LAPTOP LENDING STATUS SYSTEM

Team Information - sddec20-02

Client
Parks Library Tech Lending Center

Faculty Advisor
Md Maruf Ahamed

Team Members

Farouk Al-Obaidi	farook@iastate.edu	S E
Ryan Ray	ryanr@iastate.edu	S E
Aaron Thune	adthune@iastate.edu	CPR E
Camden Thomas	camdent@iastate.edu	S E
Zoe Sanders	zsanders@iastate.edu	S E
John Wagner	jhwagner@iastate.edu	CPR E

Problem Statement

The Parks Library Tech Lending Center lends over 200 devices to the ISU student body. Larger devices such as laptops are stored on racks (Figure 1) and smaller devices such as chargers are hung in bags on metal bars. (Figure 2) Employees need to know what the status of each device is (e.g., Checked Out, Overdue, etc). Currently they use colored stickers to label devices. This system is time-consuming to keep up to date.

Proposed Solution

This project aims to replace the printed sheets of paper with RGB LEDs. Each RGB LED will display a color that will correlate to the status of each laptop.

Figure 1 (Right): Racked larger devices. Figure 2 (Left): Bagged smaller devices.

Design Requirements

Functional Requirements

-RGB LEDs that display the status of various electronic devices.
-Each device has a corresponding LED.
-LEDs updates with status changes in ISU Libcal API in real time.

Non-Functional Requirements

- System is scalable (to adapt to more devices).
- Adding new devices and racks to the system is a simple process.
- System must be able to support more than 500 unique devices.
- Each rack module must be able to support at least 20 LEDs
- All Raspberry Pi Code should be in Python
- Status-color mappings should be able to be changed.

Operating Environment

Parks Library Tech Lending Center device room. Air-conditioned and well-maintained with plenty of access to outlets and internet.

Intended Uses/Users

The intended use of this system is to monitor the statuses of devices in the Parks Library Tech Lending Center. Intended users are the employees of Tech Lending Center.

Design Approach

Software

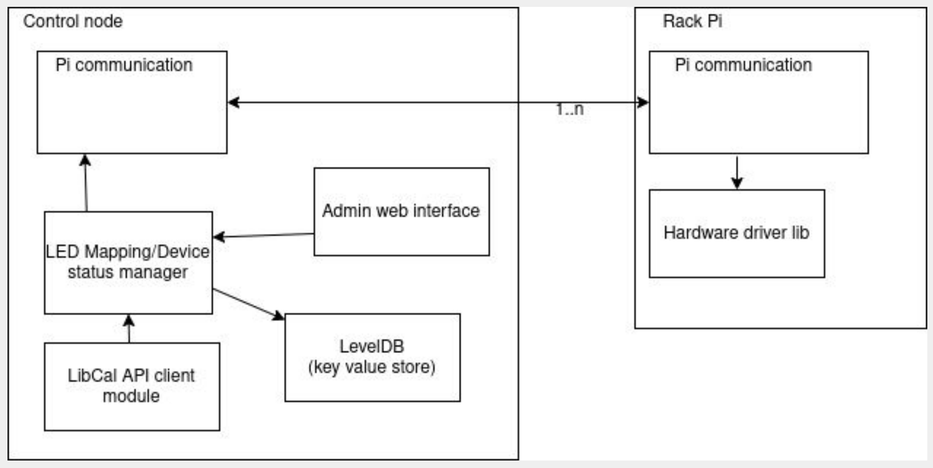
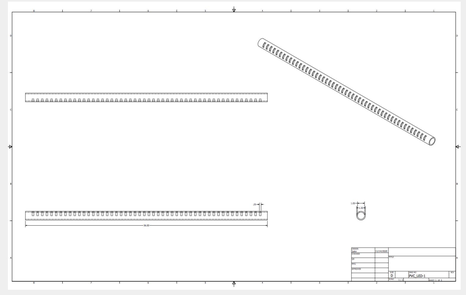


Figure 3: Software architecture concept sketch

Hardware



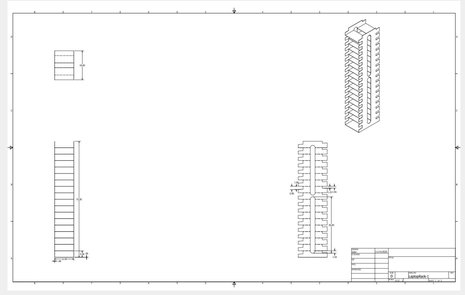
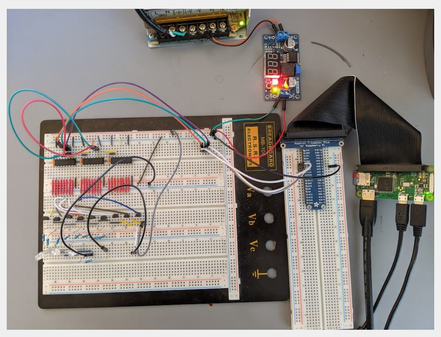


Figure 4 (left): Enclosure for LEDs. Figure 5 (right): LED enclosure on rack.



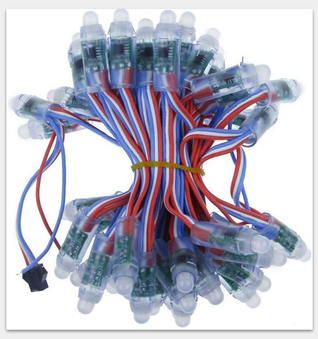


Figure 6 (far left): LED Driver Circuit which can control 24-50 RGB LEDs.
Figure 7 (right): Individually addressable LEDs used for device status representation

Technical Details

Software

Control Node: Central server that handles the following via the following modules. ***Docker***

- Pi Communication: Sends status updates to Rack Pis and handle new Rack Pis. ***Python***
- LED Mappings Manager (LMM): Receives and forwards updates from LibCal API client module and Admin Web Interface to the Pi Communication module. Also receives data from Pis and stores in LevelDB. ***Python***
- LibCal API Client Module: Receives updates from LibCal API and forwards these updates to the LMM. ***Python***
- Admin Web Interface: Web UI to easily view device statuses and Pi health. Also allows users to add, edit and delete status-to-color mappings. ***nginx, Javascript/HTML/CSS***

Hardware

- Raspberry Pi Zero W on each rack/bar to control LEDs.
- Shift registers controlling transistors connecting power to LEDs
- Autodesk Inventor CAD Tool for enclosure design

Testing

Software

- PyTests
- Requests
- Python Unit Testing
- Docker Containers to simulate a Pi-LED network.
- Python Mock Tests

Resources

- Budget of roughly $500
- Input from Clients
- Raspberry Pi Zero Ws
- Equipment from ETG

Standards

- HTTP and JSON standards.
- WS2811 LED Standard

Acknowledgements

We'd like to thank the Parks Library Tech Lending team for their great idea and continued feedback and support. We would also like to thank our faculty advisor Md Maruf Ahamed for his support and insight