College of Engineering

CS Capstone Problem Statement

October 21, 2017

100K Spaceport America Demonstration Rocket Project

Prepared for

School of MIME

Nancy Squires ____________ Signature ____ Date

Prepared by

Group 42

Glenn Upthagrove ____________ Signature ____ Date

Sam Hudson ____________ Signature ____ Date

Michael elliot ____________ Signature ____ Date

Abstract

In this problem statement we shall define the basic requirements of the 100K Spaceport America Demonstration Rocket Project. These include the creation of a transmission protocol capable of transmitting enough data uncorrupted to sufficiently track the rocket, a graphical user interface to view the data, and the actual recovery of the rocket. There shall also be a discussion of the proposed solution to this problem. This solution is the creation of a protocol, similar to the User Datagram Protocol, which will detect errors and throw away the corrupted packets, but not correct the errors. The final piece shall be description of performance metrics that will define the success of the project. These performance metrics primarily focus on the collection of the rocket and measurable update speeds for the interface at the ground control.

CONTENTS

1 Definition of Problem

This project requires the tracking and successful retrieval of a rocket. This presents interesting challenges from a software standpoint. The most important considerations from a software standpoint are handling dropped and corrupted packets of data, interaction with hardware that is not common place, and the interaction of multiple software modules in vastly different environments. The software must also account for hardware reading errors. As hardware noise will be unavoidable in this project, the software must make a best estimate of the actual values rather than rely on the hardware measurement.

2 Proposed Solution

The proposed solution is the creation of a transmission protocol for transmitting the data from the rocket back to the ground station. This protocol shall be based off of the User Datagram Protocol (UDP), used in modern media streaming applications online. A protocol based off of UDP would be an appropriate solution to this problem, since UDP is fast and lightweight. The lightweight nature of this protocol would be beneficial, most importantly when considering the relative lower specifications of a micro controller, such as the Telemega, that will most likely be used as the main board in the final rocket. We expect to be transmitting packets several times a second, more than enough data to track and collect the rocket. Since several packets will be transmitted and not all of them absolutely necessary, we will not attempt to correct any errors, but instead focus on detecting errors, and dropping a packet if it is corrupted. This will keep the processing on the rockets side to a minimum, as well as keep the packet size small, and thus increase our chance of complete and uncorrupted communication.

As our telemetry data will be loss tolerant our aim would be to implement a unidirectional communication transmission protocol which would be ideal for broadcasting. As each data is independent of one another and there's not a requirement for bidirectional transmission this is the most suitable approach. A bidirectional protocol implementation would not be suitable for our specific use case. Attempting to correct errors would make packets longer, increasing the chance of failure. In additional, large packet sizes would cause more computational stress to the limited set of hardware we have available on the rocket.

The data will be organized into packets of equal length, long enough to hold either all of the data, or more likely, the longest output from the sensors. The exact size will be decided upon when the hardware is finalized. Most likely each sensor will get to transmit one at a time in a round robin style, to minimize packet sizes. The data will also be converted into JSON format or something similar, for easier interpretation and use with existing tools. Each packet will carry a parity bit to allow for fairly quick detection of errors. A Kalmin filter will also then examine the data and compare it to an ideal mathematical model.

All of the retrieved data will be stored at the ground station for later review as well as the tracking of the rocket while it is in flight. This data will be displayed at a reduced rate such that it is possible for the rocket to be tracked. This would mean formatting the data in such a way that it is human readable and updated at a speed reduced enough for humans to keep up. All of this will be handled by fairly simple algorithms on the ground station side, and not done on the rocket, as this would cause undue stress on the reduced hardware profile. The data will then be used to update the GUI.

3 Performance Metrics

The most important and also most basic performance metric shall be, assuming proper construction by the other sub-groups, the retrieval of the rocket. If the rocket survives flight, then the retrieval of the rocket shall be considered a success for the tracking system.

Another metric worth noting is the transmission is good enough, that barring complete signal loss, the GUI can be updated at least once a second. thus making the GUI human readable and smooth.

The protocol must be simple enough that the hardware on the rocket can still collect, package, and transmit the data from each sensor at least 4 times/second, under perfect conditions.

The GUI shall be able display the estimated latitude, longitude, and altitude as raw numbers. The GUI shall also present the estimated position overlaid onto a map.

The visualization shall also be able to display a line representing the trajectory overlaid a map in 3D based off of data read in from a file of any length.