Automation Station

Database: Andrew Klonitsko
Hardware: Andrew Gates
User Interface: Reagan Stovall

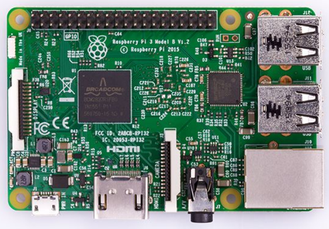
OVERVIEW

Our project is designed to be a automation station that can be interfaced through a simple phone app, an emailed dataset, or through an capacitive touch screen. This automation station will allow for fine control of a range of sensors and outputs for the average Homebrewer, Vintner, Biologist, or anyone who wants to automate some aspect of their life but does not have much experience with a project like this. The user will be able to simply plug in up to 8 analog and 8 digital sensors and allow them to modify the settings via the touch screen for quick and easy setup of any automation system that they want. They will be able to control 4 different single directional motors, or 2 different bi-directional motors as well as 4 AC relays and 4 12V relays. These can all be controlled through functions that the user creates based on the sensors that they are using. All of the sensor data will also be output to the database so that it can be easily viewed and graphed via the phone app. They will also have access to additional GPIO ports to allow for drop on modules to extend the amount of sensors, motors, and relays that they can possibly control.

CONTROLLER

Raspberry Pi 3

- 28 GPIO pins
- Bluetooth
- Wi-Fi

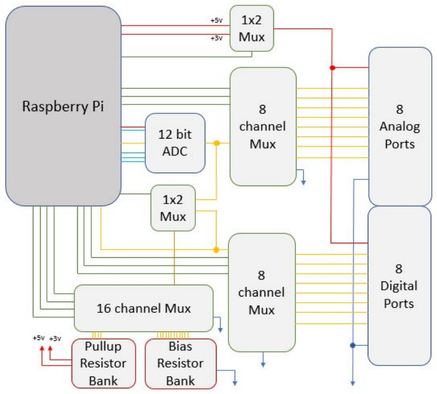


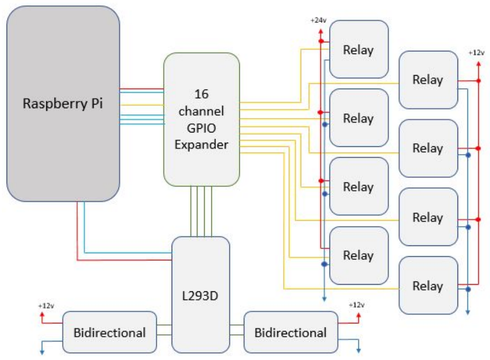
We chose the Raspberry Pi 3 for it's low price point and versatility. With this users will be able to display and interact with the sensors and output devices in a intuitive and tactile way. It's processing power will make generating calibration functions a simple onboard process, and it's array of easy to use GPIO's will make the device control simple and reliable.

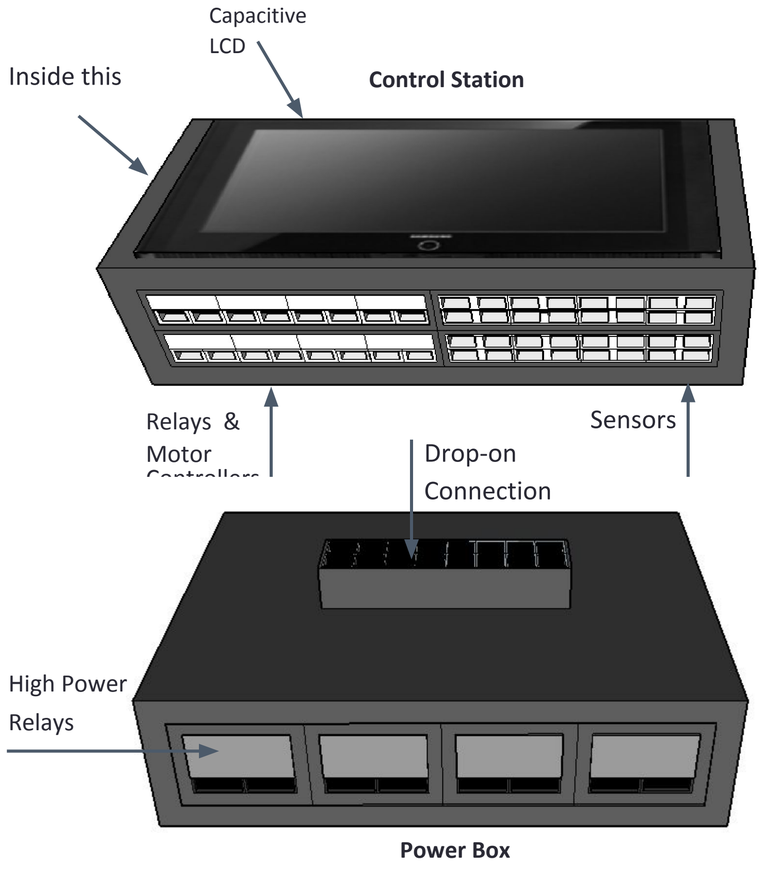
OBJECTIVES

- **Automation Station**
  - A device for reading sensors and controlling a small number of output devices.
  - Drop on attachments for expanding the controller to handle more devices with a wider range of power needs.
- **Interface**
  - Preset Templates for basic Automations/Monitoring projects
  - Sensor and output device selections for easy control and setup
  - Outputs to both phone and LCD screen.
- **Database**
  - Save sensor data, templates, and previous builds for easy access

HARDWARE DESIGNS

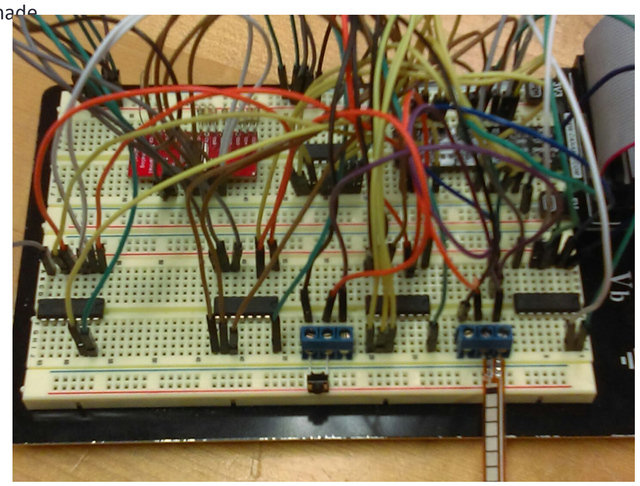




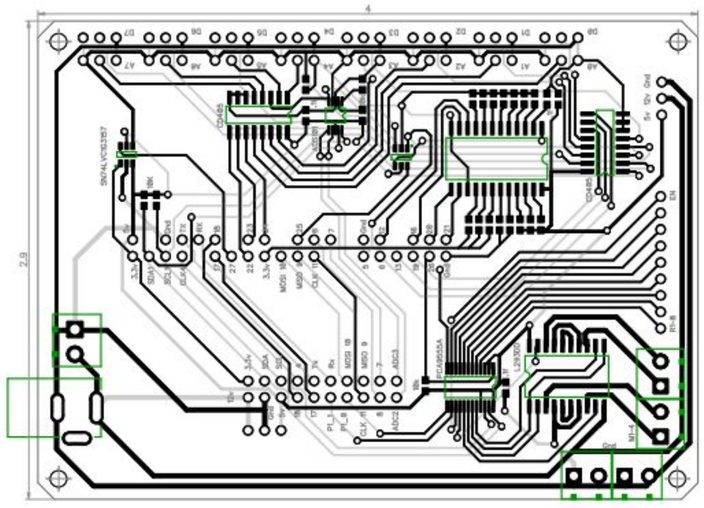


TESTING

To see if our design was applicable we first needed to test out the basic concept on a circuit board first, before we jumped to PCB production. We used the circuit below to test and evaluate our design before we had our PCB made.



Once we thoroughly tested this design we designed the PCB layout shown below, which is what we are currently using in our final product.



SUPPORT

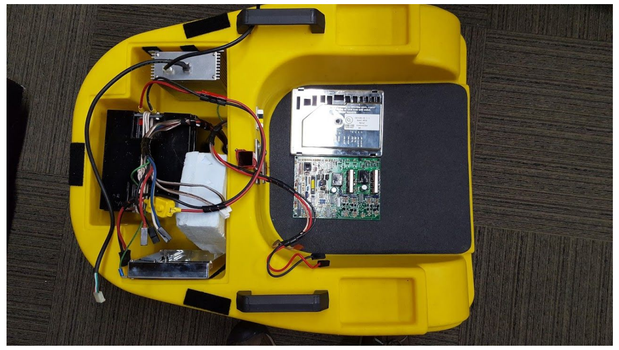
Recipients of the $2500 NSF I-Corps (National Science Foundation) grant presented on behalf of CoMotion and the Buerk Center for Entrepreneurship.

Go Bot Cart

Members: Jesse Wiklanski,
Andrew Gates,
and Regan Stovall
Overseer: Professor Robert Gutmann
For: Mary Bridge Children's Therapy Unit at Good Samaritan

OVERVIEW

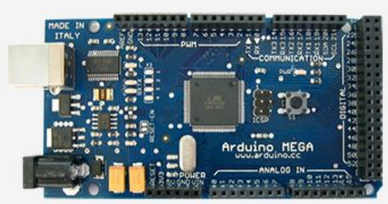
Our project was to take the cart shown, and retrofit it to be used with a variety of controllers. Implementing a new system would allow the child to control the cart with analog controllers, digital controllers, or simple switches that they can control with presses from their head, hands, feet, and etc. The child can be either sitting or standing up while controlling the cart. It would also allow the parent to have a parental override to block out the child's controller when the parent was controlling the cart.



CONTROLLER

Ardunio Nano: 22 Digital Pins, 6 PWM Pins, and 8 Analog Pins

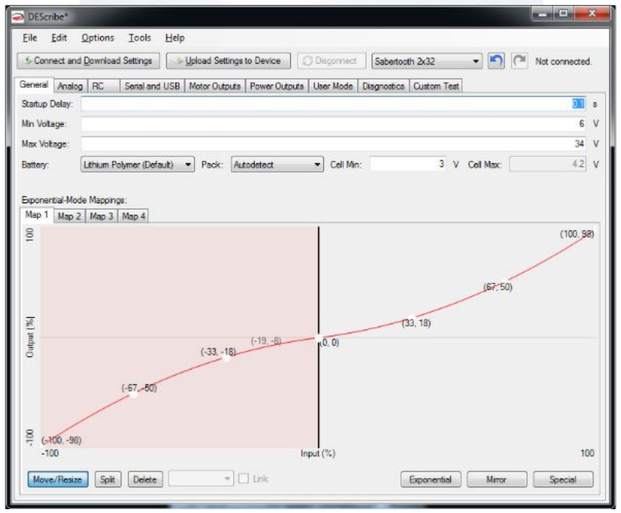
Used to read the front controller box inputs from the analog joystick, the speed potentiometer, the seat switch, and the digital switches. All of these values were sent across from the Nano to the Mega using a CAT5 cable and Serial communication



Ardunio Mega: 54 Digital Pins, 15 PWM Pins, and 16 Analog Pins

The Arduino Mega was used to collect all of these values, as well as reading the inputs for the controllers plugged into the back, if any. The Mega would then take the corresponding controller and switch data and send all of the data to the corresponding motor controller which controls the movement of the cart and the seat.

MOTOR CONTROLLERS



Sabertooth 2x32:

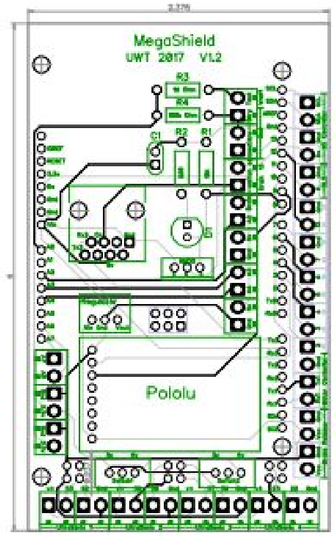
The Sabertooth 2x32 motor controller is used to control the main motors that drive the wheels of the cart. This motor controller is controlled via a Serial communication line from the Arduino. Sending over a command and then a value allows us to control the drive and turn speeds simultaneously. The Sabertooth controller's output can also be controlled by the DEScribe software, which allows for changing of acceleration curve, response time, current output, and more.

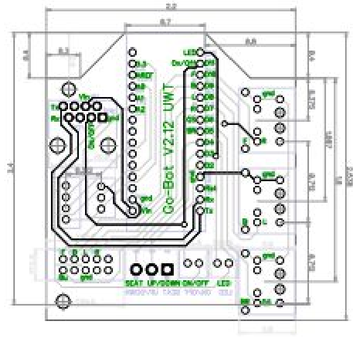
Pololu G2 High-Power Motor Driver 24v21

The Pololu G2 High-Power Motor Driver is used to control the raising and lowering of the seat. Based on different combinations sent to the PWM and DIR lines we can control the direction of the seat, either raising or lowering it.

HARDWARE

We created a custom PCB that we used as a shield for our Arduino Mega. This shield allowed us to set up our power system, Pololu motor controller, controller jack, and the CAT5 connection from the Arduino Nano all on one simple shield. The schematic that we designed and use is shown below. We also created a custom PCB for the Arduino Nano as well. This allowed us to interface all of our switches, joystick, and potentiometer to the Nano to allow for communication to the back Mega through the attached CAT5 connector





CONTROLLERS

Our cart was designed to be used with a variety of controllers shown below. Including analog controllers, digital controllers, wireless controllers, and simple switches. All of them allowing the user to control the cart with ease.