CS Capstone Problem Statement

October 10, 2017

OSU Robotics Club
Mars Rover Ground Station

Prepared for

OSU Robotics Club

Nick McComb

Prepared by

Group 30

Kenneth Steinfeldt
Christopher Pham
Corwin Perren

Abstract

For this project our group will design and create the ground station software for the Oregon State University Robotics Club. When the Mars rover embarks on its mission it will require a ground control station that will direct and monitor the rover through the mission. In addition to directing the rover the control station will also monitor various aspects of the rover. These include a video feed, a navigational map, an accurate compass indicator, waypoint editing and placement, core system status, arm joint position, etc.

In order to achieve this we will be using the Python 3 language following the PEP8 standard. The software will use the Robot Operating System (ROS) framework to monitor the rover and the Python QT framework to power the user interface. The Ground Control Station will communicate to the rover via a remote ethernet link that can dynamically adjust bandwidth according to necessity. When complete the Control Station will allow the user to fully control and monitor the Mars Rover remotely through the variety of challenges and obstacles laid before it at the University Mars Rover Challenge in Hanksville, Utah.

CONTENTS

1 INTRODUCTION

The Oregon State University Robotics Club will be competing in the University Mars Rover Challenge held in Hanksville, Utah, May 31 - June 2, 2018. The rover must be a standalone, off the grid, mobile platform. In order to accomplish this task a ground station is required in order to fully monitor and operate the mobile rover as it navigates the obstacles of the competition. The following document lays out how our group will design and write the software that will accomplish this task.

2 REQUIREMENTS

2.1 programming

The software must be written in Python 3.6 following the PEP 8 standard. The GUI must be written using QT and the PyQT5 framework. The software must interact with the rover's Robot Operation System (ROS) framework in order to handle the transmission of: control, status information, and video feed over a remote ethernet connection. Bandwidth must be dynamic in order to adjust to varying environmental constraints, this should be both automatic and manual.

2.2 Graphic User Interface

The GUI will cover 2 monitors running at 1080p resolution. The left monitor will display the following:

- selectable video display
- main navigational map
- compass
- IMU status
- waypoint indicators
- arm position
- science data
- logging information

The right monitor will display the following:

- primary video stream
- video tools, such as record to disk

2.3 Hardware

The software will running on an Intel NUC device running Ubuntu 16.04 LTS able to drive the two required 1080p monitors. The primary control interface will be two USB joysticks. A keyboard and mouse will be connected to the NUC and usable. A remote ethernet network connection will be provided via 2 Ubiquiti Rocket M2 radios. The base station will be self-contained for easy deployment.