System Specification

Go-Bot Cart for Mary Bridge Children's Therapy Unit at Good Samaritan

By: Reagan Stovall (reaganstovall@gmail.com)
Jesse Wiklanski (jswiklanski09@gmail.com)
Andrew Gates (ag8833@uw.edu)
In conjunction with Professor Robert Gutmann.

Version: 1.0

Revision History Block

Version 0.1: Changes made include,

- Initialization and formatting of all portions of the document

Version 1.0: Changes made include,

- Each topic and subsequent topics of the System Test Plan has been filled out
    - 1. Purpose
    - 2. Scope
    - 3. System Overview
    - 4. System Architecture
    - 5. Testing
    - 6. References
    - 7. Appendix

1. Purpose

1.1. What is the point of this specification? What does it describe? Who is the intended audience?

This specification is intended for anyone who wants to understand how the Go Bot Cart works at a high level. This is not intended for the children who will be using the system, but more so for the owners of the Go Bot who will want to understand how it is assembled in order to replicate, modify, or fix the current design.

2. Scope

2.1. Identify product to be described.

The product to be described here is a retrofitted Go Bot Cart. This cart has been modified to allow for control of the cart through various controllers such as a digital Atari style controller, a Wii Nunchuck controller, a wireless controller, the attached joystick, or various single input buttons. This will allow children of various disabilities or parents themselves to drive the cart, giving the child more freedom than they would normally have in a cart like this.

2.2. Explain what the overall system does and how it interacts with users or environment.

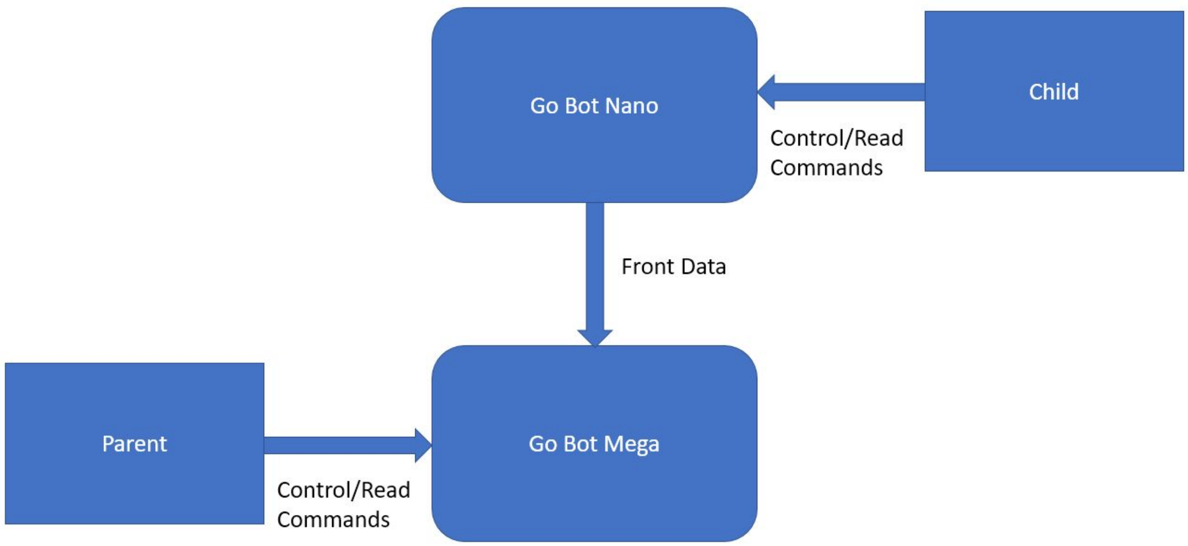
The system itself will interact with the user through a variety of controllers. The child who will be using the cart can control the cart via the joystick, or by plugging in an Atari style controller, or various single input buttons. The parent who will be monitoring the child in the cart can override the child via an Atari style controller, or a Wii Nunchuck controller that can be plugged into the back.

2.2.1. Describe the benefits of the system.

This system offers the child control of their environment via this cart where they may not have had control before. With the retrofitted cart a child of multiple disabilities can operate this cart when they may not have had the fine motor skills to do so before. This cart will also allow them to stand up and control the cart, or sit down and control the cart. The user can place a single input button anywhere on the cart so that the child can control the cart using their head, body, feet, arms and etc.

3. System Overview

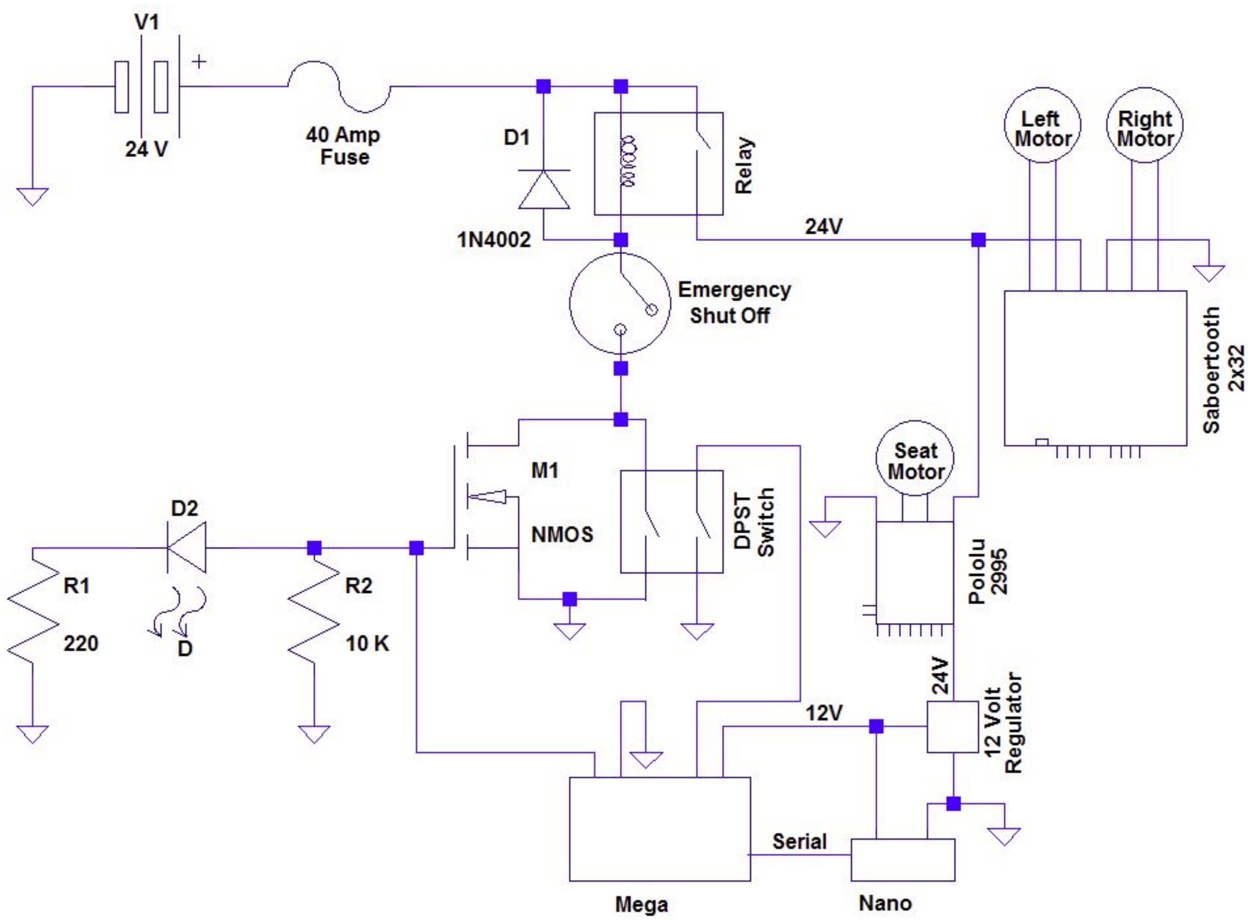
3.1. Introduce the system context.



This is the overall system context for the cart. The front controller box will contain an Arduino Nano that will collect all of the data from the front joystick, front joystick jack, side button jacks, seat switch, and miscellaneous button. It will send all data to the back Arduino Mega which will process the data as well as data that it receives from the parent's Atari controller or Wii Nunchuck controller.

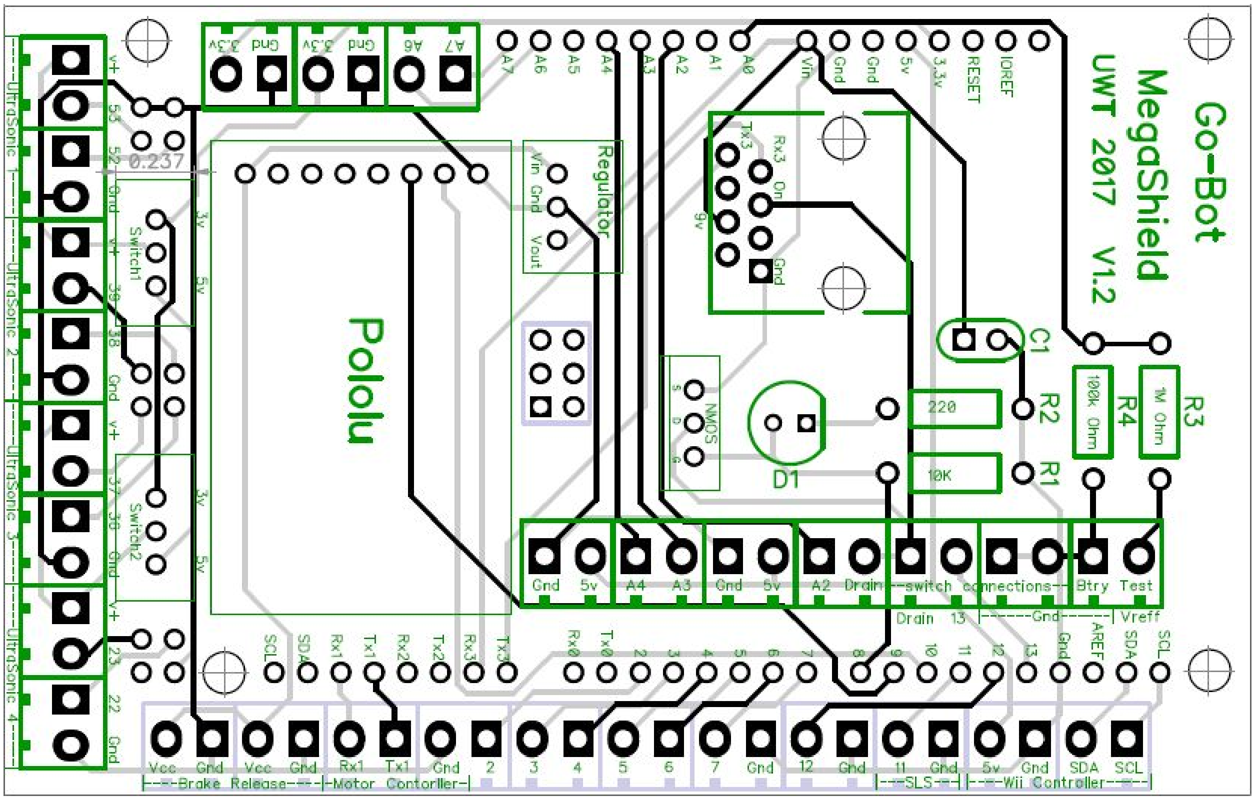
4. System Architecture

4.1 Power System



4.1.1. Describe the architecture as shown in your architecture diagrams.

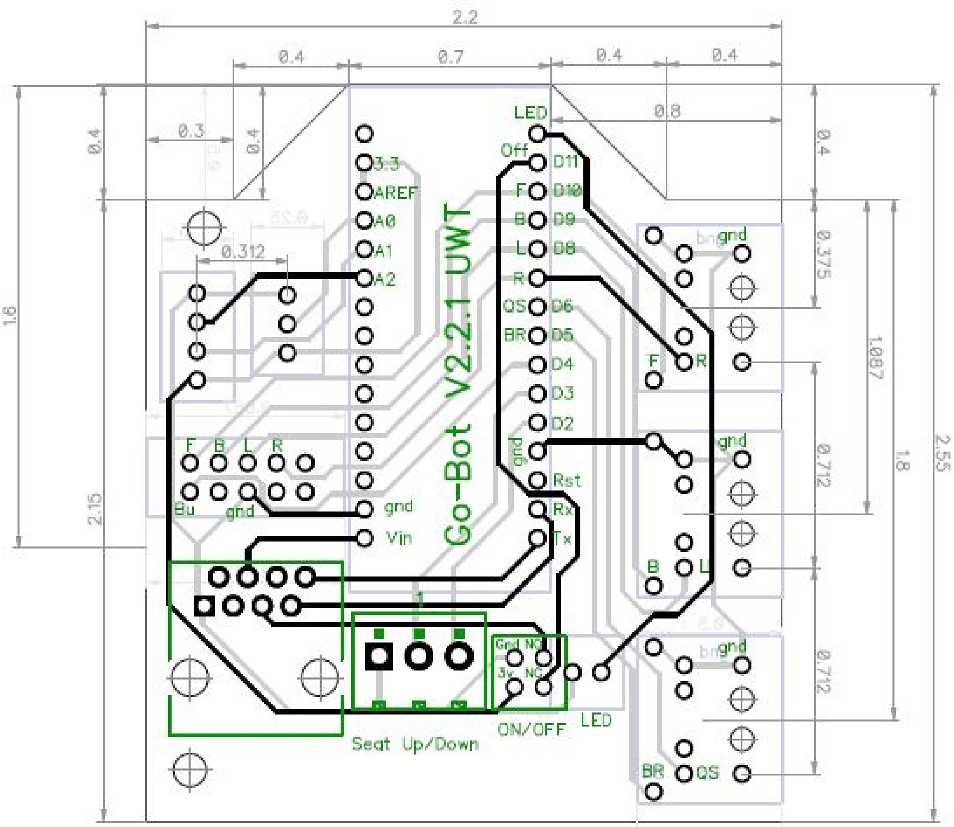
4.2. Mega Shield



4.2.1. Describe the architecture as shown in your architecture diagrams.

The Mega shield acts as hub connection for all the various inputs.

4.3. Nano Shield



4.3.1. Describe the architecture as shown in your architecture diagrams.

4.5 Use one complete section to describe the software design you used in your system. Remember that this section should have sufficient detail to reproduce your system.

Our software design uses a server/client type approach between the two Arduinos, the Mega serves as a server to the Nano who serves as a client. Through a Serial connection between the two the Nano will send all of the data it processes from the front controller box to the Mega. The Mega will then read the front data, read the rear data, process the data for the seat, process the data for the cart, and then send the data to the Sabertooth controller for the cart motors. It will also process and check a variety of other things such as checking to ensure the data from the front was sent correctly, checking the current battery level, checking to see if the cart has been on too long, checking to see if the system shutdown button was pressed, and etc. All of this happens in a loop that will cycle in conjunction with a loop on the Nano side to receive new data every cycle.

5. Testing

5.1. Describe how you tested the system.

We tested the system rigorously through various testing procedures. We tested every controller combination that is possible in the system, using all of them and ensuring that the data was accurate and the cart controlled properly. We also tested every switch, button, LED to ensure that they activated properly.

5.2. Describe the procedures necessary to test at relevant stages of development. For example, did you test individual hardware components before assembly? How? Did you unit test your software? How?

Through various iterations of our shields we tested every individual hardware components when disassembling and reassembling each shield. We ensured that all of the connections worked properly and that everything was drawing the correct amount of voltage and current. For our software we tested it by printing every output and verifying that the outputs changed accordingly to whatever test we were running on our cart at the time.

5.3. Describe test cases you used that are required to ensure full functionality.

The main test case in order to ensure full functionality is testing to make sure every single output, input, and the connection between the Nano and the Mega works properly. If this is true and every controller works properly as well as the back controllers properly overriding the front controllers, then full functionality is ensured.

5.4. Describe conditions for expected failure. Did you test those and how?

Conditions of expected failure can occur when -

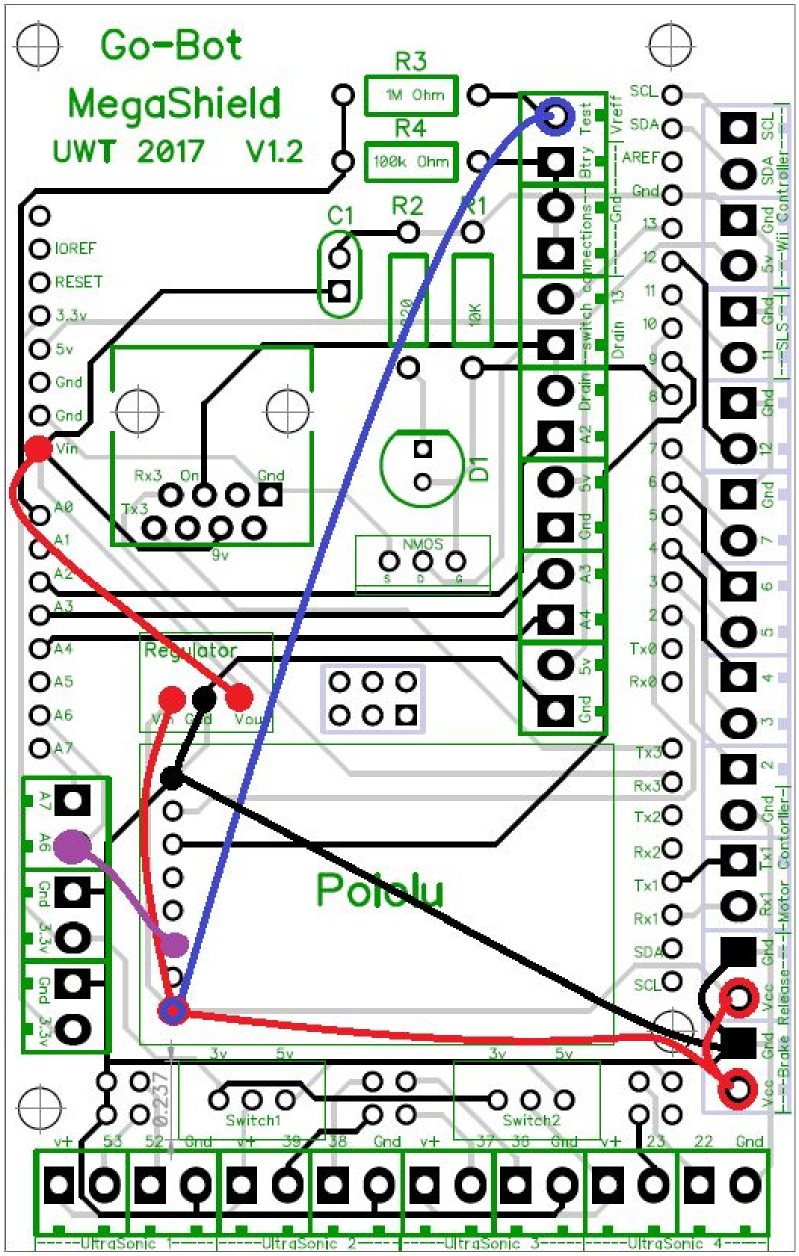
- The system is overloaded with data from 3+ controllers at the same time.
- The CAT5 connection from the Nano to the Mega is disconnected.
- The connection from the Nano to the Mega stalls.
- The seat gets stuck trying to raise or lower.

We evaluated all of these expected failure points and tested each of these to find out how to replicate them. When we were able to replicate and pinpoint one of these expected failures we put in safeguards against them, such as limiting the controlls when 3+ controllers are inputting data at the same time. Shutting down the system when the CAT5 connection is disconnected or the connection from the Nano to the Mega stalls. And measuring the current output from the seat motor controller to shut down the system if the current gets too high, meaning the seat is stuck trying to raise or lower.

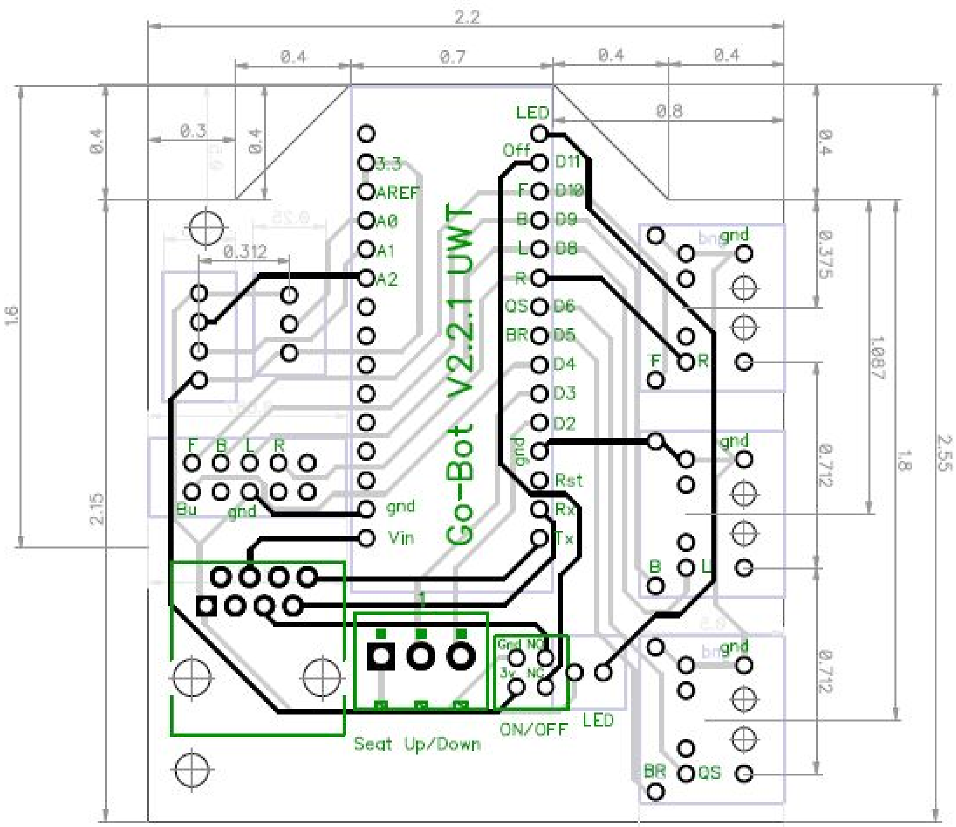
6. References

Arduino Mega - https://www.arduino.cc/en/Main/arduinoBoardMega
Arduino Nano - https://store.arduino.cc/usa/arduino-nano
Sabertooth Motor Controller - https://www.dimensionengineering.com/products/sabertooth2x32
Pololu Seat Controller - https://www.pololu.com/product/2995
1N4002 Diode - http://www.onsemi.com/PowerSolutions/product.do?id=1N4002
Voltage Regulator - https://www.recom-power.com/pdf/Innoline/R-78Cxx-1.0.pdf
Arduino Nunchuck Library - https://github.com/GabrielBianconi/ArduinoNunchuk
24 Volt 50 Amp Relay -
https://www.waytekwire.com/item/75561/Bosch-Tyco-332002250-Power-Relay/
Go-Bot Documentation -
http://www.mobilityfordiscovery.net/415.html
https://www.google.com/patents/US5701968
http://rehabrobotics.org/icorr1999/papers/papers/wright-ott.pdf

7. Appendix


Go-Bot
MegaShield
UWT 2017 V1.2
R3
1M Ohm
R4
100k Ohm
C1
R2
R1
D1
NMOS
IOREF
RESET
3.3v
5v
Gnd
Gnd
Vin
A0
A1
A2
A3
A4
A5
A6
A7
Rx3
On
Gnd
Tx3
9v
Regulator
Vout
Pololu
SCL
SDA
AREF
Gnd
13
12
11
10
9
8
7
6
5
4
3
2
Tx0
Rx0
Tx3
Rx3
Tx2
Rx2
Tx1
Rx1
SDA
SCL
Switch1
Switch2
3v
5v
v+
53
52
Gnd
39
38
37
36
23
22
UltraSonic 1
UltraSonic 2
UltraSonic 3
UltraSonic 4
Wii Controller
SLS
Motor Controller
Brake Release
Vcc
Btry
Test
Vreff
switch connections
Drain

Figure



LED
Off
D11
F
D10
B
D9
L
D8
R
QS
D6
BR
D5
D4
D3
D2
gnd
Rst
Rx
Tx
3.3
AREF
A0
A1
A2
gnd
Vin
Go-Bot V2.2.1 UWT
F B L R
Bu
gnd
Seat Up/Down
ON/OFF
LED

Figure

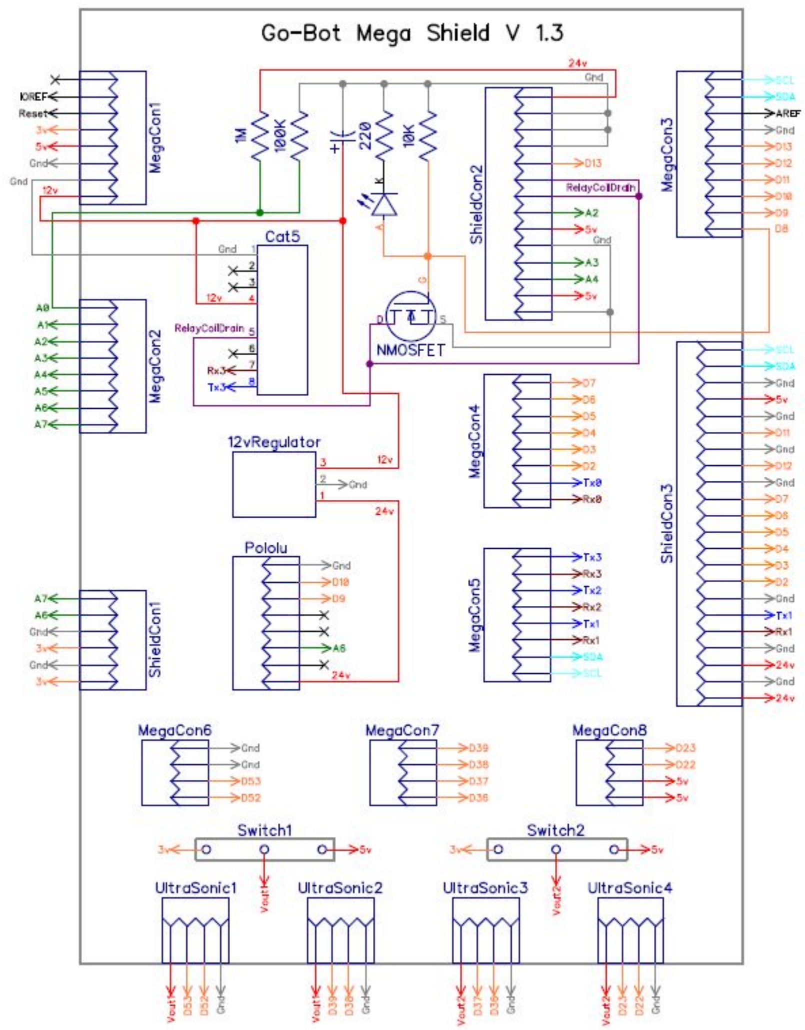


Figure X. MegaShield Schematic

Name	Description
220, 100K, 10K, 1M	Shut Off Circuit Resistors Battery Voltage Referance
12vRegulator	12vRegulator
C1	1f capacitor
Cat5	Cat5 Connection Port
D5	Basic LED
MegaCon1, MegaCon2, MegaCon3, MegaCon4, MegaCon5, MegaCon6, MegaCon7, MegaCon8,	Mega Power Header Mega Analog Header Mega Digital[D8,...,D13] & I2C Mega Digital[D2,...,D7] & UART Mega Communications Mega Gnd & Digital[D52,D53] Mega Digital[D36,D37,D38,D39] Mega 5v & Digital[D22,D23]
NMOSFET	Nchannel MOSFET Transistor
ShieldCon1	Low Power Analog Connections
ShieldCon2	Analog, Voltage Test, and SPDT momentary Switch Connections
ShieldCon3	Atari, Wii Nunchuck, Sabertooth Motor Controller, and 24v Connections
Switch1, Switch2	Unimplemented: controls power to UltraSonic1-4
UltraSonic1, UltraSonic2, UltraSonic3, UltraSonic4	Unimpleneted: Hardware to support an array of different UltraSonic devices for future possible updates

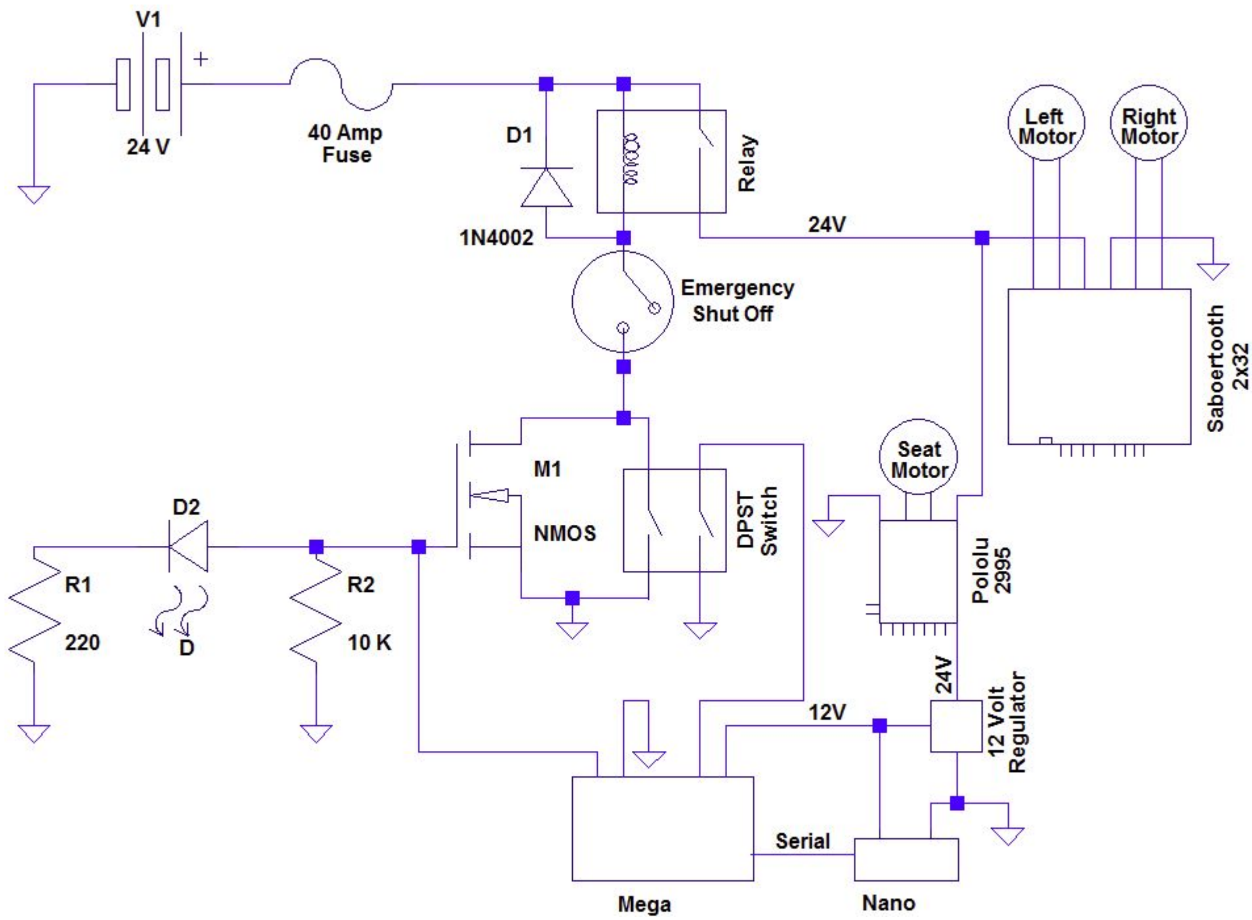
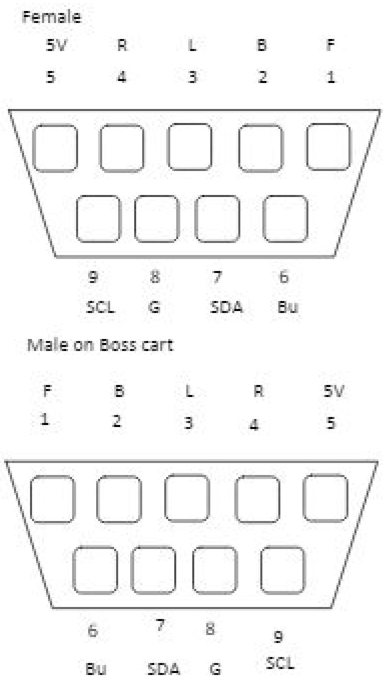


Figure Y Power Schematic


Female
5V
R
L
B
F
5
4
3
2
1
9
8
7
6
SCL
G
SDA
Bu
Male on Boss cart
F
B
L
R
5V
1
2
3
4
5
6
7
8
9
Bu
SDA
G
SCL

Figure