Craig Siron

ITMT492

Professor Hajek

Final Project

Remote-Halo-Temperature-System

For our project, we decided early on that we wanted to do something with color-changing LEDs. There were a few options to us, mostly pertaining to the hue lighting system. We took some time to discuss it and we came up with the current project we have, the Remote-Halo-Temperature-System. It sounds like a simple concept but overall it was a fun project to work on and provided a few different challenges along the way.

The basic concept of our project was to create a system that would gather the temperature and transmit the data to our arduino to and then use that information to turn on a set of LEDs that would change color depending on the temperature received. This left us with a few options to consider. What would we use to transmit the temperature data? How would the device that collects temperature operate outside? How are we going to set up the LEDs? After we did some research we decided on how we were going to solve these problems. To transmit the data, we would use the Xbee radios that we used previously in class. They have the capability to, with the help of a breakout board and temperature sensor, collect the data we need and then transmit it to another Xbee radio setup inside. That Xbee can then be read off of to get the relevant information. They also have a communication range large enough to have one operating outside while another radio remains inside to stay connected to the arduino and power. We also found out that, the Xbee with temperature sensor can be connected to a rechargeable battery to give it power and with the help of a capacitor that battery can then be connected to a solar panel. The Xbee can also be programmed to sleep in specific intervals so it could enter sleep mode and only report once every five minutes which would save a lot of battery. These things combined would be able to power it, in theory, until the battery itself goes out or a component breaks. Along with that, it can be placed in a weatherproof case to help protect it against water and cold as well as animals that might tamper with it. For the LEDs we had a couple different options. We could hook up an array of different colored LEDs that would light up different ones depending on the data received, but it would be cluttered and look simple. We could use a LED bar but we would have had to order one and figure out how to set it up. There was also the potential for the Hue lights to be used, but with another group using them it would have been difficult to test and we wanted to stand out compared to them. In the end, we settled on the Adafruit NeoPixel ring LED which is a circular array of LEDs that can be programmed with the use of Adafruit's library to whatever specifications the user wishes and it is easy to connect with just a power connection, ground connection and a data in connection.

With the design issues out of the way, our next task was to build and configure the project. We started with the Xbee configuration. We had to change the settings on them so that they would communicate properly. This caused us many problems ranging from wrong values to bad radio to bad shields. Eventually, we managed to get the two radios we were using to communicate. From there, we had to connect the Xbee radios to their individual systems. The outside radio with the breakout board and temperature sensor needed some solder work as well as the battery and solar panel components. The other Xbee radio needed to be connected to a shield and hooked up to the arduino. Compared to the radios, this part of the project was fairly simple. The next thing we had to do was figure out the coding. The main things we needed to do coding wise were learn how to properly gather the right data from the information the Xbee radio was sending, how to read that data, how to convert that data into temperature and then finally implement it with the rest of the code. With a little bit of research we were able to read the data from the correct parts of the data packets that the main Xbee was receiving and then from there we were able to run a couple simple conversions to get the proper temperature out of the data received. After that, we had to download the libraries and do some reading and research on how the NeoPixel ring operated and what classes they have included in their libraries. Once we had these individual parts figured out we combined them into one and moved on to testing making sure that we were getting the right temperatures and the lights were turning on accordingly. While there was a sufficient amount of research into the other parts of the project, the Xbee radios gave us the hardest time between all the components.

Our project works by connecting one Xbee radio to a temperature sensor by soldering them both to a breakout board. This allows the radio to read the data from the sensor. The radio then transmits this data in analog form to the other radio connected to it that is stationed inside. That radio is connected to an arduino uno gathering power through the arduino board and letting the arduino read the analog information that the radio has received. The arduino then takes this information and processes it to find what part of the data it needs to read to find the temperature, which is given in voltage. Then it converts the voltage amount into Celsius using a conversion provided from the temperature sensor manufacturer. From there, it converts it to Fahrenheit for more understandable readings for us. Then, using if statements, it turns the NeoPixel ring a specific color depending on the Fahrenheit temperature. If, for example, the Fahrenheit temperature reading shows as 37 degrees, then the NeoPixel will be lit up purple. If the temperature reading shows as 93, then the NeoPixel will be red.

Besides the project working, our main goals throughout were to demonstrate the working concepts of electronics, data collection, data transmission and data presentation, which we have done through the components of our project. We solved the electronics aspect in two ways. The primary system with the Arduino can be connected inside to a normal power unit or to a computer to both run the code and supply power to the Arduino which in turn also supplies power to the Xbee radio connected to it. We satisfy the data collection requirement through the use of the temperature sensor as it outputs voltage proportional to the temperature which can be recorded and is reliable. The Xbee radios accomplish the data transmission portion since they send the data from one to the other at a range of about 300 to 400 feet, though higher end radios can transmit up to 40 miles. Finally, we accomplish the data presentation in two ways. The first is serial output from the computer connected to the Arduino board. It presents the voltage, Fahrenheit and Celsius data, as well as the packet data, for the user to see. The other method of data presentation is through the NeoPixel ring. It may have a small curve to learn what color means what temperature, but once you have that figured out the ring presents the temperature to you in the form of a color-coded LED so you know generally what to expect when you go outside just by looking at the color.

While our project does work as advertised, that doesn't mean it can't be improved upon. There are many different ideas for ways to modify or upgrade the existing project in the future. The first idea we had to make it better would be to incorporate a mesh of Xbee radios. This would allow you to place a couple different radios with temperature sensors around your property and there would be a light corresponding to each one and then an aggregate light. This could be useful, for example, by placing one in a sunny area around your property and then the other in a shady area. It would give the user a general idea of the temperature outside because it is unlikely that you would be in direct sunlight or direct shade, but then lets the user know how it is in those environments. Another idea would be to implement more of the code available for the NeoPixel ring. Since it is fully programmable, it could be made to tick up to the proper color like a thermostat showing all the different colors before the current temperature and ending on the current temperature color. Aside from that, it could potentially be incorporated into some other projects. You could combine it with the doorbell project, for example, and have the doorbell lights be the color corresponding to the temperature received, or it could be combined with the color reading project. Since the NeoPixel uses RGB values to set the colors, you could add extra code to set the colors to your own specifications by holding up the color you want to use.

Overall, this project was a good way for us to practice and implement the major concepts presented to us throughout the course of the semester. It was also a fun way to learn and apply these concepts and practices to real-world ideas and solutions to problems.