Josh Howard

Dr. Majumder

CSCI 509

April 24, 2020

Bluetooth Wall Outlet

The purpose of this project was to build a Bluetooth wall outlet. The outlet will allow one to cut on and off the power to the outlet with another Bluetooth device. Cutting the power on and off will be used to turn small appliances on and off as well. To accomplish this, an Arduino Nano IoT was used. Taking this board, which had a Bluetooth module built in, it was wired to a relay switch and a battery. The Arduino was then programmed to look for certain input from a connected device. That input would be read to decide rather or not to switch the relay to open or closed. The live wire between the input and output of the 2-prong outlet was put into the switch portion of the relay. Doing this allowed for the live current to be completely cut when the relay was closed, and completely open for use with the relay open.

This project was to create something that would let me remotely control an outlet in my apartment. As I go between rooms in my apartment, and will usually get busy in another, I often times forget to turn things off. Either this be a light, a game console or other small appliances. Making something like this would allow me to quickly check and see if I left something on and if so, cut it off from the other room. The time I would spend going to the other rooms and checking things could be saved and put into what I’m currently working on. With that in mind I looked into ways do accomplish this. Since I always have my cellphone on me, I decided the best option would be to integrate that as being the main hub that I can control the outlet’s from. The easiest short-range option I found was using Bluetooth.

First, in implementing this I needed some form to hold the components. Something that wasn’t bigger than a usual outlet found in most home to safe on space. I found a power brick to an old phone I had lying around. The shell of it was the perfect size to fit everything, and it also already had the prongs that plug into the wall. Next was the Arduino. Arduino’s smallest board is the Arduino Nano, so that was the one that was needed. I happened to find that Arduino makes a nano with built in Wi-Fi and Bluetooth modules called the Arduino Nano 33 IoT. That specific board is the one I ended up buying and using for this project. The relay I used was a “Tolako 5v Relay Module for Arduino” that I found on amazon. This one was picked as it used 5v to power the actual relay and could run up to 250VAC through it’s switch circuit. So this relay was able to withstand the full power coming out of the wall without burning up. The Arduino connected to the relay via digital pin 8 on the board. This pin is sends out a “HIGH” or “LOW” signal to the switch pin on the relay, which turns it off and on. Then I needed to power both the Arduino and the relay while it was in use. A simple 9v battery was used in this function. The 9v battery was wired so that it powered both other components at the same time. The last portion of the hardware was the input/output wall pins on the box. Since the box already had the pins to go into an outlet, I had to worry about the two that other things would plug into. For this, I measured and cut two slits into the front side so the prongs could go into. For the connections inside the box I used 4 small pieces of copper, 2 for each prong. Each set were soldered on the ends, with one end being soldered to a piece of wire. These were placed in position and hot glued into place. The wire from the input and output pins were then connected to the relay. This completed the physical part of the system.

For the software side of my project, I used two different programs. The main one used here was the Arduino IDE. This was used for all of the programing for the Arduino’s control functions. Programming the Arduino wasn’t terrible, but it did have it’s hardships. Luckily though the Bluetooth module that came installed on the Arduino had its own library. So I didn’t have to worry about coding a separate module to talk with the board. To start with I just imported the BLE.h library into my Arduino project. After that I set up the service that will be broadcast over Bluetooth named “PowerSwitch”. A “Characteristic” was then setup. This characteristic is what allows the connected devices to interact with the Arduino by sending and receiving data to and from it. Once the Arduino is powered on it will start to broadcast its signal, and will loop until its broadcast does start. It will then add the “PowerSwitch” name and the characteristic to the broadcast. Then wait for connection. Once a device is connected to it, it will loop in a cycle. Within this cycle the Arduino checked the characteristic value every loop. If this cycle equals a set value it will turn on the relay. If the value equals another set value, it will turn the relay off. If the Arduino ever looses connection it will keep the relay in the same state and continue to broadcast until another device is connected. This is so that if the relay is on, it will continue to stay on even if I do disconnect and leave. Only to be turned back off when I mean for it to be.

The second program I used was an android app called “nRF Connect”. This program allows me to look for, and connect to, Bluetooth Low Energy devices, or BLE. This program I did not have to do any programming for. It is a free app on the google play store. When using this app it will tell me all of the nearby Bluetooth devices that it picks up. My Outlet will read “PowerSwitch” within this app. Connecting to it will show the characteristic that I added. It give the user a button to read the characteristics current value. Then sending a “0” to the device will cut it off, or sending a “1” will cut the device on.

After many hours of working on this project, I would say the results are better than I’d hoped. I had never done any programing with connections over Bluetooth. Although this gave me the most trouble out of all, it worked just like I had hoped. I have tested it on a few different things, mainly lamps, cellphone chargers and my living room lights, and each one worked great. I was able to see that I’d left some lights on before going to bed and turned them off with my phone without going to said room. The distance on connection wasn’t amazing, but good enough for my purpose. With testing, the range was about 15 meters. I was able to hold connection from almost anywhere within my apartment, but outside it the connection dropped as I had gotten too far away. The number of devices that it could control at once I wasn’t able to test. Seeing as it is only a two prong plug, I had no two prong power strips. So at the current moment only one device can be connected to it at any given moment. One last thing for this one, isn’t so much a bad thing on the devices part than it is mine. This was the first project I had actually soldered components together. So occasionally I would have to open the case and fix a wire that had come loose. It’s possible it could not work, but the issues I had with it were the bad solder points on my wires.

In conclusion, I would say the project was a success. It did give me a fair bit of trouble along the way, some spending a few days trying to figure out and fix. It turned out to do exactly what I needed and wanted it to do. There are a few other things I would like to add or change if I were to add to this project later on. One big one would be using the built in Wi-Fi on my Arduino. Using that would allow me to control my relay over the internet, giving me a world range as long as I had connectivity where I went. Then I could check in if I were at work, on a vacation or somewhere else. The next one I would like to implement is a three prong plug. Right now, the two prong is limiting on what I can plug into my device. Adding the third ground plug would open up more options and higher-powered appliances like my computer system. Other than those few things, the end project, except for the bad soldering, was pretty much how I envisioned the project from the start. Also, The experience and knowledge I gained from this was vastly more than I realized I would have needed. So know going forward I have that to put toward other projects and programs.

References

“ArduinoBLE.” Arduino, Arduino, 25 Dec. 2019, [www.arduino.cc/en/Reference/ArduinoBLE](http://www.arduino.cc/en/Reference/ArduinoBLE).

“Bluetooth Low Energy Overview : Android Developers.” Android Developers, 27 Dec. 2019, developer.android.com/guide/topics/connectivity/bluetooth-le.

Houston, Monica. “Getting Started with Bluetooth Low Energy (BLE).” Arduino Project Hub, 8 Mar. 2017, create.arduino.cc/projecthub/monica/getting-started-with-bluetooth-low-energy-ble-ab4c94.

“NRF Connect for Mobile.” Nordic Semiconductor - Home, www.nordicsemi.com/Software-and-tools/Development-Tools/nRF-Connect-for-mobile.