Universiteit Antwerpen

Title: HARDWARE NODEMCU

Satish Singh
Smetankin Daniel

Intro

In this report we're going to analyze how every hardware aspect of the NodeMCU is configured (with esp8266 on board).

Materials and Methods

NodeMCU
Breadboard
Jumper cables
HAMEG HM8012 Multimeter

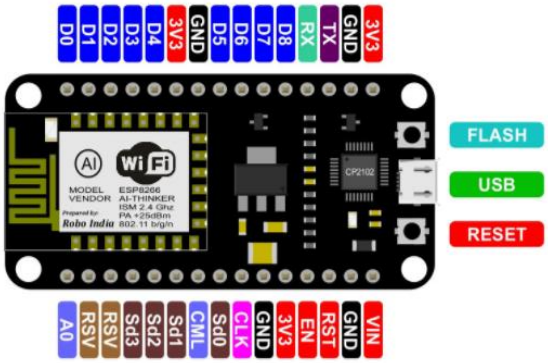
Information & conclusion

For this part we took the documents delivered with the nodeMCU. These include all the information we need for the analysis of the nodeMCU.

On this picture we see the pin configuration of the nodeMCU. We are going to use a maximum voltage of 3V. The maximum recommended to use is 3.3V. We connect our voltages to the 3V3 pin.
We use the D ports (these are the GPIO) pins to connect the functions of the car.
The other pins are unused but may come in handy. We use GND to connect to ground, RSV is reserved, CLK is to connect a clock to the nodeMCU. We won't be using as many pins as given.
The nodeMCU is equipped with a esp8266 on board.

We measured the current that the I/O Pins deliver, and this was 12mA.
We also measured the voltage that the I/O pins deliver, and this was 3.3V

The board is equipped with a micro-usb port to connect straight to our pc and write the code on the nodemcu.



This is the WiFi module that we will be using.

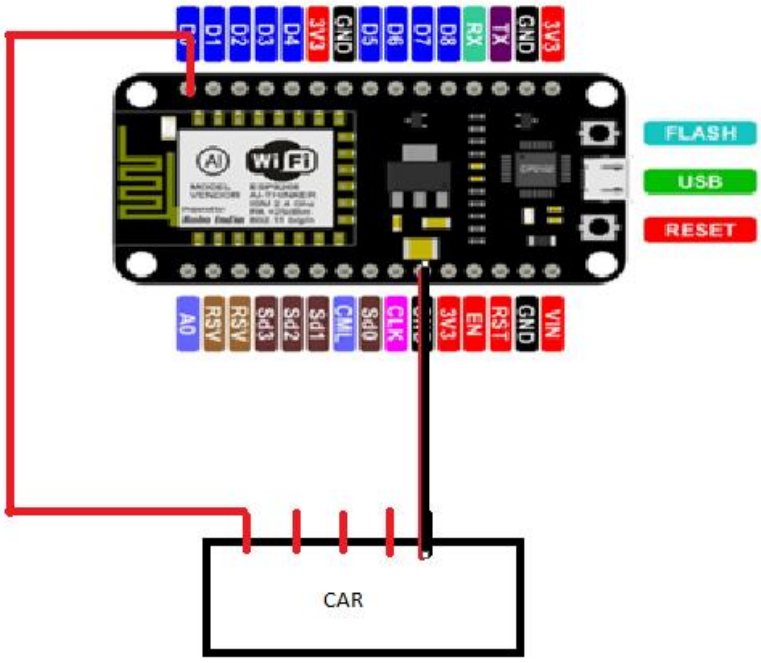
This little module has his own pins and configurations that come with it. These are connected straight to the nodeMCU.

Table 7 Absolute Maximum Ratings

Rating	Condition	Value	Unit
Storage Temperature		-40 to 125	°C
Maximum Soldering Temperature		260	°C
Supply Voltage	IPC/JEDEC J-STD-020	+3.0 to +3.6	V

Here you see that the maximum voltage we can supply is 3 – 3.6V. We will be using 3V.

To make it work we will make this configuration.

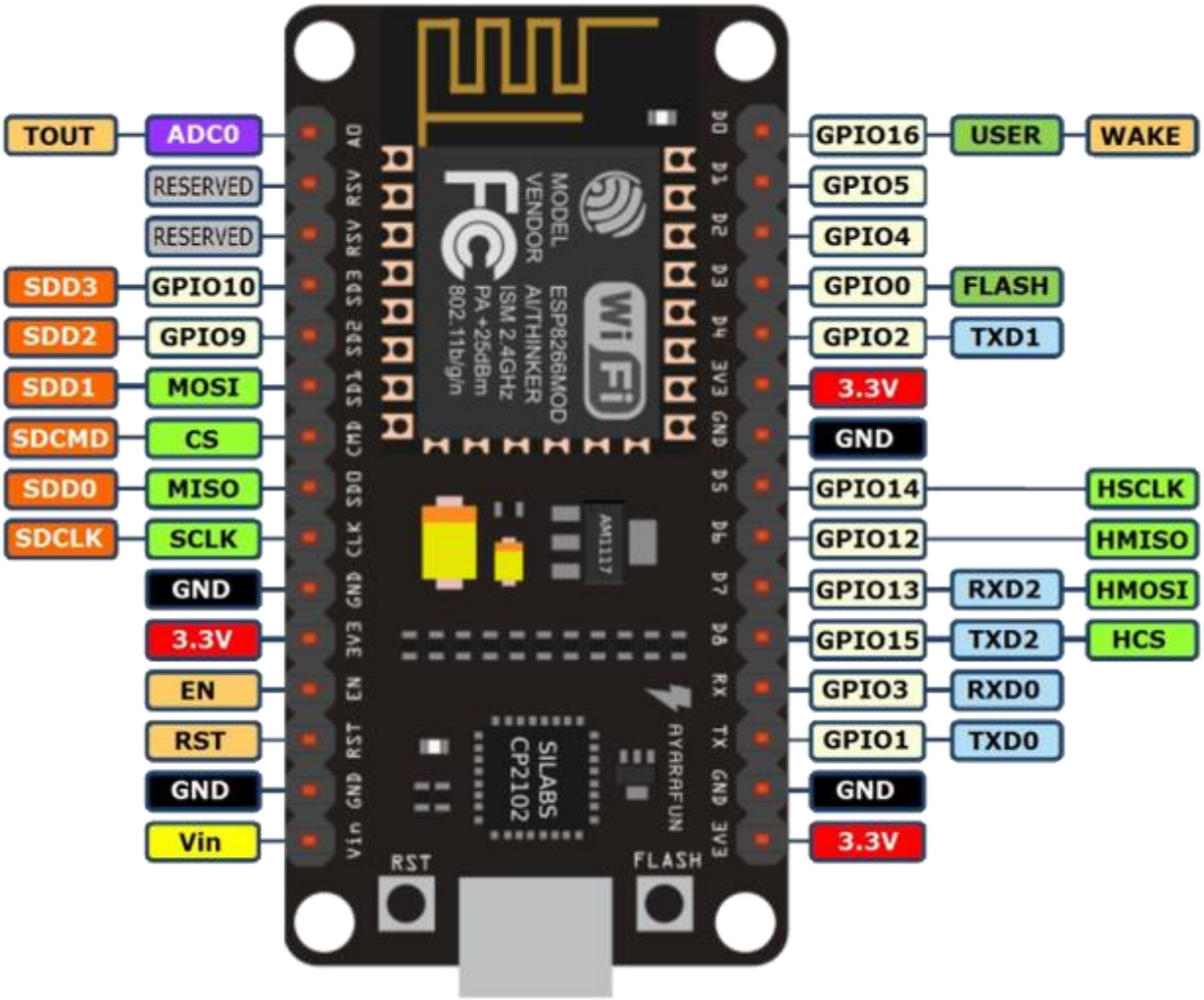


We will have to measure the output current to make sure we can put a resistor in between the car and the nodeMCU.
It Is very important to make the measurements right for the pin configuration.

To flash the code on the nodeMCU we will use this configuration.

GPIO15	GPIO0	GPIO2	Mode
0V	0V	3.3V	Uart Bootloader
0V	3.3V	3.3V	Boot sketch (SPI flash)
3.3V	x	x	SDIO mode (not used for Arduino)

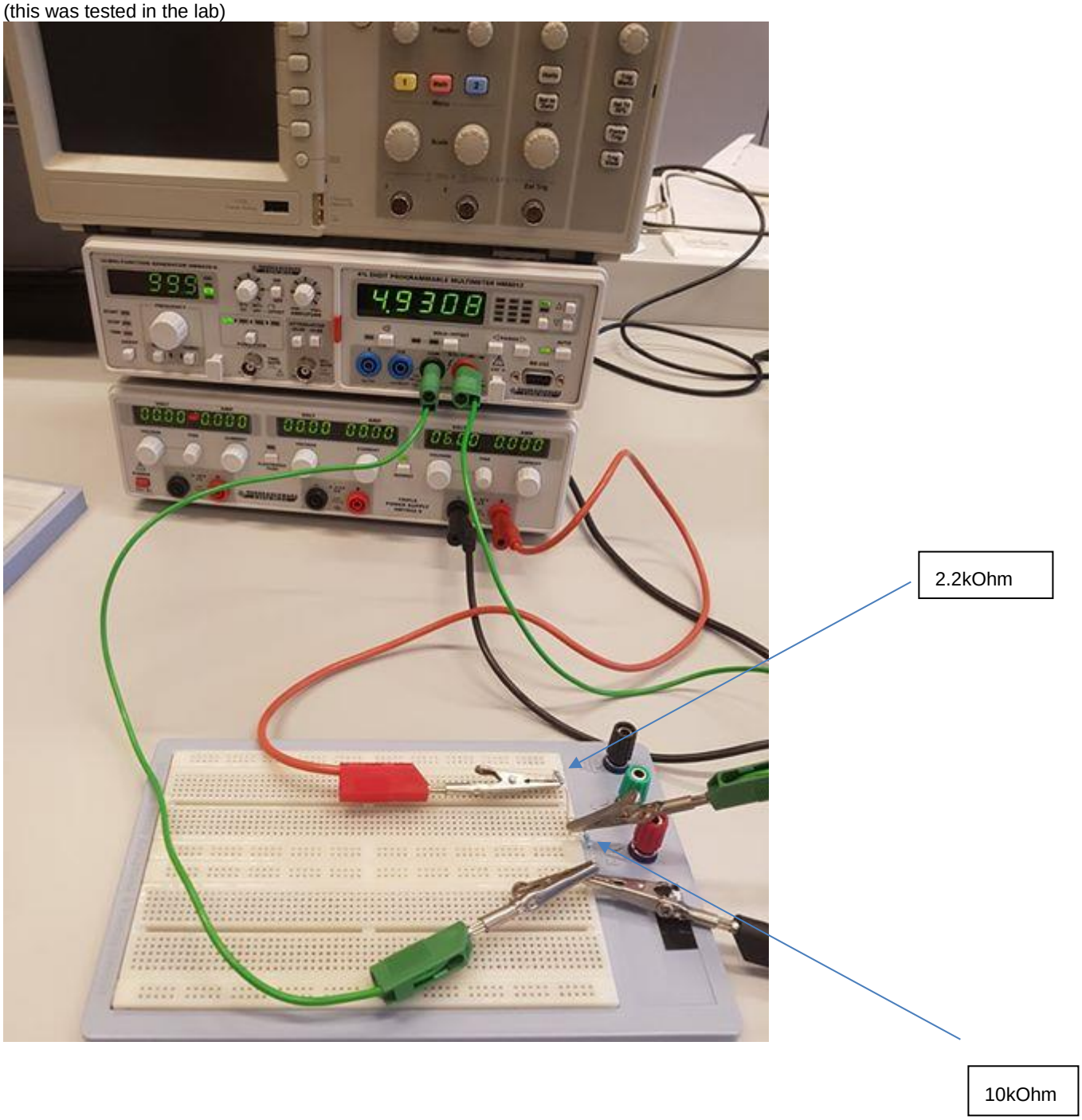
Here we see that the port 15, 0 and 2 are already taken. We can't use them for anything other than this configuration.
Port 15 Is standard 0 and doesn't have to be pulled to 0. But D3 is standard high so we have to pull it down.



On this picture you see how the nodeMCU pinout configuration is done.

On the Vin port we have to get 5V, if the voltage of the car is 6V we will use a voltage divider to get 5V.
Vout = Vin * R2/ (R1+R2)
We will choose a correct R1.

R1 = ((Vin*R2) – VoutR2)/Vout
R1 = (6*10k – 5*10k)5 = 2kOhm
(this was tested in the lab)



Referencelist

http://www.kloppenborg.net/images/blog/esp8266/esp8266-esp12e-specs.pdf
https://roboindia.com/tutorials/nodemcu-amica-esp8266-board-installation
https://tttapa.github.io/ESP8266/Chap04%20-%20Microcontroller.html
https://iotbytes.wordpress.com/nodemcu-pinout/