

Introduction

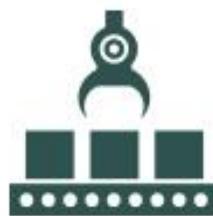
4th Industrial Revolution



The 1st Industrial Revolution

18th Century

Steam engine
based
mechanization
revolution



The 2nd Industrial Revolution

Early 19-20th Century

Electricity based
mass production
revolution



The 3rd Industrial Revolution

Latter Half of the 20th Century

Computer/Internet
based knowledge
and information

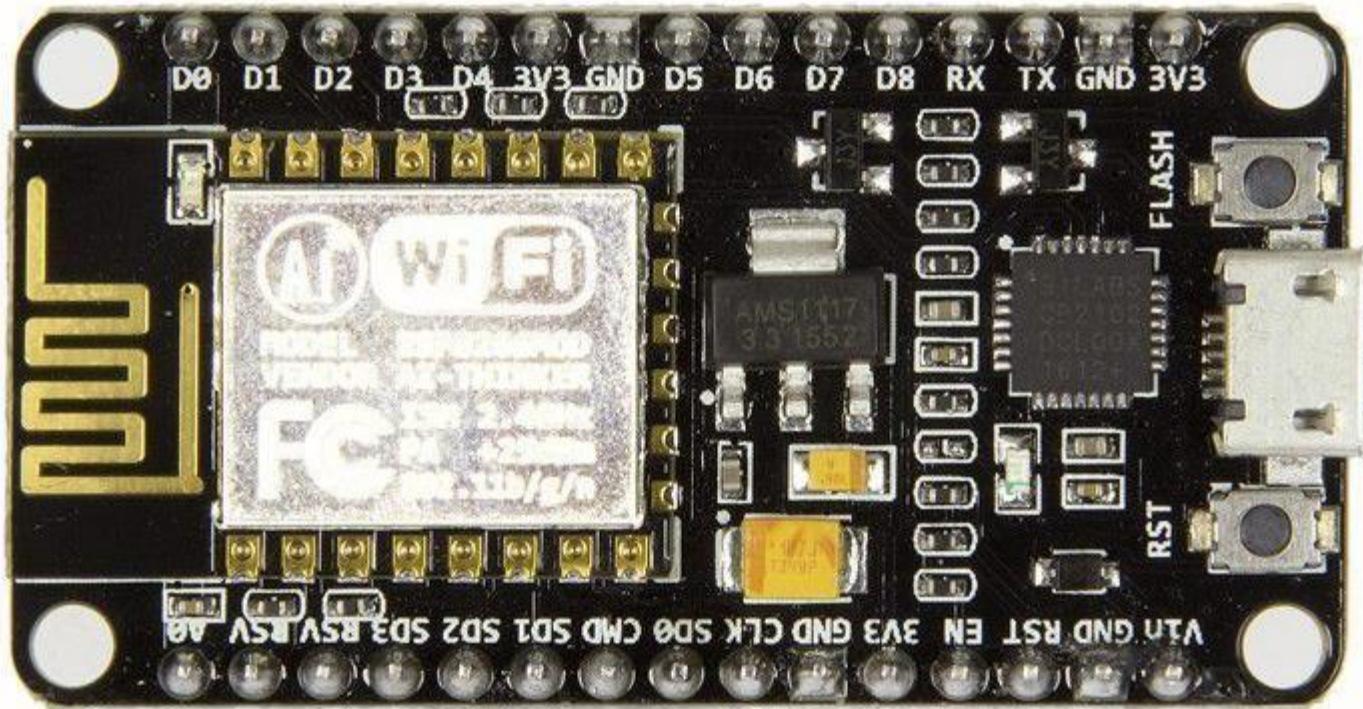


The 4th Industrial Revolution

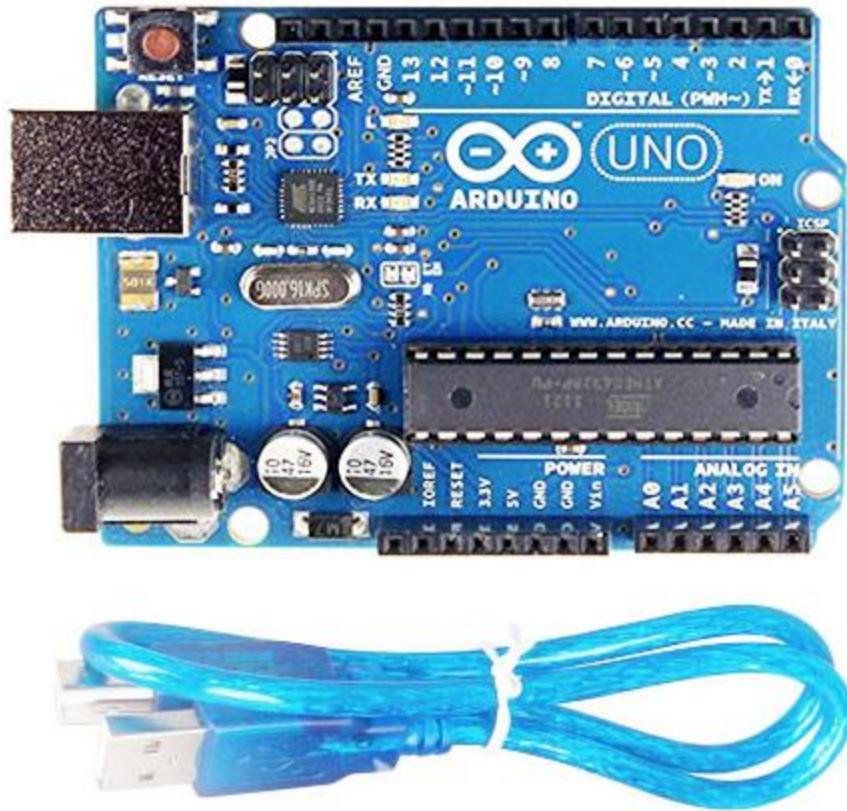
Early 21st Century -

Big Data/AI/IoT
based
hyperconnectivity
revolution

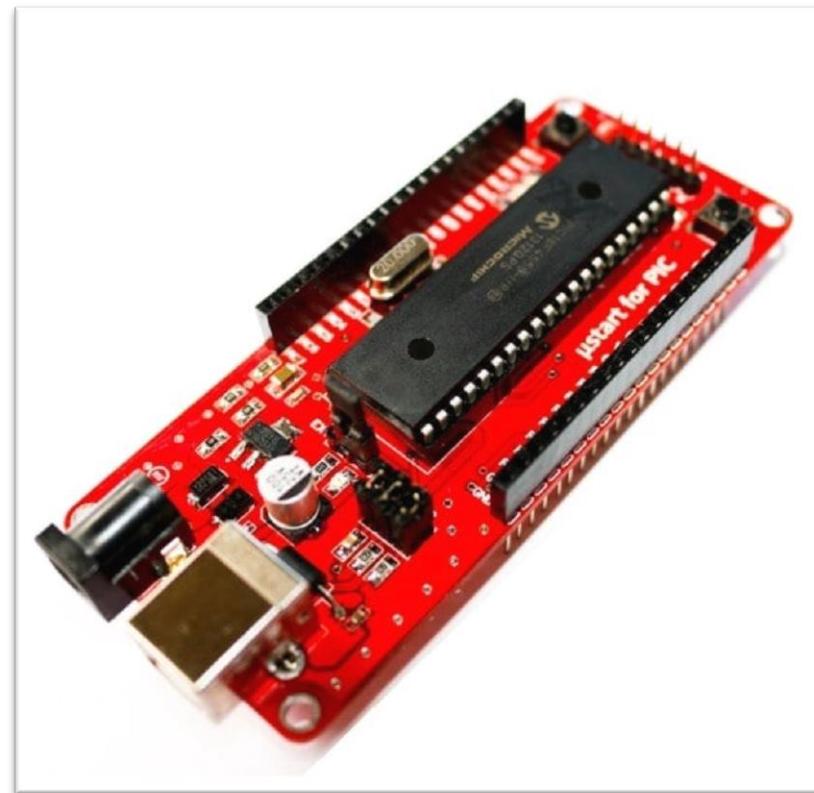
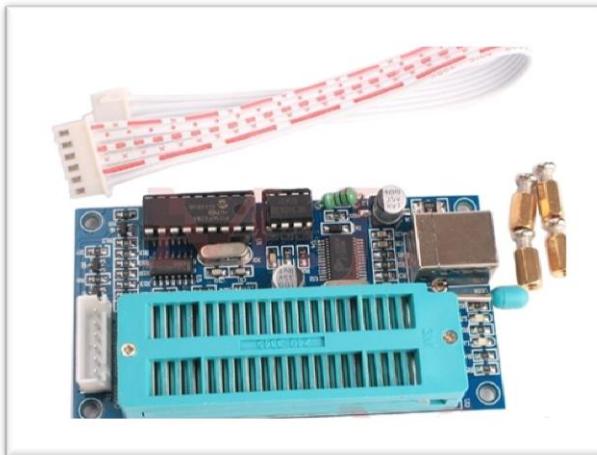
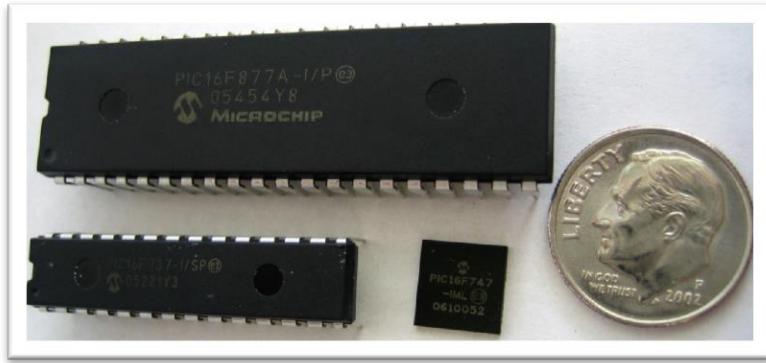
Mini Computer



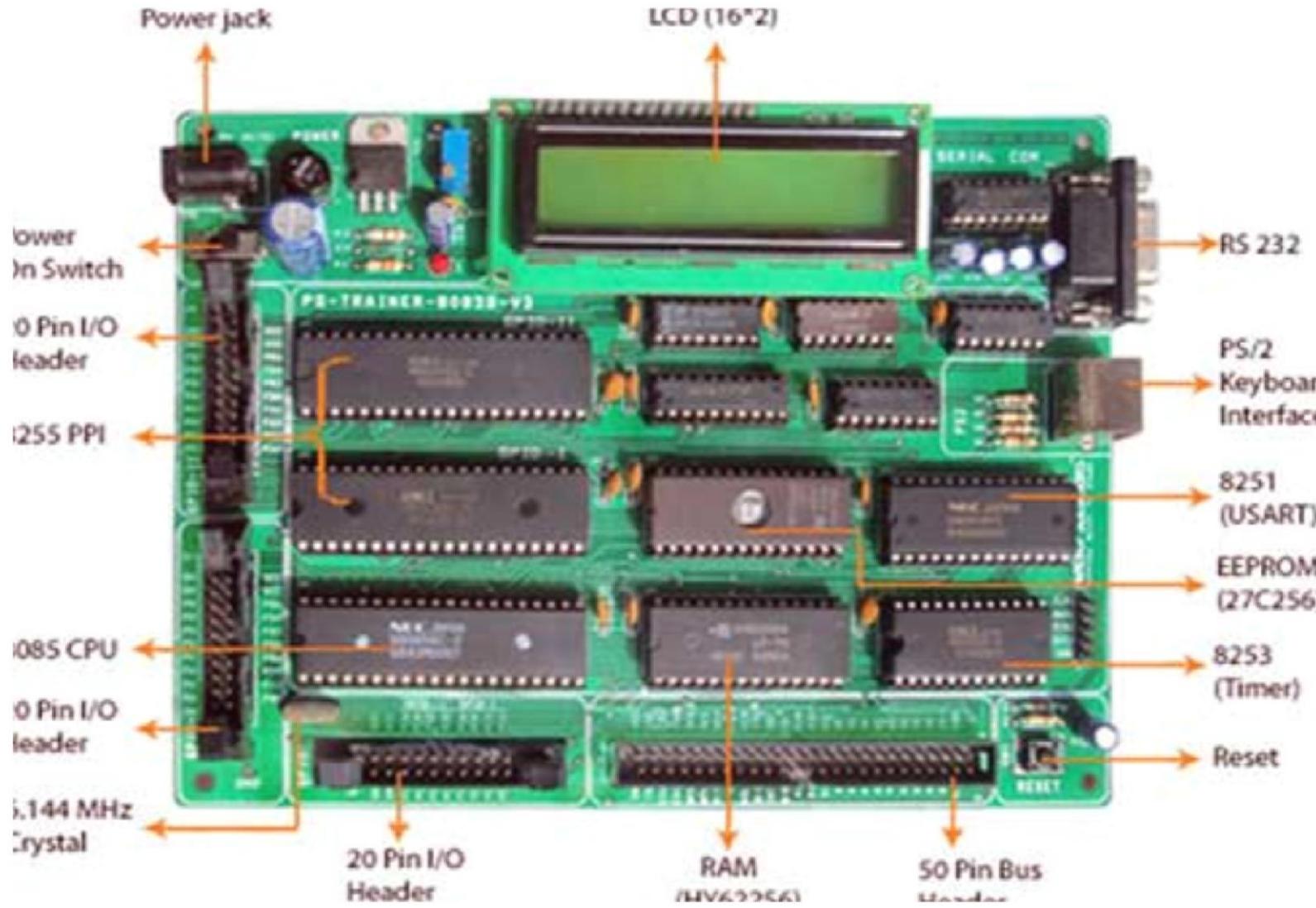
Mini Computer



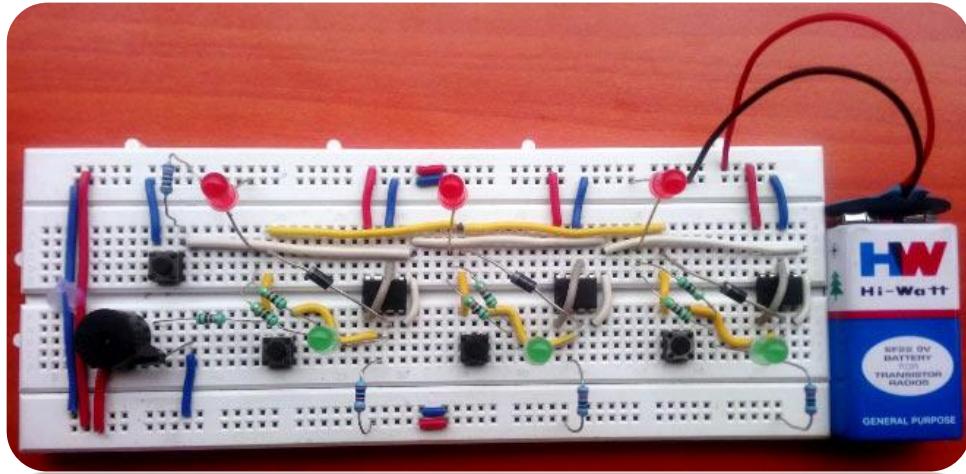
Microcontrollers



Micro Processor

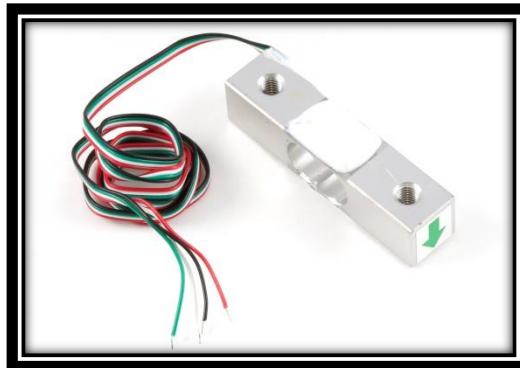
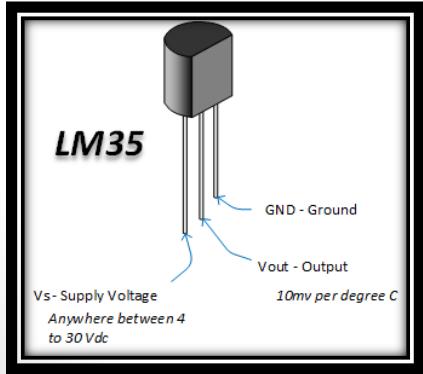
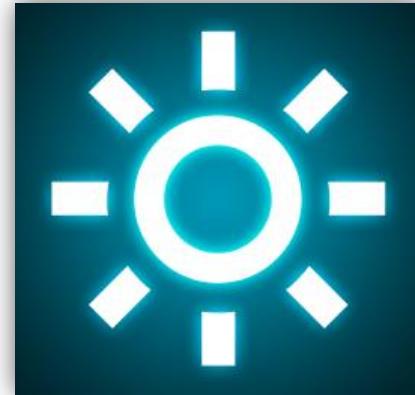


Introduction to Arduino



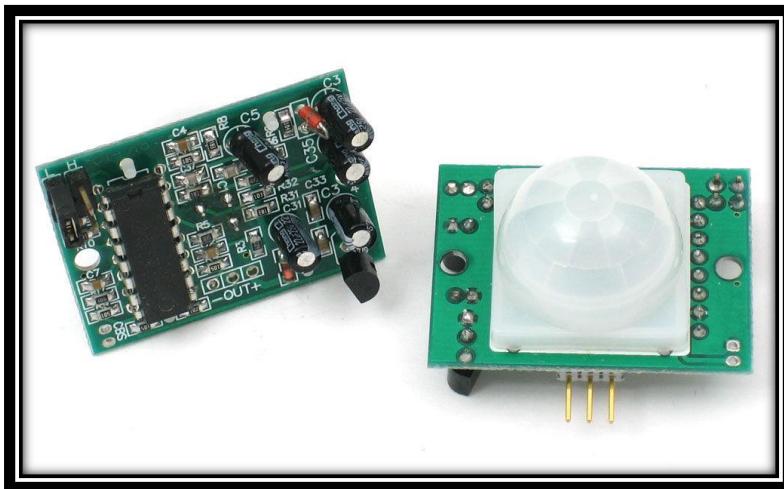
What are Things

- What Are Things ?
 - Sensors



What are Things

- What Are Things ?
 - Sensors



What are Things

- What Are Things ?
 - Other sensors : [youtube video](#)

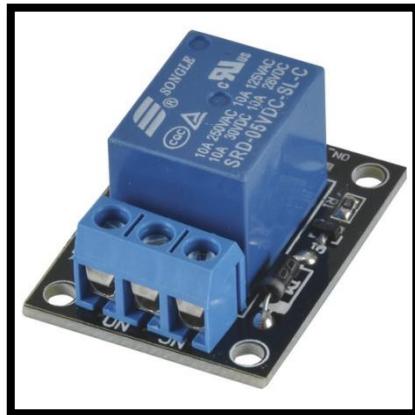
What are Things

- What Are Things ?
 - Sensors
 - Actuators



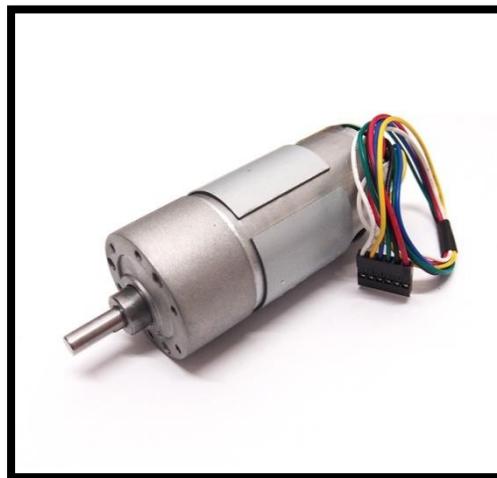
What are Things

- What Are Things ?
 - Sensors
 - Actuators



What are Things

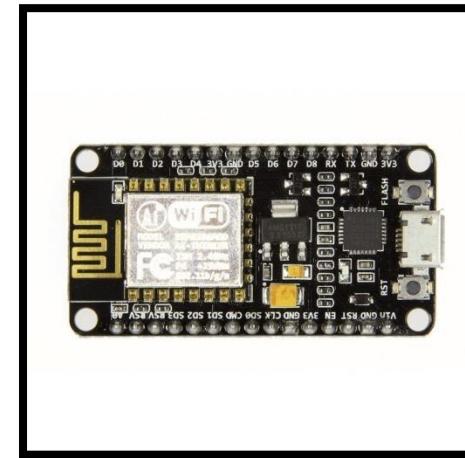
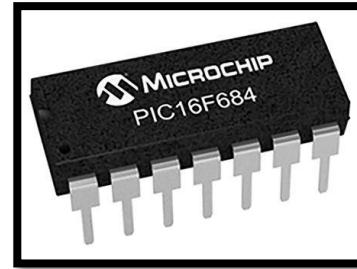
- What Are Things ?
 - Sensors
 - Actuators



What are Things

- What Are Things ?
 - Sensors
 - Actuators
 - Electronic parts

Embedded with Microcontroller, Which is a Mini Computer.



What Is ... ?

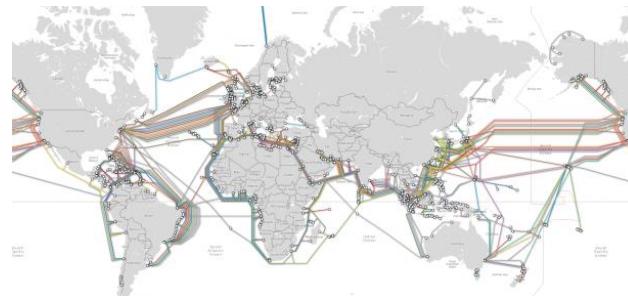
Internet of Things

- What is Internet ?

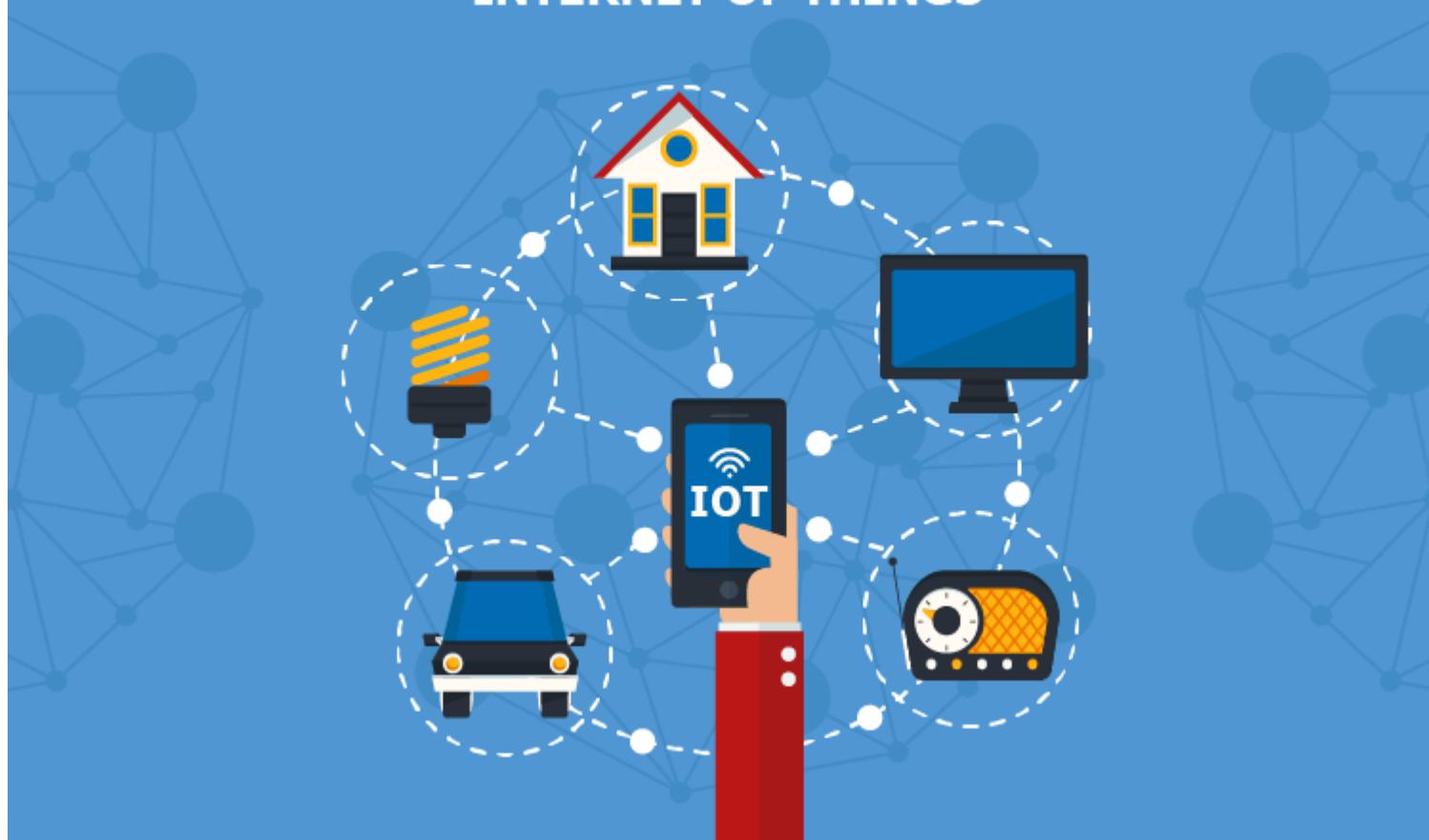


- Is a worldwide system of computer networks - a network of networks in which users at any one computer can, get information from any other computer (and sometimes talk directly to users at other computers).

[Video : How the internet works](#)

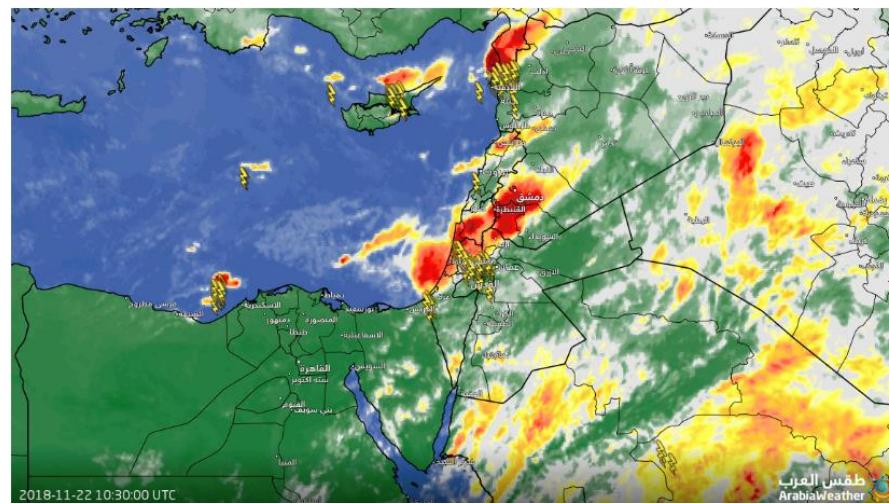


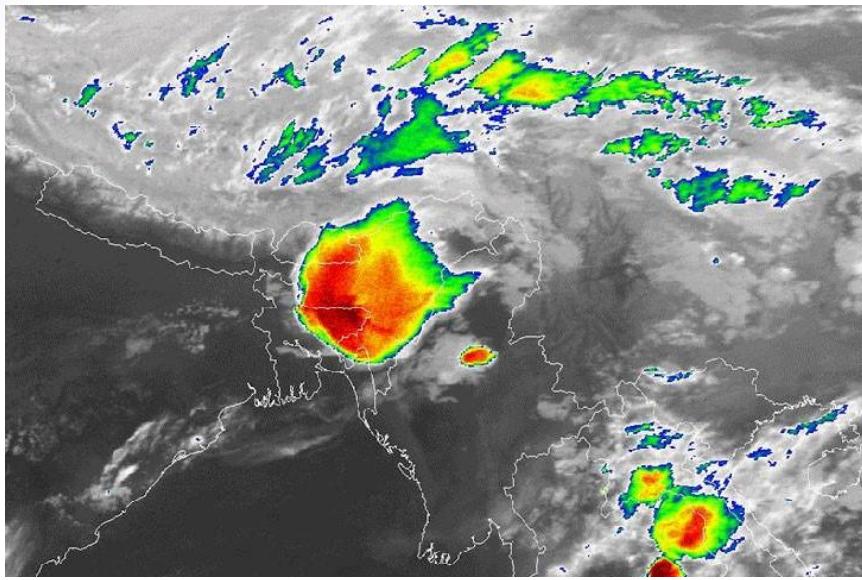
INTERNET OF THINGS



Real Example of IoT

- Arab Weather





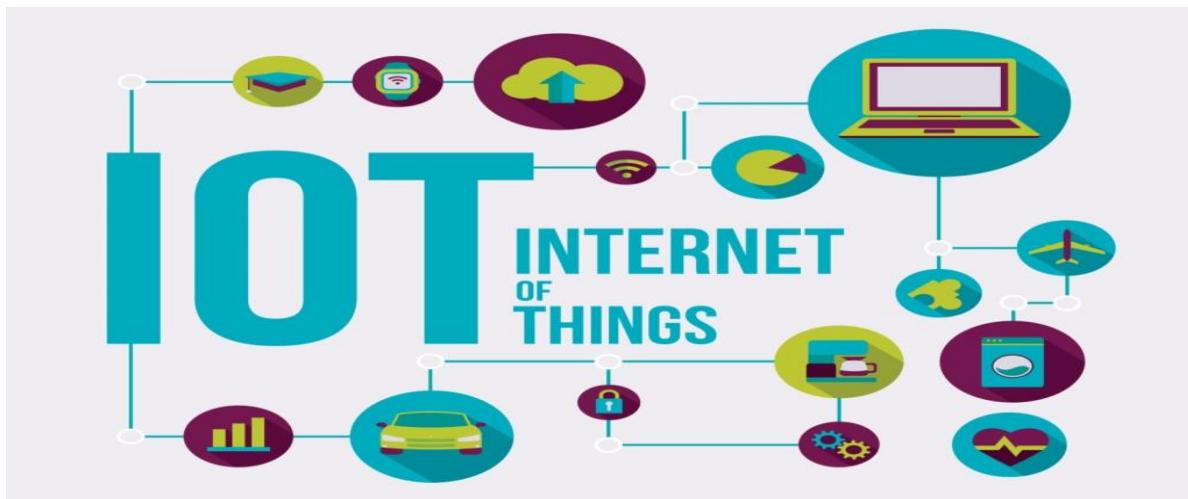
Weather Satellite

Weather Station



Introduction to IoT

- Internet of Things is the network of devices, vehicles, and home appliances that contain electronics, software, actuators, and connectivity which allows these things to connect, interact and exchange data.



Examples of IoT



Wearable Tech



Healthcare



Smart Appliances

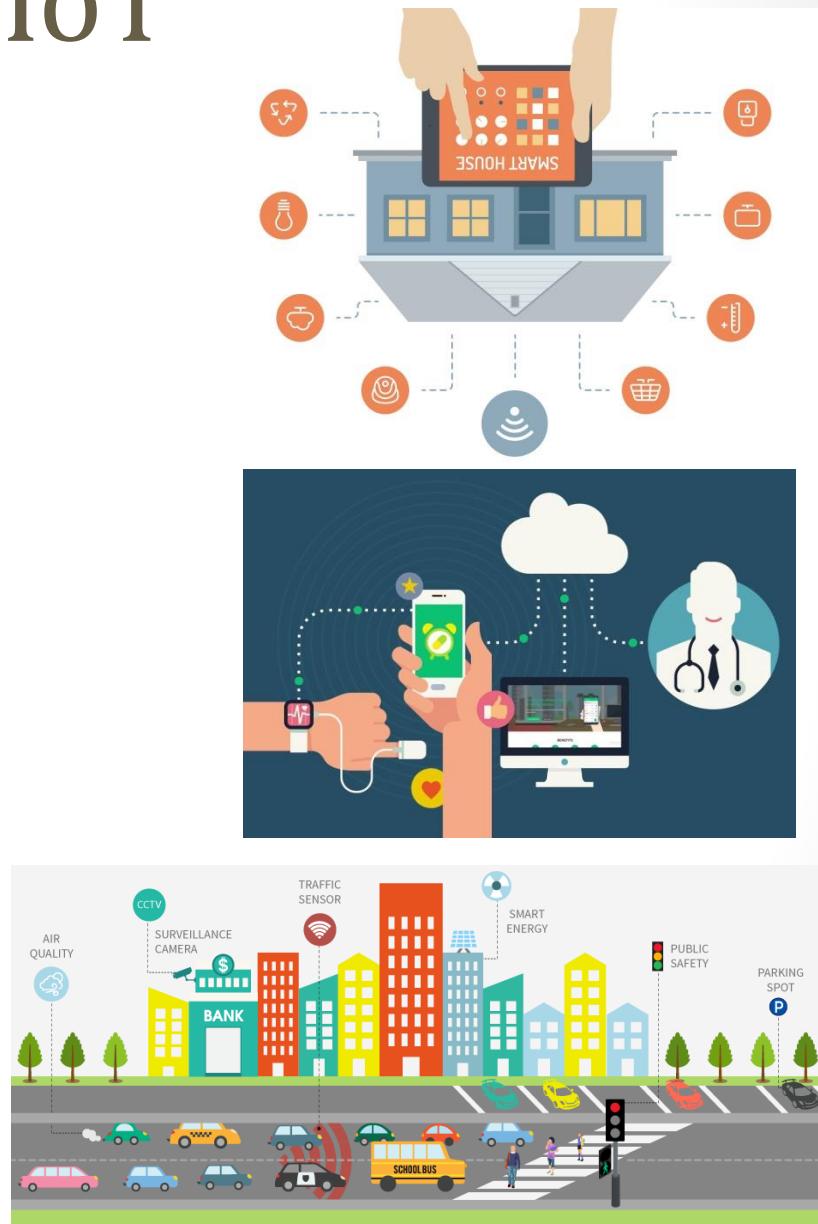
Examples of IoT



Smart Home

Introduction to IoT

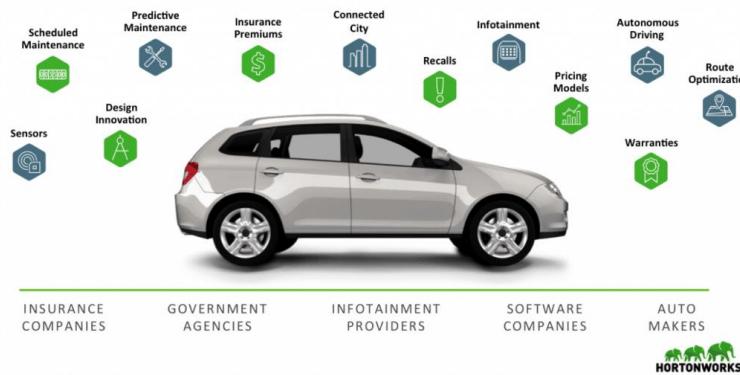
- Home
 - Automation
 - Monitoring
- Health and fitness
 - Patients
 - Players
- City
 - Traffic
 - Security



Introduction to IoT

- Connected cars
 - Insurance companies
 - Repair centers
 - Car tracking

Data Drives the Connected Car

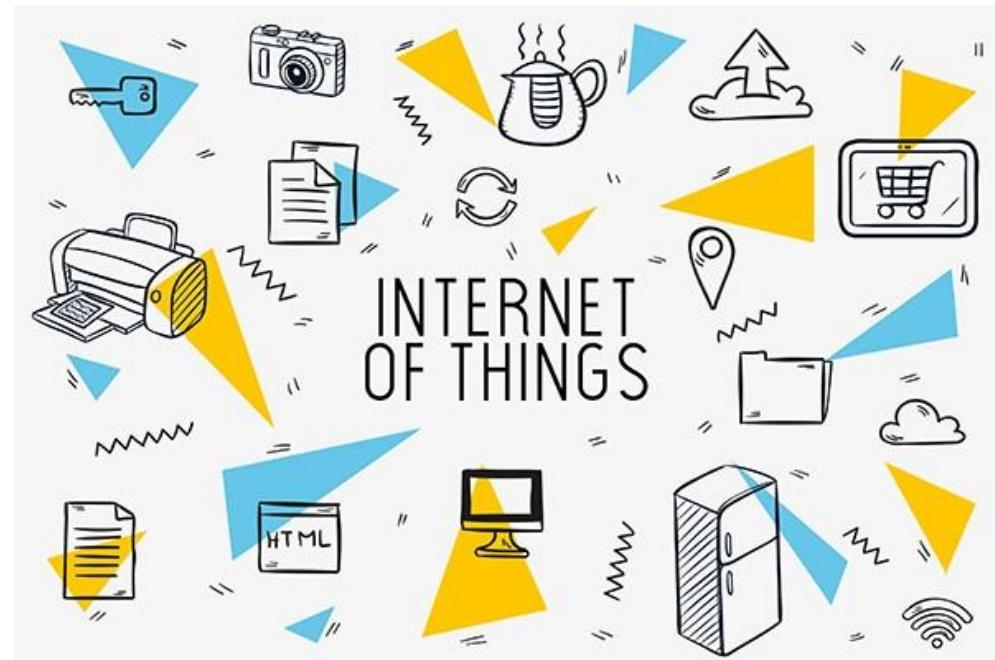


- Environment
 - Global warming
 - Natural disasters
 - Pollution



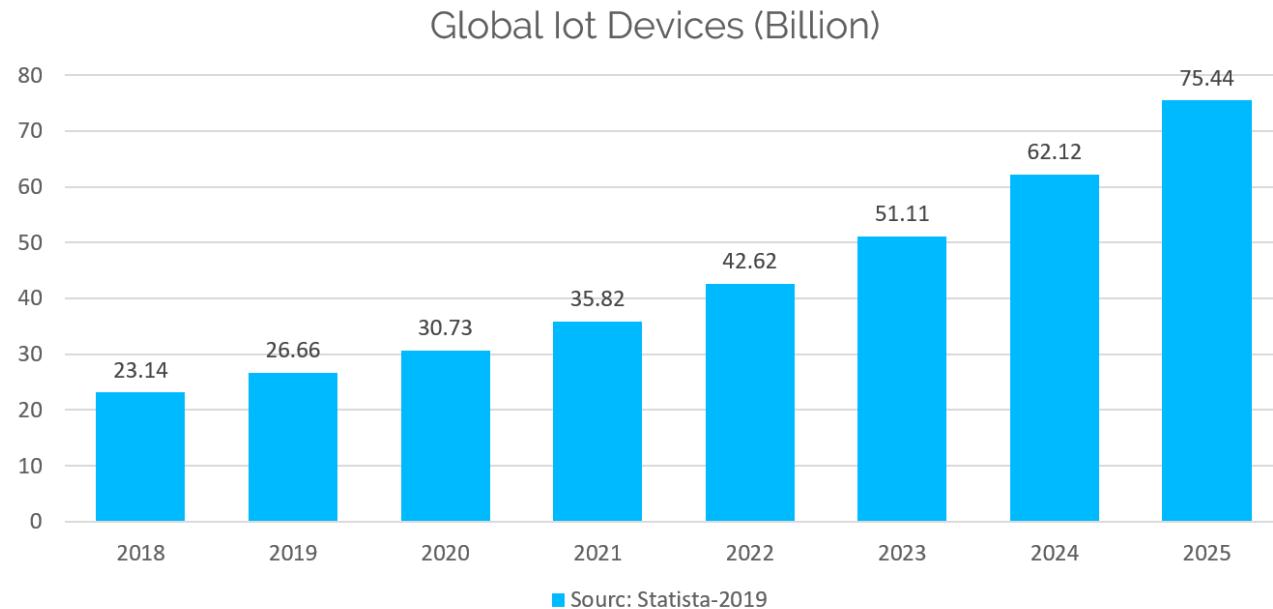
Introduction to IoT

- Home
- Health and fitness
- Business
- City
- Connected Cars
- Environment



Info about IoT field

- Future and Growth of IoT.



By 2020, more than
65% of enterprises
(up from 30% today)
will adopt IoT products.

IoT Overview

User Interface (UI):

- Web development
- Mobile Application

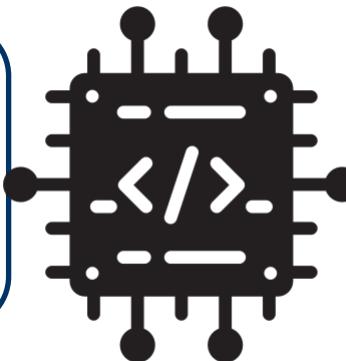
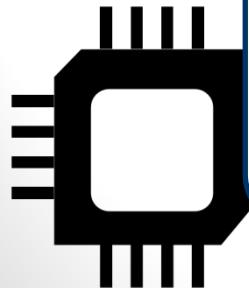


Networking

- LAN
- Internet



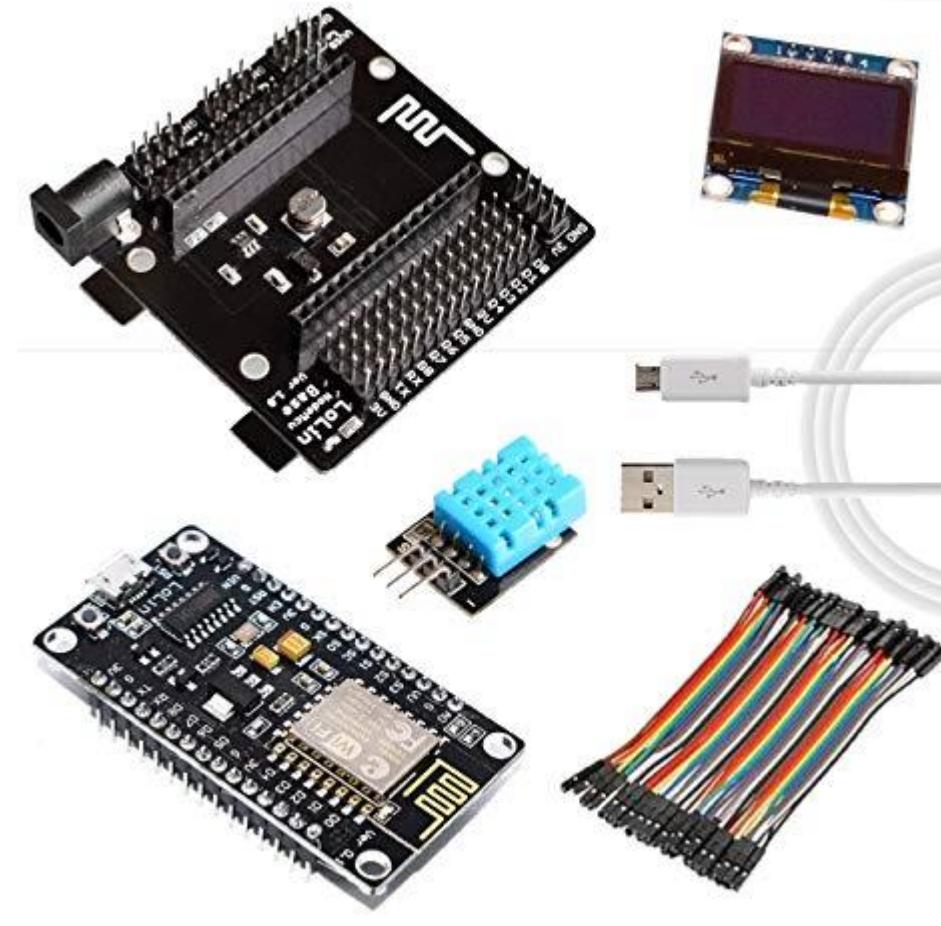
Electronics
Sensors
Actuators



Programming
Skills and tools



Let's go !



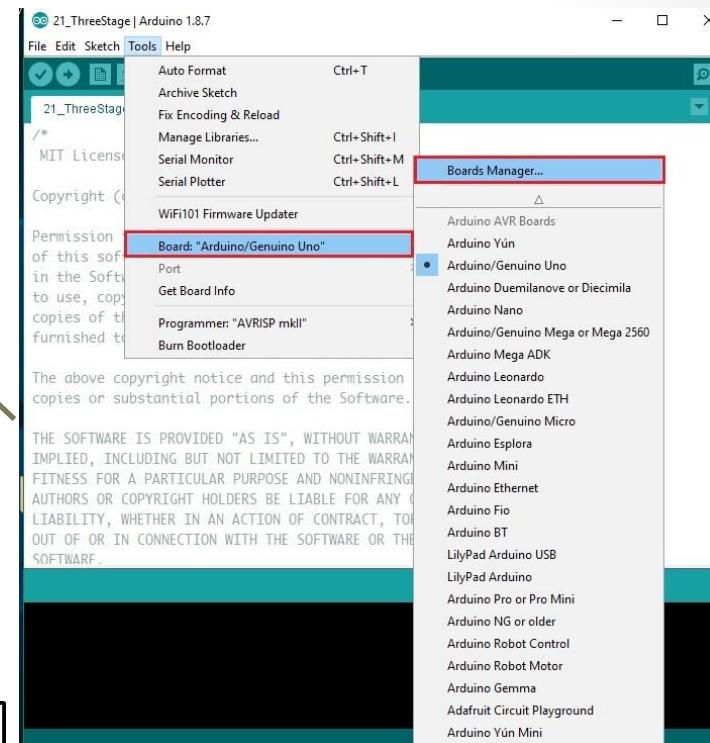
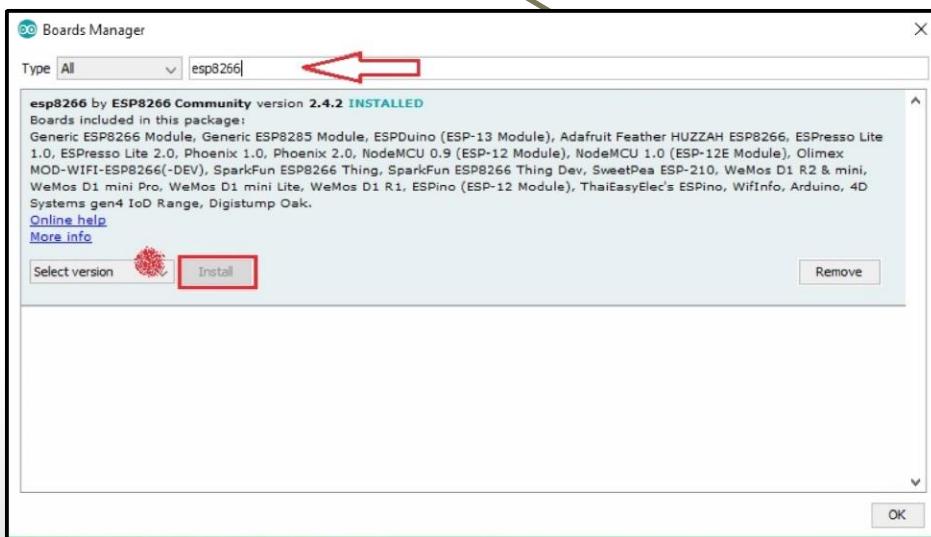
Installation

- Arduino IDE



Installation

- Arduino IDE
- ESP8266 core



Beginning with ESP8266

- **Install board :**
 - Add link to Preferences → Additional Boards
 - Go to Tools → Board → Boards Manager and install esp8266
- **Blink :**
 - Open File → Examples -> Blink
 - Select “NodeMCU 1.0” from Tools → Board:
 - Hit Upload.

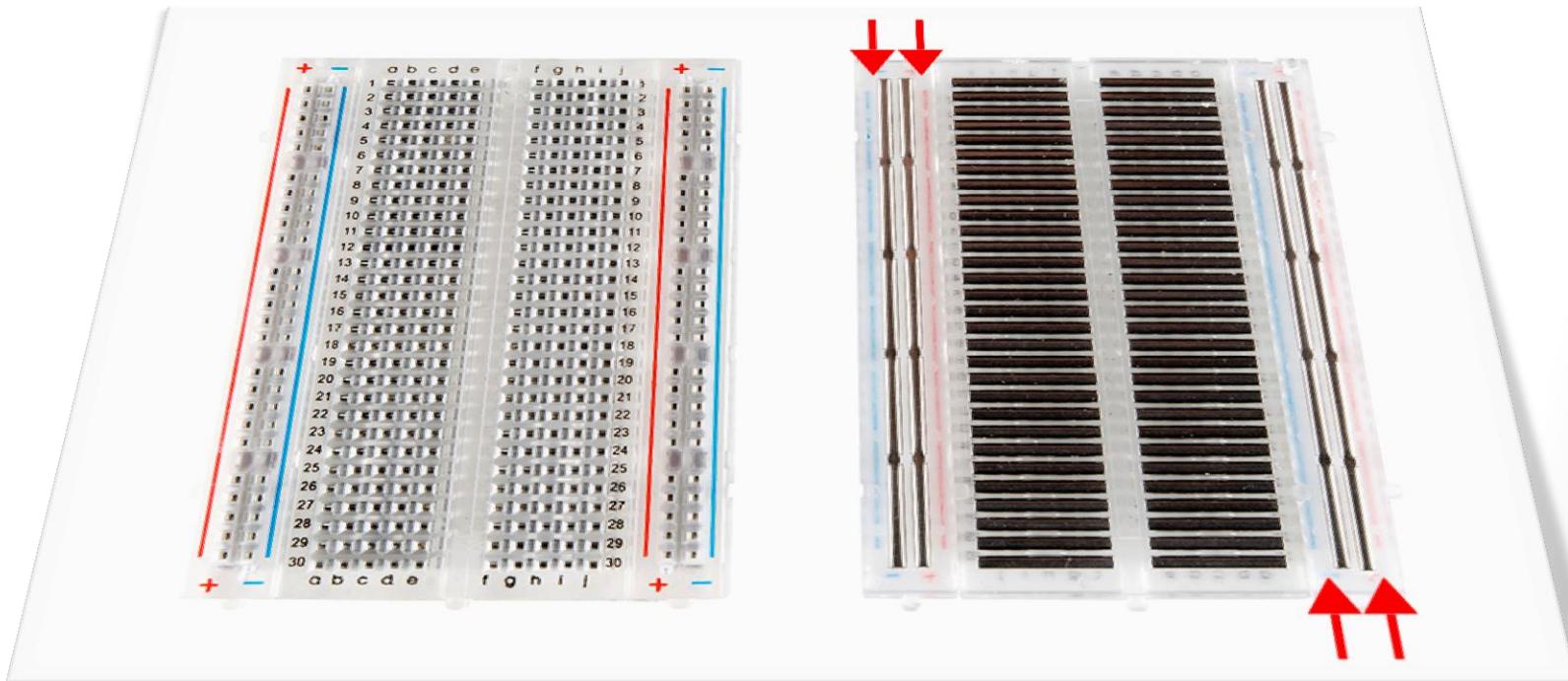
Start

- Blinking LED.



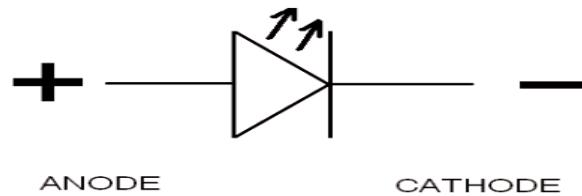
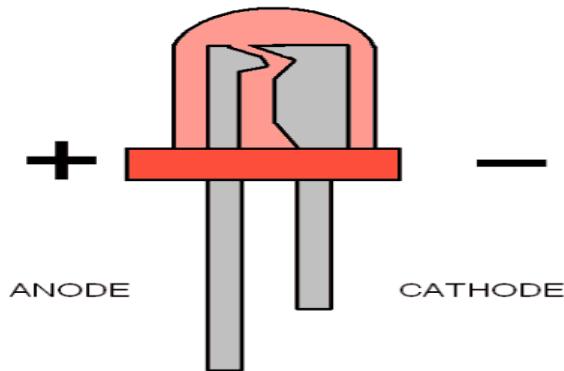
Electronics and Wiring

- Bread board



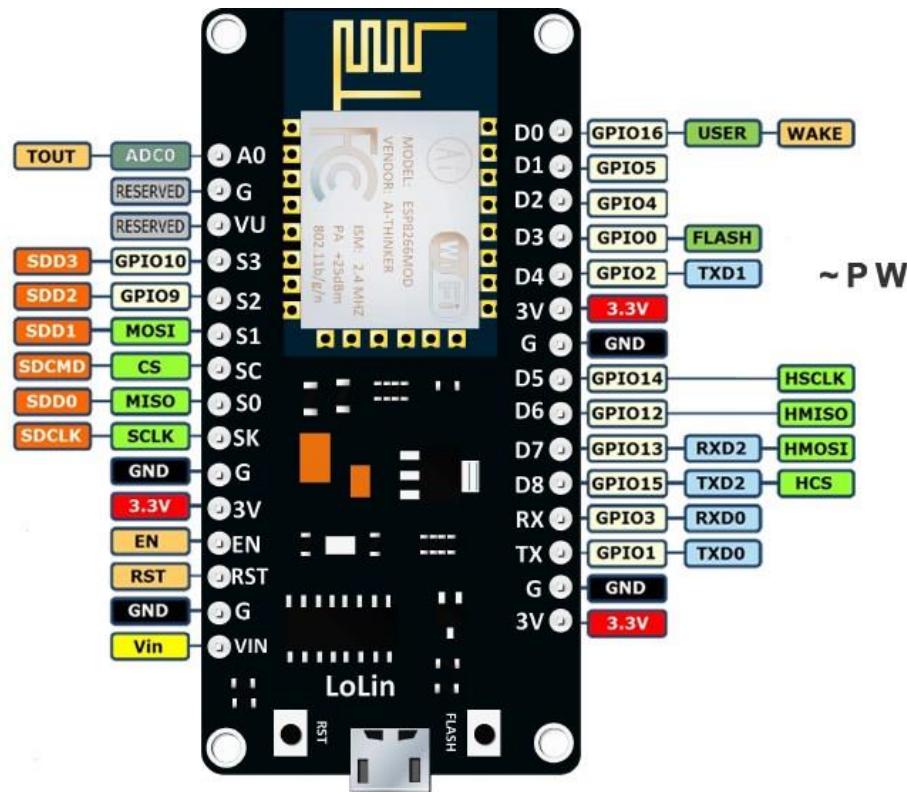
Electronics and Wiring

- Bread board
- LEDs



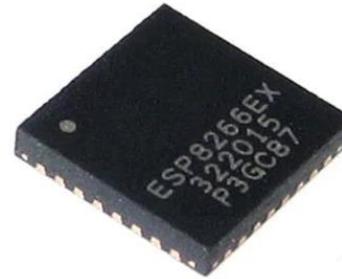
Electronics and Wiring

- Bread board
- LEDs
- ESP8266 NodeMCU pin map

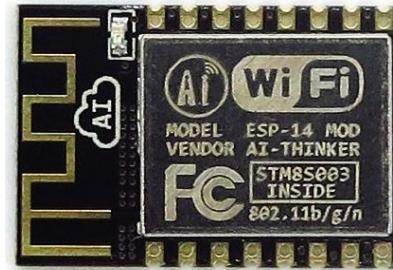


About ESP8266

- Chip
- Module
 - Flash memory
 - Antenna
- **ESP-01**
- **NodeMCU**
 - 3.3 V regulator
 - USB to UART



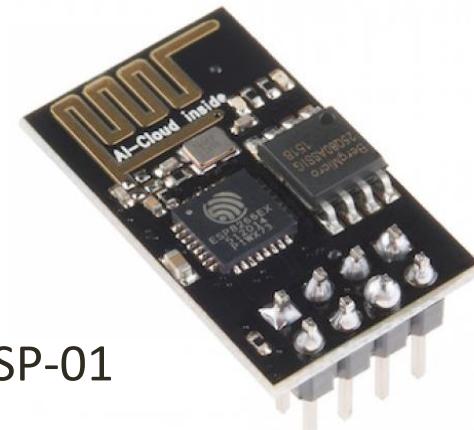
ESP8266ex



ESP-12

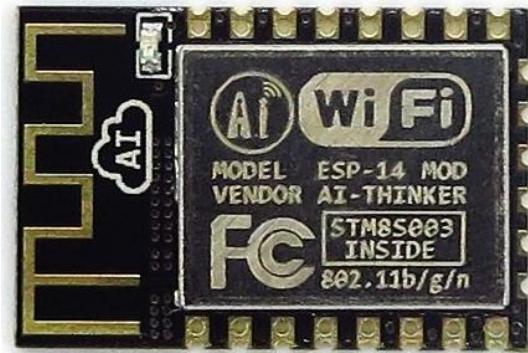
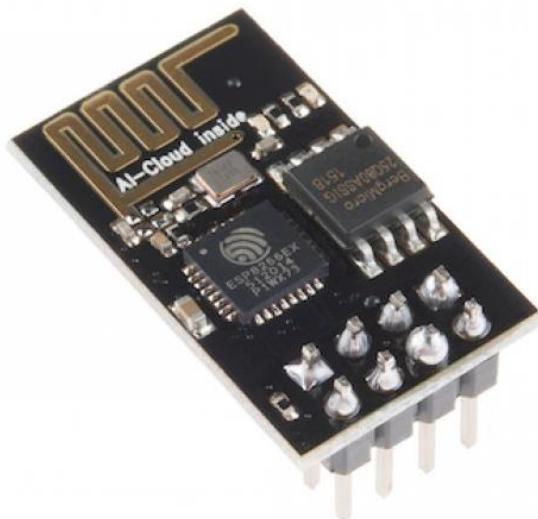


NodeMCU



ESP-01

On a scale

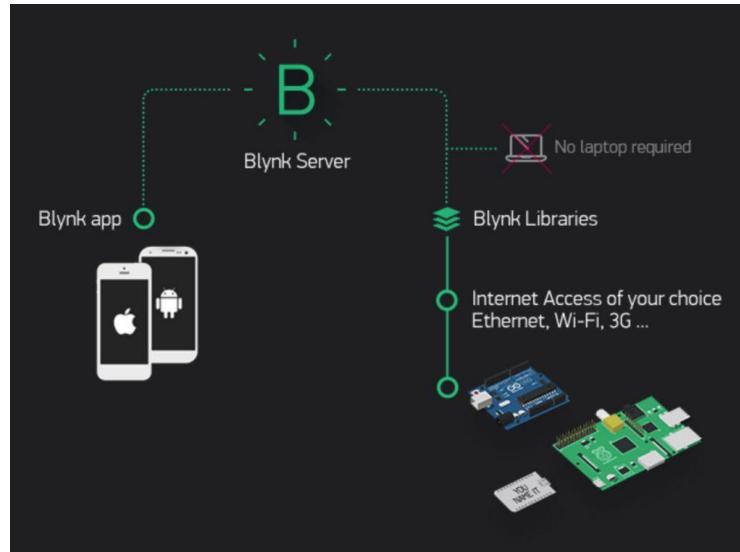


Comparison :

Specification \ MCU	Arduino UNO	ESP8266 NodeMCU
Connectivity	None	WiFi
Clock Frequency	16 MHz	80/160 MHz
Flash Memory	32 KB	4 MB
RAM	2 KB	64 SRAM / 96 KB DRAM
GPIO (Available)	14 (14)	16 (10)
Analog Inputs	6	1
PWM	6	9
EEPROM	1KB	None
Power Supply	5 V to 12 V	3.3 V to 20 V
Current Consumption (max)	40 mA	70 mA

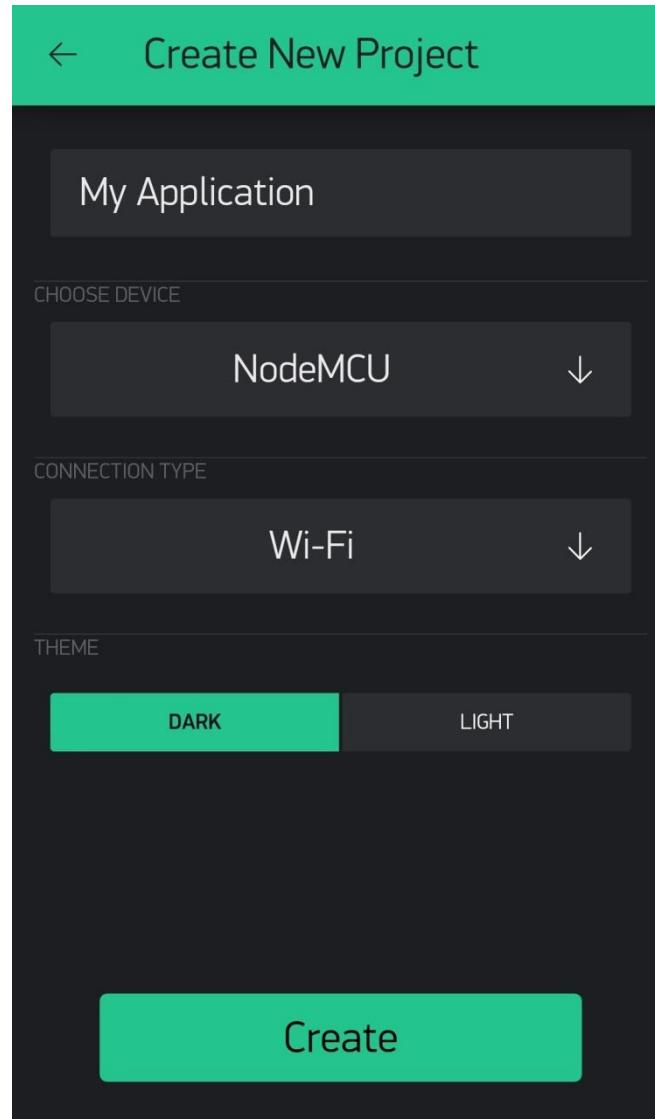
Blynk App

- **Blynk** is an Internet of Things platform with a drag-n-drop mobile application builder that allows to visualize sensor data and control electronics remotely.
- **Platforms**
 - **platform** is a group of technologies that are used as a base upon which other applications, processes or technologies are developed.



Blynk App

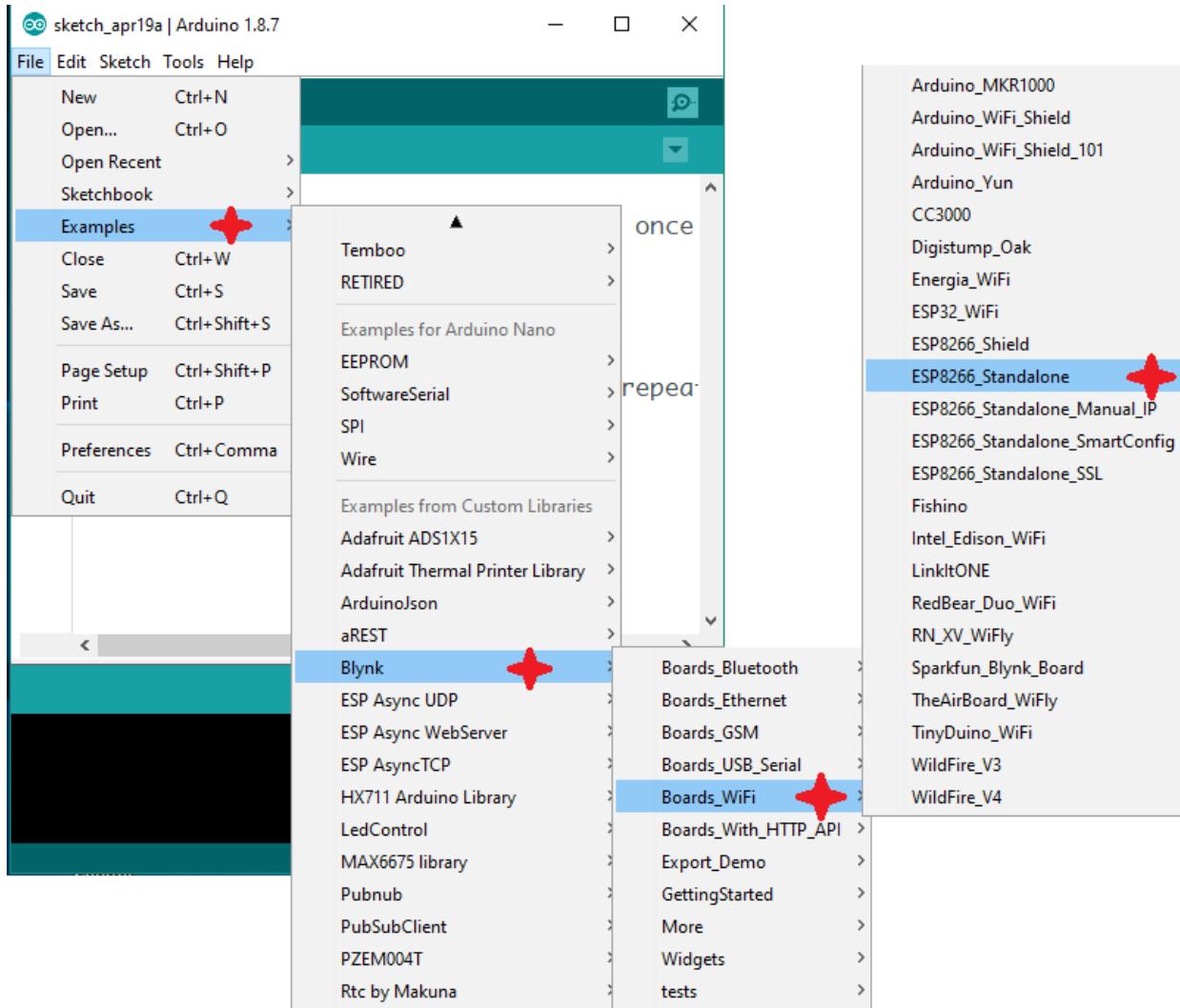
- Install blynk app.
- Instantiate new project



Blynk App

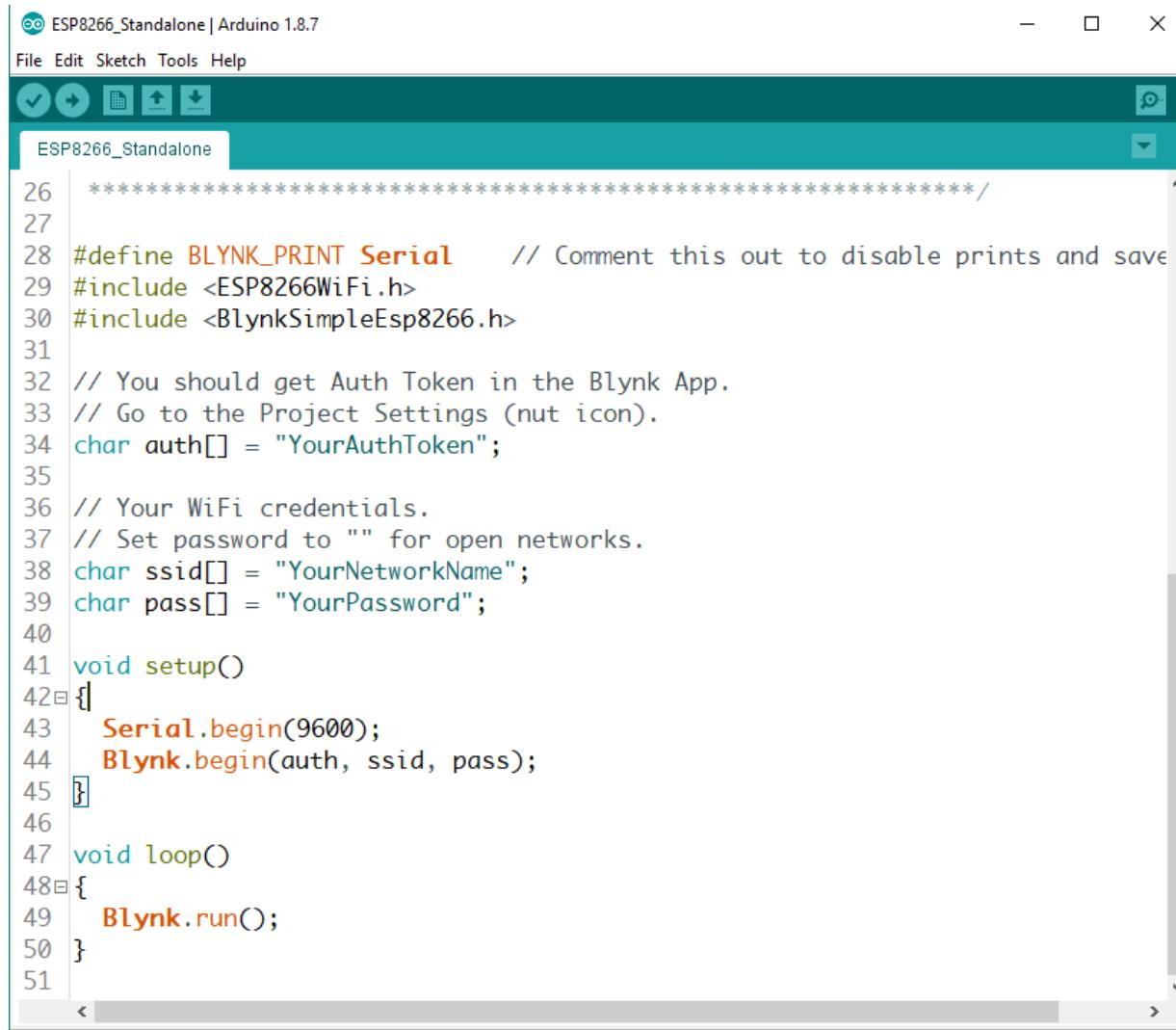
- Install blynk app.
- Instantiate new project
- Copy Auth Token to paste in Arduino example.
- Open Arduino Example

Blynk App



File -> Examples -> Blynk -> Boards_WiFi -> ESP8266_Standalone

Blynk App



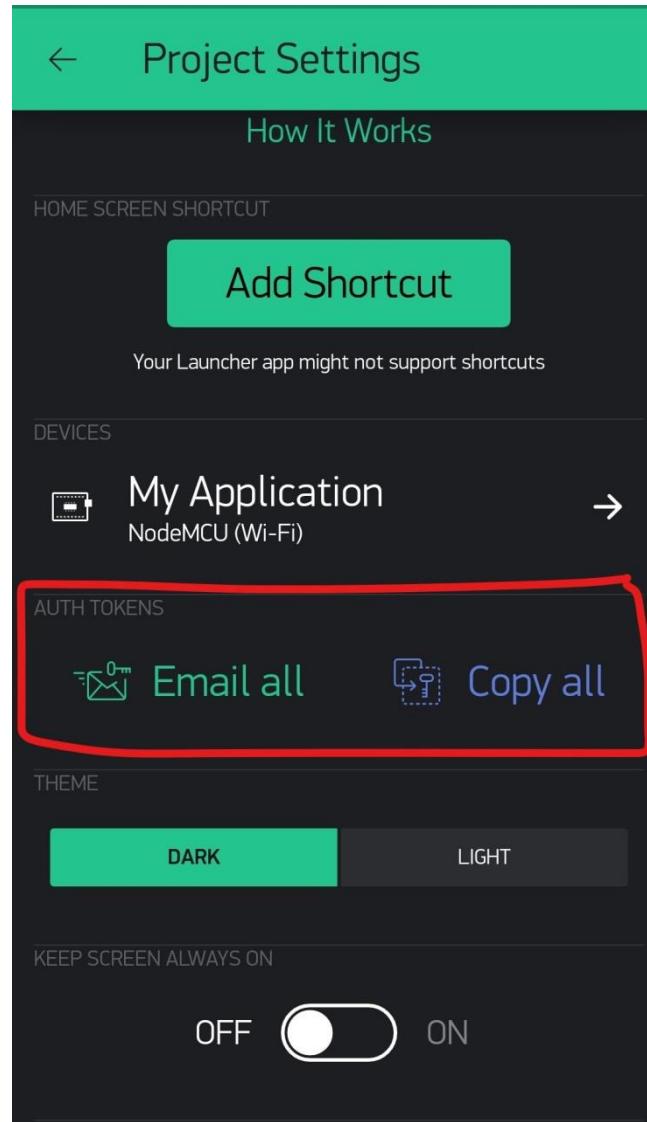
The screenshot shows the Arduino IDE interface with the title bar "ESP8266_Standalone | Arduino 1.8.7". The menu bar includes File, Edit, Sketch, Tools, and Help. Below the menu is a toolbar with icons for back, forward, upload, and download. The main window displays the code for the "ESP8266_Standalone" sketch. The code is a Blynk library example for an ESP8266. It defines a BLYNK_PRINT macro pointing to Serial, includes the ESP8266WiFi.h and BlynkSimpleEsp8266.h headers, and sets up WiFi credentials (ssid and password). The setup() function initializes the serial port at 9600 bps and starts the Blynk connection. The loop() function runs the Blynk run() method.

```
26 // ****
27
28 #define BLYNK_PRINT Serial // Comment this out to disable prints and save power
29 #include <ESP8266WiFi.h>
30 #include <BlynkSimpleEsp8266.h>
31
32 // You should get Auth Token in the Blynk App.
33 // Go to the Project Settings (nut icon).
34 char auth[] = "YourAuthToken";
35
36 // Your WiFi credentials.
37 // Set password to "" for open networks.
38 char ssid[] = "YourNetworkName";
39 char pass[] = "YourPassword";
40
41 void setup()
42 {
43     Serial.begin(9600);
44     Blynk.begin(auth, ssid, pass);
45 }
46
47 void loop()
48 {
49     Blynk.run();
50 }
```

https://github.com/HamzaHajeir/IoT_16h

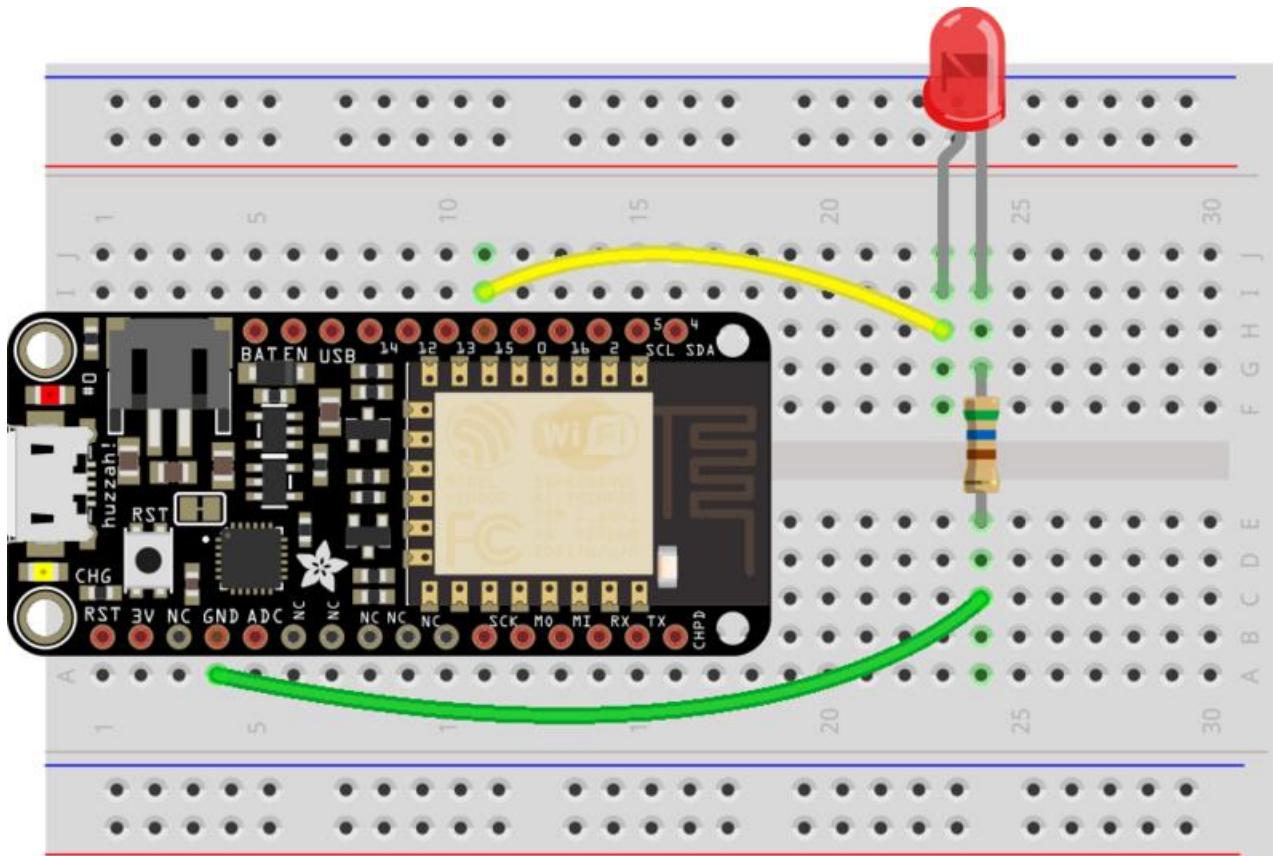
Blynk App

- Install blynk app.
- Instantiate new project
- Copy Auth Token to paste in Arduino example.



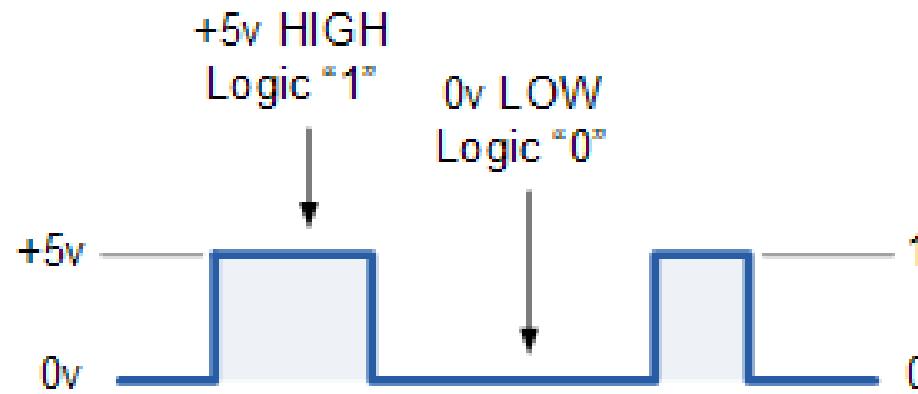
Electronics and Wiring

Circuit wiring

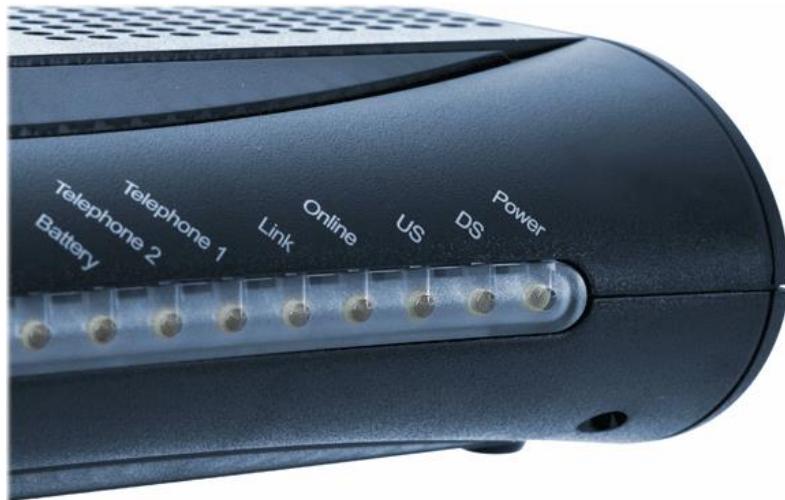


fritzing

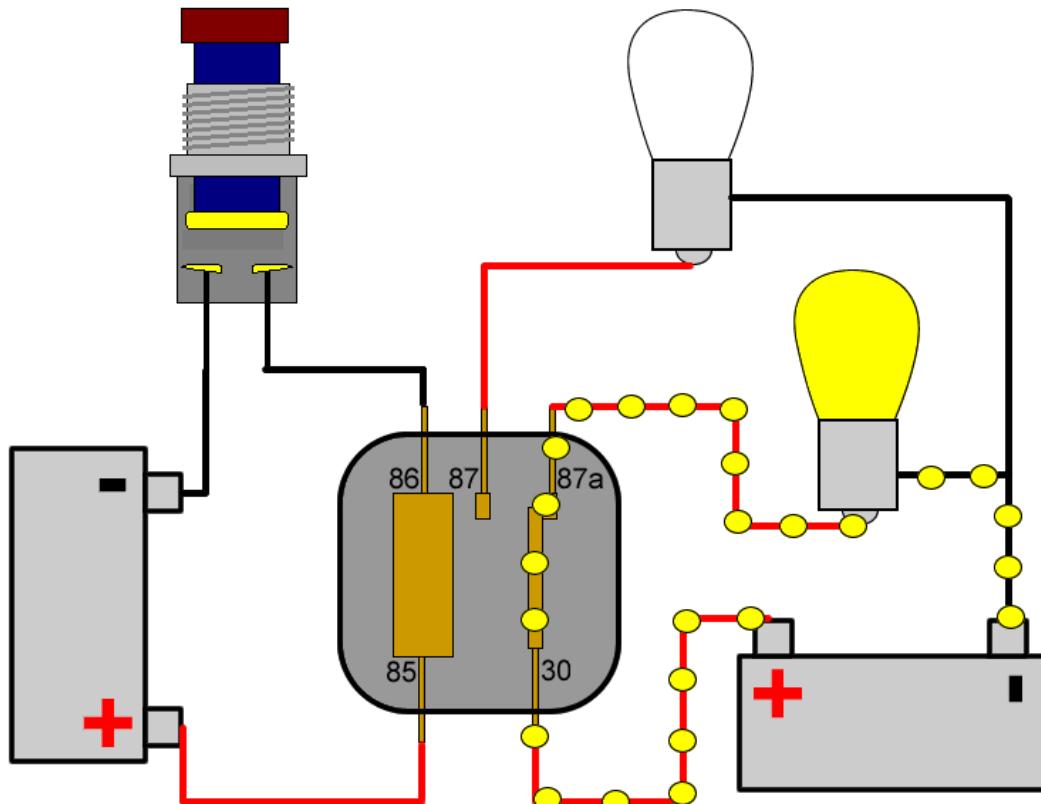
Voltage/Digital relationship



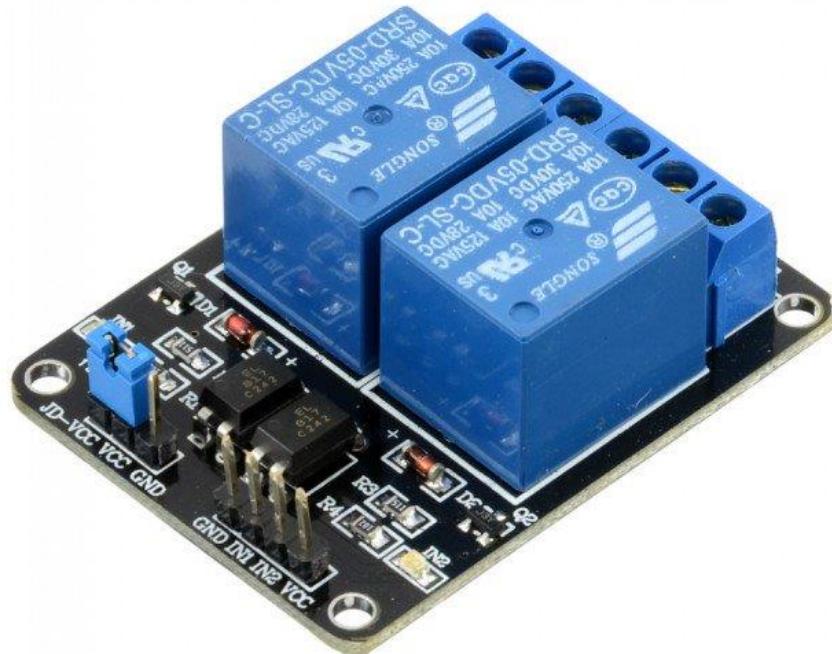
In Real Life :

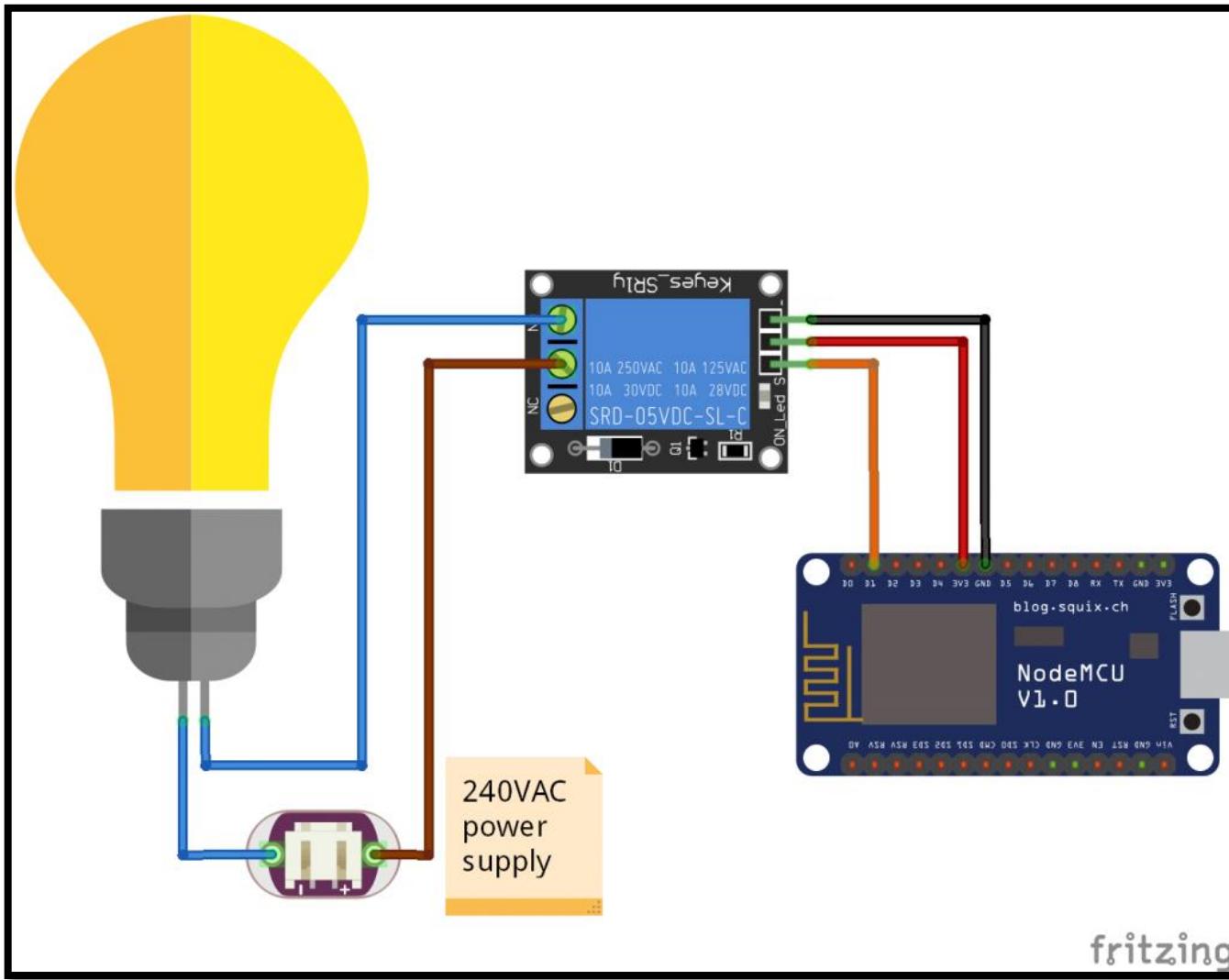


Relay

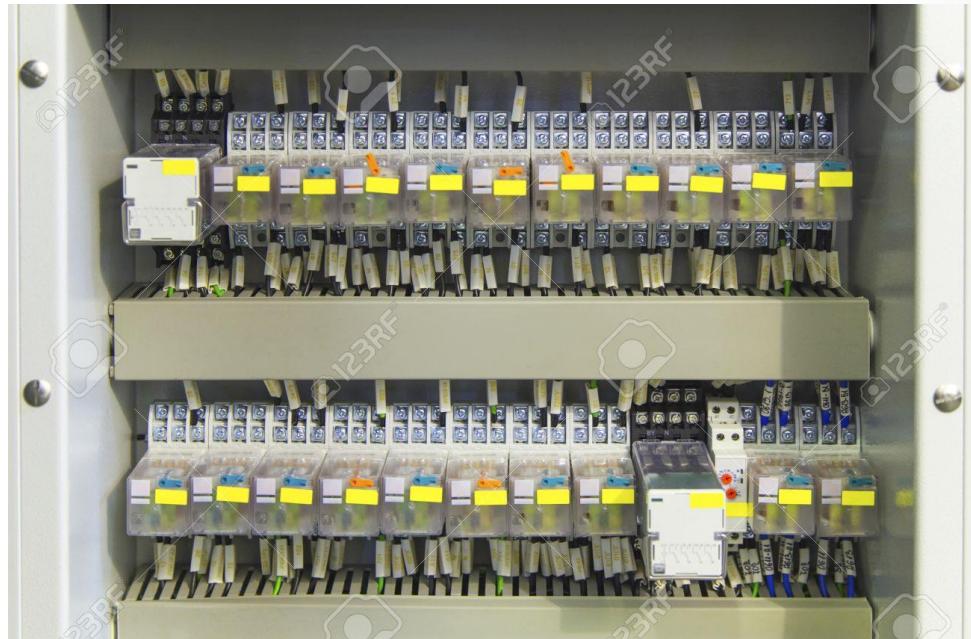


Relay



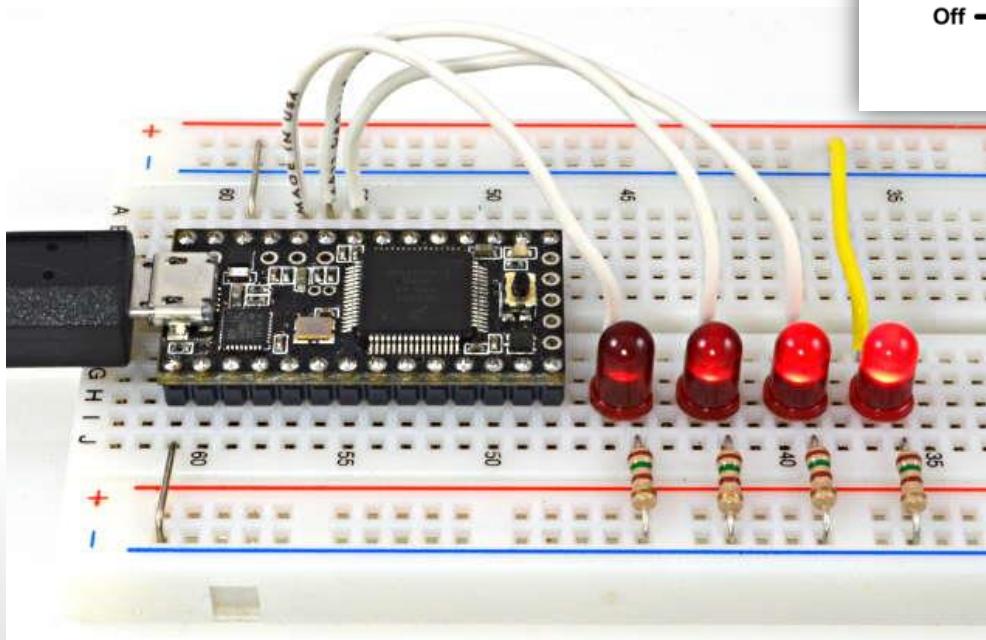
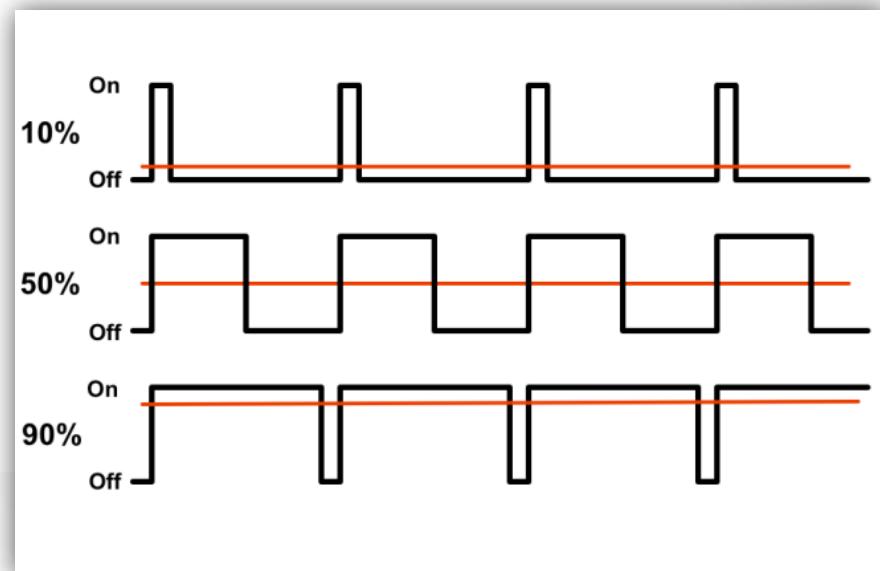


In Real Life



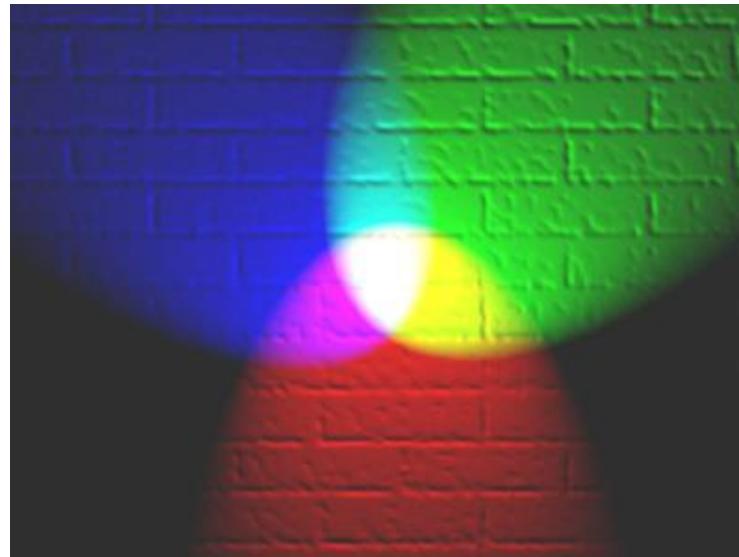
Analog Output - PWM

- Fade in/Show up LED:
 - PWM



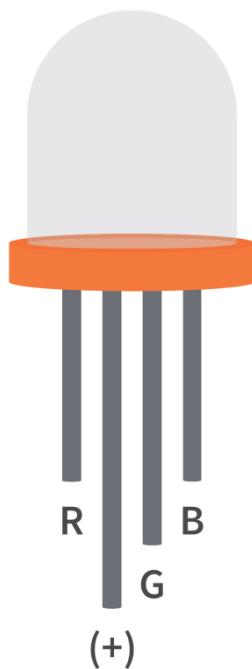
RGB LEDs

- Controlling RGB LEDs

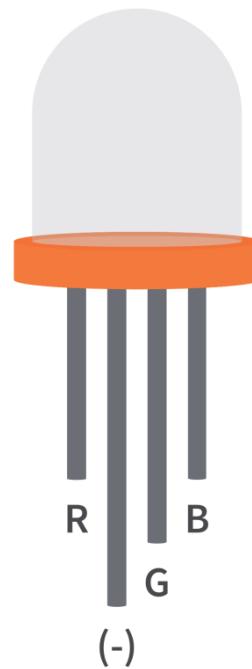


RGB LEDs

- Controlling RGB LEDs



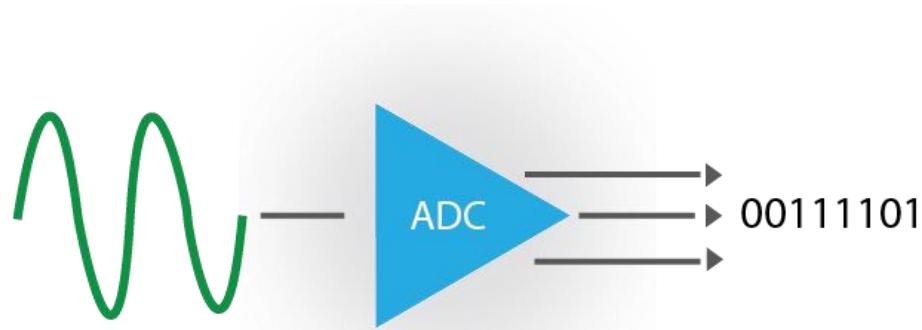
Common Anode



Common Cathode

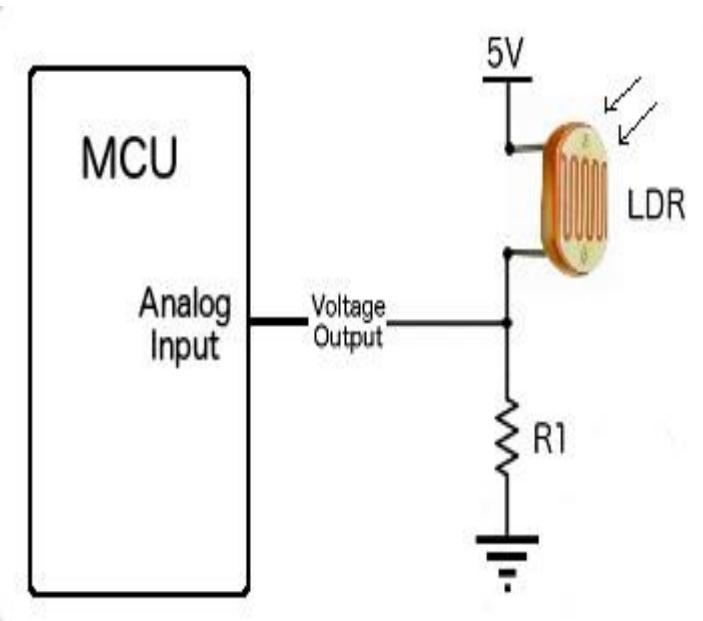
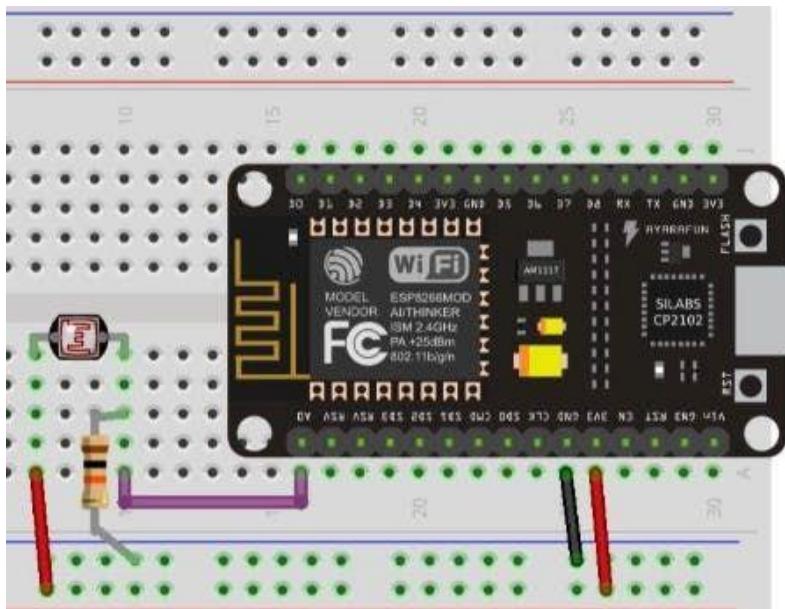
Light Sensor - LDR

- Analog :
 - Sensors :



Light Sensor - LDR

- Analog :
 - Sensors :



Blynk App

Virtual pins : push data.

```
Blynk.virtualWrite(V5, sensorData); //sending to Blynk
```

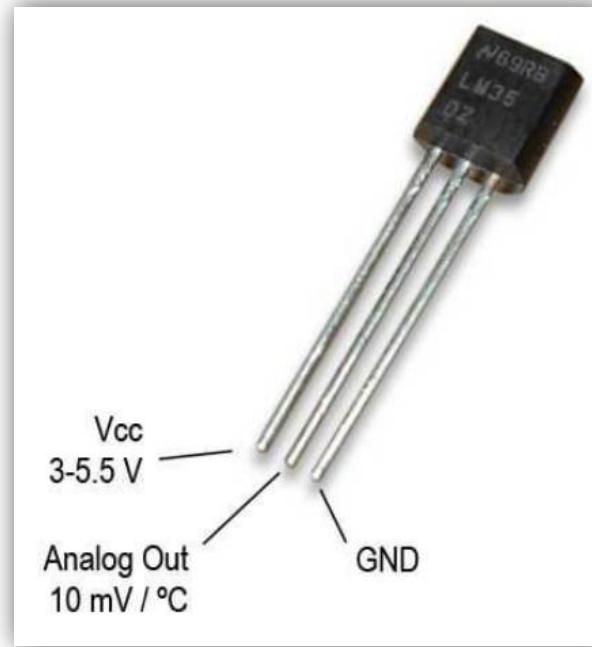
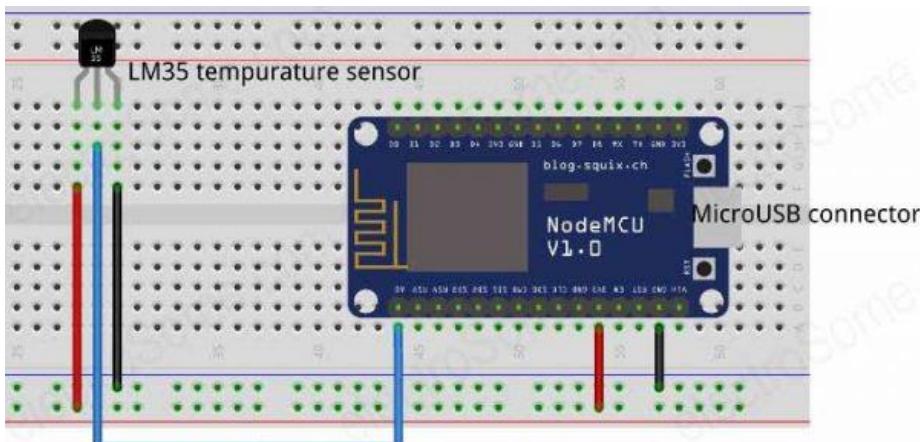
Virtual pins : read commands.

```
BLYNK_WRITE(V1)
{
    int pinValue = param.asInt();
    Serial.print("V1 Slider value is: ");
    Serial.println(pinValue);
}
```



Temperature sensor – LM35

- Communicate with MCU:
 - Serial communication
 - Read Temperature by LM35.

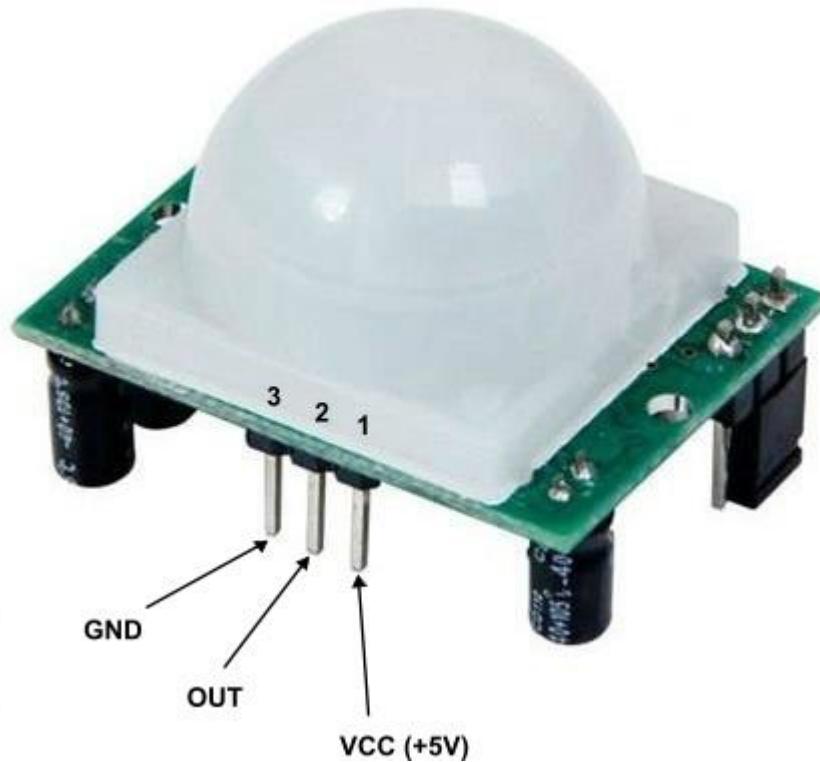


In Real Life



Motion sensor – PIR sensor

- Reading Motion sensor





Blynk App

- In-App Notification

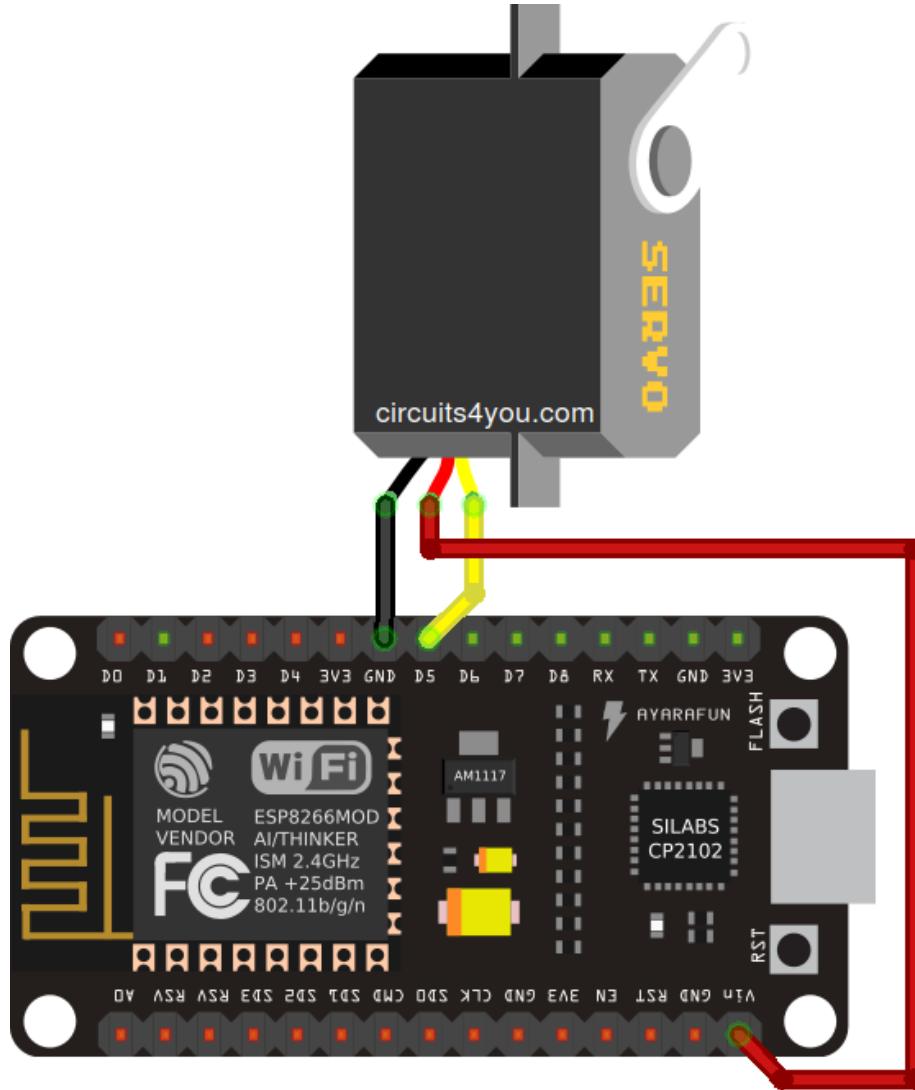
```
Blynk.notify ("Alert !");
```



Angular motion - Servo Motors



Angular motion - Servo Motors



Coding ...

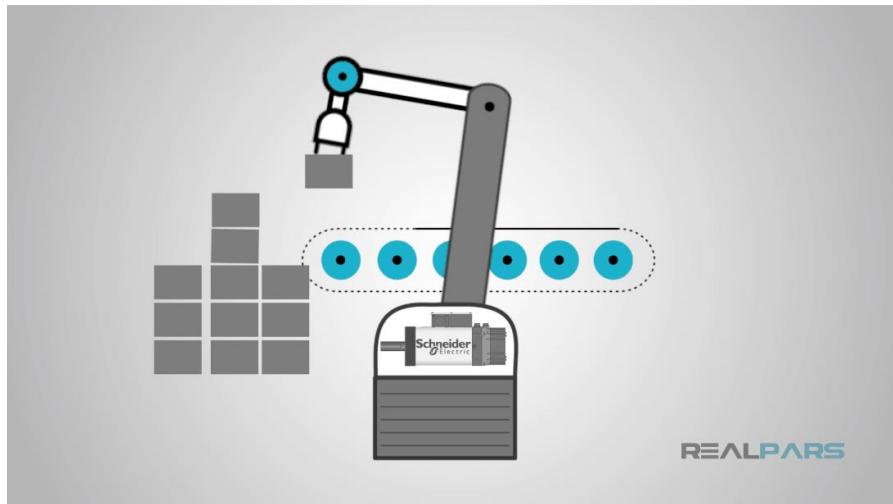
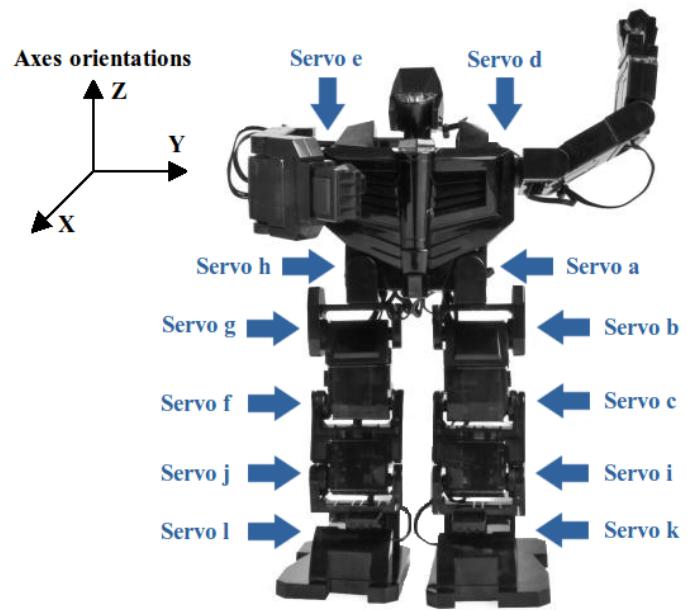


```
uint32_t prevmillis; //As global variable.

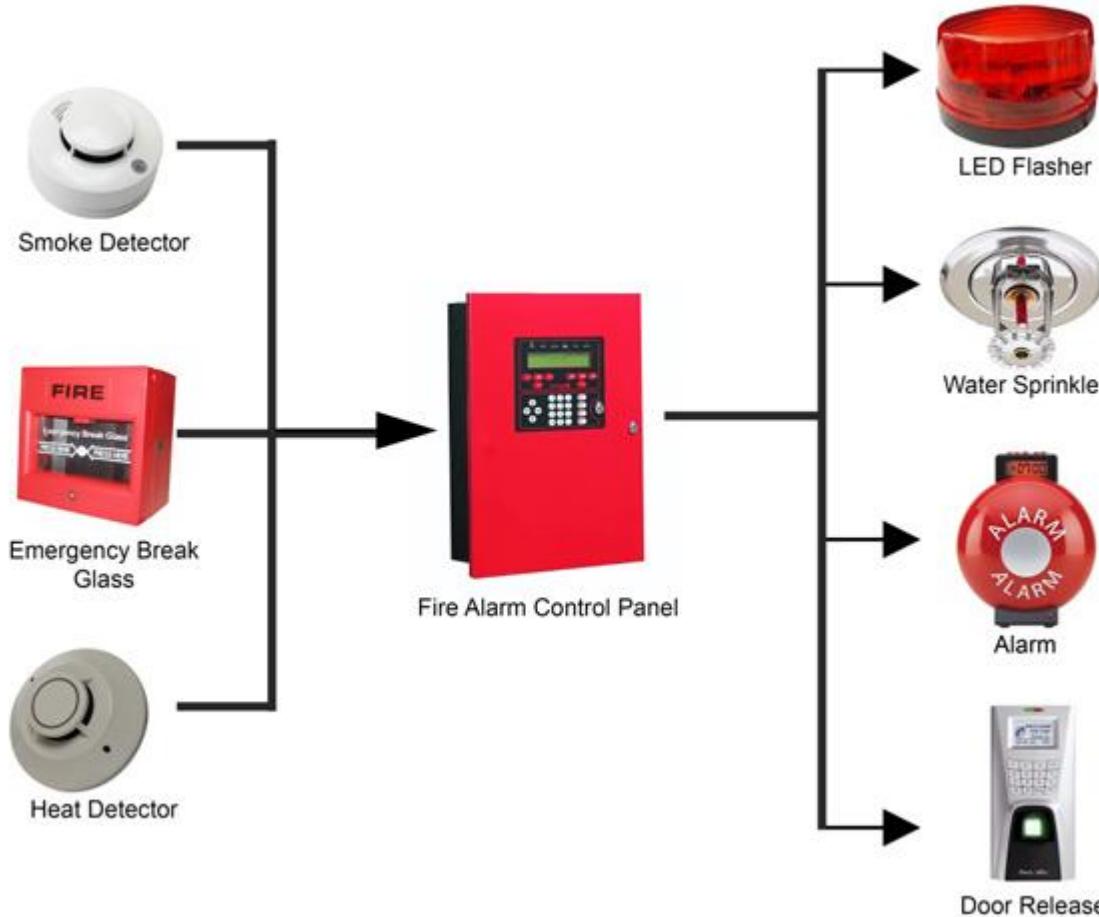
if(millis() - prevmillis > 1000){ //At main loop.
    digitalWrite( ledPin , !digitalRead(ledPin) );

    prevmillis = millis();
}
```

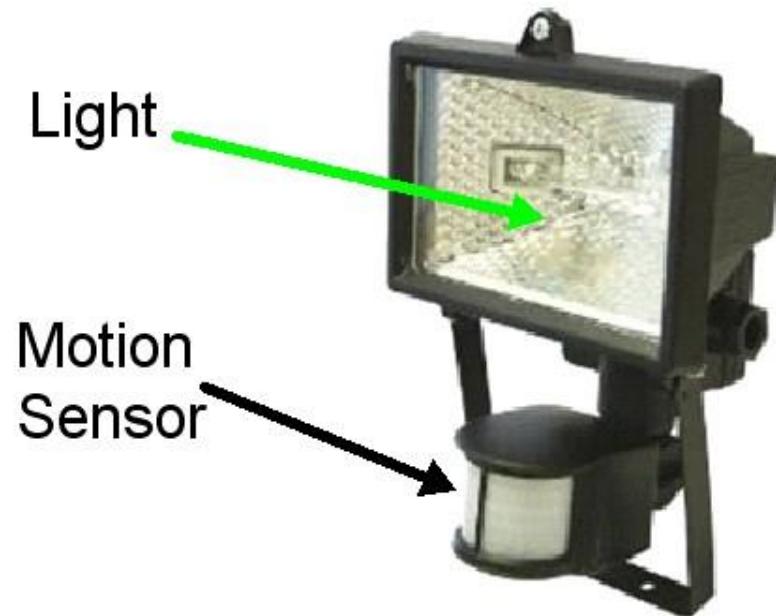
السر وراء الاستجابة الفورية : شرح لهذه الطريقة



Systems analysis



Systems analysis



Systems analysis



Systems analysis



Systems analysis



Systems analysis



SONOFF

- Easy Configurable AC Switch
- ESP8266 Core



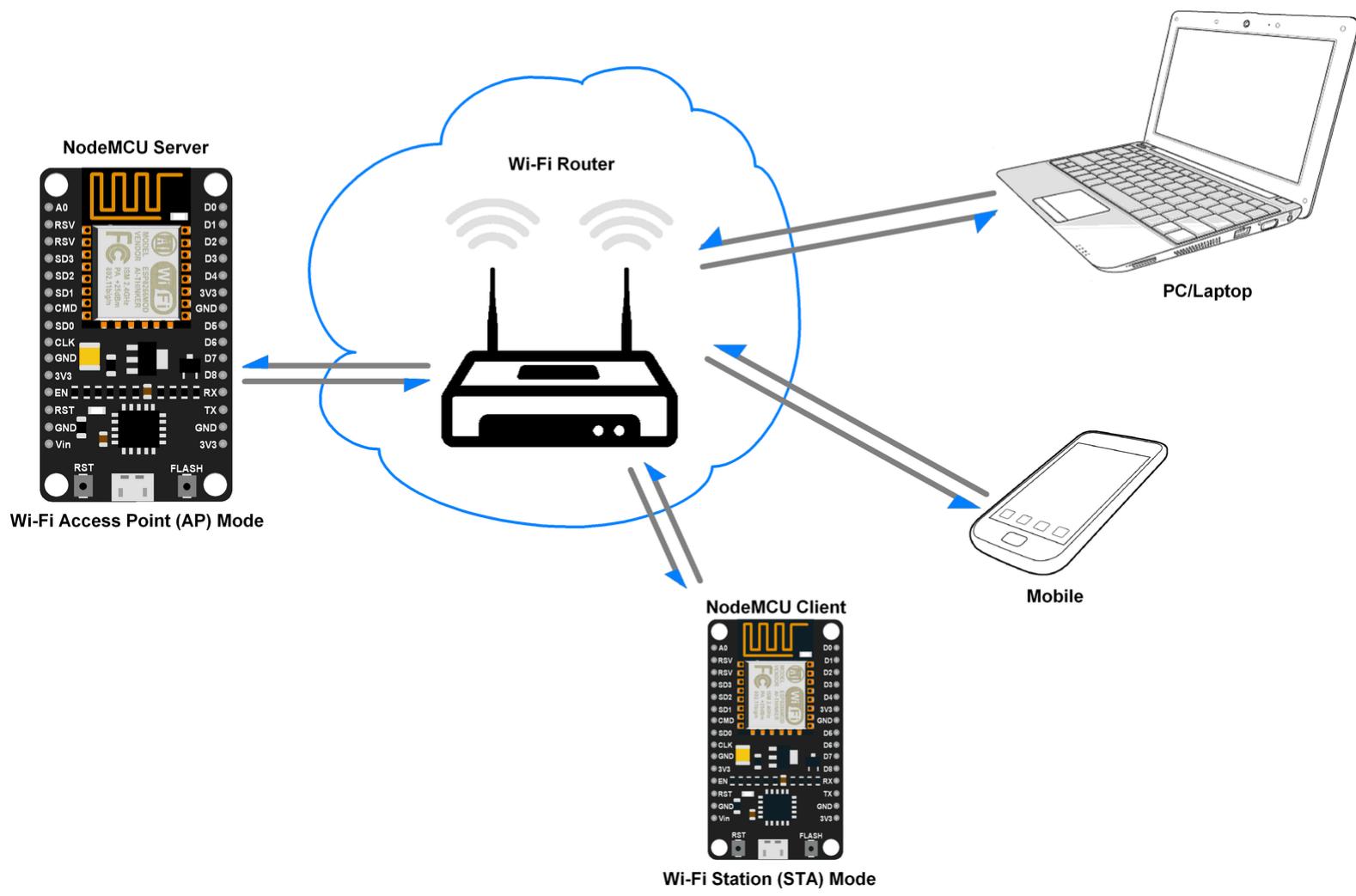
SONOFF

- 4 Channels



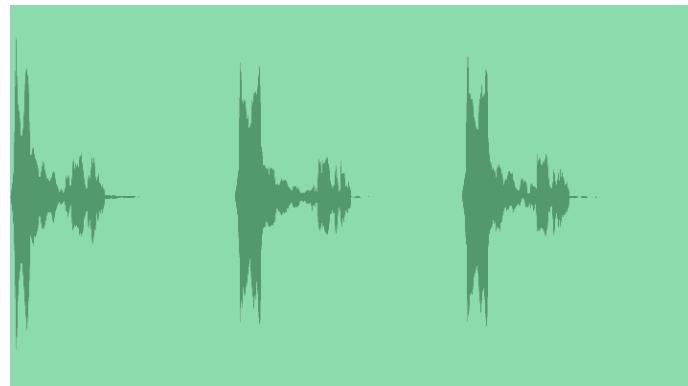
Arduino ESP8266

LAN

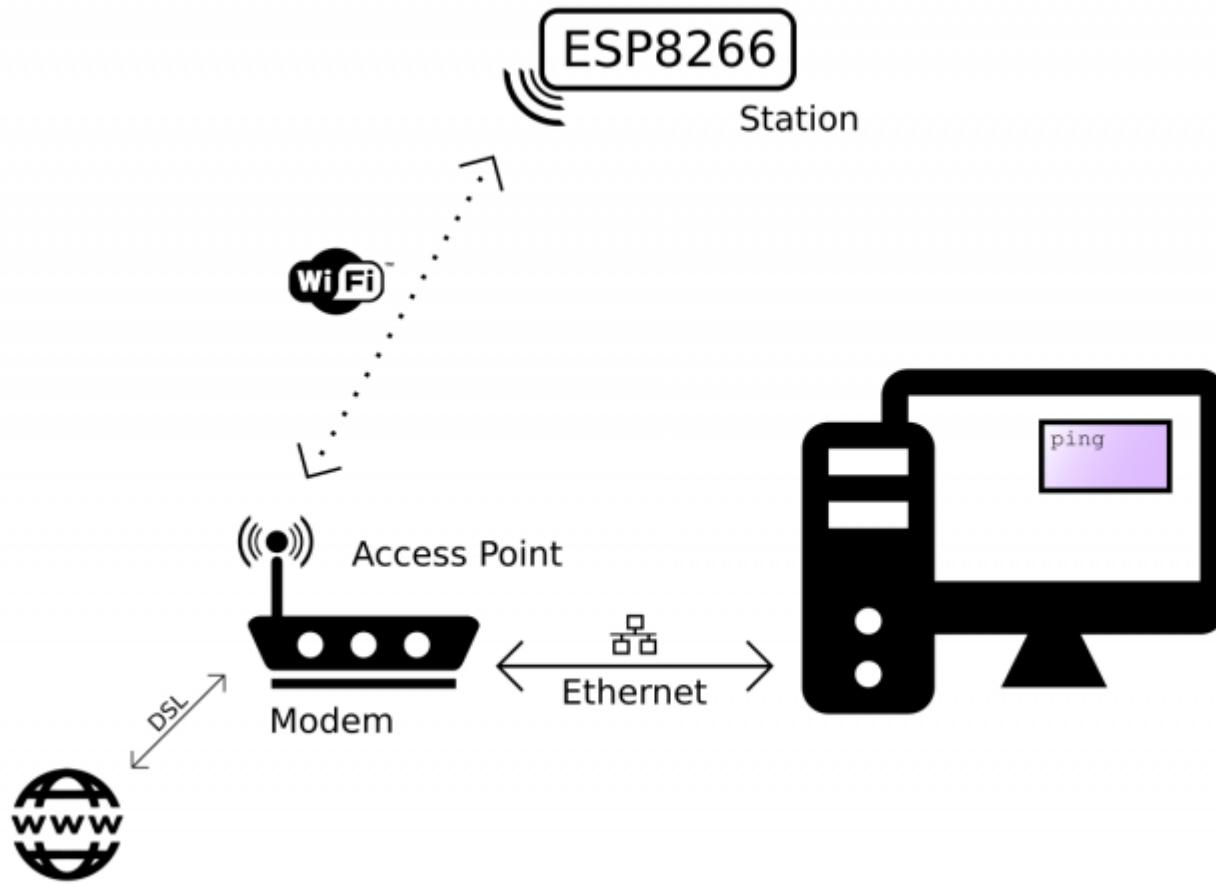


Beginning with ESP8266

- **Connect :**
 - `ESP8266WIFI.h`
 - `WiFi.begin(SSID,password)`
 - `WiFi.waitForConnectResult()`
 - Obtain and Print IP Address
 - Ping
 - Ping ESP's local IP
 - Ping google.com



Beginning with ESP8266



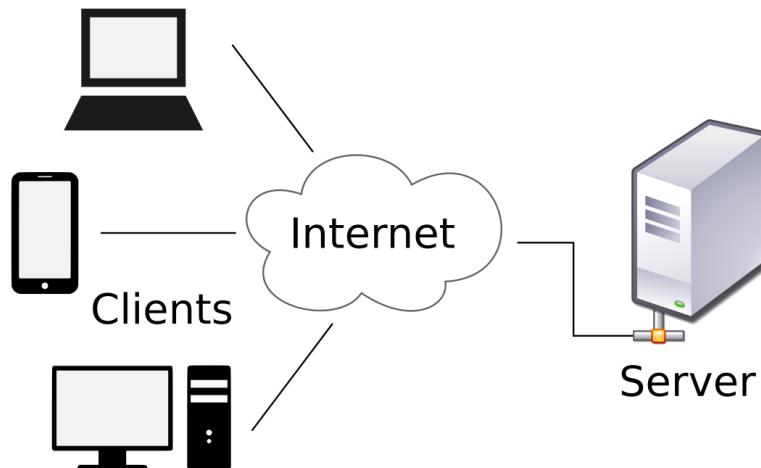
Web Server

- Open Second project “WebServer”
- Modify Credentials
- Go to :
 - <http://wha.tev.er.ip/OFF>
 - <http://wha.tev.er.ip/ON>
 - Ex : <http://192.168.1.15/ON>

Web Server

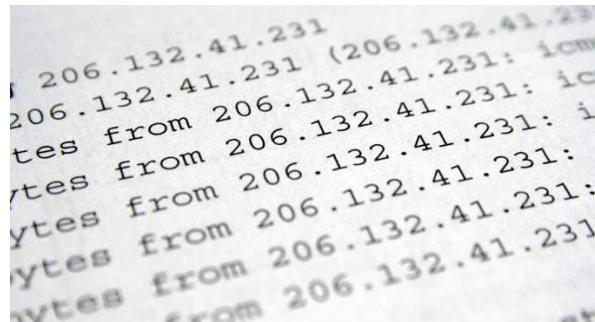
Server : A **server** is a computer that provides data to other computers. It may serve data to systems on a local area network (LAN) or a wide area network (WAN) over the Internet.

Client : A **client** is a piece of computer hardware or software that accesses a service made available by a server. The server is often (but not always) on another computer system, in which case the client accesses the service by way of a network.

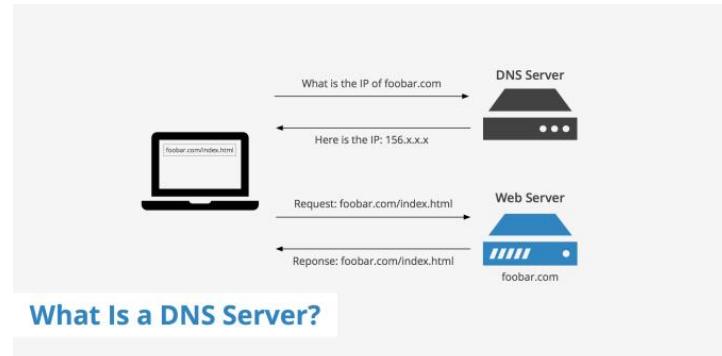


Web Server

IP Address : An Internet Protocol **address (IP address)** is a numerical label assigned to each device connected to a computer network that uses the Internet Protocol for communication.

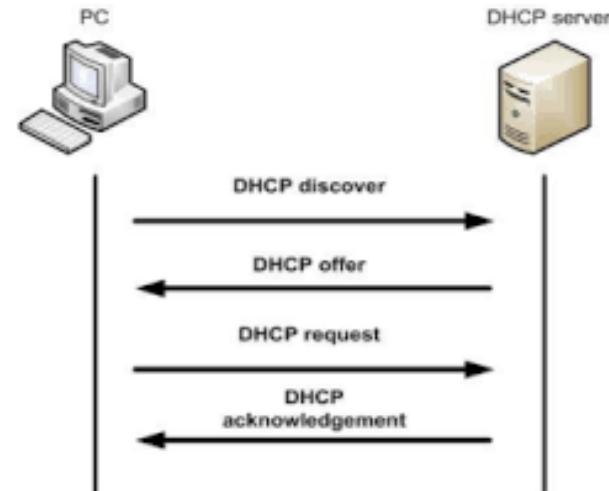


DNS : Domain Name **Servers (DNS)** are the Internet's equivalent of a phone book. They maintain a directory of domain names and translate them to Internet Protocol (IP) addresses.

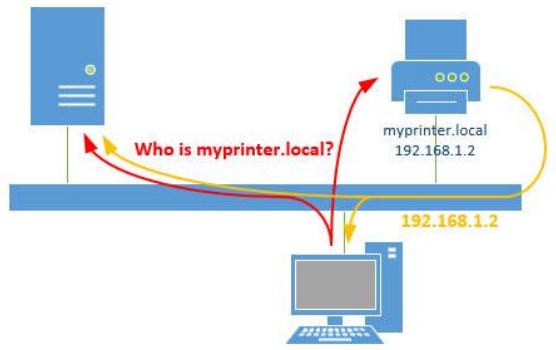


Web Server

DHCP : The **Dynamic Host Configuration Protocol (DHCP)** is a network management protocol used on UDP/IP networks whereby a DHCP server dynamically assigns an IP address and other network configuration parameters to each device on a network so they can communicate with other IP networks.



mDNS : **multicast DNS (mDNS)** protocol resolves hostnames to IP addresses within small networks that do not include a local name server.



Web Server

LAN : A **local area network (LAN)** is a computer network that interconnects computers within a limited area such as a residence, school, laboratory, university campus or office building.

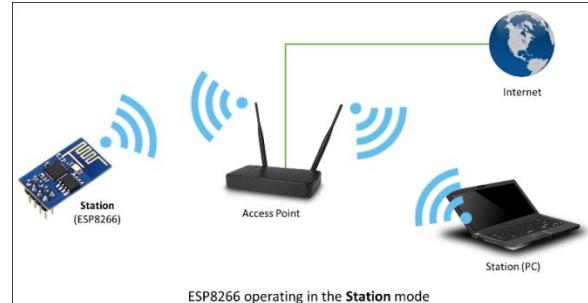


Station :

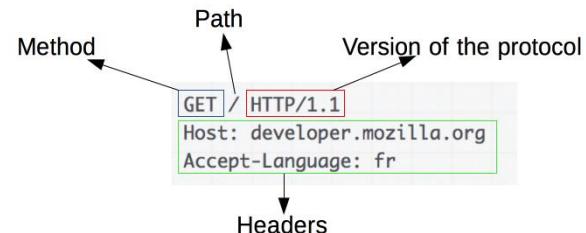
In IEEE 802.11 (Wi-Fi) terminology, a **station** (abbreviated as **STA**) is a device that has the capability to use the 802.11 protocol. For example, a station may be a laptop, a desktop PC, PDA, access point or Wi-Fi phone. An STA may be fixed, mobile or portable.

Web Server

Access Point : **access point (AP)**, is a networking hardware device that allows other Wi-Fi devices to connect to a wired network. An AP is differentiated from a hotspot, which is the physical location where Wi-Fi access to a WLAN is available.



HTTP : **Hypertext Transfer Protocol (HTTP)** is an application protocol for distributed, collaborative, hypermedia information systems.



Connecting over LAN

- Create a Web server
 - simple server's code “Hello World”:
 - HTTP responses



PAGE NOT FOUND

HTTP Status Codes		
Level 200 (Success)	Level 400	Level 500
200 : OK	400 : Bad Request	500 : Internal Server Error
201 : Created	401 : Unauthorized	503 : Service Unavailable
203 : Non-Authoritative Information	403 : Forbidden	501 : Not Implemented
204 : No Content	404 : Not Found	504 : Gateway Timeout
	409 : Conflict	599 : Network timeout
		502 : Bad Gateway

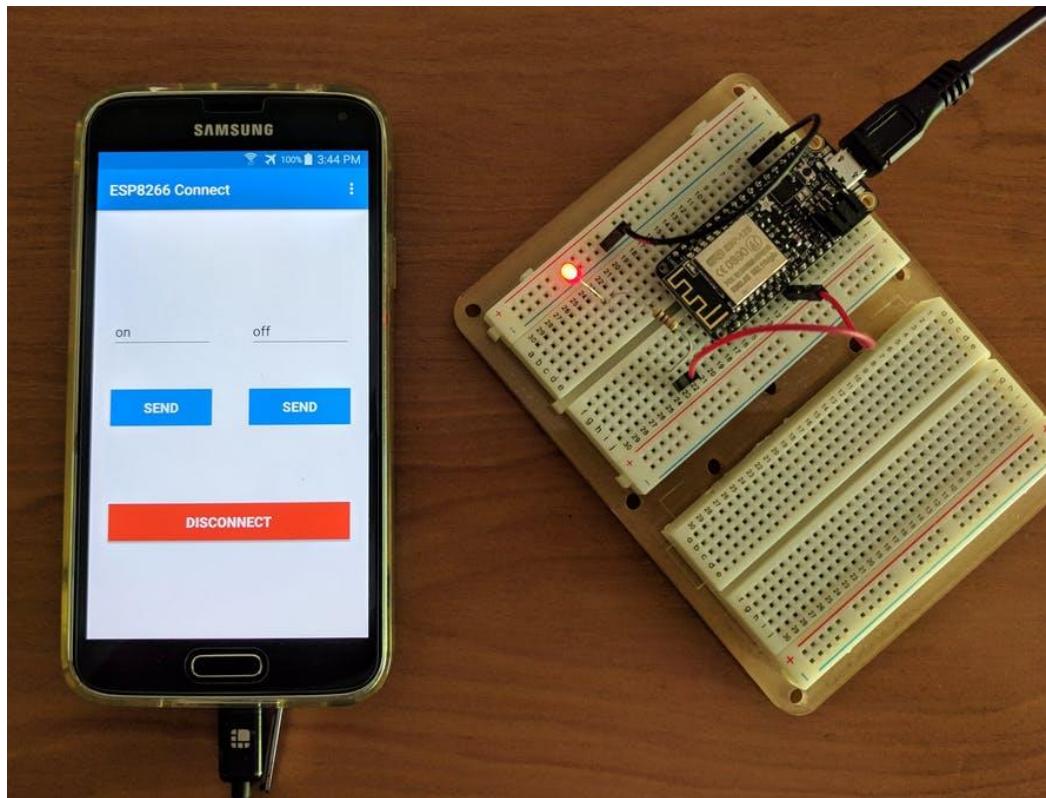
Connecting over LAN

- Create a Web server
 - simple server's code “Hello World”:
 - HTTP responses
 - HTML code

```
1 <!DOCTYPE HTML>
2 <html>
3     <head>
4     </head>
5     <body>
6         <h1 style="color:#ff0000">
7             "Hello World!"
8         </h1>
9     </body>
10    </html>
11
12
13
```

Connecting over LAN

- Create a Web server
 - Android Application
 - Google search : App inventor



As Access Point

- Open Third Project “AccessPoint”
- Upload and Monitor to ESP8266
- Connect to its network.
- Go to :
 - <http://wha.tev.er.ip/OFF>
 - <http://wha.tev.er.ip/ON>
 - Ex : <http://192.168.4.1/ON>

Coding ...

- Functions
 - Bundles a set of commands in one packet.

- Take money
- Open door
- Close door
- Go 100m forward
- Go inside shop
- Buy 1 Milk
- Come back home



- Get Milk :

Take money
Open door
Close door
Go 100m forward
Go inside shop
Buy 1 Milk
Come back home

Coding ...

- Functions
 - Bundles a set of commands in one packet.

```
int main()
```

```
{
```

```
    sum1 = sum(20,30);
```

```
    sum2 = sum(20,30,40);
```

```
}
```

```
int sum(int a,int b)
```

```
{
```

```
    return (a+b);
```

```
}
```

```
int sum(int a,int b,int c)
```

```
{
```

```
    return (a+b+c);
```

```
}
```

Coding Essentials

- “Programs must be written for people to read, and only incidentally for machines to execute.”
— **Harold Abelson**, Structure and Interpretation of Computer Programs

- “The most disastrous thing that you can ever learn is your first programming language.”

— Alan Kay

- "Every great developer you know got there by solving problems they were unqualified to solve until they actually did it." -

— **Patrick McKenzie**

Our Society

www.facebook.com/groups/ArabIEC

