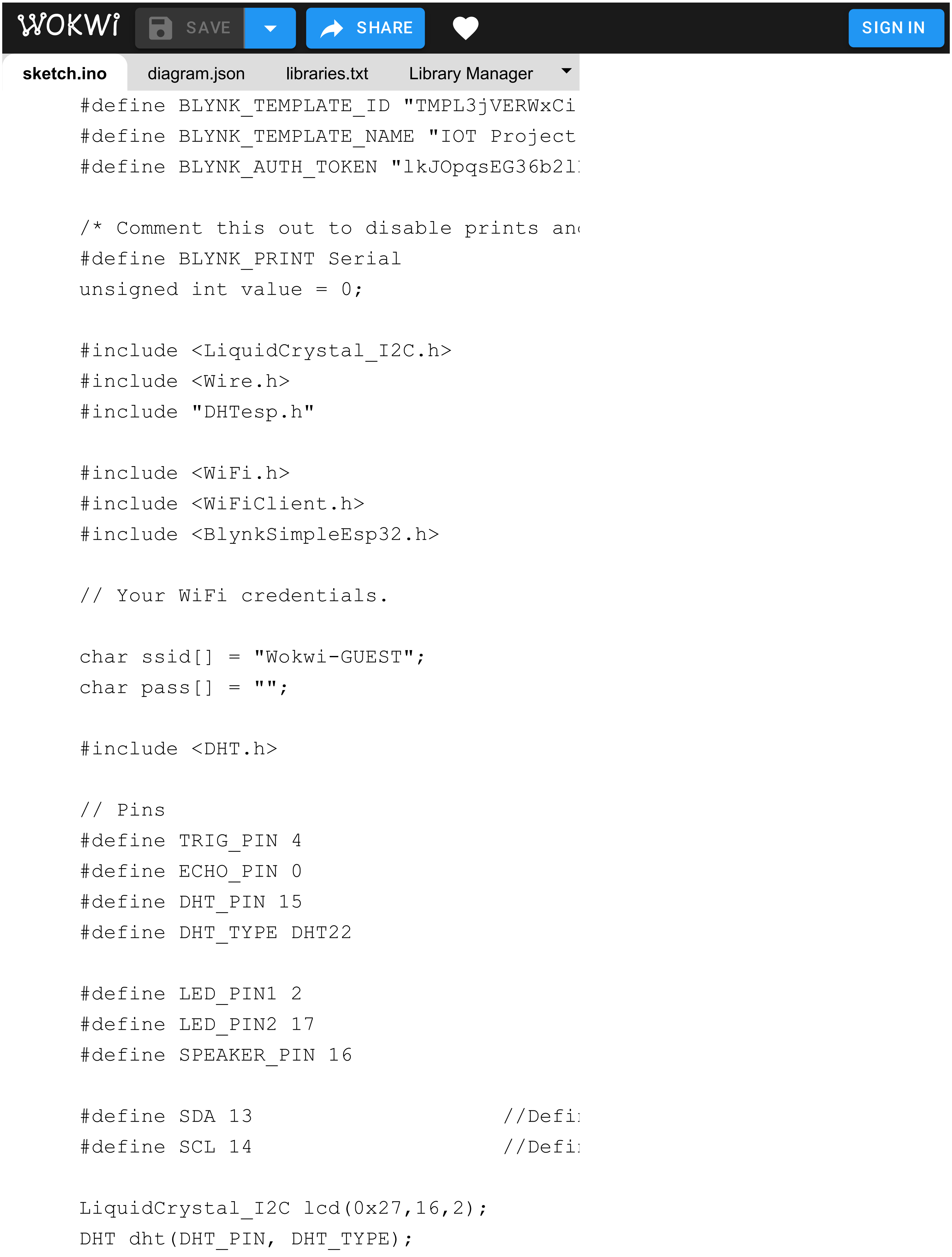

```
#define BLYNK_TEMPLATE_ID "TMPL3jVERWxCi
#define BLYNK_TEMPLATE_NAME "IOT Project
#define BLYNK_AUTH_TOKEN "lkJOpqsEG36b2l

/* Comment this out to disable prints an
#define BLYNK_PRINT Serial
unsigned int value = 0;

#include <LiquidCrystal_I2C.h>
#include <Wire.h>
#include "DHTesp.h"

#include <WiFi.h>
#include <WiFiClient.h>
#include <BlynkSimpleEsp32.h>

// Your WiFi credentials.

char ssid[] = "Wokwi-GUEST";
char pass[] = "";

#include <DHT.h>

// Pins
#define TRIG_PIN 4
#define ECHO_PIN 0
#define DHT_PIN 15
#define DHT_TYPE DHT22

#define LED_PIN1 2
#define LED_PIN2 17
#define SPEAKER_PIN 16

#define SDA 13                          //Defi
#define SCL 14                          //Defi

LiquidCrystal_I2C lcd(0x27,16,2);
DHT dht(DHT_PIN, DHT_TYPE);
```

```
void sendSensorData() {
  // Ultrasonic sensor
  long duration, distance;
  digitalWrite(TRIG_PIN, LOW);
  delayMicroseconds(2);
  digitalWrite(TRIG_PIN, HIGH);
  delayMicroseconds(10);
  digitalWrite(TRIG_PIN, LOW);
  duration = pulseIn(ECHO_PIN, HIGH);
  distance = (duration / 2) / 29.1; // C

  // DHT sensor
  float temperature = dht.readTemperatur
  float humidity = dht.readHumidity();

  // Send data to Blynk
  Blynk.virtualWrite(V2, distance);
  Blynk.virtualWrite(V0, temperature);
  Blynk.virtualWrite(V3, humidity);

  // Check if bin is full
  if (distance < 10) {
    Blynk.virtualWrite(V1,"Waste bin is 
    //tone(16, 262, 250);
    tone(SPEAKER_PIN,262,250);
    digitalWrite(LED_PIN2, HIGH);
    digitalWrite(LED_PIN1, LOW);
    lcd.clear();
    lcd.print("Bin is Full!");
  }
  else{
    Blynk.virtualWrite(V1,"Bin has Space
    digitalWrite(LED_PIN1, HIGH);
    digitalWrite(LED_PIN2, LOW);
    lcd.clear();
    lcd.print("Bin has Space");
  }
  delay(5000);
}
```

```
BlynkTimer timer;

// This function is called every time th
BLYNK_WRITE(V4)
{
  Serial.println("Inside Blynk Write");
  if(param.asInt() == 1)
  {
      Serial.println("Blynk Write: Value
     //digitalWrite(2, HIGH);
     digitalWrite(LED_PIN1, HIGH);
     Blynk.virtualWrite(V1,"Started Succ
     sendSensorData();
  }
  else
  {
    Serial.println("Blynk Write: Value i
    //digitalWrite(2, LOW);
    digitalWrite(LED_PIN1, LOW);
    digitalWrite(LED_PIN2, HIGH);
  }
}

// This function is called every time th
BLYNK_CONNECTED()
{
   Blynk.syncVirtual(V0);
   Blynk.syncVirtual(V1);
   Blynk.syncVirtual(V2);
   Blynk.syncVirtual(V3);
   Blynk.syncVirtual(V4);
   Serial.println("Inside Blynk: Blynk i
   lcd.clear();
   lcd.print("Blynk -Connected");
   delay(5000);
}

// This function sends Arduino's uptime
void myTimerEvent()
```

```
{
  Blynk.virtualWrite(V0, millis() / 1000
  Blynk.virtualWrite(V1, millis() / 1000
  Blynk.virtualWrite(V2, millis() / 1000
  Blynk.virtualWrite(V3, millis() / 1000
  Blynk.virtualWrite(V4, millis() / 1000
}

bool i2CAddrTest(uint8_t addr) {
  Wire.begin();
  Wire.beginTransmission(addr);
  if (Wire.endTransmission() == 0) {
    return true;
  }
  return false;
}

void setup()
{
  Wire.begin(SDA, SCL);
  if (!i2CAddrTest(0x27)) {
    lcd = LiquidCrystal_I2C(0x3F, 16, 2)
  }
  lcd.init();
  lcd.backlight();

  lcd.setCursor(0,0);
  lcd.print("Smart Campus");
  lcd.setCursor(0,1);
  lcd.print("Waste Management");
  //delay(2500);
  //lcd.clear();
  lcd.setCursor(0,2);
  lcd.print(" Version 0.0.1");
  //

  pinMode(5, INPUT_PULLUP);
  //pinMode(2, OUTPUT);
  //pinMode(17, OUTPUT);
  pinMode(LED_PIN1, OUTPUT);
  pinMode(LED_PIN2, OUTPUT);
  pinMode(SPEAKER_PIN, OUTPUT);
  // Begin.
  Serial.begin(115200);
  Serial.println("Serial Prints Starts..

  Blynk.begin(BLYNK_AUTH_TOKEN, ssid, pa


  pinMode(TRIG_PIN, OUTPUT);
  pinMode(ECHO_PIN, INPUT);

  dht.begin();

  // Set a timer to send sensor data eve
  timer.setInterval(10000L, sendSensorDa
}

void loop()
{
  Blynk.run();
  timer.run();

  // DHT sensor
  float temperature = dht.readTemperatur
  float humidity = dht.readHumidity();

  lcd.clear();
  lcd.setCursor(0,0);
  lcd.print("Temp: " + String(temperatur
  lcd.setCursor(0,1);
  lcd.print("Humidity: " + String(humidi


  //lcd.setCursor(0,1);
  //lcd.print("Counter:");
  //lcd.print(millis() / 1000);
  delay(5000);
}
```