In my lab tour in video #88 you saw, that I have three different work spaces and 5 LED spotlights. In this video, I want to hack a Sonoff wireless switch, build a small wireless motion detector, and combine them to a remote motion detecting system. Let’s start! And everything for about 10 $.

Currently, I use two normal in-line motion detectors attached to two groups of LEDs. One group contains three spotlights and are connected to the first detector which, in theory, should see two of the three workspaces and keep the lights on if I work there. And the other motion detector is connected to the other two LEDs which are a little remote and rarely used.

There are two problems with that design:

1. It is not easy to mount the in-line sensors that they “see” both work places
2. The sensors are completely independent of each other
3. One of the two died two weeks ago

This is, why I want to replace them with two Sonoff switches and two ESP8266 modules with small passive infrared sensors or PIR sensors. The Sonoffs are small boxes which include an ESP8266, a relay, and a 3.3V power supply and can be connected directly to mains. Keep in mind: Mains voltage is dangerous and you have to know, what you do.

The sensors and the Sonoffs are connected through Wi-Fi. This can be a one-to-one, but also a one-to many connection, which enables us to create more complex scenarios and is the solution of our issue #2.

Because the sensors and the switches are in two boxes, the sensors can exactly be placed where needed. The spotlights should keep running for a certain time, let’s say, 5 minutes, after I left the particular place or the lab.

And, of course, I want, that I can change the parameters during operation from a smartphone or a browser. I use the small built-in push button to select the mode. Just a small information which can save you some time: If you want to change the wifi mode from AP to STA or vice versa, you always have to reset the ESP and disconnect everyting. You find the details in the sketch. If you try it differently, you end up in an undefined mode, quite often in AP\_STA mode which can be a security issue. Maybe I will deal with this problem in a future, focused video.

I use the cheapest and simplest Sonoff switch which costs around 5 dollars excluding shipping and, for the sensor, a Wemos mini board and one of these small SR505 PIR sensors. They run of 5 V, but their output voltage is around 3.3V. So, I connect them to the 5V rail of the Wemos and connect the output to the D5 pin of the ESP. The specifications say, that they should reach 3 meters, but I do not think so. I have to further investigate into these sensors and also compare them with bigger ones.

To power them, I use a small 5V power supply. And, of course, I mount them in a small, 3D printed box. The PIR sensor and the DC jack is hot glued and the Wemos is mounted with a double sided sticky band.

But, how can we program these devices? And how can we make sure they can work with each other?

Let’s start with the Sonoff. To program it, at least the first time, we have to open it. Make sure, that you do this before you connect it to mains!

The PCB has 4 holes which can be used for programming, because they expose GND, 3.3V, TX and RX. To make it simple, I soldered female pins to the board and made a specific programming cable which connects my FTDI adapter with the Sonoff. I included also a small switch to interrupt VCC for programming. Make sure, that your FTDI adapter is switched to 3.3 volt!

To set the ESP on the Sonoff in programming mode you have to connect GPIO0 to ground while powering the board up. This can be done by a small push button which is already there. With our programming cable and the push button, we can start to program the device. In the IDE we select a generic board with 1M and the minimum 64k SPIFFS. This allows quite big sketches and also OTA programming.

Because, later, I want to close the Sonoff box and mount it somewhere hidden, I only want to do this programming with a cable once. Afterwards, I do not want to reopen this box again. This is a perfect scenario for the IOTappstore. So, I only load the standard IOTappstoreLoader sketch on the board. After powering up, I can connect to it with my smartphone or also with a PC. I have to enter the credentials and the address of the IOTappstore server. This only has to be done once because the values are stored in the EEPROM of the device. The IOTappstore is presented in two of my past videos. It does not yet work with the version 2.3 of the ESP in the Arduino IDE, at least not, without patching some files. It should be officially supported with the 2.4 release. But till then, you can use the programming with the cable. No worries.

And I am working with a web developer on a user-friendly version of the IOTappstore, which should be available before Christmas. Then, you just load once your loader sketch, define your new Sonoff board in the ITOappstore, and define the sketch you want to download. The rest is done automatically. It then will download my file to your Sonoff. No installation of libraries, no programming, just fun…

Let’s now come back to the sketch. After checking for new updates, we have to start a Wi-Fi server which understands just two simple commands and a status message for debugging: If it gets a HTTP request with the word “ON”, then the relay is switched on. If it gets a request with “OFF”, the relay is switched off. The off functionality is not used in today’s scenario, because we will switch the lamp automatically off if the time of the last “ON” signal is more than 5 minutes ago. This is done with a simple counter which increases every 100 ms.

If it receives the command “status” it sends the number of seconds since the last “on” command. This functionality helped me to debug the sender

The server also sends a small feedback text. This is mainly used if you want to test it with a browser.

If we start it up (still connected to the programming cable for debugging), we see the IP address of the module. We have to make sure, that this address always stays the same, otherwise we will not find it anymore in the vastness of the internet. I included also the DNS service to avoid a fixed IP address, and this worked from a browser. But so far, it did not work with another ESP. Maybe a viewer can help me out in this matter.

A fixed IP address can be assigned in the DHCP server. This is usually your router. I showed the how-to in my video about the IOTappstore.

If we now call this address with an “ON” command, we see it in the serial monitor of the Sonoff. So, this part is ready for deployment.

Now we have to build the sensor. This is a little soldering work and 3D printing. After the assembly, we also start with loading the IOTAppstoreLoader sketch and store the credentials and server names. I usually print the MAC and IP addresses on the box for future reference.

Now, we can start to write the sketch for the sensor. It monitors the PIR sensor and, if it is “ON”, it sends an “ON” command to the Sonoff. If you defined two different IP addresses, it sends the command to two Sonoffs if we want to switch more than one LED on. As said before, we do not send an “OFF” signal in this scenario.

The sketch is similar to the one before and also contains the feature to change the parameters remotely. I included also a small LED which shows the status of the PIR sensor. With its help it is easier to position the sensor. Keep in mind, that these small PIR sensors are switched on for about 8 seconds till they are switched off.

So, also this sketch is ready and we can close the device and mount it at a strategically good position.

The test shows, that, as soon as I enter my lab, the LED starts to shine, exactly as planned.

So, we have built very flexible motion detectors which also works over a certain distance, as long as you have a Wi-Fi network available. So, you could also imagine to use a water tight box for the sensor to mount it outdoors and keep the Sonoff somewhere inside your house where the conditions for electronics are better. Maybe you have other applications for this scenario not using light, but heating, motors, or other electrically driven stuff. Or you could replace the PIR sensor with any other sensor like luminosity, temperature, humidity, vibration, sound, etc. You even could combine some internet services like time, sunrise or sunset, weather forecast, etc. to blend with your own sensor data. Because the sensor box contains a beefy microprocessor, the limitation is only your imagination.

If you keep both devices behind your firewall, the security issues should be manageable, if your Wi-Fi is properly secured.

BTW: If you press the push button at the Sonoff, it creates its own WLAN and you can connect to it. I gave it the name SONOFF. And here, you can enter your Wi-Fi credentials, see the MAC address of the device for your router, enter the names of the future IOTappstores, and also other parameters for future use. Constant 5 and 6 are used by the sender to enter the IP addresses of the SONOFFs they have to switch on. If you fill only one, only one Sonoff will be switched as in my scenario.

In future videos, we will build on this knowledge and code for other Sonoff devices, also devices with built-in sensors. These devices are particularly interesting, because they include already a power supply and are, according ITEAD, really CE compliant. With such a device, we can replace an ESP, a sensor, and a power supply. Most of the Chinese standard 5-volt power supplies I use in my projects are anyway not CE compliant. So, with these Sonoffs I can sleep a little better.

Let’s now quickly calculate the material cost: The Sonoff: 6$, the Wemos clone: 3$, and the PIR sensor: 2$. So, we are close to the 10 dollars, if you use the cheapest prices available. With normal prices and including the material for the 3D box we are probably closer to 15$. Still ok for me.

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