

# **Open Source IoT Gatewa**

# USR-M100-ARD

# **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. Dimension	4 -
2. Arduino IDE Using	
3. Program burning	6 -
3.1. Programming via Arduino	6 -
3.2. Programming via ESP32 download tool	
4. Functional Test	11 -
5. Contact Us	16 -
6. Disclaimer	16 -



## 1. Introduction

## 1.1. Overview

USR-M100-ARD 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-M100-ARD Parameters
Chipset	ESP32-WROVER-E
Professor	Dual-core Xtensa <sup>®</sup> 32-bit LX6 MCU, 1.2GHz
RAM	8M
Flash	4M
Power Supply	9~36V
Hardware Interfa	ce
RS232/485	1*RS232/485, Default parameter: 115200, N, 8, 1
DI	1*digital input
DO	2*digital output
AI	1*analog input(4~20mA)
Wi-Fi	
Stadard	IEEE802.11b/g/n
Speed	Up to 150Mbps
Frequency	2412 ~ 2484 MHz
Others	
BLE	Compatible with BEL4.2

### Table 1. Parameters of USR-M100-ARD

USR-M100-ARD provides 2 indicators in total, the specific description is as follows.

### Table 2. LED indicator

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



## 1.3. Dimension

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



Figure 1. Dimension of USR-M100-ARD

# 2. Arduino IDE Using

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

2> 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					- 0	×
						<b>9</b>
0911m100akv1						
0011111000101						^
	Preferences			×		
// ///////////////////////////////////	Settings Network					
#define PHYTESTSAMPLE_ENABLE	Sketchbook location:					
#define WEBserverSAMPLE_ENABLE #define WODBUSTESTSAMPLE ENABLE	D:\文档\Arduino			Browse		
#define FEEDDOGENABLE	Editor language:	English (English)	(requires restart	of Arduino)		
#define AWS_CLOUD_ENABLE	Editor font size:	12				
	Interface scale:	Automatic 100 ** (requires restart of Arduino)				
	Show verbose output during:	Compilation Upload				
	Compiler warnings:	None v				
#include <wifi.h></wifi.h>	Dicplay line numbers					
<pre>#include <wificlient.h> #include <wificlient.h></wificlient.h></wificlient.h></pre>	Brable Code Folding					
#include <webserver.n> #include &lt;#SPaDNS.b&gt;</webserver.n>	Verify code after upload					
	Use external editor					
const char* ssid = "songhuimobile";	Aggressively cache compil	ed core				
const char* password = "13918134855";	└ Check for updates on star	tup				
HIGH AWS CLOUD FUARIF	☑ Update sketch files to ne	w extension on save (.pde -> .ino)				
#include <aws_iot.h></aws_iot.h>	✓ Save when verifying or up	loading				
	Additional Boards Manager UR	Ls: https://dl.espressif.com/dl/package_esp32_index.jso	n 🗖			
#define CUST_SSID "songhuimobile"	More preferences can be edit	ed directly in the file				
#define CUST_ROUTER_FASS_IS916154655 #define CUST_BROKER "awwisQu7xuagf.ats.iot.cn=north-	C:\Users\Administrator\AppDa	ta\Local\Arduino15\preferences.txt				
#define CUST_CLOUD_CLIENT_ID "1234"	(edit only when Arduino is n	ot running)				
<pre>#define CUST_MQTTPUBLISHTOPIC "arduino_m100"</pre>			OK	Cancel		
char HOST_ADDRESS[]=CUST_BROKER;					1	
char CLIENT_ID[]= CUST_CLOUD_CLIENT_ID;						~
		ESP32-WROOM-DA Module, Defa	ault 4MB with spiffs (1.2MB AF	PP/1.5MB SPIFFS),	240MHz (WiFi/BT), QIO, 80MHz, 4MB (32Mb), 921600, Core 1, Core 1, None, Disabled or	COM15

## 3> Install ESP32 Boards

### Tools->Board Manager->Search "ESP"

0911m100okv1   Arduino 1.8.5 File Edit Sketti Tools Help		- 0 ×
		Q.
0911m100okv1		
	begin////////////////////////////////////	^
#define PHYTESTSAMPLE ENABLE 2	S Boards Manager	
#define WEBserverSAMPLE_ENABLE	Type All 2 ESP	
#define MODBUSTESTSAMPLE_ENABLE		
#define FEEDDOGENABLE	Arouno Avk Boards by Arouno Version 1.3.0 INSTALLED Boards included in this package:	
≓define AVS_CLOUD_ENABLE	Arduino Yún, Arduino UNO, Arduino UNO, Mini, Arduino UNO WFF, Arduino Dieeimilia, Arduino Nano, Arduino Mega ADK, Arduino Neno Farta Meda ADK, Arduino Adka Adka Adka Adka Adka Adka Adka Adka	
#include < <b>WiFi.h</b> >		
<pre>#include <wificlient.h></wificlient.h></pre>	Arduino ESP32 Boards by Arduino	
#include < WebServer. h>	Boards included in this package: Arduino Nano ESP32.	
#include <espmdns.h></espmdns.h>	More info	
const charge said = [construinshile];		
const char* password = "13918134855":	esp32 by Espressif Systems version 2.0.10 INSTALLED	
• 4	Boards included in this package: E6030 Dex Board, E6031 S3 Dex Board, E6033 S3 Dex Board, E6033 C3 Dex Board,	
#ifdef AWS_CLOUD_ENABLE	More info	
#include <aws_iot.h></aws_iot.h>		
#define CUST_SSID `songhuimobile`		
#define CUSI_ROUTER_FASS 13918134855	Close	
#define CUST_CLOUD CLIENT ID "1234"		
#define CUST NQTTPUBLISHTOPIC "arduino m100"		
char HOST_ADDRESS[]=CUST_BROKER;		
<pre>char CLIENT_ID[]= CUST_CLOUD_CLIENT_ID;</pre>		~
	ESP32.WPOONLDA Module, Default 4MB with shifts /1 2MB APP/1 5MB SPIEFS). 240MHz (MEURT), DID 80MHz 4MB (32Mb) 921600, Core 1	Core 1 None Disabled on COM15

#### 4> Add library to installation path

PUSR offers library source code of AWS, Modbus, users can download from Github: 插入代码连接



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

Note: Users are strongly encouraged to use the AWS and Modbus libraries we provide.

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

Sketch->Library Manager->Search " Modbus" ->Install

© 0911m100okv1   Arduino 1.8.5		- 0 ×
File Edit Sketch Tools Help		
		۱ <mark>-۹</mark> ۲
0911m100okv1		
// ///////////////////////////////////	begin////////////////////////////////////	
<pre>#include <vifi.h> #include <vificlient.h> #include <vebserver.h> #include <espadms.h> const char* ssid = "songhuisobile"; const char* passvord = "13918134855";</espadms.h></vebserver.h></vificlient.h></vifi.h></pre>	AccCRC by Brian T. Park Cyclic Redundancy Check (CRC) algorithms: crc8, crc16ccit, crc16modbus, crc32. Programmatically converted from C99 code generated by https://gyrc.org to Arduino C+. For each CRC algorithm, 4 variants are available: The 'bit' variant uses brute force bit-by-bit loog (smallest and slowest); the 'nibble' variant uses a 4-bit table (potentially good balance between size and speed); 'nibblem' variant is the same as 'nibble' but 1:9X-2:X faster on the ESP8266; and the 'byte' variant uses an 8-bit table (largest but fastest). More info AgualaboSensor by Andres Sabas, based in the work of Libelium AgualaboSensors. Aqualabo Sensors communication with Modbus slaves over R5232/485 (via RTU protocol). Requires an R5232/485 transceiver. More info	
#ifdef AWS_CLOUD_ENABLE #include <aws_iot.h></aws_iot.h>	DFRobot_RTU by Anya DFRobot Modbus RTU library for Arduino. A library to use an Arduino as master to control and communicate via modbus protocol. More info	
#define CUS1_SSID songhuimobile #define CUST_ROUTER_PASS "13918134855"	~ ~	
#define CUST_BROKER "awwis0u7xuagf.ats.iot.cn-north-1.amaz	Close	
<pre>#define CUST_CLOUD_CLIENT_ID "1234" #define CUST_NQTTPUBLISHTOPIC "arduino_m100" char HOST_ADDRESS]-CUST_ERKER; char CLIENT_ID[]= CUST_CLOUD_CLIENT_ID;</pre>		·
31	FSP22JJRCO1JLDA Modula D45aitf 4/JR with splits (1 2JJR APP/1 5JJR SPlif58); 24/JUH-7 MJR5/	RTh DID 9004-7 LMR (730b) 491600 Core 1 Core 1 None Disabled on COM15
21	E3F32-WROUM-DA Module, Default 4MB with spins (1.2MB APP) . SMB SPIFFS), 240MP2 (Wirke	B1), GIO, BOIRH2, 4MB (S2MB), 821000, CORE 1, CORE 1, NORE, DISABLED OF COM15

# 3. Program burning

## 3.1. Programming via Arduino

1> Hardware connection

USR-M100-ARD\*1

USB to TTL converter \*1

Wi-Fi antenna\*1

The hardware connection is like this:





2> Downloading parameters settings

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

Tools-->Board-->Port-->Select the right COM

File Edit Sketch	ools Help			
0911m100okv1	Auto Format Archive Sketch Fix Encoding & Reload Serial Monitor Serial Plotter	Ctrl+T Ctrl+Shift+M Ctrl+Shift+L		ع ب
#define PHYTEST #define TEBsery #define NOBUSJ #define NOBUSJ #define ATS_CLO #include <tific #include <tific =include <tific< td=""><td>WiF101 Firmware Updater Board: "ESP32-WROOM-DA Module" Flash Mode: "QIO" Flash Size: "AMB (32Mb)" Events Run On: "Core 1* Partition Scheme: "Default 4MB with spiffs (1.2MB APP/1.5MB SPIFFS) Erase All Flash Before Sketch Upload: "Disabled" Arduino Runs On: "Core 1* Flash Frequency: "B0MHz" CPU Frequency: "B0MHz" CPU Frequency: "B0MHz" Upload Speed: "921600" Core Debug Level: "None" Port Get Board Info</td><td></td><td>Arduino Vún Mini Arduino Industrial 101 Linino One Arduino Uno WiFi ESP323 Rov Module ESP325 Dev Module ESP32 WROOM-DA Module ESP32 WROOM-DA Module ESP32 WROOM-DA Module ESP32 WROOM-DA Module ESP32 ROOM-DA MOD</td><td></td></tific<></tific </tific 	WiF101 Firmware Updater Board: "ESP32-WROOM-DA Module" Flash Mode: "QIO" Flash Size: "AMB (32Mb)" Events Run On: "Core 1* Partition Scheme: "Default 4MB with spiffs (1.2MB APP/1.5MB SPIFFS) Erase All Flash Before Sketch Upload: "Disabled" Arduino Runs On: "Core 1* Flash Frequency: "B0MHz" CPU Frequency: "B0MHz" CPU Frequency: "B0MHz" Upload Speed: "921600" Core Debug Level: "None" Port Get Board Info		Arduino Vún Mini Arduino Industrial 101 Linino One Arduino Uno WiFi ESP323 Rov Module ESP325 Dev Module ESP32 WROOM-DA Module ESP32 WROOM-DA Module ESP32 WROOM-DA Module ESP32 WROOM-DA Module ESP32 ROOM-DA MOD	
<pre>#Include Value const char* ssi const char* ssi include value #include value #define CUST_BRO #define CU</pre>	Programmer: "AVRISP mkll" Burn Bootloader ford = "19918134855"; JD_EHABLE f. h> 0 "songhuisobile" FER_PASS '13918134855" ERE "avrigut"Stugf.ats.iot.cn=north=1.asazona*s.com.cn" D_CLEHT_ID '1244' PUPBLISHOPIC "arduino_s100" S(1=CUST_GBOKER; cuST_CLOUD_CLEHT_ID;		ESP32-S3-USB-OTG ESP32S CAM LCD ESP32S CAM LCD ESP32S Varive USB ESP32 Wrover Kit (all versions) UM TinyPICO UM FeatherS2 Neo UM TinyS2 UM RMP UM TinyS3 UM PROS3 UM FeatherS3 S.ODI UITa v1 LilyGo T-Display-S3 mirroS2	





3> 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-M100-ARD. Do not release the reload button during the program

download process, otherwise the program will fail to be burned.



## 3.2. Programming via ESP32 download tool

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





### 2> Open flash\_download\_tool, Chipset-ESP32

🔅 D:\0产品部\3、产品上架\USR-M100-3.8\M100开源版本\flash_download_tool_3.9.5_0\flash_download_tool_3.9.5\flash_download_tool_3.9.5.	_	< [
		^
□ DOW – □ ×		
ChinType: ESP32 V		
		:
WorkMode: Develop V		
LoadMode: UART 🗸		
ОК		
		~

#### 3> 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

Select COM port, BAUD is 921600.

SPIFlashConfig: 40MHz, DIO, DoNotChgBin



ESP32 FLA	SH DOWNLOAD TOOL V3.9.5	-	Ū.	×
SPIDownloa ✓ D:\v0908 ✓ D:	d lokv1\0908m100okv1.ino.bin lokv1\0908m100okv1.ino.bootloader.bin lokv1\0908m100okv1.ino.partitions.bin	0 0 0 0 0	0x10000 0x1000 0x8000	
		 @ @		]
SPI SPEED     O     40MHz     26.7MHz     20MHz     80MHz	SPI MODE QIO QUO QOUT O DONotChgBin Bash vendor: DEh : ZB Flash devID: 4016h QUAD;4MB COMbineBin QUAD;4MB Crystal: 40 Mhz			
3. DownloadPar	el 1			
SYNC 等特上电同步	2.			
START	STOP ERASE BAUD: 921600			``

4> Power off the USR-M100-ARD, hold on the reload button, power on the USR-M100-ARD device, click "START" button, it starts to programming.

SPIDownload         D:\v0908ekv1\0908m100ekv1.ine.bin       @ 0x1000         D:\v0908ekv1\0908m100ekv1.ine.bootloader.bin       @ 0x8000         D:\v0908ekv1\0908m100ekv1.ine.partitions.bin       @ 0         D:\v0908ekv1\0908m100ekv1.ine.partitions.bin       @ 0         D:\v0908ekv1\0908m100ekv1.ine.partitions.bin       @ 0         SPIFlashConfig       @ 0         SPI MODE       DetectedInfo         I LockSettings       flash vendor: ^ 5th : ZB         flash devID:       @ 0         Q20MHz       0 DOUT         O DoUT       Default         QUAD;4MB       @ 0         Q:
O 26.7MHz     O QOUT     LockSettings     SEn : 2B       O 20MHz     O DIO     CombineBin     Guilabrian       O 80MHz     O DOUT     Default     Crystal:       40 Mhz     O Mhz     CombineBin     Guilabrian
OFASIRD

5> Finished the programming.



ESP32 FLA	SH DOWNLO	AD TOOL	V3.9.5					_		$\times$
SPIDownloa	d									
D:\v0908         D:\v0908	ig SPI MODE QUO QUO QOUT DIO FASTRD	100okv1.ir 100okv1.ir 100okv1.ir	no.bin no.bootlo no.partitic NotChgBi cSettings ombineBin Default	in	bin in DetectedInfo flash vendor: 5Eh : ZB flash devID: 4016h QUAD;4MB crystal: 40 Mhz	^		0 0 0 0 0 0	0x10000 0x1000 0x8000	
- DownloadPar FINISH 完成	nel 1 AP: AOB765F BT: AOB765F	D8DF9 ST D8DFA ET	A: A0B7( HERNET:	65FD	8DF8 3765FD8DFB			 		
START	STOP	ERASE	COM: BAUD:	CON 921	M15 600					~

# 4. Functional Test

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

- 1> Preparatory Work
  - > Download the Modbus Slave tool and MQTTfx tool first.
  - > Find a router, change the SSID to "usriot", and change the password to "0123456789".

USR-G816	Status	Mode: Master SSID: usriot BSSID: D4:AD:20:67:FC:17 Channel: 11 (2.462 GHz) Tx-Power: 20 dBm
> Services		
✓ Network	Enable	
WAN	Hide SSID	
LAN	SSID	usriot
Cellular Network		
Network Switch	Encryption	mixed-psk V
Wireless	Кеу	0123456789
WWAN	HW Mode	iing ~
DUCD	1	If STA is enabled, the configuration is affected by STA.

2> Power on the USR-M100-ARD device. If the work indicator is blinking, the Wi-Fi connection is successful.



3> Connect the USB to TTL to USB port, open the Uart Assistant tool, users can check the IP address.

Note: After connecting the USB to TTL, the work indicator is steady on.

· ·	Uart Assistant	×□- ₩
COM Configs	Data log UartAssi	<u>st V5.0.2</u> 🗇 🗘
Channel COM17 #L		^
Baudrate 115200	[2023-09-27 19:58:03 888]# RECV ASCII)	
	[loop_line_395]->hornbill.publish: slave_deviceid->42 success	
Stophite 1	[2023-09-27 19:58:07.408]# RECV ASCII>	
Flowetri NONE		
	WEBserverSAMPLE_ENABLEBEGIN	
	was sarray TP address: http://192.168.1.142	
Recy Options	web server if address. http://102.100.1.142	
ASCII C HEX	ARDUINO ESP32	
Cog Display Mode	KESSETVETSAMPLE_ENABLE END	
Auto Linefeed		
Hide Received Data	APDICTNO PCP22	
j Save Recv to File	PHYTESTSAMPLE_ENABLE BEGIN	
AutoSoroll Clear		
Send Options	[2023-09-27 19:58:08.402]# RECV ASCII>	
ASCII O HEX	[loop_line_395]->hornbill.publish: slave_deviceid->42 success	
✓ Use Escape Chars(i)		~
Auto Append Bytes	Data Send 1. DCD • 2. RXD • 3. TXD • 4. DTR • 5. GND • 6. DSR • 🐓	Clear 👠 Clear
Cucle 50000 mo	at+ver	
Shortcut <u>History</u>		Send
👉 Ready!	2929/0 RX:884360 TX:0	Reset

4> Connect the PC to the same router with the USR-M100-ARD device, and enter 192.168.1.142 on browser url blank. The IP address should be the one of USR-M100-ARD.

Click OUT1\_ON button, the DO1 indicator will turn on.

Click OUT1\_OFF button, the DO1 indicator will turn off.



5> Connect the Ethernet port to the LAN port of a router, the following information will be displayed on Uart Assistant tool. Via this IP address(192.168.1.145), users can access the built-in webpage to control the IO indicator also.



	Uart Assistant	₩ - □ ×
COM Configs	Data log Uart	Assist V5.0.2 🗇 🗘
Baudrate 115200  Paritybits NONE	ARDUINO ESP32	^
Stopbits 1 Flowctrl NONE	ARDUINO ESP32	
Recv Options ASCII CHEX Log Display Mode Auto Linefeed Hide Received Data Save Recv to File	<pre>[2023-09-27 20:04:57.339]# RECV ASCII&gt; [loop_line_395]-&gt;hornbill.publish: slave_deviceid=&gt;42 succes [2023-09-27 20:04:59.267]# RECV ASCII&gt; [case ARDUINO_EVENT_ETH_CONNECTED] ETH Connected case:ARDUINO_EVENT_ETH_GOT_IP: ETH MAC: A0:B7:65:FD:8D:FBA0:B7:65:FD:8D:FB IPv4: localip:192.168.1.145, FULL_DUPLEX,100Mbps</pre>	2
Send Options C ASCII O HEX Use Escape Chars (i) Auto Append Bytes	[2023-09-27 20:04:59.840]# RECV ASCII> [loop_line_388]->hornbill.publish failed Data Send 1.DCD • 2.RXD • 3.TXD • 4.DTR • 5.GND • 6.DSR	✓ Clear ▲ Clear
Send from File Cycle 50000 ms <u>Shortcut</u> <u>History</u>	at+ver	Send

#### M100: set DOUT

OUT1\_ON OUT1\_OFF OUT2\_ON OUT\_2OFF geticurrent

## 6> RS485 and AWS test

> 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



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:(東面)(123\67d7ec097c-private.pem.key
	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 S	Status Log	
arduino_m100	Subscribe QoS 0 QoS 1 QoS 2 Autoscroll Q	٠
arduino_m100 3	arduino_m100	1
Dump Mess M Unsubscri	arduino_m100	2
	arduino_m100	3
		_
	arduino m100	3
	Opt 0 27-09-2023 20:17:57.730	
	deviceid42 [0]->-1	
	Plain JSON Hex Base64	

Connect the PC and USR-M100-ARD device via USB to RS485, open Modbus Slave tool, set the Slave ID to 42, and set the address to 31000.



式 Modbus Slave - Mbslave1			- 🗆	$\times$
File Edit Connection Setup Display View	Window Help			
0 🖆 🖬 😂   🗂   🗏 👜 🦉 😽				
Image: Constraint of the second se	Slave Definition       X         Slave D       42         Function:       03 Holding Register (4k)         Address:       31000         Quantity:       10         Outure       00         Provs       01         01       20       50       100         Firto Quantity:       10         Error Simulation       PLC Addresses (Base 1)         Error Simulation       Insert CRC/LRC error (Not when using TCP/IP)         0       [ms] Response Delay         Return exception 06, Busy			
For Help, press F1.		Port 15: 115200-8-N-1		

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

式 Modbus Slave - Mbslave1		– 🗆 X
File Edit Connection Setup Display View Window Help		
Mbslave1	Connection Setup	
	Connection OK Serial Port Cancel	
	I Stop Bit     Implies a case only       TCP/IP Server       IP Address       O Any Address       Ignore Unit ID	
For Help, press F1.		Port 17: 115200-8-N-1

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	MQT1.TX - 1.3.1		~
File Edit Connection Setup Display	File Extras Help		
D 📽 🖬 🚳 🛅 🖳 🎰 😵 🕺	MQTT-Test	Connect Disconnect	
Mbslave1			
ID = 42: F = 03	Publish Subscribe Scripts Broker S	Status Log	
Al'			
Allas 31000	arduino_m100	Subscribe Qos 0 Qos 1 Qos 2 Autoscro	
	arduino_m100 405	arduino_m100	400
1 0	Dump Mess M Unsubscri	arduino m100	401
2 0			101
3 0		arduino_m100	402
4 0			
5 0		arduino_m100	403
6 0			
7 0		arduino_m100	404
		arduino m100	405
		arduino_m100	405
		005 0 27-00-2023 20:24:50 740	
		dearcerday [[0]_>//	

## 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 Click to view more: Product Catalog & Facebook & Youtube

关注有人微信公众号 登录商城快速下单