Maybe you have heard of the Amazon Dash Button. This is a small device you can place where ever you want. And every time you press the button, it orders whatever is printed on the device. I think, the button is a good idea. But I do not like, that you can only use It for one purpose. This is, why I wanted my own, much more flexible button. And in addition, you can use the principle used here for many other applications where you want a device running on small batteries for years.

But before we start: Rumors are, that YouTube changed some of their algorithms and give small channels like this one more chances. But what these channels need is likes and comments. So, if you want to help and also help other to discover this channel, like and comment as much as possible. Thank you!

Now, let’s start:

For my frequent viewers, it is no surprise that I will use an ESP8266 for my prototype. But for the real thing, it is better to use the new ESP8285. Then the device is not much bigger than the Amazon Dash

In video #58 I used a coin cell to power the ESP. And it ran quite long. But the button we want to build today has to last much, much longer. And therefore, we cannot use ESP deepsleep. It uses way too much power. We need a concept which uses less, or ideally, no current during sleep. How is this possible? In my video #4 I used a simple concept for that: I used a mechanical switch to power the ESP. The sketch has only a setup() part. The loop does nothing, just wait. If you press the button long enough, everything works fine and you have a device which uses absolutely no current during deep sleep. Great idea, isn’t it?

The only thing is, that you have to wait and for sure press the button long enough till all tasks are executed, but not much longer to save batteries. Unfortunately, this concept is not applicable, if you want to give this device to your kids or your Grandma. So, we have to add something: The microcontroller has to have the possibility to keep the “lights on” as long as he needs them. So, we add a second connection from the battery to the ESP: A relay. The relay also has the advantage, that it does not need any power when switched off. Only if it is switched on, it needs power. If we now connect the coil of the relay to one of our IO pins, we can switch it on and off with our ESP.

If we need the ESP for a task, we press the button and the setup starts to run. Right at the beginning of the sketch, after a few microseconds, we switch the relay on. From now on, we can release the button and the ESP is still powered. The rest of the setup does all the needed tasks, and, as the last step, switches the relay off. If we released the button in the meantime, the ESP switches off and does not use any power till the next action.

If we were not fast enough to release the button, the ESP just waits in the loop and does nothing till we release the button. So, our it can run now for years if you only have rare events.

Now, we can change the concept a bit. For example, I had no 3.3-volt relay in my lab. So, I used a P-channel MOSFET instead. I had to add a resistor to keep the gate high. That’s all. Here, I use a big MOSFET, but only, because it is breadboard-friendly. In reality, you can use a much smaller one.

The MOSFET has a small off-current and the battery will not hold forever, but still much, much longer than with deep sleep.

Or we replace the relay with a reed switch. A reed switch closes contacts if a magnet comes close. So, we can trigger the ESP with a mechanical movement like a door or a window.

Now we have everything ready to do something useful: We connect our button with IFTTT. And from there, we can do whatever we want. Followers of my twitter account saw the tweet which was initialized by this device. I created a “maker“channel and connected it in a “recipe” with twitter.

Two remarks: Because I still do not know how to use https connections, I had to write a php file on my webserver as a forwarder. And IFTT is not at all real-time. My twitter message was sent about 15 minutes after I pressed the button.

Of course, I could have also connected the button event with google sheets and I would have a logger for events. Or, with a connection to SMS, my 91-year-old father could press a button and release an SMS to me or to someone close to him to help if he is in trouble. Or I could order a pizza (no, not only at amazon). And that’s what I am going to do now!

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

Or the fans of node red connect it to their server