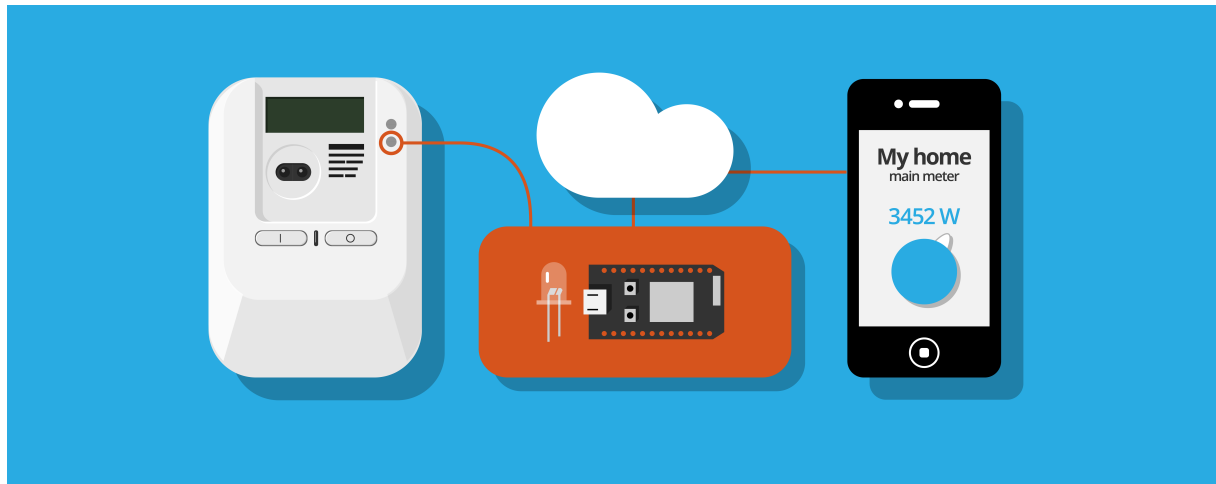


Electric meter monitor Workshop



What?

In this tutorial we will build a electric meter monitor from the very inexpensive NodeMCU WIFI module and a phototransistor. Many electric meters have an LED that flashes each time a certain amount of energy has been used. If you use the metaphor of water it would be like having a lamp flashing each time that you have used a liter of water. If you are pouring water slowly there will a long period of time between each flash and if you open the tap up fully it will be flashing with short intervals.

How?

The theory of the monitor is that the phototransistor detects the flashing light on the electric meter and the NodeMCU module then sends it over your WIFI to the "cloud" and makes is available on for example your mobile phone. There are several commercial solutions that does this but in this workshop we will create our own monitor based on a €2.5 WIFI chip and open source software.

The draw backs of this solution compared to most commercial ones is that its not battery powered. In other words you will need a power outlet and USB charger close to where you want to install it. The

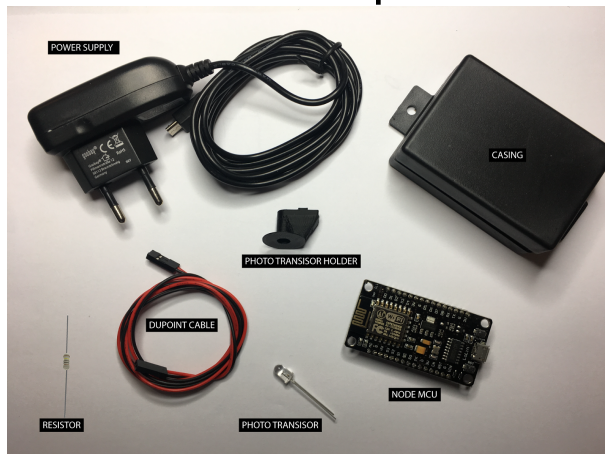
benefits of the solution is that the platform can easily be modified. It can potentially also host its own web service locally on your network if you would like to skip the cloud functionality.

Why?

We believe that in order to manage our resource waste we need to build really good smart energy services. And in order to build smart energy services they need to be build by people who understand what kind of things that would be smart in your life. And who is better to understand those things than ourselves. Our goal is to make it dead simple to experiment with new concepts and ideas in this area, allow for people to get first hand experience of how smart energy applications could be design or change the applications they use them when needed.

In later tutorials we will also explore how to use existing open source software or widgets to analyse and visualize energy data. If you are really interested we will also show how to analyse the data and write your own visualizations. And connect the data to emerging platforms such as the Apple Homekit or Alexa.

Hardware set-up

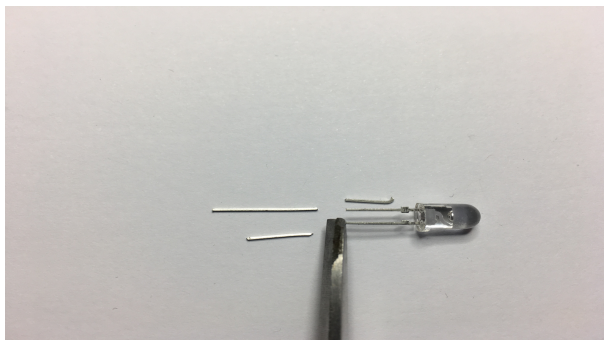


These are the parts that we will need to connect together to create the hardware for our energy monitor. Follow the description below watch the movie linked by this QR-code:



<https://youtu.be/4lkSjUP7SJo>

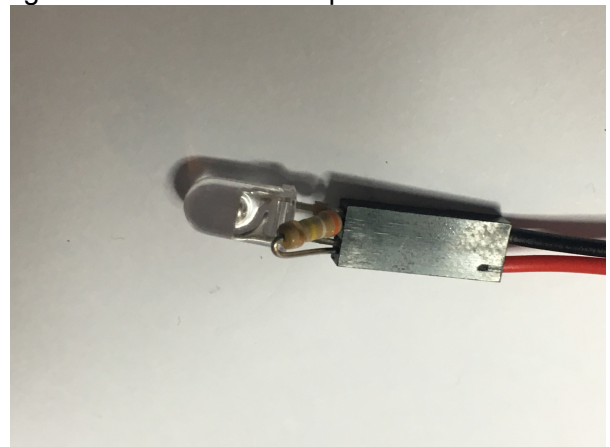
1. Connecting the phototransistor



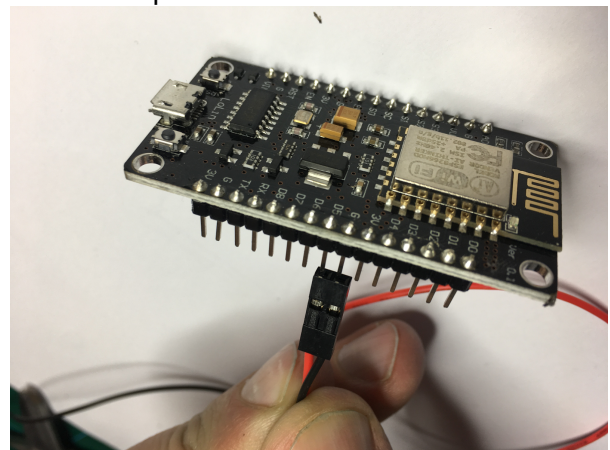
Begin with cutting the legs off from the phototransistor a bit.



To increase the sensitivity of the phototransistor you might also have to add a resistor in parallel with it. In that case cut it as shown in the picture. The value of the resistor can be between 80-400kOhm. The smaller the resistance the more sensitive the phototransistor will be and the less light it needs to detect a pulse.



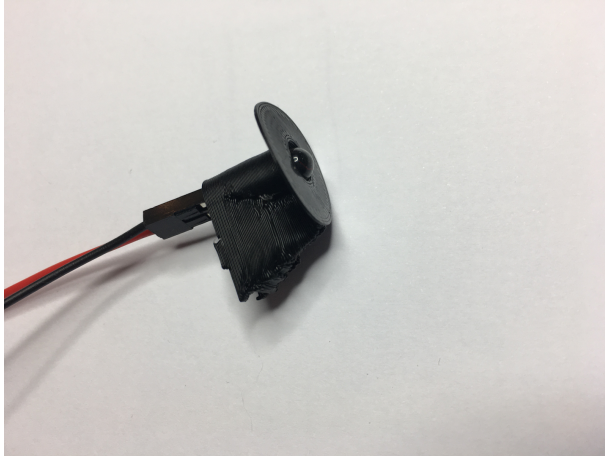
Slide both the phototransistor and resistor into the Dupont connector.



Connect a other side of the Dupont cable to the GND and GPIO5 (situated next to

each other) on the NodeMC. Make sure the phototransistor is turned the right way. The short leg with the broad stop piece and GPIO5 should be connected through the red cable and the long leg and GND to with the black.

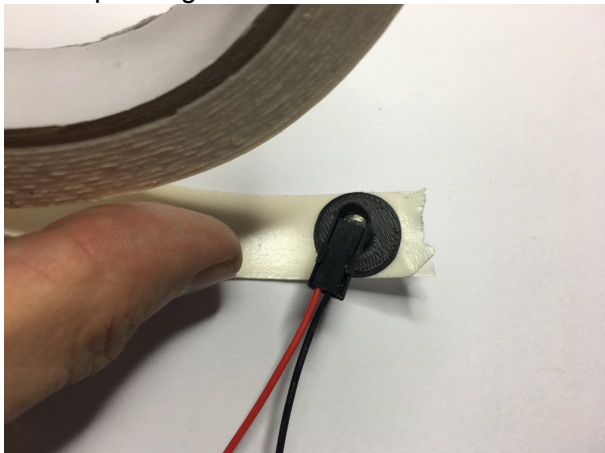
2. Mount the holder.



Push the phototransistor into the holder so that it sticks out 1-2mm.



Bend the Dupont housing down into the positioning the picture. If you have the resistor as well it can be left outside the holder pointing forward.



Cut out some double adhesive tape (the foam type works well) and stick it to the opening of the phototransistor holder.



Make a whole for the phototransistor in the adhesive tape.

3. Installing the sensor



Your sensor is now ready use. To mount it you simply remove the protective plastic of the adhesive tape and stick onto the LED of on the electric meter. You need to make sure the phototransistor is centred above the LED and that you have picked the right LED. Some meters have several LEDs. The right one usually have a marking saying "kWh" or "1000 imp/kwh", or something similar next to it. In the picture the top LED is providing pulses in the unit kvarh (reactive power) and the lower in kWh which is what we want. Before we mount the sensor however we will need to upload the software for counting the pulses onto the NodeMCU and check that everything works OK.

Software set-up

Uploading the software requires these steps.

1. Install development tools

This is only needed if you want to program the device from your own computer. If you are in a workshop and there is a computer provided already you can jump to step 2.

a) Download or copy all the files that are in the workshop bundle online (<https://drive.google.com/open?id=0B4cGiPZuiQZtaUNZUnE4N2lvSUK>) or from the USB sticks that will be available during the physical workshop to your computer.

Name	Date Modified
OpenEnergy	Yesterday 23:29
Drivers	Yesterday 22:45
OSX	Yesterday 22:45
CH34x_Install_mac	Yesterday 22:44
CH34x_Install.pkg	02 Nov 2015 02:46
ReadMe.pdf	03 Nov 2015 16:05
CH34x_Install_mac.zip	Yesterday 22:42
Windows	Yesterday 22:45
Firmware	Yesterday 23:29
Tools	Yesterday 22:46
Virtual Machines	Yesterday 22:38
VirtualBox	Yesterday 22:39

b) Open the driver folder and install the driver matching your operating system.

Name	Date Modified
OpenEnergy	Yesterday 23:29
Drivers	Yesterday 22:45
Firmware	Yesterday 23:29
Tools	Yesterday 22:46
Virtual Machines	Yesterday 22:38
VirtualBox	Yesterday 22:39
VirtualBox-5.1.8-111374-OSX.dmg	Yesterday 22:39
VirtualBox-5.1.8-111374-Win.exe	Yesterday 22:39

c) Open the folder named VirtualBox run the installation file that matches your operating system.

OpenEnergy	Yesterday 23:29
Drivers	Yesterday 22:45
Firmware	Yesterday 23:29
Tools	Yesterday 22:46
Virtual Machines	Today 00:49
OpenEnergyToolbox.zip	Today 00:47
VirtualBox	Yesterday 22:39

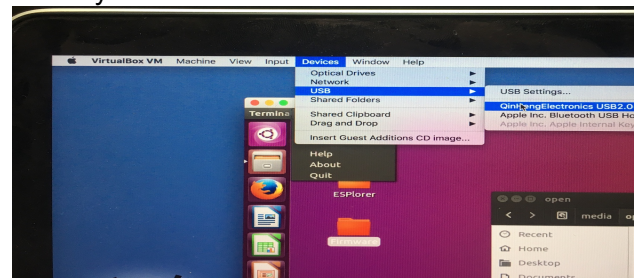
Open the folder named "Virtual machines" and click on the file named **OpenEnergyToolbox.zip**. To unzip the the virtual

machine (this can take some time).

Name	Date Modified	
Logs	Today 00:25	
OpenEnergyToolbox.vbox	Today 00:41	5
OpenEnergyToolbox.vbox-prev	Today 00:41	5
OpenEnergyToolbox.vdi	Today 00:41	7,9

Open the folder that was created named OpenEnergyToolbox and double click on the OpenEnergyToolbox.vbox file. A virtual machine will now start with all the development tools preinstalled.

Connect your NodeMCU with a USB cable to your computer if you haven't done so already.



Once in the virtual machine in the menu click: Device -> USB and select Qin-SengElectronics USB2.0-Serial (the name might vary on different operating systems). The name should appear after you plugged in the NodeMCU and disappear when you remove it.

2. Upload the program

Open the ESPlorer program. It can be found in a folder on the desktop called ESPlorer and is started by double clicking the **ESPlorer.jar** file. Wait until it starts.

Check that settings in the top right area are as in the picture and click the **Open** button on the right side of the program.

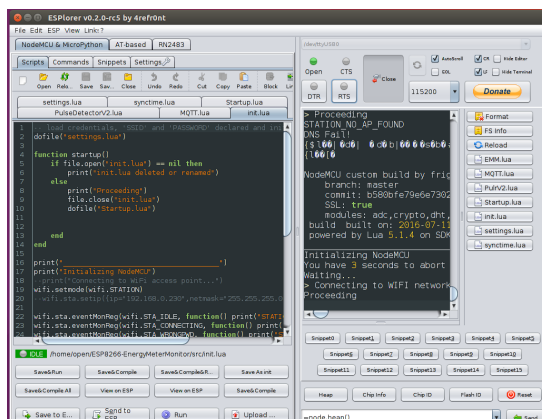
To check if you are connected to the NodeMCU you can now either press the heap button in the bottom towards the middle or press the reset button on the module itself and you should get some printouts in the terminal.

Now click open and open the file **set-tings.lua** inside the ESP8266 folder. This

file contains WIFI credentials and some other setting that you will need to change when deploying the monitor in your home. Among other things how much each pulse from your electric meter represents (this varies some across different manufacturers and meters).

Press the Save to ESP button in the left bottom corner to send the file to the NodeMCU.

Now open all files inside the **src** folder and send them to the NodeMCU with the same button. Take care to save the init.lua file to last. After you are done you can press the reload button on the far right (blue arrows) and it should look like this.



The list of files on the right represents the files exists on the NodeMCU.

Finally reboot the NodeMCU and write down the data url printed by the NodeMCU in the ESPlorer console. It begins with <https://op-en.se/emm/?id=...>

When you have that link enter it into the webbrowser and then proceed to connect your sensor to a meter or some other flashing light for testing.

Test your set-up

Copy the url that is shown in the console window of ESPlorer once the device is booted and past it into a browser.

It should look something like this:

<https://op-en.se/emm/?id=XX:XX:XX:XX:XX:XX>

Where the XX:XX:XX:XX is the unique of your device.

Once in the webpage you can then either swipe a flashlight past the photo sensor or mount it on an electric meter. If you use a flashlight flashing it once every second should produce a value of 3600W.

The blue light on the NodeMCU board should light up every time you shine the flashlight into the sensor or when the power meter that you attached it to flashes. If it doesn't try reversing the phototransistor or simply use stronger flashlight or add the parallel resister as described earlier. Also check the cables and try rebooting the device. You might also have to reprogram it. If the blue LED is instead always on try using a smaller resistor.

If successful it should look like in this movie.



<https://youtu.be/zxHxe6TVfRg>

If you manage to get everything working this far you can now process to the next tutorial and start building your own web service of app.