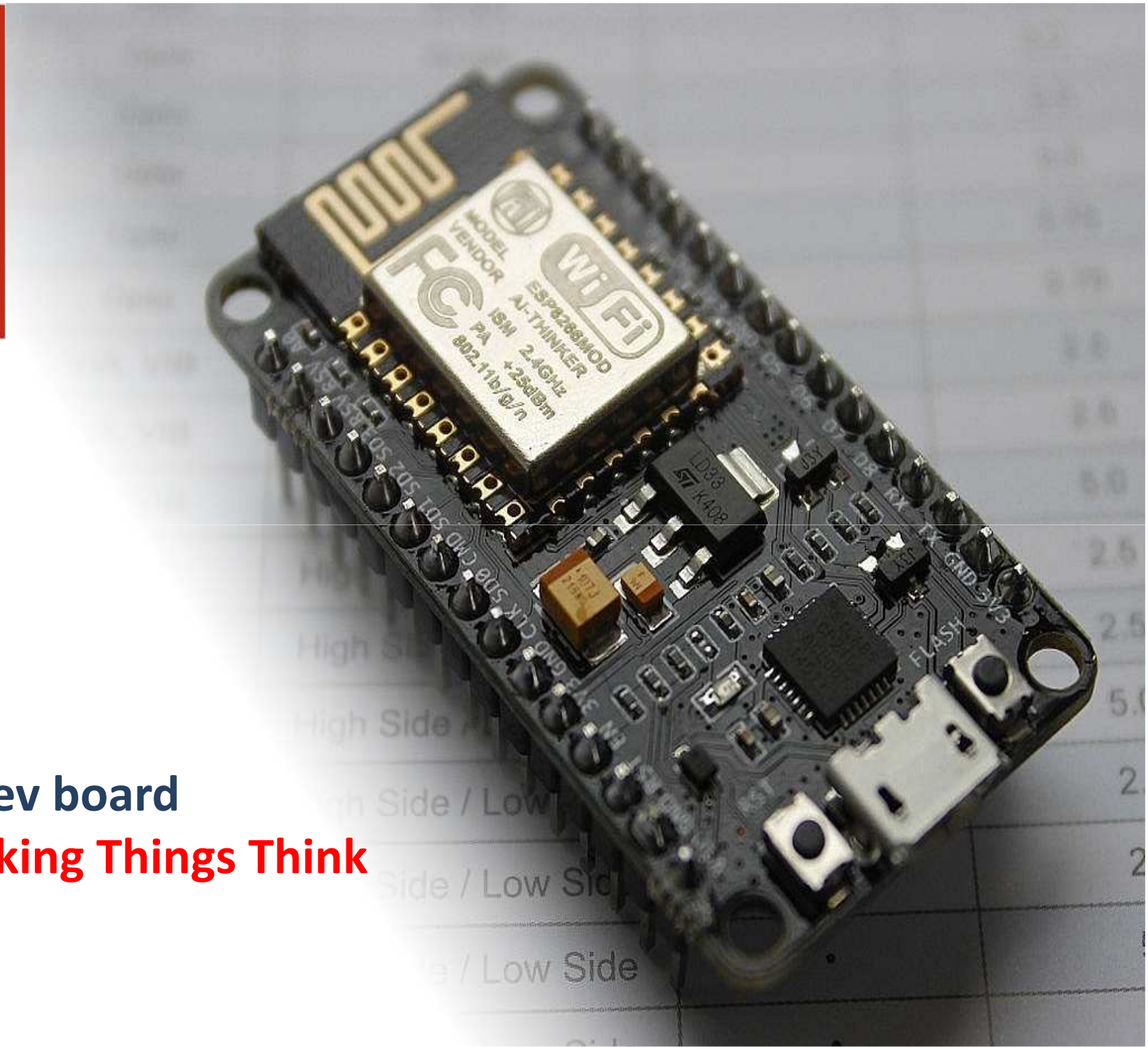


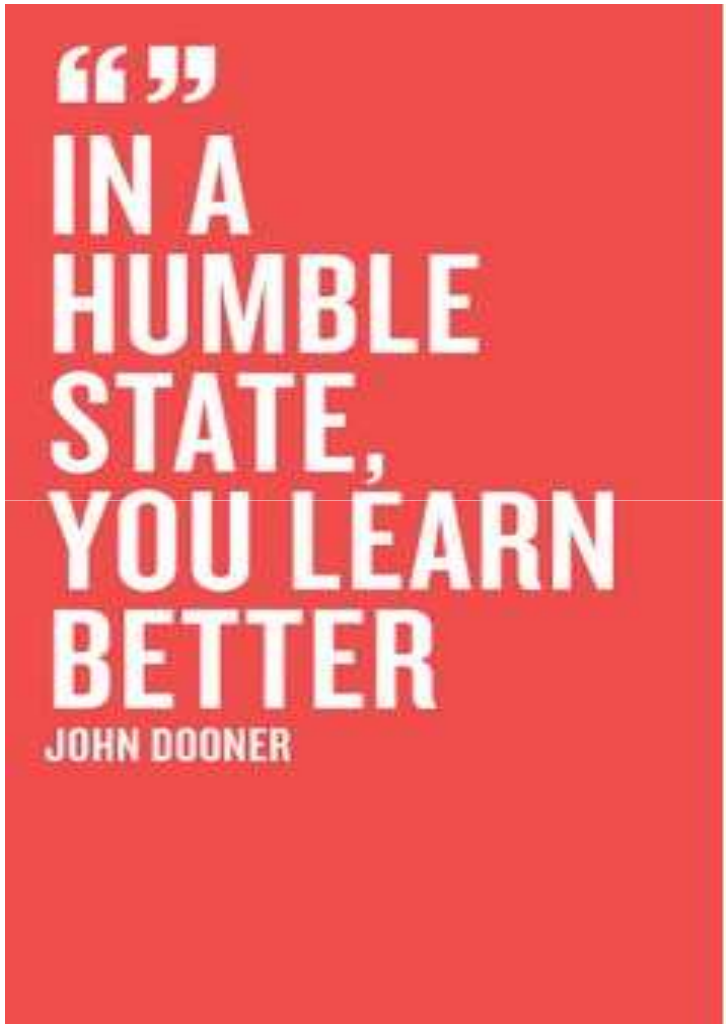
NodeMCU
ESP8266 dev board

-Making Things Think

Internet of Things

Internet of Things

IoT is a network of physical objects with embedded electronics that collect and share data”


“”
IN A HUMBLE STATE, YOU LEARN BETTER
JOHN DOONER

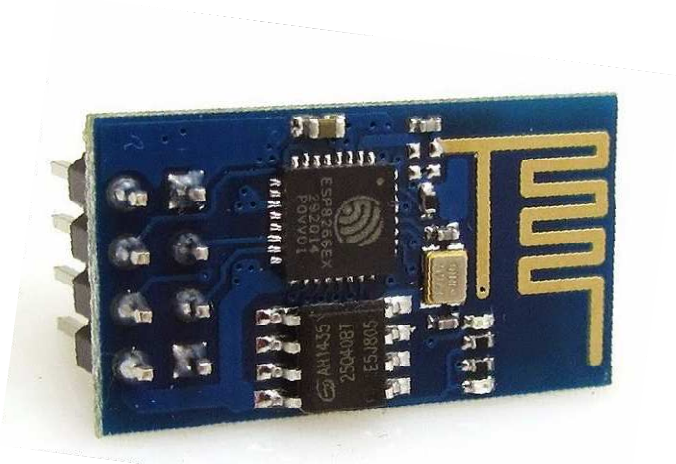

Always be
willing to learn
HAIKU DECK

Development Boards

Arduino Ethernet shield

Raspberry pi

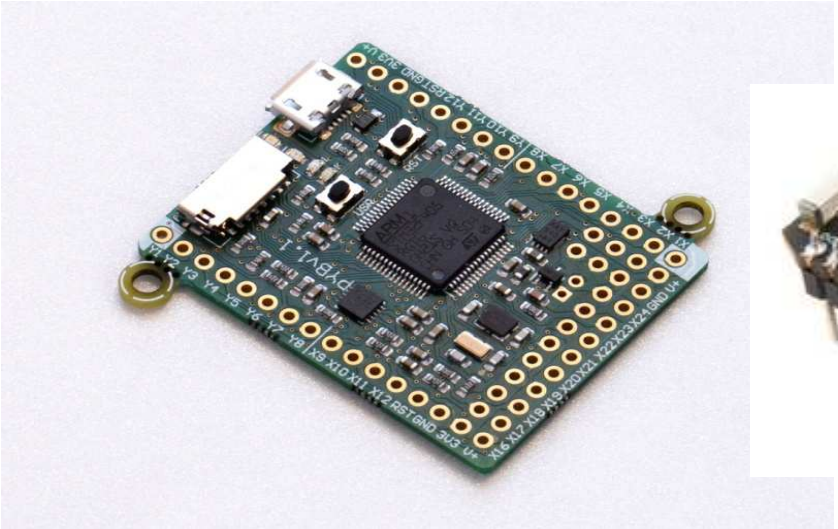
Intel Edison



ESP8266 WiFi Module

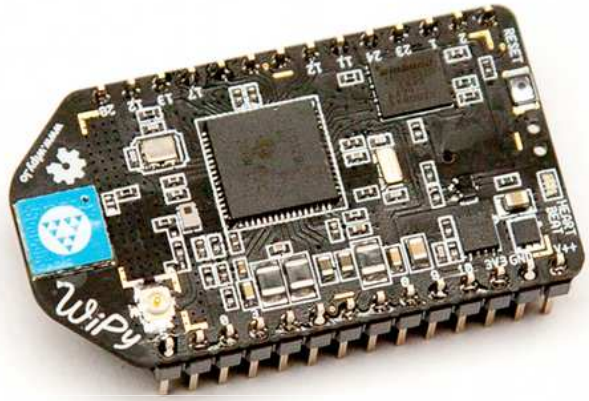
ESP8266 -12 node MCU

Development Boards

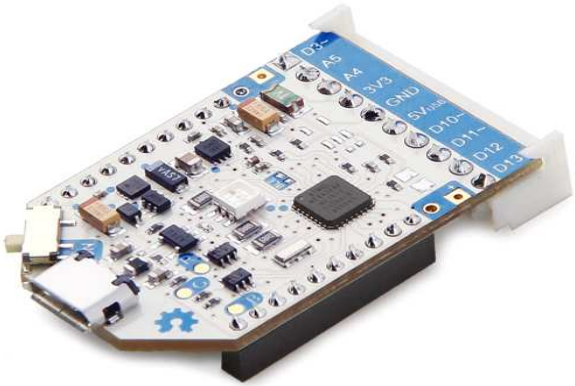


Micropython

Partical Photon

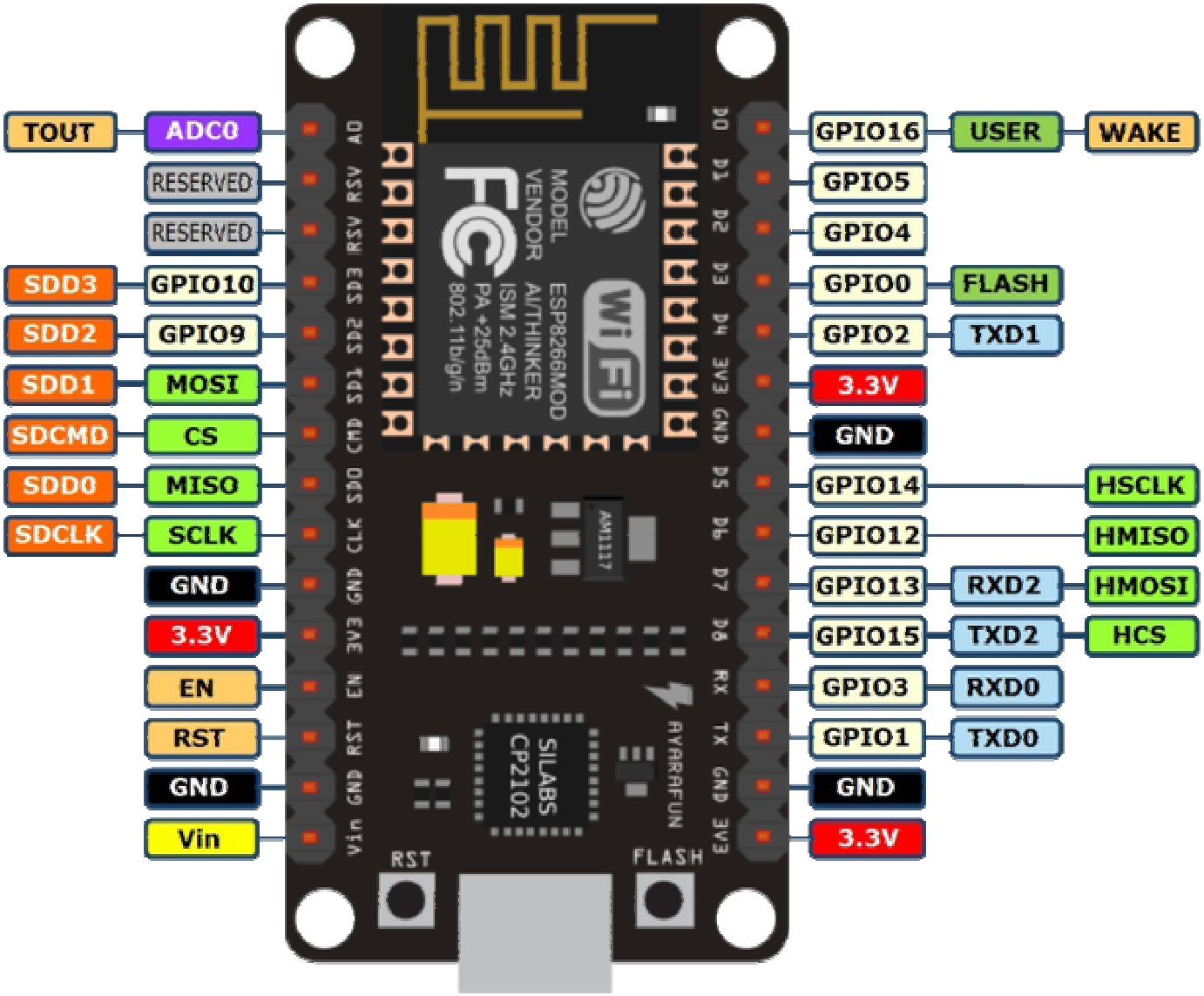


WiPy Board



Air Board

NB-IoT Board


TOUT
ADC0
RESERVED
RESERVED
SDD3
GPIO10
SDD2
GPIO9
SDD1
MOSI
SDCMD
CS
SDD0
MISO
SDCLK
SCLK
GND
3.3V
EN
RST
GND
Vin
GPIO16
USER
WAKE
GPIO5
GPIO4
GPIO0
FLASH
GPIO2
TXD1
3.3V
GND
GPIO14
HSCLK
GPIO12
HMISO
GPIO13
RXD2
HMOSI
GPIO15
TXD2
HCS
GPIO3
RXD0
GPIO1
TXD0
GND
3.3V
MODEL
VENDOR
ESP8266MOD
AI/THINKER
ISM 2.4GHz
PA +25dBm
802.11b/g/n
Wi Fi
AM1117
SILABS
CP2102
AYARAFUN
RST
FLASH

Arduino

The name is an Italian masculine name, *meaning* "strong friend"

I hear and I forget. I see and I remember.
I do and I understand.

-Confucius

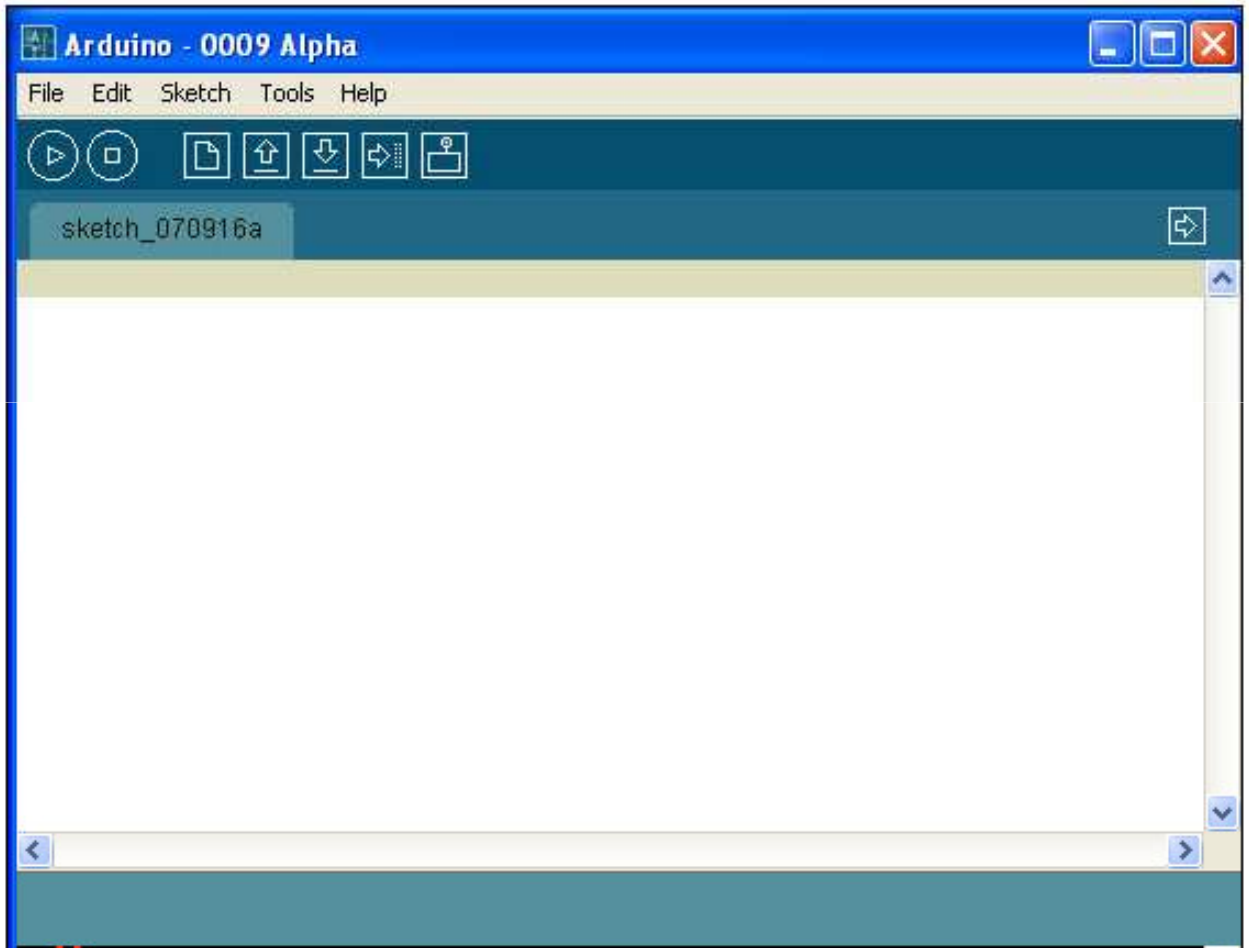
'Reform, Perform And Transform Is Our Mantra’

-Shri Narendra Modi

What do we need?

Nodemcu - Periphery

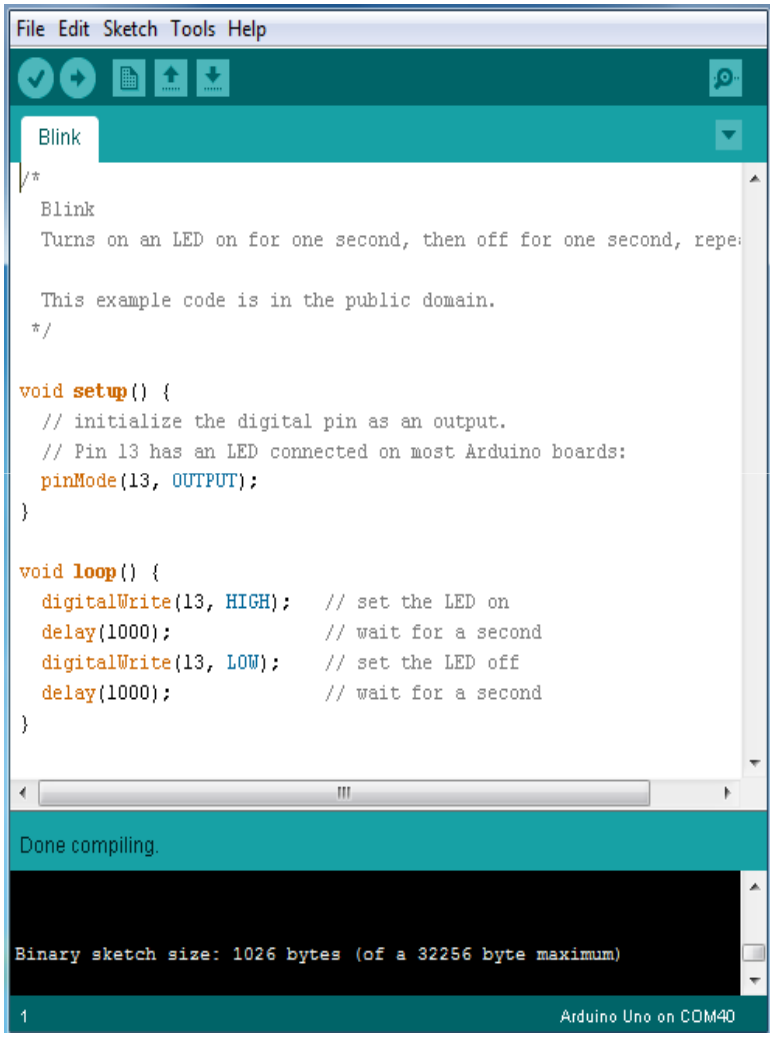
Download and install the Arduino Software



ARDUINO

Programming an Arduino

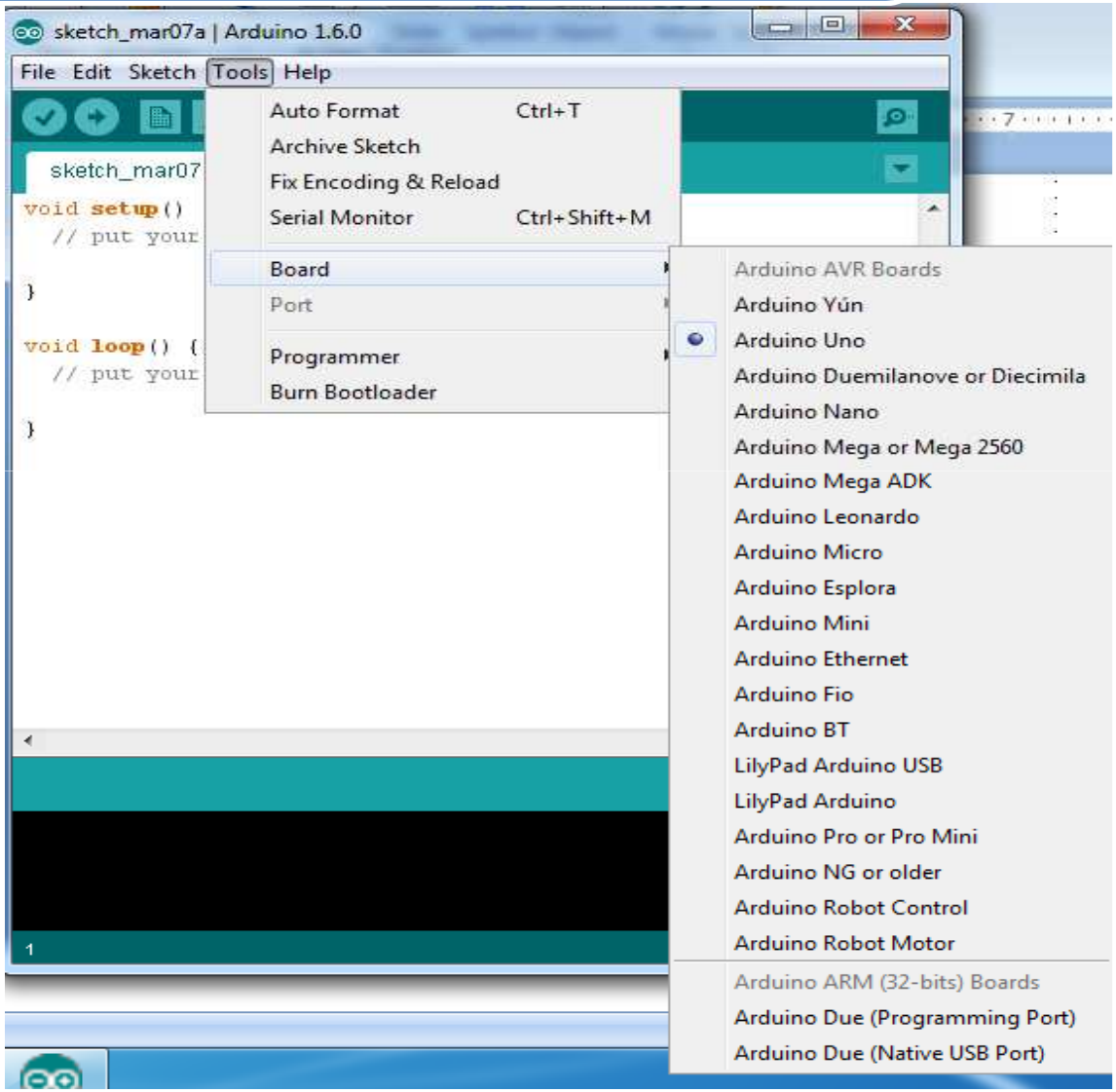
- The Arduino software consists of a development environment (IDE) and the core libraries.
- The IDE is written in Java and based on the processing environment.
- The core libraries are written in C and C++ and compiled using avr-gcc compiler.



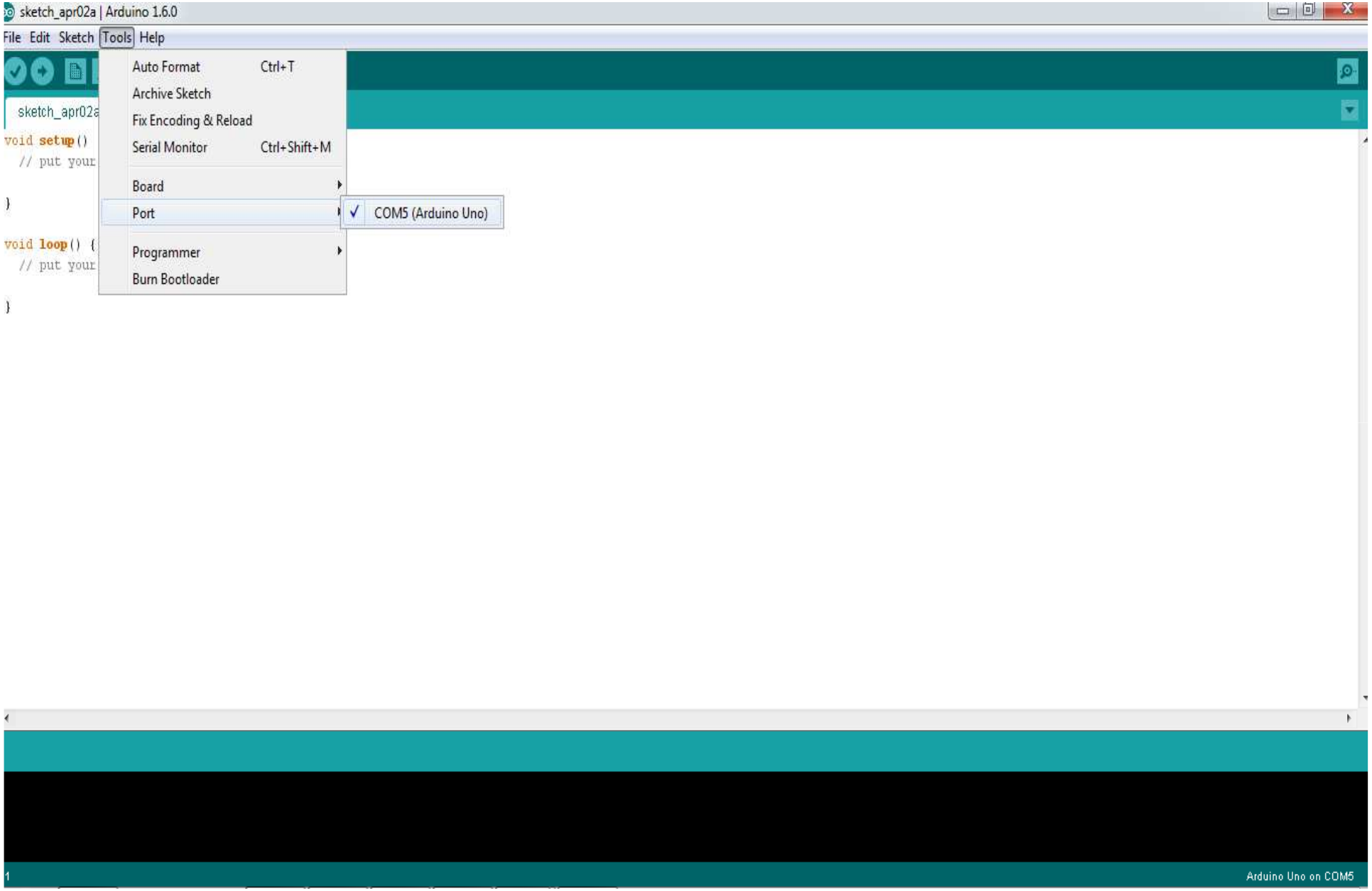
ARDUINO

FIRST PROGRAM ON IDE

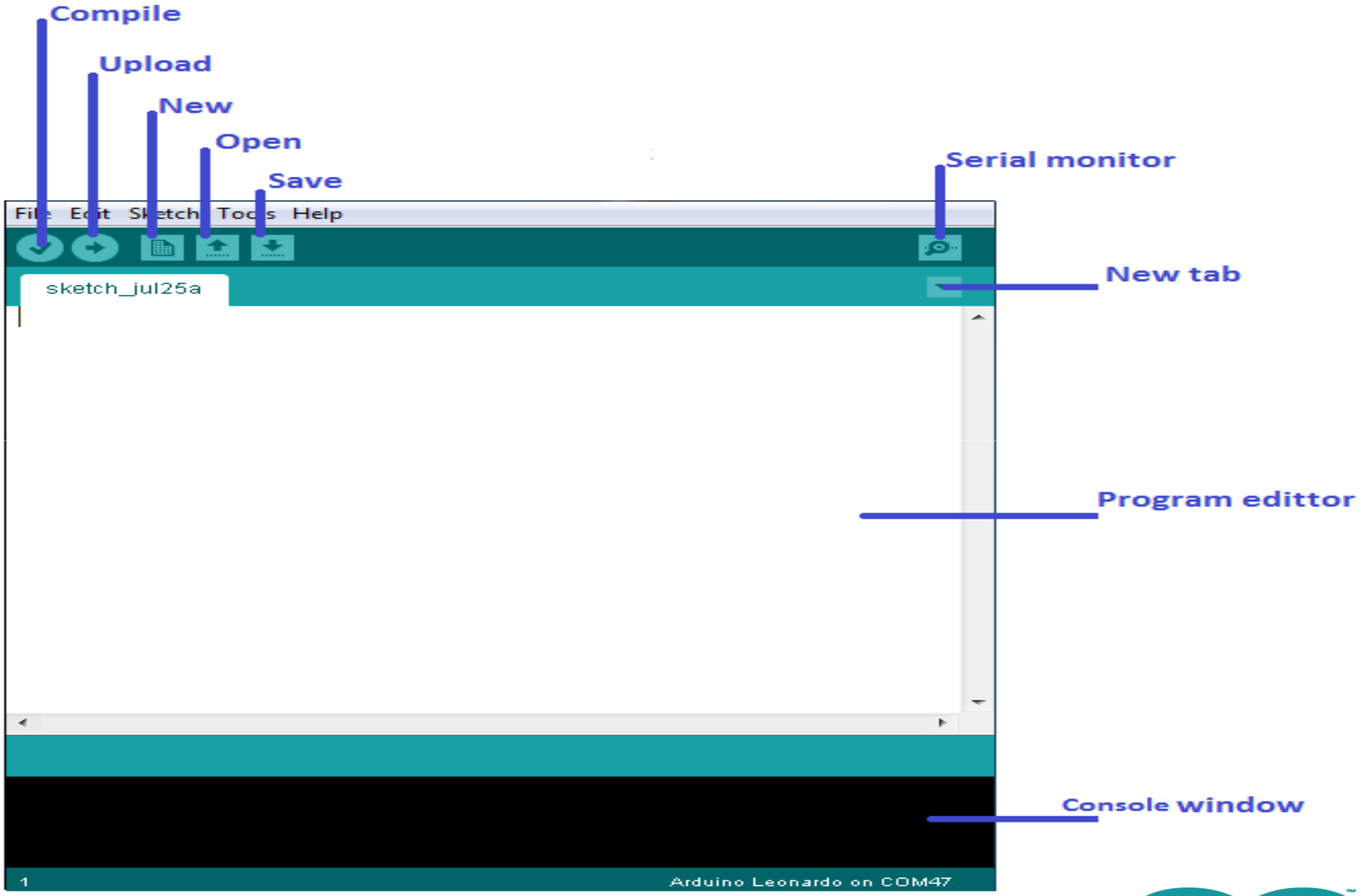
- **Click on arduino (∞) IDE**
- **Tools –Select board**
- **Serial port –COM port**



Configure the Serial Port



Arduino environment



ARDUINO

Structure of Arduino Uno code

sketch_mar12a

```
void setup() {
  // put your setup code here, to run once:

}

void loop() {
  // put your main code here, to run repeatedly:

}
```

ARDUINO

Serial Communication

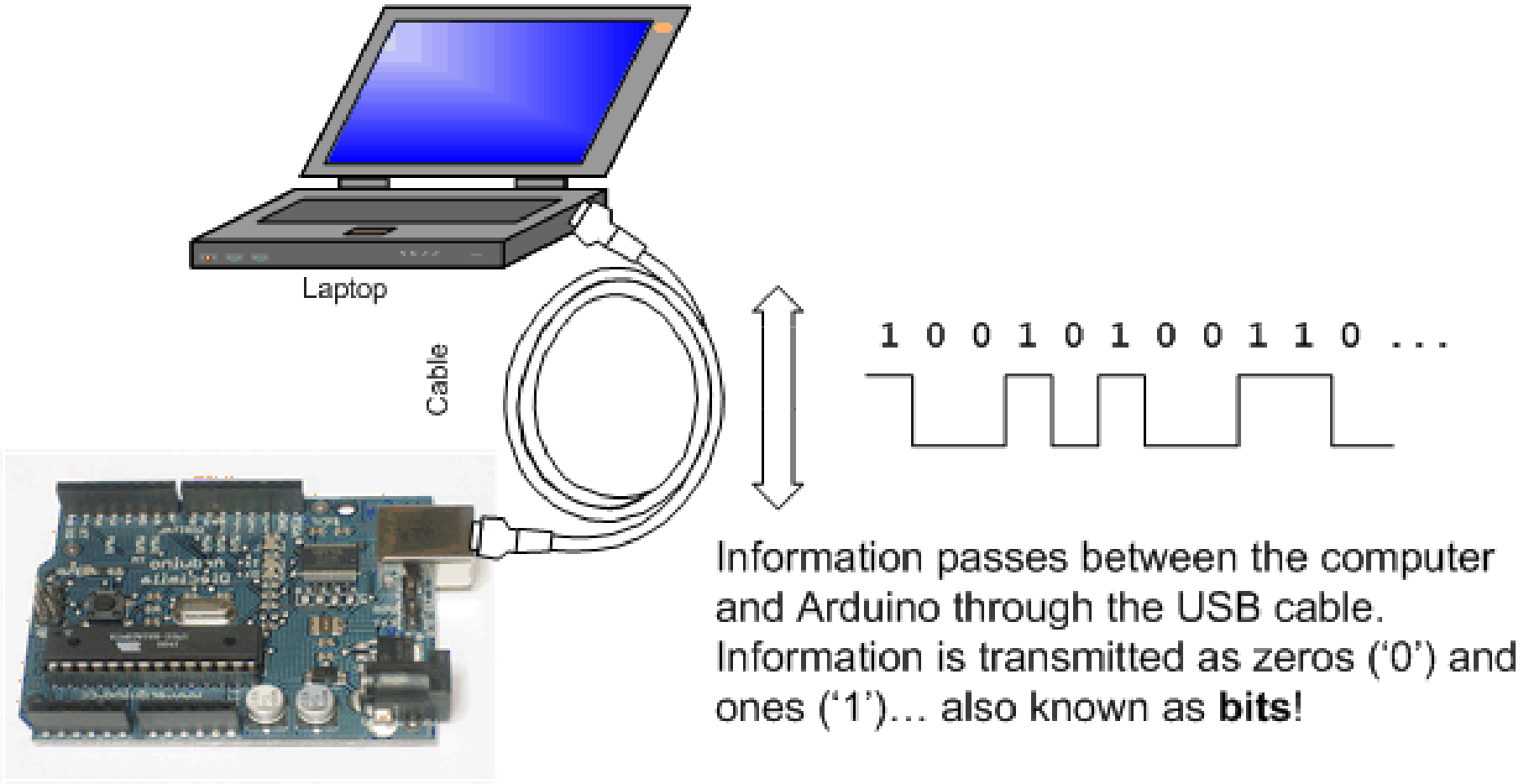
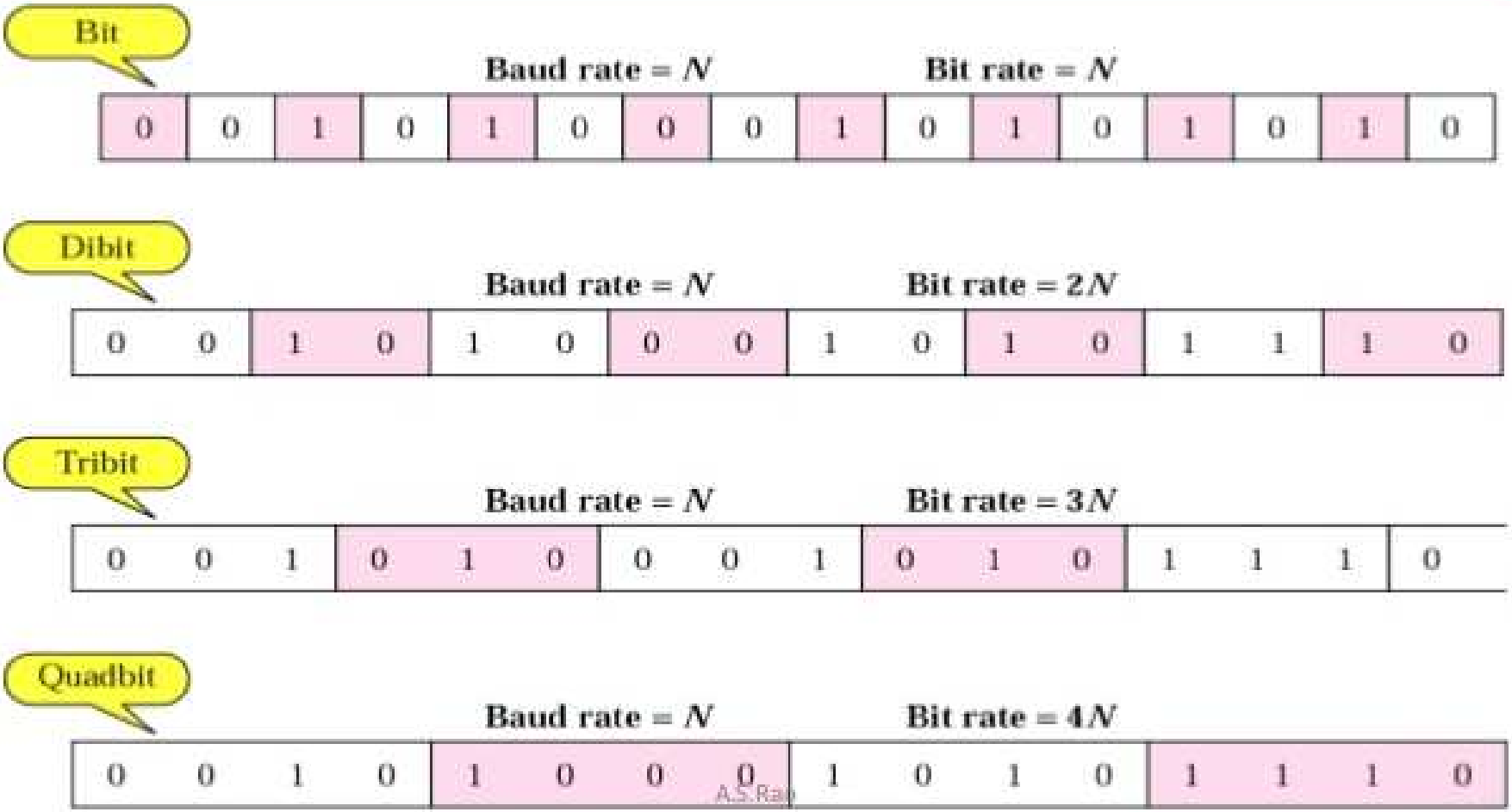


Image from http://www.ladyada.net/learn/arduino/lesson4.html

Bit Rate Vs Baud Rate

Bit rate is the number of bits per second. Baud rate is the number of signal units (symbols) per second. Baud rate is less than or equal to the bit rate.



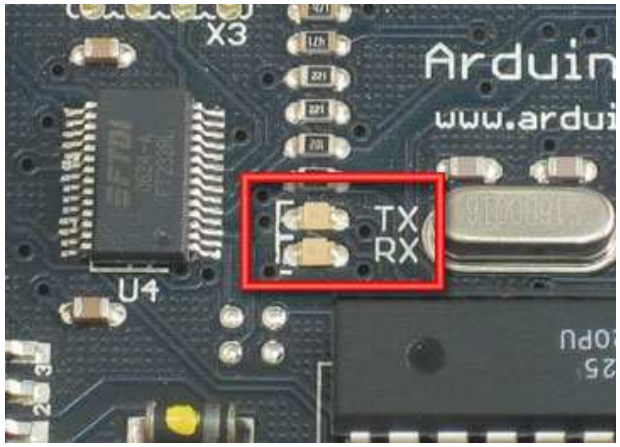
Serial Communications

- "Serial" because data is broken down into bits, each sent one after the other down a single wire.
- The single ASCII character 'B' is sent as:

```
'B' =  0 1 0 0 0 0 1 0
    =  L H L L L L H L
    =
```

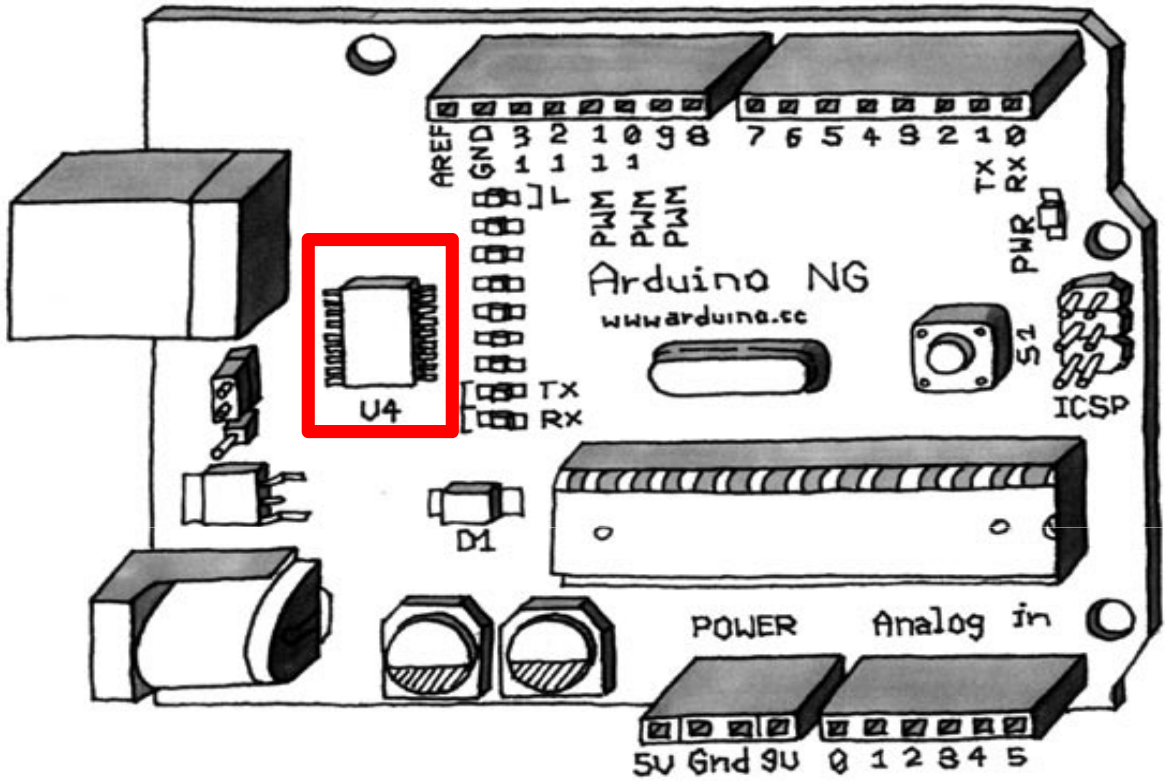
- Toggle a pin to send data, just like blinking an LED
- You could implement sending serial data with `digitalWrite()` and `delay()`
- A single data wire needed to send data. One other to receive.

Serial Communication



- ***Compiling*** turns your program into binary data (ones and zeros)
- ***Uploading*** sends the bits through USB cable to the Arduino
- The two LEDs near the USB connector blink when data is transmitted
  - **RX** blinks when the Arduino is receiving data
  - **TX** blinks when the Arduino is transmitting data

Serial-to-USB chip---what does it do?



The LilyPad and Fio Arduino require an external USB to TTY connector, such as an FTDI “cable”.
In the Arduino Leonardo a single microcontroller runs the Arduino programs and handles the USB connection.

Two different communication protocols

Serial (TTL):

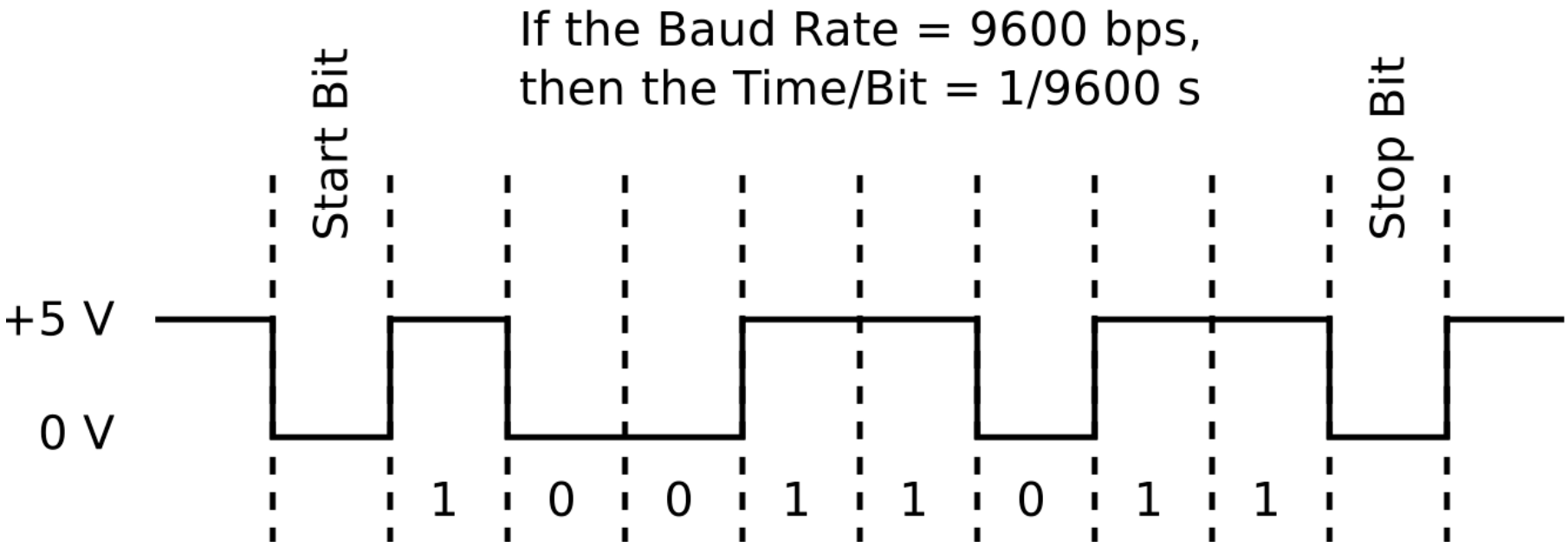


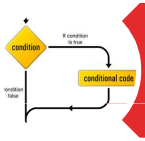
Image from http://www.fiz-ix.com/2013/02/introduction-to-arduino-serial-communication/

BIG 6 CONCEPTS

digitalWrite()

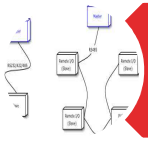
analogWrite()

digitalRead()



if() statements / Boolean

analogRead()



Serial communication

Delay()

ARDUINO

pinMode()

Syntax:

pinMode(pin, mode)

Parameters

pin: the number of the pin whose mode you wish to set

mode: INPUT, OUTPUT

commands to know…

- **pinMode**(pin, INPUT/OUTPUT);
- ex: **pinMode**(13, OUTPUT);

- **digitalWrite**(pin, HIGH/LOW);
- ex: digitalWrite(13, HIGH);

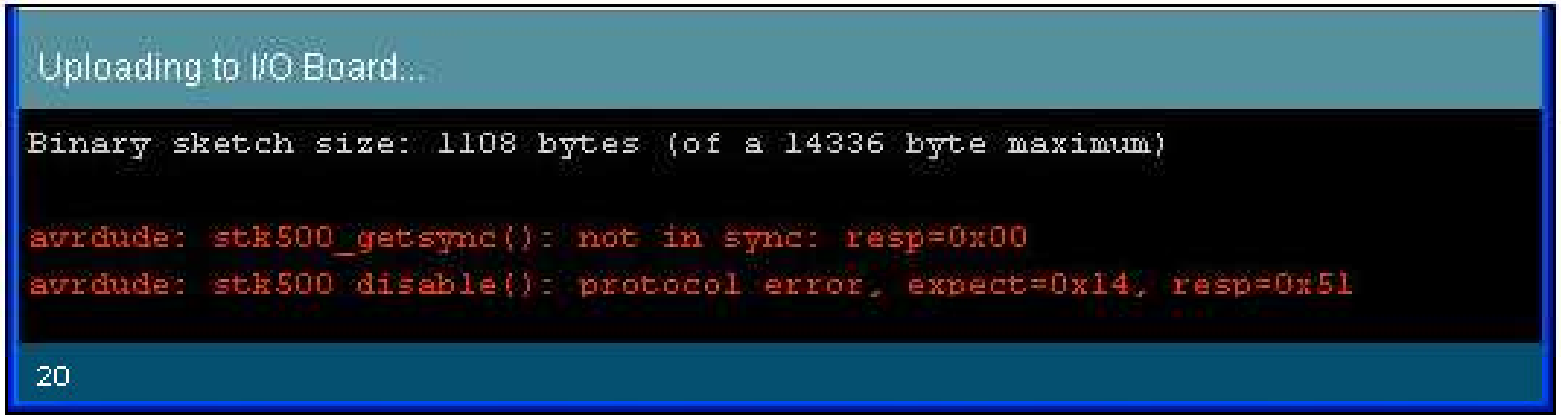
- **delay**(time_ms);
- ex: delay(2500); // delay of 2.5 sec.

- **// NOTE: -> commands are CASE-sensitive**

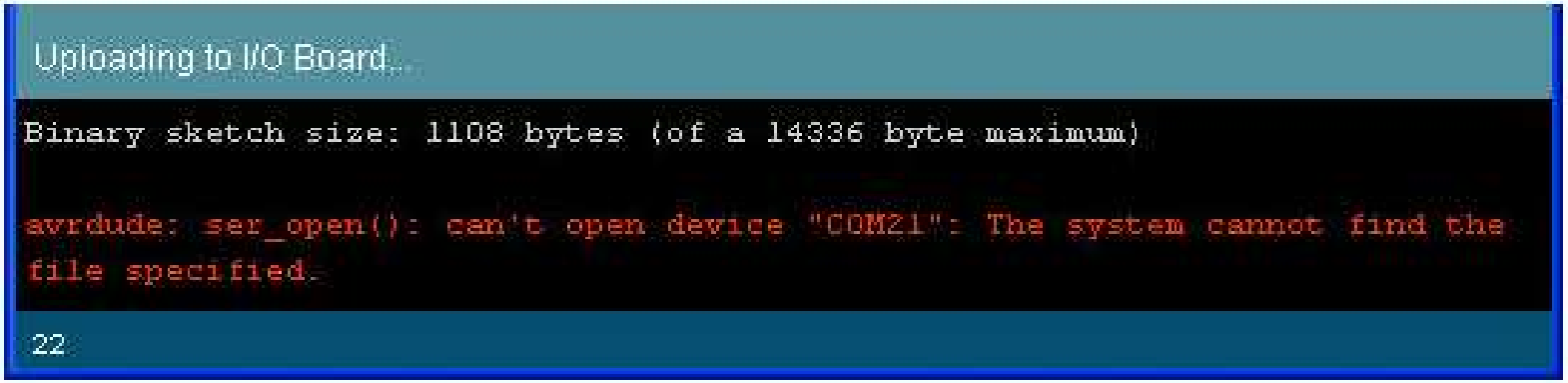
ARDUINO

Error

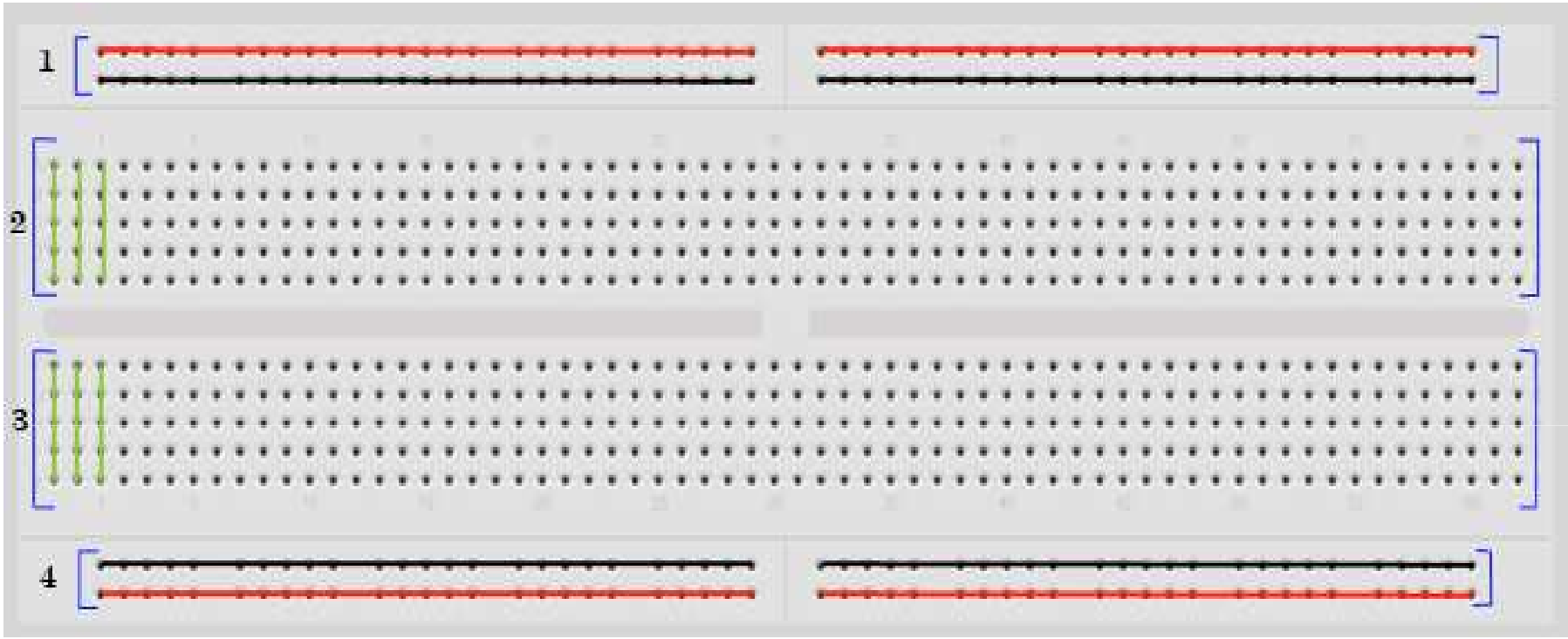
- avrdude: stk500_getsync(): not in sync: resp=0x00



- can't open device "COM10": The system cannot find the file specified

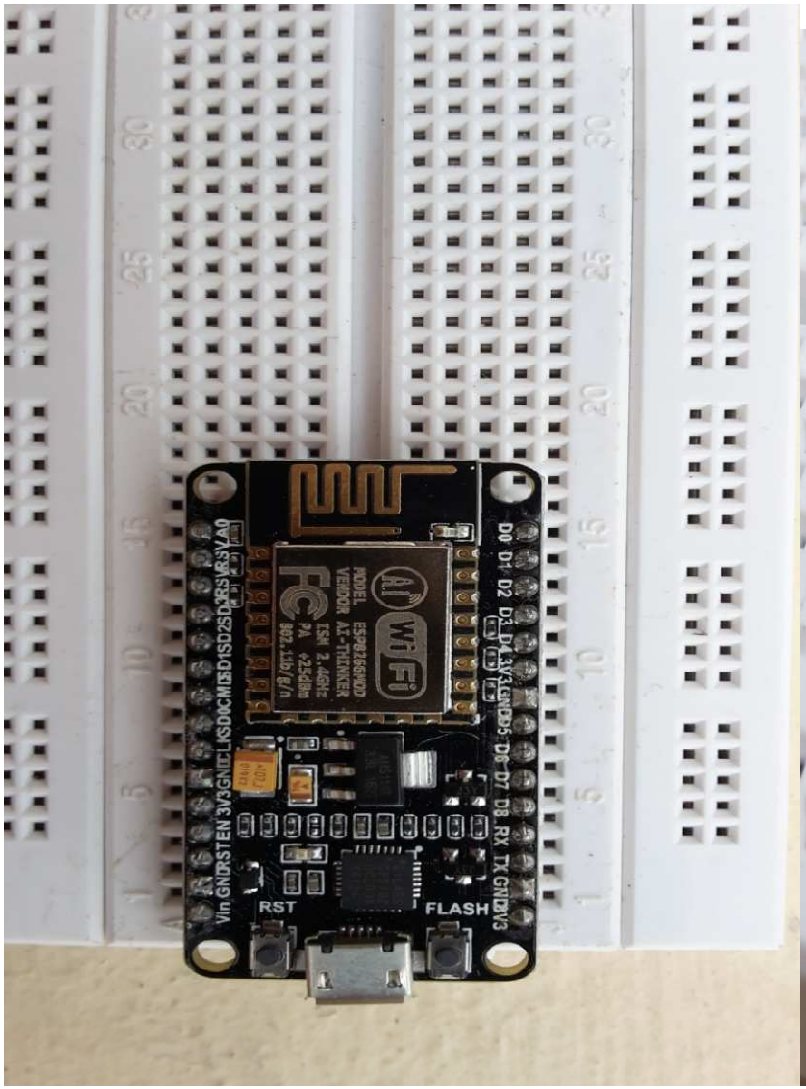


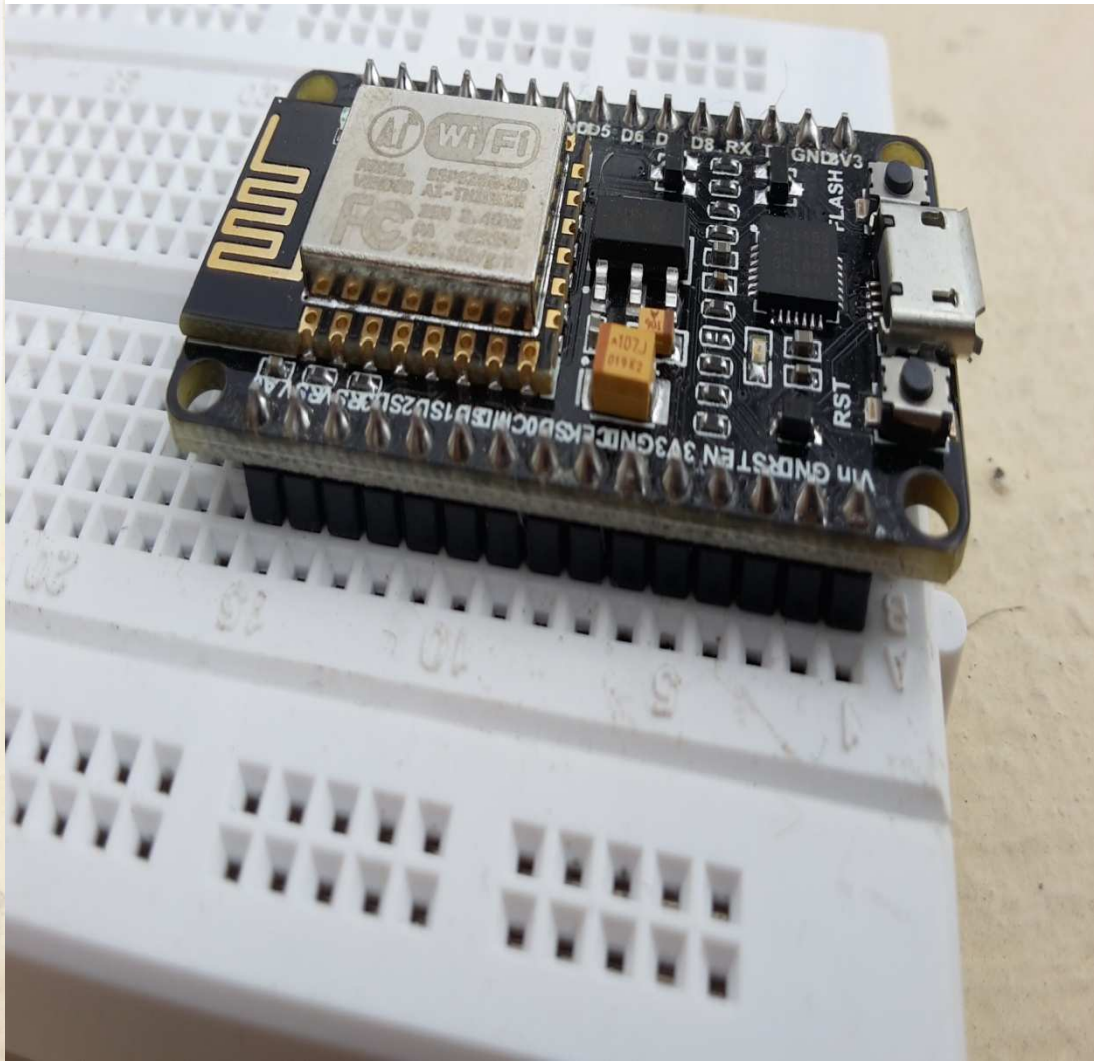
BREAD BOARD



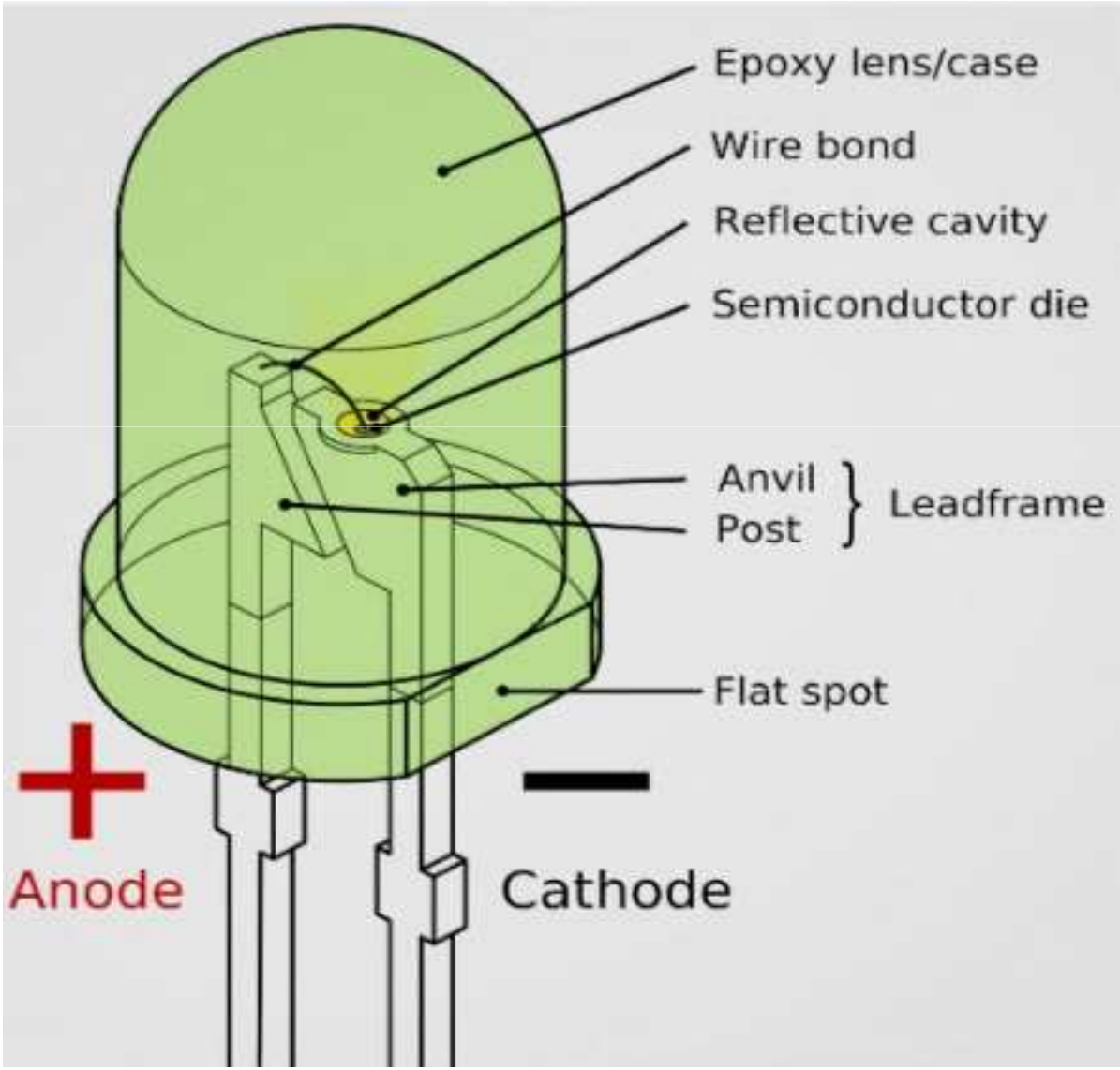
ARDUINO

Nodemcu assembling on Breadboard





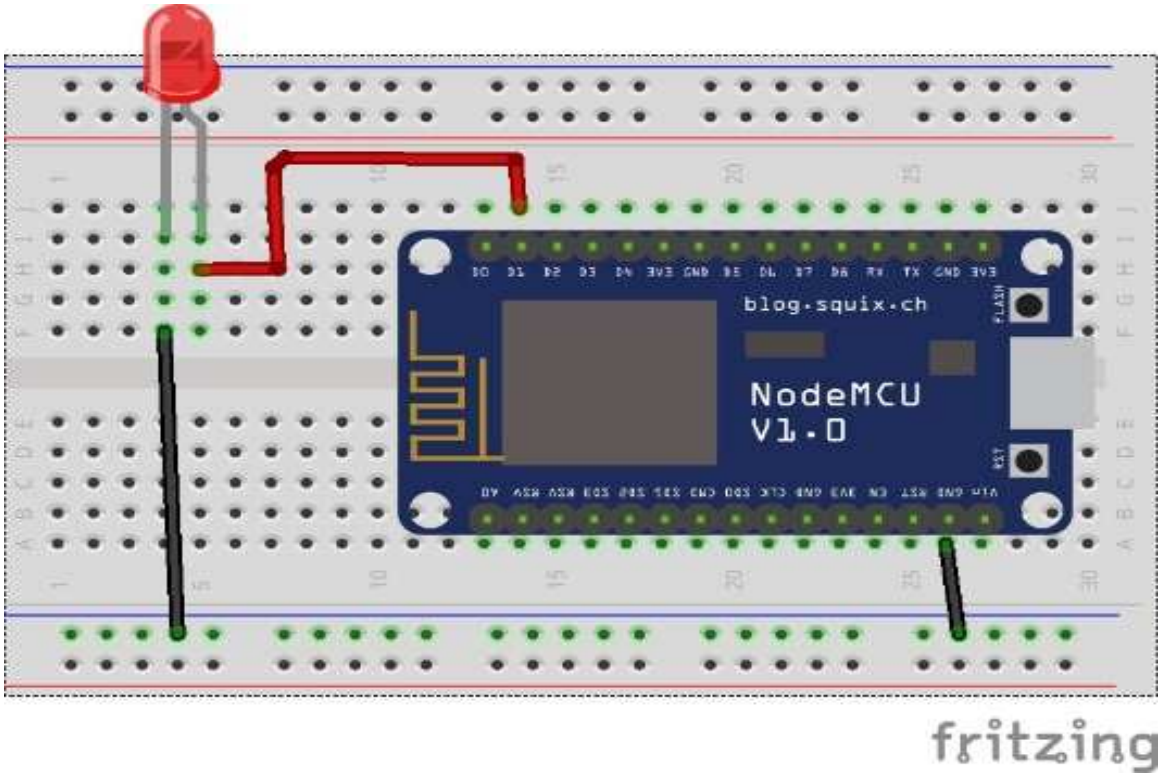
LED (Light emitting diode)



ARDUINO

BLINKING OF AN LED

Interfacing with NodeMcu



+ve of led to GPIO5
-ve of led to GND

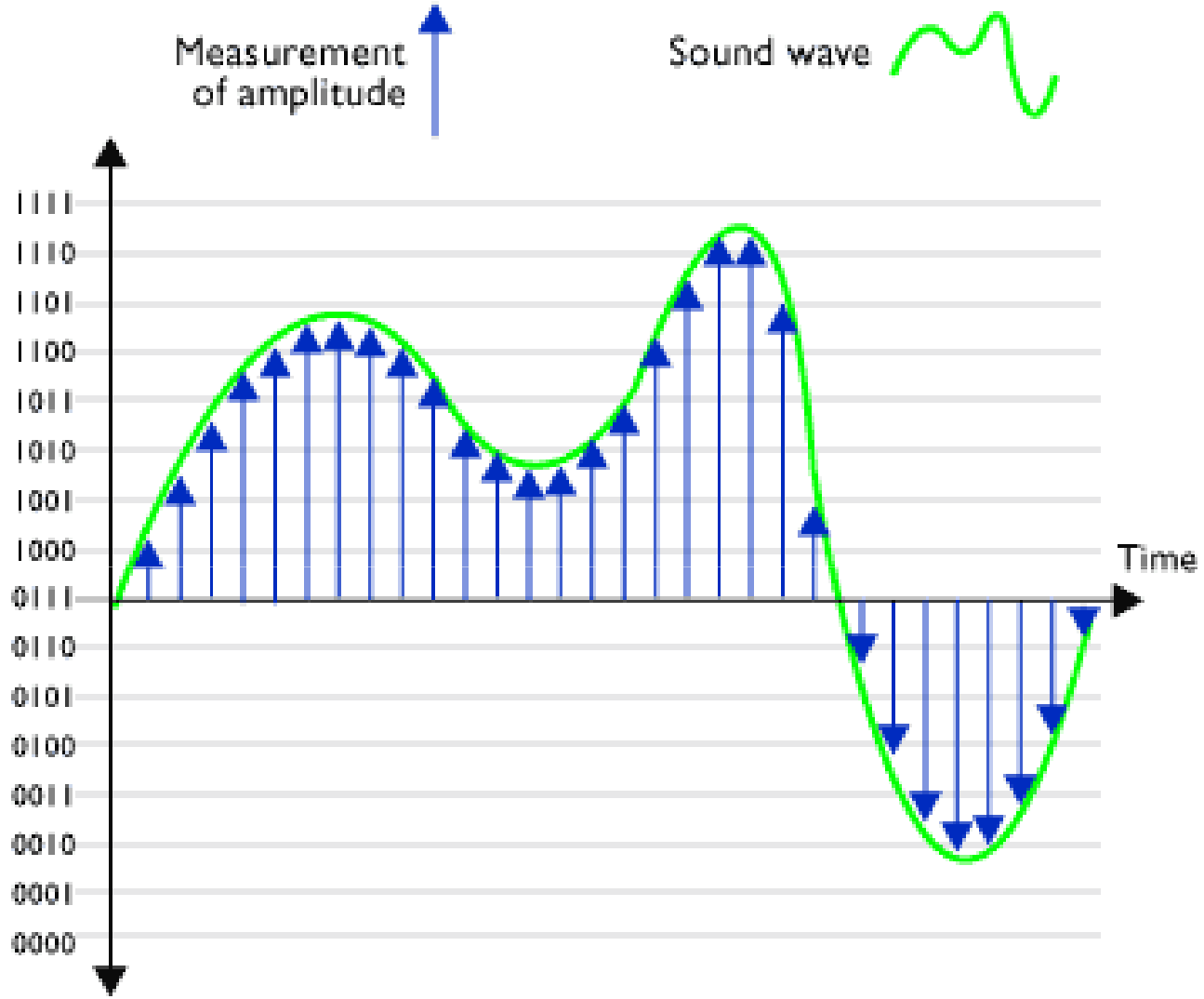
analogRead()(pin:A0)

- Reads the value from the specified analog pin. The Arduino board contains a 6 channel.
- 10-bit ADC. This means that it will map input voltages between 0 and 5 volts into integer values between 0 and 1023. $\frac{1023}{5} = \frac{ADC\ Reading}{Analog\ Voltage\ Measured}$
- It takes about 100 microseconds to read an analog input

- **Syntax**
- **analogRead(pin)**

- **pin: the number of the analog input pin to read from (0 to 5 on most boards)**

- Resolution of ADC..?
- Effect of resolution



Each measurement is assigned a number (byte) according to its amplitude. The end result is a file comprising a string of bytes, eg ...
1001 1110 0001 1010 0111 0100 1111 1101 etc

digitalRead() (pin 0-13)

- **Description**
- **Reads the value from a specified digital pin, either HIGH or LOW.**

Syntax

digitalRead(pin)

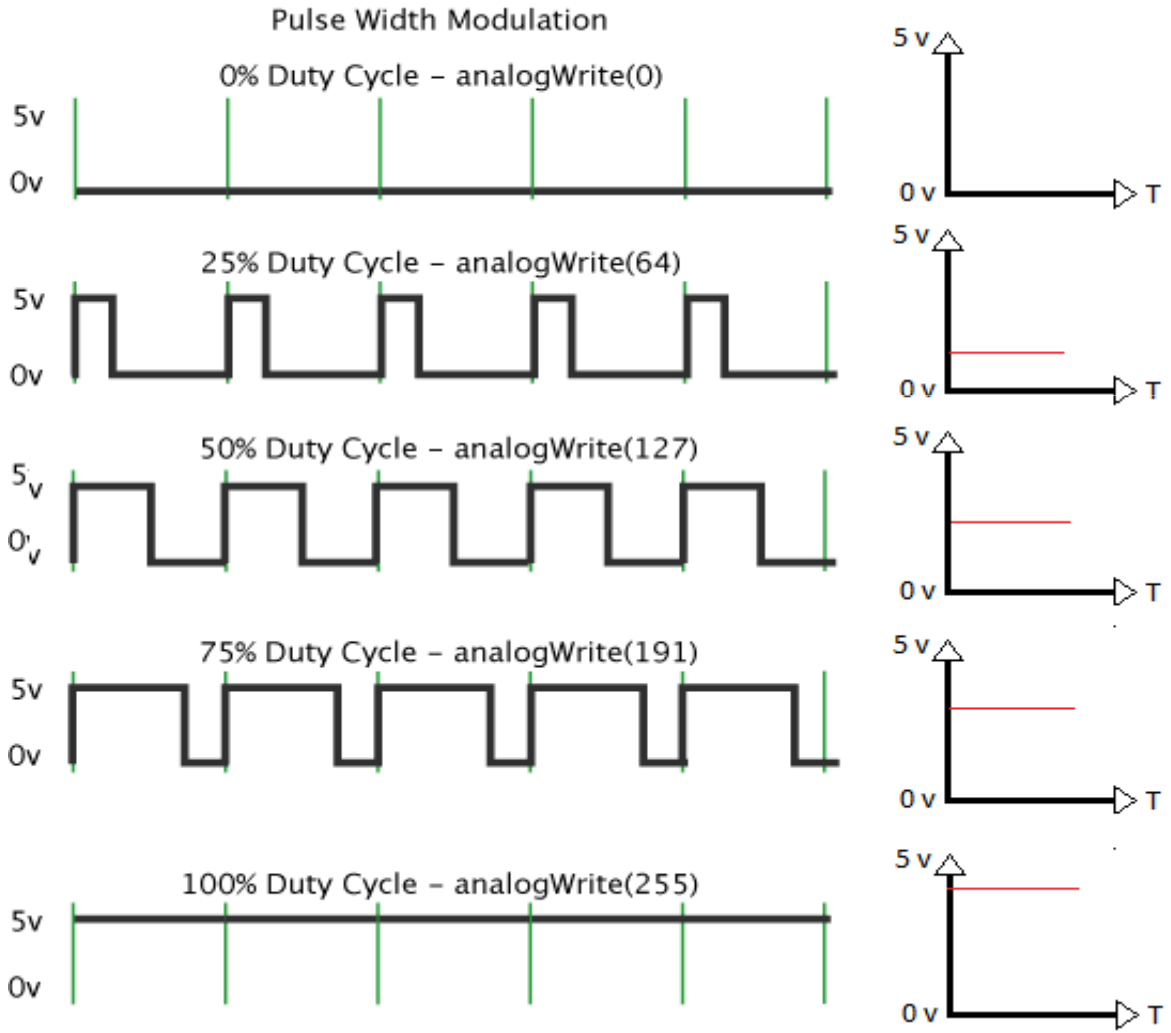
pin: the number of the digital pin you want to read (*int*)

Returns

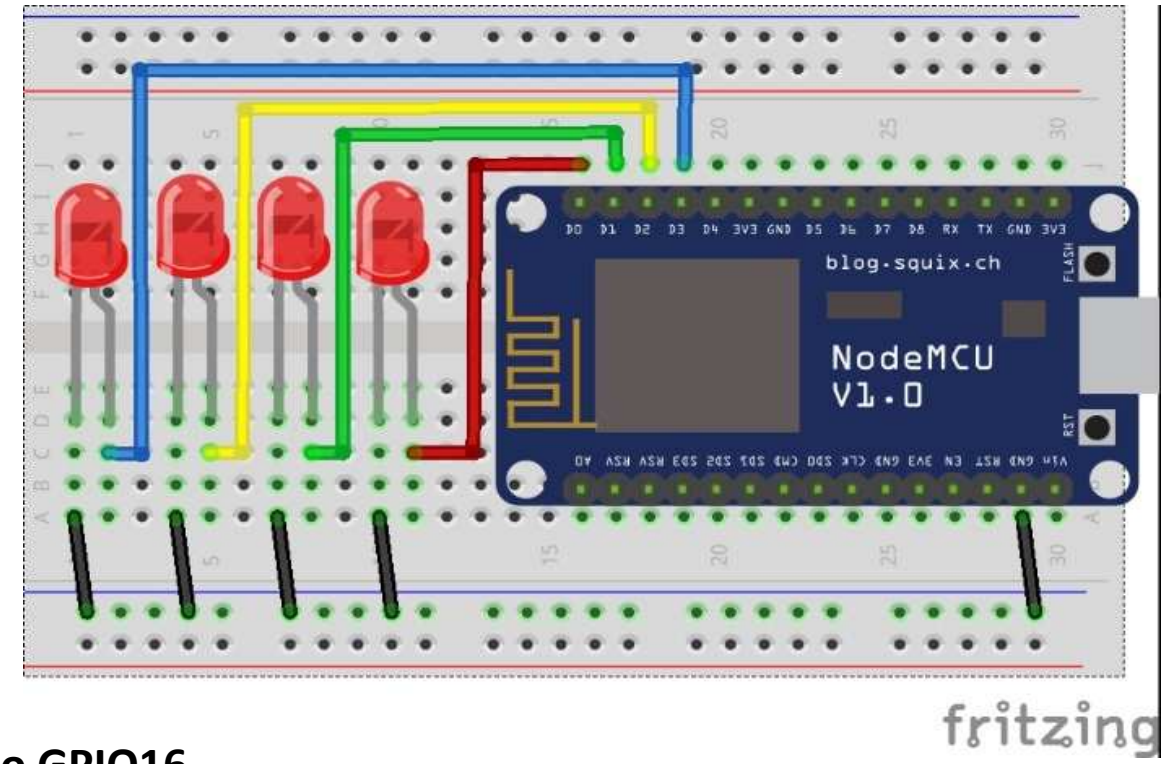
HIGH or LOW

```
int analogPin = 3;connected to analog pin 3
          // outside leads to ground and +5V
int val = 0;                  // variable to store
the value read

void setup()
{
  Serial.begin(9600);                //  setup
serial
}

void loop()
{
  val = analogRead(analogPin);     // read
the input pin
  Serial.println(val);                  // debug
value
}
```

Pulse Width Modulation(PWM)



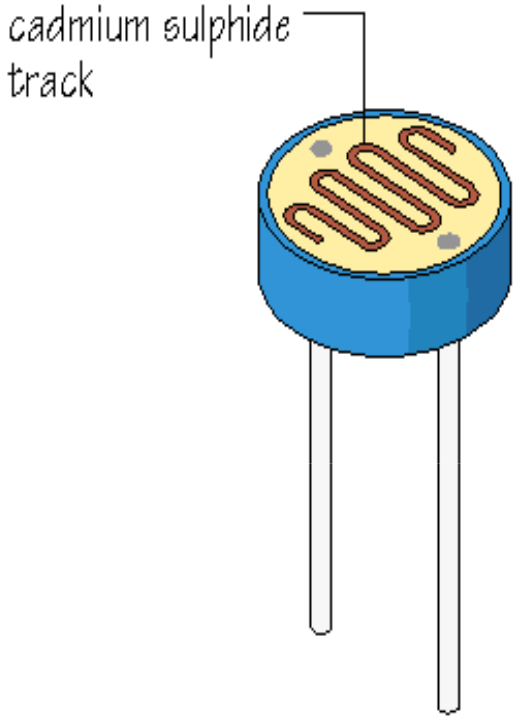
LED Pattern using NodeMcu

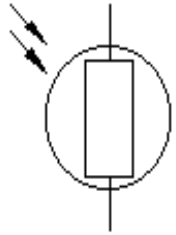


+ve of led1 to GPIO16
+ve of led2 to GPIO5
+ve of led4 to GPIO4
+ve of led1 to GPIO0
-ve of led to GND

LDR Sensor With NodeMcu

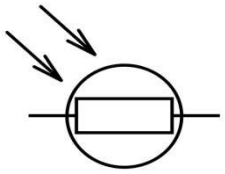
LDR(LIGHT DEPENDENT RESISTOR)





circuit symbol

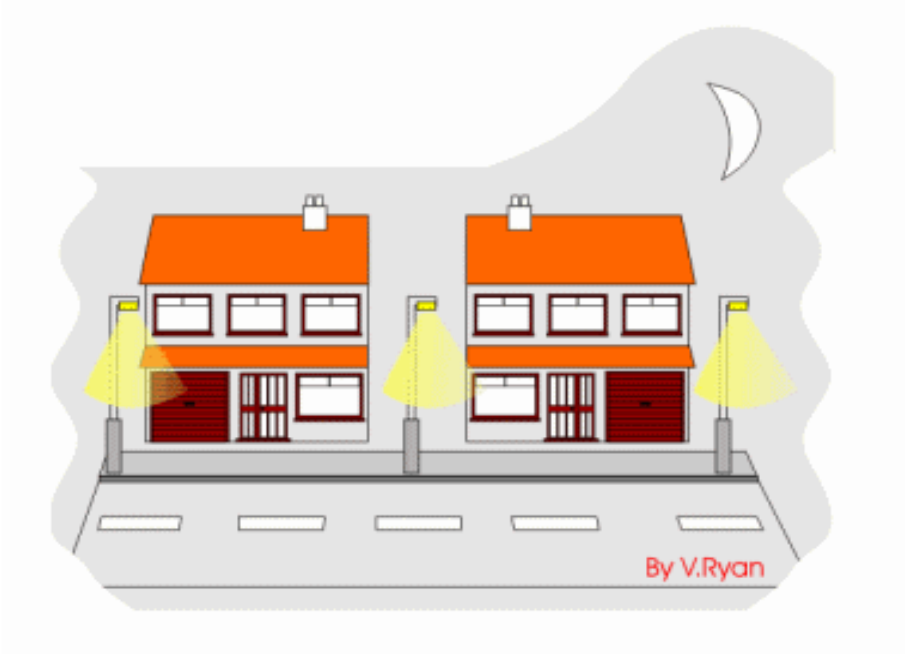
Symbol of LDR:

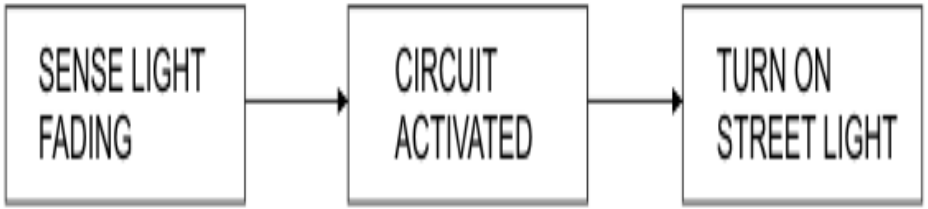


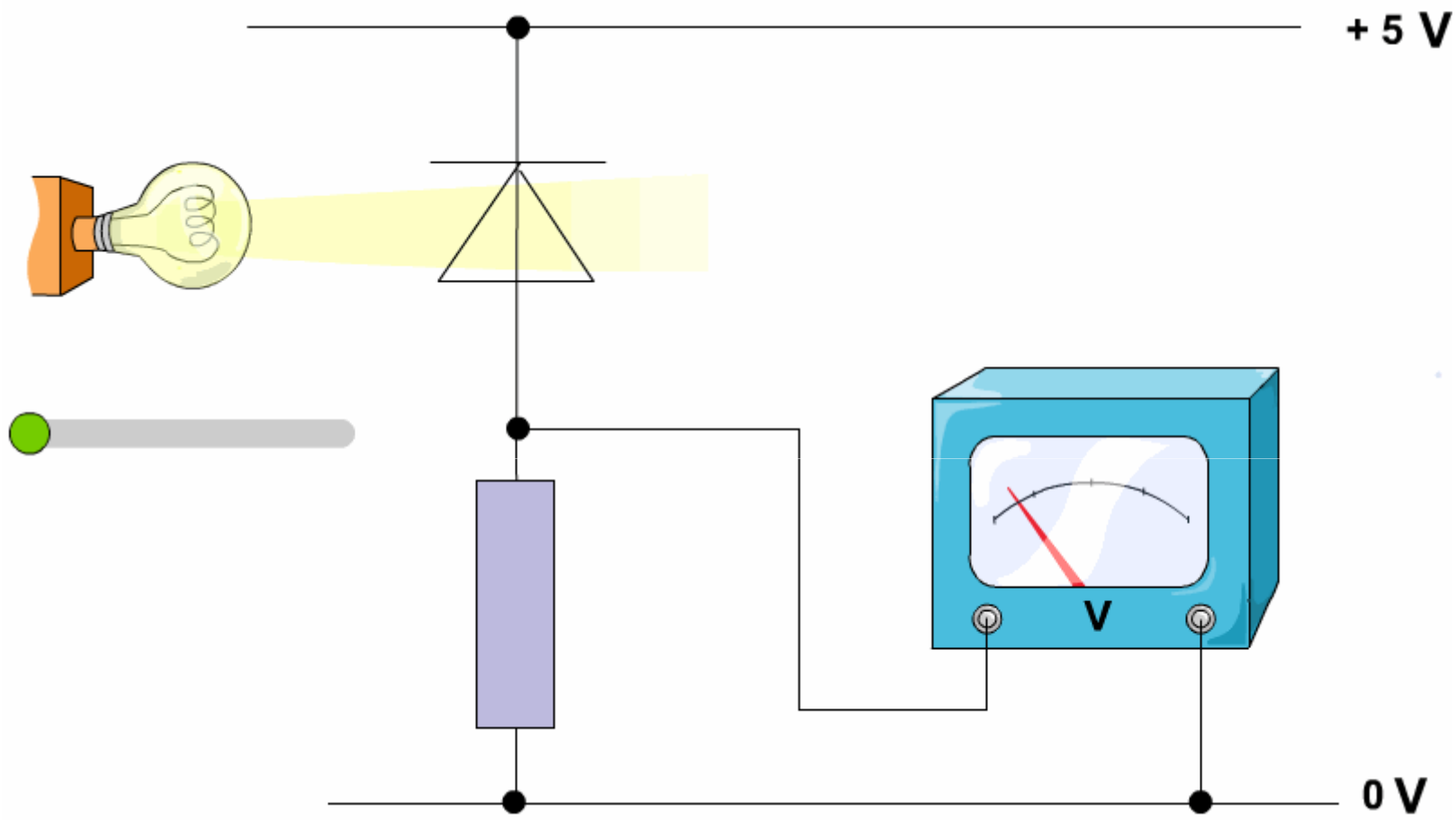
- **LDR stands for Light dependant resistor. An LDR is usually made of a semiconductor material(Normally Silicon) doped with a small percentage of a valence 5 material (commonly Arsenic), to make it an "N" material.**
- **Another word for LDR is photoresistor.**
- **The resistance of LDR decreases with increase in the intensity of light. An LDR works in the similar manner as any other analog device would work.**

APPLICATIONS

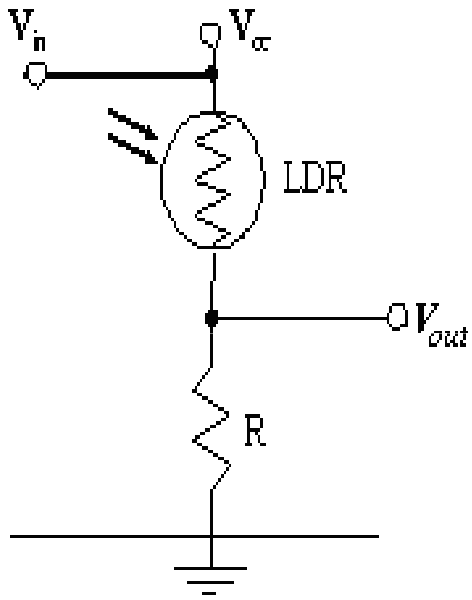
- **They can be used to respond to events such as the transition from daytime to night-time (and vice versa) for home automation**
- **Gardening applications, and are often used to control street lighting.**





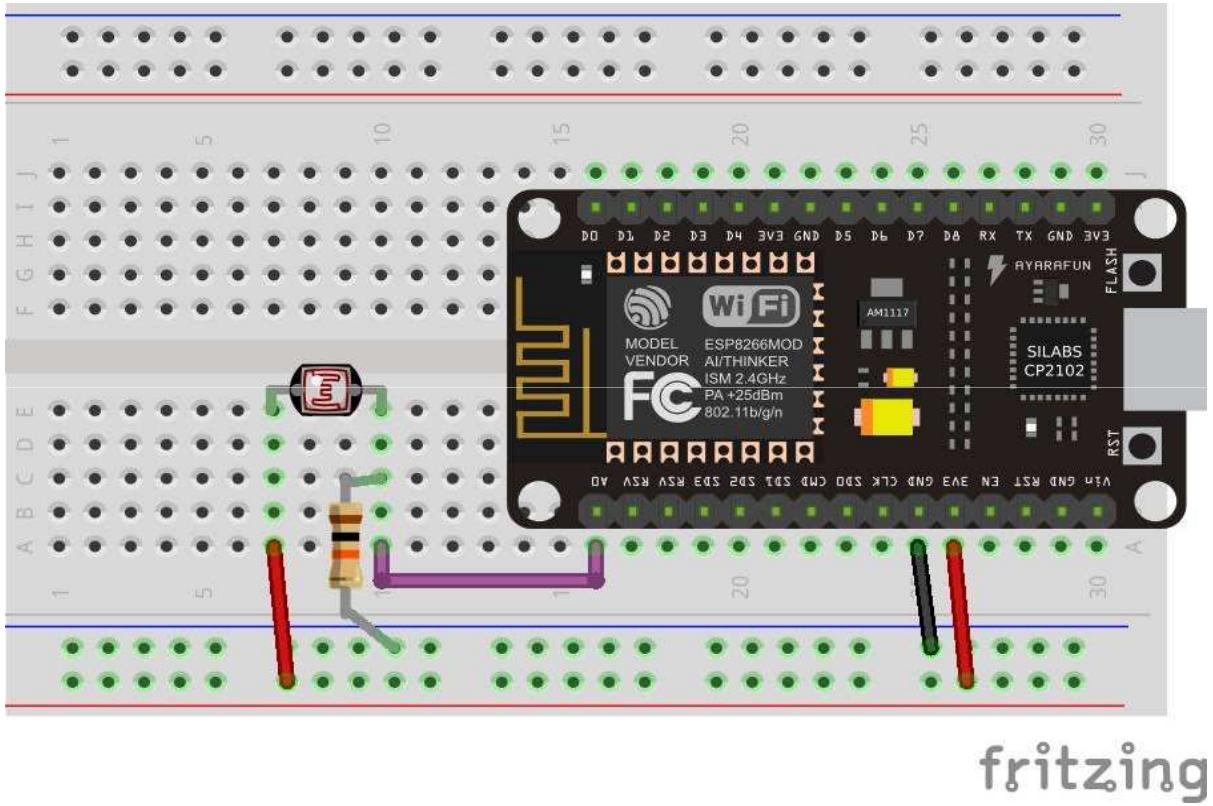

+ 5 V
V
0 V

INTERFACING LDR WITH NodeMcu



Darkness -> LDR high -> Vout is low

Bright -> LDR low -> Vout is high

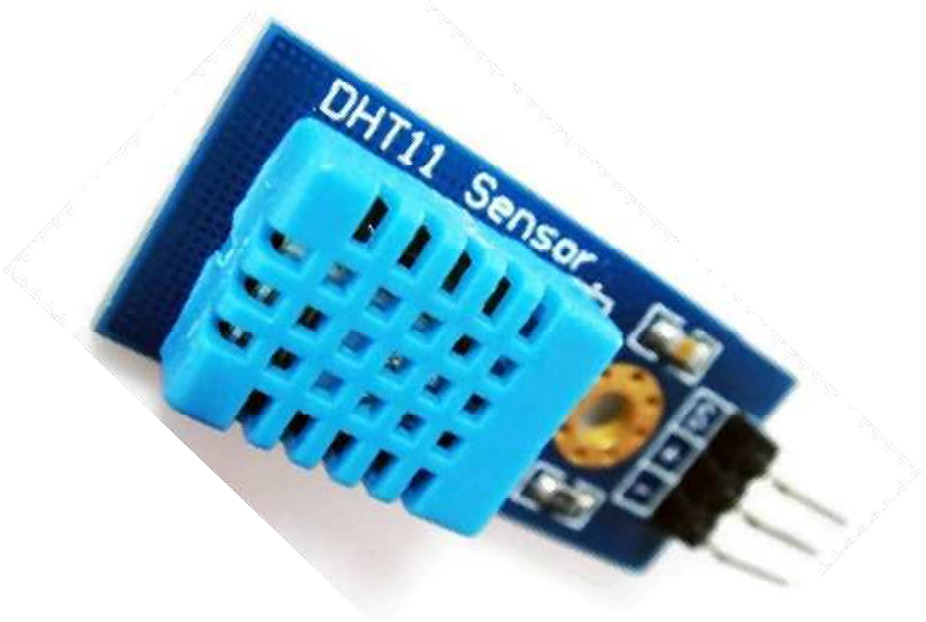


DHT11 Temperature and Humidity Sensor

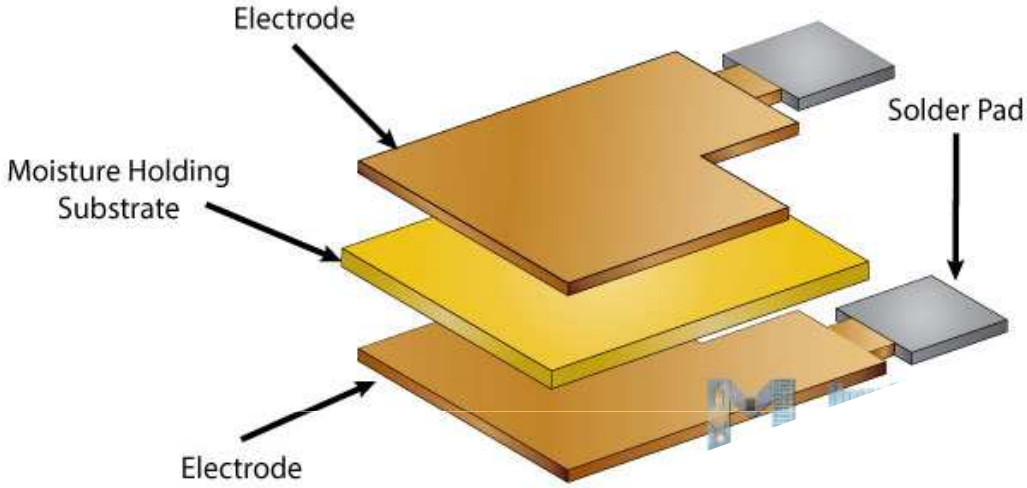
- **DHT11 is a basic, ultra low-cost digital temperature and humidity sensor**
- **Capacitive humidity sensor and a thermistor to measure the surrounding air**
- **Detects water vapor by measuring the electrical resistance between two electrodes**
- **Humidity sensing component is a moisture holding substrate with electrodes applied to the surface**

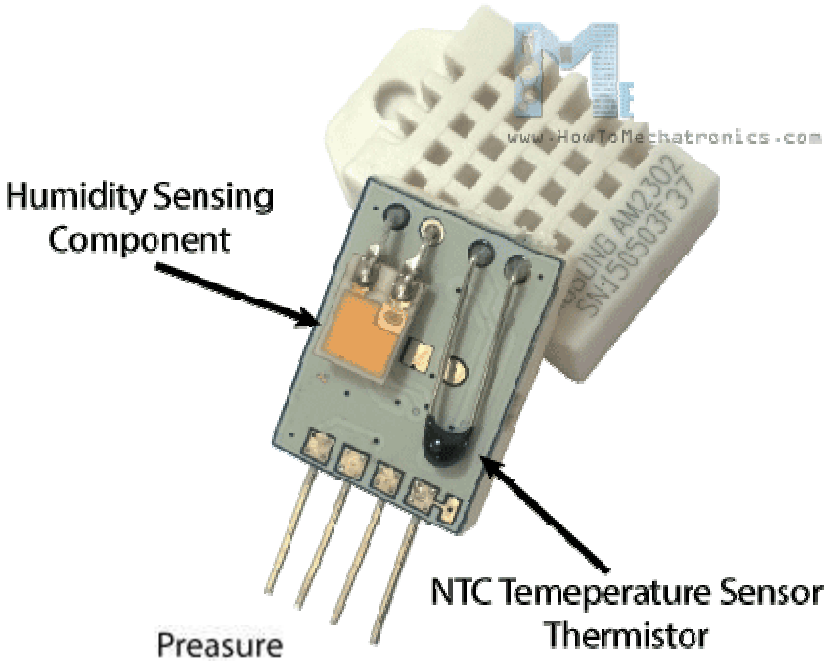
Technical Specification:

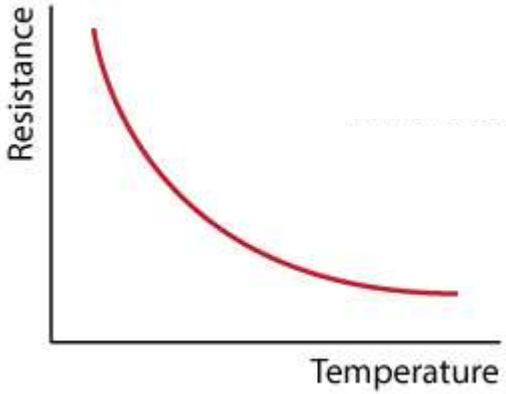
- Humidity Range: 20-90% RH
- Humidity Accuracy: ±5% RH
- Temperature Range: 0-50 °C
- Temperature Accuracy: ±2% °C
- Operating Voltage: 3V to 5.5V

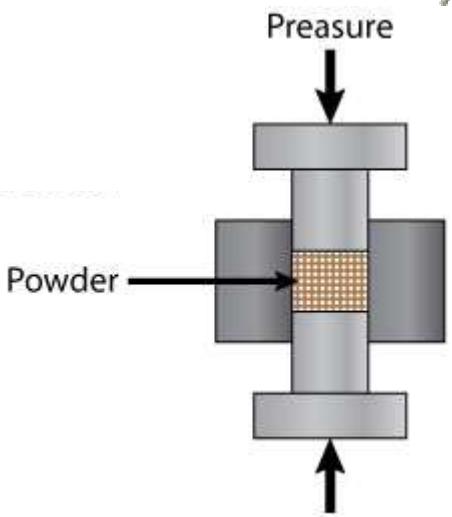


DHT11 Temperature and Humidity Sensor







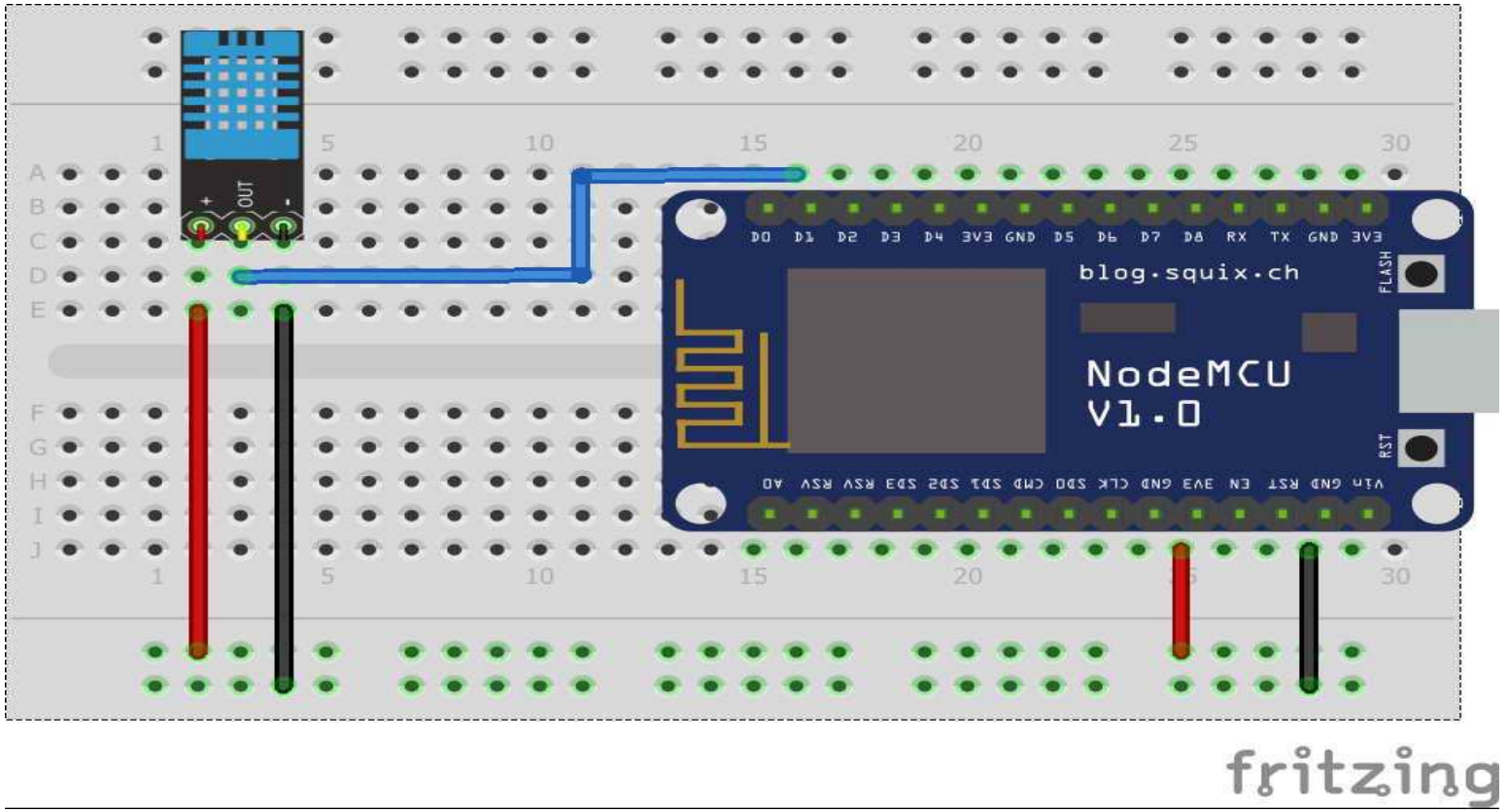


INTERFACING DHT11 WITH NodeMcu

+ve of DHT to 3.3v

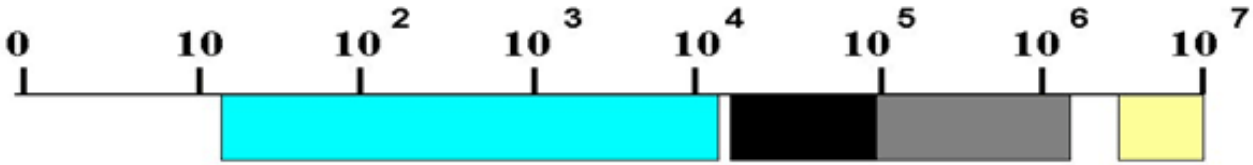
-ve of DHT to GND

DATA of DHT to D1(GPIO5)

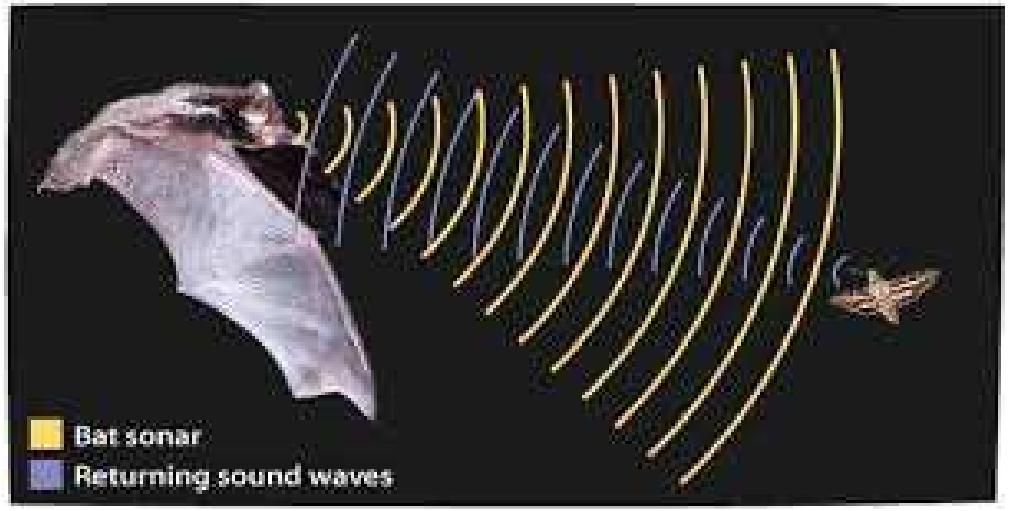


Ultrasonic

The Frequency Ranges of the Sound



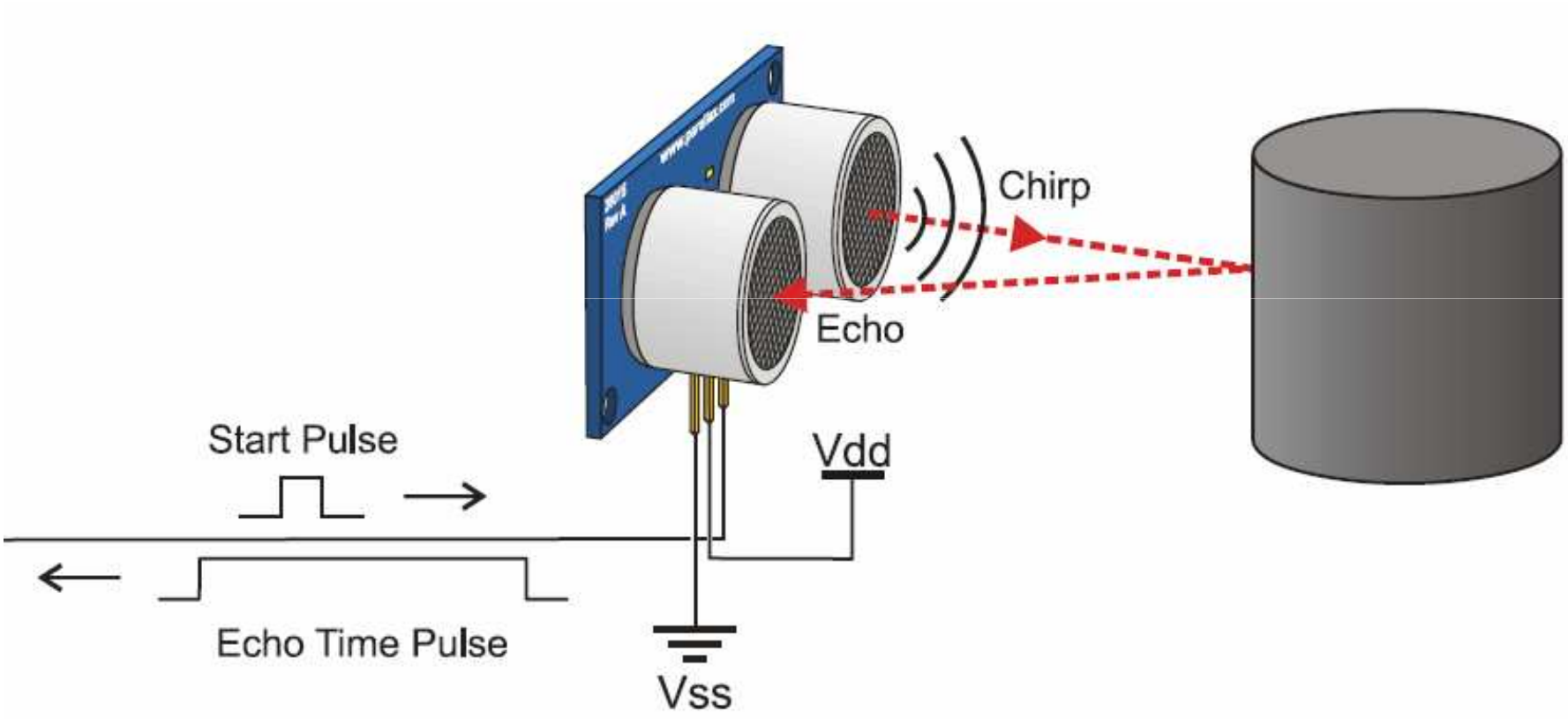
Human hearing		16Hz - 18kHz
Conventional power ultrasound		20kHz - 100kHz
Extended range for sonochemistry		20kHz - 2MHz
Diagnostic ultrasound		5MHz - 10MHz



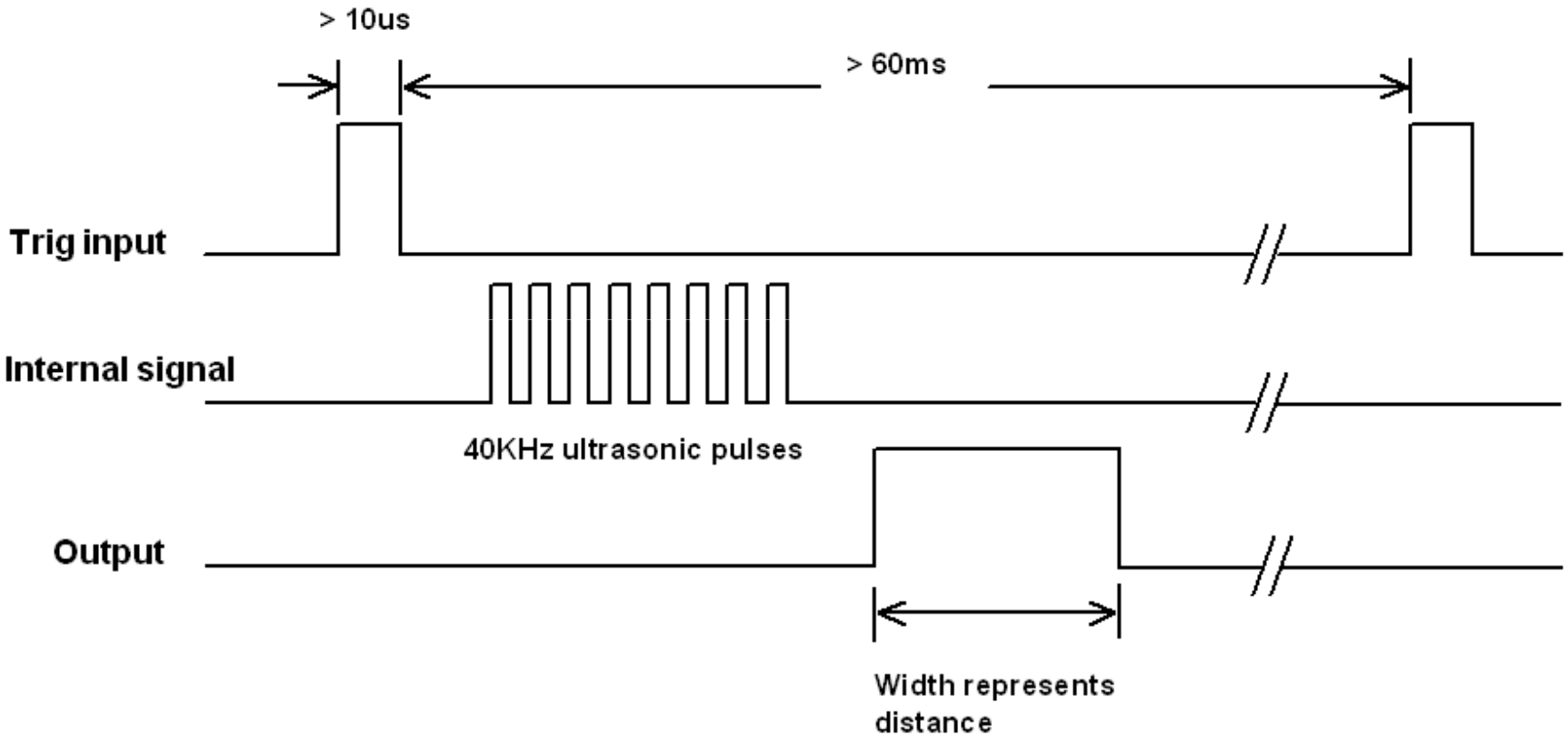
Pin Outs

VCC Trig Echo Gnd

Working

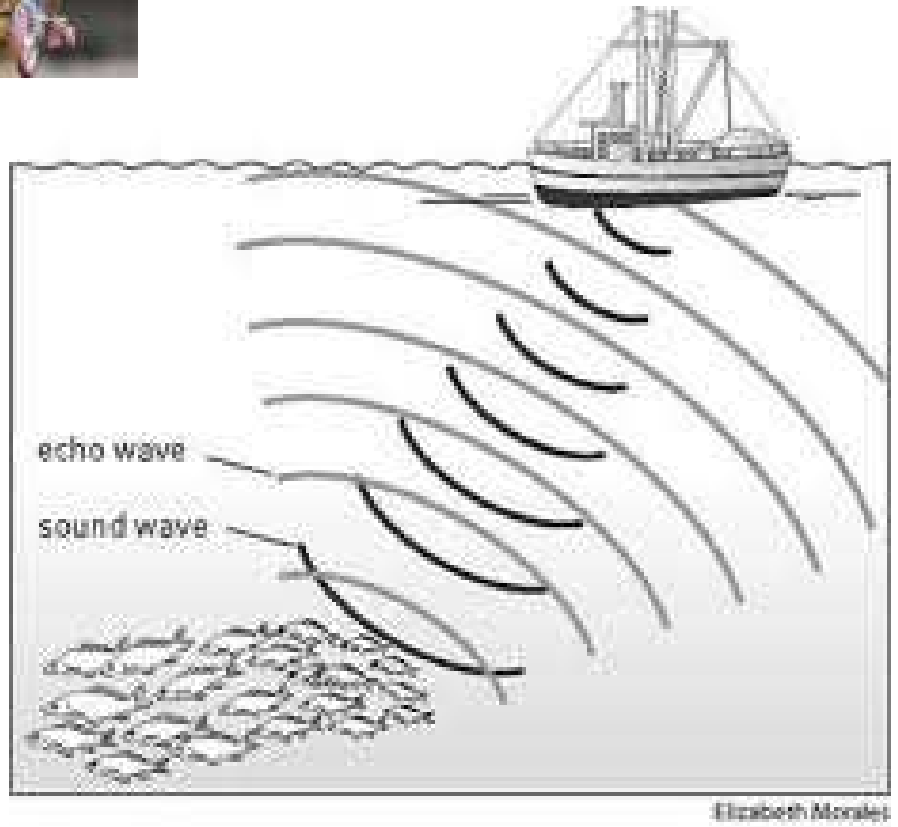


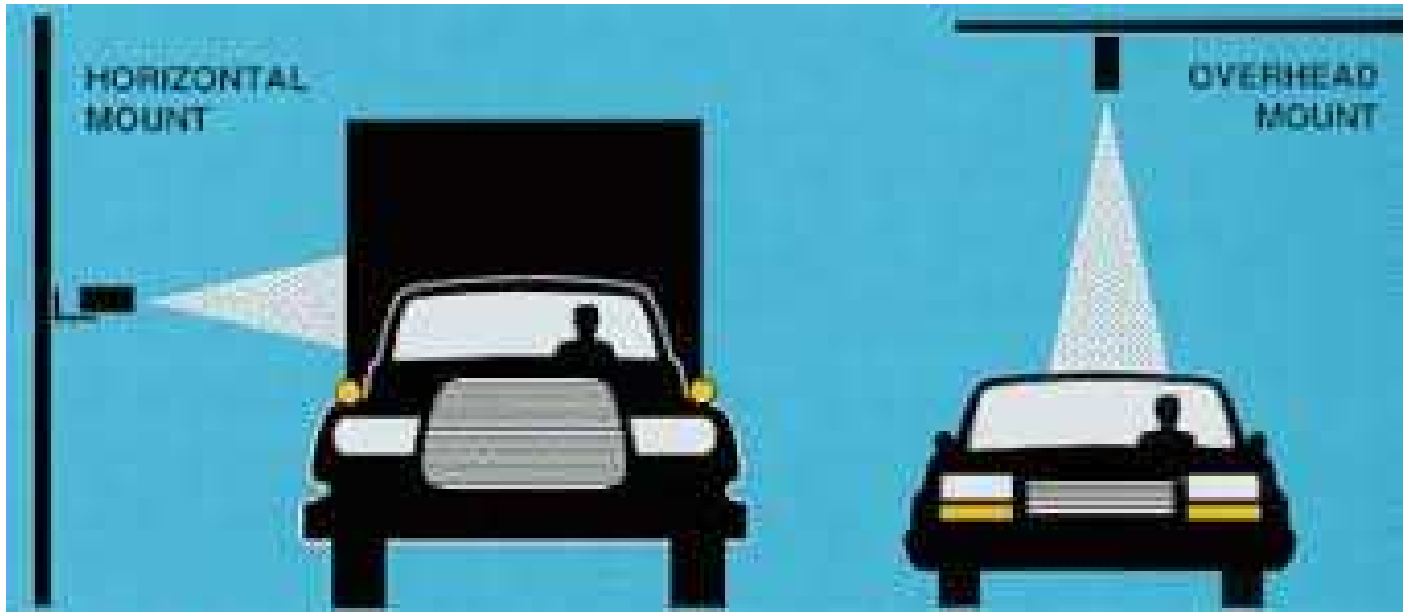
Timing Diagram

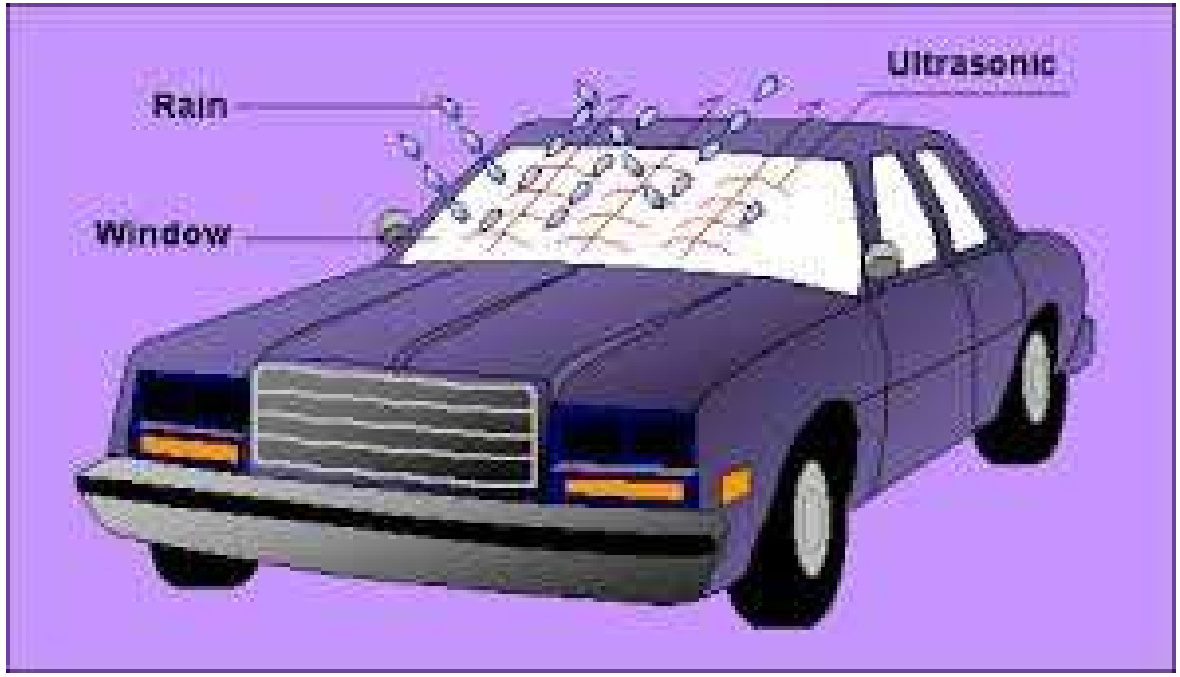


Applications

- Park assistance
- Distance measuring device
- SONAR
- Fishing



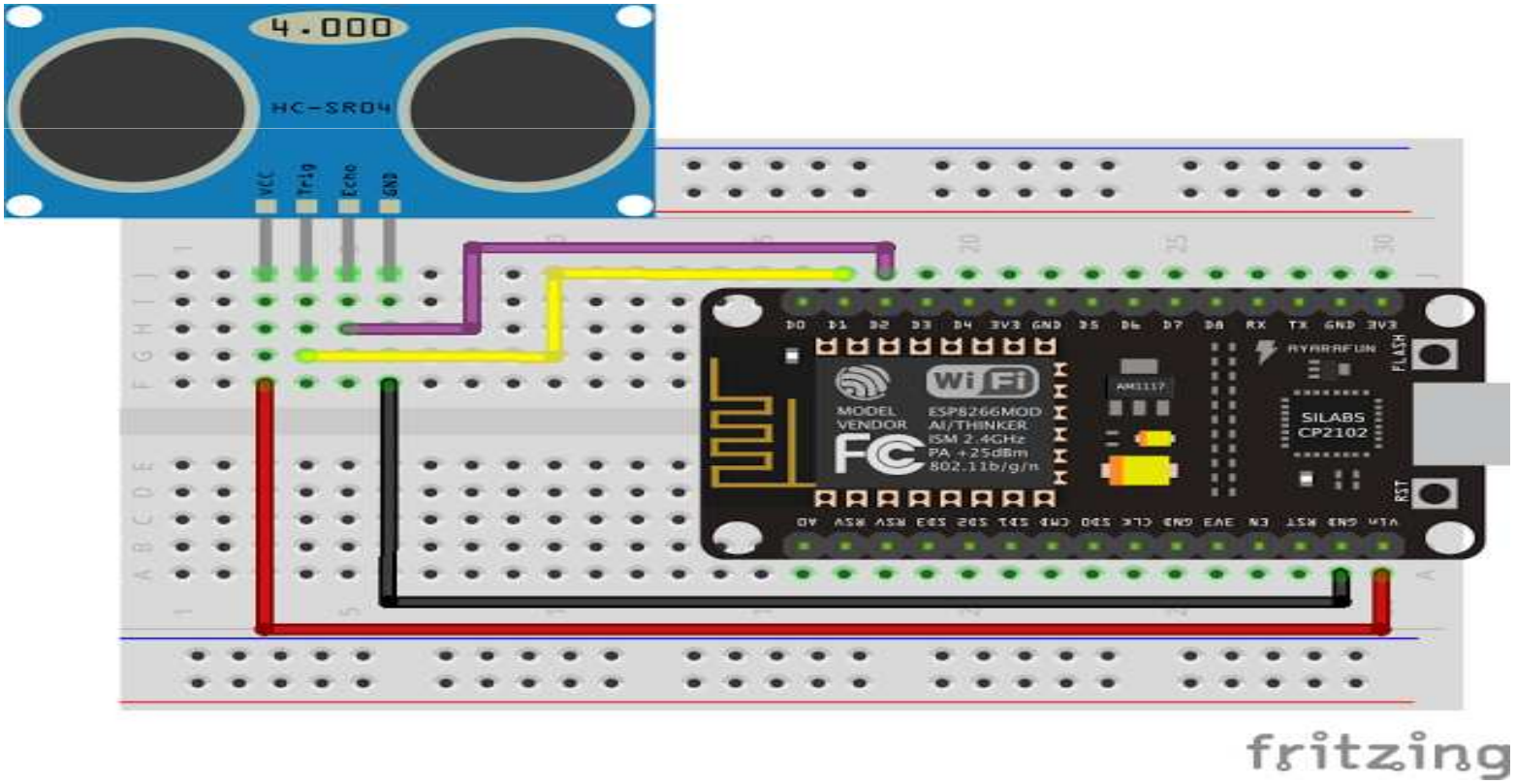


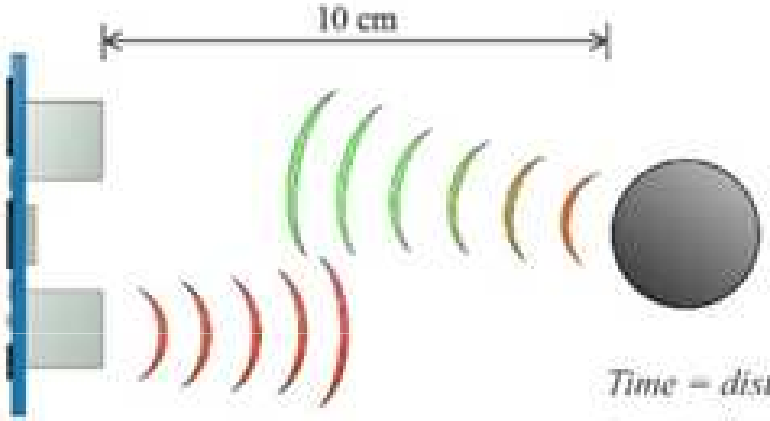


ARDUINO

Interfacing Ultrasonic Sensor WITH NodeMcu

Trig pin→ GPIO5(D1)
Echo pin → GPIO4 (D2)
VCC → Vin
Gnd → Gnd





speed of sound:

$v = 340\ m/s$

$v = 0{,}034\ cm/\mu s$

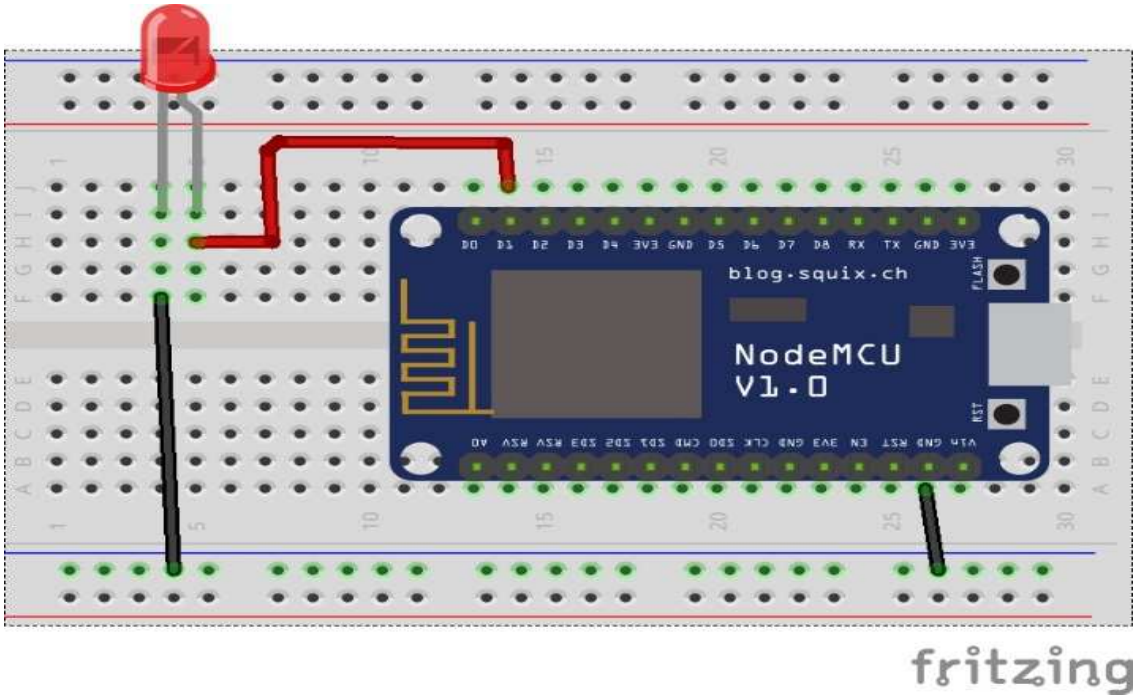
Time = distance / speed:

$t = s / v = 10 / 0{,}034 = 294\ \mu s$

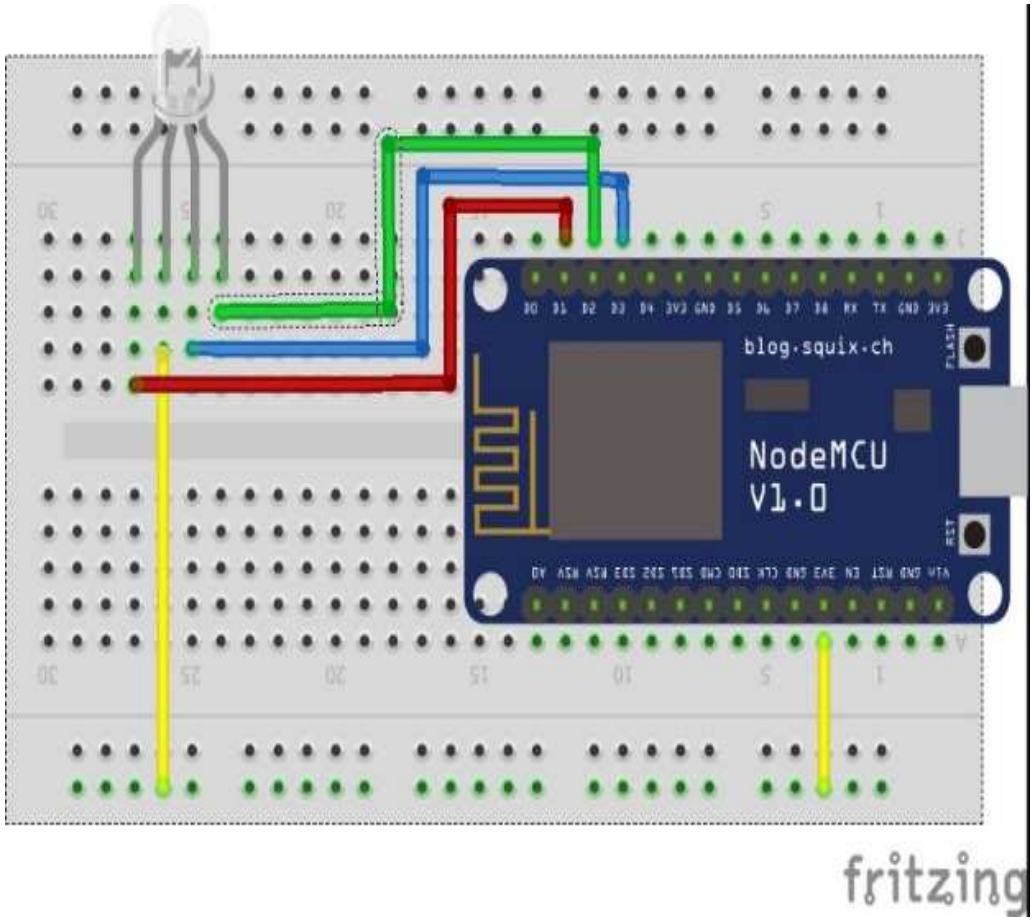
Distance:

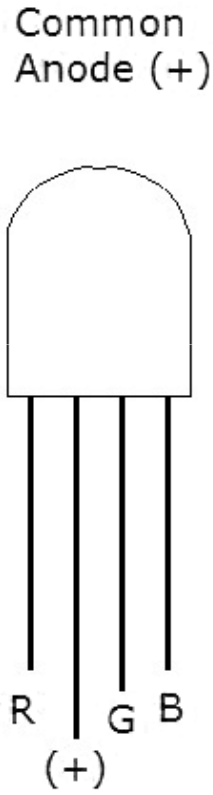
$s = t \cdot 0{,}034 / 2$

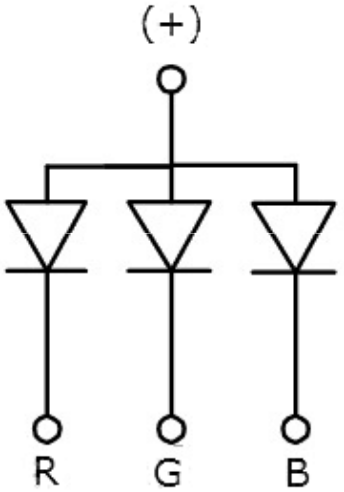
LED Blinking Using Web Server Programme



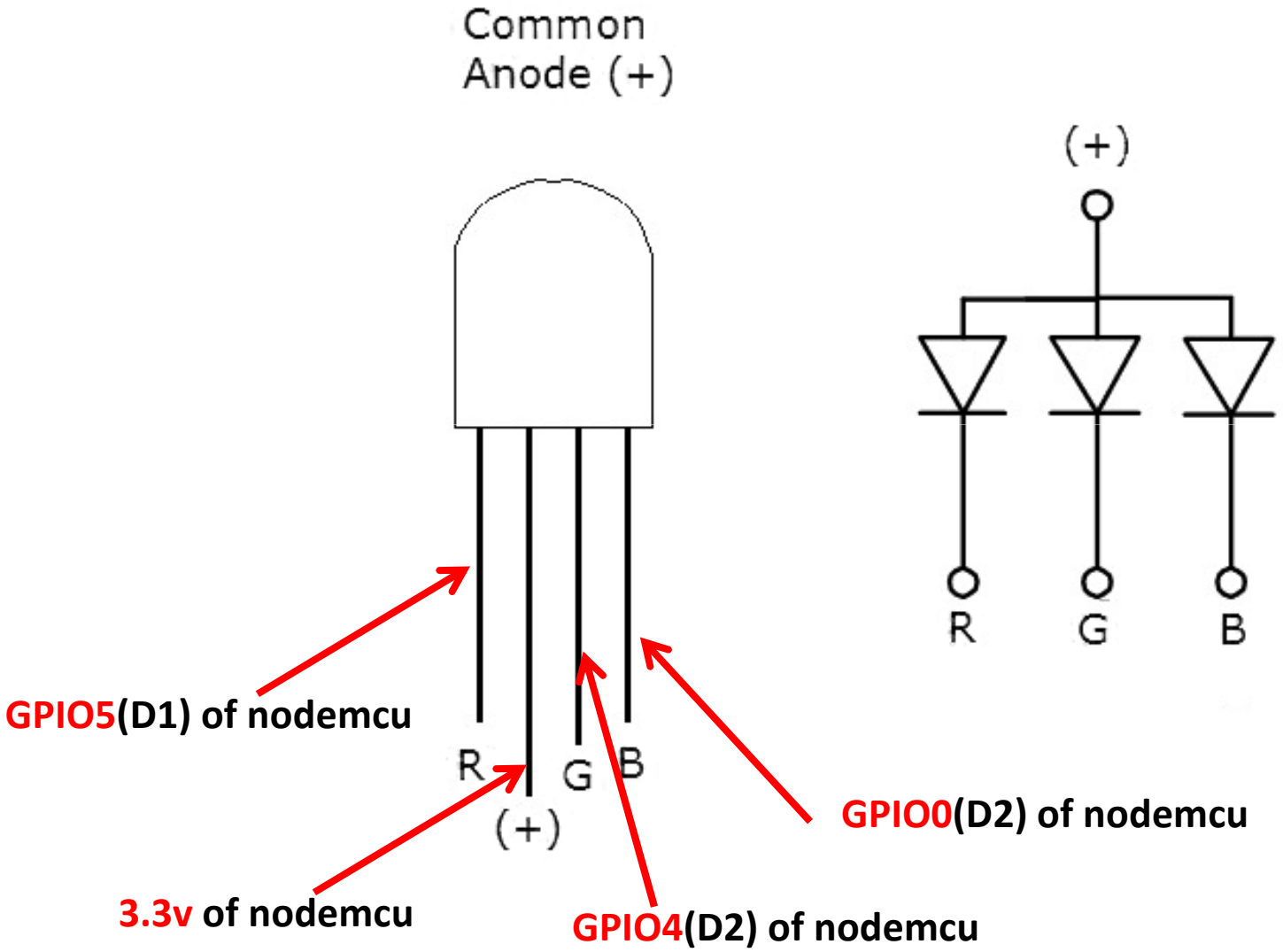
RGB led control using webserver





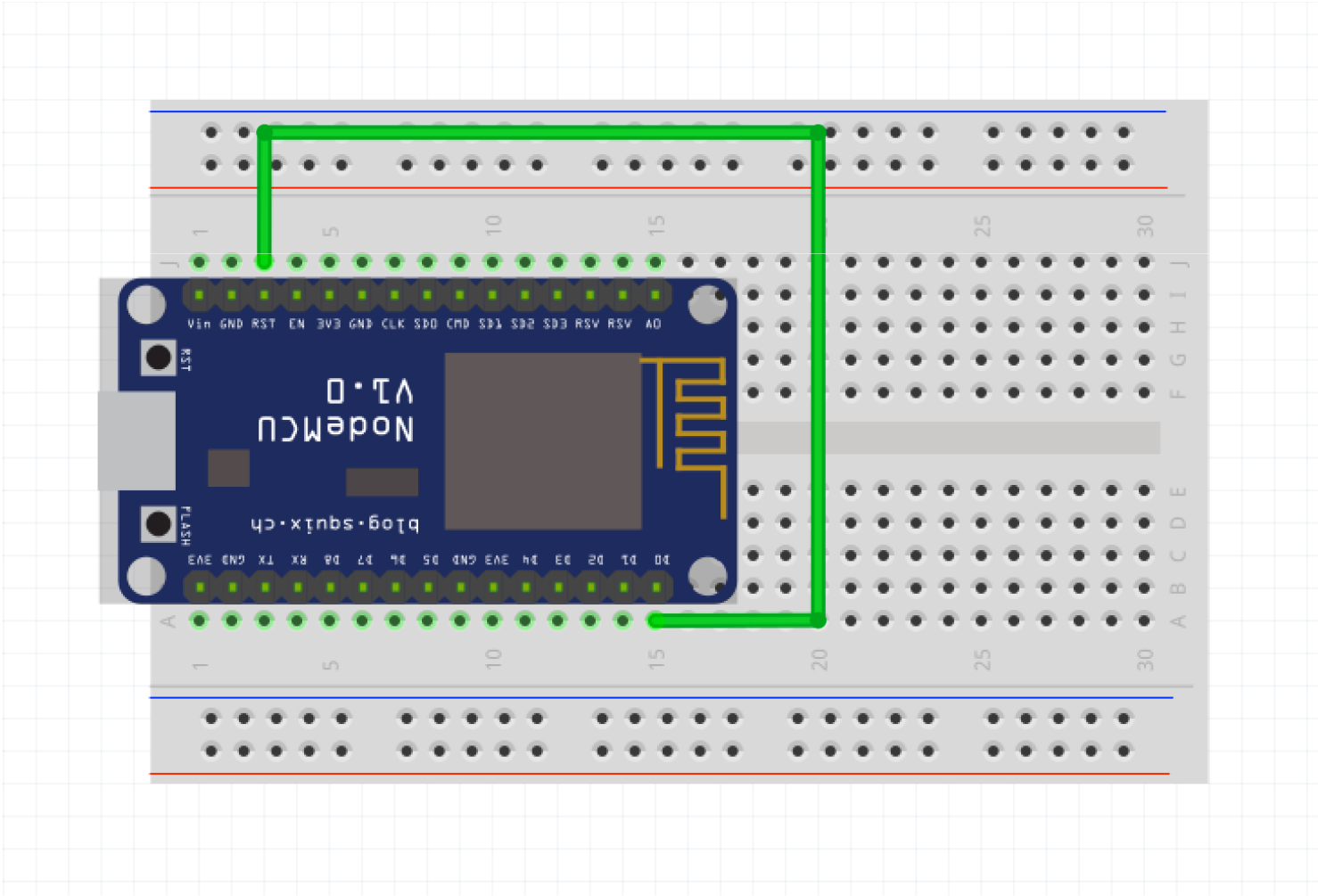


RGB led Pinouts



Deep Sleep Mode

Short Reset(RST) and D1(GPIO16) of nodemcu



If you're not prepared to be wrong, you'll never come up with anything original.

—Sir Ken Robinson