Learning Holiday Lights Project

Team Members: Ash Singh, Chris Woods, Jacob Martin, Joyeux Noel, Mitchell Wadle, Ty Gardner
Client and Advisor: Dr. Daniels
sdmay21-48

Problem Statement

Currently people are not getting the full holiday experience when they light up their Christmas tree because the light patterns are static and cannot be manually configured by the user. Therefore, Team 48 has decided to enhance the experience of Christmas by creating a unique way to visualize the patterns that get displayed on the tree.

Solution

String of controllable color LEDs (cone(tree), cylinder, sheet). Calibration is run and takes images of the lights using a Raspberry Pi camera mounted on a tripod. Images then used to produce a 3-D mapping of each of the lights. With calibration completed, lights are individually programmed to display different colors, or sequence of colors, using a web-tool. The coordination of each of these pieces is done with another Raspberry Pi, that is always connected to the lights.

Subsystem Description

Calibration

- Take photo of the LEDs using Camera Pi
- Use image processing to locate xy coordinates
- Send these coordinates to TreePi and create json files

Pattern Rendering

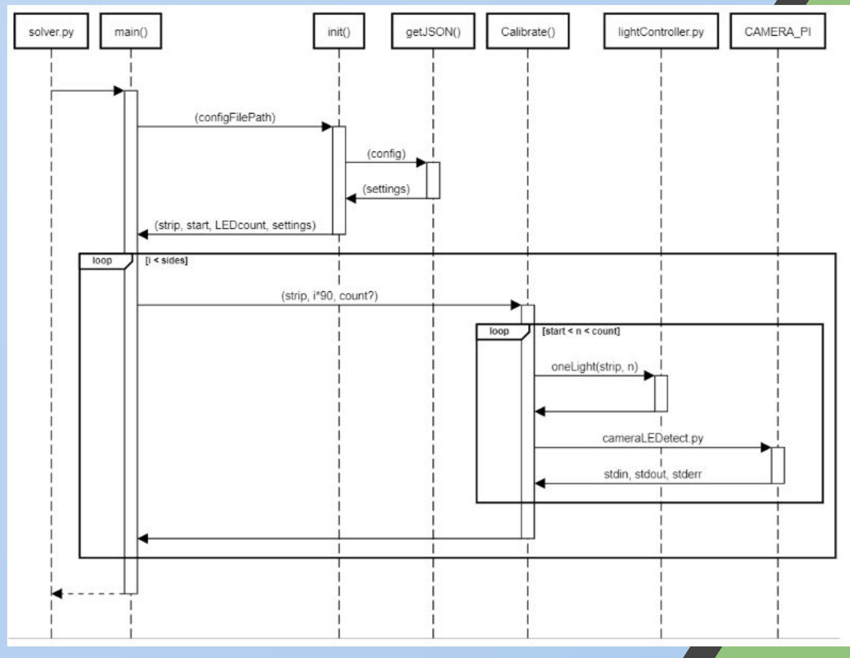
- Map patterns from user-defined 2D images to 3D cylindrical light coordinates

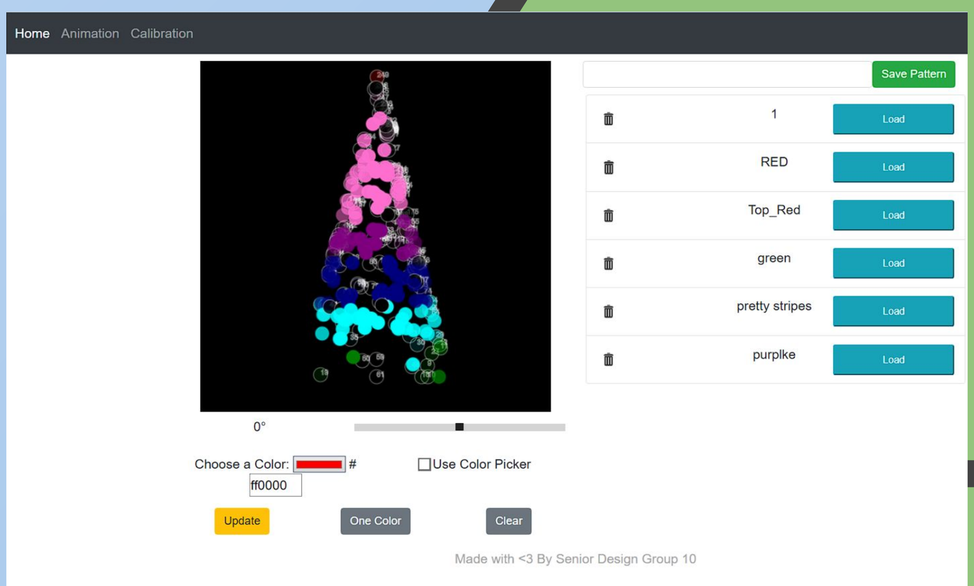
Lazy Susan

- Rotate the tree by 90⁰ four times during calibration - once for each face

Web application

- Calibrate LEDs
- Animate Tree with patterns





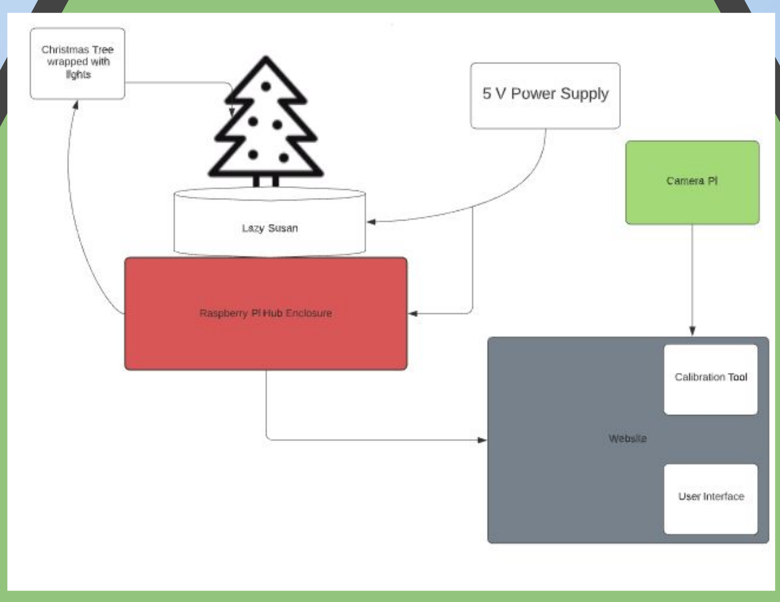
Design Requirements

Functional requirements

- Easy to install
- Controlled remotely via web server
- Operate independently
- Calibrated with camera
- Calibration is easy to complete

Operating environment

- Calibration works regardless of background
- Housing protects components from weather, applicable both indoors and outdoors



Engineering Constraints

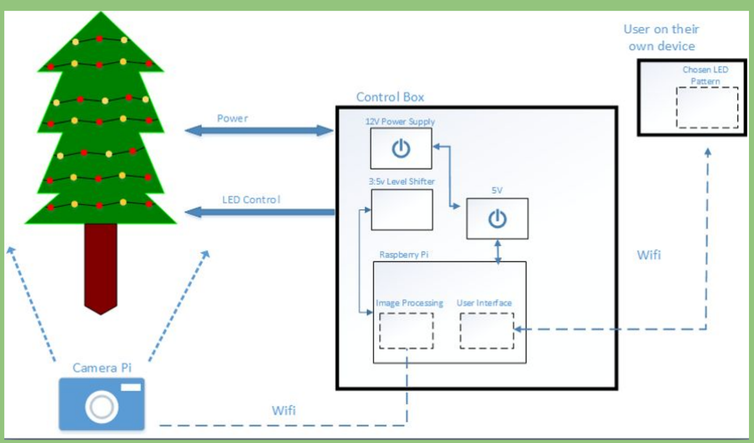
- Camera Pi and Tree Pi don't have internet access
- Work around previous teams incomplete calibration process
- Regulate voltages going to the stepper motor to protect the Tree Pi

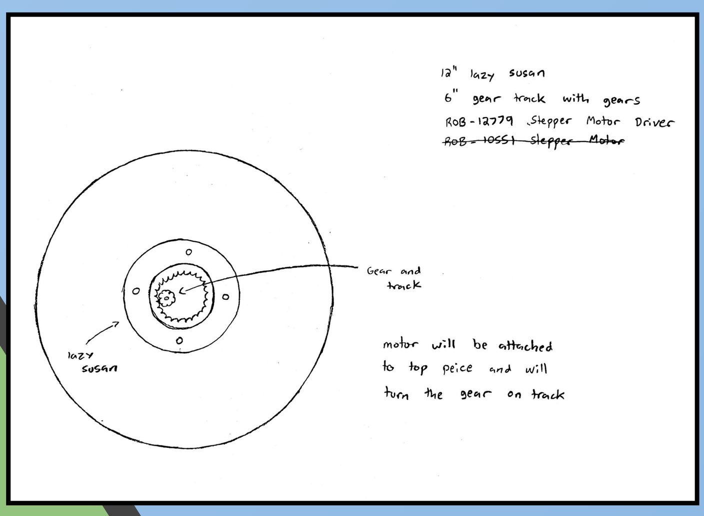
Non-functional requirements

- Material cost under $100
- Calibration takes less than 1 hour
- Time to update patterns is less than 3 seconds

Design Approach

- Two Raspberry Pis:
  - Tree Pi
    - Controls the lights array
    - Turns motorized lazy susan
    - Hosts the web application
  - Camera Pi
    - Takes pictures of each light on the tree
    - Finds their positions for calibration purposes
- Patterns saved as circular images which are mapped top-down onto the tree
- Lights data, calibration, and patterns all secure on the Tree Pi, only accessible through connection to the Wireless Access Point or direct plugin to the Pi.





Technical Details

Hardware

- Two Raspberry Pi Model 3B/3B+
- Control Box
  - 12 V, 30 A power supply
  - LM2896 SMPS to step down 12V-6V
- Raspberry Pi Camera V2
- WS2811 LED strip
- ROB-12779 Stepper driver and ROB-10551 12V Bipolar stepper motor

Software

- Apache HTTP Server
  - PHP
- Python 3
  - Paramiko(SSH)
  - OpenCV(computer vision)
  - Numpy(data processing)
  - Rpi_ws281x (LED control)
  - ZeroMQ(message queue)

Intended Users & Uses

This project was designed so that any and all users would be able to set up, calibrate, and update everything with ease.

This system is intended to be used exclusively indoors. The system, as it is, is not waterproof and would be destroyed outdoors. The controls box should be placed on the top of the lazy susan and should remain uncovered to allow for proper airflow.

Testing Environment

There are a few things to consider when setting up the testing environment. First, the tree needs to be placed away from windows or reflective backgrounds. This helps to reduce the error reflections sometimes picked up from the reflective surfaces. Next, the camera needs to be place on the tripod 3 feet off the ground and far enough from the tree so that the camera fits the tree in the picture. Distance between camera and tree varies depending on how tall the tree is.

Hardware Testing Strategy

Testing of the hardware consisted of using an oscilloscope, voltmeter for testing components and making sure everything would work properly. Testing the motor required having the tree on the lazy susan and assuring that the motor could turn the weight of the tree.

Standards

IEEE 802.11 - WiFi Communication
1789-2015 - IEEE LED safety standard
IPv4 - Routing internet traffic
I2C - Multi-controller system

Software Testing Strategy

Testing the software includes verifying that all of the calibration components work together and that all of the code runs properly. Testing included running individual tests on different components integration testing when we put everything together.