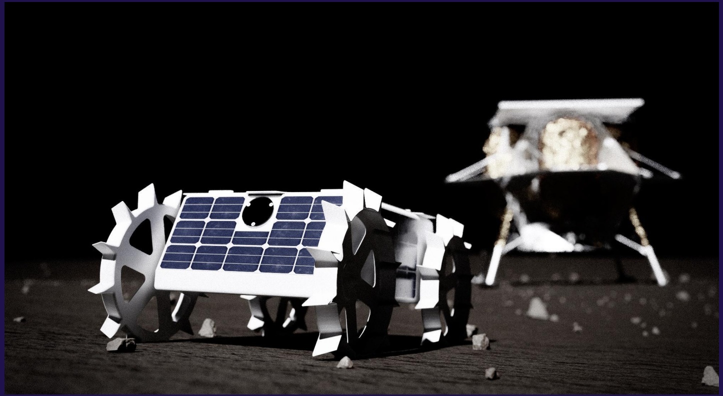
PROJECT BELKA

John Paul Harriman, Sam Adams, Mia Han

Data with integrity

Current Cube Rover



A 4.4 pound rover that is sent to the moon to collect and send data and images back to earth over wifi and is controlled from earth.

- Currently no way to verify collected data is accurate
  - Data may be corrupted by moon effects like radiation and light
- Not currently space grade
- Rover talks to Lunar Lander using UART and Wifi.
- Rover components talk to each other using I2C and QSPI.
- Currently there is no error handling for flipped bits and dropped packets.
- Moon radiation makes it very likely for bits to be flipped and packets to be dropped.

Use Cases

Solution: Create protocols between components in the CubeRover architecture (MCU, I2C, QSPI, UART, GPIO) that detect and correct errors

We will recreate the architecture of the CubeRover and use a microprocessor to simulate the potential effects of the moon on the data like bit flips and data packet dropping.

- A GUI will be used to control the microprocessor to test the accuracy and efficiency of our protocols
- ECE Areas: Software and hardware

Requirements of Projects

Power Consumption/Capacity

Battery Capacity	60 Wh
Power Consumption without Solar	1.02 W
Power Consumption with Solar	.64 W
Effective Battery Capacity	150 Wh
Energy/Instruction	TBD

Error Correction/Detection

Packet Recovery	100%
Code Rate	70%
Bit Recovery	100%
Max Handshake Instructions	4 Calls

Space Specifications

Upstream Ground-to-Rover	2038 Bytes/Payload
Downstream Rover-to-Ground	2^16 Bytes/Payload
Time from Earth-Moon	~2.6 seconds

Key Technical Challenges

Maximizing Efficiency for Error Correction:

- Each protocol needs different error corrections due to specification of each.
- UART and I2C - Defined and restricted number of bits
- SPI and UDP - Undefined number of bits/packets

Timing:

- Accurate timing is not only dependent on our code (interrupts, scheduling, etc)
- Want code to have minimal impact (<1%) on existing code timing.

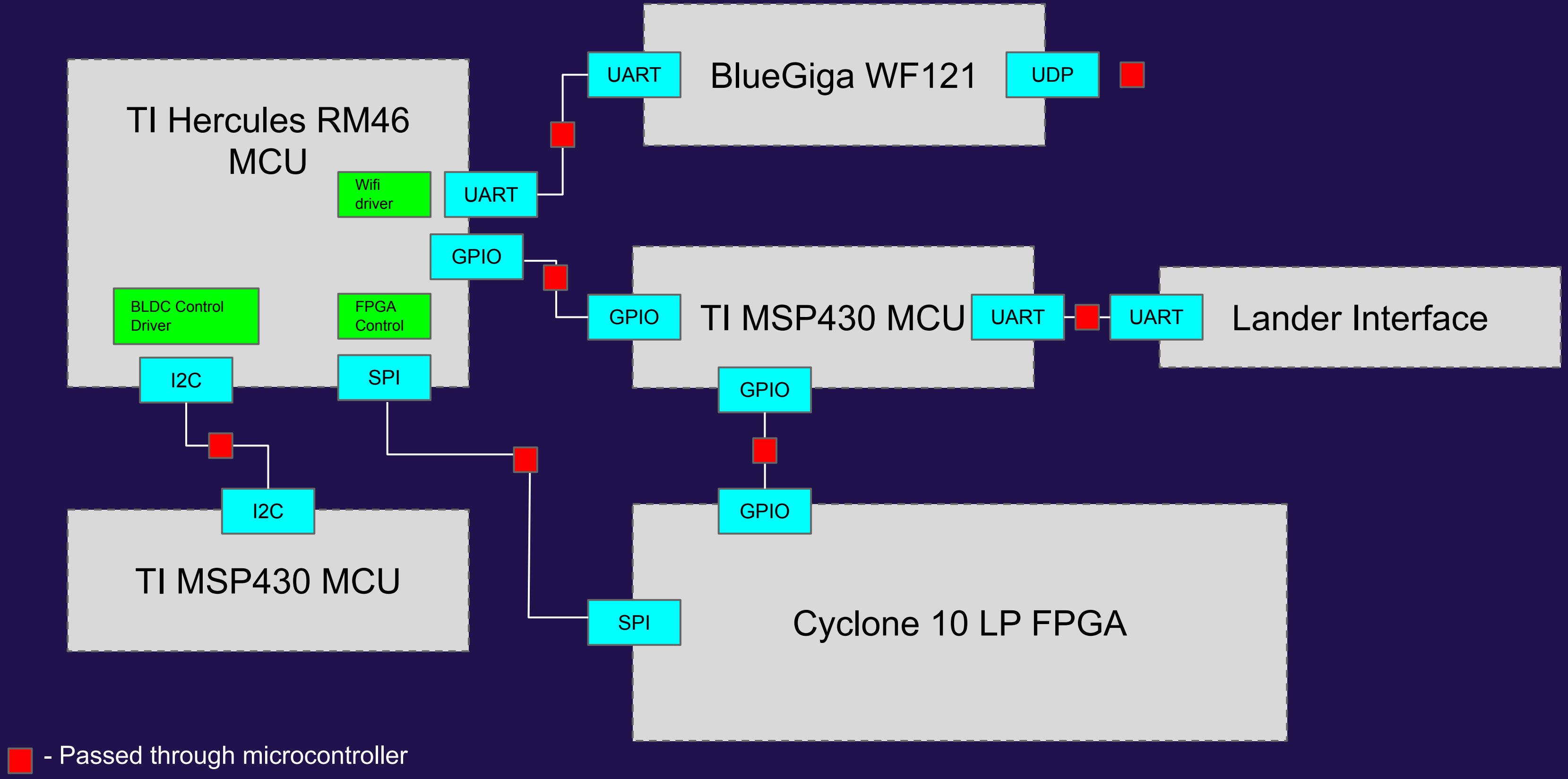
Drivers

- Each driver can be unwieldy and we are working with multiple components
- Have to create wrapper for each driver such that we can implement our correction on top of the established software
- Drivers should have C implementations, but may only contain binaries

Accurate Simulation

- How can we accurately simulate the radiation effect on the hardware.

Overall Architecture



Microcontroller Architecture

Microcontroller Discovery Board - STM32F4DISCOVERY

Design Rationale:

- Contains all necessary peripherals
    - 3 I2C (2 needed), 2 UART (2 needed), 2 SPI (2 needed).
- Easy to use programming interface
- Easily expandable/easily moddable
- Software interface similar to Arduino
- Controlled using Arduino IDE

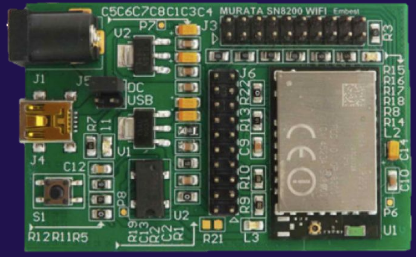
Discover WiFi Add-on board for STM32F4

Design Rationale:

- Can easily set-up UDP protocol
- Supported module for Discovery board
- Will use last I2C connection

Microcontroller Discovery Board

WiFi Add-on board



Design - Error Correction/Detection

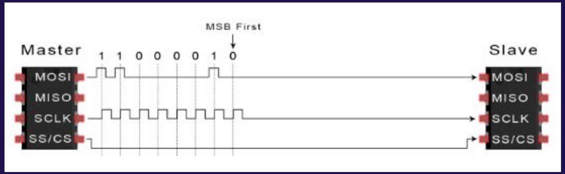
Example I2C Protocol

Start	7 or 10 bits	R/W	ACK	8 Bits	ACK	8 Bits	STOP

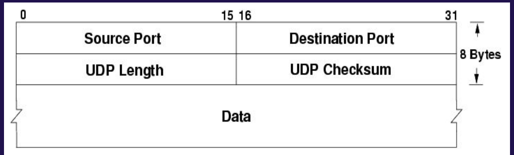
Example UART Protocol

Start	5 or 9 bits	Parity	STOP

Example SPI Protocol



Example UDP Protocol



Correction Code Possibilities:

Cyclic Redundancy Check

- Pros - Simple to implement, highly accurate
- Cons - Does not contain error correction

Reed-Solomon:

- Pros - Error Correction and Detection, Easily scalable
- Cons - Codec needs extra memory, expensive due to matrix computation

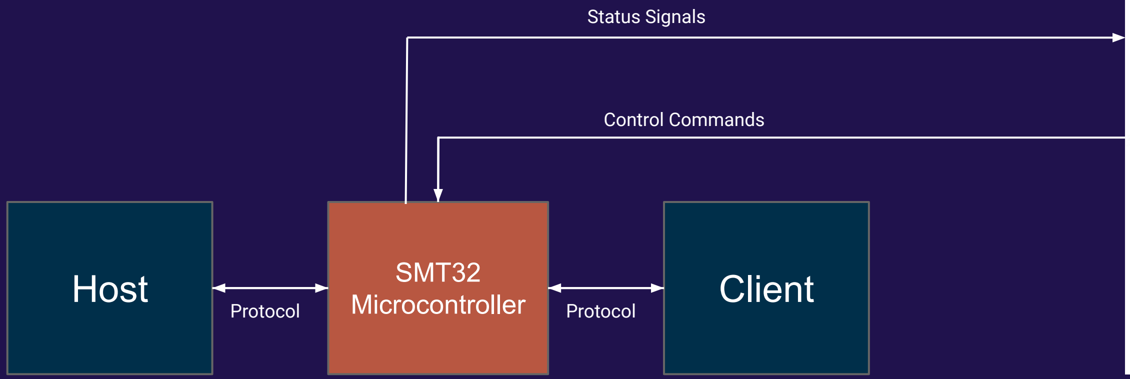
*Low-Density Parity Check:

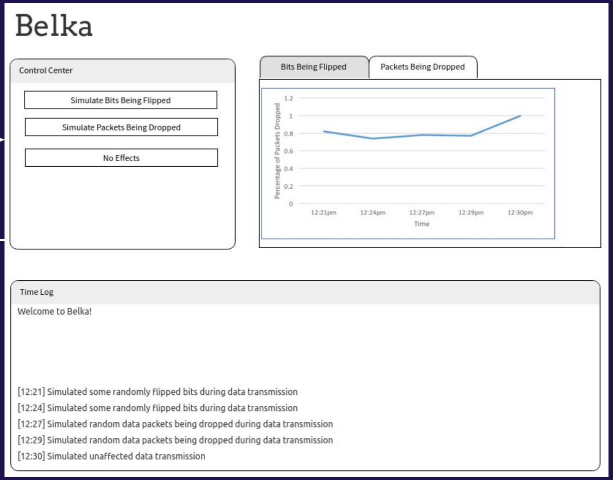
- Pros - Scalable, Parity bits are accounted for
- Cons - High drop in code rate

Design - Verification

GUI Implementation

- Implemented in Python using PythonUSB and microUSB libraries





Testing, Verification and Metrics

- Latency:

$$T = \frac{instructions}{program} \times \frac{cycles}{instructions} \times \frac{time}{cycles}$$

- Verification and Accuracy: GUI
  - Configure the number of bits flipped or packets dropped in GUI
  - Errors detected and corrected will be sent back to the GUI
  - Testbenches will run several tests to determine percentage of accuracy
  - Goal: 100% accuracy data detecting and correcting

- Power Verification
  - Use spec of each component to calculate the amount of power used per instruction
  - Verify power consumption by using a multimeter

Division of Labor

John Paul	Mia	Sam
Leads designing protocol for error detection and correction for serial data transmission	Leads GUI design, implementation, and integration	Leads designing protocol for error detection and correction for UDP and network data transmission

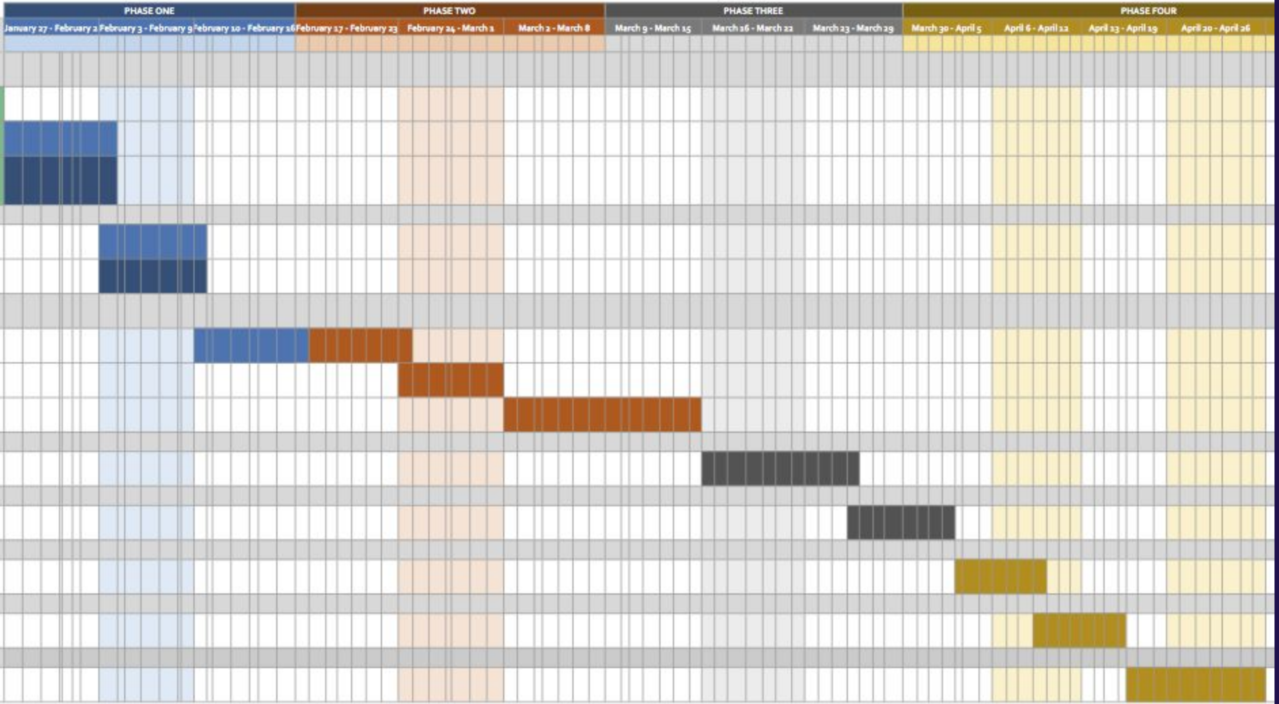
All

- Software implementation of error checking (C/C++)
- Creating GUI (Python)
- Programming MCU to simulate moon effects
- Creating and putting together components to recreate CubeRover's architecture

Gantt Chart

GANTT CHART

PROJECT TITLE	Belka
PROJECT MANAGERS	Mia Han, John Paul Harriman, Sam Adams



NUMBER	TASK TITLE	TASK OWNER	START DATE	End Date	DURATION	PCT OF TASK COMPLETE
1	Requirements Definition and Analysis					
1.1	Project Abstract	JP, Mia, Sam	1/22/20	1/23/20	1	100%
1.2	Project Proposal	JP, Mia, Sam	1/27/20	2/3/20	6	100%
1.3	Research: Components & Drivers, Methods and Approaches	JP, Mia, Sam	1/27/20	2/3/20	6	100%
2	Architecture Design					
	Design Protocols for Error Detection and Error Handling	JP, Sam	2/3/20	2/10/20	7	0%
2.1	Design UI Layout and Implementation	Mia	2/3/20	2/10/20	7	0%
3	Implementation: Protocols and GUI					
	Implementing Serial Transmission	JP, Mia, Sam	2/10/20	2/24/20	14	0%
	Implementing UDP/Wifi Protocols	JP, Mia, Sam	2/24/20	3/2/20	8	0%
3.1	Implementing GUI	JP, Mia, Sam	3/2/20	3/16/20	14	0%
4	Integration					
4.1	Integrating Components + GUI	JP, Mia, Sam	3/16/20	3/26/20	10	0%
5	Verification					
5.1	Initial System Analysis + Revision	JP, Mia, Sam	3/26/20	4/3/20	7	0%
6	Transition					
6.1	Refine prototype	JP, Mia, Sam	4/3/20	4/10/20	7	0%
7	Validation					
	Ensuring Components are close to meeting requirements	JP, Mia, Sam	4/10/20	4/15/20	5	0%
7.1	Slack					
	Slack + Creating Final Presentation	JP, Mia, Sam	4/19/20	4/26/20	7	0%