# Team UV - Final Report

Simeon Babatunde sbabatu@g.clemson.edu
Benjamin Coomes bcoomes@g.clemson.edu
Kristopher Kohm kckohm@g.clemson.edu
Miriam Lozneanu mloznea@g.clemson.edu

## Project Summary

Our system is a set of wearable devices that inform the user about UV exposure. One device senses UV radiation and the other displays UV radiation levels. The display device shows the current and average UV index. The display is in the form of two LED bars that 'fill up' as exposure increases, and 'empty' as exposure decreases. The sensor device can be clipped onto a hat or backpack. The display device is worn on the user's wrist. Our system lets people know when they have experienced a healthy amount of radiation and when their exposure is approaching dangerous levels.

## Project Description and Implementation

### Changes

- Battery - Originally the batteries for each component were going to be a series of coin cell batteries. This was changed to the Lithium Ion batteries used for the final project for three reasons, the first of which was rechargeability. Once the coin cell batteries ran out of power they would need to be replaced, but the Lithium Ion batteries could simply be recharged. The second reason was that we wanted an extra 0.1 Volts from our battery which the Lithium Ion batteries offered. The final reason was the capacity of the batteries, because the coin cell batteries would not have lasted nearly as long as the Lithium Ion batteries. The coin cell batteries offered 110mAh at 3.6 Volts, whereas the Lithium Ion batteries offered 1000 mAh at 3.7 Volts.
- Sensor Device Displays - Originally one of the sensor bars was supposed to display the instantaneous changes in UV levels and the other was supposed to display the total amount of radiation received since the device had been on. The instantaneous bar was kept the same, but the total bar was changed to an average amount of radiation received since the sensor device had been on. This was both more useful and computationally efficient because we did not have to worry about overflow from the total.

**Software:**

- Sensor Board - The sensor board runs a program that loops between three states: get UV index, compute average UV index, and transmit data. The UV index is calculated based on values read from the UV sensor. We found the algorithm to calculate the UV index as a part of the Sparkfun software library for the ZOPT2201 UV sensor. After getting the current UV index, it is used to calculate the average UV index from the time the sensing device has been on. Once the instantaneous and average UV index have been obtained, they are sent to the sensor board. Each message sent is only two bytes. The first byte contains the current UVI and the second contains the average UVI. Transmission is done using Josiah Hester's CC1101 library.
- Display Board - The display board checks to see if any radio transmissions have been received every 100 ms. If a message has been received, the current UVI is retrieved from the first byte and the average UVI is retrieved from the second byte. The program then sets the LEDs on each LED bar according to the values received. We wrote an LedBar class to simplify controlling each LED bar.

**Hardware**

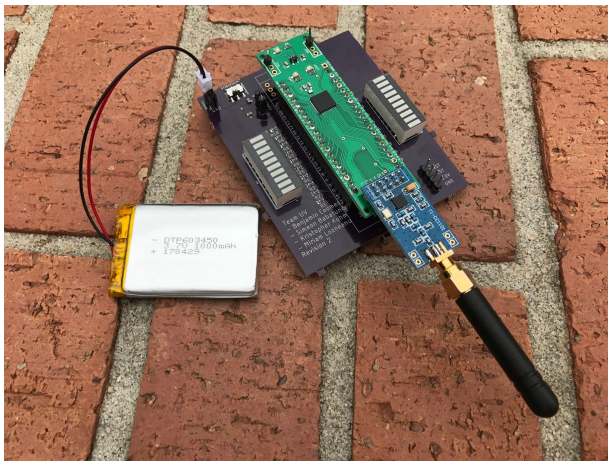
The following is the list of electronic components used in the implementation of our system:

- ZOPT2201 UV Sensor: this component was used in the sensing unit for sensing 310nm ultraviolet radiation (UVB). It gives accurate UV index measurement without interpolation from ambient light like most other sensors. It utilizes simple $I^2C$ (Inter-Integrated Circuit) interface for communicating with the MSP430fr5969 microcontroller, and rated for 3.3V and 110μA. It uses dual Qwiic connectors for seamless connection.
- MSP430FR5969 Microcontroller**:** This is the major component used in both the sensing and display unit of the system. It is an ultra-low-power (ULP) FRAM platform, utilizing 16-bit RISC Architecture for up to 16-MHz clock. It is responsible for controlling and interfacing with other peripherals within the system. Powered by (1.8 V to 3.6 V) and consumes approximately 100μA/MHz. The first Microcontroller is responsible for fetching the sensed data from the UV sensor, compute the average UVB index and then transmit the data to the display unit via the CC1101 radio transceiver. The second microcontroller is responsible for retrieving transmitted data from the radio transceiver and then drives the display section based on the signals received.
- CC1101 Radio Transceiver**:** It is a low-cost sub-1GHz transceiver designed for very low-power wireless applications. It was used for initiating remote data transfer between the sensing and display module. It interfaces with both

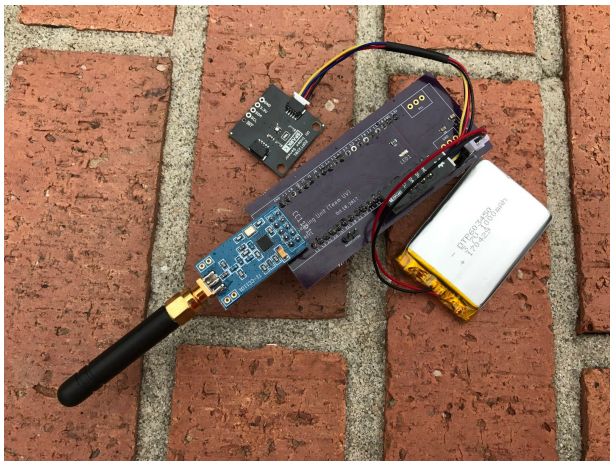
microcontrollers and operates in the frequency range 387 – 464 MHz with instantaneous maximum working current less than 30mA. It interfaces with the MSP430 microcontroller using Serial Peripheral Interface Bus (SPI).

- YSLB-10251B5-10 LED bar: This is a simple strip of 10 blue LEDs. Each LED is controlled directly by the microcontroller via the IO pins. Each LED operated between 3.2 and 3.6 V, with a maximum forward current of 20mA. We used two of these on our display board to indicate the current and average UV index. We used 330 Ohm resistors to drop the current to 10mA on each LED.
- Li-Ion Battery PRt-13813: These rechargeable Lithium Ion batteries offered 1000 mAh at 3.7 Volts. These were connected to the Vin pin of the Voltage Regulator which converts it into 3.3V. One was supplied for each unit and they could easily be attached and removed via headers on each board.
- NCP583 LDO Voltage Regulator: This was used on both units to regulate the 3.7V from the battery into 3.3V used to power the components.
- Resistor: We utilized a 330 ohms resistor for limiting the LED current to 10mA. Thus, minimizing the energy consumption of the system.
- Capacitor: We used both 1µF and 0.1µF as filtering capacitors at the input and output of the voltage regulator on both units.
- Surface Mount DPDT Switch: the switch was used as the main controller for switching the system ON/OFF. It is capable of switching up to 360mA at 3.3v.

**Board Pictures:**



Display Board



Sensor Board

**Challenges:**

- Radios: The biggest challenge we faced during demonstrations was reliable radio communication. We did not perform any checks on received data, so interference and corrupted packets could cause our display device to malfunction. We could

have solved this problem by establishing a session between the display and sensor boards before either begins sending or receiving data. A session key would be chosen and appended to the first byte of every message. Messages without this key would be ignored. We also should have checked the values of the UV indexes, and ignored any invalid values.

- Pin configurations: We wanted to use the green MSP430 boards since they already has an MSP430 chip soldered to them, as well as a place for the CC1101 radio. However, we discovered that pin 3.6 was wired to the radio on the green board. We needed this pin for I2C communication with the UV sensor. We managed to move the pin by modifying the pins.h file in Josiah Hester's CC1101 library. We also had to change Energia.h under energia-1.6.10E18\hardware\energia\msp430\cores\msp430 to add pin definitions to energia, since the green board exposes some pins that Energia does not define.

## Resources

- Josiah Hester's CC1101 library: This library provides a simple interface to the CC1101 radios that we used in our project. It can be found at the following buffet repository: https://buffet.cs.clemson.edu/vcs/u/jhester/cc1101-energia/. We modified the pin configuration slightly for the sensor board, since the library uses pin 3.6 by default, which we needed for the ZOPT2201 UV sensor. We also referred to the TX and RX examples included in this library when writing code for our devices.
- ZOPT2201 library: This library was developed by Nathan Seidle at Sparkfun. It contain the firmware and sample code for interfacing the ZOPT2201 UV sensor with the microcontroller via the I2C protocol. We customized the library by creating a .cpp version for energia which also contains the logic for computing the UVB average. The library can be found here: https://github.com/sparkfunX/Qwiic_UV_Sensor-ZOPT220x/tree/master/Firmware

## Class Feedback

**What was the most challenging part of this class?**

Ben: The most mentally challenging part was learning how to select parts that would work together and designing a circuit to connect them. The most challenging part overall was putting the accelerometer on the sensor board.

Simeon: The class on state machines and low power modes took me some time to comprehend.
Kristopher: Constructing a properly functioning sensor board (for the class, not the project) was fairly difficult. In addition, working with hardware for the first time is daunting coming from a Computer Science background.

Miriam: I used light traffic examples, but coding was hard for me. I tried to figure out how to get files on Energia working, but that was difficult too.

**What would you like to see added to this class in future semesters?**
Ben: I would like to see more pizza on demo day. Other than that, I can't think of anything else to add.

Simeon: I would appreciate if a deadline is set earlier on in the semester for the assembling of the sensor board before taking the class on Accelerometry and signal processing. Also, having accelerometer with a bigger footprint would be fun to assemble.

Kristopher: I would like to do more hands on, in class demonstrations and activities. Perhaps more of them could rely on use of the red boards instead of the sensor boards in case people had trouble with them.

Miriam: I am interested in solar energy, and like to learn about it.

**Free response on the class in general.**
Ben: This was a fun and practical class. I think that I could build other similar embedded systems on my own now. Before this semester, I would have been too intimidated by putting hardware together to try.

Simeon: The class does not only cover the technical aspect of Embedded Systems, but also helps in developing interpersonal skills like teamwork, presentation, and creative thinking.

Kristopher: I appreciated not having to stress about exams in this class because that made it much more enjoyable to come to. Also, the Computer Science major does not offer many practical applications of hardware combined with software, so it was beneficial to see how these two important elements could come together.

Miriam: I enjoyed the class so far, and it gave me an great opportunity to learn about hardware and software.