

Scenery Mover v2

An Instructional Handbook Compiled by:

Steven Johnson

Casey Horton

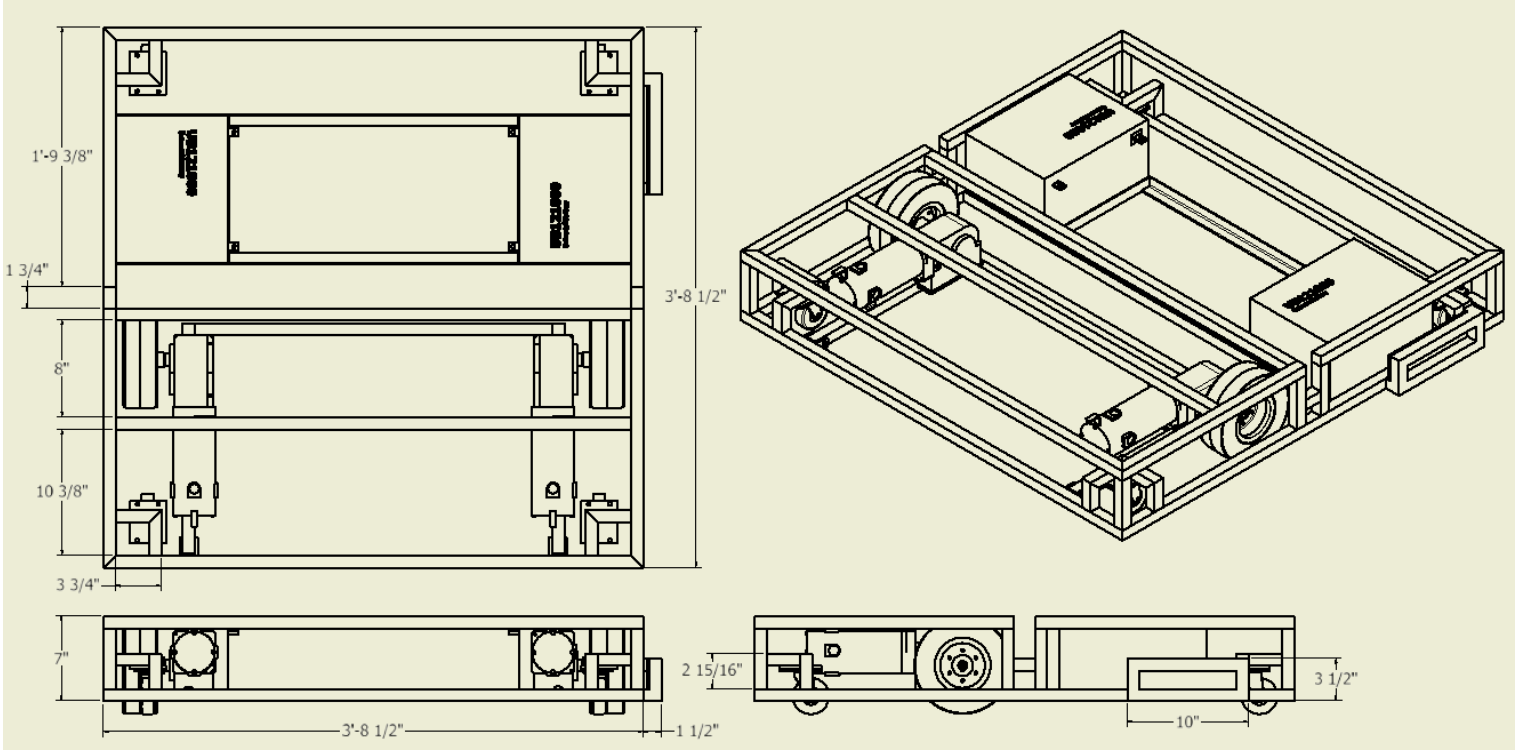
Danton Wein

Contents:

1. Equipment Layouts
 - 1.1. Technical Drawings
 - 1.2. Circuit Diagrams
 - 1.3. Progress Pictures
2. Controlling the Robot
 - 2.1. Control Setup Overview
 - 2.2. Programming Description
 - 2.3. Example Operations
3. Maintenance
 - 3.1. Parts List with Technical Specifications
 - 3.2. Inspection and Troubleshooting
4. Known Problems

1. Equipment

1.1. Technical Drawings



1.2. Circuit Diagrams

1.2.1. Wiring Color Guide

22 AWG

Red - 5V

Black - Ground input

Orange -5V input

Yellow - 3rd pin: Jack Upper limit signal input. 8th pin: Xbee Estop signal input

White - 2nd pin: Brake signal output. 7th pin: reset signal input

Blue - Arduino header: serial communication output to motor controllers. Relay header: relay signal inputs

Green - 4th pin: Jack Lower limit signal input. 9th pin: Arduino Estop signal output

10, 12, & 18 AWG

Red - 24 VDC +

Black - Ground (Negative)

Orange - 12 VDC +

Yellow - 5 VDC +

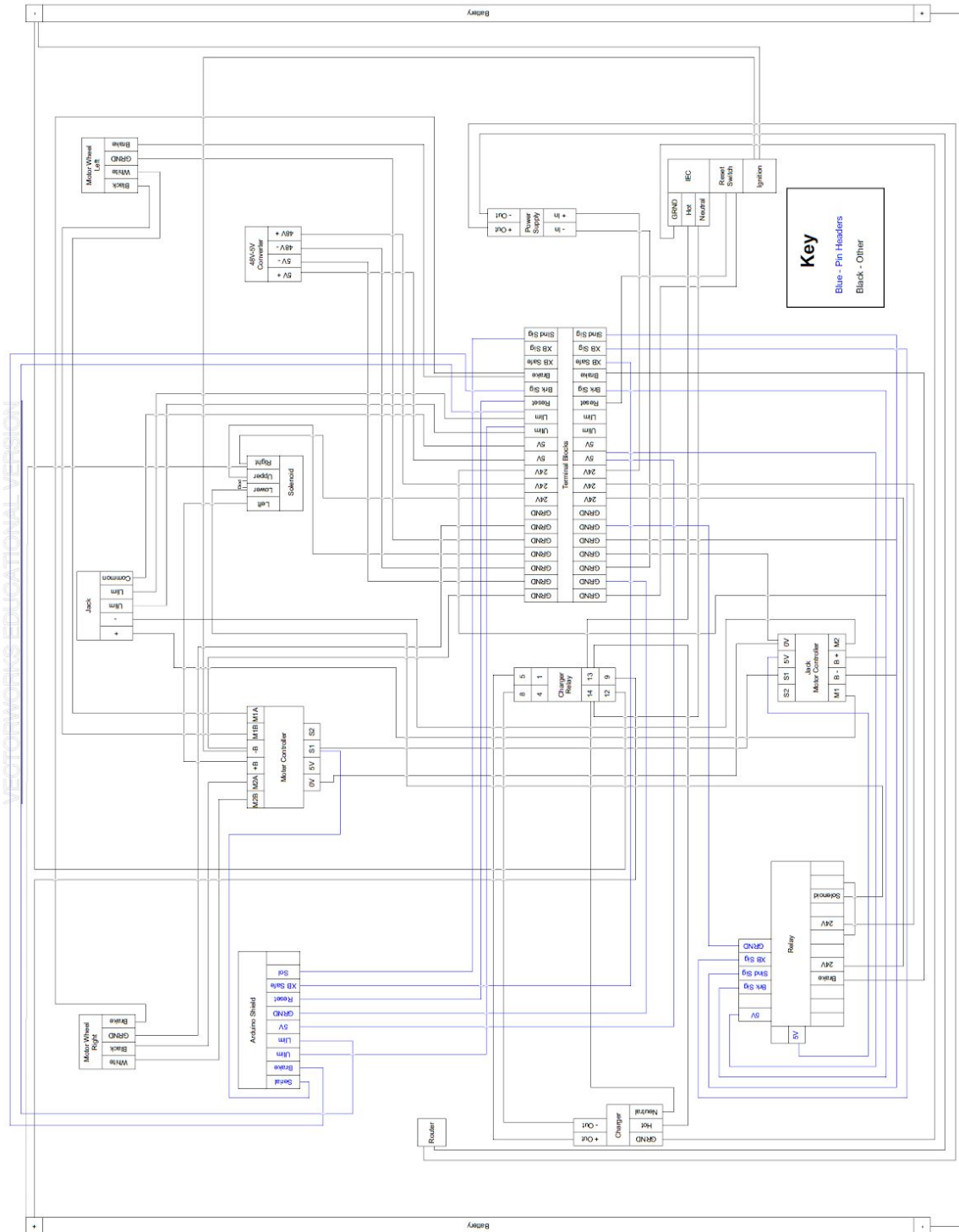
Green - Ground

Brown - Ground

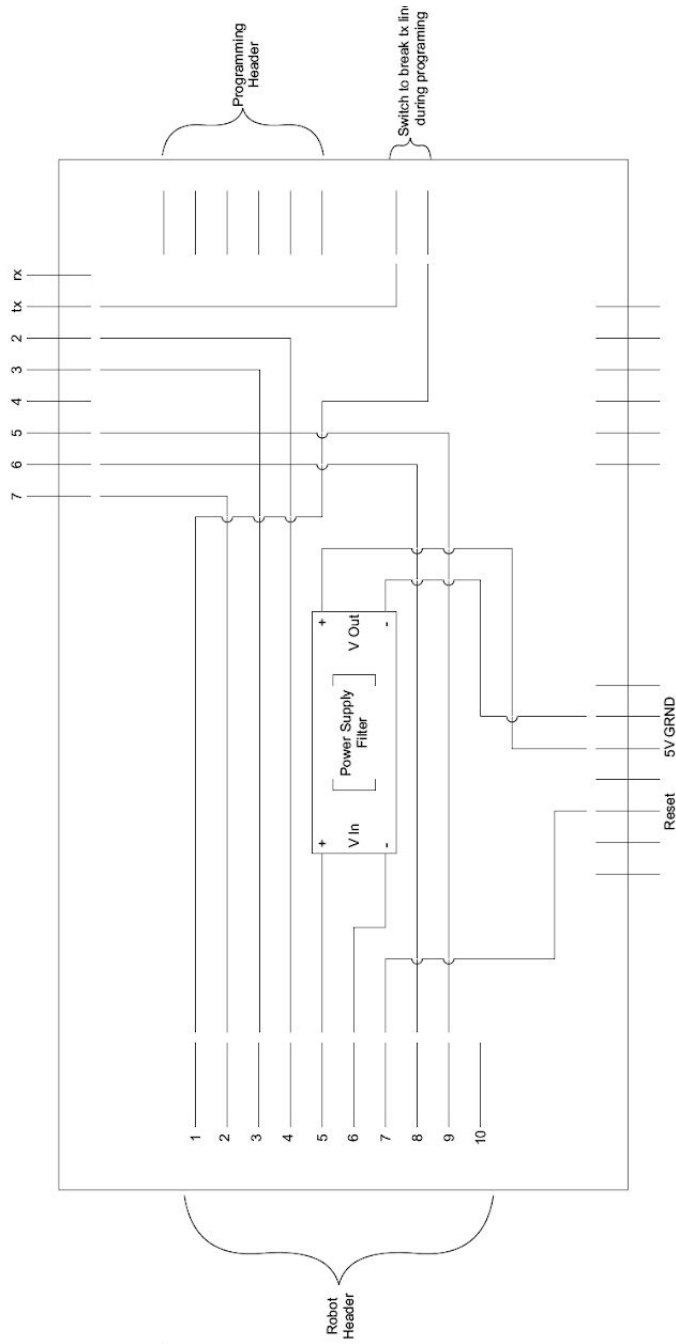
Grey - Miscellaneous (Brake Wire)

White - Miscellaneous/Legacy Wiring

1.2.2. Wiring Diagram

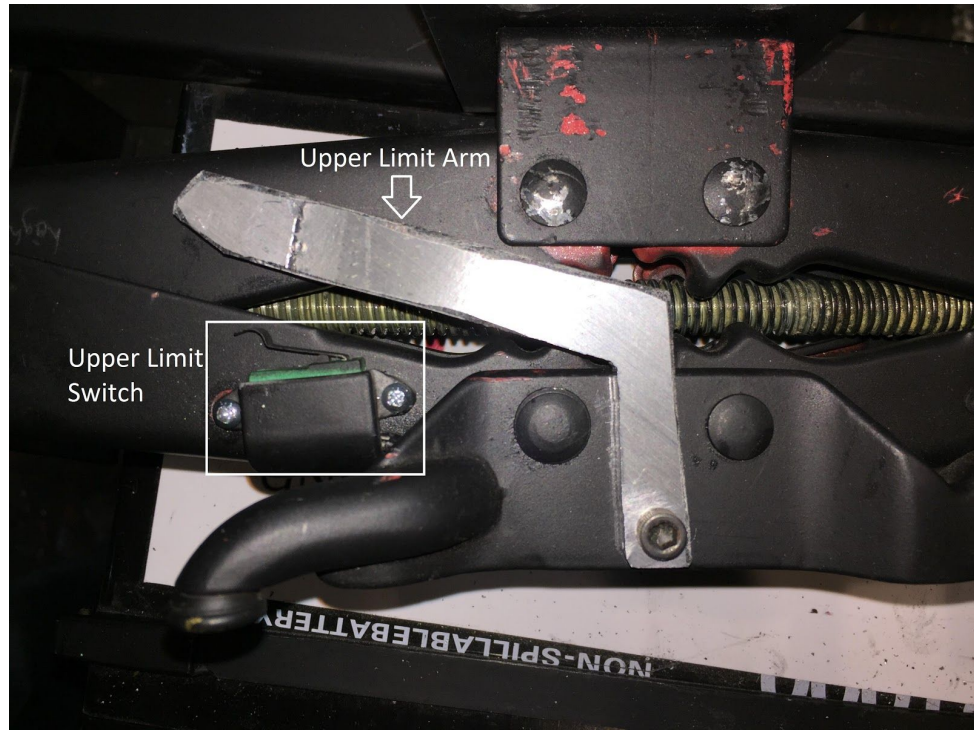


1.2.3. Robot Header Diagram



- Robot Header
- 1 = tx
- 2 = Brake Signal
- 3 = Jack Upper Limit
- 4 = Jack Lower Limit
- 5 = 5V In
- 6 = Ground
- 7 = Reset
- 8 = XBee EStop Signal for Solenoid
- 9 = Arduino EStop Signal for Solenoid
- 10 Placeholder

1.3. Progress Pictures Jack Upper Limit Assembly



2. Controlling the Robot

2.1. Control setup Overview

Computer Side

- Xbox controller takes in commands from the user (motor speed/direction, brake activation, jack direction, sensitivity for thumbsticks, maximum motor speed, night/night view)

Controller Inputs

- Independent driving mode: Left thumbstick = left motor, right thumbstick = right motor
- Collective driving mode: Left thumbstick = throttle, Right thumbstick = steering
- Both triggers = deadman switch (both have to be pressed for computer to send data)
- Both bumpers = EStop (stop communicating with robot)
- DPad: right = Collective; left Independent; up + A button = Jack up; down + A button = Jack down
- Hold X button and left stick = adjust maximum speed for wheel motors
- Hold Y button and left stick = adjust sensitivity of thumbsticks
- B = activate brakes (brakes automatically disengage when user moves thumbsticks)
- Back button = Day/Night mode

Robot Side

- Router receives packets from computer via wifi and sends them to the onboard arduinoEthernet. The arduinoEthernet parses the data (motor directions/speeds, jack direction, brake activation) and directs each to its respective hardware. Motor directions/speeds and jack directions are sent to the Sabertooth 2X60 motor control and Syren10 via serial. The brake signal is sent to the brake relay, where the relay will open or close the connection depending on the value given.

2.2. Description of Code

Computer Side

The computer application is run through Visual Basics with the MainForm.vb file being the main operating file holding all the logic and computation. In order from top to bottom of file:

- Variables declared
- Method loading the visual window form (Form1_Load())
- Method holding most of the computational code (Timer1_Tick())
 - Establishing outdated but not removed from code Estop junction box serial communication
 - Enabling Timer1
 - If statement declaring that all following code can only be accessed if the controller is connected
 - If both bumpers are pressed then Estop value is True (stop sending data)
 - If both riggers are pressed and Estop is true, depending on the selected driving mode, map the values of the two thumbsticks into a byte range (LPos = Map1() RPos = Map1()) and update visual loadbar and text on the window. If not that situation then if the two triggers are not pressed then send a stop values for the motors and update visual window. If neither of those display "disabled" and send stop values to motors and update visual window.
 - If A button is pressed on controller then accept values from pu and down positions of the D-Pad for jack directions and update visual window. If not that situation then display "Lift Disabled," send stop values for jack, and update visual window
 - If right position on D-Pad is pressed then change driving mode to Collective (both motors are giving same value for straight forward and reverse and ratio values for position of right thumbstick), else if left position is pressed on the D-Pada then change driving mode to Independent (each thumbstick position is mapped to each corresponding motor).
 - If BACK button is pressed change display mode between day and night mode

- If X button is being held adjust maximum speed of the wheel motors with the left stick's x-plane's position
- If Y button is being held adjust sensitivity of thumbsticks with left stick's x-plane's position
- If B button is pressed turn brakes on
- Assign positions in a byte array for Right motor, jack up/down, left motor, proximity override value, and brake on/off
- Method to connect computer to wifi router on robot when the "Connect" button is pressed
- Map1() method takes the -1 to 1 value of either thumbstick's Y position and maps it to the 0 - 256 range of a byte. If the Sensitivity has been activated then scale the thumbstick's Y position value to the selected sensitivity. If the MaxSpeed has been set then restrict the speed to that range i.e. if max speed is 50% then the highest value can only be 64 in both directions (forward and backward).
- Map2() takes the left thumbstick's Y position value and the right thumbstick's x position value. If the right thumbstick is at 0 then both left and right motors receive the left thumbstick's value. If the right thumbstick is less than 0 (to the left) then proportionally reduce the value the left motor receives to how far in the scale between -1 and 0 (how far left) the right thumbstick is. If the right thumbstick is greater than 0 (to the right) then proportionally reduce the value the right motor receives to how far in the scale between 0 and 1 (how far right) the right thumbstick is. Map all values to 0 - 256 for the byte array.
- Next four methods are for comport operations that will be removed
- Method that changes window display to night mode (turns everything dark)
- Method that changes window display to day mode (turns everything white)
- Next four methods are for displaying help information when the appropriate window button is clicked
- Method that takes in a value from the user when the max speed window button is clicked
- Shows the arduino code currently in the robot

- Method monitoring if the override has been activated, and if so start the override counter
- Method that takes in the sensitivity scroll bar value and assigns it to the sensitivity value for the thumbsticks
- Method that monitors the sensitivity check box to activate the sensitivity input
- Method that takes in the password from the user to override the proximity sensor

Robot Side

The brains of the robot is the onboard ArduinoEthernet running the ArduinoSceneryBot_v3 file.

- Declaring byte array, IP Address, and port number for communicating to the router
- Declaring a byte array that will hold the incoming byte array from the computer
- Declaring values that monitor if the robot has received data, the last time the robot has received data, and the proximity sensor
- Declaring a UDP variable to receive incoming packet
- Declaring the wheel and jack motor controllers
- Void setup()
 - Connect robot to router and begin serial communication via ethernet
 - Begin serial communication with the motor controllers
 - Declare various values as input or output and initiate brake as high (brakes are disengaged)
- Void loop()
 - Call xferdata() method to perform those operations
 - Check if the user has clicked override
 - Only if the robot has received data in the last 100 ms and the proximity sensor
- Void xferdata()
 - If there is a packet, read it's contents into the joystick_data array
- Void enabled() - opened only if data has been received within the last 100 ms
 - If the brake value = 0, set the brake pin to Low, if the brake value = 1 set the brake pin to High
 - Send the motor controller for the wheels the appropriate data

- If the upper limit switch is pressed on the jack and the jack value is to raise the jack, stop the motor; if the lower limit switch is pressed on the jack and the jack value is to lower the jack, stop the motor; else send the jack in the direction the value dictates
- Update lastState to say the robot was enabled
- Void disabled() - if the robot has not received data within the last 100 ms
 - Update the lastState to say the robot was not enabled
 - Send stop values to all motor controllers

2.3. Example Operation

This section is to provide an example operation that the user will use to help with understanding the flow of data during use. The example we will use is the user turning the robot driving mode to Collective and moving the robot forward. We assume the program is already open and connected to the robot and no issues have arisen either computer or robot side. We will only follow the path the new data takes.

The user will press the right side of the DPad on the XBox controller to switch to Collective driving mode, meaning the left thumbstick is the throttle and the right thumbstick is the steering. The computer program, which is continually repeating through the Timer1_Tick() method, enters the if statement "if currentstate.dpad.right = buttonstate.pressed" and changes the boolean variable Drivingmode "false." When the program loops back again it enters the if statement "ElseIf DrivingMode = False Then" and calls the Map2() method.

Since the max speed and sensitivity have not been selected the first segments of code for these settings don't change any data. In the Map2() method it takes the y position of the left thumbstick and since the left thumbstick is pushed forward and the right thumbstick is not touched (we are moving the robot forward) it enters the the if statement "if right = 0." Since the thumbstick data range is between -1 and 1 it maps the value from this range into the byte range 0 - 256 and is applied to both the right and the left motor variables.

After it leaves this program and reaches the end of the Timer1_tick() main method, the new values for the two wheel motor variables are placed in the byte array and is sent via wifi to the robot's router. On the robot, the router receives this packet and sends it to the ArduinoEthernet via ethernet cable. The arduino reads this packet into an array it can read. The

appropriate value from the array is given to each motor on the motor controller (ST1.motor(1,) and ST1.motor(2,)) and subtracts 127 from this value. This is because the motor controller accepts values in the range -127 - 128, -127 for full reverse and 128 for full forward. This sends the value to the motor controller via packetized serial on the tx serial line out of the arduino. The motor controller receives the data and rotates the motors forward to move the robot forward.

3. Maintenance

3.1. Parts List

A. Arduino Ethernet R

- a. [Arduino Ethernet Information Page](#)
- b. [Arduino Ethernet R3 Schematic](#)

Microcontroller	ATmega328
Operating Voltage	5V
Input Voltage Plug (recommended)	7-12V
Input Voltage Plug (limits)	6-20V
Input Voltage PoE (limits)	36-57V
Digital I/O Pins	14 (of which 4 provide PWM output)
Arduino Pins reserved:	
	10 to 13 used for SPI
	4 used for SD card
	2 W5100 interrupt (when bridged)
Analog Input Pins	6
DC Current per I/O Pin	40 mA
DC Current for 3.3V Pin	50 mA
Flash Memory	32 KB (ATmega328) of which 0.5 KB used by bootloader
SRAM	2 KB (ATmega328)
EEPROM	1 KB (ATmega328)
Clock Speed	16 MHz
W5100 TCP/IP Embedded Ethernet Controller	
Power Over Ethernet ready Magnetic Jack	
Micro SD card, with active voltage translators	
Length	68.6 mm
Width	53.3 mm
Weight	28 g

c.

B. Arduino Uno R3

- a. [Arduino Uno R3 Product Page](#)
- b. [Arduino Uno R3 Information Page](#)
- c. [Arduino Uno R3 Schematic](#)

Microcontroller	ATmega328P
Operating voltage	5v
USB	14
Digital I/O Pins	6
Analog Input Pins	6
Flash Memory	32 KB
SRAM	2 K
EEPROM	1 K
Clock Speed	16 MHz
Length	68.6 mm
Width	53.4 mm
Weight	25 g

- d.
- e. Note: Two identical Arduino Unos installed for XBee communication

C. Sabertooth 2x60 Motor Controller

- a. [Sabertooth 2x60 User's Guide](#)
- b. [Sabertooth 2x60 Product Page](#)

Product Specs

Model: Sabertooth 2X60

Specifications: 60A continuous, 120A peak per channel
6-30V nominal, 33.6V absolute maximum

Synchronous regenerative drive
Ultra-sonic switching frequency
Thermal and overcurrent protection
Lithium protection mode

Input modes: Analog, R/C, simplified serial,
packetized serial

Size: 3" x 3.5" x 1.8"
76 x 89 x 46 mm

c.

D. SainSmart 4-Channel Relay Module

- a. [SainSmart 4-Channel Relay Module Schematic](#)
- b. 5V 4-Channel Relay interface board, and each one needs 15-20mA Driver Current
- c. Equipped Equipped with high-current relay, AC250V 10A ; DC30V 10A
- d. Standard interface that can be controlled directly by microcontroller (Arduino , 8051, AVR, PIC, DSP, ARM, ARM, MSP430, TTL logic active low)
- e. Opto-isolated inputs
- f. Indication LED's for Relay output status

- E. SMAKN 48V DC to 5V DC Step-Down Converter
 - a. [Amazon Product Page](#)
 - b. Input voltage: DC 8-60V
 - c. Output voltage: DC 5V
 - d. Output current: 5A
 - e. Output power: 25W
 - f. Efficiency: up to 94%
 - g. Material Case: Die-Cast Aluminum
 - h. Potting: Epoxy Sealed
 - i. Dimension: (L) 64 *(W) 63*(H) 23 mm
 - j. Weight: Approx. 125g
 - k. Voltage regulation: < 1%
 - l. Load regulation: < 2%
 - m. Ripple: < 20 mv
 - n. Line length: 17 cm
 - o. Working temperature: - 30 °C to + 80 °C

- F. AutomationDirect RHINO PSP12-DC24-2 DC to DC Converter
 - a. [RHINO PSP Series Information Sheet](#)
 - b. [PSP12-DC24-2 AutomationDirect Product Page](#)

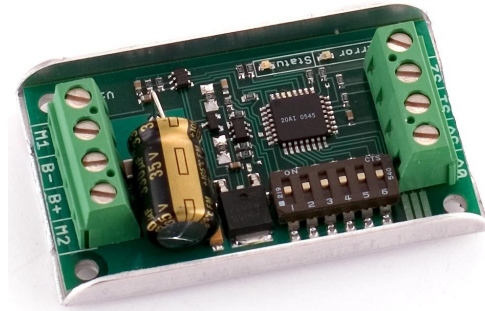
- G. AutomationDirect 782-2C-SKT Relay Mount
 - a. [782-2C-SKT Information Sheet](#)
 - b. [782-2C-SKT AutomationDirect Product Page](#)

- H. AutomationDirect 782-2C-120A Ice Cube Control Relay
 - a. [782-2C-120A Information Sheet](#)
 - b. [782-2C-120A AutomationDirect Product Page](#)

- I. White Rodgers 24V 100A 4 Terminal Solenoid 124-114111
 - a. [24V 100A 4 Terminal Solenoid Information Sheet](#)
 - i. Note: Contains Schematic For SPNO Contact Form
 - b. [24V 100A 4 Terminal Solenoid Spec Sheet](#)

- J. L-C Power Supply Filter 16V 2A
 - a. [Amazon Product Page](#)
 - b. Max. 16V input
 - c. Maximal current: 2A
 - d. Reverse Polarity protection

- K. SyRen 10A Motor Driver



- a.
- b. [SyRen 10A Motor Driver User's Guide](#)
- c. [SyRen 10A Motor Driver Product Page](#)

Model:	SyRen 10
Specifications:	Up to 18V in: 10A continuous, 15A peak 24V in: 8A continuous, 10A continuous with additional heatsinking/airflow, 15A peak
	Synchronous regenerative drive Ultra-sonic switching frequency Thermal and overcurrent protection Lithium protection mode
	Input modes: Analog, R/C, simplified serial, packetized serial
	Size: 1.4" x 2.25" x .55" 35 x 57 x 14 mm

d.

L. NPC B82HT High Torque Right-Side Geared Motor

- a. [NPC B82HT Motor Product Page](#)
- b. 24V-36V
- c. Reversible, Permanent Magnet
- d. 24:1 gear ratio
- e. 20.4 pounds untrimmed
- f. 17 pounds trimmed
- g. 120 rpm
- h. Overall length of motor & gearbox = 14 5/8"
- i. Width of gearbox = 3-3/8"
- j. Diameter of motor = 3-3/4"
- k. Length of output shaft 1-3/4"
- l. Output shaft diameter = 3/4 inch
- m. Keyway = 3/16 inch

M. NPC B81HT High Torque Left-Side Geared Motor

- a. [NPC B81HT Motor Product Page](#)

- i. Note: Same Specifications as Right-Side Geared Motor (above)
- N. Universal Power UB121000 12V 100Ah Sealed Lead Acid Battery
 - a. [UB121000-45978 Amazon Product Page](#)
 - i. Note: Two Identical batteries are installed
- O. DSR Pro Series 24V 5A Battery Charger
 - a. [DSR Pro Series 24V 5A Battery Charger Grainger Product Page](#)
- P. 12V Car Jack - Unknown Model
 - a. [Black Bull 12V Car Jack Amazon Product Page](#)
 - i. Note: Current Jack model unknown, similar jack is listed here
 - ii. Note: Sensor Wire Configuration:
 1. White - Common
 2. Blue - Lower Limit
 3. Red - Upper Limit
- Q. XBee Grove Development Board
 - a. [XBee Grove Development Board Information Page](#)
 - b. [XBee Grove Development Board Schematic](#)
- R. XBee-Pro 900MHz XSC S3B RF Module
 - a. [XBee-Pro 900MHz XSC S3B Product Page](#)
 - b. [XBee-Pro 900MHz XSC S3B Data Sheet](#)
 - i. Note: Two Identical RF modules are installed
- S. ES-P-230001 Emergency Stop Control Station
 - a. [ES-P-230001 Emergency Stop Product Page](#)
 - b. [Emergency Stop Information Sheet](#)
 - c. [Emergency Stop Operating Instructions](#)
- T. 3" Diameter Polyurethane Plate Caster
 - a. [Polyurethane Plate Caster Grainger Product Page](#)

TECHNICAL SPECS

Caster Load Rating Range **Light-Duty** Load Rating **125 lb.** Overall Height **3-3/4"**
Plate Size **2-3/4" x 3-3/4"** Caster Wheel/Tread Material **Polyurethane**
Caster Core Material **Polyolefin** Caster Wheel Shape **Standard** Number of Wheels **1**
Caster Lock Type **No Lock Included** Caster Plate Hole Pattern **D**
Inside Bolt Hole Spacing **1-3/4" x 3"** Outside Bolt Hole Spacing **1-3/4" x 3"**

- b.
- c. Note: Four Identical Casters are installed

U. Robot Marketplace NPC-PH804 Hub

- a. [NPC-PH804 Hub Product Page](#)
- b. 3/4" bore with a 3/16" inch keyway
- c. Weight: .5 lb
- d. Machined out of 1/2" x 3.5" 6061 Aluminum
- e. Bolt Circle: (4) @ 2 3/4" and (4) @ 2 13/16"
- f. Note: Hubs were custom machined to fit tire and rim assembly
- g. Note: Two Identical Hubs are installed

V. Robot Marketplace WHE4N 4" Narrow-Style Rim

- a. [WHE4N 4" Narrow-Style Rim Product Page](#)
- b. Two piece split rim, made from spun aluminum
- c. Fits 4" tires up to 2.25" wide
- d. Weight: 9.9 oz (including mounting bolts)
- e. Hole centers are on 2.5" radius, 6 holes
- f. Weight per side: 3.59oz

W. Carefree 8/3.00-4 Smooth Tread Tires

- a. [Carefree 8/3.00-4 Smooth Tread Tire Product Page](#)
- b. 8.25" X 3.1"
- c. Fits Rim Size: 4" X 2.50"
- d. Weight: 2 lbs
- e. Dimensions: 8 x 3 x 8 in.
- f. Static Load Capacity: 295 lb
- g. Max Load Capacity of 195 lbs @ 3MPH
- h. Max Load Capacity of 125 lbs @ 6MPH
- i. Max Load Capacity of 95 lbs @ 8MPH

X. IEC Male panel mount connector

- a. [IEC Panel mount connector Amazon Product Page](#)
 - i. Note: Exact model unknown, similar product is linked here

- Y. Momentary SPST switch
 - a. [Momentary SPST Switch Amazon Product Page](#)
 - i. Note: Exact model unknown, similar product is linked here
- Z. 75A Pollak Push/Pull switch
 - a. [75A Pollak Push/Pull Amazon Product Page](#)
 - i. Note: Exact model unknown, similar product is linked here
- AA. Fasteners
 - a. Motor
 - i. [6 x M6-1.0x40 Bolt](#)
 - ii. [6 x M6-1.0x20 Bolt](#)
 - b. Casters
 - i. 16 x 5/16" Diameter 1" Length Hex Bolt
 - ii. 16 x 5/16" NyLock Hex Nut
 - c. Motor Arm
 - i. 2 x 1/2" Diameter 2½" Length Hex Bolt
 - ii. 2 x 1/2" NyLock Hex Nut
 - iii. 2 x 1/2" Washer
 - d. Wheel Hub
 - i. 12 x 1/4" Diameter 1½" Length Hex Bolt
 - ii. [12 x 1/4" x 20 NyLock Hex Nut](#)
 - e. Wheel Axle
 - i. 2 x 1/2" NyLock Hex Nut
 - f. Jack Mount
 - i. 4 x 3/8" Diameter 3½" Length Hex Bolt
 - ii. [4 x 0.6" Diameter 2" Length Spring](#)
- BB. AutomationDirect T1-1015W 1" wide x 1½" height Wire Raceway
 - a. [AutomationDirect Wire Raceway Product Page](#)
 - b. [Wire Raceway Spec Sheet](#)
- CC. AutomationDirect Aluminum DIN Rail 35mm Width x 10mm Height
 - a. [DN-R35SAL1 Aluminum DIN Rail Product Page](#)
 - b. [DIN Rail Spec Sheet](#)

4. Known Problems

- Jack Upper Limit Switch Malfunctioning
- Brake Relay Overloads Arduino
- Power to jack motor controller overloads when solenoid turns on during initial startup - error light
- Xbee E-Stop not yet implemented