Lighting Control Based on Speaker Presentation

Jaiwanth Mandava*, Kalyan Adithya† and Vivek Pamnani‡
Team 25, International Institute of Information Technology
*Email: *jaiwanth.mandava@students.iiit.ac.in, †kalyan.adithya@students.iiit.ac.in, ‡vivek.pamnani@research.iiit.ac.in*

***Abstract*—Energy and Time : Both are precious resources that we often take as granted. Hence, it is paramount that we make efforts towards energy conservation and save the otherwise missed time, for time is fleeting. In this paper, we document a proposed project and analyze its impact on environment. In this project, the step we chose to make had to be a step which is cost-efficient and convenient for the stakeholders, easy enough to implement, and something that creates a deep impact towards saving Energy and Time.**
Teaching has revolutionized, it is easier for teachers now with presentations. However, we cannot afford to interrupt a class and waste time over ambient lighting. So, using this project we can automate that process. In a typical classroom, about one-fourth of the lights are focused on the screen. Thus we can save up to 25% of the net power consumption by lights in a typical classroom. Coupled with a low-cost implementation, the potential benefits of up-scaling this project seem exceedingly large.

***Keywords*—Energy, time, conservation, ambient lighting, automate, low-cost implementation.**

Developer's Document

1. Introduction

1.1. Problem Statement, Scope, and Purpose

Often professors and students find the need to switch the lights during a presentation for immersive focus. Hence, professors have to interrupt the class to switch the lights which proves to be inconvenient and obstructs the flow of the class. Automation of this switching of lights could save time and ensure the flow of the class. As an added benefit, it can help save the otherwise wasted power during long presentations.

Further, it is not possible for students to attend every class for various reasons resulting in a student's loss. What would be helpful here, is the ability to stream an ongoing lecture in such as case. Along with automation of lights, we are able to trigger a camera which will stream the ongoing lecture on our server.

1.2. Overview

Our project achieves the above objective by utilising an LDR sensor. The sensor detects the light emitting from the projector and sends a signal to a micro-controller which switches the lights accordingly and triggers the camera.

2. Design Document

2.1. System Requirements

- The Enclosure:
  - 5V adapter for the MCU.
  - 230V supply for the Relay module.
  - Access to lighting switches.
  - Suitable insulation for the otherwise exposed 230V wires.
- ESP32-CAM:
  - 5V adapter.
  - Suitable mounting provision.

2.2. System Specifications

- ESP32-WROOM-32D (Micro-Controller Unit):

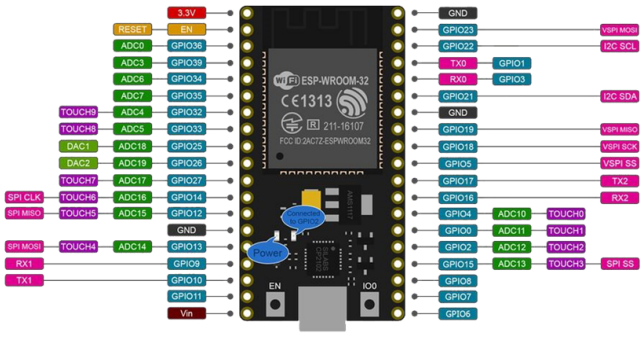

Figure 1. Pin Diagram: ESP32-WROOM-32D.

Categories	Items	Specifications
Wi-Fi	Protocols	802.11 b/g/n (802.11n up to 150 Mbps)
		A-MPDU and A-MSDU aggregation and 0.4 µs guard interval support
	Frequency Range	2.4 GHz ~ 2.5 GHz
Bluetooth	Protocols	Bluetooth v4.2 BR/EDR and BLE specification
	Radio	NZIF receiver with –97 dBm sensitivity
		Class-1, class-2 and class-3 transmitter
		AFH
	Audio	CVSD and SBC
Hardware	Module Interfaces	SD card, UART, SPI, SDIO, I2C, LED PWM, Motor PWM, I2S, IR, pulse counter, GPIO, capacitive touch sensor, ADC, DAC
	On-chip sensor	Hall Sensor
	Integrated crystal	40 MHz crystal
	Integrated SPI flash	4 MB
	Operating voltage	3.0 V ~ 3.6 V
	Operating current	Average: 80 mA
	Operating temperature	–40 °C ~ +85 °C
	Moisture sensitivity level (MSL)	Level 3

Figure 2. Specifications: ESP32-WROOM-32D.

- Light Dependent Resistor (LDR):

Items	Specifications
Operating Voltage	3.3 V ~ 3.5 V
Output	Analog
Operating Current	15mA

Figure 3. Specifications: roboCraze LDR.

- ESP32-CAM with OV2640 Camera Module:

Figure 4. Pin Diagram: ESP32-CAM.

Items	Specifications
Operating Voltage	5V
WiFi protocol	IEEE 802.11 b/g/n/e/i
Bluetooth	Bluetooth 4.2 BR/EDR and BLE
Operating Temperature	-20°C ~ 85°C
Dimensions (mm)	40.5 x 27 x 4.5

Figure 5. Specifications: ESP32-CAM.

2.3. Stakeholders

- Students and professors of the institution using the classroom where the project is deployed will be benefited by the automated lighting control.
- The recordings/pictures of the presentation taken using ESP32-CAM and mobile camera can be utilised by students for reference purposes.
- In addition to this, the college management will save approximately 22.7% of the net light power consumption for that particular classroom.

2.4. Main Components

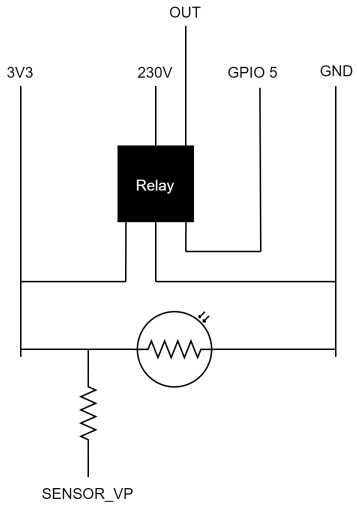

Figure 6. The Circuit Diagram.

Figure 6 decribes how all the main components are connected.

- ESP32-WROOM-32D (Micro-Controller Unit):
  ESP32-WROOM-32D is a powerful, generic Wi-Fi + BT + BLE MCU module that targets a wide variety of applications, ranging from low-power sensor networks to the most demanding tasks, such as voice encoding and music/video streaming.

- Light Dependent Resistor (LDR):
  An LDR is essentially a photresistor which is made of a high resistance semiconductor. On impact of photons, the electrons jump to the conduction band and allow flow of electricity. Thus, resistance is inversely proportional to the light intensity.

- Relay Module:
  A relay (figure 7) is an electromagnetic switch operated by a relatively small electric current that can turn on or off a much larger electric current. Works like an 'electric lever': switch it on with a tiny current and it switches on another appliance using a much bigger current.

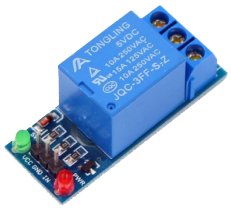

Figure 7. The Relay Module.

- ESP32-CAM with OV2640 Camera Module:
  The ESP32-CAM (figure 8) is a compact camera module with the ESP32-S chip. We use it for media streaming. In addition to the OV2640 camera, it can store captured images and recorded videos on a mounted microSD card.

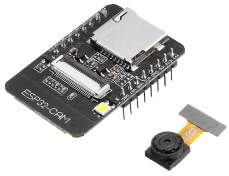

Figure 8. ESP32-CAM.

2.5. Design Details

2.5.1. Conceptual Flow and Entity Interaction.

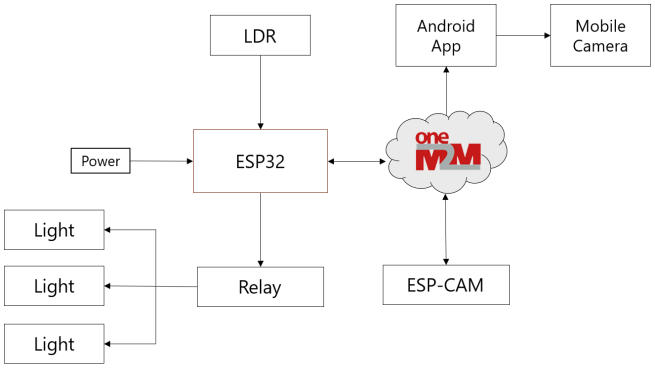

Figure 9. The Block Diagram.

Figure 9 describes a flow chart of the conceptual model of the project. Here, oneM2M is the common Service Layer API which facilitates the communication between IoT devices.

The ESP32 board, powered by a 5V source, interprets the analog readings from the LDR and accordingly sends a signal to the Relay module. On switching, the relay then toggles the lights and the board then posts data (such as projector status) on the oneM2M server.

This data can be retrieved by a set of APIs by the CAM module and the Android application. Hence, whenever the projector is turned ON, the Android application and the CAM module then trigger their respective cameras for recording.

2.6. Operational Requirements

2.6.1. System Needs.

- The ESP32 board requires a 5V power source via a Micro USB port.
- The ESP32 CAM module requires a 5V power source, via a Micro USB port, at a suitable mounting provision to record the class.
- An Android device, if used for recording, requires a suitable mounting provision to record the class. No power source required, except for periodic charging of the battery.
- The oneM2M server is to be hosted on a network with sufficient bandwidth for uninterrupted streaming. Data on the server can either be manually be accessed either via logging in to the server using a web browser or via the Android application on an Android device.
- Connections to the switchboards of the room with sufficient insulation to prevent accidental electrical shocks.

2.6.2. UI Design.

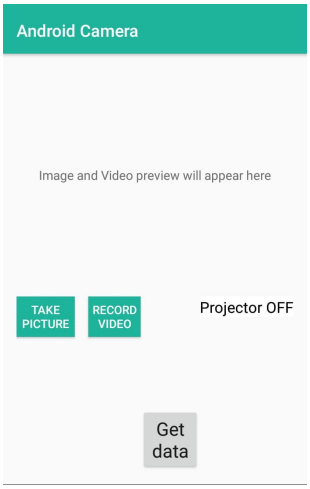

Figure 10. UI in the application.

Figure 10 shows the UI of the android app that is triggered whenever the projector is turned ON.

- **Get Data**: To get the current status of the projector.
- **Record Video**: To manually turn ON and OFF the recording of the presentation irrespective of the status of the projector.
- **Take Picture**: To take pictures during the presentation.

2.6.3. Analytical System.

In order to analyse the data in oneM2M server we have a script running which gets the latest instance of data and stores locally.

This data can now be used to generate graphs with interesting insights like the one shown in the figure 11.

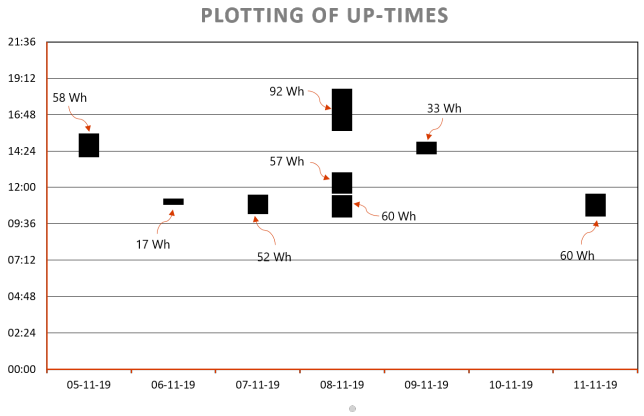

Figure 11. Up-time/Power Consumption Analysis

User's Document

3. Introduction

3.1. Objective

This document provides comprehensive guidelines and step-by-step instructions for working with the project: Lighting control based on speaker presentation.

3.2. Scope

Often professors and students find the need to switch the lights during a presentation for immersive focus. Hence, professors have to interrupt the class to switch the lights which proves to be inconvenient and obstructs the flow of the class. Automation of this switching of lights could save time and ensure the flow of the class. As an added benefit, it can help save the otherwise wasted power during long presentations.

Further, it is not possible for students to attend every class for various reasons resulting in a student's loss. What would be helpful here, is the ability to stream an ongoing lecture in such as case. Along with automation of lights, we are able to trigger a camera which will stream the ongoing lecture on a server.

4. Operational Requirements

4.1. Hardware Requirements

- The **ESP-32** board requires a constant 5V power supply through its micro-USB adapter.
- **Light Dependent Resistor (LDR)** which reports light intensity in analog values.
- **ESP32-CAM** which is triggered whenever projector is turned ON.
- **Relay** module which is used for switching lights that operate on 220V based on 3.3V input from ESP-32.

4.2. Software Requirements

- Arduino Software (Version - 1.8.10)
- Drivers for ESP32
- Operational code for taking data from sensor and switching lights can be found **here**. This code has to be flashed onto ESP32 micro-controller using arduino software.
- Code for getting latest instance of data from the server and triggering camera accordingly can be found **here**.
- Android app which retrieves current status of the projector and provides functionality like recording video and taking picture can be found **here**.
- Android Studio

5. System Working Model

5.1. Base State

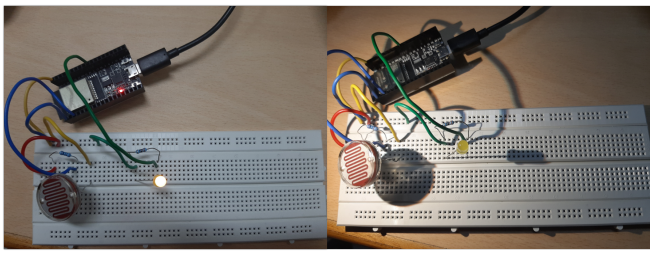

Figure 12. Initial State as a Prototype

Figure 12 depicts the testing phase of the project as a prototype. The initial phase corresponds to correctly configuring the circuit and calibrating the sensor by doing a week's analysis of a chosen deployment site.

5.2. Working State

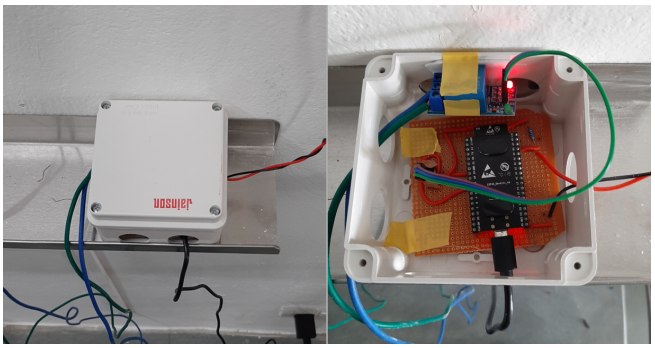

Figure 13. The Final Stage

Figure 13 depicts the final deployment stage of the project. The Relay module will glow red (as shown) when the projector status is OFF, and green when the projector status if ON.

At the same time, the ESP32 board interprets the values sent by the LDR sensor, and posts data to the oneM2M server every time the Relay is toggled. The camera then retrievess the data from the server and toggles the recording automatically (also the Android phone with app installed).

The server data can be obtained on the Android phone via the application just by the click of a button.

IoT Project Components

6. Communication

OneM2M is used for communication between the Node and Server. It has the following uses:

- Security and privacy aspects(authentication, encryption, integrity verification).
- Identification and naming of devices(for differentiating different devices).
- Data aggregation, buffering in case of missing connectivity and synchronisation upon connectivity re-establishment.
- Group management and application and data discovery functions.

The status of the projector is posted to the server(OneM2M) by the ESP32 board from where the Camera module and also the Android app fetches the status of the projector and triggers their respective cameras accordingly to record the presentation. Refer 9

7. Software Specifications

- To select the required WiFi network, change the following lines in the source code:

```
char* wifi_ssid =     SSID     ;
char* wifi_pwd =    PASSWORD    ;
```

  Change SSID and PASSWORD to the SSID and password of your own WiFi network, and make sure to flash the code onto the ESP-32 board again after making the required changes.
- To upload the data to a custom server, change the following lines in the source code:

```
String cse_ip = IP_ADDRESS;
String cse_port = PORT;
```

  These lines can be changed appropriately to send the data to a specific OneM2M server in the form of a POST request. Once again, make sure to flash the code onto the ESP-32 board again after making the required changes. Do the same for the code to be flashed on to ESP32-CAM.
- Follow steps mentioned here to run the app code given on either an actual device or an emulator.

8. Data Handling Model / Analysis Framework

Data from the server can be scraped out using a simple python script and we can store that in an excel sheet/text file through which we make day or date-wise graph for analysis as shown below. Code for it can be found here.

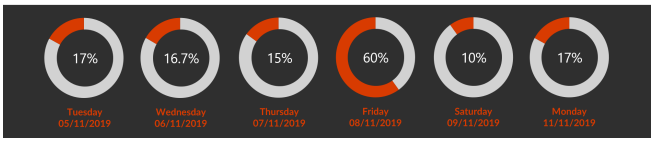

Figure 14. Percentage Up-time Analysis

DAYS	Percentage	Up-Time (minutes)	Working Time
Tuesday	17%	87	510
Wednesday	16.7%	85	510
Thursday	15%	77	510
Friday	60%	310	510
Saturday	10%	51	510
Monday	17%	86	510
TOTAL	22.7%	696	3060

Figure 15. Percentage Up-time Analysis Per Week

Thus we can save about 22.7 % of the net power consumption due to lights.

Acknowledgments

The work described in this project was conducted within IIIT Hyderabad and also funded by IIIT Hyderabad under the course Embedded Systems Workshop EC3.202. We sincerely thank Prof.Prakash Yalla and Teaching assistants Sushanth Reddy and Shivani Chepuri for their support and guidance through the whole project. We also like to thank the Embedded Systems Workshop professors for their guidance throughout the project.

9. References

[1] http://www.onem2m.org/about-onem2m/why-onem2m
[2] https://www.arduino.cc/en/main/software
[3] https://www.fecegypt.com/uploads/dataSheet/1522335719_relay%20m
[4] https://randomnerdtutorials.com/esp32-cam-video-streaming-face-recognition-arduino-ide/
[5] http://www.esp32learning.com/code/esp32-and-ldr-example.php