

# GSM 3G 4G LTE Cellular IoT Gateway



S175

S375

User Manual

Ver 1.5












Date Issued: 2020-07-29

King Pigeon Hi-Tech. Co., Ltd.

[www.IOT-Solution.com](http://www.IOT-Solution.com)



## Table of contents

1. Brief introduction.....	5
2.Safety Directions.....	5
3. Standard Packing List.....	5
4. Features and Specification.....	6
4. 1 Mainly Features.....	6
4. 2 Specifications.....	7
5. Physical Layout and Installation Diagram.....	8
5.1 Control Unit size.....	8
5.2 Interface definition.....	8
5.3 LED Indicator Definition.....	9
6. Settings&Operation.....	11
6.1 Start to Configure.....	11
6.2 Setting Self-checking.....	13
6.3 Configuration software interface and running.....	14
 Basic Settings.....	14
 Number Settings.....	15
 T&H Trigger Settings.....	16
 T&H Alarm Settings.....	17
 Timer Settings.....	18
 Interlock Settings.....	19
 RS485 Serial Port Settings.....	19
 Slave Settings.....	20
 Register Settings.....	22
 Cellular network Settings.....	22
 Ethernet Settings(only for S37X).....	24
 Historical Record.....	25
7.Configuration and Reset.....	26
7.1 Export Profile.....	26
7.2 Load Profile.....	27
7.3 Reset.....	28
8. Connection and Application.....	28
8.1 Wire Connection.....	28
8.2 Quickly Verify Device Communication Function.....	29
8.3 Modbus RTU Slave Application.....	30



8.4 Modbus RTU Master Application.....	33
8.5 Modbus TCP Master Application.....	39
8.6 Transparent Transmission DTU Application.....	45
8.7 Device connect to cloud Application.....	46
9. Device SMS Command.....	48
10.S17X/S37X Register Address.....	50
11.Slave Mapping Register Address.....	50
12. MQTT Write Identifier & Cloud Platform Read.....	52
13. Upgrade Firmware.....	54
14. Cellular Module Upgrade.....	55
15. Warranty.....	55

This handbook has been designed as a guide to the installation and operation of S17X/S37X GSM/SMS/GPRS/3G/4G Cellular IoT Gateway.

Statements contained in the handbook are general guidelines only and in no way are designed to supersede the instructions contained with other products.

We recommend that the advice of a registered electrician be sought before any Installation work commences.

King Pigeon Hi-Tech.Co., Ltd, its employees and distributors, accept no liability for any loss or damage including consequential damage due to reliance on any material contained in this handbook.

King Pigeon Hi-Tech.Co., Ltd, its employees and distributors, accept no liability for GSM Network upgrading or SIMCard upgrading due to the technology specifications contained in this handbook.

### 【UPGRADE HISTORY】

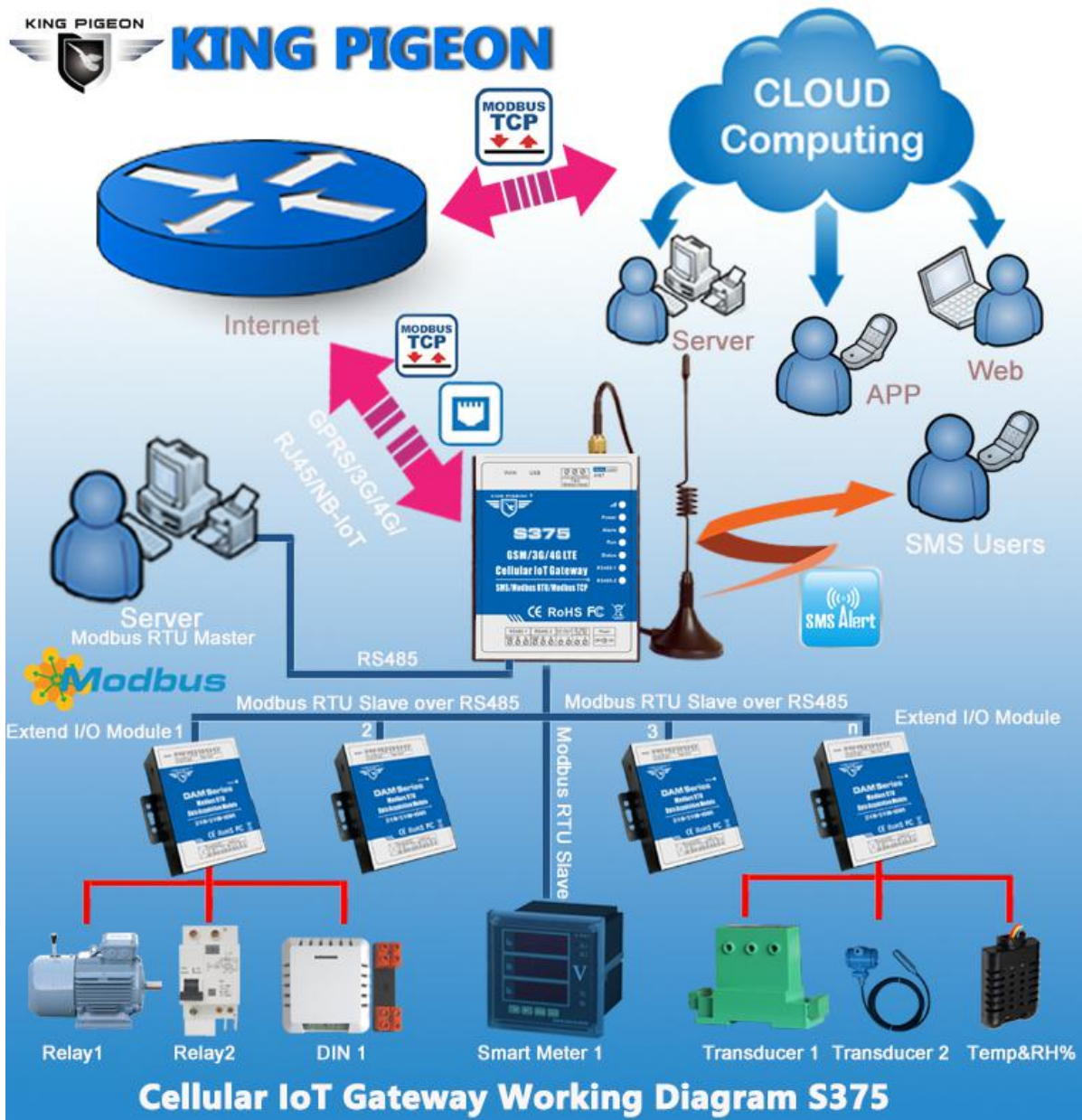
DATE	FIRMWARE VERSION	HARDWARE VERSION	DESCRIPTION
2018.12.10	V 25	V1.3	Modified the start address of the 32-bit map register in Appendix C
2019.03.06	V 26	V1.3	Network port and cellular add MQTT connection to cloud platform
2019.04.20	V27	V1.3	Adding S175E,375 E Series Products
2019.8.15	V28	V1.3	Adding the function of Ethernet as Modbus TCP master
2019.9.23	V28	V1.3	1.Modified Cellular network Settings default parameter 2.Adding MQTT protocol instruction
2020.04.09	V28	V1.3	Revised MQTT data release format
2020.7.29	V28	V1.3	Remove S173 174,S373 S374

## Model List

Model	GSM/3G /4G	AI/DIN/DO	TH	RS485	Ethernet	SD Card	Extend I/O tags			
							Boolean	16-Bit	32-Bit	64-Bit
S175	√	×	1	2	×	8G	64	128	64	64
S375	√	×	1	2	1	8G	64	128	64	64
S175E	×	×	1	2	×	8G	64	128	64	64
S375E	×	×	1	2	1	8G	64	128	64	64

**Notice**

1. Default version is GSM/GPRS module inside.
2. For 3G WCDMA, 4G LTE version, please tell our sales where would you like to use them.
3. The S175 ,375E series has no communication module, so it can't call and SMS to alarm.



## 1. Brief introduction

The Cellular IoT Gateway S17X/S37X is an industrial class, high reliability, high stability, and programmable Remote Terminal Unit (RTU). It embedded 32-Bit High Performance Microprocessor MCU, inbuilt industrial Cellular module. It provides 1 ambient sensor input for monitoring onsite temperature and humidity, and RS485 serial port, supports 128/224/320 mapping registers via Modbus RTU protocol. It can monitor and operate the I/O ports by SMS, APP, Web Server, internet, timers and programmed inter-lock events automatically.

The Cellular IoT Modbus Gateway S17X/S37X inbuilt TCP/IP protocol stack make it suitable for internet of things (IoT) applications, it can be easily to operate by the provided cloud, app, and web server, or integrated to you IoT applications according to Ethernet or the TCP/UDP protocol, or integrated to SCADA systems by standard Modbus TCP protocol, too. This is very useful if you need remote control onsite devices with low cost solution.

The Cellular IoT Modbus Gateway S17X/S37X is designed for working in the harsh industrial application environment, widely used in a variety of industrial automation, security monitoring system, automatically measurement and control system, BTS monitoring, remote data acquisition, telemetrically systems, automatically control system. It can be used as a remote RS485 data transmission, remote extend I/O tags control, remote smart PLC, meters monitoring.

The Cellular IoT Modbus Gateway S17X/S37X supports 2 RS485 transparent data transmission and performs as Modbus Slave over GPRS/3G/4G/NB-IoT network and RS485 serial port. Moreover, the Cellular IoT Modbus RTU can be used as Modbus RTU Master to reading smart meters, I/O modules, PLC, and converts to SMS alert once triggered the threshold value, or transmit data to remote server over GPRS/3G/4G/NB-IOT network. 2 RS485 ports can be used one as Modbus RTU Master and the other as Slave at the same time.

### Typically applications:

BTS Monitoring, Security Alarm System applications, Supervision and monitoring alarm systems, Automatic monitoring system, Vending Machines security protection, Pumping Stations, Tanks, Oil or Water levels, Buildings and Real Estate, Weather Stations, River Monitoring and Flood Control, Oil and gas pipelines, Corrosion protection, Temperatures, water leakage applications, Wellheads, boat, vehicle, Energy saving, street lights control system, Valve controls, Transformer stations, Unmanned machine rooms, Control room application, Automation System, M2M, etc.

## 2. Safety Directions



### Safe Startup

Do not use the unit when using GSM/3G/4G equipment is prohibited or might bring disturbance or danger.



### Interference

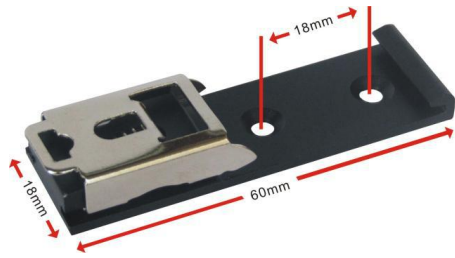
All wireless equipment might interfere network signals of the unit and influence its performance.

## 3. Standard Packing List

Gateway X1; AC/DC Adaptor X1; GSM/3G/4G Antenna X1; User Manual X1; PC Configurator X1 .

*Note: The package does not include any SIM card.*

**Optional:** 35mm Standard DIN rail fixed Bracket



## 4. Features and Specification

### 4.1 Mainly Features

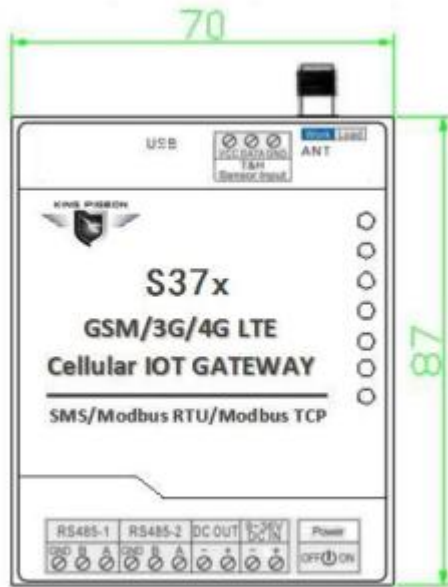
- GSM/GPRS/3G/4G network communication, can be operated from anywhere, no distance limitation;
- Wide range power supply 9~36VDC with over voltage and phase-reversal protection;
- Embedded ARM<sup>®</sup> Cortex<sup>™</sup> -M4 32 Bit RISC Core, 168 MHz inside, RTOS system, reliable performance with in-built watchdog;
- 1 temperature & humidity sensor input for monitoring onsite environment, the sensor model is AM2301, Measures temperatures from -40-80°C, 0.5°C accuracy, Relative Humidity from 0-99RH%, accuracy is 3%;
- Inbuilt 8G SD card to save up to tens of thousands historical data and events;
- Supports dual sim card, improve communication quality;
- 2 RS485 port supports Modbus RTU Master and Slave, can be used to extend I/O ports or meters;
- Powerful SMS function: Threshold high SMS alert, SMS set, SMS inquiry, SMS command for Modbus PLC..., and SMS monitoring communication with Slaves;
- Provides 1 channel 9~36VDC power source output for external device, saving wiring cost;
- Automatically resend the data while communication interrupt or failure, and failure will alert by SMS text to users;
- Supports remotely restart the RTU, and configure & operate it by SMS commands remotely;
- 10 SMS Alert and auto dial numbers for receiving alarm message, can program to receive specified alarm message;
- Inbuilt inter-lock logic programmer and powerful timer program function;
- Modular structure design, replace a module can upgrade the network from 2G to 3G/4G or 3G to 4G;
- Support SMS, dial, GPRS, 3G, 4G network for alert, USB port for configuration and upgrade firmware;
- Inbuilt large capacity automatically rechargeable backup battery, alert when external power failure;
- Support TCP/UDP, Modbus TCP, Modbus RTU over TCP, King Pigeon IoT RTU protocol and data transparent transmission function;
- Using metal shell, protection class IP30. Metal shell and system security isolation, especially suitable for industrial applications in the field;
- L70 \* W88 \* H30mm, compatible wall installation and DIN35mm industrial rail installation.

**4. 2 Specifications**

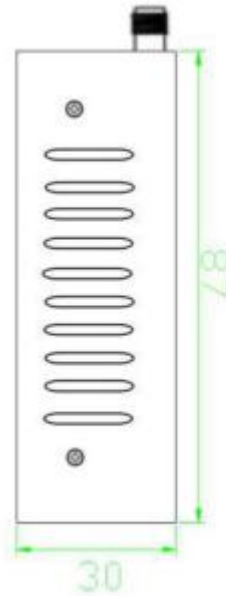
Item	Reference Scope
DC Power supply	Standard adapter: DC 12V/1.5A Range 9-36VDC
Power consumption	Standby:12V/70mA; Working Max.: 12V/300mA
GSM Frequency	850/900/1800/1900Mhz
3G/4G/NB-Iot	Optional: WCDMA/TDD-LTE/FDD-LTE/NB-IoT
TCP/IP stack	TCP,UDP
SIM interface	Dual SIM Card,supporting 3V and 1.8V SIM Card
External antenna	SMA Antenna interface, 50 Ohm, Gain: 3dB
Serial Interfaces	1 USB Port
Protocols	SMS, GPRS UDP,TCP, Modbus RTU, Modbus TCP and more equipment protocols can be added according to requirements.
Ethernet (only for S37X)	1 RJ45 Ethernet port for connect internet, WAN or LAN.
RS485	2 RS485, Support Transparent transmission and Modbus RTU Slave, Modbus RTU Master.
Temp.&Hum Inputs	Temperature range: -40°C to +80°C, Humidity Range: 0~100%RH;
Power Outputs	1 Port, for external device;
Mapping Registers	Max.320
Memory Capacity	Internal 8G SD card inside, can save the data for 100000events.
Backup Battery	3.7V 1200mAH
Temperature range	-10°C ~ +70 °C
Humidity range	Relative humidity 95% (condensation free)
Exterior dimension	70mm*88mm*30mm
Net Weight	500g

## 5. Physical Layout and Installation Diagram

### 5.1 Control Unit size



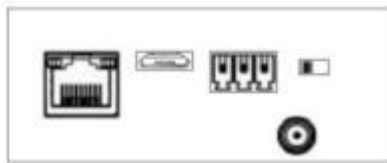
(Front view)



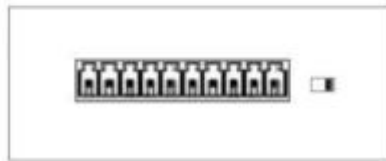
(Left side view)



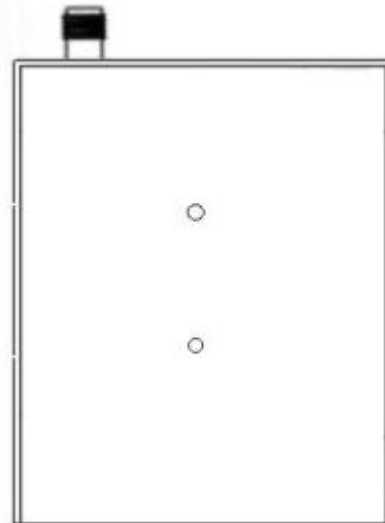
(Right side view)



Top view

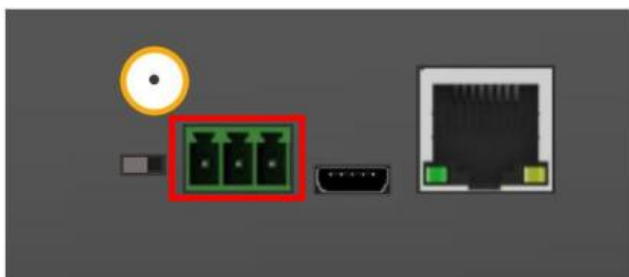


(Upward view)



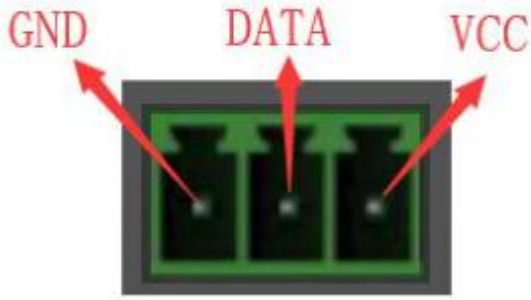
(Bottom view)

### 5.2 Interface definition



Function	Description
----------	-------------



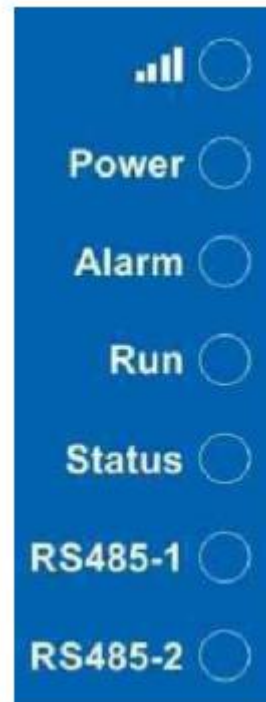


RS485-1	RS485-2	DC OUT	9~36V DC IN
GND B A	GND B A	- +	- +


GND	GND for input
DATA	Temperature/humidity data
VCC	Power for Tem/Hum sensor AM230x

Function	Description
GND	GND
485_1 B	485_1 B data
485_1 A	485_1 A data
GND	GND
485_2 B	485_2 B data
485_2 A	485_2 A data
DC_OUT -	Power source output port, Negative electrode.
DC_OUT +	Power source output port, positive electrode.
DC_IN -	External DC Power input port, negative electrode.
DC_IN +	External DC Power input port, positive electrode.

### 5.3 LED Indicator Definition



**LED Indicator Definition**

	Cellular network indicator. When 2G register network, off 2 seconds, on 0.2s and so on; When 3G 4G register network, on 2s, off 0.2s...Flicks quickly means data transmission.
<b>Power</b>	RTU status indicator, LED ON when switched RTU on..
<b>Alarm</b>	Alarm Indicator, alarm will ON and flick. Normally is OFF;
<b>Arm</b>	Arm/Disarmed Indicator, Arm is ON, disarmed is OFF.
<b>Run</b>	RTU running status indicator, ON or OFF stands for RTU halted, flicks slowly stands for RTU running.
<b>RS485-1</b>	When transmitting data by RS485, the LED will flick, otherwise, it is off.
<b>RS485-2</b>	When transmitting data by RS485, the LED will flick, otherwise, it is off.

**Switch & Button Definition**

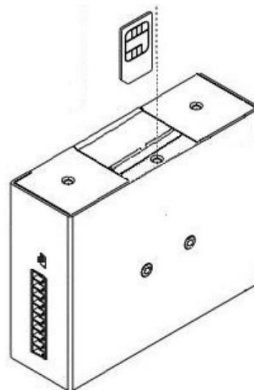


**Power Switch** For switch ON or OFF the RTU



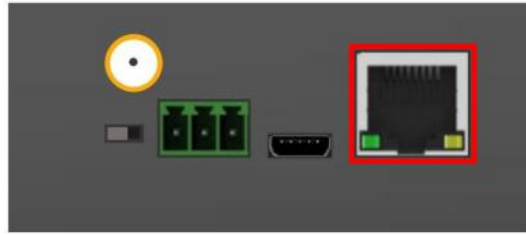
**Upgrade Firmware Switch** For upgrade firmware purpose only. Only when upgrade new firmware version will use it, otherwise keep it at Work Side all the time.

**SIM Card Slot**



**SIM Card Slot** For SIM Card Installation, only supports 1.8V/3V SIM Card

**Ethernet Connector Definition(only for S37X)**



Ethernet (Only for S37x)	Rate indicator: ON stands for 100Mbps;OFF stands for 10Mbps Link indicator:ON means connecting;OFF means disconnect;flick means transmitting data
<b>ATN Port Connector Definition</b>	
<b>ATN</b>	GSM/3G/4G Antenna connector, 50Ohm, SMA male.
<b>USB Port Connector Definition</b>	
<b>USB</b>	USB port, for configuration and upgrading firmware and exporting historical data;

## 6. Settings&Operation

The Cellular IoT Gateway is user-friendly design. The user can setup it or export historic data by the PC Configuration through USB cable, and upgrade firmware by USB port. The Cellular IoT Gateway also can be configured some basically parameters by SMS Commands, please refer to **SMS Command List**.

### Tips!

- 1) Please insert the SIM Card firstly, and install the GSM/3G/4G Antenna, please power on to check the LEDs status according to above mentioned LED Definitions, keep switch on it during the programming.
- 2) The PC Configuration in the CD, please click it to run. Also can download from [www.GPRS-M2M.com](http://www.GPRS-M2M.com) under S17X/S37X page directly.

Below is the steps to setup the parameters by PC Configuration, please follow it step by step.

### 6.1 Start to Configure

#### Step1: Install the Configuration software

The Configuration software in the CD or download from [www.GPRS-M2M.com](http://www.GPRS-M2M.com), then installs it on the computer.

#### Step2: Connection

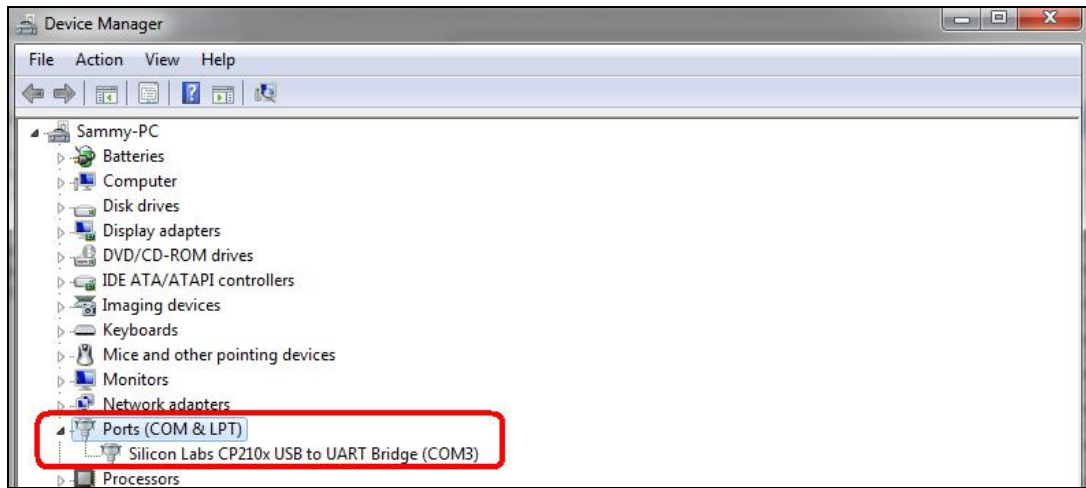
Please insert the SIM Card, and install the GSM/3G/4G Antenna.

#### Step3: Connect the Gateway to the PC by USB Cable. And connect the external DC Power to DC Power Ports, Power on, and switch on the device, see below:



#### Step4: Install USB Driver

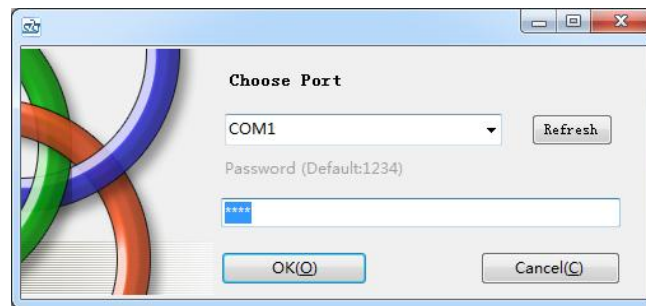
Install the USB Driver to the computer from the CD firstly. When successful, it can be found out at the device manager of the XP or Windows 7 or Win8/Win10, please see the below photo. Also, the driver for different OS can be downloaded from Silicon Laboratories, Inc. <http://www.silabs.com>, the model is CP210x.



**Step5:** Run the Configuration software (Compatible with Windows XP/7/8/10)

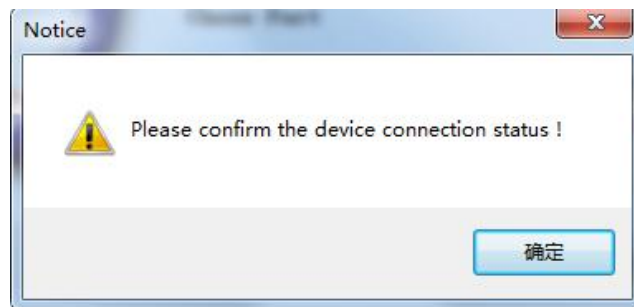
**Tips:** In some computer, it required download net framework 4.0 while installation, then please click "Yes" to go to Microsoft website to download this service pack.

Please click S17X/S37X configuration software to run it. Enter the password, default is 1234. Then you can enter the configuration page as below:



**Notice:**

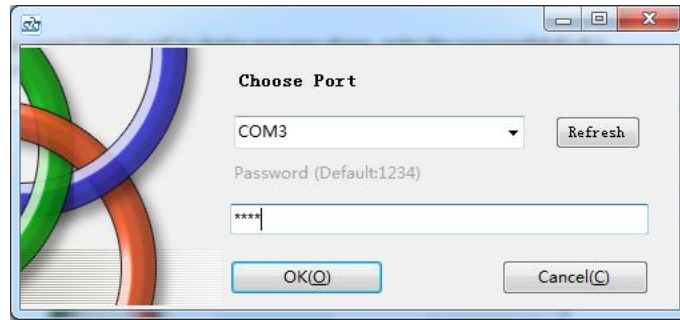
If display the below windows, then means the RTU connect to the PC failure. The reasons are below:



- 1) USB Driver installation failure;
- 2) USB Cable connection is disconnected;
- 3) The Upgrade Firmware Switch at Load side, not at Work side.
- 4) Power Switch switched off or DC Power Connection is disconnected.

**Step6:** Choose the correct "COM port" in device manager above, enter the password(default is 1234),click the "OK" to connect and start to program

Details please check the picture as below:



**Tips:** If not connect successfully, will not enter into next step. Pls check if USB connect well, or COM port and password correct or not.

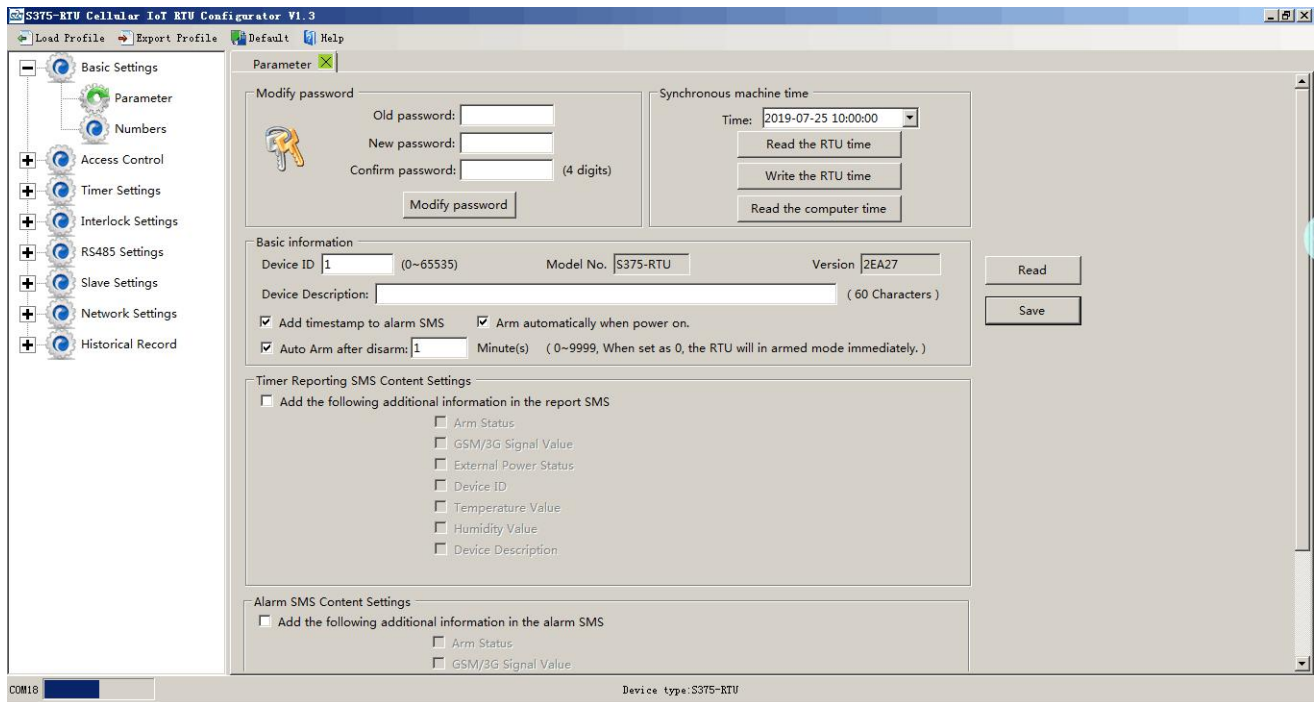
## 6.2 Setting Self-checking

Phenomenon	Possible Reason
Can't enter software	<ol style="list-style-type: none"> <li>1. USB Driver installation failure;</li> <li>2. COM Port not correct or USB driver installation failure;</li> </ol>
After switching panel on, only Power light on, panel can't work	<ol style="list-style-type: none"> <li>1. The Upgrade Firmware Switch at Load side, not at Work side. Solution: Switch the power switch to OFF----&gt;Upgrade switch to Work side----&gt; Power switch to ON;</li> <li>2. SD card fall out from the slot. Solution: Shake panel to listen if there is voice or not;</li> <li>3. In upgrade mode, use upgrade tool erased the firmware.</li> </ol>
Can't enter into working mode	<ol style="list-style-type: none"> <li>1. The Upgrade Firmware Switch at Load side, not at Work side. Solution: Switch the power switch to OFF----&gt;Upgrade switch to Work side----&gt; Power switch to ON;</li> </ol>
Can't find COM Port	<ol style="list-style-type: none"> <li>1. Have not installed driver;</li> <li>2. PC system problem cause driver installation failure, can't support Apple OS system.</li> <li>3. Check USB line, and try other common driver software such as "Drive TheLife".</li> </ol>
In working mode, the device not response the Modbus command	Have not set the device ID. Solution: In setting mode, set device ID---->Switch the device to Run mode.
After switching panel on, not running according to parameter setting	<p>After parameter setting, forget to click "Save" button in the menu.</p> <p>Solution: Back to Set mode----&gt;Click "Save" Button after setting one page in the menu.</p>

### Terms usually used by Cellular IoT Gateway

Cellular IoT RTU, Modbus RTU, Modbus Master, Modbus Slave, Modbus RTU Over TCP, Modbus TCP, Arm, Disarm...

## 6.3 Configuration software interface and running



**Load Profile:** Click it to load additional Profile to the PC Configurator;

**Export Profile:** Click it to save the present configuration parameters as a profile for next RTU configuration or backup the parameter settings.

**Tips:** *The Load Profile and Export Profile is very useful while you need to program bulks of RTU with similar parameters. After programmed the first unit then you can export profile to save it, for the second RTU then you can load profile directly to save you time.*

**Default:** Click it to recovery the parameters to factory defaults.

**Notice:**

1. *After setting or revising parameter, need to click the "Save" button of this page for saving parameters in device*
2. *Before export profile, need to read Slaves configuration details first, to avoid Slaves information missing.*
3. *Easy way to revise parameter: Open parameter setting page---->Click "Read" button to get device current value ---->Revise and click "Save" button in the menu.*
4. *Reboot the device, switch the Power Switch to OFF, then switch it to ON, the device will enter into normal running mode after that.*



## Basic Settings

**Reminder:** *Please click the "Read" for previous parameter before starting to set.*



**Modify Password:** This is for modifying the RTU's Password, default is 1234.

**Synchronous machine time:** This is to setup the RTU's time for daily report or other timers. After click **Write the RTU Time**, the RTU will be synchronous the same time as the PC. If connect to King Pigeon Cloud Server, no need this step.

**Device ID:** Non-necessary. This is mainly for monitoring center to identify the RTU; If communicate via Modbus protocol, device ID only can be 1~247.

**Device Description:** This is the description of the RTU, e.g.: installation address, usage instructions and so on.

**Add Timestamp to alarm SMS:** Tick it stands for while alarm occurrence, the Alarm SMS will include the RTU'S current time information at the SMS Content.

**Arm automatically when power on:** Tick it stands for once the RTU powered up, the RTU will enter into Arm Mode automatically.

**Auto Arm after disarmed:** Fill the timeout to enter into Armed Mode automatically after disarmed operation. This is useful for security protection applications.

**Tips:**

**Arm:** Under this mode, any alarm occurrence will send SMS and dial the authorized numbers immediately. And execute the programmed I/O outputs.

**Disarmed:** Under this mode, alarm occurrence will not send SMS & dial the authorized numbers.

**Timer Reporting SMS Content Settings:** Tick the related items to add its value/status to the Timer report SMS contents.

**Alarm SMS Content Settings:** Ticks the related items to add its value/status to the Alarm SMS Contents.

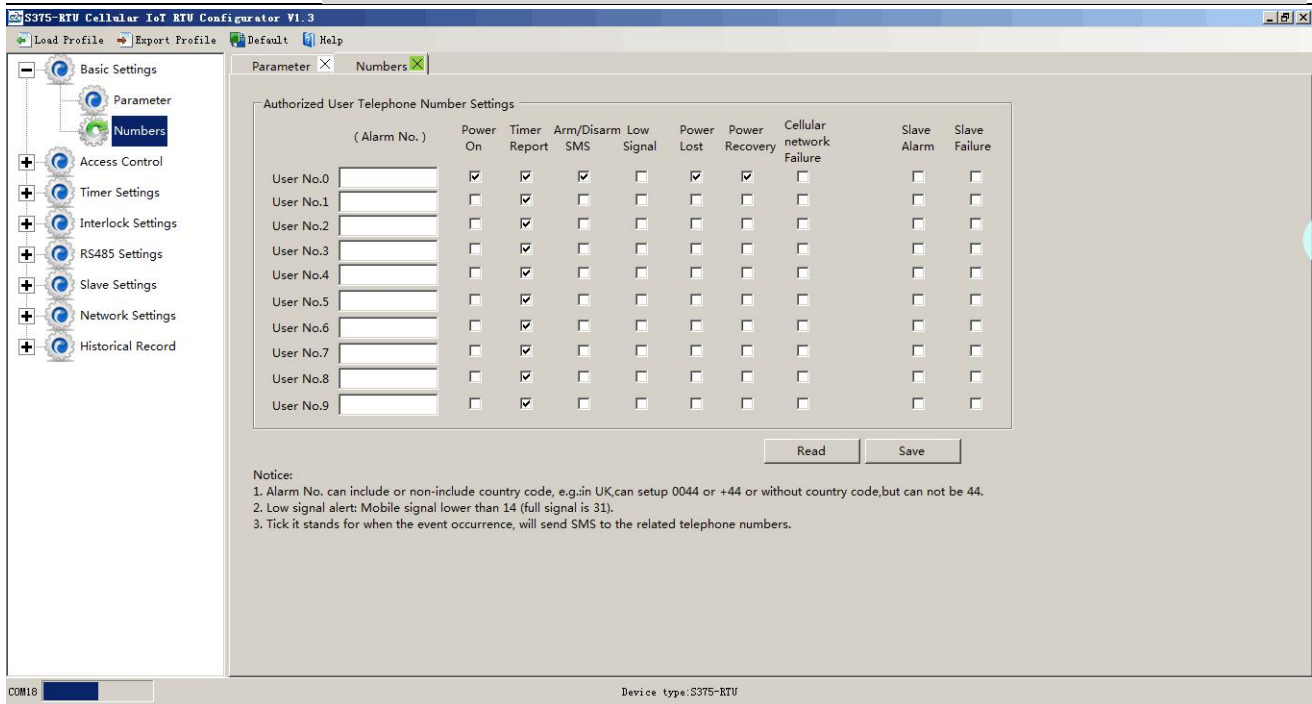


## Number Settings

This is to setup the Authorized User Telephone Numbers to receive the Alarm SMS or dial. Tick it stands for while the related event alarm occurrence will send SMS to this number.

**Reminder:**

Please remember that click "Save" button to save it after parameter be written, below pages are the same.



**Power On:** Tick it stands for while the RTU powered up, will automatically send SMS to this number, include device model, version, description, IMEI, status, signal value etc....

**Timer Report:** Tick it stands for Timer report SMS will send to this number.

**Arm/Disarm SMS:** Tick it stands for Arm or Disarm the RTU, will send SMS to this number.

**Low Signal:** Tick it stands for while GSM/3G/4G Network signal strength lower than 14 will send SMS to this number.

**Power Lost:** Tick it stands for while external DC Power loss will send SMS to this number.

**Power Recovery:** Tick it stands for while external DC Power recovery, will send SMS to this number.

**Cellular network Failure:** Tick it stands for while GPRS connection re-try 3 times and still failure will send SMS to this number.

**Relay Switch:** Tick it stands for while relay is activated, will send SMS to this number.

**Slave Alarm:** Tick it stands for the slave tag triggered will send SMS to this number.

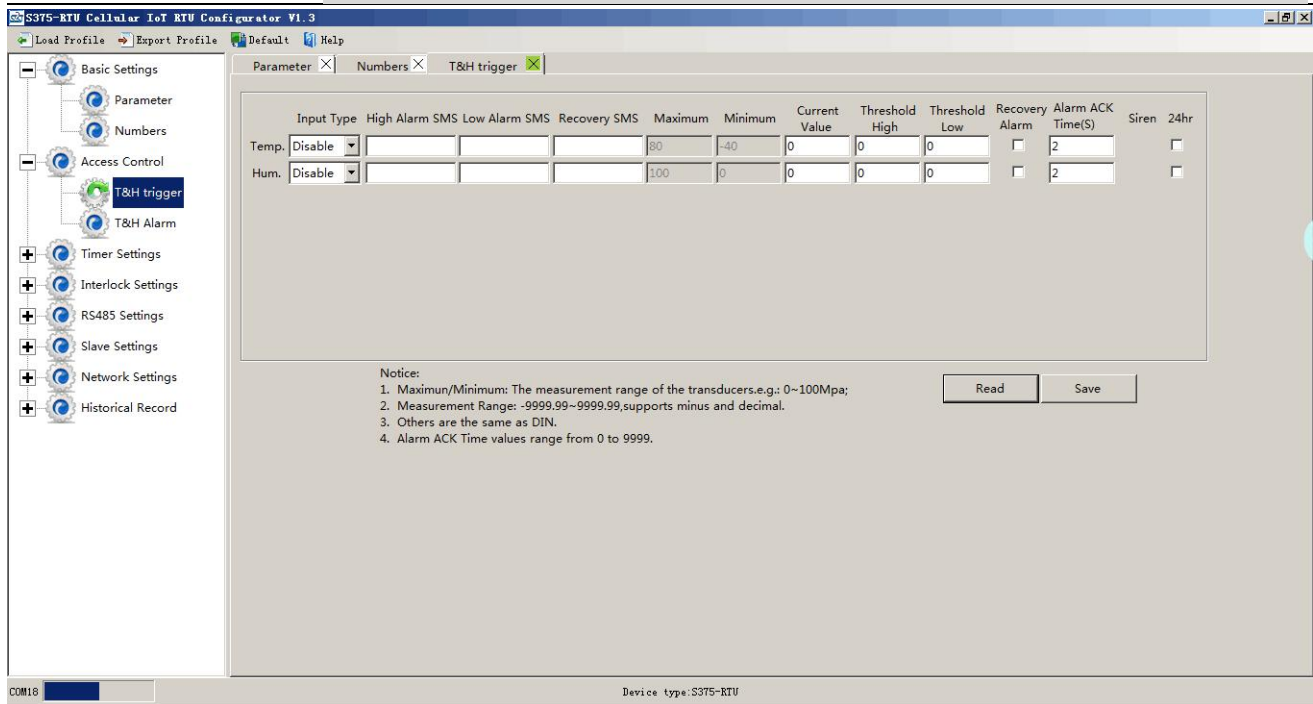
**Slave Failure:** Tick it stands for when slave communication failure alarm verify time arrive, will send SMS to this number.

## T&H Trigger Settings

This page is to setup the temperature/humidity input alarm conditions parameter. It can be used for monitoring temperature and humidity, can be connected as below:







## Reminder:

*T&H input default Disable.*

**Input Type:** The user can choose the input type for related channel. Includes: Enable and Disabled.

- 1) Disabled: Not use this channel.
- 2) Enable: Use this channel.

**High Alarm SMS:** Under Arm or 24h status, once current value higher than threshold high value will send this SMS content to authorized numbers.

**Low Alarm SMS:** Under Arm or 24h status, once current value lower than threshold low value will send this SMS content to authorized numbers.

**Recovery SMS:** Under Arm or 24h status, if tick the "Recovery Alarm", when current value recovery normal will send this SMS content to authorize number.

**Maximum:** The transducer's maximum measure range. E.g.:100 Celsius degree. Usually it can be found out at the transducer's specification.

**Minimum:** The transducer's minimum measure range. E.g : -50 Celsius degree. Usually it can be found out at the transducer's specification.

**Current Value:** Stands for input's current value of the transducers.

**Threshold High:** The high value(reached) need to alarm; Example: set 50Celsius degree to alert.

**Threshold Low:** The low value(reached) need to alarm; Example: set -30Celsius degree to alert.

**Recovery Alarm:** Tick it stands for when the T&H input recovery, will send SMS to the authorized numbers.

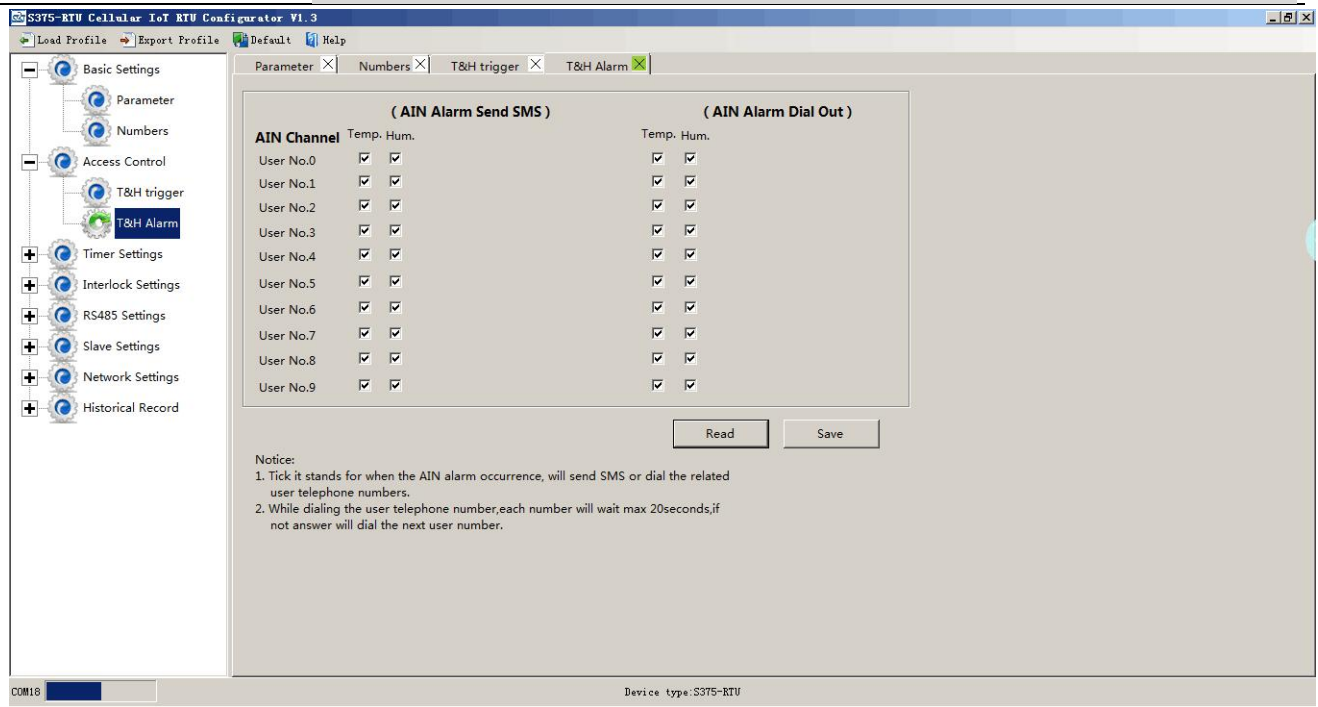
**Alarm ACK Time:** Stands for when the temperature/humidity value high or low lasted time more than this value, will be treated as a true alarm, if less than this value, then will not alarm.

**24Hr:** Tick it stands for no matter the RTU is in Arm or Disarmed mode, this input triggered will alarm.



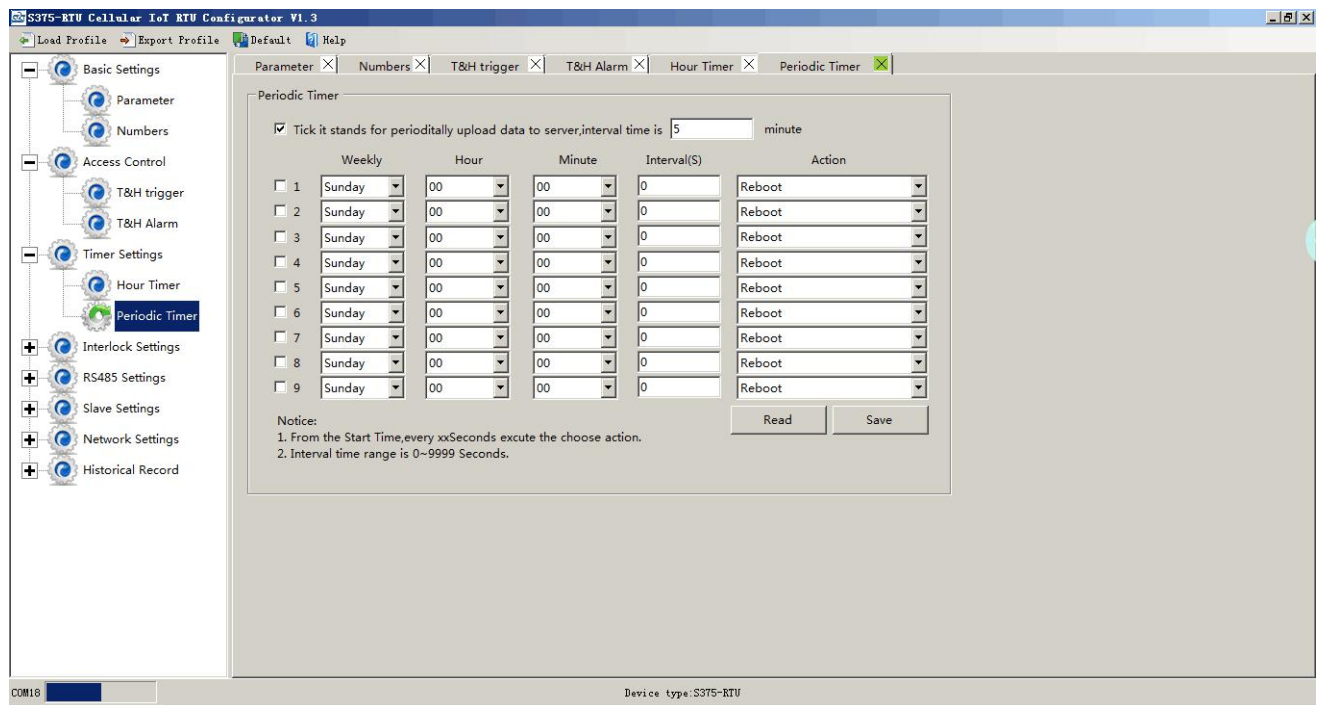
## T&H Alarm Settings

This page is for setup while temperature/humidity alarm, send SMS & Dial to which authorized numbers. Tick it stands for enable to send SMS or dial the related authorized number.



## Timer Settings

This page is for setup hour timer and periodically timer, it is useful for scheduling when to execute what action automatically or it with repeat this action according to the interval time. Tick it stands for enable this timer event:



### Reminder:

When GPRS/3G/4G data transmission protocol is King Pigeon IoT RTU Protocol, the periodically auto upload default enable and upload every 5 minutes.

Tick stands for enable this timer function, otherwise is disable.

**Weekly+Hour+Minute:** Stands for what day and at what time does the RTU should start to



execute the action and interval how many seconds then repeat to execute the action.

**Interval:** Stands for interval how many seconds does the RTU should repeat to execute the action.

If setup it as 0, then this event will not be repeated.

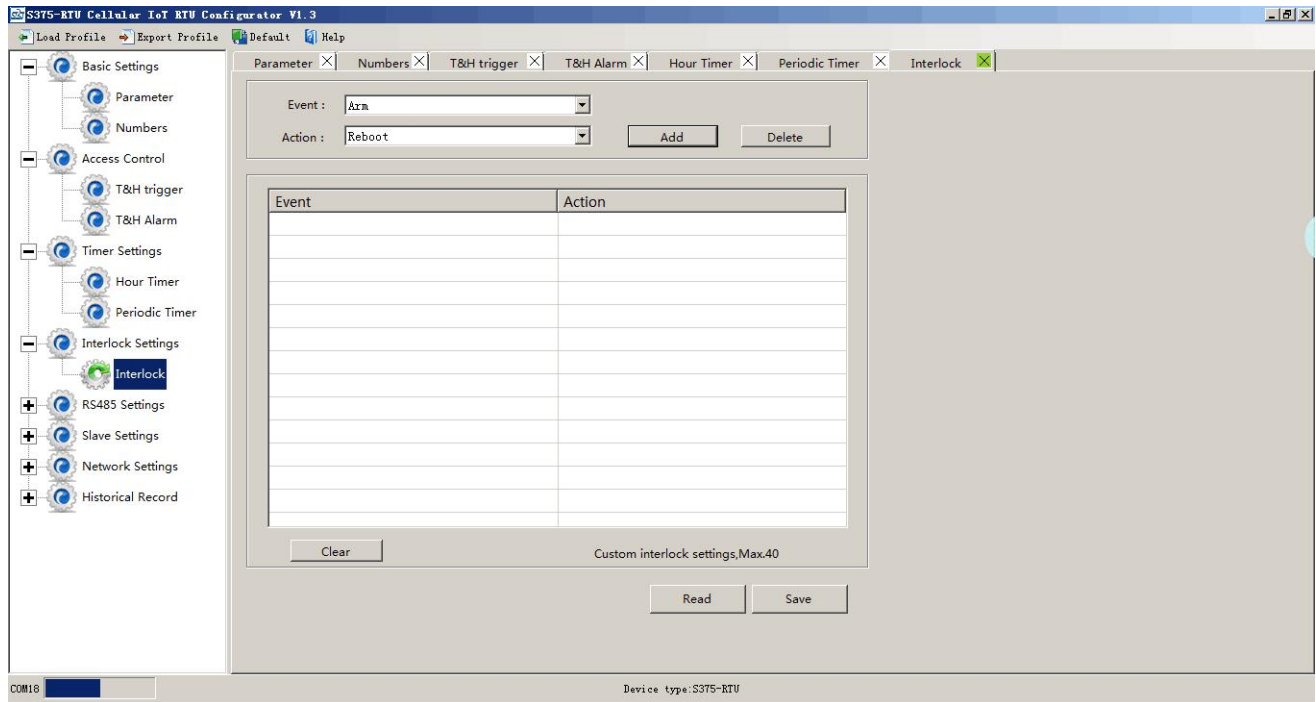
**Action:** Stands for what action does the RTU should to execute at the specified time.

**Question:** Have set the timer SMS report, but finally not get the SMS.

**Solution:** Have no ticked the "Timer Reporting SMS Content" in first Basic Parameter Settings page.

## Interlock Settings

This page is for setup if what happen, then what action does the RTU should execute, it is a programmable logic events. Total can program up to 40 logic events for automatically control purposes.

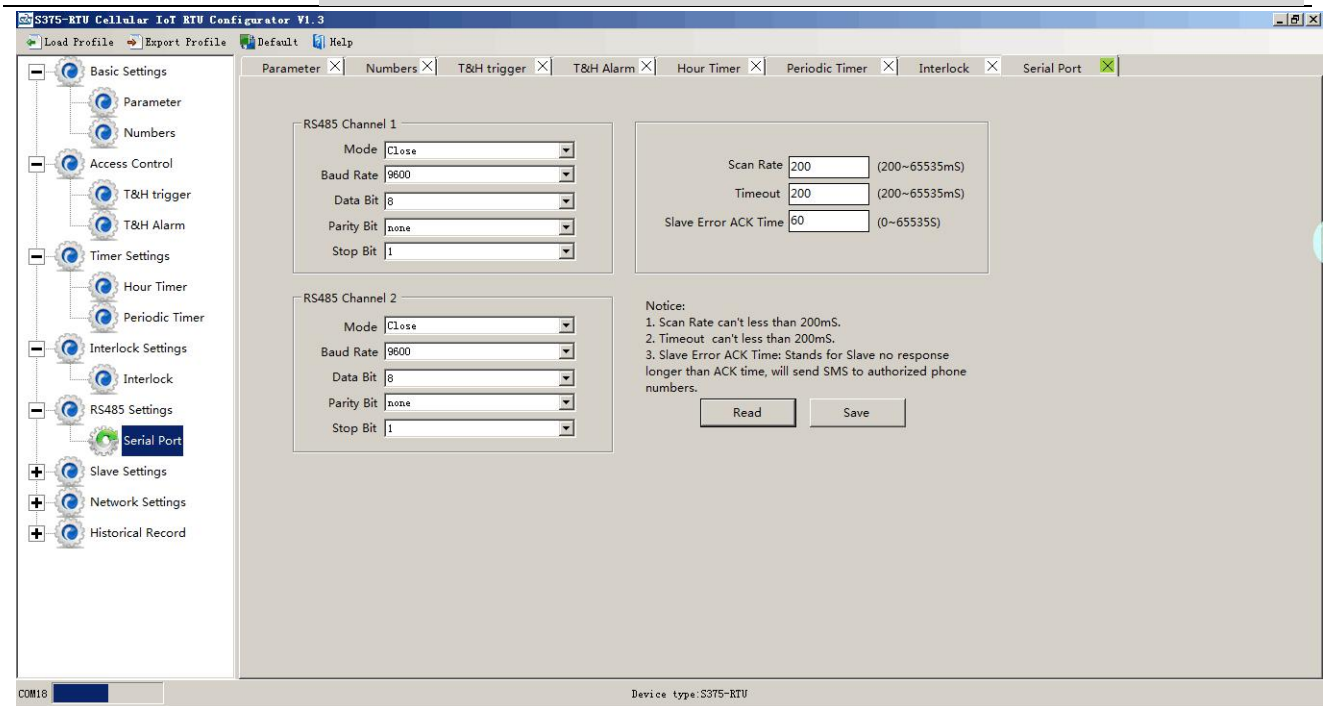


**Event:** Stands for if this occurrence.

**Action:** Stands for then what action does the RTU should execute.

## RS485 Serial Port Settings

This page is for setup the serial port parameters. Over the RS485, the S275 RTU can be used as Modbus RTU Slave, Modbus RTU Master and transparent transmission.



**Modbus RTU Master:** Stands for the RS485 used for Modbus RTU Master.

**Modbus RTU Slave:** Stands for the RS485 used for Modbus RTU Slave, and the "Scan rate", "Time out" and "Slave failure verify time " of Master function will be disable.

**Baud Rate:** 1200/2400/4800/9600/19200/38400/57600/115200 optional.

**Data Bit:** 8 bit.

**Parity Bit:** None, Even and Odd optional.

**Stop Bit:** 1 or 2 stop bit optional.

**Scan rate:** When RS485 used as Master, the interval time between two polling command.

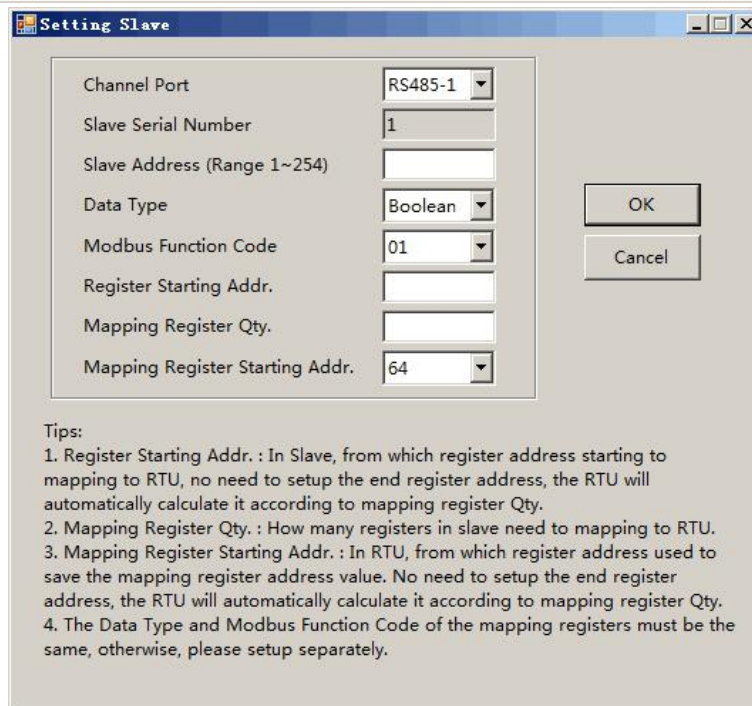
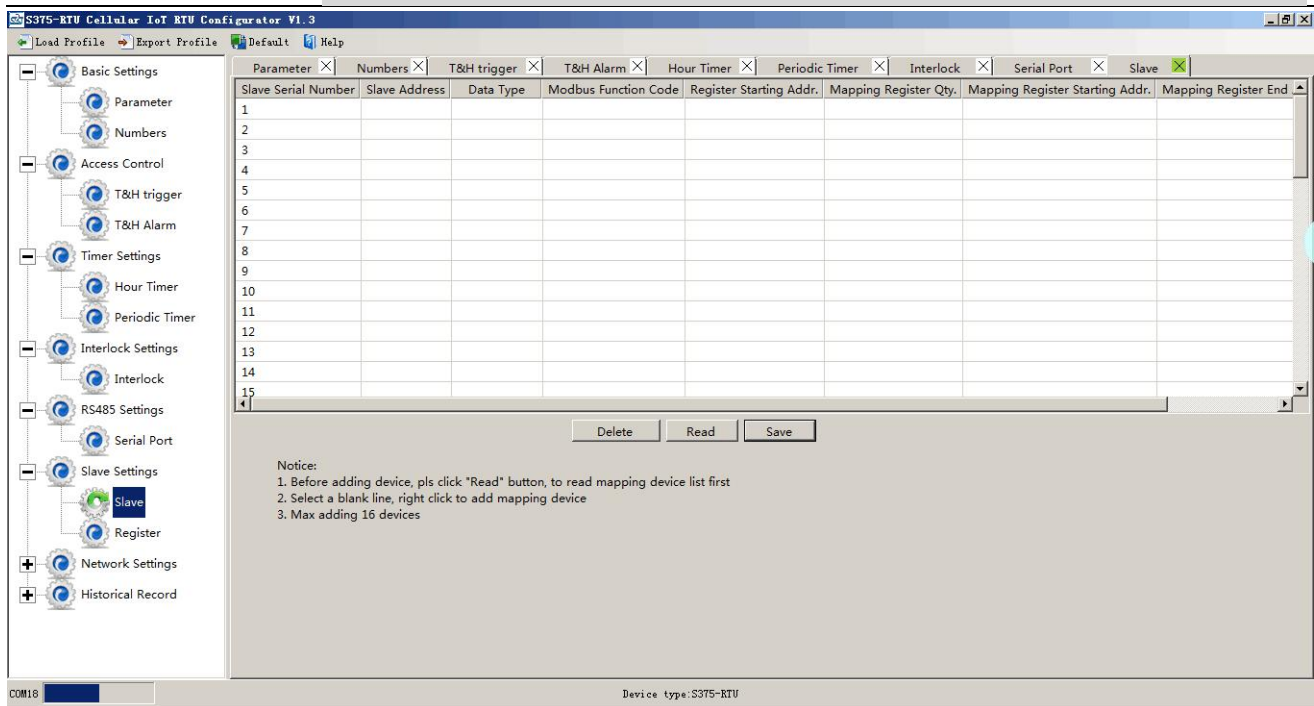
**Time out:** When RS485 used as Master, after sending command to slave, the longest time waiting for slave data back. If longer than this setting value, will ensure slave no response.

**Slave Error ACK time:** When RS485 used as Master, if no response time between Master and Slave longer than this value, will send SMS to authorize number.

## Slave Settings

This page is for adding, revising and deleting the slaves. When used as Modbus RTU master, slave data mapping can be added to local register. When editing a slave, just select a row and right click to complete the delete, add, modify parameters, etc.

**Note:**When adding a slave, first read the list of slaves that have been mapped to prevent the new slave covering the added slave device.



**Channel Port:** The communication way of the slave connected to the RTU.

**Slave Serial Number:** The number of slave.

**Slave Address:** Stands for the Modbus RTU Slave ID.

**Data Type:** Stand for "Boolean", "16 Bit", "32 Bit", "64 Bit".

**Modbus Function Code:** Stand for Modbus RTU protocol function code, command for slave reading and writing.

**Register Starting Address:** The starting register address for slave data reading and writing.

**Mapping Register Qty:** How many data quantity need to read, used for mapping to device register address.

**Mapping Address Starting Address:** Stand for mapping the slave start register data to the device start mapping address.

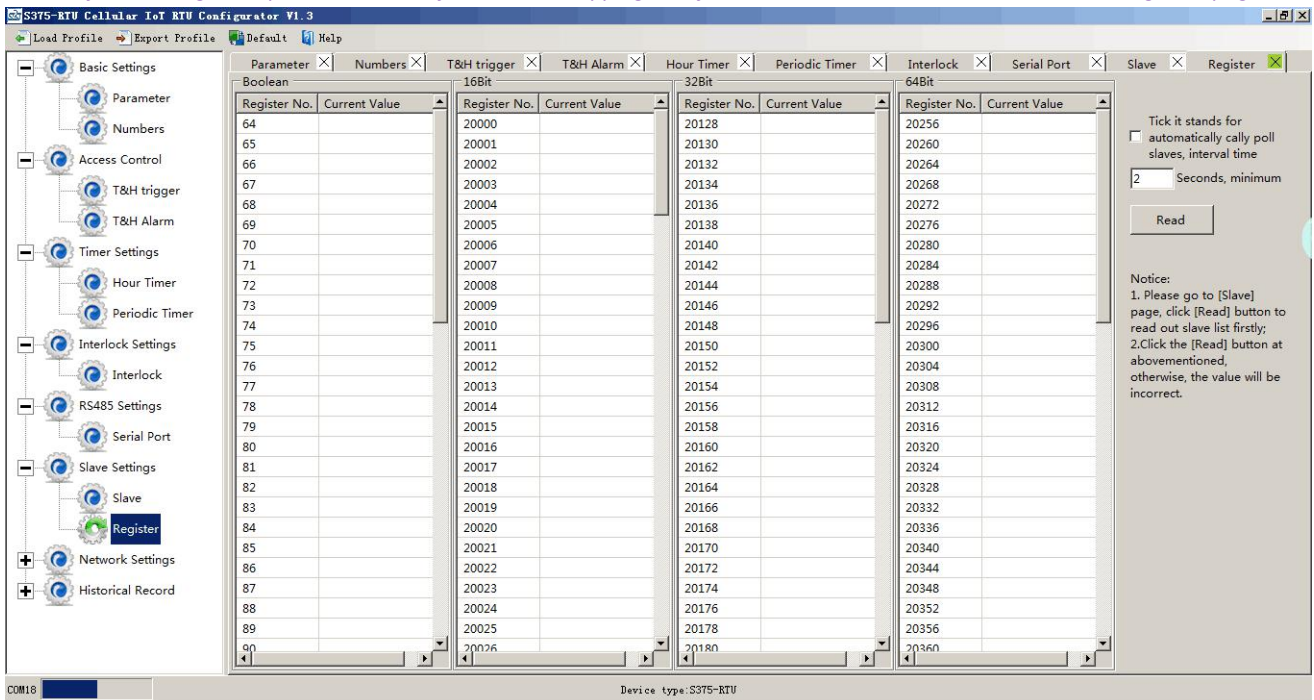
**Mapping Address End Address:** Calculate the end mapping address according to start address and reading data quantity.

## Register Settings

Click "Register" page to real time check slave current value.

### Reminder:

*Before reading data, pls read slave list from "Slave Mapping List" first, then can check slave current value in "Register" page:*



The screenshot shows the 'Register' tab selected in the software. The main window contains four data tables for different register types: Boolean, 16Bit, 32Bit, and 64Bit. Each table has two columns: 'Register No.' and 'Current Value'. The 16Bit, 32Bit, and 64Bit tables are populated with register numbers ranging from 20000 to 20360. On the right side, there is a 'Read' button and a 'Notice' section with instructions: '1. Please go to [Slave] page, click [Read] button to read out slave list firstly; 2. Click the [Read] button at abovementioned, otherwise, the value will be incorrect.'

## Cellular network Settings

This page used for setting device parameters connect to networks. The device can be compatible with many third party upper computer system. And it can communicate with monitoring software or cloud via GPRS/3G/4G as below:

- 1) Modbus RTU Protocol, means Modbus RTU over TCP, communication with upper computer system. For example, connect to www.my-m2m.com cloud server. Domain: modbus.dtuip.com, Port: 6651.
- 2) Modbus TCP Protocol, communication with upper computer system. For example, connect to www.my-m2m.com cloud server. Domain: modbus.dtuip.com, Port: 6655.
- 3) King Pigeon IoT RTU protocol, transmit the King Pigeon IoT protocol on TCP, to communication with upper computer system. Advantage is when device abnormal, can send data to upper system automatically, not waiting for polling, then answer. For example, connect to www.rtu-m2m.com cloud server.



The screenshot shows the configuration interface for the S375-RTU Cellular IoT Gateway. The 'Cellular network' tab is active, displaying various settings for cellular connectivity and MQTT communication. The interface includes a sidebar with navigation options like Basic Settings, Parameter, Numbers, Access Control, and Network Settings. The main area contains three sections: Cellular Network Settings, Message Settings, and MQTT Settings, each with specific input fields and dropdown menus.

**Cellular Data:** "Disable", "Modbus RTU protocol", "IoT RTU protocol" or "Modbus TCP protocol" optional.

**Protocol:** TCP or UDP optional.

**Access Point Name:** APN, GSM operator provide.

**APN User Name:** User Name, GSM operator provide.

**APN Password:** Network password, GSM operator provide.

**Sever 1/2 IP/DNS:** Server IP address or DNS.

**Server Port:** Stands for the server's port.

**Server Strategy:** Only support "Prefer server 1" function, no "Both connection" now. When server 1 disconnect, will connect to server 2 automatically.

**Server Response Timeout, Device Re-connect Interval Time:** Connecting server fail 3 times, then the interval time of next time reconnecting.

**Login Packets:** Server register handshake protocol package. When transparent transmission or Modbus protocol, this item used for device ID, provided by cloud. Contact King Pigeon sales if need to connect [www.my-m2m.com](http://www.my-m2m.com) cloud server.

**Login ACK Packets:** Once set, device need response within 10 seconds after device send login message, otherwise it will continue send login message according to "Reconnection Times", still not response will offline once time, then try to reconnect, according to "Server Offline 3 Times, Device Reconnection Time".

**Logout Packets:** Once server send to device, device will be offline.

**Heartbeat Packets:** Heartbeat content to avoid network offline.

**Heartbeat ACK Packets:** Once set, device need response within 6 seconds after device send heartbeat message, otherwise it will continue send login message according to "Reconnection Times", still not response will offline once time, then try to reconnect, according to "Server Offline 3 Times, Device Reconnection Time".

**Heartbeat Interval:** Network keep online heartbeat interval time.

**Retransmission Times:** After setting heartbeat and login message, if server no response, the times which server will send data.

**Login Message Strategy:** "Send Once When Login Server", "Plus It In Front Of Every Packet", "Both Of Them" optional. "Plus It In Front Of Every Packet" when data transmission.

## MQTT Setting

**MQTT Client ID:** The device ID that needs to be subscribed after connecting to the cloud platform through the MQTT protocol. After subscribed, the device will send message to the users. this item used for device ID,

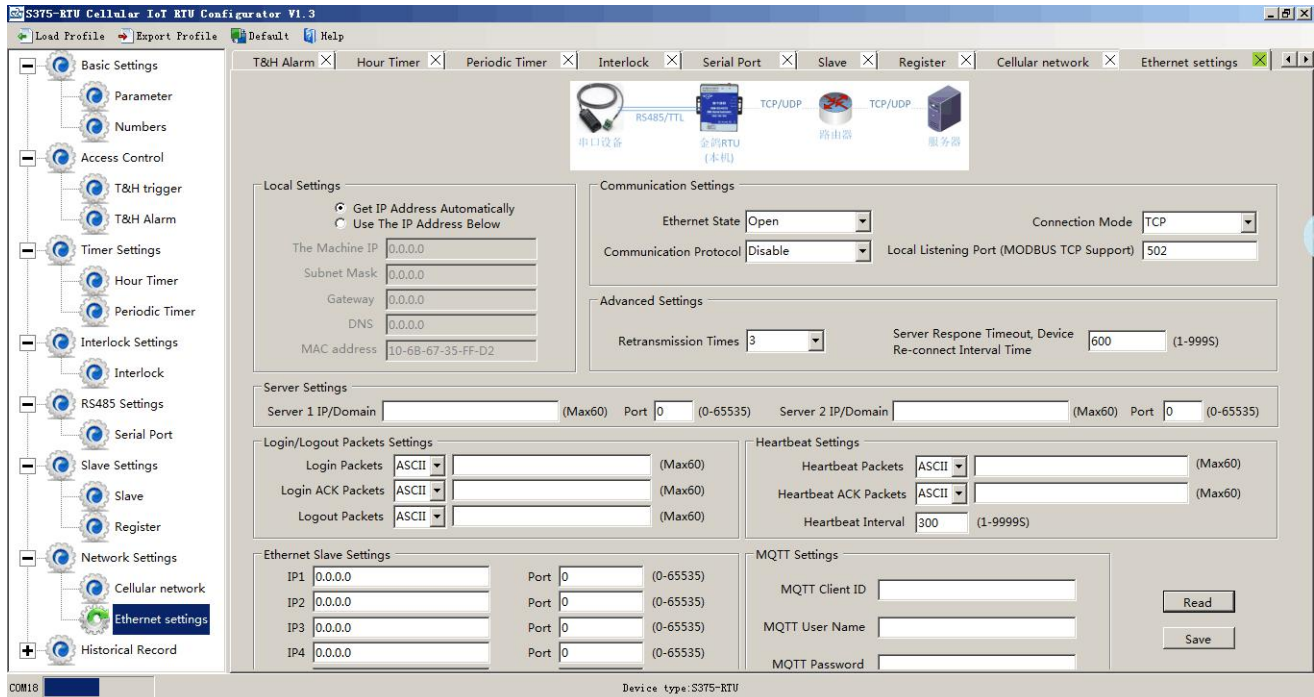


provided by cloud. Contact King Pigeon sales if need to connect www.my-m2m.com cloud server.

**MQTT User Name:** User accounts using the MQTT protocol.

**MQTT Password:** User password using the MQTT protocol.

## Ethernet Settings(only for S37X)



**Get IP address automatically:** Tick it stands for: the device automatically obtains the IP address in the LAN. Only when the router in the LAN allows the dynamic allocation of IP addresses can be used.

**Use the IP address below:** Tick it stands for the user setup a fixed IP address for the module.

### 01) Communication Settings

**Ethernet State:** Open or Close optional.

**Communication protocol:** "Disable", "Modbus RTU protocol", "IoT RTU protocol" or "Modbus TCP protocol" optional.

**Connection mode:** TCP or UDP optional.

**Local listening port:** it can be used to set a port number to listen to the visitor's data, the default is 502.

### 02) Advanced Settings

**Retransmission times:** After setting heartbeat and login message, if server no response, the times which server will send data.

**Server Response Timeout, Device Re-connect Interval Time:** Connecting server fail 3 times, then the interval time of next time reconnecting.

### 03) Server Setting

**Server IP/DNS:** Server IP address or DNS.

**Port:** Stands for the server's port.

#### Reminder:

*Server 1 is primary server, server 2 is backup server; connecting server 1 first; if it is not successfully in 50 seconds, will connect to server 2 automatically; can't "both connect" at a time.*

### 04) Login/Logout Packets Settings





**Login Packets:** Server register handshake protocol package. When transparent transmission or Modbus protocol, this item used for device ID, provided by cloud. Contact King Pigeon sales if need to connect [www.my-m2m.com](http://www.my-m2m.com) cloud server.

**Login ACK Packets:** Once set, device need response within 10 seconds after device send login message, otherwise it will continue send login message according to "Reconnection Times", still not response will offline once time, then try to reconnect, according to "Server Offline 3 Times, Device Reconnection Time".

**Logout Packets:** Once server send to device, device will be offline.

## 05) Heartbeat Setting

**Heartbeat packets:** Heartbeat content to avoid network offline.

**Heartbeat ACK Packets:** Once set, device need response within 6 seconds after device send heartbeat message, otherwise it will continue send login message according to "Reconnection Times", still not response will offline once time, then try to reconnect, according to "Server Offline 3 Times, Device Reconnection Time".

**Heartbeat Interval:** Network keep online heartbeat interval time.

## 06) Ethernet Slave Settings

**IP:** The IP address of Modbus slave.

**Port:** The port of Modbus slave.

## 07) MQTT Setting

**MQTT Client ID:** The device ID that needs to be subscribed after connecting to the cloud platform through the MQTT protocol. After subscribed, the device will send message to the users. this item used for device ID, provided by cloud. Contact King Pigeon sales if need to connect [www.my-m2m.com](http://www.my-m2m.com) cloud server.

**MQTT User Name:** User accounts using the MQTT protocol.

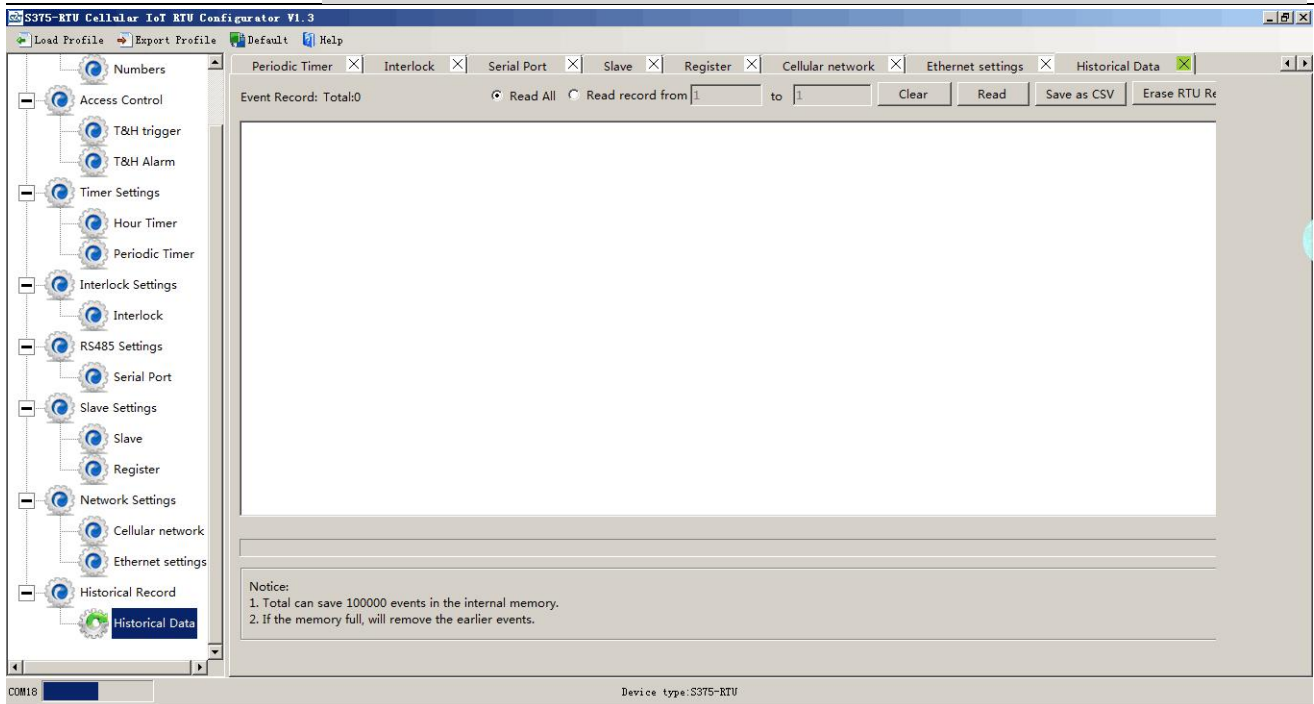
**MQTT Password:** User password using the MQTT protocol.



## Historical Record

The device inbuilt 8G SD card, store alarm and historical records. For saving historical records, need to set the saving historical records interval time in "Periodically Timer" page.

For historical record, once it full, will automatically remove the earlier records for new records. And can save as CS format for other purpose usage.



**Total:** Display device current historical records qty, "Read All" or "Read Record from xx to xx" optional.

**Clear:** Clear the screen.

**Read:** Read historical records.

**Save as CSV:** Historical records export as CSV file.

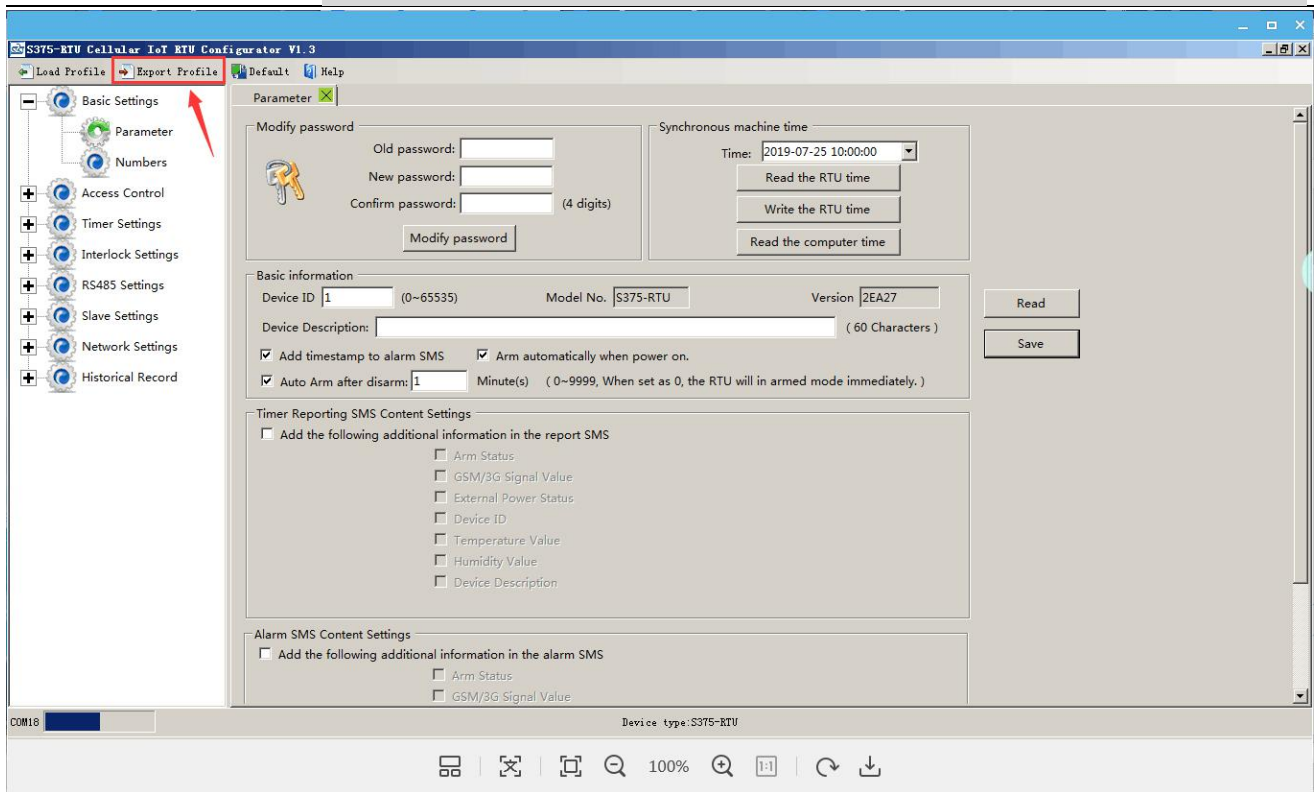
**Erase RTU Records:** Click this button will erase all device historical records, be careful.

## 7.Configuration and Reset

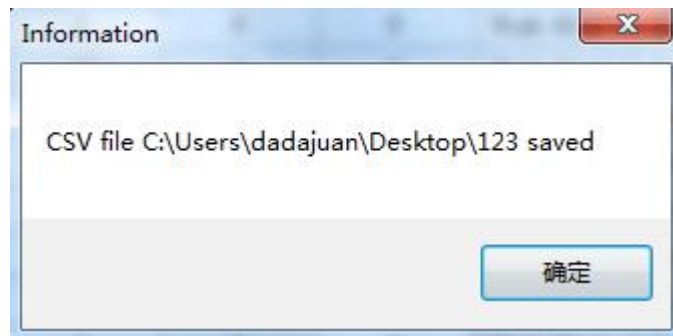
The Load Profile and Export Profile is very useful while you need to program bulks of RTU with similar parameters. The "Default" function can reset the device to factory default status.

### 7.1 Export Profile

Click "Export Profile" button----> *chose the path and input the name to save.*

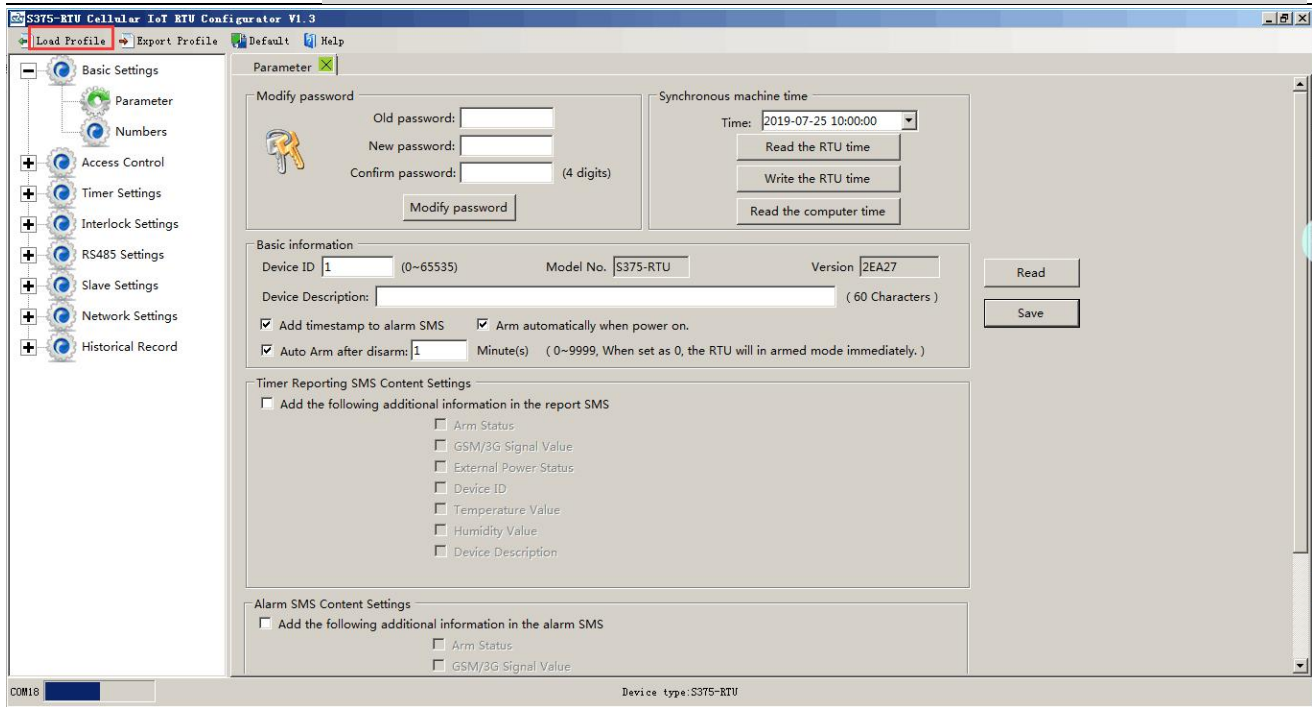


Then it will display as below after a moment:



## 7.2 Load Profile

Click "Load Profile" button----> chose the file which need to load.



Then it will display as below after a moment:



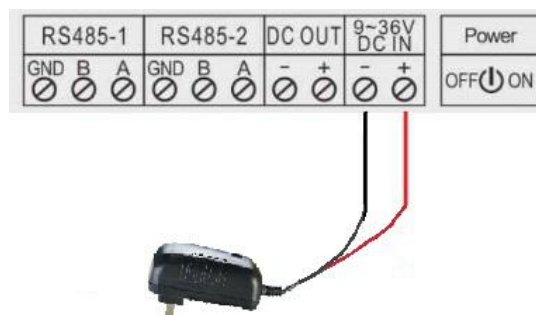
## 7.3 Reset

When device is on, connect the PC configuration software, click the “Default” button on software to reset. It will restore all parameters of the device to the factory default initial value. Please contact King Pigeon sales if forget password, website [www.gprs-m2m.com](http://www.gprs-m2m.com).

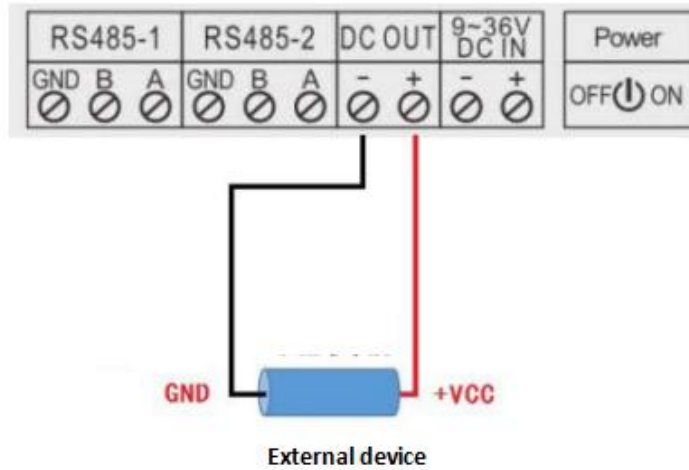
## 8. Connection and Application

### 8.1 Wire Connection

#### 8.1.1 Power wire connection:

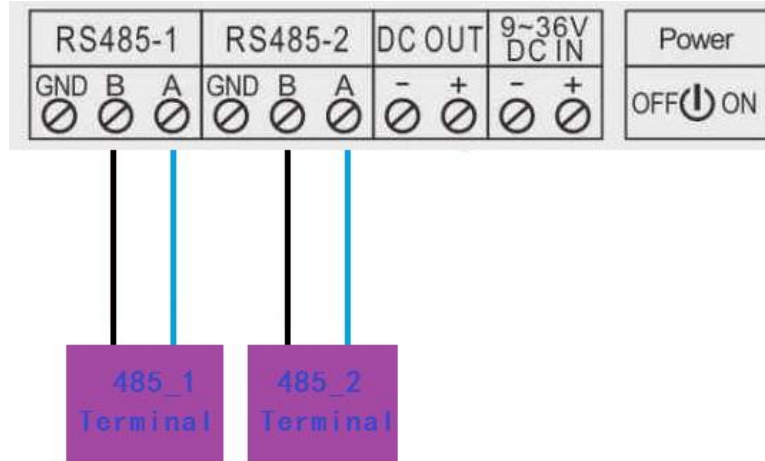


**8.1.2 DC output:**



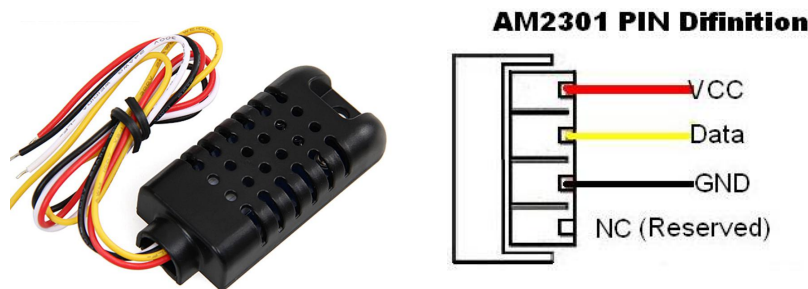
**8.1.3 RS485:**

S17X/S37X supports one RS485 port for communication, connection as below:



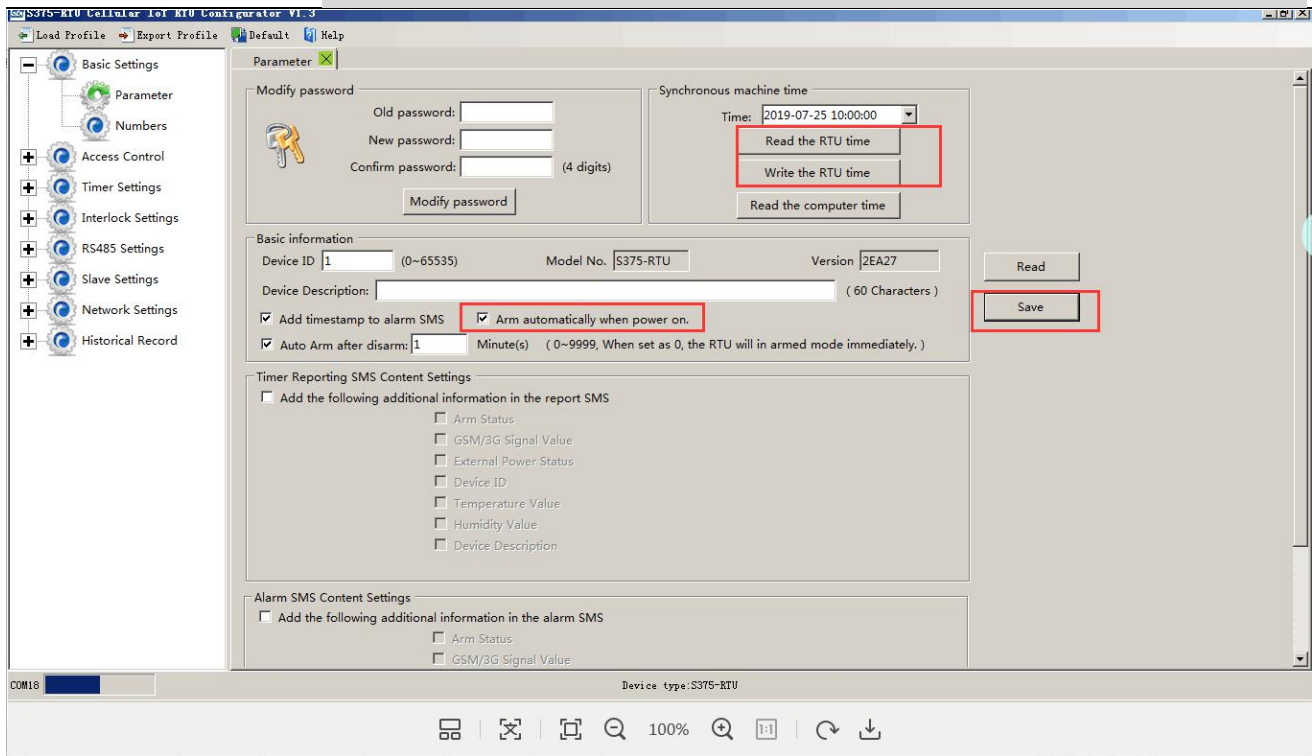
**8.1.4 Temperature/Humidity input:**

S17X/S37X supports one channel temperature and humidity input for sensor AM230X as below:

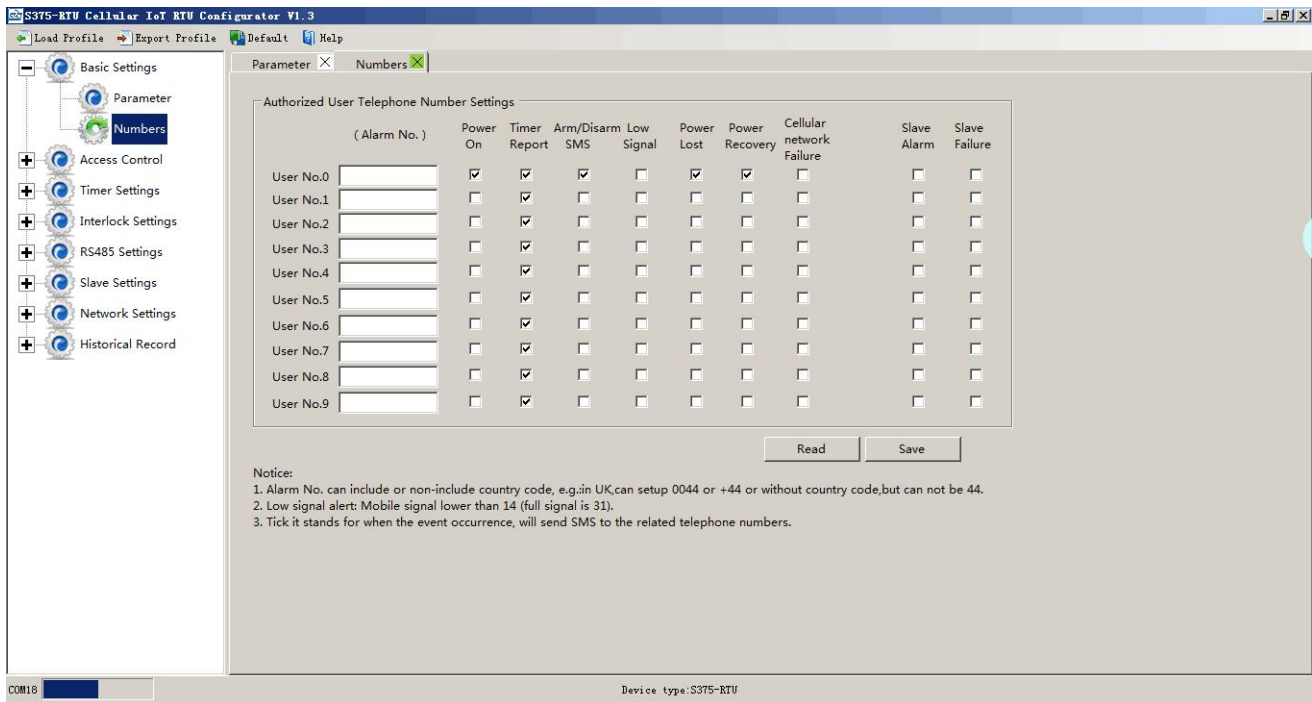


**8.2 Quickly Verify Device Communication Function**

Connect power supply to the device then switch on the device, run the configuration software, and log in. In “Basic Settings” page, click [Read the RTU Time] > [Write the RTU Time] to write the correct time to the device. Choose [Arm automatically when power on]. Then click the [Save]. As below:



In “Numbers” page, enter the phone number used to receive the alarm, choose the options according to request, such as power on, power lost.... then click [Save], as below:



Then switch off the device, insert sim card, switch on the device. after the device register network successfully, the phone number can receive SMS sent by the device. If you cut off the device power, the phone number will receive power failure SMS sent by the device.... That shows the device work normally.

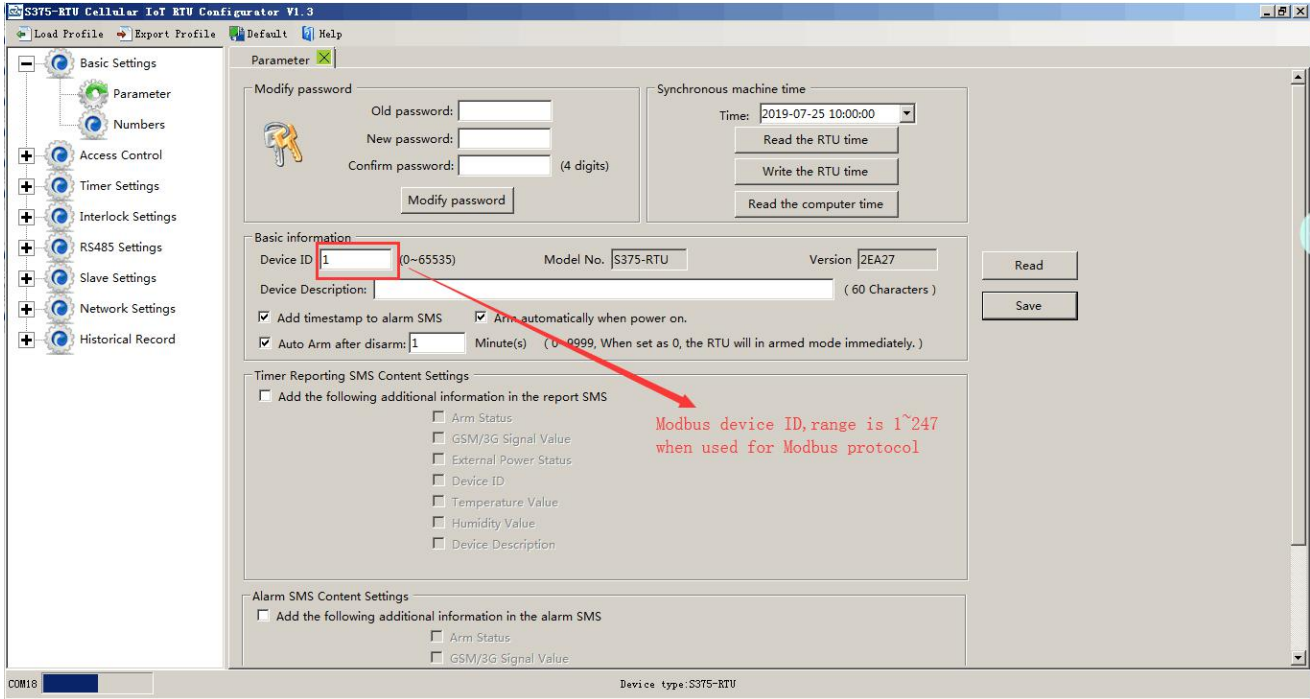
### 8.3 Modbus RTU Slave Application

Device support Modbus RTU slave function, can be connected to HMI, SCADA, DCS, MES system. It can be used for fieldbus data acquisition, remote SMS alarm, remote dial alarm and GPRS/3G/4G to cloud...



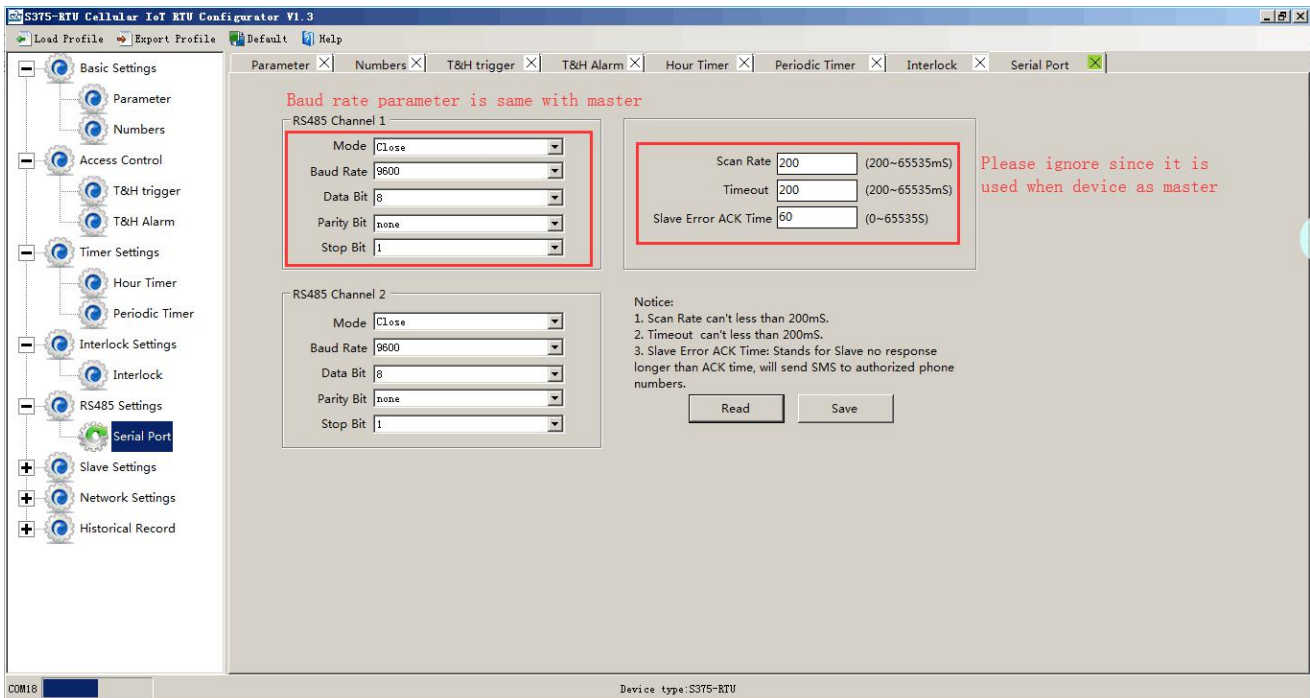
For example, when device as Modbus RTU slave, connect to HMI as below:

- (1) Connect device to HMI via RS485 port, set HMI RS485 port parameter;
- (2) In "Basic Parameter Settings" page, set "Device ID", range is 1~247 in Modbus protocol as below:



(3) In "Serial Port" setting page, set device parameter as below:

- A) RS485 used as "Modbus RTU Slave";
- B) Baud Rate, Data Bit, Parity Bit, Stop Bit setting should be corresponding with HMI, otherwise communication will be failure. If multiple Masters, all Masters parameter should corresponding with device;
- C) No need set: "Scan Rate", "Time Out", "Slave Fault Verify Time";
- D) Click "Save" button.



(4) In HMI configuration software, set the Modbus RTU Register address of device. Refer to **["S17X/S37X register"]**;



(5) Switch the device on, enter into working mode, device running according parameter setting.

### 8.3.1 Read device power supply voltage value

#### Master Send Data Format:

Content	Bytes	Data (H: HEX)	Description
Device Address	1	01H	01H Device, Range: 1-247, according to setting address
Function Code	1	04H	Read input register, function code 04
Register Starting Address	2	00 0EH	Power supply voltage address: 00 0EH
Read Register Qty	2	00 01H	1* 16 Bit register address
16CRC Verify	2	50 09H	CRC0 CRC1 low byte in front, high byte in behind

#### Receiver Return Data Format:

Content	Bytes	Data (H: HEX)	Description
Device Address	1	01H	01H Device, Range: 1-247, according to setting address
Function Code	1	04H	Read input register
Return Bytes Qty	1	02H	Return data length
Returning Data	2	04 B6H	Returning power supply voltage value
16 CRC Verify	2	3A 46H	CRC0 CRC1 low byte in front, high behind

Example: Inquiry device power supply voltage value, then:

**Server Send:** 01 04 00 0E 00 01 50 09

01= Device address; 04= Read input register; 00 0E= Origin register address; 00 01= Read 1 input register value; 50 09= CRC verify.

**Device Answer:** 01 04 02 04 B6 3A 46

01= Device address; 04= Read input register; 02=Return data bytes Qty; 04 B6=Returning power supply voltage value, the value is expanded 100 times (Real value= this value/100); 3A 46=CRC Verify.

### 8.3.2 Read device temperature and humidity value:

#### Master Send Data Format:

Content	Bytes	Data (H: HEX)	Description
Device Address	1	01H	01H Device, Range: 1-247, according to setting address
Function Code	1	04H	Read input register, function code 04
Register Starting Address	2	00 18H	Temperature address: 00 18H; Humidity address: 00 19H
Read Register Qty	2	00 02H	total 2* 16 Bit register address
16CRC Verify	2	F1 CCH	CRC0 CRC1 low byte in front, high byte in behind

#### Receiver Return Data Format:





Content	Bytes	Data (H: HEX)	Description
Device Address	1	01H	01H Device, Range: 1-247, according to setting address
Function Code	1	04H	Read input register
Return Bytes Qty	1	04H	Return data length
Returning Data	2	0B 0E 19 DCH	Returning temperature and humidity value
16 CRC Verify	2	92 6AH	CRC0 CRC1 low byte in front, high behind

Example: Inquiry device address, 2\*16 Bit register start from 18H. Match to device temperature, humidity, then:

**Server send:** 01 04 00 18 00 02 F1 CC

01= Device address; 04= Read input register value; 00 18= Temperature starting register address; 00 02= Serial reading 2 input register value; F1 CC= CRC verify.

**Device answer:** 01 04 04 0B 0E 19 DC 92 6A

01= Device address; 04= Read input register value; 04= Returning data bytes; 0B 0E 19 DC= Returning temperature and humidity value. 2 bytes in front stands for temperature, 2 bytes behind stands for humidity, and the value expand 100 times (Real value= this value/100). Converter to Decimal as:

Temperature	0B 0E	2830	Real value= this value/100
Humidity	19 DC	6620	Real value= this value/100

So temperature and humidity real value is 28.3 °C, 66.2%.

92 6A= CRC verify.

### 8.4 Modbus RTU Master Application

When RS485 as Modbus RTU Master, can extend I/O tags, support slaves for connecting Remote I/O data acquisition module, Smart meter, Power monitoring module, Smart transducer...; Can mapping register value from Slave to Master, these registers' can be setup high or low threshold value, and NC/NO type, moreover, can enable to send SMS to users once alarm occurrence by the registers if required. Also can remote control Slaves by writing coil.

#### Mapping Register Table and function code:

**Reminder:**

1. Use this function code when connect to Modbus RTU/Modbus TCP upper computer via GPRS/3G/4G (Device as Modbus RTU Slave). Stands for when Cloud communication with S17X/S37X, the S17X/S37X is Modbus RTU Slave of Cloud Server.
2. When device connect to Modbus RTU/Modbus TCP upper computer via GPRS/3G/4G (Device as Modbus RTU Slave). The I/O of S17X/S37X itself refer to "Modbus RTU Slave Application" above.

For example, when device as Modbus RTU master, as below:

**Step1:** Connect the slave to device RS485 port.

**Step2:** Find the slave port communication parameter and register address from user manual.

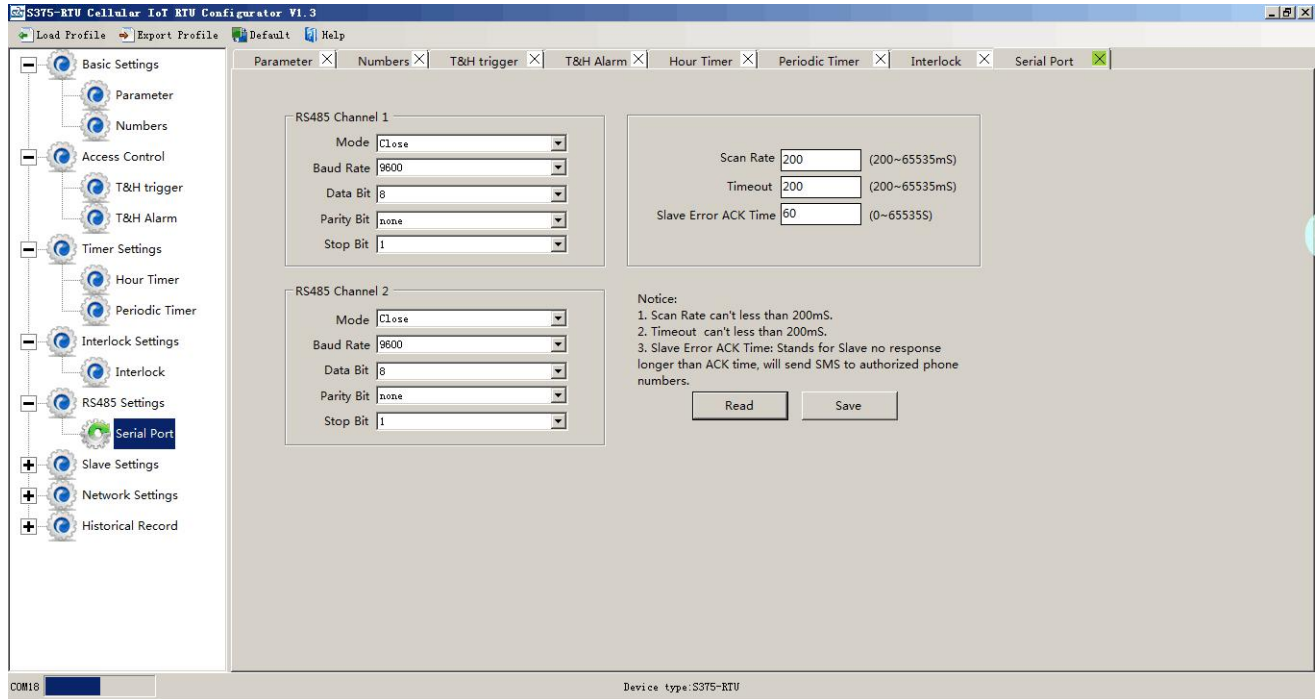
**Step3:** Write device RS485 parameter according to slave port communication parameter, pls ensure both parameter are same, others communication failure, refer to "port setting" part.



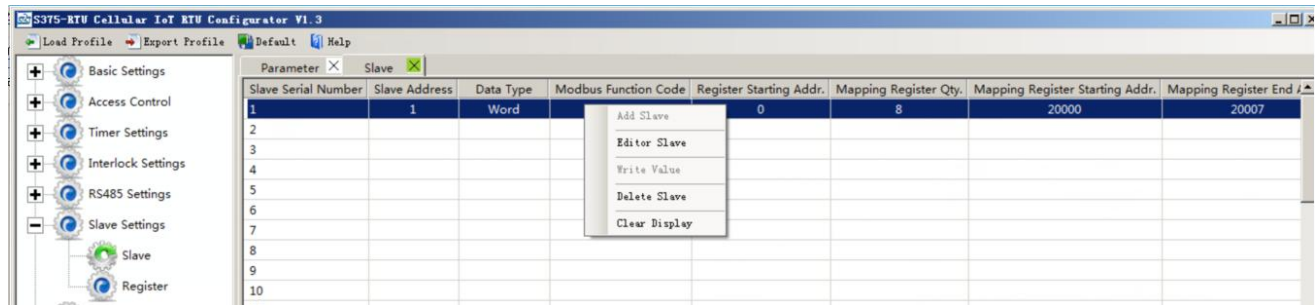
**Step4:** Set RS485 port as Modbus RTU Master, then set polling and time out parameter, refer to "port setting" part.

Serial Port Setting:

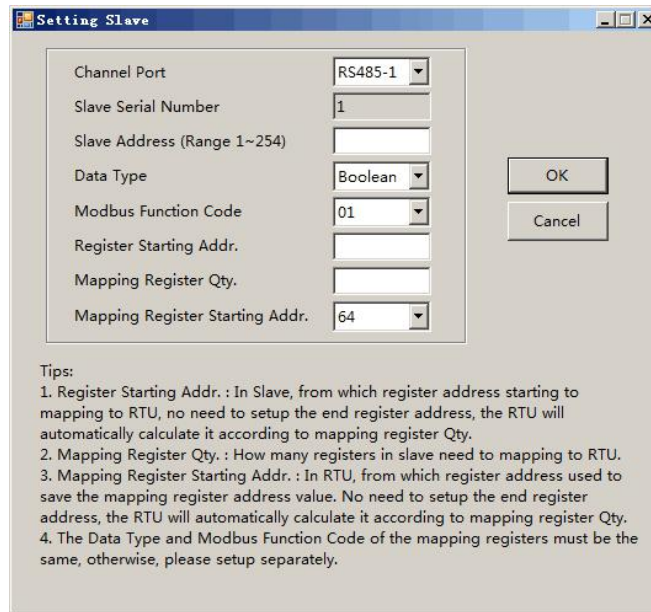
In "Serial Port" setting page, choose RS485 port as "Modbus RTU Master". Baud Rate, Data Bit, Parity Bit and Stop Bit parameter need to be same as connected device; Scan Rate, Time Out and Slave Failure Verify Time can be set as default:



**Step5:** Back to Slave Mapping page as below, right click the line to add.

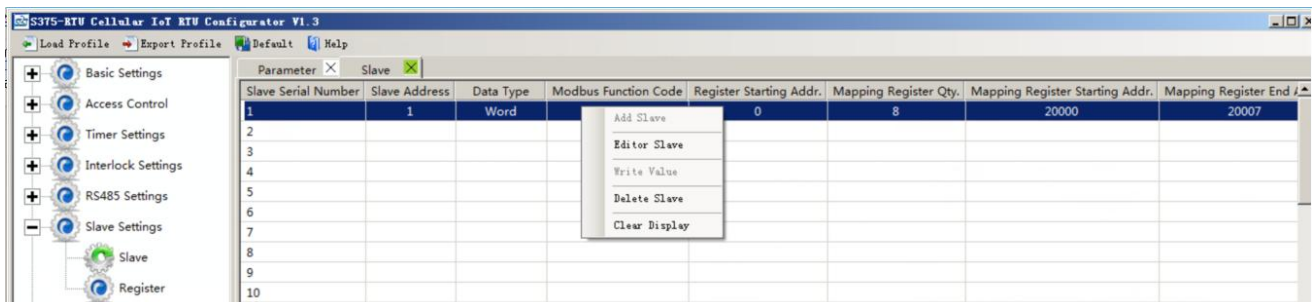


Click Add Slave as below:

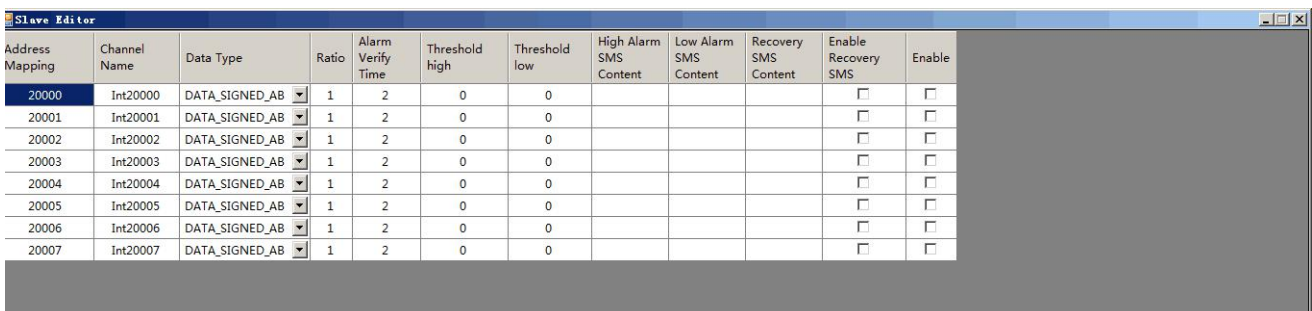


If one slave have multi register, then need to add seperately according to register type; For example, Mxxx remote I/O module, with digital and analog inputs, need to add the digital(Boolean) first, then add the analog(16 Bit).

**Step6:** Right click the line to edit the slave.



Click the "Editor Slave" to set channel name, alarm SMS content, recovery SMS content and relay active etc. as below:



Address Mapping	Channel Name	Data Type	Ratio	Alarm Verify Time	Threshold high	Threshold low	High Alarm SMS Content	Low Alarm SMS Content	Recovery SMS Content	Enable Recovery SMS	Enable
20000	Int20000	DATA_SIGNED_AB	1	2	0	0				<input type="checkbox"/>	<input type="checkbox"/>
20001	Int20001	DATA_SIGNED_AB	1	2	0	0				<input type="checkbox"/>	<input type="checkbox"/>
20002	Int20002	DATA_SIGNED_AB	1	2	0	0				<input type="checkbox"/>	<input type="checkbox"/>
20003	Int20003	DATA_SIGNED_AB	1	2	0	0				<input type="checkbox"/>	<input type="checkbox"/>
20004	Int20004	DATA_SIGNED_AB	1	2	0	0				<input type="checkbox"/>	<input type="checkbox"/>
20005	Int20005	DATA_SIGNED_AB	1	2	0	0				<input type="checkbox"/>	<input type="checkbox"/>
20006	Int20006	DATA_SIGNED_AB	1	2	0	0				<input type="checkbox"/>	<input type="checkbox"/>
20007	Int20007	DATA_SIGNED_AB	1	2	0	0				<input type="checkbox"/>	<input type="checkbox"/>

**Step7:** Restart the device, enter into working mode, device running according parameter setting, include alarm SMS and call. If set the network communication function, then can remote transmit data to cloud server via GPRS/3G/4G.

**Reminder:**

*After adding slaves, device switched off/on to restart is necessary.*

### 8.4.1 Read Boolean Mapping Address Data:

#### Master Send Data Format:

Content	Bytes	Data (H: HEX)	Description
Device Address	1	01H	01H Device, Range: 1-247, according to setting address
Function Code	1	01H	Read holding coil type, function code 01
Boolean Register Starting Address	2	00 40H	Range: 0040H-007FH, address refer to [" <b>Slave Mapping Register Address</b> "] at manual bottom
Read Register Qty	2	00 0AH	Range: 0001H-0040H, total 64 address for Boolean mapping
16 CRC Verify	2	BD D9H	CRC0 CRC1 low byte in front, high byte in behind

#### Receiver Return Data Format:

Content	Bytes	Data (H: HEX)	Description
Device Address	1	01H	01H Device, according to the data Master send
Function Code	1	01H	Read holding coil
Return Byte Length	1	02H	Return Data Length
Returning Data	2	73 01H	
16CRC Verify	2	5D 0CH	CRC0 CRC1 low byte in front, high behind

Example: Start from address 64, read 10 Boolean mapping data value, then:

**Server send:** 01 01 00 40 00 0A BD D9

01= Device address; 01= Read holding coil; 00 40=Read Boolean data start from address 64; 00 0A = Serial to read 10 Boolean status;

BD D9= CRC verify.

**Device answer:** 01 01 02 73 01 5D 0C

01= Device address; 01= Read holding coil; 02= Return Byte Length; 73 01= Return 10 Boolean status. High byte stands for low address data, low address stands for high address. According to Modbus protocol, fix 73 01H real value to be 01 73H, converter to Binary as below:

Register mapping address	Invalid	Invalid	Invalid	Invalid	Invalid	Invalid	73	72
Value	0	0	0	0	0	0	0	1
Register mapping address	71	70	69	68	67	66	65	64
Value	0	1	1	1	0	0	1	1

The address value higher than 10 digits will be seen as invalid.

5D 0C =CRC verify.

### 8.4.2 Modify Boolean Mapping Address Data:

If control relay status which connected to RS485, need to add slave in slave list of configurator. Write command 15 for mapping, when mapping address value modified, will write to RS485 matched slave address.

### Master Send Data Format:

Content	Bytes	Data (H: HEX)	Description
Device Address	1	01H	01H Device, Range: 1-247, according to setting address
Function Code	1	05H	Write single holding coil, function code 05H
Boolean Mapping Register Address	2	00 40H	Range: 00 40H-00 7FH, address refer to [" <b>Slave Mapping Register Address</b> "] at manual bottom
Write value	2	FF 00H	This value: FF 00H or 00 00H, FF 00H stands for write 1; 00 00H stands for write 0
16 CRC Verify	2	8D EEH	CRC0 CRC1 low byte in front, high behind

### Receiver Return Data Format:

Content	Bytes	Data (H: HEX)	Description
Device Address	1	01H	01H Device, according to the data Master send
Function Code	1	05H	Write single holding coil
Boolean Mapping Register Address	2	00 40H	Range: 0040H-007FH
Write value	2	FF 00H	This value: FF 00H or 00 00H. FF 00H stands for write 1, 00 00H stands for write 0.
16 CRC Verify	2	8D EEH	CRC0 CRC1 low byte in front, high behind

Example: Modify Boolean mapping address 64 status, modify to 1, then:

**Server send:** 01 05 00 40 FF 00 8D EE

01= Device address; 05= Write Boolean value; 00 40=The mapping address which need to revise; FF 00 = Write 1;

8D EE = 16 Bit CRC verify.

**Device answer:** 01 05 00 40 FF 00 8D EE

01= Device address; 05= Write Boolean value; 00 40= The mapping address which need to write; FF 00= Write 1.

8D EE = 16 Bit CRC verify.

If need multi modify, pls check function 15 of Modbus protocol.

### 8.4.3 Read Data Type Mapping Address Data:

#### Master Send Data Format:

Content	Bytes	Data (H: HEX)	Description
Device Address	1	01H	01H Device, Range: 1-247, according to setting address
Function Code	1	03H	Read holding coil, function code 03



Mapping Register Starting Address	2	4E 20H	One address can read 2 bytes. Mapping data type address range, refer to [ <b>“Slave Mapping Register Address”</b> ] at manual bottom.
Read Mapping Register Qty	2	00 0AH	Read input register qty.
16 CRC Verify	2	3D 2FH	CRC0 CRC1 low byte in front, high behind

### Receiver Return Data Format:

Content	Bytes	Data (H: HEX)	Description
Device Address	1	01H	01H Device, according to the data Master send
Function Code	1	03H	Read holding coil
Range Data Bytes	1	14H	One address can read 2 bytes
Returning Data	20	00 14 00 1E 00 28 00 32 00 4B 00 41 00 0A 00 25 00 14 00 2AH	Returning Data
16 CRC Verify	2	FB 34H	CRC0 CRC1 low byte in front, high behind

Example: Mapping address start from 20000, read 10 address data, then:

**Server send:** 01 03 4E 20 00 0A D3 2F

01= Device address; 03= Read holding coil; 4E 20=Mapping register starting address, current is Decimal data 20000; 00 0A = Read 10 register value;  
D3 2F = 16 Bit CRC verify.

**Device answer:** 01 03 14 00 14 00 1E 00 28 00 32 00 4B 00 41 00 0A 00 25 00 14 00 2A FB 34

01= Device address; 03= Read holding register; 14= Returning 20 byte; 00 14 00 1E 00 28 00 32 00 4B 00 41 00 0A 00 25 00 14 00 2A = Returning data.

Register Mapping Address	20000	20001	20002	20003	20004	20005	20006	20007	20008	20009
Value	00 14	00 1E	00 28	00 32	00 4B	00 41	00 0A	00 25	00 14	00 2A

FB 34 = 16 Bit CRC verify.

### 8.4.4 Modify Data Type Mapping Address Data:

If need to revise slave data which RS485 connected, need to add slave in slave list of configurator. Write command 16 for mapping, when mapping address value modified, will write to RS485 matched slave address. If address 20000 mapping slave data type is Signed Int, sort AB.

### Master Send Data Format:

Content	Bytes	Data (H: HEX)	Description
Device Address	1	01H	01H Device, Range: 1-247, according to setting address
Function Code	1	06H	Write single holding register, function code 06
Mapping Register Address	2	4E 20H	Address range: 4E 20H-50 1CH. Mapping data type address range, refer to [ <b>“Slave Mapping Register Address”</b> ] at manual bottom.



Write Data	2	00 64H	Data writing value is Decimal data 100
16 CRC Verify	2	9E C3H	CRC0 CRC1 low byte in front, high behind

### Receiver Return Data Format:

Content	Bytes	Data (H: HEX)	Description
Device Address	1	01H	01H Device, according to the data Master send
Function Code	1	06H	Write single holding register
Mapping Register Address	2	4E 20H	Address range: 4E20H-501CH, mapping data type address range.
Write Data	2	00 64H	Write 100 successfully
16 CRC Verify	2	9E C3H	CRC0 CRC1 low byte in front, high behind

Example: If address 20000 mapping slave data type is Signed Int, sort AB, modify mapping address 20000 register to 100, then:

**Server send:** 01 06 4E 20 00 64 9E C3

01= Device address; 06= Modify single holding register value; 4E 20=Modify address 20000 register value; 00 64 = Write Decimal value 100; 9E C3 = 16 Bit CRC verify.

**Device answer:** 01 06 4E 20 00 64 9E C3

01= Device address; 06= Modify single holding register value; 4E 20= R Modify address 20000 register value; 00 64= Modify to Decimal value 100. 9E C3 = 16 Bit CRC verify.

If need to modify multi data type mapping address, pls check function code 16 in Modbus protocol.

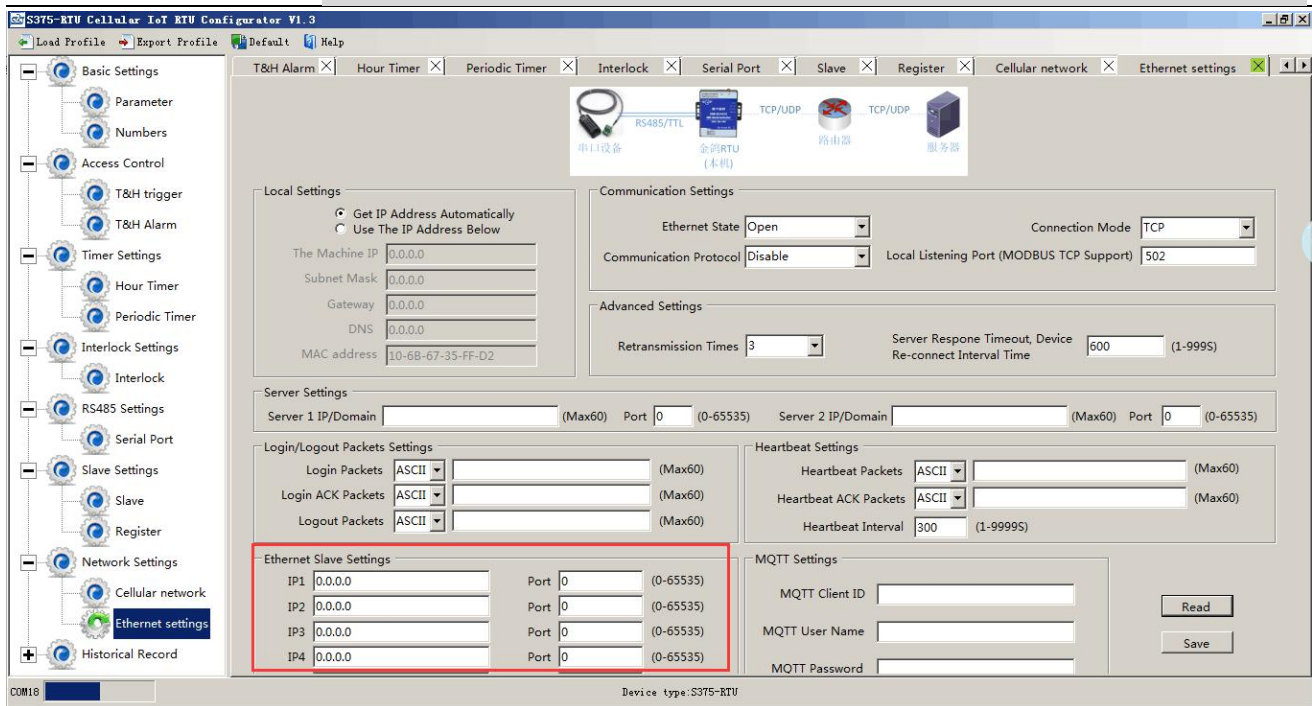
## 8.5 Modbus TCP Master Application

When Ethernet as Modbus TCP master,can extend I/O tags, support slaves for connecting Remote I/O data acquisition module, Smart meter, Power monitoring module, Smart transducer... Can mapping register value from Slave to Master, these registers' can be setup high or low threshold value, and NC/NO type, moreover, can enable to send SMS to users once alarm occurrence by the registers if required. Also can remote control Slaves by writing coil.

For example, when device as Modbus TCP master,as below:

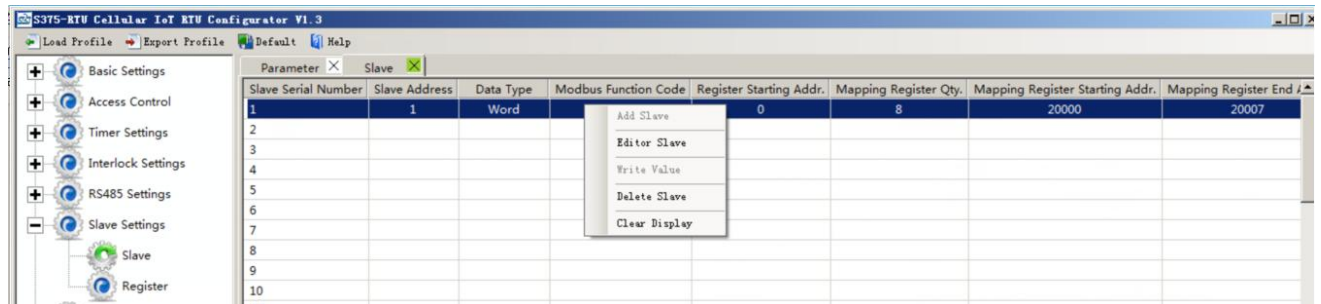
1)Ethernet Setting:

In "Ethernet Settings" page, choose correct slave IP and port,click "save" to add,support up to 5 Ethernet slave.

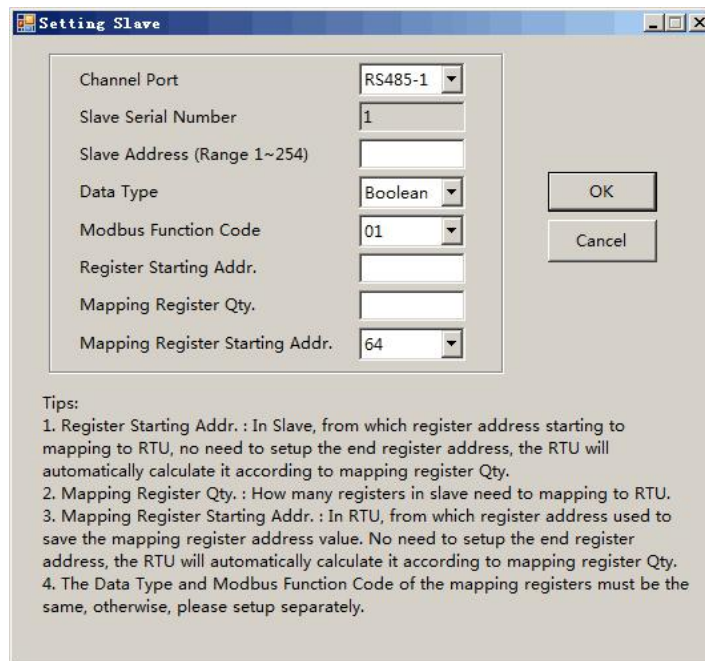


## 2)Slave Setting:

In "Slave" page, right click the line to add, as below:



Click Add Slave as below:

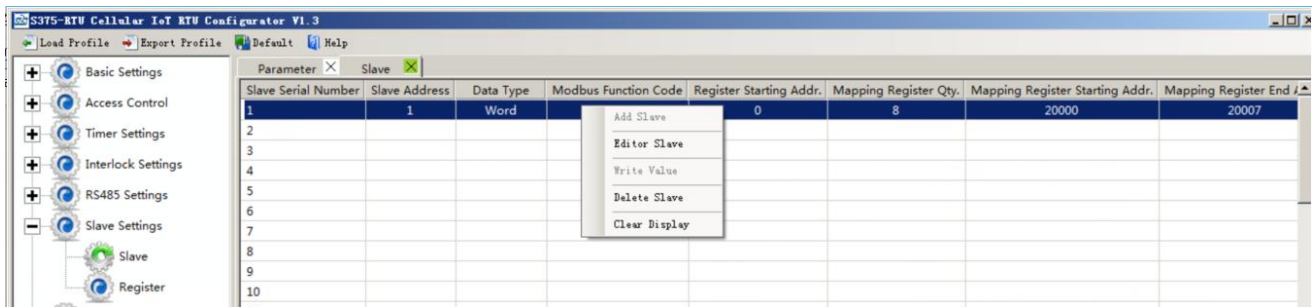


If one slave have multi register, then need to add separately according to register type; For example, Mxxx remote I/O module, with digital and analog inputs, need to add the digital(Boolean) first, then add the

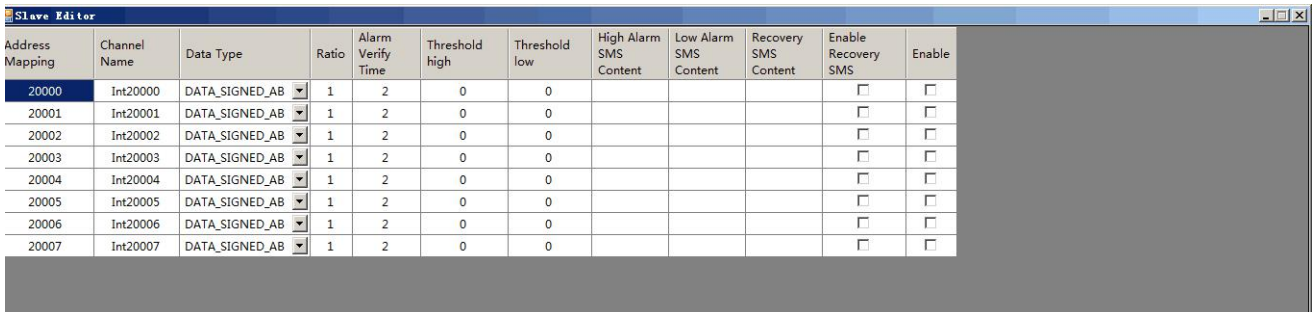


analog(16 Bit).

**Step6:** Right click the line to edit the slave.



Click the "Editor Slave" to set channel name, alarm SMS content, recovery SMS content and relay active etc. as below:



Address Mapping	Channel Name	Data Type	Ratio	Alarm Verify Time	Threshold high	Threshold low	High Alarm SMS Content	Low Alarm SMS Content	Recovery SMS Content	Enable Recovery SMS	Enable
20000	Int20000	DATA_SIGNED_AB	1	2	0	0				<input type="checkbox"/>	<input type="checkbox"/>
20001	Int20001	DATA_SIGNED_AB	1	2	0	0				<input type="checkbox"/>	<input type="checkbox"/>
20002	Int20002	DATA_SIGNED_AB	1	2	0	0				<input type="checkbox"/>	<input type="checkbox"/>
20003	Int20003	DATA_SIGNED_AB	1	2	0	0				<input type="checkbox"/>	<input type="checkbox"/>
20004	Int20004	DATA_SIGNED_AB	1	2	0	0				<input type="checkbox"/>	<input type="checkbox"/>
20005	Int20005	DATA_SIGNED_AB	1	2	0	0				<input type="checkbox"/>	<input type="checkbox"/>
20006	Int20006	DATA_SIGNED_AB	1	2	0	0				<input type="checkbox"/>	<input type="checkbox"/>
20007	Int20007	DATA_SIGNED_AB	1	2	0	0				<input type="checkbox"/>	<input type="checkbox"/>

**Step7:** Restart the device, enter into working mode, device running according parameter setting, include alarm SMS and call. If set the network communication function, then can remote transmit data to cloud server via GPRS/3G/4G.

**Reminder:**

*After adding slaves, device switched off/on to restart is necessary.*

### 8.5.1 Read Boolean Mapping Address Data

**Master Send Data Format:**

Content	Bytes	Data (H: HEX)	Description
Transaction ID	2	00 00H	Identification of Modbus requests/responses transactions
Protocol Identifier	2	00 00H	00 00: Modbus protocol
Data Length	2	00 06H	The length of subsequent data
Device Address	1	01H	01H Device, Range: 1-247, according to setting address
Function Code	1	01H	Read holding coil type, function code 01
Boolean Register Starting Address	2	00 40H	Range: 0040H-007FH, address refer to [" <b>Slave Mapping Register Address</b> "] at manual bottom
Read Register Qty	2	00 0AH	Range : 0001H-0004H

**Receiver Return Data Format:**

Content	Bytes	Data	Description
---------	-------	------	-------------



		(H: HEX)	
Transaction ID	2	00 00H	Identification of Modbus requests/responses transactions
Protocol Identifier	2	00 00H	00 00: Modbus protocol
Data Length	2	00 05H	The length of subsequent data
Device Address	1	01H	01H Device, Range: 1-247, according to setting address
Function Code	1	01H	Read holding coil type, function code 01
Return Data Length	1	02H	Range: 0000H-0003H
Returning Data	2	73 01H	

Example: Start from address 64, read 10 Boolean mapping data value, then:

**Server send:** 00 00 00 00 00 06 01 01 00 40 00 0A

00 00= Transaction ID; 00 00 = Protocol Identifier, 00 00 stands for Modbus protocol; 00 06 = The length of subsequent data; 01= Device address; 01 = Read holding coil; 00 00 = Read Boolean data start from address 64; 00 0A = Serial to read 10 Boolean status;

**Device answer:** 00 00 00 00 00 05 01 01 02 73 01

00 00= Transaction ID; 00 00 = Protocol Identifier, 00 00 stands for Modbus protocol; 00 05 = The length of subsequent data; 01= Device address; 01 = The function code of reading Boolean; 02= Return Data Length; 73 01= Return 10 Boolean status. High byte stands for low address data, low address stands for high address.

According to Modbus protocol, fix 73 01H real value to be 01 73H, converter to Binary as below:

Register mapping address	Invalid	Invalid	Invalid	Invalid	Invalid	Invalid	73	72
Value	0	0	0	0	0	0	0	1
Register mapping address	71	70	69	68	67	66	65	64
Value	0	1	1	1	0	0	1	1

The address value higher than 10 digits will be seen as invalid.

### 8.5.2 Modify Boolean Mapping Address Data

If control slave's relay status which connected to Ethernet, need to add slave in slave list of configurator. Write command 15 for mapping, when mapping address value modified, will write to Ethernet matched slave address.

#### Master Send Data Format:

Content	Bytes	Data (H: HEX)	Description
Transaction ID	2	00 00H	Identification of Modbus requests/responses transactions
Protocol Identifier	2	00 00H	00 00: Modbus protocol
Data Length	2	00 06H	The length of subsequent data
Device Address	1	01H	01H Device, Range: 1-247, according to setting address
Function Code	1	05H	Write single holding coil, function code 05H



Boolean Mapping Register Address	2	00 40H	Range: 00 40H-00 7FH, address refer to [" <b>Slave Mapping Register Address</b> "] at manual bottom
Write value	2	FF 00H	This value: FF 00H or 00 00H, FF 00H stands for write 1; 00 00H stands for write 0

### Receiver Return Data Format:

Content	Bytes	Data (H: HEX)	Description
Transaction ID	2	00 00H	Identification of Modbus requests/responses transactions
Protocol Identifier	2	00 00H	00 00: Modbus protocol
Data Length	2	00 06H	The length of subsequent data
Device Address	1	01H	01H Device, according to the data Master send
Function Code	1	05H	Write single holding coil
Boolean Mapping Register Address	2	00 40H	Range: 00 40H-00 7FH, address refer to [" <b>Slave Mapping Register Address</b> "] at manual bottom
Write value	2	FF 00H	This value: FF 00H or 00 00H. FF 00H stands for write 1, 00 00H stands for write 0.

Example: Modify Boolean mapping address 64 status, modify to 1, then:

**Server send:** 00 00 00 00 00 06 01 05 00 40 FF 00

00 00= Transaction ID; 00 00 = Protocol Identifier, 00 00 stands for Modbus protocol; 00 06 = The length of subsequent data; 01= Device address; 05= Write holding coil; 00 40=The mapping address which need to revise; FF 00 = Write 1;

**Device answer:** 00 02 00 00 00 06 01 05 00 40 FF 00

00 00= Transaction ID; 00 00 = Protocol Identifier, 00 00 stands for Modbus protocol; 00 06 = The length of subsequent data; 01= Device address; 05= Write holding coil; 00 40= The mapping address which need to write; FF 00= Write 1;

If need multiple modify, pls check function 15 of Modbus protocol.

### 8.5.3 Read Data Type Mapping Address Data

#### Master Send Data Format:

Content	Bytes	Data (H: HEX)	Description
Transaction ID	2	00 00H	Identification of Modbus requests/responses transactions
Protocol Identifier	2	00 00H	00 00: Modbus protocol
Data Length	2	00 06H	The length of subsequent data
Device Address	1	01H	01H Device, Range: 1-247, according to setting address
Function Code	1	03H	Read holding register, function code 03
Mapping Register Starting Address	2	4E 20H	One address can read 2 bytes. Mapping data type address range, refer to [" <b>Slave Mapping Register Address</b> "] at manual bottom.



Read Mapping Register Qty	2	00 0AH	Read input register qty.
---------------------------	---	--------	--------------------------

### Receiver Return Data Format:

Content	Bytes	Data (H: HEX)	Description
Transaction ID	2	00 00H	Identification of Modbus requests/responses transactions
Protocol Identifier	2	00 00H	00 00: Modbus protocol
Data Length	2	00 17H	The length of subsequent data
Device Address	1	01H	01H Device, according to the data Master send
Function Code	1	03H	Read holding register
Range Data Bytes	1	14H	One address can read 2 bytes
Returning Data	20	00 14 00 1E 00 28 00 32 00 4B 00 41 00 0A 00 25 00 14 00 2AH	Returning Data

Example: Mapping address start from 20000, read 10 address data, then:

**Server send:** 00 00 00 00 00 06 01 03 4E 20 00 0A

00 00= Transaction ID; 00 00 = Protocol Identifier,00 00 stands for Modbus protocol; 00 06 = The length of subsequent data; 01= Device address; 03= Read holding register ; 4E 20=Mapping register starting address, current is Decimal data 20000; 00 0A = Read 10 register value;

**Device answer:** 00 00 00 00 00 17 01 03 14 00 14 00 1E 00 28 00 32 00 4B 00 41 00 0A 00 25 00 14 00 2A

00 00= Transaction ID; 00 00 = Protocol Identifier,00 00 stands for Modbus protocol; 00 17 = The length of subsequent data;01= Device address; 03= Read holding register; 14= Returning 20 byte; 00 14 00 1E 00 28 00 32 00 4B 00 41 00 0A 00 25 00 14 00 2A = Returning data.

Register Mapping Address	20000	20001	20002	20003	20004	20005	20006	20007	20008	20009
Value	00 14	00 1E	00 28	00 32	00 4B	00 41	00 0A	00 25	00 14	00 2A

### 8.5.4 Modify Data Type Mapping Address Data

If need to revise slave data which Ethernet connected, need to add slave in slave list of configurator. Write command 16 for mapping, when mapping address value modified, will write to Ethernet matched slave address.

If address 20000 mapping slave data type is Signed Int, sort AB.

### Master Send Data Format:

Content	Bytes	Data (H: HEX)	Description
Transaction ID	2	00 00H	Identification of Modbus requests/responses transactions
Protocol Identifier	2	00 00H	00 00: Modbus protocol
Data Length	2	00 06H	The length of subsequent data



Device Address	1	01H	01H Device, Range: 1-247, according to setting address
Function Code	1	06H	Write single holding register, function code 06
Mapping Register Address	2	4E 20H	Address range: 4E 20H-50 1CH. Mapping data type address range, refer to [ <b>“Slave Mapping Register Address”</b> ] at manual bottom.
Write Data	2	00 64H	Data writing value is Decimal data 100

### Receiver Return Data Format:

Content	Bytes	Data (H: HEX)	Description
Transaction ID	2	00 00H	Identification of Modbus requests/responses transactions
Protocol Identifier	2	00 00H	00 00: Modbus protocol
Data Length	2	00 06H	The length of subsequent data
Device Address	1	01H	01H Device, according to the data Master send
Function Code	1	06H	Write single holding register
Mapping Register Address	2	4E 20H	Address range: 4E20H-501CH, mapping data type address range.
Write Data	2	00 64H	Write 100 successfully

Example: If address 20000 mapping slave data type is Signed Int, sort AB, modify mapping address 20000 register to 100, then:

**Server send:** 00 00 00 00 00 06 01 06 4E 20 00 64

00 00= Transaction ID; 00 00 = Protocol Identifier,00 00 stands for Modbus protocol; 00 06 = The length of subsequent data; 01= Device address; 06= Modify single holding register value; 4E 20=Modify address 20000 register value; 00 64 = Write Decimal value 100;

**Device answer:** 00 00 00 00 00 06 01 06 4E 20 00 64

00 00= Transaction ID; 00 00 = Protocol Identifier,00 00 stands for Modbus protocol; 00 06 = The length of subsequent data; 01= Device address; 06= Modify single holding register value; 4E 20= R Modify address 20000 register value; 00 64= Modify to Decimal value 100.

If need to modify multiple data type mapping address, pls check function code 16 in Modbus protocol.

## 8.6 Transparent Transmission DTU Application

Device can support data transparent transmission: DTU function. Could server transmit data to device via GPRS/3G/4G, device will transfer the data to RS485 port directly without deal with. Once device receive data from RS485, also transmit to cloud server directly via GPRS/3G/4G. When device RS485 port no need mapping slave, or connect to others which is not standard Modbus RTU protocol, then can choose transparent transmission as below:

- (1) Items connect to device via RS485, set RS485 port parameter;
- (2) Basic setting page to set device ID;



(3) In "Serial Port" setting page, device parameter as below:

- A) Choose RS485 as "Transparent Transmission";
- B) Baud Rate, Data Bit, Parity Bit, Stop Bit setting should be corresponding with items, otherwise communication will be failure. If multiple items, all items parameter should corresponding with device;
- C) No need set: "Scan Rate", "Time Out", "Slave Fault Verify Time";
- D) Click "Save" button.

**Reminder:**

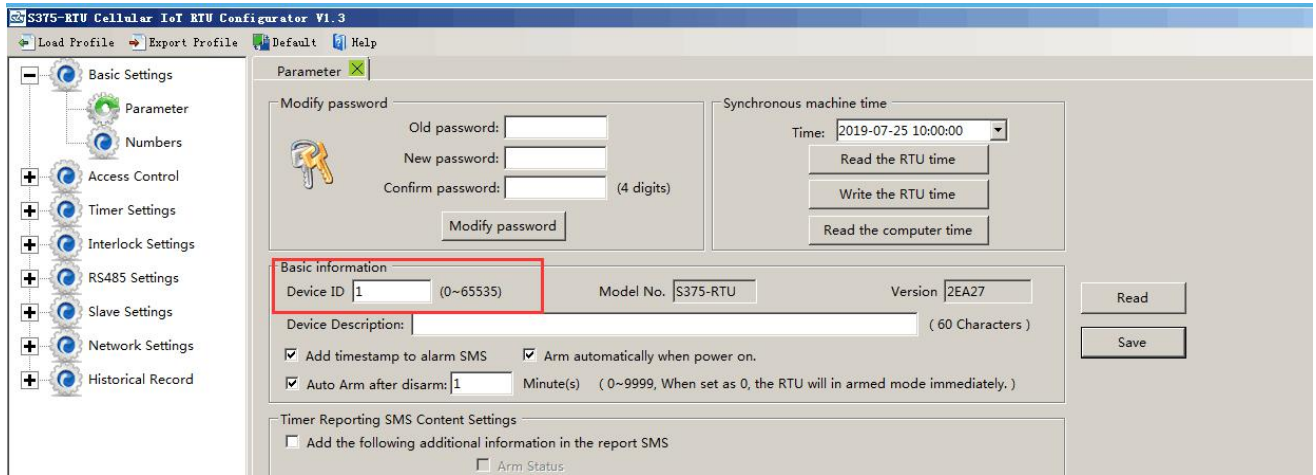
*The device ID which connect to RS485 can't be same with S257's device ID.*

## 8.7 Device connect to cloud Application

Device can connect to cloud and SCADA via GPRS/3G/4G network, also can connect to clients own server and King Pigeon [www.My-M2M.com](http://www.My-M2M.com) cloud server. King Pigeon [my-m2m.com](http://my-m2m.com) cloud as sample below:

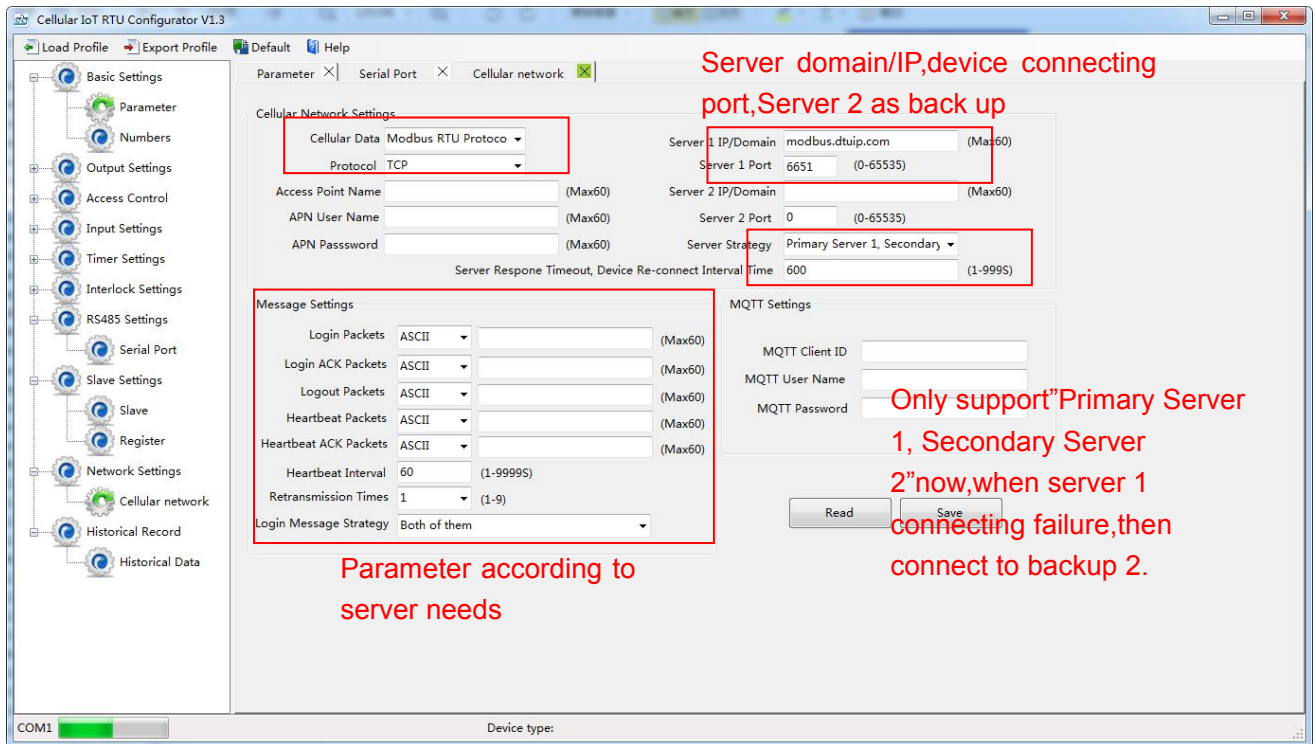
King Pigeon my m2m cloud support Modbus TCP, cloud configuration, wechat alarm function, welcomed editable function.

(1) In "Basic Parameter" setting page, set device ID, range 1~247 in Modbus RTU protocol as below:



(2) In "Cellular network" setting page, set parameter as below:

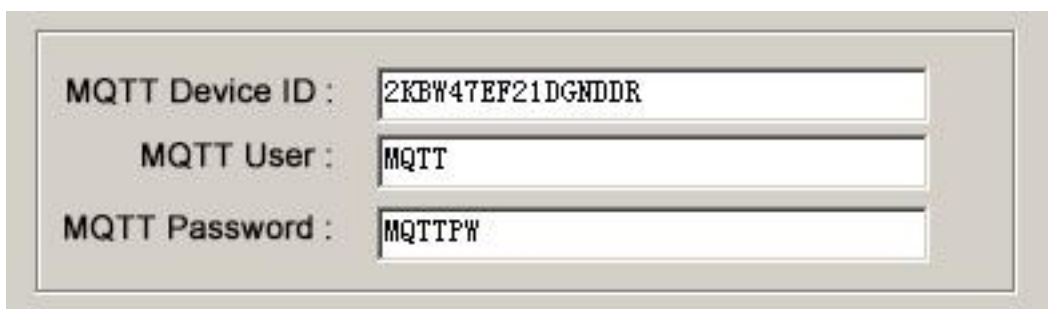
When Communication Data as "Modbus RTU Protocol", then server IP/DNS should be: modbus.dtuip.com, port is 6651, pls contact King Pigeon Sales for "Login Message Writing";



(3) Click "Save Settings" in the menu, then switch device off.

(4) Switch the device on, enter into working mode, then Slave and Master I/O can connect to network.

(5) Need enter MQTT parameter when choose MQTT protocol:



## 9. Device SMS Command

1. The default Password is 1234.
2. The unit cannot support PIN Code Protected SIM Card.
3. You can program the GSM unit with SMS commands using your phone.
4. Remember that commands must be CAPITAL LETTERS. It is PWD not pwd, CAP not Cap etc. Don't add spaces or any other character.
5. In some GSM operators they use different SMS parameter; the units can't return the SMS confirmation in some gsm operators, but it can performance the functions correctly. Also, you can try to add the country code

For example:  
 E.g.: the country code is 0086, or +86.  
 The user cell phone number is 13600000000 and has been assigned as a SMS Alert number, the simcard number in the panel is 13512345678.  
 When you setup the number as the authorized number, please setup as 008613600000000 or +86136000000000. Not 13600000000.

before the number, see the below settings:

6. If the password is correct but the command is incorrect, the device will return: SMS Format Error, Please check Caps Lock in Command! So please check the Command, or add the country code before the telephone number or check the input is in ENGLISH INPUT METHOD and CAPS LOCK. If password incorrect then will not any response SMS.
7. Once the Unit received the SMS Command, will return SMS to confirmation, if no SMS return, please check your command or resend again.
8. The SMS commands that you will certainly use in the GSM units are the following:

**\*\*SMS Commands For Program and Operation the S17X/S37X\*\***



### SMS Command List:

The SMS commands will be used for remote control the RTU are below:

#### 1) **Commands error return SMS**

Event	Return SMS Content
Any incorrect Command	SMS Format Error, Please check Caps Lock in Command!

#### 2) **External DC Status**

Event	Return SMS Content
External DC goes off	External DC Power Goes OFF
External DC Power Goes ON	External DC Power Goes ON

#### 3) **Modify Password, 4digits, default is 1234**

SMS Command	Return SMS Content
Old Password + P + New Password	This is the New Password, please remember it carefully.

#### 4) **Reboot**

SMS Command	Return SMS Content
Password + Reboot	

#### 5) **Arm/Disarm SMS Command**

SMS Command	Return SMS Content
-------------	--------------------





Arm	password+AA	Armed
Disarm	password+BB	Disarmed

6) **Set RTU time**, format is 2015-05-22 15:20:30W01, the W01 stands for Monday, W07 stands for Sunday.

SMS Command	Return SMS Content
password+Dxxxx-xx-xxTxx: xx: xxWxx	xxxx(Y)XX(M)XX(D)xx(H)X(M)xx(W)

7) **Inquiry Current Status SMS Command**

SMS Command	Return SMS Content
password+EE	Armed/Disarmed Model: Version: IMEI: GSM Signal Value: External DC Power Goes OFF/ON

8) **Setup 10 User number**(Alarm Number&Access Control Number), max 21digits. ( Return 0~4 or 5~9 separately while setting.)

SMS Command	Return SMS Content
Setup password+A+series number+T+tel number  <b>Notice:</b> Series number = 0~9	Tel1: --- Tel2: --- Tel3: 13570810254 Tel4: --- Tel5: ---
Inquiry password+A	Return all numbers
Delete password+A+series number	Return 0~4 or 5~9 numbers.

9) **Setup Daily Report time**

SMS Command	Return SMS Content
Setup password+DR+series number+T+time  <b>Notice:</b> Series number =0~9, e.g.: 1234DR1T12:30	Daily SMS Report at: xx:xx
Inquiry password+DR	
Delete password+DRDEL	

10) **Set Server Parameter(Can not setup DNS by SMS)**

SMS Command	Return SMS Content
Set Server IP password+IP+ IPaddress+P+Com port	Server: Port:
Inquiry password+IP	
Delete password+IPDEL	

11) **Set GPRS APN/USER NAME/PASSWORD**

SMS Command	Return SMS Content
Set password+AP+apn+#+username+#+userpassword	APN:
Inquiry password+AP	User Name:
Delete password+APDEL	Password:

12) **GPRS Online**

SMS Command	Return SMS Content
password+GPRSONline	GPRS always online

13) **Delete Historical Data**



SMS Command	Return SMS Content
password+HISDEL	Delete all historical records

## 10.S17X/S37X Register Address

**Tips:** All address in S17X/S37X Register Table displayed as Decimal data.

Read Input Register (Function Code 04: Read Input Register.)			
Register Address (Decimal)	Definition	Data Type	Description
14	Power supply voltage	16bit int	Y=X/100
15-23	Reserved, not available	---	---
24	Temperature	16bit int	Y=X/100
25	Humidity	16bit int	Y=X/100

**Tips:** In description, the parameter definition as below:

Y= Real value; X=The value stored in register;

"Y=X/100" means "Real value= Current value stored in register/100".

## 11.Slave Mapping Register Address

### Boolean Slave Register Assignment Table

Holding Coil (Function Code 01, Function Code 05, Function Code 15.)			
Boolean Register Address (Decimal)	Definition	Flag (MQTT)	Description
64	Boolean 64	REG64	Boolean type, slave mapping address, can mapping slave input coil and holding coil status.
65	Boolean 65	REG65	Same as above
.....	125 data similar as above	.....	Same as above
127	Boolean 127	REG127	Same as above

### 16 Bit Slave Register Assignment Table

Read and Write Holding Register (Function Code 03,Function Code 06, Function Code 16)				
16 Bit Register Address (Decimal)	Definition	Flag (MQTT)	Data Type	Description
20000	16 Bit data 20000	REG20000	Sort AB, its data type according to slave mapping data type	According to configurator set mapping rules, this address will sort slave mapping data to AB, stock in this address, for cloud easy reading together, can mapping slave inputting and holding register.
20001	16 Bit data 20001	REG20001	Same as above	Same as above
20002	16 Bit data	REG20002	Same as above	Same as above



	20002			
.....	124 data similar as above	.....	Same as above	Same as above
20127	16 Bit data 20127	REG20127	Same as above	Same as above

**32 Bit Slave Register Assignment Table**

Holding Register				
32 Bit Register Address (Decimal)	Definition	Flag (MQTT)	Data Type	Description
20196	32 Bit data 20196	REG20196	Sort ABCD, its data type according to slave mapping data type	According to configurator set mapping rules, this address will sort slave mapping data to ABCD, stock in this address, for cloud easy reading together, can mapping slave inputting and holding register.
20198	32 Bit data 20198	REG20198	Same as above	Same as above
20200	32 Bit data 20200	REG20200	Same as above	Same as above
.....	60 data similar as above	.....	Same as above	Same as above
20254	32 Bit data 20254	REG20254	Same as above	Same as above

**64 Bit Slave Register Assignment Table**

Holding Register				
64 Bit Register Address (Decimal)	Definition	Flag (MQTT)	Data Type	Description
20256	64 Bit data 20256	REG20256	Sort ABCDEFGH, its data type according to slave mapping data type	According to configurator set mapping rules, this address will sort slave mapping data to ABCDEFGH, stock in this address, for cloud easy reading together, can mapping slave inputting and holding register.
20260	64 Bit data 20260	REG20260	Same as above	Same as above
20264	64 Bit data 20264	REG20264	Same as above	Same as above
.....	60 data similar as above	.....	Same as above	Same as above



20508	64 Bit data 20508	REG20508	Same as above	Same as above
-------	----------------------	----------	---------------	---------------

**12. MQTT Flag and Cloud Platform**

**MQTT Introduction**

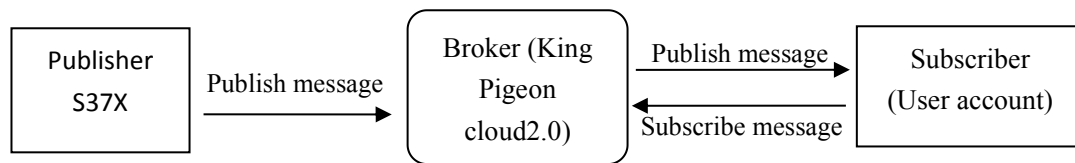
MQTT is a client-server based message publish/subscribe transport protocol. The MQTT protocol is lightweight, simple, open, and easy to implement, and these features make it very versatile. In many cases, including restricted environments such as machine to machine (M2M) communication and the Internet of Things (IoT). It is widely used in satellite link communication sensors, occasionally dialed medical devices, smart homes, and some miniaturized devices. The MQTT protocol runs on TCP/IP or other network protocols, providing ordered, lossless, two-way connectivity.

**MQTT Principle**

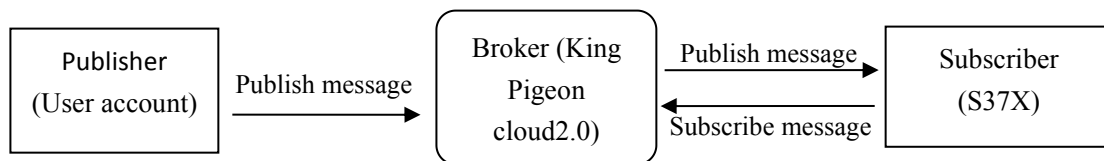
There are three identities in the MQTT protocol: Publisher (Publish), Broker (Server), Subscriber (Subscribe). Among them, the publisher and subscriber of the message are both clients, the message broker is the server, and the message publisher can be the subscriber at the same time.

Devices use MQTT communication through only two steps.

- 1.Devices publish the Topic through a proxy server;
2. Users can create a account on the proxy server to subscribe to the device to achieve monitoring.



(S37X uploads data to Broker)



(The S37X receives the downlink message from the Broker to implement control of the S37X)

The specific implementation process is as follows:

**1.Topic Publish of Devices**

**1)Device configuration**

- 1.Domain name: the domain name of the broker (King Pigeon Cloud 2.0 default: mqtt.dtuip.com)
2. Port: the port of the broker (King Pigeon Cloud 2.0 default 1883)
3. MQTT device ID: The unique Identification of the device, which can be the serial number, device ID, sensor ID or IMEI number and mac address of the module chip, ensuring uniqueness and distinguishing from other devices.

(King Pigeon 2.0 device default ID is serial number)



4.MQTT account: The account of device publishing the topic to the broker (King Pigeon cloud 2.0 default account: MQTT)

5.MQTT password: The account password of device publishing the topic to the broker ( King Pigeon cloud 2.0 default account password:MQTTPW)

## 2)Topic Publish Format

Topic: MQTT Device ID

Payload content is json data format

```
{
  "sensorDatas":
  [
    {
      "value":"10.0",           //Data type and value
      "flag":"AIN1"           //flag
    },
    {
      "lat":39.9,
      "lng":116.3,
      "flag":""
    },
    {
      "switcher":"1",
      "flag":"D11"
    },
    {
      "str":"1",
      "flag":"REG20014"
    }
  ]
}
```

### Note:

Each I/O point must contain three types of information when the device publish message: add Time, data type and value, flag;

// Data type and value: according to the type is divided into the following

1. The numeric character is "value" followed by: "data value"
  2. The positioning character is "lat" (longitude) followed by: "longitude value"  
"lng" (latitude) is followed by: "latitude value"
  3. The switch character is "switcher" followed by: "data value"
  4. The string character is "str" followed by: "data value"
- //Flag: the character is "flag" followed by " flag"

For example KingPigeon Cloud 2.0 , after the user subscribes to the device, each I/O port has a fixed flag when the device issues a message. After the flag ID is filled in, can read and write the value. The I/O port flag of the device on the cloud platform is default and no need to be processed.Each I/O port has a fixed flag when the device publish a message. The specific flags are as follows:

### Device own I/O Port

1) Temperature: The flag is "TEMP" and the data type is 16-bit type. The actual value is 2 decimal places, platform has been mapped.



Humidity: The flag is "HUMI" and the data type is 16-bit type. The actual value is 2 decimal places, platform has been mapped.

2)Power supply voltage: The flag is "EXTPWR" and the data type is 16-bit type. The actual value is 2 decimal places and the platform has been mapped.

### **Extend I/O port**

The flag of the extend I/O port is unified to "REGx" (x is the register address). For details of the flag, reference to [["Slave Mapping Register Address"](#)].

## **2. User topic subscription**

### **1) Subscribe to the device**

After the device publish the message on the proxy server, client can monitor the device through the subscription.

Subscription format: serial number / + (subscription topic needs to add the wildcard "/" after the serial number)

### **2) Delivery format of device control**

```
{
"sensorDatas":
[
{
"sensorsId": "211267", //cloud platform sensor ID
"value": "225", //switch type data, 0 is off, 1 is closed
"flag": "REG20116" // flag
}
],
"down": "down" // platform downlink message
}
```

#### **Note:**

The data sent by the device control must contain three types of information: sensor ID, data type, flag flag, and downlink message packet.

//Sensor ID: The character is "sensorsID", and the ID is automatically generated according to the platform definition.

// Data type and value: according to the type is divided into the following

1. The numeric character is "value" followed by: "data value"
2. The switch character is "switcher" followed by: "data value"
3. The string character is "str" followed by: "data value"

// Flag: the character is "flag" followed by "read and write flag"

// Platform downlink message: Confirm data is sent.

## **13. Upgrade Firmware**

The device supports upgrade firmware via USB port directly. If you required upgrade, please contact us to discuss and modify the firmware according to you requirements, we can provide the upgraded firmware to you to upgrade them.

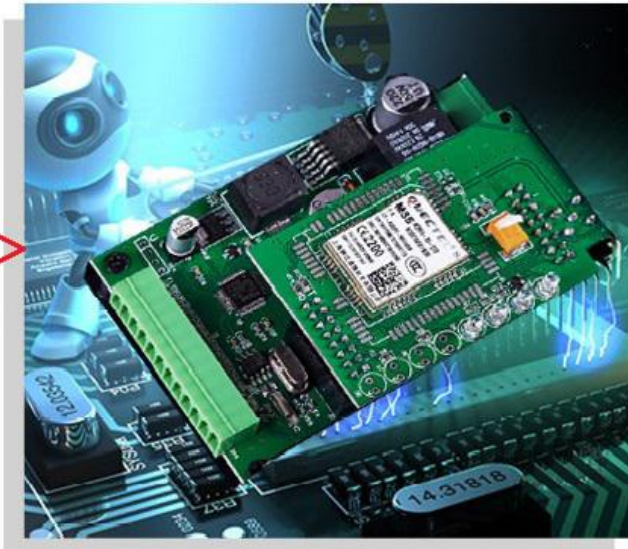
**14. Cellular Module Upgrade**

The device adopt modular structure design, when user local Gsm operator upgrade network, no need to replace the whole hardware, only need to replace inbuilt communication module, easily upgrade Gsm to 3G, or 3G to 4G network.

**Cellular Module Upgrade**

**Users can easily upgrade GSM (or 3G) to 3G/4G, NB-IoT or 5G network.**

*No need to replace whole device again when local network upgrade, only pick Gsm module out, put a 3G/4G module in, then device can support 3G/4G.*



**15. Warranty**

- 1) This system is warranted to be free of defects in material and workmanship for one year.
- 2) This warranty does not extend to any defect, malfunction or failure caused by abuse or misuse by the Operating Instructions. In no event shall the manufacturer be liable for any alarm system altered by purchasers

The End!

Any questions please help to contact us feel free.

[Http://www.IOT-Solution.com](http://www.IOT-Solution.com)