

Open Source IoT Gateway

USR-EG118

User Guide



V2.0

Be Honest & Do Best

Your Trustworthy Smart Industrial IoT Partner

Content

1. Introduction	3 -
1.1. Overview	3 -
1.2. Parameters table	3 -
1.3. Pin Mapping	4 -
1.4. Indicators	5 -
1.5. Dimension	5 -
1.6. Development Environment	5 -
2. Development Environment Setup	5 -
2.1. Main steps to set up a development Environment	5 -
2.2. Specific Step Instructions	6 -
2.2.1. Download and install Arduino IDE	6 -
2.2.2. Add Boards Manager URLs	6 -
2.2.3. Install ESP32 Boards	
2.2.4. Install the ArduinoModbus library file,	7 -
2.2.5. Add AWS_IOT library file	8 -
3. Program burning	- 8 -
3.1. Hardware connection	8 -
3.2. Programming via Arduino	9 -
3.2.1. Downloading parameters settings	9 -
3.2.2. Download Program	10 -
3.3. Programming via ESP32 download tool	10 -
3.3.1. Export compiled binary file	11 -
3.3.2. Load BIN file	11 -
4. Functional Test	错误!未定义书签。
5. Contact Us	20 -
6. Disclaimer	20 -



1. Introduction

1.1. Overview

USR-EG118 is an open source IoT gateway powered by Espressif's ESP32 module equipped with low power Xtensa® 32-bit LX6 MCU 240 MHz processor with 4MB pSRAM and up to 8MB SPI flash memory on-board. It integrates rich hardware interface: RS485/232, DO*2, DI*1, AI*1, downloading port, Wi-Fi, Ethernet port and BLE. PUSR provides basic source code demo for users to develop their own application. For the rich hardware interface and networking methods, this device can be applied in a variety of scenarios, especially scenarios where users need to collect or control serial devices and IO devices simultaneously, such as: Wastewater Treatment, Agricultural Irrigation, Solar Energy, Intelligent Farming etc.

1.2. Parameters table

	USR-EG118 Parameters
Chipset	ESP32-WROVER-E
Professor	Dual-core Xtensa® 32-bit LX6 MCU, 1.2GHz
RAM	8M
Flash	4M
Power Supply	9~36V
Hardware Interface	2
05222/405	1*RS232/485
R5232/485	Default parameter: 115200, N, 8, 1
DI	1*digital input, (0 - 2 V detected as logic low, 9 - 36 V detected as logic high)
	2*digital output
DO	DC contacts rating @R(at resistive load)10A / 28V DC, NC
	AC contacts rating @R(at resistive load)10A / 277V AC,NC
	AC contacts rating @R(at resistive load)5A /250V AC, NO
AI	1*analog input(4~20mA)
Wi-Fi	
Standard	IEEE802.11b/g/n
Speed	Up to 150Mbps
Frequency	2412 ~ 2484 MHz
Others	
BLE	Compatible with BEL4.2

Table 1. Parameters of USR-EG118



1.3. Pin Mapping

The table shows the corresponding relationship between the hardware interface pins of M100 and the pins of the main control ESP32.

Users can also download the schematic diagram corresponding to the M100. Download link:

https://www.pusr.com/support/download/Schematic-Diagram-Hardware-schematic-diagram-Of-USR-

EG118.html

Pins of EG118		Pins of ESP32		Function		
No.	Interface	Pin	Pin Name	No.	Description	
1		RMII_MDIO	IO18	30		
2		RMII_MDC	1023	37		
3		RMII_REF_CLK	100	25		
4		RMII_TXD0	IO19	31		
5	RMII Interface	RMII_TXD1	1022	36	For Ethernot port	
6		RMII_RX0	1025	10	For Ethernet port	
7		RMII_RX1	IO26	11		
8		RMII_TXEN	1021	33		
9		RMII_CRS_DV	1027	12		
10	PHY Reset	PHY_RESET	105	29		
11		UART1_TX	1032	8		
12	RS485/232	UART1_RX	1033	9	For serial port	
13		RS485_EN1	1012	14		
14		UART2_TX	IO13	16		
15	Reserved RS485	UART2_RX	1034	6	Reserved, for IO expansion machine	
16		RS485_EN2	IO14	13	machine	
17		OE	104	26		
10	Latch control	IE	IO15	23	Register input/control pin	
10			102	24		
19	DO	DO1	DO1	Q1	Digital output(Not connect to	
20	00	DO2	DO2	Q2	ESP32 directly)	
21	DI	DI1	SENSOR_VN	5	Digital input	
22	AI	ADC1	IO35	7	Analog input	
23	Watchdog	MCU_WDI	RXD0	34	Pin for feeding watchdog	
24	Reload	Reload	SENSOR_VP	4	 Reset to factory settings Enable program flashing mode 	
25	LED	LED_WORK	U0_RX	34	Work indicator	
26	Reset	RESET	EN	3	Reset pin for watchdog	
27	Elashing	SWCLK	U0_TX	35	Drogram Flaching	
28	FIDSIIIIIg	SWDIO	U0_RX	34		



1.4. Indicators

USR-EG118 provides 2 indicators in total, the specific description is as follows.

Name	Description
PWR	Steady on: power supply is normal. Off: No power supply or abnormal power supply.
WORK	Blinking: The system works normally.

Table 2. LED indicator

1.5. Dimension

- > DIN-Rail mounting and wall mounting supported.
- > 79.6*58*110.0mm (L*W*H, accessories not included)



Figure 1. Dimension of USR-EG118

1.6. Development Environment

Users can use the Arduino IDE provided by Arduino for development, or they can use the ESP-IDF tool provided by ESP32 for development. This document is introduced based on the Arduino IDE development environment.

2. Development Environment Setup

2.1. Main steps to set up a development Environment

For the steps to build a secondary development environment, refer to the picture below. The specific



operation steps will be introduced in detail in the following chapters.



2.2. Specific Step Instructions

2.2.1. Download and install Arduino IDE

Download and install Arduino IDE: https://www.arduino.cc/en/Main/Software

2.2.2. Add Boards Manager URLs

After installing the IDE, add additional boards manager URLs:

https://dl.espressif.com/dl/package_esp32_index.json

File->Preference-> Add Additional Boards Manager URLs

© 0911m100okv1 Arduino 1.8.5 File Edit Sketch Tools Help	- 1	٥	×
			ø
			-
			^
//////////////////////////////////////			
Settings Hetrork			
#define PHYTESTSAMPLE_BNARLE Sketchbook location:			
Rdeline #EDServerSaMPL_ENABLE Broze Broze			
#define FEEDDOGENAELE Editor language: English (English)			
Hefine AVS_CLOUD_ENABLE Editor font size: 12			
Interface scale: 🖂 Automatic 100 🗘 (requires restart of Arduino)			
Show verbose output during: ☑ compilation ☐ upload			
Cospiler warnings: Hone v			
sinclude (%17).h> Display line numbers			
finclude (TebServer.h) Enable Code Folding			
Finclude (KSPaDNS.h> Vrify code after upload			
oner chart ceid = "conchuischila".			
const char said = Sugmander,			
\Box (Thek for updates on startup			
sidef AVS_CLODE_DRAELE			
Additional Reards Wanazze WEIse (https://dl.estysessif.cog/dl/nackazz.esty2 index ison			
Adefine CUST_SSID 'songhuiabile' Nor preferences can be ditectly in the file			
#define CUST_ROUTER_PASS 19918134855" C:\USers\Administrator\AppData\Local\Arduino15\preferences.txt			
addine USI_DAORAE awisud.augi.ais.iot.crrneruf define USI_CLOUD_CLIENT_D 1:234' (dit only when Arduino is not running)			
#define CUST_MQTTPUBLISHTOPIC "arduino_m100"			
char HOST_ADDRESS[-COST_EBOKER:			
char (LIERI_IDI)= CUSI_CLOUD_CLIERI_ID;		_	~

2.2.3. Install ESP32 Boards

Tools->Board Manager->Search "ESP"



ESP32-WROOM-DA Module, Default 4MB with splifs (1 2MB APP/1 5MB SPIFFS), 240MHz (WIF/BT), QIO, 80MHz, 4MB (32Mb), 921600, Core 1, Core 1, None, Disabled on COM15

0911m100okv1 Arduino 1.8.5		- 0 ×
		Q
0911m100okv1		
// ///////////////////////////////////	Boards Manager X Type All SESP Arduino AVR Boards by Arduino version 1.8.6 INSTALLED Boards Indued in this package: Arduino Yún, Arduino UNO, Arduino UNO WFI, Arduino Diecimila, Arduino Nano, Arduino Mega, Arduino Mega ADK, Arduino Leonardo, Arduino Leonardo, Arduino Saloze: Arduino Yún, Arduino LNO, Arduino DNO, WFI, Arduino Diecimila, Arduino Ethernet, Arduino Ming, Arduino Bardino, Arduino Bir, Arduino BT, Arduino DI, Arduino LUNPad, Arduino Lineardo Ethernet, Arduino Nano, Arduino Robot Motor, Arduino Ethernet, Arduino Nini, Arduino Antone, Arduino Ethernet, Arduino Ming, Arduino Nano, Arduino Ming, Arduino Diecimila, Arduino Lonardo Ethernet, Arduino Ming, Arduino Robot Control, Arduino Robot Motor, Arduino Gemma, Adafruit Circuit Playground, Arduino Yún, Arduino Industrial 101, Linino One. Contine helo More Info	~
<pre>#include (TiFi.h> #include (TiFicHent.h> #include (TeSP=DUS.h> #include (ESP=DUS.h> const char* ssid = "songhuisobile"; const char* password = "13918134855"; 4 #iddd_ 45 C(OUD_EUEEE </pre>	Arduino ESP32. Boards by Arduino Boards included in this package: Arduino Nano ESP32. More info Boards included in this package: ESP32 by Espressif Systems version 2.0.10 INSTALLED Boards included in this package: ESP32 by Espressif Systems version 2.0.10 INSTALLED Boards included in this package: ESP32 by Espressif Systems version 2.0.10 INSTALLED Boards included in this package: ESP32 by Espressif Systems version 2.0.3 Dev Board, ESP32-C3 Dev Board,	
<pre>Filed avg_co_nnuc Einclud (Avg_CO_nnuc Edefine CUST_SSID 'songhuiaobile' #define CUST_ROUTER_PASS '13918134855' Edefine CUST_ROUTER_PASS '13918134855' Edefine CUST_ROUTER_TO '1234' Edefine CUST_CLOUD_CLIENT D' '1234' Edefine CUST_MOTTPOELTSNOPIC 'archuino_1100'</pre>	Close	
<pre>char nusi_subdess[]=CUST_ENUMER; char CLIENT_ID[]= CUST_CLOUD_CLIENT_ID;</pre>		Ŷ
	ESP32-WROOM-DA Modulie, Default 4MB with splifs (1 2MB APP/1.5MB SPIFFS), 240MHz (WFVBT), 010, 80MHz, 4MB (32Mb), 921600, Core 1, Core 1, Core	re 1, None, Disabled on COM15

2.2.4. Install the Needed library file

If users test the demo code provided by the PUSR, the following Library files need to be installed first:

- ESPAsyncWebServer
- ArduinoRS485
- ArduinoModbus
- PubSubclient
- OneButton

•Sketch->Library Manager->Search " ESPAsyncWebServer" ->Install

Users can also download the other libraries needed on Arduino IDE.





2.2.5. Add AWS_IOT library file

PUSR offers library source code of AWS, users can download from:

https://www.pusr.com/support/download/Library-File-AWS-IOT-File-Provided-by-PUSR.html

The downloaded libraries should be copied to the installation path of Arduino IDE.

Note: If the AWS code is needed, users are strongly encouraged to use the AWS libraries we provide.

3. Program burning

There are two main ways of burning firmware: through Arduino IDE and the tool provided by ESP32. The specific usage methods of the two burning tools are introduced below.

No matter which method is used to burn, the download cable needs to be connected in advance.

3.1. Hardware connection

Before you start, some preparations is needed.

- USR-EG118*1
- USB to TTL converter *1
- Wi-Fi antenna*1
- USB to RS485/232 Cable*1
- 12V/1A Power Adapter*1

The hardware connection is like this:





3.2. Programming via Arduino

3.2.1. Downloading parameters settings

Tools-->Board-->" ESP32-WROOM-DA Module"

Auto Format Ctrl+T Auto Format Ctrl+Shift+H DB11m100M Fix Encoding & Reload Serial Monitor Ctrl+Shift+H WiFi101 Firmware Updater Ctrl+Shift+L Board: "ESP32 WROOM-DA Module" Arduino Vin Mini Hash Mode: "QIO" Arduino Vin Mini Fash Size: "4MB (32Mb)" Arduino Industrial 101 Edefine MEBServ Flash Size: "4MB (32Mb)" Edefine VEBServ Flash Size: "4MB (32Mb)" UP or trauency: "30MHz WiFi (51-2MBSE (51-2MBSE)") Elnelude CHFIC Frase All Flash Freezeency: "30MHz Finclude CHFIC For Debus Level: "None"	File Edit Sketch To	ools Help		
Archive Sketch Ball Monitor Serial Monitor Serial Monitor Serial Monitor Ctrl+Shift+L Wifi01 Firmware Updater Baard "ESP32-WROOM-DA Module" Hash Modito" 'QiO' Fadarin & Child State File State Finclude Ciffic		Auto Format	Ctrl+T	
0911m100xkr Fix Encoding & Reload Serial Monitor Serial Monitor Ctrl+Shift+M Ctrl+Shift+M WiFi01 Firmware Updater Arduino Vin Mini #define PETEST Flash Moder "Q(0" Flash Moder "Q(0" Arduino Vin Mini define PETEST Flash Moder "Q(0" redstine VESP32 WROOM-DA Module" Arduino Undustrial 101 Linino One Arduino Vin Mini define VESP32 Flash Size "MM8 (23Mb)" define VESP32 Pentition Schemer "Default 4MB with spiffs (1.2MB APP/1.5MB SPIFFs)" define VESP32 Pentition Schemer "Default 4MB with spiffs (1.2MB APP/1.5MB SPIFFs)" define VESP32 Pentition Schemer "Default 4MB with spiffs (1.2MB APP/1.5MB SPIFFs)" define VESP32 Pentition Schemer "Default 4MB with spiffs (1.2MB APP/1.5MB SPIFFs)" define VESP32 Pentition Schemer "Default 4MB with spiffs (1.2MB APP/1.5MB SPIFFs)" define VESP32 Pentition Schemer "Default 4MB with spiffs (1.2MB APP/1.5MB SPIFFs)" einclude (TiFi) Flash Frequency: "BOMH2" Upload Speed: "921600" ESP322 Dev Module ESP322 Dev Module ESP322 Dev Module einclude (TiFi) Port einclude (TiFi) Porgammer: AVRISP Mkll" Core		Archive Sketch		
Serial Monitor Ctrl - Shift - M. Serial Plotter Ctrl - Shift - M. WHF101 Firmware Updater Arduino Yún Mini #define FRITEST Hash Size: "4MB (32Mb)" #define FRITEST Hash Size: "4MB (32Mb)" #define FRITEST Flash Size: "4MB (32Mb)" #define fRITEST Events Run On: "Core 1" #define fRITEST Events Run On: "Core 1" #define fRITEST Erase All Flash Before Sketch Upload: "Disabled" Arduino Runs On: "Core 1" ESP322 Dew Module ESP322 Dew Module ESP322 Dew Module CPU Frequency: "240MHz (WF/V)") ESP322 Dew Module Upload Speed: "221600" ESP322 Dew Module Esp322 Dew Module ESP322 Dew Module Esp32 S3-Dew Module ESP32 S3-Dew Module Esp32 S3-Dew Module ESP32 S3-Dew Module Esp32 S3-Dew Module ESP32	0911m100okv1	Fix Encoding & Reload		
Serial Plotter Ctrl+Shift+L Wiff101 Firmware Updater Arduino Yún Mini Board: "ESP32-WROOM-DA Module" Arduino Yún Mini Fdefine FETER Hash Mode: "Qto" Tadsino Kone: "Qto" Arduino Industrial 101 Edefine FETER Flash Size: "4M8 (32Mb)" Arduino No. "Core 1" Edefine FETEROC Partition Scheme: "Default 4MB with spiffs (1.2MB APP/1.5MB SPIFFS)" Arduino MWFi Edefine FETEROC Partition Scheme: "Default 4MB with spiffs (1.2MB APP/1.5MB SPIFFS)" ESP3252 Arduino Edefine FETEROC Partition Scheme: "Default 4MB with spiffs (1.2MB APP/1.5MB SPIFFS)" ESP3252 Arduino Edefine FETEROC Partition Scheme: "Default 4MB with spiffs (1.2MB APP/1.5MB SPIFFS)" ESP3252 Arduino Edefine FETEROC Partition Scheme: "Default 4MB with spiffs (1.2MB APP/1.5MB SPIFFS)" ESP3252 Arduino Edefine ATS_CLOU CPU Frequency: "240MHz (WiF/WT)" ESP3252 Dev Module Einclude CHFIC Core Debug Level: None" ESP3252 Wood Module Einclude CHFIC Port ESP3252 PCO-DA Finclude CHFIC Port ESP3252 Wroov ARL Finclude CHFIC Port ESP3252	•	Serial Monitor	Ctrl+Shift+M	
WiFi01 Firmware Updater Board: "ESP32 WROOM-DA Module" #define PETERST Hash Mode: "Q(0" Flash Flast Edore: Sketch Upload: "Disabled" Arduino Runs On: "Core 1" Flash Frequency: "BOMHz" CPU Frequency: "BOMHz" Upload Speed: "921600" Upload Speed: "921600" Upload Speed: "921600" Upload Speed: "921600" ESP322 Dev. Module ESP322 Dev. Module ESP32 Dev. Module ESP32 NCO-DA Module Esp32 NCO-DA Module Esp32 NCO-DA		Serial Plotter	Ctrl+Shift+L	
Board: "ESP32:WROCM-DA Module" A Fieldine VEIServ Flash Mode: "QIO" Arduino Nún Mini Arduino Didustrial 101 Linko One Arduino Industrial 101 Linko One Flash Mode: "QIO" Arduino Industrial 101 Linko One Flash Mode: "QIO" Arduino Industrial 101 Linko One Flash Frequency: Team Arduino Mini Arduino Ersex All Flash Before Sketch Upload: "Disabled" Arduino Une WFi Esp32 Arduino Esp322 Dev Module Esp32 Dev Module Esp322 Dev Module Esp32 Sause Esp32 Sause Finclude (FIFI Port Esp32 Sause Esp32 Sause Esp32 Sause Esp32 Sause Esp32 Sause Esp32 Sause Esp32 Saus		WiFi101 Firmware Updater		
addition PETTEST Flash Moder "Q(Q" addition Description Events Run On: "Core 1" addition EEEEDOC Partition Scheme: "Default 4MB with spiffs (1.2MB APP/1.5MB SPIFFs)" ESP32 Arduino addition Description Erase All Flash Before Stecht Upload: "Disabled" ESP322 Over Module addition Current ESP32 Description ESP322 Description atloadition Current ESP32 Description ESP32 Description atloadition Carrent ESP32 Description ESP32 Description atloadition Carrent ESP32 Description ESP32 Description atloadition Carrent Port ESP32 Description ESP32 Description atloadition Carrent Port ESP32 Description ESP32 Description atloadition Carrent Port ESP32 Description ESP32 Description atloaditio		Board: "ESP32-WROOM-DA Module"		
statistic Filash Size: "4MB (32Mb)" Addition Unitability (32Mb)" statistic Addition One Addition One statistic Exerts Run On: "Core 1" Linino One statistic Partition Scheme: "Default 4MB with spliffs (1.2MB APP/1.5MB SPIFFs)" Exerts Run On: "Core 1" statistic Partition Scheme: "Default 4MB with spliffs (1.2MB APP/1.5MB SPIFFs)" EsP3232 Arcluino statistic Arduino Uno: "Core 1" EsP3232 Der Module Flash Frequency: "40MHz" ESP3252 Der Module CPU Frequency: "40MHz" ESP3252 Der Module Upload Speed: "921600" ESP3252 Der Module stinclude <fiffic< td=""> Core Debug Level: "None" sinclude <fiffic< td=""> Fordament: AVRISP mkli" Poorgammer: "AVRISP mkli" ESP3252 Der Module sinclude <fiffic< td=""> ESP3252 Der Module sinclude <fiffic< td=""> Forgammer: "AVRISP mkli" Poorgammer: "AVRISP mkli" ESP3253 CAM LCD const char's ssit Burn Bootolader const char's pastword = "19918134855"; ESP3252 Wroover Kit (all versions) sidefine CUST_ROUBE_RASE, "senghui abbils"," UM TinyS2 sidefine CUST_ROUBE_RASE, "senghui abbils"," UM TinyS3 sidefine CUST_ROUBER_RASE, "senghui abbils"," UM TinyS3 sidefine CUST_ROUBER_RASE, "senghui abbils"," UM TinyS3</fiffic<></fiffic<></fiffic<></fiffic<>	#define PHYTEST	Flash Mode: "QIO"	:	Arduino Industrial 101
elstine NOBUST Events Run On: "Core 1" Advaino Uno WiFi edstine FEDDO Partition Scheme: "Default AMB with splifs (1.2MB APP/1.5MB SPIFFS)" Advaino Uno WiFi #dstine VEDDO Partition Scheme: "Default AMB with splifs (1.2MB APP/1.5MB SPIFFS)" ESP322 Archimo #dstine VEDDO Partition Scheme: "Default AMB with splifs (1.2MB APP/1.5MB SPIFFS)" ESP32 Archimo #dstine VEDDO Partition Scheme: "Default AMB with splifs (1.2MB APP/1.5MB SPIFFS)" ESP323 Dev Module #lash frequency: "BMNH2" ESP3225 Dev Module ESP3225 Dev Module EVEDO CPU Frequency: "BMNH2" ESP322 Dev Module Upload Speed: "921600" ESP322 Wrover Module ESP322 Wrover Module #include (TiFI) Core Debug Level: "None" ESP322 Wrover Module #include (TiFI) Port ESP322 Sa Dev Module #include (TiFI) Port ESP322 Wrover Module #include (TiFI) Port ESP322 Sa Dev Module #include (TiFI) Burn Bootloader ESP322 Sa Dev Module	#define WEBserv	Flash Size: "4MB (32Mb)"	:	Linino One
Elefine FEEDOC Partition Scheme: "Default 4MB with splifs (1.2MB APP/1.5MB SPIFFS)" ISP32 Arduino Elefine AVS_CLO Erase AHT Hash Before Steketh Upload: "Disabled" ESP322 Arduino Esp32 Arduino ESP322 Dev Module ESP32 Arduino ESP322 Dev Module Esp32 Dev Module ESP322 Dev Module Esp32 Dev Module ESP32 Dev Module Esp32 Dev Module	#define MODBUST	Events Run On: "Core 1"	:	Arduino Uno WiEi
#define AYS_CLO Ersex All Flash Before Sketch Upload: "Disabled" ESP3253 Dev Module #define AYS_CLO Esp3253 Dev Module ESP3253 Dev Module ESP3252 Dev Module ESP3252 Dev Module Esp325 Dev Module ESP3252 Dev Module Esp325 Dev Module ESP3252 Dev Module Esp325 Dev	#define FEEDDOG	Partition Scheme: "Default 4MB with spiffs (1.2MB APP/1.5MB SPIFFS)	1	ESP32 Arduino
Arduino Runs On: "Core 1" ESP322G Dev Module Hish Frequency: "240MHz" ESP325 Dev Module CPU Frequency: "240MHz" ESP325 Dev Module Upload Speed: "921600" ESP325 Dev Module Esp325 Dev Module	#define AWS_CLO	Erase All Flash Before Sketch Upload: "Disabled"	:	ESP32S3 Dev Module
Flash frequency: 240MHz (WF/BT)* ESP322S Dev Module Upload Speed: *221600* ESP32 Wroot MLz (WF/BT)* Upload Speed: *221600* ESP32 Wroot MLZ (WF/BT)* Esp32: WROOM-DA Module ESP32 Wroot MLZ (WF/BT)* Port ESP32 Wroot MLZ (WF/BT)* Finclude (*IFIC Port Esp12: WROOM-DA Module ESP32 Wroot MLZ (WF/BT)* Port ESP32 Wroot MLZ (WF/BT)* Esp22: WROOM-DA Module ESP32 Wroot MLZ (WF/BT)* Port ESP32 Wroot MLZ (WF/BT)* Finclude (*IFIC Port Port ESP32: SUSB-DOF Esp32: SUSB-DOF ESP32: SUSB-DOF Esp1: SUSB-SUSD: SUSB-SUSD: SUSB-SUSD: SUSB-SUSD: SUSB-SUSD: SUSB-DOF ESP32: SUSB-DOF Esp1: SUSB-SUSB-SUSB-SUSB-SUSD: SUSB-SUSD: SUSB-SUSD: SUSB-SUSB-SUSD: SUSB-SUSB-SUSB-S		Arduino Runs On: "Core 1"	:	ESP32C3 Dev Module
CPU Prequency: 240MHz (WiHyll)" ES932 Dev Module #include (TiFI: Core Debug Level: "None" ES932 WROOM-DA Module #include (TiFI: Core Debug Level: "None" ES932 WROOM-DA Module #include (TiFI: Core Debug Level: "None" ES932 Wrover Module #include (TiFI: Get Board Info ES932 Salow #include (TiFI: Get Board Info ES932 Salow #include (TiFI: Get Board Info ES932 Salow #orgammer: 'AVRISP mkli" ES932S Native US8 #include (TiFI: ES932S Native US8 #include (ATS_IOT. h) UM TinyPICO #include (STSI: SID "Songhitabile" UM TinyPICO #define (UST_SROKER "swisp@ufmach1e", 193134855" UM RMP #define (UST_SROKER "swisp@ufmach1e", 193134855" UM RMP #define (UST_SROKER "swisp@ufmach1e", 1934 UM TinyS3 #define (UST_UST_ROKER, 100)* UM TinyS3 #define (UST_UST_ROKER, 100)* UM TinyS3 #define (UST_UST_ROKER, 100)* UM TinyS3		Flash Frequency: "80MHz"		ESP32S2 Dev Module
Include (TiFI) Upload Speed: '921600'' ESP32-WROOM-DA Module Einclude (TiFI) Core Debug Level: 'None'' ESP32-WROOM-DA Module Einclude (TiFI) Port ESP32-WROOM-DA Module Einclude (TiFI) Fort ESP32-WROOM-DA Module Esp32-S3-USB-OR ESP32-S3-USB-OTG ESP32-S3-USB-OTG Const char* ssi Burn Bootloader ESP32-S3-USB-OTG const char* passroof = '13918134855'; ESP32 Wrover Kit (all versions) #ifdef a MS_CLOUD_ENABLE UM TinyS2 Enclude (ATS_10T: ho UM TinyS2 #define CUST_SID "songhusobile" UM TinyS2 #define CUST_BONER "arvis0u'Tungf, ats.iot.ct-morth-1.aszon#s.cos.cn" UM TinyS3 #define CUST_BONER "arvis0u'Tungf, ats.iot.ct-morth-1.aszon#s.cos.cn" UM TinyS3 #define CUST_BONER "arvis0u'Tungf, ats.iot.ch-morth-1.aszon#s.cos.cn" UM TinyS3 #define CUST_UNTEL_ISTENC' arvis0u'Tungf, ats.iot.ch-morth-1.aszon#s.cos.cn" UM TinyS3 #define CUST_CUST_BONER "arvis0u'Tungf, ats.iot.ch-morth-1.aszon#s.cos.cn" UM TinyS3 #define CUST_CUST_BONER "arvis0u'Tungf, ats.iot.		CPU Frequency: "240MHz (WiFi/BT)"		ESP32 Dev Module
Linkuda (Tirki) Core Uebug Level: 'None" ESP32 Worver Module Linkuda (Tirki) Port ESP32 (PCO-D4 Eincluda (Tirki) Get Board Info ESP32 (PCO-D4 Eincluda (Tirki) Porgrammer: 'AVRISP mkl" ESP32 (PCO-D4 Const char's password = '13918134855'; ESP32 (PCO-D4 Eincluda (ATS_ION_ERLASLE) UM Tirki (Leverions) #ifdef aTS_CLOUD_ERLASLE UM Tirki (Leverions) #includa (ATS_ION_ERLASLE) UM Tirki (Leverions) #idefine (DST_SSTD 'songhuisobile' UM Tirki (Leverions) #idefine (DST_ROKER, 'ass' i3918134855') UM Tirki (Leverions) #idefine (DST_ROKER, 'ass' i3918134855') UM Tirki (Leverions) #idefine (UST_ROKER, 'ass' i3918134855') UM Tirki (Leverions) #idefine (UST_ROKER, 'ass' i001'songf, ats.iot.cm-north-1.asazonass.cos.cn'' UM Tirki (Leverions) #idefine (UST_ROKER, 'ass' i001'songf, ats.iot.cm-north-1.asazonass.cos.cn'' UM Tirki (Leverions) #idefine (UST_ROKER, 'ass' i000'songf, ats.iot.cm-north-1.asazonass.cos.cn'' UM Tirki (Leverions) #idefine (UST_RO	zinclude (ViRi	Upload Speed: "921600"		ESP32-WROOM-DA Module
Hinchold CHAPS ESP32 P(C-O-D4 Einclud CHAPS Get Board Info Finclud CHAPS Get Board Info Programmer: 'AVRISP mkll' ESP32-S3-Box Const char* ssi Bum Bootloader Const char* ssi Bum Bootloader State State ESP32-S3-LOSE Esp32-S3-LOSE ESP32-S3-LOSE Esp32-S3-LOSE ESP32-S3-LOSE Esp32-S3-LOSE ESP32-S3-LOSE Esp32-S3-LOSE ESP32-S3-LOSE Esp32-S3-LOSE UM TimyPiCO Esp32-S3-LOSE UM TimyPiCO Edefine CUST_SSID 'songhusobile'' UM TimyS2 Edefine CUST_BOKER 'ass'159134855' UM RAP Edefine CUST_BOKER 'ass'1591455' UM RAP Edefine CUST_BOKER 'ass'15914555' UM RAP Edefine CUST_BOKER 'ass'15914555' UM RAP Edefine CUST_BOKER 'ass'15914555' UM RAP Edefine CUST_BOKER 'ass'15916* UM RAP Edefine CUST_BOKER 'assisted in '1234' UM FeatherS3 Edefine CUST_ENDER_BOKER 'assisted in '1234' UM FeatherS3 Edefine CUST_ENDERER S.ODI Ultra v1 Char HOST_ADDRESSIC=CUST_BOKER; LUKjGo T-Display-S3	#include (WiFiC	Core Debug Level: "None"		ESP32 Wrover Module
Espa://site Get Board Into Espa:/site Espa:/site Findude (SFFM Programmer: 'AVRISP MkI' Espa:/site	Finclude (TehS	Port		ESP32 PICO-D4
Programmer: "AVRISP mkll" ESP32-S3-UBS-OTG Const char* ssi Burn Bootloader ESP32-S3-UBS-OTG Const char* ssi Burn Bootloader ESP32-S3-UBS-OTG Const char* ssi Burn Bootloader ESP32-S3-UBS-OTG Const char* ssi Tis918134805"; ESP32-S2-UBS-OTG #ifdef ATS_CLOUD_ENABLE UM TinyPiCO UM TinyPiCO #include <ats_iot.h> UM TinyS2 UM FeatherS2 #define CUST_SDID_Songhuiabile" UM TinyS3 UM TinyS3 #define CUST_DROKER_YASS '13918134805" UM TinyS3 UM TinyS3 #define CUST_DROKER_VASS'13918134805" UM TinyS3 UM RAP #define CUST_DROKER_VICTURE_TASS '13918134805" UM TinyS3 UM RAP #define CUST_DROKER_VICTURE_TASS '13918134805" UM WAROS3 UM FeatherS3 #define CUST_DROKER_VICTURE_TASSICON_CONTRAGE ASSIC CONTRAGE ASSIC CONTRAGE</ats_iot.h>	#include <espal< td=""><td>Get Board Info</td><td></td><td>ESP32-S3-Box</td></espal<>	Get Board Info		ESP32-S3-Box
const char* ssi ESP32S3 CAM LCD const char* ssi ESP32S3 CAM LCD const char* sessord = '15918134855'; ESP32Wrower Kit (all versions) sifdef ATS_CLOUD_ENBALE UM TimyPICO sinclude (ATS_IOT.h) UM Teather52 define CUST_SSID 'songhuisobile" UM Keather52 idefine CUST_SSID 'songhuisobile" UM TimyS2 define CUST_BOKER 'swis00/magf.ats.iot.cm-north-1.sazonaws.cos.cm" UM TimyS3 define CUST_BOKER 'swis00/magf.ats.iot.cm-north-1.sazonaws.cos.cm" UM Feather53 idefine CUST_UNUTER_MERS '13918134855' UM Feather53 idefine CUST_BOKER 'swis00/magf.ats.iot.cm-north-1.sazonaws.cos.cm" UM TimyS3 idefine CUST_UNUTER_IDE '1234' UM Feather53 idefine CUST_UNUTERUSTING' carduing_a100' S.ODI Ultra v1 char HOST_ADDRESSIC=CUST_BOKER; UD/S iden CUST_UNUTER_IDE CUST_BOKER; S.ODI Ultra v1		Programmer: "AVRISP mkll"		ESP32-S3-USB-OTG
const char* password = "13918134855"; ESP322x Native USB tidef ATS_LCUDD_EINABLE UM TimyPICO tinclude (ATS_1OT.b) UM TeatherS2 tedfine CUST_SSID "songhinobile" UM TimySICO tedfine CUST_SSID "songhinobile" UM TimySICO tedfine CUST_SSID "songhinobile" UM TimySICO tedfine CUST_SUTER_PASS "13918134855"; UM TimySI tedfine CUST_SUTER_PASS "13918134855"; UM TimyS3 tedfine CUST_SUTER_PASS "13918134855"; UM TimyS3 tedfine CUST_SUTER_PASS "13918134855"; UM TimyS3 tedfine CUST_ROKER, "swiioUrizuagf.sts.iot.cm-north-1.amazonaws.com.cm" UM TimyS3 tedfine CUST_CUST_ROKER; UM FeatherS3 scond UKUT Public CUST_CUST_ROKER; ScODU UKW and I they CUST_ROKER; UK yoo T-Display-S3	const char* ssi	Burn Bootloader		ESP32S3 CAM LCD
bifdef ATS_CLOUD_ENTABLE UMT impRiCO sinclude (ATS_LOUD_ENTABLE) UM FeatherS2 sinclude (ATS_LOUD_ENTABLE) UM FeatherS2 sinclude (ATS_LOUD_ENTABLE) UM FeatherS2 sinclude (ATS_LOUD_ENTABLE) UM TimyS2 sinclude (ATS_LOUD_ENTABLE) UM TimyS2 sinclude (ATS_LOUTE_PASS '13918)34855' UM TimyS2 sinclude (ATS_LOUD_CLIENT_DATS_1)34855' UM TimyS3 sinclude (UST_LOUD_CLIENT_DATS_1)34835'' UM TimyS3 sinclude (UST_LOUD_CLIENT_DATS_1)34835'' UM TimyS3 sinclude (UST_LOUD_CLIENT_DATS_1)34835'' UM TimyS3 sinclude (ATS_LOUD_CLIENT_DATS_1)34835'' UM TimyS3 sinclude (ATS_LOUD_CLIENT_DATS_1)34835'' UM TimyS3 sinclude (ATS_LOUD_CLIENT_DATS_1)34835'' UM TimyS3 sinclude (ATS_LOUD_CLIENT_DATS_1)34835'' S.ODI Ultra v1 sinclude (ATS_LOUD_CLIENT_DATS_1)34835'' S.ODI Ultra v1 sinclude (ATS_LOUD_CLIENT_DATS_1)34835'' S.ODI Ultra v1	const char* passw	rord = "13918134855";		ESP32S2 Native USB
sifder ATS_CLOUD_ENTABLE UM FeatherS2 sinclude (ATS_IOT.h) UM FeatherS2 Neo define CUST_SIDI "songhuiaobile" UM TrayS2 define CUST_SIDI "songhuiaobile" UM RMP define CUST_SIDI "songhuiaobile" UM TrayS3 define CUST_SIDI "songhuiaobile" UM TrayS3 define CUST_SIDUTER_PASS "1391814455" UM TrayS3 define CUST_SIDUTER_PASS "1391814455" UM TrayS3 define CUST_UROKER "avris0uTraugi, ats.iot.cm-north-1.amazonavs.com.cm" UM FeatherS3 define CUST_UROKER (avris0uTraugi, ats.iot.cm-north-1.amazonavs.com.cm" UM FeatherS3 define CUST_UROKERSICUST_BOKER; UM FeatherS3 char HOST_ADDRESSIC-CUST_BOKER; S.ODI Ultra v1 char HOST_ADDRESSICUST_BOKER; LilyGo T-Display-S3				LIM Tim/PICO
<pre>#include (AFS_IOT.h) UM FeatherS2 Neo UM FeatherS2 Neo UM TinyS2 #define (UST_SSID "senghiabebile" #define (UST_SUTER_PASS "13918134855" #define (UST_ROUTER_PASS "13918134855" #define (UST_ROUTER_PASS "13918134855" UM TinyS3 #define (UST_CLOUD_CLEUT_ID "1234" UM FeatherS3 #define (UST_CLOUT_ENT_ROUTER_#DO" </pre>	#ifdef AWS_CLOU	ID_ENABLE		UM FeatherS2
#define CUST_KOUTER_PASS '13918124855' UM TimyS2 #define CUST_ROUTER_PASS '13918124855' UM KMP #define CUST_ROWTER_PASS '13918124855' UM TimyS3 #define CUST_ROWTER_PASS '13918124855' UM TimyS3 #define CUST_ROWTER_PASS '13918124855' UM TimyS3 #define CUST_ROWTER_TANS_01 UM FROS3 #define CUST_ROWTER_UST_ROWTERSIG=CUST_ROWTERSIG=	<pre>#include <aws_iot< pre=""></aws_iot<></pre>	t.h>		UM FeatherS2 Neo
Fadefine CUST_SSID CUM RMP Fadefine CUST_SUBTE_RASS: T9918148557 UM TinyS3 Fadefine CUST_BROKER * swis0u7magf.ats.iot.cm-north=1.asazonaws.cos.cm* UM PROS3 #adfine CUST_ROKER * swis0u7magf.ats.iot.cm-north=1.asazonaws.cos.cm* UM PROS3 #adfine CUST_CLOUD_CLENT_ID * 1234* UM FeatherS3 #adfine CUST_CUST_BROKER; S.ODI Ultra v1 char HOST_ADDRESSID=CUST_BROKER; LUKGOT-Display-S3		ALL STREAM A		UM TinyS2
Heartine UUST_RRWER* (avsis0f*usd) UM TinyS3 #define CUST_RRWER* (avsis0f*usd), ats.ot. cm-north-1. amazonaws. com. cm ² UM PROS3 #define CUST_RRWER* (avsis0f*usd), ats.ot. cm-north-1. amazonaws. com. cm ² UM PROS3 #define CUST_RRWER* (avsis0f*usd), ats.ot. cm-north-1. amazonaws. com. cm ² UM PROS3 #define CUST_RRWER* (avsis0f*usd), ats.ot. cm ² UM FeatherS3 #define CUST_ROTIPUELISHTOPIC *arduino_m100* S.OOU Utra v1 char. HOST_ADDRESS[=CUST_RRWER; LilyGo T-Display-S3	#define CUST_SSID) "songhuimobile"		UM RMP
Hoefine (USI_DOWLEX #VISUUXUBE1.45.)0.Chtmorth1.Basconavs.com UM PROS3 Hedefine (USI_CUDUC_LIENT_0) '234' UM FeatherS3 #define (USI_KQTIPUELISHTOPIC 'arduino_B100' S.OOU UHra v1 char HOST_ADDRESS[=CUSI_BROKER; LilyGo T-Display-S3	#define CUST_ROUT	ER_PASS 13918134855		UM TinyS3
Image: Image: Image: Construction (Line) UM Feather S3 define: CUST_USUTFUELISET(C) "and duino_min0" S.ODI Ultra v1 char: HOST_ADDRESS[-CUST_BROKER; LilyGo T-Display-S3	#define CUST_BROK	IEK awwisUu/xuagi.ats.iot.cn-north-1.amazonaws.com.cn ID CLIENT ID (1994)		UM PROS3
char HOST_ADDRESS[]=CUST_EROKER; S.ODI Ultra v1 LilyGo T-Display-S3	#define CUST_CLOU	PUDITCHIOTC Cardwine =100		UM FeatherS3
Char CLERT TDTT LilyGo T-Display-S3	char HOST ADDRESS	Carterio arterio arterio		S.ODI Ultra v1
	char CLIENT ID[]=	CUST CLOUD CLIENT ID		LilyGo T-Display-S3

Tools-->Board-->Port-->Select the right COM. Users can check the COM port in device manager like the

following picture. If there is not this COM port in device manager, users can install the serial port driver.

🛃 Computer Management <u>File Action View H</u>elp 🗢 🏟 🙍 📅 🔽 🗩 🔚 Computer Management (Local) USR-FEUWTMNMYOU Actions System Tools Task Scheduler Event Viewer Audio inputs and outputs Device Manager . Biometric devices Bluetooth More Actions Generation of the second Cameras Computer Disk drives Display adapters Firmware Services and Applications Human Interface Devices Keyboards Memory technology devices Mice and other pointing devices Monitors Ports (COM & LPT) USB-SERIAL CH340 (COM19) ~ Printers Processors Security devices Software devices Sound, video and game controllers Storage controllers Universal Serial Bus controllers Universal Serial Bus devices Universal Serial Bus device USB Connector Managers

Download link: https://www.pusr.com/support/download/Serial-Driver-USB-To-Serial-Driver.html





3.2.2. Download Program

Users could edit, compile, and download code on the Arduino IDE.

Click the "Upload" button to compile the code first, When the compilation is about to be completed, press and hold the reload button, power on the USR-EG118. Do not release the reload button during the program download process, otherwise the program will fail to be burned.

Note: The

Image: Second	File Edit Sketch Tools Help	
Diminicade Image: State		Q
<pre>Enclude (FIFLb) Enclude (</pre>	0911m100okv1	×
<pre>Hinduk (IFI/Lim: h) Hinduk (ASPAUS: h) Hinduk</pre>		
<pre>end of with a start of the start of the</pre>	Munduda /WIWI ha	
<pre>Huble' (%bServer.b) Finduk' (%bServer.b) Findu</pre>	#include (WiFiClient.h>	
<pre>Findude (SP*005.h) Cont that* sid = "somphiseble"; cont that* somphiseble"; cont that* somphiseble cont that* somethiseble cont that* some</pre>	#include < WebServer .h>	
<pre>const charr soid = "souphulabils"; const charr sarved = "lostBladsos"; eifde AS_COUP_BABLE include 4AS_COUP_CABLE =define COST_SOUTER_PASS 'lostBladsos" =define COST_COUTER_PASS 'lostBladsos =define COST_</pre>	#include < ESPaDNS. h>	
<pre>cont thr* parserd = '1918134895'; iiidef AYS_(CLOUD_DIAGE incluée 'AYS_(CLOUD_DIAGE incluée 'AYS_(CLOUD_DIAGE) idefine CUST_SUTION_DIAGE 'AS '1918134895' idefine CUST_SUTION_DIAGE 'AS '1918134895' idefine CUST_SUTION_CLIENT_D' 'Archine_alou' idefine CUST_SUTTRENESTOPE' 'Archine_alou' idefine CUST_CUDECATION_CLIENT_D' idefine CUST_CUD</pre>	const char* ssid = "songhuimobile";	
<pre>stdia 475_CLOUD_ENALLE tinclude 475_107.b) stdefing CUTS_LSDD *snohlabile' stdefing CUTS_LSDD *snohlabile' stdefing CUTS_LSDD *snohlabile' stdefing CUTS_LSDD *snohlabile' stdefing CUTS_LDDD_LIDE_ID' table 1000_LIDE_ID' t</pre>	const char* password = "13918134855";	
<pre>initial algo and algo and algo and algo algo algo algo algo algo algo algo</pre>	Wifelaf ANG CIOND ENADLE	
<pre>#define CUST_SKUE *semphuisebile" #define CUST_ROUTE_PLAS *1381814655" #define CUST_ROUTE_PLAS *1381814655" #define CUST_CUDU_CLET_ID '1244" #define CUST_CUDUCUT_CUTU_ENCEEN; char CLERT_ID '1244" #define CUST_CUDUCUT_CUTUE_</pre>	#include <aws_iot.h></aws_iot.h>	
<pre>#define CUST_SSID *:enaphulaobile" #define CUST_ROXER_MASS 19918134895" #define CUST_ROXER_MASS 19918134895" #define CUST_ROXER_MASS 100"*usef, ats.iot.cmmorth-1.amzonavs.com" #define CUST_ROXERS[:-CUOUC_LIENT_D':234" #define CUST_ROXERS</pre>		
Hermin Uos_JONGRER Text Softwarf, ask Isot.chronth-Laszonaws.cos.cn [*] Hermin CUST_ROKER Text Softwarf, ask Isot.chronth-Laszonaws.cos.cn [*] Hermin CUST_LOUD_CLIENT_ID * 1234 [*] Hermin CUST_ADDRESS[]=CUST_EROKER; char CLENT_ID[]= CUST_CLOUD_CLIENT_ID; Done uploading Friting at GADDOCENSA (N %) Friting at GADDOCENSA (N %) Friting at GADDOCENSA (S %) Friting At GA	#define CUST_SSID "songhuimobile"	
<pre>rdefine CUST_CLOUD_CLIENT_DD '1234' rdefine CUST_KOITPORLISHTOPIC '1234' rdefine CUST_KOITPORLISHTOPIC '1234' rdefine CUST_KOITPORLISHTOPIC '1234' char CLIENT_DD]= CUST_CLOUD_CLIENT_ID; Doneuploading rdefine at 0x000c8524 (75 %) rdefine at 0x000c8524 (75 %) rdefine at 0x000c8524 (84 %) rdefine at 0x000c8524 (88 %) rdefine at 0x000c8524 (98 %) rdefine at 0x000c</pre>	#define CUST_BROKER "awwis0u7xuagf.ats.iot.cn-north-1.amazonaws.com.cn"	
<pre>#define CUST_NOTIFEURSENCEST = stating_m100" char CUI_ENT_IDI]= CUST_CLOUD_CLIENT_IDI] CONSUMPTIONESSI.===================================</pre>	#define CUST_CLOUD_CLIENT_ID "1234"	
<pre>char KORESS[_UDST_ROKER; char CLENT_DD= CUT_CLOUD_CLENT_DD; Dene uploading riting at 0000c8234 (10 %) riting at 0000c8234 (20 %) riting at 0000d124 (39 %) riting at 0000d124 (39 %) riting at 000004588 (40 %) riting at 000004588 (100 %) rector 901984 bytes (017011 compressed) at 0x00010000 in 9.7 seconds (effective 782.6 kbit/s) Hash of data verified. leaving Ration estima via ETS pin</pre>	#define CUST_MQTTPUBLISHTOPIC "arduino_m100"	
Done uploading. Friting at 0x000c257a (78 %) Friting at 0x000c109 (84 %) Friting at 0x000c109 (84 %) Friting at 0x000c4104 (89 %) Friting at 0x000c4805 (98 %) Friting at 0x000c4805 (97 %) Friting at 0x000c4805 (97 %) Friting at 0x000c4805 (100 %) Frote 951984 bytes (017011 compressed) at 0x00010000 in 9.7 seconds (effective 782.6 kbit/s) Hash of data verified. Leaving Easting vis 815 pin	char HOST_ADDRESS[]=CUST_EROKER;	
Doneuploading Triting at 0x000c257a (3% %) Triting at 0x000c257a (4% %) Eriting at 0x000d2524 (8% %) Triting at 0x000d2524 (9% %) Triting at 0x000d2526 (3% %) Triting at 0x000d2582 (3% %) Eriting at 0x000d2582 (10% %	char chien_ibii= cosi_ccoub_chien_ib;	
<pre>kriting at 020002234 (19 %) Friting at 0x000e324 (36 %) Friting at 0x000e334 (36 %) Triting at 0x000d3454 (39 %) Friting at 0x000d3454 (39 %) Friting at 0x000e3826 (49 %) Friting at 0x000e3826 (10 %) Friting at 0x000e3826 (10</pre>	Done uploading.	
<pre>triing at 0x000cs105 (34 %) rriing at 0x000cs105 (35 %) rriing at 0x000cs128 (35 %) rriing at 0x000seBesL (45 %) Triing at 0x000seBesL (45 %) Friing at 0x00seBesL (45 %) Friing</pre>	Writing at 0x000c2b/a (78 %) Writing at 0x000c8324 (81 %)	
<pre>triting at 0x000d124 (89 %) Triting at 0x000d124 (89 %) Friting at 0x000e48bb (22 %) Friting at 0x000e48bb (22 %) Friting at 0x000e48bb (24 %) Friting at 0x000e430d (97 %) Friting at 0x000e430d (100 %) Frote 951994 bytes (017011 compressed) at 0x00010000 in 9.7 seconds (effective 782.6 kbit/s) Eash of data verified. Leaving Eaving via 8T5 pin</pre>	Writing at 0x000ce109 (84 %)	
<pre>Rriting at 0x000dr124 (89 %) Friting at 0x000dr480 (29 %) Friting at 0x000e480 (24 %) Friting at 0x000e480 (07 %) Friting at 0x000f480 (10 %) Frote 951984 bytes (017011 compressed) at 0x00010000 in 9.7 seconds (effective 782.6 kbit/s) Hash of data verified. Leaving Hard resetting via RTS pin</pre>	Writing at 0x000d3b2b (86 %)	
<pre>Arting at 0x000e8882 (34 %) Arting at 0x000e8882 (37 %) Friing at 0x000e18882 (100 %) Friing at 0x000e18882 (100 %) Friing at 0x000e18882 (100 %) Erote 851984 bytes (017011 compressed) at 0x00010000 in 9.7 seconds (effective 782.6 kbit/s) Each resetting via 8TS pin Each resetting via 8TS pin</pre>	Writing at 0x000dc124 (89 %)	
Triting at 0x000ef30d (97 %) Friting at 0x000f498d (100 %) Ercte 951984 bytes (617011 compressed) at 0x00010000 in 9.7 seconds (effective 782.6 kbit/s) Eash of data verified. Leaving Leaving Hercesting via 875 pin	Writing at 0x000098e3 (92 %)	
Rriting at 0x000f4984 (100 %) Wrote 951984 bytes (817011 compressed) at 0x00010000 in 9.7 seconds (effective 782.6 kbit/s) Hash of data verified. Leaving Hard resetting via RTS pin	Writing at 0x000ef30d (97 %)	
Frote 951994 bytes (01701 compressed) at 0x00010000 in 9.7 seconds (effective 781.8 kbit/s) Harb of data verified. Leaving Hard resetting via RTS pin	Writing at 0x000f498d (100 %)	
Leaving Ard setting vis RTS pin	Wrote 951984 bytes (617011 compressed) at 0x00010000 in 9.7 seconds (effective 782.6 kbit/s) Hash of data verified	
Leaving Eard resetting via ETS pin		
Kard resetting via RTS pin	Leaving	
	Hard resetting via RTS pin	
	<	>

3.3. Programming via ESP32 download tool

Users can export the BIN file through Arduino or other tools, and then download the BIN file through FLASHDOWNLOAD TOOL.



3.3.1. Export compiled binary file

• Users can edit code on Arduino IDE, and export compiled binary file.



3.3.2. Load BIN file

xxx.ino.bin	Start register address:	0x10000
xxx.ino.bootloader.bin	Start register address:	0x1000
xxx.ino.partitions.bin	Start register address:	0x8000

The xxx.ino.bin file is the binary file exported above. The bootloader and partitions can be downloaded



from our official website:

https://www.pusr.com/support/download/Reference-Document-USR-EG118-Demo-and-Reference-

Document.html

Select COM port, BAUD is 921600.

SPIFlashConfig: 40MHz, DIO, DoNotChgBin

ESP32 FLA	SH DOWNLOAD TOOL V3.9.5		-		×
SPIDownlos Dtv0900 Dtv0900 Dtv0900 Dtv0900 Dtv0900 1. 1.	d iokv1\0908m100okv1.ino.bin iokv1\0908m100okv1.ino.bootloade iokv1\0908m100okv1.ino.partitions.	r.bin	000000	0x10000 0x1000 0x8000	
SPIFlashConf SPI SPEED 40MHz 26.7MHz 020MHz 080MHz 3.	QUO QUO QOUT © DIO © DOUT © FASTRD	DetectedInfo flash vendor: SEh : ZB flash devID: 4016h QUAD:4MB crystal: 40 Mhz	 100	1	~
DownloadPar SYNC 等待上电同步 START	stop erase com: co BAUD: 92	M15 1600			< > > >

1> Power off the USR-EG118, hold on the reload button, power on the USR-EG118 device, click "START" button, it starts to programming.



SPIDownload D:\v0908okv1\0904 D:\v0908okv1\0904 D:\v0908okv1\0904 D:\v0908okv1\0904 SPIFlashConfig SPIFlashConfig SPI SPEED SPI MOI O:\v0008clash	8m 100 okv 1. in o. bin 8m 100 okv 1. in o. bootloade 8m 100 okv 1. in o. partitions. I	r.bin bin			00000	0x10000 0x1000 0x8000	
D:\v0908okv1\0908	8m100okv1.ino.bin 8m100okv1.ino.bootloade 8m100okv1.ino.partitions.l	r.bin			00000	0x10000 0x1000 0x8000	
D:\v0908akv1\090 D:\v0908akv1\0908akv1\090 D:\v0908akv1\090 D:\v0908akv1\090	8m100okv1.ino.bin 8m100okv1.ino.bootloade 8m100okv1.ino.partitions.l	r.bin Din			0 0 0 0 0 0	0x10000 0x1000 0x8000	
D:\v0908okv1\090/ D:\v0908okv1\090 D:\v0908okv1\090 D:\v0908okv1\090 D:\v0908okv1\090 D:\v0908okv1\090 D:\v0908okv1\090 D:\v0908okv1\090 D:\v0908okv1\090	8m100okv1.ino.bootloade 8m100okv1.ino.partitions.l	r.bin			0 0 0 0	0x1000 0x8000	
	8m100okv1.ino.partitions.l	bin		···· ··· ···	0 0 0	0x8000	
SPIFlashConfig SPI SPEED SPI MOI					0 0 0		
SPIFlashConfig SPI SPEED SPI MOI					0		
SPIFlashConfig SPI SPEED SPI MOI					@		
SPIFlashConfig SPI SPEED SPI MOI				March			-
SPIFlashConfig SPI SPEED SPI MOI					0		1
SPI SPEED SPI MOI					0		
26.7MHz QQUT 26.7MHz QQUT 20MHz ODO 380MHz ODOUT CFASTR	DE DoNotChgBin LockSettings CombineBin Default	Detectedinto flash vendor: 5Eh : ZB flash devID: 4016h QUAD;4MB crystal: 40 Mhz	~				
DownloadPanel 1							
SYNC							^
STIT L. U. M. D.							~
START STOP	ERASE COM: CO	M15					×
0.01	BAUD: 921	1600					~

2> Finished the programming.

SPIDownloa	d						
☑ D:\v0908	30kv1\0908m1	00okv1.ino.bin			 0	0x10000	
D:\v0908	30kv1\0908m1	00okv1.ino.bootload	ler.bin		 0	0x1000	
D:\v0908	30kv1\0908m1	00okv1.ino.partitions	s.bin		 0	0x8000	
					 0		
					 0		
					 @		
					 0		
					 0		
 40MHz 26.7MHz 20MHz 80MHz 	QIO QOUT ● DIO ○ DOUT ○ FASTRD	DoNotChgBin LockSettings CombineBin Default	flash vendor: 5Eh : ZB flash devID: 4016h QUAD;4MB crystal: 40 Mhz	^			
DownloadPar TNISH 完成	nel 1 AP: A08765FD BT: A08765FD	BDF9 STA: A0B765F BDFA ETHERNET: A	D8DF8 .0B765FD8DFB OM15	v			
START	STOP	RASE					



4. Function introduction and testing

PUSR provides a simple demo program to facilitate customers to test various hardware interface functions. The main functions of the Demo program include: IO port testing, setting AP/STA mode, data transparent transmission over TCP/MQTT protocol.

Note: All the following tests in this section are based on code provided by PUSR.

4.1. Get starting

The default parameter of USR-EG118 is shown in the following table.

Item	Value
SSID	usriot
Password	012345678
IP Address	192.168.4.22

Users can connect to EG118 via Wi-Fi. Then enter 192.168.4.22 in URL blank of browser.

	🕴 Search for M100-PUSR IOT 🛛 🗙	M100 rootweb	🕺 edge cpmputing IoT gateway_O 🗙 🗎	🔋 USR-TCP232-E2-H7外观尺寸图。 → 🖌 💼 USR-TCP2.
\leftarrow C	▲ 不安全 192.168.4.22			
M100	: m100 config			
1.0 dout cont	ig 2.0 wifi_sta_mode 3.0 wifi_SOF	T_AP_mode 4.0 TCP_DTU_MODE	5.0 MQTT_DTU_MODE 7.0AWSONLY	MODE 8.0set_IDLE_MODE

4.2. IO Interface

Click the first button on the webpage, it will navigate to the following page. Users can control the on and off of

DO through the switch on this interface.



4.3. STA setting

Click the second button, and enter the SSID and password of the router which USR-EG118 will connect to. With this demo code, the length of the SSID and password should be less than 10 characters.

\leftrightarrow \rightarrow C \blacktriangle	不安全 192.168.4.22/wifi_	sta_settingPAGE	
method='POST'>	G806s	12345678	submit
	SSID	Password	

The setting will take effect after restarting. To improve efficiency, users can set all parameters at once and then



restart.

4.4. TCP mode setting

Click 4.0 TCP_DTU_MODE button, then set the server IP and port. In TCP mode, the device work at TCP client mode only for now.

	192,168,4,22/	tcpserversettingPA	GE	a di	
method='POST'>	rver IP	server	port	submit	eset
Server IP: 172.16.11.221 Server port: 8081					
← → C ○ Q 192.168.1.142/tu ★书签 □ 手机书签 ■ Wifi速率相关知识	:pserversettingPAGE 12019爆款总结, 11 📌 济雨	南有人物联网技 🔥 济南有人物联	€网技 ▲ ⁸	7月管理 🦂 MES 有人物]	联网 🤺 用户登录 -客户支: 🔞
method='POST'> 172.16.11.221 On TCP server side.	8081	submit reset			

.

Local Host: 172.16.11.221

Local port: 8081

From the following picture, we can see the data can be transmitted between the serial port and TCP server.

••	串口调试助手	- M-		Network Assistant	- U ×
串口设置	数据日志	<u>VartAssist V4.3</u>	Settings	Data log	NetAssist V5.0.2
出口号 Com18 #02 →	[2023-11-30 17:51:51 692]# SEND ASCII)		TCP Server		~
成将率 PPPE	data from serial port		(2) Local Host Addr	[2023-11-30 17:49:30.866]# Client 172.16.10.50:52287 ge	ts online.
秋堀台 8 ▼	[2023-11-30 17:51:54.677]# RECV ASCII> data from TCP server		172.16.11.221	[2023-11-30 17:51:51.867]# RECV ASCII FROM 172.16.10.50	:52287>
· / / / / / / / / / / / / / / / / / / /			(3) Local Host Port	data from Serial port	
			8081	[2023-1 30 17:51:54.622]# SEND ASCII TO ALL>	
关闭			· Close	A A ON ACA SEATER	
接收设置					
• ASCII C HEX			Recy Perions		
▶ 按日志模式显示		/	ASCII C HEX		
□ 接收区自动换行			Auto Linefeed		
白动浓展 海栓编版		\times	Hide Received Data	<	
			Save Recv to File		
分包设置 界面主题			AutoScroll Clear		
			AutoBenly Themes		× .
安洋沿黑			BatchSend Export	Data Send Clients: All Connections (1)	n 🚽 Clear 🐔 Clear
• ASCII C HEX			ASCIT-Man_Donate	data from TUP server	
□ 自动解析转义符			ASCIL C HEX		
☑ AT指令自动回车			Use Escape Chars ()		
□ 自动发送校验位	数据发送 1. DCD ● 2. RXD ● 3. TXD ● 4. DTR ● 5. GHD ● 6. DSR ● 7. RTS ● 8. CTS ● 9. R	【● √清除 ~清	🔲 🔲 Auto Append Bytes		Send
「循环周期 500 ms	data from serial port		Send from File		
快捷定义历史发送		友氏	Shortent History		
(● 就绪!	, 4245/5 RX:265466	TX:74 复位计参	Call Contract Instory		
Contraction Contraction			Heady!	171 BX:21	IX:20 Heset

4.5. MQTT mode setting

In MQTT mode, users can set server IP, username and password to connect the MQTT server to communicate.



The setting will take effect after restarting. To improve efficiency, users can set all parameters at once and then

restart.

6 0	Rearch for M100-PUSR IOT	× 192.168.4.22/mqttserve	rsetting ^{p,} × 📌 edge cpmp	outing loT gateway_O 🗙 📔 USR-TO	CP23
← C	▲ 不安全 192.168.4.22/	mqttserversettingPAGE			
method=	='POST'>			submit reset	
	server IP	Username	Password		

In this document, we set the parameters like the following:

Server IP: 123.233.249.74

Username: mqtt_10

Password: USR-test

Publish: songhuioutTopic

Subscribe: mqtt_to_rs485

← C	▲ 不安全 192.168.1.142	/mqttserversettingPAGE		
method='POS	ST'> 123.233.249.74	mqtt_10	USR-test	submit reset

Settings of EG118

🛞 моттх		- 🗆 X
File Edit View Window Help		
Connections New Collection	< Back Edit	Connect
• MQTT@123.233.249.7	General	
	* Name MQTT	
	* Client ID 123.233.249.74	C O
<u>ч</u>	* Host mqtt:// v 123.233.249.74	
+	* Port 1883	× ×
	Username mqtt_10	
	Password	
	SSL/TLS	
	Advanced A	
O	Connect Timeout 10	(s)
	Keep Alive 60	(s)
0	Clean Session 🧿 true 🔷 false	
	Auto Reconnect 🔿 true 💿 false	

Settings of MQTTx software





4.6. Connecting to AWS platform

To use this code connecting to the AWS platform, you need to ensure that the EG118 is connected to a router that can connect to the Internet through Wi-Fi.

EG118 can connect to the AWS platform through the MQTT protocol, and the corresponding parameters have been encapsulated in the demo code. Users do not need to care about the parameter settings on the EG118 side. Users can use MQTTfx.exe to test this function.

۶	Open MQTTfx.exe
	Broker Address: awwis0u7xuagf.ats.iot.cn-north-1.amazonaws.com.cn
	Broker Port: 8883
	Client ID: Generate
	SSL/TLS: Load the certificates, check the PEM Formatted
	Publish/Subscribe topic: m100_ard
	The needed certificate file can also downloaded from the official website:
	https://www.pusr.com/support/download/SSL-Certificate-SSL-Certificate-Files-for-AWS.html



Edit Connection Profiles	- D X
M2M Eclipse MQTT-Test	Connection Profile
	Profile Name MQTT-Test
	Broker Address awwis0u7xuagf.ats.iot.cn-north-1.an Broker Port 8883 Client ID 6b17a842d19f491792cb1234280f54 Generate
	General User Credentials SSL/TLS Proxy Last Will and Testament
	CA signed server certificate CA certificate file CA certificate keystore Self signed certificates
	CA File D:\桌面\123\ca1.pem
	Client Certificate File D:\桌面\123\67d7ec097c-certificate.pem.crt
	Client Key File D:\桌面\123\67d7ec097c-private.pem.key
	Client Key Password
	PEM Formatted 🗸
	Self signed certificates in keystores
+ -	Revert Cancel OK Apply

> Click "Connect" button to connect the AWS server, subscribe to the "Arduino_m100" topic.

MQTT.fx - 1.3.1		×
File Extras Help		
MQTT-Test	Connect Disconnect	•
Publish Subscribe Scripts Broker	Status Log	
arduino_m100 💌	Subscribe QoS 0 QoS 1 QoS 2 Autoscroll	0°*
arduino_m100 3	arduino_m100	1
Dump Mess M Unsubscri)	arduino_m100	2
	arduino_m100	(3)
	arthring m100	3
	0.5 0 27-00-2022 20-17-57 730	5
	deviceid42 [0]->-1	
	Plain JSON Hex Base64	

Connect the PC and USR-EG118 device via USB to RS485, open Modbus Slave tool, set the Slave ID to 42, and set the address to 31000.



نغ Modbus Slave - Mbslave1	- 0	×
File Edit Connection Setup Display View Window Help		
He clat connection Step Display View Window Help Window Help Image: Step Display View Window Help Image: Step Display View Window Help Image: Step Display View Window Help Image: Step Display View Window Help Image: Step Display View Window Help Image: Step Display View Window Help Image: Step Display View Window Help Image: Step Display View Window Help Image: Step Display View Window Help Image: S		
For Help, press F1. Port 15: 115200-8-N-1		

> Serial settings, distinguish the RS485 COM from the Download COM.

式 Modbus Slave - Mbslave1			×
File Edit Connection Setup Display View Window Help			
Pile Lan Connection Setup Display View Window Help Image: Setup Display View Window Hel	Connection Setup X Connection OK Serial Port Cancel Serial Settings Cancel USB-SERIAL CH340 (COM15) Cancel I115200 Baud Mode ® Rtu ASCII Flow Control Flow Control IDSR CTS I Stop Bit Image: Imag		
	TCP/IP Server IP Address I27.0.0.1 Any Address Ignore Unit ID		
For Help, press F1.	Port 17: 115200-	3-N-1	.4

Change the value of register to 77, and the changed data can be received by the MQTTfx. It means the device connect to AWS and can communicate with the RS485 device.



Modbus Slave - Mbslave1	MQ11.TX - 1.3.1		- u x
File Edit Connection Setup Display	File Extras Help		
	MOTT-Test	Connect Disconnect	0
Mbslave1	indir fest	Disconnect	
ID = 42: F = 03	Publish Subscribe Scripts Broker	Status Log	
Alias 31000	arduino_m100 🗸	Subscribe QoS 0 QoS 1	QoS 2 Autoscroll
077_	arduino m100 405	arduino_m100	400
1 0	Dump Mess M Unsubscri	arduina m100	401
2 0			
3 0		arduino_m100	402
4 0		anting m100	(102)
5 0		arduno_mioo	405
6 0		arduino_m100	404
7 0			
8 0		arduino_m100	405
0		arduino m100	405
		-	
		deviceid421[0]->77	
		and the second for the second s	

5. Contact Us

Jinan USR IOT Technology Limited Address : Floor 12 and 13, CEIBS Alumni Industrial Building, No. 3 Road of Maolingshan, Lixia District, Jinan, Shandong, China Official website: https://www.pusr.com Official shop: https://shop.usriot.com Technical support: http://h.usriot.com/ Email : sales@usriot.com Tel : +86-531-88826739 Fax : +86-531-88826739-808

6. Disclaimer

The information in this document provided in connection with Jinan USR IoT technology ltd. and/or its affiliates' products. No license, express or implied, by estoppel or otherwise, to any intellectual property right is granted by this document or in connection with the sale of USR IoT products. EXCEPT AS SET FORTH IN THE TERMS AND CONDITIONS AS SPECIFIED IN THE LICENSE AGREEMENT FOR THIS PRODUCT, USR IOT AND/OR ITS AFFILIATES ASSUME NO LIABILITY WHATSOEVER AND DISCLAIMS ANY EXPRESS, IMPLIED OR STATUTORY WARRANTY RELATING TO ITS PRODUCTS INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTY OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NON-INFRINGEMENT. IN NO EVENT SHALL USR IOT AND/OR ITS AFFILIATES BE LIABLE FOR ANY DIRECT, INDIRECT, CONSEQUENTIAL, PUNITIVE, SPECIAL OR INCIDENTAL DAMAGES (INCLUDING, WITHOUT LIMITATION, DAMAGES FOR LOSS OF PROFITS, BUSINESS INTERRUPTION OR LOSS OF INFORMATION) ARISING OUT



OF THE USE OR INABILITY TO USE THIS DOCUMENT, EVEN IF USR IOT AND/OR ITS AFFILIATES HAVE BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. USR IOT and/or its affiliates make no representations or warranties with respect to the accuracy or completeness of the contents of this document and reserves the right to make changes to specifications and product descriptions at any time without notice. USR IOT and/or its affiliates do not make any commitment to update the information contained in this document.







Official Website: www.pusr.com Official Shop: shop.usriot.com Technical Support: h.usriot.com Inquiry Email: inquiry@usriot.com Skype & WhatsApp: +86 13405313834 关注有人微信公众号 登录商城快速下单 Click to view more: Product Catalog & Facebook & Youtube