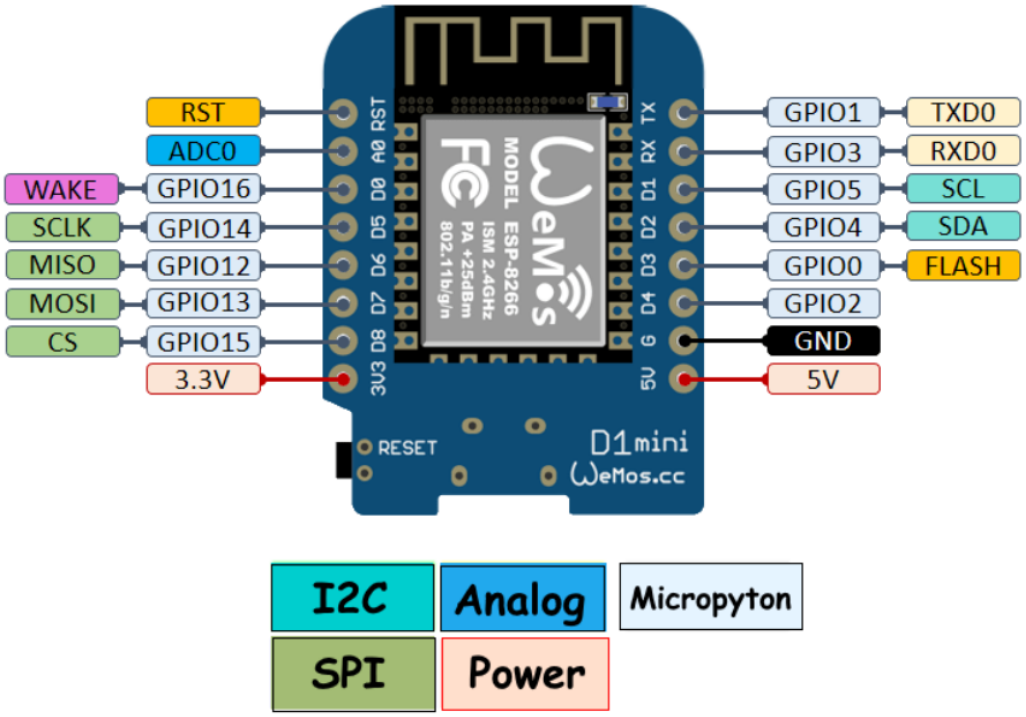
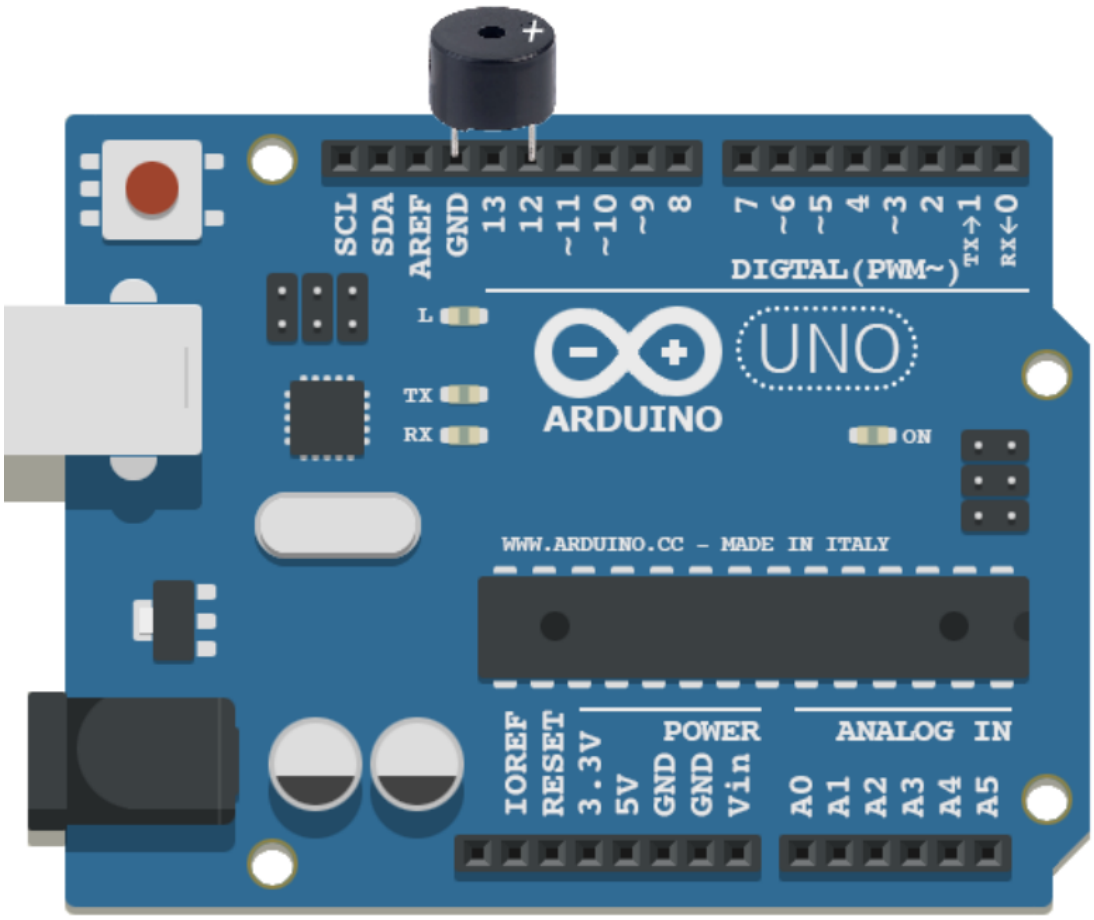
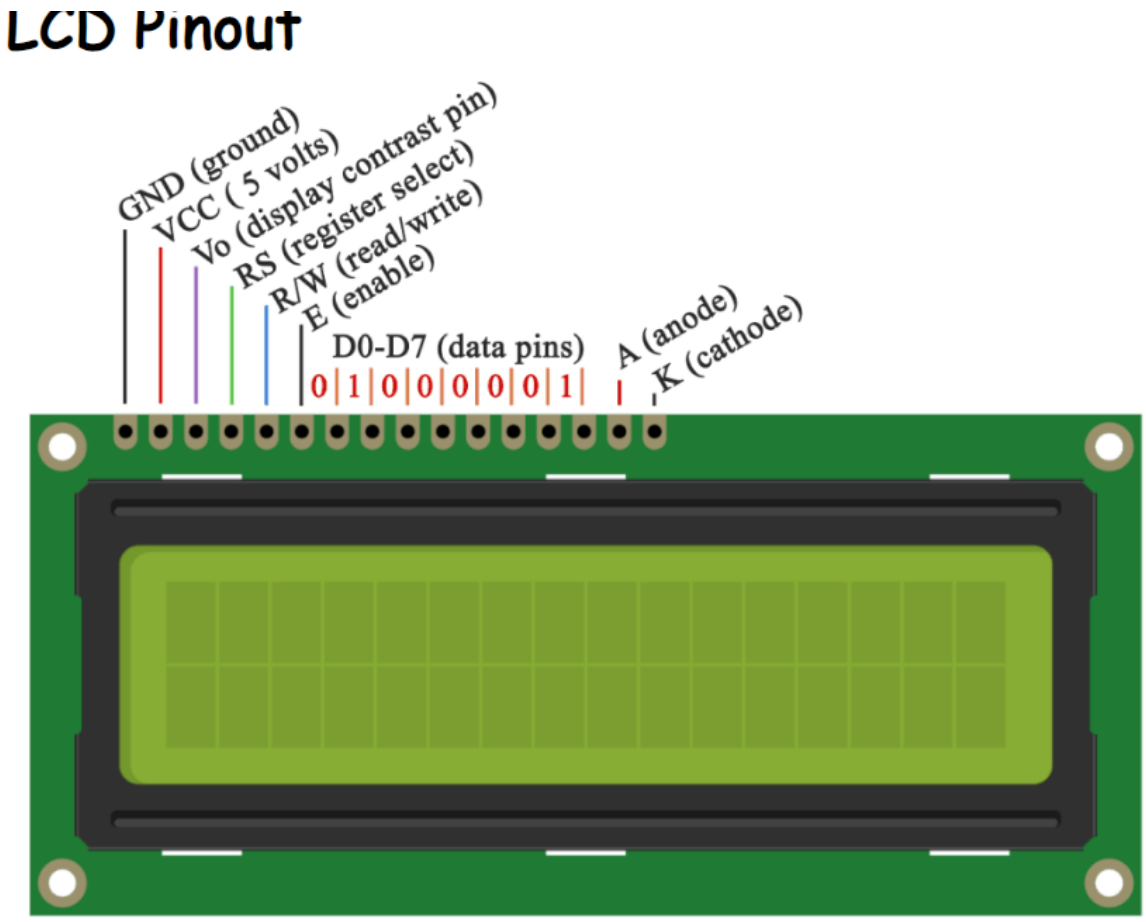
Tools



- **BUZZER**



- **LCD(Liquid Crystal Display.)**

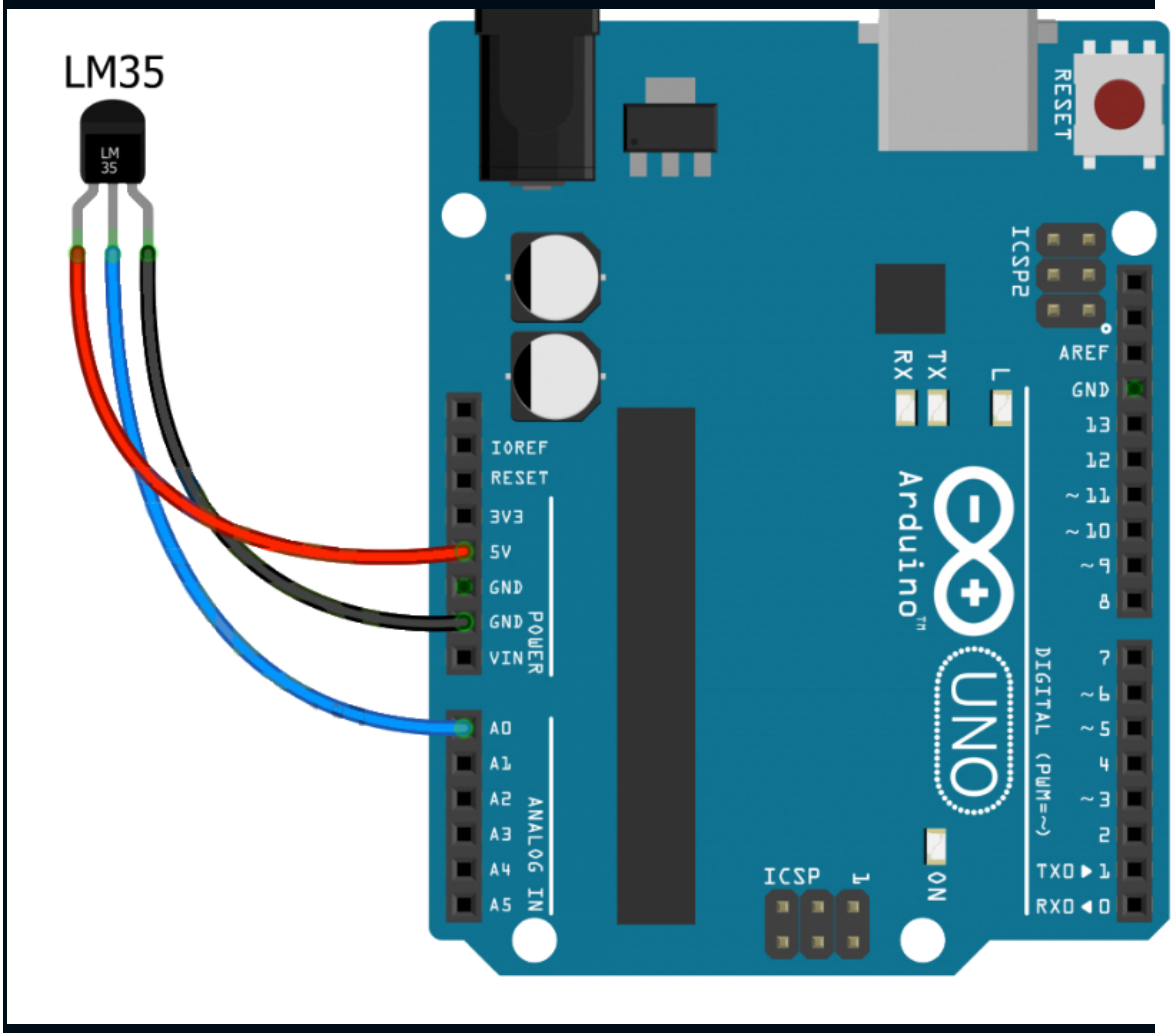


links

https://gist.github.com/irumvanselme/bbcfff03fc5b4690091f0fbbf016270d
https://github.com/izereuwonkunda/embedded_iot_practice
https://github.com/PatrickNiyogitare28/embedded-systems-starter

LCD code

```
#include<Wire.h>
#include<LiquidCrystal_I2C.h>
LiquidCrystal_I2C lcd(0x27, 16 , 2) ;
void setup(){
 lcd.begin();
 lcd.backlight();
```




```
lcd.print("hello");
}
// the loop function runs over and over again forever
void loop(){
//lcd.clear();
//lcd.print("hello");
//delay(500);
//lcd.clear();
}
```

distance_sensor

Download `LiquidCrystal_I2C.h` k

https://github.com/fdebrabander/Arduino-LiquidCrystal-I2C-library.git

```
#include<Wire.h>
#include<LiquidCrystal_I2C.h>

int redPin = 13;
int greenPin = 12;//GPIO 13 corresponds to pin D7. You can check the
pinout.
const int trigPin = 14;
const int echoPin = 16;
long duration, distance;
LiquidCrystal_I2C lcd(0x27, 16, 2);
void setup(){
 lcd.begin();
 lcd.backlight();
// initialize ledPin as an output.
Serial.begin(9600);
pinMode(redPin, OUTPUT);
pinMode(greenPin, OUTPUT);
 pinMode(trigPin, OUTPUT);
  pinMode(echoPin, INPUT);
}
// the loop function runs over and over again forever
void loop(){
 digitalWrite(trigPin, LOW);

  delayMicroseconds(2);

  digitalWrite(trigPin, HIGH);

  delayMicroseconds(10);

  digitalWrite(trigPin, LOW);
```

```
  duration = pulseIn(echoPin, HIGH);

  distance = (duration / 2) / 29.1;



  if (distance >= 100 || distance <= 0){
digitalWrite(greenPin,HIGH);
digitalWrite(redPin,LOW);
   Serial.println("Out of range");
lcd.print(distance);
   }

   else {
digitalWrite(greenPin,LOW);
digitalWrite(redPin,HIGH);
     Serial.print(distance);
     Serial.println(" cm");
   }
   delay(500);
}
```

Wifi

distance_seinsor_wifi(send data)

```
#include<Wire.h>
#include<LiquidCrystal_I2C.h>
#include <ESP8266WiFi.h>
WiFiClient wifiClient;

int redPin = 13;
int greenPin = 12;//GPIO 13 corresponds to pin D7. You can check the
pinout.
const int trigPin = 14;
const int echoPin = 16;
long duration, distance;
LiquidCrystal_I2C lcd(0x27, 16, 2);
void setup(){
```

```
WiFi.begin("RCA-WiFi", "rca@2019");
lcd.begin();
lcd.backlight();
// initialize ledPin as an output.
Serial.begin(9600);
pinMode(redPin, OUTPUT);
pinMode(greenPin, OUTPUT);
 pinMode(trigPin, OUTPUT);
  pinMode(echoPin, INPUT);
}
// the loop function runs over and over again forever
void loop(){
  String host = "192.168.1.68";
  String path = "/iot/";
    int port = 8000;
 digitalWrite(trigPin, LOW);

  delayMicroseconds(2);

  digitalWrite(trigPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigPin, LOW);
  duration = pulseIn(echoPin, HIGH);
  distance = (duration / 2) / 29.1;



  if (distance >= 100 || distance <= 0){
digitalWrite(greenPin,HIGH);
digitalWrite(redPin,LOW);
   Serial.println("Out of range");

   }

   else {
digitalWrite(greenPin,LOW);
digitalWrite(redPin,HIGH);

   }
   lcd.clear();
```

```
   lcd.print("distance");
   lcd.setCursor(0,1);
   lcd.print(distance);
    String request = "POST
/iot?device=RCA0125BGE&distance="+(String)distance+" HTTP/1.1";
  wifiClient.connect(host, port);
  wifiClient.println(request);
  wifiClient.println("Host: "+host);
  wifiClient.println("User-Agent: ESP8266/1.0");
  wifiClient.println("C: ESP8266/1.0");
  wifiClient.println();
  Serial.println("Response: "+wifiClient.readStringUntil('\n'));
   delay(500);
}
```

Temperature

```
#include<Wire.h>
#include<LiquidCrystal_I2C.h>
#include <ESP8266WiFi.h>
WiFiClient wifiClient;

int redPin = 13;
int greenPin = 12;//GPIO 13 corresponds to pin D7. You can check the
pinout.
int buzzerPin=16;
LiquidCrystal_I2C lcd(0x27, 16, 2);
void setup(){
WiFi.begin("RCA-WiFi", "rca@2019");
lcd.begin();
lcd.backlight();
lcd.println("temperature");
// initialize ledPin as an output.
Serial.begin(9600);
pinMode(redPin, OUTPUT);
pinMode(greenPin, OUTPUT);
pinMode(buzzer,OUTPUT);
```

```
lcd.clear();
}
// the loop function runs over and over again forever
void loop(){

    int rawData = analogRead(A0);
    float vcc = 5.0;
    float voltage = rawData * (vcc / 1024.0);
    float temperature = voltage * 100;
//    float vcc = 3.3;
//    float millivolts = rawData * (vcc / 1024.0);
//    float temperature = millivolts * 100;
//    float offset = 3.0;
//    temperature = temperature-offset;
      if (temperature == 35){
    digitalWrite(greenPin,LOW);
    digitalWrite(redPin,HIGH);
    digitalWrite(buzzer.HIGH);    delay(500);

    digitalWrite(buzzer.HIGH);
   String host = "insecure.benax.rw";
  String path = "/iot/";
  int port = 80;
 // String request = "POST
/insecure.benax.rw/iot/?device=RCA0125BGE&temperature="+(String)temp
erature+" HTTP/1.1";
  String mData="";
  String device = "kerie";
  mData = "device="+device+"&distance="+(String)temperatire;
  sendData(port, host,path,mData);
      }
     else {
    digitalWrite(greenPin,HIGH);
    digitalWrite(redPin,LOW);
    digitalWrite(buzzer.LOW);
       }
    lcd.clear();
    lcd.setCursor(0,1);
    lcd.print(temperature);
    delay(1000);
```

```
}

void sendData(const int httpPort, const char* host,const char*
filepath , String data){
  wifiClient.connect(host, httpPort);
  wifiClient.println("POST "+(String)filepath+" HTTP/1.1");
  wifiClient.println("Host: " + (String)host);
  wifiClient.println("User-Agent: ESP8266/1.0");
  wifiClient.println("Content-Type:
application/x-www-form-urlencoded");
  wifiClient.println("Content-Length: " +(String)data.length());
  wifiClient.println();
  wifiClient.print(data);
  Serial.println("Response: " + wifiClient.readStringUntil('\n'));

}
```