

SPRINT 3

INITIALIZING HARDWARE

Team ID	PNT2022TMID52810
Project Name	Project - Signs with smart connectivity for Better road safety

Integrate hardware that can connect to cloud services and modify output based on output processing.

```
#include <string>
#include <LiquidCrystal_I2C.h>
#include <WiFi.h>
#include "ArduinoJson.h"
#include <HTTPClient.h>
LiquidCrystal_I2C lcd(0x27,20,4);
const char* ssid = "harrish";
const char* password = "endhapassword";
String payload;
void setup() {
  // put your setup code here, to run once:
  Serial.begin(9600);
  Serial.println();
  Serial.print("Connecting to ");
  Serial.println(ssid);

  WiFi.begin(ssid, password);

  while (WiFi.status() != WL_CONNECTED) {
    delay(50);
    Serial.print(".");
  }

  Serial.println("");
  Serial.println("WiFi connected");
  Serial.println("IP address: ");
  Serial.println(WiFi.localIP());
  lcd.begin();
  lcd.backlight();
}
void link(void) {
  HTTPClient http;
  String url="https://node-red-wrxia-2022-11-13.eu-gb.mybluemix.net/data";
  //Serial.print(url);
  Serial.print("Making a request");
  http.begin(url.c_str()); //Specify the URL and certificate
  http.setFollowRedirects(HTTPC_STRICT_FOLLOW_REDIRECTS);
  int httpCode = http.GET();
  if (httpCode > 0) { //Check for the ing code
    payload = http.getString();
  }
}
```

```

        Serial.println(httpCode);
        Serial.println(payload);
    }
    else {
        Serial.println("Error on HTTP request");
    }
}
http.end();
}
void loop() {
    // put your main code here, to run repeatedly:
    link();

    //JSON
    Serial.println("Parsing start: ");

    //char JSONMessage[] = payload; //Original message

    StaticJsonBuffer<1000> JSONBuffer;           //Memory pool
    JsonObject& parsed = JSONBuffer.parseObject(payload); //Parse message

    if (!parsed.success()) { //Check for errors in parsing

        Serial.println("Parsing failed");
        delay(5000);
        ;
    }

    const char * sensorType = parsed["location"]; //Get sensor type value
    const char * sensorType1 = parsed["weather_report"];
    const char * sensorType2 = parsed["wea_alt"];
    const char * sensorType3 = parsed["schl_tmg"];
    const char * sensorType4 = parsed["spd_alt"];
    int value = parsed["humidity"]; //Get value of sensor measurement
    double value1 = parsed["temperature"];
    int value2 = parsed["pressure"];
    Serial.print("Location: ");
    Serial.println(sensorType);
    Serial.print("Temperature: ");
    Serial.println(value1);
    Serial.print("Pressure: ");
    Serial.println(value2);
    Serial.print("Humidity: ");
    Serial.println(value);
    Serial.print("Weather Report: ");
    Serial.println(sensorType1);
    Serial.print("Message: ");
    Serial.println(sensorType2);
    Serial.print("Timing: ");
    Serial.println(sensorType3);
    Serial.print("Speed Limit: ");

```

```
Serial.println(sensorType4);
Serial.println();
lcd.clear();
lcd.setCursor(0, 0);
lcd.print("Welcome IBM IOT");
lcd.setCursor(0, 1);
lcd.print("Loc: ");
lcd.print(sensorType);
delay(1500);
lcd.clear();
lcd.setCursor(0, 0);
lcd.print("Temp: ");
lcd.print(value1);
lcd.print((char)223);
lcd.print("C");
lcd.setCursor(0, 1);
lcd.print("Hum: ");
lcd.print(value);
lcd.print("%");
delay(1500);
lcd.clear();
lcd.setCursor(0, 0);
lcd.print("Wea: ");
lcd.print(sensorType1);
lcd.setCursor(0, 1);
lcd.print("Time: ");
lcd.print(sensorType3);
delay(1500);
lcd.clear();
lcd.setCursor(0, 0);
lcd.print(sensorType2);
lcd.setCursor(0, 1);
lcd.print(sensorType4);
}
```

HARDWARE

The output from esp32 after fetching location details is shown in figure 9.2.



Fig 9.2 LCD monitor output 1

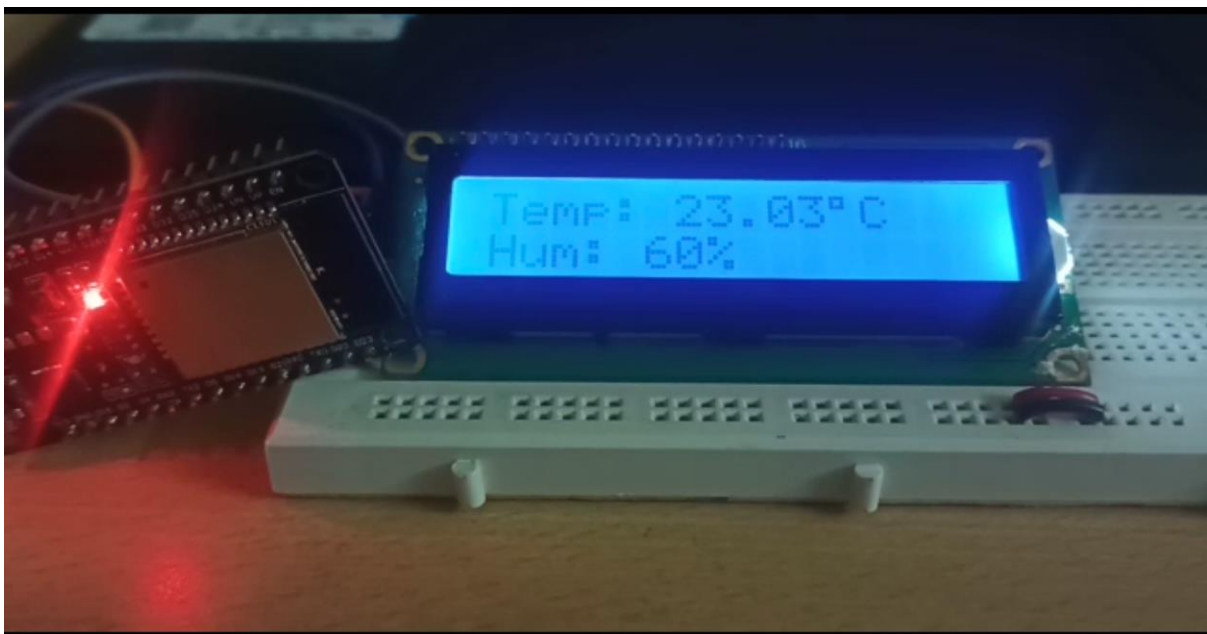


Fig 9.3 LCD monitor output 2

The output from esp32 after fetching temperature and humidity details is shown in figure 9.3.



Fig 9.4 LCD monitor output 3

The output from esp32 after fetching weather and time details details is shown in figure 9.4.



Fig 9.5 LCD monitor output 4

The output from esp32 after fetching weather and time details is shown in figure 9.5.