
MediDetect Reflection

A Reflection of the iOS and CC2500/Arduino System

Alaina Finley - April 27th, 2019

This project was easily the most significant and difficult project I have yet to work on. I felt that this project significantly pushed me, and enabled me to grow substantially as a software engineer.

The hardware used in this project was significantly harder to use than anticipated. I was used to using hardware that had already been preconfigured, and therefore pretty straightforward to working with. The CC2500 was not at all any of these things. Having to learn how to read data sheets to learn what each register value needed to be in order to achieve the settings I wanted was not at all what I was anticipating. And then having to figure out the binary number of the value and then convert that value to hex was again, unexpected and quite the challenge.

Another thing working with this hardware taught me was how to write software when there are little to no resources available. Because this project was quite unique, there were very little resources available for learning how to interface the CC2500 with an Arduino. The sample projects I discovered (there were two) used the wrong modulation, didn't compile, and when I finally was able to get them to compile, they didn't work. This forced me to have to look at the data sheets to understand the functionality of the hardware, and then figure out how to write the functionality in C.

In addition to all of these hardware lessons, I also had very little experience programming in C prior to this project. This project ended up being an extremely C intensive project. I learned quickly how to overcome the language obstacle and how to break the problem apart, think about it in a language I might understand better, and then figure out how to translate that into this newer language.

Switching over to the software side of things, I learned a significant amount as well. My goal of creating an iOS application was certainly met. For one, I was new to Swift programming, so learning how to create an application in a completely new language definitely had its challenges. For one, I struggled with figuring out how to persistently save data. Because of the frameworks needed in my application, I needed to learn how to not only create an application that is written in Swift, I also had to create an application that can work with Objective-C as well. This ended up not being an incredibly difficult endeavor, but it was certainly useful to learn how to work with bridging headers.

I learned how to do UI testing. This was an element of testing I had never worked on before. All of my testing experience was with unit testing, so it was fascinating to learn how to test the flow of the UI. I also learned how to work with storyboards, however, I noticed that although they may seem to make things simple, they also take away from a lot of UI capabilities. The next time I create an iOS application, I would like to try and create the UI programmatically.

Looking at the entirety of this project, there was very little that I knew how to do beforehand. I think overall, that says a lot about the growth I had during this project. Beforehand, I had no idea what modulation was, what data sheets were (let alone how to read one) or how to configure registers/determine register values. I was novice to C programming and completely new to iOS development. Being able to overcome all of these learning obstacles and manage my time, I am very proud of everything I was able to accomplish. With all of this, this project clearly demonstrates the Westminster College Learning Goals of Creativity and Critical Thinking, and the Computer Science Department's Program Goals of acquiring significant project experience working alone and to develop problem solving skills.