Watson Capstone Projects (WCP)
WCP09 Autonomous Beacon Location System (ABLS)
Fall Design Report

Sponsor: Lockheed Martin

Submitted by:
Jonathan Felder, ME
Henry Chen, CoE
Haosen Zheng, CoE
Ethan Terwilliger, EE
Joseph Bourque, EE

Faculty Advisor: Prof. Jack Maynard
Industry Mentor: Alfredo Iturralde, Lockheed Martin

November 30, 2018
Revision: -

Approved for public release; distribution is unlimited.
Submitted in partial fulfillment of WCP academic requirements.

Thomas J. Watson School of Engineering and Applied Science
Binghamton University
Binghamton, NY

Executive Summary

The Automated Beacon Location System (ABLS) solution addresses a problem that park services is facing with the search and rescue of missing hikers. In recent years there have been increasingly insufficient personnel and resources available for park services to respond to stress calls from lost hikers and the ABLS looks to correct this problem. The ABLS provides an autonomous solution that will assist park services in locating the lost hiker by scouting a route for the park service personnel to get to the hiker. The ABLS consists of three parts: a beacon carried by a hiker, a vehicle that will navigate autonomously to that beacon, and a base station from which the operator will control and monitor the vehicle and track its location, speed, and other various data.

The beacon being used for the project is a simple radio signal emitter. The vehicle chosen is a small rover that is equipped with a first person view camera, an on-board Raspberry Pi controller, sensors of various types, transmitters for the video feed and telemetry data, and receivers for the beacon signal and operator commands. These operator commands can be issued from the base station at any time, including an emergency situation. The vehicle is capable of locating the direction of the beacon and navigating towards it autonomously, while avoiding obstacles it encounters. During its operation, the vehicle will also relay its status, telemetry data and Global Positioning System (GPS) information to the base station.

The software for the ABLS system will be written mostly in C++, with the graphical user interface written in Java and some of its algorithm written in Python. The software system will be developed and ran on Unix-like environments.

Currently, the team has been successful in clearly defining the problem statement and has come up with many different alternative designs that have been consolidated to the final rover design mentioned in detail in this report with targeted parts and components. The team is planning to proceed onto the next phase which entails the purchasing of parts and components. This process is expected to be completed by January 21st. In the spring semester, the team will be building and integrating the subsystems and its parts. Testing will be performed alongside the development process as subsystems and parts are being completed and integrated into the ABLS.

The sponsor of this project is Lockheed Martin. A funding limit for the ABLS has been set at $750. After choosing the parts that will be used, the expected cost of the ABLS is $635, well within the allocated budget from the sponsor. The expected reserve of $115 will be used to cover the cost of maintenance, replacement, and/or repair of the ABLS during the building and testing phase.

Table of Contents

List of Figures

List of Tables

1. Introduction

1.1 Initial Scope

The goal of the Autonomous Beacon Location System (ABLS) is to navigate to a locator beacon with minimal assistance from an operator. The vehicle will send video feed and telemetry data to the base station where the operator is located. The base station will also map the path the vehicle takes to reach the beacon.

1.2 Applicable Documents

Refer to the following documents for more information about the project.

1. WCP09 Autonomous Beacon Location System Project Specification, published September 28,2018.
2. WCP09 ABLS Conceptual Design Document, published November 4, 2018.
3. WCP09 Autonomous Beacon Location System Project Development Plan, published November 15, 2018

2. Problem Definition

The goal of ABLS is to prove a concept, rather than find the best solution for a problem. The test scenario provided for proof of concept is that of locating lost hikers. Each hiker will have a locator beacon. The park rangers will be positioned at the ABLS base station while the vehicle determines and travels a path to reach the hiker. At the base station, the rangers require a map of the vehicle's path with GPS coordinates, the total distance of the path, and a live first person view (FPV) video feed as it travels. Furthermore, the system will display the battery life, speed, and direction of the vehicle, as well as the distance from the vehicle to the beacon.

The ABLS will be tested in a park located near Binghamton, New York. Binghamton is mostly forested with some large bodies of water that are off limits to all hikers. ABLS will be expected to operate in wet conditions, including small amounts of rain and snow. It will also be required to deal with a variety of obstacles, from plants and animals to holes, trees, and fallen debris.

2.1 Project Scope

The ABLS is a proof of concept to an autonomous solution that will locate and navigate to a radio frequency (RF) signaling beacon, while gathering and providing information needed for a person to travel to that signaling beacon. The environment that the ABLS is expected to operate in is forested areas with obstacles and terrain, including potholes, puddles, snow, mud, fallen branches, trees, and rocks. The ABLS will operate with minimal operator input, have the ability to avoid obstacles, seek goals, and map routes, and communicate wirelessly with a base station. The ABLS will gather and relay telemetry data and positional information to the operator wirelessly via radio signals.

2.2 Technical Review

For hikers, essential skills include reading contour maps; determining what is a mountain, valley, river, creek, etc is important to a successful trip into the wilderness. For hikers it is also important to leave a detailed description of where they are going and when they expect to be back with close friends and family. Essentially hikers need a plan and contingencies in case they get lost or injured during their ventures. However, even with all the precautions, the unexpected still happens. Currently, there is no system in place to locate lost hikers. Most searches are conducted by people, preferably trackers, and begin where the hiker was last seen. This process is not highly technical and involves some luck to be successful. When hikers are lost, it is often first the responsibility of the park services to locate and help these hikers, before police or other personnel are called in.

Recently, there has been a high volume of lost hikers and park services does not have enough personnel to respond to all of the cases. Currently, park services personnel must handle each case one at a time and hope they can make it to all lost hikers in time. In dire situations, external help is called in which can be expensive. With the influx of lost hikers, the park rangers simply cannot help everyone at once.

The ABLS solution is to create an autonomous ground-based vehicle that will relay first person video and Global Positioning System (GPS) data to a base station to map a route to the beacon, one of which will be carried by each hiker. The beacon will send out an RF signal containing its location, which will allow the base station and vehicle to determine the lost hiker’s location.

2.3 Design Requirements

The ABLS was initially proposed by the customer with thirteen requirements. From this given list, lower level requirements were derived to help mold the system's design. A complete list of all the project requirements can be found in *WCP09 Autonomous Beacon Location System Project Specification*.

From the customer requirements, seven were critical factors in determining both the overall system design and the construction of its subsystems. The most important was that *{WCP09-R-002}* The System sh*all be a ground-based or air-based vehicle capable of autonomously navigating to the hidden beacon.* This led to the development of two main types of designs; one based on a rover and the other based on a drone. While a drone would have to comply with *{WCP09-R-018} The System and the Operator shall comply with FCC rules and regulations*, and *{WCP09-R-019} The System and the Operator shall comply with FAA rules and regulations*, a rover would only be required to meet those of the FCC, making it a slightly more attractive option.

The subsystems were also driven by customer requirements. First, the system shall be capable of autonomously navigating to the beacon. This is required by, *{WCP09-R-001} The System shall be capable of locating an amateur radio signaling beacon from an undefined location.* Second, the system shall be capable of navigating around obstacles as stated by, *{WCP09-R-006} The System shall be capable of navigating around obstacles*. This led to the creation of the navigation subsystem, which utilizes multiple sensors to detect obstacles, such as trees, branches, puddles, holes, and cliffs. An algorithm will be implemented that will use the sensor data, as well as the beacon's signal to safely navigate the vehicle to the beacon.

Other customer requirements specified the need for a base station to be created as well. This is stated in, *{WCP09-R-011}The System shall display telemetry data on the base station*, which will display the directional path of the vehicle and map out its course. It shall also display telemetry data, as defined in, *{WCP09-R-012} Telemetry data shall include: distance traveled, distance to the beacon from the rover, direction of the rover (N, S, E, W), speed of the rover, and battery life.* It shall display FPV (first person view) of the vehicle's mission. *{WCP09-R-010} The System shall have a First-Person View of the ground-based vehicles mission.* These requirements will be satisfied by the development of a graphical user interface (GUI) on the base station, as well as a communication subsystem to relay information from the vehicle to the base station. The vehicle will be equipped with a GPS locator, which will provide the required telemetry data. *{WCP09-R-003} The System shall map the location and directional* *path of the ground-based vehicle on the base station using GPS.* As stated in, *{WCP09-R-005} The System shall be capable to operate a mission for a minimum of 20 minutes.* This will allow appropriate time for the vehicle to accomplish its task. while navigating various terrain to get to its end goal as stated in, *{WCP09-R-007}The System shall be durable enough to handle various terrains.* Lastly, *{WCP09-R-021} Each Operator shall obtain an Amateur Radio License*, requires a team member to obtain a license for testing and demonstration purposes. In order to not be confined by that specific team member's schedule, two team members will obtain the certification.

3. Design Description

3.1 Overview

The first major decision made for the ABLS was between a ground-based or air-based system. The information and reasoning for selecting the ground-based rover is contained in the *WCP09 ABLS Conceptual Design Document*. This document also contains alternative designs for the drone and some of the subsystems.

ABLS is an autonomous rover, designed to locate a beacon. One of the receivers accepts the beacon's location while front mounted sensors look for obstacles or water in front of the vehicle. An algorithm will use this information to navigate a safe course towards the beacon. These external system interactions are shown in Figure 1 below.

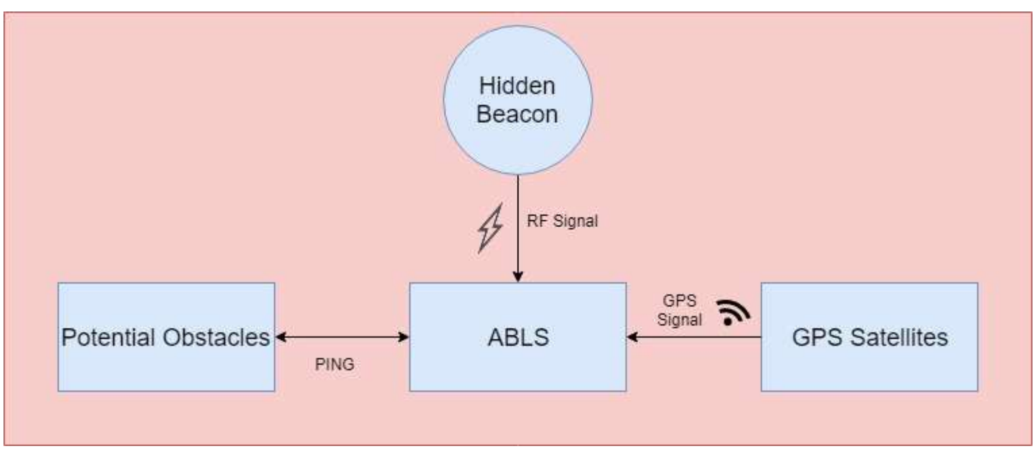


Figure 1: Operational Context Diagram

While the vehicle is in motion, its speed, heading, and geographical position will be determined by a GPS module attached to the vehicle. Camera feed will be transmitted to provide a first person view of the vehicle. All of this data will be received at the base station and displayed through a GUI to the operator. Lastly, the vehicle will have a built in override command so that the operator can take manual control of the vehicle at any point in time.

3.2 Detailed Description

3.2.1 Chassis

The chassis selected is the Dagu Wild Thumper four wheel drive. This rugged, 4-wheel-drive chassis from Dagu Electronics has an independent suspension for each of its spiked 120mm-diameter wheels. This allows for traction over various uneven terrain. Its body is composed of 2mm-thick anodized aluminum with 4mm holes, to allow electronics and accessories to be mounted to the chassis. The following are the specifications for the chassis. Figure 2 shows the 3D model of the chassis.

- Size: 280 × 300 × 130 mm (11" × 12" × 5")
- Weight: 1.9 kg (4.1 lb)
- Ground clearance: 60 mm (2.5") when lightly loaded
- Recommended motor voltage: 2 – 7.5 V
- Stall current at 7.2 V: 6.6 A per motor
- No-load current at 7.2 V: 420 mA per motor
- No-load output shaft speed at 7.2 V:
    - 160 RPM for the version with 75:1 gearboxes
- Stall torque at 7.2 V:
    - 11 kg-cm (160 oz-in) per motor for the version with 75:1 gearboxes

Figure 2: 3D model of the vehicle chassis. [2]

To protect the electronics from the outside elements, an enclosure is to be placed over the chassis. The enclosure will be 3D printed and made of Clear FLGPCL03 Resin. This resin was chosen due to its high heat deflection temperature when it is postcured. Figure 3 shows the specifications of Clear FLGPCL03 Resin.

	Green[3]	Postcured[4]
Mechanical Properties		
Ultimate Tensile Strength	38 MPa	65 MPa
Young's Modulus	1.6 GPa	2.8 GPa
Elongation at Failure	12 %	6.2 %
Flexural Modulus	1.25 GPa	2.2 GPa
Notched IZOD	16 J/m	25 J/m
Thermal Properties		
Heat deflection temp. @ 264 psi	42.7 °C	58.4 °C
Heat deflection temp. @ 66 psi	49.7 °C	73.1 °C

Figure 3: Clear FLGPCL03 Resin specifications

To avoid blurry video from the camera, a 10 mm hole will be cut out in the enclosure for the lens. The camera lens will fit into the hole via a press fit. An interference fit was not chosen due to the risk of the plastic cracking when assembling the camera and the enclosure. To avoid extra weight, the enclosure is very thin (6 mm). Due to this, no screw threads will be incorporated into the 3D printed design. Instead, a simple pilot hole will be incorporated into the design. Each pilot hole will be at least 12.7 mm away from the edge of the enclosure to avoid cracking during assembly. The pilot hole sizes will be 1.524 mm to accommodate the screw. A 18-8 stainless steel socket head screw will be used. It is a 0.06-80-UNF-3A screw, chosen to keep the enclosure securely mounted so no electronics on the chassis are exposed to any moisture.

3.2.2 Motors

The motors are provided with the chassis and have a gear ratio of 75:1. The motors support a differential-drive, which means that turning is accomplished by driving the motors at different speeds. The two motors on each side of the robot are wired in parallel, so only two channels of motor control are required to get this chassis moving. The specifications for the motors are intended for a maximum nominal operating voltage of 7.2 V with a minimum of 2V, and a stall current of 6.6 A and a no-load current of 420 mA at 7.2 V. But the motors will draw the full stall current when suddenly starting from rest and almost two times the stall current when suddenly changing directions at full speed. For this reason, a motor driver capable of supplying 14A combined per-channel stall current of these motors at 7.2 V is needed. To accomplish this, the TReX dual motor controller offers an RC interface to accomplish the task. It can act as a simple electronic speed control (ESC) as well as a serial interface for autonomous operation, and it features the ability to combine the two for purposes of this project.

3.2.3 First Person View Camera

Figure 4: Runcam Racer

The camera module selected is the Runcam Racer, shown above in Figure 4. The specifications are as follows [5]:

- Super Wide Dynamic Range (WDR) CMOS Sensor image sensor
- 700 TVL Horizontal Resolution
- Latency as low as 6ms
- M8 2.1mm lens – 150° FOV
- Switchable 4:3/16:9 Aspect Ratio
- Input Voltage: DC 5V – 36V
- Power consumption: 110mA@5V or 40mA@12V
- Supports Camera Control via UART

- Weight: 5.5g

The traits that were considered included, cost, wide dynamic range (WDR), field of view (FOV), size, weight, imaging sensor (CCD vs CMOS), and latency.

WDR was important since it determines how well a bright or dark spot is portrayed. This is shown in Figure 5 below.

Figure 5: Effect of WDR

The field of view provides the range of degrees that will be captured by the camera. While a larger range would see more terrain, too high a range can cause a fish eye effect to occur which can distort images. After visual analysis, it was determined that a range of 130 to 150 degrees is desirable.

The next criteria to consider was the imaging sensor, including the choice between CCD (charge-coupled device) or CMOS (complementary metal-oxide semiconductor). CCD typically has less "jello" effect in footage due to global shutter, less digital noise in low light, and generally warmer color. CMOS has lower latency and higher resolution, but can also have more digital noise, more natural image colour, and better low light. This choice was made for CMOS since in forested areas, better low light becomes an extremely important feature for a camera.

Latency is the time required for the FPV camera to capture and process an image. Latency can be a deciding factor when controlling a vehicle. Although the vehicle will not be traveling at high speeds it will be helpful to the operator to view FPV footage with minimal delay possible.

3.2.4 Video Transmitter and Receiver

Figure 6: 1.2 GHz transmitter and receiver modules

The receiver and transmitter chosen for the camera module is a generic model found online. The driving factors behind this choice were cost effectiveness and working frequency. There were several options for the frequency of the receiver and transmitter, including 1.2 GHz, 2.4 GHz, and 5.8 GHz. Simply put, the higher the frequency, the more data that can be sent, but the tradeoff is that those frequencies are more prone to interference. Since the system is designed to operate in a forested area, a 5.8 GHz signal may get blocked by the numerous trees in the way. The 1.2 GHz frequency was selected since it is low enough to have good penetration through objects but also high enough to transmit all of the necessary video data. The following is the specification for the receiver and transmitter module:

TX1000:

- Voltage Input : DC12V
- Working Current : 350mA
- Working Frequency : 1.2GHz
- Channel : 8CH
- Transmission Power : 1000mW
- Size : 50 x 25 x 8mm
- Weight : 30g

RX02:

- Voltage Input : DC12V
- Working Current : 300mA
- Working Frequency : 1.2GHz
- Channel : 12CH
- Video Output : 1NP-P
- Size : 114 x 80 x 20mm
- Weight : 110g

3.2.5 Controls Transmitter and Receiver

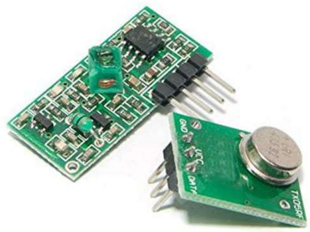


Figure 7: 433 MHz transmitter and receiver modules

The transmitter and receiver that will be tasked with the sending and receiving of telemetry data and commands from the operator will be the WINGONEER 433Mhz RF Wireless Transmitter Module and Receiver Kit for Arduino/Raspberry Pi. The same reasoning was applied here compared to the video transmitter and receiver. Two will be used for the project, with one set traveling with the rover and the other set stationary at the GUI each will transmit data between the rover and base station. The first set will be used for rover control commands from base station to rover, while the second will be used to transmit telemetry data from rover back to the base station. To differentiate between the two signals a header will be used on the data outputs to distinguish control messages from data messages. The following are specifications of the modules:

Receiver:

- Operating voltage : DC 5V
- Quiescent Current : 4mA
- Receiving frequency : 433MHZ
- Receiver sensitivity : -103DBm
- Size : 30 x 12 x 7mm
- Working temperature : -20°C ~ +70°C

Transmitter:

- Launch distance : 20-200 meters (different voltage, different results)
- Operating voltage : 3-12V
- Transfer rate : 4KB / S
- Transmitting power : 10mW
- Transmitting frequency : 433MHZ
- Working current : 20-28mA
- Standby current : 0mA
- Output power : 16dBm (40mW)
- Transfer rate : <10Kbps

- Working temperature : -10°C +70°C
- Size : 19×19×8mm
- Pinout from left to right : (DATA; VCC; GND)

3.2.6 Beacon

The beacon is not part of ABLS. It is however, required for testing ABLS and is not provided by the customer. The only specifications for the beacon were derived by the team. It must comply with FCC regulations so the system can be used in the United States. It also must be small enough and waterproof enough that a hiker would reasonably carry it on a hike. While these are by no means perfect requirements, they provide enough basis to create a simple beacon by which to test the system.

Currently, there are two designs that will both be created for the beacon and tested to determine which works better. The first is based around a Raspberry Pi Zero and the second around a 555 timer. The Pi Zero Model will utilise a 315 Mhz transmitter to broadcast a simple message that can be detected by a corresponding 315 Mhz receiver on the rover. The design for the 555 timer model is shown below in Figure 8. The beacon will be powered by a 5V battery and encased in a waterproof enclosure.

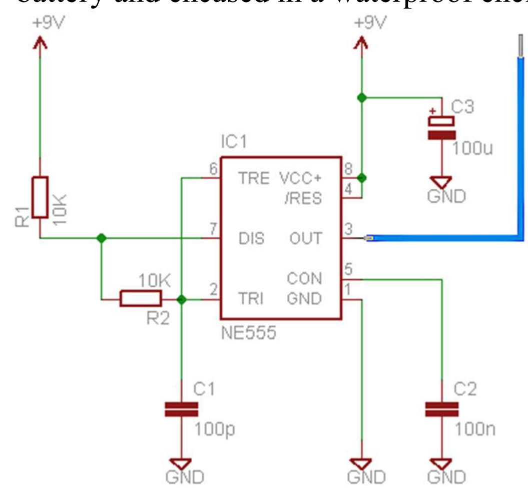


Figure 8: 555 Timer Beacon Circuit Diagram [6]

3.2.7 Global Positioning System and Gyroscope

To fulfill *{WCP09-006}The System shall map the location and directional path of the ground-based vehicle on the base station using GPS,* the system will use the GPS FeatherWing module. The FeatherWing connects into the Raspberry Pi and provides precise location identification. The FeatherWing is capable of keeping track of time once it is synced with the satellites. The following are the specifications[4]:

- 165 dBm sensitivity, 10 Hz updates, 66 channels
- 20mA current draw
- RTC with coin battery backup
- Built-in datalogging
- PPS (pulse per second) output on fix
- Internal patch antenna + u.FL connector for external active antenna
- Fix status LED

The gyroscope will provide valuable data on the orientation of the vehicle. The scenario may be that the vehicle has rolled over and the gyroscope can alert the operator that the vehicle is in need of assistance. The selected module is the Adafruit 9-DOF Accel/Mag/Gyro+Temp Breakout.

This is a 9 degree of freedom sensor that can determine which direction is down towards the Earth or how fast the board is accelerating in 3D space. The board is capable of sensing where the strongest magnetic force is coming from, and can measure spin and twist.

3.2.8 On Board Computer (Raspberry Pi)

For the on board computing device, the Raspberry Pi was chosen to handle all controls for the rover's autonomous navigation and communications back to the base station. The rationale for the choice was simple; Raspberry Pi boards contain a diverse amount of ports and have excellent compatibility with other parts. The Raspberry Pi 3B+ features can be seen on the specification list here [3]:

- CPU type/speed ARM Cortex-A53 1.4GHz
- RAM size 1GB SRAM
- Integrated Wi-Fi 2.4GHz and 5GHz
- Ethernet speed 300Mbps
- Bluetooth 4.2

The Raspberry Pi 3B+ was selected from all the Pi models due to its combination of low cost and flexibility to reduce risks later on. If changes or additions are necessary at a further point in the project, this model will have enough ports and computing power that no set backs will occur. This decision includes paying more for the Raspberry Pi 3B+ compared to Raspberry Pi zero but in terms of the lifetime of the project, the Raspberry Pi 3B+ is a safer option with its extra ports and more powerful hardware features.

Figure 9 below shows how different components will be connected to the Raspberry Pi.

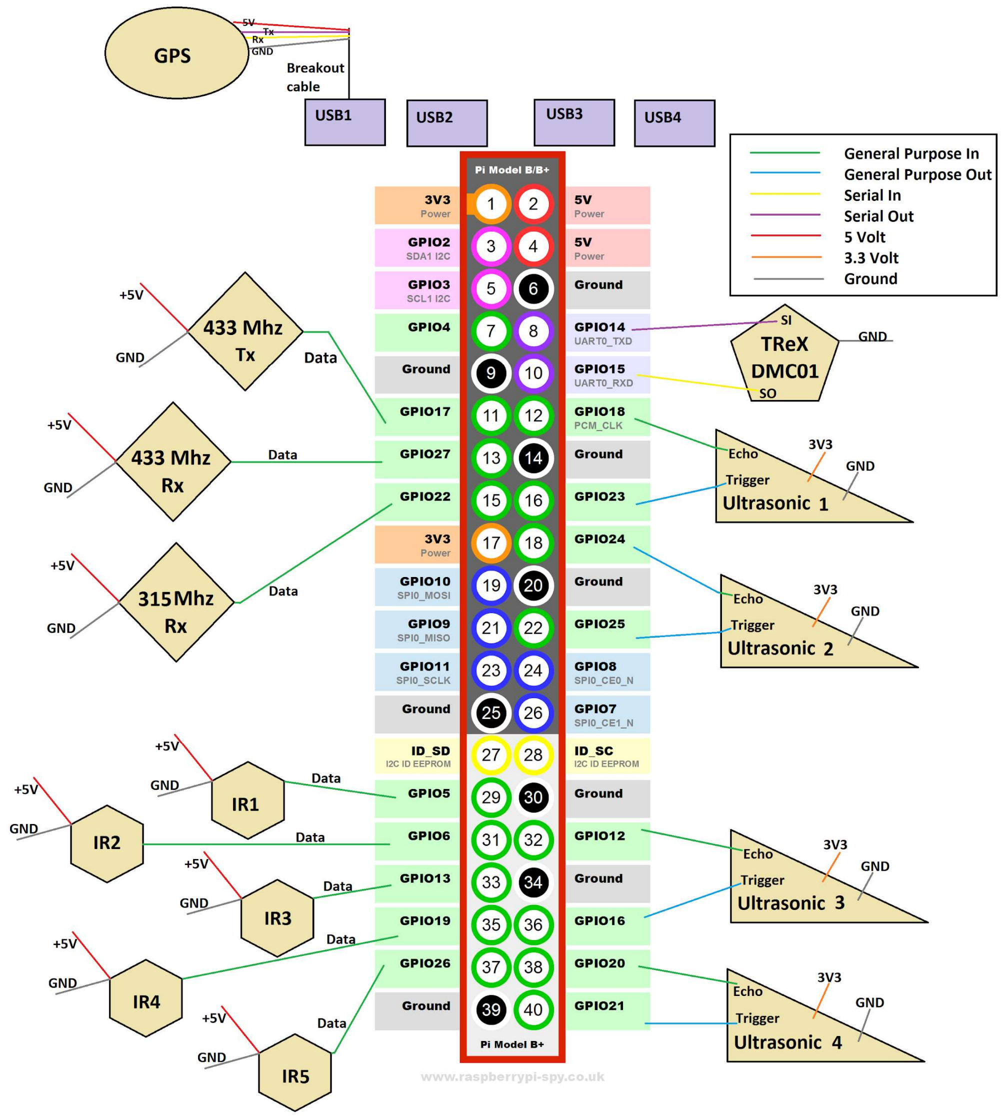


Figure 9: Raspberry Pi Port Connections

3.2.9 Sensors

The rover will be equipped with five infrared sensors and four ultrasonic sensors. The five infrared sensors have a 35 degree wide detection beam. This fits into the avoidance algorithm which uses ~180 degree detection in front of the vehicle to avoid obstacles. The five sensors will cover 175 degrees which will be sufficient for the avoidance algorithm.

The four ultrasonic sensors will be mounted to the front of the rover to detect water. Two of the sensors will be angled to detect water about one foot in front of the rover while the other two sensors will be angled to detect water about four feet in front of the rover. If all four sensors detect water, the rover is approaching a large body of water and the obstacle

avoidance algorithm will navigate the rover away from this body of water. Otherwise, if only two of the ultrasonic sensors are detecting water, then the rover is likely approaching a small puddle it can drive through. The first person view camera for the operator at the base station will be used as a failsafe. If the rover does not navigate away from a large body of water, or cannot detect a large body of water, the operator will override the autonomous rover and direct it away from the body of water.
The following are the specifications for the infrared sensors:

Figure 10: Infrared Sensor Module

Features:

- There is an obstacle, the green indicator light on the circuit board
- Digital output signal
- Detection distance: 2 ~ 30cm
- Detection angle: 35 ° Degree
- Comparator chip: LM393
- Adjustable detection distance range via potentiometer:
- Clockwise: Increase detection distance
- Counter-clockwise: Reduce detection distance

Specifications:

- Working voltage: 3 - 5V DC
- Output type: Digital switching output (0 and 1)
- 3mm screw holes for easy mounting
- Board size: 3.2 x 1.4cm

The following are the specifications for the ultrasonic sensors shown in Figure 11:

Figure 11: Ultrasonic Sensor Module

- Working Voltage DC: 5 V
- Working Current: 15mA
- Working Frequency: 40Hz
- Max Range: 4m

- Min Range: 2cm
- Measuring Angle: 15 degree
- Trigger Input Signal: 10uS TTL pulse
- Echo Output Signal Input: TTL level signal and the range in proportion
- Dimension: 45*20*15mm

3.2.10 Battery

Figure 12: Battery Pack

The batteries chosen for the rover and various components on the rover are the Turnigy Stick Pack Sub-C 3000 mAh 7.2V NiMh High Power Series batteries. Two are being purchased to be used on the project. The reason behind choosing these batteries is because the chosen rover chassis suggests two 7.2V sub-c battery packs. Additionally, this battery pack is cheaper than alternate options. The specifications for the battery are shown below:

- Category: Sub-C battery
- Capacity: 3000mAh
- Discharge: 10C constant / 15C burst
- Voltage: 7.2V
- Chemistry: Ni-MH rechargeable
- Weight: 340g
- Dimensions: 133x45x23mm
- Discharge Plug: Tamiya style

3.2.11 High Level Overview

Most of the hardware elements contained in the system can readily be seen in the system and subsystem diagrams. The base station, shown in Figure 13 below, is very simplistic, consisting of a laptop, two receivers, and one transmitter. The 433 Mhz receiver will be for the telemetry data while the 1.2 GHz will be for the video from the rover. A 433 MHz transmitter will enable the base station to send control commands to the rover.

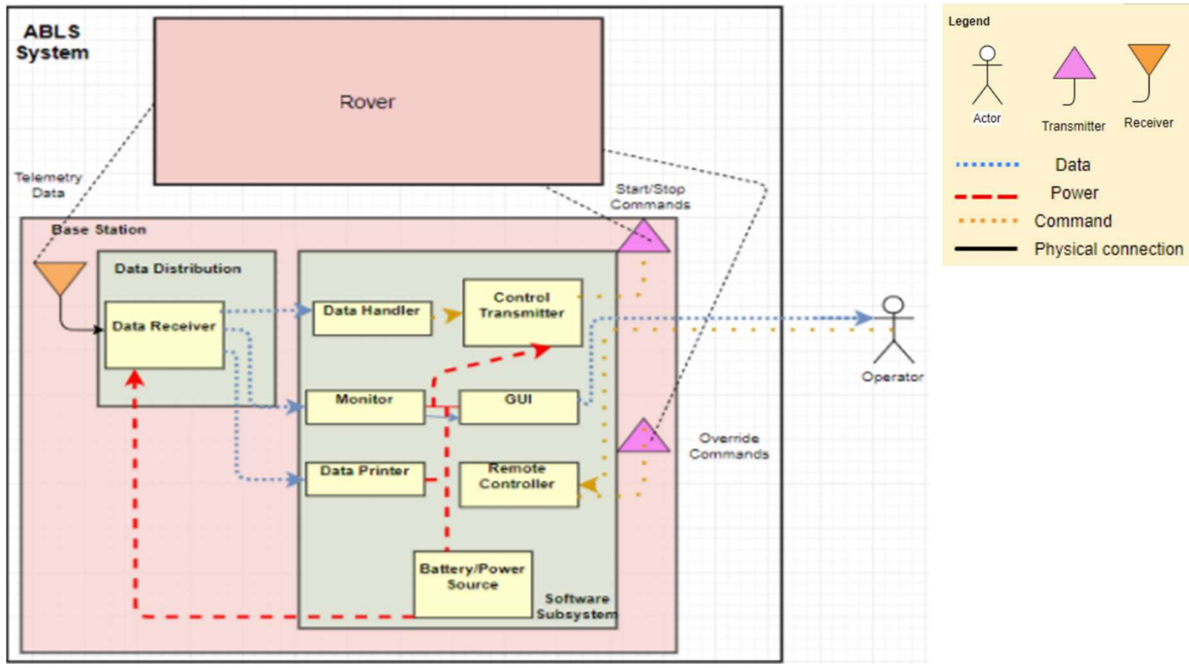


Figure 13: Upper Level System Diagram

The ground-based system, shown below in Figure 14, will be built on top of a Dagu Rover Wild Thumper chassis. Multiple battery packs will provide power for the different components and subsystems on the vehicle. The vehicle will be equipped with two transmitters and two receivers. A 1.2 GHz transmitter will send video to the base station while a 433 MHz transmitter will send the telemetry data. A 433 Mhz receiver will receive controls and commands in return from the base station. The messages will be differentiated by their headers. Finally a 315 MHz receiver will be used for detecting the locator beacon.

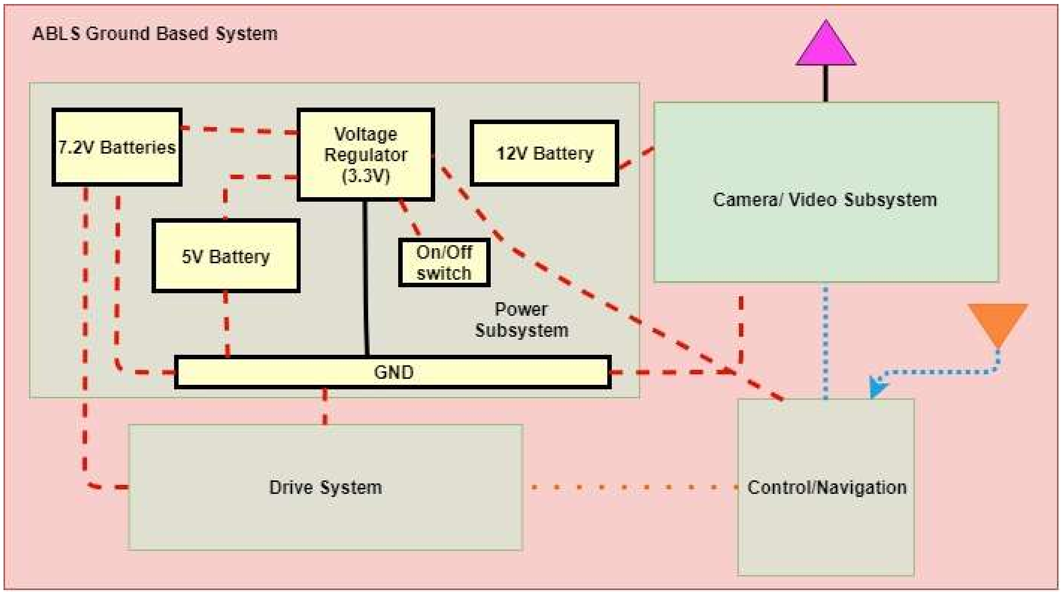


Figure 14: Rover System Diagram

The hardware components of the subsystems are illustrated in Figure 15 below. The navigation subsystem consists quite simply of a Raspberry Pi 3B+ and a number of ultrasonic and infrared sensors. A GPS tracker is also used to provide positional data and speed to the vehicle. The drive system will receive motor commands from the navigation system at the TReX DMC01 motor controller. This device will manage the motor speeds and directions. Lastly, a Runcam Racer camera will be used to provide FPV video to the base station. The camera will be mounted to the rover chassis using a mount that will be either pre-built or 3D printed. The FPV captured by the camera will be transmitted over the 1.2 GHz video transmitter to its receiver attached to the base station.

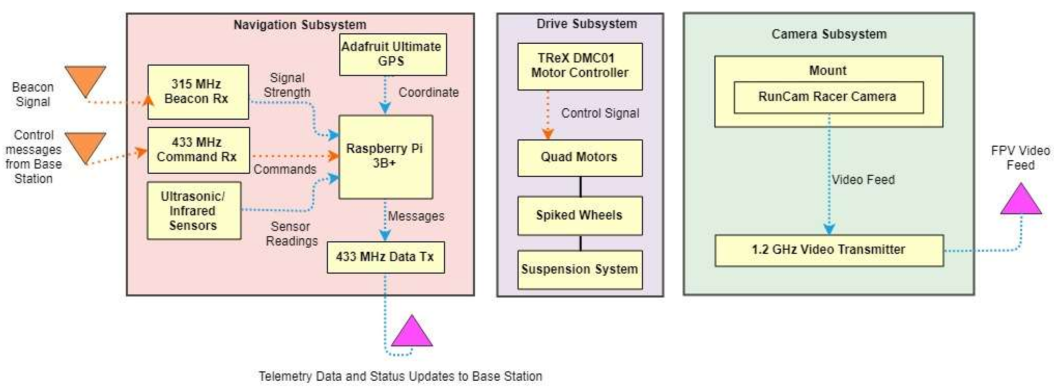


Figure 15: Rover Subsystem Diagram

3.2.12 Software Toolchain

For the software system of the entire project, a combination of C++, Java, and Python were selected as the programming languages, with the support of Robot Operating System (ROS) framework. The software system for both the base station and the vehicle will be expected to run on a Unix-like operating system. The base station software system is expected to run on Ubuntu 18.04, and the vehicle software system is expected to run on Raspbian.

Since the software system will be used mainly for controlling a rover, the relatively lower level and yet more user friendly C++ were chosen as the main programming language for this project. Python is expected to be used to write mathematically complex algorithms as Python offers numbers of very elaborated, high level, and user friendly libraries that supports mathematical modeling and implementation. Alongside C++ and Python, the ROS framework was also chosen to facilitate the development and integration of many software components. ROS was designed to be as modular and distributed as possible. On top of that, the ROS community has contributed numerous packages that are available for potential use in this project.

Java will be used mainly to create a simple graphic user interface to provide the operator with an easier method of controlling the vehicle. Java was chosen to be the language that the GUI would be written in because one of the team members has some prior experience using JavaFx for GUI development. The Java GUI is expected to interface between the less user friendly terminal function calls with buttons and other graphical feature that the operator can interact with.

Software that is written in C++ and Python will be developed using Visual Studio Code that contain plugins/extensions which enables syntax checking and highlighting, as well as integrated terminal, version control, and debugger. The compiling of C++ source code will be supported by the GNU Compiler Collection. Java development will be done via the Java Integrated Development Environment(IDE) IntellIJ IDEA. All of the software source code will be version controlled via Git, and be hosted on a public repository in GitHub.

3.2.13 Top Level Design Description

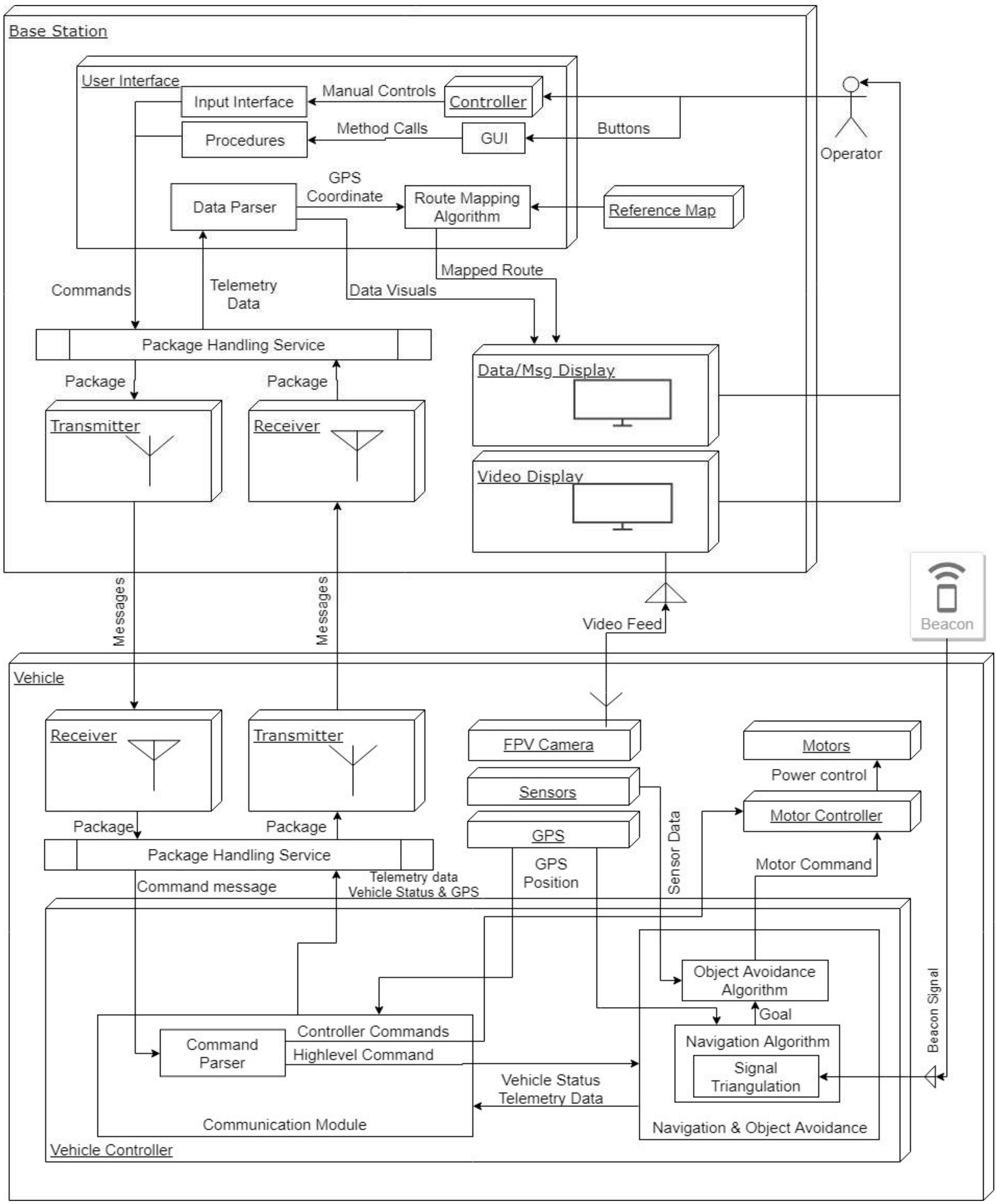


Figure 16: Software High Level block Diagram

The Software system, shown in Figure 16, of the ABLS is composed of two major subsystems who will constantly communicate with each other throughout the operation. The first software subsystem is the base station subsystem, this subsystem will serve as the operator system interface, whose job is two fold: taking data and present it to the operator, and take input from the operator and act correspondingly.

The base station subsystem will be responsible for receiving telemetry data from the vehicle and present the information to the operator through a graphic user interface(GUI). The Base station subsystem will also take input from the operator, package these commands and send them to the vehicle. The GUI will provide the operator with a simplified and high level method to control the vehicle, the operator will be able to issue command such as “start the vehicle”, “stop the vehicle”, “pause the vehicle”, etc. Should the operator wish to have more direct control over the vehicle, the base station also provides the operator with the option to control the vehicle with a controller, the base station will listen to controller input and package the controller input into commands and send it to the vehicle.

The Vehicle subsystem have three major responsibilities: driving the vehicle autonomously toward the beacon, receiving commands from the base station and act accordingly, and send telemetry data back to the base station.The autonomous control aspect of the vehicle will be the brain of the vehicle, this subsystem will be responsible for making the decision as to what the vehicle will do without the commands from the operator. While the vehicle is operating autonomously, telemetry data such as speed, direction, position, battery life, and operating status are being sent to the base station; the vehicle is also listening to any operator commands that might be sent over from the base station.

The autonomous portion of the system contains two major modules, the navigation and the object avoidance. The Navigation portion of the vehicle controller will be responsible of determining where the vehicle should go, or in another words, setting the goal for the vehicle to pursuit. The object avoidance algorithm will be the ‘person’ behind the steering wheel of the vehicle, who will control the vehicle to drive toward the goal set by the navigation module, while sensing the environment through a combinations of infrared(IR) sensors and ultrasonic sensors to avoid obstacles. The object avoidance algorithm will have controls over the motor through the motor controller, which will interface between motor control instructions and actual power delivery to the motor.

3.2.14 Vehicle Base Station Communication

The vehicle and the base station will communicate with each other via radio signals. Commands and telemetry data will be communicated over the same channel with a pair of wireless transmitter and receiver that will be placed on both the base station and the vehicle. The FPV video feed will be transmitted in its own channel with a standalone pair of transmitter and receiver on the vehicle and base station respectively. Communication of commands and data will follow a communication protocol. Data and commands will be split into packages of fixed size and their metadata will be located in the header of the package allowing the recipient to correctly identify the data.

The operator commands will have a higher priority than the autonomous decisions that the vehicle has made on its own. To ensure that operator will have full control of the vehicle while using the controller, the vehicle will only listen to the operator's command once an operator command is received, decisions that are made autonomously will be disregarded. The vehicle will resume following its autonomous decisions only if the vehicle receives the command from the operator which instructs it to do so.

3.2.15 Graphical User Interface (GUI)

The base station will implement a simple GUI through which the operator communicates with the vehicle. The GUI will offer buttons, which allows the operator to easily issue high level commands to the vehicle. These buttons will be linked to function call underneath the cover, which is the actual action that would issue the commands. The GUI will also feature a table of telemetry data which will be updated using data received from vehicle. There will be a console that outputs other system monitoring information about the base station and the vehicle. This system monitoring info will be read by the operator for a deeper understanding of the current status of the vehicle for the purpose of debugging. This information would include if the vehicle is running live, the status of the communication between the vehicle and the base station, and some of the actions that the base station and the vehicle performed under the cover. The GUI will feature a window that displays the route travelled by the vehicle mapped onto a reference map. The FPV video feed will be displayed on a stand alone display. Figure 17 below shows a concept of the above described GUI.

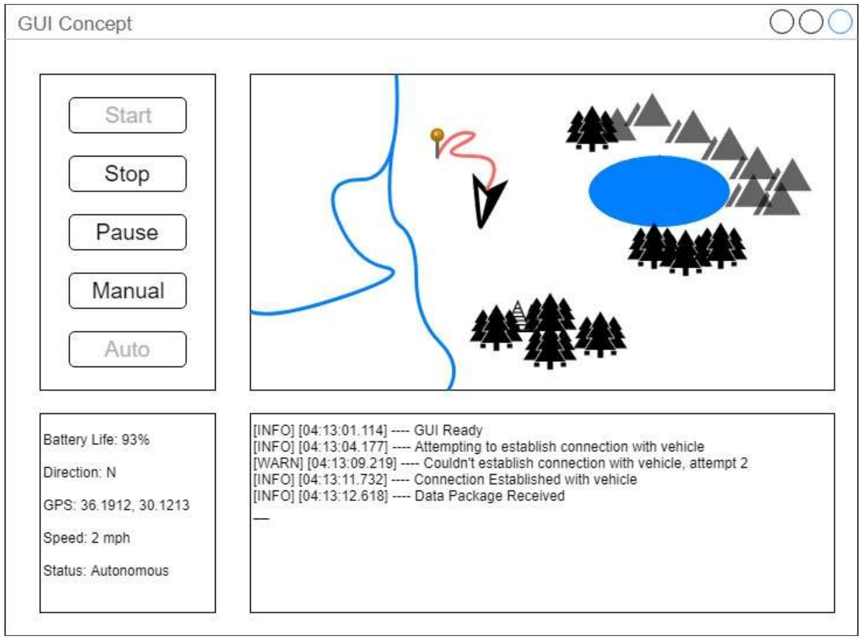


Figure 17: GUI Concept

3.2.16 Obstacle Avoidance Algorithm

The ground-based design will have infrared sensors covering 180 degrees of the front of the rover. It is important to note that the detection range of the sensors can be adjusted. This will consist of roughly 5 infrared sensors with a range of about 35 degrees each. The sensor reading can then be taken and represented as a graph which can be used to determine our best path.

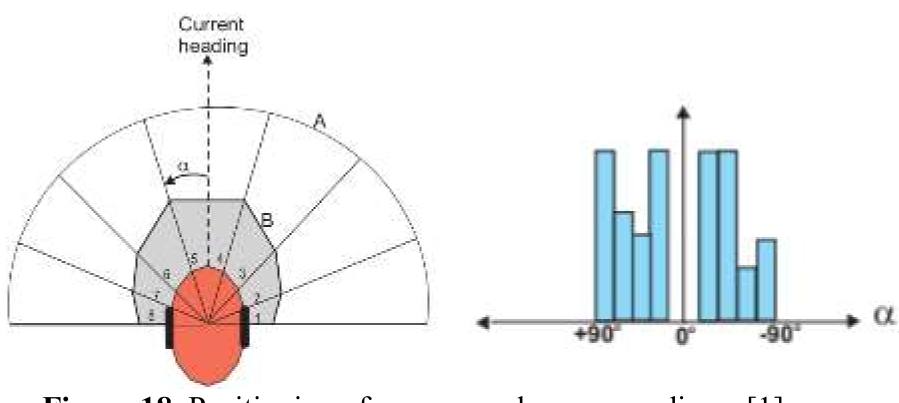


Figure 18. Positioning of sensors and sensor readings. [1]

Figure 18 above shows an snapshot of the obstacle avoidance algorithm.It shows that there are detected obstacles to the left and right of the vehicle. The ideal choice is to keep moving forward (in the direction of 0 degrees).

The algorithm will have the rover initially move toward the direction of the beacon. If an obstacle is detected by the sensors, the rover will head into the direction with the least obstacles until the path to the beacon is clear again. Figure 19 below will illustrate a certain scenario.

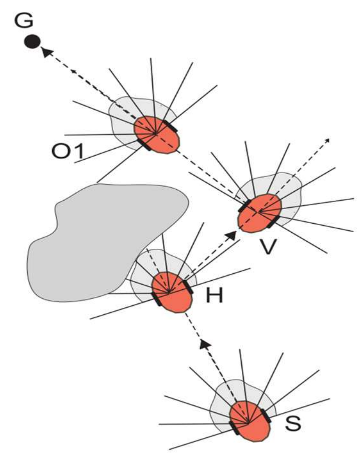


Figure 19: Illustration of avoidance process. [1]

Figure 20 below shows the flowchart for the obstacle avoidance algorithm the rover will obey.

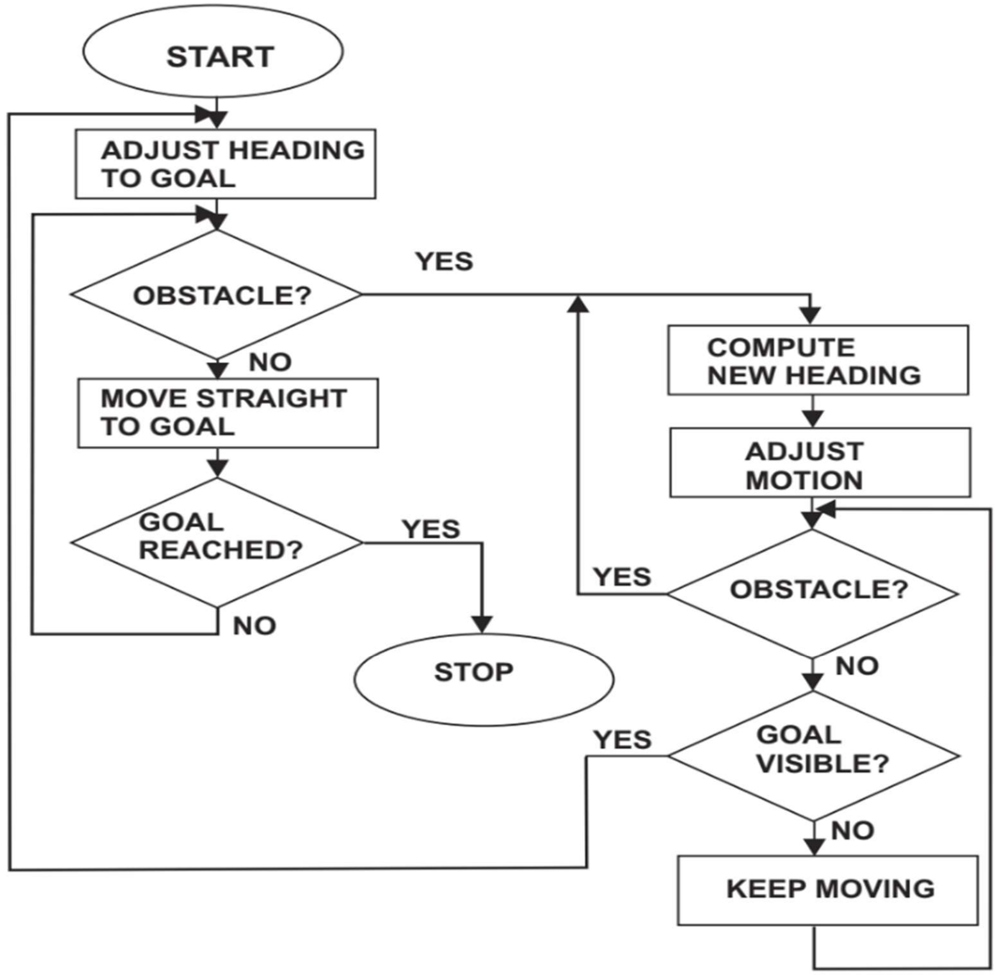


Figure 20: Flowchart for obstacle avoidance algorithm

3.2.17 System Integration

To understand and record how all components will communicate with each other, interface matrices were used. Each matrix highlights a major subsystem of the ABLS system. The remaining subsystems that are not highlighted are marked with an 'X' if if communicates with the highlighted subsystem. Also included in the interface matrices is a message column stating what is being communicated between the two subsystems. Using these matrices creates a simpler system to keep record of all communications between the various components of the design.

3.3 Use

The rover will be able to autonomously navigate through wooded areas to find hikers who are lost or injured. The hiker will carry a beacon for the rover to locate. Once the rover receives the signal from the beacon, it will attempt to locate the direction of the beacon, and then will navigate to the beacon. Telemetry data such as GPS coordinates and distance traveled will be relayed to an operator at a base station.

4. Implementation

This section is not applicable in this document because the prototype is not complete.

5. Finances

For the ABLS project a funding limit was given of $750. After deciding on the components that were going to be used, the total budgeted amount for the project is $635, this amount gives a minimum reserve amount of $115. The reserve amount will allow some margin for replacement parts, unforseen unknowns, or any other parts that may needed in the future. Table 1 shows the finances for this project.

Table 1. Project Finances

Items	Original Estimate $	Actuals to Date $	Estimate to Completion $	Estimate at Completion $
Dagu Rover 4WD	175	0	175	175
FPV Camera	35	0	35	35
RF Tx and Rx 315Mhz	5	0	5	5
RF Tx and Rx 433Mhz	15	0	15	15
Video Tx and Rx	40	0	40	40
Raspberry Pi	40	0	40	40
GPS	40	0	40	40
USB to TTL Serial Cable	10	0	10	10
Beacon Antenna	10	0	10	10
Motor Controller	100	0	100	100
Batteries	20	0	20	20
Battery Charger	25	0	25	25
Sensors	20	0	20	20
Misc.	100	0	100	100
Subtotal:	**635**	**0**	**635**	**635**
Minimum Reserve:	115		Current Reserve:	115
Funding Limit:	750			750

6. Evaluation and Plans

Thus far the ABLS team has been very successful in the problem definition and design phases. Numerous exercises were employed to help the team get a firm grasp on the problem at hand, and create a solid set of requirements from which to design a viable solution. The selected design also features many concepts which various team members have prior experience with. Obstacle avoidance was a major component of the Computer Engineering Junior Design Project. The Electrical Engineering Sophomore and Junior labs also focused on rover motor control, albeit not with ROS (robot operating system). The main weaknesses of the team are a lacking of experience in two key areas: radio communication, and ROS. In an effort to combat this, team members are currently following online tutorials and classes in an effort to gain some experience.

The next steps in the project will be to begin building and testing the communications system, as well as the beacon. The electrical engineers will be focused on developing communications between two Raspberry Pis while the computer engineers will look into the communication between two PCs using ROS. The mechanical engineer will be designing mounts to hold the sensors and hardware to the robot chassis to give a full 180 degree field of vision for obstacles. These tasks should be complete before the spring semester begins.

Upon the start of the spring semester, the various components will slowly be built from the communication system outwards. First the rover chassis will be added and manual control tests will be run. Next the sensors will be added and automated control testing will commence. In parallel with these tests, a separate setup will be used to test the integration of components with the communication subsystem. Obtaining and transmitting both telemetry data and video feed can be tested with the setup, before they are integrated onto the physical rover. The goal is to have completed the subsystem and component testing by the end of February so as to begin full testing in March, or after spring break at the latest.

Full system testing will begin by spring break. This will begin in an open room with obstacles being progressively added. Once the navigation is working consistently well and the rover is able to locate the beacon, testing will proceed outdoors, weather permitting, to an open field. Similar to the indoor testing, the tests will gradually feature more obstacles, specifically trees, fallen debris, and puddles. Outdoor testing will begin by April. The testing will be done by a subset of the team, depending on schedules and availabilities. However one of the two radio licensed operators must always be present for testing. Figure 21 shows the Gantt Chart for this project.

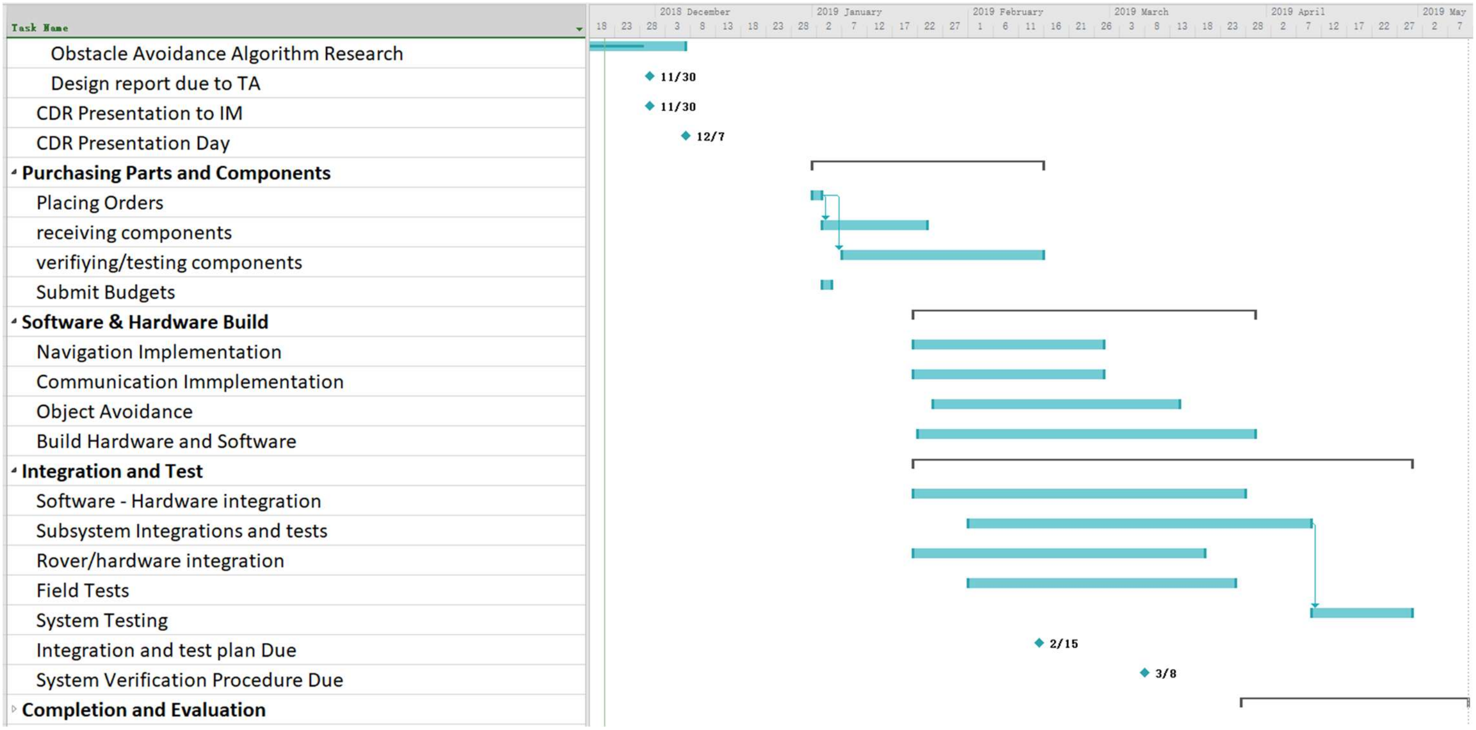


Figure 21: Project Gantt Chart

7. References

[1] I. Susnea, V. Minzu, G.Vasiliu, "Simple, Real-Time Obstacle Avoidance Algorithm for Mobile Robots," Recent Advances in Computational Intelligence, Man-Machine Systems and Cybernetics, Romania, University "Dunarea de Jos", 2009.

[2] Mechatronics Art. (2016, May 22). Dagu Wild Thumper 4WD Chassis. [Online]. Available: https://grabcad.com/library/dagu-wild-thumper-4wd-chassis-1

[3] Maker.io. (2018). Meet the New Raspberry Pi 3 Model B+. [online] Available: https://www.digikey.com/en/maker/blogs/2018/meet-the-new-raspberry-pi-3-model-b-plus [Accessed 23 Nov. 2018].

[4] Industries, A. (2018). *Adafruit 9-DOF Accel/Mag/Gyro+Temp Breakout Board - LSM9DS1*. [online] Adafruit.com. Available at: https://www.adafruit.com/product/3387 [Accessed 23 Nov. 2018].

[5] Runcam.com. (2018). *RunCam Racer User Manual*. [online] Available at: https://www.runcam.com/download/Racer-manual.pdf [Accessed 23 Nov. 2018].

[6] Electroschematics.com. (2016). *Build Your Own Radio Beacons.* [online] Available at: https://www.electroschematics.com/12634/build-your-own-radio-beacons/ [Accessed 23 Nov. 2018]

Appendix A: Design Makeability Checklist

Watson School Design Makeability Review

Design Makeability Checklist

- Project
  - Responsible organization Lockheed Martin
  - Project ID WCP09
- Conceptual Design Briefing ✓
- DFx
  - Assembly ✓
  - Test ✓
  - Safety x – not considered at this point
- Electrical
  - Circuit schematics ✓ Have for beacon
  - Circuit layouts ✓ Have connections for battery power and Raspberry Pi connections
  - Proto board plans ✓ Have for beacon
  - PCB designs if applicable N/A
  - Device wiring ✓ → shown in system diagrams will be shielded by plastic enclosure from water
  - Control panel wiring ✓
  - Color codes ✓
  - Connections/Terminals ✓ → Rasp Pi layed out
- Manufacturing Drawings
  - Material selection ✓ → plastic resin
  - Layout ✓
    - Title block ✓
      - Project Name x
      - Project ID x
      - Primary contact email x

      will add to drawing
    - Dimensioning ✓ – need to measure
    - Tolerances ✓ – need to double check manufacturing process
    - Assembly drawing n/a – parts are purchased
- Bill of Materials (BOM)
  - Parts ✓ – Acquiring parts
  - Materials (finished, chemicals, etc.) ✓
  - Estimated costs ✓ – tabulated
  - Vendors ✓ – tabulated
- Software
  - Language, libraries selected ✓
  - Development environment ✓
  - Modules and their interfaces defined ✓
  - Architecture diagram(s) with data and control flow ✓

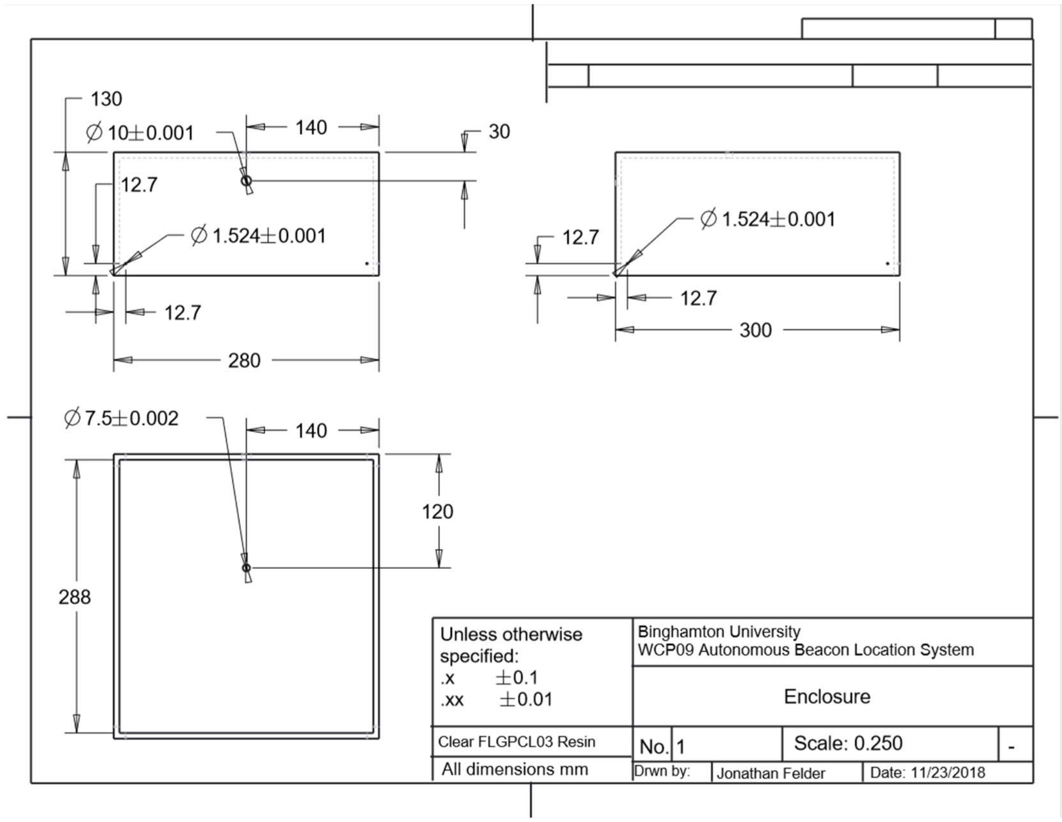
  C++, Java (Java Fx)
- Project Timeline
  - Tasks by subsystem, functions, etc. ✓
  - Integration, test ✓
  - Installation (if applicable) ✓
  - Assigned team members ✓
  - Due dates ✓
  - Dependencies ✓

  in Gantt Chart

Watson School Design Makeability Review

- Project
  - Responsible organization Locthead Martin
  - Project ID WCP09
- Makeability review completed, approved
  - Approved by Canzler
  - Signed
  - Date 11/30/18

Appendix B: CAD Drawings



Appendix C: Bill of Material

Part Desc.	Place of Purchase	Part Name:	Part #	Quantity:	Price:	Total Price:
Rover Chassis	pololu.com	Dagu Wild Thumper 4WD, 75:1	1566	1	$ 174.95	$ 174.95
Batteries	Hobbyking.com	Turnigy Stick Pack Sub-C 3000mAh 7.2v NiMH High	9440000002	2	$ 9.00	$ 18.00
Battery Charger	"	6-12V NiMh/NiCd smart charger	RB-Ten-02	1	$ 21.95	$ 21.95
Motors	N/A	comes with dagu				$ -
Tires	N/A	comes with dagu				$ -
ESC	N/A	comes with TReX motor controller				$ -
Radio Transmitter/Receiver(Beacon)	amazon.com	HiLetgo 315Mhz RF Transmitter and Receiver		1	$ 4.69	$ 4.69
Radio Transmitter/Receiver(Vehicle Co	amazon.com	WINGONEER 433Mhz RF Transmitter and Receiver		2	$ 5.99	$ 11.98
FPV Camera	amazon.com	Runcam Racer Micro		1	$ 34.99	$ 34.99
FPV transmitter/receiver	banggood.com	1.2G TX1000 1W 1000mW 8CH Transmitter	1191192	1	$ 39.99	$ 39.99
Raspberry pi cpu	amazon.com	Raspberry Pi 3b+		1	$ 39.00	$ 39.00
GPS	adafruit.com	Adafruit Ultimate GPS Breakout	746	1	$ 39.95	$ 39.95
USB to TTL	amazon.com	HiLetgo USB to TTL Serial Cable		1	$ 5.19	$ 5.19
Antenna for Beacon	amazon.com	315Mhz-470Mhz Omnidirectional Antenna		1	$ 8.98	$ 8.98
Motor Controller	pololu.com	Pololu TReX Dual Motor Controller DMC01	777	1	$ 99.95	$ 99.95
IR Sensor	aliexpress.com	Infrared Obstacle Avoidance Sensor Module		5	$ 0.38	$ 1.90
Ultrasonic Sensors	digikey.com	SparkFun Electronics Ultrasonic Sensor HC-SR04	SEN-13959	4	$ 3.95	$ 15.80
steel screw	mcmaster.com	18-8 Stainless Steel Socket Head Screw	92196A051	1	$ 6.73	$ 6.73
					Total Amount:	$ 524.05

Appendix D: Interface Matrices

Communication	Navigation	Base Station	Power	Camera/Video	Drive	Messages	Description	Type/Unit(?)	Value Range
	X					Motor Override Command	According to the Motor Controller Instruction set, the motor command should be a byte in length.		
		X				Status Message	Status update of the rover, expected to include its operating mod, Vehicle status, etc		
		X				Operation Mode	Flag that indicates the mode of operation that the vehicle is currently in. Autonomous and Manual	Categorical	Autonomous Manual
		X				Vehicle status	a few bits that indicates the current status of the vehicle.	Categorical	Operating Immobilized Awaiting Instructions Suspended
		X				Battery status	Indicates the estimated battery life	Categorical	High, Medium, Low, Very Low
		X				Vehicle Position	A GPS Coordinates indicates the location of the vehicle		
		X				Orientation	Orientation that the vehicle is facing	Categorical	N, NW, W, etc
		X				Beacon signal Strength	Signal Strength of the Beacon	Categorical	
		X				Beacon Direction	Direction of the beacon relative to our vehicle	Categorical	to the east/ north east/south etc
		X			X	Vehicle Speed	The speed of the vehicle based on the GPS postiion	Numeric	0 - 2 mph

Communication	Navigation	Base Station	Power	Camera/Video	Drive	Messages	Description	Type/Unit(?)	Value Range
X					X	Motor Command	According to the Motor Controller Instruction set, the motor command should be one byte in length.		
X		X				Beacon signal Strength	Signal Strength of the Beacon	Categorical	
X		X				Beacon Direction	Direction of the beacon relative to our vehicle	Categorical	to the east/ north east/south etc
X					X	Motor Speed	As specified in Motor Controller Instruction set		

Communication	Navigation	Base Station	Power	Camera/Video	Drive	Messages	Description	Type/Unit	Value Range
X	X				X	Commands	High level commands sent to the vehicle controller	Categorical	Start Stop Autopilot mode Manual mode Shutdown
X	X				X	Controller input, throttle	Input from the Operator's controller indicating how fast the motor would run	Numeric	0.0 - 1.0
X	X				X	Controller input, Turns	Input from the Operator's controller indicating directions the vehicle will turn to	(direction of turning, duration of turn)	

Communication	Navigation	Base Station	Power	Camera/Video	Drive	Messages	Description	Unit	Value Range
X	X					Battery Life	Percentage battery life remaining on the rover	Percentage	0%-100%
X	X			X	X	Voltage	Voltage needed to power devices	Volts	0 - 7.2

Communication	Navigation	Base Station	Power	Camera/Video	Drive	Messages	Description	Type/Unit	Value Range
		X				Video feed	FPV video feed	Analog	

Communication	Navigation	Base Station	Power	Camera/Video	Drive	Messages	Description	Type/Unit	Value Range
X	X					Motor Command	According to the Motor Controller Instruction set, the drive system will return some specified information		