Cognitive Robotics Lab
Lab 2: Design and Implement Smart Home Automation System

Srikar Ganesh 22BAI1080

04-08-2025

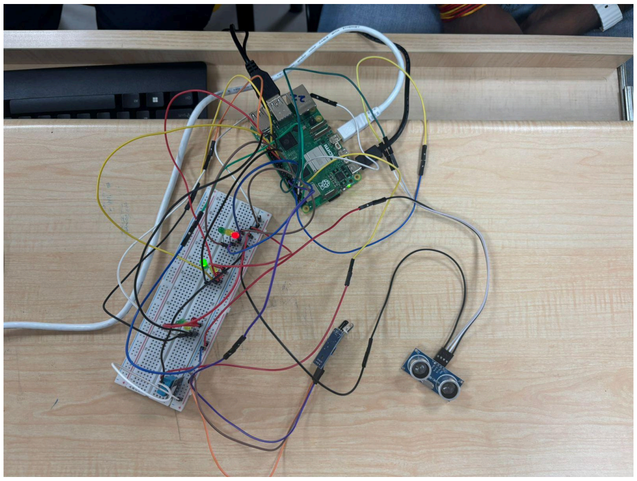
Aim:

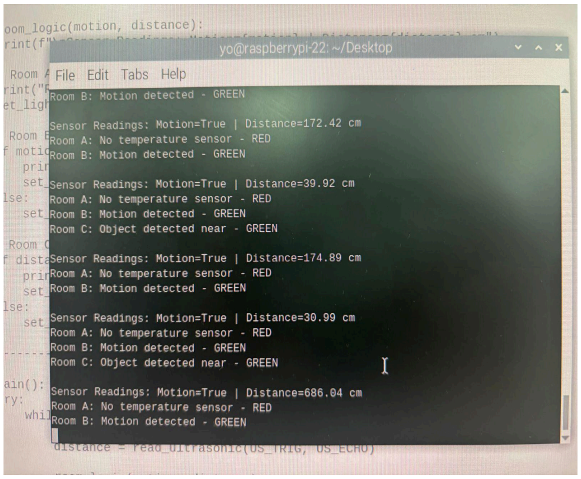
To design and implement a Smart Home Automation System using appropriate sensors, actuators, and microcontrollers (such as Arduino or Raspberry Pi), enabling control and monitoring of home appliances for improved convenience, energy efficiency, and security.

Components:

1. **Raspberry Pi** – Serves as the central controller to process inputs and control outputs.
2. **DHT11 Sensor** – Measures temperature and humidity levels in Room A.
3. **IR Sensor** – Detects motion or presence in Room B.
4. **Ultrasonic Sensor (HC-SR04)** – Measures distance to detect nearby objects in Room C.
5. **RGB LEDs (or individual Red, Yellow, Green LEDs)** – Indicate room status based on sensor readings.
6. **Resistors (220Ω or 330Ω)** – Limit current to protect the LEDs.
7. **Breadboard** – Enables temporary connections for easy prototyping.
8. **Jumper Wires** – Connect sensors and LEDs to the Raspberry Pi GPIO pins.
9. **Power Supply (5V, 2.5A or above)** – Powers the Raspberry Pi and peripherals.
10. **MicroSD Card (with OS)** – Stores the Raspberry Pi operating system and Python script.

Output:





```
Room B: Motion detected - GREEN

Sensor Readings: Motion=True | Distance=172.42 cm
Room A: No temperature sensor - RED
Room B: Motion detected - GREEN

Sensor Readings: Motion=True | Distance=39.92 cm
Room A: No temperature sensor - RED
Room B: Motion detected - GREEN
Room C: Object detected near - GREEN

Sensor Readings: Motion=True | Distance=174.89 cm
Room A: No temperature sensor - RED
Room B: Motion detected - GREEN

Sensor Readings: Motion=True | Distance=30.99 cm
Room A: No temperature sensor - RED
Room B: Motion detected - GREEN
Room C: Object detected near - GREEN

Sensor Readings: Motion=True | Distance=686.04 cm
Room A: No temperature sensor - RED
Room B: Motion detected - GREEN
```

Source Code:

```
import RPi.GPIO as GPIO
import time

GPIO.setmode(GPIO.BCM)
GPIO.setwarnings(False)

# --- Pin Definitions ---
# Room RGB LEDs
PIN_MAP = {
    'A': {'R': 17, 'Y': 27, 'G': 22},
    'B': {'R': 5,  'Y': 6,  'G': 13},
    'C': {'R': 19, 'Y': 26, 'G': 21},
}

# Shared sensors
DHT_PIN = 4
IR_PIN = 23
US_TRIG = 18
US_ECHO = 24

# --- Setup GPIOs ---
for room_pins in PIN_MAP.values():
    for pin in room_pins.values():
        GPIO.setup(pin, GPIO.OUT)
        GPIO.output(pin, GPIO.LOW)

GPIO.setup(IR_PIN, GPIO.IN)
GPIO.setup(US_TRIG, GPIO.OUT)
GPIO.setup(US_ECHO, GPIO.IN)

# --- DHT11 Without Adafruit ---
def read_dht11(pin):
    data = []
    GPIO.setup(pin, GPIO.OUT)
    GPIO.output(pin, GPIO.LOW)
    time.sleep(0.02)  # 20ms
    GPIO.output(pin, GPIO.HIGH)
    GPIO.setup(pin, GPIO.IN)

    count = 0
    last = -1
    while GPIO.input(pin) == GPIO.LOW:
        continue

    while GPIO.input(pin) == GPIO.HIGH:
        continue

    for i in range(40):
        while GPIO.input(pin) == GPIO.LOW:
            continue
        t = time.time()
        while GPIO.input(pin) == GPIO.HIGH:
            continue
        if time.time() - t > 0.00005:
            data.append(1)
        else:
            data.append(0)

    # Convert bits to bytes
    humidity = int("".join(str(bit) for bit in data[0:8]), 2)
    humidity_dec = int("".join(str(bit) for bit in data[8:16]), 2)
    temperature = int("".join(str(bit) for bit in data[16:24]), 2)
    temperature_dec = int("".join(str(bit) for bit in data[24:32]), 2)
    checksum = int("".join(str(bit) for bit in data[32:40]), 2)

    calc_checksum = (humidity + humidity_dec + temperature + temperature_dec) & 0xFF
    if checksum == calc_checksum:
        return temperature, humidity
    else:
        return None, None

# --- Other Sensors ---
def read_ir(pin):
    return GPIO.input(pin) == GPIO.HIGH

def read_ultrasonic(trig, echo):
    GPIO.output(trig, False)
    time.sleep(0.05)

    GPIO.output(trig, True)
    time.sleep(0.00001)
    GPIO.output(trig, False)

    pulse_start = time.time()
    timeout = pulse_start + 0.04
    while GPIO.input(echo) == 0 and time.time() < timeout:
        pulse_start = time.time()

    pulse_end = time.time()
    timeout = pulse_end + 0.04
    while GPIO.input(echo) == 1 and time.time() < timeout:
        pulse_end = time.time()

    duration = pulse_end - pulse_start
    distance = duration * 17150
    return round(distance, 2)

def set_light(room, color):
    pins = PIN_MAP[room]
    for c in ['R', 'Y', 'G']:
        GPIO.output(pins[c], GPIO.LOW)
    GPIO.output(pins[color], GPIO.HIGH)

def room_logic(temp, humidity, motion, distance):
    print(f"\nSensor Readings: Temp={temp}°C | Humidity={humidity}% | Motion={motion} |
Distance={distance} cm")

    # Room A: temperature
    if temp is not None and temp > 28:
        print("Room A: Hot - GREEN")
        set_light('A', 'G')
    elif temp is not None and temp < 20:
        print("Room A: Cold - YELLOW")
        set_light('A', 'Y')
    else:
        set_light('A', 'R')

    # Room B: motion
    if motion:
        print("Room B: Motion detected - GREEN")
        set_light('B', 'G')
    else:
        set_light('B', 'R')

    # Room C: distance
    if distance < 100:
        print("Room C: Object close - GREEN")
        set_light('C', 'G')
    else:
        set_light('C', 'R')

# --- Main Loop ---
def main():
    try:
        while True:
            temp, humidity = read_dht11(DHT_PIN)
            motion = read_ir(IR_PIN)
            distance = read_ultrasonic(US_TRIG, US_ECHO)

            room_logic(temp, humidity, motion, distance)
            time.sleep(5)
    except KeyboardInterrupt:
        print("\nExiting. Cleaning up GPIO...")
    finally:
        for room in PIN_MAP:
            set_light(room, 'R')
        GPIO.cleanup()

if __name__ == "__main__":
    main()
```

Explanation:

The Smart Home Automation System is designed to monitor and respond to environmental conditions in three rooms using different sensors and RGB LEDs to indicate the room status:

- Room A uses a DHT11 sensor to read temperature and humidity. Based on the temperature:
    - If > 28°C, the room is considered hot – Green LED is turned on.
    - If < 20°C, the room is cold – Yellow LED is turned on.
    - Otherwise, normal temperature – Red LED is turned on.
- Room B uses an IR sensor to detect motion:
    - If motion is detected, the Green LED is turned on.
    - If no motion, the Red LED is turned on.
- Room C uses an ultrasonic sensor to measure object distance:
    - If an object is closer than 100 cm, the Green LED turns on.
    - Otherwise, the Red LED turns on.

Each room is represented by a set of RGB LEDs connected to GPIO pins. The code continuously reads the sensor data every 5 seconds, processes it, and updates the LED color for each room accordingly.

Result:

The Smart Home Automation System was successfully implemented using a Raspberry Pi and various sensors. The system accurately monitored environmental conditions in three separate rooms and provided real-time visual feedback using RGB LEDs. In Room A, the DHT11 sensor detected temperature changes, and the corresponding LED changed color to indicate whether the room was hot, cold, or at a normal temperature. Room B effectively detected human presence using an IR sensor, lighting the green LED when motion was sensed and red when the room was vacant. Room C utilized an ultrasonic sensor to detect the proximity of objects, turning on the green LED when an object was within 100 cm. The system ran in a continuous loop, displaying live sensor readings and LED status updates in the terminal, thus demonstrating a reliable and functional smart home monitoring setup.