Steven Al-Sheikh

Justin Nam

Moe Jawadi

Smart Room Availability System

What is the project? The Smart Room Availability System is a project aimed at monitoring the availability of rooms in real-time, allowing users to easily check if rooms, such as study spaces or meeting areas, are occupied or free. The purpose of this system is to improve the efficiency of room utilization and save time for individuals by providing up-to-date occupancy information through a simple web interface.

Objective: Create a system using Raspberry Pi and motion sensors to monitor and display the real-time availability of rooms, such as study spaces or meeting rooms, in a building via a simple web interface.

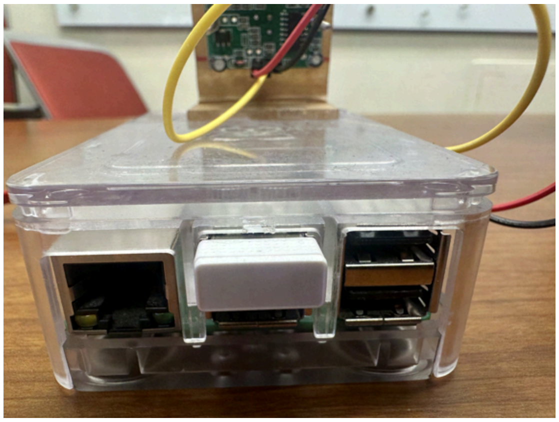
Key Features:

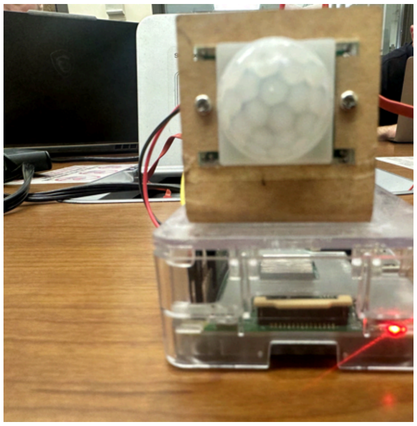
1. **Real-Time Data Processing:** Process sensor data in real-time to determine the occupancy status of each room.
2. **Web Interface:** Develop a web interface where users can view the current occupancy status of rooms by refreshing the page.
3. **Network Communication:** Ensure continuous and secure communication between the motion sensors, Raspberry Pi, and the server that hosts the web interface to maintain accurate and up-to-date room statuses.

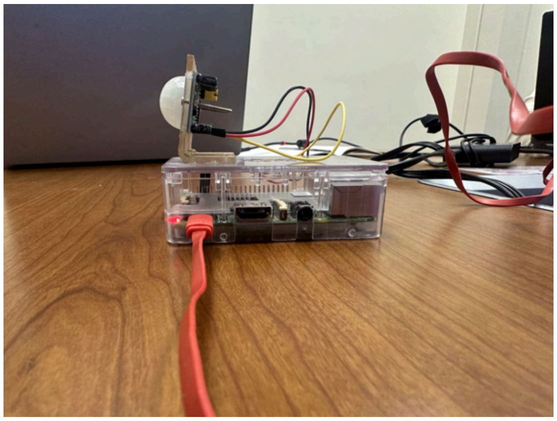
Technologies Used:

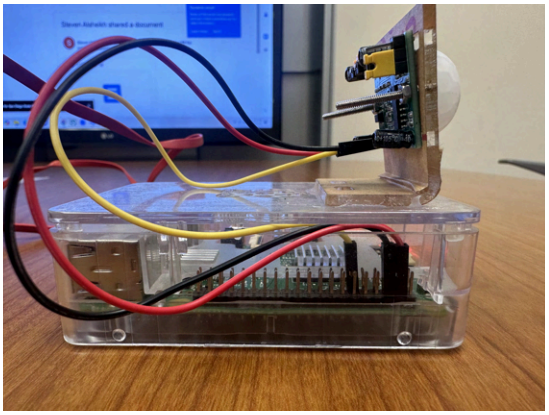
- **Hardware:** Raspberry Pi, motion sensors
- **Software:** Python for scripting, HTML/CSS for the web interface, and potentially Node.js or Python Flask for backend development to manage sensor data and serve it to the web interface.

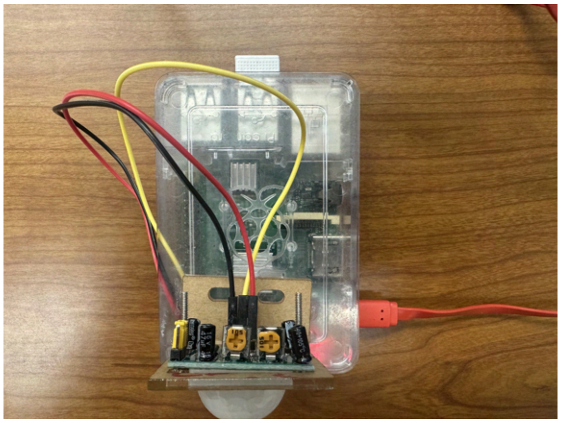
Application: Ideal for educational institutions, libraries, or office buildings where managing room occupancy and availability is critical.



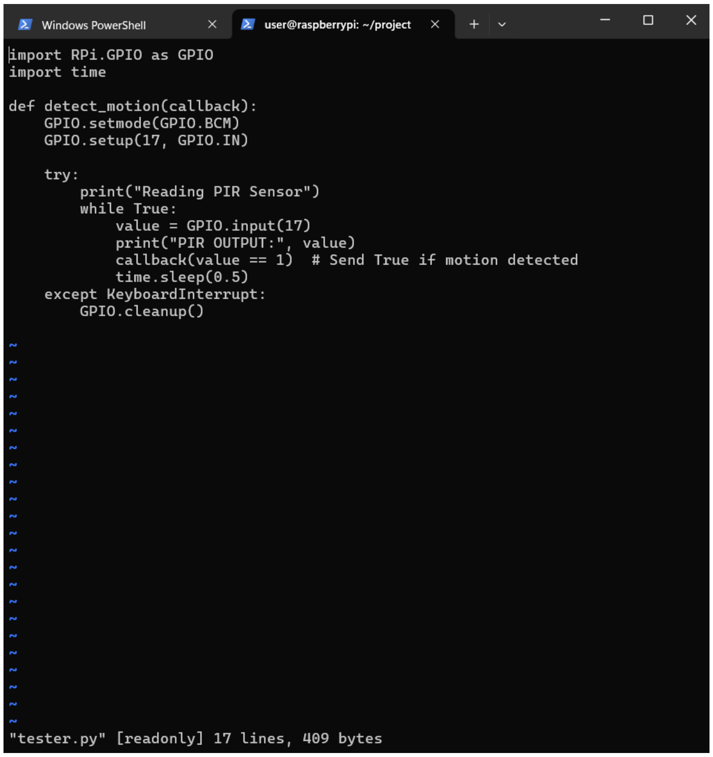








Source Codes:

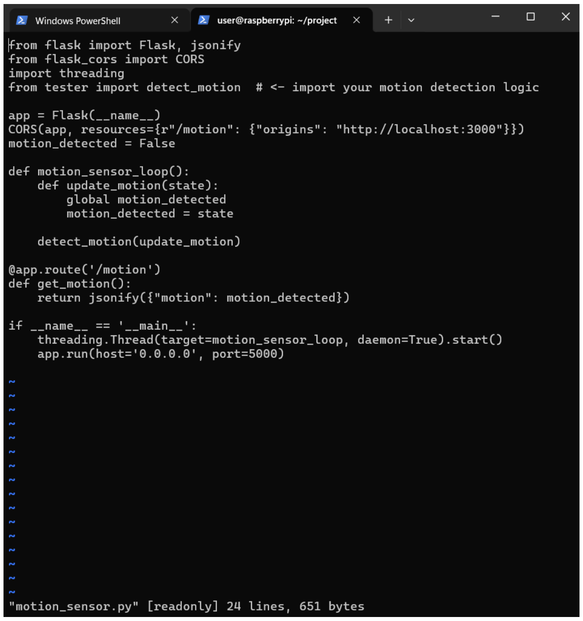


```
import RPi.GPIO as GPIO
import time

def detect_motion(callback):
    GPIO.setmode(GPIO.BCM)
    GPIO.setup(17, GPIO.IN)

    try:
        print("Reading PIR Sensor")
        while True:
            value = GPIO.input(17)
            print("PIR OUTPUT:", value)
            callback(value == 1)  # Send True if motion detected
            time.sleep(0.5)
    except KeyboardInterrupt:
        GPIO.cleanup()
```

```
"tester.py" [readonly] 17 lines, 409 bytes
```



```
from flask import Flask, jsonify
from flask_cors import CORS
import threading
from tester import detect_motion  # <- import your motion detection logic

app = Flask(__name__)
CORS(app, resources={r"/motion": {"origins": "http://localhost:3000"}})
motion_detected = False

def motion_sensor_loop():
    def update_motion(state):
        global motion_detected
        motion_detected = state

    detect_motion(update_motion)

@app.route('/motion')
def get_motion():
    return jsonify({"motion": motion_detected})

if __name__ == '__main__':
    threading.Thread(target=motion_sensor_loop, daemon=True).start()
    app.run(host='0.0.0.0', port=5000)
```

```
"motion_sensor.py" [readonly] 24 lines, 651 bytes
```

Front-End Code

https://github.com/MoeJawadi/CS578-MotionDectector