LDWR R0, 152
LDWR R2, 131
MOVXX 2
LDWR R0, 130
LDWR R14, 0x5A03
MOVXR R14, 0 , 1
LDWR R0, 156
MOV R18, R2, 2
MOVXX 2
MOV R16, R0, 2
MOVXR R14, 0 , 1
CALL WAIT4
CALL SEND
GOTO OVER
COM9:
IJNE R9, 9 , STATE
; Judge M1, M2 / M3 status
LDWR R0, 4
MOVXR R24, 1 , 1
IJNE R25, 0 , OVER
MOV R20, R0, 2
MOVXR R24, 1 , 1
IJNE R25, 0 , OVER
; <3, The platform continues
LDWR R0, 144
LDWR R26, 1
MOVXR R26, 0 , 1
GOTO OVER
BACK:       ; M1, M2 / M3 Return home
CALL PLUS ; Flag 10
CALL ORIGIN1
LDWR R14, 0x5A02
```

```
MOV R16, R0, 2
MOVXR R14, 0 , 1
CALL WAIT4
CALL SEND
GOTO OVER
STATE:      ; Judge M1, M2 / M3 status
LDWR R0, 4
MOVXR R24, 1 , 1
IJNE        R25, 0 , OVER
MOV         R20, R0, 2
MOVXR R24, 1 , 1
IJNE        R25, 0 , OVER
CALL WAIT1
CALL WAIT2
CALL        SEND
; Replace the platform
LDWR R0, 147
LDWR R12, 0
MOVXR R12, 0 , 1          ; The number of production platforms cleared
LDWR R0, 144
LDWR R8, 0
MOVXR R8, 0 , 1 ; Production flag is cleared
; Change platform flag
GOTO CHANGE
; Flag + 1 function
PLUS:
INC         R8, 1 , 1
LDWR R0, 144
MOVXR R8, 0 , 1
RET
; Function confirmation command sent
SEND:
LDWR R0, 0x0800
LDWR R26, 0x005A
MOVXR R26, 0 , 1
RET
; M1 Standby function
WAIT1:
LDWR R14, 0x5A00
LDWR R0, 130
MOVXR R14, 0 , 1
RET
; M2 / M3 Standby function
WAIT2:
LDWR R14, 0x5A00
MOV R16, R0, 2
MOVXR R14, 0 , 1
RET
; M4 Standby function
WAIT4:
LDWR R14, 0x5A00
LDWR R0, 139
MOVXR R14, 0 , 1
RET
; M1 Return to origin function
ORIGIN1:
LDWR R14, 0x5A01
LDWR R0, 130
MOVXR R14, 0 , 1
RET
; M4 Return to origin function
ORIGIN4:
LDWR R14, 0x5A01
LDWR R0, 139
MOVXR R14, 0 , 1
```

RET

[Note] the DWIN OS Among programming, there are two main difficulties, it is necessary to exploit the use of the production process flag, to achieve control of the entire production process; second is to clever use of intermediate variables, M2 with M3 The control program written in the same paragraph, to reduce the program branches. Also, note that DWIN OS Each cycle, we have to re-run the program once, the flow control is particularly important. In the debugging process, through the serial port DGUS Screen and PC Link up in PC End use modbus poll
Or other host device simulation tools to monitor, debug DGUS Screen, which makes debugging more intuitive and convenient.

After completion of all of the above to be designed for use SD Card will DWIN_SET Folder (including the configuration files, pictures, etc. interface) to download DGUS Screen, and then test the system, the final version by following commissioning.

Step6 Version of Archiving: After the fixed version, the configuration files, images, fonts, icons, libraries, etc. DGUS Files stored in a covered screen SD Card can be mass production.