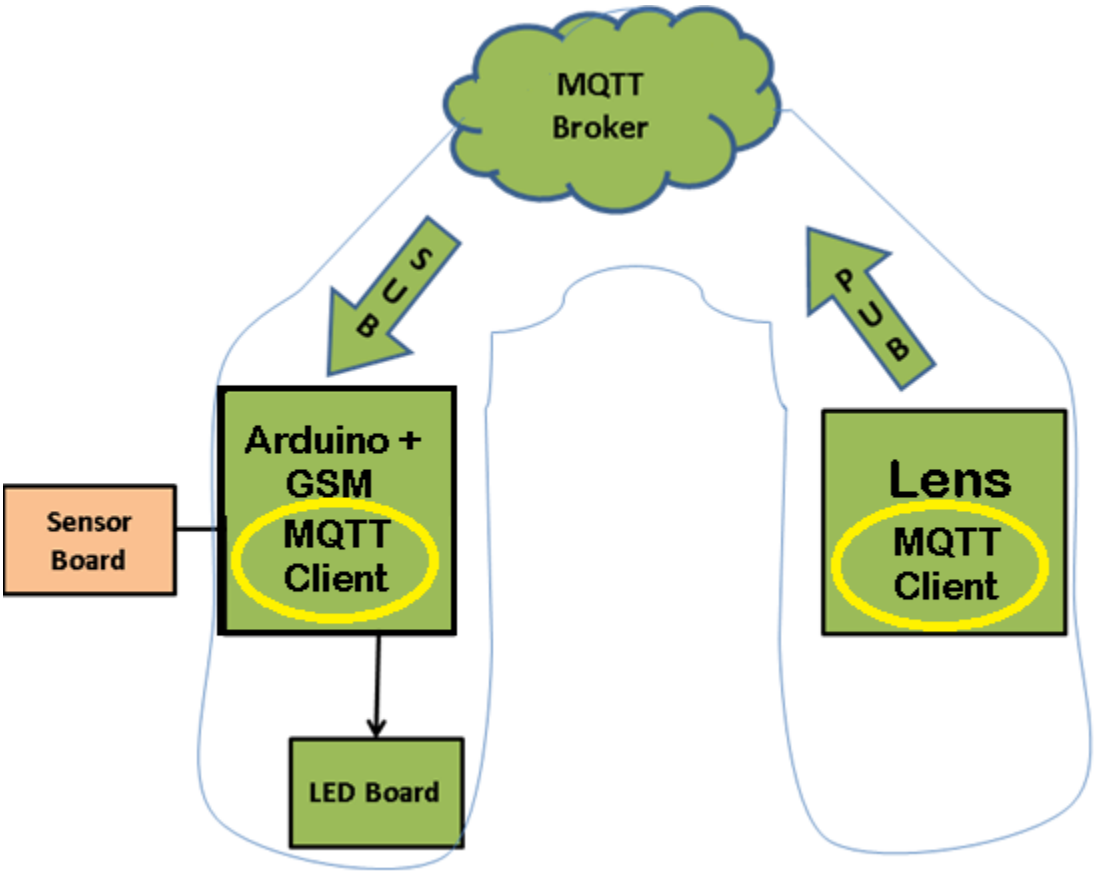
User Manual

MQTT Subscribe

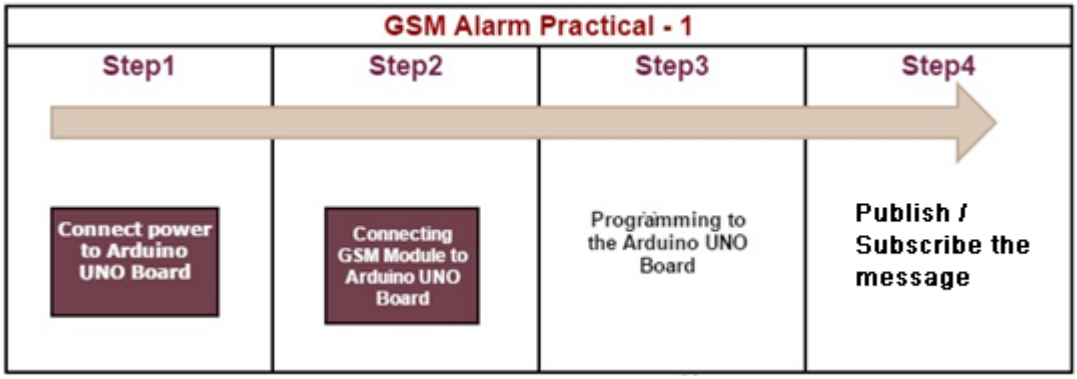
Practical's Objective:

Send commands to arduino board to on/off the LED.

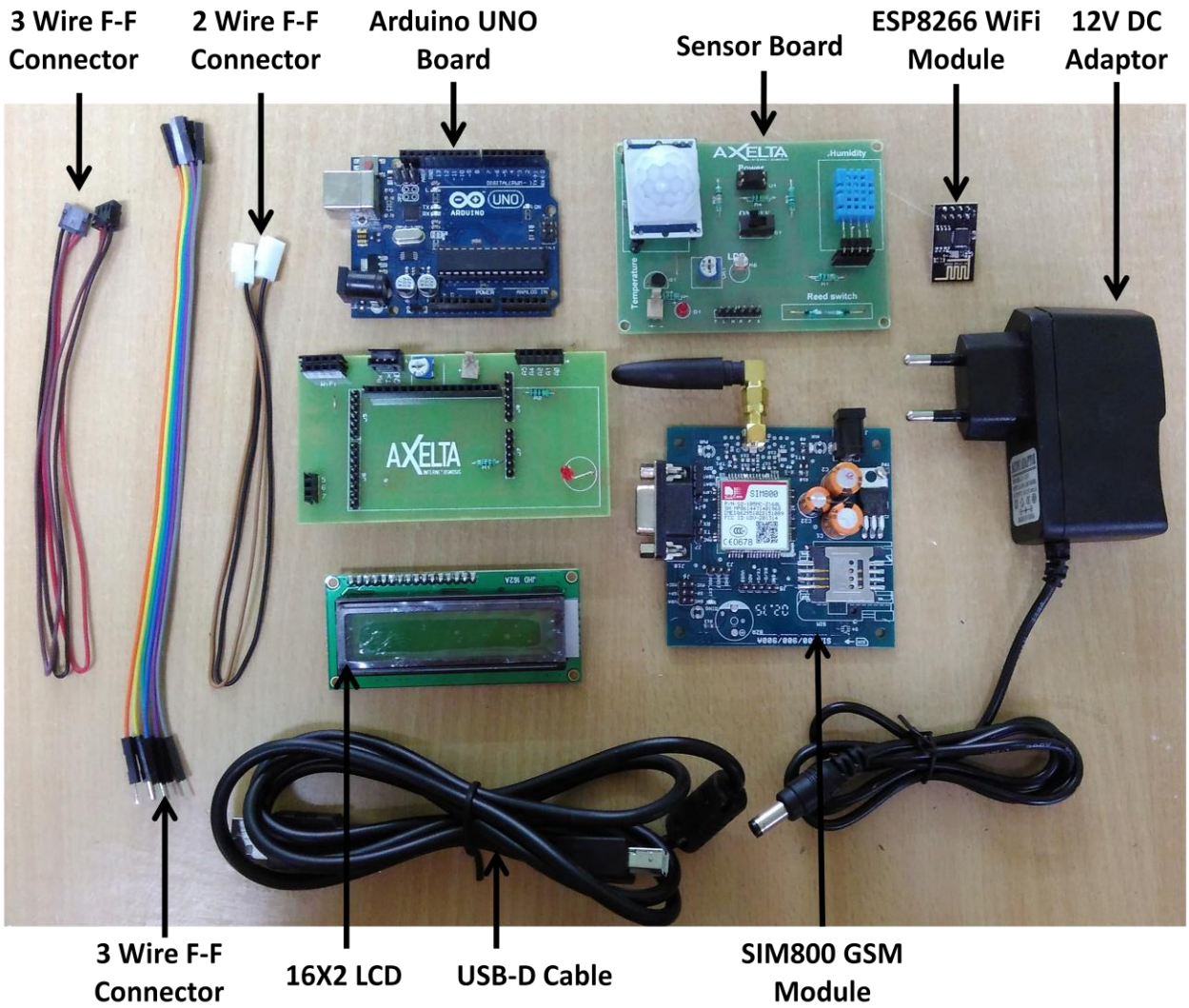
1. **End-End IoT Flow Diagram:**



2. Sensors Data Posting Using JSON To Do:



3. Hardware requirements:

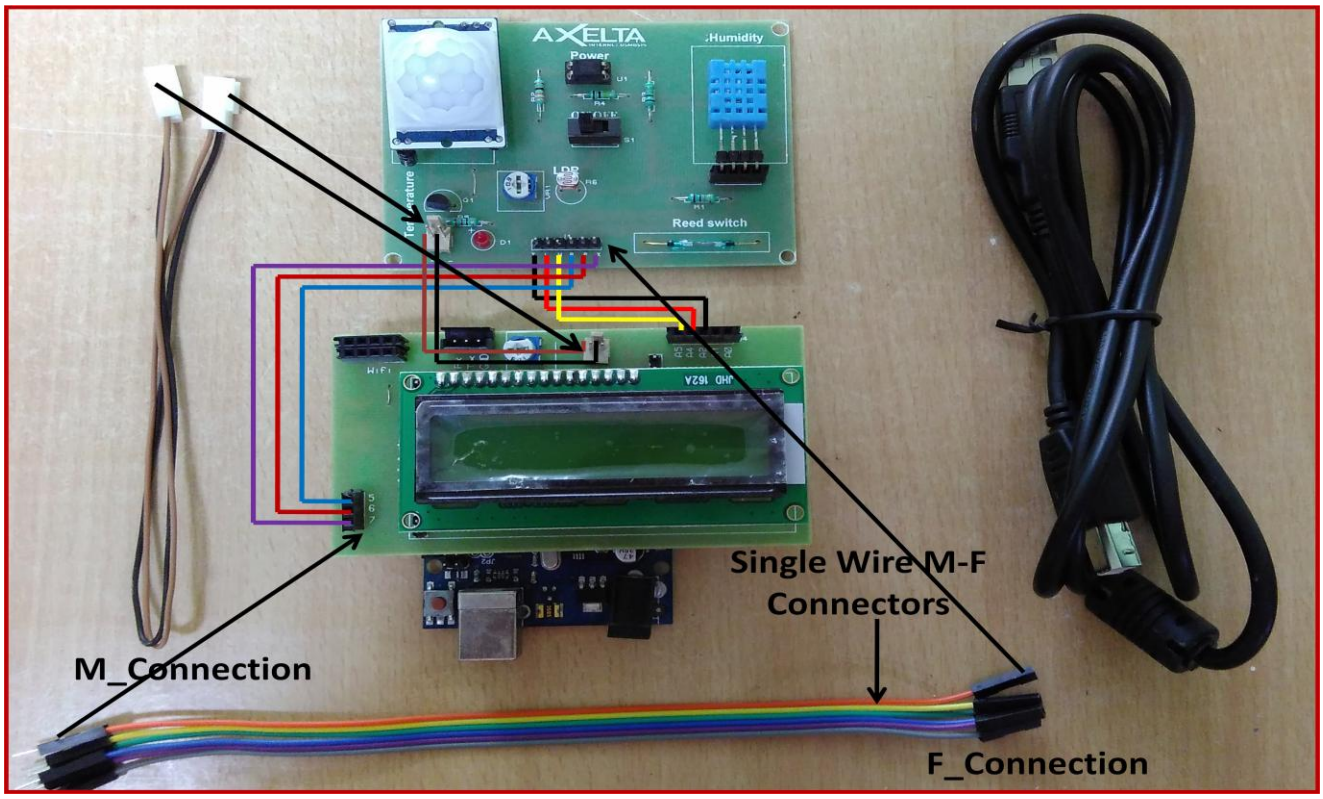


Software Requirement:

- Arduino IDE

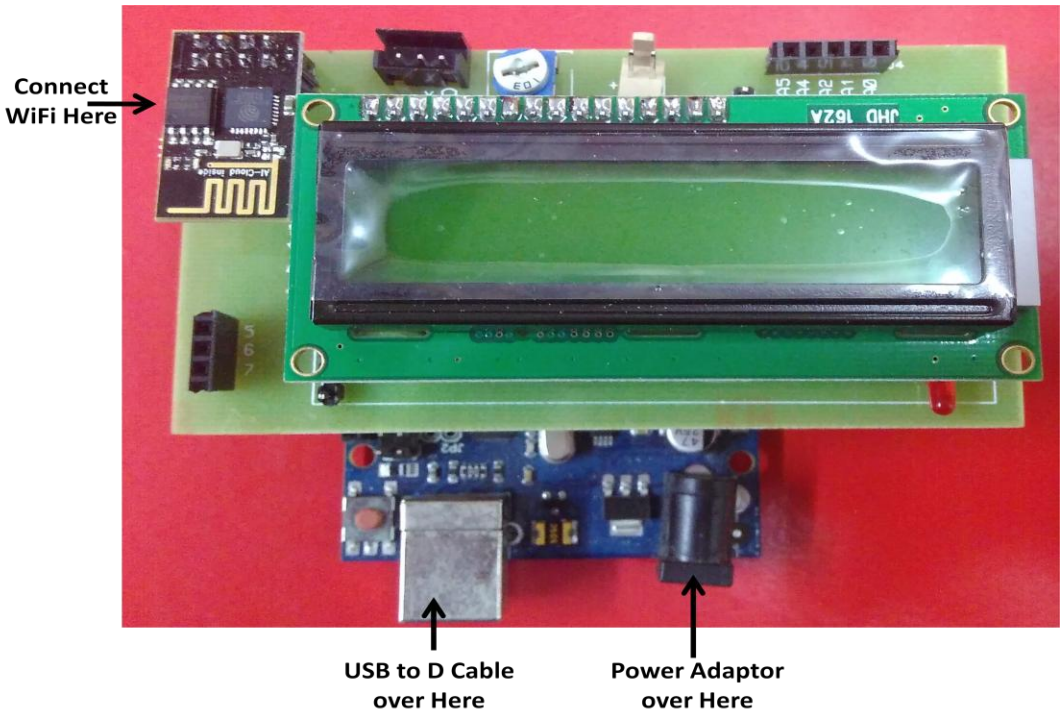
Arduino UNO Board Connections with Sensor Board:

Don’t change the sensor connections. They will also remain same.



Note:

1) **Don’t connect WiFi Module ESP8266 to the J9 on LCD Shieldunless you give12V Supply to the Arduino Board.**
2) After plugging ESP8266 as shown below connect your PC to Arduino Board via USB Cable.

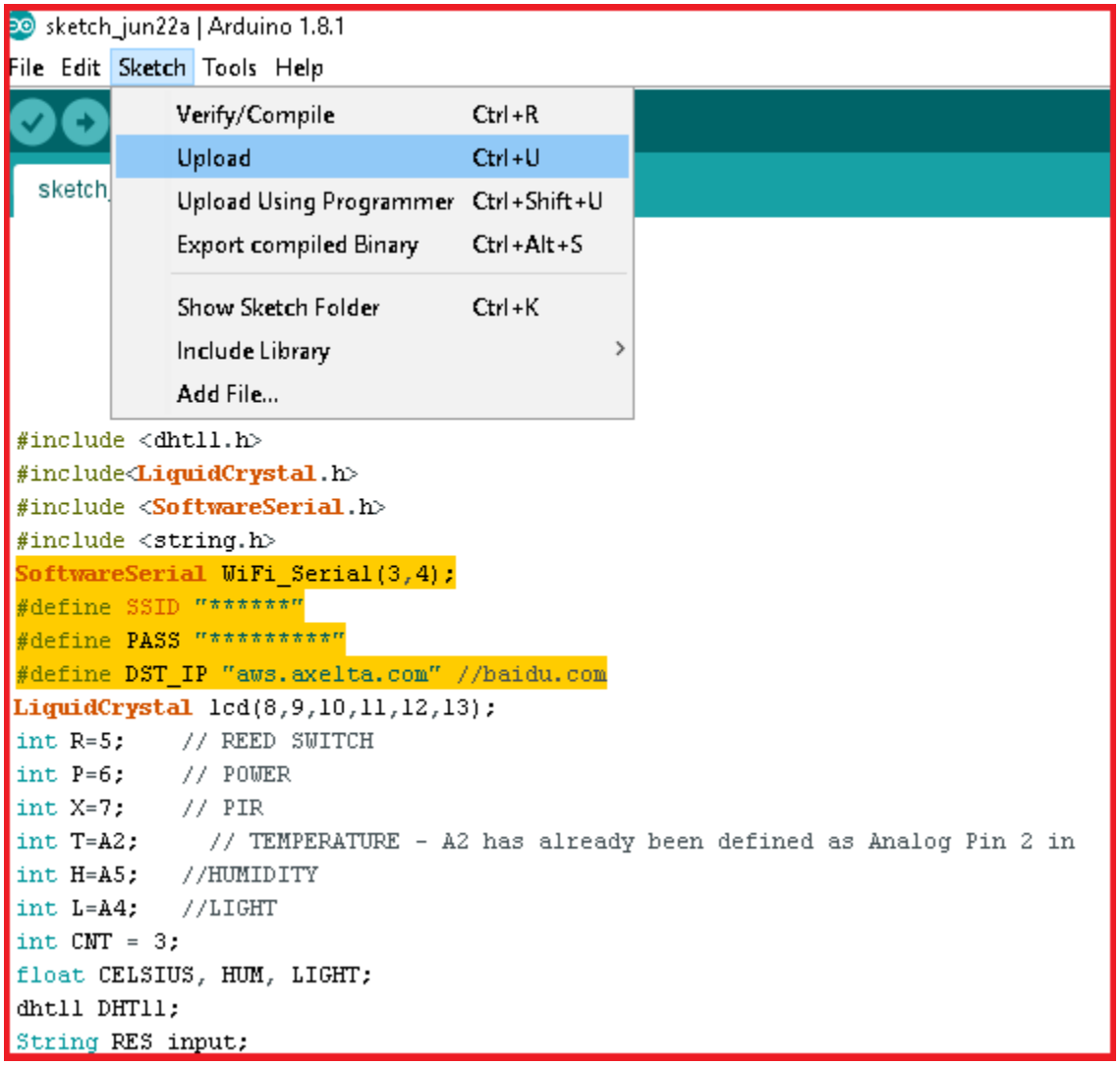


3) Make sure that all the Hardware connections are proper as explained above.

Programming:

5) You can find code by the link:
6) https://drive.google.com/open?id=1YgWxG_xlyApDU5lkU-scWzWsV_wJRmUb
7) Go to **File** menu, and Click on **save**, give a File name and click **ok**.

8) Copy, paste & upload the code in Arduino board



If it is uploaded successfully then you can see its output on Serial Monitor. But before that Open the serial monitor& set correct **baud rate as 9600.**

A simple 'C' program in Arduino.

/*Here we are Sending data to web server using WIFI/GPRS modem and we are interfacing WiFi modem using Uart */

```
#include <WiFiEsp.h>
#include <WiFiEspClient.h>
//#include <WiFiEspUdp.h>
#include "SoftwareSerial.h"
#include <PubSubClient.h>

char server[] = "osmosis.axelta.com";
char ssid[] = "Axelta";          // your network SSID (name)
char pass[] = "Axelta140218";          // your network password
int status = WL_IDLE_STATUS;   // the Wifi radio's status
char* subtopic = "LEDtopic";   /*MQTT Subscribe topic*/
const int A = A3;
String clientName1;
char message_buff[10];

// Initialize the Ethernet client object
WiFiEspClient espClient;

PubSubClient client(espClient);

SoftwareSerial soft(3, 4); // RX, TX
void setup() {
  // initialize serial for debugging
  Serial.begin(9600);
  // initialize serial for ESP module
  soft.begin(9600);
  // initialize ESP module
  WiFi.init(&soft);
  pinMode(A, OUTPUT);
  digitalWrite(A, LOW);

  // check for the presence of the shield
  if (WiFi.status() == WL_NO_SHIELD) {
    Serial.println("WiFi shield not present");
```

```
    // don't continue
    while (true);
  }

  // attempt to connect to WiFi network
  while ( status != WL_CONNECTED) {
    Serial.print("Attempting to connect to WPA SSID: ");
    Serial.println(ssid);
    // Connect to WPA/WPA2 network
    status = WiFi.begin(ssid, pass);
  }

  // you're connected now, so print out the data
  Serial.println("You're connected to the network");

  //connect to MQTT server
  client.setServer(server, 1883);
  client.setCallback(callback);
}

//print any message received for subscribed topic
void callback(char* topic, byte* payload, unsigned int length) {

  int i = 0;
  for (int i = 0; i < length; i++) {
    message_buff[i] = payload[i];
  }
  //message_buff[i] = '\0';
  //for (int i=0;i<(sizeof(message_buff));i++)
  String message;
  for (int i=0;i<length;i++)
  {
    message += String(message_buff[i]);
  }
  Serial.println("stringformed: " +message);

    if (message == "on")
    {
      Serial.println("led on");
      digitalWrite(A, HIGH);
    }
    else if (message == "off")
    {
      Serial.println("led off ");
      digitalWrite(A,LOW);
    }
}
```

```
void loop() {
  // put your main code here, to run repeatedly:
  if (!client.connected()) {
    reconnect();
  }
  //Serial.println((String(message_buff)));


  //client.subscribe(subtopic);
  client.loop();
}


void reconnect() {
  // Loop until we're reconnected
  while (!client.connected()) {
    Serial.print("Attempting MQTT connection...");
    // Attempt to connect, just a name to identify the client
    if (client.connect("arduinoClient")) {
      Serial.println("connected");
      // Once connected, publish an announcement...
      client.publish("command", "hello world");
      // ... and resubscribe
      client.subscribe("presence");
    } else {
      Serial.print("failed, rc=");
      Serial.print(client.state());
      Serial.println(" try again in 5 seconds");
      // Wait 5 seconds before retrying
      delay(5000);
    }
  }
}
```