

# Internet of Things (IoT)

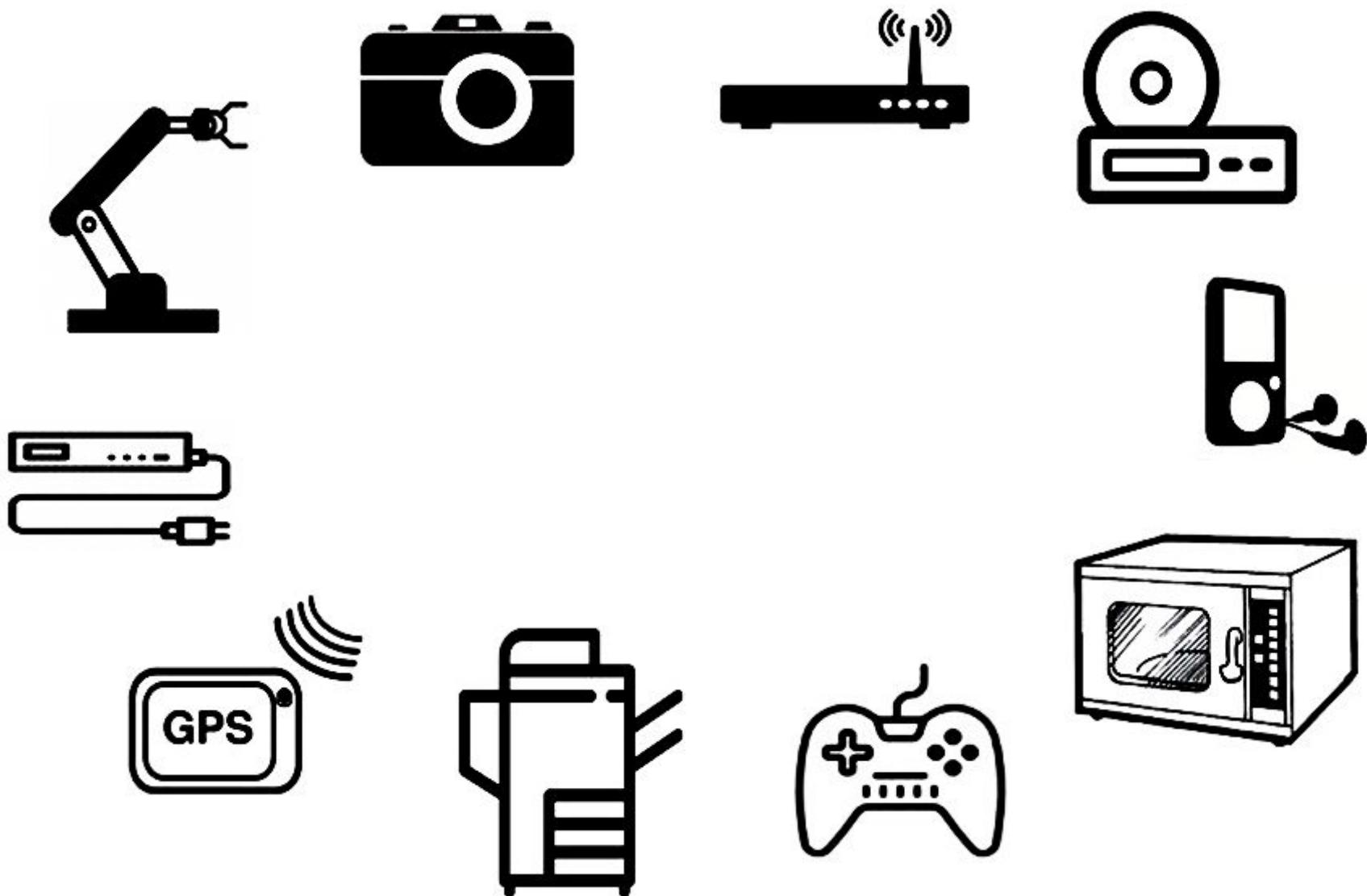
## Make It Yourself



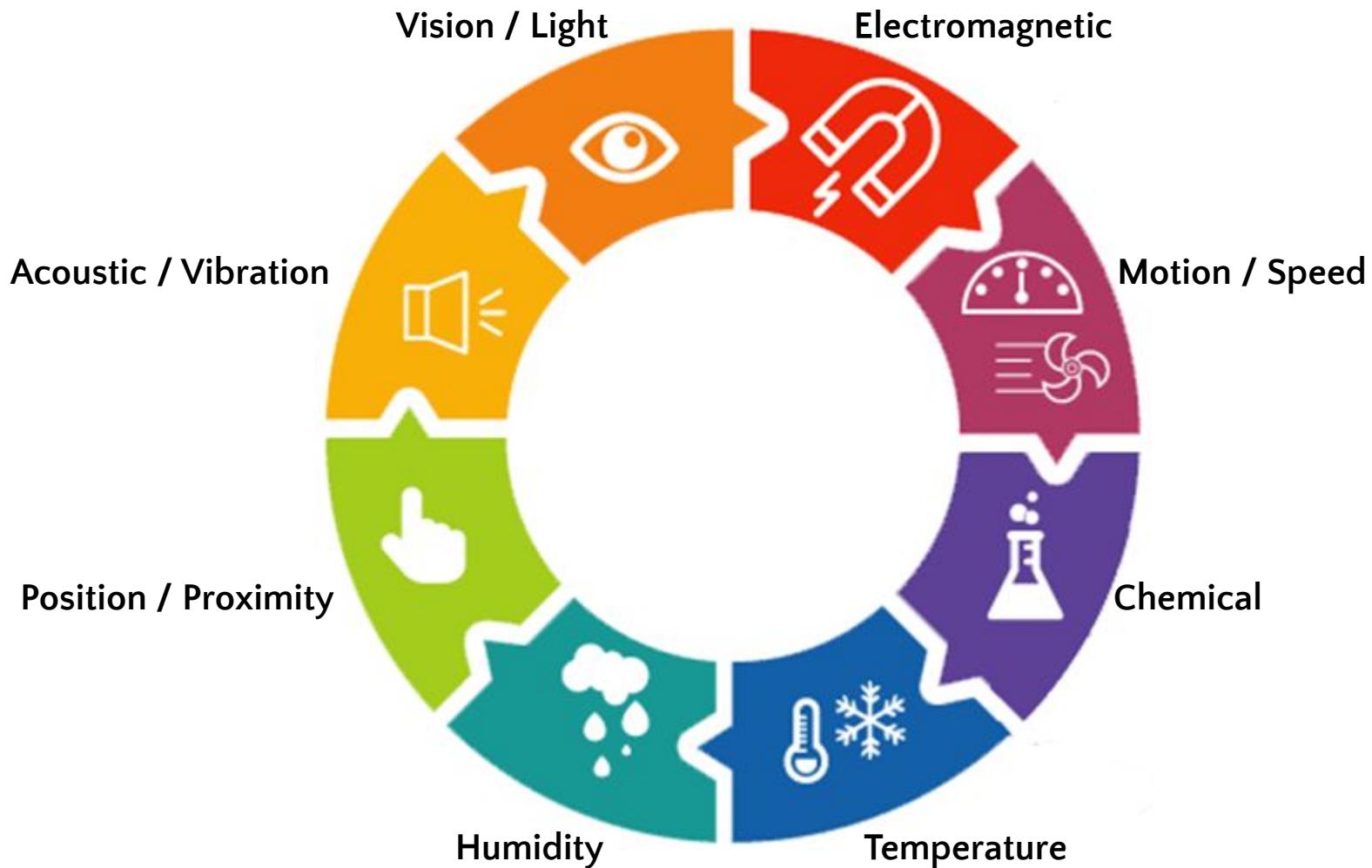
# Overview

- **Introduction to Internet of Things (IoT)**
- **IoT System Components**
  - Embedded Systems
    - Hardware & Software
    - Sensors & Actuators
  - Communication Network
- **Hands-On with IoT Development Platforms**
  - Hardware – NodeMCU
  - Software – Arduino Integrated Development Environment (IDE)
  - Do It Yourself (DIY) Experiments
- **Familiarizing Cloud Platforms**
  - ThingSpeak – [www.thingspeak.com](http://www.thingspeak.com)
  - Ubidots – [www.ubidots.com](http://www.ubidots.com)
- **IoT Projects**

# Embedded Systems



# Sensors



# Actuators



Image Source: <https://techsee.me/wp-content/uploads/2018/10/IoT-onboarding.jpg>

# Programming



# Cloud



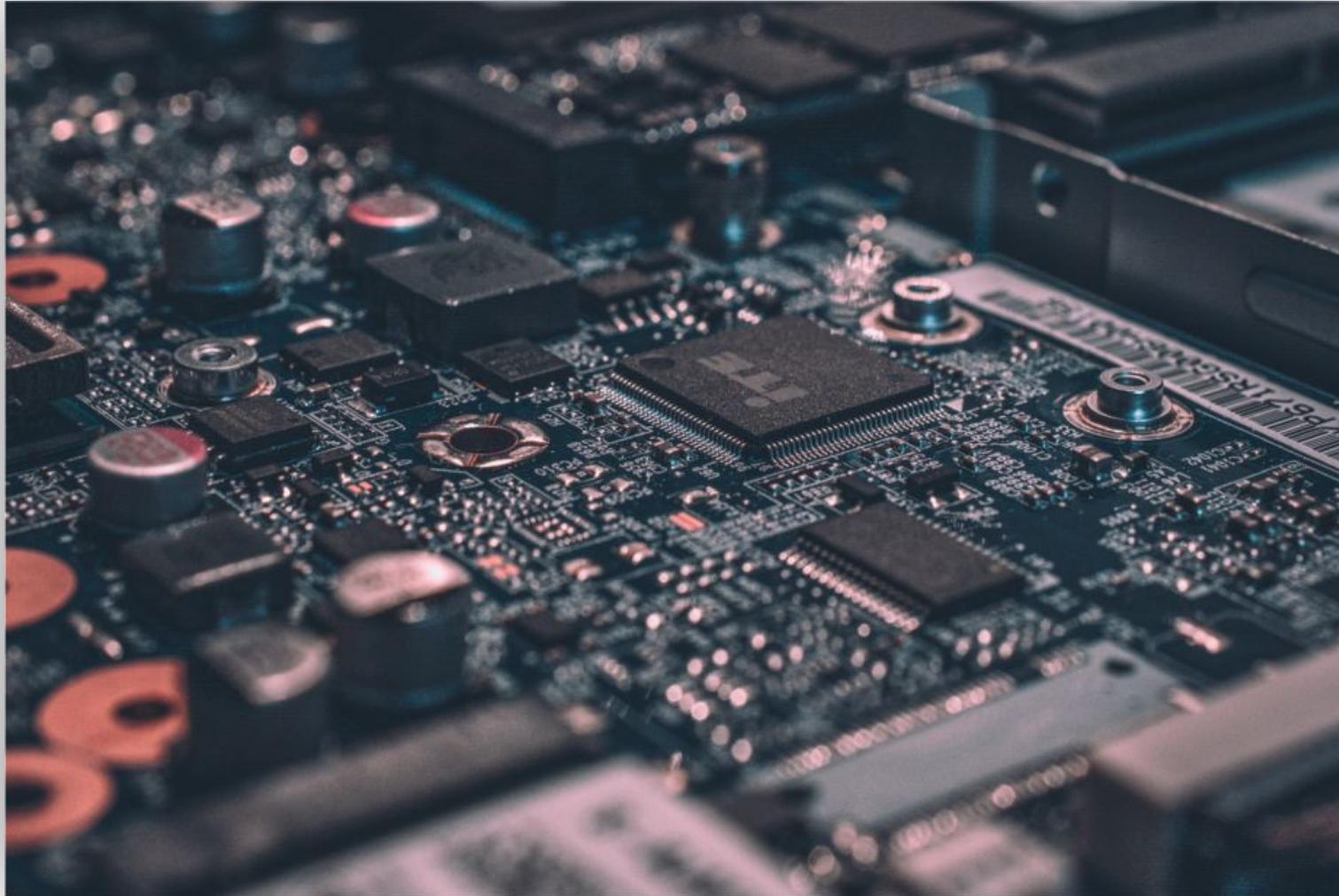
Image Source: <https://d15sh11kswkct0.cloudfront.net/wp-content/blogs.dir/1/files/2011/09/cloud-computing-3.jpg>

# Internet of Things (IoT)



Image Source: Wikipedia®

# Embedded Systems



# Embedded Systems ...



Image Source: [https://images-na.ssl-images-amazon.com/images/I/81pTG21EL7L.\\_SL1500\\_.jpg](https://images-na.ssl-images-amazon.com/images/I/81pTG21EL7L._SL1500_.jpg)

# Embedded Systems ...



Image Source: <https://ae01.alicdn.com>

# Embedded System meaning

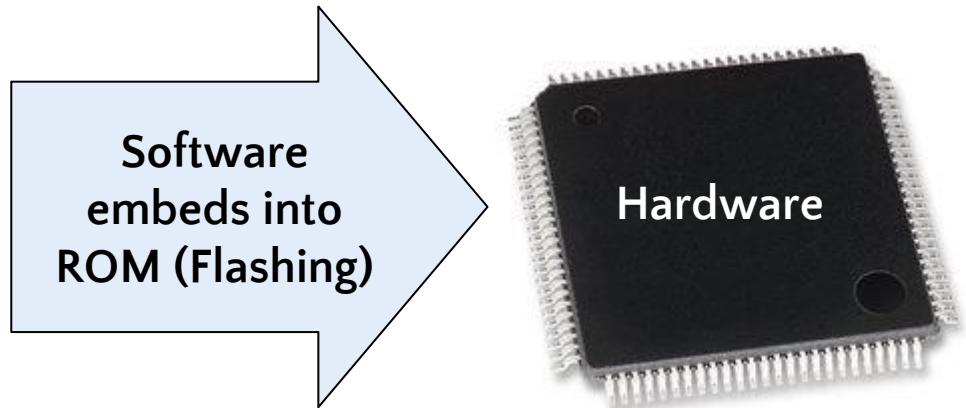
An **Embedded System** is one that has Computer Hardware with Software embedded inside.

## Software Program

```
#include "NodeMCU.h"
#define ledPin D0

void setup()
{
    pinMode(ledPin, OUTPUT);
}

void loop()
{
    digitalWrite(ledPin, HIGH);
    delay (1000);
    digitalWrite(ledPin, LOW);
    delay (1000);
}
```



# Roles of Embedded Systems in Modern World



Health Care



Transportation



Home Appliances



Multimedia



Defence



Manufacturing

# Demonstration

## Obstacle Range Finder

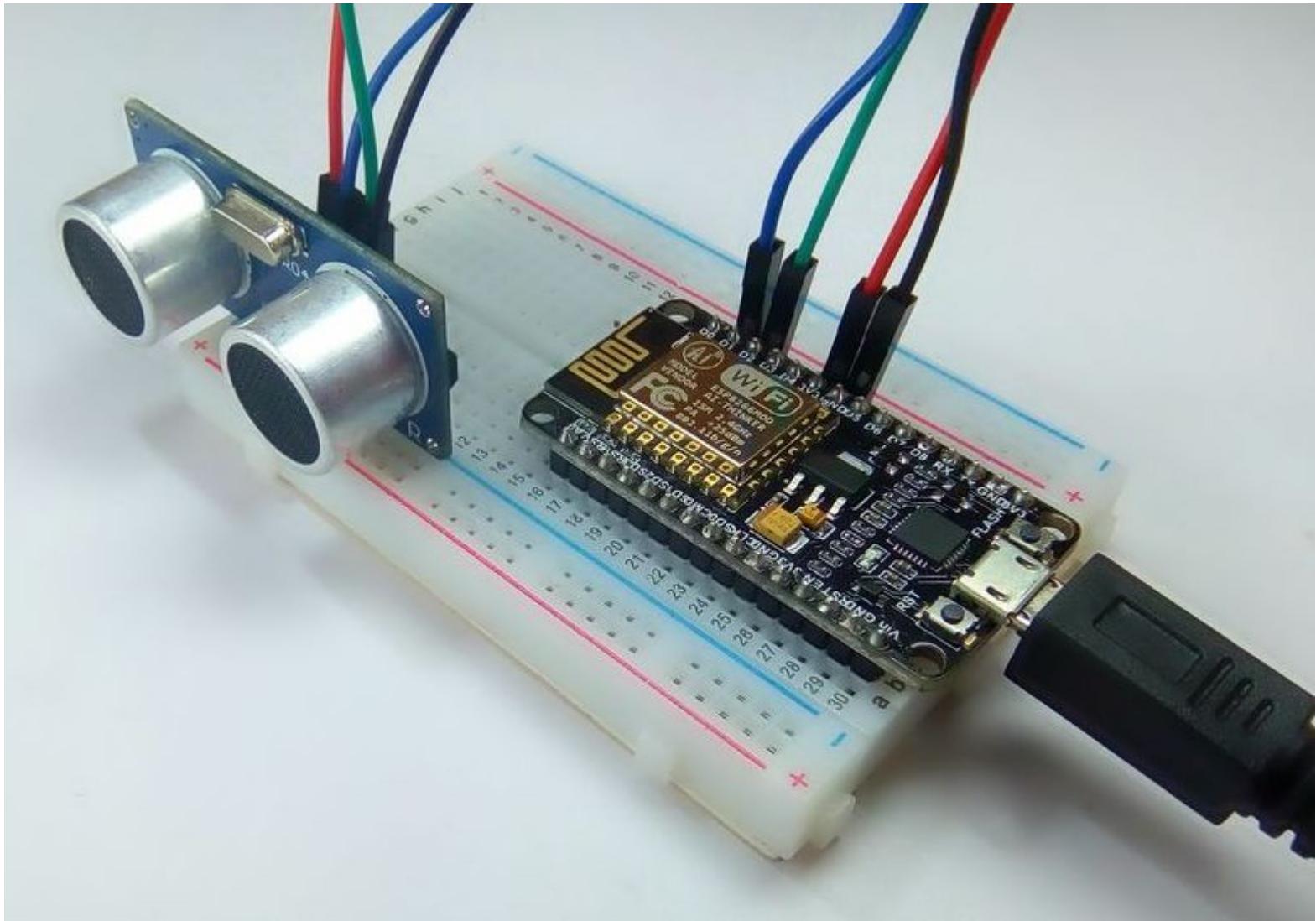
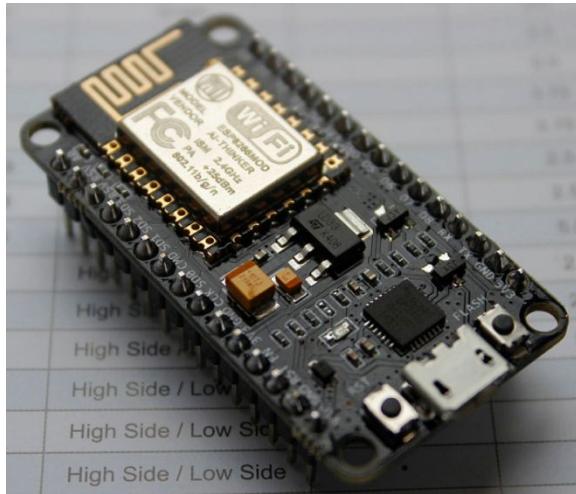


Image Source: <https://cdn.instructables.com/F9F/7UFQ/J48Q4APW/F9F7UFQJ48Q4APW.LARGE.jpg>

# Demonstration

## Obstacle Range Finder - Components

### Hardware



#### NodeMCU

ESP8266 WiFi SoC based Open Source  
Hardware Development Platform



#### Ultrasonic Sensor

HC-SR04  
5V-12V, 0.2 – 0.45m range  
with 0.03m precision

### Software



#### ARDUINO 1.8.9

The open-source Arduino Software (IDE) makes it easy to write code and upload it to the board. It runs on Windows, Mac OS X, and Linux. The environment is written in Java and based on Processing and other open-source software.

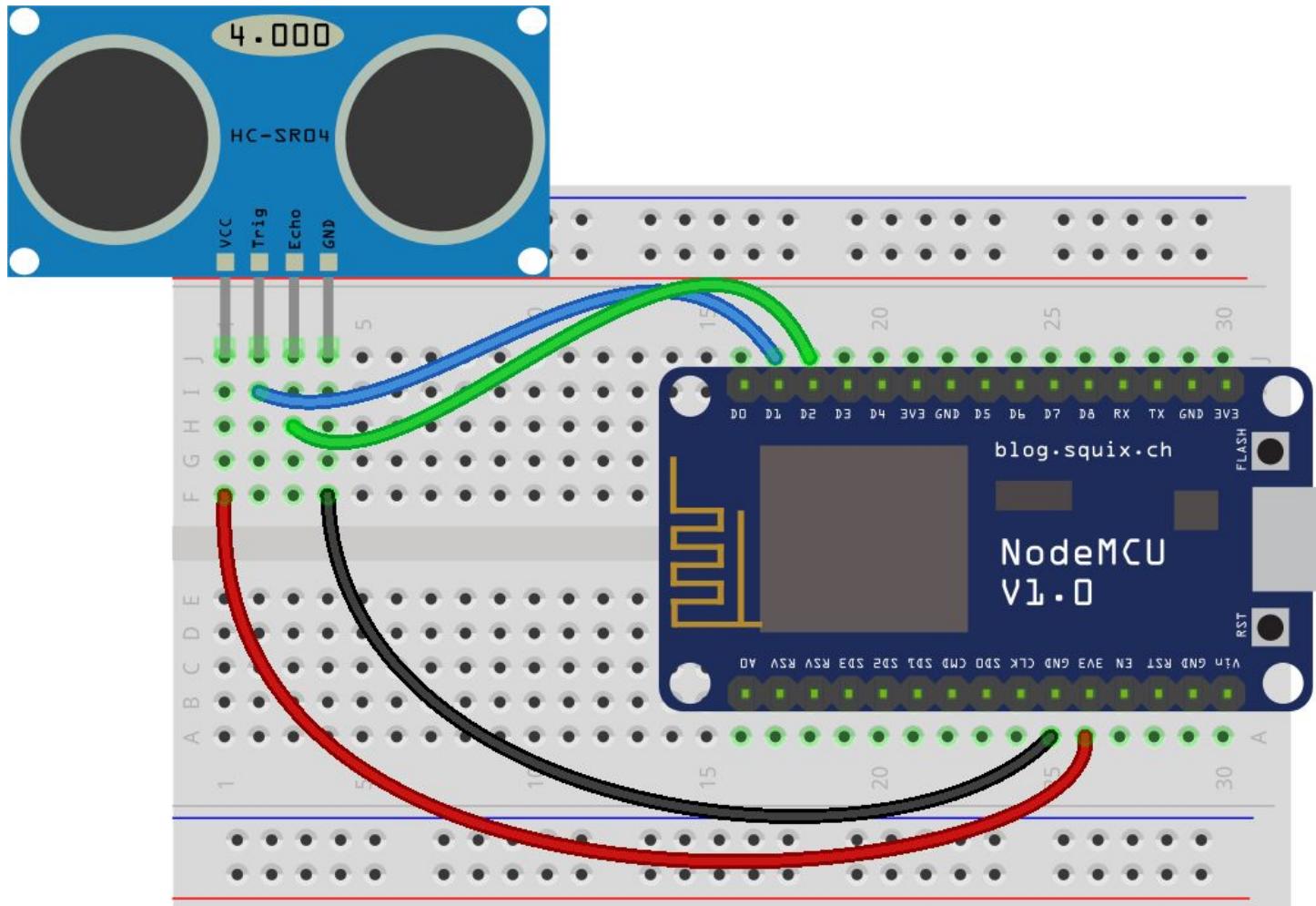
This software can be used with any Arduino board. Refer to the [Getting Started](#) page for [Installation](#) instructions.

#### Arduino IDE

C++ based Embedded Software  
Integrated Development Platform

# Demonstration

## Obstacle Range Finder - Wiring



Layout designed using Fritzing® - <http://fritzing.org>

NodeMCU part Source: <https://github.com/squix78/esp8266-fritzing-parts/tree/master/nodemcu-v1.0>

fritzing

# Internet of Things

# What is IoT?

**Internet of things (IoT)** refers to the concept of connecting **any thing** to the Internet. It is the Idea that people can communicate with the objects and physical things

**Things** can be Humans, Animals, Buildings, Energy Stations, Smartphones, Tablets, Bicycle, Sensors, Cameras, Vehicles, Health monitoring devices and so on...

# Choosing the right Hardware Platform

## Major Factors

- Processor
- Chipset
- Memory
- On-Board Peripherals
  - I<sup>2</sup>C, SPI, CAN, USB, HDMI, Parallel ports etc.
- System Clock options
- Software / Firmware Development support
- Cost

# Smart Wrist Band

Heart Rate Monitor



Pedometer



Sports Monitor



Alarm Clock



Sedentary Monitor



Calorie Monitor



Call Alert



Sleep Monitor

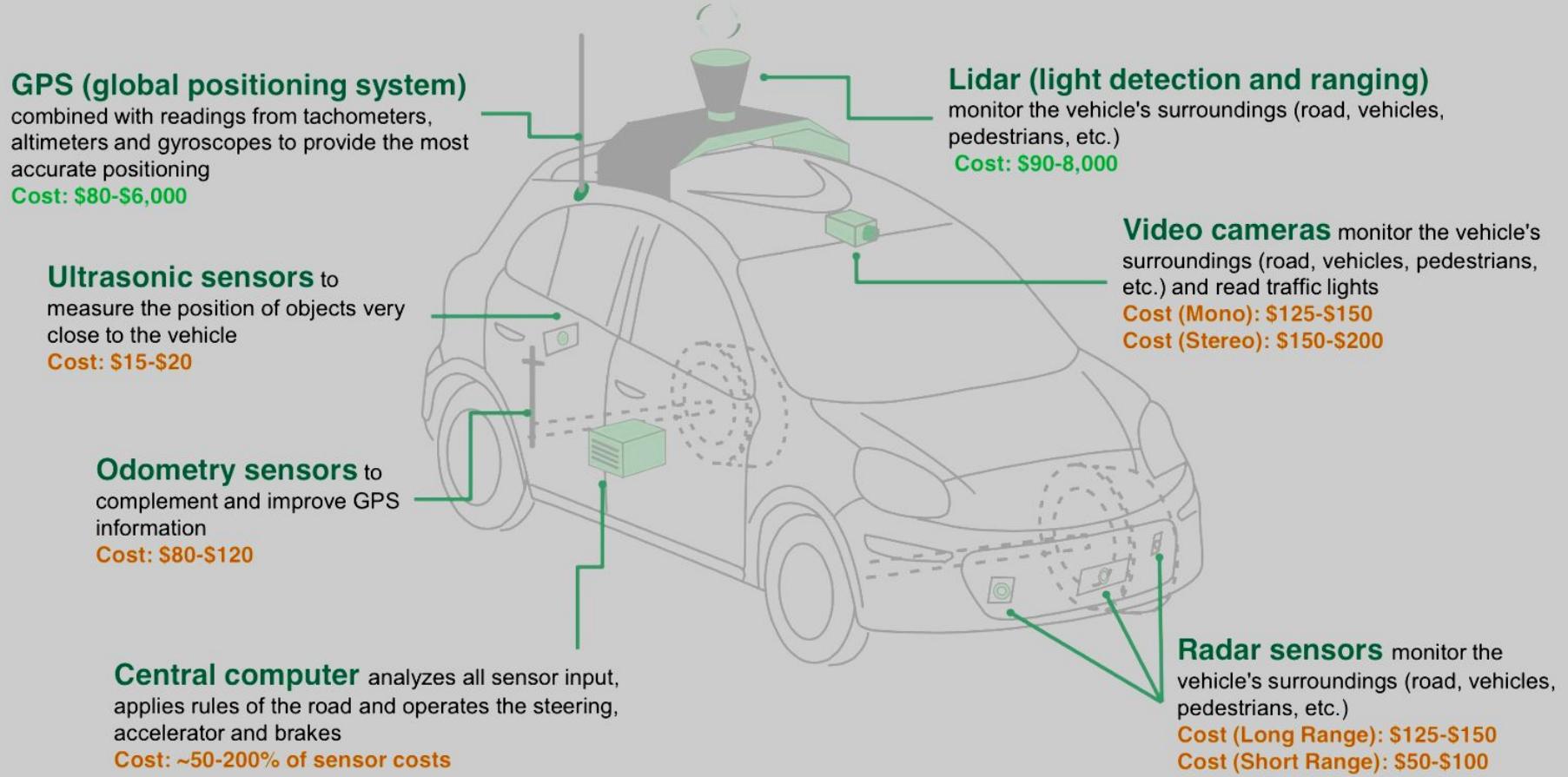


# Driverless Cars



Image Source: <https://fortunedotcom.files.wordpress.com/2014/05/google-gadgets-car.jpg>

# Driverless Cars



# Smart Power Outlets



Wireless Control · Energy Monitoring & Conservation · Smart Device Integration

# Smart Key Chains & ID Cards



# Things !!!



# How Things Communicate ?

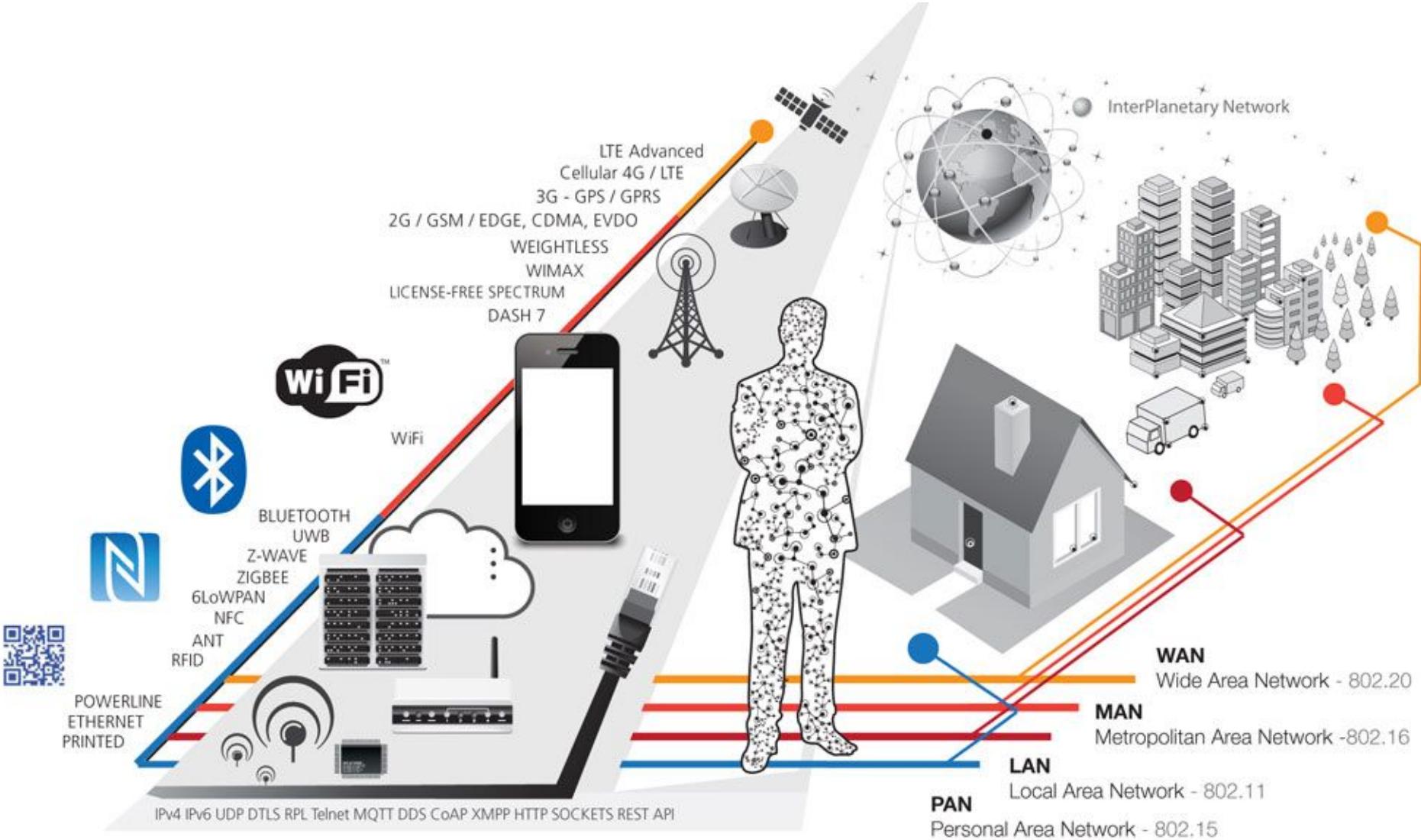
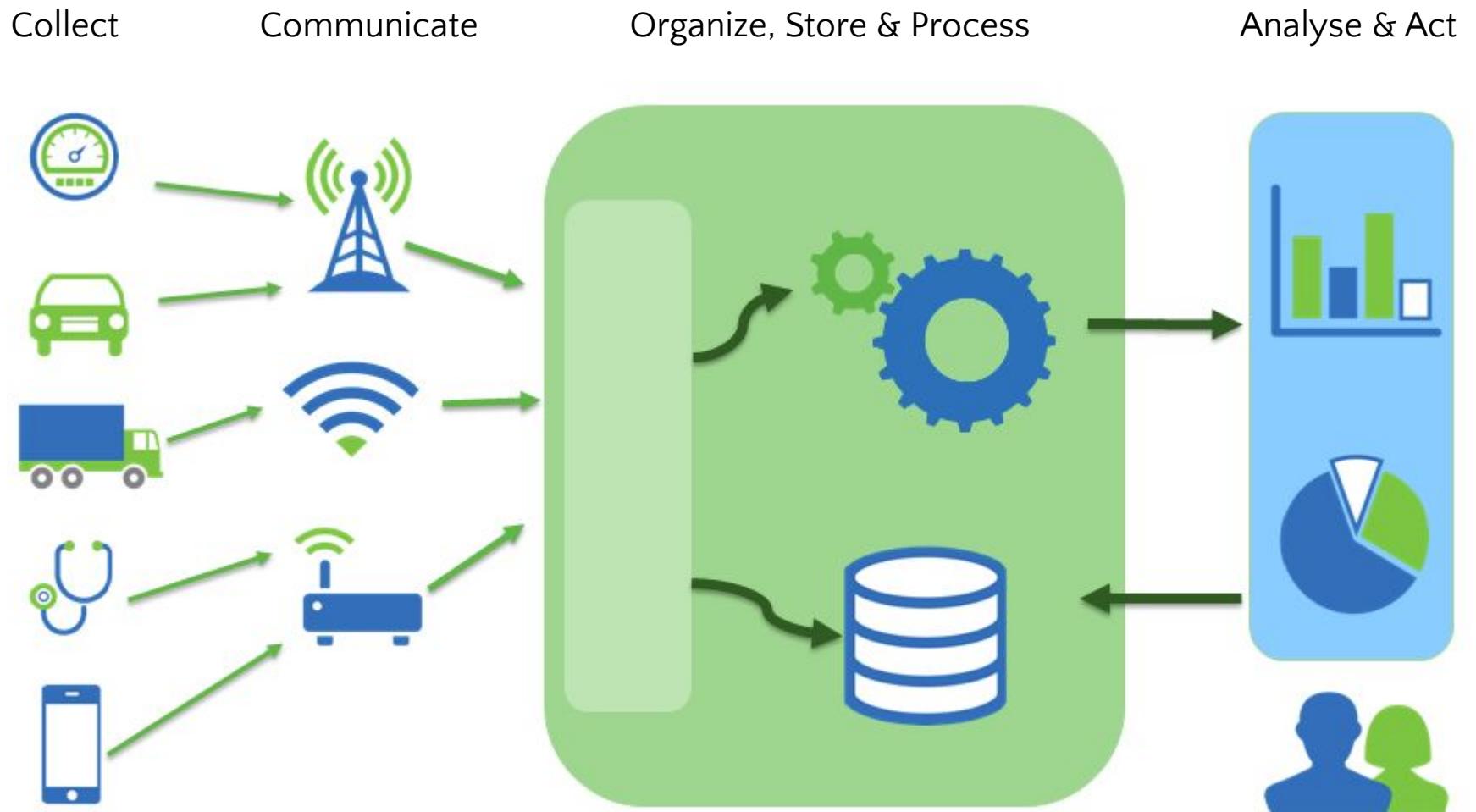


Image Source: <https://www.postscapes.com/wp-content/uploads/2018/03/connectivity-diagram.jpg>

# Data Analysis & Action



# IoT Life Cycle



# NodeMCU Development Board

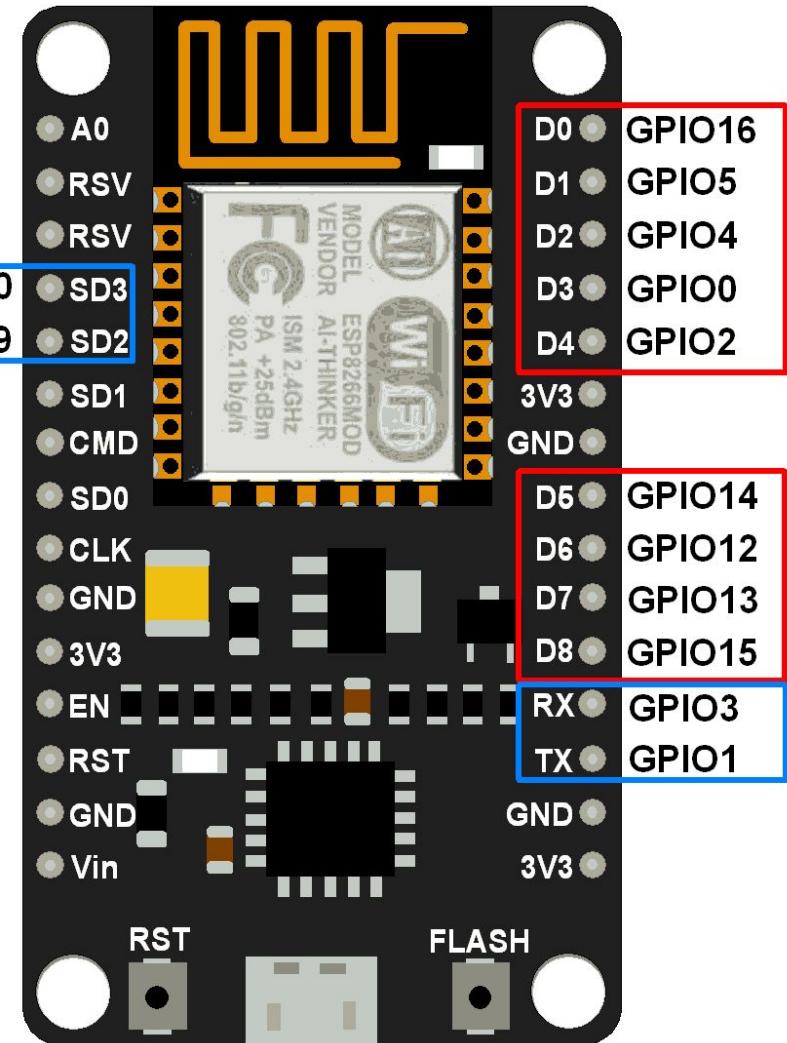


Image Source: <https://dziadainfpolx.cloudfront.net/blog/wp-content/uploads/2015/09/official-nodemcu-development-board.jpg>

# NodeMCU Development Board

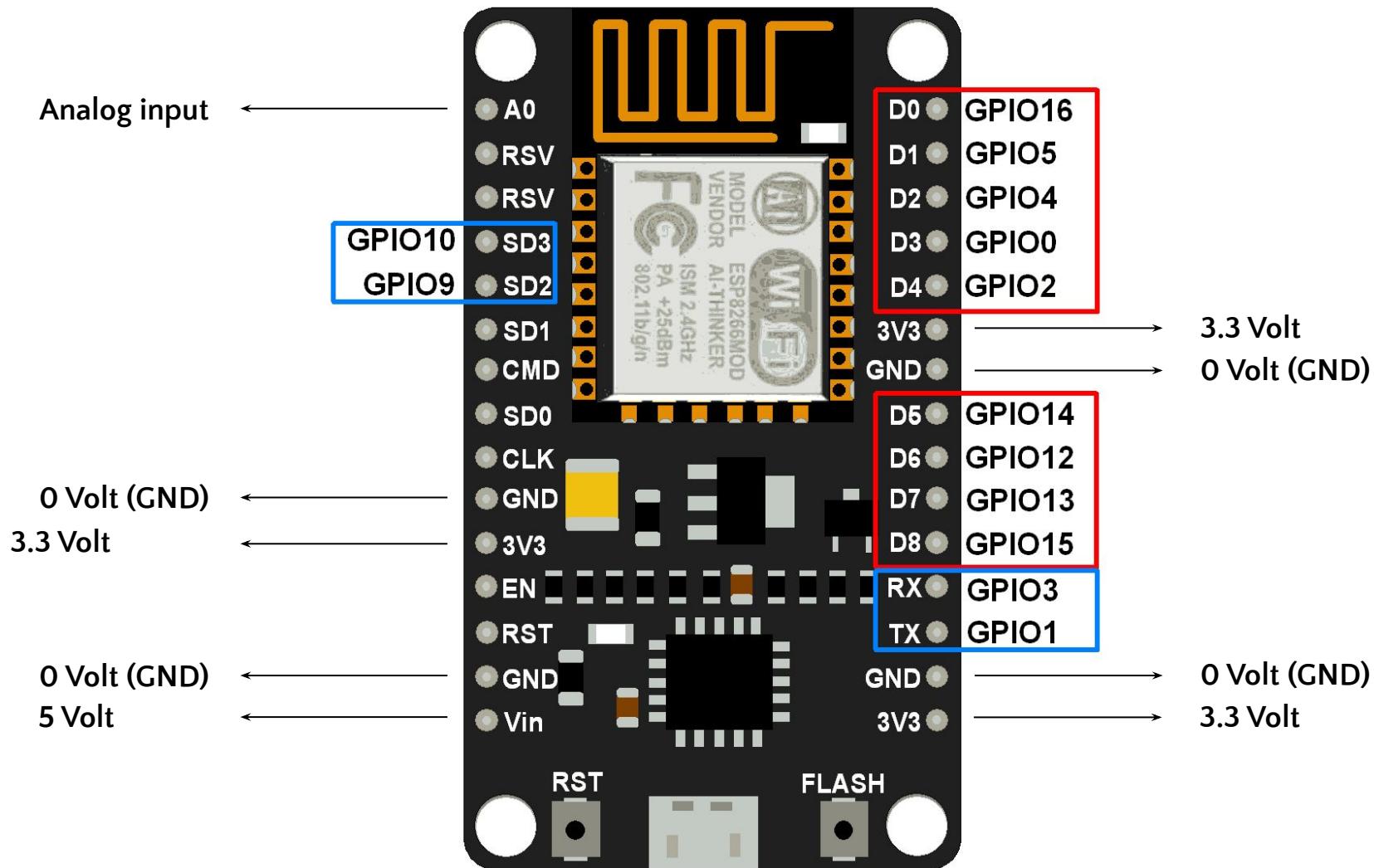
## Specification & Pin Out

- Tensilica L106, 32-bit RISC Processor, 160 MHz
- 9 Digital I/O Pins (digital input/output pins)
- 1 Analog pin (10 Bit ADC)
- 4MB Flash (Program) memory
- 128KB Data Memory
- Inbuilt WiFi 802.11 b/g/n
- UART(2), SPI(2), I<sup>2</sup>C (1)
- USB Powered
- Operating Voltage 3.3V
- 3.3V & 5V Outputs available for external use



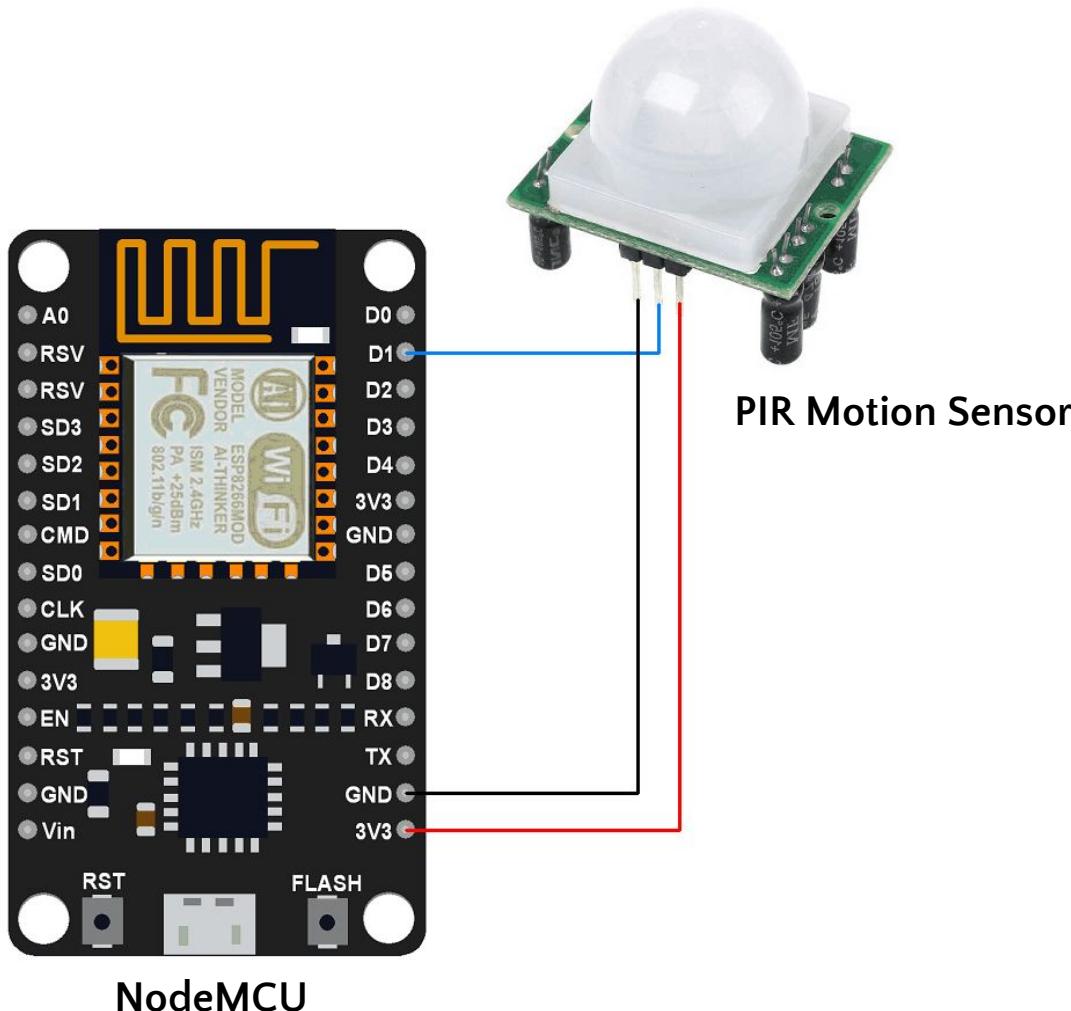
# NodeMCU Development Board

## Pin Out



# DIY Experiments with NodeMCU

## Interfacing PIR Motion Sensor – HCR501



# Software Development Platform

Arduino® Integrated Development Environment (IDE)



## ARDUINO

The open-source Arduino Software (IDE) makes it easy to write code and upload it to the board. It runs on Windows, Mac OS X, and Linux. The environment is written in Java and based on Processing and other open-source software.

This software can be used with any Arduino board. Refer to the [Getting Started](#) page for installation instructions.

<https://www.arduino.cc>

**Windows** Installer, for Windows XP and up

**Windows** ZIP file for non admin install

**Windows app** Requires Win 8.1 or 10



**Mac OS X** 10.8 Mountain Lion or newer

**Linux** 32 bits

**Linux** 64 bits

**Linux ARM** 32 bits

**Linux ARM** 64 bits

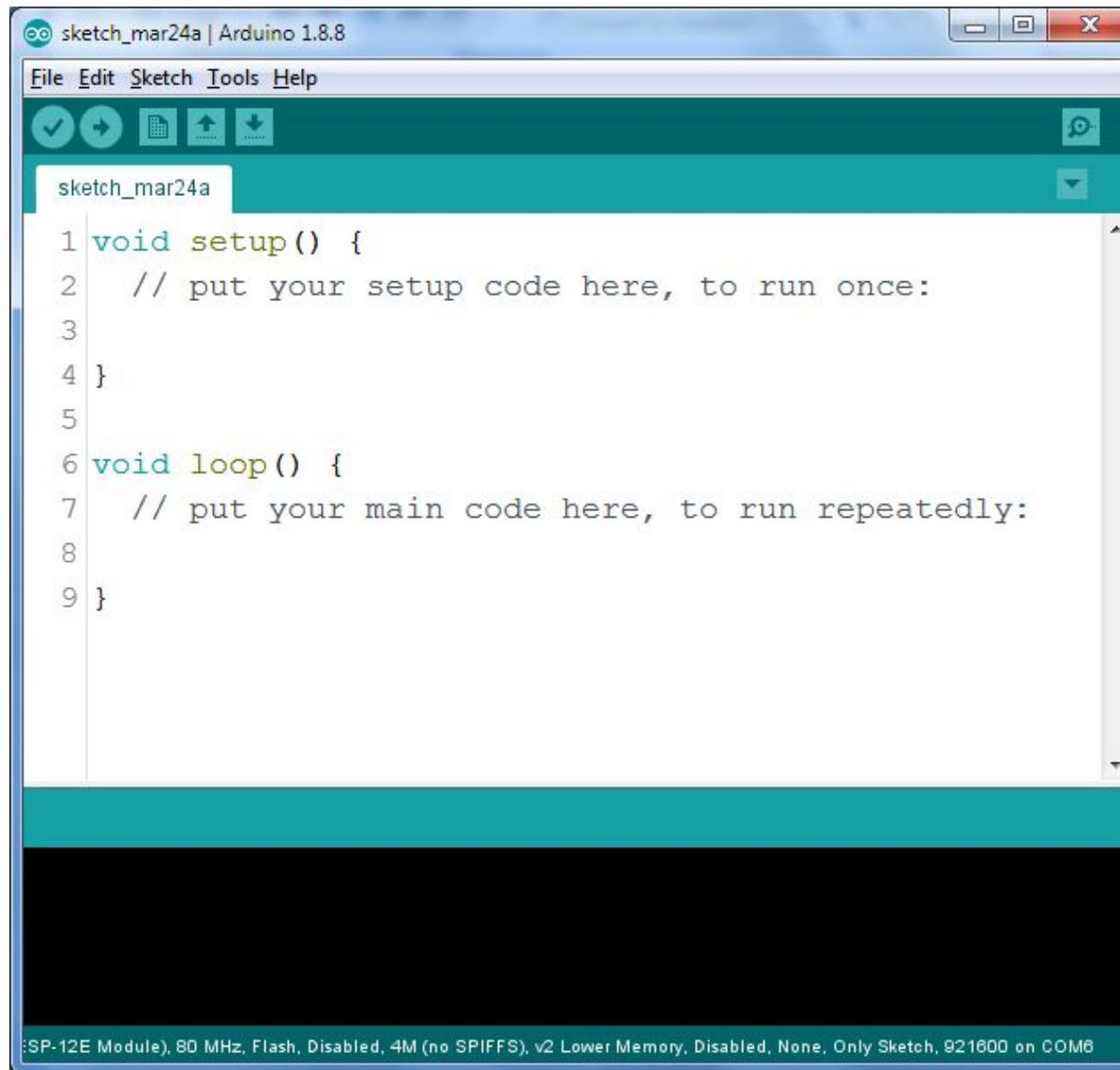
[Release Notes](#)

[Source Code](#)

[Checksums \(sha512\)](#)

# Software Development Platform

## Arduino® Integrated Development Environment (IDE)

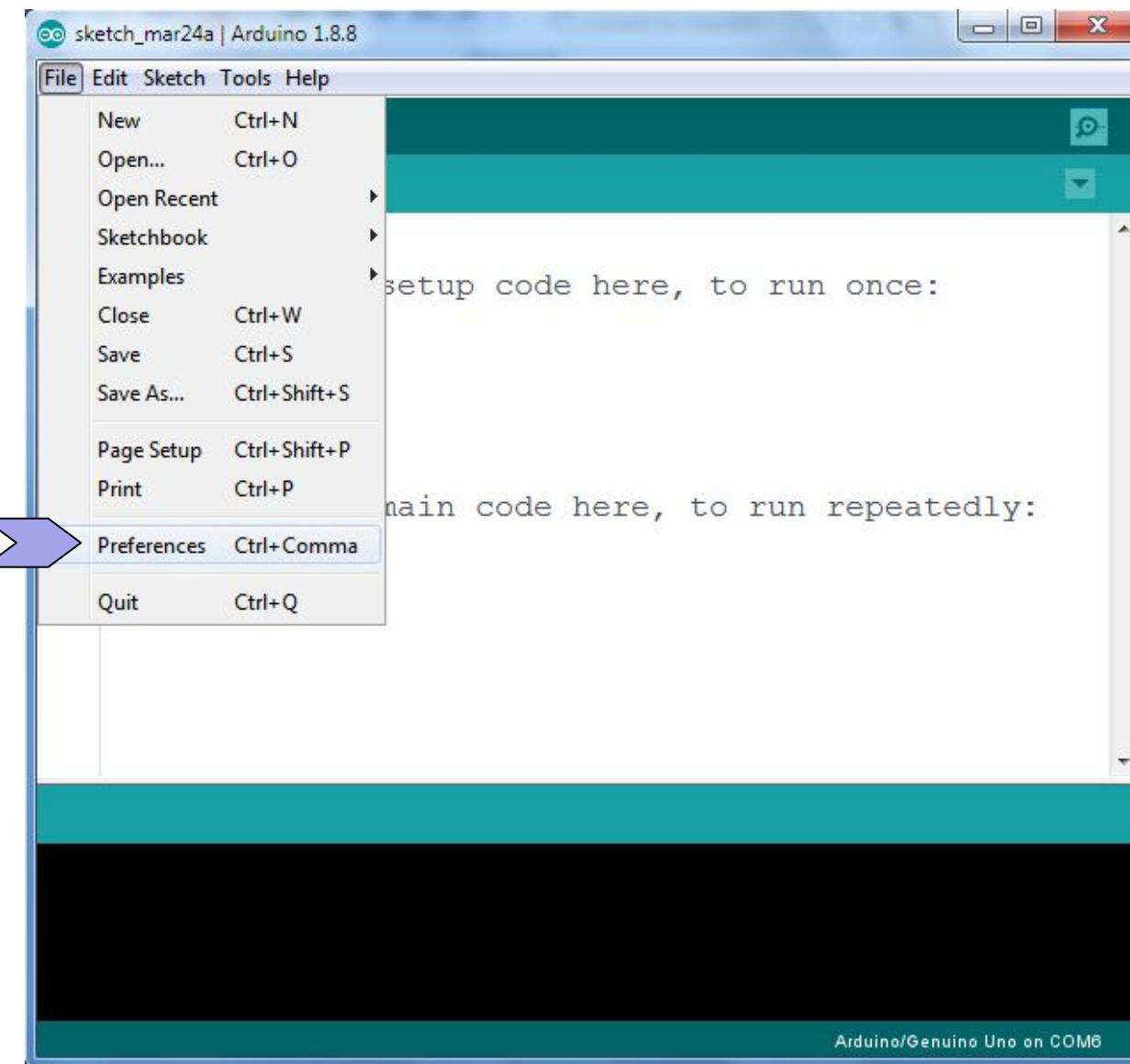


The screenshot shows the Arduino IDE interface with the following details:

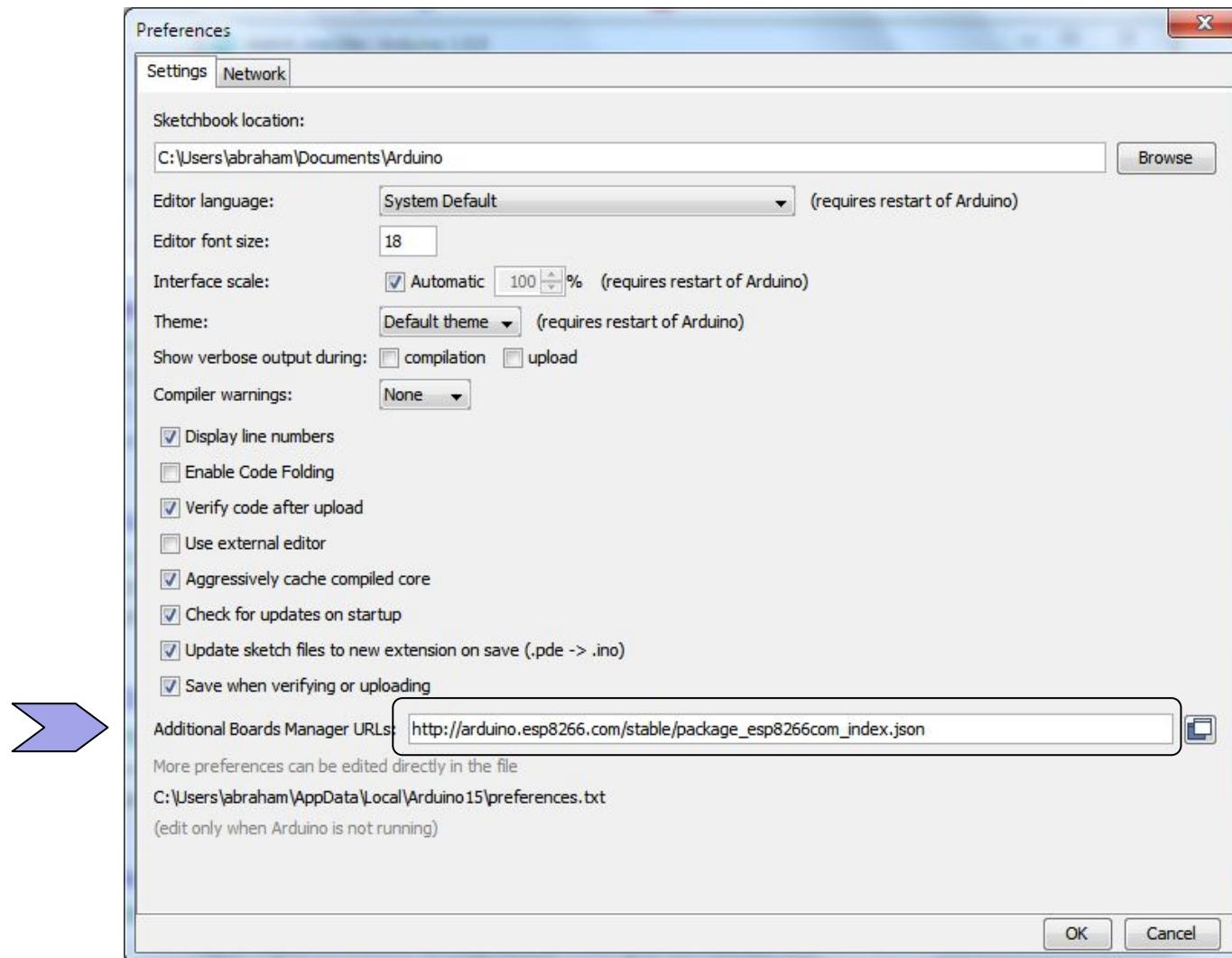
- Title Bar:** sketch\_mar24a | Arduino 1.8.8
- Menu Bar:** File Edit Sketch Tools Help
- Toolbar:** Includes icons for Save, Undo, Redo, Open, Upload, and Download.
- Sketch Editor:** Displays the code for 'sketch\_mar24a'. The code consists of two functions: setup() and loop().

```
1 void setup() {  
2     // put your setup code here, to run once:  
3  
4 }  
5  
6 void loop() {  
7     // put your main code here, to run repeatedly:  
8  
9 }
```
- Status Bar:** Shows the board configuration: (ESP-12E Module), 80 MHz, Flash, Disabled, 4M (no SPIFFS), v2 Lower Memory, Disabled, None, Only Sketch, 921600 on COM6.

# Setting up Arduino IDE for NodeMCU Board

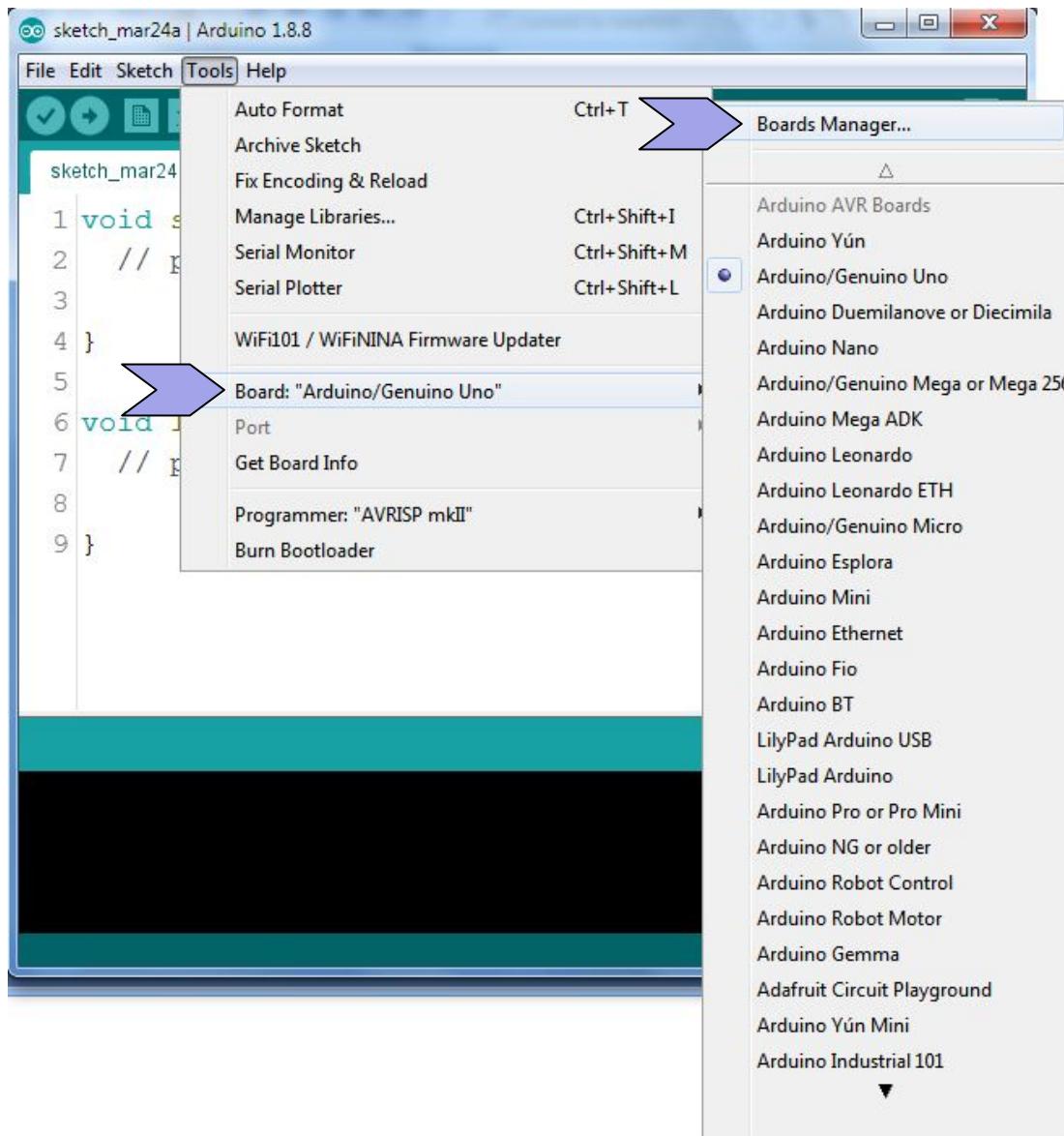


# Setting up Arduino IDE for NodeMCU Board

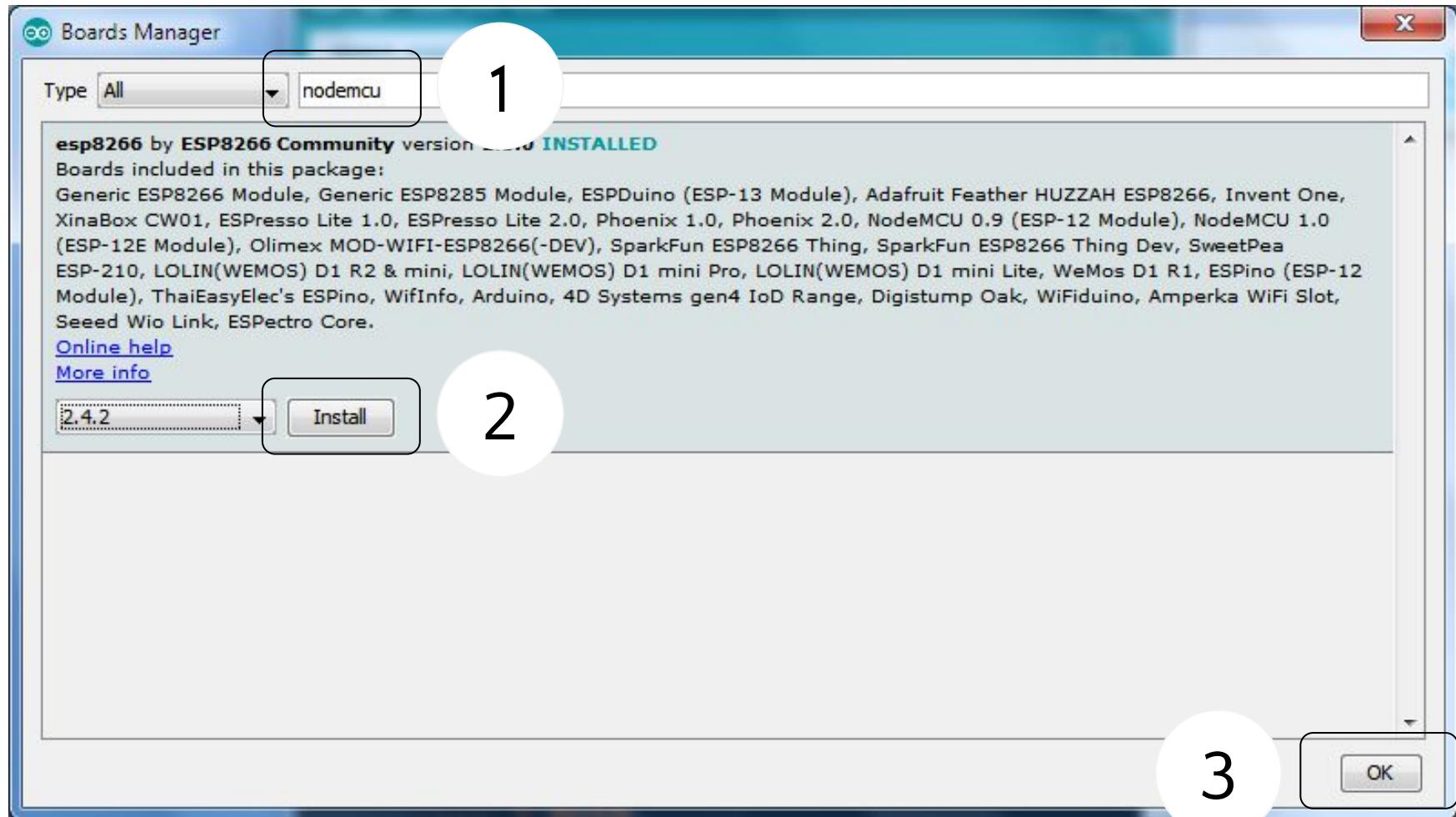


Google Search keyword: “Arduino additional Board manager URL for NodeMCU”

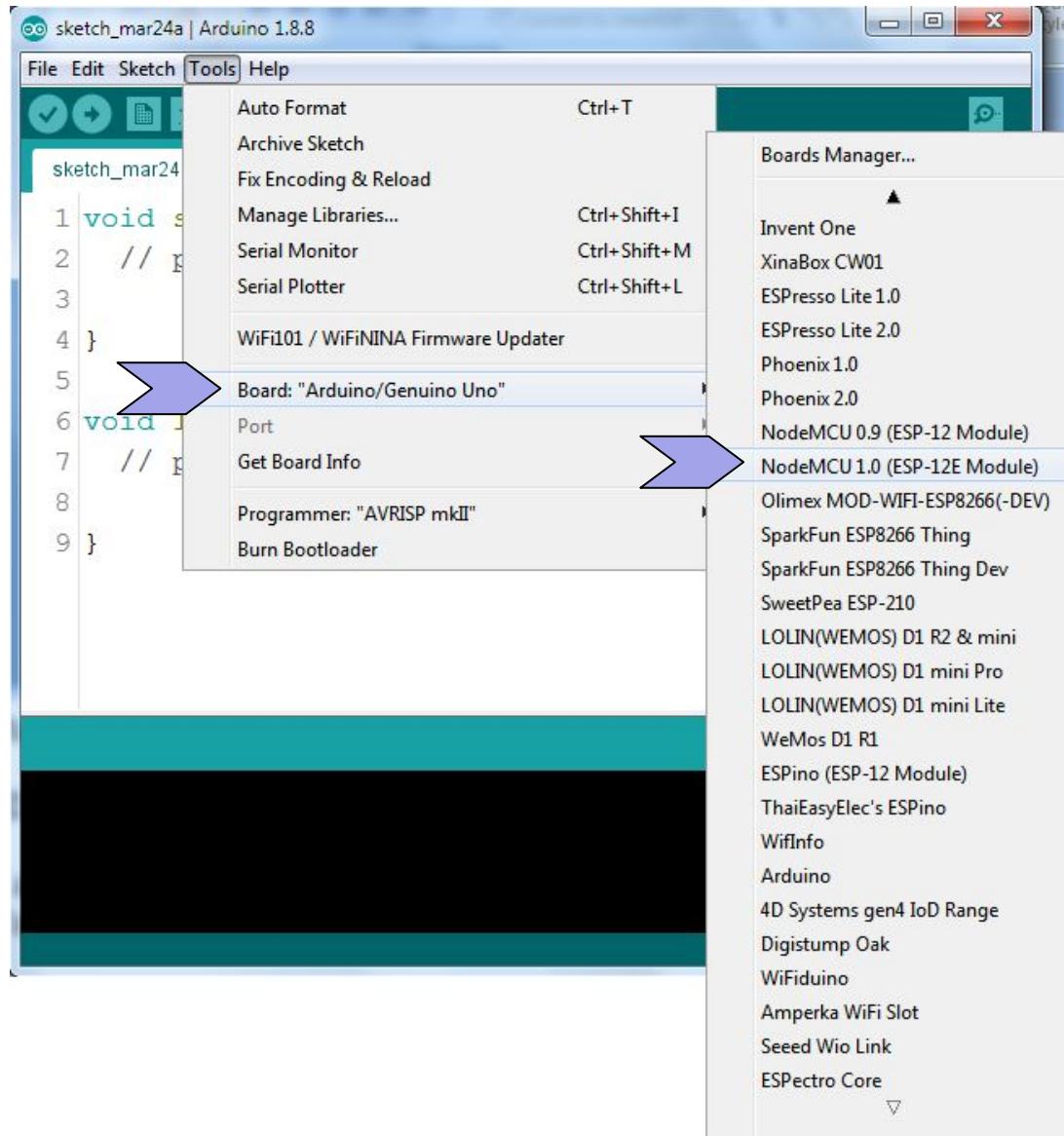
# Setting up Arduino IDE for NodeMCU Board



# Setting up Arduino IDE for NodeMCU Board



# Setting up Arduino IDE for NodeMCU Board



# Programming NodeMCU

## Commonly used Arduino Functions

Function	Purpose	Example Syntax
<code>pinMode()</code>	Used to configure pins as input/output	<code>pinMode(D0, INPUT);</code>
<code>digitalRead()</code>	Used to read digital signals from a pin	<code>digitalRead(D0);</code>
<code>digitalWrite()</code>	Used to write digital signals on to a pin: HIGH for 5V & LOW for 0V	<code>digitalWrite(D0, HIGH);</code>
<code>analogRead()</code>	Used to read analog signals from a pin using internal A to D Conversion	<code>analogRead(A0);</code>
<code>analogWrite()</code>	Used to generate width varying digital pulses using Pulse Width Modulation (PWM) which can be used to produce low frequency Analog signals with the help of additional filters.	<code>analogWrite(D6, 400);</code>

# Programming NodeMCU

## Commonly used Arduino Functions ...

Function	Purpose	Example Syntax
<code>Serial.print()</code>	Used to transmit data through serial interface for debugging / monitoring / communication purposes	<code>Serial.print("Hello World");</code>
<code>Serial.begin()</code>	Sets the data rate in bits per second (bps or baud) for serial data transmission.	<code>Serial.begin(115200);</code>
<code>delay()</code>	To delay the next command for a certain duration in milliseconds.	<code>delay(1000);</code>

Let's  
Make It Yourself  
Hands-On!!!

# Let's connect them together

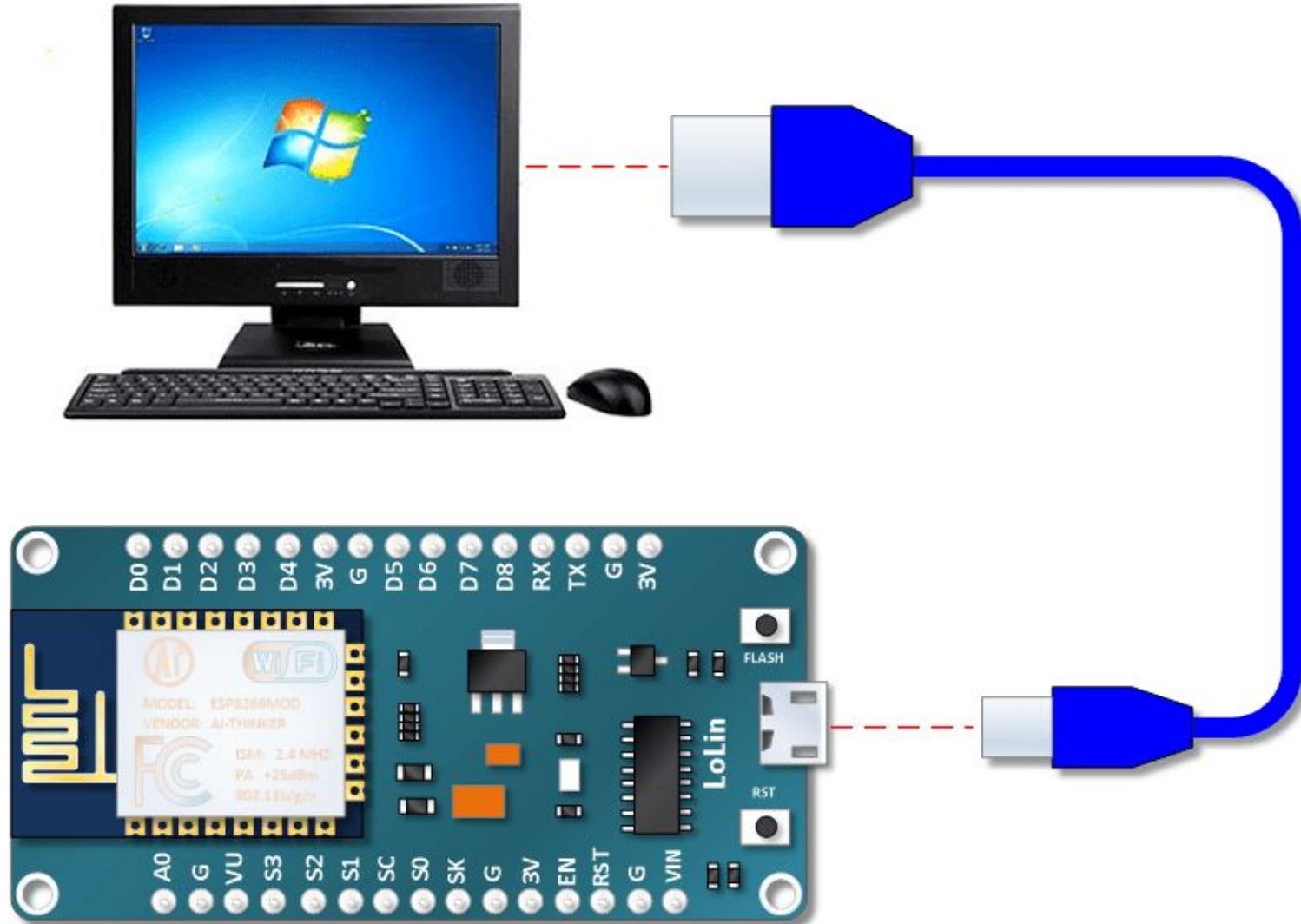


Image Source: <https://i0.wp.com/henrysbench.capnfatz.com/wp-content/uploads/2016/09/Connect-to-Computer.png>

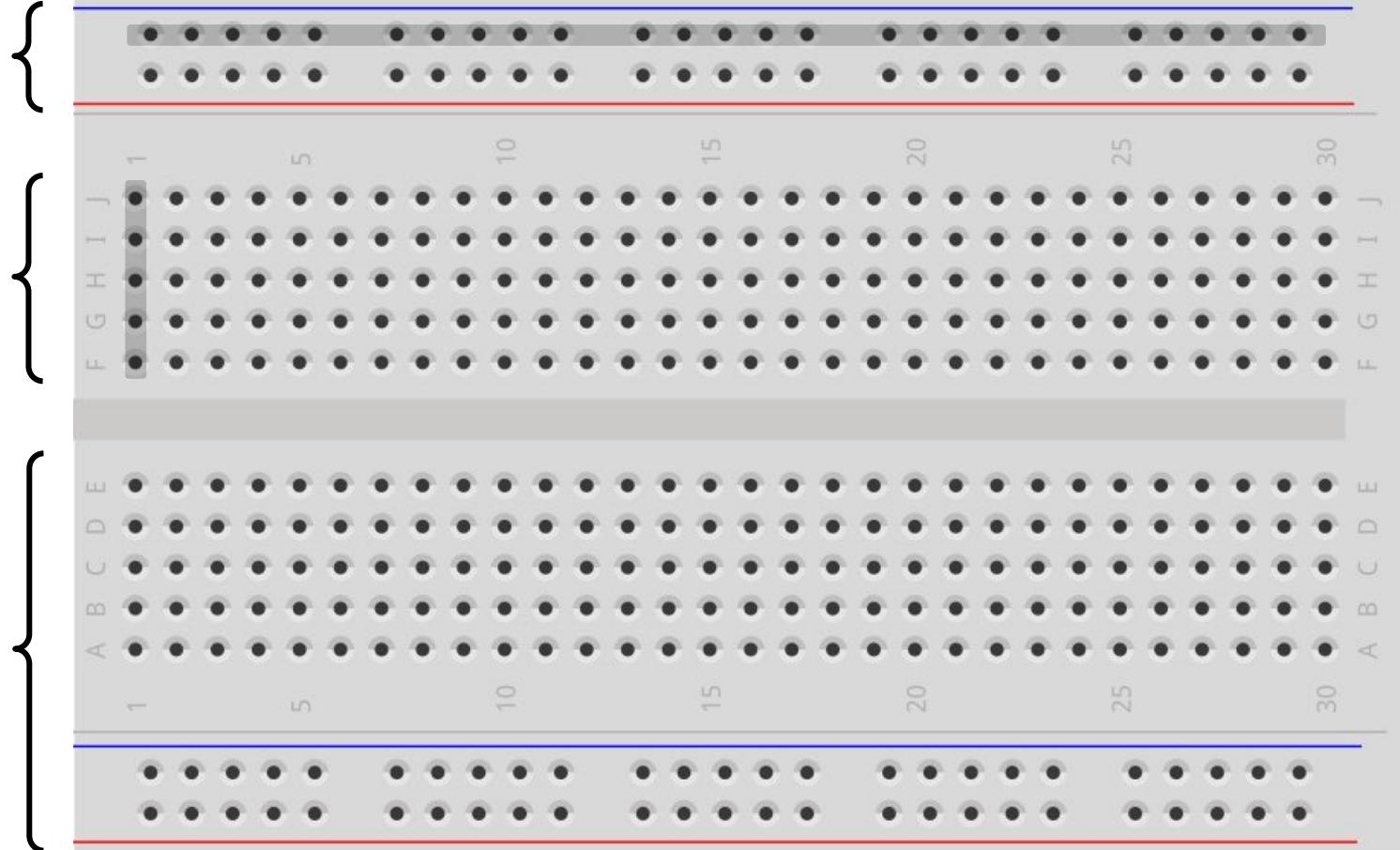
# Breadboard Connection

The horizontal strips that run the length of the breadboard are electrically connected inside. The strips are usually used for power and ground connections

Power Bus

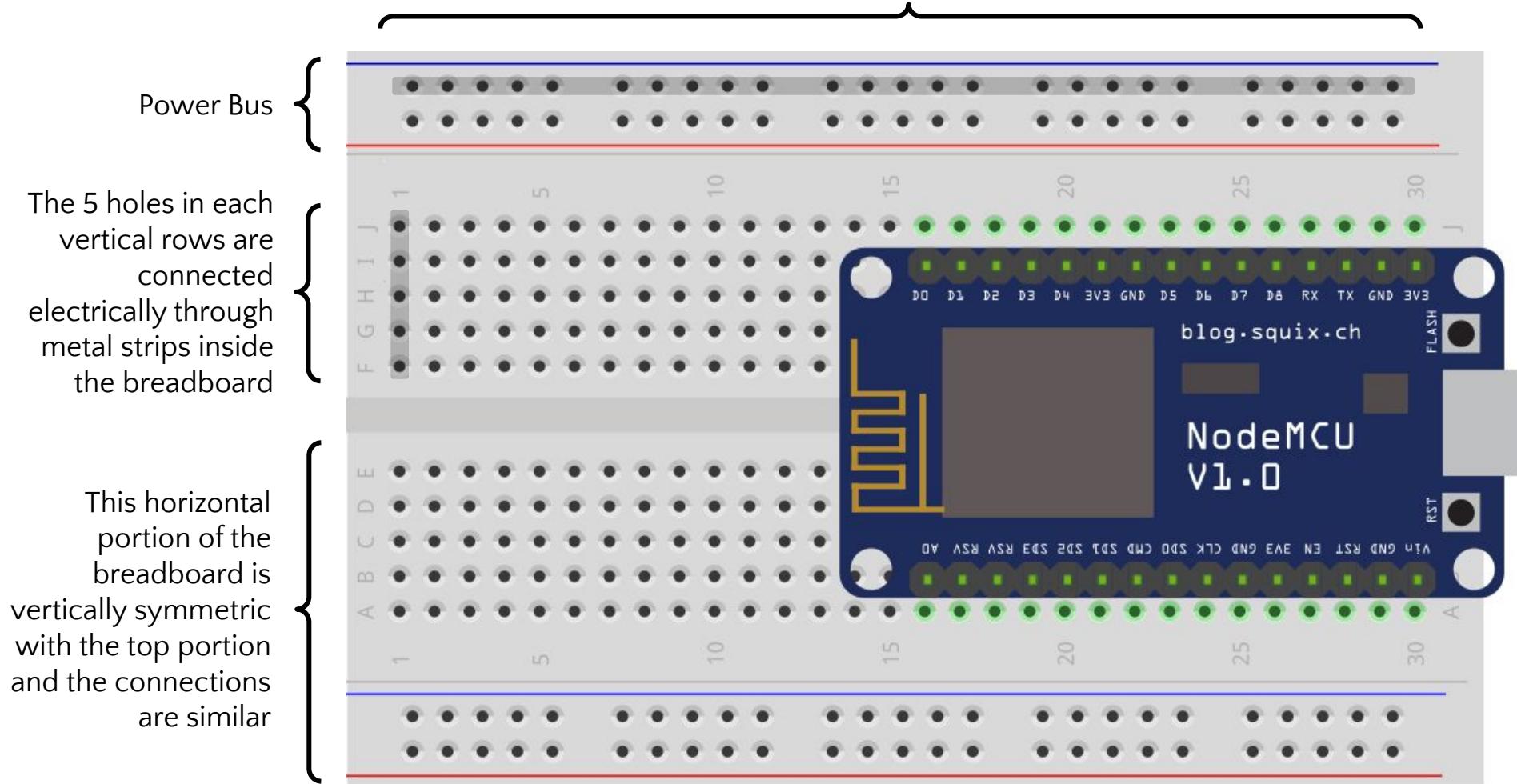
The 5 holes in each vertical rows are connected electrically through metal strips inside the breadboard

This horizontal portion of the breadboard is vertically symmetric with the top portion and the connections are similar



# Breadboard Connection

The horizontal strips that run the length of the breadboard are electrically connected inside. The strips are usually used for power and ground connections



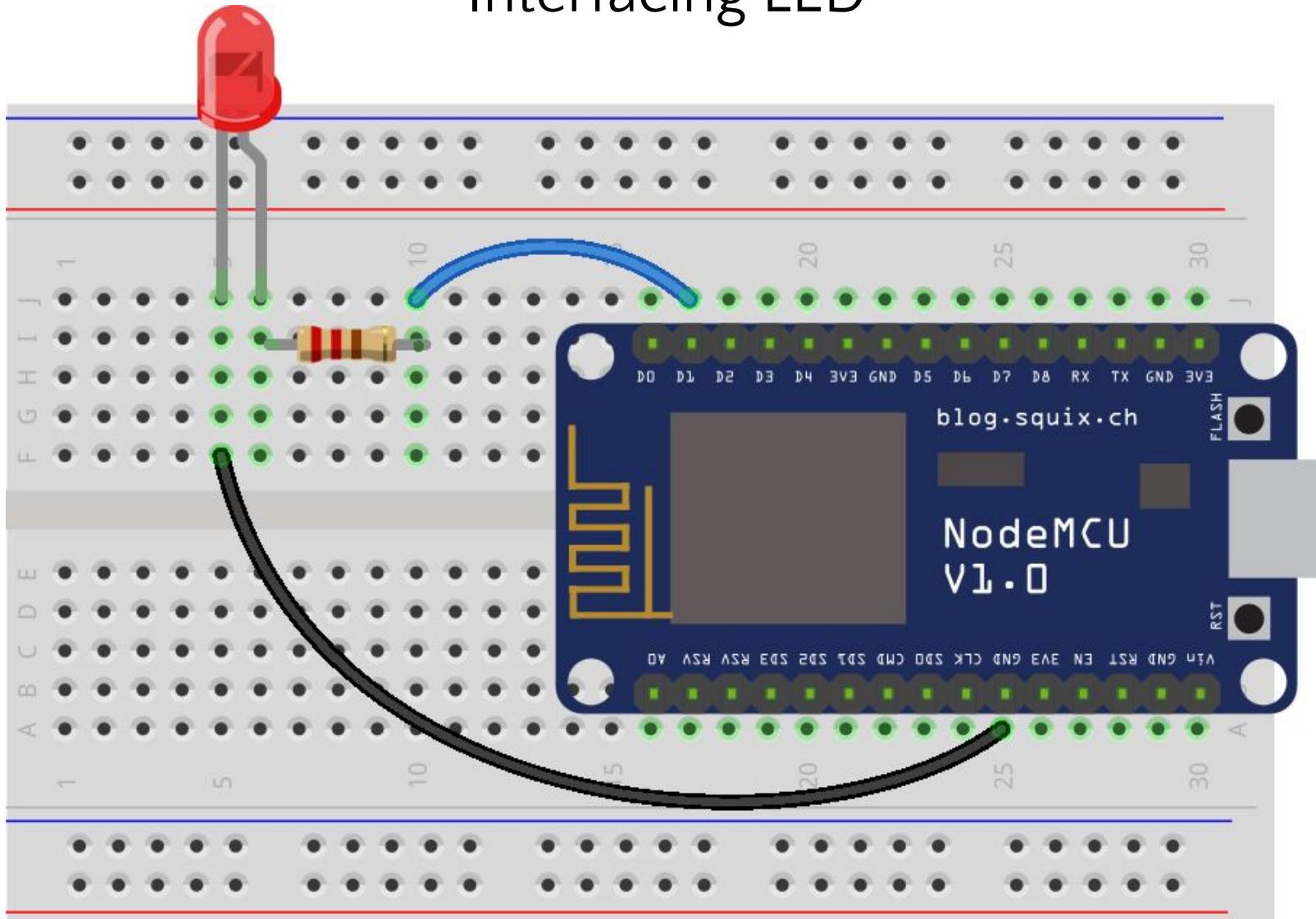
Reference: <https://diygeeks.org/wp-content/uploads/2018/01/BreadBoard-Labelling-1012x1024.jpg>

Layout designed using Fritzing® - <http://fritzing.org>, NodeMCU part Source(s):<https://github.com/squix78/esp8266-fritzing-parts/tree/master/nodemcu-v1.0>

fritzing

# DIY Experiments with NodeMCU

## Interfacing LED



Layout designed using Fritzing® - <http://fritzing.org>  
NodeMCU part Source(s):<https://github.com/squix78/esp8266-fritzing-parts/tree/master/nodemcu-v1.0>

fritzing

# Momentary Tactile Push Button / Switch

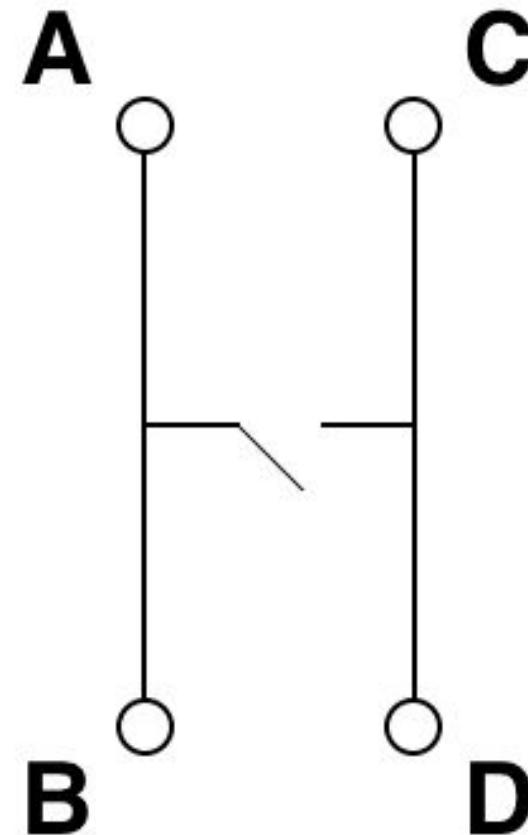
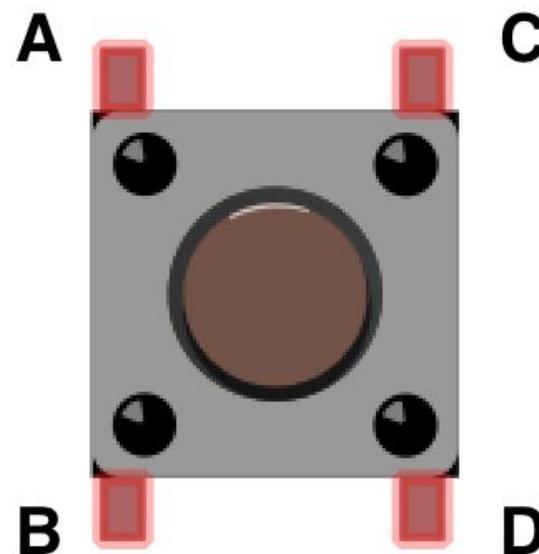
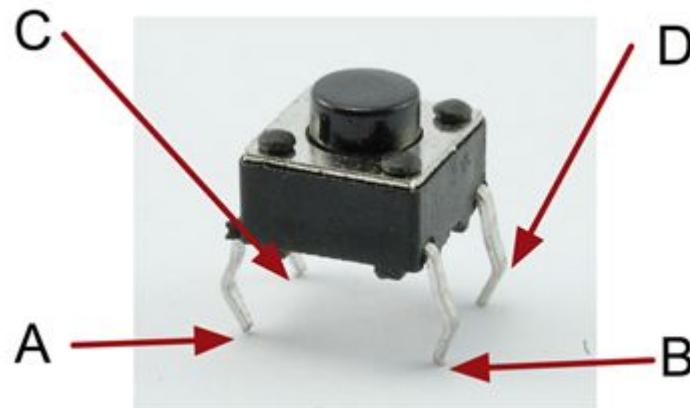


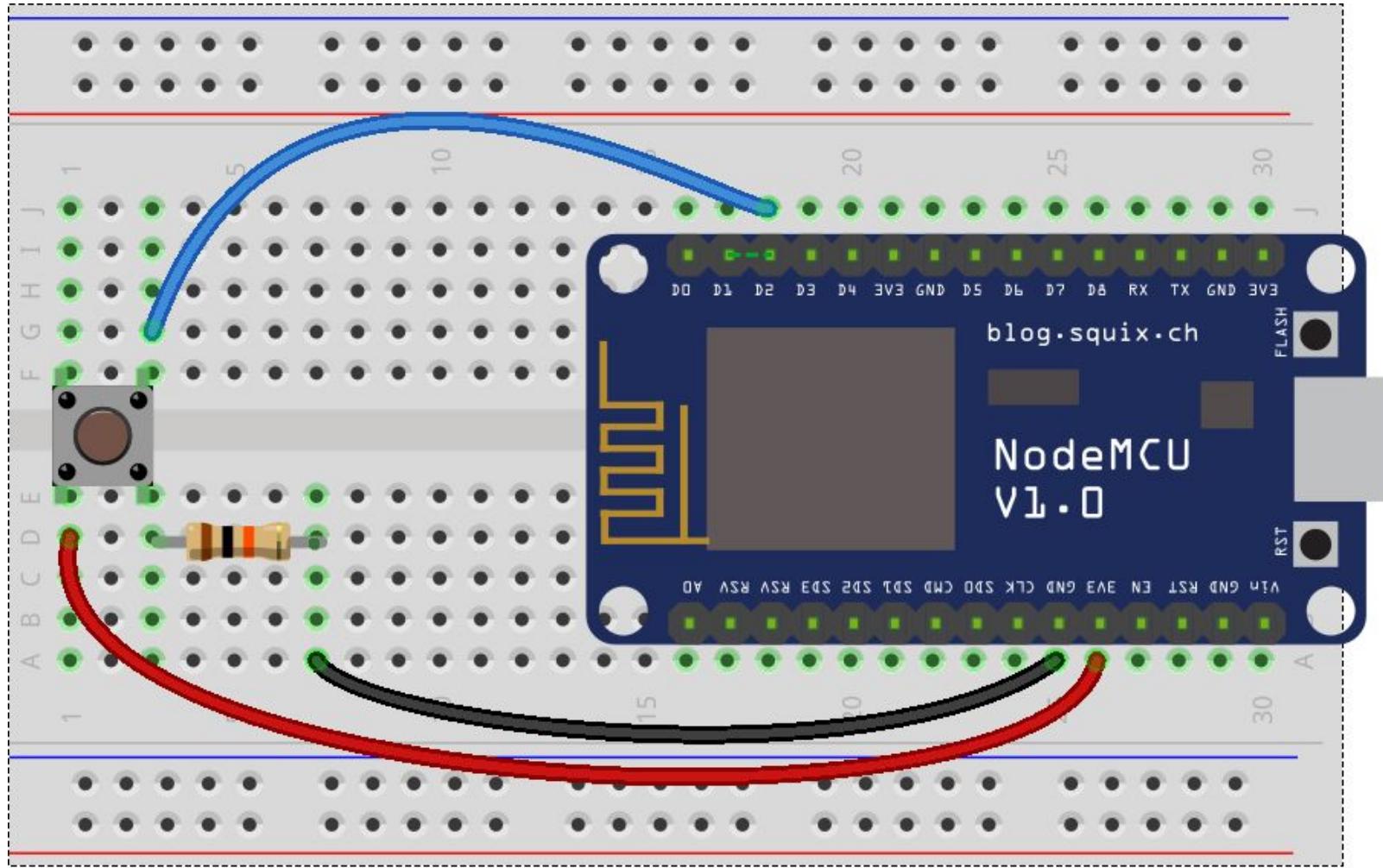
Image Source(s):

[http://razzpisampler.oreilly.com/images/rpck\\_1102.png](http://razzpisampler.oreilly.com/images/rpck_1102.png),

[https://docs.labs.mediatek.com/resource/linkit7697-arduino/files/en/12880064/12880062/1/1498095674923/button\\_sch.png](https://docs.labs.mediatek.com/resource/linkit7697-arduino/files/en/12880064/12880062/1/1498095674923/button_sch.png)

# DIY Experiments with NodeMCU

## Interfacing Push Button Switch



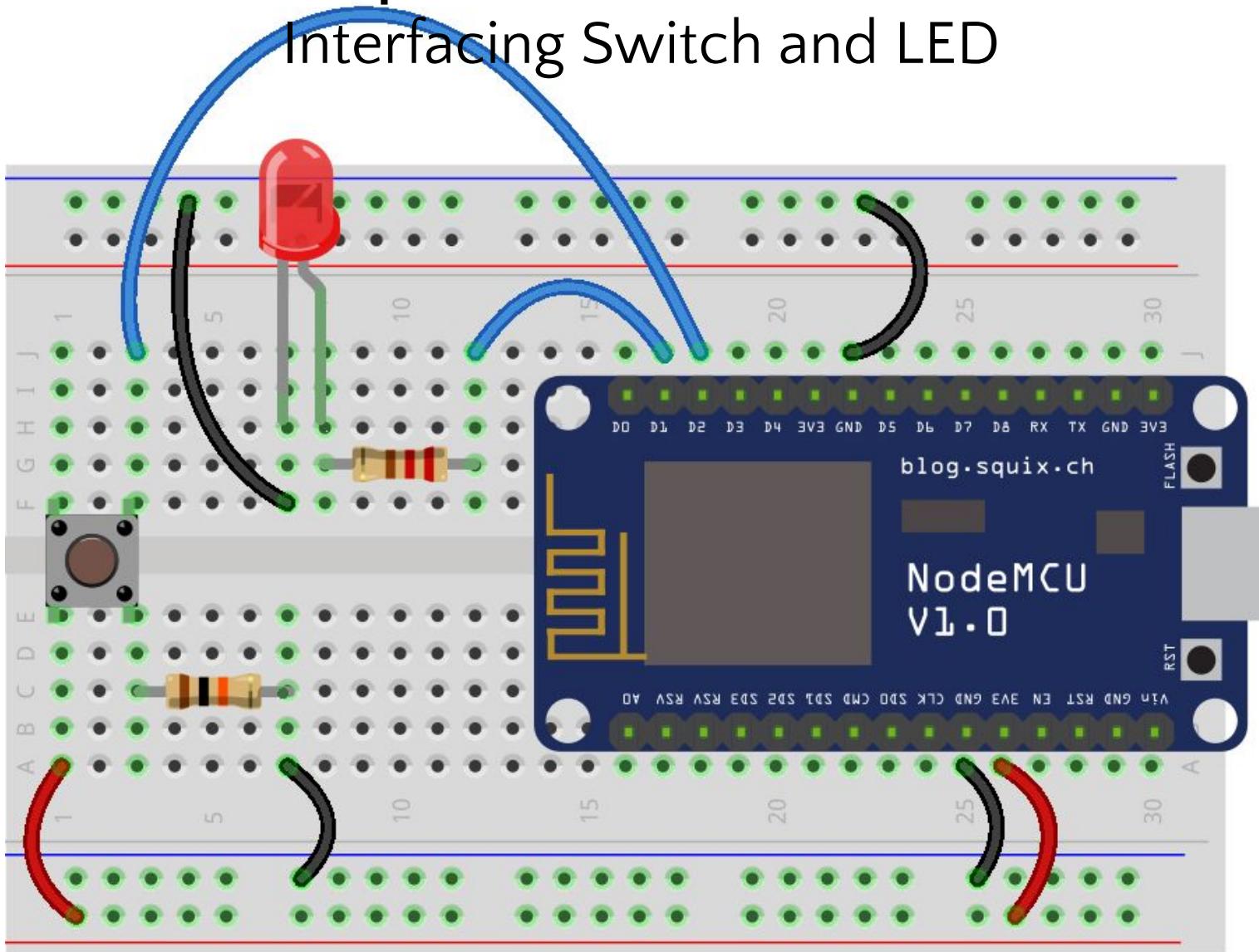
Layout designed using Fritzing® - <http://fritzing.org>

NodeMCU part Source: <https://github.com/squix78/esp8266-fritzing-parts/tree/master/nodemcu-v1.0>

fritzing

# DIY Experiments with NodeMCU

## Interfacing Switch and LED



Layout designed using Fritzing® - <http://fritzing.org>

NodeMCU part Source: <https://github.com/squix78/esp8266-fritzing-parts/tree/master/nodemcu-v1.0>

fritzing

# Light Dependent Resistor (LDR)

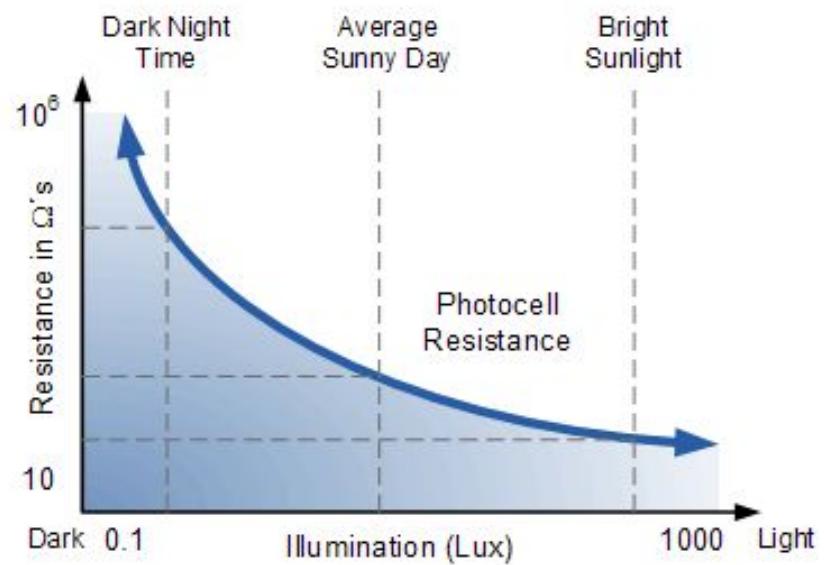
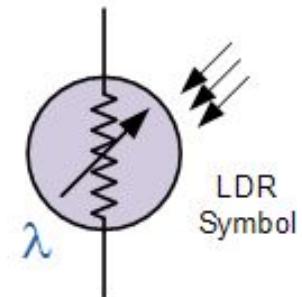


Image Source(s):

<http://www.electricalibrary.com/wp-content/uploads/2017/08/LDR-destaque.jpg>

<https://www.electronics-tutorials.ws/io/io19.gif>

# LDR application circuit diagram

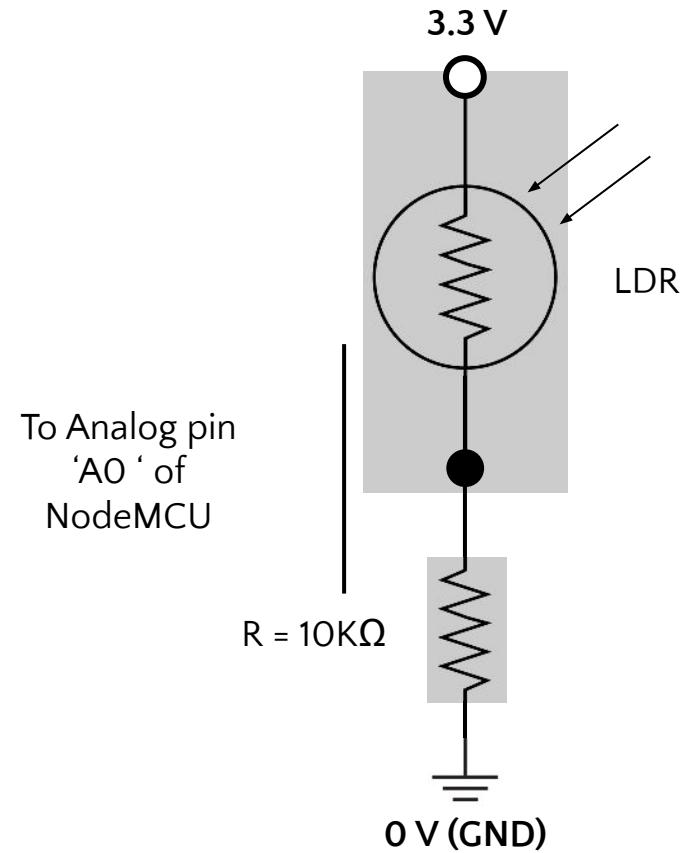
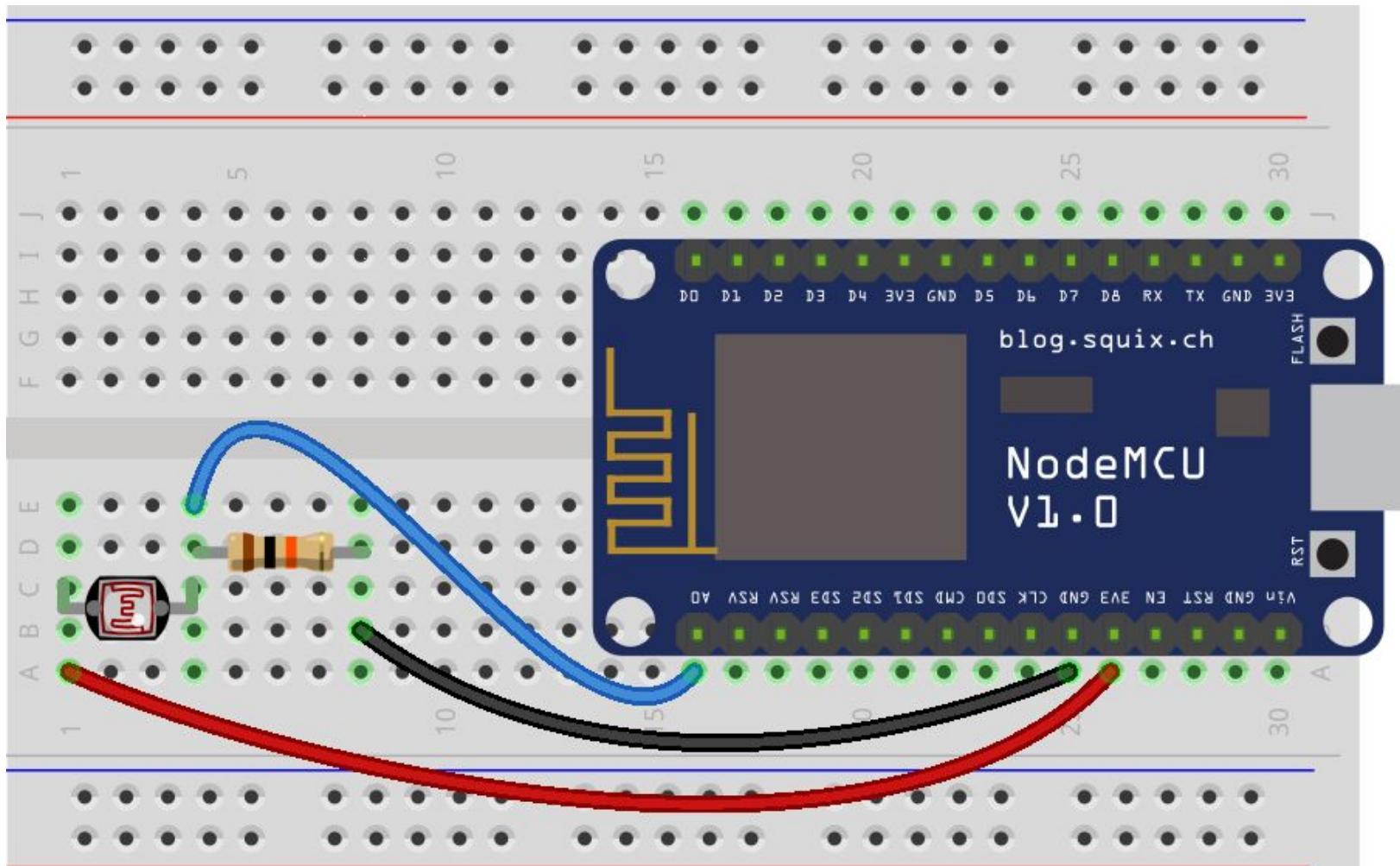


Image Source: <https://www.electronics-notes.com/images/light-dependent-resistor-photoresistor-circuit-symbol.svg>

# DIY Experiments with NodeMCU

## Interfacing LDR

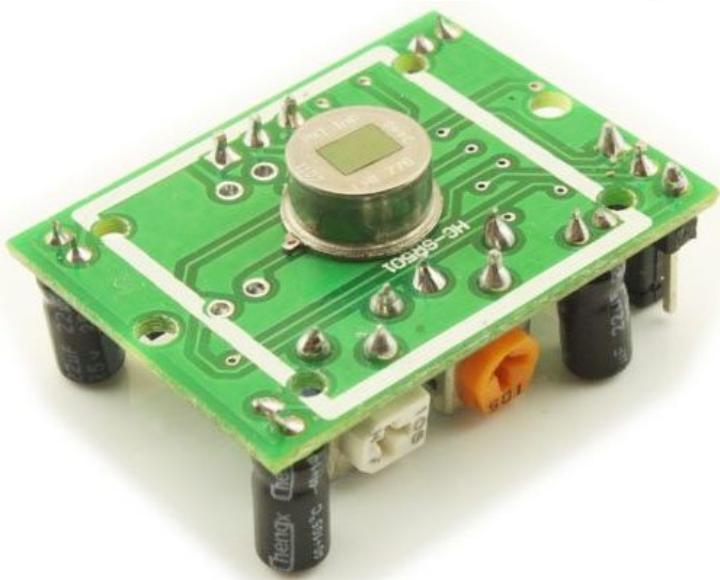


Layout designed using Fritzing® - <http://fritzing.org>

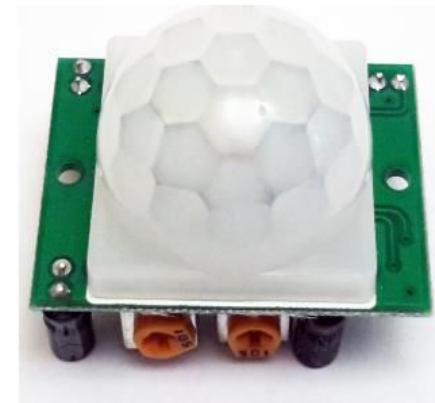
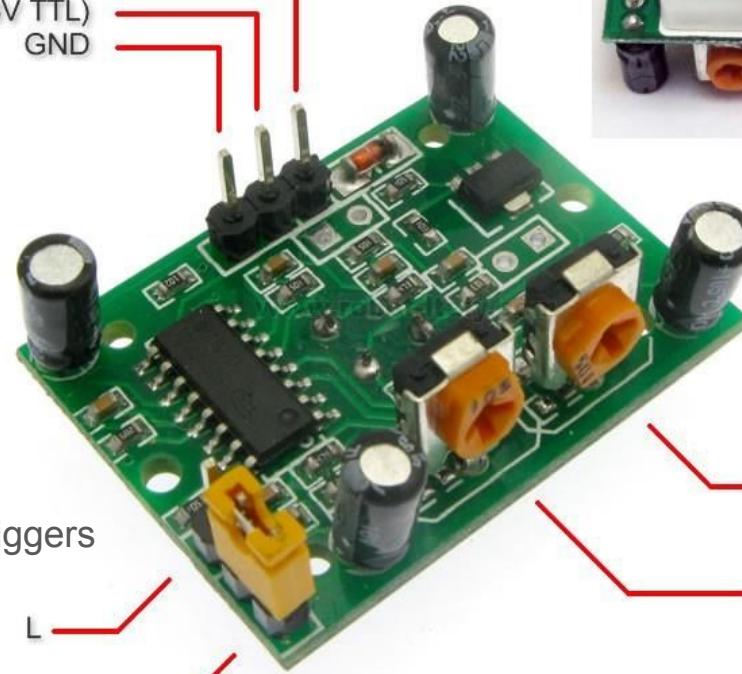
NodeMCU part Source: <https://github.com/squix78/esp8266-fritzing-parts/tree/master/nodemcu-v1.0>

fritzing

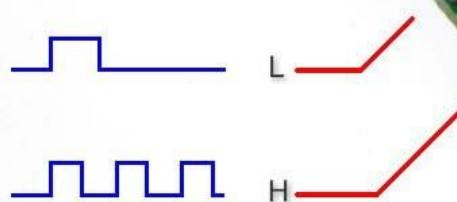
# PIR Motion Sensor – HCR501



VCC 5-12VDC  
OUT (3.3V TTL)  
GND



Setting for Single or Repeatable Triggers



# PIR Motion Sensing

## Fresnel Lens Focusing

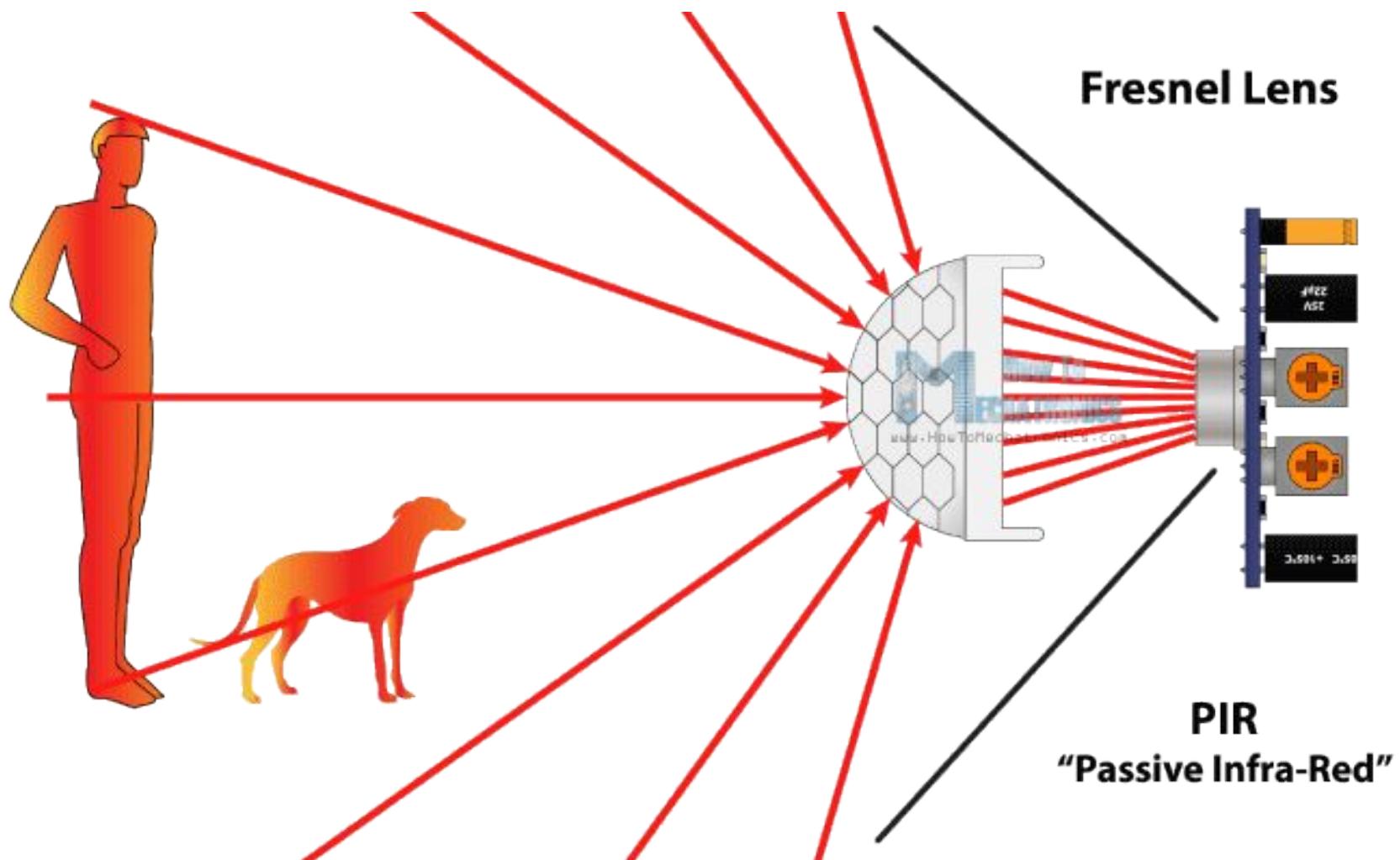


Image Source:: <https://howtomechatronics.com/wp-content/uploads/2015/09/PIR-Motion-Sensor-How-It-Works.png>

# PIR Motion Sensing

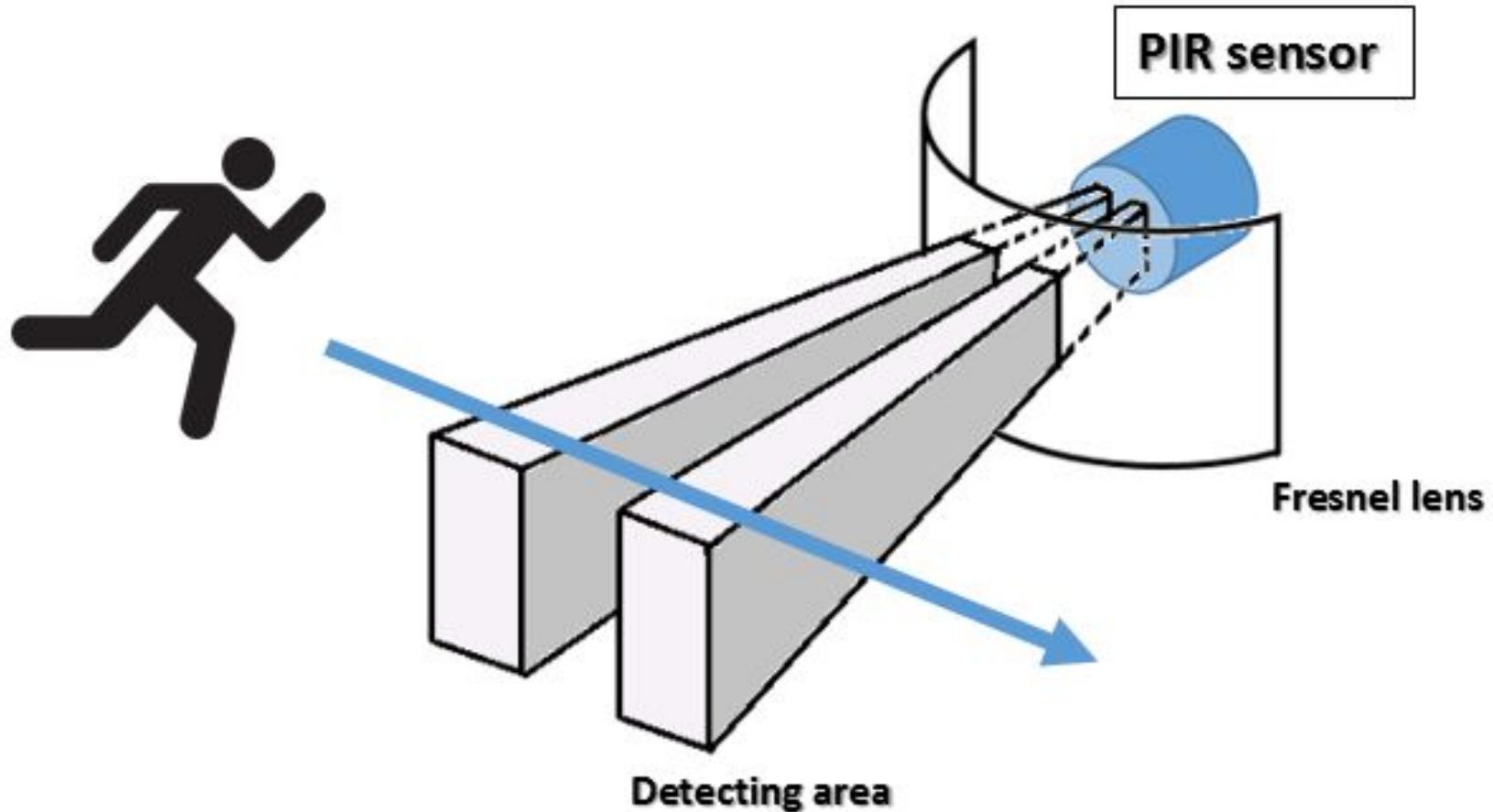


Image Source::

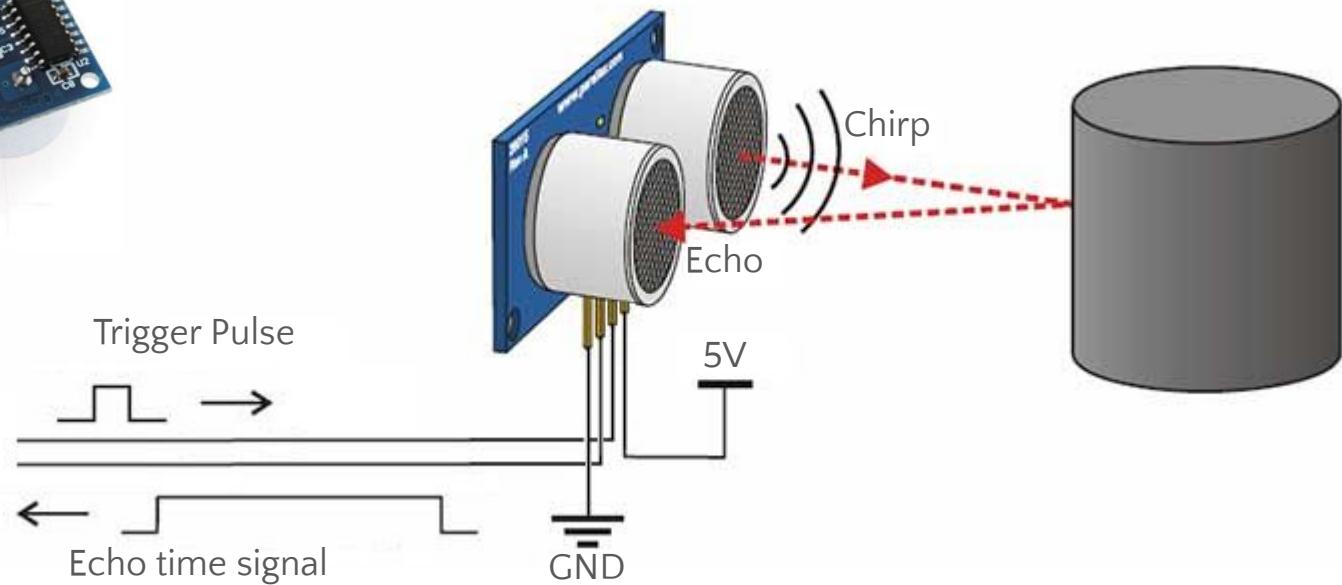
<https://static1.squarespace.com/static/557f31e6e4b0971f7f8bcac6/t/5b04a73e352f53cc9e14f413/1527031679239/Meccanismo-Complesso-PIR-sensor-how-to.png>

# Ultrasonic Sensor – HC-SR04



HC-SR04

## Working of HC-SR04



# Ultrasonic Sensor HC-SR04

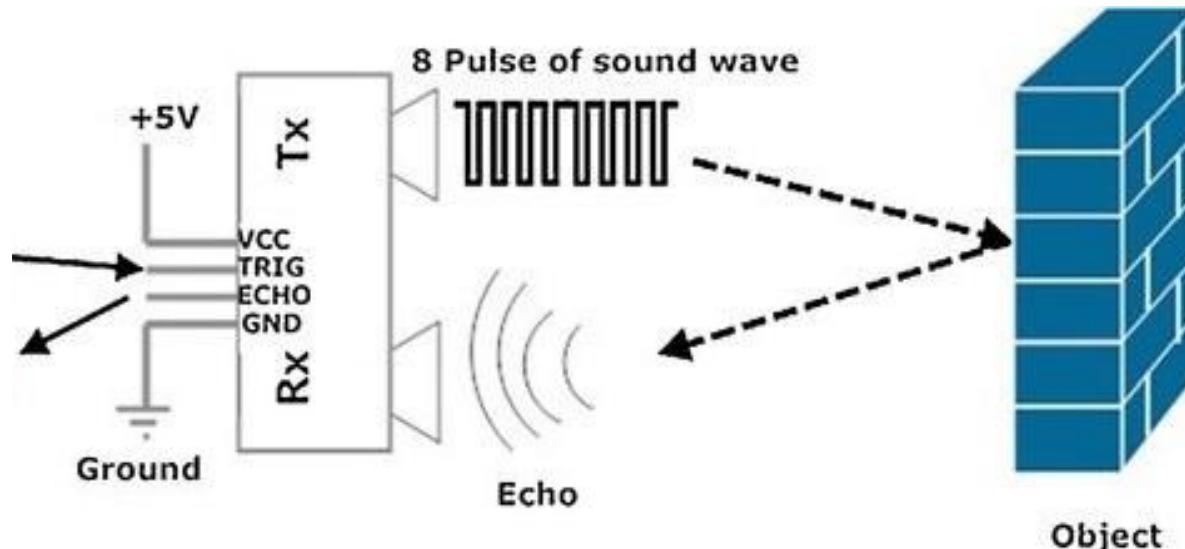
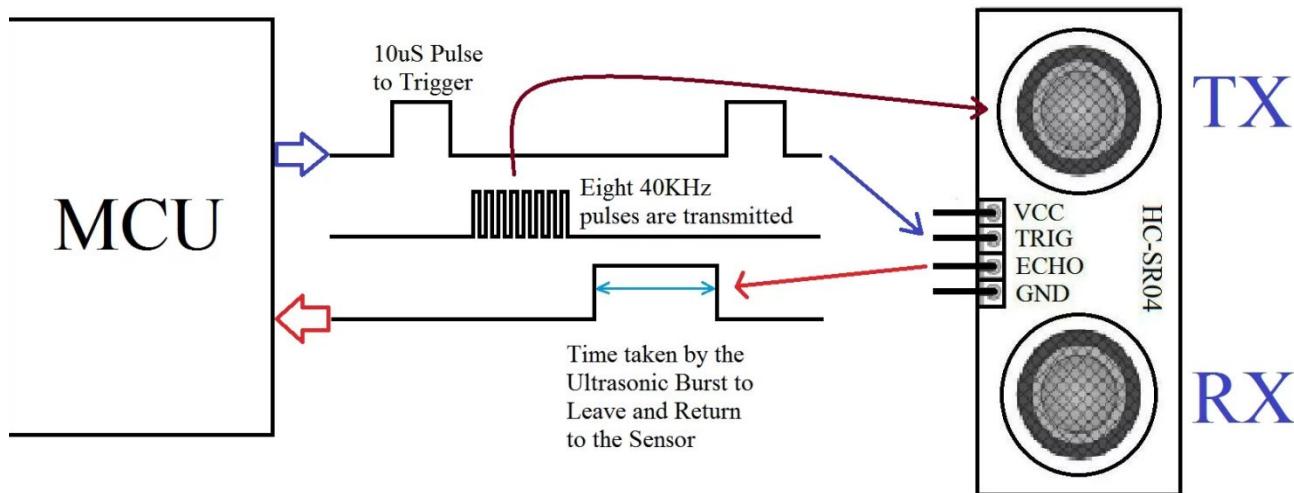


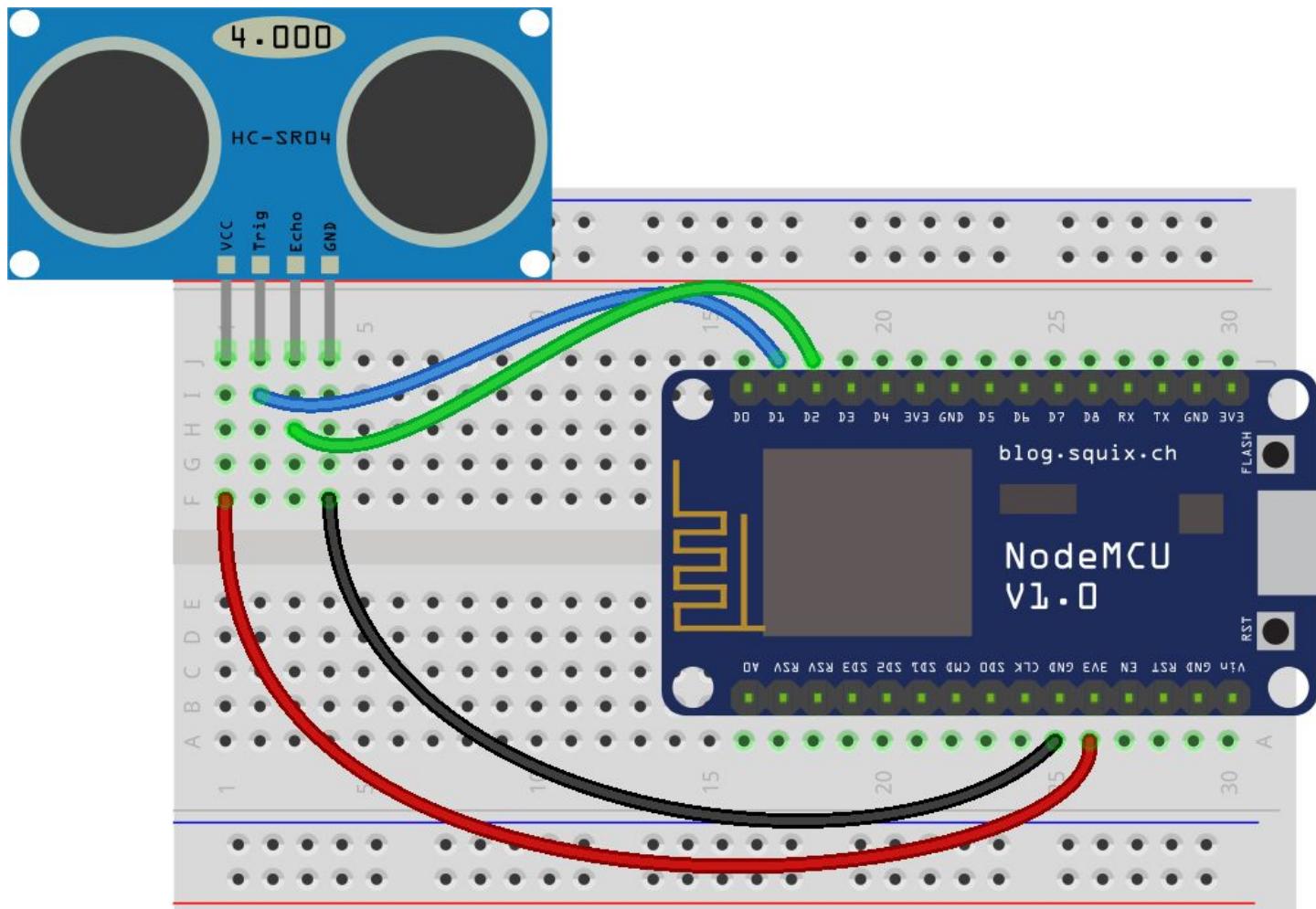
Image Source:

<https://electrosome.com/wp-content/uploads/2014/08/Working-of-HC-SR04-Ultrasonic-Sensor.jpg>

[https://www.researchgate.net/figure/Working-principle-of-an-ultrasonic-sensor\\_fig1\\_304822025](https://www.researchgate.net/figure/Working-principle-of-an-ultrasonic-sensor_fig1_304822025)

# DIY Experiments with NodeMCU

## Interfacing Ultrasonic Sensor

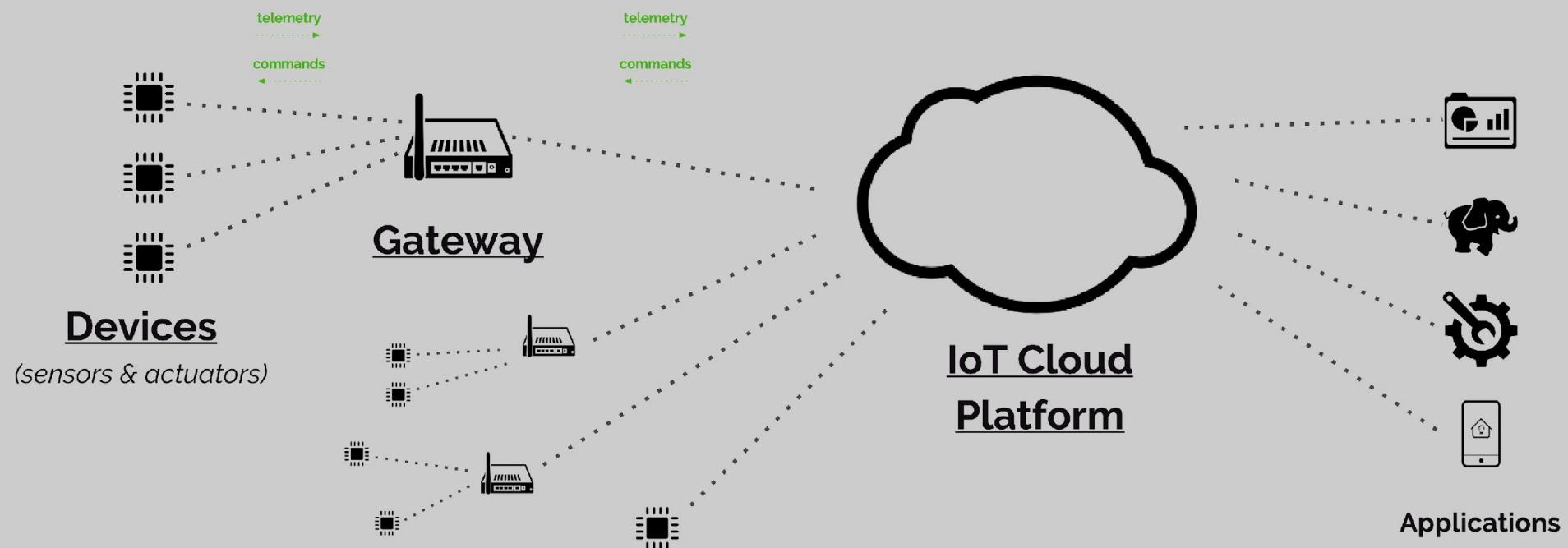


Layout designed using Fritzing® - <http://fritzing.org>

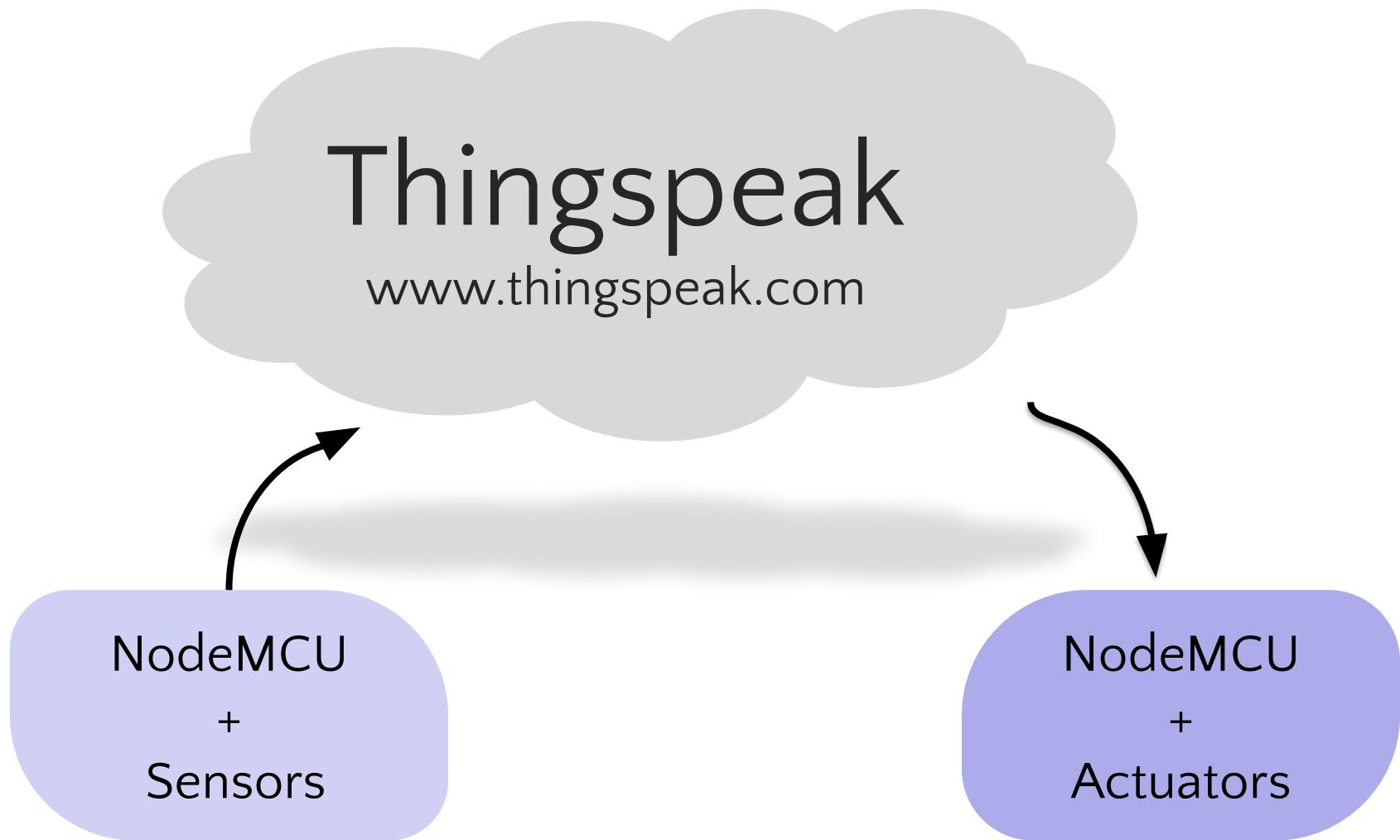
NodeMCU part Source: <https://github.com/squix78/esp8266-fritzing-parts/tree/master/nodemcu-v1.0>

fritzing

# Cloud Platforms



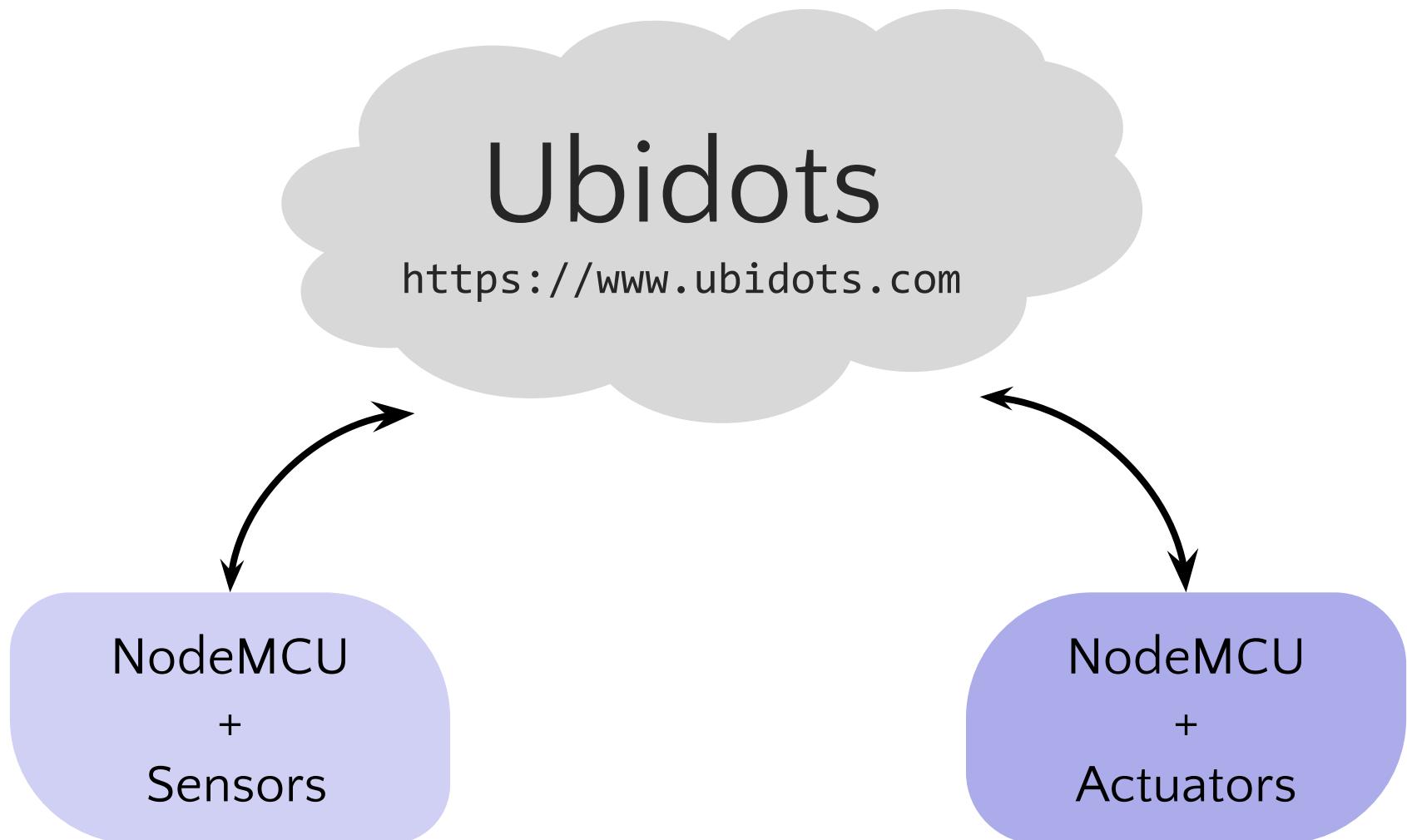
# Cloud Platforms



# IoT Projects

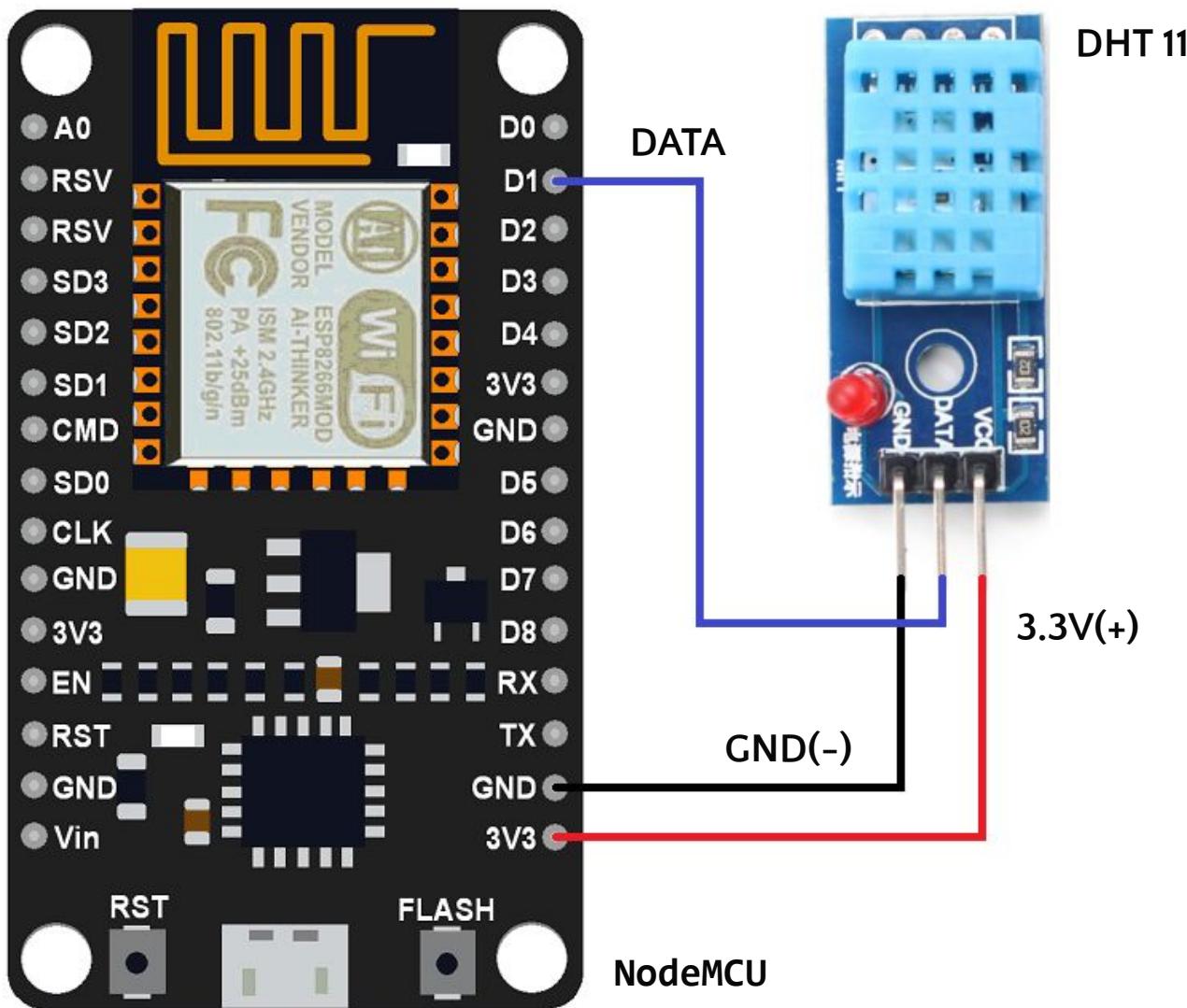
Application of IoT in Smart Cities  
Use Case: Urban Waste Management

# Cloud Platforms



# DIY Experiments with NodeMCU

## Interfacing Temperature & Humidity Sensor



# IoT Projects

Application of IoT in Home Automation  
Use Case: Remote Switching of Air Conditioner