MCGREEN 2.0
Senior Showcase

MCGreen 2.0 – Robot Operating System (ROS)

Raj Joshi

Aditya Chandaliya

MCGreen 2.0 – Games Team

Gil Shteyman

Nishil Patel

MCGreen 2.0 – Mobility

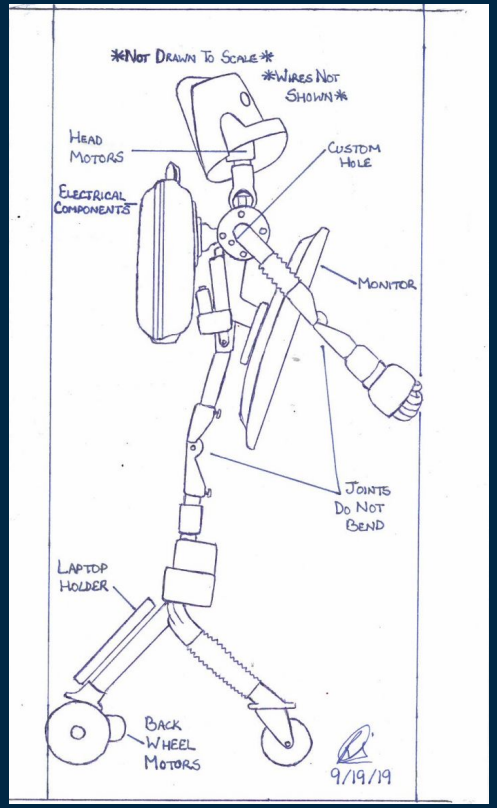
Mihir Rastogi

Maxim Grebinskiy

Sanjana Gunda

Problem Addressed

Updating the MC Green 1.0 robot for the Middlesex County Improvement Association (MCIA)



Applications

- The robot will be used by the Middlesex County Improvement Authority to educate children in Middlesex County about the environment and sustainability

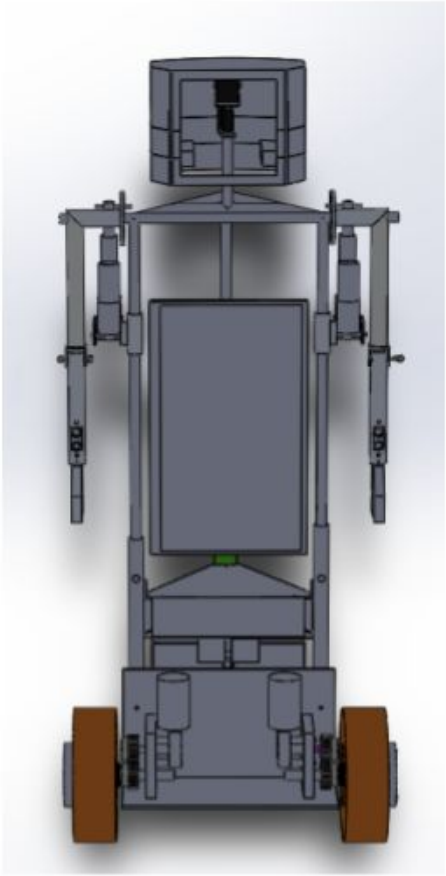
- Games will create an interactive experience for children to learn about proper recycling and environmental sustainability practices

End User Customers

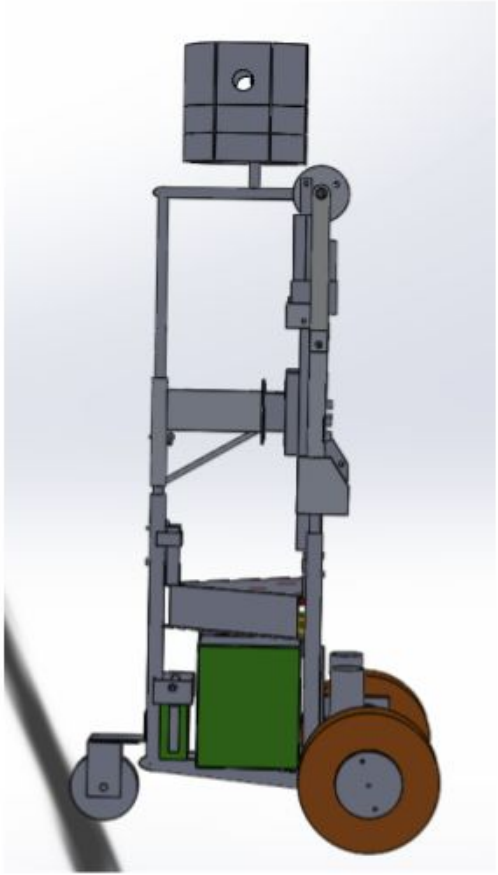
- The Middlesex County Improvement Authority:
  - A public entity created to improve the quality of life of Middlesex county residents
  - Heavy focus on environmental awareness
  - Offer free programs to schools, civic and business organizations, senior citizen clubs and other groups in which they educate and entertain using the MC Green 2.0 Robot
- The robot will interact with children in Middlesex County
- User guide
  - Information regarding how to turn on/off the robot, how to dismantle the robot, maintenance/troubleshooting, etc.

Our Solution

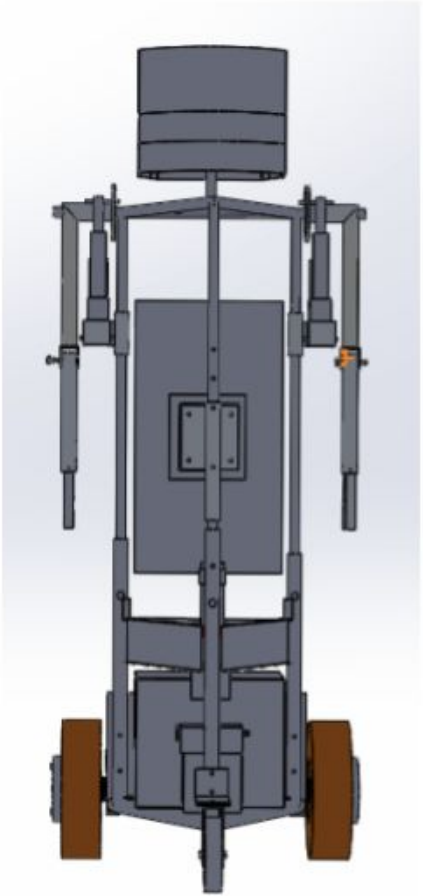
- ROS interfaces computer and hardware
- Games team creates a good user experience through interactive games and faces
- Mobility and Power team ensures that the MCGreen 2.0 robot is able to move appropriately regardless of the surface type and that the robot's battery life lasts for a duration of 6 hours.
- Overall robot specifications: 1.8 mph, maximum incline of 15 degrees, 80 pound weight limit



Front View



Side View

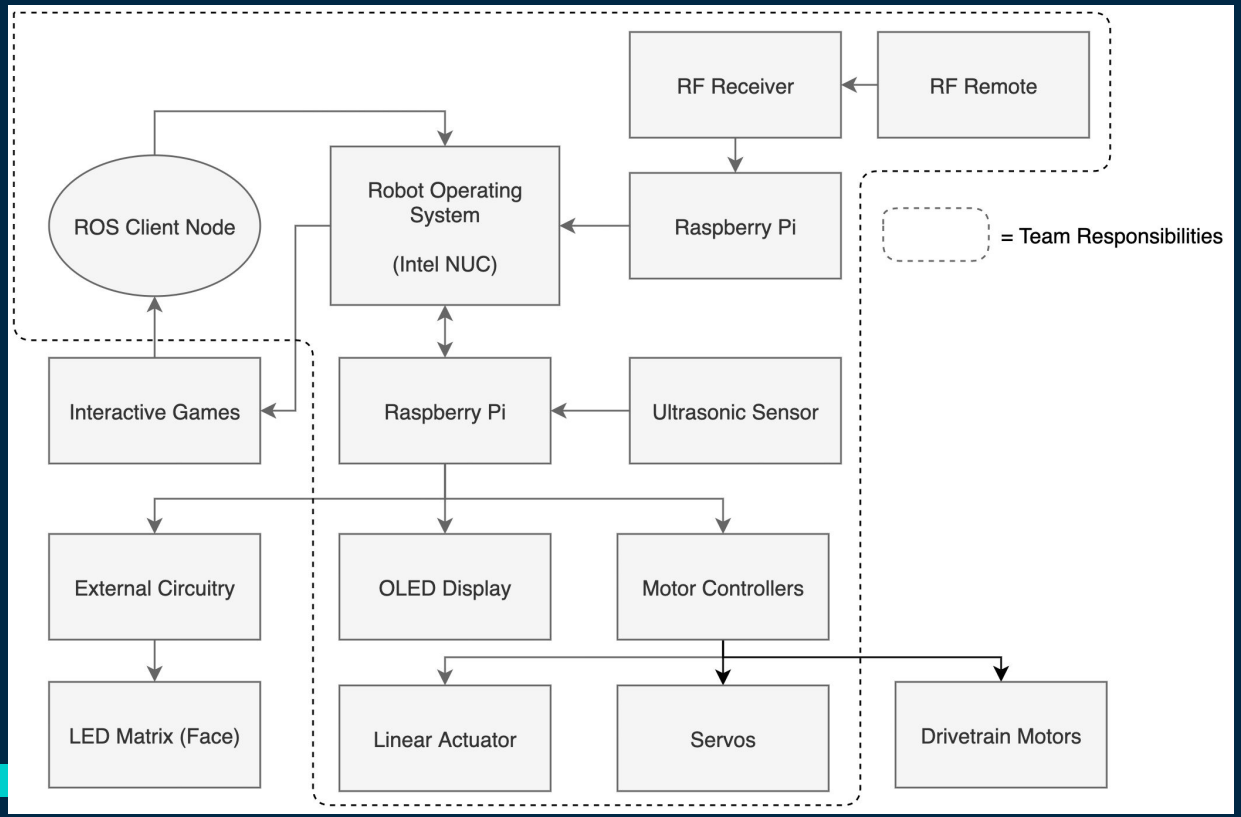


Back View

COVID, Transfer of Knowledge

- Greatly impacted by COVID-19
  - Delays in the manufacturing of the frame
- Response to the problem
  - Despite not being able to get our hands on the physical robot, we have done all of the planning and all of software and hardware is completed and ready to be mounted on the robot when received.
  - Created a detailed plan for what steps should be completed to put the robot together after the frame arrives.
  - To facilitate a transfer of knowledge to the juniors we have created extensive documentation for each team regarding part selections, calculations, how to use different tools, and any other information necessary

ROS


RF Receiver
RF Remote
ROS Client Node
Robot Operating System
(Intel NUC)
Raspberry Pi
= Team Responsibilities
Interactive Games
Raspberry Pi
Ultrasonic Sensor
External Circuitry
OLED Display
Motor Controllers
LED Matrix (Face)
Linear Actuator
Servos
Drivetrain Motors

ROS Team Updates

Received (June 2020)

- Messy file structure, extraneous and outdated files
- Completed code for:
  - serial signal receiver
  - one ultrasonic sensor
  - most of the internal processing code (mode selection, safety toggle manipulation)

Now (June 2021)

- Updated file structure; easier to understand and more streamlined
- Updated detection method for ultrasonic sensor and created a node for the second sensor
- Integrated games with the NUC
- Completed code for drivetrain and servo motors
- Added a physical startup mechanism for the user

Games Team Updates

Received (June 2020)

- 5 buggy games with unmanageable code
- Games played on horizontal monitor
- Games played with keyboard and mouse
- No integration with ROS
- No Face matrix hardware or code

Now (June 2021)

- 5 games with structured code and easy editability
- Games played with vertical monitor
- Games played with touch screen only
- Integration with ROS
- Working face matrix hardware and code

Mobility/Power Team Updates

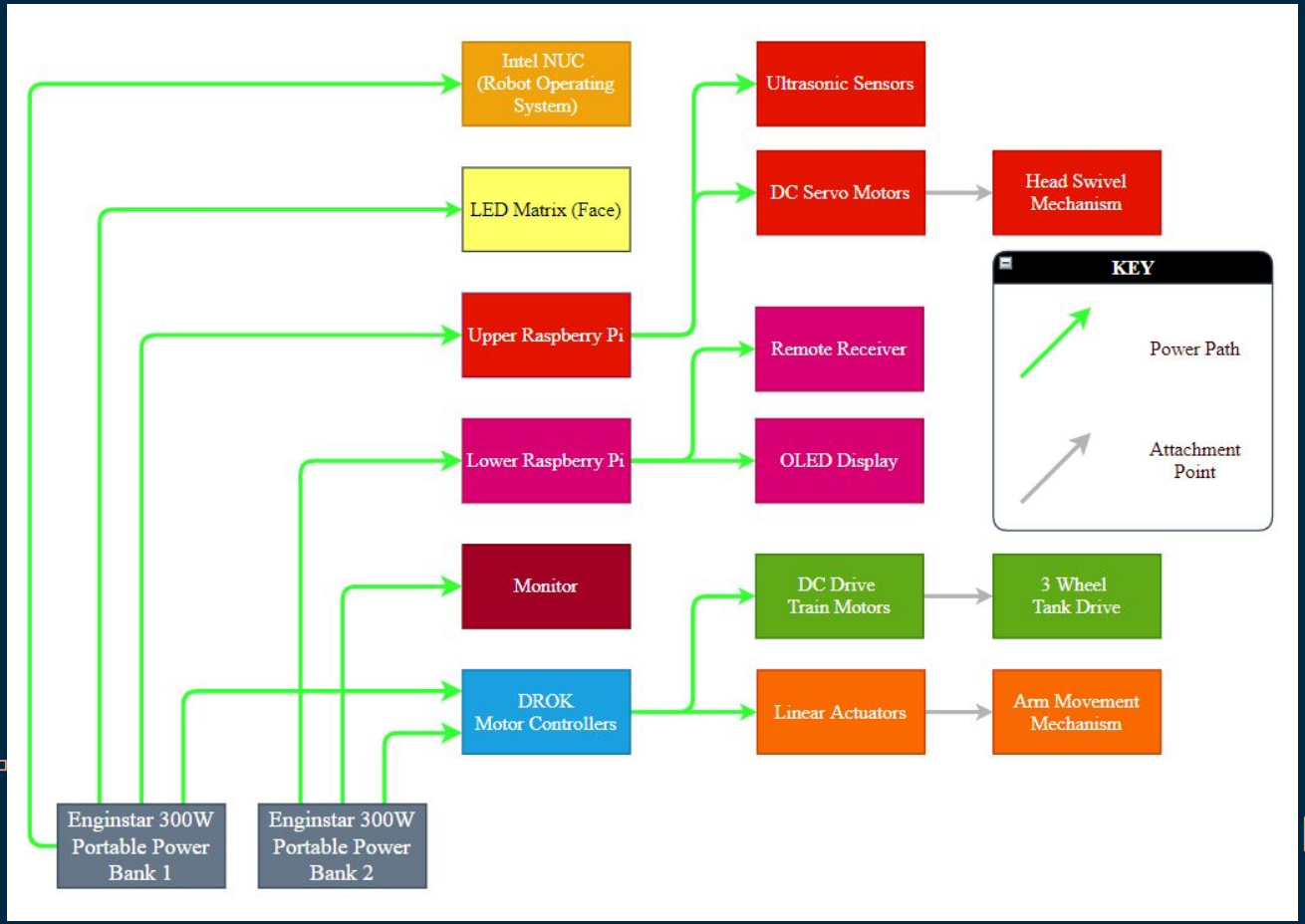
Received (June 2020)

- Parts specifications and power calculations: drivetrain, drivetrain motor controllers, servo motors, linear actuators

Now (June 2021)

- All components selected: battery, wires, wire connectors (ribbon cable connector)
- Complete wire schematic with all components and connections; a plan to make robot dismantlable (at the torso)
- New battery that is more reliable with a longer battery life
- Drivetrain motor code is written and has been tested

Block Diagram



Component Power Requirements

Name of Component	Quantity	Power Draw	Duty Cycle	W used every hour
Drivetrain Motor (DC)	2	60 W	40%	24 W
Linear Actuator (DC)	2	40 W	25%	10 W
Raspberry Pi Model 3 (powered by USB hub) (AC)	2	~ 10 W	100%	10 W
Monitor (AC)	1	27 W = ~ 30 W	100%	30 W
NUC (AC)	1	35 W	100%	35 W
LED Matrix (AC)	1	~ 15 W	100%	15 W

Enginstar 300W Battery

- Capacity: 298 Watt-hours
- Ports:
  - 2 AC output ports (110V 300W)
  - 2 DC output ports (12V)
  - 1 Cigarette Lighter output port (+1 DC port) (12V)
  - 3 USB output ports
- Weight: 8 pounds
- Battery 1 and Battery 2 power components such that both spend 84 Wh each hour (on average)
- Total Power Capacity Needed: (168 Wh per hour) * 6 hours = 1008 Wh → 85% efficiency results in ~ 1186 Wh needed
- Power Requirement satisfied with 2 sets of 2 Enginstar batteries

Wiring Connectors

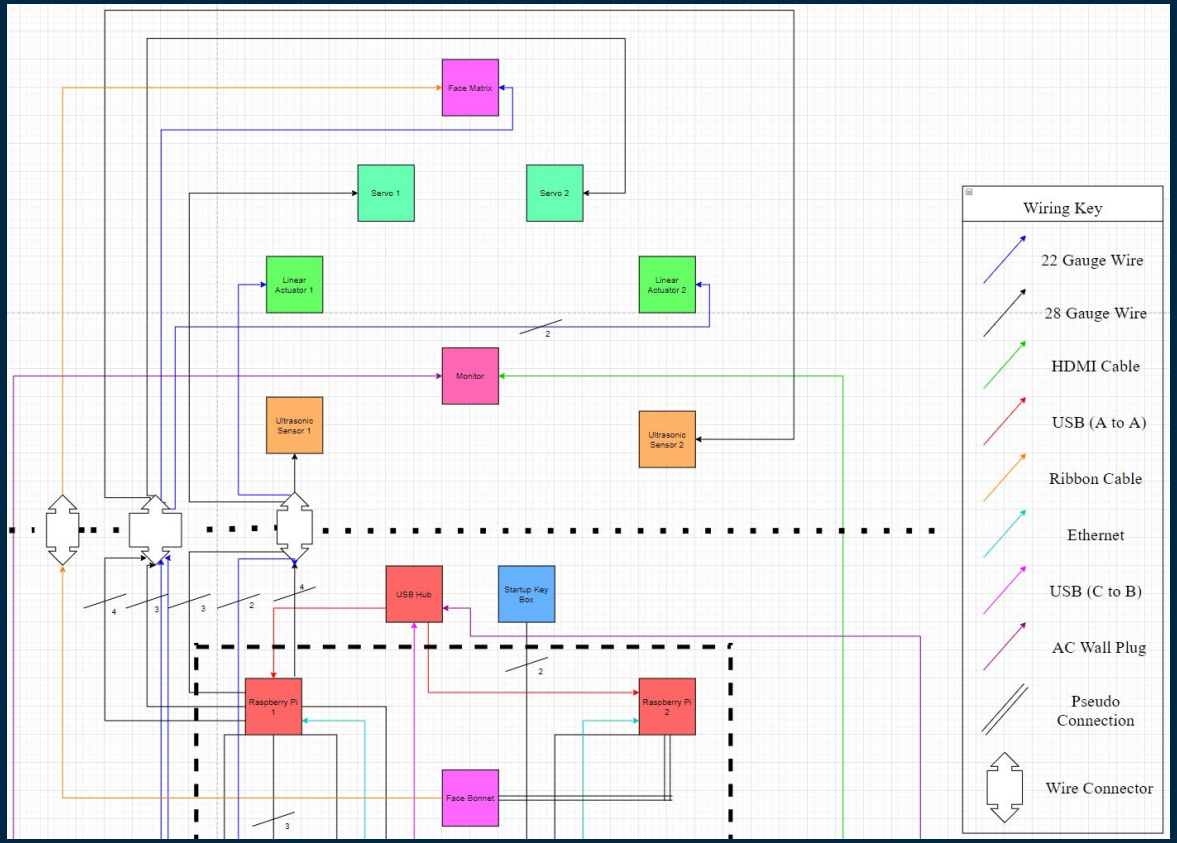
Glarks 12 pin wire connector

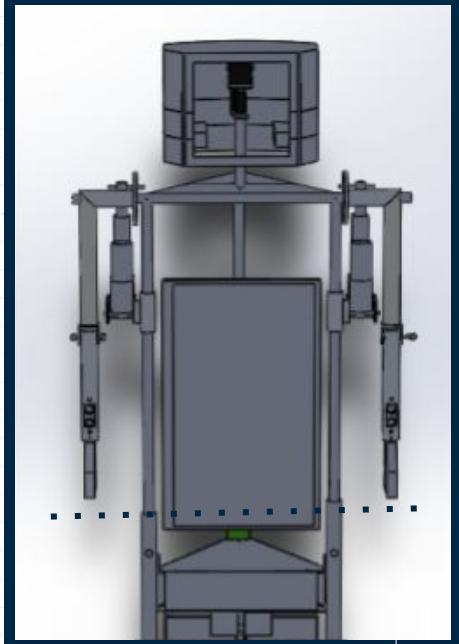
Wire crimping tool

Ribbon cable connector

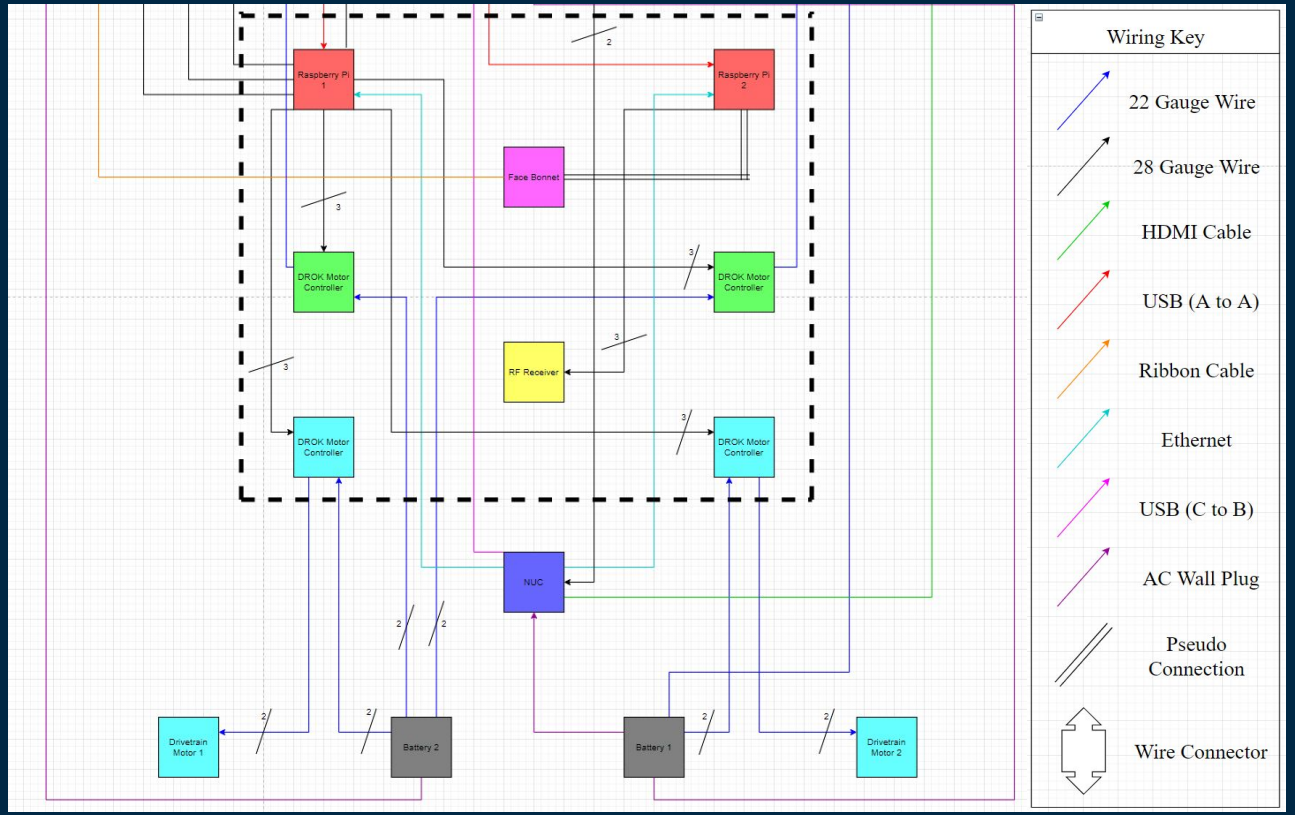
DEMO

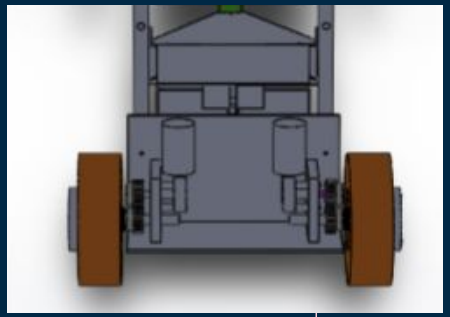
Wiring Diagram





Wiring Diagram (cont.)

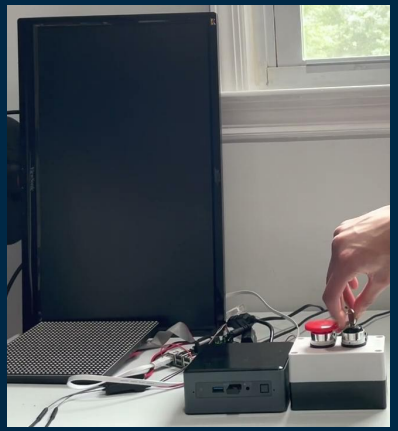


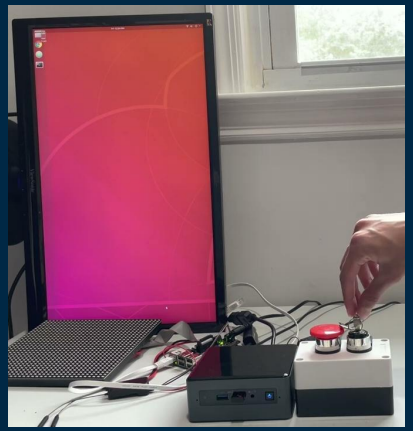


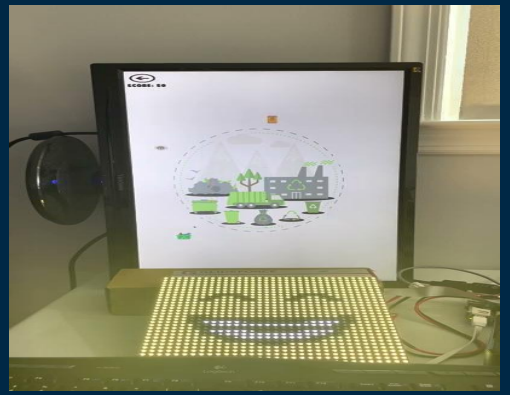
Wiring Plan

- Separated into three phases of how robot will be built and includes connections, wire type, approximate wire length, number of wires, and tools needed
- Additionally, includes detailed description of which wires connect to which pin in connectors as well as a color breakdown of which connections will be wired in what color
- https://docs.google.com/document/d/1TAzO_j_mAv-AyaRI0-6zynhUeJCJzPrbGJ07gK_Vr_o/edit

Video Demos







Questions?