Using NodeMCU ESP8266 + RFID + PHP for doing something

The assembled self-made RFID access “thing”

Recently I was playing around with some cheap modules and sensors – mostly ordered directly from China using Alibaba or Amazon.

I had this idea for an RFID controlled door ever since I started working in a huge R&D building where every single door could only be opened with your personal badge. I wanted to have that at home. Why? Well.. because it’s possible :-)But commercial solutions are just way to expensive and it is actually so much more fun to build stuff.

So after having this idea in sleep mode for a couple of years it finally surfaced again when I cleaned up the basement and found an old LCD which I got from a Raspberry Starter Kit – but actually never used.

I also knew that the nodeMCU can handle this LCD – but I miss the I2C adapter for it. So I ordered it from Amazon.

Whilst I was playing around with some cheap other RFID readers and a relay – which I considered as highly unreliably and uncostomizable I decided to give it a try and build the stuff myself.
The basic idea:

NodeMCU + RFID reader + LCD + Buzzer connected to a server to manage access right.

So what do we need? Some electronics first.

The part list:

1. nodeMCU ESP8266 ESP-12E: http://amzn.to/2iE3GdM
2. Mifare RFID RC522 Radio frequency IC card + Keyfob: http://amzn.to/2zSVEBS
3. IIC / I2C / LCD1602 2004 LCD-Adapter: http://amzn.to/2xr60qZ
4. LCD Display Module: http://amzn.to/2yUvrVC

Passiv buzzer from the Starter Kit (also includes the LCD and jumper wires: http://amzn.to/2zKZeND

Whilst we wait for all the HW to arrive let’s turn towards SW.

We need following SW:

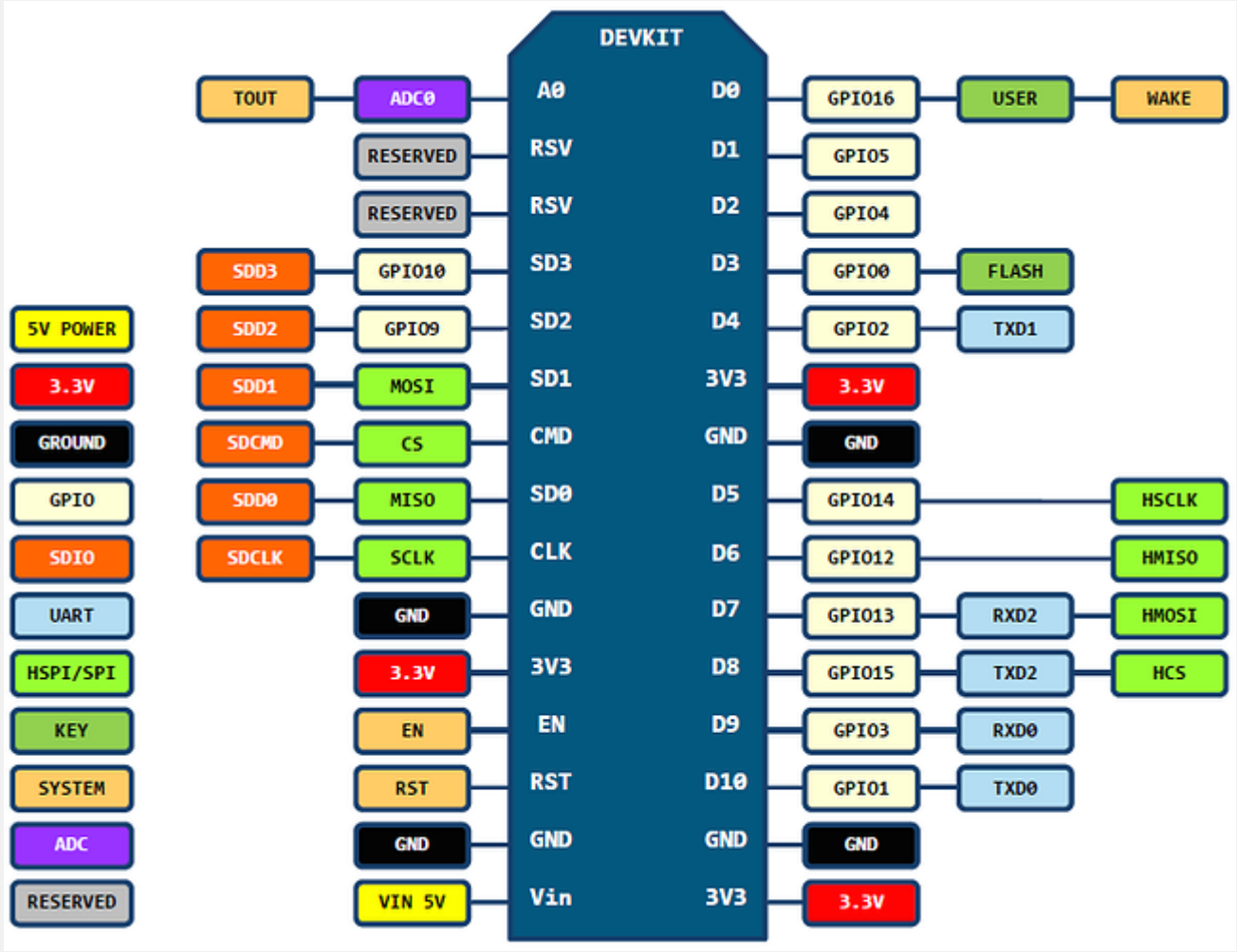
1. Arduino IDE
2. ESP8266 Board Manager the Arduino IDE
3. Libraries for I2C LCD and RFID, Wifi and HTTP client
4. Code

1. and 2. A very short and straight forward quick-start guide for setting up Arduino IDE for ESP8266 is available here: https://www.hackster.io/Aritro/getting-started-with-esp-nodemcu-using-arduino-ide-aa7267. It also includes the download links – so I am not posting it here again.
3.a The library to talk to the I2C LCD can be downloaded from https://github.com/marcoschwartz/LiquidCrystal_I2C
3.b As library for RFID you can use the RFID library by miguelbalboa.
3.c The Wifi library can be found on Github here: https://github.com/esp8266/Arduino/tree/master/libraries/ESP8266WiFi
3.d and finally the HTTP Client also from Github here https://github.com/esp8266/Arduino/tree/master/libraries/ESP8266HTTPClient.
4. All libraries come with samples. Build on that. I also link some more samples in the following paragraphs.

Download and unzip both files to the libraries folder of the Arduino IDE (typically under C:\Programs(X86)>Arduino>libraries) – and restart the IDE afterwards.

Additional hint:
Whilst it is easy and straight forward to use the Arduino IDE it sometimes causes a bit of confusion (at least for me) with the I/O mapping between of Arduino and the NodeMCU. When using the Arduino IDE your code should use the Arduino pin mapping but you have to keep in mind that the actual NodeMCU Pins are different – For example if you want to use the NodeMCU's D4 PIN your code in the Arduino IDE has to point to Arduinio's GPIO 2 PIN

Here is the pin definition:



Wiring the HW

I was thinking to provide full wiring diagrams. But actually I think the real learning is by doing all that stuff step by step. So I'd recommend to follow the examples provided with the Demos of Arduino and NodeMCU.

1. First we talk to the LCD:
You can follow this instruction here: https://www.losant.com/blog/how-to-connect-lcd-esp8266-nodemcu. It also has a very short example of how to talk to the LCD and includes a description on the wiring as well.

In my case I first had to connect the LCD itself with the I2C adapter and the I2C adapter with the NodeMCU – I used an old IDE cable for that – its has the marked cable for PIN1 so it makes it easy to not connect it wrong – of course you could also use a lot of single jumper wires for that.

Small hint – the display uses 5V – nodeMCU is running on 3.3V – but if you

power it from USB you can grab the 5V from the VIN pin.

2. Let's utilize the RFID module.
TheCurcuit published a fairly easy example of how to use RFID for access management. You can find it here: http://www.instructables.com/id/MFRC522-RFID-Reader-Interfaced-With-NodeMCU/.

3. Bringing this together – Your turn
E.g. you could grab the reading from the RFID into a variable and have it displayed on the LCD.

4. Adding the connectivity and PHP part
The easiest way if you do not yet have local Webserver handy is probably to quickly install XAMPP its installer is fairly straight forward and it already comes with Apache and PHP preconfigured.

Connectivity to our NodeMCU is added by using the WiFi library and an HTTP Client library, which I mentioned earlier. As my code is so messy, I really do not want to show it here… but you can go and have a look at: https://techtutorialsx.com/2016/07/17/esp8266-http-get-requests/ to get an understanding on how to setup the NodeMCU for Wifi and how to perform HTTP requests.

In my project what I do is I basically grabbing the RFID card's IDs and sending them via HTTP to a PHP file. The php file opens a txt file stored on the server: This file contains the "allowed IDs" of the card. The php file checks if the transmitted Card ID is stored in the textfile. If yes, it replies with an "Access Granted" and the webserver itself sends an HTTP request to IFTTT using webhooks and if the ID does not match it just says "Access Denied". Additionally I log the IDs that access the main php file into another text file – this way it is super easy to add additional cards – I just need to hold it in front of the reader – it is rejected but it also is logged and this number I then can copy to the other "allowed IDs" textfile. The next time access will be granted. You could then also initiate the webserver to do something like placing a call to an IFTTT maker webhooks to trigger something using FOPEN / Wrappers.

You can use this code as an inspiration:

```
<?php
$search = $_GET['user']; //Variable from the NodeMCU
$user_file = file('users.txt'); //TXT file containing IDs with access
```

```
for ($i = 0, $found = FALSE; isset($user_file[$i]); $i++) {
 if (trim($user_file[$i]) === $search) {
 $found = TRUE;
 } }

if ($found == TRUE) {
 echo "Yeah, User found!";
 }

if ($found == FALSE) {
 echo "No user found!";
 }
?>
```

A good read to understand the PHP part is: https://davidwalsh.name/basic-php-file-handling-create-open-read-write-append-close-delete,