Competition Directive: Design assistive Rovers for a manned mission to Mars.

There are two main competitions that we will be competing in fro the 2015 Rover Season: the University Rover Challenge and The European Rover Challenge. The University Rover Challenge (URC) takes place in Hanksville, Utah and has been the competition we have been competing in for the past few years. The European Rover Challenge will be held in Poland and will be a new competition for us this season. The main goal of the competitions are the same. However, there are some tasks that are more important in one competition than the other. The following will discuss how these changes affect the Telemetry and Controls Team for each competition.

University Rover Challenge Rule Changes

"Due to the overwhelming growth of the competition in recent years, the University Rover Challenge has introduced a competitive down selection process as one of the biggest rule changes in 2015. Teams must declare their intent to compete by December 5, 2014, submit a Team Proposal no later than January 23, 2015, and go through a Critical Design Review by March 14, 2015.

The Critical Design Review (CDR) will take a much more comprehensive look at the team's overall system design, and progress-to-date of the final system.

The Presentation Task will now be a video submitted mid-May."

Other Design Requirement Updates:

- *Extremely difficult terrain may include continuous steps, slopes even steeper than 60 degrees, and even vertical faces. If a rover cannot manage an obstacle, or it is deemed too hazardous, teams will be able to skip it and try their luck at another obstacle.*

- *Rovers may now be repaired in the field during interventions.*

- *The Sample Return Task has a new emphasis on geology, not strictly the biological focus of past years.*

- *Context of the sample location must be provided with respect to the local geography and other features of interest.*

- *The wide-angle panorama is not required to be a 360 degree horizon image, and can be a single wide-angle shot or stitched as teams see fit.*

- *The requested scale indication must merely provide an order-of-magnitude level of spatial awareness.*

- *As a coarse backup navigation, passive magnetometers will be allowed.*

- *Anything deployed by the rover should be collected before returning to the start gate.*

- *If deployed objects are not collected you will forfeit the points for returning the rover to the start gate on time.*

- *Antenna height is now limited to 3m*

European Rover Challenge Rule Changes

The European Rover Challenge is the new competition that we will be competing in this year. Therefore, I will discuss the criteria that differ from the criteria of the URC.

Design Requirements:

- *Accurate GPS: The ERC relies heavily on the location being given by GPS*
- *Autonomy: For extra points, the rover will need to move from one location to the other just being given a GPS location*
- *Publishing data to Internet: For extra points, the rover should upload the data it has collected to the internet (i.e. email or website)*
-

The document that follows contains the preliminary designs and future plans.

Motherboard Squad

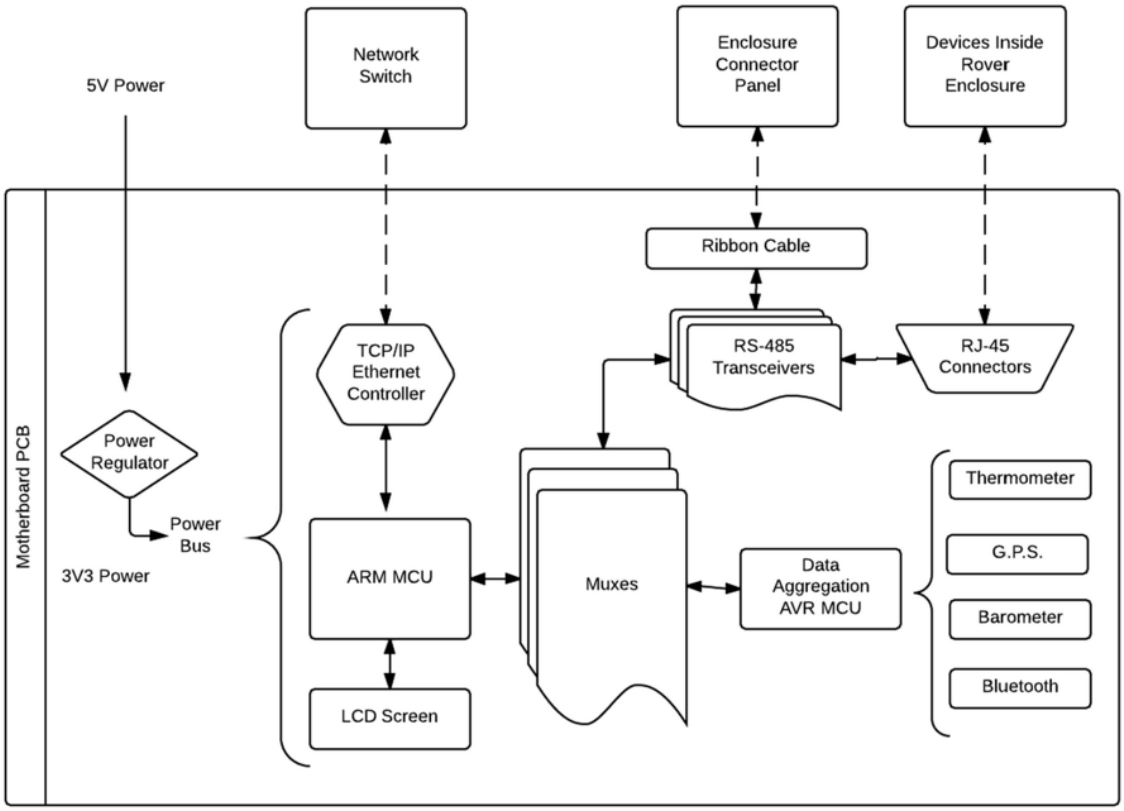
- Design requirements
  - Functional requirements
    - To receive control commands from the base station, interpret them, and issue the appropriate control signals to the rest of the rover.
    - To aggregate and transmit data from the rover subsystems to the base station.
    - To house several primary sensors including a G.P.S. module.
    - To control the rover in an autonomous mode.
  - Design requirements (i.e. hard requirements, such as size/weight constraints, voltages, data formats, etc.)
    - Physical Requirement
      - 6 in. X 6 in. X 2 in.
    - Power Input
      - 5 volts
    - Data connection to Base Station

        - Data connection to other on Rover electronics
- Points of interface
    - Physical connection to rover
        - No. 2 screw standoffs
    - Power Input
        - 5 volts
    - Data Connection to Base Station
        - Ethernet connection to the network switch
        - TCP Connection to base station
        - Custom data format for data layer
    - Data Connection to onboard rover devices
        - Power Board
            - RJ-45 Connector
            - RS-485 Electrical Standard
            - Custom Data Format
        - All Other Rover Devices
            - Ribbon cable connection to patch panel which contains RJ-45 Connectors that go to other devices
            - RS-485 Electrical Standard
            - Custom Data Format
- Issues in last year's system
    - Poor connector placement
    - No timeout on serial connections
    - No ability to determine heading, just G.P.S. position
    - Drivers for TCP/IP processor were never written
    - Ferrite bead was very small
    - GPS Diode was very small

2014 Design

The design described in this section applies to the 2014 Rover. It is important to understand the 2014 design for context of the 2015 design.

Hardware Design



At the center of the motherboard, sits an A.R.M. M4 micro-controller that acts as the primary processor for the rover. In early iterations of the motherboard, this processor was soldered directly to the motherboard. However, the pitch of the processor pins proved to be too small for the team to reliably manufacture. To circumvent this problem, the team chose to utilize the Texas Instrument LM4F120 LaunchPad Evaluation Board as the main processor. This development board is approximately 13 USD and has female header pins on the bottom of the board. This allowed the team to place easy to solder male header pins on the printed circuit board and simply insert the entire Launchpad board onto the motherboard. This method had the added advantage, of allowing the team to remove the Launchpad in order to program, flash and debug the main processor without requiring the rest of the motherboard. The circuit board layout is shown in figure 4 below.

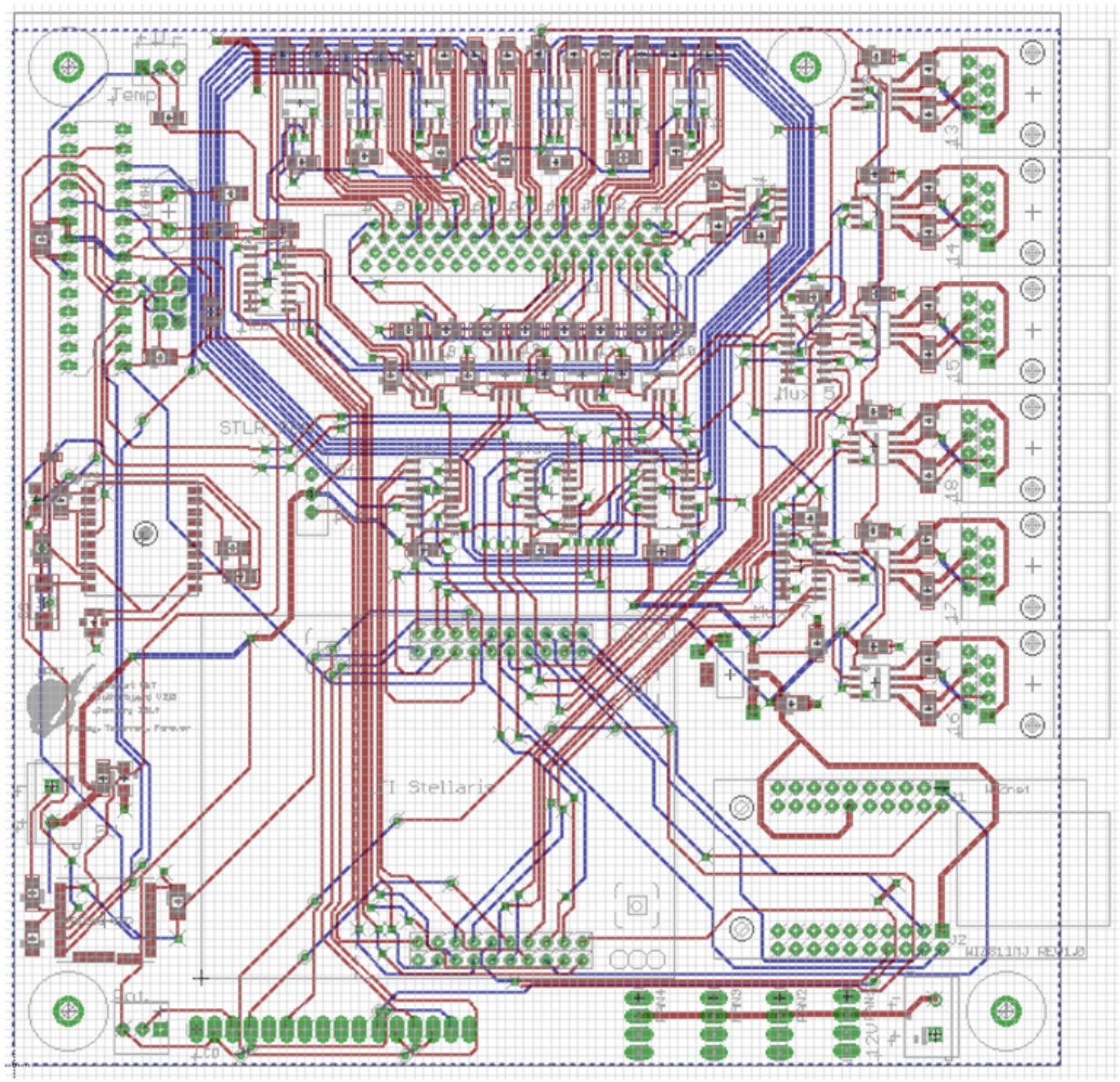


Figure 4. - Motherboard P.C.B. Layout

The link to the base station comes from a TCP / IP controller chip, the Wiznet 5100. This chip was mounted in the same way as the Launchpad, using the breakout headers since the pitch was also very small. All communication with the base station came through this path, flowing first through this chip, then through a network switch on board the rover and finally out of the 2.4 GHz radio.

A secondary micro-controller on the motherboard was designed to manage and aggregate data from several sensors on the motherboard, thus relieving the primary processor of the low level collection and management tasks. This processor was an Atmel ATmega328P. The team used

this processor for all onboard computing tasks, with the exception of the main TI processor. This allowed the team to make use of the easy to use and learn libraries provided as part of the Arduino environment. The common processor also allowed the team to minimize the number of spare parts required.

This auxiliary processor is directly connected to a G.P.S. receiver, a temperature and humidity sensors, and a Bluetooth transceiver. The team made use of both the G.P.S. receiver and the temperature / humidity sensor. However, due to time constraints, the team did not implement software for the Bluetooth transceiver.

All other electronics on the rover connect to the motherboard via RS-485 connections. This differential pair signal standard works well since it is full duplex and can be used with off the shelf ethernet cable and RJ-45 jacks. The motherboard has 20 RS-485 transceivers: 6 are connected to RJ-45 jacks mounted directly on the motherboard, 12 are connected to a ribbon cable that connects to a patch panel mounted on the exterior wall of the rover's electronics housing. The main processor on the rover contains only 6 accessible hardware universal asynchronous receiver / transmitter, which the rover was designed with 18 devices in mind. To solve this issue, a series of dual channel multiplexers are connected between the main processor and the RS-485 transceivers.

Software Design

The team was also responsible for developing all of the software for the motherboard. This included both the main processor and the sensor co-processor. All software was written in either C or C++ and is discussed further below.

Main Processor Software

In order to create a responsive system, the team decided that they would need to use a real time operating system (RTOS) that could handle scheduling and prioritizing tasks to run on the single core of the processor. Texas Instruments provides a free RTOS called SYS/BIOS that already has many drivers written for the the LM4F120 processor. Thus, the team chose to use SYS/BIOS to schedule our tasks on the processor.

The team defined two tasks, a command processing task and a telemetry aggregation and transmission task. The command tasks was given a higher priority than the telemetry task and would preempt the telemetry task if a command was received. The command task had to first decode the JSON message. It then changed the appropriate multiplexer to the correct RS-485 transceiver and sent the message in the format described below. When no commands were in the command queue, the telemetry task cycled through all devices connected, polling for the latest telemetry values. JSON messages were then constructed for each telemetry parameter and sent into the TCP connection to the base station.

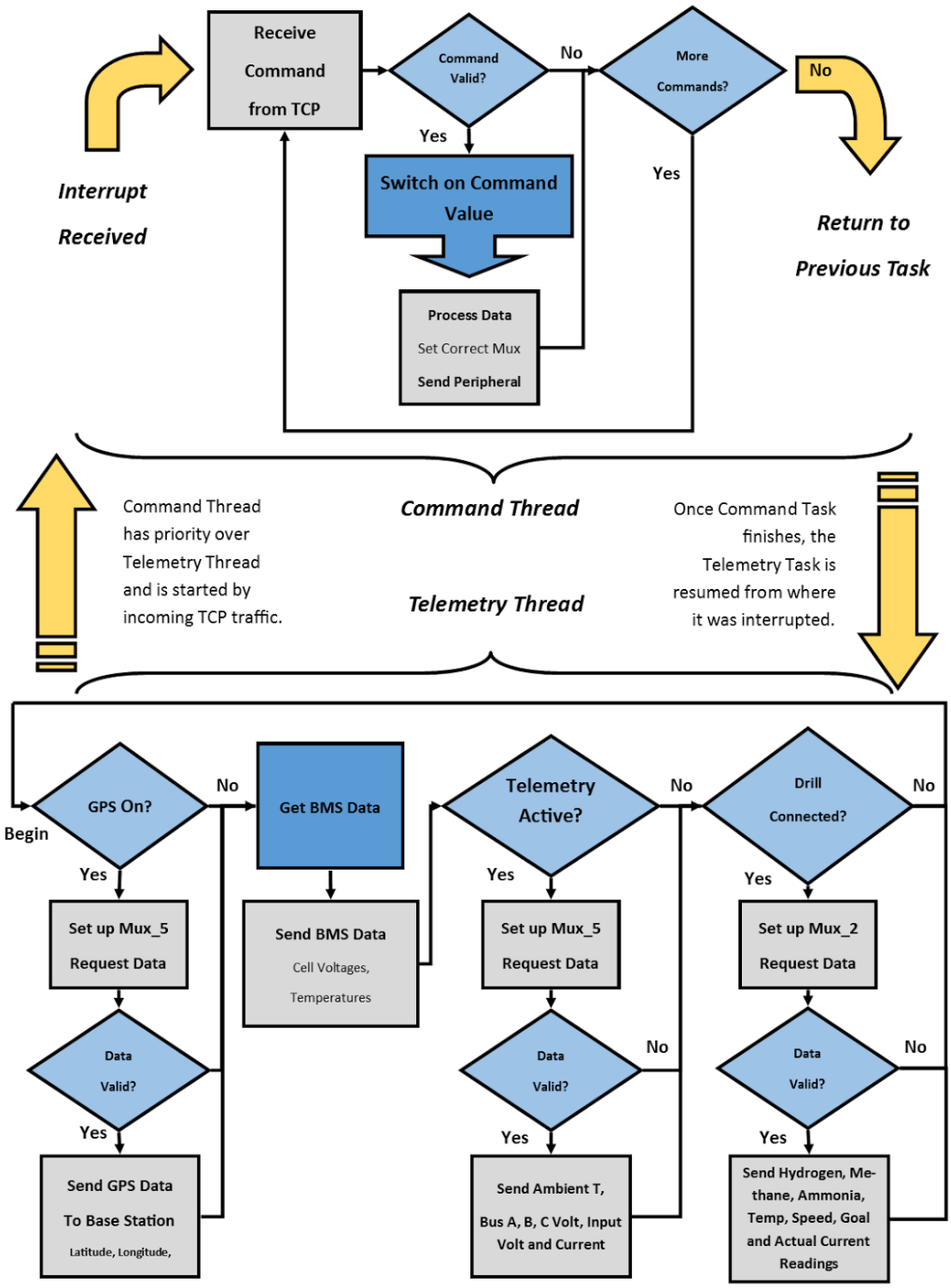


Figure 5. - Main Processor Thread Diagram

Sensor Co-Processor

The sensor co-processor software consisted of a single, infinite loop that would execute the following sequence in order:

1. Poll G.P.S. receiver for position
2. Poll temperature sensor for temperature & humidity
3. Send data to motherboard

Base Station Message Format

In order to support legacy base station code, the javascript object notation (JSON) was required for commands to the rover and for telemetry sent to the base station. Telemetry and commands followed a common format. Each message contained an identification field and a value field as shown in Figure 5. The identification field was a four digit number that served as the key for each message. The value was either the value of the telemetry parameter or the value for the command.

```
{
    “MessageID” : “1002”,
    “Value” : “50”
}
```

Figure 6. - Example Base Station Message

Inter-Rover Message Format

Communication internal to the rover was done by serializing standard c structs. Each message contained the following components shown in Figure 6, where each row represents an eight bit byte. The two start bytes in the beginning are to ensure that the message is correctly byte aligned. The next byte, the size, indicates the number of payload bytes. The payload is the serialized struct with any padding bytes removed. Finally, a checksum is added to ensure that the data arrives intact. The software on the receiving side, will discard any messages that fail the checksum.

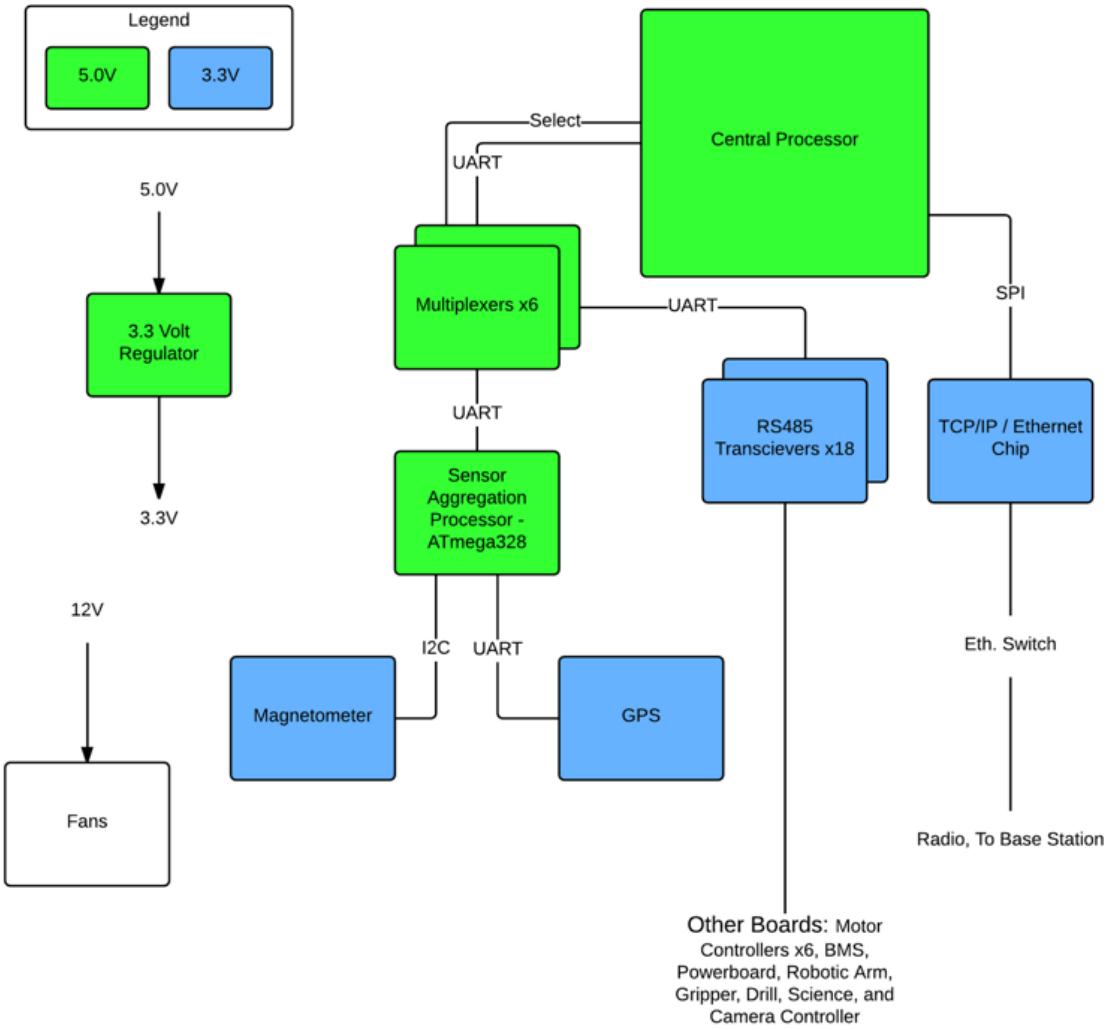
Start Byte 1
Start Byte 2
Size of Payload
Payload Byte 1
Payload Byte 2
Payload Byte 3
…..
Payload Byte Size
Checksum

Figure 7. - Inter-rover Message Format

2015 Design

While Phoenix achieved the second place award in the competition last year, the MST Rover design contained many contingencies that proved to be extraneous, while encountering a few bottlenecks in a few technical aspects of the competition. Honing in on the success and failures of the 2014 competition showing, the 2015 motherboard will be upgraded, streamlined, and will increase the Rover's functionality and reliability, all at a reduced footprint.

Motherboard Hardware Model



This section contains planned improvements for the 2015 Rover

Hardware Design

New Primary Processor

The team is proposing, and evaluating, an upgrade from last year's M3 cortex (LM4F120 LaunchPad,) to this year's M4 cortex integrated development board, a more recent offering from Texas Instruments. The M4 Texas Instruments Launchpad Tiva C Connected (EK-TM4C1294XL) merges the TCP/IP/Ethernet chip and ARM MCU on to a single board, and also allows the team to physically remove the Microcontroller in order to iteratively program, flash, and debug the main processor, without requiring any physical attachment to the rest of the board. As such, the team will be focusing on iterative

development and design tools, in order to refine last year's successes, while mitigating last year's failures, in a progressive enhancement development cycle.

Reduced Circuit Board Size

The team is focusing on removing non-essential components from the 2014 Motherboard design. This includes removing all but one of the onboard RJ-45 jacks since only one is required for the 2015 Motherboard. This change allows the motherboard team to remove two multiplexers and five rs-485 transceivers.

Larger Part Selection

Some of the Motherboard components were incredibly small and difficult to reliably hand solder. The team will ensure that all parts are selected and placed for ease of manufacturing.

Addition of Magnetometer

In order to provide heading data to the base station, a magnetometer has been added to the updated Motherboard Design.

LCD Screen Removal

The LCD screen has been removed from the 2015 Motherboard. It occupied a large amount of space on the circuit board and did not provide much added value.

Better Connector Placement

In addition to the reduction in size of the Motherboard, the placement of connectors will be optimized for ease of cable routing and placement.

Software Design

Inter-Rover Communication

The 2014 Rover's internal communication protocol worked well when all external devices were functioning correctly, but suffered some reliability issues when devices behaved unexpectedly or stopped responding. For the 2015 rover, a minor redesign of the software layer to improve robustness in these situations and improve communication efficiency on the peripheral side is planned.

Interrupts

The team plans on utilizing interrupts to service all serial communication in and out of the Motherboard processor. The interrupts will dump data into large software buffers that will then be processed by the threads.

Threading

The 2014 Rover's software was divided into two main threads, each performing multiple tasks. In order to implement the autonomy system for the Rover, this thread design may need to be changed to allow for the new capabilities. The team is evaluating a variety of threading models that will provide the best system reliability and performance.

Base Station communication

With last years' time constraints, the JSON web service protocol provided a 'quick to market' implementation and components in the JSON protocol were successfully utilized. JSON message structures are messages of simple web formatted text strings, typically utilized by high level RESTful web services. As full strings containing text character representations of numeric values, parsing the JSON constructs in and out of a text-based representations does have a measurable processing cost associated with it.

While these messages are ideal for web service endpoints, they used a disproportion amount of the embedded system resources on the Rover, as Phoenix locally decoded the text strings on the embedded real time processors inherently designed for serialized c structures and without support layers for uplifted web components in mind.

The Base Station team and the Motherboard team are jointly assessing custom protocol message formats, in the attempt to shift this overhead off the local Real Time Management micro systems in the Rover's embedded layer, and onto the power of the Base Stations ULSI parallel PC chipset and managed code runtime layer.

During competition, Phoenix behaved unexpectedly or stopped responding at times. For the 2015 rover, a minor redesign of the software layer on the peripheral side is planned that should improve the robustness and communication efficiency and mitigate these situations.

Watchdog Resets

In case the Rover the becomes unresponsive, implementation of a watchdog timer reset will allow it to reset itself without the Team requiring to take a touch penalty. This could save points at competition and may be a greater necessity in case the autonomy system prevents the rover from responding to base station inputs.

Automation

The Motherboard squad has the goal of adding a level of automation to this year's rover. The current goal is give the rover the ability to navigate to a given G.P.S. coordinate autonomously. To start, the rover will not have the ability to avoid obstacles, however, this is not likely to be a problem given the competition is held in the desert where there are few obstacles that need to be avoided. This will be accomplished using the G.P.S. transceiver and magnetometer. It will also need to be accommodated for in the new threading architecture. The team is currently evaluating the accuracy of these sensors and designing a control algorithm.

Base Station Squad

- Design requirements
    - Functional Requirements
        - Communicate with rover via TCP connection
        - Interpret messages from the rover
        - Display telemetry intuitively
        - Provide intuitive controls for rover functions
    - Points of interface (mini-ICD)
        - TCP Connection to Rover that follows proprietary protocol.
    - Issues in last year's system (if applicable)
        - None
    - Description of preliminary design (**complete system** review)
        - RED - v1 - Phoenix
            - TCP server for simple sending/receiving of data and automatic management of the connection
            - JSON Communication Protocol
            - Xbox controller input for controls
            - Modular UI for customizing telemetry display
        - RED - v2 - Next Rover (2015)
            - Full application unit testing (proprietary code only)
            - New networking stack
                - Enabled multiple RED applications running on the same network to communicate and coordinate control of the rover.
            - GPS mapping

            - Offline maps and designation of waypoints
        - Automatic application updates
        - Persistent storage for telemetry data (database)

            - RED will store data locally temporarily and then migrate it to IT-supported database on campus once on-campus internet connectivity is established.
        - User-level configuration of telemetry modules.
            - Changes to telemetry settings on the rover will not need to require modification of the source code. The user will be able to customize modules as they see fit by adding, removing, modifying existing modules.
        - Keeping Xbox controller. Would really like to implement haptic feedback but it requires sensors on the rover.
    - Trade studies of alternative options
        - We could rewrite everything as one of the following
            - GUI - Qt Application (C++ - All platforms?)
            - GUI - Cocoa Application (Objective C or Swift - Mac)
            - GUI - WinForms (C# - Windows)
            - GUI - Java Swing (Java - All platforms *sighs*)
            - Command Line Application (many languages to choose from)
            - ***All are bad ideas for the following reasons:***
                - Team has a lot of time invested learning WPF
                - WPF works great and has met all of our needs in some capacity
    - Reliability and Risk Management
        - Revert to previous build of RED

2014 Design

For the Base Station last year, a C#/.NET Application was created that would handle all the data packaging and communication to the rover. The Application had a modular GUI that would allow for the user to select what data they wish to see. The Base Station was one of the few systems last year that didn't have any major issues. It functioned properly with little errors. One weak point in the Base Station would be the communication protocol. The rover used JSON as its communication with the Base Station which is very easy to use and was readable by the user. However, JSON protocol comes with a lot of overhead which slows down the embedded system on the rover. Other than that the Base Station was completely functional and effective.

2015 Design

In this upcoming year, the Base Station squad is looking to build off the platform that was created last year. Since the Base Station is such a functional system, the squad doesn't really need to go back to the design board on anything. We will just be looking to improve the existing system through minor aesthetic changes and speeding up communication. The Base Station is currently in contact with the Motherboard squad trying to get a new protocol set up. The current goal is to implement a binary based protocol that will have little to no overhead. The other main adjustment we are hoping to implement is the creation of a mobile Base Station. The Mobile

Base Station has multiple functions. A mobile version can be used for PR events for showing off the rover. Another use for this project would be a more modular approach to reading the data during the competition. We are going to create a system that will allow for specific data to be displayed on a tablet. The use for this is apparent in the Science Experiment in which members of our science group could look at the tablet in front of them and only see the data that is important to them (Camera feeds, Experiment data, etc).

Robotic Arm Squad

- Design Requirements
    - Functional Requirements
        - Receive commands from the motherboard
        - Facilitate algorithms up to a six degree of freedom inverse kinematic model
        - Control joints and end effectors
        - Supply voltage to motors
    - Design Requirements
        - Physical Requirement
            - Under 7.2” x 7.5” and fit on the base of the arm
        - Power Input
            - Multiple
                - Pack Voltage
                - 12 Volts
                - Possibly others as needs arise
        - Data connection to motherboard
        - Power and data connections to joint motors
        - Data connection to end effectors
    - Points of interface
        - Physical connection to rover
            - Screw standoffs
        - Data connection to motherboard
            - RS-485
        - Data connection to joints
            - RS-485 or TTL for Dynamixel joints
            - Power connections to joint motors
        - Data connection to end effector
            - Xbee 2.4GHz IEEE 802.15.4
    - Issues with previous system
        - Command overflow on I2C bus
        - Not able to facilitate inverse kinematics
        - No control scheme - incremental control
        - No motor braking

2014 Design

The design of the hardware for last year's robotic arm was made in the master-slave configuration.

Board Level Design

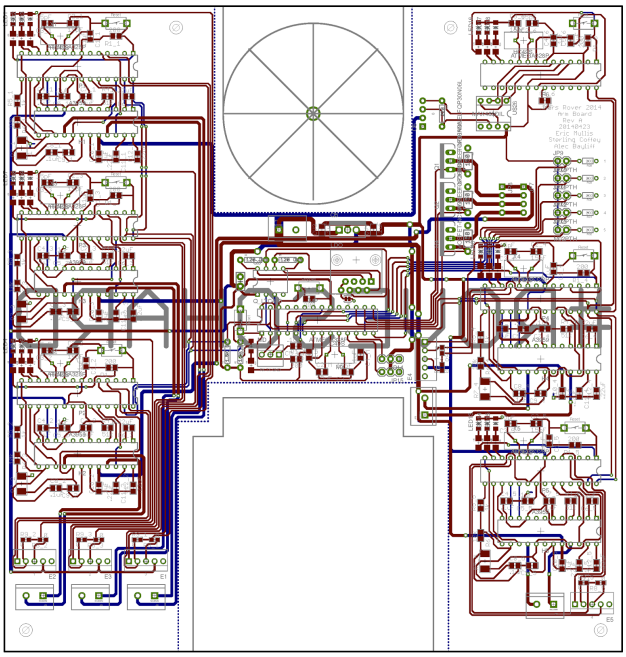
The Motherboard would communicate directly to one of the ATmega328's on the robotic arm board. From there, the main chip would communicate with the five other slave chips through I2C. The I2C bus proved to be difficult because the bus could have command overflow and would cause packets of information to be dropped.

Electromechanical

The electromechanical hardware chosen for last year's rover was basic 12V motors with encoders attached to them. The motors worked well for torque output and the ability to lift and move the robotic arm. However, the encoders and control scheme proved to be more difficult than it should have been.The hardware for the rover last year had no support for a more advanced control scheme without coding magic.

Control Scheme

The only control scheme that could be easily implemented was an incremental control in which a button push at the Base Station will correspond to a certain period of movement for a motor or a servo. The user found the control to be intuitive, but not very precise. Some of the precision was lost due to the fact that the weight of the arm would overcome the static friction of the arm. The robotic arm couldn't hold it's location if any weight was placed at the end of the arm.



2015 Design

The design of the robotic arm this year will focus on control and precision. In order to get the precision required for the competition, the hardware will need to be changed drastically.

Board Level Design

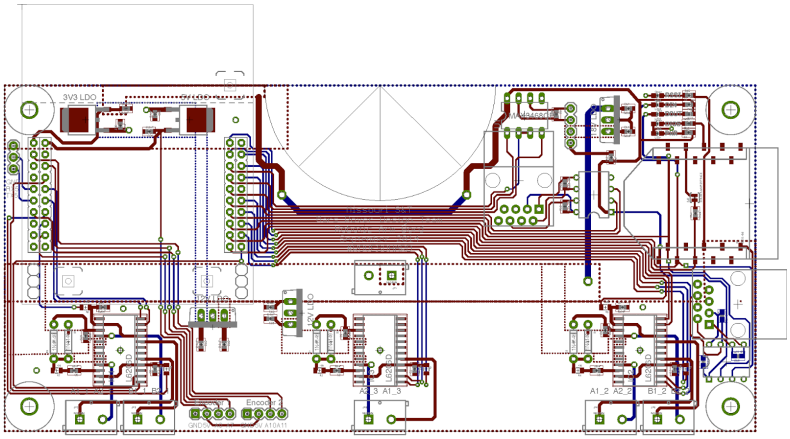
The robotic arm board will get rid of the master-slave configuration completely. The main processor now will be the Texas Instrument LM4F120 LaunchPad Evaluation Board. This was the board used by the Motherboard last year it has a much higher processing speed which will allow for a control scheme such as forward kinematics to be implemented. If more I/O pins are needed for feedback then a co-processor of an ATmega328 will be placed on the board to handle those calculations.

Electromechanical

The next hardware step that will be implemented will be the switching of the motors on the elbow and the wrist with Dynamixel servos. The Dynamixel servos will have the same amount of torque that the previous motors will have except they will be much more controllable. The servos will allowed for a more accurate control scheme with its built in position control and feedback.

Control Scheme

The control scheme that will be implemented this year will be a form of forward kinematics. Initially the robotic arm team will be getting the control scheme back to what we had last year. The reason we will be doing that is because the control is intuitive and can be easily learned and the new servos will allow for more precision. With the new servos, the control scheme from last years will work a lot better because the movement periods will be more precise and braking on the arm will be easier. The more advanced scheme of forward kinematics will be discuss later in this document.



Auxiliary Squad

Cameras

- Design Requirements
    - Functional requirements
        - To supply clear camera feeds
        - To supply low latency camera feeds
        - To supply multiple, quality camera feeds

    - Design requirements
        - Field of View
        - Operation Temperature range
        - Price
        - Sensor Size
        - Focal Length
        - Image Resolution
        - CCD/CMOS Sensor
        - Manual Control
        - Weight

    - Points of interface
        - Mechanical
            - Mounting of the Cameras
            - Range of motion for Gimbal
        - Communication
            - Sending Data Over Radios
            - Space on the Ethernet Switch
        - Power
            - Supply Voltage Bus

    - Issues in last year's system
        - Lack of dynamic range
        - Auto-exposure woes
        - Incorrectly focused cameras.
        - Late Decision Making
        - No Trade Studies Done
        - High Latency
        - Low Image Quality

    - Description of preliminary design
        - This year, we plan on having a minimum of 4 cameras on the rover.

- A mast-mounted, fixed wide-angle camera for the drivers to navigate with
- A gimbal-mounted camera capable of zooming in to scan the horizon/ahead
- A camera mounted on the arm to help direct the arm
- A base camera

- Trade studies of alternative options:
  - Top Choices - Full Trade Study Shown Below
  - DLINK- DCS-5222L
    - Mast Camera
    - Pros:
      - Good Price
      - Weight
      - Field of View
      - Easily Controllable
    - Cons:
      - Lower Operating Temperature
  - Sony - EVI-D100/D100P
    - Gimbal Camera
    - Pros:
      - Turning Speed
      - Zoom Range
      - Wide Range of View
    - Cons:
      - Analog Camera

Cameras we considered:	Sony - EVI-D100/D100P	DLINK- DCS-5222L	BOSCH- MIC IP dynamic 7000 HD	FCBEX1010P	FCB-EX48C
position	gimbal	mast	Base/arm	Base/arm	Base/arm
Ranking	#2	#1	#5	#4	#3
focal length	3.1 to 31mm	3.6 mm	4.3 mm to 129 mm		
Field of View	6.6 to 65 degrees	70 degrees	2.3 to 65 degrees	1.7 to 57.8 degrees	2.8 to 48.0 degrees
CCD/CMOS sensor	CCD	CMOS	CMOS	CCD	CCD

Manual Control	N/A	N/A	Yes		
weight	860 g (1lb 14oz)	340g (0.75lb)	6.7kg(14.7lb)	8.1 oz (230 g)	8.1 oz (230 g)
price		$170			
Operating Temperature	0 to 40C (32 to 104F)		-40C to +60C (-40F to 140F)	0 to 50C (32 to 122F)	0 to 50C (32 to 122F)

- Cost/budget/timeline
  - Cost
    - The cost for the the camera we are considering buying is estimated around $170. We will consider the purchase of additional cameras after full review of the current ones
  - Budget
    - The current budget for the Cameras is $
  - Timeline:
    - We plan on getting our first choice camera(DLINK- DCS-5222L) by October 30th and start testing. At the same time, we will start reassessing the current cameras and compare them to the potential backup plans. We want to have a more thorough review of the current devices and take any possible updates into consideration.
- Reliability and Risk Management backup plans
  - FCB-EX1010P
    - Base/Arm Cameras
    - These two cameras came very close ranking because they are both box cameras with similar specifications.
    - Pros:
      - Stationary Cameras
      - Size Weight
      - Manual Control
    - Cons:
      - Price
      - Analog
  - FCB-EX48C,
    - Gimbal Camera
    - Pros:
      - Operating Temperature

        - Size Weight
        - Control
    - Cons:
        - Price
        - Analog

- BOSCH-MIC IP Dynamic 7000 HD
    - Base/Arm Camera
    - Pros:
        - Operating Temperature
        - Control
        - Field of View
    - Cons:
        - High Voltage
        - Power over Ethernet

Field of View

The horizontal Field of View of a camera/lens system is given by the equation: FoV=2arctan d/2f where “FoV” is the field of view in degrees, “d” is the size of the image sensor, measured horizontally in millimeters and “f” is the focal length. Most camera specs should state at least the focal length and field of view; many will also state the sensor size. We noticed that the actual field of view is just as dependent on the sensor size as it is on the lens.

Operating Temperature

For many cameras the storage temperature is different from the Operation temperature range. Taking into consideration the climate and weather conditions during competition, we agreed that this had to be considered as a deciding factor. After further discussion with more experienced team members who previously attended competition, we came to the conclusion that around 50°C (122°F) is an acceptable maximum. Anything above that would be considered a plus and would be ideal as it allows us to not worry about cooling and various other issues.

2014 Design

After review of last year’s cameras and further discussion with those who operated the base station, we have determined that most of the camera issues came not from a lack of resolution, but from a lack of dynamic range, auto-exposure woes, and incorrectly focused cameras. In order to improve the cameras, we looked at the difference between image resolution and image quality. The second problem with the camera interface was the latency. We are currently looking to solving our latency problem through the radio frequency and data packaging.

Integration was something that was not discussed about the cameras last year. They didn't communicate with the mechanical team about how they wanted the cameras to mount the rover itself which lead to makeshift connections.

2015 Design

The main focus this year will be on early integration and correct selection of motors. The Auxiliary Squad has laid out that they believe that full vision around the rover can be completed by using 4 precisely mounted cameras. The squad looked into getting cameras to their specific mounting location. They will be taking into account the focus and focal length of all the cameras. The following trade study shows their top 6 choices and why they have selected these cameras for their respective locations. The Auxiliary squad will, also, prototype and test these cameras intensively. They will look into calibrating them to meet the proper specifications and getting the clearest image they can.

End Effectors

- Design Requirements
    - Functional Requirements
        - To create automated tools to go on the end of the robotic arm
        - To allow the rover to complete competition taks
        - To pick up tools and earth samples
    - Design Requirements
        - Physical Size
            - 3in x 3in x 1in
            - Must fit on the end of the robotic arm
            - As small as possible
        - Power Requirements
            - Supplied with 18V
            - Needs 12V to power the motors
            - Voltage will come from the robotic arm
        - Control/Communication
            - Atmega32U4/Atmega328
            - Data will be sent through XBee
            - Data needs to be wireless for Continuous rotation
    - Last Year's Issues
        - Lack of Continuous rotation
        - Unreliability
        - Slow motion of the gripper
        - Communication problems
        - Last minute integration
        - Precision

- Reliability and Risk Management
  - Remove complexity
  - Revert back to simplified form of last years system

2014 Design

During the 2014 rover season, the team focused mainly on the driving functionality of the rover and little on the auxiliary equipment. The oversight of these systems lead to integration errors and unreliable systems.

On the 2014 rover, there were two main end effectors: the gripper and the drill. They were both needed for specific tasks during the competition.

The gripper was power by a simple hobby servo. In order for the mechanical design to work, the servo was required to be geared down. The process of gearing down the servo made it noticeable slow. The speed of the opening and closing of the gripper was a cause for points being lost in the competition. The rover can't function while the gripper is opening or closing and caused dead or wasted time during the competition task. The gripper was powered and controlled using the board seen in the following figure. The gripper took in 12V power and stepped it down to 5V in order to power the servo and microcontroller. The microcontroller used on this board was the atmega328.

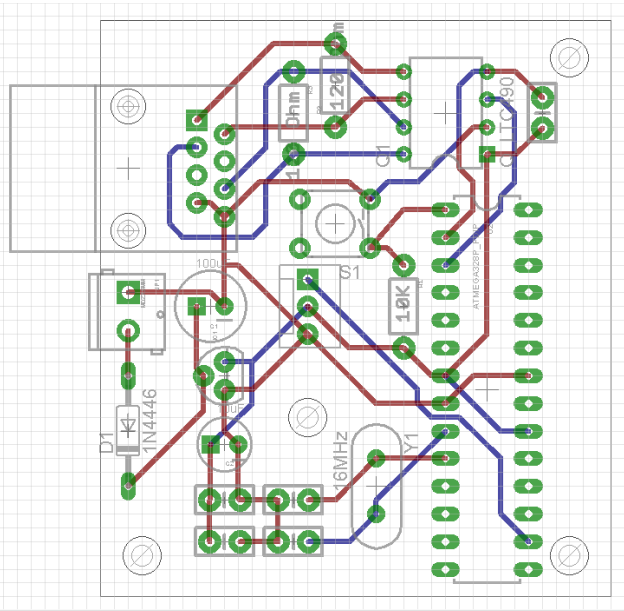
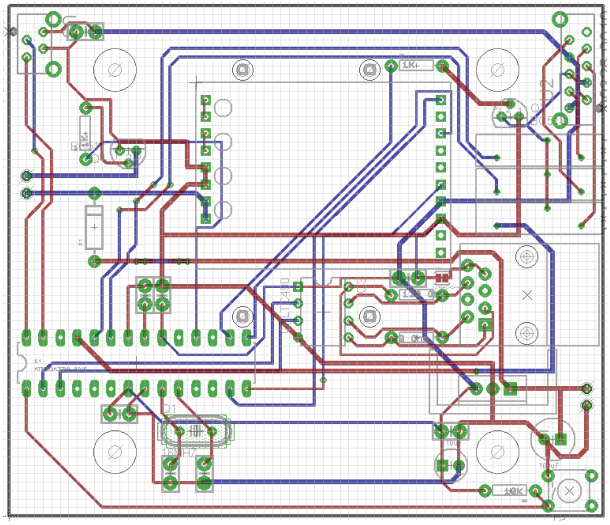


Figure - Gripper Board 2014

The drill was the second End Effector used on the 2014 Rover. The drill was used for the sample collection task. The drill needed to be able pick up a soil sample, so that scientific tests could be run on the soil. The drill acted as more than just a sample collector; it's secondary functionality was as a science experiment. The drill was intended to test the soil before the sample was taken in order to get valuable samples. The drill had two main sensor arrays: gas sensors and thermal conductivity. Neither of these systems were able to fully integrated and used. The drill was powered and controlled by the following figure. The drill was powered using 12V. The board took 12V and stepped it down 5V for the microcontroller. The motor that was used was a 12V motor and was powered using an offboard motor controller.



2015 Design

During the upcoming season, the Auxiliary squad will address all of the errors we ran into during the last season and have a primary focus on reliability. In the following section, I will discuss our changes for future.

New Communication Connections

The 2015 rover will have communication from the robotic arm to the end effectors through the wireless connections of XBees. Using XBees will solve a major problems from last year which was the lack of continuous rotation. The gripper needs to have continuous rotation in order to improve it's functionality. However, tests still need to be conducted to ensure that the reliability of the communication remains the same.

12V motors

The 2015 rover will be using 12V motors for both of the end effectors which will speed up the gripper. The motor will spin much faster that the hobby servo even if it is geared down to improve the torque.

Feedback Sensors

For both the gripper and the drill, feedback sensors will be added to improve the reliability and precision of the end effectors. Last year, we had no feedback sensors which allowed for our system to be damaged. On the gripper, we will be implementing pressure sensors and limit switches. On the drill, we will be implementing current sensor to ensure that the motor doesn't draw more current than it should.

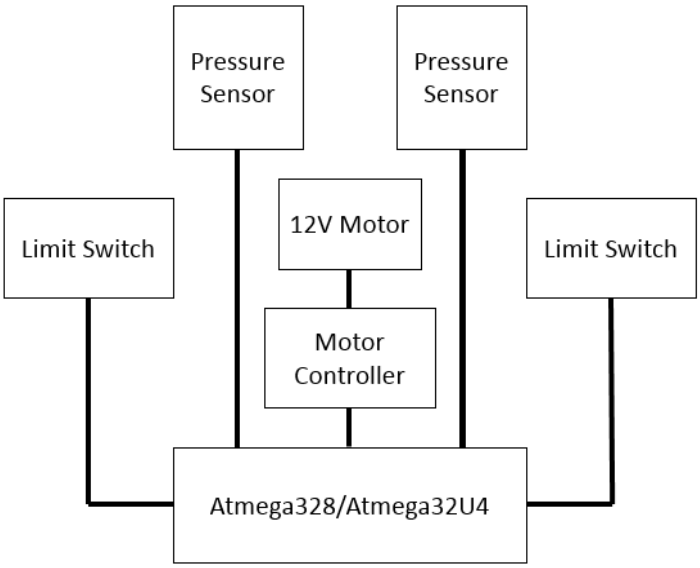


Figure - Gripper Black Box

Power Controls

- Design requirements
  - Functional requirements
    - Balance charge the battery system using both constant voltage and constant current based off of the current battery voltage in order to prolong the life of the battery
    - Monitor the battery voltages during operation
    - Monitor the amperage draw from the battery during rover operation, in addition to monitoring the amperage draw of the different busses
  - Design requirements (i.e. hard requirements, such as size/weight constraints, voltages, data formats, etc.)

        - All hardware requirements are being handled by the BMS squad on the Power team (to my knowledge)
        - Voltages needed will be dependant on microcontroller picked.
        - The smaller the footprint of the microcontroller used the better in order to save as much space on the PCB as possible
        - The more ADCs on board the microcontroller the better as to have the ability to read from multiple sources
        - The faster the ksps of the ADC reads the better as to improve response time
- Points of interface
    - Physical connection to rover
        - Should be handled by the BMS squad on the Power sub-team
    - Power Input
        - Voltage line dependant on selected microcontroller, most likely 3.3V
    - Data Connection to Base Station
        - Relay of information based off of current voltages and amperage draw
    - Data Connection to onboard rover devices
        - I'm assuming just a standard connection with a way to communicate voltage and current information to the motherboard for relay purposes
- Issues in last year's system
    - Hardware for balanced charging with both constant current and constant voltage was in place, however it was not implemented via software
    - Monitoring of onboard currents was not very accurate
    - Monitoring of onboard batteries was also not accurate
- Minimal requirements
    - Be able to charge the batteries
    - Be able to accurately monitor battery charge
    - Be able to accurately monitor battery discharge amperage

2014 Design

The design of last year's Battery Management System and Power Board controls, were not completely finished. The software designer weren't given a feasible amount of time to complete their projects

Battery Management System

The Battery Management system was design to be the protection for the batteries and to be the charging circuitry needed to charge the batteries. These two functions were "completed" We didn't get 100% functionality out of these systems. The protection of the batteries sent back telemetry that said the batteries weren't stable when they were, so it was not a reliable system The charging hardware was created,so that the Battery Management System (BMS) could charge the batteries using Constant Current Charging and Constant Voltage Charging.The BMS

was also equipped to have individual cell balancing while the rover was charging. The only charging functionality that was implemented last year was the Constant Voltage Charging. The BMS had major hardware failures that made it hard to see whether or not the software was a problem.

Power Board

The code for the Power board was never flashed to any of the chip we used on the rover and therefore never fully integrated into the system. The goal of this control was to have the ability to turn on/off the different power buses. The use for this would be for a sequential startup or emergency stop for one of the motors.The Power Board like the BMS had major hardware failures that would have made any software void of purpose. However, telemetry from the Power Train Boards will be helpful for any errors that should occur during the normal function of the rover.

2015 Design

For the coming year, the goal is reliability just like most other systems. However, the Power Controls group one and only focus this year will be reliability. As a team, we need to trust the system and the first step is to trust the Power Train Hardware which can be created through good communication to and from those boards.

Microcontrollers

Last year, the atmega328 was used the main microcontroller for both the BMS and the Power Board. However, after a trade study, this squad has decided that the ATmega32U4 is a better fit. The USB programming and extra analog pins should make it easier on programmers to program and collect data from these boards. In the figure below, the trade study that was completed by the squad on different ATmega microcontrollers.

	ATmega328P	**ATmega32U4**	**ATtiny828**	**ATxmega16e5**	**ATxmega32E5**
Flash (Kbytes)	32	32	8	16	32
Pin Count	32	44	32	32	32
Max. Operating Freq (MHz)	20	16	20	32	32
SRAM (Kbytes)	2	2.5	0.5	2	4
EEPROM (Bytes)	1024	1024	256	512	1024
Max I/O Pins	23	26	28	26	26
Operating Voltage (V)	1.8-5.5	2.7-5.5	1.7-5.5	1.6-3.6	1.6-3.6

Timers	3	4	2	3	3
32kHz RTC	Yes	No	Yes	Yes	Yes
Temp. Sensor	Yes	Yes	No	Yes	Yes
ADC Resolution	10	10	10	12	12
ADC Channels	8	12	28	16	16
ADC Speed (ksps)	15	15	15	300	300
DAC Channels	0	0	0	2	2
UART	1	1	1	2	2
SPI	2	2	1	1	1
I2C	1	1	1	1	1
USB Interface	No	Device	No	No	No
PWM Channels	6	8	4	16	16
Approximate price for 10 ea	$3.22	$6.09	$2.19	$2.52	$2.70

Figure - Microcontroller trade study

Battery Management System

The software design of the BMS will not change greatly from last year's design. The software design last year was rushed and only had minimal functionality. Hopefully with the time that given this year the software will be easy to read and well-developed. The software designers on this squad will work closely with hardware designers on the Power Train Team in order to understand the system fully and write efficient code. The basic function of the BMS will be finished first which is the monitoring of the battery cells and the sending of the data back to the Base Station.The next part of the code will handle the charging which will have a three part process. Constant Current Charging will be the most useful followed closely by Constant Voltage Charging. The Constant Voltage will occur after the batteries have been charged most of the way with Constant Current. The final part of the charging code will be the balancing. The balancing part relies heavily on the hardware being placed in the correct locations and functioning properly. The BMS goals didn't change from last year to this one, but more attention and structure will be placed on the design of the code.

Power Board

The software design for the Power Board will be very different and reliant upon the hardware design. However, the main goal remains the same which is to be able to turn the voltage buses on and off. There are multiple reasons why control of the different buses is important. The first being sequential turn on of the rover. The inrush current on last year's rover was incredibly high and dangerous to the system. A way of minimizing the inrush current is to

turn on different parts of the rover systematically and sequentially. The second reason for power board control would be stopping the different parts of the rover for drawing large amounts of current. All the motors of the rover can draw very large amounts of current. If the rover hits a rock or tries to go up a steep hill the motors can draw a current larger than the system can handle. It would be nice in a situation like this to be able to turn off a voltage bus if the current reaches a dangerous level. I think this could have protected some of our systems that were damaged or broken last year.

New Design Squads

The following two squads were created this year to solve some problems faced at last year's competition. The controls The first is the UAV team. The UAV team will be needed to relay radio signals over large obstructions in which the rover is not longer in site of the radios. The second group added to this team is a Controls Squad. The Controls squad will tackle two very difficult problems: creating a control scheme for the wheels and creating a control scheme for the robotic arm. The controls squad will be what takes this rover to the next level because with a control scheme implemented on both of these systems, the rover will be more reliable. The Controls and UAV squads will be pushing the limits of this team and striving for the future.

Controls

Functional Requirements

Functionally, our control design must reduce the error between the prevailing state of the controlled systems (robotic arm and drive system) and the desired state of these systems.

In the robotic arm, our design will take as an input the desired angle of each major joint with respect to a common reference angle as well as the concurrent angle of each joint. It will then use software tools to determine the difference between these angles and determine what must be done from a power standpoint to reduce this error to (ideally) zero. Our main goal is to give the arm the ability to change the load it is holding without the need for the operator to compensate for this change in load by adjusting the power being supplied to the motors.

In the drive system, our design will take as an input the desired rotational velocity of each wheel as well as the concurrent rotational velocity of each wheel. It will then use software tools to determine the difference between these velocities and determine what must be done to reduce this error. Our main goal with the drive system is to give the rover the ability to move at a

constant speed, no matter the condition of the terrain. This will also let us reduce how often and to what degree our wheels slip in the loose, Martian-like terrain.

Hard Requirements

Because our system is primarily theory-based, we have very little constraints that do not extend to the jurisdiction of other sub-teams. However, we must still design our system so that it works well with other sub-teams such that the integration of our systems into theirs does not hinder their abilities to meet their design requirements.

One example of this is the complexity of our design. Since our control designs are being implemented by four points of motion (2 joints) in the arm and 6 points of motion in the drive system, it is a good idea to keep our designs as simple as possible so as to not consume more processing power than that which may hinder other sub-teams from performing their own software tasks.

One design requirement that is mostly our responsibility is the physical limitations of our arm and drive system. Because we are designing the control interface, we must incorporate control and software limits, as well as perhaps electrical hardware limiting technologies, into our design so that the mechanical systems maintain a healthy distance between their physical limitations and the extent to which we give them the ability to move. One such case is that of the linear actuator on the robotic arm. Since an over-extension of the linear actuator can make it difficult for the operator to maintain effective control of the arm, we must implement software, and perhaps hardware, limits on the range of motion of the arm. Another example is the configuration of the gears in the major joints of the robotic arm. Because each symmetrical, motored gear contributes to two different types of motion (same direction = pivot of shaft; opposite direction = rotation of shaft), we much synchronize each motor with each other such that motion of arm acts as desired.

Preliminary Design Overview

Robotic Arm Controls

Currently the controls group is working on the design for forward kinematics which will be used on the robotic arm. Forward Kinematics find the location of the end of the robotic arm using the lengths of the arm lengths and the angles at each of those links. The arm will need to be modeled and measured accurately in order for the calculations to be correct. The location is then calculated through a series of matrix multiplications. The figure below lays out the basic feedback loops that are needed to control the motors.

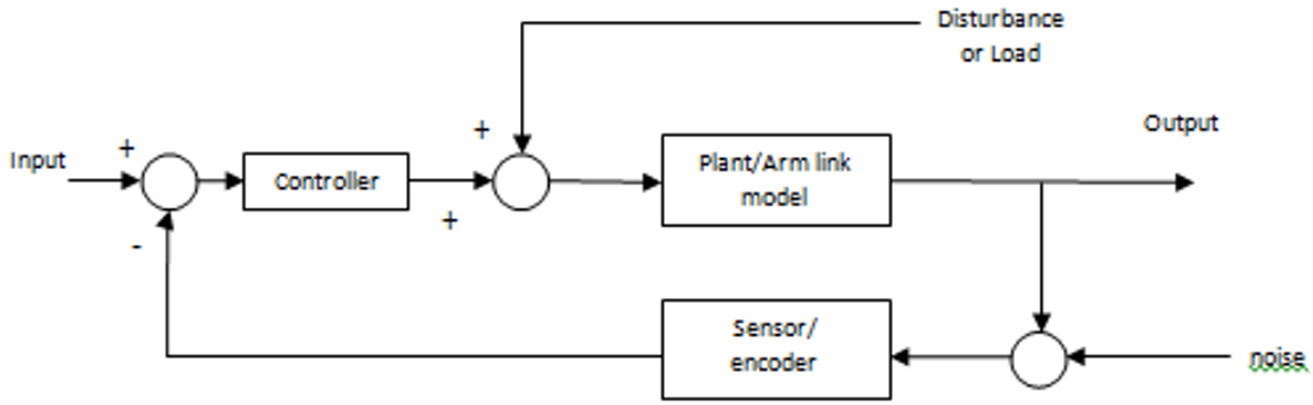


Figure - Basic Feedback Loop

Once the basic feedback loop is created then a more defined model can be created. The start to the creation of the forward kinematics model starts with a dynamic arm model as shown below.

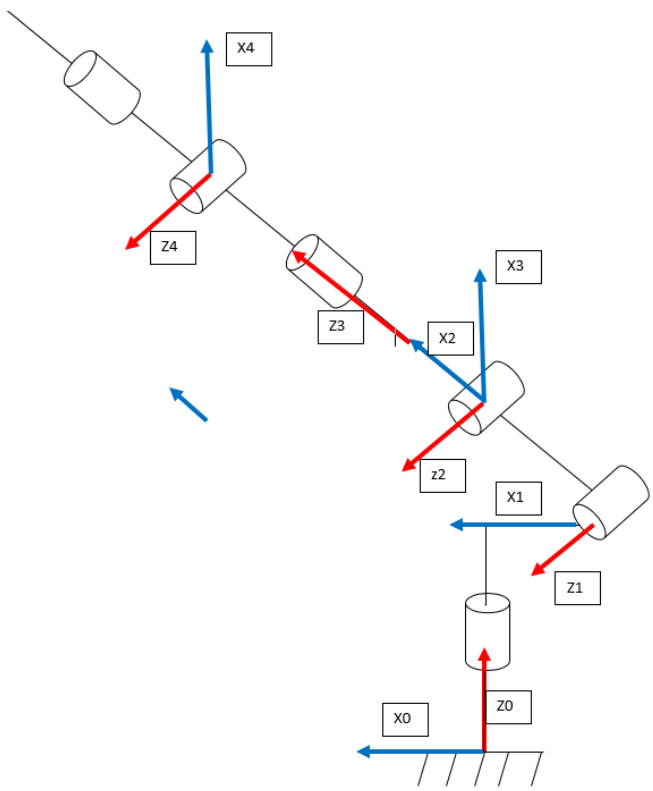
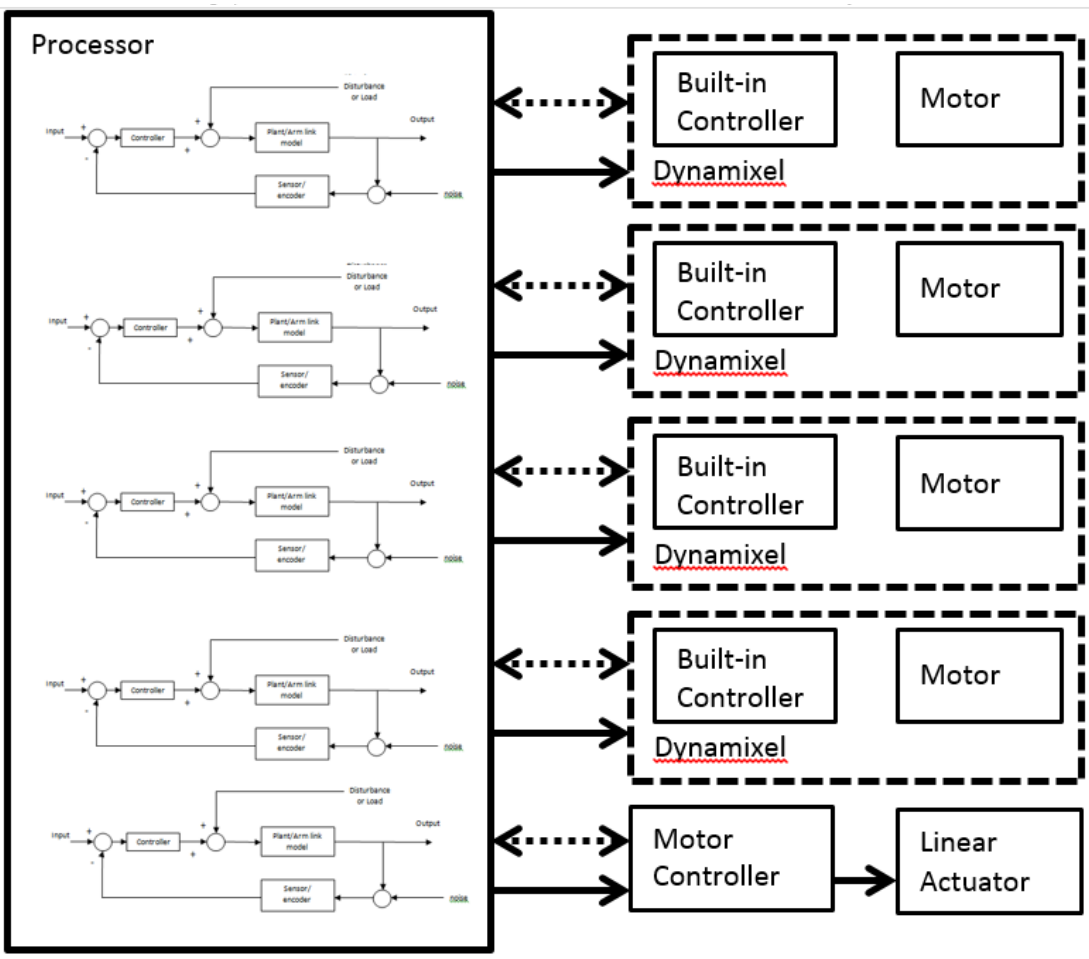


Figure - Dynamic Arm Model

The Black Box Model for the Robotic Arm is shown in the figure below. This figure represents the new design for the robotic arm that includes the Dynamixel servos.



Dynamixel has built in controller which can use angular position, velocity, and/or torque as both an input and/or an output, which makes it almost ideal when implementing a control design. Also very strong, fully rotational. The Dynamixel servos will be easier to control once the platform is learned. However, the servos that we use can be communicated to by a communication very similar to what we have on the rover or through basic TTL the communication and control should be easier than a basic motor encoder set up.

Propulsion Controls

The rover will be using brushless DC motors which will be much trickier to control that standard brushed DC motors. Last year the ideal system would have had torque control on the wheels which read the current flowing into the motor. After the current is read, the calculation of the torque is made and then the programmer can write the code to sustain that torque. For the

system this year, the controls group will implement a similar system. The need for torque control stems from the problem of climbing steep hills during competition. The rover last year was unable to grab enough traction and the wheels began to spin uncontrollably. If torque control was implemented the the rover would have slowed the wheel spin and in an attempt to gain more grip. A torque controlled propulsion system makes the system safer and improves the functionality of the rover.

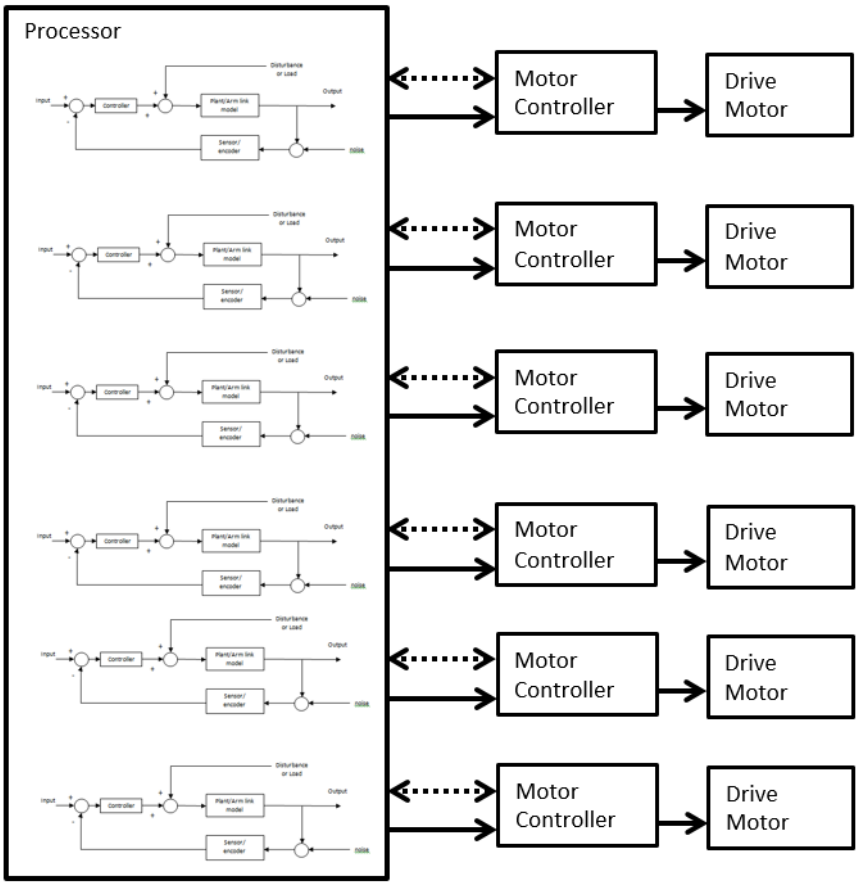


Figure - Black Box Model of Propulsion

The Motor will all be controlled as individual motors in the modeling of the system. However in the application of these designs, the motors on each side of the rover will work as one continuous group. The individual motor control will be useful to get the rover out of tight spots or during a strenuous terrain traversal task.

Alternatives and Risk Management

Both the robotic arm and propulsion schemes have fallbacks if the originals don't work. The robotic arm will fall back to a incremental control which will be made more precise through the work of the controls team. They will add feedback sensor and pull data from the servos to get more precise position measurements. The propulsion team will fall back to a velocity sensor feedback system. They will read the power being consumed by the motors combined with other sensors to get the velocity and use that to find more data like position and acceleration.

UAV Controls

The UAV is a solution to the problem of the Rover losing communications from base station due to the terrain or other obstacles blocking it. We chose for the UAV to be a quadrotor. The UAV has several requirements it must meet to be a viable option.

"

-- UAV Specific Requirements

- *Must carry a dead weight equal in weight to the battery (Competition Requirement)*
- *Must be able to take off from and land on the rover or near the rover*
- *Able to fly continuously in the air for longer than 10 minutes*
- *Must meet FAA regulations (legal issues)*
- *Must be easily secured and easily released from the rover (competition rule)*
- *Must be able to fly (Competition Requirement),-- lighter than air vehicles are illegal*
- *Must be able to carry weight of repeaters and other equipment necessary to perform its designated tasks*
- *Must have stability control*
- *Must have fail safes in place in case of communication loss occurs or high winds are encountered*
- *Must have reasonable positioning accuracy (TBD)*

-- Non UAV Requirements

- *Must relay radio signal from the base station up to 20-30m away from rover for purposes of extending communication range*
- *Must provide real time imagery from above for astronaut assistance*
- *Repeat signal for up to 40 min to complete the task within time allotted*
- *Ability to provide Clear Images for purposes of surveying terrain and identifying obstacles and geological landmarks*

--Requirements that are not needed unless we do drop boxes

- *Ability to see repeater from 20-30m to be able to find repeaters at end of task (for statically deployable repeaters)*
- *Repeater has to be easily picked up by rover (so we don't waste time on it) (for statically deployable repeaters)"*

There are several components that were included that needed its own requirements.

Batteries

Batteries is a major part to be selected. In the URC 2015 rules in part 2 paragraph f it instructs us to simulate Mars' gravity. To accomplish this we must "carry a dead weight equal to the weight of the battery(s) used.". This made the topic of flight time very sensitive. The larger the capacity of the battery the heavier the battery and that increases the extra payload as well.

The type of battery that we are using are Lithium-Polymer batteries. They are used for small scale aircraft a majority of the time. This is because of RC LiPo batteries are lightweight

and can be made in almost any shape and size, have large capacities, and RC LiPo batteries have high discharge rates to power the most demanding electric motors.

Rc LiPo batteries have 3 things to look for: capacity, number of cells, and C rating (discharge). The capacity is is basically how long the battery will last. The number of cells affects the voltage that can be output. Each individual cell supplies at max charge 4.2v, so if you are going to power something that requires 12v as a minimum to operate you will use 3 cells or more. Last is the discharge, the constant current that can be output by the battery. This is based off of your average current draw during flight.

The weight for each specification can be represented to a mathematical model as shown below.

4 Cell 30 C discharge

Battery Weight (g)	Cells	Capacity (mAh)	Discharge (C)	Product link
258	4	2200	30	http://www.hobbyking.com/hobbyking/store/__9460__Turnigy_2200mAh_4S_30C_Lipo_Pack.html
452	4	4000	30	http://www.hobbyking.com/hobbyking/store/__9511__Turnigy_4000mAh_4S_30C_Lipo_Pack.html
526	4	5000	30	http://www.hobbyking.com/hobbyking/store/__8588__ZIPPY_Flightmax_5000mAh_4S1P_30C.html
568	4	5800	30	http://www.hobbyking.com/hobbyking/store/__9933__ZIPPY_Flightmax_5800mAh_4S1P_30C_.html
845	4	8000	30	http://www.hobbyking.com/hobbyking/store/__16226__ZIPPY_Flightmax_8000mAh_4S1P_30C.html

weight = 0.098x +39
x = mAh

4 Cell 35 C discharge

253	4	2200	35	www.hobbyking.com/hobbyking/store/__11945__Turnigy_nano_tech_2200mah_4S_35_70C_Lipo_Pack.html
364	4	3000	35	http://www.hobbyking.com/hobbyking/store/__10100__Turnigy_3000mAh_4S_35C_Lipo_Pack.html
427	4	4000	35	http://www.hobbyking.com/hobbyking/store/__17242__Turnigy_nano_tech_4000mah_4S_35_70C_Lipo_Pack.html
466	4	4500	35	http://www.hobbyking.com/hobbyking/store/__21422__ZIPPY_Compact_4500mAh_4S_35C_Lipo_Pack.html

weight = 0.089x +73

3 Cell 30 C discharge				
Battery Weight (g)	Cells	Capacity (mAh)	Discharge (C)	Product link
190	3	2200	30	
197	3	2200	30	http://hobbyking.com/hobbyking/store/__9394__Turnigy_2200mAh_3S_30C_Lipo_Pack.html
218	3	2450	30	http://hobbyking.com/hobbyking/store/__10274__Turnigy_2450mAh_3S_30C_Lipo_Pack.html
300	3	3300	30	http://hobbyking.com/hobbyking/store/__24776__B_Grade_3300mAh_3S_30C_Lipoly_Battery.html
324	3	4000	30	
359	3	5000	30	http://hobbyking.com/hobbyking/store/__37357__ZIPPY_Traxxas_5000mAh_3S1P_30C_Lipo_Pack_Suits_TRA2872_.html
522	3	6400	30	http://hobbyking.com/hobbyking/store/__37358__ZIPPY_Traxxas_6400mAh_3S1P_30C_Lipo_Pack_Suits_TRA2857_.html
644	3	8000	30	http://hobbyking.com/hobbyking/store/__16225__ZIPPY_Flightmax_8000mAh_3S1P_30C.html
	weight = 0.076x +24			
	x = mAh			
3 Cell 35 C discharge				
210	3	2450	35	
199	3	2200	35	http://hobbyking.com/hobbyking/store/__10383__Turnigy_2200mAh_3S_35C_Lipo_Pack.html
220	3	2650	35	
297	3	3700	35	http://hobbyking.com/hobbyking/store/__21412__ZIPPY_Compact_3700mAh_3S_35C_Lipo_Pack.html
360	3	4500	35	http://hobbyking.com/hobbyking/store/__21421__ZIPPY_Compact_4500mAh_3S_35C_Lipo_Pack.html
	weight = .0712x +36			
	x=mAh			

The conclusion from all of this is that 3 cell weighs less (smaller slope) but has a lower voltage than the. higher discharge also results in a smaller slope for weight but a higher y intercept (base weight). There is a break even point where the higher discharge begins to weigh less. For 4 cell it is around 3778mAh.

For our use we want to get the weight as low as we can but we chose a 4 cell so the motors can drive our large propellers. If we could get low enough K/v motor we may be able to go to 3 cell (this will be explained in the motor section)

Motors

For motors there are brushed and brushless motors. We decided to go for brushless motors because they are more widely used for hobbyist aircraft and for good reason too. They are a lot more efficient. Compared to the brushed motors that has the same output, the brushless ones are lighter and is said to produce more torque.

Most motors on quadcopters are outrunners. In outrunner motors, the rotating part is on the outside, and not the inside (as opposed to the inrunner motors). Because of this layout this type of motors can generate much more torque and for a quadrotor its important to have as much torque as you can using the least amount of power.

The biggest part though when judging motors are the the K/v which is the rpm per volt. This can be important because the lower the K/v the more torque the motor can have and you can achieve higher altitudes for less power.

Flight Control Boards

- Design requirements

    - Must be consistent
    - Can output things like voltage
    - has gps capabilities
    - Software can be edited
- Flight control options that meets or is near meeting the requirements
    - the open pilot cc3d
    - aeroquad
    - ardupilot
    - several dji options

Out of the options presented we decided on the Ardupilot Mega board because of it's open source functionality, familiar technology for the team (Arduino) and the notoriety the board has for its preformance

Communication methods

Transmitter/Receiver for flight control

For hobbyist aircraft (or most RC things) in 2014 most devices are communicated to through 2.4Ghz. The devices for those are the easiest to find and they are pretty cheap new. There are issues with using 2.4Ghz though. In the URC rules they state, "Teams shall be limited to using no more than three channels in the 2.4 GHz band". Quadcopter operates with a minimum of 4 channels, 1 for every axis of movement (pitch, yaw, roll, and throttle). The number of channel gets even higher when you add in other functions we need like camera gimbal control.

To solve this issue we are looking at alternatives like 72Mhz or 433Mhz to circumvent the regulation issue.

Transmitter/Receiver for video

The off the shelf solutions for video (FPV) communication come in a variety of frequencies. We are currently choosing a 900Mhz solution because of its cheap price for the full system. We may change to 1.2 Ghz to avoid any issues with the regulations set by URC for 900Mhz.

Repeater for rover communication

For the repeater we are going with a device named the Ubiquiti Picostation. This device is a lightweight low power solution that also has a small footprint but can repeat the for a really far distance.

Important Deadlines

October 20th - Motherboard/Robotic Arm Bill of Materials ordered
October 30th -Cameras Selected
October 30th - Fully Define UAV needs
November 7th - Motherboard/Robotic Arm E-Boards Populated
November 14th - Motherboard/Robotic Arm Full Board Review
November 14th - UAV Board Designed/Ordered off the shelf
November 21st - Motherboard/Robotic Arm Final Board/BOM
November 21st - BMS Code - Monitoring Cells
November 21st - Cameras Full System Review
November 21st - Power Board - Monitoring Buses/Turning them on/off
November 21st - Gripper/Drill Schematics Completed
November 21st - Purchase Tablet for Mobile Base Station
November 21st - Propulsion Controls Defined
November 30th - Gripper/Drill Final Board/BOM
November 30th - UAV Board Full System Review
November 30th - Motherboard Software 80%
November 30th - Forward Kinematics Tested on Phoenix
November 30th - Robotic Arm Software 80%
December 15th - Propulsion Controls Testing
January 15th - Propulsion Controls Full System Review

Summary

During the 2015 Rover Season, the Telemetry and Controls Sub-Team will focus on reliability and testing. The Telemetry systems last year suffered from inadequate time to complete and test the full communication and software. This year we are adopting a much more aggressive schedule that finishes the bulk of our work in the first semester leaving most of the second semester for integration and testing. The whole team will be doing more testing and Telemetry and Controls will be a crucial part of the full system tests. In this document, the team discussed what the failures and successes were for the rover last season and our preliminary design decisions that we have made to fix those problems. During this review we have started our design process and some of the many options the team has to pursue.