EECS 113 Final Project
Building Management System
Esther Anaya
82547392
Due 6.13.25

Project Overview

The goal of this final project was to design and implement a **Building Management System (BMS)** on a Raspberry Pi using sensors and actuators. The system provides an integrated solution for controlling ambient lighting, room temperature (HVAC), fire alerts, and perimeter monitoring through sensors and a real-time LCD interface. The solution includes real-time weather-based decisions via the OpenWeatherMap API and logging of all system activity.

System Features Implemented

1. Ambient Light Control

- **Hardware Used:** PIR sensor, Yellow LED
- **Function:** Detects motion and activates the light. If no motion is detected for 10 seconds, the light turns off automatically.
- **LCD Display:** Shows light status (ON/OFF).
- **Log Events:** "LIGHTS ON", "LIGHTS OFF"

2. Room Temperature Control (HVAC)

- **Hardware Used:** DHT11 sensor, Red and Blue LEDs, Push buttons
- **Inputs:**
    - Temperature: Read from DHT11 every 1s and averaged over the last 3 readings.
    - Humidity: Retrieved from OpenWeatherMap API using API key: `b02542f41eb133c138feabe966165e01`
- **Weather Index Calculation:**
- **User Control:**
    - Desired temperature adjustable via external interrupts (push buttons)
    - Temp range: 65°F to 95°F
- **HVAC Logic:**
    - Turns on **AC** if weather index ≥ desired + 3
    - Turns on **Heater** if weather index ≤ desired - 3
    - Turns off HVAC if window/door is open or if no heating/cooling needed
- **Hysteresis:** Prevents rapid toggling between heater and AC
- **LCD Display:** Shows real-time temp, HVAC status, door status, and light status
- **Log Events:** "HVAC AC", "HVAC HEAT", "HVAC OFF"

3. Fire Alarm System

- **Trigger:** Weather index exceeds 95
- **Actions:**
    - HVAC turned off
    - Door opened (Green LED)
    - Fire LED (Red) flashes
    - LCD shows evacuation message

- **Log Events:** "FIRE ALERT", "DOOR OPEN", "HVAC OFF"

4. Security System

- **Hardware Used:** Door/window sensor (Button), Green LED
- **Functionality:**
    - Button state triggers LCD alert for 3 seconds
    - HVAC system disabled while door is open
    - System resumes when door is closed
- **LCD Display:** Displays "DOOR/WINDOW OPEN" or "CLOSED"
- **Log Events:** "DOOR OPEN", "DOOR CLOSED", "HVAC OFF"

5. Internal Clock and Logging

- **Functionality:**
    - Uses `datetime` to timestamp each event
    - Appends to `log.txt`

Code Summary and Main Functions

The core functionality is built around an infinite loop that monitors sensor data, responds to user input, and updates the display and log files in real-time. Below are the key parts of the code and what they do:

1. Initialization

- Sets up all sensors and devices using GPIO and `lgpio`
- Reads the API key from `key.txt`
- Initializes LCD and prints a startup message
- Prepares a log file to record system events

2. PIR Light Control

- **Variables:** `curr_state`, `prev_state`, `last_motion_time`
- **LED Pin:** GPIO20
- **Logic:**
    - Detects motion via `PIR_PIN`
    - If motion detected, turns on yellow light and logs "LIGHTS ON"
    - If no motion for 10s, turns off the light and logs "LIGHTS OFF"

3. Temperature and Humidity Measurement

- Uses the `adafruit_dht` library to read temperature
- Calls `get_irvine_humidity()` to fetch live humidity data from OpenWeatherMap
- Averages the last 3 readings to reduce sensor error

- Rounds the weather index using the formula:
  weather_index = round(avg_temp + 0.05 * humidity)

4. User Temperature Control

- Push buttons on GPIO24 (up) and GPIO25 (down)
- Handled with `Button.when_pressed` to adjust `desired_temp`

5. HVAC Control Logic

- Based on comparison between `weather_index` and `desired_temp`
  - Turns on AC (GPIO12) if `weather_index >= desired_temp + 3`
  - Turns on heater (GPIO16) if `weather_index <= desired_temp - 3`
  - Turns both off if within range or door is open
- Logs every state change and displays temporary notifications on LCD

6. Fire Safety Trigger

- Fires if `weather_index >= 95`
- Activates door open LED, turns off HVAC, flashes fire LED, and displays emergency message on LCD
- Loop pauses for half-second intervals to create flashing effect

7. Door/Window Security

- Push button on GPIO22 simulates open/close
- On open:
  - Door LED (GPIO5) turns on
  - LCD shows door open message and HVAC off
  - Logged as "DOOR OPEN" and "HVAC OFF"
- On close:
  - LED turns off
  - Resumes normal HVAC control

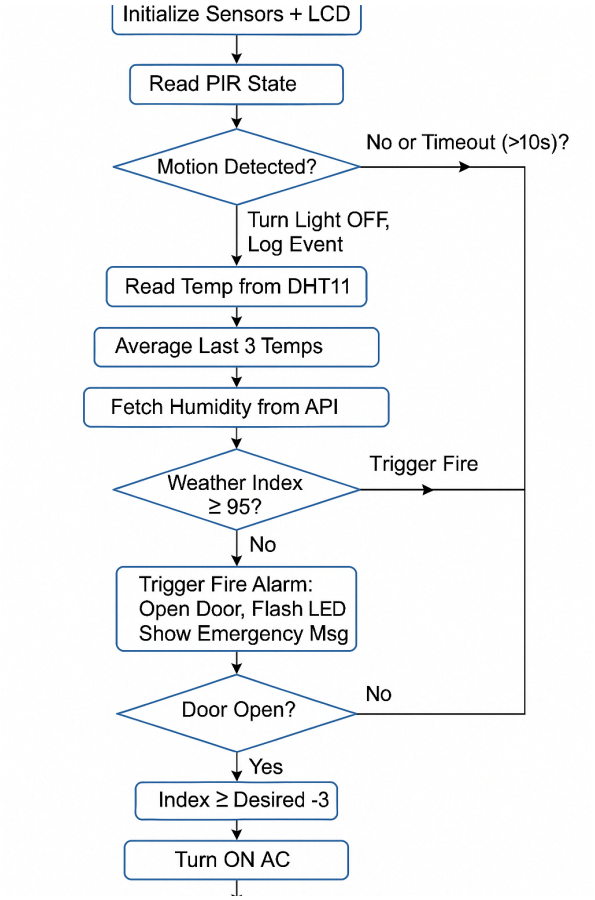
8. LCD Updates

- LCD shows status: temperature, HVAC mode, door state, light state
- If an event occurs (fire alert, door opened), it clears and shows that alert temporarily before returning to status screen

9. Graceful Shutdown

- Registered SIGINT (Ctrl+C) handler ensures all devices are turned off
- Logs system stop event and prints shutdown message

Flowchart of Functions



Hardware Setup Summary

Raspberry Pi GPIO Pins Used:

- PIR Input: GPIO17
- PIR LED: GPIO20
- AC LED (Blue): GPIO12
- Heater LED (Red): GPIO16
- Fire LED (Red): GPIO6
- Door LED (Green): GPIO5
- Temp Up Button: GPIO24
- Temp Down Button: GPIO25
- Door/Window Button: GPIO22

Wiring is neatly arranged on a breadboard with components securely connected to their assigned GPIO pins. PIR sensor is placed to face a general motion zone; DHT11 sensor is mounted for unobstructed airflow.

Images

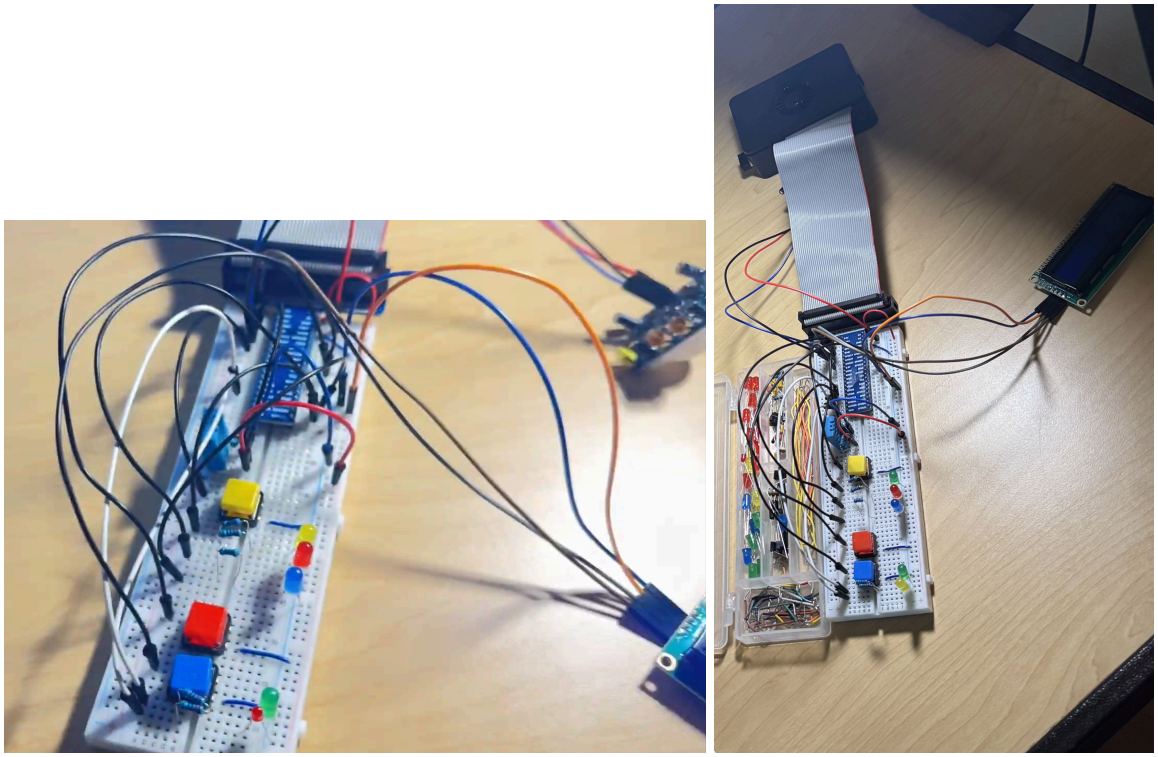


Image 1 & 2. Breadboard wiring and components setup.

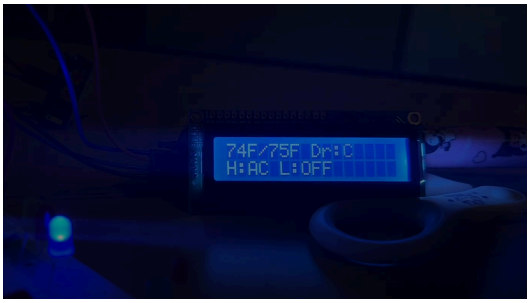


Image 3. LCD display including desired/ actual temp, Door Status (Open, Closed), HVAC status (AC, Heat), and Light status (ON, OFF)

Image 4. LDC display when there is a fire (temp goes over 95 degrees fahrenheit).

Conclusion

This project demonstrates the integration of software and hardware to create a fully functional BMS. It showcases real-time sensor data processing, decision-making algorithms, user interaction via interrupts, and external API communication. The system successfully handles HVAC automation, lighting control, fire safety alerts, and basic perimeter security.