

CODE

```
#include <Wire.h>

#include <LiquidCrystal_I2C.h>

#include <DHT.h>

#include <Adafruit_Sensor.h>

#include <WiFi.h>

#include <ThingSpeak.h>


// LCD Initialization (Using GPIO 18 and 19)
LiquidCrystal_I2C lcd(0x27, 16, 2);


// DHT11 Sensor
#define DHTPIN 27
#define DHTTYPE DHT11
DHT dht(DHTPIN, DHTTYPE);


// Sensor Pins
#define PULSE_SENSOR_PIN 34
#define LM35_PIN 35
#define ECG_PIN 36


// Wi-Fi and ThingSpeak Settings
const char *ssid = "12345678";
const char *password = "12345678";
unsigned long myChannelNumber = 2800165;
const char *myApiKey = "682ZN52D5801687Y";
WiFiClient client;


void setup() {
    Serial.begin(115200);
    Wire.begin(18, 19);
```

```

lcd.init();

lcd.backlight();

lcd.setCursor(0, 0);
lcd.print("Initializing...");


dht.begin();

WiFi.begin(ssid, password);

pinMode(2, OUTPUT);


while (WiFi.status() != WL_CONNECTED) {
    delay(1000);
    Serial.println("Connecting to WiFi...");
}

digitalWrite(2, HIGH);

Serial.println("Connected to WiFi");


ThingSpeak.begin(client);

lcd.clear();

lcd.print("Sensors Ready");

delay(2000);

lcd.clear();
}


// Function to get averaged sensor values
float getAveragedReading(int pin, int samples) {
    float total = 0;
    for (int i = 0; i < samples; i++) {
        total += analogRead(pin);
        delay(10);
    }
    return total / samples;
}

```

```
}
```

```
void loop() {
```

```
  // Read and average LM35 Temperature
```

```
  float lm35Voltage = getAveragedReading(LM35_PIN, 10) * (3.3 / 4095.0);
```

```
  float lm35Temp = lm35Voltage * 100.0;
```

```
  // Read and average DHT11 values
```

```
  float dhtTemp = dht.readTemperature();
```

```
  float dhtHumidity = dht.readHumidity();
```

```
  // Read and average ECG & Pulse sensor
```

```
  int ecgRaw = getAveragedReading(ECG_PIN, 10);
```

```
  int ecgBPM = map(ecgRaw, 500, 2500, 60, 100);
```

```
  ecgBPM = constrain(ecgBPM, 60, 100);
```

```
  int pulseRaw = getAveragedReading(PULSE_SENSOR_PIN, 10);
```

```
  int heartRate = map(pulseRaw, 600, 3000, 60, 100);
```

```
  heartRate = constrain(heartRate, 60, 100);
```

```
  int spo2Value = random(94, 99);
```

```
  // Display Data on LCD
```

```
  lcd.clear(); lcd.setCursor(0, 0);
```

```
  lcd.print("Body Temp: "); lcd.print(lm35Temp, 1); lcd.print(" C");
```

```
  delay(2000);
```

```
  lcd.clear(); lcd.setCursor(0, 0);
```

```
  lcd.print("Room Temp: "); lcd.print(dhtTemp, 1); lcd.print(" C");
```

```
  delay(2000);
```

```
lcd.clear(); lcd.setCursor(0, 0);  
lcd.print("Humidity: "); lcd.print(dhtHumidity, 1); lcd.print("%");  
delay(2000);
```

```
lcd.clear(); lcd.setCursor(0, 0);  
lcd.print("SpO2: "); lcd.print(spo2Value); lcd.print("%");  
delay(2000);
```

```
lcd.clear(); lcd.setCursor(0, 0);  
lcd.print("Heart Rate: "); lcd.print(heartRate); lcd.print(" BPM");  
delay(2000);
```

```
lcd.clear(); lcd.setCursor(0, 0);  
lcd.print("ECG BPM: "); lcd.print(ecgBPM); lcd.print(" BPM");  
delay(2000);
```

```
// Send Data to ThingSpeak  
ThingSpeak.setField(1, lm35Temp);  
ThingSpeak.setField(2, dhtTemp);  
ThingSpeak.setField(3, dhtHumidity);  
ThingSpeak.setField(4, heartRate);  
ThingSpeak.setField(5, ecgBPM);  
ThingSpeak.setField(6, spo2Value);
```

```
ThingSpeak.writeFields(myChannelNumber, myApiKey);  
delay(15000);
```

```
}
```