IOT FINALS

Home Content Contact Others

SMART HOME ENERGY-SAVING LIGHT CONTROLLER

Gian - Aireen - Love

INTERNET OF THINGS

IOT FINALS

Covered Today

What's in it?

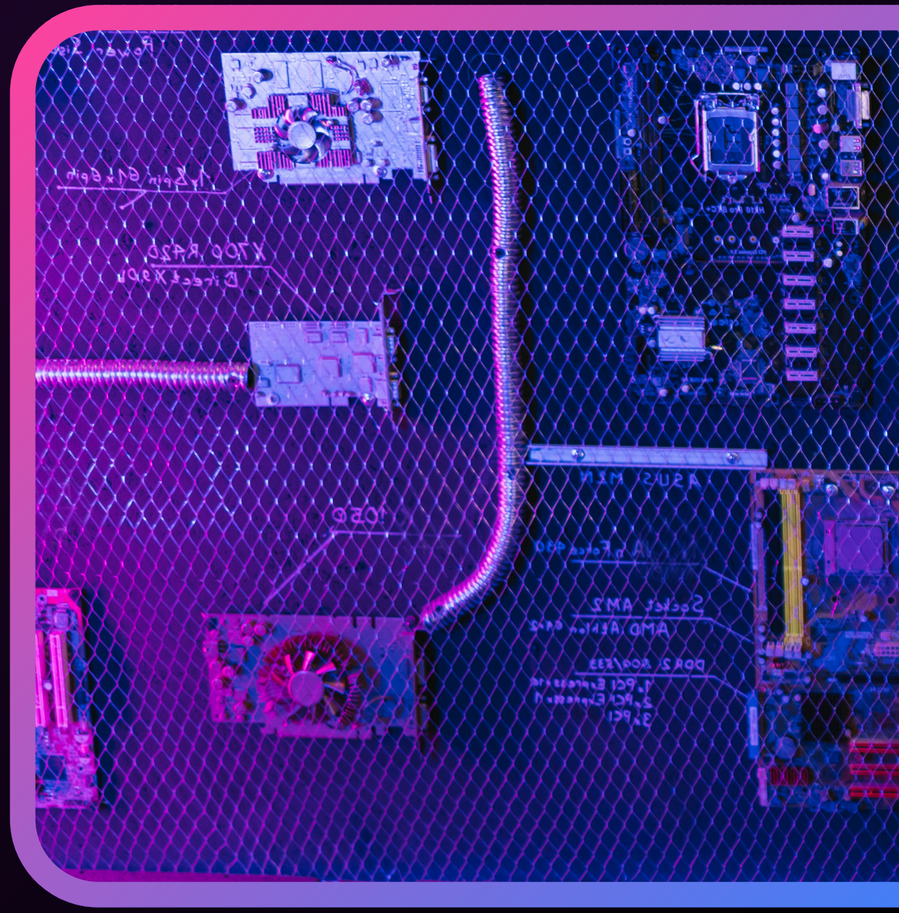
- Introduction
- Background of the project
- SDG Requirements
- Security Implementation
- Challenges Encountered
- Summary
- Critical Reflection

INTRODUCTION

Purpose and Features

Uses Raspberry Pi Pico W to control an LED light based on occupancy (motion sensor) and ambient light (photoresistor), with temperature and humidity monitoring for comfort.

- Automatic light control to reduce energy waste.
- Web interface for real-time data and manual control.
- Data sent to ThingSpeak for remote monitoring.



Background

- Homes account for ~30% of global energy use, with lighting contributing 10–15% of household electricity.
- Inefficient lighting (e.g., lights left on) wastes energy, increasing costs and environmental impact.
- IoT solutions can optimize energy use through automation and monitoring.

Selected SDGs

HOW DO WE GET THERE?

SDG 7: AFFORDABLE AND CLEAN ENERGY

- Reduces electricity consumption by activating lights (LED) only when needed (motion detected, low light).
- Example: Saves ~0.3 kWh/day per light (assuming a 60W bulb replaced by a 0.1W LED, used 5 hours less daily).

SDG 12: RESPONSIBLE CONSUMPTION AND PRODUCTION

- Promotes efficient resource use through automated lighting and user awareness via web interface tips.

SDG 11: SUSTAINABLE CITIES AND COMMUNITIES:

- Lowers urban energy demand by optimizing home lighting, reducing environmental footprint (Target 11.6).
- Enhances home livability with comfort monitoring (Target 11.7).

INNER WORKINGS OF THE PRODUCT

HARDWARE

- Core: Raspberry Pi Pico W (Wi-Fi enabled).
- Sensors:
  - Infrared Motion Sensor (GP15): Detects occupancy.
  - Photoresistor (GP26, ADC): Measures ambient light (0–1000 lux).
  - Temperature and Humidity Sensor (DHT22, GP14): Monitors room conditions.
- Actuator: Stepping Motor Driver (GP10–13) controls a simulated switch for an LED light (GP17).
- Indicator: Status LED (GP16) shows light state (on/off).
- Power: USB-powered, low-power mode for efficiency.

INNER WORKINGS OF THE PRODUCT

SOFTWARE

- Written in MicroPython.
- Logic:
  - If motion detected and light <150 lux, turn on LED and rotate stepper motor (10 steps clockwise, “on”).
  - If no motion for 5 minutes or light >300 lux, turn off LED and rotate stepper motor (10 steps counterclockwise, “off”).
  - Monitor temperature/humidity; alert if >30°C or >80% (ventilation needed).
- Web Server: HTTPS (or HTTP if simplified) on Pico W displays data and allows manual light control.
- IoT: Publishes data to ThingSpeak via MQTT every 5 minutes.
  - Diagram: Show wiring schematic (e.g., sensors to Pico W, stepper motor, LED).

RESULTS

- LOW LIGHT AND MOTION , MOVES THE MOTOR WHICH MIMICS THE OPENING OF A WINDOW OR CURTAI, AND TURNING OFF THE LIGHTS
- HIGH LIGHT, AND MOTION, DOES NOT MOVE THE MOTOR

IOT FINALS

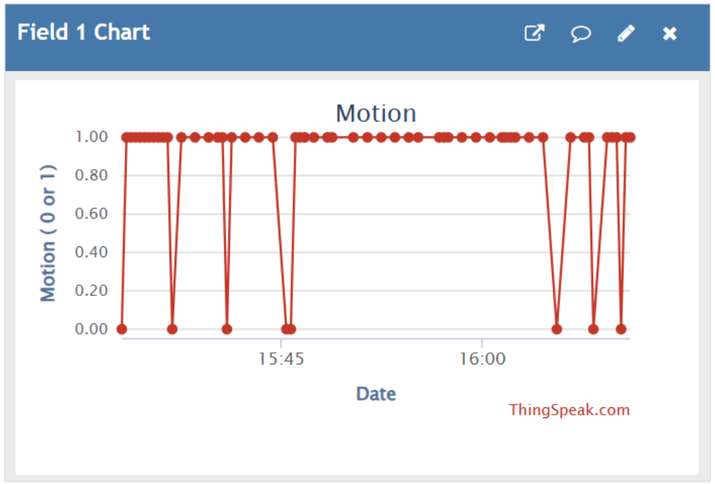
DEMO

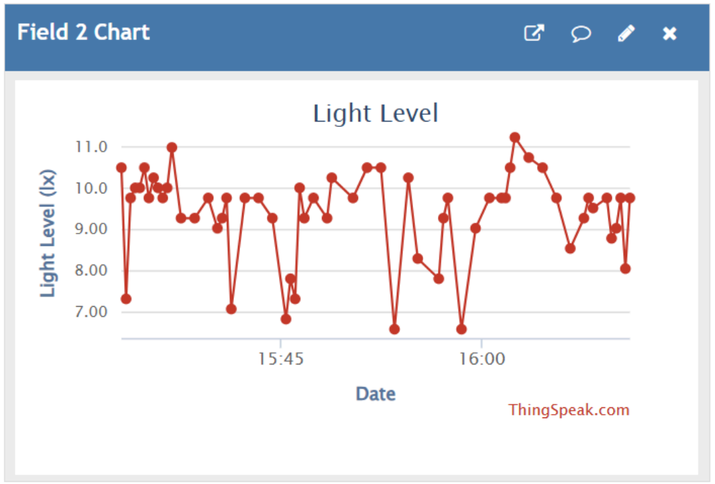

```
Sent to ThingSpeak: Motion=1, Light=9.8lx, Temp=23C, Humid=44%
Debug - Light: 9.5lx, Temp: 23C, Humid: 44%
Sent to ThingSpeak: Motion=0, Light=9.5lx, Temp=23C, Humid=44%
Debug - Light: 8.5lx, Temp: 23C, Humid: 44%
Sent to ThingSpeak: Motion=1, Light=8.5lx, Temp=23C, Humid=44%
Debug - Light: 9.8lx, Temp: 24C, Humid: 44%
Sent to ThingSpeak: Motion=1, Light=9.8lx, Temp=24C, Humid=44%
Debug - Light: 9.8lx, Temp: 24C, Humid: 44%
Sent to ThingSpeak: Motion=1, Light=9.8lx, Temp=24C, Humid=44%
Debug - Light: 8.8lx, Temp: 24C, Humid: 44%
Sent to ThingSpeak: Motion=1, Light=8.8lx, Temp=24C, Humid=44%
Debug - Light: 9.0lx, Temp: 24C, Humid: 43%
Sent to ThingSpeak: Motion=1, Light=9.0lx, Temp=24C, Humid=43%
Debug - Light: 9.8lx, Temp: 24C, Humid: 43%
Sent to ThingSpeak: Motion=0, Light=9.8lx, Temp=24C, Humid=43%
Debug - Light: 8.1lx, Temp: 24C, Humid: 43%
```

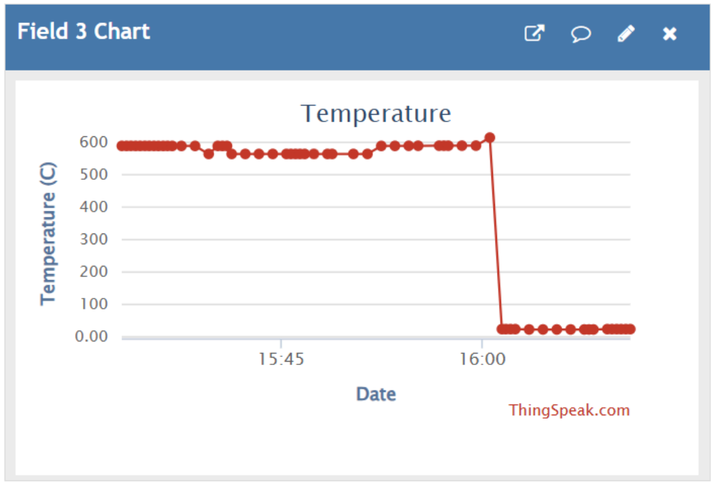
IOT FINALS

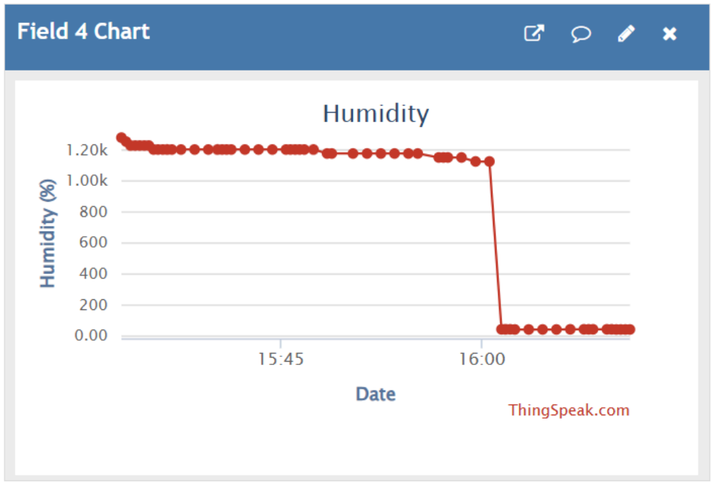
Home Content Contact Others

DEMO





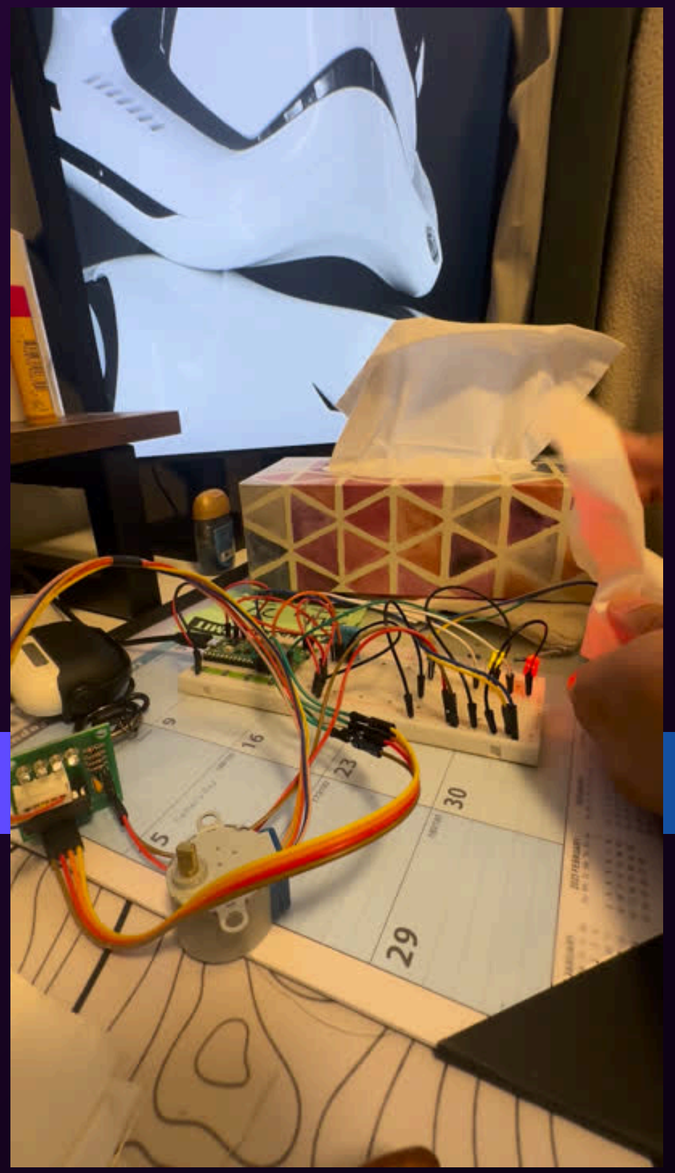




IOT FINALS

Home Content Contact Others

DEMO



SECURITY IMPLEMENTATION

DATA TRANSMISSION

- TLS encryption for MQTT communication with ThingSpeak (port 8883).
- Uses ThingSpeak's MQTT broker with secure API key for data publishing.

AUTHENTICATION

- ThingSpeak channel protected by a write-only API key, preventing unauthorized access.
- Web server uses basic HTTP authentication (username/password) to restrict access to household users.

LOCAL SECURITY

- No physical display, so local access is via the web interface, protected by password.
- Credentials stored securely in MicroPython code (hardcoded for simplicity, with instructor approval).

LIMITATIONS

- Pico W's limited processing power restricts advanced encryption (TLS and basic auth are sufficient).
- Regular API key rotation recommended for long-term use.

Challenges Encountered

Hardware Challenges:

- Photoresistor Calibration: Difficulty converting ADC values to accurate lux readings; simplified to relative thresholds (<30% = dark, >70% = bright).
- Stepping Motor Setup: Limited torque for precise dimming; used simple on/off switch simulation with LED.
- Component Limitation: No bulb available; used LED to simulate lighting, confirmed with instructor.

Challenges Encountered

Software Challenges:

- Web Server: Setting up HTTPS on Pico W was complex due to resource constraints; used HTTP with instructor approval for simplicity.
- Wi-Fi Stability: Occasional dropouts required reconnection logic in code.

Challenges Encountered

IoT Challenges:

- MQTT Setup: Initial issues with ThingSpeak credentials; resolved by following official documentation.
- Data Frequency: Balancing 5-minute intervals to avoid exceeding ThingSpeak’s free-tier limits.

Challenges Encountered

Time Management:

- 3-week timeline was tight; prioritized LED over bulb and omitted thermistor to save time.
- Group coordination required daily check-ins to align tasks.

Summary

- Developed an IoT system to control home lighting (LED) using Raspberry Pi Pico W, sensors, and a stepper motor.
- Integrated with ThingSpeak for data monitoring and hosted a web interface for user interaction.
- Aligned with SDGs 7, 12, and 11 by reducing energy use, promoting responsible consumption, and supporting sustainable urban homes.
- Successfully met project requirements within 3 weeks, despite hardware and time constraints.

IOT FINALS

Home Content Contact Others

Critical Reflection

Applications in life

- Strengths
- Weaknesses
- Lessons Learned
- Future Improvements

IOT FINALS

Strengths

EFFECTIVE ENERGY-SAVING LOGIC

- USER-FRIENDLY WEB INTERFACE FOR REAL-TIME MONITORING AND CONTROL.
- CLEAR SDG ALIGNMENT, ENHANCING HOME AND COMMUNITY SUSTAINABILITY.

Weaknesses

- Lack of a physical display limited local interaction; web interface required Wi-Fi access.
- LED less realistic than a household bulb for demo; limited by available components.
- Simplified security (HTTP instead of HTTPS) due to Pico W constraints.

IOT FINALS

LESSONS LEARNED

Importance of early testing to identify hardware/software issues.

Value of teamwork in dividing tasks (hardware, coding, IoT, documentation).

Need for clear instructor communication to align on simplifications (e.g., LED, HTTP).

FUTURE IMPROVEMENTS

Bulbs

Integrate a real bulb with a dimmer circuit for realism.

Security

Enhance security with full HTTPS and advanced authentication.

QUESTIONS?

WE'D LOVE TO HEAR YOUR THOUGHTS

IOT FINALS

Home Content Contact Others

Smart Home Energy-Saving Light Controller

THANK YOU

IOT FINALS