Universiteit Antwerpen

How do you program a nodeMCU

De Borrekens Gauthier

Intro

This report will explain the very basics of programming a nodeMCU. We will first discuss how to set up everything. This will be followed by a brief introduction on how the Arduino software works and some specifications for the hardware we need to keep in mind.

Materials and Methods

Arduino IDE
NodeMCU
Breadboard, some cables and a resistor

Information & conclusion

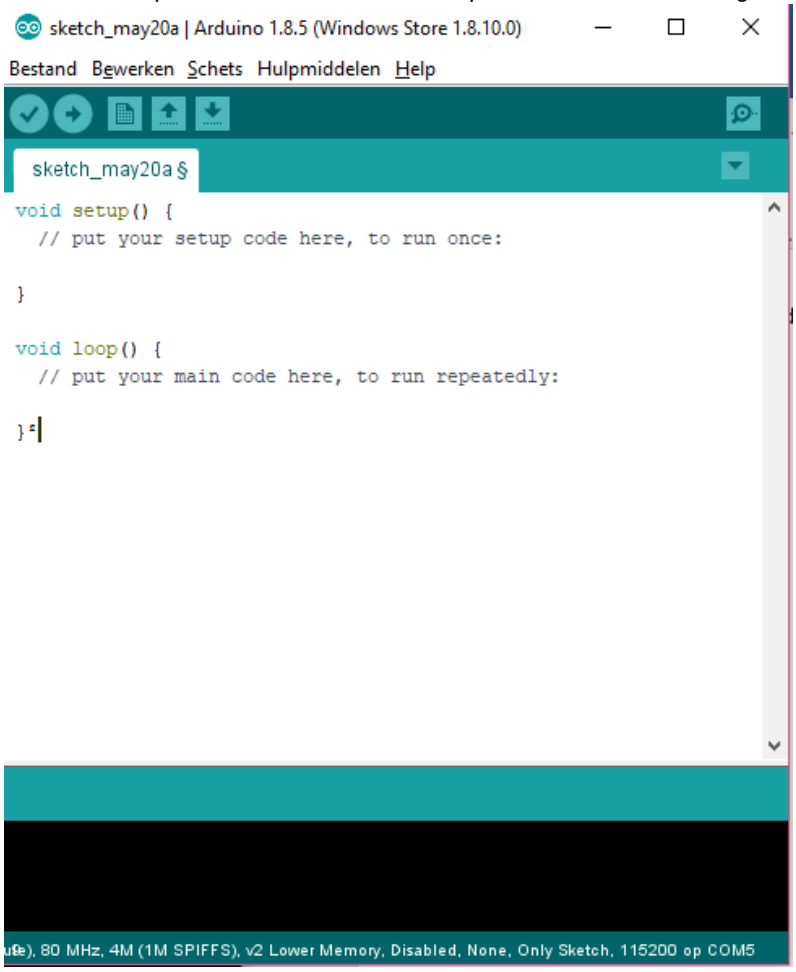
Setup

First of all, we need to pick a coding environment in which we want to work. The possibilities range from microcontroller to microcontroller and can easily be found online. In our case, ESP8266 has a couple of environments available in different languages, like LUA or java.

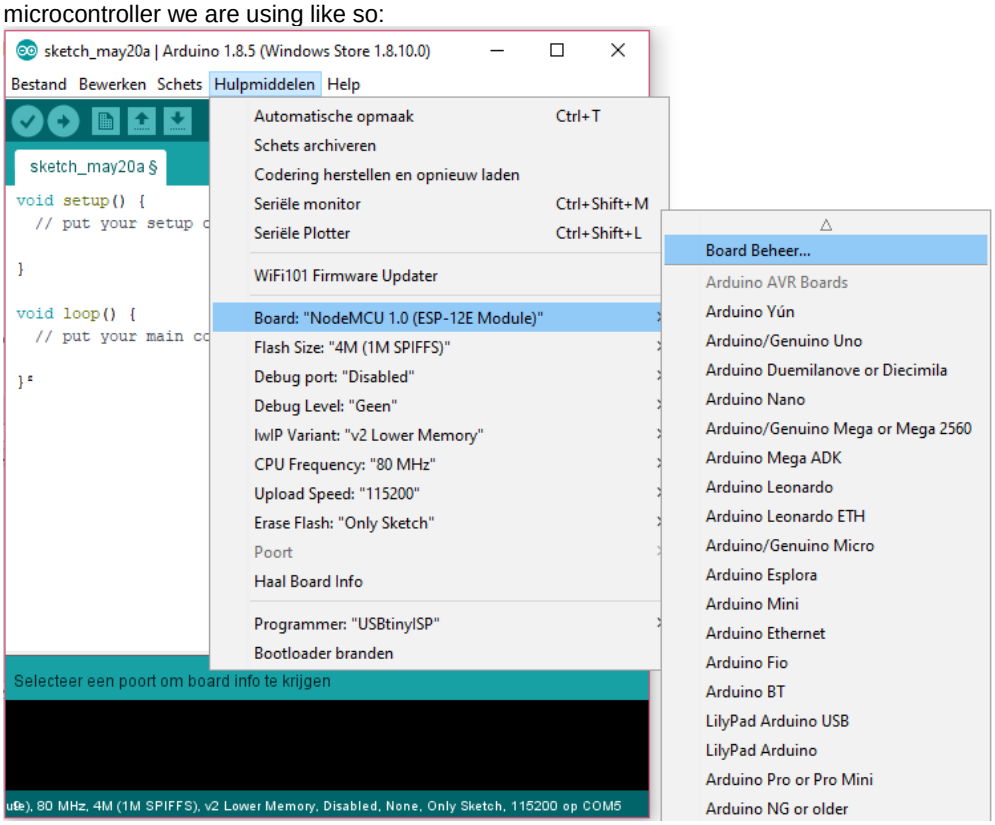
We have selected the Arduino IDE environment as it uses Java-code which we are familiar with and is well documented and widely used, which makes this ideal for looking up some examples or using existing libraries.

Arduino IDE can be downloaded from the Microsoft Store. We use the latest version, which is currently “Arduino 1.8.5”.

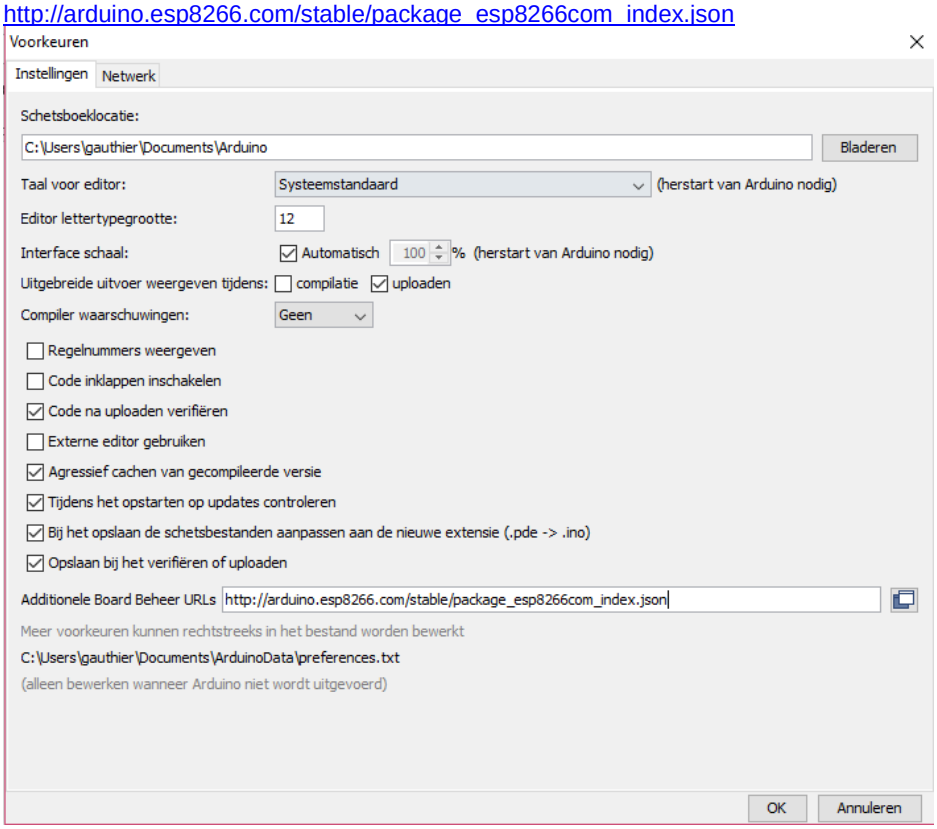
Once this is downloaded and opened, a new sketch shows up. This will look something like this:



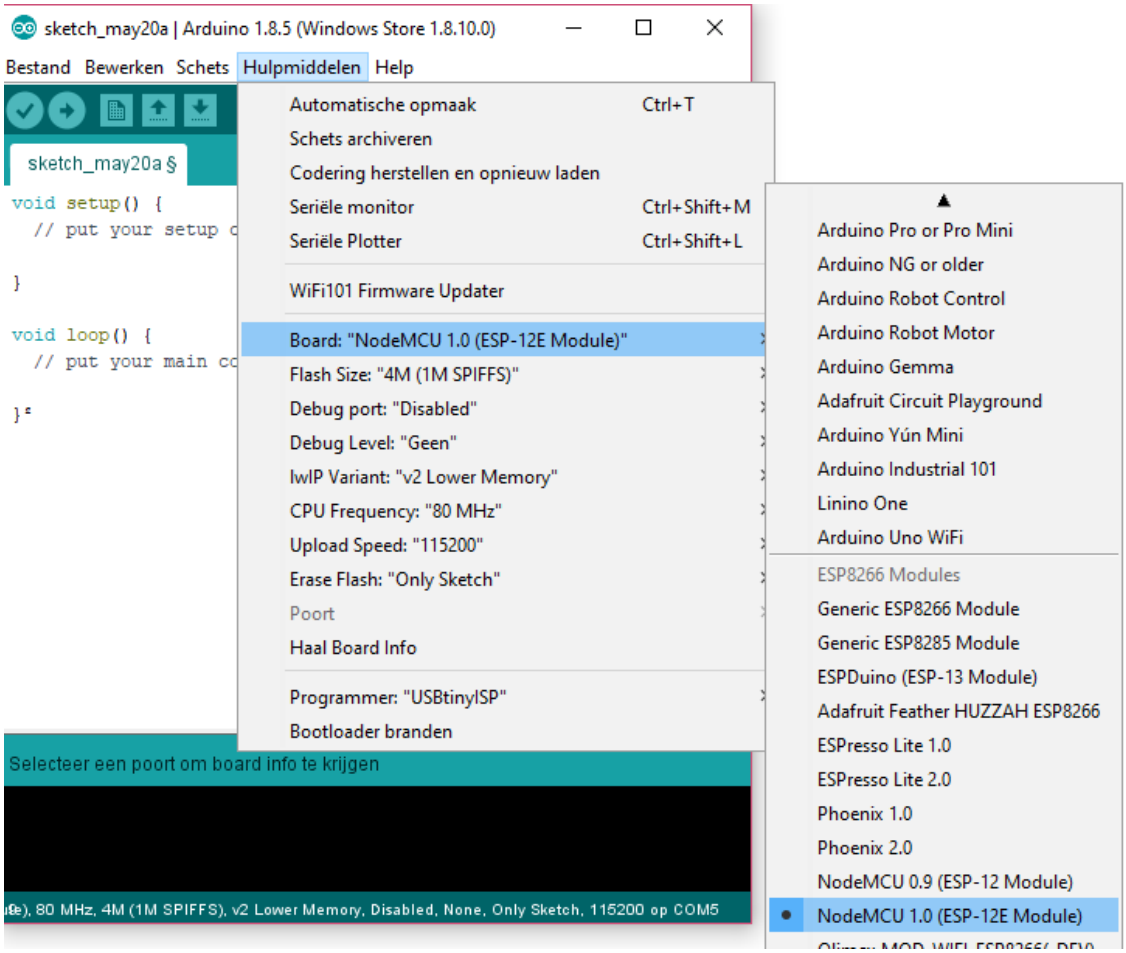
Before we head in to the coding part, we first need to configure the application to our needs. We can select which type of microcontroller we are using like so:



Unfortunately, NodeMCU is not built-in in the Arduino IDE library. To import it, we need to add an additional board manager. The menu can be opened with Ctrl+comma. The URL we need to import is http://arduino.esp8266.com/stable/package_esp8266com_index.json



Now that Arduino recognizes the NodeMCU boards, we just need to search and download the ESP8266 – package from the board manager we opened earlier. When we reopen the board select menu, we see that there are new boards added. Select the NodeMCU.



We will be using a baudrate of 115200 so this needs to be changed as well.
To flash this software on the NodeMCU, we need to put it in bootloader mode. This is done by connecting the GPIO0 pin to the ground (Careful to put a high resistor in between these pins to limit the current).
And that's it. Any code can now be uploaded by connecting the nodeMCU through USB and pressing Ctrl+U

Program

An Arduino program consists of two functions, setup(), which will run once at the start of the program, and loop(), which will continuously run once setup() is finished.

Setup

In setup, all the configurations and connections are made. Some examples are;

- Declaring the pin mode of the GPIO pins
- Starting the serial connection

These are things that need to run once and only once.

Loop

The loop-function contains the actual program. The actual thing you want to do with your code. Some examples are;

- Communicating through WiFi / Serial
- Using sensors / actuators

The biggest part of the code will be in the Loop function.

Hardware info

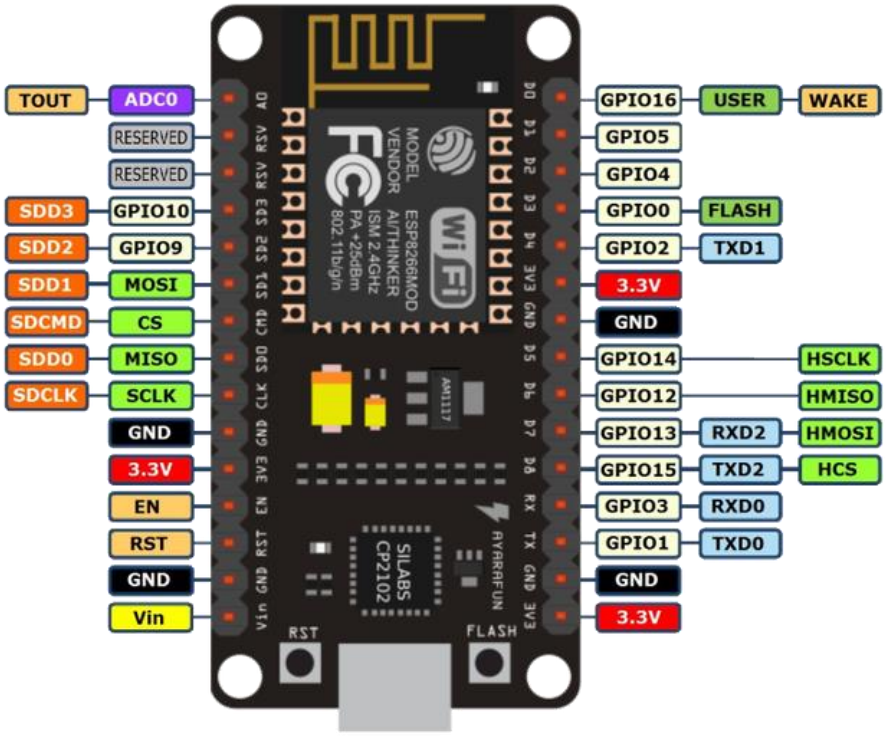
The NodeMCU has 11 usable GPIO pins (0-5 and 12-16) which provide 3.3V voltage.

Attention

Applying more than 3.6V voltage on a pin can fry the board.

Drawing more than 12mA current from a pin can damage the board.

For the pins, the code uses the GPIO index, but the board uses d-notation. The relation is seen in the pin diagram.



For example: If you want to connect something to pin D4, this pin will need to have index “2” in the Arduino code as D4 stands for GPIO2.

Referencelist

https://tttapa.github.io/ESP8266/Chap03%20-%20Software.html

Extra Documents

See the Arduino Code-document