
Smart Street Lightning System.

Step 1: Install Required Libraries and Package.

Ensure your Raspberry Pi is set up with the necessary libraries and dependencies.

1. Update Raspberry Pi:

- `sudo apt-get update`
- `sudo apt-get upgrade`

2. Install Python3 and pip (if not already installed):

- `sudo apt-get install python3`
- `python3-pip`

3. Install Required Python Libraries:

- RPi.GPIO: For GPIO control.
- `sudo pip3 install RPi.GPIO`
- Adafruit_DHT: For DHT11 sensor.
- `sudo pip3 install Adafruit_DHT`
- smtplib: Built-in Python library for sending emails. No installation required.

4. Enable I2C and SPI (Optional):

- Enable the necessary interfaces for your sensors via the Raspberry Pi configuration tool:
- `sudo raspi-config` Navigate to "Interface Options" and enable I2C and SPI.

Step 2: Hardware Setup

Connect the components to the Raspberry Pi as described:

1. DHT11 Sensor:

- Connect the VCC pin to 3.3V.
- Connect the GND pin to GND.
- Connect the DATA pin to GPIO 4.

2. Ultrasonic Sensor (HC-SR04):

- Connect VCC to 5V.
- Connect GND to GND.
- Connect TRIG to GPIO 19.
- Connect ECHO to GPIO 26 (with a voltage divider if necessary)

3. Light Sensor:

- Connect the sensor's output to GPIO 13.

4. LED Streetlights:

- Connect LEDs to GPIO pins 18, 12, 3, and 5 with appropriate resistors.

Step 3: Email Configuration

1. Gmail Settings:

- Enable "Less Secure App Access" or "App Passwords" in your Gmail account.
- Replace the following placeholders in the script with your credentials:
 - sender_email = "your_email@gmail.com"
 - password = "your_app_password"
 - receiver_email = receiver_email@gmail.com

2. Test Email Sending (Optional):

- Use a basic script to confirm email functionality:
 - import smtplib sender = your_email@gmail.com
 - password = "your_app_password"
 - receiver=receiver_email@gmail.com with smtplib.
 - SMTP("smtp.gmail.com", 587) as server:
 - server.starttls()
 - server.login(sender, password server.
 - sendmail(sender, receiver, "Subject: Test\n\nEmail works!")

Step 4: Load and Run the Script

1. Transfer the Script to Raspberry Pi: Save your Python script (e.g., smart_streetlight.py) on the Raspberry Pi.
2. Make the Script Executable: `chmod +x smart_streetlight.py`
3. Run the Script: `python3 smart_streetlight.py`

Step 5: Observe and Debug

The script will run continuously, controlling LEDs and sending email alerts as per the schedule and sensor readings.

Check the output logs for sensor readings, LED statuses, and email alerts.

Screen Shots



