Cesar Escoto & John Moore

Jeremy Hajek  
ITMT 492 – Embedded Systems

Chromiris

Have you ever wondered what the color of a certain object was? Well with our project, you can do just that. Meet Chromiris. Chromiris is a relatively simple device that will allow the user to get the HEX and RBG value for any object placed near its sensor. It is easy to use; by placing the sensor just 1-2 inches away from the object, you can determine any color. The whole experience of creating Chromiris was very eye opening. In this paper, I will document the idea behind the product and the build process that contributed to Chromiris. I will start by discussing our team’s brainstorming process, and how we decided on the idea. Working with a team was a good experience for this project, because we were able to collaborate and come up with creative ideas. Then I will discuss some problems we faced. As with any project, we had to solve some issues that arose. At first we had to eliminate ideas that would require special hardware, and then once we had out idea, it took some effort to find the right software that would do exactly what we needed. Luckily our team was able to locate some online resources that aided us in understanding the software. Lastly, we compare our design to an existing product. We make a comparison based on functionality and cost. Chromiris proves to be a competitive design, with an even more competitive price point.

Originally it was difficult to come up with the idea for Chromiris. Our team spent a lot of time brainstorming what to design and build. We shared common interests art and music, so our ideas came from this area. One initial idea we had was, an automated guitar tuner that would hear the note of a particular string and tune accordingly to the scale you wanted. We figured out that we would need special hardware, such as tuning pegs, for the guitar to make this idea a reality. Due to our time constraints and available resources, we decided to keep brainstorming. Both of us enjoy visual arts and design and thought a colorimeter would be a cool gadget to make. The initial idea was to place the colorimeter on any object and it would instantly tell you the HEX and RGB values of said object. It was cool to to us that you could place the colorimeter on any object and get a value for that color. We tossed the idea around and it seemed like something that was achievable with our available resources. With this idea in our minds, we began the process of designing the colorimeter. Originally, we wanted to have data transmission through an Xbee to a computer to make the colorimeter work at a distance; using the Adafruit Flora made it difficult for us to figure out how to implement that feature, so we decided to scrap that feature. The Adafruit Flora was not as versatile as we had hoped. In addition, the Adafruit color sensor we were using was only capable of accepting 3V. The color readings we were getting with the 3V color sensor were murky and everything was registering as a darker shade than the actual color. The Flora and color sensor were not getting the job done for us, so we decided to make a change.

To help solve our issue we turned to Arduino. We thought that having something powered by the Arduino would be easier to code and wire, especially with the use of a breadboard. With this idea, we started again, this time using the Arduino. With the Arduino, we decided to upgrade to the breakout version of the Adafruit color sensor that had a pinout. This was easier to attach to a breadboard, but more importantly, it allowed for 5V, which subsequently made the colors more accurate. With the new enhancements we began the majority of our coding. The bulk of the code was used from YouTube video we found. The video gave us a link to a German blog for Arduino projects and 3D printing (Dax, A.). We needed to include the Adafruit\_TCS34725 library so the color sensor would be recognized. The sample code gave us a solid base but we still needed to find out a way to display the RGB and HEX values. We had to add code to allow the display on the LCD. We wrote code to label and display each RBG value on the LCD. We also had to map the values to 255 because at first the values given to us were way too high. Once they were brought down, it showed the correct values. One problem we faced was the LCD was holding onto the third digit if the object caused the corresponding RBG go below 100. To overcome this challenge, we had to modify our code to set the hundreds place digit to a space if the RBG value went below 100. After that, we began to tackle the problem of displaying the HEX value for each object. Luckily, the Arduino print library has a convenient HEX second parameter. Using that we were able to print out each RGB HEX value under each label. The two worked separately but we then wanted to add a button so we were able to switch between modes. Usually, any button presses are registered only for as long as the button is being held down, and once the button is released it goes back to its original state. We had a hard time trying to code a button to hold the state it was in. We used the sample debounce code in the Arduino library to help us get started. We decide to make each of the RBG and HEX print outs separate functions. Doing that we were able to say if the button is in the ‘0’ state to run HEX function and when in the state of ‘1’ to run the RBG function. That worked quite well, but then we faced the same problem on the LCD. If one of the RBG values was over 100 and we switched to HEX mode, the hundred-place number would remain on the screen. Simply printing a space in the hundred place for each value cleared that up.

Physically Chromiris is not very complex. Chromiris is a colorimeter that is powered by an Arduino Uno and using an Adafruit color sensor. It entire apparatus consists of four six parts, including wires to hook everything together. All of the parts we needed were included in the Arduino starter kit, except the color sensor. The LCD is wired normally with 6 analog pins giving it data to process, 3 ground wires, and 2 power wires with one having a 220k ohm resistor to power the backlight of the LCD display. The button was simple to wire up. There is power running into one side and then a wire runs to a digital input and a 1k ohm resistor out to the ground. Finally, the Adafruit color sensor only required 4 pins, power, ground, and the last two SCL and SDA, for data transmission. We found while testing our project, that glossy items are not represented as well as matte objects or clothing. Chromiris uses its sensor to collect data for each color. Then the information that is gathered is formatted and mapped to corresponding numbers in the code. Once the numbers are understandable to us they are displayed on the LCD.

The main colorimeters in the market are significantly more sophisticated. Most of them are designed to test the colors in liquids, pH levels, and even any minerals that might be in said water. We could not find a colorimeter that is as primitive as ours. The premium colorimeter we found was the YSI pHotoFlex STD. It is waterproof and housed in a case for maximum durability. “The compact pHotoFlex STD handheld precision meter enables you to carry out the following measurements quickly and reliably: Concentration measurements (colorimetric measurements), absorbance measurements, and transmission measurements (pHotoFlex).” It has the built in capability to transfer its data to a remote computer. It has very large data storage with 100 data sets possible. With all of these features it is easy to see why it costs astronomically more than Chromiris. The YSI pHotoFlex is around $1000.00, comparing to our design, which came in at about $50.00 total cost. With that being said, Chromiris does compete with the YSI pHotoFlex, however, it is overall a much more basic product.

As a team, we learned a lot though the process of designing and building Chromiris. From working with others to actually fabricating a gadget from nothing, it was a good learning experience. It was also fun to actually make something from scratch. Working on a team for this project was beneficial, because it required a lot of brainstorming and creative thinking. We even though of a couple of ideas to enhance our project in the future. We would like to add a user interface in the form of a touch screen to allow for easier use. In addition, we would like to add a database functionality. If a color you just scanned corresponds with a HEX value in the database, it can give you the name of that color. For example, we thought that teaming up with a paint manufacturer we be a good use for the product. People would be able to scan a wall in their house, and it would be sent straight to the manufacturer and a paint that matches the exact color could be produced. There could be a list of all the rooms in your house with the colors they use stored into Chromiris. Hopefully we will be able to make enchantments, and addition features such as this, a reality in the future.

In conclusion, the Chromiris is a colorimeter that can display the HEX and RGB values of any object you place in front of it. It is not a perfect design, but we were able to learn a lot by making it. We also came up with many ideas on how to improve our design and ways to innovate it. By comparing our design to an existing similar product, we were able to see some areas where our design can grow. Working with a team greatly helped our communication and collaboration skills. We did not feel like presenting was our team’s strong point, but it gave us good experience for future presentations. The presentations, also allowed us to share our ideas with our class, and learn from the other groups. We faced some problems while designing our product, and even had to scrap some of our initial ideas, before coming up with a design that was possible to make within our given time and resource constraints. Chromiris is a cost effect design with endless applications for use. In the future we hope we can enhance our project to increase its functionality.

References

pHotoFlex (2016, April 23). Retrieved from https://www.ysi.com/photoflex

Dax, A. (2015, January 12). *Recognize colors with RGB sensor TCS34725 and the Arduino*. Retrieved from http://www.makerblog.at/2015/01/farben-erkennen-mit-dem-rgb-sensor-tcs34725-und-dem-arduino/