SPRING 2018

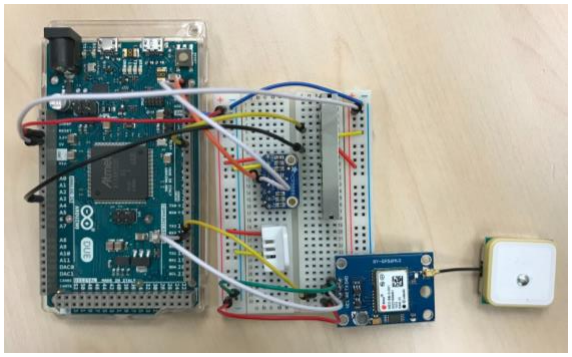
ECE 103 ENGINEERING PROGRAMMING

PROJECT REPORT 1

ABDULRAHMAN ALAMAR: AALAMAR@PDX.EDU
CHUCK FABER: CFABER@PDX.EDU
EBTEHAL ALENEZI: EBTE@PDX.EDU
PORTLAND STATE UNIVERSITY

Table of Contents

HARDWARE



In the hardware aspect of our project we've been making good progress. Our payload tracker's heart is the Atmel ATMega328 in the form of an Arduino board. We've purchased and obtained most if not all of the hardware required for this project, and have spent significant time researching the data sheets of our main components and sensors. This research has allowed us to create the tentative schematic we have below. Our main tracking components include the uBlox Neo 6m GPS module and the Radiometrix HX-1 radio transceiver which operates on 144.39 MHz. We've learned that the GPS module requires 3.3V while the radio transceiver requires 5V. We found out that we will require level shifters to shift the transmission and receiving pins on the GPS module up and down from 3.3V to 5V and vice versa. We are in the process of testing each hardware piece individually to better understand how they work and will interface with our project.

SOFTWARE

So far, in regards to software, our primary goals have been researching how to accomplish the tasks we want to accomplish. The nature of our product involves using several systems which require the use of complicated communication and data protocols. For example, every APRS data packet transmission requires the processing and framing of the data to be transmitted and understanding the broadcasting rules on this frequency.

Using GPS requires understanding of how GPS operates, and understanding the NMEA protocols and strings that are retrieved from GNSS so that we can process them accordingly. Some decisions were made in light of understanding the growing complexity of our task. Originally we were interested in possibly creating our own ground station and writing our own code for decoding the APRS packets, but this turned out to be not feasible. In communicating with several engineers in the field and amateur radio enthusiasts we were informed that this would be too daunting a task for the scope of this project.

Despite the barriers, we made good strides in understanding the data processing required to accomplish this project. During our research we had to look at the code that others have used to accomplish similar tasks, and learned quite a bit about programming concepts we weren't yet familiar with. For example, structures were widely used in the programming and understanding how they were referenced and declared was important. We learned a lot about the various PWM modes on the Arduino and how they could be changed using particular timer pins, and how the timers could be used to identify particular "interrupt" protocols. These protocols called ISRs (interrupt service routines) took a vector as

input and could run regardless of where the program was at in its execution as long as the interrupt condition was met. We learned about volatile and static data types as well as quite a bit more. Having this understanding will make creating our own code a more achievable goal. We are still working to understand how look-up tables are utilized in the signal processing aspect of data processing.

Throughout the process, we've had several points in this project where we've attempted to construct pseudocode. We started off with very general pseudocode as our understanding of the underlying processes was still lacking, and have progressively filled in the details.

Further Work

We still need to do more work to understand the concepts about the data processing we will be doing. In particular, we need better understanding of GPS and processing NMEA codes, as well as the bit-wise operations required to process transmission data. We also need to continue to read the data sheets of the sensors in order to get their I2C addresses and other related communication information. Further individual testing needs to be done on the GPS and the transceiver and functional code should be written for both and tested before trying to merge the systems together. Individual code should also be written for each of the sensors. Once we accomplish the basic requirements of this project we can begin looking into creating a flight-ready prototype.

Discussion

Overall, it has been a challenge to understand the code that others have written to accomplish similar tasks and to understand these data transmission protocols. This project has many interconnected pieces that must work together and some pieces are dependent upon others. We eventually realized to use the sensors it would require the use and understanding of I2C communication protocols. We are slightly worried that time might not be on our side and this quarter is coming to a close rather faster than anticipated. But we've been making steady progress throughout the quarter, and we've learned quite a lot so far.