WiFi Module Interface User Guide V1.3.4

Contents

1. Overview

To enable DWIN smart LCD screens to connect to the internet easily, DWIN has established our own cloud server and integrated the interfaces for Internet connection based on the DGUS system. This allows developers to create network-enabled DWIN screens with minimal effort.

Currently supported LCD platforms are T5UID1, T5UID2 and T5L_ASIC. Supported products include 86 box products, network screen, etc.

The advantages of DWIN WIFI scheme are mainly reflected in the following four aspects:

1. Development Efficiency

Based on the DGUS architecture, once standalone functions are implemented, network configuration and data upload can be achieved with zero code on the MCU. Engineers only need to plan data addresses. Complex JSON protocol parsing is handled by the cloud, saving MCU resources and improving development efficiency. This allows customers to focus on core product features.

2. Stability

The solution fully integrates protocols between the main ASIC (GUI), WiFi module, and MQTT server, creating a UI+IoT integrated solution. This reduces development steps, communication costs, and uncertainties, enhancing overall system stability.

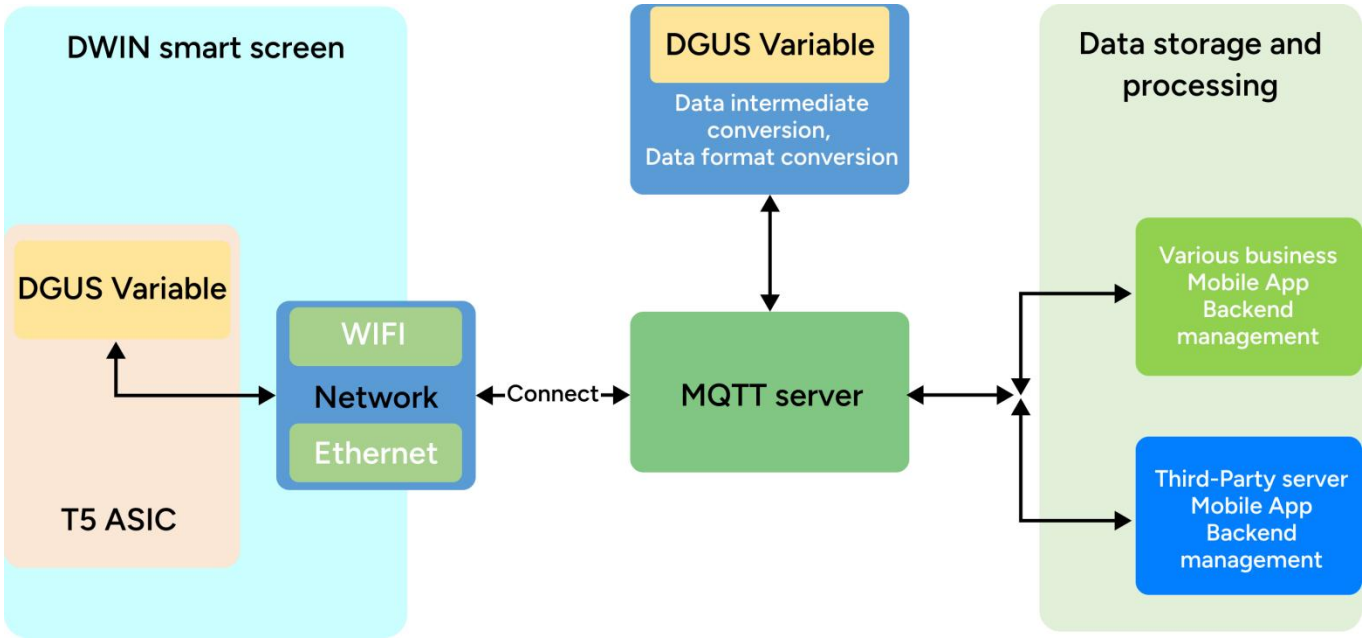
3. Cost-effectiveness

As part of DWIN's IoT ecosystem, the WiFi module benefits from scale advantages serving tens of thousands of customers across industries. Direct cooperation with original WiFi & ETHERNET chip manufacturers ensures cost competitiveness.

4. Flexible options

The solution supports both WiFi and Ethernet to meet diverse networking needs. When Ethernet is connected, the module automatically disables WiFi. If Ethernet connectivity is lost, WiFi is reactivated—providing customers with optimal network performance.

1.1 System Block Diagram



1. Through simple configuration, the smart screen equipped with the DWIN WiFi module can connect to the MQTT server through the "connection" channel.
2. The WiFi module can transfer the DGUS register variables of the DWIN smart screen and the corresponding DGUS variables on the server up and down.
3. The mobile terminal and the background management conduct inter-server data communication and interaction through the MQTT server. The mobile terminal mainly realizes the remote UI function and variable modification function. The background management can realize the functions of variable modification control and file upgrade.
4. The third-party server can be the customer's server. By providing an interface, it can operate DGUS variables and perform file upgrades.

1.2 Access Methods

The DWIN WiFi module, MQTT server, and data transfer server must be from DWIN. The servers for business data processing, storage, and handling can be either DWIN's or the customer's.

1.3 Characteristic Parameters

The features of the WiFi module are as follows:

(1)Main chip: ESP8266 + 4MB Flash

(2) Supports 802.11 (2.4 GHz), with a frequency range of 2.4G-2.5G (2400M-2483.5M)

(3) Antenna type: PCB-onboard antenna

(4) Maximum transmit power: 14dBm

(5) UART transmission rate: 921600bps (device-side data transmission rate)

(6) Ethernet card interface (For the hardware interface diagram of the Ethernet card, please refer to the eth-esp32 document)

1.4 Hardware Interface

(1). WIFI-10

The hardware encapsulation definition of WIFI-10 is shown in the following figure, with a pin pitch of 2.0mm:

This WiFi module has a total of 7 pins: VCC (3.3V), GND, TXD, RXD, C/D, TXD_C, and RST.

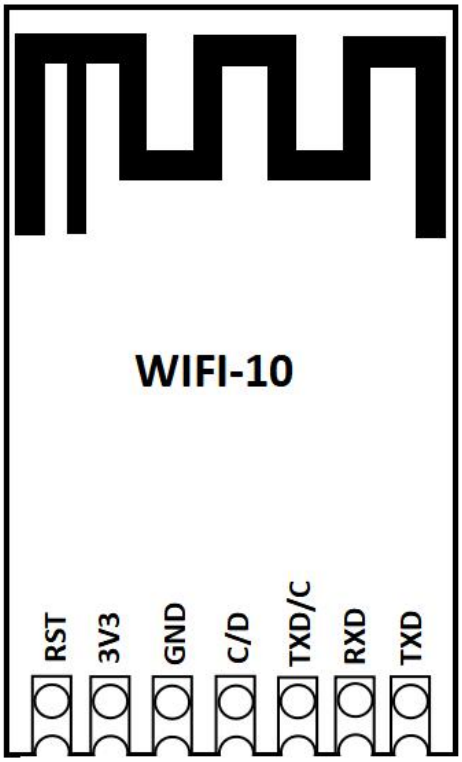
TXD, RXD: Data channels, with the baud rate of 921600.

TXD_C: A pin for printing internal information, which is not related to the business. It should be left floating.

C/D: Enables the download function of the WiFi module. There is an internal pull-up resistor of 10K. When the pin is at a low level during module power-on, it indicates entering the boot mode. It can be left floating during normal use.

3V3, GND: Need to ensure a peak current of 300mA.

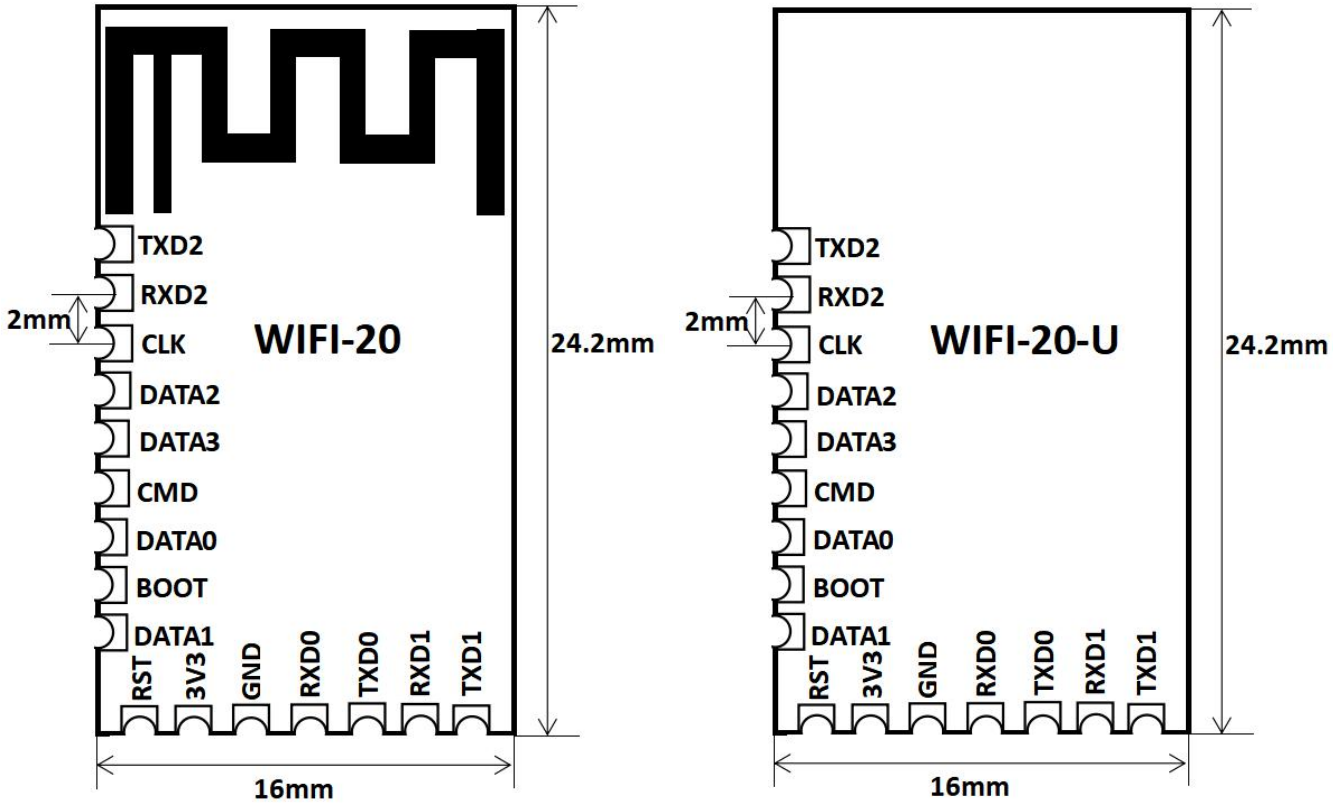
RST: The reset pin. It needs to be connected to a reset chip for reliable reset. A low-level signal resets the WiFi module.



This version uses the esp8266 chip and does not support independent connection to the client's server.

(2). WIFI-20,WIFI-20-U

The hardware encapsulation definition of WIFI-20 and WIFI-20-U are shown in the following figures:



This version is compatible with WIFI-10 pins, and adds SD card support.

For the SD card, there are CLK, CMD, DATA0, DATA1, DATA2, DATA3.

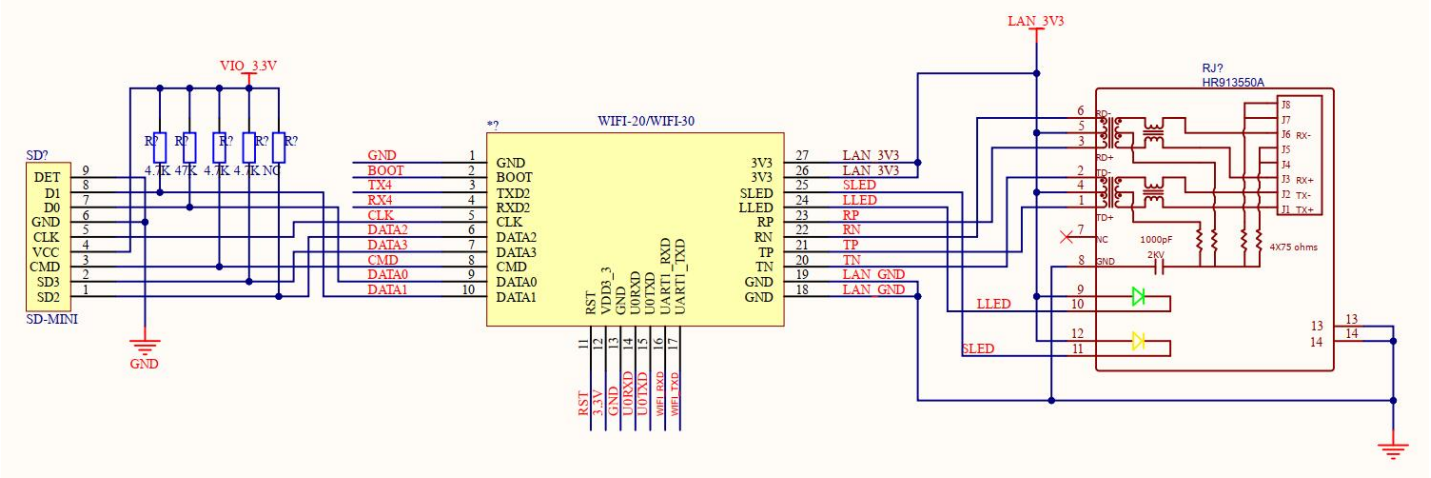
Connect BOOT to the ground to enter the programming mode.

UART: TXD2, RXD2.

WIFI-20 uses a built-in antenna.

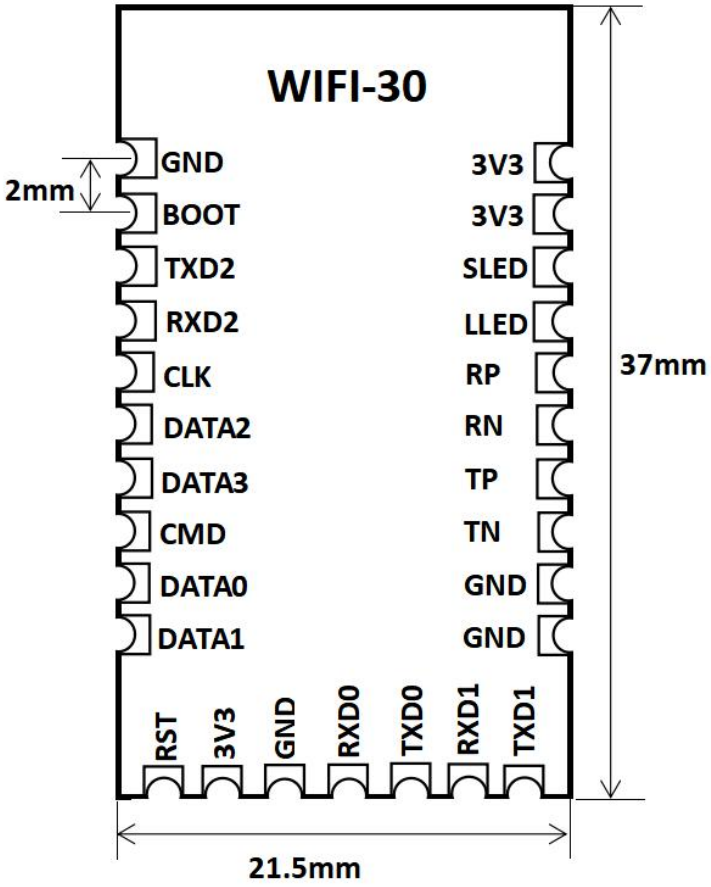
WIFI-20-U uses an IPX external antenna.

Note: Pull-up resistors are required for these pins when connecting the SD card. Otherwise, some chips may fail to start.



(3). WIFI-30

The hardware encapsulation definition of WIFI-30 is shown in the following figure:



This version supports both WIFI and Ethernet.

Note: When an Ethernet cable is detected, the WIFI will be automatically turned off.

For the SD card, there are CLK, CMD, DATA0, DATA1, DATA2, DATA3.

Connect BOOT to the ground to enter the programming mode.

UART: TXD2, RXD2.

For Ethernet: SLED indicates the sending status; LLED indicates the connection status; RP, RN, TP, TN correspond to the wiring of RJ45.

Software differences:

WIFI-10: Only supports connecting to the DWIN Cloud Server. All data is docked through the DWIN Cloud API interface. Customers need to develop their own servers.

WIFI-20: Retains the connection function with the DWIN Cloud Server as WIFI-10, and adds TCP, UDP, and WEBSOCKET connections to the customer's server.

WIFI-30: It is the Ethernet version, and also has a WIFI antenna IPX socket. It adds TCP, UDP, and WEBSOCKET connections to the customer's server.

1.4.1 T5UID1_86 platform and WIFI module Interface

The T5UID1_86A platform is an application of the standard T5UID1 platform on an 86-box. It has additional definitions for I/O and the UART used to connect to WIFI. Eventually, **only UART4 is used**, and the rest are not needed.

T5_IO	WIFI_IO	Description
IO2	C/D	Connect a resistor for hardware-configured download enable. In practice, it is not required and can be left floating.
IO3	--	Used for power-on button detection.
UART4_TX	RXD	UART
UART4_RX	TXD	UART
IO1 UART6_RX	TXD_C	For debugging information output. It can be ignored and is not actually used, so it can be left floating.

1.4.2 T5L hardware platform Interface

Only UART1 is used, and the rest are not needed.

T5L_IO	WIFI_IO	Description
IO18	C/D	Connect a resistor for hardware-configured download enable.
IO19	TXD_C	For debugging information output. It can be ignored.
UART1_TX	RX	UART data
UART1_RX	TX	UART data

2. Development Steps

2.1 Platform

Platform	Description	Example Products
T5UID1, T5UID2	1. Need to configure and load the 22.bin file. 2. The 8283 instruction needs to be implemented in the OS. The provided example code can be used.	TR028C11W04 TC035C21W04 TR028C12W00 TC035C22W00
T5L ASIC	Only need to configure and load the 22.bin file.	

2.2 Standalone Device Program Development

It is the same as the original DGUS standalone development and no special treatment is required.

2.3 Cloud Access Operations

Write the 22.bin file to meet the network access requirements, including the product ID, display of network connection status icons, display of the version number and MAC address, etc.

2.4 Mobile Terminal and Cloud Background Development

Log in to http://merchant.dwinhmi.com.cn/ for background management and develop the App.

3. DGUS Platform Interface Define

3.1 DGUS System Read-Write Interface

The DGUS system generally interacts with external systems through memory data. The memory variable space ranges from 0x0000-0xFFFF. Among them, 0x0000-0x0FFF is the DGUS system area, and the network communication interface is located in the range from 0x400-0x4FF within the system area. The area from 0x1000-0xFFFF is the user memory area.

The bus interface for external devices to read and write the DGUS system is implemented through UART, mainly using the 82H and 83H commands.

Data Block	1	2	3	4	5
Definition	Frame Header	Data Length	Command	Data	CRC Check (Optional)
Data Length	2	1	1	N	2
Description	0x5A5A	Including command, data, check	0x82/0x83	-	-
Example (with Check)	5A A5	04	83	00 10 04	-
Example (without Check)	5A A5	06	83	00 10 04	25 A3

Command	Data	Description
0x82	Transmission: Variable space start address (0x0000-0xFFF) + written data	Write a data string (word data) starting from the specified address to the variable space. Do not write to the system-reserved space.
	Response: 0x4F 0x4B	Write command response
0x83	Transmission: Variable space start address (0x0000-0xFFF) + length of read word data (0x0-0x7D).	Read word data of the specified length starting from the specified address in the variable space.
	Response: Variable space start address + variable data word length + read variable data.	Data response.

It should be noted that the maximum length of a single 82H/83H command is 256 bytes, after removing the frame header, frame trailer, and CRC, the typical maximum payload is around 240 bytes. Therefore, when a large amount of data needs to be interacted with the DGUS system, it is necessary to split it into small packets for interaction. For example, if 4K of data from the cloud is sent to the WIFI module, the WIFI module needs to split it into small packets and write them into the DGUS system variables one by one.

3.2 Definition of Network Interface in DGUS System

The definition addresses of the network part in the DGSU system are in the 0x400-0x4FF area, and the definitions are as follows.

Definition	Address	Length	Description	Recommended Value
Network Interface Switch	0x400	1	0x5AA5 indicates that the network communication interface is enabled.	5AA5
RMA Alarm	0x401	3	D5:D4 = 0x5AA5 (MQTT)/0x7AA7 (UDP)/0x7AAC (TCP with format header)/0x7ABC (TCP-MODBUS) indicates to immediately start uploading the data in the specified RMA space to the server, and clear it after processing. D3:D2 = The word address of the RMA variable memory to be uploaded. D1:D0 = The word length of the RMA variable memory to be uploaded.	All 00
Reserved	0x404	12	Reserved.	All 00
Device Description	0x410	1	The high-byte = 0x5A indicates that the device description is valid. The low-byte = The encoding method and length of the device description text. .7 - .6: The encoding method, 0x00 = UNICODE, 0x01 = GBK, and GBK is recommended. .5 - .0: The length of the description text is 0x00 - 0x34.	5A45
	0x411	2	4-Byte device manufacturer ID, allocated by DWIN. The segment 0xFFFF:0000 - 0xFFFF:FFFF is reserved.	00000001
	0x413	1	2-Byte classification of each manufacturer's device. Device manufacturers classify them by themselves according to DWIN's classification standards.	0001
	0x414	2	4-Byte individual device number under each type of device. Manufacturers number them by themselves.	Custom - defined
	0x416	26	Device description text, up to 52 Bytes.	DGUS2_T5UID1
RAM Mapping	0x430	1	The time interval for the RMA to auto-refresh the server, 0x0000 - 0xFFFF, with the unit of 0.1 seconds. 0x0000 indicates that no automatic refresh is required.	012C

Definition	Address	Length (word)	Description	Recommended Value
RAM Mapping	0x431	1	Starting address of RMA (variable memory mapped to server) read space, word address. Reading is not allowed if out of bounds.(Due to limited space, only read space is unrestricted)	1000
	0x432	1	Size of RMA read space (unit: 128 Words). Current maximum setting is 16 (4 KBytes). 0x0000 indicates read prohibited.	0004
	0x433	1	The starting address of the RMA write space. It can overlap with the read space, and writing is not allowed if it exceeds the boundary.	2000
	0x434	1	Size of RMA write space (unit: 128 Words). Current maximum setting is 16 (4 KBytes). 0x0000 indicates write prohibited.	0004
	0x435	3	Reserved.	All 00
Remote Upgrade	0x438	1	Remote upgrade interface enable. 0x5AA5 indicates that the device enables the remote upgrade interface, and the system will automatically detect the Buffer.	5AA5
	0x439	1	Configuration of the timeout timer for remote upgrade packets, (unit: 0.1 seconds).	0064
	0x43A	4	Definition of the first remotely upgradable space of the device (aligned to 4KB)	
			D7=0x5A indicates that this remote upgrade space is enabled. D6 -D3: 32-bit starting address of the upgradable space (with the lower 12 bits being 0), with a maximum of 4GB.	5A 0001 0000 0000F0
			D2 - D0: The size of the upgradable space, with the unit of 4KB and a maximum of 4GB.	
	0x43E	4	Definition of the second remotely upgradable space of the device.	5A 0000 0000 008000
	0x442	4	Definition of the third remotely upgradable space of the device.	All 00
	0x446	4	Definition of the fourth remotely upgradable space of the device.	All 00

Definition	Address	Length (word)	Description	Recommended Value
Remote Upgrade	0x44A	2	Definition of the remote upgrade buffer interface	
			D3 = 0x5A indicates that the remote upgrade buffer is valid. D2 = Upgrade mode. 0x00 means the communication terminal is responsible for verifying the data CRC. If there is an error frame, it will notify the host to resend. D1 = The number of buffers available for remote upgrade, ranging from 0x01 to 0x10, with a maximum of 16. D0 = The high byte of the starting address (word address) of Buffer0. The lower 8 bits are 0x00.	
			Each buffer fixedly occupies 2,304 words (0x900) and is arranged in sequence.	5A
			Definition of a single buffer (the first 512 bytes are the control interface, and the subsequent 4KB are the data): D0: 0x5A indicates starting a remote upgrade for this buffer, which will be cleared to zero after being processed by the CPU. D1: Selection of the remote upgrade space, 0x00 - 0x03, representing one of the four remote upgrade spaces.	00 08 B000
			D2 - D5: Write the target address of the remote upgrade space, with the lower 12 bits being 0 (aligned to 4KB).	
			D6 - D7: Data byte length, 0x0001 - 0x0FFF.	
			D8 - D9: Data CRC checksum.	
			D10 - D511: Reserved.	
			D512: Data starts, with a maximum of 4096 bytes.	
	0x44C	4	Reserved.	All 00
Reserved Device Description	0x450	48	QR code of the device.	All 00

Definition	Address	Length (word)	Description	Recommended Value
Description of Communication Equipment	0x480	16	D31: 0x5A indicates that the communication device description data is valid. D30: Encoding method and length of the device description text. .7 - .6: Encoding method. 0x00 = UNICODE, 0x01 = GBK. It is recommended to use GBK. .5 - .0: Length of the description text, ranging from 0x00 to 0x14. D29: Communication device category. 0x01 = Wi-Fi. D28: Communication device status, bit definition. .7: Device working status. 0 = Configuring, 1 = Normal; .6 - .3: Undefined, write 0; .2 - .0: Signal quality, 0x00-0x07, with a total of 8 levels. 0x00 indicates the worst, and 0x07 indicates the best. D27 - D22 = 6-Byte MAC address of the communication device. D21 - D20: Reserved. D19 - D0: Communication device description information in text format, with a maximum of 20 Bytes.	User-defined
Fixed IP Flag Setting	0x485	2	d1 - d0: d1 = 0x01 indicates enabling the fixed IP for Wi-Fi. d0 = 0x01 indicates enabling the fixed IP for Ethernet.	
Reserved Communication Device	0x490	112	Customization of the communication device, such as the username and password for Wi-Fi.	User-defined

The device description information is configured by the 0x0800 - 0x09FF in the 22 initialization file (regardless of whether the variable buffer function of 22 initializing file is enabled).

4. WIFI Memory Interface Definition

The addresses 0x400-0x47F are the DGUS system communication interface, and the addresses 0x480-0x4FF are the function and status interfaces of specific communication modules or devices.

Different network modules and devices have different definitions. The definition of the WIFI module is as follows.

Definition	Address	Length	Description	Recommended Value
Communication Device Description	0x480	16	D31: 0x5A indicates that the communication device description data is valid. D30: Encoding method and length of the device description text. .7 - .6: Encoding method. 0x00 = UNICODE, 0x01 = GBK. It is recommended to use GBK. .5 - .0: Length of the description text, ranging from 0x00 to 0x14. D29: Communication device category. 0x01 = WiFi D28: Communication device status, bit - defined .7: Device working status. 0 = Configuring, 1 = Normal; .6 - .3: Undefined, write 0; .2 - .0: Signal quality, 0x00-0x07, with a total of 8 levels. 0x00 indicates the worst, and 0x07 indicates the best. D27 - D22 = 6 Bytes: MAC address of the communication device. D21 - D20: Reserved. D19 - D0: Communication device description information in text format, with a maximum of 20 Bytes. Among them, D19:D18 are fixed - bit indicators “OK”, and D17 - D0 are the version numbers of the wifi module.	User-defined
Start Fixed IP Flag	0x485	2	d1 - d0: d1 = 0x01 indicates enabling the fixed IP for Wi - Fi. d0 = 0x01 indicates enabling the fixed IP for Ethernet.	
WIFI Hardware Parameters Setting Hard_Set	0x490-0x494	8	It is effective when restarting the WIFI module and is used during the restart initialization process. [D15:D14] Baud rate setting reservation. Currently fixed at 9216(00), not configurable. D13 - D9: Reserved for storing the local IP address. D13=5A indicates that the IP address is valid. D12 - D9: IP address XX.XX.XX.XX (e.g., 192.168.10.150 in decimal). D8: DEBUG firmware of switching the WIFI module . 5A is effective. [D7:D0] WIFI module model. Default is “TT - DW10”.	

Definition	Address	Length	Description	Recommended Value
WIFI Configuration Setting Cfg_set	0x498	4	D7: 5A indicates starting the WIFI network configuration. After the configuration is completed, this value is cleared to 0. D6=0x5a indicates starting the connection to the router using the network name and password method (the network name and password are stored at 0x4B0). D6=0xa5 indicates starting the scanf scanning function, and the results are placed at 0x4D0. For D7 - D6, once started, D7 - D6 will be cleared simultaneously. The two Internet access methods cannot be started simultaneously. If both are 5A, one - key network configuration takes precedence. D5: Network time synchronization enable. 00 or 5A indicates enabling the DGUSII with RTC(0x9C). RTC library(0xF430). 7A indicates using network time (write to 0x9C). D4: 0x5A indicates that the UDP port setting is valid. D3 - D2: UDP port number. [D1] WIFI module automatic upgrade. 5A indicates enabling. [D0] Soft reset. 5A=starting the reset. After the reset is completed, it clears to 0.	
IP Write to WIFI Save Setting	0x498	1	[D0] b5=writing the wifi's IP into the wifi module. b7=writing the fixed Ethernet IP into the wifi module.	
Enable and Disable IP Setting Flag	0x49B	2	[D0] b5 indicates enabling wifi fixed IP. b6 indicates not enabling wifi fixed IP. b7 indicates enabling fixed Ethernet IP. b8 indicates not enabling fixed Ethernet IP.	
Electronic Table Card	0x49C	4	D7: 5A indicates starting the electronic table card broadcast, once every 5 seconds. D6: Indicates the current download progress. 0= Start of download. 1-99= Download in progress. 100= Download completed. 101= Download failed. D5D4: Indicate the ID of the electronic table card. D3: 5A indicates enabling weather forecast. D1D0: The current update number of the T5L file.	

Definition	Address	Length	Description	Recommended Value
Network Status	0x4A0	8	[D15:D14] (0x4A0) Reserved. The Wi-Fi version number is displayed in ASCII code at the position 0x487. [D13:D12] (0x4A1) Network configuration status feedback. 00: Network not configured. 01: Start network configuration. 02: Network configuration in progress. 03: Network configuration succeeded. 0F: Network configuration failed. [D11:D10] (0x4A2) Network connection status. 00: Username and password not obtained. 01: Wi-Fi router connection succeeded. 02: Wi-Fi module self-upgrading. 03: Connecting to the server. 04: Logged in to the server. 05: Connected to the Cloud. [D9:D8] (0x4A3) Network strength (same as 0x481). [D7:D6] Wi-Fi connection to the router error. 01: Router does not exist. 02: Password is incorrect. 03: Connection timed out. [D5:D2] Reserved. [D1:D0] (0x4A7) Hold counter. It will be incremented by 1 in each cycle (<1s). If the DGUS finds that this value hasn't changed for a long time, it can be considered that the Wi-Fi module has crashed, so as to carry out the next step of processing.	
Program Running Status	0x4A8	2	D3: State machine D2: UART state machine	
Esp32 Memory Limitation and Network Function Selection	0x4AA	2	[D3:D0]: Direct connection to the client-server. D2: 1=Enabling CRC16 check for communication with T5L. D1: 0=Not connecting to DWIN Cloud; 1=Connecting to DWIN Cloud, which means closing MQTT to connect to DWIN Cloud. D0: Different bits indicate whether different functions are enabled. 0 means enabled, 1 means disabled. D0 -> 0: 0 indicates enabling TCP, 1 indicates disabling TCP. D0 -> 1: 0 indicates enabling WebSocket, 1 indicates disabling WebSocket. D0 -> 2: 0 indicates enabling common MQTT, 1 indicates disabling common MQTT.	

Definition	Address	Length	Description	Recommended Value
Network RTC	0x4AC	4	D7: 5A indicates the time is valid. D6 - D0: Year, month, day, weekday (0 - 6), hour, minute, second.	
Network Parameters Router_set	0x4B0	32	0x4B0: SSID, must end with 0xFF. 0x4C0: SN, must end with 0xFF.	
	0x4D0	48	0x4D0: Save, the SSID names separated by \n.	

Registers where DGUS sends commands to the WIFI module, such as 400H, 490H, etc., will be actively read by the WIFI module every 0.1s-0.2s. The status registers of the WIFI module, such as 480H, 4A0H, will be written to the DGUS registers by the WIFI module every 3-5 seconds.

Operation Instructions

1.Edit the 22.bin file template and modify the values you need.

2.Router network name and password settings: Write 005AH to the 498H register to enable, and set the SSID and password at 4B0H. Alternatively, you can start one-key network configuration in the application without setting these values.

3.T5UID1 platform: You need to add the instruction code of 8283 to the OS and use an interrupt to detect the WIFI serial port. Refer to the example for details. The T5L platform does not require any processing.

UI Operation Instructions (Icons are needed for status indication)

1.One-key Network Configuration Page on Mobile Phone

(1).Touch operation: Press the "Start One-key Network Configuration" button (writing 0x5A00 to the address 0x498 when the button is pressed).

(2).Display:

- One-key network configuration status icon (Address 0x4A1: 00 means not configured; 01 means starting network configuration; 02 means network configuration in progress; 03 means network configuration succeeded; 0F means network configuration failed).
- WIFI network connection status icon (Address 0x4A2: 00 means username and password not obtained; 01 means successful connection to the WIFI router; 02 means the WIFI module is self-upgrading; 03 means connecting to the server; 04 means logged in to the server; 05 means connected to the Cloud).
- WIFI module MAC address (Addresses 0x482-484: Hexadecimal display, 12-byte length ASCII display).
- WIFI module version number (Address 0x487: Displayed as the ASCII string with the length of 18 bytes).

2.Manual Input of Network Name and Password

(1).Touch operation:

- Text input for network name at address 0x4B0. Text input for network password at address 0x4C0.
- Press the "Start Network Connection" button(“Return Key Code”control, write 0x005A to the address 0x498).

(2).Display:

- WIFI network connection status icon (Address 0x4A2: 00 means username and password not obtained; 01 means successful connection to the WIFI router; 02 means the WIFI module is self-upgrading; 03 means connecting to the server; 04 means logged in to the server; 05 means connected to the Cloud).
- WIFI module MAC address (Addresses 0x482-484: Hexadecimal display, 12-byte length ASCII display).
- WIFI module version number (Address 0x487: Displayed as the ASCII string with the length of 18 bytes).

After that, the WIFI module will automatically connect to the Internet and start data synchronization.

Working Process of the Serial Port Side of the WIFI Module

1.Read the configuration information: After the WIFI module is powered on, it reads the configuration information from 0x400 to 0x4FF. After successfully reading it, it enters the next state.

2.Status between the WIFI module and DGUS:

a. Lowest priority 0: Read DGUS variables (Instruction 83) every 0.1 to 0.2 seconds. Addresses such as 400H and 490H are involved in the instructions issued by DGUS. Write the necessary status of the WIFI module into DGUS through Instruction 82 every 3 to 5 seconds.

b. Second lowest priority 1: Check the synchronization timer for the server interface reached its threshold, and then perform the periodic upload of specified large data.

c. Second lowest priority 2: Detect whether the RMA alarm interface is enabled, and then perform the trigger-based upload of specified large data.

d. Second highest priority 3: Trigger-based upload to DGUS for network configuration, network timing, and changes in network status. For example, during network configuration and network login, once the status changes, it will be immediately written into DGUS.

e. Highest priority 4: File upgrade. After receiving the first packet of the file upgrade package, suspend the scheduling of ordinary tasks, and try to ensure that resources are sent to DGUS via MQTT and the serial port. During the process of sending the packets to DGUS, other Instructions(82H/83H) are not allowed to be generated.

Working Process of the MQTT Side of the WIFI Module

1.Power on and initialize, and wait to obtain the 512 bytes configuration word.

2.Wait to connect to the router.

3.Determine whether to upgrade the WIFI module. If the WIFI module needs to be upgraded, upgrade it. After the upgrade is successful, it will automatically restart.

4.Access the REG registration API through the HTTP protocol to obtain the username and password for logging in to the MQTT server.

5.Re-establish the connection and log in to the MQTT server with the username and password.

6.Subscribe to the topic channel and send an online notification.

7.After the data channel is established, you can perform read and write instructions of 82H and 85H with the server for register synchronization, and perform file Buffer operations through Instructions 92H and 93H.

5. Communication between WIFI and the Server

The maximum buffer for one frame in the WIFI module is 5K (5120), which means the maximum packet length between the WIFI module and the server is 5K.

5.1 Overview

1. Currently, there is no response mechanism for data synchronization in terms of instructions. Why?

(1)Data is centered around the device, and the device's data serves as the basis. The device itself has an automatic timed reporting mechanism. The data on the mobile-phone side is only used for UI display and cannot reflect the real status of the device.

(2)The data transmission adopts the mechanism of transmitting all data in one packet. That is, when the device sends data to the server, each frame tries to transmit all the data.

(3)If the mobile phone controls an operation and the device fails to receive it, the mobile-phone side can first lock the display data in the request state. After a timeout (for example, 5 seconds), it can display the real value to overwrite the display value. The data on the mobile-phone side is considered unreliable data, and the real data will always be synchronized through the device's reporting.

(4)The non-response mechanism simplifies the synchronization mechanism of each process, making complex problems easier. Eventually (if the automatic reporting interval is 30 seconds), all display terminals will show the status of the device.

2. Without a response, how is data convergence achieved?

Data automatic convergence is mainly achieved through the device's active reporting and regular reporting.

(1)When the server writes data to the device, there will be no response.

(2)However, after the server writes data to the device, it may immediately receive new data. This is because after the device detects a change in variables, its automatic reporting function is triggered, and it sends Instruction 0x85, which is used to upload variables of the entire block.

(3)If the data sent by the server to the device is not updated, there will be no automatic reporting. This instruction itself has no response. If the device does not detect any variable changes, it will not trigger immediate reporting, so the server will not receive Instruction 0x85.

Currently, the instructions supported between the WIFI module and the server side are as follows:

1.The server writes data to the device using 0x82. When the WIFI-ETH module receives Instruction 82 from the server, it writes data to the device. Currently, there is no response for this instruction (if a response is required, according to the protocol, it should be 0x83, and the response should be “OK” (4F 4B)).

2.The server requests to read data from the device using 0x84. The WIFI-ETH module receives the instruction of type 0x84 from the server. After receiving this instruction, the WIFI-ETH module replies to the server with Instruction 0x85 as a response, and attaches the data at the same time. After the server receives Instruction 85, it needs to write the data into the mapped memory block of the server.

3.The device writes data to the server.

4.The device requests to read data from the server.

In fact, ultimately, data only needs to be transmitted from the device to the server or from the server to the device. The function described in 3 is replaced by Instruction 0x85, and the function described in 4 is replaced by Instruction 0x82.

Finally, only Instructions 0x82, 0x84, and 0x85 are retained. Instruction 0x82 realizes the data transfer from the server to the device, Instruction 0x85 realizes the data transfer from the device to the server, and Instruction 0x84 enables the server to actively query the registers of the device.

5.2 Frame Structure

5.2.1 RAM Write

RAM write operation: Mainly uses RAM write, specifically referring to the server writing to the device's RAM.

Header (2B)	Remaining Frame Length (2B)	Frame Type	Address (2B)	Word Data Stream
AA 55	Maximum: 5117	0x82: Write	XXXX	

RAM write response (currently not implemented).

Header (2B)	Remaining Frame Length (2B)	Frame Type	Address (2B)	Word Data Stream
AA 55	Maximum: 5117	0x83: Write Response	XXXX	4F4B

5.2.2 RAM Write

RAM write operation: Mainly uses RAM write, specifically referring to the server writing to the device's RAM.

Sender:

Header (2B)	Remaining Frame Length (2B)	Frame Type	Address (2B)	Length (2B) (word)
AA 55	5	0x84: Read		

RAM write response (currently not implemented).

Responder (This response can be used as an instruction for the device's active reporting):

Header (2B)	Remaining Frame Length (2B)	Frame Type	Address (2B)	Word Data Stream
AA 55	Maximum: 5117	0x85: Read Response		

5.2.3 Upgrade BUF Frame

Server to Device

Header (2B)	Remaining Frame Length (2B)	Frame Type	BID (2B)	BUF Structure
AA 55	Maximum: 5117	0x92: Upgrade BUF Write	BUF ID	D0: 0x5A indicates starting a remote upgrade for the buffer, which is cleared to zero after CPU processing. D1: Remote Upgrade Space Selection. 0x00–0x03, representing one of the four remote upgrade spaces. D2–D5: Write the target address for the remote upgrade space, with the lower 12 bits set to 0 (aligned to 4KB). D6–D7: Data byte length, 0x0001-0x1000. D8–D9: Data CRC checksum. D10–D511: Reserved. D512: Data starts, with a maximum of 4096 bytes.

RAM write response

Header (2B)	Remaining Frame Length (2B)	Frame Type	BID(2B)
AA 55	0x02	0x92: Upgrade BUF Write	BUF ID

5.3 Initialization Process

WIFI and Server Operation Flow: (Mainly Communicating via DGUS Business Topic)

1.After the WIFI module is powered on, it uses the content of addresses 0x411–0x413 as the PID and the MAC address as the DID to register with the registration server, obtain login credentials, and then log in to the server.

2.Subscribes to the DGUS business topic.

3.Writes data from addresses 0x400–0x4FF in DGUS to the server via the DGUS business topic.

4.Once initialization is complete, all subsequent data within the topic will follow the defined data frame structure.

5.4 Data Communication

RMA Operations: Frame types are 0x82 or 0x83.

Communication with Server: The frame header is 0xAA 55 data stream.

Upon receiving the server's data stream, the WIFI-ETH module converts it into the DGUS frame structure for reading/writing to the DGUS system.

5.5 BUF Upgrade

The server converts corresponding addresses (4G space) according to DGUS file rules, and constructs a BUF application frame.

The entire BUF frame is embedded as the payload within the communication frame.

When the WIFI module receives the BUF frame, it responds with the BUFID.

5.6 Usage of UDP485 Bus

1) Set the address to be uploaded at 0x402, for example: 0x3800.

2) Set the length of the data to be uploaded at 0x403, for example: 0x0010.

3) Set 0x401 to 0x7AA7. It will be automatically cleared after the transmission is completed.

The data will be broadcast via UDP to all devices in the local area network. Other devices will acquire the data and write it to the address 3800 on their own machines, overwriting the original data. How to use the specific data is determined by the devices.

5.6.1 Write Frame Structure of UDP Bus

UDP write operation: Mainly use RAM write to write to the device's RAM.

Header (2B)	Remaining Frame Length (2B)	Frame Type	Address (2B)	Word Data Stream
AA 55	Maximum: 1024	0x72: Write	XXXX	

The User Datagram Protocol (UDP) is designed to enable a host to identify multiple destination addresses while allowing multiple applications on the same host to operate independently in sending and receiving data packets.

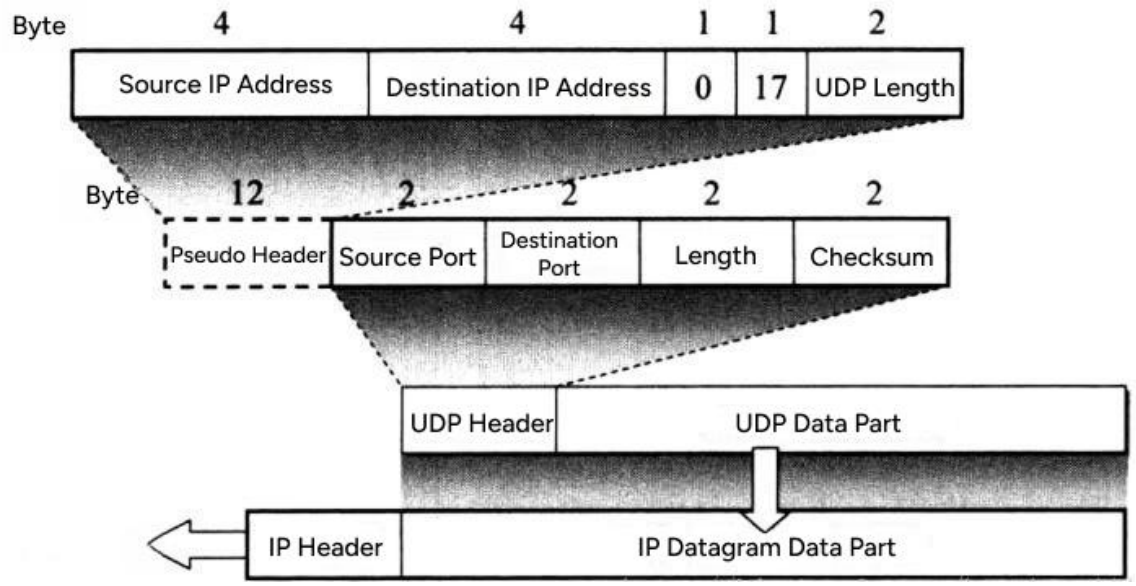
UDP uses the underlying Internet Protocol to transmit messages. Similar to IP, it provides unreliable, connection - less packet transmission services. It does not offer functions such as message arrival confirmation, sorting, and flow control.

The UDP Helper can relay and forward broadcast messages on a specified UDP port. That is, it converts the broadcast messages on the specified UDP port into unicast messages and sends them to the specified server, acting as a relay.

5.6.2 Message Format

In the UDP protocol hierarchical model, UDP is located above the IP layer. Applications access the UDP layer and then use the IP layer to transmit datagrams. The data part of an IP packet is a UDP datagram. The IP header indicates the addresses of the source and destination hosts, while the UDP header indicates the source and destination ports on the hosts. The UDP segment consists of an 8-byte header and a payload field.

The UDP header is composed of 4 fields, each of which occupies 2 bytes. Specifically, it includes the source port number, destination port number, packet length, and checksum.



Introduction to the Local Area Network Device Protocol

Hardware devices formulate the local area network UDP protocol according to actual business needs. The UDP protocol format has two modes: read and write. When the uppe -level computer (Android, iOS, or PC) sends a command, the MCU will provide corresponding feedback data.

When the host computer sends a write command (0x72), the device returns a command (0x73), which is the feedback of the written data content and has the same content.

When the host computer sends a read command (0x74), the device returns a command (0x75), which is the content of the read data, that is, the value at the read memory address.

5.6.3 UDP Read/Write Protocol Format

Own MAC address (12 bytes) + instruction header AA55 (2 bytes) + subsequent content length (2 bytes) + command word (1 byte) + read/write address (2 bytes) + content (no content for read instructions).

12-byte MAC address, corresponding to the ASCII code of the MAC address content.

Command word: Write command 0x72->0x74; Read command 0x73->0x75.

5.6.4 Device Broadcast

Devices automatically send discovery instructions via broadcast on the default port 5556 (user-configurable).

The configured UDP port + 1. The configured UDP port is the receiving port.

Device Self-Discovery (Broadcast) `01 deviceNum,clientIP,clientPort,cardId`

Own MAC address (12 bytes) + instruction header AA55 (2 bytes) + subsequent content length (2 bytes) + command word 0x01 (1 byte) + character-type content `deviceNum,clientIP,clientPort,cardId`

5.6.5 UDP Communication Flow

The device sends the device automatic reporting and discovery instruction once every 5 seconds. The host computer receives it on port 5556. If the host computer fails to receive the instruction for more than 5 times, it indicates that the device is offline.

When the device sends read and write instructions of 0x72 and 0x73, it will provide corresponding feedback data.

5.7 Definition of the Weather Forecast Protocol

Weather data (The network module writes into the vp variable, and the starting position of the weather data is 0x0780).

5.7.1 The network module actively obtains weather information

Weather detail data (The red part is for reporting, and the green part is the weather data issued by the server):

		Basic Information	Current Day Weather	+1 Day Weather	+2 Day Weather	+3 Day Weather	+4 Day Weather	+5 Day Weather
	0x0780	0x0790	0x07A0	0x07B0	0x07C0	0x07D0	0x07E0	0x07F0
0	Flag bit, Time zone	Flag bit, Flag bit	Flag bit, Year	Flag bit, Year	Flag bit, Year	Flag bit, Year	Flag bit, Year	Flag bit, Year
1	Language, encoding	Time zone, Year	Month, Day	Month, Day				
2	Postal code	Month, Day	Weather	Weather				
3		Hour, Minute	Temperature	Max. Temperature				
4		Language, Encoding	Apparent temperature	Min. Temperature				
5		City name	Relative humidity	Relative humidity				
6	International area code		Wind direction, Wind force	Wind direction, Wind force				
7			Air quality index	Air quality index				
8			PM2.5	PM2.5				
9								
A								
B								
C								
D								
E								
F								

Note: All reserved data is replaced with 00.

Time zone: Signed char, range [-12 to +12], default is +8, which is the Chinese time zone.

Language: Unsigned char, range [0 - 255], default is 1. Language codes: 1. Chinese; 2. English; 3. Spanish; 4. French; 5. Arabic; 6.

Encoding: Unsigned char, range [0 - 255], default is 3. Encoding codes: 1. ASCII; 2. UTF - 8; 3. GBK.

Postal code: ASCII string.

Estimated area code: Unsigned two-byte integer.

Flag bit: All use 5A, other values are invalid.

Year, month, day, hour, and minute: Use the time in the corresponding time zone. Unsigned char.

For example, 0C0C1E0C04 represents 12:04 on December 30th, 2012.

City name: A string in the specified language and encoding, with a maximum of 22 bytes.

Weather: 2-byte Unsigned Integer

Code mapping table for weather phenomena

Code	Chinese	English	Icon
0	晴（国内城市白天晴）	Sunny (Domestic city,day)	
1	晴（国内城市夜晚晴）	Sunny (Domestic city,night)	
2	晴（国外城市白天晴）	Sunny (Foreign city,day)	
3	晴（国外城市夜晚晴）	Sunny (Foreign city,night)	
4	多云	Cloudy	
5	晴间多云	Partly Cloudy	
6	晴间多云	Partly Cloudy	
7	大部多云	Mostly Cloudy	
8	大部多云	Mostly Cloudy	
9	阴	Overcast	
10	阵雨	Shower	

Code	Chinese	English	Icon
11	雷阵雨	Thundershower	
12	雷阵雨伴有冰雹	Thundershower with Hail	
13	小雨	Light Rain	
14	中雨	Moderate Rain	
15	大雨	Heavy Rain	
16	暴雨	Storm	
17	大暴雨	Heavy Storm	
18	特大暴雨	Severe Storm	
19	冻雨	Ice Rain	
20	雨夹雪	Sleet	
21	阵雪	Snow Flurry	

Code	Chinese	English	Icon
22	小雪	Light Snow	
23	中雪	Moderate Snow	
24	大雪	Heavy Snow	
25	暴雪	Snowstorm	
26	浮尘	Dust	
27	扬沙	Sand	
28	沙尘暴	Duststorm	
29	强沙尘暴	Sandstorm	
30	雾	Foggy	
31	霾	Haze	
32	风	Windy	
33	大风	Blustery	

Code	Chinese	English	Icon
34	飓风	Hurricane	
35	热带风暴	Tropical Storm	
36	龙卷风	Tornado	
37	冷	Cold	
38	热	Hot	
99	未知	Unknown	

Temperature: 2-byte Signed Integer in Degrees Celsius

Wind direction: 1 E, 2 ESE, 3 SE, 4 SSE, 5 S, 6 SSW, 7 SW, 8 WSW, 9 W, 10 WNW, 11 NW, 12 NNW, 13 N, 14 NNE, 15 NE, 16 ENE.

Wind force: Unsigned char.

Air quality index: 2-byte unsigned integer.

PM2.5: 2-byte unsigned integer.

The device actively obtains the weather (up to 4 times a day): 0x0401 Write 2aa2

5.7.2 Obtain the Payment QR Code (the QR code is located at 0x0760)

The device obtains the payment QR code: 0x0401 Write 2aa3

5.7.3 Obtain the Weather Configuration QR Code Address (the QR code is located at 0x0760)

The device's weather configuration QR code: 0x0401 Write 2aa4

5.8 TCP Data Transparent Transmission Protocol

TCP data transparent transmission (the network module writes to the VP, with the starting address for TCP data transparent transmission set as 0x0730) can be divided into two types based on the header format:

1.Standard format defined by DWIN: AA55 + transparent data length (two bytes) + transparent data.

2.Custom format defined by the client: Used for sending data with self-defined header formats.

5.8.1 Address 0x730: DGUS System Communication Interface

Definition	Address	Length	Description	Recommended Value
TCP Data Transparent Transmission	0X730 0X731 0X732	16	D31-D30: 0x5AA5 indicates starting a TCP service connection. Connect once. D29: Unused. D28: Connection status: 0x00: Not connected 0x02: Connecting 0x03: Connection successful 0x04: Connection is broken D27-D24: Store the TCP IP address. D23-D22: Store the TCP port number. D21-D20: Specify the address for TCP return data. If not filled in, the default is 0xD400. D19-D18: 0x5AA5 indicates that the TCP data writing is successful. D17-D16: Store the length of the data written to TCP, which should not exceed 1024. The buffer provided by the OS is 1024 bytes long. D15-D14: Whether to enable heartbeat detection: 5A; The byte following 5A represents the maximum interval time in minutes. If not set, it is 00, and the default maximum interval is 3 minutes. (For example, 05 represents a maximum time of 5 minutes. If no heartbeat response is received within 5 minutes, the TCP connection will be broken and re-established.) D13-D0: Reserved.	User-defined

5.8.2 Usage of TCP Transparent Transmission

1. You need to set the TCP IP address at 0x732-0x733. For example: 0xC0A80AD1 (decimal: 192.168.10.209).
2. You need to set the TCP PORT address at 0x734. For example: 0x1A0A (decimal: 6666).
3. You need to set the address for receiving data sent by the TCP service at 0x735: 0xD400. This address can be configured in the 22.bin file.
4. To start the TCP connection, you need to set 0x730 to 0x5AA5, and the device will automatically connect to the specified service address.
5. Receive data from the TCP service and save the data to the address set in step 3 (the maximum length of a single-received data is 4kb).
6. After receiving the data from the TCP service, write the length of the received data into 0x737. At the same time, write the data-writing success flag, setting 0x736 to 0x5AA5.
7. The device connected to T5L processes the above mentioned data. If there is a need to return data after processing, perform the following operations:
   1) Set 0x402 to the address of the data to be reported. For example: 0xD400.
   2) Set 0x403 to the length of the data to be reported. For example: 0x0010.
   3) Set 0x401 to 0x7AAC for sending. After the sending is completed, the value of 0x401 will be automatically cleared.
   4) The TCP server receives the data sent from the device connected to T5L.

5.8.3 Process of TCP Transparent Transmission

1. Write the well-formatted data into the sending area (the value stored in address 0x402).
2. Set the sending flag (address 0x401, value 0x7AAC (TCP with formatted header)/0x7ABC (TCP-custom header format)). After that, the data in the sending area will be automatically read and sent via TCP.
3. When the receiving area (the address stored in address 0x402) receives the complete data, it will set the data length (address 0x737) and the data readable flag (address 0x736, value 0x5AA5) to the specified positions.
4. After receiving the data from TCP, write the data into the receiving area (the value stored in address 0x735), and set the data length (address 0x737) and the data readable flag (address 0x736, value 0x5AA5).
5. The upper-layer application determines whether new data has been received by scanning the data readable flag (address 0x736).
6. When the upper-layer application reads the data, it needs to clear the data readable flag.
7. The transparent transmission protocol format between the host computer and the WiFi module (the following is in HEX):

0x7AAC (TCP with formatted header)

AA55 + length of transparent transmission data (two bytes) + transparent transmission data;

Example: Transparent transmission data: DD EE FF

Communication format: AA 55 00 03 DD EE FF

0x7ABC (TCP-custom header format)

The transparent transmission data is in a use -defined format.

1).After the OS program sets the TCP IP, port, and return address, a buffer of 1024 bytes is required by default.

2).The communication module writes the data from the network to this address, with a maximum of 1024 bytes.

3).Write the length of the data put into the buffer this time into 0x737.

4).Write 0x5AA5 into 0x736.

5).Wait for the OS to take away the data. The OS continuously scans whether 0x736 is 0x5AA5. If so, take away the data of the length set in 0x737. Then set 0x736 to 0x0000, indicating that the data has been taken away.

6).The communication module also continuously scans 0x736. If it is 0x5AA5, it means the data has not been taken away, and it will wait.

7).The OS is responsible for parsing the data.

The underlying layer automatically sends: AA550000. This is the underlying heartbeat, and the host computer must return a response.

5.8.4 Set Static IP, Subnet Mask, Gateway, DNS

Definition	Address	Length	Description	Recommended Value
Configure WiFi, Ethernet Static IP, Subnet Mask, Gate IP, DNS	0x740	32	// In T5L, 4 bytes starting from address 0x740 represent a group in order. 0x740-0x747: WiFi configuration. 0x748-0x74F: Ethernet configuration. // 0x740-0x747: WiFi IP, Subnet Mask, Gate IP, DNS // 0x748-0x74F: Ethernet IP, Subnet Mask, Gate IP, DNS	

The format for setting Ethernet static IP, Subnet Mask, Gate IP, and DNS is as follows:

address	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
00000740	0000	0000	0000	0000	0000	0000	0000	0000	c0a8	0aee	ffff	ff00	c0a8	0a01	0808	0808

Operation for enabling static IP, Subnet Mask, Gate IP, and DNS settings

(1) Write the WiFi static IP, subnet mask, gate IP, and DNS address at addresses 0x740-0x747;

Write the Ethernet static IP, subnet mask, gate IP, and DNS at addresses 0x748-0x74F;

(2) At address 0x485:

Write 01 at the D0 position as the Ethernet fixed IP enable flag.

Write 01 at the D1 position as the WiFi fixed IP enable flag.

(3) At address 0x498:

At the D0 position, entering "b5" means writing the Wi-Fi IP address to the Wi-Fi module;

entering "b7" means writing the fixed Ethernet IP address to the Wi-Fi module.

(4) At address 0x49B:

Write b5 at the D0 position to indicate that the device reboots after activating the WiFi fixed IP and successfully connects to the WiFi static IP.

b6 indicates not activating the WiFi fixed IP.

b7 indicates that the device reboots after activating the Ethernet fixed IP and successfully connects to the Ethernet static IP.

b8 indicates not activating the Ethernet fixed IP.

5.9 DWIN Mini Program, Universal Remote Controller

Definition	Address	Length	Description	Recommended Value
Device Binding Verification Code Display	0x750	2	Display the verification code sent from the WeChat Mini Program. It consists of 4 characters or digits and is used for verification. The device only needs to check whether it is consistent with the code sent from the mobile phone.	
Device Binding Verification Confirmation	0x752	1	Set whether the verification is passed. 1 indicates that the verification is passed, and 0 indicates that the verification fails.	

6.0 Address 0x760: WebSocket Communication Interface

Definition	Address	Length	Description	Recommended Value
WebSocket Transparent Transmission Communication	0x760 0x761 0x762	16	D31-D30: Unused. User - defined. D29: Unused. D28: Connection status: 0x00 not connected; 0x01 connecting; 0x02 connected successfully; 0x03 connection disconnected. D27 - D0: Unused.	User-defined

6.1 Address 0x770: Electronic Table Card Application

Definition	Address	Length	Description	Recommended Value
Electronic Table Card	0x49C	4	D7: 5A indicates to start the electronic table card broadcast, once every 5 seconds. D6: Represents the current download progress. 0: Download starts. 1-99: Download in progress. 100: Download completed. 101: Download failed. D5D4: Represents the ID of the electronic table card. D3: 5A indicates to enable the weather forecast. D1D0: The current update number of the T5L file.	

1. Command Format

AA55 + Length (2 bytes) + Command Word (1 byte) + Content Body

2. Default Electronic Table Card Broadcast

The device broadcasts its information periodically. The format is:

`Mac Address,Local IP,Local Port,Table Card ID`

Tablet ID = 0: The tablet is not bound to any host computer.

Tablet ID > 0: The tablet is bound to a host computer, sending directed heartbeats instead of broadcasts.

1).Device Broadcast (Command Word: 0x01)

Sent every 5 seconds.

Content Body: `deviceNum,clientIP,clientPort,cardId`

Example: `123456789abc,192.168.0.1,5444,0`

2). Device Association to Host Computer (Command Word: 0x03)

A response to command 0x02, after joining the host computer.

Content Body: `deviceNum,clientIP,clientPort,cardId,serverIp,serverPort`

3). Heartbeat Transmission (Command Word: 0x04)

Sent periodically after joining the host computer (network entry completed).

Content Body: `deviceNum,clientIP,clientPort,cardId,serverIp,serverPort`

3. Host Computer Command (Command Word: 0x02)

Content Body: `Electronic table card ID`

4. Host Computer Response (Command Word: 0x05)

A response to the device's broadcast message (`Mac Address,Local IP,Local Port,Table Card ID`).

Content Body: `Assigned electronic table card ID`

If the device does not receive this response, it assumes the host computer is offline and resumes broadcasting.

5. Host Computer File Download Command (Command Word: 0xba)

Content Body:

```
[
  {
    "dw_file_number": 60,
    "dw_file_url": "http://1.117.163.53:8001/uploadfile/60_nocolor.icl"
  },
  {
    "dw_file_number": 60,
    "dw_file_url": "http://1.117.163.53:8001/uploadfile/60_nocolor.icl"
  }
]
```

Device Status Reports:

Start Download:

Mac Address (12 bytes) + aa55 + Length (2 bytes) + Command Word (0xbb) + 514-byte content + URL

Download Progress:

Mac Address (12 bytes) + aa55 + Length (2 bytes) + Command Word (0xbe) + Current Packet (2 bytes) + Total Packets (2 bytes) + 510-byte content + URL (identifies the file via URL)

Download Success:

Mac Address (12 bytes) + aa55 + Length (2 bytes) + Command Word (0xbd) + 514-byte content + URL

Download Failure:

Mac Address (12 bytes) + aa55 + Length (2 bytes) + Command Word (0xbc) + 514-byte content + URL

6.2 General Common MQTT Communication Interface

Similar to WebSocket communication: Place the specific content at a certain address, read this address, and then read the content.

The MQTT communication setting is 0x2A for general settings, and the command words are 0xC1, C2, C3, C4. Each instruction has its own command format. Since MQTT data is in text form, the relevant data uses <==> as the data segment connector.

The address layout is: Command word (2) + Address (2) + Word length (2)

Instruction Format and Processing Flow:

(1) Set the MQTT connection URL, username, and password.

a) 0x2AC1 + 0x2000 + 0x004 indicates the 0x2AC1 command, which reads 8-byte data at 0x2000.

b) At 0x2000, the data is placed in the format:

```
URL (including port) <==> Username <==> Password <==> Will topic <==> Will message
```

c) The length is converted to the word length.

d) The URL is in the form of domain name:port or IP:port, and the port must be specified.

e) Multiple parameters need to be separated by <==>. If there is no parameter, it is not necessary.

f) Example: `xxx.xxx.xx.xx:1883<==>name<==>pass<==>willtop<==>willmessage`

g) It must contain xxx.xxx.xx.xx:1883.

(2) Set the command word, starting address, and length for returning data.

a) 0x2AC2 + 0x3000 + 0x006 indicates setting the address where the data returned by MQTT subscription is placed. The length represents the maximum data length that can be accepted.

The structure of this address consists of the MQTT network connection status (2) + the length of the returned data (2) + the content body.

b) For the MQTT network connection status, 0 means the network connection is not started, 1 means the network is being connected, 2 means the connection is successful, and 3 means the connection fails.

If the connection is not successful, it will automatically reconnect every 2 seconds. 0 may also mean that the network connection has not been configured, and step (1) needs to be followed.

c) The returned data length is in bytes. > 0 means there is data to be read. After reading, it needs to be cleared to 0. The network will judge that if it is = 0, new data will be put in. If it is > 0, it means the data has not been taken away, and the data sent by the underlying network will be discarded until the T5L takes away the data. Note that when the T5L sets it to 0, it means the data has been taken away. Only when it is = 0 will the network layer put in data.

d) Content body: `the subscribed topic <==> content`, separated by <==>.

e) This area is occupied by the underlying network as a data exchange area and cannot be used for other purposes.

(3) Set the subscribed topics. Step (3) can be executed repeatedly to set multiple subscribed topics.

a) 0x2AC3 + 0x4000 + 0x006 indicates setting the address where the MQTT subscribed topics are placed. The length represents the data length of the subscribed topics.

b) After setting a subscribed topic, the underlying network starts the MQTT automatic connection.

c) It can be executed multiple times to subscribe to multiple topics.

(4) Send topic data. Step (4) can be executed repeatedly to send multiple topic data.

a) 0x2AC4 + 0x5000 + 0x006 indicates setting the address where the MQTT sending content body is placed. The length represents the data length of the sending content body.

b) Content body: `the sending topic <==> content`, separated by <==>.

c) It can be executed multiple times to send multiple topics.

Operation Processing Flow:

1) Set the connection IP or URL, port, username, password, will topic, and will content. This only needs to be set once.

2) Set the address for returning data and the MQTT connection status. This only needs to be set once.

3) Set the subscribed topics. Multiple subscribed topics can be set, with a maximum of 5.

4) Send topics and content, which can be sent arbitrarily.

7.0 Address 0xFF80: Bluetooth Special Settings

Definition	Address	Length	Description	Recommended Value
Bluetooth Name	0xff80	8	Name with a maximum of 16 bytes	
SN Code	0xff88	4	SN code must be 8 bytes, and the first 8 bytes of the transmitted data body must be the SN code. Bluetooth connection status: 1 = connected, 0 = not connected. Write Instruction: //0xaa55000f 820732 0102030405060708 c0a80ad1 Read Instruction: //0xaa55000583073c 0102030405060708 0003	

Instruction Format:

1. Instruction Header: 0xaa55 + Length (2 bytes) + Command Word + Address (2 bytes) + SN Code (8 bytes) + Transmitted Content (2-byte aligned).

2. Length (2 bytes): Calculated as Command Word + Address (2 bytes) + SN Code (8 bytes) + Transmitted Content (2-byte aligned).

3. Command Word 82: Write instruction.

4. Command Word 83: Read instruction.

5. Command Word 90: Instruction to read the SN code.

6. Add a special command for the remote control to obtain the SN code set by ESP32.Use command 83 to directly read the address 0x768, which reads 8 bytes

Configuration Instructions

The MAC address for BLE Bluetooth connection is stored in T5L at addresses: 0x0739-0x073B.

The PIN code for BLE Bluetooth connection is stored in T5L at addresses: 0x073C-0x073D.

The flag for setting the BLE Bluetooth connection PIN code is stored in T5L at address 0x073E. Setting it to 0x5a makes it valid, other values are ignored.

For the first use, the maximum MTU value must be set. Otherwise, the default value is 23, which may not meet communication requirements.

To activate and update the BLE PIN code:

● After setting the BLE PIN code, it will not take effect immediately. You need to set the value of address 0x0402 to 0x073C, the value of address 0x0403 to 0x0003, and the value of address 0x0401 to 0x2aa3. This triggers the set BLE PIN code to take effect and saves it to the flash. For subsequent use, the saved PIN code will be used directly.

● When connecting via Bluetooth, enter the corresponding PIN code. The connection succeeds after verification passes.

● After successful connection, read and write operations are performed on the service UUID 0x00FF and characteristic value UUID 0xff01.

Valid data read by BLE must start with 0xaa55.

The BLE write data format is: `0xaa550007820732c0a80ad1`.

Read instruction sent via the Bluetooth channel: `0xaa55000583073c0003`.

Return value: `0xaa55000983073c000a2c2a5a5a`.

Write instruction sent via the Bluetooth channel: `0xaa550007820732c0a80ad1`.

Return value: `0xaa55000982777269746520 6f6b`.

Data reported by the screen: `0xaa550007850732c0a80ad1`.

Invalid data read by BLE is `0x656d7074792064617461`. Reply with invalid data to prevent read freezing.

Revision Records

Version	Revise Date	Content	Editor
V1.0	2019-12-15	Implemented basic WIFI module protocol functions	Wang Jintai
V1.1	2019-12-30	Added hotspot scanning function	Wang Jintai
V1.2.1	2020-04-15	1. Added UDP broadcast protocol 2. Added UDP broadcast protocol usage instructions (Refer to Section 5.6)	Li Jianmin
V1.2.2	2020-04-20	1. Added IP write-back function 2. Added port configuration	Wang Jintai
V1.2.3	2020-06-04	Modified 487 address description	Li Jianmin
V1.2.4	2021-01-13	Added 49C electronic table broadcast startup and 4A0 [D7 D6] WIFI router connection error description	Li Jianmin
V1.2.5	2021-03-30	Added UDP weather forecast information retrieval	Li Jianmin
V1.2.6	2021-05-27	Added TCP transparent transmission communication function	Li Jianmin
V1.2.7	2022-04-20	1. Supported static IP, gateway, subnet mask, DNS 2. Enabled dynamic/static IP switching	Li Jianmin
V1.2.8	2022-05-31	Added TCP-Modbus transparent transmission; removed heartbeat; parsed Modbus format length	Li Qiang
V1.2.9	2022-08-16	Added WIFI20, ethernet module WIFI-30	Li Qiang
V1.3.0	2022-09-26	Added DWIN Cloud mini-program remote device binding mechanism	Li Qiang
V1.3.1	2023-07-01	Added generic MQTT; default disabled Bluetooth/TCP/WebSocket/MQTT connections to DWIN Cloud	Li Qiang
V1.3.2	2023-07-25	Standardized electronic table; added UDP heartbeat, HTTP download update	Li Qiang
V1.3.3	2023-08-25	Added generic MQTT direct connection to customer servers	Li Qiang
V1.3.4	2023-08-25	Added T5L CRC16 checksum support	Li Qiang

Please contact us if you have any questions about the use of this document or our products,or if you would like to know the latest information about our products:

- Customer service Tel: +86 400 018 9008
- Customer service email: dwinhmi@dwin.com.cn
- DWIN Developer Forum: https://forums.dwin-global.com/

Thank you all for continuous support of DWIN,and your approval is the driving force of our progress!