Light Party

CS M117

Kevin

Byunghun Lee; ehtk0502@gmail.com

Yanyin

Bey-Ru Hsu

William Tsang

Introduction

For this project, our group came up with a hardware and software-based product. The product that we came up with is a medium-sized standing LED lamp that can be manipulated wirelessly. Some of its functionalities include switching the LED lamp on and off, creating random light patterns, and three hundred and sixty degree rotation capabilities. The motivation behind our design for the lamp was to build something using products that can be found in everyday life. We carried out this plan using easily obtainable objects such as paper clips and an array of LED lights. Raspberry Pi 3 model B was used to implement the wireless technology. Raspberry Pi 3 allows us to connect to the LED lamp through WiFi using any mobile or non-mobile device that supports WiFi wireless connection. In addition, we also set up a web server that communicates with Raspberry Pi through previously stored execution codes.

Motives

As our team was comprised of members with a background in electrical engineering, it was only natural to consider building our project around something that involved both software and hardware usage. In recent years, there has been a definite increase in the number of smart home appliances that can be controlled remotely. Following this trend, we decided to make our own version of a smart home product that is simple in design and can be used by everyone with ease. As we contemplated on what product would be the best fit for this project, we knew that an LED lamp was it. When used at home, the LED lamp can be controlled wirelessly without having to move

from your seat or having to touch the product. The LED lamp can also be used to create a refreshing and upbeat environment in any room — essentially functioning as a mood enhancer that gives you a little bit more energy to last throughout the day.

Technical Design

The LED cube we wanted to make consists of 64 LEDs in a 4x4x4 cube matrix. To light up an LED, we need to generate a positive difference in voltage between the anode and cathode of the LED as the LED is a forward biased electrical component. We determined that GPIO pins, when set to be ON or OFF, had a voltage output of 3.3V and 0V respectively and a current of 3mA. As our LED specifications allowed 3.3V potential differences and up to 5mA of current, we did not have to create voltage divider networks or design any voltage/current reducing circuitry.

The primary issue to solve in our design is that we need a total of 64 pins and a common ground to control all of the LEDs uniquely; however, we have only a maximum of 26 GPIO outputs. Careful consideration was taken to identify methods of reducing GPIO outputs. We ended up designing four 4x4 layers of LEDs stacked together, with each layer having its own unique ground. For each column of LEDs, we connected the anodes together.

We would set the initial (or ‘off’) state of the LED cube to have all layer cathode GPIO pins output an ON signal (or 3V high) and all column anode GPIO pins output an OFF signal (or 0V low). To turn on a particular LED on, we would set the LED’s ground pin to OFF and the anode for the LED to ON, which would allow current flow through only that pin. Initially we designed for the cube to utilize a multiplexer to select which ground cathode to connect with the 0V common ground pin; however, the multiplexer component was causing problems, that is why

the GPIO ground layer method was developed. Using the above procedure, we could display any combination of LEDs in any particular layer of the cube.

We used both calculations in LED light attenuation and the Flicker Fusion Theory to design a method to display any combination of LEDs. The Flicker Fusion Theory is the same theory behind laptop and TV screen refresh rates, for which large flickering rates are unnoticed by human eyes and determined to be lit continuously. LED light after the stopping of a voltage source also continues to provide light but attenuates very quickly. These two ideas mean that if we flash between the layers, (display one layer of LEDs, then the next layer, and the next layer, and finally the last layer) we could create the illusion that a unique combination of LEDs on each layer were lit up. This is the theory and design behind the electrical components utilized in crafting the LED cube.

Implementation

In order to implement our LED cube, we used paper clips, copper wiring, and LEDs. The paper clips acted as a great building material as it was cheap, conductive, and provided much needed structural support that our thin copper wire lacked. Many of the paper clips, other than for structural support, were used in the construction of the four ground planes. Each of the four planes had 16 LEDs soldered onto it in a grid alignment, with the cathodes attached to act as ground. The four ground planes were then connected together with four more paper clips, one at each corner of the grids running perpendicular to the planes. With some space in between each grid, this allowed us to form the frame of the 4x4x4 cube. The LEDs in each column of the grid were connected to the same anode signal with copper wiring so that only 20 GPIO pins (16 anode connections for 16 columns, 4 cathode connections for the 4 planes) would be needed to control 64 (16 LEDs per plane, 4 planes) individual LEDs.

Aside from the LED lamp itself, we also had to implement the WiFi and functionality onto the Raspberry Pi 3. This process started with installing Raspbian OS onto the Pi's microSD card. Raspbian would later be needed to install several libraries during the setup process that we used in the implementation of our code. After installing Raspbian, we connected the Pi to a monitor with HDMI input, and then plugged in a mouse and keyboard to run commands on the Pi. The first command we ran was to allow it to connect to a WiFi network. Once that was done we were able to disconnect the Pi from the monitor and disconnect the mouse and keyboard. Since it was connected to WiFi we could SSH into the board as well as use VNC. These two together allowed us to remotely control the Raspberry Pi over the WiFi network. Once this was setup we installed the WiringPi and Apache HTTP libraries needed to setup instructions for our lamp. Then we coded our commands for the LEDs seen in the LED.py file as well as instructions

for the motor seen in motor.py. Both files utilize the WiringPi library since we needed to send high and low signals to the GPIO pins on the Pi itself. High and low signals on the GPIO pins allow us to make potential differences that dictate when each specific LED is lit up depending on which GPIO pin signal is high for one of the 16 anodes, and which GPIO pin signal is low for one of the 4 cathode ground planes. After the LED and motor instructions were written, we wrap up implementation by building a webpage that allows other devices on the network to control the Pi without having to use SSH or VNC. This is when we use the Apache HTTP server we installed onto Raspbian OS. Once the webpage is setup with Index.html and Gpio.php, we are presented with a webpage that allows us to turn the LEDs off, on, or make them blink in a sequence. We can load the webpage by entering the Pi's static IP address into a web browser. An example of this can be seen in our demo video.

Wireless Technology

Wireless Technology that we chose to use for our project was WiFi. Since the Raspberry Pi is like a little computer without a monitor, a keyboard, and a mouse, it would be very inefficient and inconvenient if we always need to have all equipment when we want to use the Pi. Therefore, we decided to use WiFi that could help us to set up SSH and VNC and control the Pi through other devices such as phones, laptops, and tablets. The way we connected the internet router to the pi by WIFI was that firstly we edited the file called wpa_supplicant.conf to add the network to the Pi. Editing this file required permission, so we ran the command line with "sudo" in the front of the line. In that file, we added network={.....}, and inside the network, we specified ssid and psk. Ssid would be the network name that we want to connect to, and psk would be the password for that network. By using this method, we could add as many network as we want, moreover, we could also delete the network that we don't want to connect to anymore.

Conclusion

In conclusion, we achieved our goal in understanding how hardware is controlled and manipulated wirelessly. Few of our teammates didn't really know how to implement Raspberry Pi 3 model B before we start the project. In this project, each of our team had contributed and figured out with WiFi wireless connection, such as mobile or non-mobile device, Raspberry Pi 3 allows us to connect to the LED lamp and light it up. The final product works as what we expected. The most successful things we had done is we applied the wireless technology we had learned in class to our products and gained more hands-on experience in real world. Since we only had a few weeks to finish our project (as our equipment did not arrive until very late into the quarter), there are still many improvements that can be made in the future. First, we can make our product have a better appearance such as designing a cover for it because some long or short clips make people hard to handle it. In addition, we can add a speaker and add more libraries and string sequence in our codes to let the Raspberry Pi 3 model read the sequence and play a song. Also, since we now only has small coding projects and a simple webpage to manipulate the LEDs, we can try to develop a more fancy webpage or even an app to let more users to play on it. In this case, we need to have a stronger background in programming skills. Moreover, though we started our project with a goal of create a mood and feel at home, in the future, we can develop a bigger project based on what we had now and apply to our environment. For example, we can help to decorate our streets and other public area, so it no longer just targets at small groups of people. All in all, it is a great experience to work in a team and build the LED product using WiFi; the success of this project has helped us develop more interest in the vast array of wireless technologies.

How to Use

1. Turn on Raspberry Pi by connecting to a power source through micro-USB
2. The Pi should already be connected to WiFi through the school's network UCLA_WEB
    a. If not, connect the Pi to an HDMI display and use mouse and keyboard to control the Pi.
    b. Connect the Pi to a WiFi network nearby
3. SSH and VNC to the Pi to remotely control it
4. Ensure that the GPIO pins have a secure connection to the LED Lamp (Diagram below).
    a. 4 orange wires should be connected to the ground plane, one wire per plane
        i. Connect to GPIO pins: 2, 3, 4, 14
    b. Motor should be connected to 5V, Ground, and GPIO pin 21
    c. Other 16 wires should be connected to the anode pin columns
        i. Connect to GPIO pins: 5, 6, 7, 8, 9, 10, 11, 13, 15, 17, 18, 22, 23, 24, 27
5. Execute the code to turn on the motor and LED (directly)
    a. sudo python motor.py
        i. Turns on the motor to turn left, center, and right
        ii. Motor can only be turned on directly, (no HTTP implementation)
    b. sudo python LED.py
        i. Turns on the LEDS
    c. Turn off either with
        i. Sudo python code.py (set all gpio pins to high)
        ii. Sudo python code1.py (set all gpio pins to low)
6. Execute the code to turn on the LEDs (HTTP)

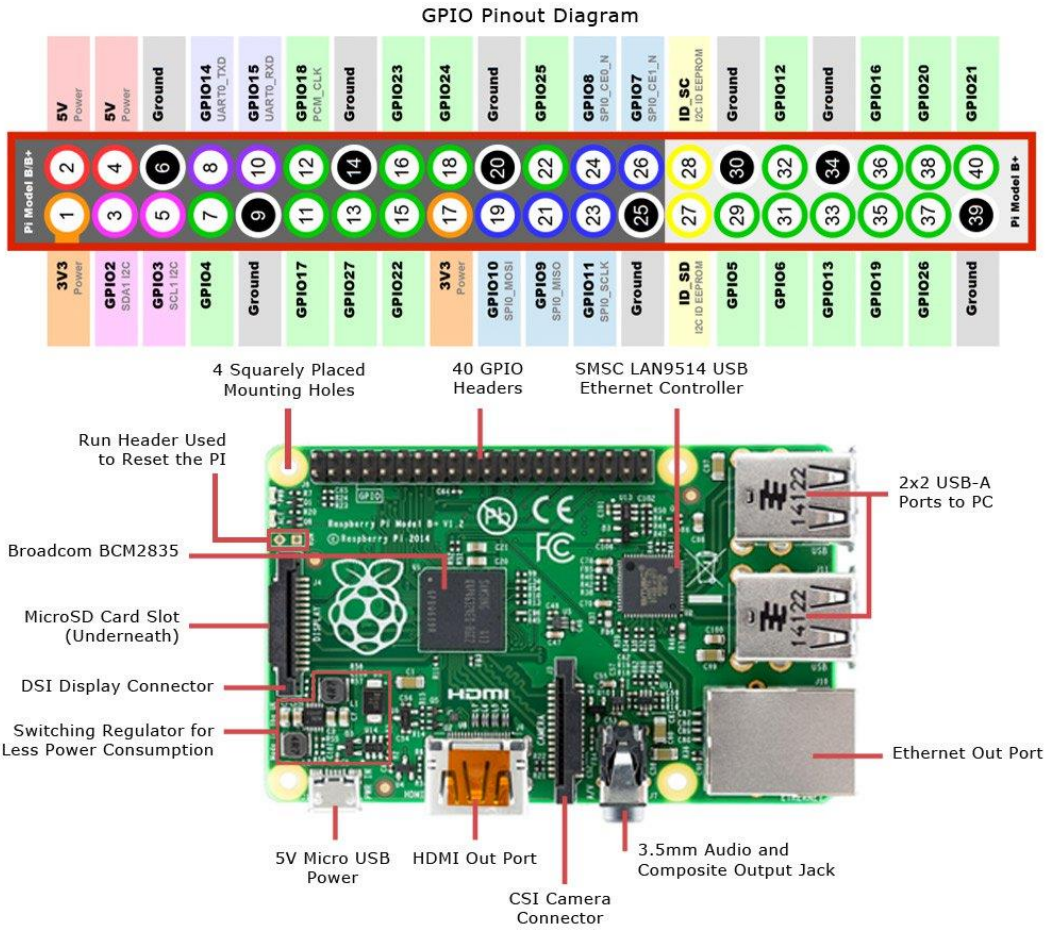
a. Index.html for the outlook of the webpage

b. Gpio.php for controlling LEDs

c. Enter the Raspberry Pi's static IP address into your browser

   i. Loads webpage with options to turn LEDs: off, on, and blink.

   ii. Click the corresponding buttons to do each action



Follow this diagram when connecting pins.