Today, we do something really dangerous. But it is also a very rewarding addition to your Home Automation System. We hack the newest child of the Sonoff family: The Sonoff POW. This device can not only wirelessly switch power on and off. It can also measure how much Power and current is currently flowing through the device. This is very handy because we can monitor the consumption of our devices without spending too much money: The Sonoff currently sells at about 10$ plus shipping.

The device itself carries a CE sign and is therefore not dangerous if you use it as intended. But as soon as you open it, it becomes, of course, a little dangerous. This is the same with all devices which are connected to mains. But hacking this one is much more dangerous than hacking any other device. Why is this?

If we take a normal Sonoff switch, we have a complete electrical separation between mains and the logic circuit and we measure infinite resistance between live or phase wire and logic ground. There is no danger if you connect your devices to the logic circuitry.

If we do the same thing on our Sonoff POW, we measure nearly zero ohm. So, the whole logic is connected to the live wire of mains and we measure the full 230 Volt between logic ground and Earth. So, stop watching if you do not know exactly what I told you up till now!

For the rest of us: pay extremely attention that you remove all serial cables before you plug the device into 220 Volt. Otherwise your computer can easily be roasted. Think twice, before you do something.

Enough about safety. What’s in the box? As usual, it is an ESP8266 and a relay. In addition, ITEAD built-in a chip which can measure current and voltage, and calculate power. It even should be able to use True RMS calculations, if I understand or believe the datasheet. And here comes the problem: The chip is from a Chinese manufacturer and the documentation is completely in Chinese. I used Google translate to get at least some information which I will share with you.

The chip is connected to the high side of the load. This means, that a 0.01 Ohm resistor is inserted into the live wire and the chip measures the voltage across it. This is also the reason for the inexistence of a separation between Mains and logic ground.

In addition, the chip measures also mains voltage. With the two values, it does its calculations and as an output, we get two signals which vary in frequency. Therefore, we only can measure power and voltage or power and current. For my purpose, and because our net here is quite stable, today, I do not measure voltage. But if you live in another place, this could be quite interesting. You send an alarm if the voltage drops under a particular value. Or you could immediately switch the device off if this happens.

Fortunately, I got a small library from ITEAD which measures these frequencies and provide the Power, Current or Voltage readings. I added also the capability to read frequency. I will show you later, why.

So, we can read power, current, and frequency. But as we saw before, we must not touch the system and we must not use Serial connection. Here comes all the investment we did in all the previous videos: We connect our device completely wirelessly, which means:

1. We load or update the sketch over the air with the newly created iotappstore principle.
2. We use telnet to monitor the values for a first test. For this purpose, I use a very good library called “remoteDebug”. You find a link to it in the description.
3. Later, we will send the measured data and get the command for the relay via MQTT

So, we can do our first tests. And it works. Values are shown and transmitted over WiFi.

I choose MQTT, because many of my viewers insisted, that MQTT is better than just call a REST API. So, I use it here. If you want to use a different service with a REST API, you find all the information in my past videos.

So, our Sonoff publishes three MQTT topics:

1. The power reading
2. The current reading
3. The state of the relay

I use PubSub as a library and Adafruit.io as a MQTT broker, because I still have no own broker in my lab. But I know, that many of you already have your own Home Automation system capable to deal with MQTT.

The Sonoff itself subscribes to a “command” topic, which switches the relay on and off. So, let’s look at the dashboard. We have the command switch, and, as soon as I switch the device on, the light goes on, and the switch indicator represents the actual state of the relay. The same happens, if we switch it off.

Below the two switches, we see power and current. Both values are not measured frequently. This has to do with the limitations of Adafruit.io which does not accept more than 2 postings per second. Because we have 3 values, we only can update every 1.5 seconds. There is also some averaging done in the library which delays the reading.

Anyway, you find the library on Github and I would be glad, if you have a look at it and help to improve it. Maybe somebody also looks if the accuracy of the frequency measuring code is already ok or if it can be improved.

As a last step, we need a possibility to enter and store the WiFi credentials and also the address and credentials of the broker service. For this, I use a fork of the WiFi manager. Also here, you find the link in the description. And if you look closely to my code, you will see, that I separate the configuration mode completely from the measuring code and do the switching using RTC memory. The switching is done by pressing the button on the Sonoff for more than 4 seconds. If you press the button for about 2 seconds, a new sketch version is loads from the IOT appstore, of course, only, if yours is not the actual one.

Now we have everything together for our great addition to our home Automation system. But how exact can this device measure power and current? For that, I built a small test system consisting of a Sonoff device and a load. As loads, I use different light bulbs from 12W to 100W. To check, if the results are ok, I use a normal power meter and also my UNI-T multimeter with true RMS measurement. This is not important for old Wolfram lamps. But for the biggest load, I use a vacuum cleaner. This happens to be one of the bigger consumer of Energy in our household, which can be plugged to a wall connector. It contains a motor, and therefore, I need RMS measurement.

The voltage is constant for the smaller loads. Just with the vacuum cleaner it is a little lower.

Here are the results:

At a first glance, we see, that power measurements are more precise than current measurements. For loads around 100 W it both are pretty accurate. For small loads, the difference for the current is bigger.

For the vacuum cleaner, the difference is big again. Because this is just a first test, and I have only one device, I cannot tell you where these differences come from. If they come from the chip itself, from the library, or from the way the chip calculates the RMS power. Maybe you do your own tests.

One thing is important, however. The chip produces a frequency and the library uses a simple formula to calculate the Power and current:

A x frequency + b. So, you can calibrate your device to the range you need.

Going through the code we see, that we can store a and b for power, current, and voltage by using the commands set parameter. The variables are then a and b.

To measure, you call the functions getPower or getCurrent. You see also, that I used double instead of float format. This is, because the ESP, differently than the Arduino, provides more precision with this data type.

After reading, I publish the values to MQTT. The rest of the code deals with the infrastructure of wireless communication as mentioned before.

I used 0 and 100W to calculate the two factors a and b. If you plan to use it for other ranges, you can calibrate it to this range by changing the factors in the code. If you do not remember, how this is done: I added a link to a video.

This is, why I also display the frequency reading. You need these readings for your calculations.

If we watch the measured values a little closer, we see, that the 100 W readings are quite stable. For small readings, however, also the frequency readings vary. So, the results are also not as accurate. But as I said: Maybe we can tweak the library a bit…

Summarized, this is a very interesting device for all sorts of Home Automation Projects because you can measure on which power level a device is currently working, and also, how much energy it consumes during 24h. You could also switch consumers on- or off depending on your power generation situation if your use for example solar energy.

Together with the MQTT interface and the Webupdate through the IOTappstore as well as the entry of all credentials via a Smartphone or browser it is already a very usable tool.

I hope, this video was useful or at least interesting for you. Bye

https://www.youtube.com/watch?v=AvSXCQi75i0