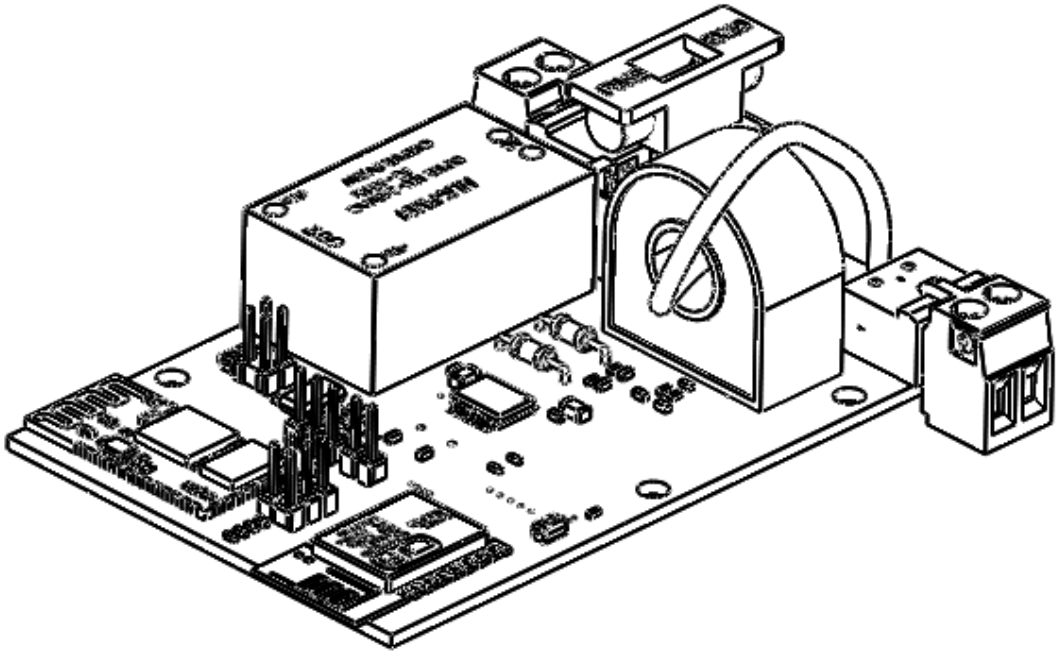
PQduino Board

The arduino based Power meter



User manual

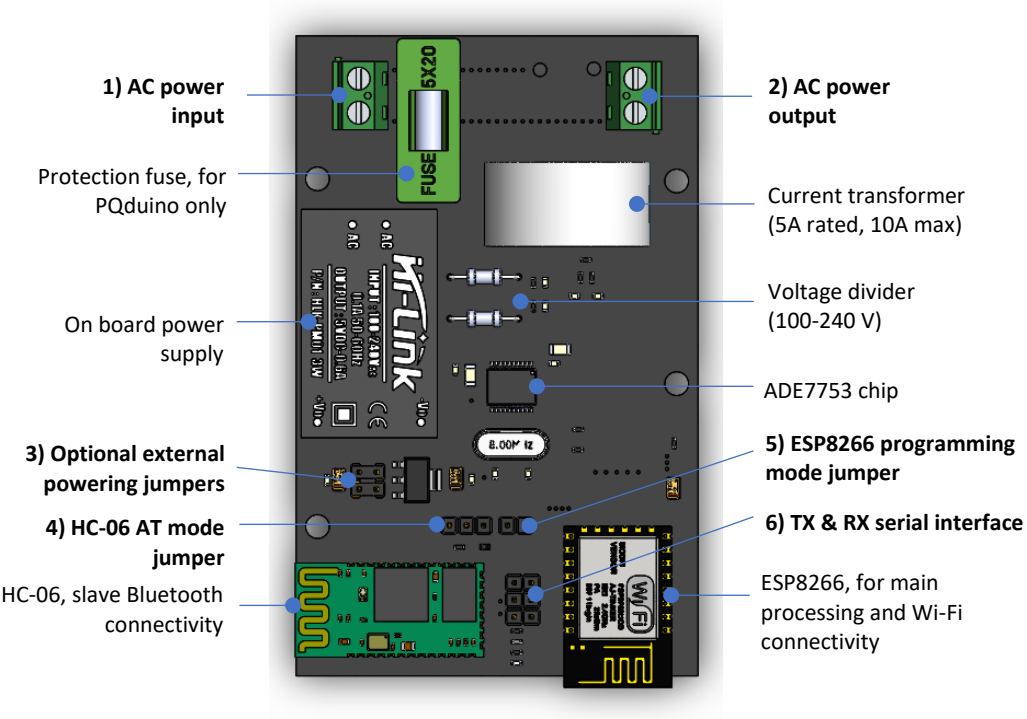
Content

1. Technical specifications

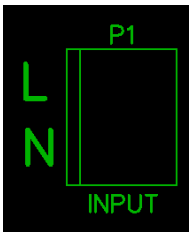
Overall dimensions:	
Height	95.3mm (3.75in)
Width	59.7mm (2.35in)
Deep	30.9mm (1.22in)
Weight	<100g
Operating temperature:	–40°C to +85°C
Power supply:	
Rated input voltage:	90-245 Vac
Input voltage range:	85-264 Vac
The maximum input voltage:	270 Vac
The maximum input current:	<150 mA
Typical input current:	11 mA (Wifi + BT connected)
Fuse protection:	200 mA, 250 Vac, 5x20mm
Measuring:	
Rated voltage measuring range:	10-270 Vac
Maximum voltage measuring:	<600 Vac (separation of power input and measuring pcb tracks is needed)
Rated current measuring range:	5 A
Maximum current measuring:	10 A
Possibility to measure big currents:	YES, please refer point 3h) in this manual
Measuring error	0.1%
Phase error	±0.05
Analog inputs maximum input:	0.5 Vpk max

2. Pinout

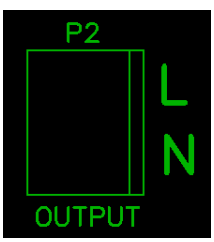
General reference:



1) **AC power input**. Connect here the main feeder entrance, L for Line and N for Neutral. Typically, you can use an AC plug. Use of insulated ferrules is required to connect the wires.

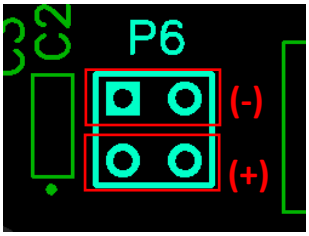


2) **AC power output**. Connect here the load you want to measure, L for Line and N for Neutral. Typically, you can use an AC outlet. Use of insulated ferrules is required to connect the wires.

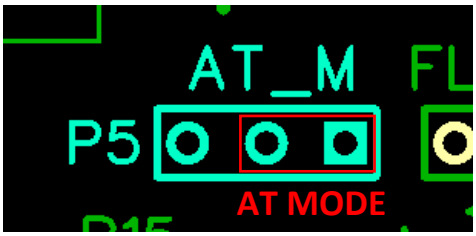


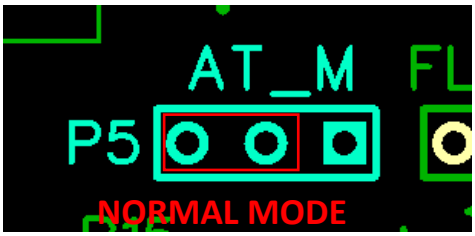
2. Pinout

3) **Optional external powering jumpers**. These jumpers are used to isolate the 5Vcd output from on board power supply, this is needed when you want to debug the PQduino code and the board stay unconnected from AC power system. To powering the board from external supply (USB outlet for example), you need to use the right side pins (- and +) and connect to 5 Vcd source.



4) **HC-06 AT mode jumper**. PQduino adds a Bluetooth module, it is used to establish debugging communication and for connectivity functions. This Bluetooth is also used for wirelessly flash the board.





PQduino can haves one of two these Bluetooth modules:

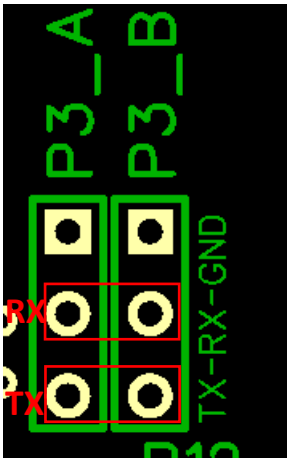
HC-06. If you want to change the name or password, set the jumper to AT MODE, in the right side. For normal use, set the jumper to left side or simply let floating. See point 3c) of this manual for establish connection in AT MODE.

SPP-C. This module stay in AT MODE while it is not connected to a Bluetooth master, you don't need to do anything in the AT_M jumper. See point 3c) of this manual for establish connection in AT MODE.

2. Pinout

5) **ESP8266 programming mode jumper**. If you need to upload a new code to PQduino Board you can use one of various ways to do that, FLASH jumper is used only when you need to upload the code with a universal UART converter, or with the Bluetooth module (wirelessly), **this jumper is not needed when you upload the code via OTA programing**. See point 3f) of this manual for flashing methods.

6) **TX & RX serial interface**. ESP8266 communicates with Bluetooth module via serial interface, however this interface can be used to add a screen or to flash a new code, or more. That's why these jumpers are enabled, in normal operation and flashing with the Bluetooth module these jumpers must be set. GND jumper is really unnecessary cause there is a direct connection between these pins.



3. Main functions

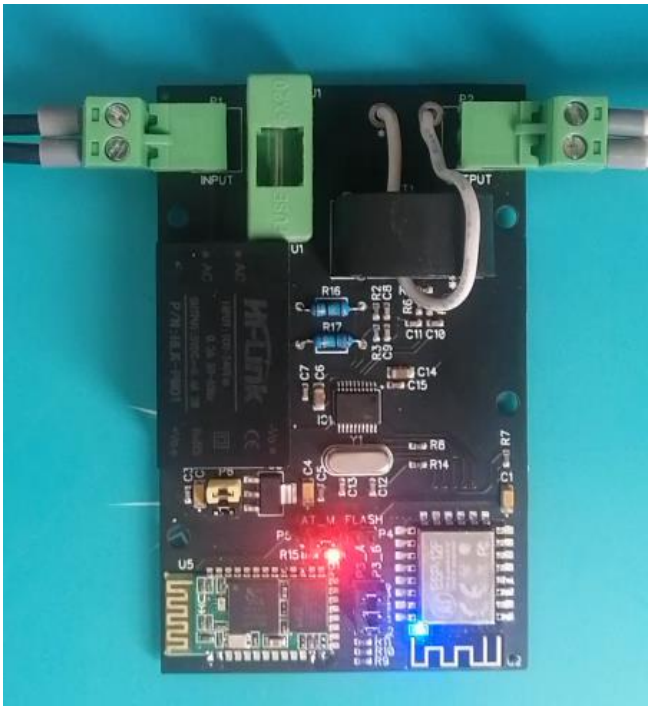
a) First power on

To star to using PQduino please follow these steps:

1. Connect a plug to input and outlet to output, fix the board in a secure place or enclosure that fit your needs. Connect the load you want to measure to the outlet, for example your laptop. Make sure the load current is less than 10 A if you are using PQduino without any modification (please refer 3h) in this manual).

   Use 14 AWG wire size and insulated ferrules to connect the HT5.08 male input and output connectors. We recommend the use of 10A thermomagnetic or fuse protection in the supply side.

2. Connect the plug to a right range voltage power supply. Led in ESP8266 blink once and then starts to blink repeatedly if a Wi-Fi connection was made successfully, but in the first time power on this led only blink once. Red Led of Bluetooth module starts to blink until you connect to it.



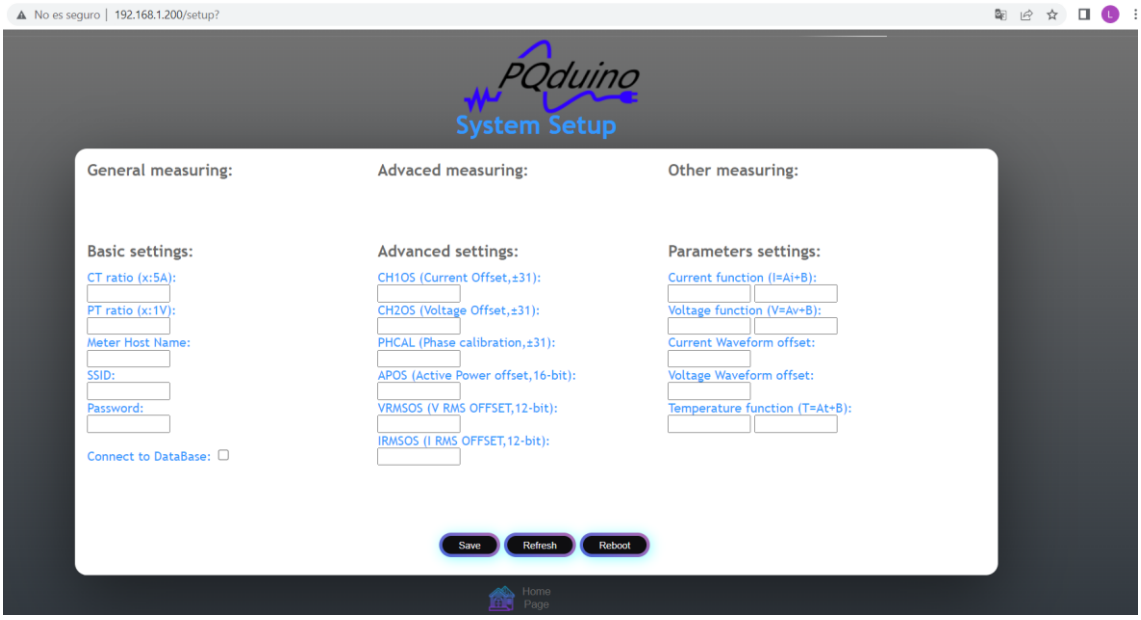
3. Main functions

Since PQduino will not be able to connect to a Wi-Fi network in the first power on, it will change to AP MODE (Access point Mode), and you can connect to ESP_Settings network, password for this network is **123456789**.

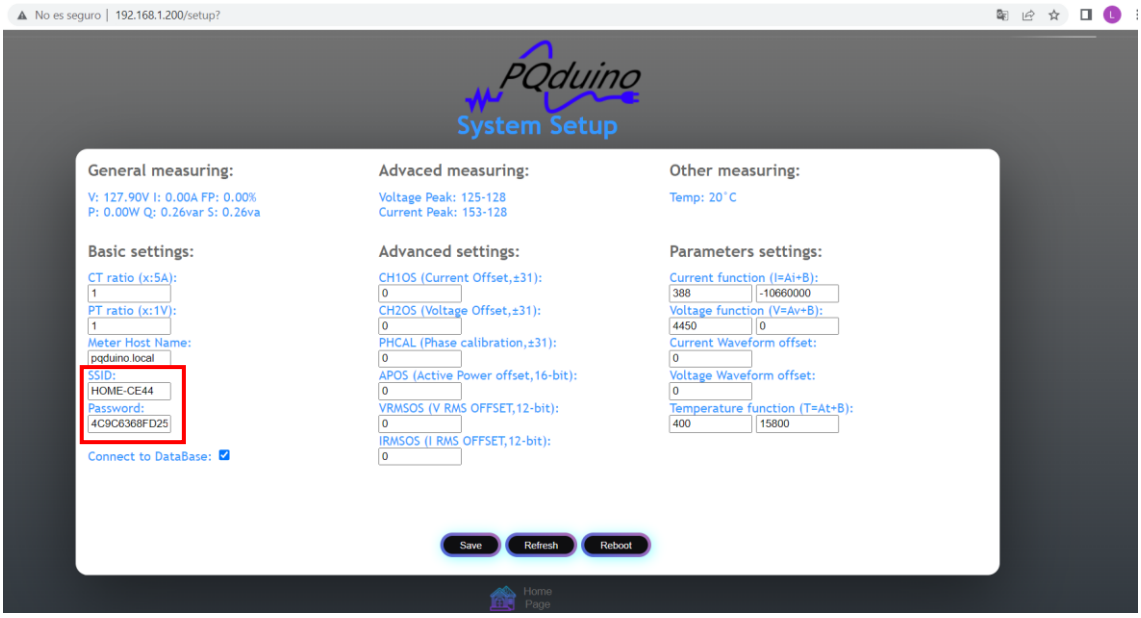
Once you are connected to ESP_Settings, go to **192.168.1.200** address to open the web server interface.

3. Main functions

Click in “System Setup” and this will open this tab:



Wait few seconds…



In this Settings page you can change the SSID and password to your own settings. Click “SAVE” button. Then you can click “REFRESH” button to make sure you have saved this change, at this point PQduino is still running in AP MODE. Click in “REBOOT” button and PQduino will restart, if everything goes ok PQduino will connect to your home Wi-Fi network.

You can now to check for IP assigned to PQduino and open that IP address in your PC, remember that you PC now must be connected to your home Wi-Fi network too.

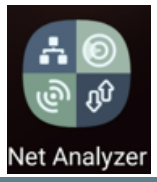
3. Main functions

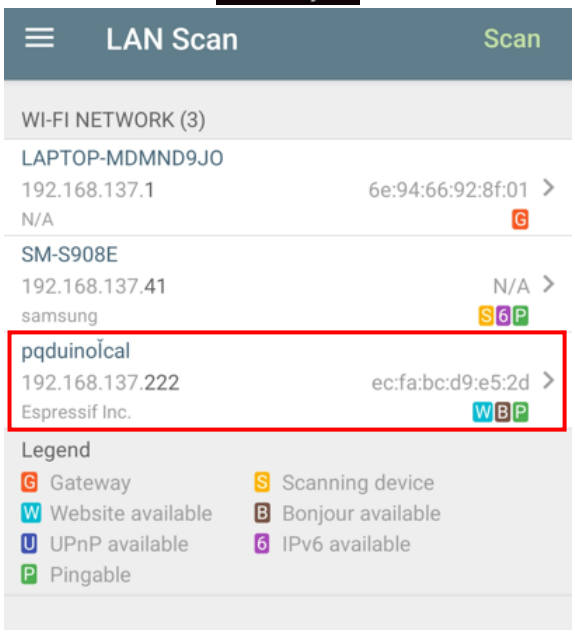
b) Establish communication

There are three basic ways to get real-time data from PQduino:

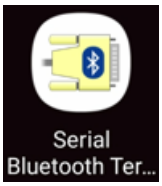
Wi-Fi.

Once you set your own SSID and password of your home Wi-Fi network and reboot PQduino, you can use this default host name "**pqduino.local**" to access to webserver, but sometimes MDNS doesn't work due to the modem configuration or something. So, you can use any tool like Net Analyzer (for Android) and scan to search for PQduino assigned IP.



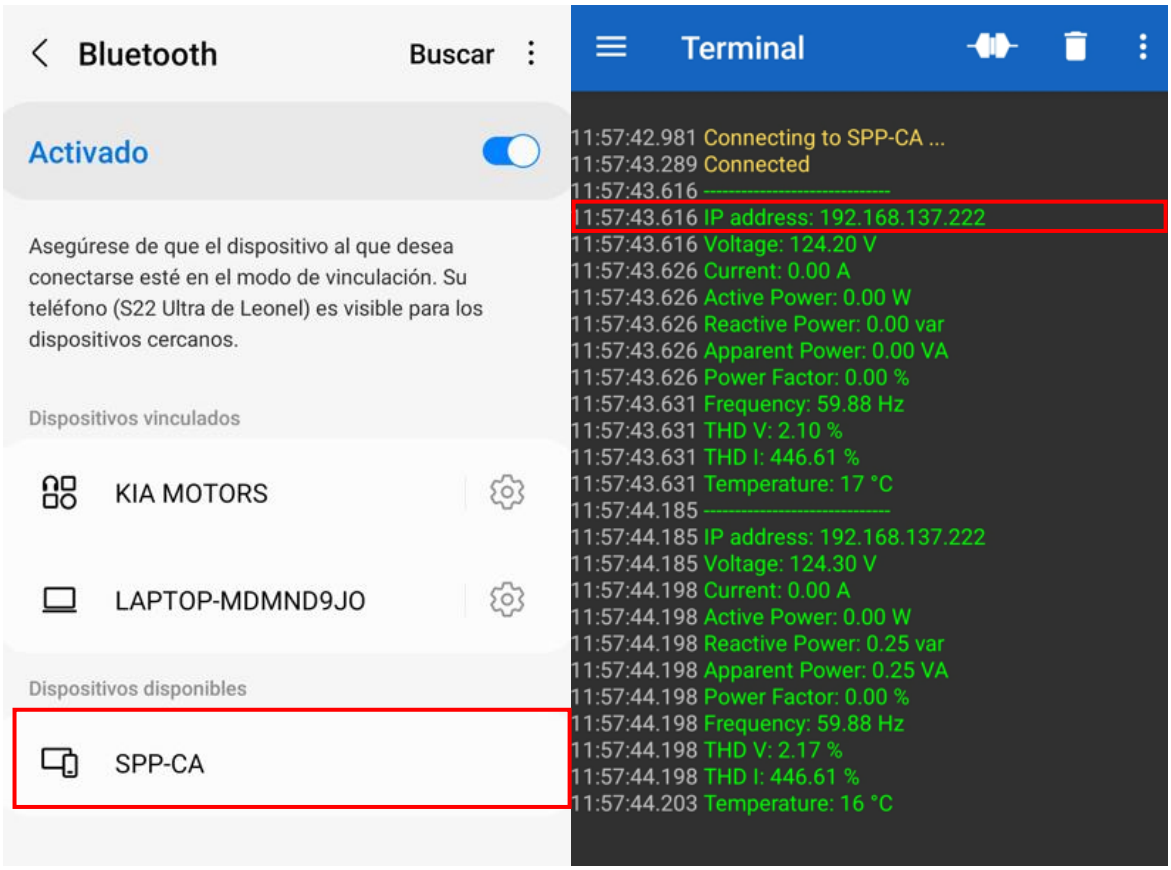


Other way to know IP is by Bluetooth, just connect to it (I use my phone to do this) and use a tool like Serial Bluetooth Terminal (for Android).



3. Main functions

Connect to PQduino, name can be “MyHomePowermeter”, “HC-06” or “SPP-CA”. Connect to it using password **4321** and open Serial Terminal. You will see IP in the collected data.

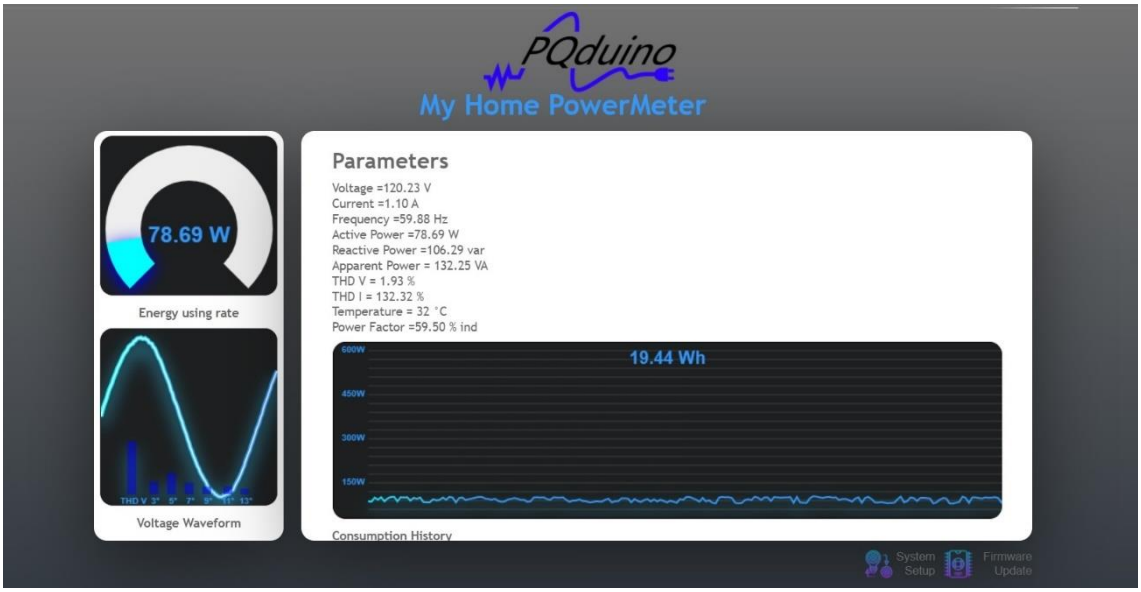


You can use PuTTY software for windows, or even a Serial Monitor of Arduino IDE, the serial speed is 115200. This is the link to video explaining this:

https://www.youtube.com/watch?v=PzobCweuAZY&t=14s

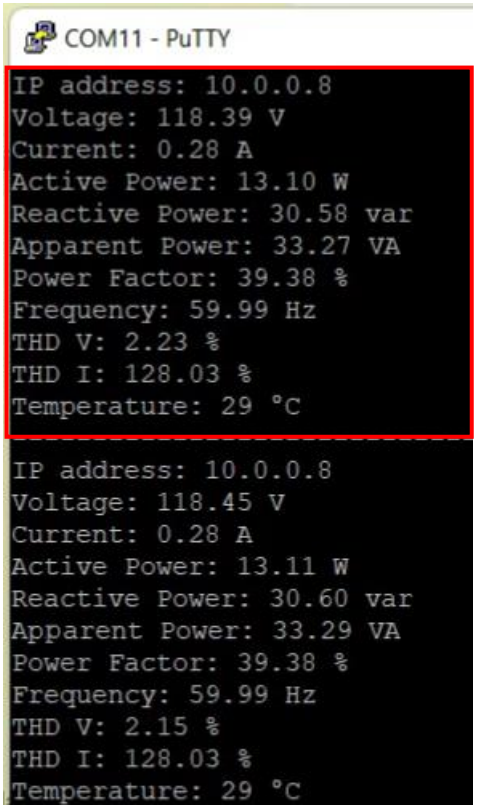
Once you know the IP just open it in a web browser, you will get data in the web server interface, PQduino sends data by a Websocket.

3. Main functions



Bluetooth.

PQduino also sends measured parameters by Bluetooth, just connect to it in the way explained before, and get the data. PQduino sends new data every 500ms, maybe you want to change that, it is possible by modifying the open-source code:



3. Main functions

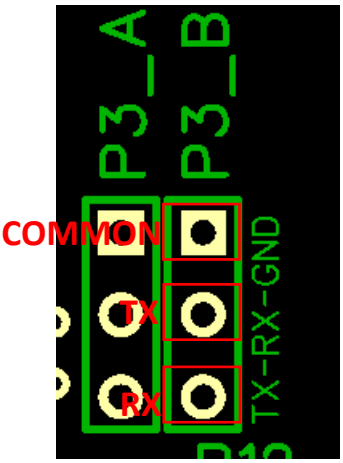
Serial Port.

You can connect physically to UART serial port in the board and send this data to another Arduino board for example that maybe will show the data in a screen, in fact, the possibilities are numerous. Connect **RX in board to TX of external module, TX in board to RX of external module**, GND in the board to GND in the external module. **Use right side pins of board.**

Speed is 115200 bauds, and operating voltage is **3.3V**, please make sure you are making right adequations if you will connect to a 5V external board.

You can change the string type that PQduino sends in order to facilitate serial process in the external board, like in this example:

https://github.com/PQduino/PQduino-Board/tree/main/Examples/ArduinoMEGA2560_BT-Client-forPQduino



c) Bluetooth settings

Before is explained the ways you can connect to Bluetooth in order to receive the data PQduino sends. The parameters of Bluetooth module are follows:

Mode:	Slave, (HC-06 is slave only, SPP-C is master/slave)
Operating voltage:	**3.3 V**
Speed	115200 bauds
Name:	MyHomePowermeter
Password:	4321

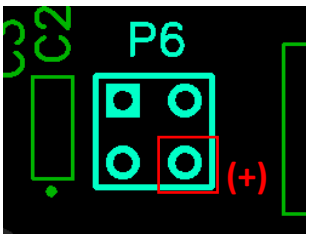
3. Main functions

If you want to change these parameters you will need to connect to TX and RX of Bluetooth module and enter in AT MODE.

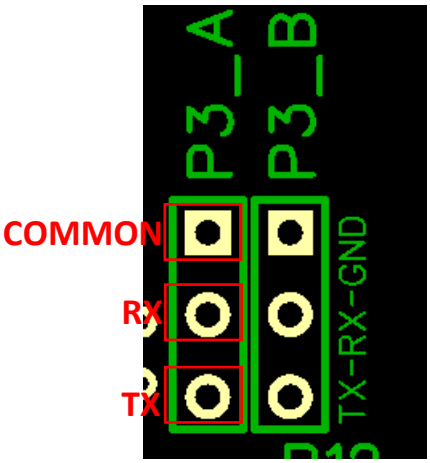
You will need an USB to TTL Serial 3.3V UART Converter.

Disconnect PQduino from AC power system.

Take out optional external powering jumpers and connect the 5Vcd pin of the converter to (+) pin in the right side of jumper.



Take out serial jumpers and **use left side pins in the board**. Connect **RX in board to RX of converter, TX in board to TX of converter**, GND in the board to GND in the converter.



Connect the converter to an USB port in your PC, identify the assigned port and open PuTTY or Serial Monitor in Arduino IDE, use a speed of 115200 and with No line ending, wait for 4 seconds and type “AT” in command and click send, you must receive “OK” as a response.

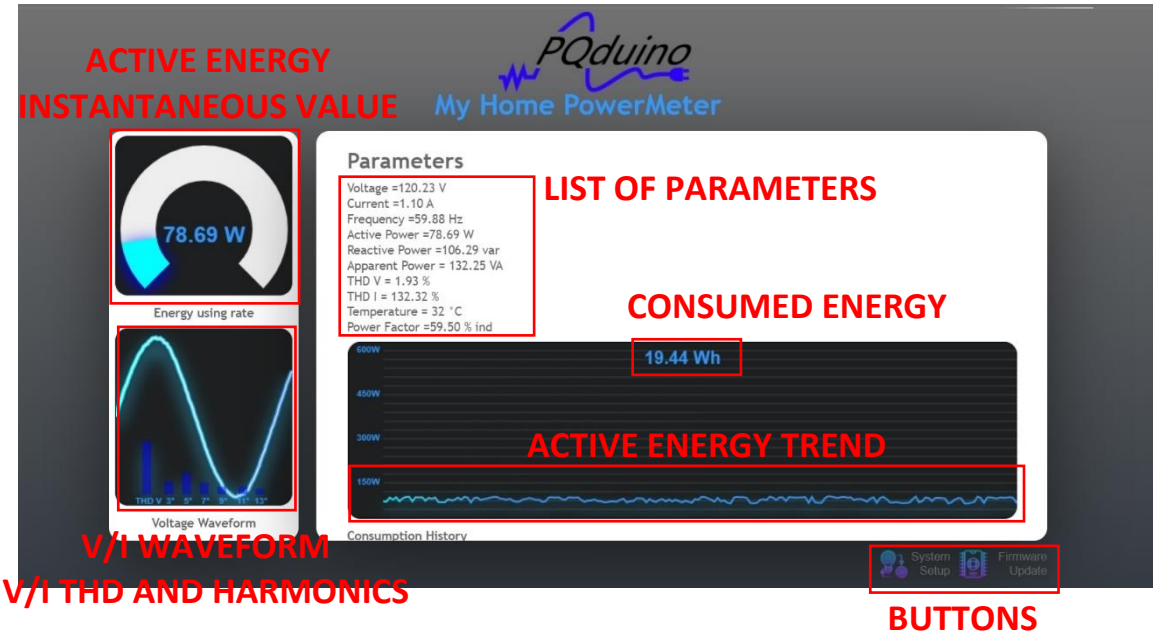
New Bluetooth modules have a default speed of 9600, but you receive a previously programmed module with a speed of 115200. If you change this speed you will need to update de code in ESP8266 with same speed too.

Send AT+PINxxxx to change password, xxx is the new password, you must receive OKsetPIN as a response.
Send AT+BAUDx to change speed, x is speed option (Note: 1 for 1200, 2 for 2400, 3 for 4800, 4 for 9600, 5 for 19200, 6 for 38400, 7 for 57600, 8 for 115200), you must receive something like OK9600 as a response.
Send AT+NAMExxxx to set the name of the module for example: AT+NAMEMyHomePowermeter, response is OK.

3. Main functions

d) Webserver

PQduino will show the measured parameters by a webserver and a websocket to update this data every 500ms. Webserver is accessible if you open the assigned IP in a web browser. Measured parameters are follows:



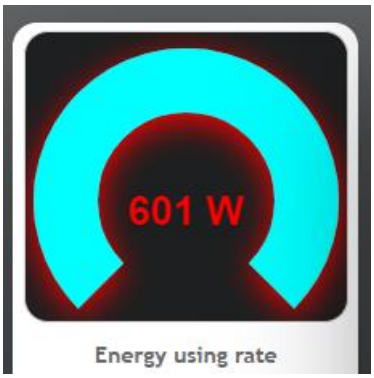
List of parameters. You get:

Voltage in Volts [V], this is RMS voltage level.
Current in Amperes [A], this is RMS apparent (active + reactive) current.
Frequency in Hertz [Hz], this is the fundamental frequency of voltage signal.
Active Power in Watts [W], this is the real power demand of your load.
Reactive Power in Volt-Ampere Reactive [var], this is the reactive power demand of your load, it can be capacitive (-) or inductive (+) depending of your load.
Apparent Power in Volt-Ampere [VA], this is the phasorial summatory of active and reactive powers.
THD V in percent [%]. This is Total Harmonic Distortion of voltage signal, normal values for home installations are under 2.25%, a value bigger than 3% is starting to get worrying for electronic loads.
THD I in percent [%]. This is Total Harmonic Distortion of current signal. Normal values for electronic loads like PC, TV, LED lamps, etc. are 80-180% (this is a really big distortion). Normal value for other loads like refrigerators, heaters, coffee makers, water pumps, etc. are 15-40%. THD I is what produce THD V in systems, and as a general rule if the current is low not matter if its distortion is big, it can't produce a problem.
Temperature in Celsius Degrees [°C], this is the temperature measured by ADE7753 internal sensor and can be different of ambient temperature.
Power Factor in percent [%], this is the Active Power/Apparent Power ratio, it can be capacitive or inductive depending of the load.

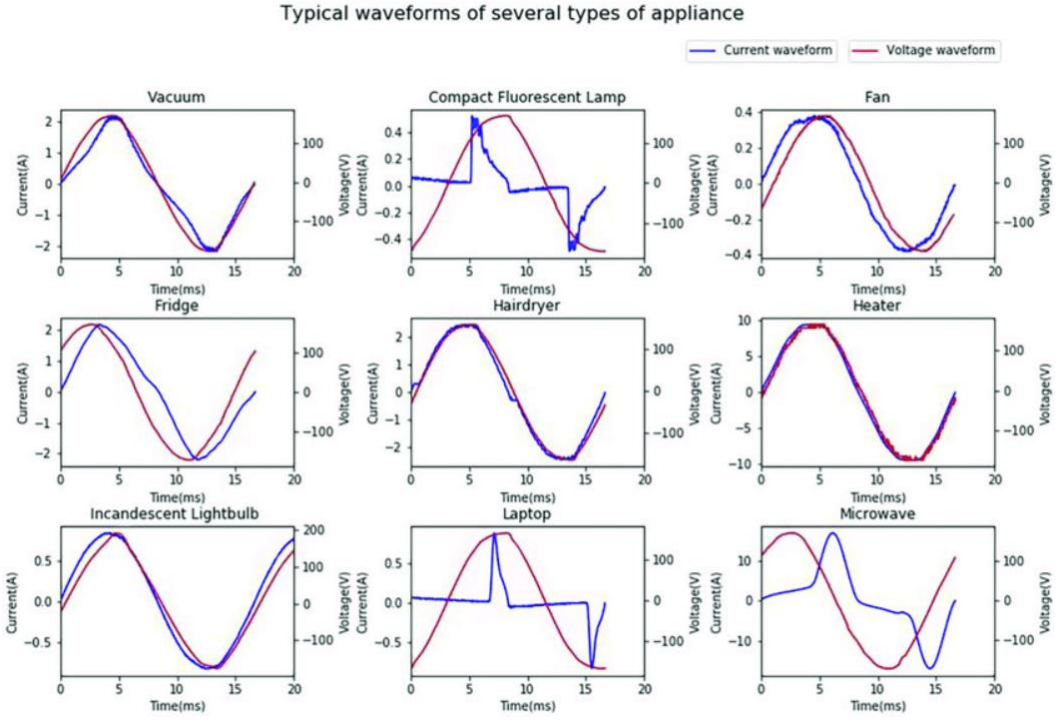
3. Main functions

Consumed energy. This is a counter of Active Energy in Watts-hour [Wh] that your load consume since PQduino starts to measure, it resets if PQduino resets. Active Energy is the time integration of Active Power demand [W]. What you will pay to supply company is this Active Energy but please notice the units.

Active Energy instantaneous value. This is again Active Power demand [W], but shown in a dial with a 600W maximum, if Active Energy is bigger than 600 W dial will show as and warning, you can edit this part in the open-source code:



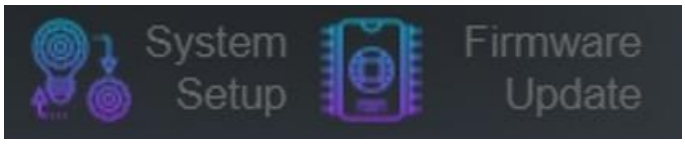
V/I Waveform. In this part is shown alternately the voltage and current waveforms (128 points), it changes every 4 seconds. Voltage always is a purer sinusoidal waveform cause THD is less, current can contain much bigger distortion and depending of the load, typical waveforms for some loads can be:



3. Main functions

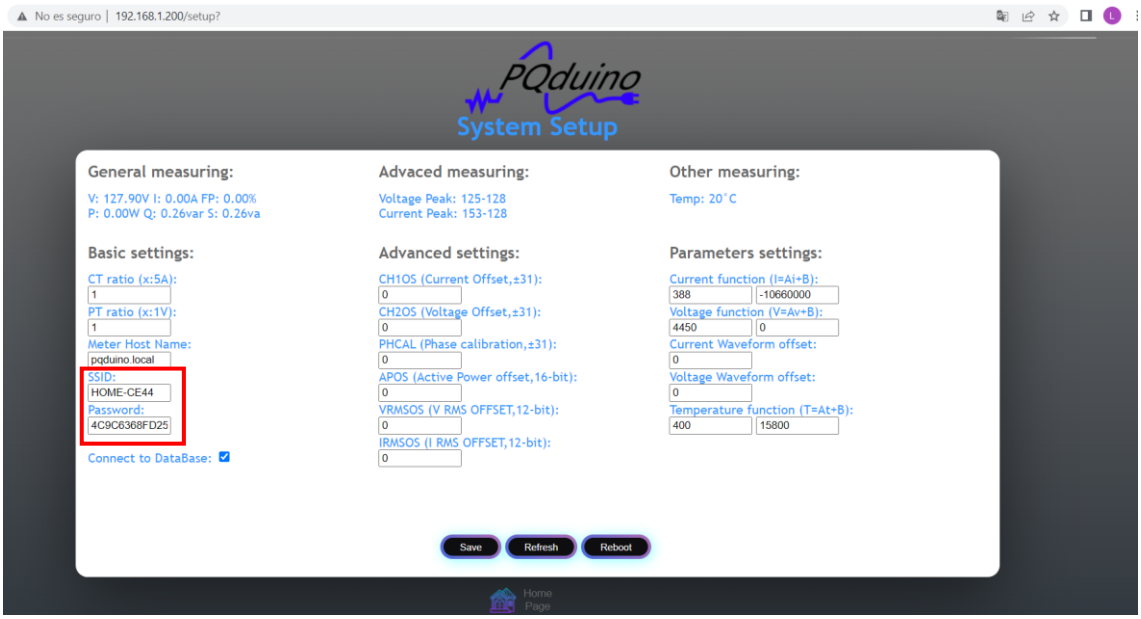
V/I THD and harmonics. In the same space a simple harmonic spectrum is shown, this is how THD is compound, only indicating first odd harmonics, PQduino calculates THD and harmonic components based in a DFT algorithm that can be modified to get more harmonic orders. When a voltage waveform is shown spectrum correspond to this voltage signal, when current waveform is shown, the spectrum is calculated to this current signal.

Buttons. Webpage can link to others that helps in additional functions, there is a System Setup and a Firmware Update buttons.



e) Webserver Settings page

If you click in System Setup button, you will get this page:



You can see basic parameters in order to can observe how changes made in the configuration impact the measured parameters, this without need of change to main page.

Basic settings.

CT ratio (x:5A). This is used to change the relation of primary current to secondary current, only if you are using an external Current Transformer (CT), please see point 3h) in this manual.

PT ratio (x:1V). This is used to change the relation of primary voltage to secondary voltage, only if you are using an external/additional Potential Transformer (PT) for voltage measuring, this is not a common situation.

3. Main functions

Meter Host Name. This is the name used for MDNS (multicast DNS), if your Wi-Fi Network can work with this you can access directly to webserver only typing this name in a web browser even unknowing the IP of PQduino.
SSID. SSID of your home Wi-Fi network.
Password. Password of your home Wi-Fi network.
Connect to database. This enable or unable connection to FIREBASE real-time database, this is an important and very useful function to monitor and save historical measured data. please see point 3g) in this manual.

Advanced settings.
These settings will affect parameters calculation inside ADE7753, and are used when exist any deviation or error in measuring.
CH1OS (Current Offset,±31). This is a register that allows channel 1 (current channel) offset in ADE7753 chip, this is used for calibration in measured value, we made this calibration previously we sent to you the board. Only can take values between ±31.
CH2OS (Voltage Offset,±31). This is a register that allows channel 1 (current channel) offset in ADE7753 chip, this is used for calibration in measured value, we made this calibration previously we sent to you the board. Only can take values between ±31.
PHCAL (Phase calibration,±31). This register allows you to reduce phase error between channels in ADE7753 chip, these errors can be introduced for using external CTs or PTs, we made this calibration considering the internal CT previously we sent to you the board. Only can take values between ±31.
APOS (Active Power offset,16-bit). This register is used to reduce Active Power offset error in ADE7753 chip, normally this is no necessary cause ADE7753 integrates a Low Pass Filter that deletes this error, an offset could exist in the power calculation due to crosstalk between channels on the PCB or in the IC itself. We made this calibration previously we sent to you the board. Only can take values between ±32760.
VRMSOS (V RMS OFFSET,12-bit). This register deletes RMS voltage calculation offset error in ADE7753 chip, we made this calibration previously we sent to you the board. Only can take values between ±2046.
IRMSOS (I RMS OFFSET,12-bit). This register deletes RMS current calculation offset error in ADE7753 chip, we made this calibration previously we sent to you the board. Only can take values between ±2046.

Parameters settings.
These settings will affect parameters calculation inside ESP8266, not inside ADE7753, and are used when advance settings can't correct any deviation or error in measuring. In GITHUB repository there is an excel file that you can use for calculate right constants.

https://github.com/PQduino/PQduino-Board

Current function (I=Ai+B). These parameters are used for adjust current value from ADE7753 to real value, A is a proportional constant like a slope of a line, B is offset constant.
Voltage function (V=Av+B). These parameters are used for adjust voltage value from ADE7753 to real value, A is a proportional constant like a slope of a line, B is offset constant.
Current Waveform offset. This is an offset corrector for current waveform.

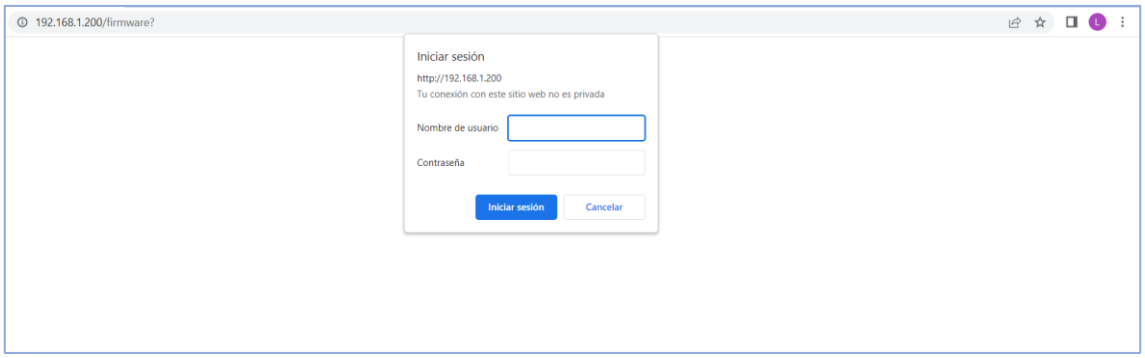
3. Main functions

Voltage Waveform offset. This is an offset corrector for voltage waveform.
Temperature function (T=At+B). These parameters are used for adjust temperature value from ADE7753 to real value, A is a proportional constant like a slope of a line, B is offset constant.

Click “SAVE” button to save any change you do. Then you can click “REFRESH” button to make sure you have saved these changes, you will see now how your changes in settings affect measured parameters. Click in “HOME PAGE” button and PQduino will back to main page.

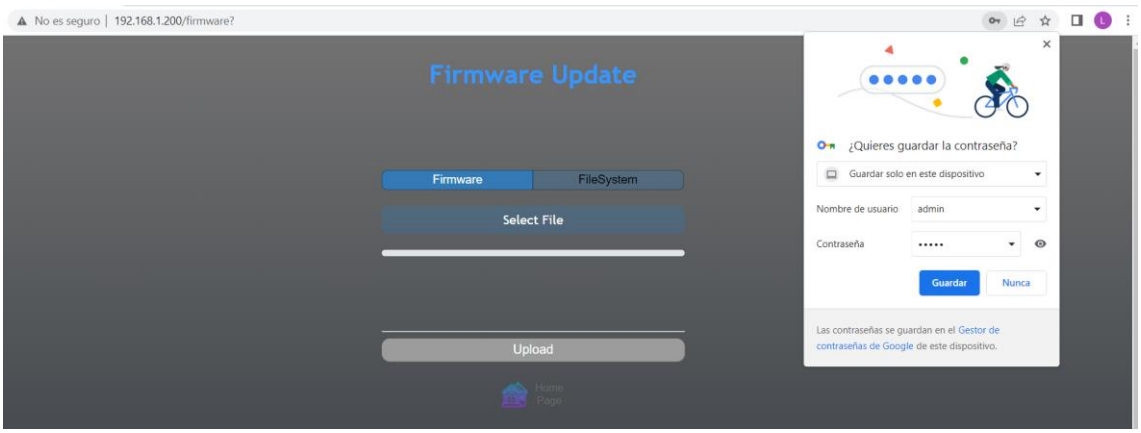
f) Upgrading the code (firmware)

There are different ways to change/upgrade the PQduino code, one is through the **OTA programming** with Firmware Update page. **You don’t need to disconnect PQduino from AC power system.** If you click in Firmware Update button, you will get this page:



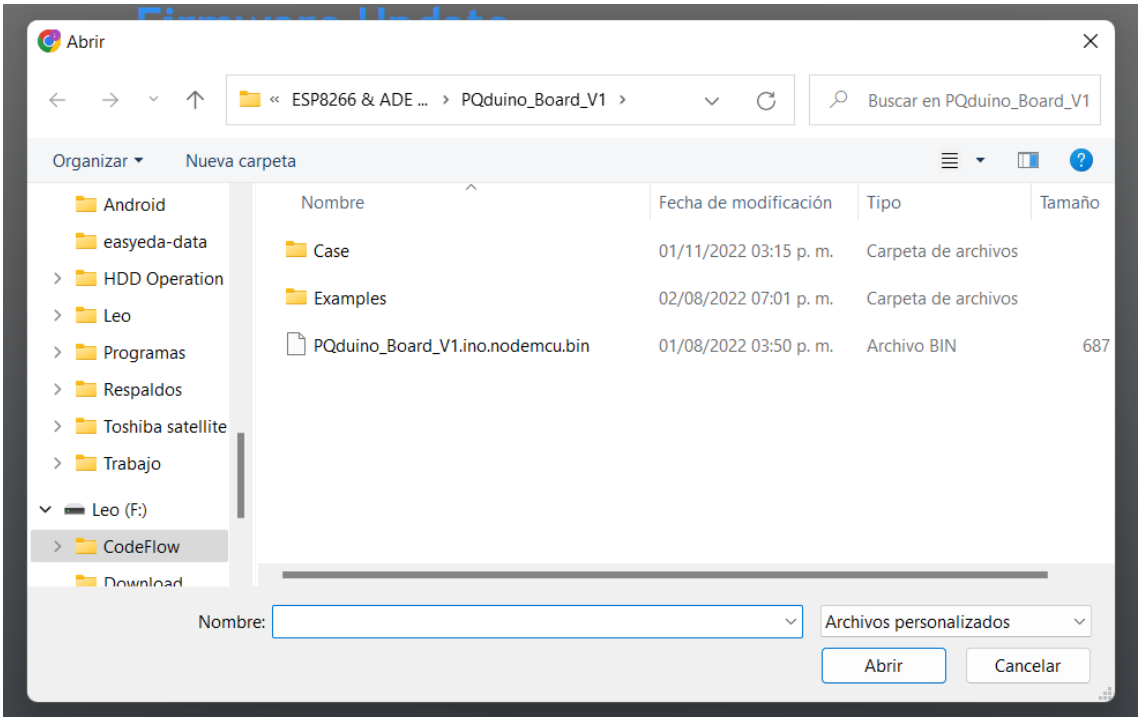
Login with:
user: **admin**
password: **admin**

You will see this page:

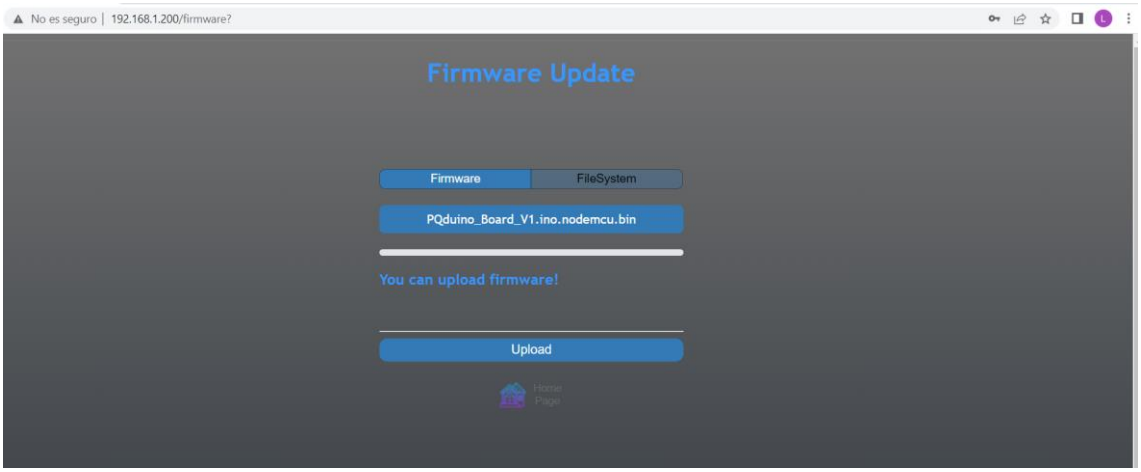


3. Main functions

Click in “Select File” and select binary file that you get from Arduino IDE by going to Sketch > Export compiled Binary.

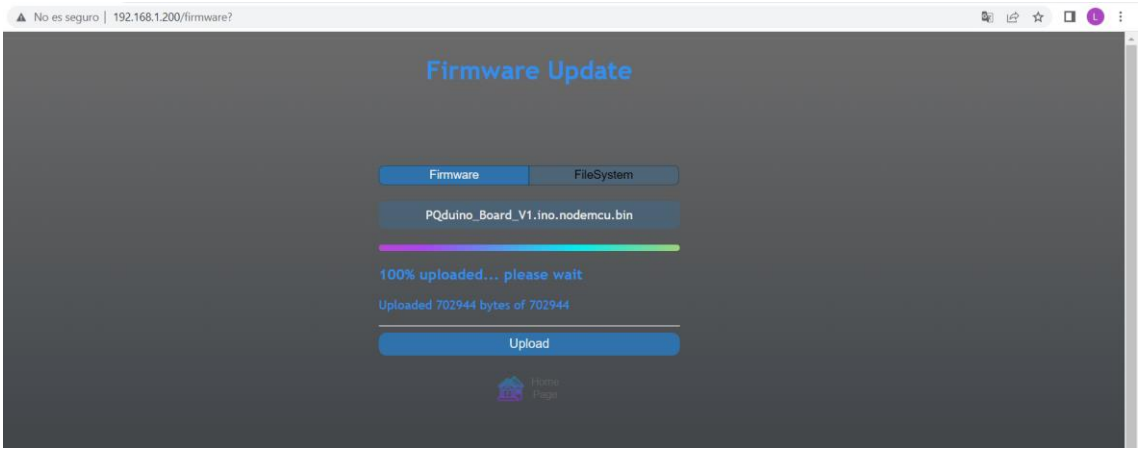


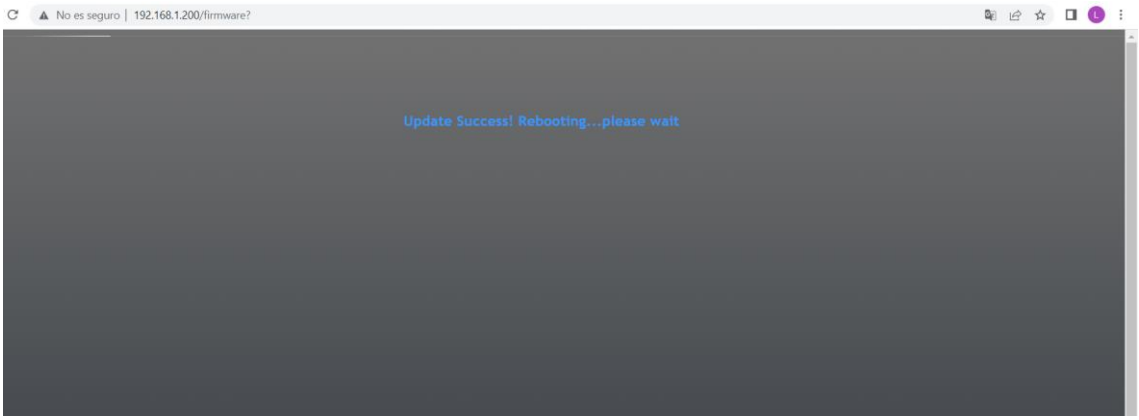
Click in “Upload”.



3. Main functions

Wait until PQduino reboot automatically, maybe you need to type IP address again to go to the Home Page. One thing you need to consider is: if new code doesn't work well and ESP8266 enter in a reset loop, you will not have possibility to upload again a firmware by OTA and you will must program the ESP8266 by other method.

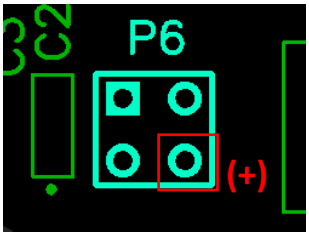




You can upgrade the code physically via **UART serial** port in the board. You will need an USB to TTL Serial 3.3V UART Converter.

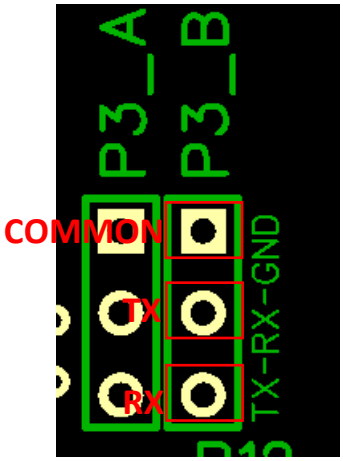
Disconnect PQduino from AC power system.

Take out optional external powering jumpers and connect the 5Vcd pin of the converter to (+) pin in the right side of jumper.



3. Main functions

Take out serial jumpers and **use right side pins in the board**. Connect **RX in board to TX of converter, TX in board to RX of converter**, GND in the board to GND in the converter.



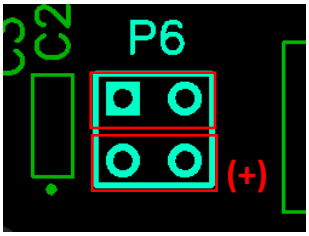
Set de FLASH jumper to enter in programming mode.

Connect the converter to an USB port in your PC, the board turns on, identify the assigned port and open Arduino IDE, select NodeMCU 1.0 (ESP-12E Module) in boards and the click compile and update.

If you don have an UART converter, you can upload a new code by **Bluetooth**, this is easier and must of time you can use this method to test code that maybe will go to reset loop.

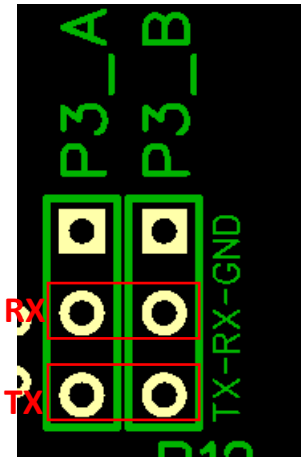
First, disconnect PQduino from AC power system.

Make sure the optional external powering jumpers are set.



3. Main functions

Make sure serial jumpers are set. GND pins don't need to have a jumper because there is a direct connection between they.



Set de FLASH jumper to enter in programming mode.

Then, connect PQduino to AC power system. PQduino turns on but enter un programming mode or flash mode, so it will not send any data how it is normally.

Open Arduino IDE, select the right port, select NodeMCU 1.0 (ESP-12E Module) in boards and the click compile and update. Make sure the speed is set to 115200 because this is the Bluetooth speed.

When uploading finish, you need to disconnect PQduino from AC power system.

Then, you can take out the FLASH jumper and connect PQduino to AC power system again.

3. Main functions

g) Firebase Real-time database

PQduino can send data to a firebase cloud service, in this way you can send data continuously and store it with a **timestamp** in a real-time database.

We prepare a webpage as a client to connect to database and shows a trend (highcharts library) of selected parameters, we set the PQduino code to send only these parameters every two minutes:

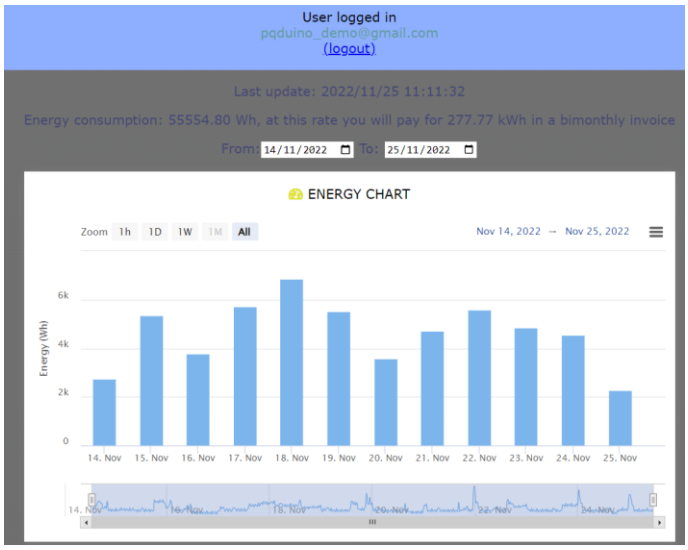
Energy. Energy consumed during this period of two minutes, in the page these values are shown as a bar graph with different times datagrouping options, so you can view the summatory of energy consumed for minutes, for hours, for days, for weeks and more.

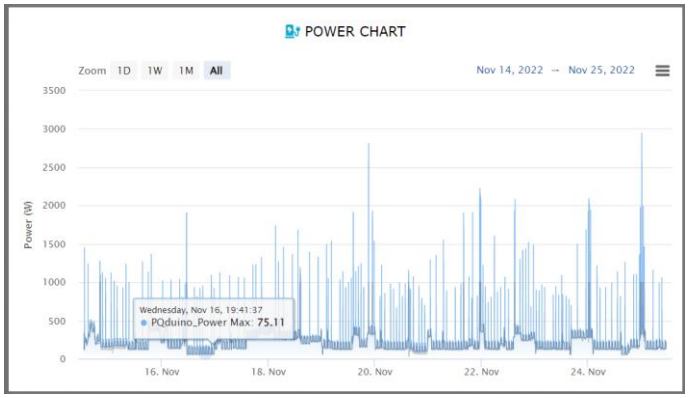
Active power maximum and minimum. Cause power takes many values between these two minutes, we select the minimum and the maximum values that power takes in this period and send these two values to database, we store each value in a different variable in database.

Voltage maximum and minimum. Same thing but for maximum and minimum values of voltage.

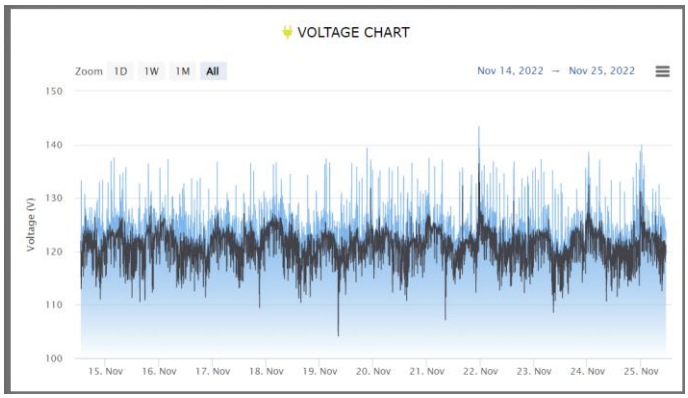
Current maximum and minimum. Maximum and minimum values of current.

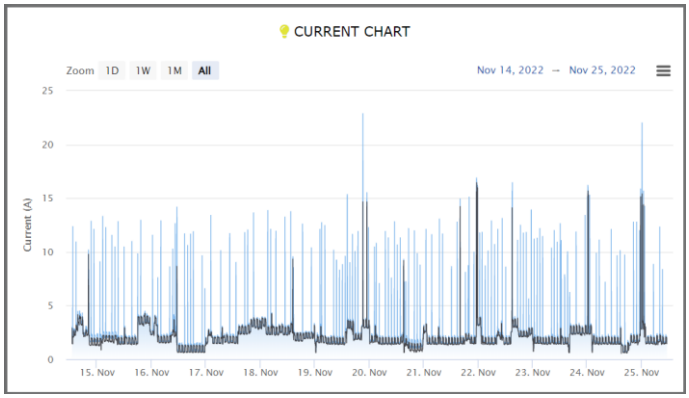
Temperature. We send only one value of temperature cause this parameter is very stable.

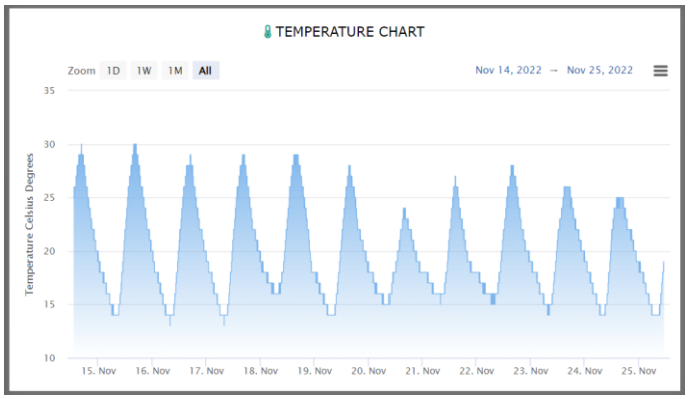




3. Main functions







You can download the client webpage with sample measured data in this link, version 2.0 is which was explained here. Version 1.0 only haves a single value for each parameter but in this way you lost some information about real behavior of parameters.

Firmware V1 of PQduino Board uploads data for V1.0 webpage and firmware V2 of PQduino Board uploads data for V2.0 webpage. Download the sample page and login with this user and password:

Firmware Version	IoT webpage client	Link	Login with:
PQduino_Board_V1.ino	V1.0	PQduino IoT v1.0.html	User: pqduino_demo@gmail.com Password: PQS2022
PQduino_Board_V2.ino	V2.0	PQduino IoT v2.0.html	

To enable data sending in PQduino you must enable "Connect to database" checkbox in System Setup page. See 3e) point of this manual.

But before you do that, please change PQduino code or you will upload data to our samples database. You can follow this excellent tutorial to create an account and a new project database to upload your own measurings:

https://randomnerdtutorials.com/esp8266-nodemcu-firebase-realtime-database/

3. Main functions

In PQduino code (Arduino IDE) you need to change this, with you own values:

```
// Change this for use your database:
//------------------------------------------------------------------------
#define API_KEY "AIzaSyC4DxVKXS0Ol_QZh0xGkx7scvNv50KWpJo"
#define USER_EMAIL "pqduino_test@gmail.com"
#define USER_PASSWORD "PQtest2022"
#define DATABASE_URL "https://pqduino-sample-default-rtdb.firebaseio.com"
#define DATABASE_SECRET "RRONfqEPzdx6HYB8OicgYTZ5Fuyyx1wWulqcxddb"
//------------------------------------------------------------------------
```

And in the PQduino IoT client webpage also you need to change this, you can open the page file in text editor to modify it:

```
// REPLACE WITH YOUR APP CONFIG IF YOU WANT TO MAKE YOUR OWN DATA LOGGER
  const firebaseConfig = {
    apiKey: "AIzaSyC4DxVKXS0Ol_QZh0xGkx7scvNv50KWpJo",
    authDomain: "pqduino-sample.firebaseapp.com",
    databaseURL: "https://pqduino-sample-default-rtdb.firebaseio.com",
    projectId: "pqduino-sample",
    storageBucket: "pqduino-sample.appspot.com",
    messagingSenderId: "577442550306",
    appId: "1:577442550306:web:d454e599bcec2b00a7707b",
    measurementId: "G-1MEZZ2FF6T"
  };
```

And in this function:

```
  auth.onAuthStateChanged(user => {
    if (user) {
      console.log("user logged in");
      console.log(user);
      setupUI(user);
      var uid = "iLPTysyT7QakXYb5MsYdhvEz5Vo2"; // Change this line to:        var uid = user.uid;
      console.log(uid);
    } else {
      console.log("user logged out");
      setupUI();
    }
```

And also in this part:

```
// get user UID to get data from database
    var uid = "iLPTysyT7QakXYb5MsYdhvEz5Vo2"; // Change this line to:    var uid = user.uid;
  console.log(uid);
```

h) Modifying current measuring range.

If you want to measure an extended current range you can use these methods:

a) Using an external CT.
You will need a Current Transformer (CT) with 5A or 1A secondary.

This is important, there are some devices called transducers that delivery a voltage signal as a secondary. These devices will not work in this method, we need a CT that delivery a current signal (5A or 1A) as a secondary. There are many CT types, use the correct for your needs.

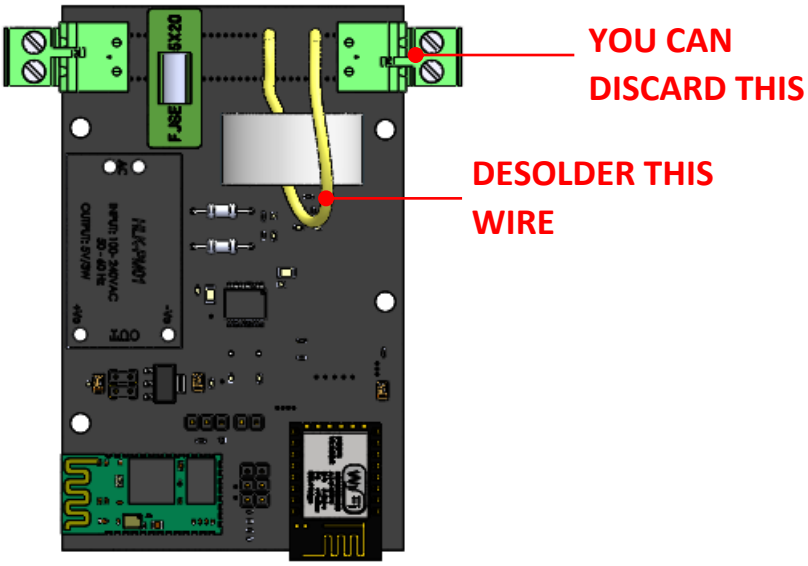
ONE OF MOST IMPORTANT THINGS IS YOU NEED TO MAKE SURE THE SECONDARY WIRES MUST NEVER BEEN OPEN WHEN A CURRENT IS PASSING IN PRIMARY SIDE.

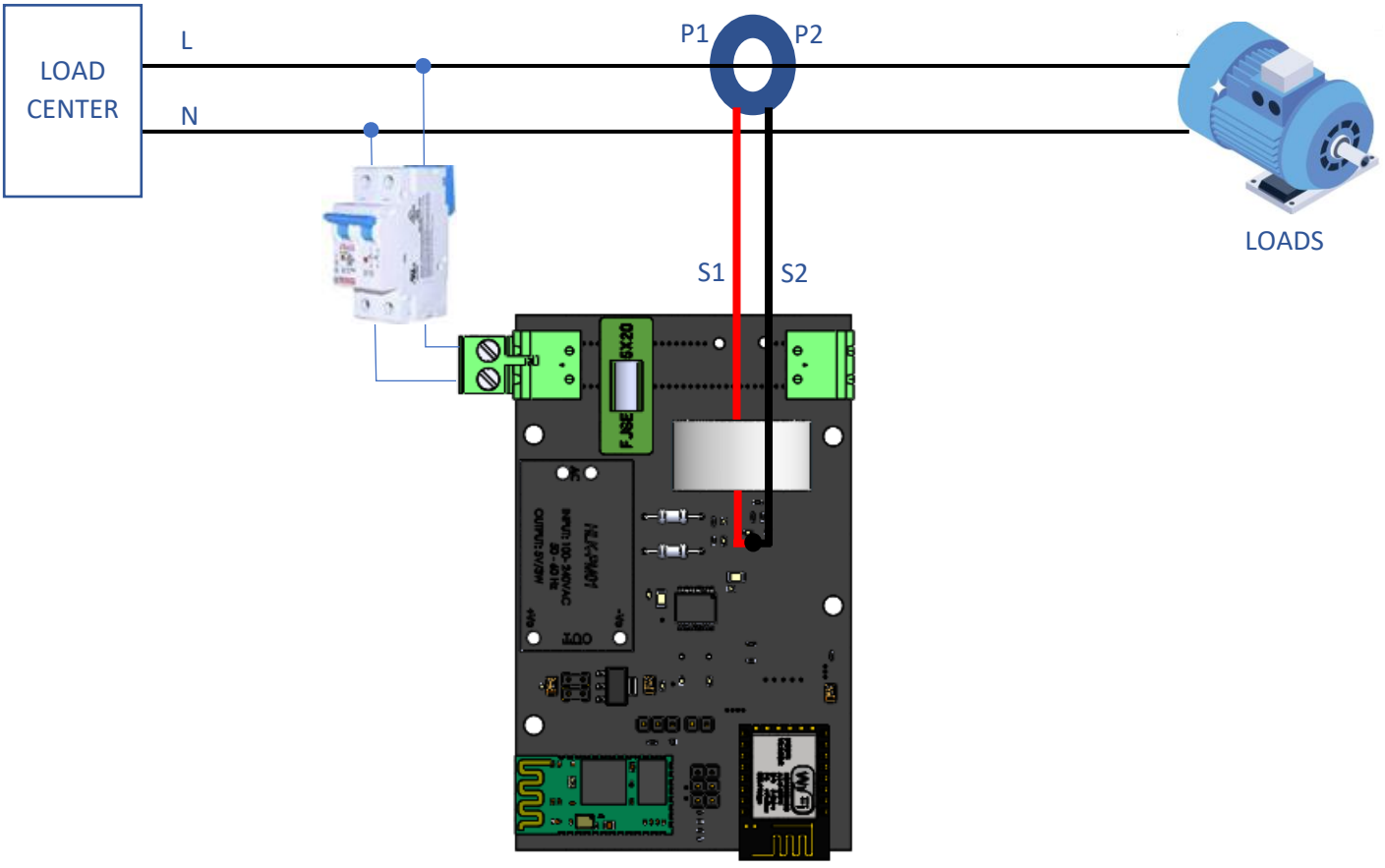
PLEASE FIRST CONNECT SECONDARY SIDE AND THEN THE PRIMARY SIDE TO AVOID ANY DISCHARGE POSIBILITY.

IF YOU FAIL TAKING THIS PRECAUTION YOU CAN PROVOQUE A BIG ACCIDENT.

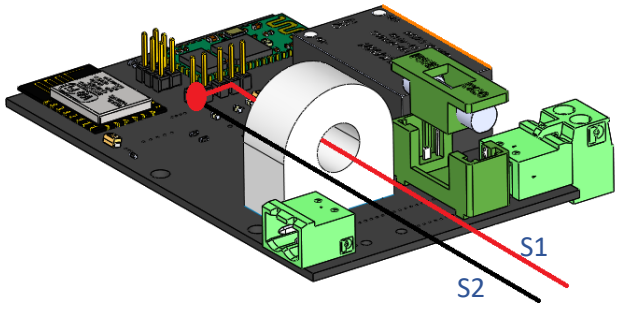
3. Main functions

Connect the CT to PQduino Board in this way:





3. Main functions



CT haves polarity, so please make sure power wire is passing from P1 to P2 into the CT, and for secondary wires of CT S1 is passing into the PQduino current sensor and S2 is passing outside the sensor and connect directly with S1.

ONE OF MOST IMPORTANT THINGS IS YOU NEED TO MAKE SURE THE SECONDARY WIRES MUST NEVER BEEN OPEN WHEN A CURRENT IS PASSING IN PRIMARY SIDE.

PLEASE FIRST CONNECT SECONDARY SIDE AND THEN THE PRIMARY SIDE TO AVOID ANY DISCHARGE POSIBILITY.

IF YOU FAIL TAKING THIS PRECAUTION YOU CAN PROVOQUE A BIG ACCIDENT.

After make connection and turn on the board, you need to change value in **CT ratio (x:5A) System Setup page. See 3e) point of this manual.** Click in “SAVE” and then in “REFRESH” to apply changes.

Value which must be introduced in CT ratio is calculated as primary side nominal current of your CT divided by secondary side nominal current of your CT, for example 100A / 5A = 20, CT ratio will be 20.

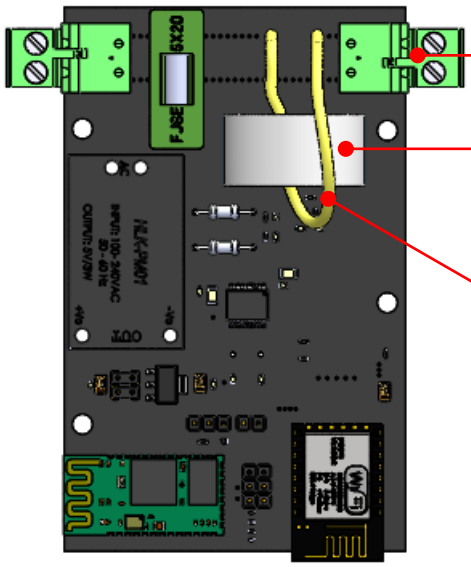
b) Using a miniCT like **SCT013-000.**

This method is easier than the previous one and we recommend it for non-expert users. There are different types of SCT013 but we are using SCT013-000 because it haves a 100A range and delivery until 50mA as a secondary signal. Also, this CT includes a protective diode as TVS in secondary side to avoid the open secondary danger.

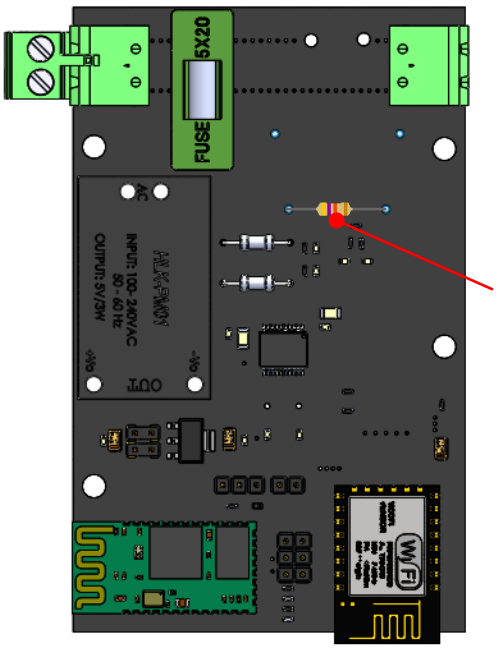
Modification to PQduino pcb that is made here is only right for this specific model of CT, if you want to use another model you must to calculate de right resistors values. In GITHUB repository there is an excel file that you can use for calculate right constants.

3. Main functions

Connect the CT to PQduino Board in this way:

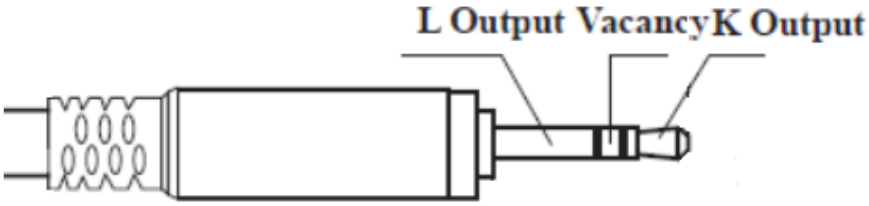


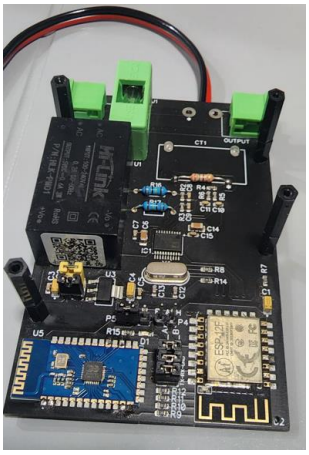
Solder a 15Ω resistor where you desolder the CT (output pins). You will get a board like this:

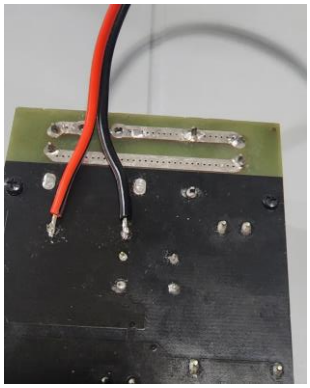


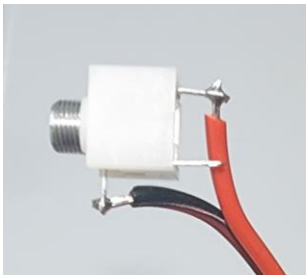
3. Main functions

In the same place (same holes) solder the terminal of **SCT013**-000, you can use a 3.5mm connector. Like this:









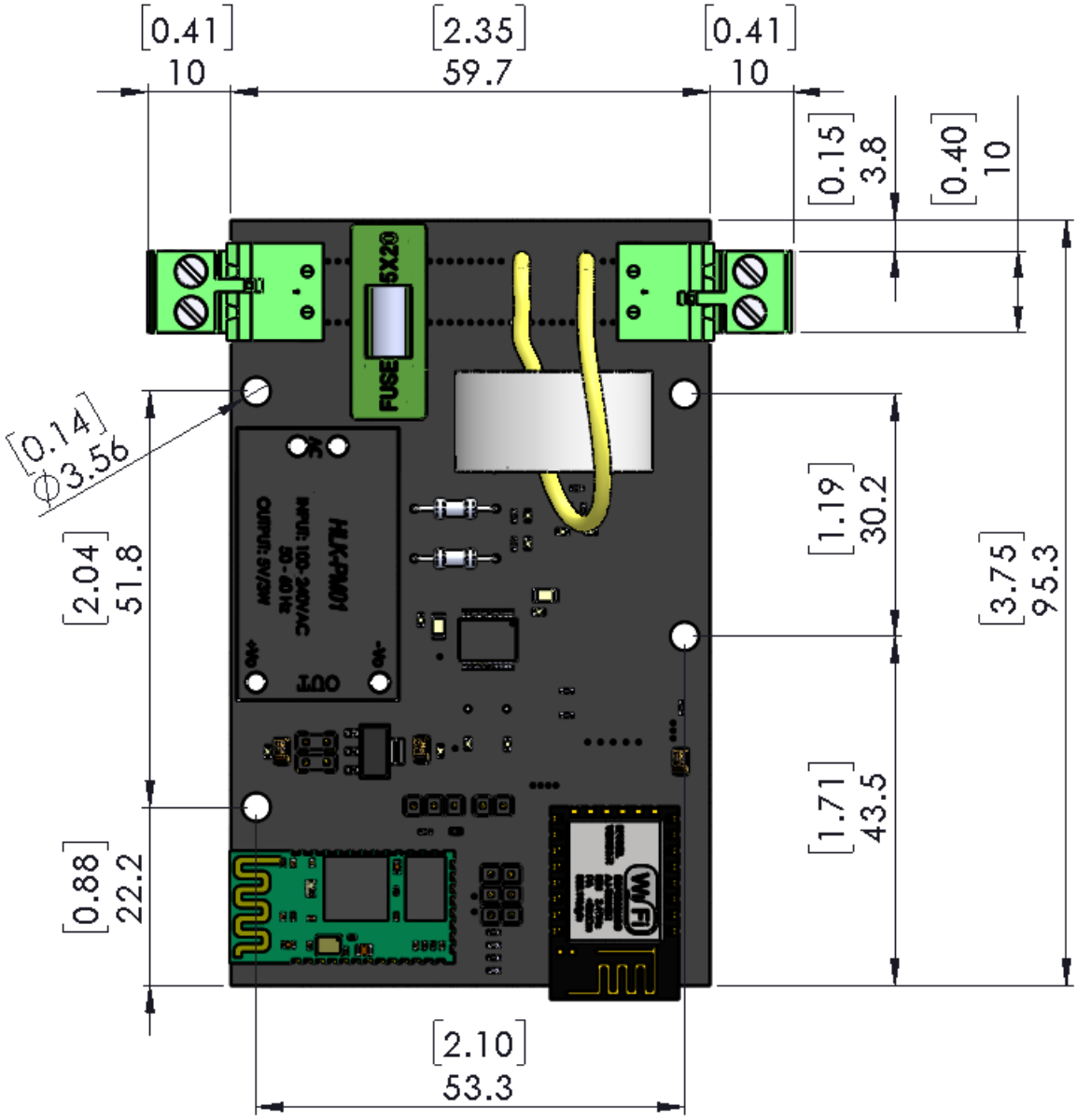
After make connection and turn on the board, you need to change value in **Current function (I=Ai+B) in System Setup page. See 3e) point of this manual.**

Introduced in A this value: 388

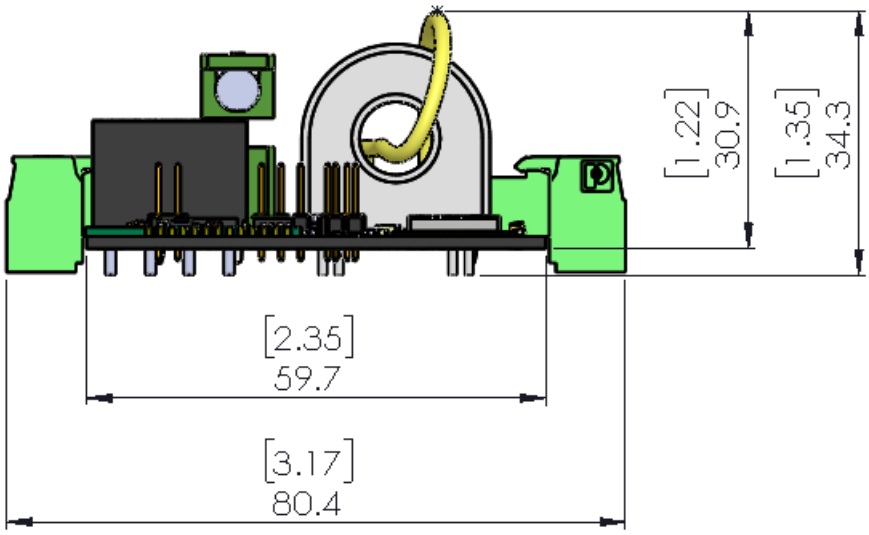
Introduced in B this value: -10660000

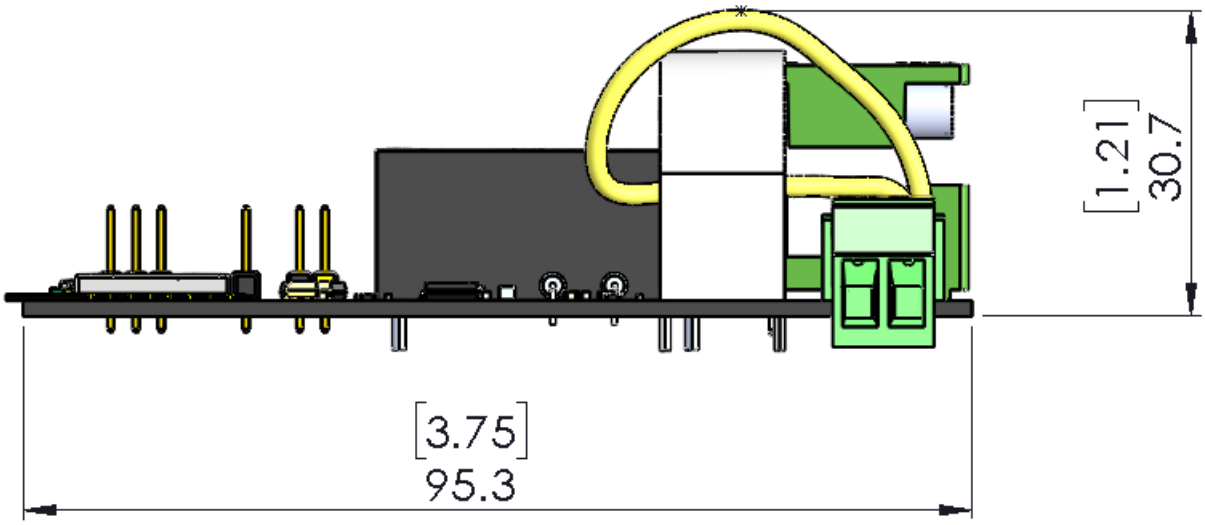
Click in “SAVE” and then in “REFRESH” to apply changes.

4. Dimensions



4. Dimensions





5. Web references

Contact pages and additional information:

YouTube channel:	https://www.youtube.com/channel/UCoWQ7Pg5kH3r624XluI_oBQ
Github repository page:	https://github.com/PQduino
Tindie store:	https://www.tindie.com/stores/pqduino/
PQduino Project:	https://create.arduino.cc/projecthub/leonel_flores/pqduino-f39d93
ArduinoMEGA2560_BT-Client-forPQduino:	https://create.arduino.cc/projecthub/leonel_flores/arduino-powermeter-9c2240?ref=user&ref_id=2164090&offset=0