|  |  |  |  |
| --- | --- | --- | --- |
| Date | Prototype Version | Key Functionality Added | Summary of Work |
| 06 - 08 April 2018 | V1.1 | * Initial Communications * Transmission and receiving data | * Component Testing * Breadboard testing * Error testing * 7-seg display * Accelerometer Test |
| 09 April 2018 | V1.1 | * Basic location detection zone based on packet loss of the transmitted signal | * System clock differential tracking ruled out * Timed ping return based tracking ruled out |
| 15 April 2018 | V1.2 | * Received Signal Strength (RSS) based distance tracking | * Implementing RSS module * Testing signal strength over distance * Mapping decay equation to measured data * Using equation to determine distance |
| 21 April 2018 | V2.1 (Iteration 2 & 3) | * De-acceleration threshold warning system * Water immersion detection system | * Integration of accelerometer and water sensor into transmitter module * Coding system state logic to respond to different stimulus * Testing entire system |
| 27 April 2018 | V2.2 | * Circuit streamlining | * Code refinement * Soldering components |

## Key Lessons from Prototyping

### Methods for Determining Distance

We started by looking into using the system clock, if the system clocks between the transmitter and receiver were synchronised, then all the transmitter has to do is transmit its on-board time, the difference between the received time and the value at the receiver would give the time delay and ultimately the distance. Unfortunately, the arduino chips don’t possess a built in system clock, so synchronising the clocks was close to impossible.

Still attempting to utilise time delay for distance calculation, since the calculation is simple and less susceptible to noise, we looked into pulsed communication (each device is capable of transmitting and receiving). Doing some basic calculations revealed an important flaw:

* Radio waves travel at 300,000,000 m/s
* Oil rigs are approximately 150m x 100m (<http://theweek.com/articles/494480/oil-rigs-cities-sea>)
* If the base station was located at the centre of the rig, and the tag at the extremity, a pulse would have to travel 150m
* This would take 0.5 microseconds
* Even if the rig was 1km in diameter and the signal had to travel 1km, it would only take 3.3 microseconds
* Arduino chips only have a maximum resolution of 4 microseconds (<https://www.arduino.cc/reference/en/language/functions/time/micros/>)

Ultimately, while it would be possible to use time delay if we had sophisticated radio equipment with extremely accurate time keeping functionality, it is beyond the budget and feasibility of this prototype.

Still attempting to adapt the current prototype, we investigated using packet loss to determine the distance between the transmitter and the receiver. Since the stream of numbers that was being transmitted followed a defined order, if any numbers were missed, the amount of data lost could be determined and then the distance between transmitter and receiver could be estimated. The transmitter and receiver code were modified and testing began. It was discovered that the amount of information lost in a short period of time had a lot of random variation. Increasing the distance between the transmitter and receiver didn’t reveal much information, there was little consistency in the randomized loss second to second over short distances. Over slightly longer distances ~20m away, the signal from the transmitter was almost completely lost and not picked up by the receiver.

This property could be used as an asset tracking system, used to establish a safe zone which could set off an alarm when one of the tags drifts outside of the zone, but would not provide that much value for the client. We eventually decided that we would have to modify the design to make use of received signal strength (RSS). The receiver module we had purchased for the prototype does not provide RSS information, and the schematic is not detailed enough to understand where we could read the value from. In order to incorporate this functionality into the prototype, so that the distance between receiver and transmitter could be determined, research into a RSS Arduino receiver antenna was done, and the part was ordered.

### Trade-off between speed and Data

* Programming in the logic for the water sensor was a bit complicated, and reveal an interesting potential behaviour of a real system
  + Since the device is primarily a tracking system, rapid location updates are essential, this favours higher quantities of smaller messages being sent from transmitter to receiver
  + However, this creates a tradeoff with the sensors included on the device, transmitting their data back would make the messages larger, slowing down the rate at which the location could be updated
  + Finding a balance is an important consideration in designing the system
* We went with a status based update system, where the data processing from the onboard sensors is done on the transmitter, and a simple status number is sent to the receiver, this ensure that simple messages can be constantly delivered, which in turn works with the RSS receiver to determine the distance of the transmitter