Smart Street Lightining System (B1-G7)

Chethana N	1AP21IS006
Ajoy Anthony	1AP21CS005
Vinod	1AP22EC409

Overview:

The aim of a **Smart Intelligent Street Light System** is to optimize the performance and efficiency of street lighting by using advanced technologies, such as IoT (Internet of Things), sensors, and artificial intelligence. This system is designed to provide several key benefits

Software Installation:

1. **Update your Raspberry Pi**
```
sudo apt-get update
sudo apt-get upgrade
```

2. **Install Python and pip if not already installed**
```
sudo apt-get install python3 python3-pip
```

3. **Install required Python packages**
   - 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 you 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"
     - receive=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!")

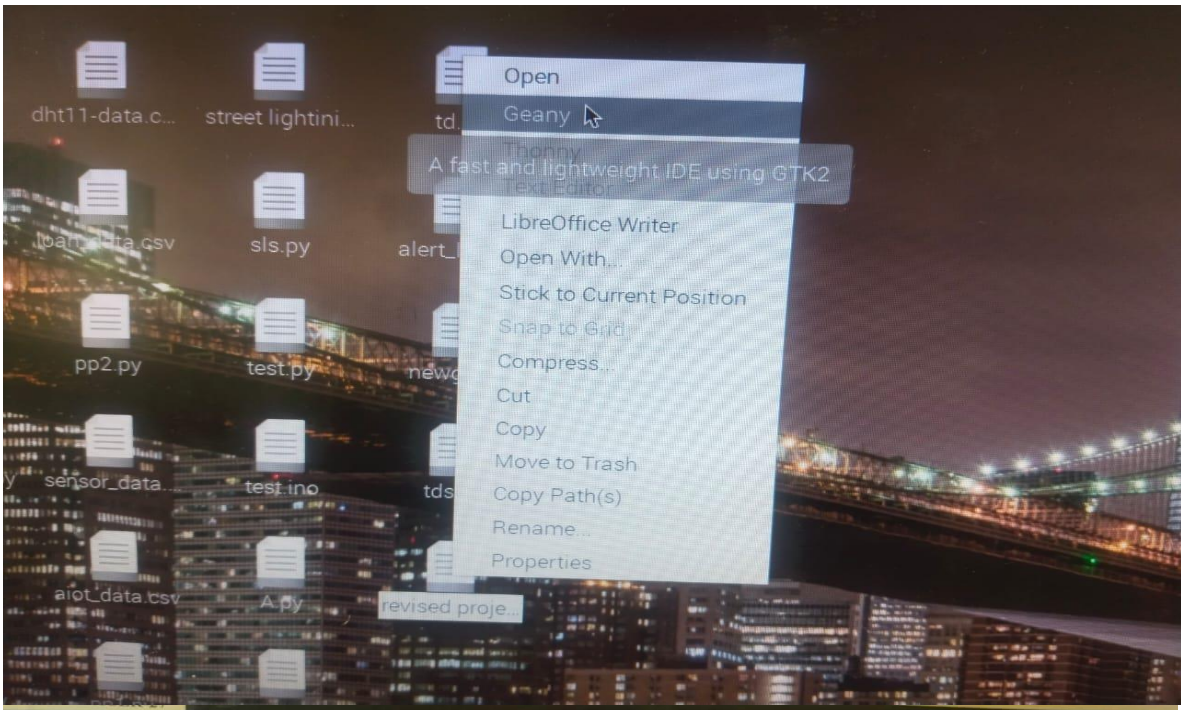
Step 4: Load and Run the Script

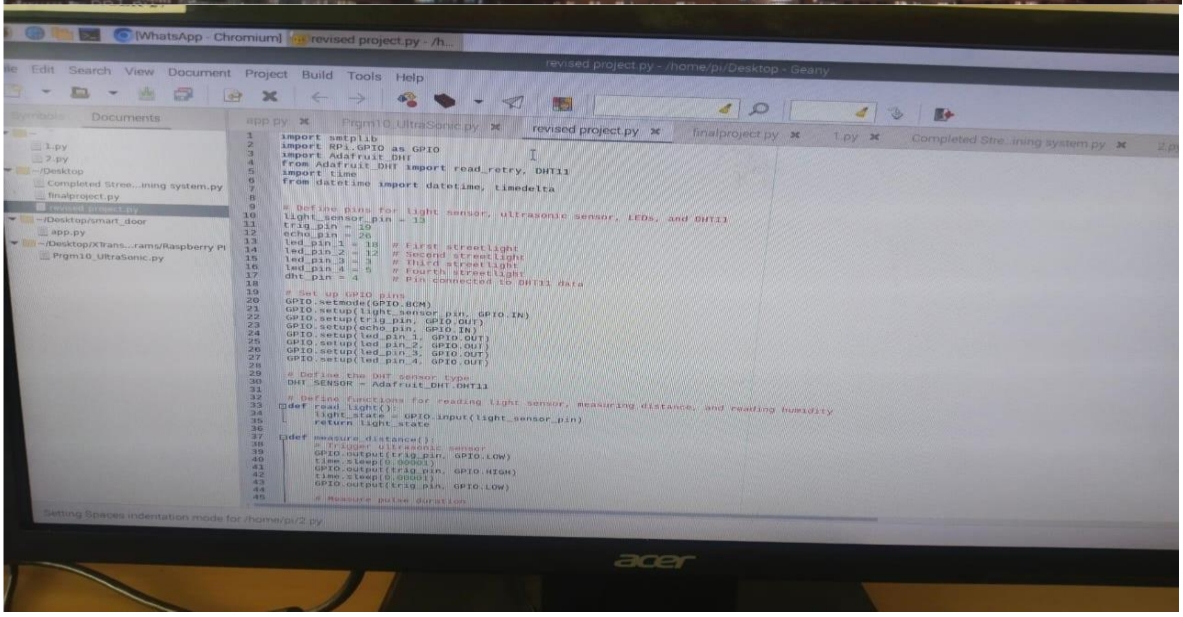
1. Transfer the Script to Raspberry Pi: Save your Python script (e.g., smart_streetlight.py) on the Raspberry Pi.
2. Make the Script Executable:chmod +x smart_streetlight.py
3. 3. Run the Script:python3 smart_streetlight.py

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

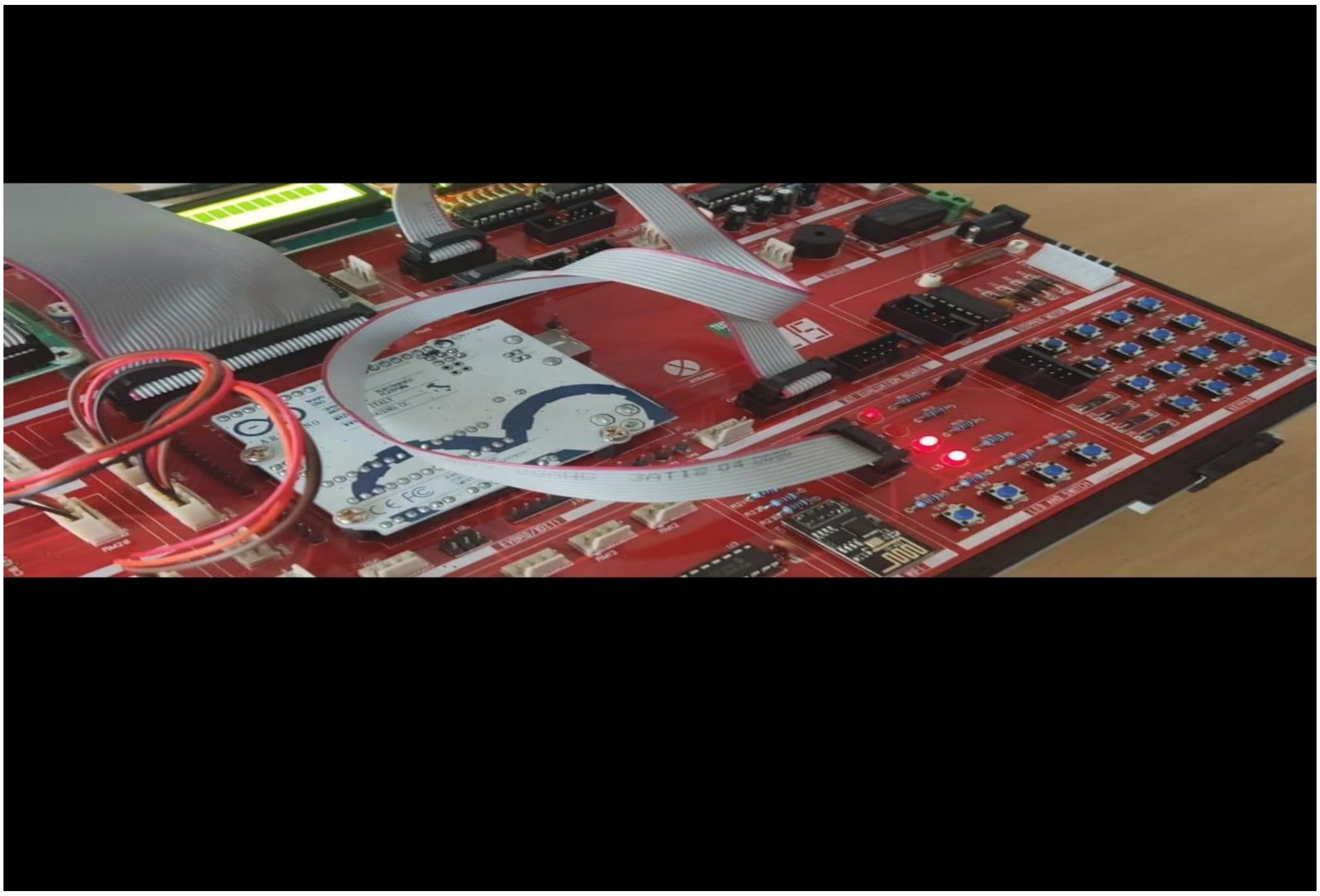
Screenshots

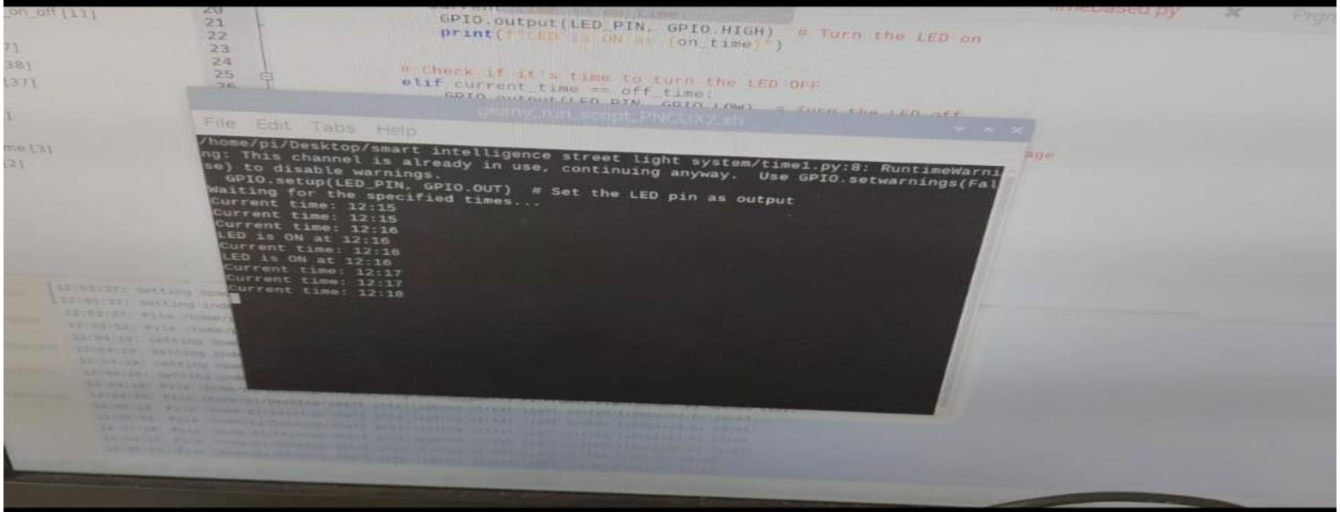



[WhatsApp - Chromium]
revised project.py - /h...
revised project.py - /home/pi/Desktop - Geany
File Edit Search View Document Project Build Tools Help
Symbols Documents
app.py
Prgm10_UltraSonic.py
revised project.py
finalproject.py
1.py
Completed Stre...ining system.py
2.py
~/Desktop
Completed Stree...ining system.py
finalproject.py
revised project.py
~/Desktop/smart_door
app.py
~/Desktop/XTrans...rams/Raspberry Pi
Prgm10_UltraSonic.py

```
import smtplib
import RPi.GPIO as GPIO
import Adafruit_DHT
from Adafruit_DHT import read_retry, DHT11
import time
from datetime import datetime, timedelta

# Define pins for light sensor, ultrasonic sensor, LEDs, and DHT11
light_sensor_pin = 13
trig_pin = 19
echo_pin = 26
led_pin_1 = 18   # First streetlight
led_pin_2 = 12   # Second streetlight
led_pin_3 = 3    # Third streetlight
led_pin_4 = 5    # Fourth streetlight
dht_pin = 4      # Pin connected to DHT11 data

# Set up GPIO pins
GPIO.setmode(GPIO.BCM)
GPIO.setup(light_sensor_pin, GPIO.IN)
GPIO.setup(trig_pin, GPIO.OUT)
GPIO.setup(echo_pin, GPIO.IN)
GPIO.setup(led_pin_1, GPIO.OUT)
GPIO.setup(led_pin_2, GPIO.OUT)
GPIO.setup(led_pin_3, GPIO.OUT)
GPIO.setup(led_pin_4, GPIO.OUT)

# Define the DHT sensor type
DHT_SENSOR = Adafruit_DHT.DHT11

# Define functions for reading light sensor, measuring distance, and reading humidity
def read_light():
    light_state = GPIO.input(light_sensor_pin)
    return light_state

def measure_distance():
    # Trigger ultrasonic sensor
    GPIO.output(trig_pin, GPIO.LOW)
    time.sleep(0.00001)
    GPIO.output(trig_pin, GPIO.HIGH)
    time.sleep(0.00001)
    GPIO.output(trig_pin, GPIO.LOW)

    # Measure pulse duration
```

Setting Spaces indentation mode for /home/pi/2.py





```
        GPIO.output(LED_PIN, GPIO.HIGH)  # Turn the LED on
        print(f"LED is ON at {on_time}")

    # Check if it's time to turn the LED OFF
    elif current_time == off_time:
```

```
/home/pi/Desktop/smart intelligence street light system/time1.py:8: RuntimeWarni
ng: This channel is already in use, continuing anyway.  Use GPIO.setwarnings(Fal
se) to disable warnings.
  GPIO.setup(LED_PIN, GPIO.OUT)  # Set the LED pin as output
Waiting for the specified times...
Current time: 12:15
Current time: 12:15
Current time: 12:16
LED is ON at 12:16
Current time: 12:16
LED is ON at 12:16
Current time: 12:17
Current time: 12:17
Current time: 12:18
```