Smart Greenhouse IoT System using ESP8266

Aachu Anna Sony (Roll No: 1)
Diya Risa Chacko (Roll No: 22)
Helna Mary Sherin (Roll No: 30)
Semester 3, Electronics & Computer Engineering (ER)
Department of Electronics & Computer Engineering
Saintgits College of Engineering

Course: Microcontroller Based System Design (MBSD) Micro Project - Phase 2

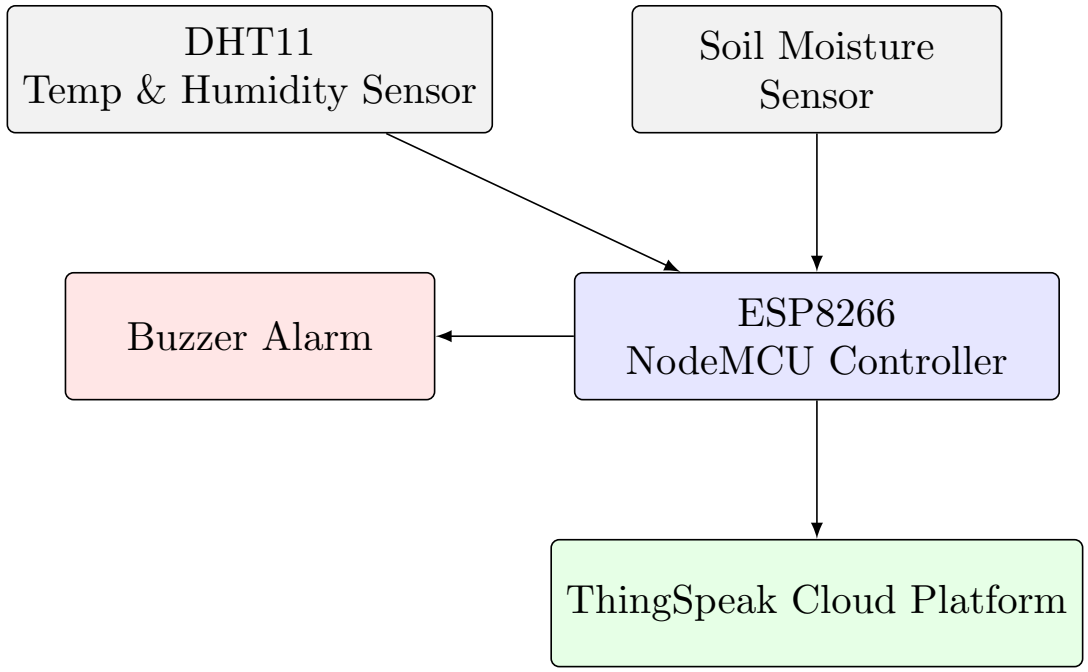
1. Introduction and Objectives

The **Smart Greenhouse IoT System** is a compact embedded setup that monitors temperature, humidity, and soil moisture using sensors connected to an **ESP8266 NodeMCU** microcontroller. The system activates a buzzer alarm whenever any parameter goes beyond safe limits and uploads all readings to the **ThingSpeak IoT Cloud** for real-time monitoring and visualization.

Objectives:

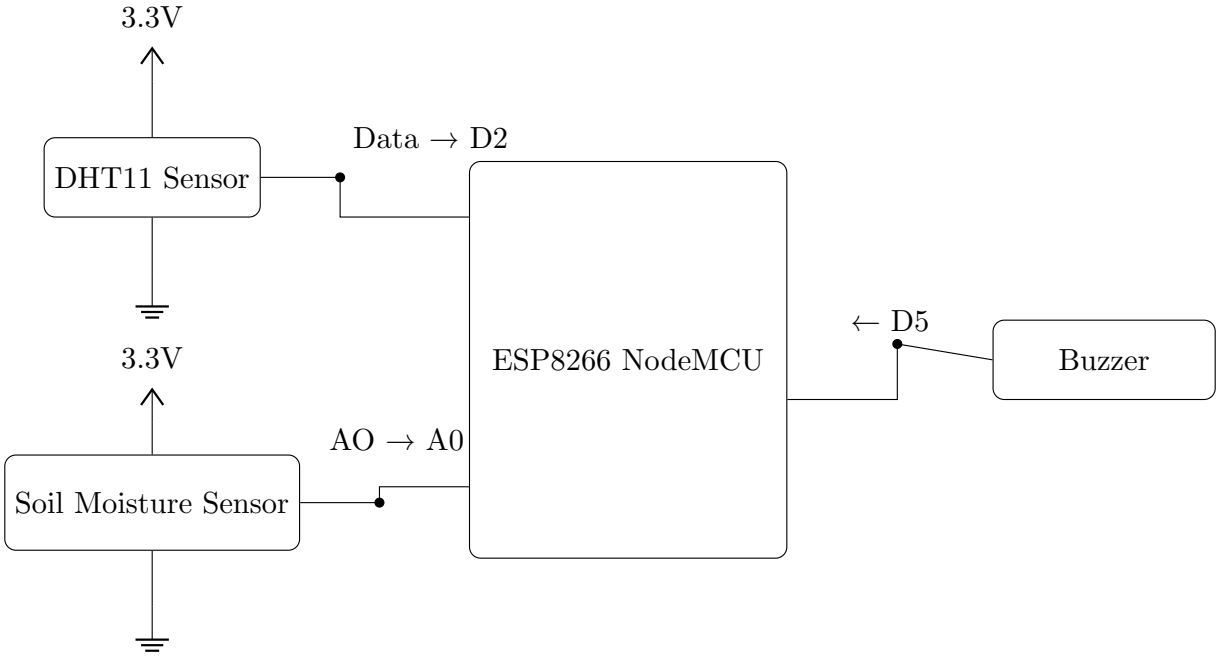
- To design an IoT-enabled greenhouse monitoring system using ESP8266.
- To measure temperature, humidity, and soil moisture in real time.
- To alert the user through a buzzer during critical environmental changes.
- To transmit sensor data to the ThingSpeak cloud using Wi-Fi.
- To operate efficiently using NodeMCU's built-in 3.3V supply.

2. Functional Block Diagram with Explanation



Explanation: The DHT11 sensor provides temperature and humidity readings, while the soil moisture sensor measures the soil dryness. The ESP8266 NodeMCU processes these sensor inputs and triggers the buzzer whenever values cross predefined thresholds. Simultaneously, it uploads the data to ThingSpeak for cloud-based IoT monitoring.

3. Circuit Diagram with Working Principle and Methodology



Working Principle and Methodology: The NodeMCU reads data from DHT11 (temperature and humidity) and the soil moisture sensor. When the temperature is above **32°C** or below **18°C**, humidity drops below **40%** or exceeds **80%**, or the soil moisture value exceeds **800** (indicating dryness), the buzzer turns ON as an alert. All sensor data are sent to the **ThingSpeak Cloud** every 20 seconds for real-time visualization. The system runs entirely on the NodeMCU's onboard 3.3 V power without needing an external 5 V supply.

4. Arduino Code and Interfacing Logic

Program Code:

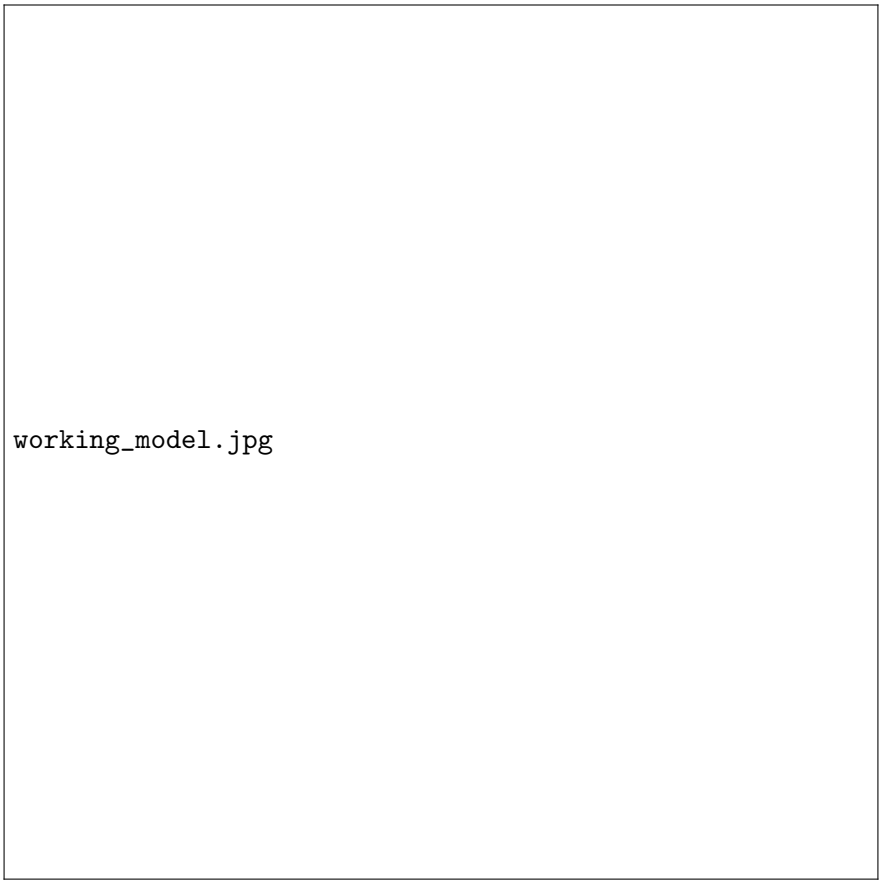
```
#include <ESP8266WiFi.h>
#include <ThingSpeak.h>
#include "DHT.h"

// ---------- PIN DEFINITIONS ----------
#define DHTPIN D2
#define DHTTYPE DHT11
#define SOILPIN A0
#define BUZZER D5

// ---------- WiFi & ThingSpeak ----------
const char* ssid = "Helnaaaa";
const char* password = "hemash_23";

unsigned long channelID = 3111183;
const char* writeAPIKey = "OD5T4LFHTSRMUWUP";

WiFiClient client;
DHT dht(DHTPIN, DHTTYPE);
unsigned long lastDHTRead = 0;
unsigned long lastThingSpeakUpdate = 0;

void setup() {
  Serial.begin(115200);
  dht.begin();
  pinMode(BUZZER, OUTPUT);
  WiFi.begin(ssid, password);
  Serial.print("Connecting to WiFi");
  while (WiFi.status() != WL_CONNECTED) {
    delay(500); Serial.print(".");
  }
```

```
  Serial.println("\n WiFi Connected!");
  ThingSpeak.begin(client);
}

void loop() {
  bool buzzerState = LOW;
  int soilValue = analogRead(SOILPIN);
  Serial.print(" Soil Moisture Value: "); Serial.println(soilValue);

  if (soilValue > 800) { Serial.println(" Soil is very dry!"); buzzerState = HIGH; }

  if (millis() - lastDHTRead > 3000) {
    lastDHTRead = millis();
    float h = dht.readHumidity();
    float t = dht.readTemperature();
    if (!isnan(h) && !isnan(t)) {
      Serial.print(" Temp: "); Serial.print(t);
      Serial.print("C, Humidity: "); Serial.println(h);
      if (t > 32 || t < 18 || h < 40 || h > 80) buzzerState = HIGH;
    }
  }

  digitalWrite(BUZZER, buzzerState);

  if (millis() - lastThingSpeakUpdate > 20000) {
    lastThingSpeakUpdate = millis();
    if (WiFi.status() == WL_CONNECTED) {
      ThingSpeak.setField(1, dht.readTemperature());
      ThingSpeak.setField(2, dht.readHumidity());
      ThingSpeak.setField(3, soilValue);
      ThingSpeak.writeFields(channelID, writeAPIKey);
    }
  }
  delay(1000);
}
```

Complete Pin Interfacing Logic:

- **DHT11 Sensor:**
  - VCC → 3.3 V (NodeMCU)
  - GND → G (Ground)
  - Data → D2 (Digital Input)
- **Soil Moisture Sensor:**
  - VCC → 3.3 V (NodeMCU)
  - GND → G (Ground)
  - AO (Analog Output) → A0 (Analog Input)
- **Buzzer:**
  - Positive (Anode) → D5 (Digital Output)
  - Negative (Cathode) → GND
- **Wi-Fi Communication:**
  - Managed internally by ESP8266 Wi-Fi module.
  - Data uploaded every 20 seconds to ThingSpeak Cloud.
- **Power Supply:**
  - NodeMCU powered via USB (5 V from laptop or adapter).
  - Onboard regulator provides 3.3 V to sensors and buzzer.

5. Image of Working Model



(Smart Greenhouse prototype using ESP8266, sensors, and buzzer alert system)

6. Expected and Observed Outcomes

- Continuously monitors soil moisture, temperature, and humidity.
- Buzzer activates when readings exceed safe limits.
- Sends live sensor data to ThingSpeak cloud every 20 seconds.
- Operates fully on NodeMCU's onboard 3.3 V supply.
- Provides an efficient IoT-based greenhouse monitoring solution.