AMPLIFIED ENGINEERING

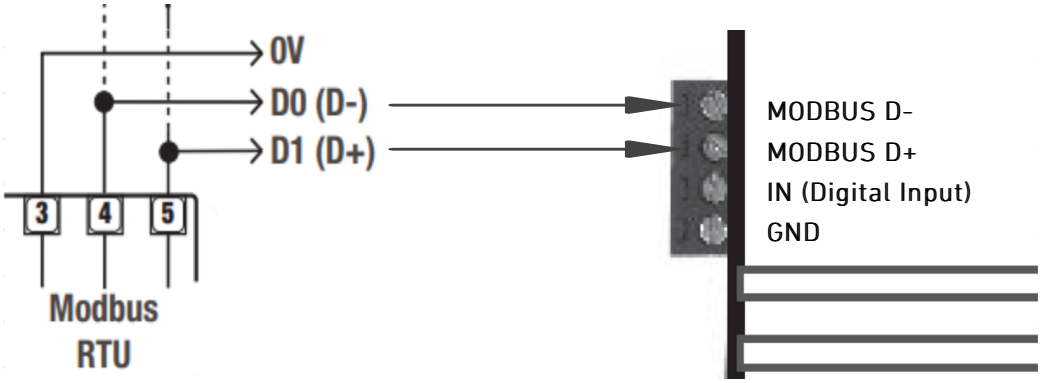
Intelligent M2M & IOT Connectivity

Connecting a Schneider energy meter to an IoT backend platform with the FATBOX G3 Gateway

Cloud monitoring energy usage via Modbus RTU

- In this tutorial, we will set up a Schneider iEM2150 Energy Meter as a **Modbus RTU slave** and interface to our FATBOX G3 gateway **Modbus master**.
- The gateway is able to support up to 32 Modbus devices, including meters and remote I/O terminal. The standard industrial Modbus/RTU protocol runs on Serial RS-485 interface, providing a robust & reliable interface.
- We will also look at **connecting the G3 IOT Gateway to a designated Cloud Service**. Users have the option of backhaul via either wired Ethernet , WIFI or even cellular (4G/3G) for remote sites/redundancy.

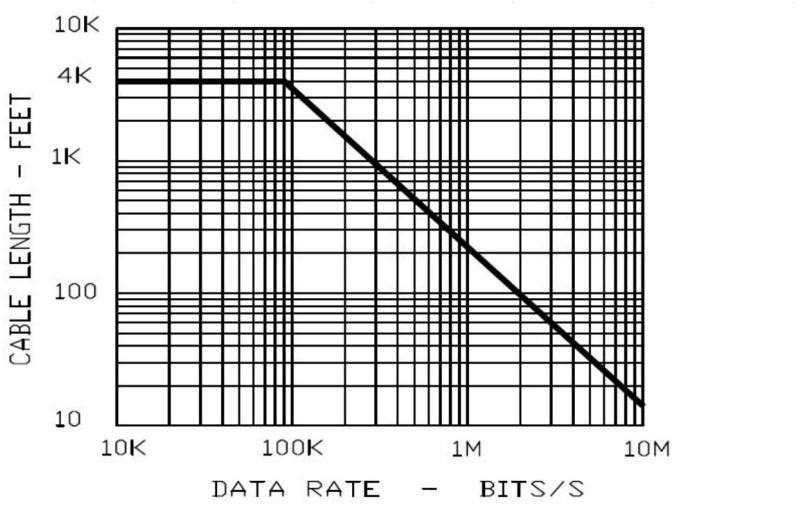
Hardware Wiring



Schneider iEM2150 *FATBOX G3*

Above shows the Serial RS-485 connection usually implemented using shielded twisted pair cable.

Note maximum cable length depends on baudrate and cable quality, e.g. using 24AWG shielded twisted pair, about 100Kbit/sec at 1000m.

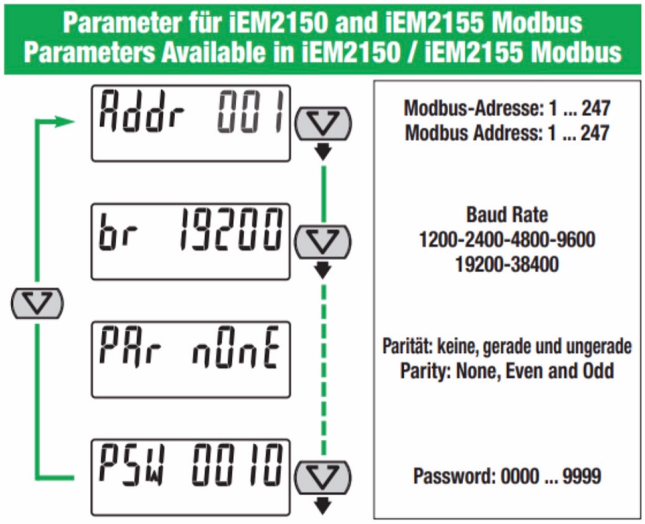


DATA SIGNALLING RATE VESUS CABLE LENGTH FOR BALANCED INTERFACE USING 24 AWG TWISTED PAIR CABLE

Image courtesy of eng-tip.com

Schneider Meter Configuration

Use the front display/button to set the serial parameters as below.



All slave devices connected on the RS-485 network must each have a unique address ranging from 1 to 247. Also, the baudrate and parity for all slave devices and master must be the same.

Connect Meter to Gateway

Our gateway uses a configuration file (\user\iotasset.txt) to map the required Modbus registers to be read during each polling cycle. This flexibility allows wide compatibility with most Modbus RTU devices from different manufacturers.

To write this iotasset.txt file we will be referencing the Modbus register table from the product user manual (see sample below):

Meter Data

Current, voltage, power, power factor and frequency

Register Address	Action (R/W/WC)	Size	Type	Units	Description
Current					
3000	R	2	Float32	A	Current
Voltage					
3028	R	2	Float32	V	Voltage
Power					
3054	R	2	Float32	kW	Active Power
3068	R	2	Float32	kVAR	Reactive Power
3076	R	2	Float32	kVA	Apparent Power

And here is a sample of the iotasset.txt. In the top example, we are configuring to read Current (A) register frm meter Modbus address 1, register 3000 (Schneider mandates a -1 offset) and name data field name, as "M1_Current".

The gateway is able to apply a customer multiplier (x0.1 in this case) and offset (0 in this case) to match the requirements of their cloud service.

```
MBM_START

TYPE,R
ADDR,1
MBFC,3
REGS,2999,2,FLOAT32ABCD,0.1,0
Key,M1_Current
Unit,A

TYPE,R
ADDR,1
MBFC,3
REGS,3027,2,FLOAT32ABCD
Key,M1_Voltage
Unit,V

TYPE,R
ADDR,1
MBFC,3
REGS,3053,2,FLOAT32ABCD
Key,M1_ActivePower
Unit,kW

MBM_STOP
```

Once you have configured your iotasset.txt file we will load it into the G3 gateway with the following steps:

For MAC users

Open a terminal session and run the following commands:

> **cd** <folder where you saved the iotasset.txt *[i.e."cd documents"]*>
> **scp iotasset.txt root@192.168.1.1:/user**
> <enter in the password for your G3 *[default is fatbox12345]*>

You can run the following commands to check that your iotasset.txt is loaded:

> **ssh root@192.168.1.1**
> <enter the password>
> **cd /user**
> **ls** *[check to see if there is a iotasset.txt file in the list]*
> **cat iotasset.txt** *[verify the contents of the iotasset.txt file]*

Go to the <Management> tab to **REBOOT** the G3 to save your settings.

For Windows Users

Go to the <Management> tab in the G3 web configuration menu. Enable the SSH option (see blow) and click the **UPDATE** button to save your settings.

Next run a file management program like Winscp and connect using the following File Protocol

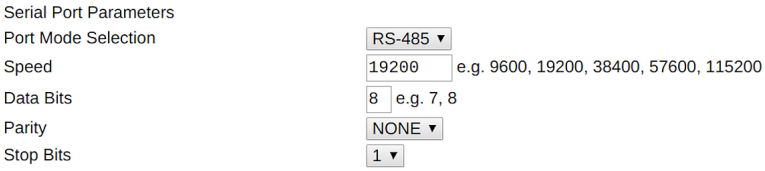
> **SCP Hostname: 192.168.1.1**
> **User/password: root/fatbox12345**
> Drag the iotasset.txt file over to the /user directory

You can run the following commands to check that your iotasset.txt is loaded:

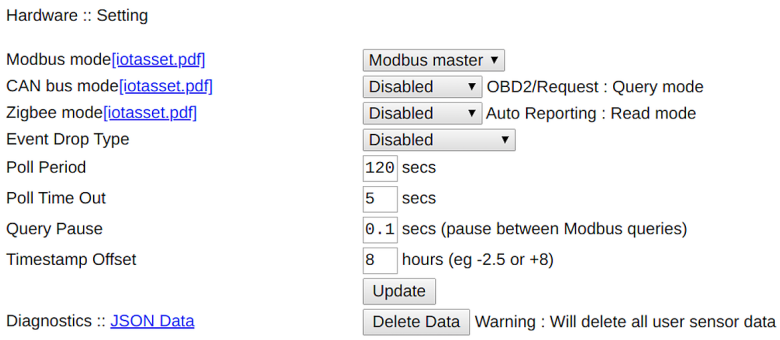
> **ssh root@192.168.1.1**
> <enter the password>
> **cd /user**
> **ls** *[check to see if there is a iotasset.txt file in the list]*
> **cat iotasset.txt** *[verify the contents of the iotasset.txt file]*

Go to the <Management> tab to **REBOOT** the G3 to save your settings.

The rest of the settings will be made using the G3's web configuration menu. Log into the G3 (the default address is 192.168.1.1) and enter in your user name and password. We will now configure the gateway's serial port to operate as required for the attached Modbus devices. Go to the *Port Settings* tab and enter the following settings.



Then, go to the *IOT Hardware* menu to set the Hardware interface to "Modbus master" and register the polling configuration as required by the user (Note that all other interfaces like CAN bus and Zigbee can also run concurrently).



After these settings are updated, **REBOOT** the gateway. The user can then check the Modbus sensor data collected (JSON Data) or delete for testing.

Connect to a Cloud Service

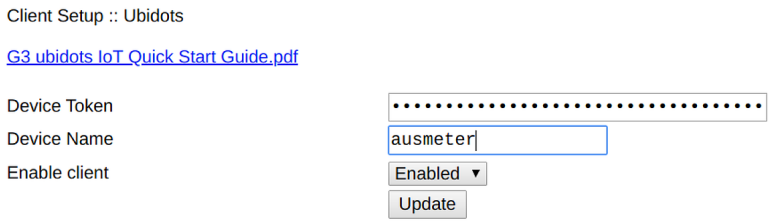
Now that the connectivity between the Schneider Meter and the G3 gateway has been set, we will then look at getting the data onto a designated cloud service. We support an open customer software and 3rd party API (e.g. REST API for HTTP, HTTPS and MQTT) integration to connect to a chosen cloud data or dashboard service (or to use for on board data processing).

Direct deployment on Azure or Ubidots can also be done as both these IoT clients have been integrated on the G3 Gateway. For this project, we are going to showcase using Ubidots as the setting is a straightforward inclusion of the secure **Device Token** from user's Ubidots account into the *IOT Client* menu.

To Obtain your Ubidots Device Token

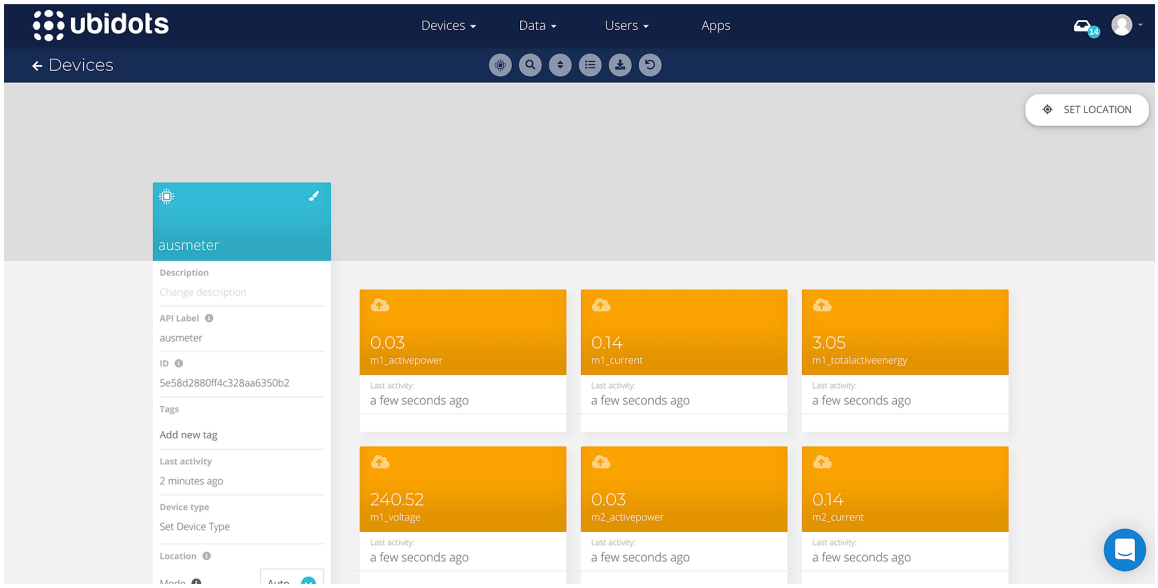
> Log in to your Ubidots account.
> Then go to <Users\Organizations> and create a new organization by pressing the plus (+) icon located at the upper right side of the platform.
> Enter the organization once it gets created.
> Next, go to the <Tokens> tab to create a new token pressing the plus (+) icon.
> Assign the token name of your preference, and press ACCEPT.
> Copy the token created.

Enter in your the Device Token and Device Name into the *IOT Client* tab on your G3 web configuration menu. Click on **UPDATE** to establish the connection to Ubidots Cloud Service.



Top

The FATBOX G3 will start sending the JSON data from the Schneider meter to Ubidots and you will be able to see your new FATBOX G3 created and listed under your 'Devices' in your Ubidots account.



Your Customised Reports Dashboard can now be easily built from the different register data (e.g. current from meter at Modbus address 1) using the friendly yet flexible Ubidots platform.