

IOT & Applications

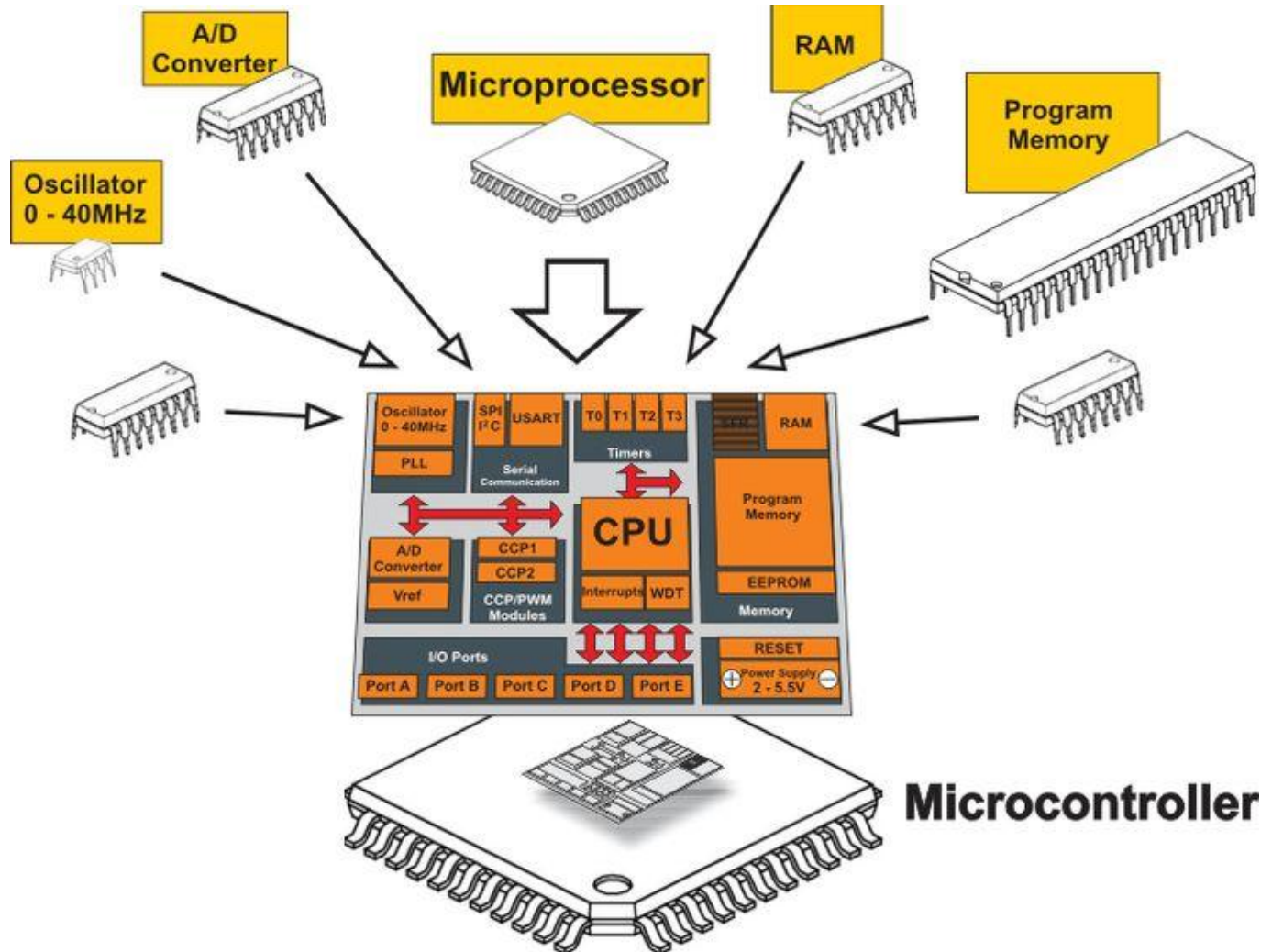
Module - 4

IoT using Arduino: Interoperability in IoT, Arduino Programming, Integration of Sensors and Actuators, Microcontrollers, Embedded C programming, Analog Interfacing

Microprocessor and Microcontroller

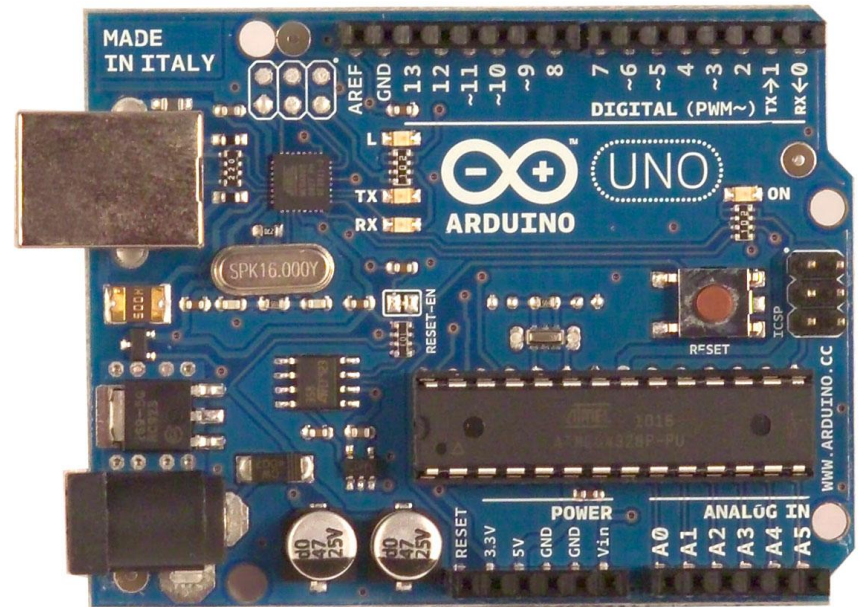
- ▶ Microprocessor is a computer processor that incorporate the function of a central processing unit on a single integrated circuit. It is the central unit that executes and manages the logical instruction passed to it.
- ▶ Microcontroller is a integrated circuit that contains a microprocessor along with memory and associated circuit like RAM, EEPROM, AD-DA converter etc.

Microprocessor and Microcontroller



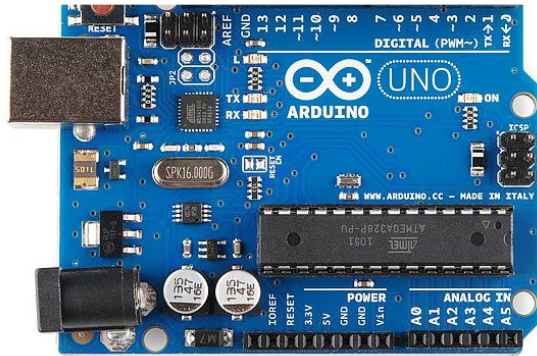
What is an Arduino ?

- ▶ **Open Source** electronic prototyping **platform** based on flexible **easy to use** hardware and software.

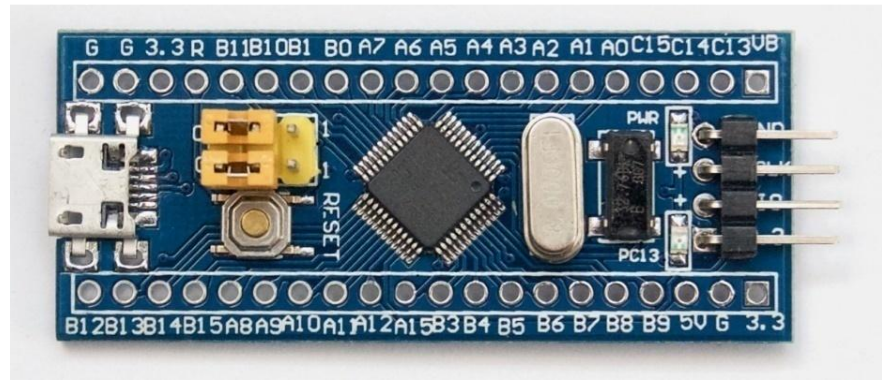


Different microcontroller board

Arduino



STM32



Beaglebone



TI MSP430



Bare minimum code

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

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


Bare minimum code

- ▶ **setup** : It is called only when the Arduino is powered on or reset. It is used to initialize variables and pin modes
- ▶ **loop** : The loop functions runs continuously till the device is powered off. The main logic of the code goes here. Similar to while (1) for micro-controller programming.

PinMode

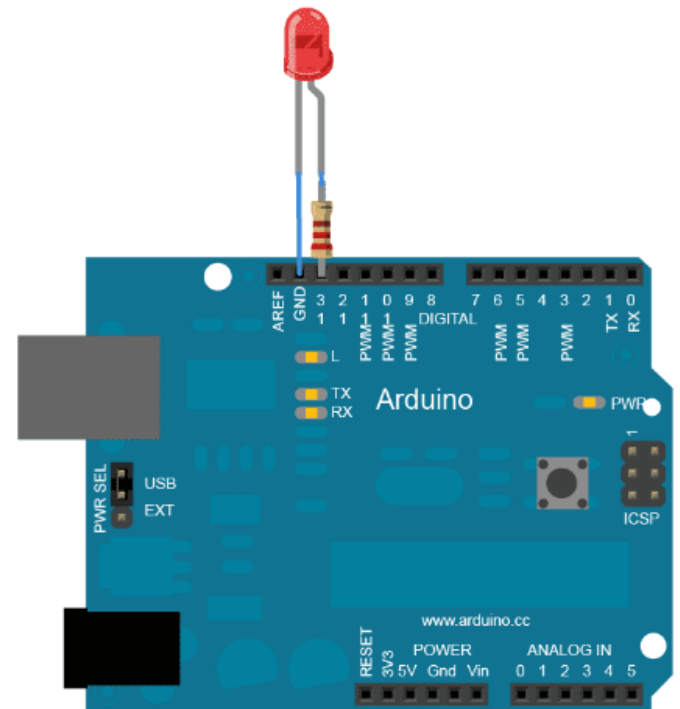
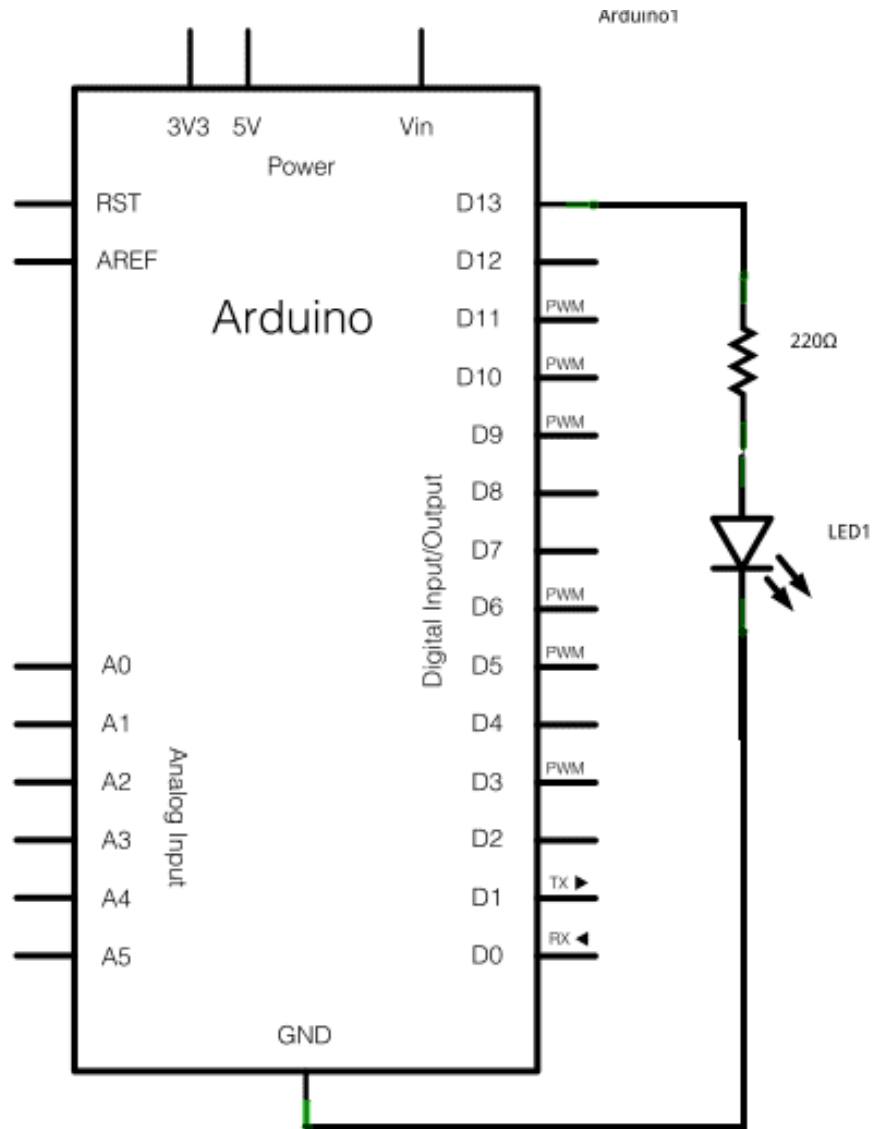
- ▶ A pin on arduino can be set as input or output by using pinMode function.
- ▶ `pinMode(13,OUTPUT);` // sets pin 13 as output pin
- ▶ `pinMode(13,INPUT);` // sets pin 13 as input pin

Reading/writing digital values

- ▶ `digitalWrite(13, LOW);` // Makes the output voltage on pin 13 , 0V
- ▶ `digitalWrite(13, HIGH);` // Makes the output voltage on pin 13 , 5V
- ▶ `int buttonState = digitalRead(2);` // reads the value of pin 2 in buttonState

Example-1: LED Blink

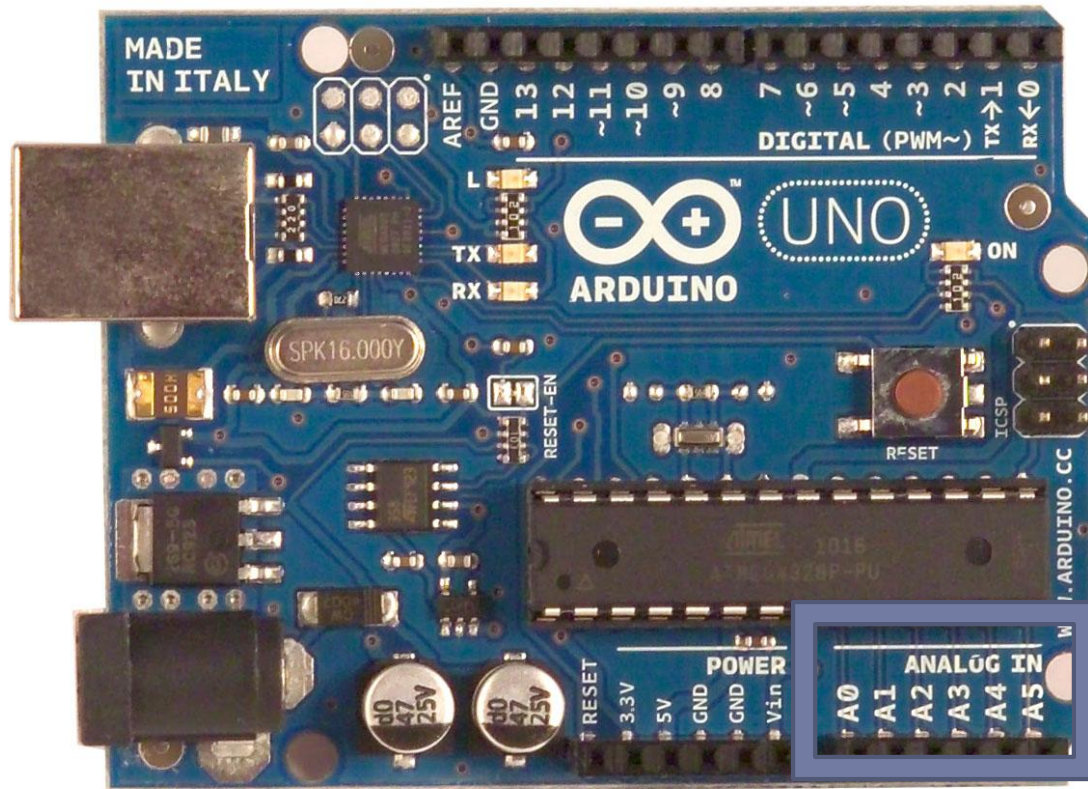
```
// the setup function runs once when you press reset or power the board
void setup()
{
  // initialize digital pin LED_BUILTIN as an output.
  pinMode(LED_BUILTIN, OUTPUT);
}
// the loop function runs over and over again forever
void loop()
{
  digitalWrite(LED_BUILTIN, HIGH); // turn the LED on (HIGH is the voltage level)
  delay(1000);                     // wait for a second
  digitalWrite(LED_BUILTIN, LOW);  // turn the LED off by making the voltage LOW
  delay(1000);                     // wait for a second
}
```



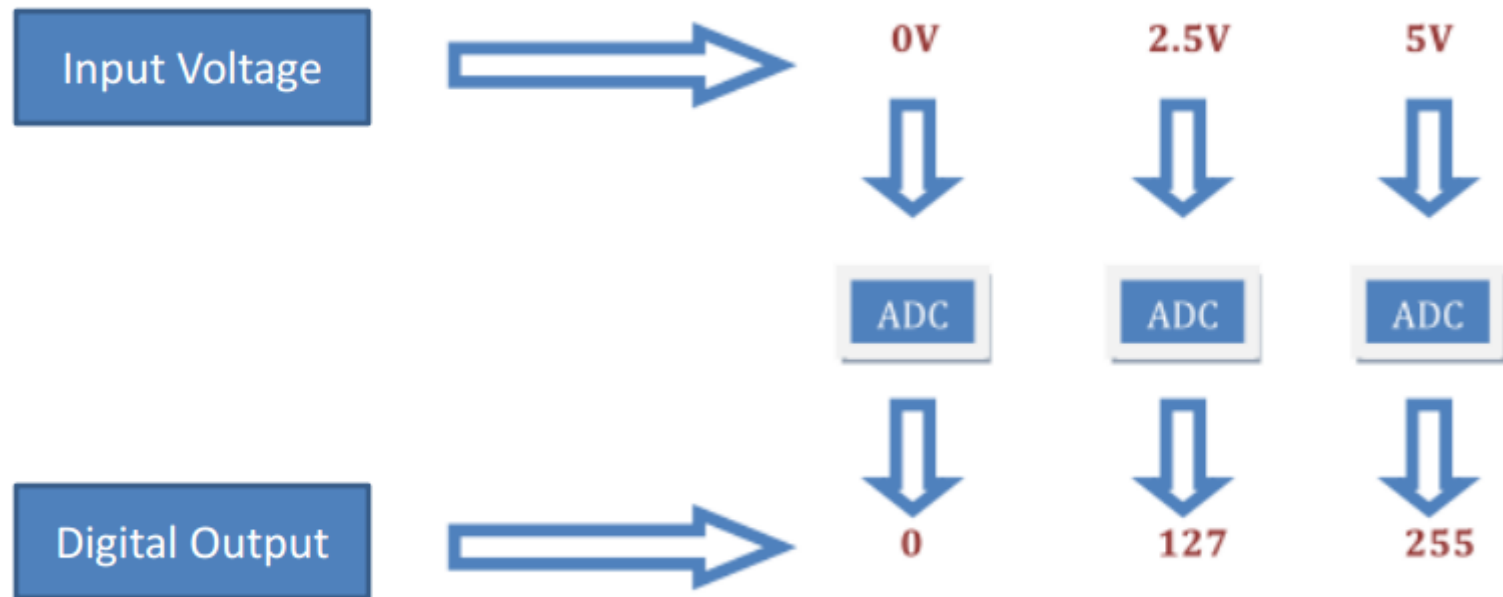
Analog to Digital Conversion

- ▶ What is analog ?
 - ▶ It is continuous range of voltage values (not just 0 or 5V)
- ▶ Why convert to digital ?
 - ▶ Because our microcontroller only understands digital.

ADC in Arduino Uno



Converting Analog Value to Digital



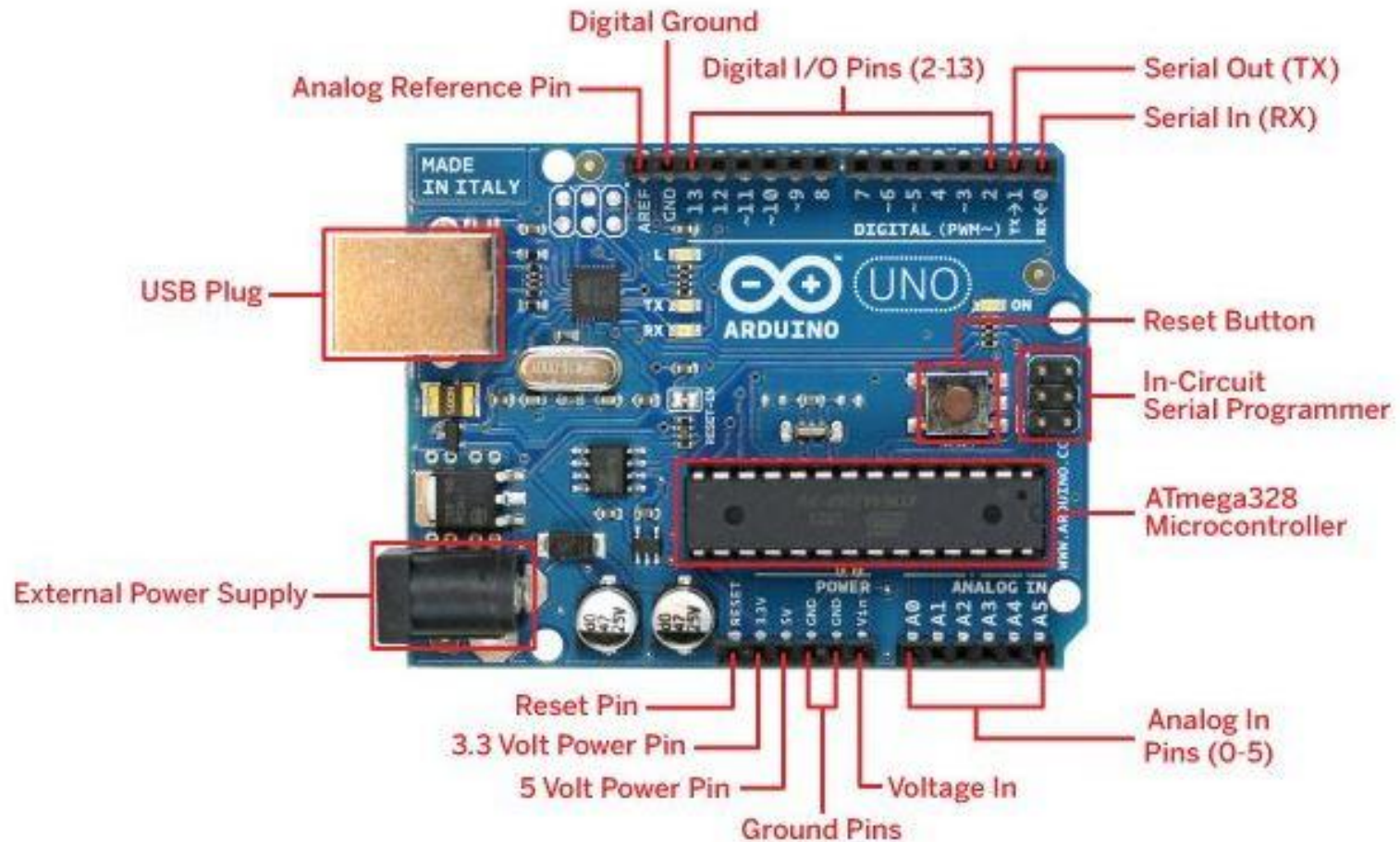
ADC in Arduino

- ▶ The Arduino Uno board contains 6 pins for ADC
- ▶ 10-bit analog to digital converter
- ▶ This means that it will map input voltages between 0 and 5 volts into integer values between 0 and 1023
- ▶ Resolution?

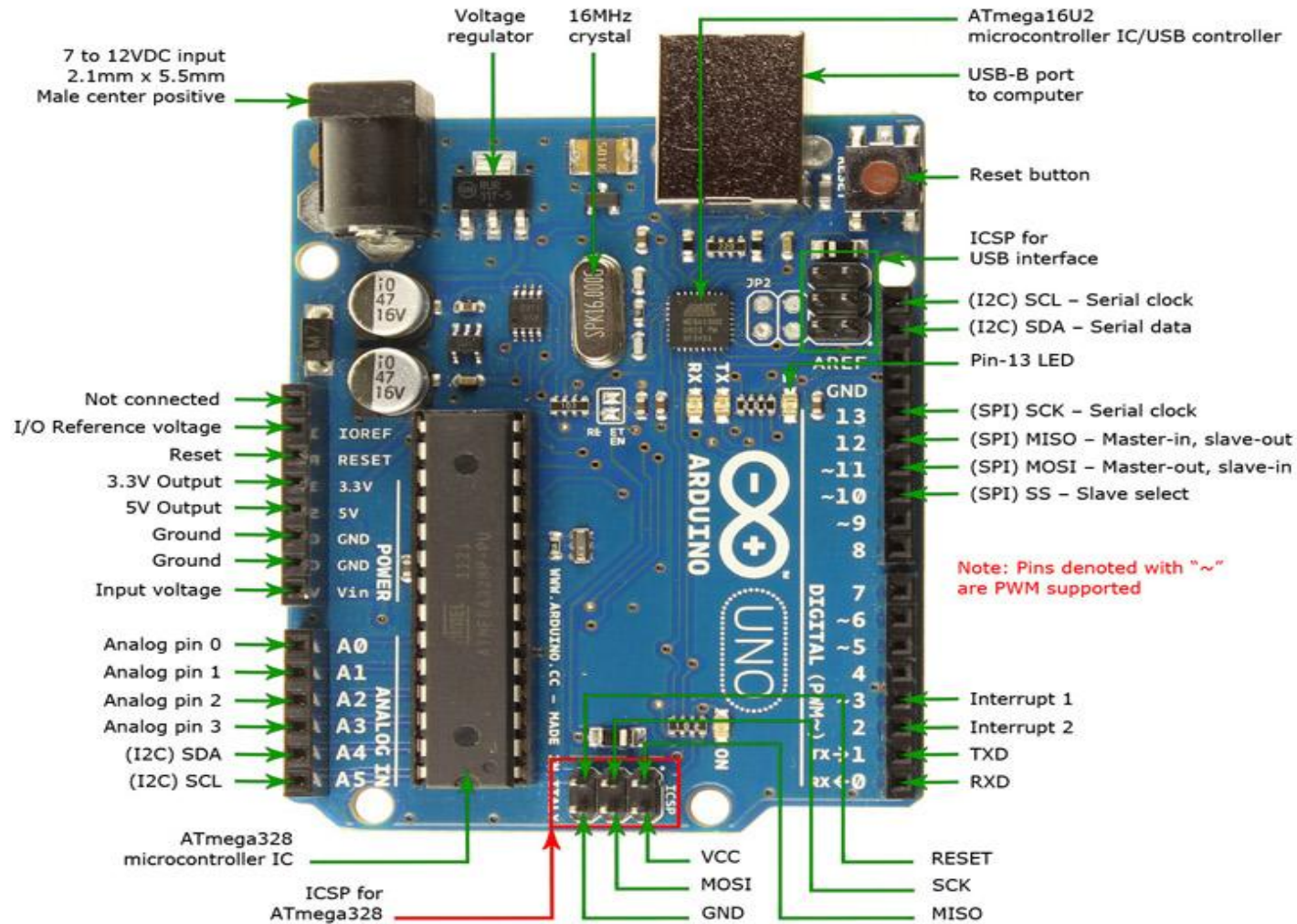
Reading/Writing Analog Values

- ▶ `analogRead(A0);`// used to read the analog value from the pin A0
- ▶ `analogWrite(2, 128);` **How much is the voltage?**

Arduino board components

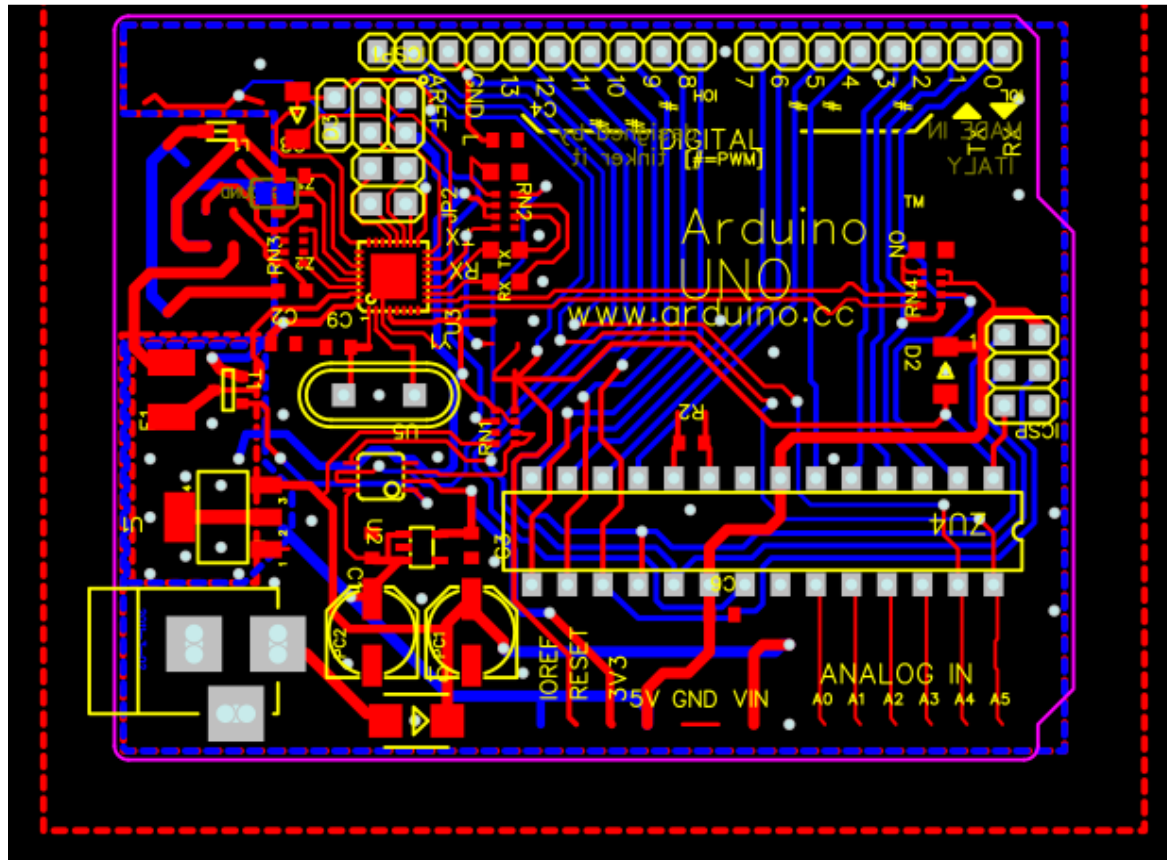


Arduino board Pin diagram



Arduino board layout

Layout can be defined in which parts/components are arranged or road laid out.



ADC Example-2: Connecting POT, scaling the value and printing in serial monitor

```
▶ // These constants won't change. They're used to give names to the pins used:
const int analogInPin = A0; // Analog input pin that the potentiometer is attached to
const int analogOutPin = 9; // Analog output pin that the LED is attached to

int sensorValue = 0; // value read from the pot
int outputValue = 0; // value output to the PWM (analog out)

void setup() {
    // initialize serial communications at 9600 bps:
    Serial.begin(9600);
}

void loop() {
    // read the analog in value:
    sensorValue = analogRead(analogInPin);
    // map it to the range of the analog out:
    outputValue = map(sensorValue, 0, 1023, 0, 255);

    // change the analog out value:
    analogWrite(analogOutPin, outputValue);

    // print the results to the serial monitor:
    Serial.print("sensor = ");
    Serial.print(sensorValue);
    Serial.print("\t output = ");
    Serial.println(outputValue);
    delay(2);
}
```

Example-3: ADC with LED Blink using LDR

► `/* Analog Input`

`int sensorPin = A0; // select the
input pin for the potentiometer`

`int ledPin = 13; // select the
pin for the LED`

`int sensorValue = 0; // variable to
store the value coming from the
sensor`

`void setup() {`

`// declare the ledPin as an
OUTPUT:`

`pinMode(ledPin, OUTPUT);
}`

`void loop()`

`{`

`// read the value from the sensor:`

`sensorValue = analogRead(sensorPin);`

`// turn the ledPin on`

`digitalWrite(ledPin, HIGH);`

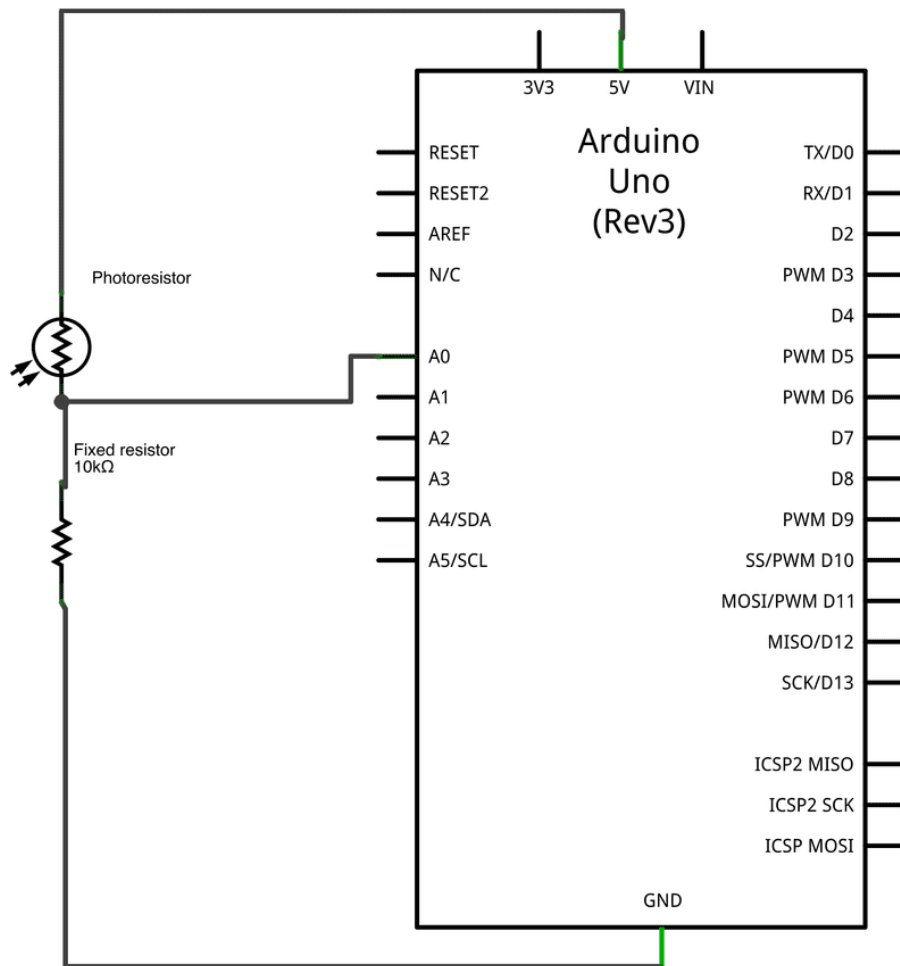
`delay(sensorValue);`

`digitalWrite(ledPin, LOW);`

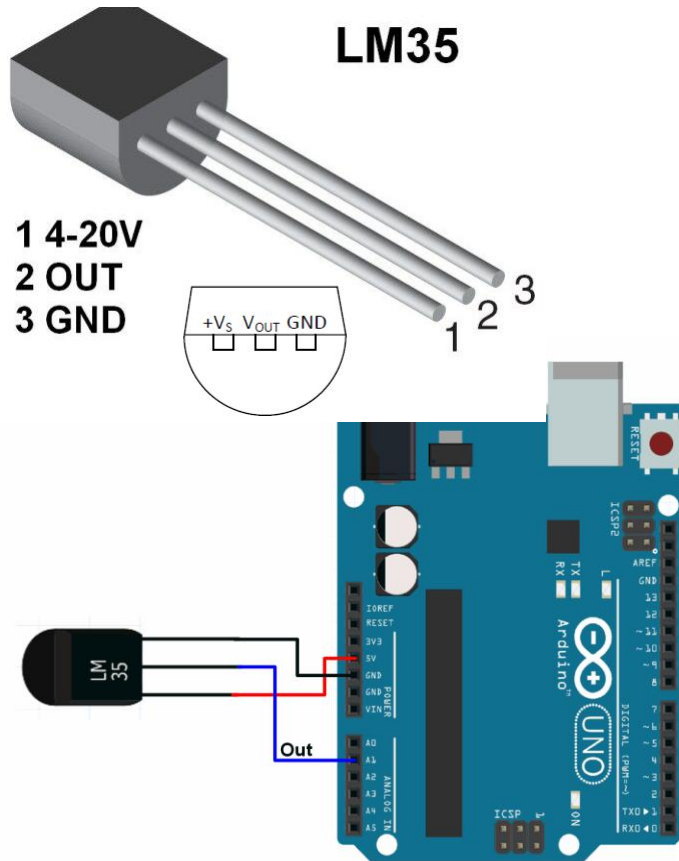
`delay(sensorValue);`

`}`

Example-3: ADC with LED Blink



Example-4: Temperature Sensor (LM35) with ADC and display temperature in degree C



```
void setup()
```

```
{ // initialize serial communication at 9600 bits per second:
```

```
Serial.begin(9600);
```

```
}
```

```
void loop() {
```

```
int sensorValue = analogRead(A0);
```

```
// Convert the analog reading (which goes from 0 - 1023) to a voltage (0 - 5V):
```

```
float voltage = sensorValue * (5.0 / 1023.0);
```

```
int temperature = (voltage/10);
```

```
Serial.println("Temperature=");
```

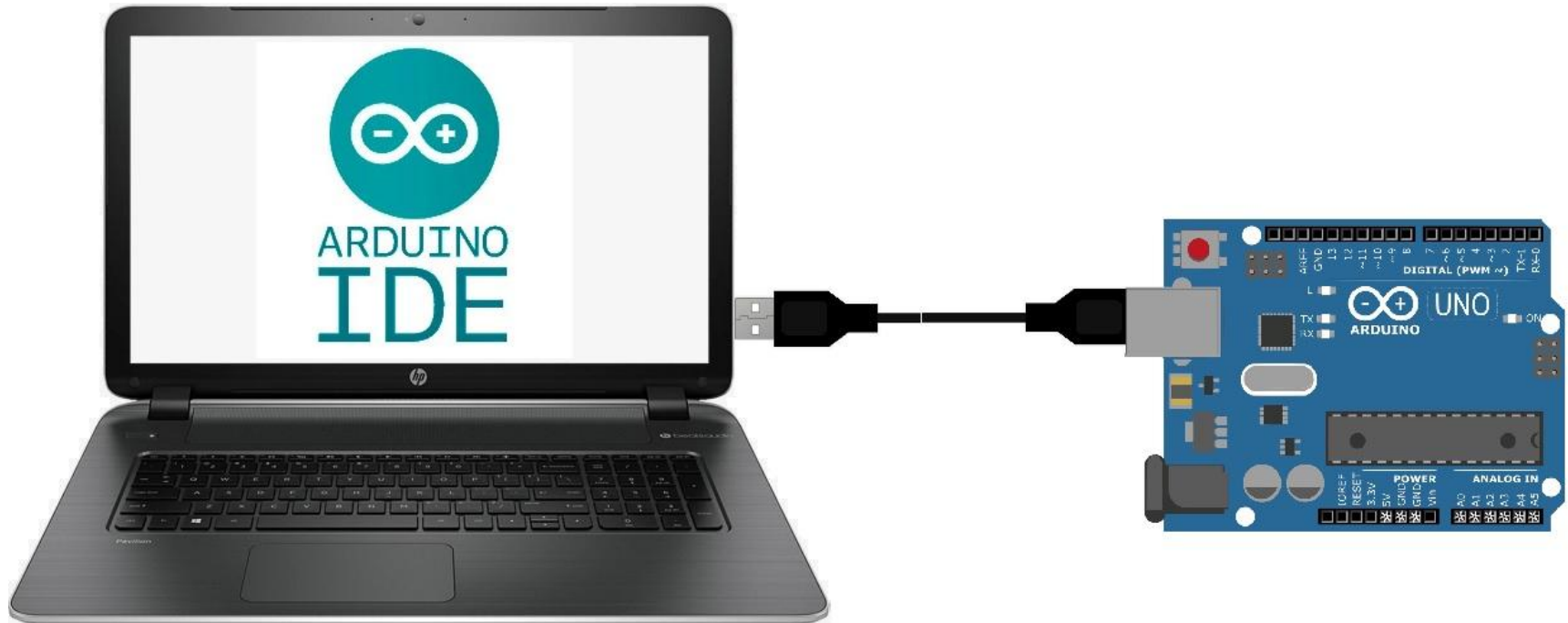
```
Serial.print(temperature);
```

```
}
```

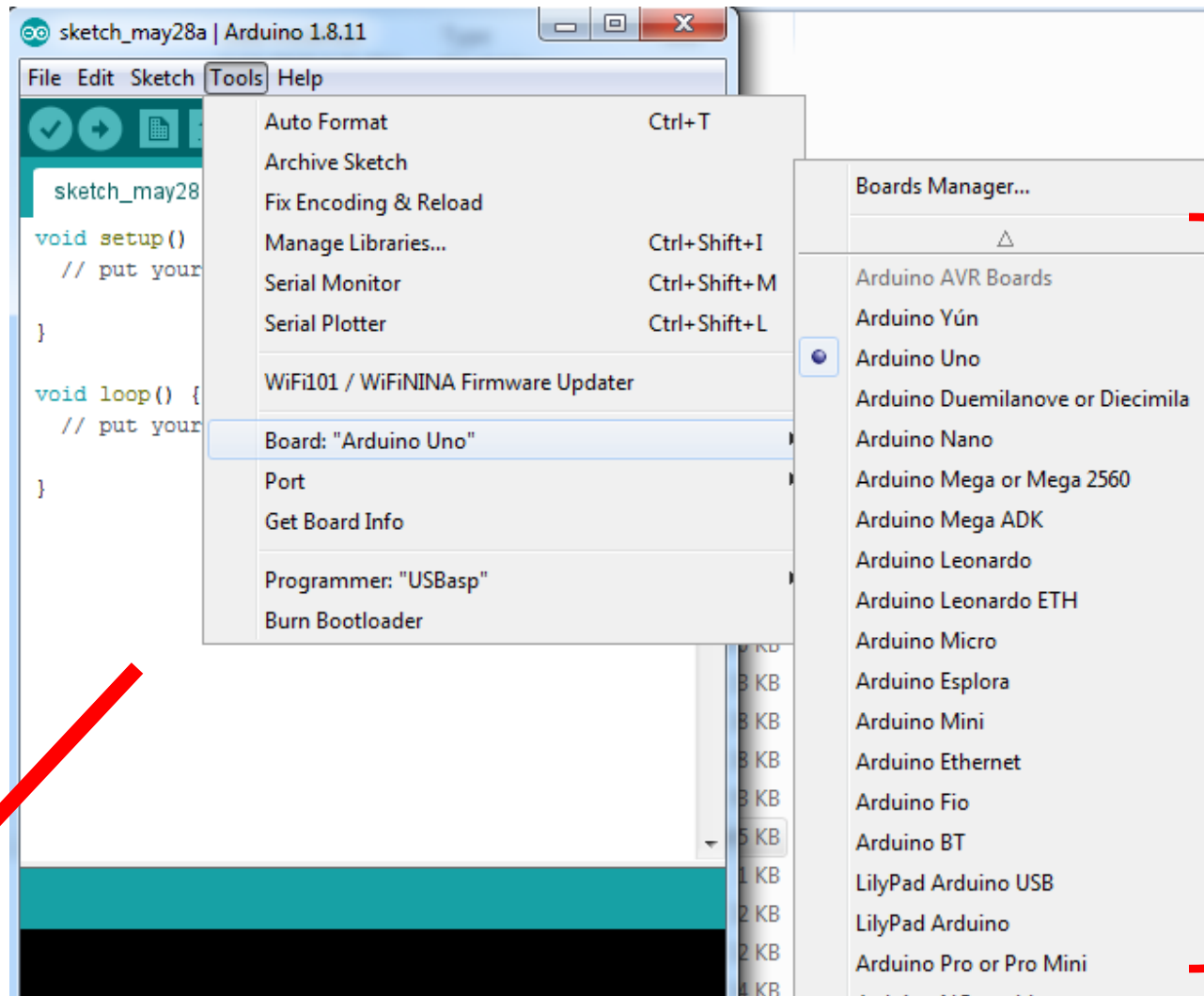
- **LM35 :Temperature sensitivity 10mv/C**
- ▶ • **Using 10 bit ADC**

Arduino Programming Procedure

PC connection with Arduino Through USB



PC connection with Arduino Through USB



Different Arduino Compatible Boards

Arduino IDE



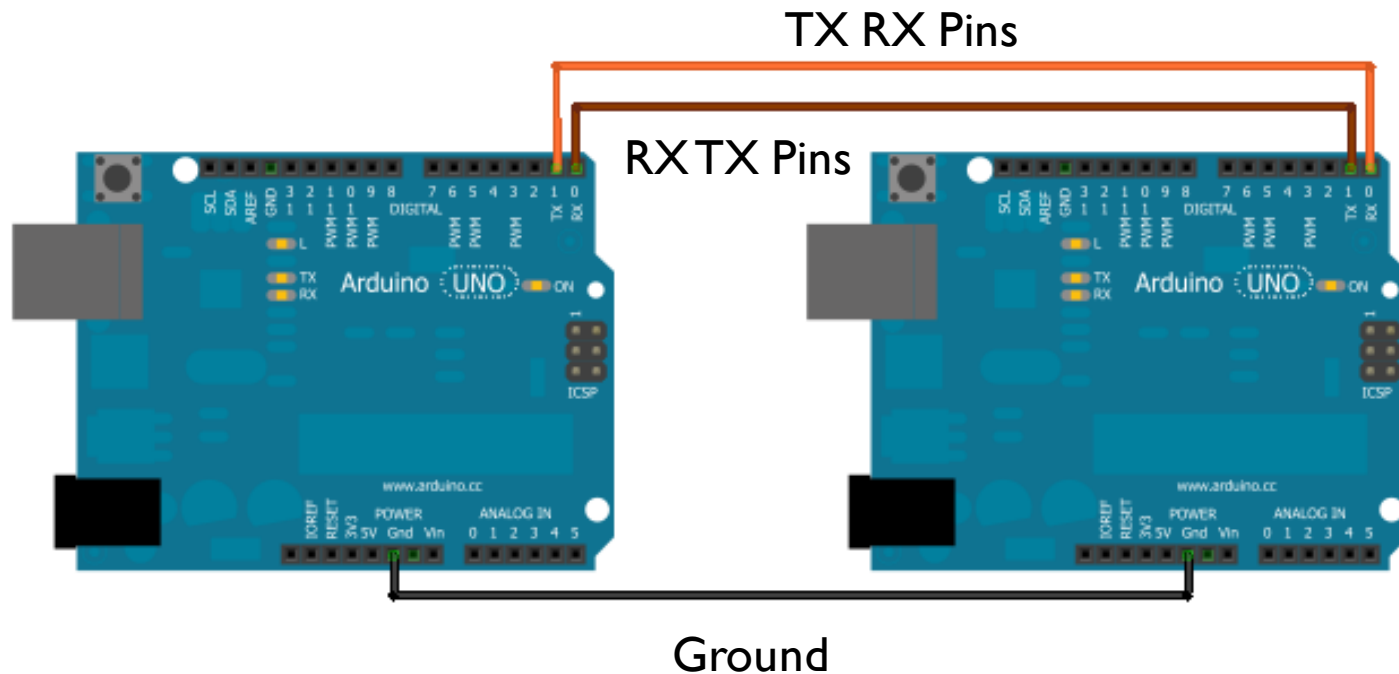
PC connection with Arduino Through USB



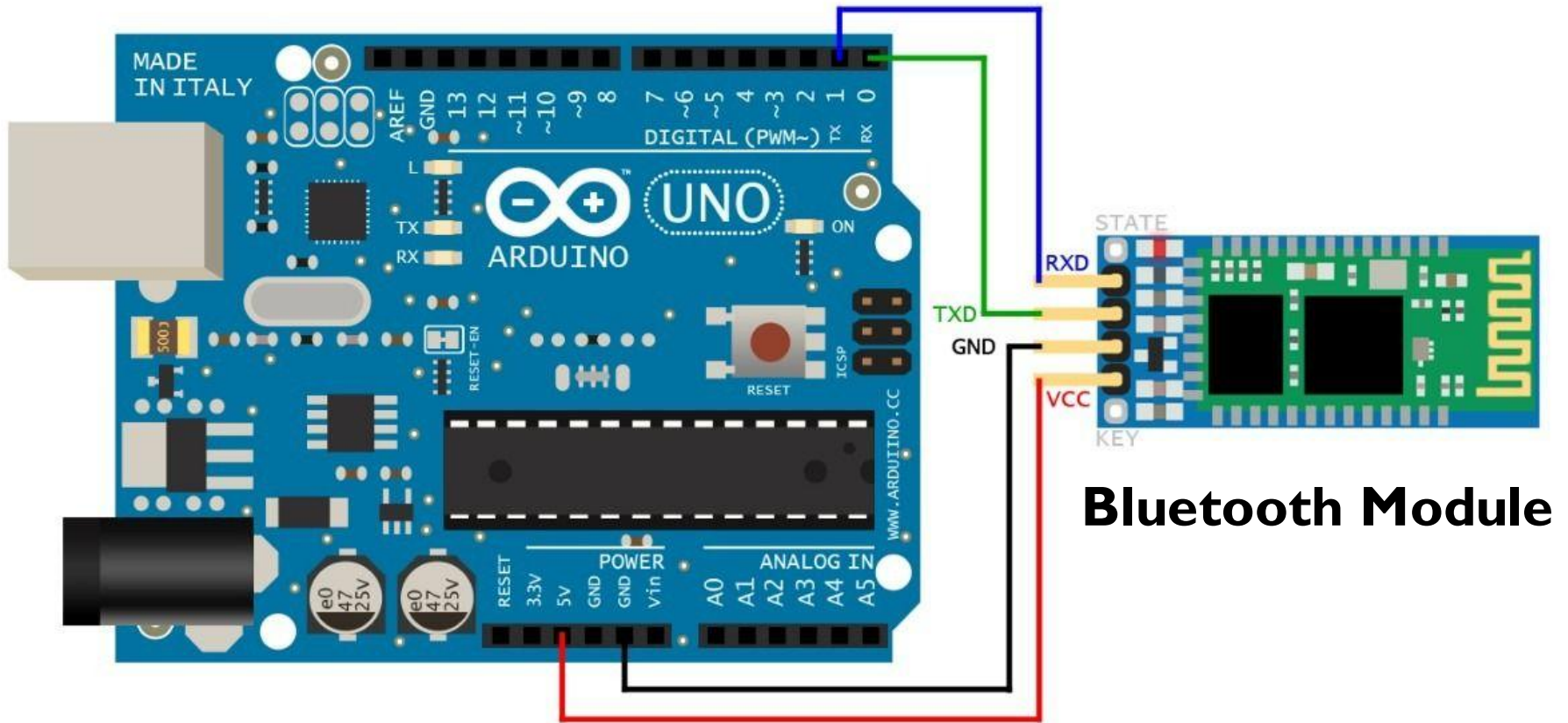
Arduino Uno Board

Arduino IDE

Example-1: Two Arduino Uno boards connected through RS232

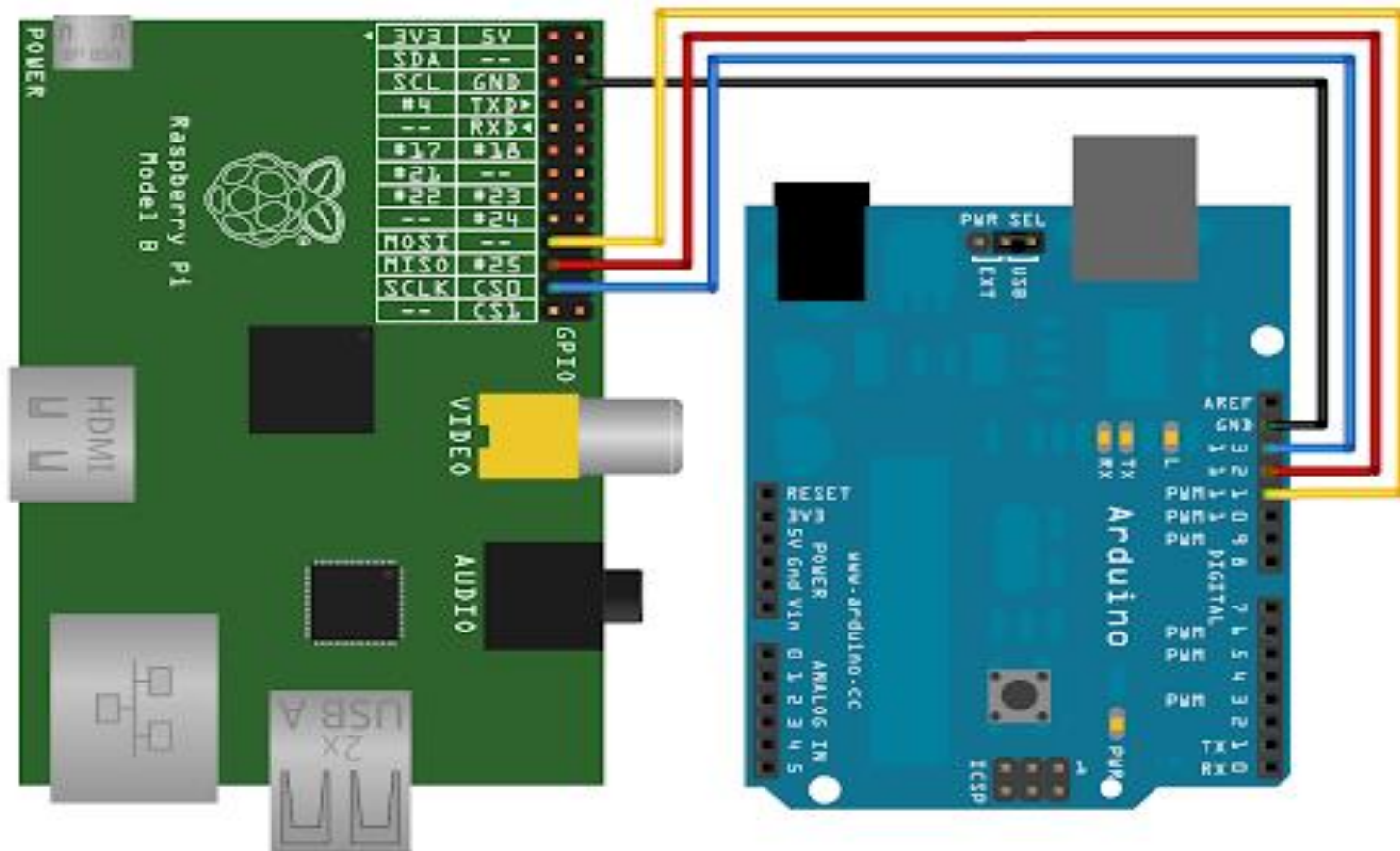


Example-2: Arduino Uno boards connected to Bluetooth module through RS232



Bluetooth Module

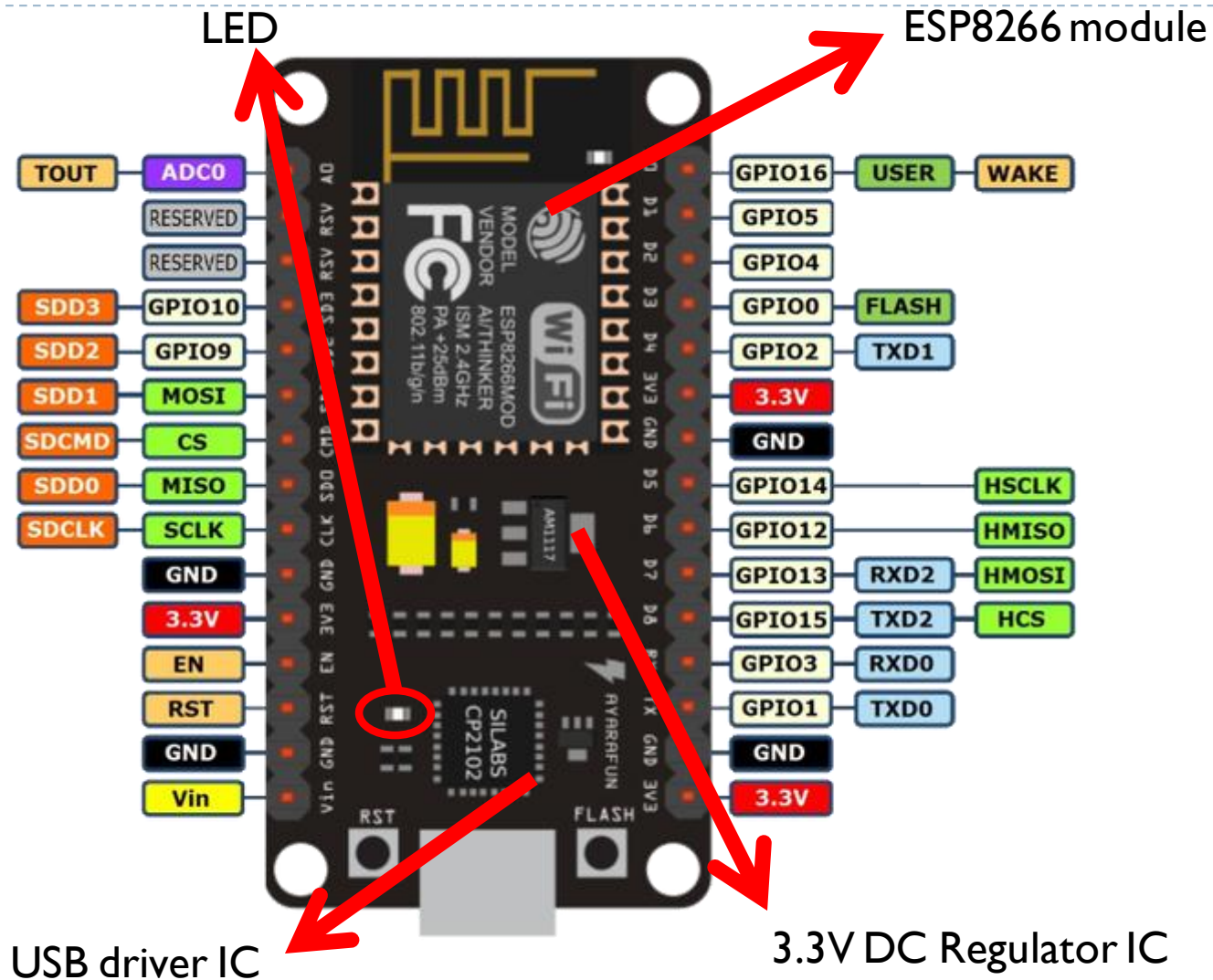
Example-3: Arduino Uno boards connected to Raspberry Pi through SPI



Raspberry Pi

Arduino Uno

Wi-Fi module (ESP2866)



Steps for Arduino IDE

1. First you must have your Arduino board and USB cable.
2. Download Arduino IDE.
3. Power on your board.
4. Launch Arduino IDE.
5. Open Arduino project.
6. Select serial port in which USB cable is connected with Arduino board.

Steps for Arduino IDE

Demonstration of Arduino IDE & Arduino Uno development board

- 1. Running a LED blink program on Arduino Uno**
- 2. Scanning of Wifi Signals using ESP8266 development board**

Example: Scanning of Wifi Signals using ESP8266 development board

```
#include "ESP8266WiFi.h"
```

```
void setup() {
```

```
Serial.begin(115200);
```

```
int numberOfNetworks = WiFi.scanNetworks();
```

```
for(int i =0; i<numberOfNetworks; i++){
```

```
    Serial.print("Network name: ");
```

```
    Serial.println(WiFi.SSID(i));
```

```
    Serial.print("Signal strength: ");
```


```
    Serial.println(WiFi.RSSI(i));
```

```
    Serial.println("-----");
```

```
}
```

```
}
```

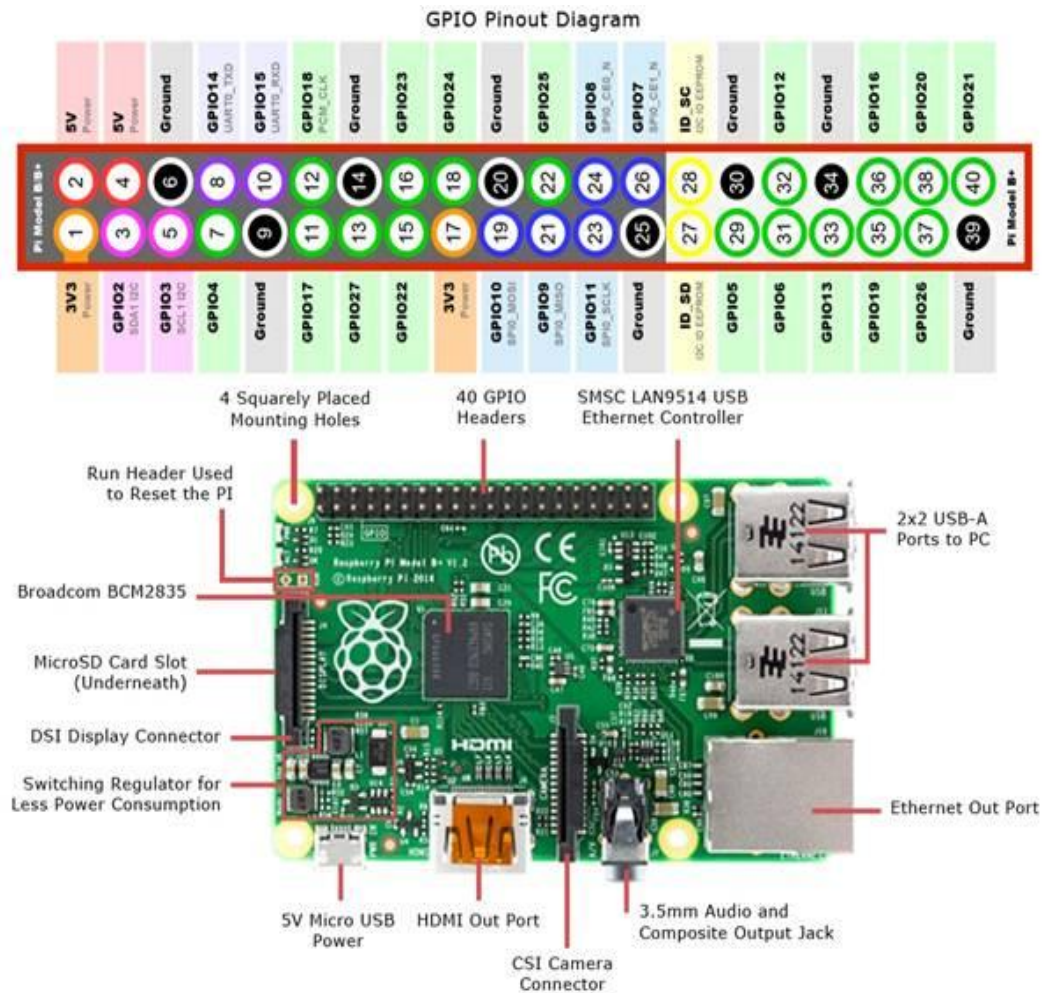
```
void loop() {}
```



```
COM5
[1 leš! Qlâ!  _ #i c!Z,8â '("BX p _0No8$no0âi b piž*(l{lp0oâ f 1 _0 c oâANl1 â žb,0Noi $æDl' 0 no 4' nržŮo c
Network name: MEO-8S
Signal strength: -90
-----
Network name: MEO-8S
Signal strength: -47
-----
Network name: DLink-8A
Signal strength: -90
-----
```


Raspberry Pi

- ▶ Raspberry Pi board comprises RAM, processor, CPU, GPU, Ethernet port, Xbee socket, UART and various interface for other external devices.
- ▶ Essential H/W specification of raspberry pi board mainly include SD card containing Linux OS, keyboard, monitor, video cable optional specification includes USB mouse, USB hub, USB Wi-Fi adapter.



THANK YOU