
Building Smart Things

SHABIR ALI

Outline

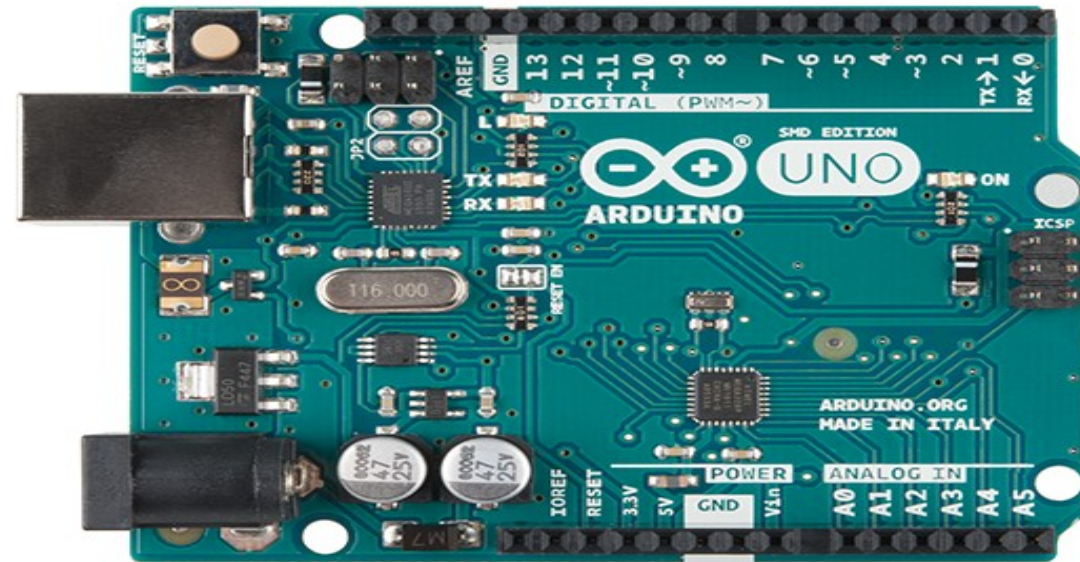
- Sensor development micro controller boards
 - Arduino Uno SMD R3
- Sensors
- Actuators
- Arduino IDE
- Connecting Sensors to Arduino UNO
 - Experiment-1
 - Experiment-2
 - Experiment-3
 - Experiment-4

Sensor development micro-controller boards

- Micro-controller (MCU) board- constrained device (limited computing power)
- The MCU boards used for development known as “dev MCU board”
- How to select MCU visit: <https://dzone.com/articles/how-to-choose-a-microcontroller-for-iot>
- The popular MCU development boards are
 - **Arduino Uno**
 - Jetson TX2 Development Kit
 - i.MX8 Development Kit
 - Thunderboard Sense 2
 - **Raspberry Pi 3 Model B**
 - Ultrahaptics Touch Development Board
 - AudioSmart 2-Mic Development Kit
 - ADALM-PLUTO
 - SensorTile Development Kit
 - Samsung ARTIK 710 Development Kit

Arduino Uno SMD R3

The **Arduino UNO** is an open-source microcontroller board based on the Microchip **ATmega328P** microcontroller and developed by **Arduino.cc**. (around 400 rupees)

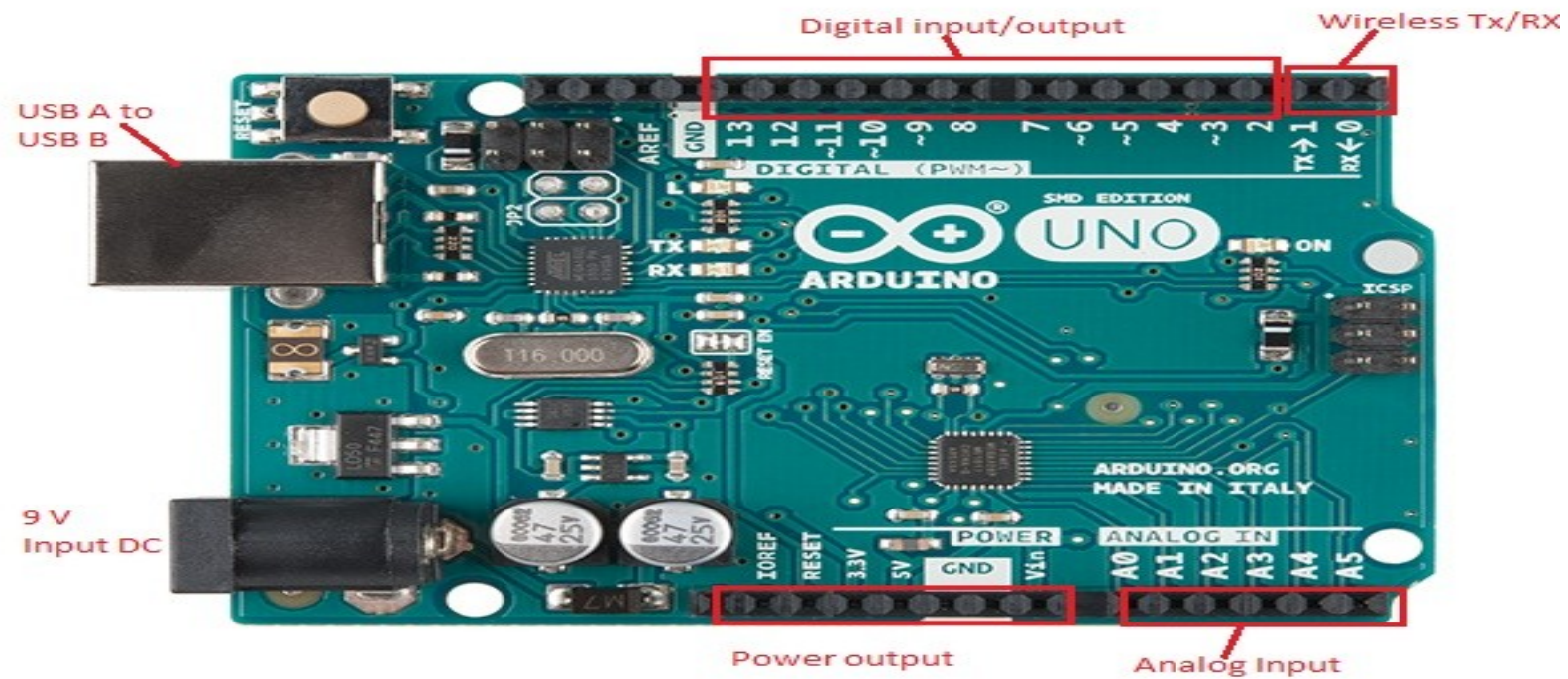


Arduino Uno SMD R3

Technical Specification

- Microcontroller: Microchip ATmega328P [7]
- Operating Voltage: 5 Volts
- Input Voltage: 7 to 20 Volts
- Digital I/O Pins: 14 (of which 6 provide PWM output)
- Analog Input Pins: 6
- DC Current per I/O Pin: 20 mA
- DC Current for 3.3V Pin: 50 mA
- Flash Memory: 32 KB of which 0.5 KB used by bootloader
- SRAM: 2 KB
- EEPROM: 1 KB
- Clock Speed: 16 MHz
- Length: 68.6 mm
- Width: 53.4 mm
- Weight: 25 g

Arduino Uno SMD R3



Arduino Uno SMD R3

General pin functions

LED: There is a built-in LED driven by digital pin 13. When the pin is high value, the LED is on, when the pin is low, it's off.

VIN: The input voltage to the Arduino/Genuino board when it's using an external power source (as opposed to 5 volts from the USB connection or other regulated power source). You can supply voltage through this pin, or, if supplying voltage via the power jack, access it through this pin.

5V: This pin outputs a regulated 5V from the regulator on the board. The board can be supplied with power either from the DC power jack (7 - 20V), the USB connector (5V), or the VIN pin of the board (7-20V). Supplying voltage via the 5V or 3.3V pins bypasses the regulator, and can damage the board.

3V3: A 3.3 volt supply generated by the on-board regulator. Maximum current draw is 50 mA.

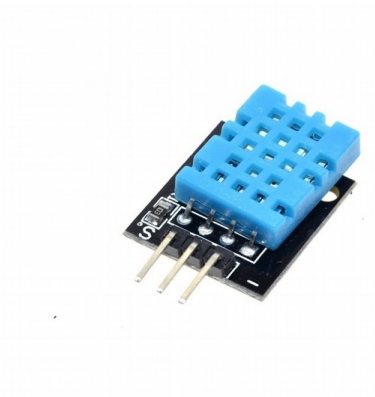
GND: Ground pins.

IOREF: This pin on the Arduino/Genuino board provides the voltage reference with which the microcontroller operates. A properly configured shield can read the IOREF pin voltage and select the appropriate power source or enable voltage translators on the outputs to work with the 5V or 3.3V.

Reset: Typically used to add a reset button to shields which block the one on the board.

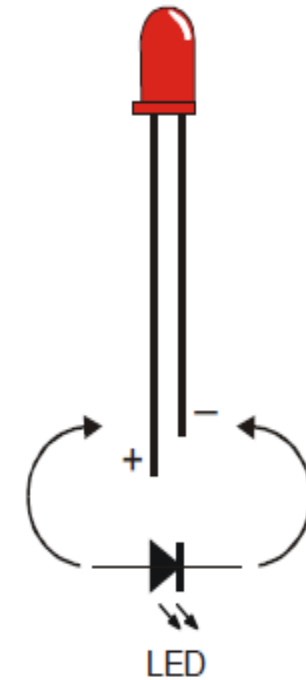
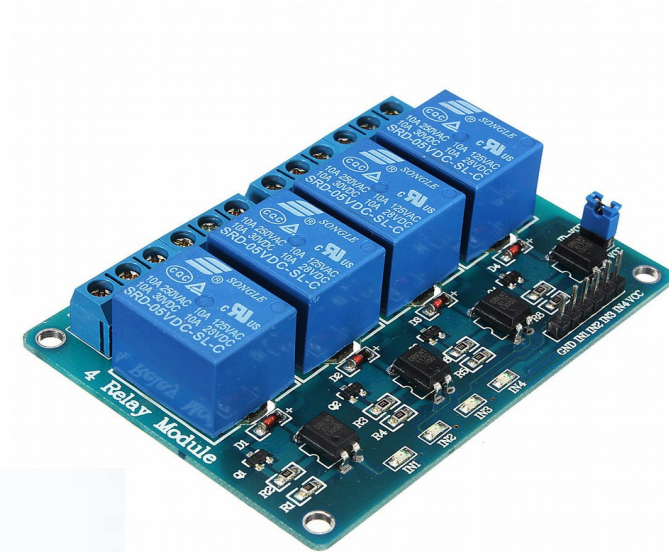
Sensors

- Temperature Sensor (DHT-11)
- Humidity Sensor, Gas sensor (MQ-135)
- Barometric (Pressure) Sensor (BMP-180)
- Gyroscope (GY-521) and Accelerometer (MPU-6050)
- Many More...



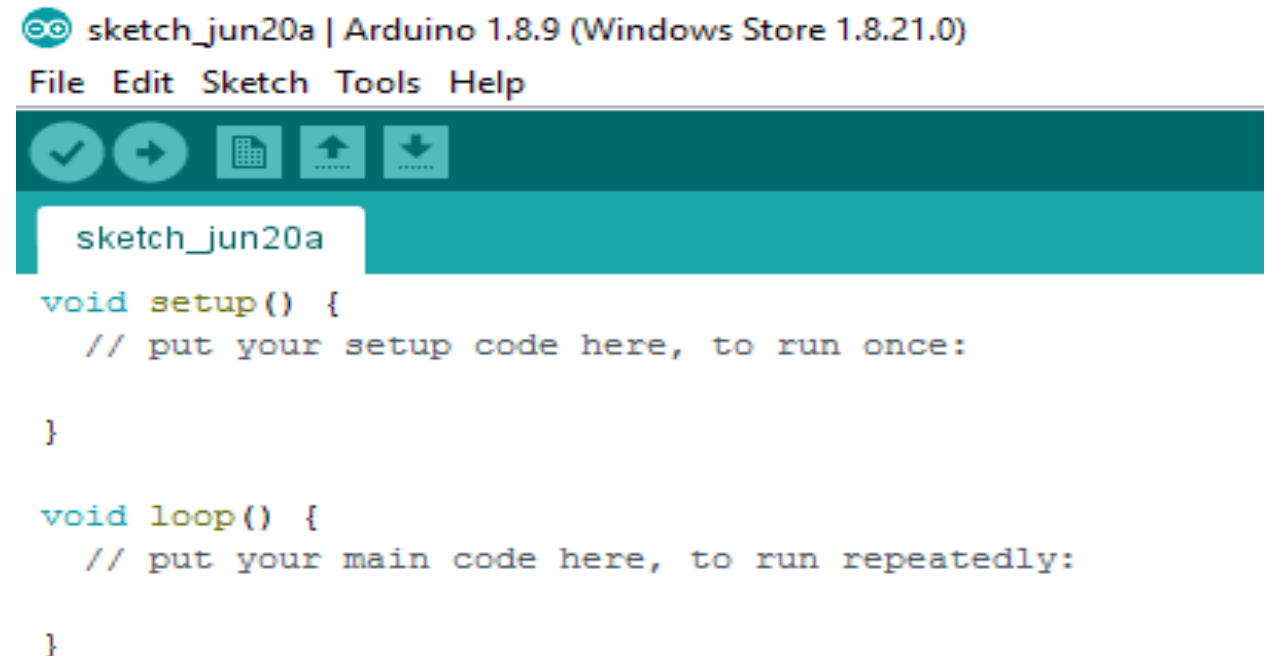
Actuators

- Light-Emitting Diode (LED)
- Relay board
- Motors
- Linear actuator (motion in straight line)



Arduino IDE

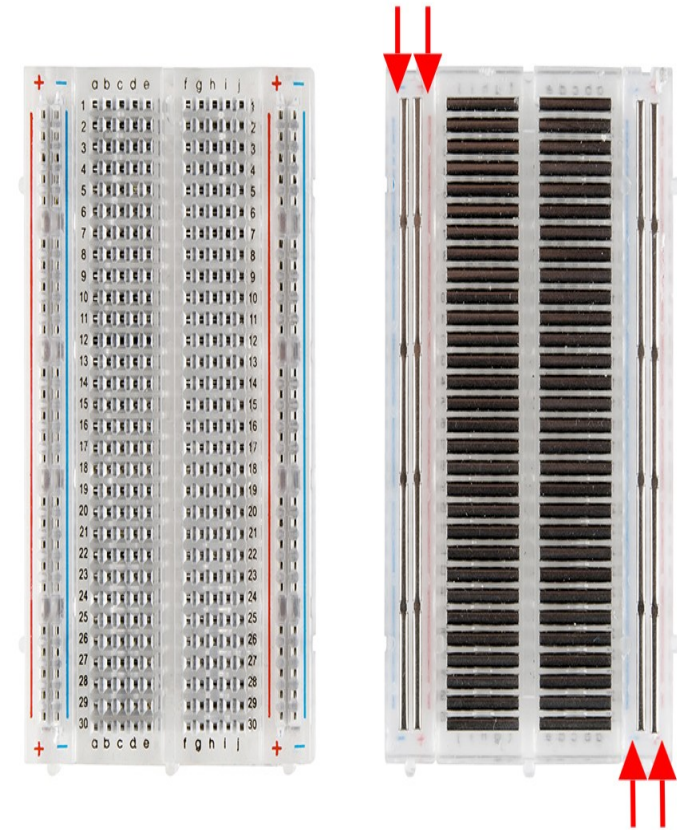
- Download from <http://arduino.cc>
- For Ubuntu VM download 64bit Linux_
- Extract it using command `tar -xvzf filename`
- Moved to arduino directory and run
- `sh install.sh`



Connecting Sensor to Arduino MCU

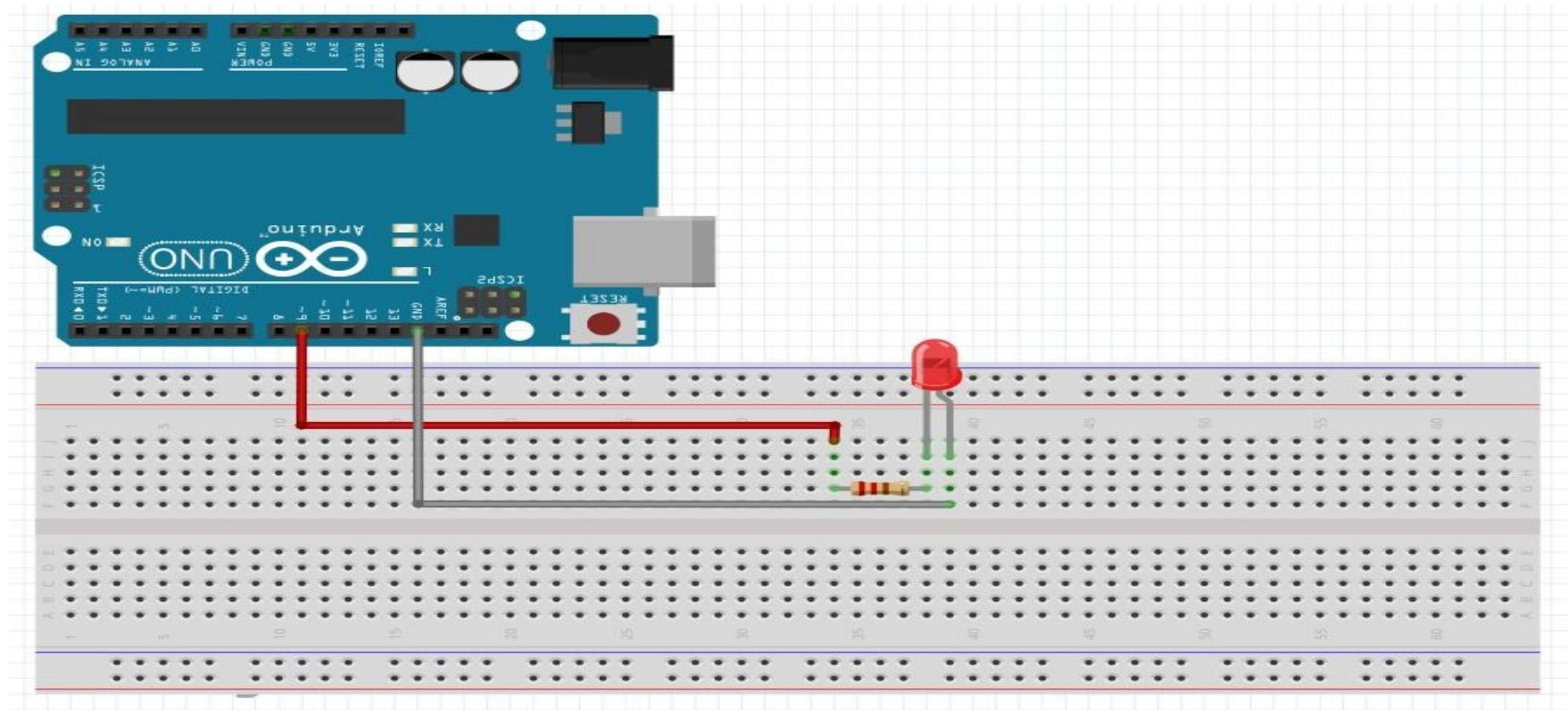
Example 1: Blink LED from Arduino MCU

- Remember 5v---> + and ground---> -ve
- Use breadboard and jumper pin to connect LED
- Connect LED +ve-->pin 13 of Arduino
- Connect ground in Arduino to -ve of LED
- Connect USB A port to machine having Arduino IDE
- In Arduino UNO select tool --> port (/dev/ttyUSB0)
- Select tool--> board Arduino UNO
- In Arduino IDE use open symbol to open basics then blink exan



Connecting Sensor to Arduino MCU

Example 1: Blink LED from Arduino MCU



Connecting Sensor to Arduino MCU

Example 1: Blink LED from Arduino MCU

//Arduino Code for blinking light

```
void setup() {  
    pinMode(LED_BUILTIN, OUTPUT);  
}  
  
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  
}
```

Verify to compile the program and upload the program to arduino UNO MCU

Connecting Sensor to Arduino MCU

Example 2: Blink LED from any digital pin of Arduino MCU

//Arduino Code to change default associated pin

```
Int ledPin=3

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

void loop() {
  digitalWrite(ledPin, HIGH); // turn the LED on (HIGH is the voltage level)
  delay(1000);                // wait for a second
  digitalWrite(ledPin, LOW);  // turn the LED off by making the voltage LOW
  delay(1000);                // wait for a second
}
```

Verify to compile the program and upload the program to arduino UNO MCU

Connecting Sensor to Arduino

MCU **Example 3: LED status at serial monitor from Arduino MCU**

```
int ledPin=2;
// the setup function runs once when you press reset or power the board
void setup() {
  // initialize digital pin LED_BUILTIN as an output.
  pinMode(ledPin, OUTPUT);
  Serial.begin(9600); //baud rate of arduino serial monitor
}

// the loop function runs over and over again forever
void loop() {
  digitalWrite(ledPin, HIGH); // turn the LED on (HIGH is the voltage level)
  Serial.print("High");
  delay(1000);                // wait for a second
  digitalWrite(ledPin, LOW);  // turn the LED off by making the voltage
  LOW
  delay(1000); // wait for a second
  Serial.print("Low");
}
```


Connecting Sensor to Arduino

MCU

Example 4: Read analog signal of sensor (MQ135)

```
#include <MQ135.h>
int sensorValue;
int digitalValue;
int digitalPin=4;
int analogPin=0;
int ledPin=3;
MQ135 gasSensor=MQ135(analogPin);
void setup()
{
  Serial.begin(9600); // sets the serial port to 9600
  pinMode(analogPin, OUTPUT);
  pinMode(digitalPin, INPUT);
  pinMode(ledPin, OUTPUT);
}
```



```
void loop()
{
  sensorValue = analogRead(analogPin); // read analog input
  pin 0
  digitalValue = digitalRead(digitalPin);
  if(sensorValue>70)
  {
    digitalWrite(ledPin, HIGH);
  }
  else
    digitalWrite(ledPin, LOW);
  Serial.println(sensorValue, DEC); // prints the value read
  float ppm = gasSensor.getPPM();
  Serial.println("PPM reading:");
  Serial.print(ppm);
  Serial.println(digitalValue, DEC);
  delay(1000); // wait 100ms for next reading
}
```

Assignment

