Individual Progress Report

Proof of Concept B

Project Details

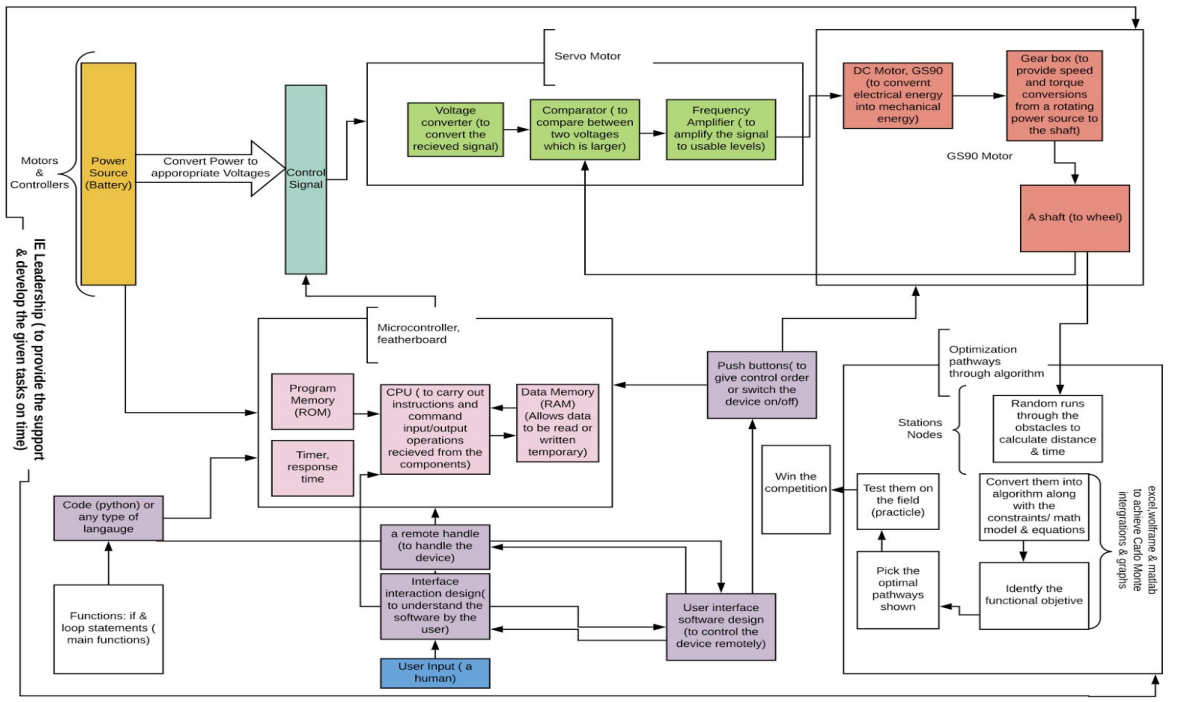
Project name	Cat's Conundrum
Group Number	C7
Author, discipline	Elizabeth Andrews, CS
Reporting period	Jan. 2018 - Feb. 2018
Date Due	Feb. 11

Summary

For this milestone, I continued to work on developing the Graphical User Interface. We just recently received our microcontroller in the mail, an Adafruit Feather Board Bluefruit LE. Once I was able to work directly with the Bluefruit, I was able to design two tests to wirelessly communicate with the Bluefruit via Bluetooth on my laptop. I used the Adafruit Bluefruit LE Connect application developed by Arduino to send commands to the Bluefruit. My first test was sending a simple "Hello World!" message to the Bluefruit, and my second test was manually toggling an LED on the board itself. With both tests, I was able to successfully perform the tests and also received a confirmation from the board.

I also decided to take a different approach to the controls on the GUI. For my initial prototype, I had four buttons that you would click one time that would then update a label to show which button was clicked. After further research and brainstorming, I will try to change the controls of the GUI so that instead of clicking on buttons, the user will press keys on a keyboard and hold them down to keep the label changed. This way, when the GUI is integrated with the microcontroller and the vehicle, we will be able to control the vehicle and steer it more precisely.

As a team, we also created the following functional diagram:



My contributions to this functional tree were mostly the lower left section, which includes the functionality of the GUI, user input from whoever is interacting with the GUI at the time, and an interface that successfully connects the GUI with the microcontroller.

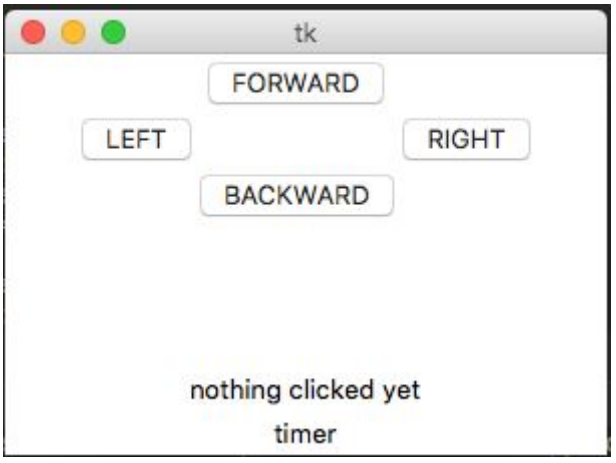
My team also created the following Morph chart:

Morph Chart (Group)					
Functions	**Solution1**	**Solution2**	**Solution3**	**Solution4**	**Solution5**
Power the vehicle	AAA batteries	AA batteries	9V battery	Rechargable batteries	Gas
Movement	4 wheels	Tracks	Legs	3 wheels	Hop
Turning	1 servo	Skid Stear	Articulation	2 servos	
Suspension	Springs	Flexible parts	Pistons	Soft wheels	
Durability	Wood	Metal	PLA	ABS	
Acceleration	Accelerometer	Changing gears	Set gears		
Graphical User Interface	Written in Python	Written in C-based language	Written in Swift		
Interface GUI with microcontroller	Wifi	Bluetooth	Long wire	Radio	Infrared
Optimization: pathways	Lindo	Excel	Matlab	Wolfram	Desmos
Charting programs	SmartDraw	Lucidchart	Excel	Draw.io	Smartsheet
Leadership Styles	The Autocrat	The Laissez-Faire	The Participative Leader	The Transactional Leader	The Transformational Leader

For the Morph chart, the functionalities I added were the “Graphical User Interface” and “Interface GUI with microcontroller”. It was difficult to think of very many different ways to address these problems because for the GUI itself, the language it is written in depends entirely on the type of microcontroller we use, and the interfacing must be wireless as well as supported by the specific microcontroller we choose. So in reality, most of these solutions will not actually be viable for my part of the project. Still, it was interesting to see other people’s ideas for some of the other functionalities that we added to the chart.

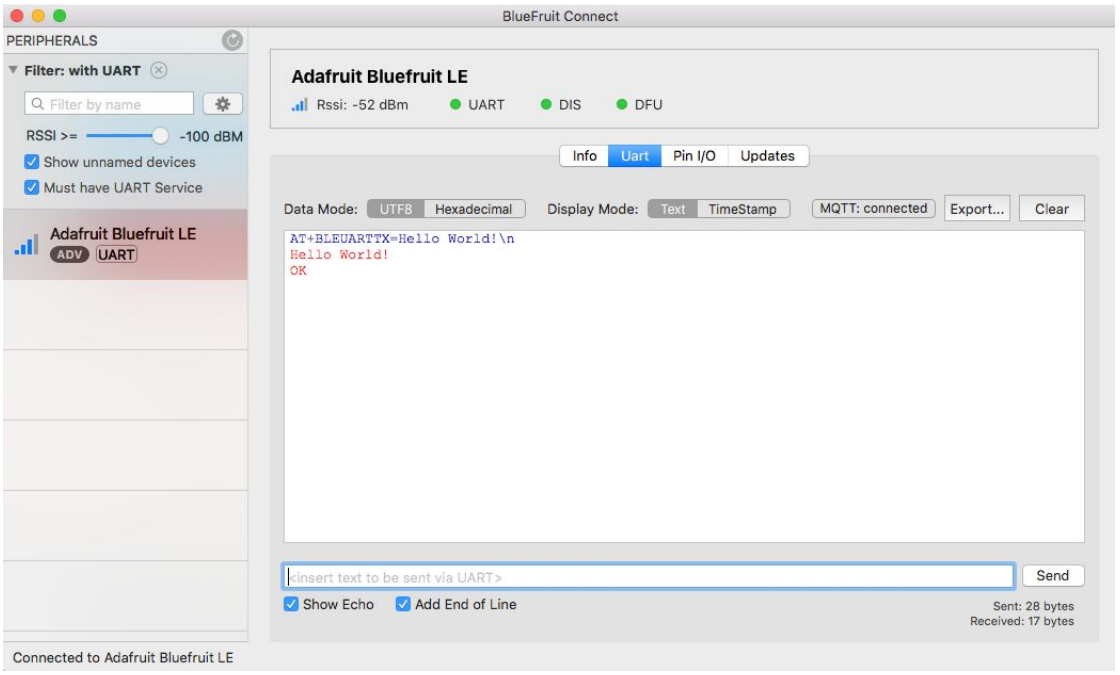
Activities

Create Graphical User Interface

Status	In progress
Objective	Write a simple GUI that can successfully interface with the chosen microprocessor.
My time on this task	1 hour
Support team member(s) time on task	N/A
Visual Progress Update	
Current Progress	For this milestone, I focused more on developing tests for the microcontroller than on the GUI. At the moment, the GUI is at the same stage as it was for the last milestone. I have planned out some updates and design changes, and I will be implementing them for the next milestone.
Outputs created	The same output as produced last milestone, the above Graphical User Interface.
System Integration Considerations	Now that we have the microcontroller that we will use, I need to begin working on integrating this GUI with the Arduino Bluefruit that we got.
Challenges/Lessons learned	N/A, since I did not focus on this activity during this milestone.

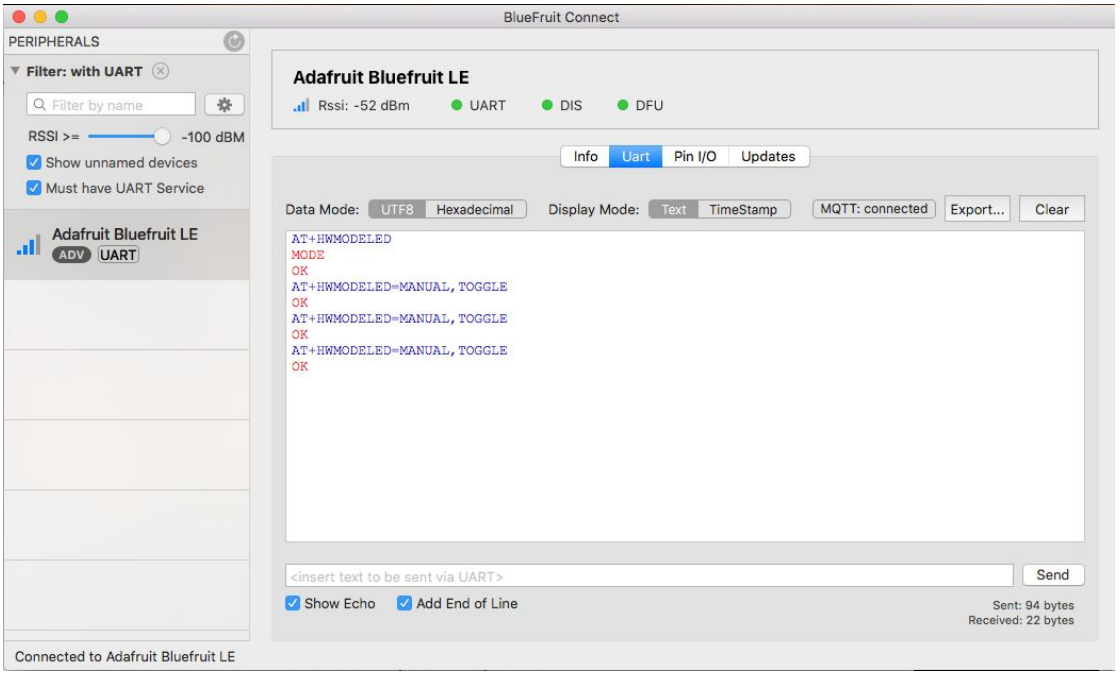
Display basic ability to communicate wirelessly

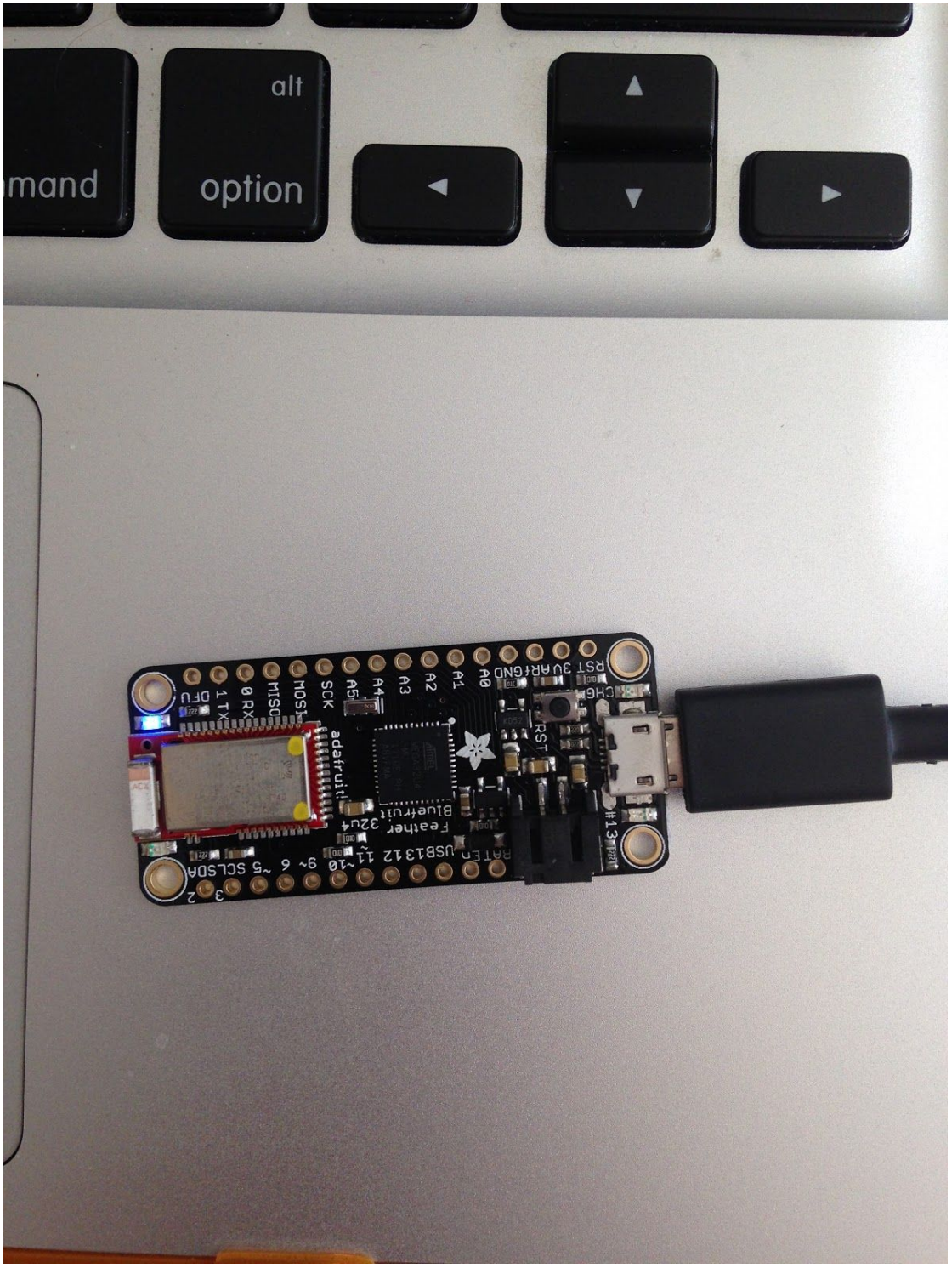
Status	Achieved
Objective	Write a simple “Hello World” program displaying the ability to communicate wirelessly with a microcontroller.
My time on this task	2 hours
Support team member(s) time on task	N/A

Visual Progress Update	
Current Progress	Now that we have received our microcontroller, an Adafruit Bluefruit, I am able to start writing code and tests for it. I connected the Bluefruit to a USB power source and used the Adafruit Bluefruit LE Connect application to try and make a connection with the Bluefruit wirelessly. I was able to connect my laptop to the Bluefruit after lots of trial and error, and figured out how to switch the Bluefruit into Command Mode so that I could send commands and get information from the microcontroller. In the image above, the blue text is the information that I sent to the Bluefruit, and the red text was the information that I received back from the Bluefruit. I sent the command 'AT+BLEUARTTX="Hello World!\n"', to print out a "Hello World!" string with a new line character. The response I got from the Bluefruit was the "Hello World!" string, and on a new line an "OK" message to confirm that the exchange of information was completed successfully.
Outputs created	I was able to communicate with the Bluefruit wirelessly over Bluetooth, and printed out a simple "Hello World!" message.
System Integration Considerations	This was only a basic test that I used to try and become more familiar with the Bluefruit and the software that I will be using to communicate with it. I need to figure out how to use what I learned from this activity to communicate with the vehicle's motors.
Challenges/Lessons learned	This is not the microcontroller that I would have chosen for this project. My Electrical Engineer was not very willing to compromise on a microcontroller that would have been easier for me to code. It's going to be extremely challenging to work with this Arduino Bluefruit, especially with a GUI that will be written from scratch.

Develop another test to display wireless communication

Status	**Achieved**
Objective	Write another test that accurately displays the ability of the GUI to communicate wirelessly with the microcontroller. This test should turn an LED on and off on the Bluefruit microcontroller.
My time on this task	3 hours

Support team member(s) time on task	N/A
Visual Progress Update	

	
Current Progress	I decided to write a test that would turn an LED on the Bluefruit on and off. After connecting to the Bluefruit via Bluetooth, I sent several commands to do this. In the first image above, the blue text is the command that I sent from my laptop, and the red text is the response I received from the Bluefruit. First, I sent the command “AT+HWMODELED” to ensure that a command existed that would allow me to interact with the LEDs on the Bluefruit. I received an “OK” in response. Then I sent the command “AT+HWMODELED=MANUAL, TOGGLE” to set the LED mode to be manually controlled and then toggle it to the opposite setting. I received an “OK” in response from the Bluefruit, and the red LED on the microcontroller switched off. I repeated the command, and the LED switched back on. I sent the command one final time and the LED turned off.
Outputs created	I was successful in writing a test that turned a small LED on the Bluefruit microcontroller on and off.

System Integration Considerations	This particular test will probably not be integrated with the final system, because the goal of this activity was merely to test my ability to communicate wirelessly with the microcontroller.
Challenges/Lessons learned	This is not the microcontroller that I would have chosen for this project. My Electrical Engineer was not very willing to compromise on a microcontroller that would have been easier for me to code. It's going to be extremely challenging to work with this Arduino Bluefruit, especially with a GUI that will be written from scratch.

Create a Morph chart with team

Status	**Achieved**
Objective	Create a Morph chart with the other team members to address functionality and possible solutions to the needed functionalities.
My time on this task	.5 hours
Support team member(s) time on task	Alyssa, Mubarak, Jacob, Ryan
Visual Progress Update	**Morph Chart (Group)** Functions / Solution1 / Solution2 / Solution3 / Solution4 / Solution5 Power the vehicle: AAA batteries / AA batteries / 9V battery / Rechargable batteries / Gas Movement: 4 wheels / Tracks / Legs / 3 wheels / Hop Turning: 1 servo / Skid Stear / Articulation / 2 servos Suspension: Springs / Flexible parts / Pistons / Soft wheels Durability: Wood / Metal / PLA / ABS Acceleration: Accelerometer / Changing gears / Set gears Graphical User Interface: Written in Python / Written in C-based language / Written in Swift Interface GUI with microcontroller: Wifi / Bluetooth / Long wire / Radio / Infrared Optimization: pathways: Lindo / Excel / Matlab / Wolfram / Desmos Charting programs: SmartDraw / Lucidchart / Excel / Draw.io / Smartsheet Leadership Styles: The Autocrat / The Laissez-Faire / The Participative Leader / The Transactional Leader / The Transformational Leader
Current Progress	While in class, our team created a very simple Morph chart in Excel. We addressed the issues of powering the vehicle, steering and turning the vehicle, the GUI needed to control the vehicle, charting programs, acceleration, and more. While it was difficult to come up with five different solutions to some of the functionalities, we had quite a few ideas and were able to come up with some unique ways to solve some of the problems.
Outputs created	A Morph chart showing some different functionalities for our team and a number of different solutions for achieving those functionalities.
System Integration Considerations	N/A
Challenges/Lessons learned	It was difficult to think of new functionalities in order to fill up all the solutions. We have done so ideating already for this project that I did not feel like filling out this chart really helped with any new ideas and it didn't really produce any real results.

Total Time On Task for this Milestone

Total time spent by me	6.5 hours
Total time spent by support team members	.5 hours

Next Steps

For the next milestone, we will be making our first prototype. I will need to put a lot of time into creating a functional GUI that can communicate with the Bluefruit in some capacity. This will require my Electrical Engineer and I to really begin collaborating and bouncing ideas off of each other for how to connect the code that I write to the servo and motors.

Communicating over Bluetooth with this microcontroller is going to be very difficult. There is not a lot of online documentation that I can refer to, and it's going to be lots of trial and error to figure out what works and what doesn't. I will probably be reaching out to several people I know who took the class last semester and find out what strategies they used for writing and interfacing the GUI. If I get to a point where I am completely unable to accomplish a working GUI, I will purchase a Raspberry Pi and use that instead.

Archived Activities

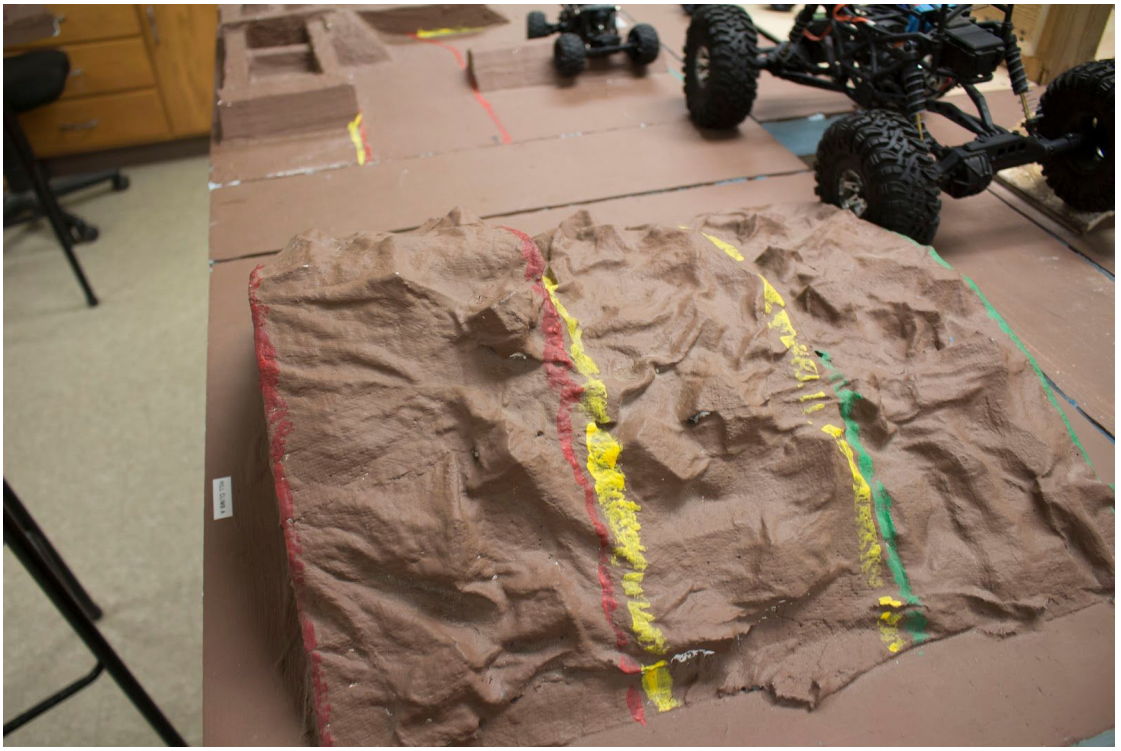
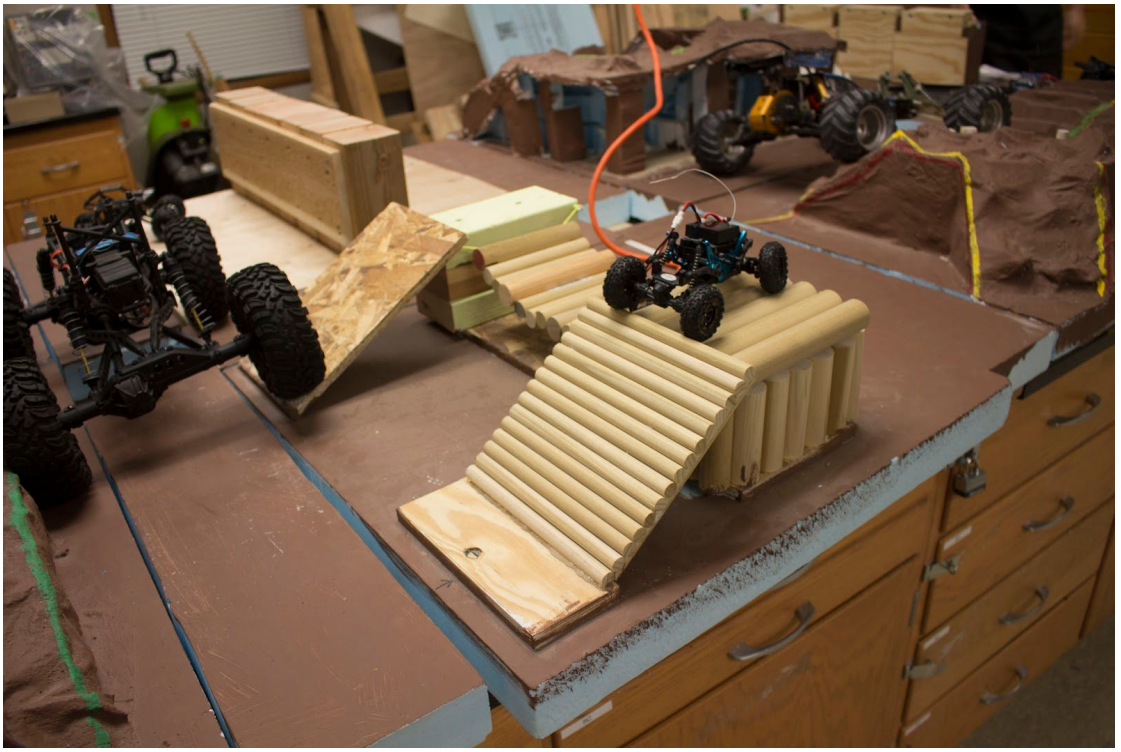
Establish Functional Requirements and Objectives

Status	**Achieved**
Objective	Establish a list of functional requirements and objectives for the CS portion of the project based on both my own needs and the needs of my team members, specifically my electrical engineer.
My time on this task	~3-4 hours
Support team member(s) time on task	Alyssa Ferry: ~2 hours
Visual Progress Update	N/A
Current Progress	Based on discussions with my electrical engineer and my own research, I created a general list of what my portion of the project needs to achieve: • A Graphical User Interface (GUI) that will run on either a phone or a laptop. The choice of either a phone or a laptop will depend on whether I use Bluetooth or a wireless connection to connect to the microprocessor. This is still to be determined at a later time. • The GUI needs to successfully interact with the microprocessor to steer and control the vehicle. • The GUI needs to be relatively simple and easy to use.

Outputs created	The above list of requirements and functionalities is what has come out of this activity.
System Integration Considerations	Since my team has an Industrial Engineer, speed needs to be considered when connecting the GUI to the vehicle. Should the GUI display the current speed and time? If so, how will I connect to and get data from the accelerometer? Should the GUI be able to speed the vehicle up or not?
Challenges/Lessons learned	My Electrical Engineer wants to use a specific microprocessor that she has used before for the project, but I would like to use a different one that would be much easier for me to code and troubleshoot. We will need to discuss the pros and cons of each option before we make a final decision.

Identify Resources Available from Makerspace

Status	**Achieved**
Objective	Visit the MSU Makerspace and identify resources and tools that are available for our team to use throughout the project.
My time on this task	1 hour
Support team member(s) time on task	Alyssa Ferry: 1 hour Jacob Johnson: 1 hour Ryan Lane: 1 hour Mubarak: 1 hour
Visual Progress Update	These are only a few of the pictures that Mubarak took of the obstacles in the Makerspace. The rest of them are located on the team's Trello board. 

	 
Current Progress	My team and I figured out a time that we were all available, and the five of us went to the MSU Makerspace together. For most of us, it was our first time visiting the Makerspace. We spent about an hour in the Makerspace looking at all the tools and talking to Matt about the available resources. Matt talked to Alyssa and I extensively about the different microprocessors that are available for purchase, and the pros and cons of the different options. Jacob and Ryan examined all of the vehicles that were there, and seemed to get some good ideas about implementation and building our team's vehicle. Mubarak took lots of pictures that we can use for future reference and documentation.

Outputs created	Our team became more familiar with the Makerspace and was able to identify all of the different tools we could use. We also were able to look at all of the obstacles available for the course and start thinking about which ones we want to use.
System Integration Considerations	N/A
Challenges/Lessons learned	The biggest challenge for this activity was figuring out everyone's schedules to find a time that was available for everyone to go. This will probably be an issue for the entire semester, since we all have lots of other time commitments.

