Team 5 Project Proposal: Electronic Directory Board for Purvine Lower Level

Prepared for: Prof. Healy

Oregon Institute of Technology

Prepared by Hayden Hutsell, Zak Rowland, James Rountree, and Beto Estrada

October 25, 2019

Table of Contents

List of Figures

Executive Summary

Directory boards are very useful for determining where places are in a building, but after someone familiarizes themselves with the building, directory boards do not serve any use to them anymore. This project will take a normal directory board and add real-time information of the rooms in the form of LEDs. An RGB will be a different color to show the room is booked, canceled, or open, as well as an LED displaying if the room is empty or not. Using an Arduino, LEDs, passive infrared sensors, and a simple web interface, we can achieve this goal. The project is intended to save time for students, faculty, and building staff.

Students would be able to glance at the updated directory board and determine what room is suitable for them. Faculty and building staff would save time by being able to display if class was canceled to all students using the building, instead of the students enrolled in the class that would receive an email. This would ease the amount of interruptions classrooms would face and help students find empty rooms without wandering around. Applying this concept to other applications, this project could be used for businesses needing a way to display if a conference room is being used.

Problem

The directory boards in Purvine are not used very often, aside from new students and visitors familiarizing themselves with the building. After the student is familiar with the building, the directory board is no longer useful. Along with their outdated design, the boards take up space on the wall instead of providing a resource to people every day, especially if they are already familiar with the building. Another issue students face is having to check what labs are open or closed at any given time. Maybe a group needs a quiet, empty lab for the time being. Currently, the group would have to hunt for an open and empty lab, spending valuable time and potentially distracting fellow students while searching for a suitable space.

When a class is canceled, sometimes the professor can't communicate to the students that they don't need to attend class. This leads to students waking up for an early lecture, driving to campus, and walking to their class to see the piece of paper posted to the door stating class is canceled. This wastes the student's time, building management's time by having to put up notices around the building, and causes extra traffic in the already chaotic campus parking lots.

A solution to these problems would help students with classes in Purvine Hall, faculty, building management, as well as reducing unnecessary traffic on campus. An updated directory board to feature CSET and EE talents would also be a great conversation piece for visitors touring the building. An electronic directory like this one could also be used by other schools or offices who share the same problem.

Solutions

The major problem to be solved by this project is identifying which rooms are open and if they are empty. The project will be updating the lower level directory board in Purvine Hall, which features 8 labs. Using RGB and colored LEDs, the electronic directory board (herein abbreviated as EDB) will quickly show students which labs are open, canceled, or booked as well as empty or occupied by sensing motion (see figure 1.) This means there will be 8 RGBs and 8 LEDs on the EDB. Worst case scenario, all RGBs will be on and all 8 LEDs will be on. 8 blue LEDs use a total of (8 * 30mA) 240mA, and if all 3 colors on all 8 RGBs are on they will require a total of (3 colors * 20 mA * 8) 480mA. Combining these values means all the lighting will draw about 720 mA. The microcontrollers we are considering using (currently the Arduino Uno WIFI Rev2 or the Arduino ESP-32) are rated for a maximum I/O current draw of about 1 amp. Since the microcontroller is the most expensive component of this project, lowering the power requirement on the Arduino to extend its life cycle would be ideal. The LEDs will be driven by an external power supply and the selected Arduino will send a small amount of current to a MOSFET or operational amplifier circuit, which will act as a switch to close the circuit and power the LEDs. This will lower the power requirement for our microcontroller.

A problem that arises from having the EDB use a dynamic schedule is having the changes to the schedule be displayed in real-time. This problem will be solved using the WIFI capabilities of the Arduino to communicate with a locally hosted website on a laptop nearby. Both the laptop and the Arduino would be connected to the same router and the Arduino would get the current time from the router. As a stretch goal, there would also be a public website that students could access from home to see the schedule for the day and check if a class was canceled without their knowledge. Full website design is out of scope for this project, but it would supplement the usefulness of the EDB.

Housing the EDB is another problem to solve. First, we need to bring the design up to modern standards. This includes contrasting colors (black on white vs. green on white), sharper lines, and most importantly, a space for the LEDs to live. The design can be made and printed at Staples. The housing is more complicated to plan without having all the pieces, but we will place it beside the board, so it is visible for all to see.

Overview of Solution

The microcontroller's purpose will be processing the external data from the sensors and website and adjusting the LEDs accordingly. Memory shouldn't be an issue to worry about with either Arduino because the most we will need to store into memory is the state of the LEDs (worst case scenario is all three are colors on and all motion detectors are on which would yield 3*8*8 = 192 bits) and storing the program, which would leave plenty of space to take in website data via WIFI.

The PIR sensors will reside in a cubby-like system representing rooms, and the sensitivity will be turned down relatively low. They typically run off 5 to 20-volt power and use very little current. In actual use, such as in a room, the sensitivity can be adjusted to fit the specific room. The max range is 7 meters. One thing to test is if the heat from the computers and lab equipment could trigger this type of sensor.

Powering the LEDs is another issue. It will be best to keep the Arduino sourcing as little current as possible to extend its longevity. Because of this, an external circuit and power supply will be powering the LEDs and the Arduino will simply trigger an operational amplifier or MOSFET to activate the circuit.

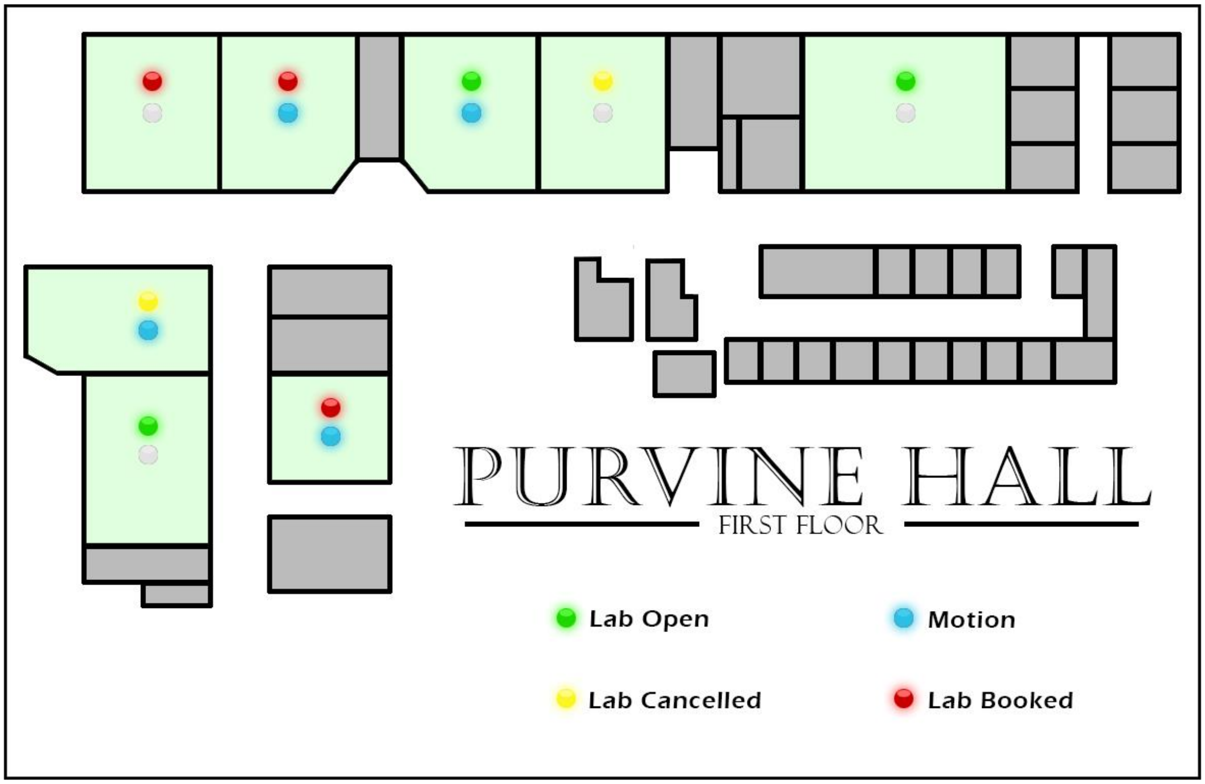
The wireless component will be a part of the Arduino. This helps solve the problem of finding an open lab because it can be changed dynamically when a lab is canceled or booked for an event. Being able to use a simple web interface to communicate with the Arduino would be quick and easy to update the schedule the Arduino is using. A problem to think about going ahead is the cost and difficulty of maintaining a public website. We need to talk to many faculty members at OIT to get approval to display anything with the University's name on it. Also, security will be an issue to research with both public and local website implementations.

Software Requirements

The EDB has four fundamental modules: the wireless communication with the localhost website, processing the data from the website, processing the data from the motion sensors, and switching the LEDs. The wireless communication aspect will consist of the Arduino and the localhost website on a laptop. Because we are using Arduino and C code, there are most likely numerous examples or references on how to use the WIFI chip present on the Arduino. The most difficult part will be creating the website with HTML, as our group has little experience in doing so. The module that processes the data from the website and sensors will need to determine which LEDs to light up and what color based on the information provided wirelessly. The module analyzing the sensor data is simple, simply turn the LED indicating if the room has activity in it or not. If there is movement detected, start a 10-minute timer to turn it off. Any additional movement detected will reset the timer. The Arduino will use C code to program these modules. Atmel IDE will most likely be used as we have experience using it, but the Arduino IDE is also an option. The biggest challenge will be the wireless communication aspect, because of our group's lack of experience in wireless communication. PSpice can be used to analyze and test our circuits before using them on live hardware. HTML will most likely be used to create the localhost website. Discord will be used to communicate with other group members, and Gitlab to keep our source code and other relevant files together and organized. For collaborating on documents, we have decided to use Microsoft Teams.

Figure 1: The EDB Design

Figure 1. The updated EDB design. Designed by Zak Rowland



Potential Customers and Markets

The main use case for our EDB is for buildings with rooms that are used at a different time around the day. Online systems exist for booking and scheduling rooms, but nothing showing the current status. Shawndell Brown, one of Purvine Hall's office specialists, said a system like ours would, "Save a lot of time for people like her, students, and faculty. Being able to update the current status of canceled or late classes would save me and others time from having to post many pink cancellation notices around on the doors." Ms. Brown seemed to like the idea, mainly quoting the EDB's time-saving features for all parties, not just students. An additional feature she mentioned was being able to set a late indicator, like orange or purple. This would be very easy to implement, and we have plans to update our design to include this. She also had the idea of making it so faculty could edit the timetables, so they could cancel the class wherever they might be. If this prototype works as planned, adding additional RGBs for the classrooms is worth researching so all the rooms can be mapped on the EDB.

Team Composition

One of our team’s biggest strengths is familiarity with the C language as well as the Arduino. Zak has some experience with graphic design and some app development. Having completed other personal microcontroller projects over the summer, Zak also has additional experience with microcontrollers outside of a classroom or lab setting. Because of this, Zak is familiar with searching for external resources and content to answer the questions he has. James has connections to people with website building experience, which will be crucial in designing a functional website for our Arduino to communicate with. Having taken many C based classes, coding in this language is his strong suit. This could also help achieve our stretch goal of designing a public website to visit from home, so they know if their class is canceled before coming to class. Beto, Hayden, and Zak are confident in constructing and analyzing circuits, which will be crucial in designing an effective design for our LEDs. Hayden also has experience with report writing for the many documents that will need to be written to showcase our plans and progress. Later in the project when we need to design housing for the EDB, Hayden has experience with power tools which will be useful if we need to drill, cut, or fabricate housing for the microcontroller and the EDB, and Zak has experience with 3D modeling for designing prototypes.

Parts List

The Microcontroller

This will include either an Arduino Uno WIFI rev2 or the ESP-32. The Uno WIFI will be purchased from Arduino themselves, and the ESP-32 would come from Amazon. The ESP is cheaper, but it isn't made by Arduino, so the quality could be compromised. That's why we are leaning towards the Uno WIFI.

The LEDs

The RGB will be a simple 5mm RGB LED, and the motion detection LED will be a 5mm blue led. There are also the option for RGB strip lighting, which is controlled by PWM. Since it uses a single I/O pin, we would save a ton of I/O pins, but more research needs to be done on this technology. Either option would require an external power source. There are so many different types from many different retailers, research needs to be done on which would be the best option to purchase from.

The Circuit Components

This includes the MOSFETS or Operational amplifier, the breadboard to put it on, resistors, wires, and housing. These things are also cheap, but we need to research if we aren't buying from the source. Buying the 358N dual op-amp directly from Texas Instruments would probably be the best bet. We will also need a shift register to expand our Arduino I/O if we are not using the RGB strip. Shift registers (74HC595) sold from Texas Instruments are about 60 cents, so buying from the source will be best. Housing can be determined later once the size of all the components put together is realized. Also, to be considered is the external power supply. Once we have all the components, we can start looking at external DC power supplies and the hundreds of places they are sold.

The Motion Sensors

We will be using a wired PIR motion sensor, which could require some minor soldering depending on the model. We found a pack of 10 from Amazon for about 20 dollars, so it's a safe guess there will be cheaper options elsewhere.

The Router

There are many routers, but we don't need the best of the best. Just a simple wireless router will do. Based on the routers we have experience using, we will most likely use a Netgear brand router based on the features their products have.

Preliminary Cost

The Arduino Uno WIFI will cost 50 dollars, while the ESP-32 is around 15 dollars from many different retailers, and there is also two included. Resistors, op-amps/MOSFETS, wires, can be acquired at very cheap prices. I would not expect this section to exceed 35 dollars, depending on how many op-amps and MOSFETs we need to implement our circuits. A DC power supply costs about 10 to 20 dollars depending on the brand. Breadboards are about 10 dollars four a pack of 4. High-quality blue LEDs are about 50 cents each, and RGB LEDs cost about the same. So, we can expect to spend about 10 dollars on LEDs if we include shipping costs. A simple Netgear wireless router costs about 20-40 dollars depending on what model is in stock at the time. Public website hosting does have a cost, but since that is a stretch goal, we will not be calculating that cost at this time. Indoor banner printing from Staples starts at 16 dollars, but we cannot get a definite price until we have a final design. Housing designs can't reliably be computed at this time, but we can set aside 20 dollars for the materials. With all these things considered, if we go with the Arduino Uno WIFI, we can expect to spend $167 on materials. This cost will most likely go down as we find cheaper sources for the same parts, free parts we already have, or surplus free parts from the CSET or EE department. If we go with the ESP-32, we can expect to spend about 132 dollars. For a table, see figure 2 on the following page.

Figure 2: Parts Table

Figure 2. Our current table of parts to purchase.

Module	Part	Qty.	Cost Each	Total Cost	Alternative
Router	Netgear AC1200	1	$38.99	$38.99	Any simple wireless router
Microcontroller	Arduino UNO WiFi	1	$44.90	$44.90	Raspberry Pi
	Arduino UNO	1	$22.00	$0.00	Raspberry Pi
	--- WiFi Module	1	$6.95	$6.95	Adafruit
Directory Board	RGB LED	25	$5.95 - 10 pack	$5.95	Any other bulk pack
	Resistors	1-3	$5.99 – 775 pack	$5.99 - $17.97	Any other variety packs
	MOSFET	25	$0.90	$22.50	ON Semiconductor (If op-amps don't work out.)
	Shift Register	3	$2.75 – 3 pack	$2.75	Any other 74HC595
	Wire	N/A	$9.48	$9.48	Any other generic wire
	Power Supply	1	$13.90	$13.90	Any other 5V/2A+
	358N op-amps	20	$0.68	$13.60	Other resellers
Motion Sensor	Passive Infrared Sensor	2	$9.49 – 5 pack	$18.98	Any other HC-SR501
	Power Supply	1	$13.90	$13.90	Any other 5V+/1A+
			Total:	~$180.42	Some parts won't be purchased if others are purchased.