Unit Testing and Integration Testing Plan of the Sensor Module

Unit Testing

Overview:

Unit testing focuses on testing individual units or components of the code in isolation to ensure that each part functions as expected. For the sensor module code, key components include functions, libraries, and specific functionalities.

Components to be Tested:

1. WiFi Connection Functions:

- `testWifi()`
- `launchWeb()`
- `setupAP()`

2. ThingSpeak Communication:

- `writeFields()`

3. Sensor Readings:

- `readTemperature()`
- `readHumidity()`
- `analogRead()`

4. EEPROM Operations:

- Reading and writing WiFi credentials.

5. Web Server Setup:

- `createWebServer()`

6. DHT Sensor Integration:

- `readTemperature()` and `readHumidity()` integration with DHT sensor.

Unit Testing Plan:

1. WiFi Connection Functions:

- Input: Simulated WiFi connection status.
- Expected Result: Correct determination of WiFi connection status.

2. ThingSpeak Communication:

- Input: Simulated data to be sent to ThingSpeak.
- Expected Result: Successful write to ThingSpeak.

3. Sensor Readings:

- Input: Simulated sensor readings.
- Expected Result: Accurate temperature, humidity, and potentiometer readings.

4. EEPROM Operations:

- Input: Simulated WiFi credentials.
- Expected Result: Correct reading and writing to EEPROM.

5. Web Server Setup:

- Input: Simulated web server setup.
- Expected Result: Proper initialization and configuration of the web server.

6. DHT Sensor Integration:

- Input: Simulated DHT sensor readings.
- Expected Result: Accurate integration with DHT sensor library.

Integration Testing

Overview:

Integration testing evaluates the interactions between different components and ensures that they work together seamlessly. It focuses on verifying that the modules interact correctly and exchange information as intended.

Components to be Tested:

1. WiFi Connection and Hotspot Setup:

- Interaction between `testWifi()`, `launchWeb()`, and `setupAP()`.

2. ThingSpeak Communication and Sensor Readings:

- Interaction between `writeFields()`, `readTemperature()`, `readHumidity()`, `analogRead()`.

3. Web Server Setup and EEPROM Operations:

- Interaction between `createWebServer()` and EEPROM functions.

4. Complete System Integration:

- Interaction between all components to ensure the entire system functions cohesively.

Integration Testing Plan:

1. WiFi Connection and Hotspot Setup:

- Input: Simulated WiFi connection and hotspot setup scenarios.
- Expected Result: Successful hotspot initiation and correct response to WiFi connection status.

2. ThingSpeak Communication and Sensor Readings:

- Input: Simulated sensor readings and data to be sent to ThingSpeak.
- Expected Result: Accurate data transmission to ThingSpeak based on sensor readings.

3. Web Server Setup and EEPROM Operations:

- Input: Simulated web server setup and WiFi credentials modification.
- Expected Result: Proper initialization of web server and successful EEPROM operations during WiFi credentials update.

4. Complete System Integration:

- Input: Comprehensive testing of the entire sensor module code.
- Expected Result: All components work together seamlessly, and the module functions as intended.

Conclusion:

Unit and integration testing are crucial for ensuring the reliability and functionality of the sensor module code. By thoroughly testing individual components and their interactions, potential issues can be identified and addressed, contributing to a more robust and stable system.