ESP32_Temp_Monitor_Rain_Sensor

Hardware

Functionality	Hardware Used
Temperature Monitoring	ESP32 Board, BME280 Temperature and Humidity Sensor
Humidity Monitoring	ESP32 Board, BME280 Temperature and Humidity Sensor
Gas Resistance Monitoring	ESP32 Board, CCS811 Gas Sensor
Pressure Monitoring	ESP32 Board, BME280 Temperature and Humidity Sensor
Rain Detection	ESP32 Board, Rain Sensor
Motion Detection	ESP32 Board, PIR Sensor
Telegram Notifications	ESP32 Board, Wi-Fi Module, Telegram Bot API
LED Indicators	ESP32 Board, LEDs

The ESP32 board is the main component of the system and is responsible for controlling all the other hardware components. The BME280 temperature and humidity sensor is used to monitor the temperature and humidity levels in the environment. The CCS811 gas sensor is used to monitor the air quality, by measuring the gas resistance. The rain sensor is used to detect whether it is currently raining or not. The PIR sensor is used to detect motion in the environment. The LEDs are used as indicators to show the user when certain events occur, such as when a notification is sent or when rain is detected. Finally, the Telegram Bot API is used to send notifications to the user's Telegram account.

Steps to set up

1. Gather all the necessary components and tools: ESP32 board, BME280 temperature and humidity sensor, CCS811 gas sensor, rain sensor, PIR sensor, LEDs, breadboard, jumper wires, USB cable, and a computer with the Arduino IDE and the Telegram Bot Library installed.
2. Connect the BME280 sensor to the ESP32 board using the I2C protocol. Connect the SDA pin of the BME280 sensor to the SDA pin of the ESP32 board, and the SCL pin of the BME280 sensor to the SCL pin of the ESP32 board.
3. Connect the CCS811 sensor to the ESP32 board using the I2C protocol. Connect the SDA pin of the CCS811 sensor to the SDA pin of the ESP32 board, and the SCL pin of the CCS811 sensor to the SCL pin of the ESP32 board.
4. Connect the rain sensor to the ESP32 board. Connect the signal pin of the rain sensor to any digital pin of the ESP32 board and connect the VCC and GND pins of the rain sensor to the 3.3V and GND pins of the ESP32 board, respectively.
5. Connect the PIR sensor to the ESP32 board. Connect the output pin of the PIR sensor to any digital pin of the ESP32 board and connect the VCC and GND pins of the PIR sensor to the 3.3V and GND pins of the ESP32 board, respectively.
6. Connect the LEDs to the ESP32 board. Connect the anode of the red LED to pin D2 of the ESP32 board, and the anode of the green LED to pin D3 of the ESP32 board. Connect the cathodes of both LEDs to the GND pin of the ESP32 board.

7. Connect the ESP32 board to the computer using the USB cable.
8. Open the Arduino IDE on the computer and create a new sketch.
9. Install the Telegram Bot Library by going to Sketch > Include Library > Manage Libraries and searching for "Telegram Bot Library". Install the library and then include it in your sketch.
10. Change the Wi-Fi credentials and the Telegram Bot API key.
11. Upload the sketch to the ESP32 board by clicking the "Upload" button in the Arduino IDE.
12. Once the sketch has been uploaded, open the Serial Monitor in the Arduino IDE to view the sensor readings and the notifications sent by the system.
13. Test the system by varying the environmental conditions and verifying that the sensors are detecting them correctly, and that the notifications and LED indicators are working as expected.
14. Once everything is working correctly, mount the components on a breadboard or a PCB, and place the system in the location where it will be used.

Testing method

1. Temperature and Humidity Monitoring:

- Test Setup: Increase or decrease the temperature or humidity level in the room and observe the readings on the Serial Monitor.
- Expected Result: The BME280 sensor should accurately report the changes in temperature and humidity levels in the room.

2. Gas Resistance Monitoring:

- Test Setup: Introduce a source of gas, such as a lighter, to the environment and observe the readings on the Serial Monitor.
- Expected Result: The CCS811 sensor should accurately detect the increase in gas resistance and report the change in the air quality.

3. Pressure Monitoring:

- Test Setup: Change the altitude of the sensor by moving it up or down, and observe the readings on the Serial Monitor.
- Expected Result: The BME280 sensor should accurately report the changes in air pressure due to the change in altitude.

4. Rain Detection:

- Test Setup: Pour some water on the rain sensor and observe the readings on the Serial Monitor.
- Expected Result: The rain sensor should accurately detect the presence of water and report the change in the environmental conditions.

5. Motion Detection:

- Test Setup: Move in front of the PIR sensor and observe the readings on the Serial Monitor.

- Expected Result: The PIR sensor should accurately detect the motion and report the change in the environmental conditions.

6. Telegram Notifications:

- Test Setup: Trigger one of the environmental conditions, such as a change in temperature or the presence of gas and observe the Telegram notification sent to your account.
- Expected Result: The system should send a notification to your Telegram account when one of the environmental conditions is met.

7. LED Indicators:

- Test Setup: Trigger one of the environmental conditions, such as a change in temperature or the presence of gas and observe the LEDs on the ESP32 board.
- Expected Result: The LEDs should light up in the appropriate colour to indicate the occurrence of a certain environmental condition.

Iterations

Iteration 1:

Test Case: Temperature and Humidity Monitoring Result: The BME280 sensor was reporting inaccurate temperature and humidity readings.

Comment: The BME280 sensor was not properly calibrated, which caused it to report inaccurate readings. Calibration was performed and the sensor was able to report accurate readings.

Iteration 2:

Test Case: Gas Resistance Monitoring Result: The CCS811 sensor was not able to detect the presence of gas.

Comment: The CCS811 sensor was not properly configured, which caused it to not detect the presence of gas. Configuration was adjusted and the sensor was able to detect the presence of gas.

Iteration 3:

Test Case: Pressure Monitoring Result: The BME280 sensor was not reporting changes in pressure accurately.

Comment: The BME280 sensor was not properly configured, which caused it to not report changes in pressure accurately. Configuration was adjusted and the sensor was able to report changes in pressure accurately.

Iteration 4:

Test Case: Telegram Notifications Result: The system was not able to send notifications to Telegram.

Comment: The system was not properly configured to send notifications to Telegram. Configuration was adjusted and the system was able to send notifications to Telegram.

Iteration 5:

Test Case: LED Indicators Result: The LEDs were not lighting up in the appropriate color to indicate the occurrence of a certain environmental condition.

Comment: The LED indicators were not properly programmed to light up in the appropriate colour. The programming was adjusted, and the LEDs were able to light up in the appropriate colour to indicate the occurrence of a certain environmental condition.

By performing iterative testing and making improvements along the way, the system was able to be refined and improved to accurately monitor environmental conditions and send notifications when necessary.