

Winter 2012

ECE 579

Intelligent Robotics I

Omar Mohsin

Ali Hassan E. Alnasser

PSU Guide Robot

Base Team

Table of Contents

I.	Introduction.....	5
II.	Robot Design	5
III.	Construction Parts.....	6
	A. Aluminum Bars	6
	B. Bolts.....	7
	C. U bolt	8
	D. Nuts	9
	E. Brackets	9
	F. Washers.....	12
	G. Spacer	13
	1. Bronze self-lubricating Spacer.....	13
	2. Aluminum Spacer	13
IV.	Base Parts	14
	A. Wheels.....	14
	B. DC Motors.....	18
	C. Gearbox	19
	D. 500 Hub Key.....	20
	E. DC Motor Controller	21
	F. Arduino	22
	G. Battery	23
V.	Actual work.....	24
VI.	Prices	27
	A. Wheels.....	27
	B. Aluminum bars	28
	C. Gearbox	28
	D. Dc motor (RS555 Brushed Motor).....	28
	E. Motor driver	29
	F. Arduino	29
	G. Hardware Stuff.....	29
VII.	Calculations	30

A.	Torque calculation	30
B.	Current calculation	31
C.	Battery Capacity Calculation	32
VIII.	Appendix A: Mecanum wheel [3]	33
IX.	Appendix B: Mecanum Wheel Assembly [4]	35
X.	Appendix C: DC Motor Specifications [5]	37
XI.	Appendix D: P60 Gearbox Preliminary [6]	38
XII.	Appendix E: Hub key dimension [7]	39
XIII.	Appendix F: DC Motor Controller [8]	40
XVI.	Appendix G: Report Video Links	78
XVII.	Reference	80

Table of Figures

Figure 1:	Robot schematic	5
Figure 2:	Square shape aluminum bar.	6
Figure 3:	L shape aluminum bar.	7
Figure 4:	Socket screw bolt.	7
Figure 5:	Machine screw bolt.	8
Figure 6:	U shape bolt.	8
Figure 7:	Nylon nut.	9
Figure 8:	1"X1/2" with 2-holes L bracket.....	9
Figure 9:	1-1/2"X5/8" with 4-holes L bracket.....	10
Figure 10:	2"X5/8" with 4-holes L bracket.	10
Figure 11:	1-1/2"X3/4" with 4-holes L bracket (brace).	10
Figure 12:	1-1/2"X1-1/2" with 6-holes L bracket (brace).....	11
Figure 13:	2-1/2"X1-1/2" with 6-holes L bracket (brace).....	11
Figure 14:	4"X4" T-shaped bracket with 5-holes.....	12
Figure 15:	Flat washer.	12
Figure 16:	1/2"X3/4"X1/2" bronze spacer.....	13
Figure 17:	Aluminum spacer.....	13
Figure 18:	Omni Wheels alignment.	15
Figure 19:	Mecanum wheels alignment.	16
Figure 20:	Mecanum wheel movement.	17
Figure 21:	DC motor.	18
Figure 22:	Motor gearbox.....	19
Figure 23:	Pinion.....	19

Figure 24: Gearbox 500 hub key.....	20
Figure 25: Gearbox key.....	20
Figure 26: Saber tooth DC motor controller.....	21
Figure 27: Arduino Mega.....	22
Figure 28: Base dimension.	24
Figure 29: Brackets holding two bars.	24
Figure 30: Mecanum wheel is connected to the gearbox and DC motor and mounted to the aluminum bar.	25
Figure 31: U shape nuts holding the gearbox to the bars.	26
Table 1: Conventional vs. Omni wheels.	14
Table 2: Omni vs. Mecanum wheels.....	14
Table 3: Mecanum wheel advantages and disadvantages.....	17
Table 4: DC motor specifications.....	18
Table 5: DC motor controller specifications.....	21
Table 6: Arduino specifications.	22
Table 7: Different Wheels characteristics with their prices.	27
Table 8: Different gearbox with their prices.	28
Table 9: Different motor drivers' prices with specifications.	29
Table 10: Hardware prices.....	29

PSU Guide Robot: Mecanum Platform

I. Introduction

The guide robot is a robot which will guide visitors to PSU Engineering building through every department offices and varies labs around the building. This report is about the base construction and functionality. Also, it shows how the base is connected to the whole body.

II. Robot Design

The expected robot look is shown in figure 1. The robot base is consists of four Omni-wheels which is connected to four DC motors via gear boxes. The robot is constructed from aluminum Bars assembled together using some brackets, bolts, nuts and washers. The maximum expected weight of the robot is about 143lb (65kg).

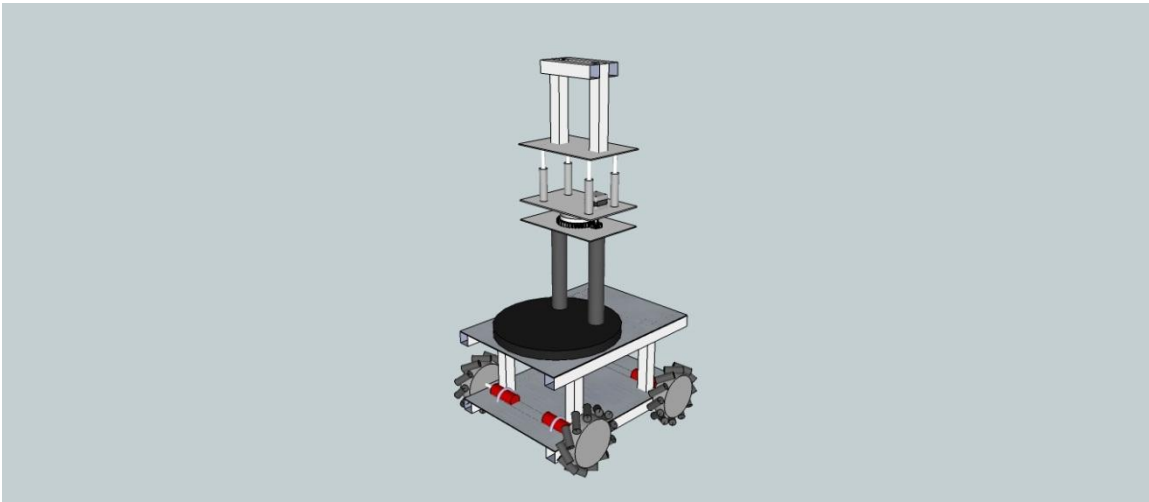


Figure 1: Robot schematic.

After seeing a lot of different designs in the internet from different universities or companies, our design was influenced by the Segway Mecanum wheel base. Some details on how the robot is built are on the actual work section. To see some video on how we constructed the base see appendix G. The link to the 3d model of the robot is found in the google 3d warehouse:

<http://sketchup.google.com/3dwarehouse/details?mid=2d5e490d368bede3ed1231cfd3b598dd&prevstart=0>.

III. Construction Parts

A. Aluminum Bars

There are several shapes of aluminum bars that have been used in this project. There is the square shape (figure 2) with .5"X.5" and thickness of 0.065" which weighs 0.13lb per foot , 1.5"X1.5" and a thickness of 0.065" which weighs .45lb per foot and 1.5"X1.5" and a thickness of 0.125" which weighs 0.732lb per foot and.



Figure 2: Square shape aluminum bar.

There is the L shape (figure 3) with 0.5"X0.5" and a thickness of 0.065" which weighs 0.068lb per foot, .5"X.5" and a thickness of 0.125" which weighs 0.131lb per foot, 0.75"X0.75" and a thickness of 0.125" which weighs 0.205lb per foot and 1.25"X1.25" and a thickness of 0.125" which weighs 0.343lb per foot.

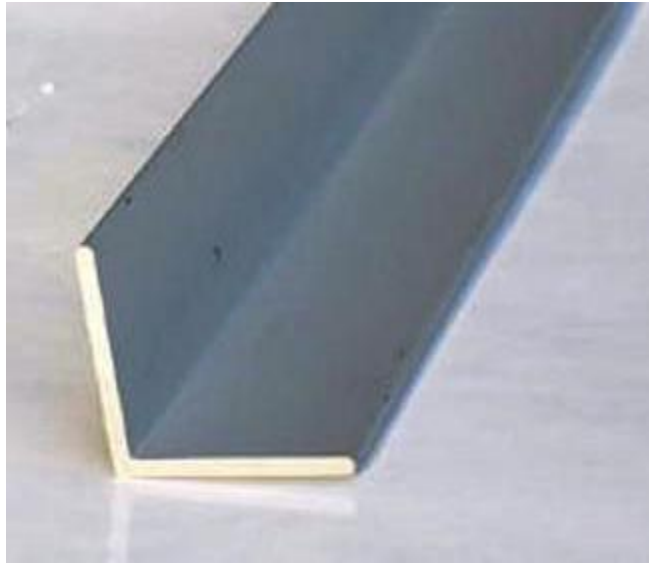


Figure 3: L shape aluminum bar.

B. Bolts

There are two types of bolts. One of them is stainless steel socket screws of quantity of 50 and 2" in length and 3/16" in width. It is shown in figure 4.



Figure 4: Socket screw bolt.

The other one is stainless steel machine screws of quantity of 50 and 2" in length and 3/16" in width. It is shown in figure 5.



Figure 5: Machine screw bolt.

C. U bolt

This U bolt (figure 6) is 4-1/2" in length and 2-1/4" in width. This U shape bolt is used to hold the gearbox to the aluminum bar, so the gearbox and the DC motor will not detached from the base during the robot movement.



Figure 6: U shape bolt.

D. Nuts

The lock nuts (figure 7) are nylon which will tighten the bolts strongly. It is $\frac{3}{16}$ " in width to support the available bolts and there are 100 nuts.



Figure 7: Nylon nut.

E. Brackets

There are several types of stainless steel brackets with L shape. There is $1 \times \frac{1}{2}$ " with 2-holes (figure 8), $1\text{-}\frac{1}{2} \times \frac{5}{8}$ " with 4-holes (figure 9), $2 \times \frac{5}{8}$ " with 4-holes (figure 10), $1\text{-}\frac{1}{2} \times \frac{3}{4}$ " with 4-holes (figure 11), $1\text{-}\frac{1}{2} \times 1\text{-}\frac{1}{2}$ " with 6-holes (figure 12) and $2\text{-}\frac{1}{2} \times 1\text{-}\frac{1}{2}$ " with 6-holes (figure 13). There is 4×4 " T shape with 5-holes (figure 14).



Figure 8: $1 \times \frac{1}{2}$ " with 2-holes L bracket.



Figure 9: 1-1/2"X5/8" with 4-holes L bracket.



Figure 10: 2"X5/8" with 4-holes L bracket.



Figure 11: 1-1/2"X3/4" with 4-holes L bracket (brace).



Figure 12: 1-1/2"X1-1/2" with 6-holes L bracket (brace).



Figure 13: 2-1/2"X1-1/2" with 6-holes L bracket (brace).



Figure 14: 4"X4" T-shaped bracket with 5-holes.

F. Washers

There are 8 flat washers each 2 of them are connected to one gearbox to prevent the hub key and the spacer from foaling off the shaft. Washers are shown in figure 15.



Figure 15: Flat washer.

G. Spacer

1. Bronze self-lubricating Spacer

The outer diameter of this bronze spacer (figure 16) is $\frac{3}{4}$ " and the inner diameter $\frac{1}{2}$ ". It is used to space the key hub in the gearbox since the gearbox shaft length is $1\frac{1}{2}$ " and the key hub is 1". The available one in the market is 1" in length, so it has to be cut in order to fit in the gearbox.



Figure 16: $\frac{1}{2}$ "X $\frac{3}{4}$ "X $\frac{1}{2}$ " bronze spacer.

2. Aluminum Spacer

These spacers (figure 17) are used to fill in the extra length in bolt that is used to attach the gearbox to the Aluminum bars underneath it. Because the length of the available bolts is 2" and the height of the aluminum bar is 1.5", we need to put this spacer.



Figure 17 Aluminum spacer

IV. Base Parts

There are several parts in the base. Each one is connected to each other.

A. Wheels

There was a discussion among all the project teams about the type of wheels that are going to be used in the guide robot. The debate was about choosing either Omni or traditional wheels. After discussing the advantages and disadvantages, the Omni wheels were chosen due to its ability to go any direction. Table 1 shows the comparison between the normal and Omni wheels.

Conventional wheels	Omni wheels
Wheels in aligned the body	Wheels are 45° with the body
2 degrees of freedom	3 degrees of freedom
High load capacity	Good for small and light robots
Simple design	Complex design

Table 1: Conventional vs. Omni wheels.

Afterward, Mecanum wheels came in the picture and the debate started again. The Omni wheels (figure 18) are connected diagonally by 45 to the base and can provide movement to any direction. On the other hand, the Mecanum wheels (figure 19) are connected in alignment to the base and can provide the same movement. Table 2 shows the comparison between Omni and Mecanum wheels.

Omni-Wheels	Mecanum Wheels
mounted diagonally (45o)	mounted parallel to the body
move in any direction (3 DOF)	move in any direction (3 DOF)
rollers are in the side of the of the wheel	rollers are constructed diagonally in the wheel
useful for small and light robot	can handle heavy objects

Table 2: Omni vs. Mecanum wheels.

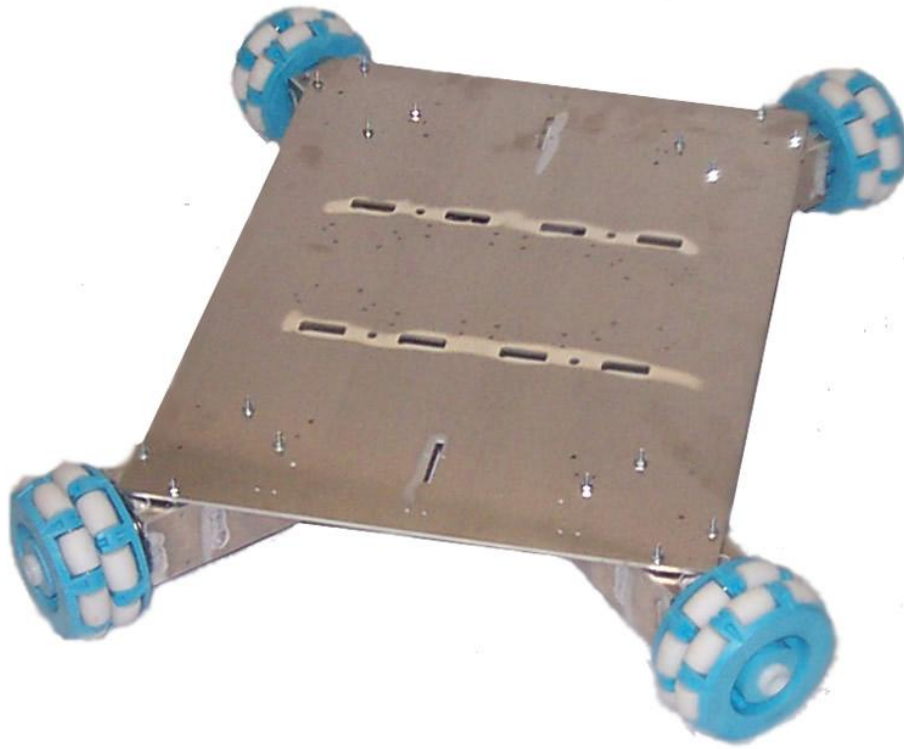


Figure 18: Omni Wheels alignment.



Figure 19: Mecanum wheels alignment.

The settlement was on the Mecanum wheels because the good looking and endurance hence they can endure more weight than Omni wheels. Also, unlike the Omni wheels, the Mecanum wheels have the rollers aliening diagonally so it will provide the movement that the Omni wheels provide to the robot. The Mecanum wheels are 8" in diameter and 2" in width and each weighs about 2.5lb. See appendix A to know more about these wheels and appendix B to see more on how to construct these wheels. Refer to appendix G to see some video on how to assemble the wheels.

The degree of freedom between Omni and Mecanum wheels does not matter, since they both have three degrees of freedom. Table 3 [1] shows the advantages and disadvantages of the Mecanum wheels.

Advantages	Disadvantages
Compact design	Very complex conceptually
High load capacity	Discontinuous wheel contact
Simple to control	High sensitivity to floor irregularities
Less speed and pushing force when moving diagonally	Complex wheel design

Table 3: Mecanum wheel advantages and disadvantages.

The Mecanum wheel movement is shown in figure 20.

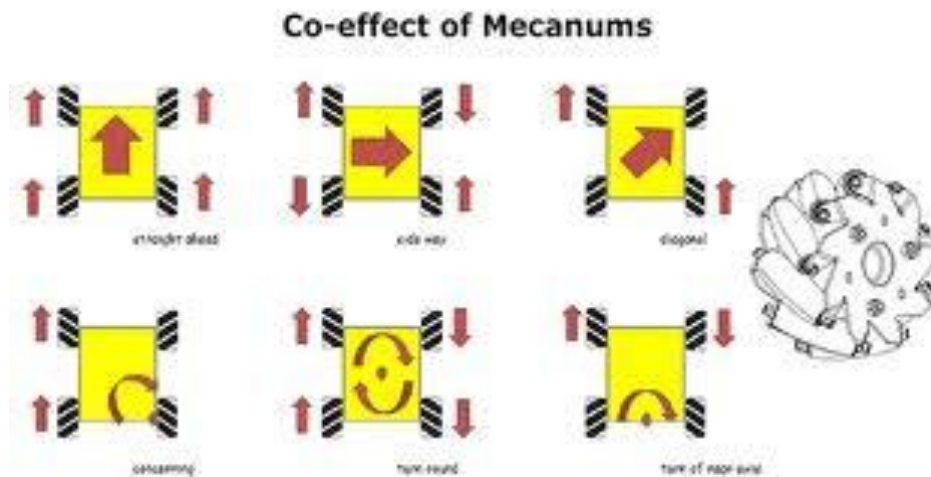


Figure 20: Mecanum wheel movement.

B. DC Motors

There are 4 of RS555 brushed DC motors (figure 21) that will run the wheels. They are connected to the Mecanum wheels via gearboxes. Table 4 shows its specifications. To see more on DC motors, refer to appendix C.

Specification	Values
Voltage (V)	5-15
Nominal Voltage (V)	12
No load speed (rpm)	7750
No load current (A)	0.4
Stall current (A)	15
Stall torque (mN-m)	205.9
Efficiency (%)	68.5

Table 4: DC motor specifications.



Figure 21: DC motor.

C. Gearbox

The P60 RS-540/550 gearbox (figure 22) has a ratio of 64:1. It is the medium between the DC motor and the Mecanum wheels and will provide the speed and torque required to move the robot. It comes with a pinion that should mount to the Dc-motor shaft (figure 23). Refer to appendix D for more information.



Figure 22: Motor gearbox.



Figure 23 Pinion

D. 500 Hub Key

This hub key (figure 24) is used to attach the wheel to the gearbox using small machine key (figure 25). Moreover, $\frac{1}{2}$ " spacer is used to fill in the extra space in the shaft. Refer to appendix E to see the hub key schematics.



Figure 24: Gearbox 500 hub key.



Figure 25: Gearbox key.

E. DC Motor Controller

The saber tooth DC motor controller (figure 26) has the ability to control two DC motors at a time. That makes it two DC motor controllers to control the 4 DC motor that we have. This controller will provide the protection to the Arduino board as well as voltage and current to the DC motors. See appendix F to see more about the controller. Table 5 shows specification about the controller.

Specifications	Voltage (V)	Continuous current (A)	Max. current (A)
Value	6-24	10	15

Table 5: DC motor controller specifications.



Figure 26: Saber tooth DC motor controller.

F. Arduino

The Arduino mega (figure 27) is the brain of the robot base and the body which will send the commands to the DC motors via DC motor controller. The DC motor controllers are connected to the Arduino via the DC motor controllers which by themselves will transfer the commands to the DC motors. Table 6 shows some of the Arduino specifications. To know more about Arduino refer to www.arduino.cc.

Specifications	Values
Operating Voltage (V)	5
Input Voltage limits (V)	6-20
Digital I/O pins	54 (of which 14 provide PWM output)
Analog I/O pins	16
DC current per I/O pin (mA)	40

Table 6: Arduino specifications.



Figure 27: Arduino Mega.

G. Battery

The battery needed to provide the necessary voltage and current is 12V. The battery capacity (Ah) is chosen based on the mass of the robot, the velocity of the robot, the maximum incline, the supplied voltage, the desired acceleration, the desired operating time and the total efficiency.

The velocity, the supplied voltage, the desired acceleration, the desired operation time and the total efficiency will not change. The remaining properties are the total mass and the maximum incline. If we take the extreme situation as 65kg and 28° , the battery capacity pack will be around 50Ah. The battery in this case will weigh 15kg which may add some weight to the robot and its dimension is 18.34X16.51X17.5cm which is quite big.

However, the robot may not reach this situation since the robot is made of aluminum which is light and the incline in the engineering building is not that steep. In that case if we take the mass as 40kg and the incline as 10° , the battery capacity pack will be around 12Ah. This battery is about 4kg and the dimension is 15X9.8X9.5cm. This is for the base dc-motors consumption. For the battery capacity of the whole body, refer to capacity calculation section of this report.

There is another crucial specification of the battery which is the battery cycle. It means the number of charge\discharge cycles of the battery can experience before it fails to meet specific criteria [2]. In our situation, the desired battery cycle life is 500.

V. Actual work

After having all parts available to build the base taking into account that it took more than 2 days to find the right materials, we started by calculating the aluminum bars dimensions. The base dimension should be 17"X 24" in addition to the width of each Mecanum wheel which is 2" as shown in figure 28.

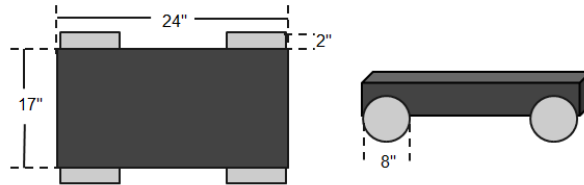


Figure 28: Base dimension.

Then we started to cut the bars using the right blade that we bought for the lab miter saw. After that, we punched the bars to make holes by using punching machine that is available in the lab and we spent more than 2 days. By using the brackets and bolts, we assembled them together (figure 29).

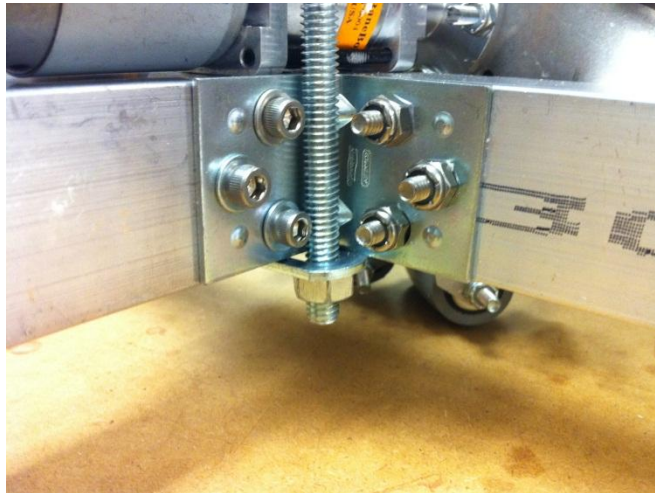


Figure 29: Brackets holding two bars.

Before assembling the wheels together with the DC motor and gearbox, the gearbox is needed to be greased, and the wheel is needed to be assembled (took around one day). To know how to do the gearbox greasing, and the wheel assembling, refer to appendix G. After doing that, we connected the gearbox with the DC motor, then to the wheel using the 500 key hub and the two flat washers. Refer to appendix G to see the video. The same procedure was done for the rest of the wheels, gearboxes and DC motors (took one day).

The next step is to put together the combination of DC motor, gearbox and wheel to the assembled aluminum bars (figure 30) and use aluminum spacer to compensate for the extra length to the bolt that connects them together. This was done by making another two holes in the aluminum bars for the gearbox, and using U shape bolts to hold the combination to the bars (figure 31). To make the U shape bolt hold the gearbox from the top, we punched the spacer that comes with the bolt, and that will make two holes that can be screwed to the gearbox.

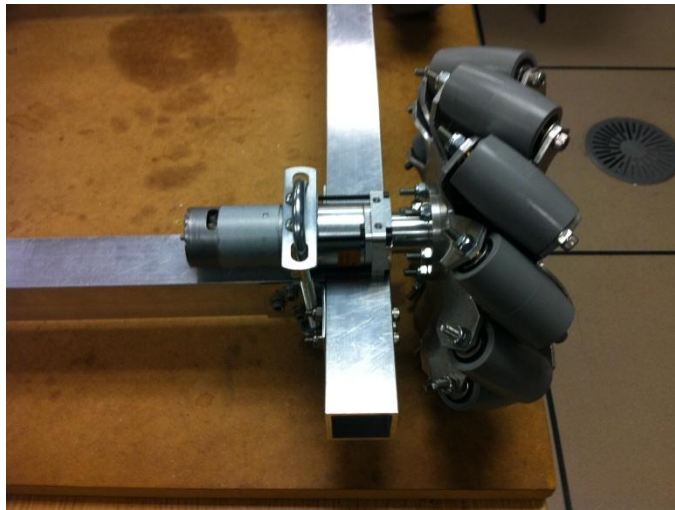


Figure 30: Mecanum wheel is connected to the gearbox and DC motor and mounted to the aluminum bar.



Figure 31: U shape nuts holding the gearbox to the bars.

The next step is to make sure that the robot can move and the whole connection is correct before engaging it to the whole body. The test is to connect the four DC motors to the NXT using two HiTechnic. The Arduino, suber tooth and the battery part will be part of our work in the next term.

VI. Prices

We looked for the parts that we needed in different places and websites. The prices are varied depending on the quality and the item characteristics. For example; the dc-motor drivers prices will become 50\$ more if we choose the 26A current instead of 12 A maximum current supplied by it.

A. Wheels

There are different wheels varied in the prices depending on the weight that they support or the quality as well. The table below lists different wheels with their prices.

Wheel	Diameter	Supported weight	Price
Plastic Omni-wheel (am-0013)	8"	50 lbs	34
Aluminum Omni Wheel (am-0098)	8"	50 lbs	54
Mecanum Wheel (am-0081)	8"	180 lbs	78
Mecanum Wheel (am-0201)	10"	440 lbs	180

Table 7: Different Wheels characteristics with their prices.

The above table is to show the comparison between the Omni wheels and the Mecanum wheels in prices and characteristics. It is easy to see the marginal difference in prices on one hand and the supported weights on the other hand.

B. Aluminum bars

We bought the Aluminum bars from metal market which is near the airport. It is a store that works only on metal stuffs. It also provides them with different characteristics like the thickness, dimension and shape like square, L shape or cylinder shape. The total cost so far for the aluminum bars is 160\$.

C. Gearbox

After looking in different websites, we found that there are a variety of gearboxes. They are shown in Table 8.

Gear	Torque	RPM	Price
12V, 58RPM 60:1 Gear Motor	254.8mN.m	58	22.58
12V, 17RPM 200:1 Gear Motor	784mN.m	17	22.58
12V, 185RPM 20:1 Gear Motor	78mNm	185	22.58
Banebots 64:1 P60 Gearbox and RS555 Brushed Motor	13 Nm (Combined with the motor)	121	70

Table 8: Different gearbox with their prices.

The Banebots gearboxes prices start at 50\$ for the 4:1 ratio, and end with 80\$ for the 256:1 ratio.

D. Dc motor (RS555 Brushed Motor)

The prices of the dc motor are included with the gearboxes prices. It is about 6\$ each.

E. Motor driver

Here are some motor drivers' prices and specifications.

Motor driver	Current	No. of motor	Price
Sabertooth driver	10	2	80
Pololu TRex Dual Motor Controller	13	2	100
Bi-directional DC Motor Speed Controller	5-48Amp (modification are needed)	1	25

Table 9: Different motor drivers' prices with specifications.

F. Arduino

Arduino Mega price is 59\$.

G. Hardware Stuffs

Some of the parts that are bought are listed in table 10.

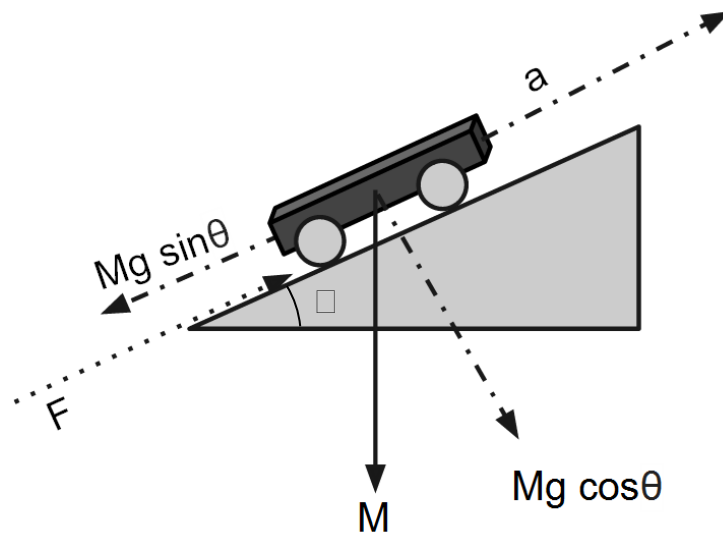
Item	Specification	Price
Bolts	Socket and Machine screws	183 for all the hardware parts
Nylon nuts		
Brackets		
Grease		10.08
Washers		1.74
Spacers	Pushing for the gears shaft	6.4 for all 4
Saw blade	For the miter saw	48

Table 10: Hardware prices.

The total cost for this base is (there is half of the Aluminum bars have not been used yet) 1151.52\$. Also we need to buy the saber tooth motor driver and the battery. That means another 300\$.

VII. Calculations

A. Torque calculation



r = Wheel radius.

M = Platform mass.

θ = maximum incline angle.

F = Fraction force That produces the torque.

a = acceleration.

N = number of the wheels.

The force result on the base axis:

$$Mg \cdot \sin \theta + Ma = F_{friction}$$

$$Mg \cdot \sin \theta + Ma = T/r$$

$$T = \frac{(a + g \sin \theta) \cdot M \cdot r}{N}$$

The result should be multiplied by 100 / the motor efficiency.

The base characteristic are:

$r = 4''$, $M = 64\text{Kg}$, $a = 0.3 \text{ m/s}^2$, $\theta = 32$, $N = 4$ and the motor efficiency is 68.5.

From the above equation we find the required torque is:

$$T = 13 \text{ Nm}$$

the RS555 Motor - 12V, torque equals 0.2059 Nm and the angular velocity equals 7750 rpm.

By using P60 Gearbox with 64:1 gear ratio, we found:

$$T = 0.2059 * 64 = 13.17 \text{ Nm.}$$

$$w_2 = 7750 / 64 = 121 \text{ rpm}$$

$$121 * 2 * \text{Pi} / 60 * 4 * 2.54 / 100 = 1.28 \text{ m/s.}$$

B. Current calculation

$$P = T \cdot w_2$$

The angular velocity in rad/sec

$$P = V \cdot I \quad \text{then, } I = \frac{T \cdot w_2}{V}$$

$$I = 13.7 \text{ A}$$

We can conclude from the previous calculation that the gear will handle the required torque. And that torque which is 13Nm is for the ultimate situation. For instance; if we change the weight to 40kg and the incline to 10 degree the resultant torque will drop to 2.96 Nm, And the current will drop to 3.12 A. There is a simple tool in the robot shop website to do all this calculations (<http://www.robotshop.com/dc-motor-selection.html>).

C. Battery Capacity Calculation

The current drawn by each part of the robot:

Platform motors = 12 A.

Waist Actuator = 10 A.

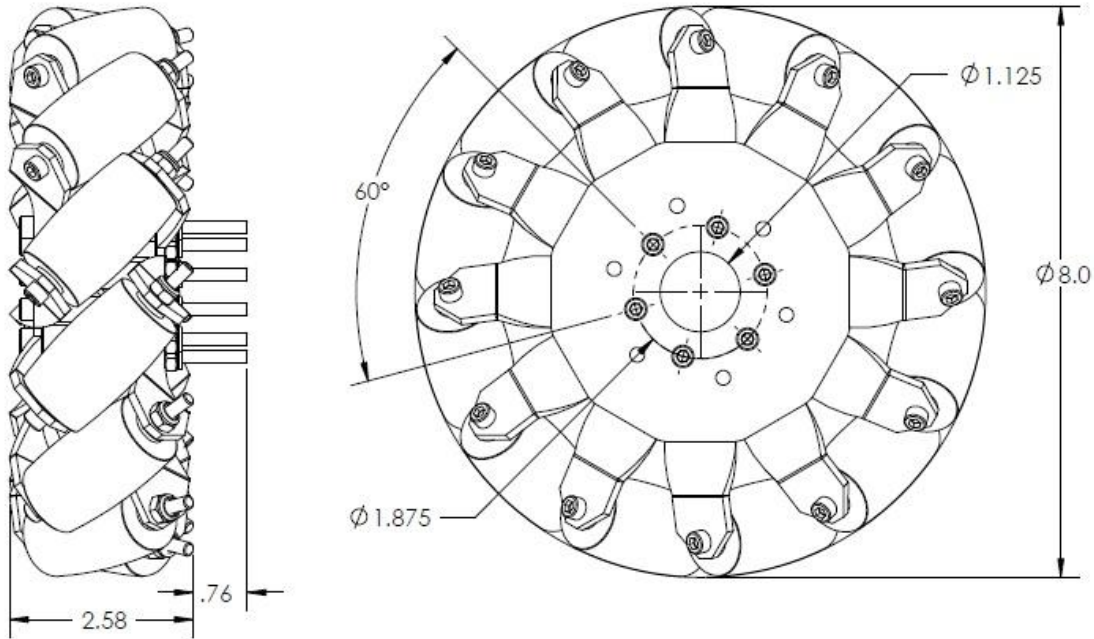
Two Kinect = 2 A.

One Arm = 3 A.

One Arduino board = 1 A.

The total A = 28 A.

VIII. Appendix A: Mecanum wheel [3]



PROPRIETARY AND CONFIDENTIAL
 THE INFORMATION CONTAINED IN THIS DRAWING IS THE SOLE PROPERTY OF <INSERT COMPANY NAME HERE>. ANY REPRODUCTION IN PART OR AS A WHOLE WITHOUT THE WRITTEN PERMISSION OF <INSERT COMPANY NAME HERE> IS PROHIBITED.

		UNLESS OTHERWISE SPECIFIED:		NAME	DATE	AndyMark Inc.	
		DIMENSIONS ARE IN INCHES		DRAWN	MK		
		TOLERANCES:		CHECKED		TITLE:	
		FRACTIONAL ±		ENG APPR.		8 inch Mecanum Wheel	
		ANGULAR: MACH ±		MFG APPR.		SIZE	
		TWO PLACE DECIMAL ±		Q.A.		DWG. NO.	
		THREE PLACE DECIMAL ±		COMMENTS:		REV	
		INTERPRET GEOMETRIC TOLERANCING PER:				8 mecanum wheel with roller III	
		MATERIAL				SCALE: 1:4	
		FINISH				WEIGHT:	
NEXT ASSY	USED ON	DO NOT SCALE DRAWING				SHEET 1 OF 1	
APPLICATION							

5

4

4

1

3

1

2

1

1



Bengt Ilon invented this type of wheel while working for the Swedish company Mecanum AB. The Mecanum-style drive base uses 4 wheels, including 2 right wheels and 2 left wheels. One right and left is on each side of the robot. Each wheel is driven independently. AndyMark Mecanum Wheels use bent sheet metal for the plates that retain the rollers.

See www.andymark.com for more details.

Mecanum Wheel Specs			
Wheel Size - (inches)	6	8	10
Wheel Weight - (pounds)	1.3	2.5	10
Load Rating- (pounds per wheel)	80	80	440



AndyMark has all the necessary hardware to mount these wheels-

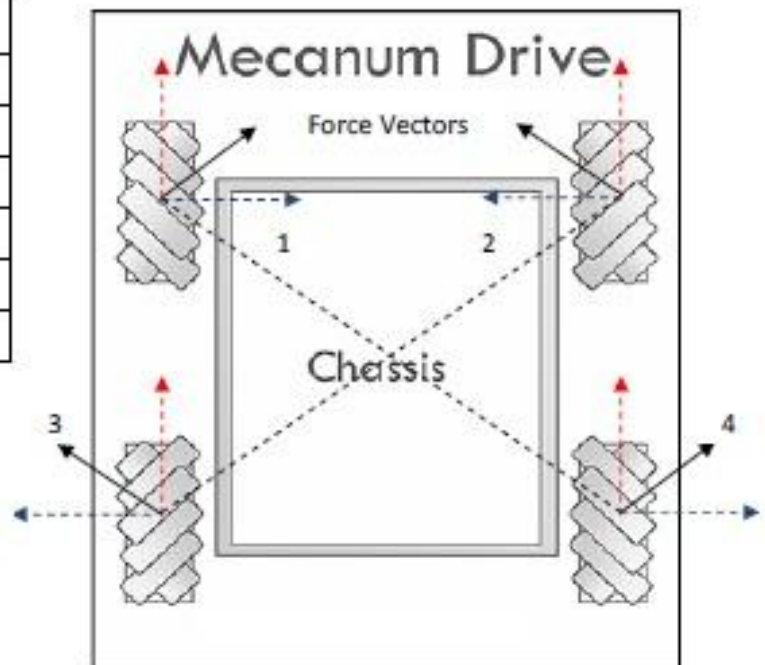
Hubs- 0.375, 0.500", 0.625" bore →

Bolt Kits- available for both to mount hubs



<u>Direction of Movement</u>	<u>Wheel Actuation</u>
Forward	All wheels forward same speed
Reverse	All wheels backward same speed
Right Shift	Wheels 1, 4 forward; 2, 3 backward
Left Shift	Wheels 2, 3 forward; 1, 4 backward
CW Turn	Wheels 1, 3 forward; 2, 4 backward
CCW Turn	Wheels 2, 4 forward; 1, 3 backward

To the right: This is a top view looking down on the drive platform. Wheels in Positions 1, 4 should make X-pattern with Wheels 2, 3. If not set up like shown, wheels will not operate correctly.



IX. Appendix B: Mecanum Wheel Assembly [4]

AndyMark 8" Mecanum Wheel Assembly Instructions

July 2011



Parts need to make one (1) 8" Mecanum Wheel

Tools needed:

Item	Component	Qty.	Part #
A	Side plate (right or left)	2	am-0630 (R) am-0631 (L)
B	Molded roller w/ black core and 2 bushings	12	am-0608
C	Brass tube	12	am-0602
D	1600 Spacer	1	am-0652
E	#10-32 x 3.0 Inch-long screw	18	am-1201 (25)
F	#10-32 Nylock nut	18	am-1212 (50)
G	$\frac{1}{4}$ " Stainless steel washer, 0.04" thick	24	am-1149 (100)

$\frac{5}{32}$ " Hex head driver
$\frac{3}{8}$ " Wrench (or socket)
Cordless drill, set on low torque setting



Set of 4 am-0083

Wheel Assembly Instructions:

Step 1: Lay one side plate (A) down on the table with inside face (inside sticker is on one side of plate). Feed screws (E) from bottom up through the 6 interior holes closest to the center bore.



Step 1



Step 2

Step 2: Place the 1600 Spacer (D) on the plate by feeding the screws (E) through the 6 holes in the spacer.

Step 3: Place second side plate (A) inside down onto wheel assembly. The "inside" of the plates should face each other. Then install nuts (F) on each of the 6 interior screws, but leave them only finger-tight.



Step 3



Step 4

Step 4: Insert brass tube (C) into roller (B), repeat for all rollers.

For more information on our products, please visit us on our website, www.andymark.com.



Step 5



Step 6

Step 5: Feed screw (E) from bottom up through perimeter hole, making sure that the screw head is on the same side as the screw heads of the interior screws. Put washer (G) onto screw (E), over plate.

Step 6: Put roller (B) onto screw (E). Put washer (G) onto screw (E), on top of roller (B). Place the washer so that the brass tube is inside the inner hole of the washer.

Step 7: Install nut (F) on the axle screw, and tighten just enough to fully engage the nylock nut.



Step 7



Step 8

Step 8: Repeat steps 5 through 7 for the remaining 11 rollers (B).

Step 9: Use the 3/8 wrench and a 5/32 hex driver (or allen wrench), and tighten the 12 perimeter screws first, keeping the washers fitted over the brass tubes and not sandwiched between the tube and the side plate. Tighten the 6 interior screws in a balanced way.



Step 9



Step 9

Images of final wheel assembly



For more information on our products, please visit us on our website, www.andymark.com.

X. Appendix C: DC Motor Specifications [5]

Rs555 DC-motor

Performance

Model	M5-RS555-12
Operating v	: 5v - 15v
Nominal v	: 12v
No Load RPM	: 7750
No Load A	: 0.4A
Stall Torque	: 29.16 oz-in 205.9 mN-m
Stall Current	: 15A
Kt	: 1.94 oz-in/A 13.7 mN-m/A
Kv	: 646 rpm/V
Efficiency	: 68.5%
RPM - Peak Eff	: 6660
Torque - Peak Eff	: 4.76 oz-in 33.6 mN-m
Current - Peak Eff	: 2.5A

Physical

Weight	: 7.5 oz (213g)
Length - for motor	: 2.24 in (57mm)
Diameter (with flux ring)	: 1.5 in (38mm)
Diameter (no flux ring)	: 1.41 in (35.8mm)
ShaftDiameter	: 0.13 in (3.2mm)
Shaft Length	: 0.3 in (7.6mm)
Mounting Screws (2)	M3

XI. Appendix D: P60 Gearbox Preliminary [6]

**BANEBOTS P60 GEARBOX
PRELIMINARY**

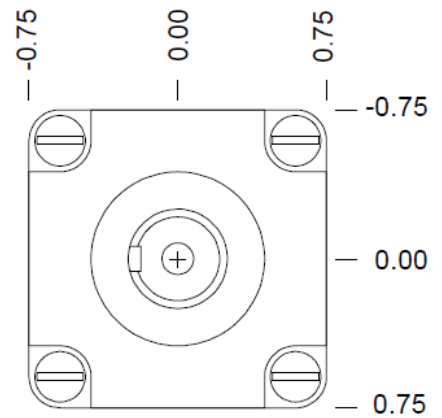
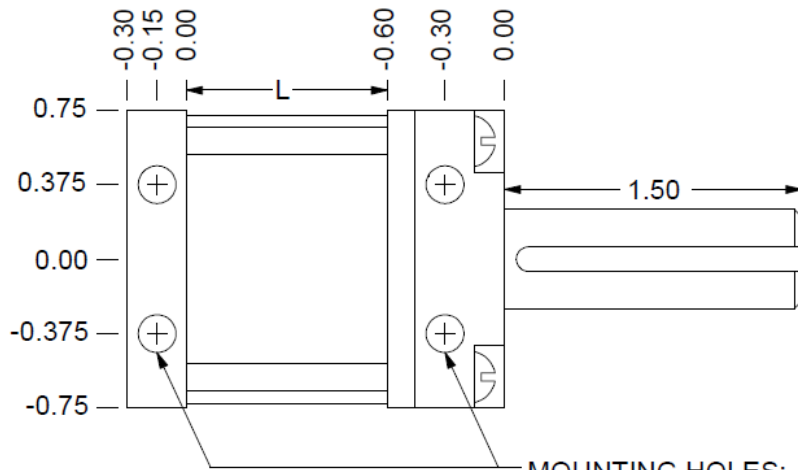
RINGGEAR LENGTH (L)

- 1 STAGE: 0.60
- 2 STAGE: 1.00
- 3 STAGE: 1.40
- 4 STAGE: 1.80

(TOTAL LENGTH = RINGGEAR LENGTH + 0.90)

STANDARD SHAFT:

- 4140 HARDENED STEEL
- 0.500 DIAMETER
- 0.125 KEYWAY TO SHAFT END
- #10-32 END TAP, 0.300 MIN DEPTH

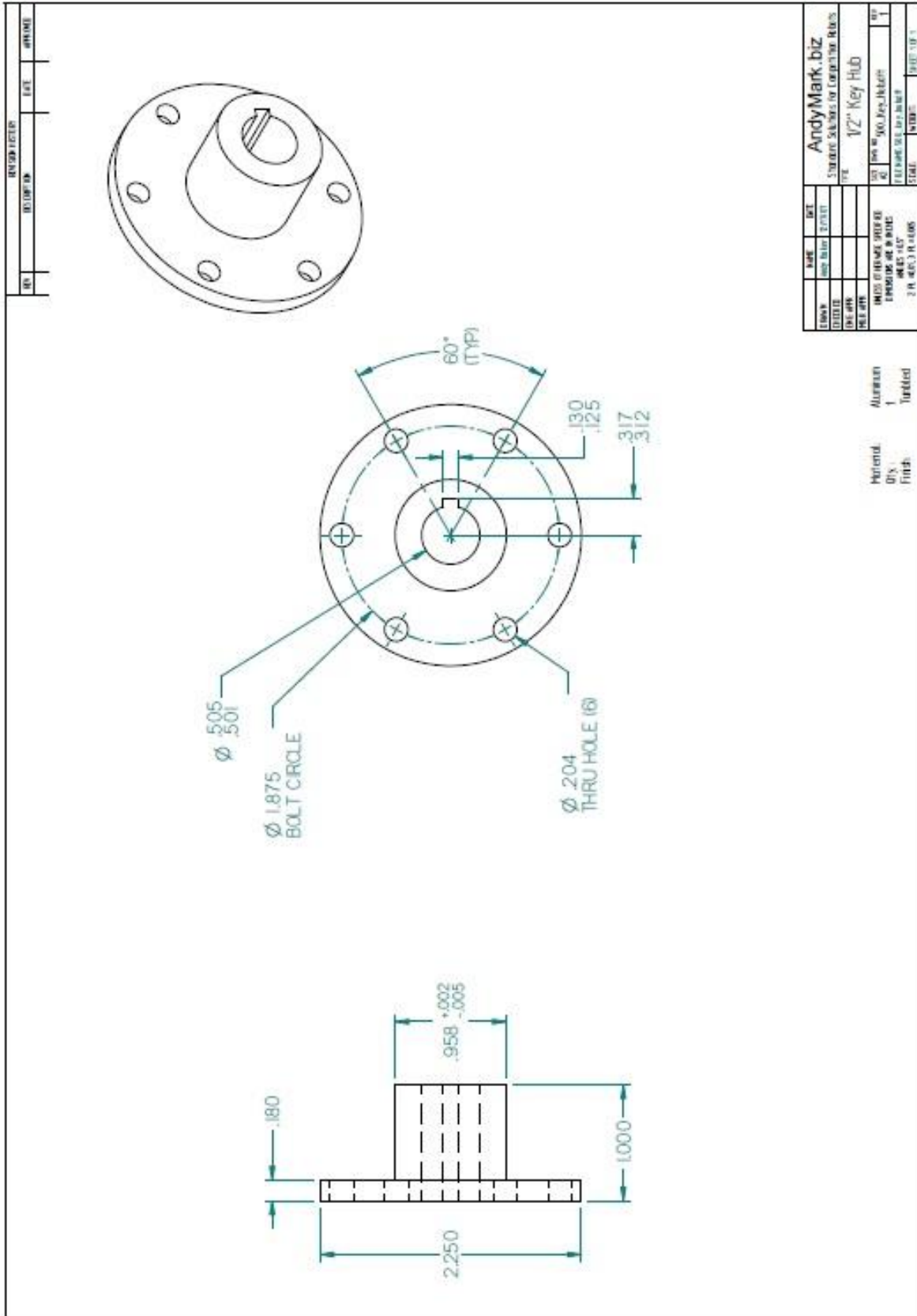


MOUNTING HOLES:
#10-32
8 TOTAL
4 TOP, 4 BOTTOM
0.30 MIN DEPTH

ALL DIMENSIONS INCHES
NO HIDDEN LINES SHOWN

**PRELIMINARY
11 JAN 09**

XII. Appendix E: Hub key dimension [7]



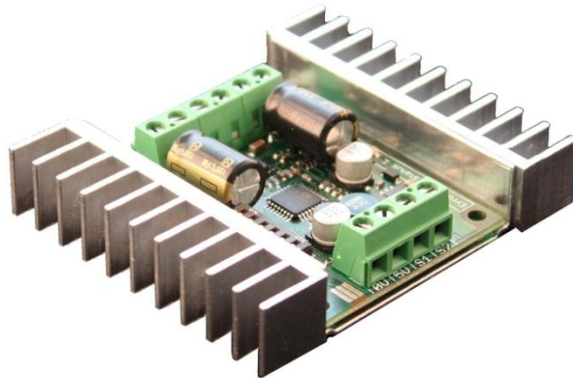
XIII. Appendix F: DC Motor Controller [8]



DimensionEngineering

Sabertooth 2x12 User's Guide

November 2010



Input voltage: 6-24V nominal, 30V absolute max.

Output Current: Up to 12A continuous per channel. Peak loads may be up to 25A per channel for a few seconds. These ratings are for input voltages up to 18v in still air without additional heatsinking.

5V Switching BEC: Up to 1A continuous and 1.5A peaks across the entire range of input voltages.

Recommended power sources are:

- 5 to 18 cells NiMH or NiCd
- 2s to 6s lithium ion or lithium polymer. Sabertooth motor drivers have a lithium battery mode to prevent cell damage due to over-discharge of lithium battery packs.
- 6v to 24v lead acid
- 6v to 24v power supply (when in parallel with a suitable battery).

Dimensions:

Size: 2.5" x 2.95" x .6" 64 x 75 x 16mm

Weight: 2.2oz

Features

Mixed and independent options:

Sabertooth features mixed modes designed especially for differential drive robots, where two motors provide both steering and propulsion. It also has independent options in all operating modes. This is useful for if you have two motors to control, but they aren't necessarily being used to drive a differential drive robot. The motors do not need to be matched or even similar, as long as they both are within Sabertooth's operating limits.

Synchronous regenerative drive:

Going one step farther than just regenerative braking, a Sabertooth motor driver will return power to the battery any time a deceleration or motor reversal is commanded. This can lead to dramatic improvements in run time for systems that stop or reverse often, like a placement robot or a vehicle driving on hilly terrain. This drive scheme also saves power by returning the inductive energy stored in the motor windings to the battery each switching cycle, instead of burning it as heat in the motor windings. This makes part-throttle operation very efficient.

Ultra-sonic switching frequency:

Sabertooth 2x12 features a PWM frequency of 32kHz, which is well above the maximum frequency of human hearing. Unlike some other motor drivers, there is no annoying whine when the motor is on, even at low power levels.

Thermal and overcurrent protection:

Sabertooth features dual temperature sensors and overcurrent sensing. It will protect itself from failure due to overheating, overloading and motor shorts.

Easy mounting and setup:

Sabertooth has screw terminals for all inputs and outputs. There are four mounting holes, which accept 4-40 screws. Mounting hardware is included. All operating modes and options are set with DIP switches – there are no jumpers to struggle with or lose. No soldering is required.

Compact Size:

Sabertooth utilizes surface mount construction to provide the most power from a compact package. Its small size and light weight mean you have more space for cargo, batteries, or can make your robot smaller and more nimble than the competition.

Carefree reversing:

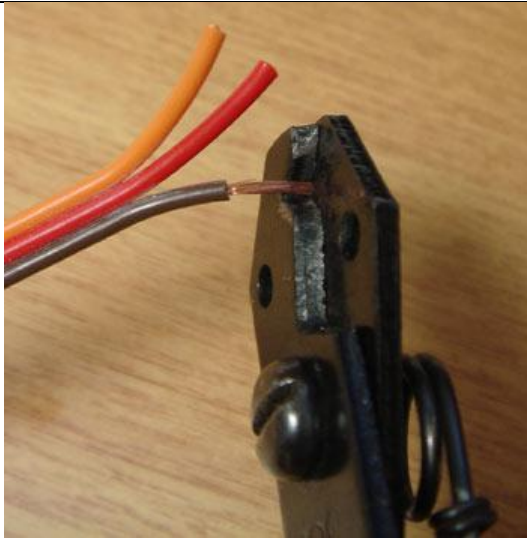
Unlike some other motor drivers, there is no need for the Sabertooth to stop before being commanded to reverse. You can go from full forward immediately to full reverse or vice versa. Braking and acceleration are proportional to the amount of reversal commanded, so gentle or rapid reversing is possible.

Many operating modes:

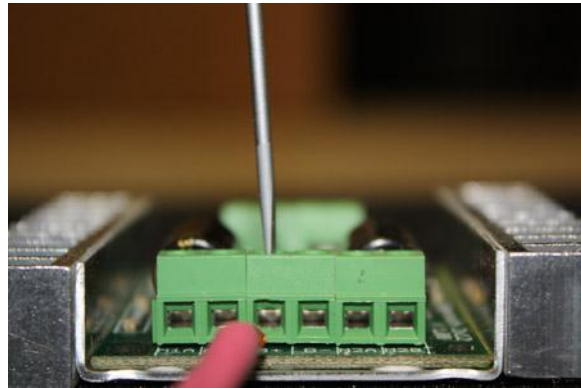
With analog, R/C and serial input modes, as well as dozens of operating options, the Sabertooth has the flexibility to be used over and over, even as your projects grow more sophisticated. Yet it is simple enough to use for your first robot project.

Hooking up the Sabertooth motor driver

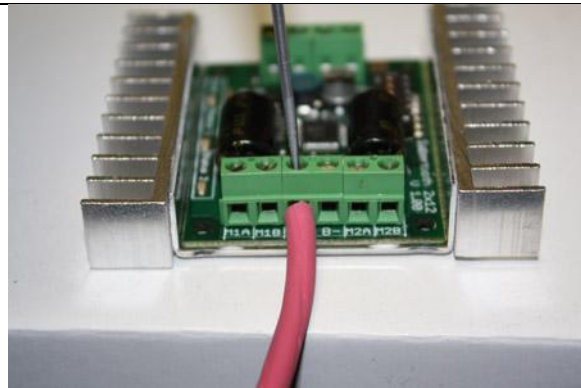
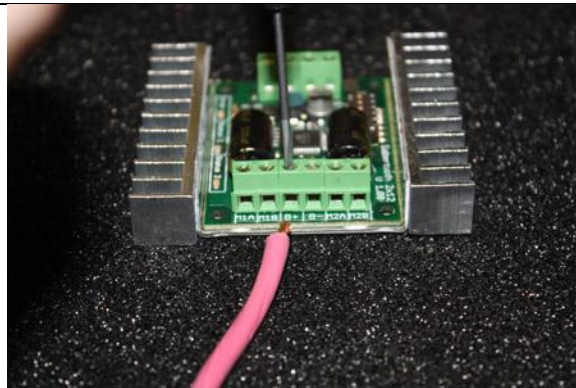
All connections to the Sabertooth are done with screw terminals. This makes it easy to set up and reconfigure your project. If you've never used screw terminal connections before, here is a quick overview.



Step 1: Strip the wire which you are using approximately $\frac{1}{4}$ " The wires may be 14 gauge to 30 gauge



Step 2: With a small screwdriver, turn the top screw counter-clockwise until it stops gently.



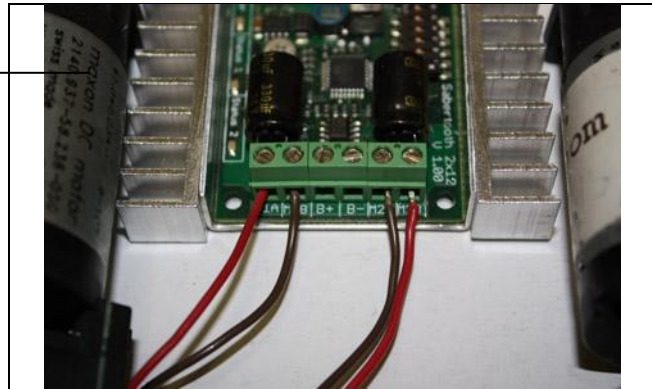
Step 3: Insert the stripped portion of the wire into the opening in the screw terminal

Step 4: Turn the top screw clockwise until you encounter resistance, then tighten the screw firmly. Pull on the wire gently to ensure that it is secured.

Motor1 Terminals

Motor 1 is connected to terminals M1A and M1B as shown below. If the motor runs in the opposite way that you want, you may reverse the motor wires to reverse rotation.

Motor 2 is connected to terminals M2A and M2B

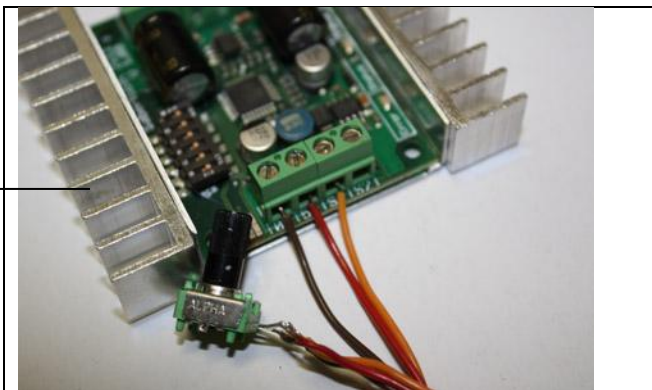


The motors connect to terminals M1A/B and M2A/B

Signal Input Terminals

S1 and S2

The input signals that control the Sabertooth are connected to terminals S1 and S2. If you are running in analog mode, it is important to have both the signal connected before applying power

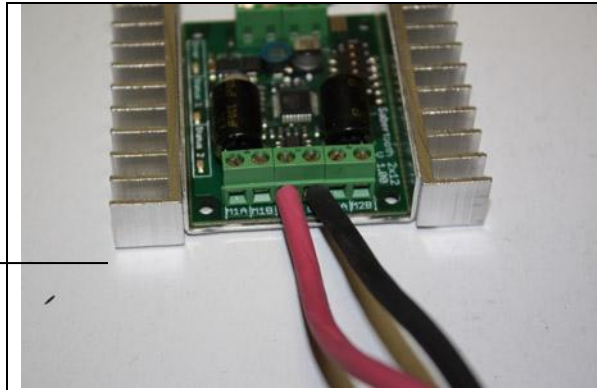


The input signals connect to terminal S1 and/or S2

to the device. Otherwise, the motors may start unexpectedly.

Battery Terminals

B+ and B-

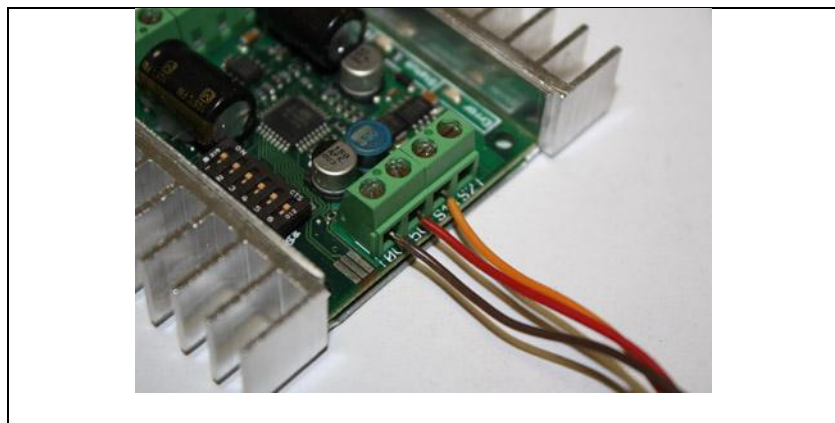


The battery or power supply is connected to terminals B- and B+. B- connects to the negative side of the battery (usually

The battery connects to terminals B+ and B-

black.) B+ connects to the positive side of the battery (usually red or yellow.) It is usually best to connect the battery through a connector instead of directly to the motor driver. This makes it easy to unplug the battery for charging, and prevents plugging in the battery backwards.

Warning! Be very careful to wire and plug in the battery and connector correctly. Connecting the battery backwards will destroy the Sabertooth and will void the warranty.



Power terminals

The 5V terminal can be used to power loads totaling no more than 1A, like a potentiometer or a radio receiver and a servo to two. The 0V signal must be connected to the ground of the device generating the input signal.

0V and 5V

The 0V and 5V connections are used to power and interface to low-power control circuits.

The 5V connection is a 5v power **output**. The 2x12 utilizes a **1A switching BEC** to power the onboard electronics as well as to provide power to your receiver and up to 4 standard analog servos. You can power anything that requires 5V straight from the Sabertooth 2x12. There is no need for an external BEC unless you need more than 1 Amp! The BEC will work at full rated output throughout the Sabertooth's operating voltage range. You can use the BEC at full capacity whether you are running 7V or 24V in!

The 0V connection is the signal ground for the Sabertooth. In order to receive input signals correctly, it must be connected to the ground of the device sending the signals.

Using the 0V and 5v connections to power a radio receiver in R/C mode and potentiometer in analog mode is shown in Figures 2.1 and 2.2. ***If you are using multiple Sabertooths running from the same radio receiver, only one should have the 5v line connected.*** You can either take the red lead out of the connection housing or just clip the wire with a pair of cutters.

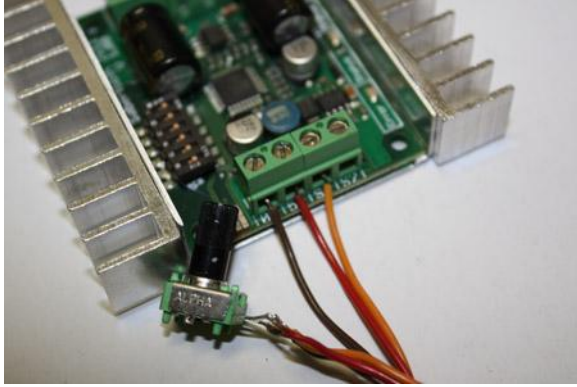


Figure 2.1: Analog input using a potentiometer powered from terminal 5V

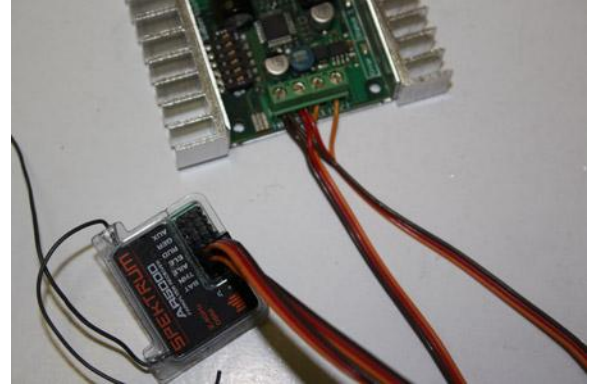
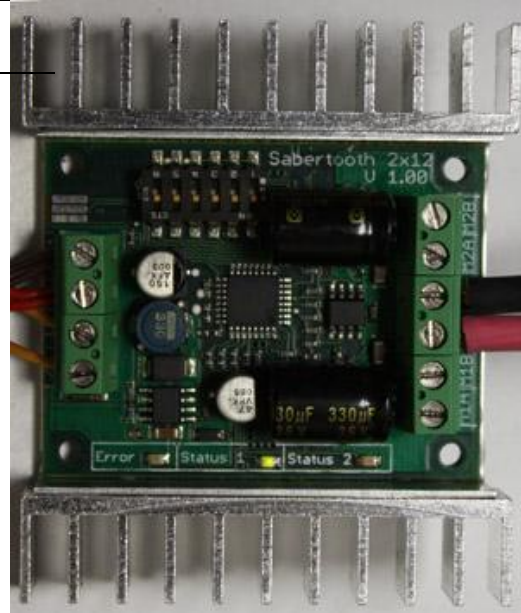


Figure 2.2: R/C input using a receiver powered from terminal 5V

Status and Error LEDs

Sabertooth 2x12 has three indicator LEDs.

The green LED marked Status1 is used to communicate various information about the current state. In most cases Status1 acts as a power indicator. In R/C mode, it glows dimly if there is no RC link present and brightly if there is an RC link.



Status1 LED on

The green LED marked Status2 is used in lithium mode. It blinks to indicate the number of lithium cells detected. Also, when Status2 and Error are flashing at the same time, and you are experiencing no output from the motors, the unit is displaying low voltage mode. Charge your batteries first, and if that does not work, you will need a larger battery. Keep in mind that when a lot of current is pulled from a battery, the voltage will drop. The more depleted the battery, the worse the voltage drop. The Sabertooth will hold the unit in the error state for longer than the battery voltage drops. This is to stop the unit from stuttering and gives the user enough time to diagnose the flashing LEDs.

The red Error LED illuminates if the Sabertooth has detected a problem. It will light if the driver has shut down due to overheating or overcurrent.

Mounting your Sabertooth 2x12

The Sabertooth is supplied with four mounting holes. These can be used to attach it to your robot. The centers of the mounting holes form a 1.5" x 2" rectangle. The holes are .125 inches in diameter. The proper size screw is a 4-40 round head machine or wood screw. Four 5/8" long machine screws and nuts are included.

If your robot or device is constructed from insulating materials such as wood or plastic, it may be necessary to mount the Sabertooth on standoffs to allow air to circulate. This is shown in Figure 2.3

If your robot or device is constructed from metal, it is usually better to attach the bottom heat spreader of the Sabertooth directly to the frame, without standoffs. This will allow your frame to act as a heat sink and will cause the Sabertooth to run cooler. If your chassis is grounded, you must insulate the heatsink from the chassis. This is shown in Figure 2.4



Figure 2.3: Mounted to a wood frame using standoffs

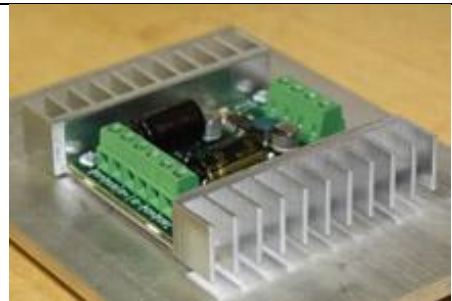


Figure 2.4: Mounted directly to a metal frame

Operating Modes Overview

Mode 1: Analog Input

Analog input mode takes one or two analog inputs and uses those to set the speed and direction of the motor. The valid input range is 0v to 5v. This makes the Sabertooth easy control using a potentiometer, the PWM output of a microcontroller (with an RC filter) or an analog circuit. Major uses include joystick or foot-pedal controlled vehicles, speed and direction control for pumps and machines, and analog feedback loops.

Mode 2: R/C Input

R/C input mode takes two standard R/C channels and uses those to set the speed and direction of the motor. There is an optional timeout setting. When timeout is enabled, the motor driver will shut down on loss of signal. This is for safety and to prevent the robot from running away should it encounter interference and should be used if a radio is being used to control the driver. If timeout is disabled, the motor driver will continue to drive at the commanded speed until another command is given. This makes the Sabertooth easy to interface to a Basic Stamp or other low-speed microcontrollers.

Mode 3: Simplified serial.

Simplified serial mode uses TTL level RS-232 serial data to set the speed and direction of the motor. This is used to interface the Sabertooth to a PC or microcontroller. If using a PC, a level converter such as a MAX232 chip must be used. The baud rate is set via DIP switches. Commands are single-byte. There is also a Slave Select mode which allows the use of multiple Sabertooth 2x12 from a single microcontroller serial port.

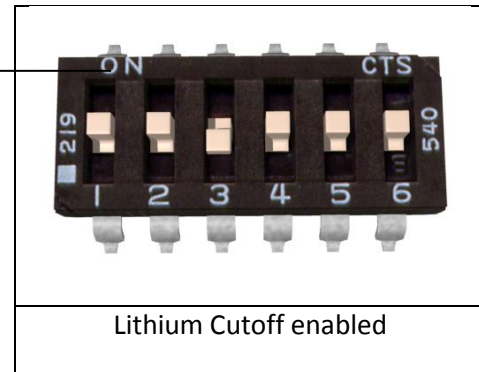
Mode 4: Packetized serial

Packetized serial mode uses TTL level RS-232 serial data to set the speed and direction of the motor. There is a short packet format consisting of an address byte, a command byte, a data byte and a 7 bit checksum. Packetized serial automatically detects the transmitted baud rate based on the first character sent, which must be 170. Address bytes are set via dip switches. Up to 8 Sabertooth motor drivers may be ganged together on a single serial line. This makes packetized serial the preferred method to interface multiple Sabertooths to a PC or laptop. Because Sabertooth uses the same protocol as our SyRen single motor drivers, both can be used together from the same serial master.

Lithium cutoff:

Switch 3 of the DIP switch block selects lithium cutoff. If switch 3 is in the down position as shown the Sabertooth will automatically detect the number of series lithium cells at startup, and set a cutoff voltage of 3.0 volts per cell. The number of detected cells is

flashed out on the Status LED. If the number of cells detected is too low, your battery is in a severely discharged state and must be charged before operation. Failure to do so may cause damage to the battery pack. When 3.0V per cell is reached, the Sabertooth will shut down, preventing damage to the battery pack. This is necessary because a lithium battery pack discharged below 3.0v per cell will lose capacity and batteries discharged below 2.0v per cell may not ever recharge. Lithium cutoff mode may also be useful to increase the number of battery cycles you can get when running from a lead acid battery in non-critical applications. Because the system will continue to draw some power, even with the motor shut down, it is important to unplug the battery from the Sabertooth promptly once the cutoff is reached when using lithium batteries. If the Sabertooth is being run from NiCd, NiMH or alkaline batteries, or from a power supply, switch 3 should be in the up position.



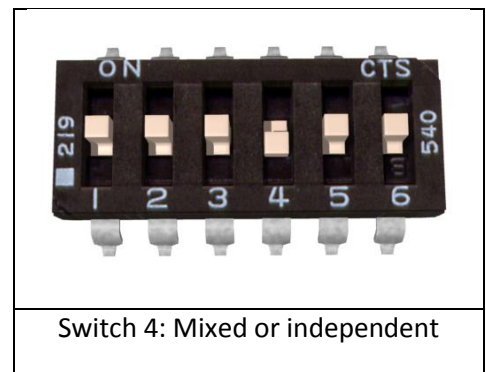
Mode 1: Analog Input

Analog input mode is selected by setting switches 1 and 2 to the UP position. Switch 3 should be either up or down, depending on the battery type being used. Inputs S1 and S2 are configured as analog inputs. The output impedance of the signals fed into the inputs should be less than 10k ohms for best results. If you are using a potentiometer to generate the input signals, a 1k, 5k or 10k linear taper pot is recommended. In all cases, an analog voltage of 2.5V corresponds to no movement. Signals above 2.5V will command a forward motion and signals below 2.5V will command a backwards motion.

There are three operating options for analog input. These are selected with switches 4, 5 and 6. All the options can be used independently or in any combination.

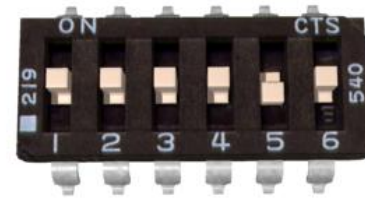
Switch 4: Mixing Mode

If switch 4 is in the UP position, the Sabertooth 2x12 is in **Mixed** mode. This mode is designed for easy steering of differential-drive vehicles. The analog signal fed into S1 controls the forward/back motion of the vehicle, and the analog signal fed into S2 controls the turning motion of the vehicle. If Switch 4 is in the DOWN position, the Sabertooth 2x12 is in Independent mode. In Independent mode, the signal fed to S1 directly controls Motor 1 (outputs M1A and M1B) and the signal fed to S2 controls Motor 2.



Switch 5: Exponential response

If switch 5 is in the DOWN position, the response to input signals will be exponential. This softens control around the zero speed point, which is useful for control of vehicles with fast top speeds or fast max turning rates. If switch 5 is in the UP position, the response is linear.

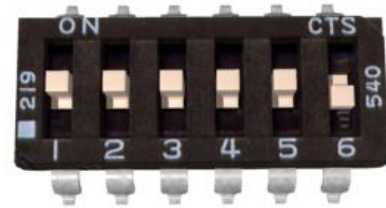


Switch 5: Exponential response

Switch 6: 4x sensitivity

If switch 6 is in the UP position, the input signal range is from 0v to 5v, with a zero point of 2.5v.

If switch 6 is in the DOWN position, 4x sensitivity mode is enabled. In this mode, the input signal range is from 1.875V to 3.125V, with a zero point of 2.5v. This is useful for building analog feedback loops



Switch 6: 4x sensitivity

Low pass filter for using PWM in Analog Mode

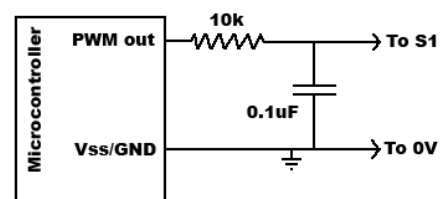


Figure 4.1: Filtered PWM

Note on using filtered PWM in Analog Mode

If you are using a filtered PWM signal from a microcontroller to generate the analog voltage, an R/C filter with component values 10k ohms and at least .1uf is recommended as shown in **Figure 4.1**. Using a larger value filter capacitor such as 1uf or 10uf will result in smoother motor operation, at a cost of slower transient response. A PWM frequency higher than 1000Hz is recommended.

Mode 2: R/C Input

R/C input mode is used with a standard hobby Radio control transmitter and receiver, or a microcontroller using the same protocol. R/C mode is selected by setting switch 1 to the DOWN position and switch 2 to the UP position. If running from a receiver, it is necessary to obtain one or more servo pigtails and hook them up according to figure 5.1. If using a receiver pack, do not connect power to the 5V line of the Sabertooth because the maximum voltage it can tolerate is 6V.

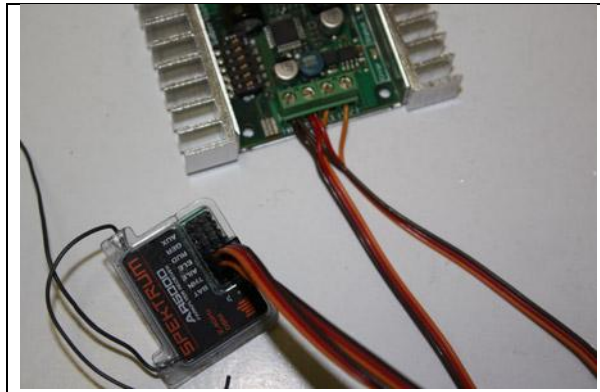
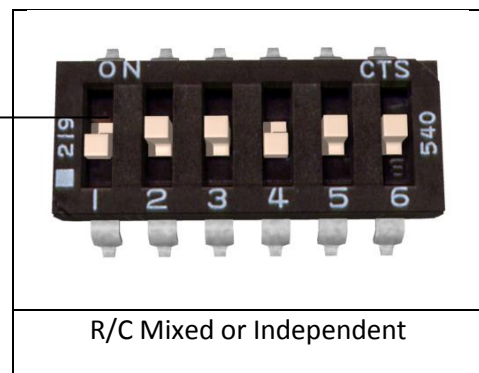


Figure 5.1: R/C connection

There are three operating options for R/C mode. These are selected with switches 4, 5 and 6.

Switch 4: Mixing Mode

When Switch 4 is in the UP position, Mixed mode is selected. In this mode, the R/C signal fed to the S1 input controls the forward/backwards motion of the vehicle. This is usually connected to the throttle channel of a



R/C Mixed or Independent

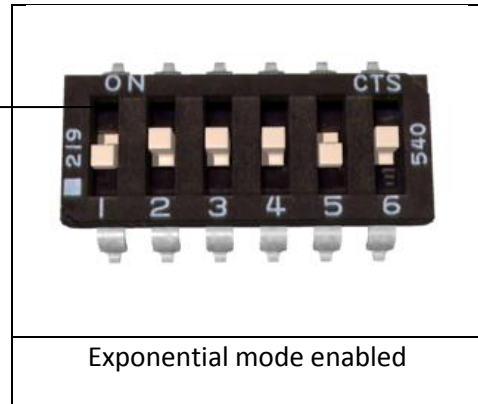
pistol grip transmitter, or the elevator channel of a dual stick transmitter. The R/C signal fed to the S2 input controls the turning of the vehicle.

When switch 4 is in the DOWN position, Independent mode is selected. In this mode, the signal fed to the S1 input directly controls Motor 1 (M1A and M1B) and the signal fed to S2 controls Motor 2.

Switch 5: Exponential response

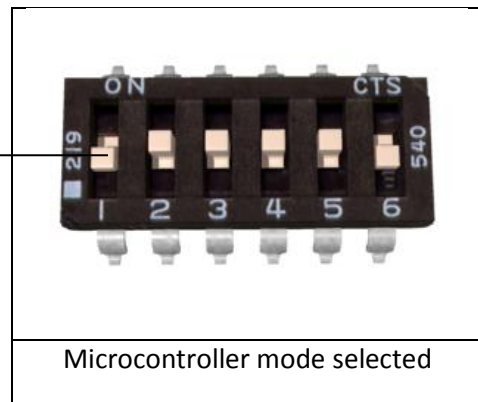
If switch 5 is in the UP position, the response is linear.

If switch 5 is in the DOWN position, the response to input signals will be exponential. This softens control around the zero speed point, which is useful for control of vehicles with fast top speeds or fast max turning rates.



Switch 6: R/C Mode/Microcontroller mode select

If switch 6 is in the UP position, then the Sabertooth is in standard R/C mode. This mode is designed to be used with a hobby-style transmitter and receiver. It automatically calibrates the control center and endpoints to maximize stick usage. It also enables a Timeout Failsafe, which will shut down the motors if the Sabertooth stops receiving correct signals from the receiver.



If switch 6 is set in the DOWN position, then Microcontroller mode is enabled. This disables the Timeout Failsafe and auto-calibration. This means that the Sabertooth will continue to drive the motor according to the last command until another command is given. If the control link is possible unreliable – like a radio - then this can be dangerous due to the robot not stopping. However, it is extremely convenient if you are controlling the Sabertooth from a microcontroller. In this case, commanding the controller can be done with as little as three lines of code.

Output_High(Pin connected to S1)

Delay(1000us to 2000us)

Output_Low(Pin connected to S1)

A note on certain microprocessor receivers

Some receivers, such as the Spektrum AR6000, will output servo pulses before a valid transmitter signal is present. This will cause the Sabertooth to autocalibrate to the receiver's startup position which may not correspond to the center stick position, depending on trim settings. This may cause the motors to move slowly, even when the transmitter stick is centered. If you encounter this, either consult your receiver manual to reprogram the startup position, or adjust your transmitter trims until the motors stop moving. As a last resort, you can enter R/C microcontroller mode which will disable Sabertooth's autocalibration.

Mode 3: Simplified Serial Mode

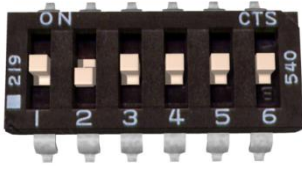
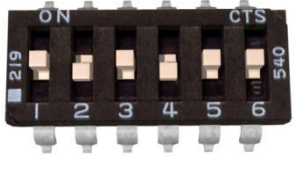
Simplified serial uses TTL level single-byte serial commands to set the motor speed and direction. This makes it easy to interface to microcontrollers and PCs, without having to implement a packet-based communications protocol. Simplified serial is a one-direction only interface. The transmit line from the host is connected to S1. The host's receive line is not connected to the Sabertooth. Because of this, multiple drivers can be connected to the same serial transmitter. If using a true RS-232 device like a PC's serial port, it is necessary to use a level converter to shift the -10V to 10V rs-232 levels to the 0v-5v TTL levels the Sabertooth is expecting. This is usually done with a Max232 type chip. If using a TTL serial device like a microcontroller, the TX line of the microcontroller may be connected directly to S1.

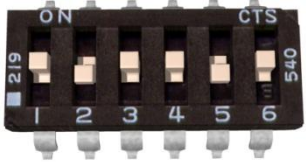
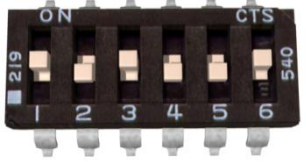
Because Sabertooth controls two motors with one 8 byte character, when operating in Simplified Serial mode, each motor has 7 bits of resolution. Sending a character between 1 and 127 will control motor 1. 1 is full reverse, 64 is stop and 127 is full forward. Sending a character between 128 and 255 will control motor 2. 128 is full reverse, 192 is stop and 255 is full forward.

Character 0 (hex 0x00) is a special case. Sending this character will shut down both motors.

Baud Rate Selection

Simplified Serial operates with an 8N1 protocol – 8 data bytes, no parity bits and one stop bit. The baud rate is selected by switches 4 and 5 from the following 4 options

	
2400 Baud: 01x00x	9600 Baud: 01x10x

	
19200 Baud: 01x01x	38400 Baud: 01x11x

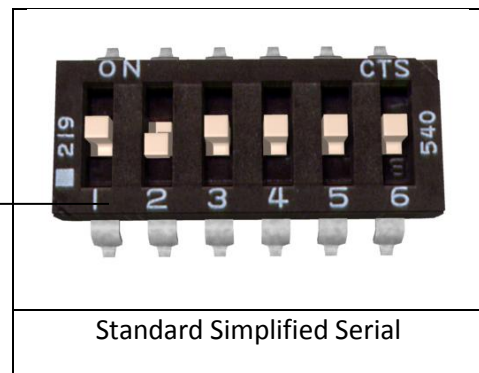
What baud rate to use is dependent on what your host can provide and the update speed necessary. 9600 baud or 19200 baud is recommended as the best starting points. If communication is unreliable, decrease the baud rate. If communications are reliable, you may increase the baud rate. The maximum update speed on the Sabertooth is approximately 2000 commands per second. Sending characters faster than this will not cause problems, but it will not increase the responsiveness of the controller either.

The baud rate may be changed with power on by changing the DIP switch settings. There is no need to reset or cycle power after a baud rate change.

There are 2 operating options for Simplified Serial. These are selected by the position of Switch 6.

Option 1: Standard Simplified Serial Mode

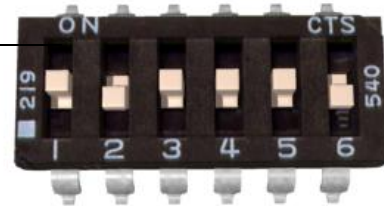
Serial data is sent to input S1. The baud rate is selected with switches 4 and 5. Commands are sent as single bytes. Sending a value of 1-127 will command motor 1. Sending a value of 128-255 will command motor 2. Sending a value of 0 will shut down both motors.



Option 2: Simplified Serial with Slave Select

This mode is used when it is desirable to have multiple Sabertooth motor drivers running from the same serial transmitter, but you do not wish to use packetized serial. A digital signal (0v or 5v) is fed to the S2 input.

This is controlled by the host microcontroller. If the signal on S2 is logic high (5v) when the serial command is sent, the driver will change to the new speed. If the signal on S2 is not high when the command is sent, then command will be ignored. Pseudo-code demonstrating this is shown below. After sending the signal, allow about 50 us before commanding the Slave Select line to a logic LOW to allow time for processing. A hookup diagram and example pseudo-code are shown in **Figures 6.2** and **6.3**.



Simplified Serial with Slave Select

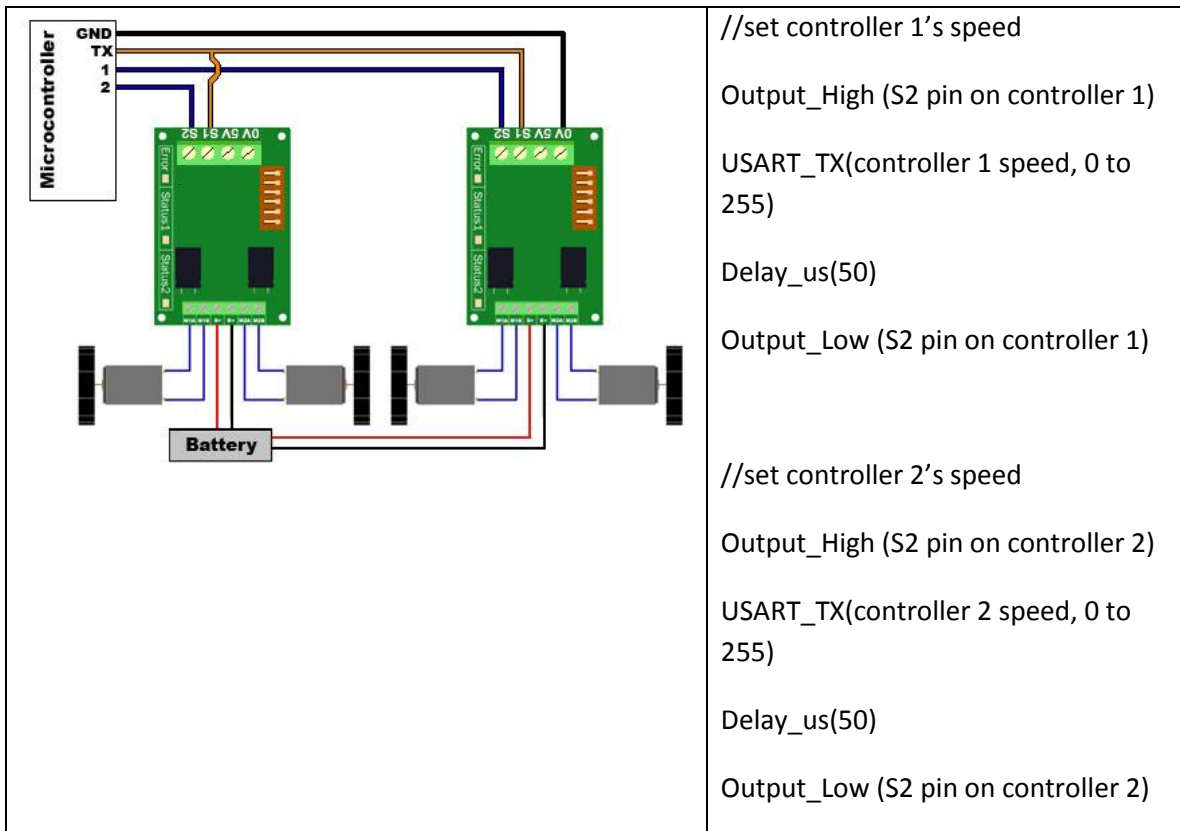


Figure 6.2: Hookup for Slave Select

Figure 6.3: Pseudocode for Slave Select

Mode 4: Packetized Serial Mode

Packetized Serial uses TTL level multi-byte serial commands to set the motor speed and direction. Packetized serial is a one-direction only interface. The transmit line from the host is connected to S1. The host's receive line is not connected to the Sabertooth. Because of this, multiple Sabertooth 2x12 motor drivers can be connected to the same serial transmitter. It is also possible to use SyRen and Sabertooth motor drivers together from the same serial source, as well as any other serial device, as long as it will not act on the packets sent to the Sabertooth. If using a true RS-232 device like a PC's serial port, it is necessary to use a level converter to shift the -10V to 10V rs-232 levels to the 0v-5v TTL. Packetized serial uses an address byte to select the target device.

Packet Overview

The packet format for the Sabertooth consists of an address byte, a command byte, a data byte and a seven bit checksum. Address bytes have value greater than 128, and all subsequent bytes have values 127 or lower. This allows multiple types of devices to share the same serial line.

An example packet and pseudo-code to generate it are shown in **Figures 7.1** and **7.2**

<u>Packet</u>	Void DriveForward(char address, char speed)
Address: 130	{
Command : 0	Putc(address);
Data: 64	Putc(0);
Checksum: 66	Putc(speed);
	Putc((address + 0 + speed) & 0b01111111);
	}

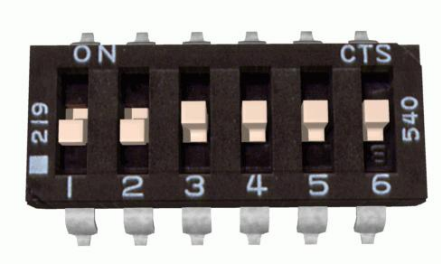
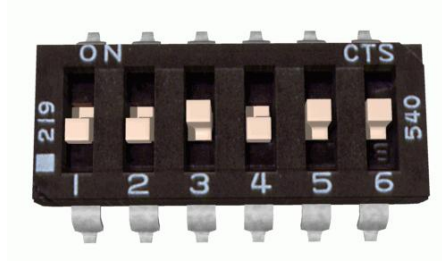
Figure 7.1: Example 50% forward	Figure 7.2: Pseudocode to generate 7.1

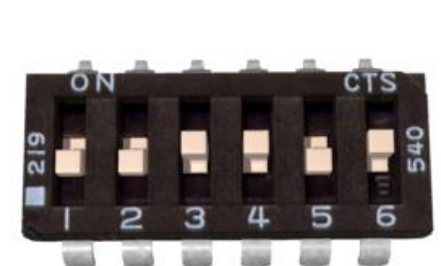
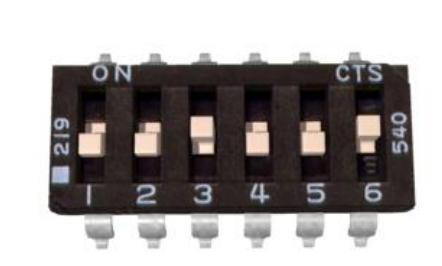
Baud Rate Selection:

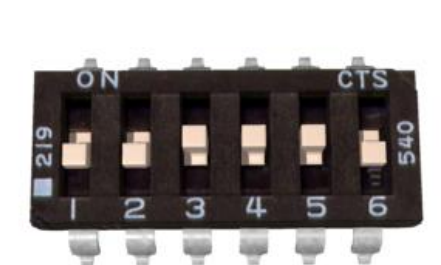
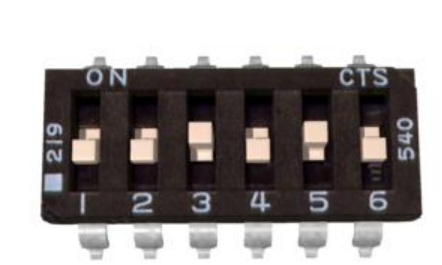
Packetized Serial operates with an 8N1 protocol – 8 data bytes, no parity bits and one stop bit. The baud rate is set at **9600 baud** from the factory. This value can be changed by sending the proper baud rate selection packet once the unit has powered on. Changed baud rates will be active after a power cycle. Once you set it, it stays that way until you change the rate again. See the 'Setting Commands' page for further details on how to change the baud rate.

Address Byte Configuration:

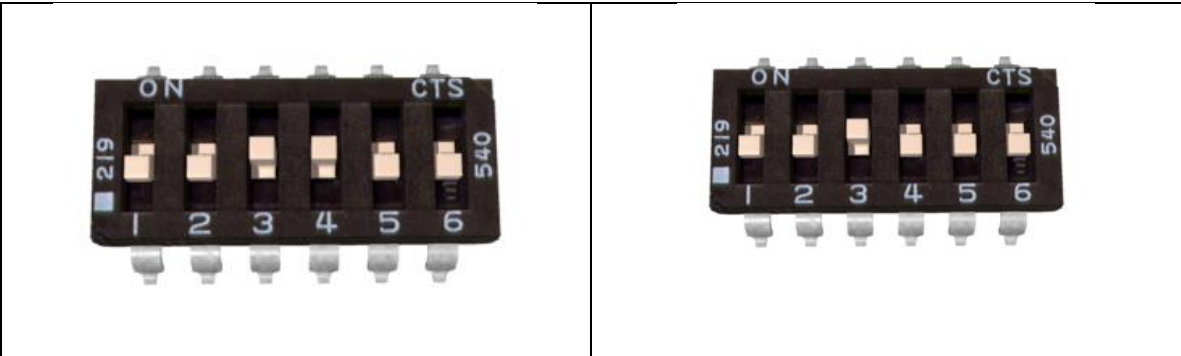
Address bytes are set by switches 4, 5 and 6. Addresses start at 128 and go to 135. The switch settings for the addresses are shown in the chart below

 A 6-pin DIP switch assembly with pins labeled 1 through 6. Switches 1, 2, and 3 are in the 'ON' position. Switches 4, 5, and 6 are in the 'OFF' position. Labels 'ON' and 'CTS' are above the switches, and '219' and '540' are on the sides.	 A 6-pin DIP switch assembly with pins labeled 1 through 6. Switches 1, 2, and 3 are in the 'ON' position. Switches 4 and 5 are in the 'OFF' position, and switch 6 is in the 'ON' position. Labels 'ON' and 'CTS' are above the switches, and '219' and '540' are on the sides.
Address: 128	Address: 129

 A 6-pin DIP switch assembly with pins labeled 1 through 6. Switches 1, 2, and 3 are in the 'ON' position. Switches 4 and 5 are in the 'OFF' position, and switch 6 is in the 'ON' position. Labels 'ON' and 'CTS' are above the switches, and '219' and '540' are on the sides.	 A 6-pin DIP switch assembly with pins labeled 1 through 6. Switches 1, 2, and 3 are in the 'ON' position. Switches 4 and 5 are in the 'OFF' position, and switch 6 is in the 'ON' position. Labels 'ON' and 'CTS' are above the switches, and '219' and '540' are on the sides.
Address: 130	Address: 131

 A 6-pin DIP switch assembly with pins labeled 1 through 6. Switches 1, 2, and 3 are in the 'ON' position. Switches 4 and 5 are in the 'OFF' position, and switch 6 is in the 'ON' position. Labels 'ON' and 'CTS' are above the switches, and '219' and '540' are on the sides.	 A 6-pin DIP switch assembly with pins labeled 1 through 6. Switches 1, 2, and 3 are in the 'ON' position. Switches 4 and 5 are in the 'OFF' position, and switch 6 is in the 'ON' position. Labels 'ON' and 'CTS' are above the switches, and '219' and '540' are on the sides.
---	--

Address: 132	Address: 133
--------------	--------------



Address: 134	Address: 135
--------------	--------------

Commands:

The command byte is the second byte of the packet. There are four possible commands in packetized serial mode. Each is followed by one byte of data

0: Drive forward motor 1 (decimal 0, binary 0b00000000, hex 0h00)

This is used to command motor 1 to drive forward. Valid data is 0-127 for off to full forward drive. If a command of 0 is given, the Sabertooth will go into power save mode for motor 1 after approximately 4 seconds.

1: Drive backwards motor 1 (decimal 1, binary 0b00000001, hex 0h01)

This is used to command motor 1 to drive backwards. Valid data is 0-127 for off to full reverse drive. If a command of 0 is given, Sabertooth will go into power save mode for motor 1 after approximately 4 seconds.

2: Min voltage (decimal 2, binary 0b00000010, hex 0h02)

This is used to set a custom minimum voltage for the battery feeding the Sabertooth. If the battery voltage drops below this value, the output will shut down. This value is cleared at startup, so much be set each run. The value is sent in .2 volt increments with a command of zero corresponding to 6v, which is the minimum. Valid data is from 0 to 120. The function for converting volts to command data is

Value = (desired volts-6) x 5

3: Max voltage (decimal 3, binary 0b00000011, hex 0h03)

This is used to set a custom maximum voltage. If you are using a power supply that cannot sink current such as an ATX supply, the input voltage will rise when the driver is regenerating (slowing down the motor) Many ATX type supplies will shut down if the output voltage on the 12v supply rises beyond 16v. If the driver detects an input voltage above the set limit, it will put the motor into a hard brake until the voltage drops below the set point again. This is inefficient, because the energy is heating the motor instead of recharging a battery, but may be necessary. The driver comes preset for a maximum voltage of 30V. The range for a custom maximum voltage is 0v-25v. The formula for setting a custom maximum voltage is

Value = Desired Volts*5.12

If you are using any sort of battery, then this is not a problem and the max voltage should be left at the startup default.

4: Drive forward motor 2 (decimal 4, binary 0b00000100, hex 0h04)

This is used to command motor 2 to drive forward. Valid data is 0-127 for off to full forward drive. If a command of 0 is given, the Sabertooth will go into power save mode for motor 2 after approximately 4 seconds.

5: Drive backwards motor 2 (decimal 5, binary 0b00000101, hex 0h05)

This is used to command motor 2 to drive backwards. Valid data is 0-127 for off to full reverse drive. If a command of 0 is given, the Sabertooth will go into power save mode after approximately 4 seconds.

6: Drive motor 1 7 bit (decimal 6, binary 0b00000110, hex 0h06)

This command is used to drive motor 1. Instead of the standard commands 0 and 1, this one command can be used to drive motor 1 forward or in reverse, at a cost of lower resolution. A command of 0 will correspond to full reverse, and a command of 127 will command the motor to drive full forward. A command of 64 will stop the motor.

7: Drive motor 2 7 bit (decimal 7, binary 0b00000111, hex 0h07)

This command is used to drive motor 2. Instead of the standard commands 4 and 5, this one command can be used to drive motor 1 forward or in reverse, at a cost of lower resolution. A command of 0 will correspond to full reverse, and a command of 127 will command the motor to drive full forward. A command of 64 will stop the motor.

Mixed mode commands:

Sabertooth can also be sent mixed drive and turn commands. When using the mixed mode commands, please note that the Sabertooth requires valid data for both drive and turn before it will begin to operate. Once data for both has been sent, then each may be updated as needed, it is not necessary to send both data packets each time you wish to update the speed or direction. You should design your code to either use the independent or the mixed commands. Switching between the command sets will cause the vehicle to stop until new data is sent for both motors.

8: Drive forward mixed mode (decimal 8, binary 0b00001000, hex 0h08)

This is used to command the vehicle to drive forward in mixed mode. Valid data is 0-127 for off to full forward drive.

9: Drive backwards mixed mode (decimal 9, binary 0b00001001, hex 0h09)

This is used to command the vehicle to drive backwards in mixed mode. Valid data is 0-127 for off to full reverse drive.

10: Turn right mixed mode (decimal 10, binary 0b00001010, hex 0h0a)

This is used to command the vehicle to turn right in mixed mode. Valid data is 0-127 for zero to maximum turning speed.

11: Drive turn left mixed mode (decimal 11, binary 0b00001011, hex 0h0b)

This is used to command the vehicle to turn left in mixed mode. Valid data is 0-127 for zero to maximum turning speed.

12: Drive forwards/back 7 bit (decimal 12, binary 0b00001100, hex 0h0c)

This is used to command the vehicle to move forwards or backwards. A command of 0 will cause maximum reverse, 64 will cause the vehicle to stop, and 127 will command full forward.

13: Turn 7 bit (decimal 13, binary 0b00001101, hex 0h0d)

This is used to command the vehicle turn right or left. A command of 0 will cause maximum left turn rate, 64 will cause the vehicle to stop turning, and 127 will command maximum right turn rate.

Setting Commands

Several parameters of the Sabertooth 2x12 can be changed using Packetized Serial mode. Some of these changes persist when the unit is powercycled and some persist when it is switched to other modes.

14: Serial Timeout (decimal 14, binary 0b00001110, hex 0h0e)

This setting determines how long it takes for the motor driver to shut off if it has not received a command recently. Serial Timeout is off by default. A command of 0 will disable the timeout if you had previously enabled it. The timeout scales 1 unit per 100ms of timeout, so a command of 10 would make a timeout of 1000ms. This setting does **not** persist through a power cycle or in any mode other than Serial.

15: Baud Rate (decimal 15, binary 0b00001111, hex 0h0f)

This value remains until it is changed and does persist through a power cycle. The values are:

1: 2400 baud

2: 9600 baud (default)

3: 19200 baud

4: 38400 baud

16: Ramping (decimal 16, binary 0b00010000, hex 0h10)

This adjusts or disables the ramping feature found on the Sabertooth 2x50. This adjustment applies to all modes, even R/C and analog mode. Values between 1 and 10 are **Fast Ramp**; values between 11 and 20 are **Slow Ramp**; values between 21 and 80 are **Intermediate Ramp**.

Fast Ramping is a ramp time of $256/(\sim 1000 \times \text{Command value})$. Ramp time is the delay between full forward and full reverse speed.

1: 1/4 second ramp (default)

2: 1/8 second ramp

3: 1/12 second ramp

Slow and Intermediate Ramping are a ramp time of $256/[15.25 \times (\text{Command value} - 10)]$

See **Figures 8.1 and 8.2** in the Appendix for a graph of values.

17: Deadband (decimal 17, binary 0b00010001, hex 0h11)

This determines the extent of the Sabertooth's deadband – the range of commands close to “stop” that will be interpreted as stop. This setting applies to all modes and persists through a power cycle. The commands range from 0 to 127 and the formula is as follows:

$$127 - \text{command} < \text{motors off} < 128 + \text{command}$$

Thus, a command of 3 would shut the motors off with speed commands between 124 and 131.

A command of 0 sets the deadband to its default, which is $124 < \text{off} < 131$ in serial mode.

Checksum:

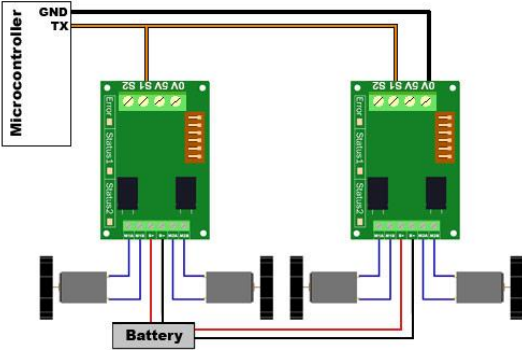
To prevent data corruption, each packet is terminated with a checksum. If the checksum is not correct, the data packet will not be acted upon. The checksum is calculated as follows:

Checksum = address byte + command byte + data byte

The checksum should be added with all unsigned 8 bit integers, and then ANDed with the mask 0b01111111 in an 8 bit system.

Example of Packetized Serial:

The following is an example function for commanding two Dimension Engineering motor drivers using Packetized Serial Mode. **Figure 7.3** shows an example hookup and **Figure 7.4** shows an example function.

	<pre>Void DriveForward(char address, char speed) { Putc(address); Putc(0); Putc(speed); Putc((address + 0 + speed) & 0b01111111); }</pre>
Figure 7.3: Packetized serial hookup	Figure 7.4: Packetized Serial Function

Example: So in this function, if address is 130, command is 0 (for driving forward), speed is 64, the checksum should calculate as follows:

$$130+0+64 = 194$$

194 in binary is 0b11000010

$$0b11000010 \& 0b01111111 = 0b01000010$$

Once all the data is sent, this will result in the Sabertooth with address 130 driving forward at roughly half throttle.

Emergency Stop:

In Packetized Serial mode, the S2 input is configured as an active-low emergency stop. It is pulled high internally, so if this feature isn't needed, it can be ignored. If an emergency stop is desired, all the S2 inputs can be tied together. Pulling the S2 input low will cause the driver to shut down. This should be tied to an emergency stop button if used in a device that could endanger humans.

Figure 8.1: Fast and Intermediate Ramp

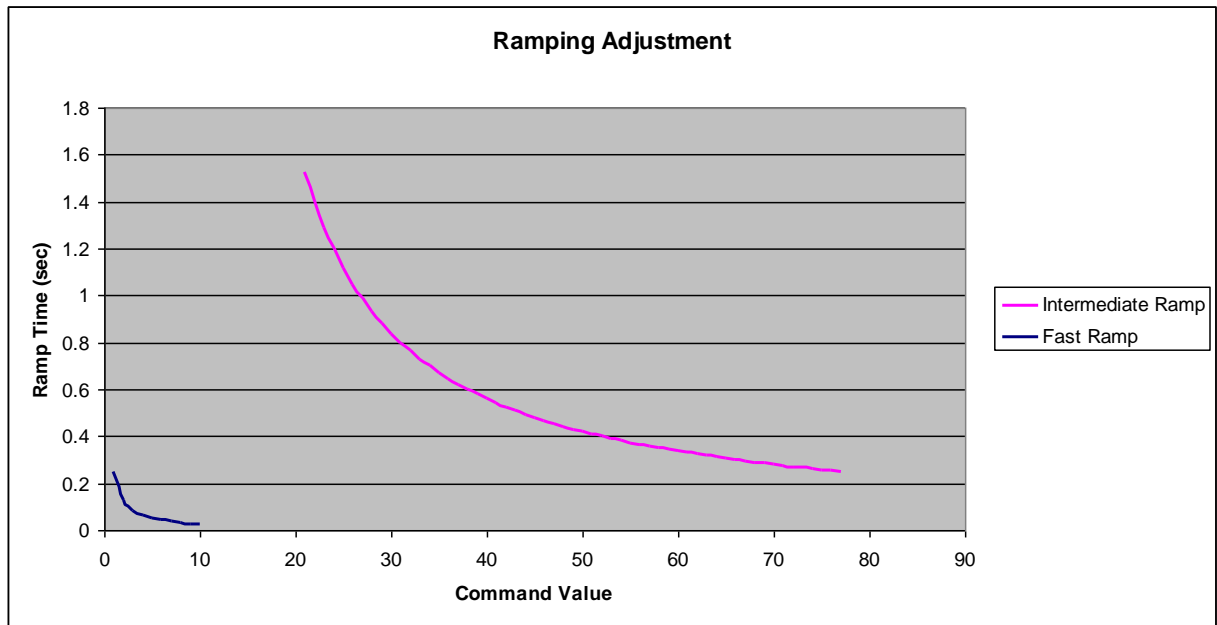
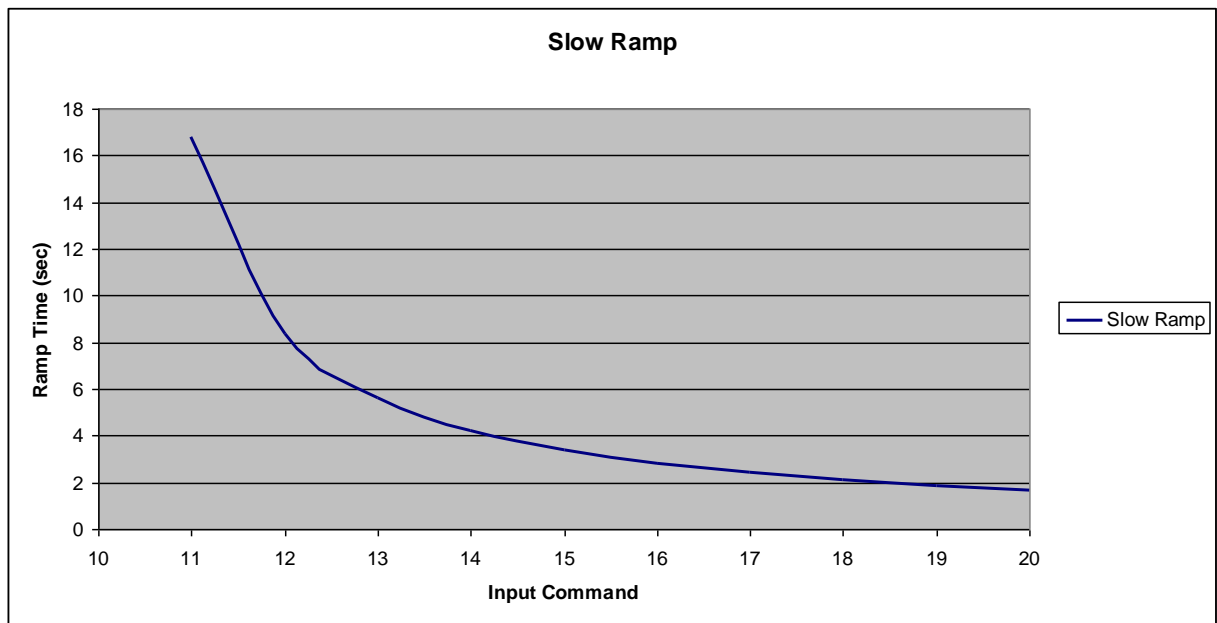


Figure 8.2: Slow Ramp



XVI. Appendix G: Report Video Links

1. Cleaning the rollers:

<http://www.youtube.com/watch?v=swEFxwaFELA&list=UULPI4grwwb-bSrqgJTe70kA&index=7&feature=plcp>

2. Assembling Mecanum wheel-A:

<http://www.youtube.com/watch?v=dUYrSBrMXG4&list=UULPI4grwwb-bSrqgJTe70kA&index=13&feature=plcp>

3. Assembling Mecanum wheel-B:

<http://www.youtube.com/watch?v=mnjQKnTE7Oo&list=UULPI4grwwb-bSrqgJTe70kA&index=12&feature=plcp>

4. Assembling Mecanum wheel-C:

<http://www.youtube.com/watch?v=HO-sOh3xxZo&list=UULPI4grwwb-bSrqgJTe70kA&index=11&feature=plcp>

5. Assembling Mecanum wheel-D:

<http://www.youtube.com/watch?v=qehzeLaXSuc&list=UULPI4grwwb-bSrqgJTe70kA&index=10&feature=plcp>

6. Assembling Mecanum wheel-E:

<http://www.youtube.com/watch?v=TU-S7gAIPzM&list=UULPI4grwwb-bSrqgJTe70kA&index=9&feature=plcp>

7. Greasing and assembling the gearbox:

http://www.youtube.com/watch?v=CObjLvuAt_Y&list=UULPI4grwwb-bSrqgJTe70kA&index=4&feature=plcp

8. Graining the bronze spacer:

<http://www.youtube.com/watch?v=jUCL66D3uTI&list=UULPI4grwwb-bSrqgJTe70kA&index=6&feature=plcp>

9. After graining the bronze spacer, attach it to the gearbox:

<http://www.youtube.com/watch?v=TU-S7gAIPzM&feature=autoplay&list=UULPI4grwwb-bSrqgJTe70kA&lf=plcp&playnext=1>

10. Cutting the aluminum bars:

<http://www.youtube.com/watch?v=1576TIWx328&list=UULPI4grwwb-bSrqgJTe70kA&index=5&feature=plcp>

11. Preparing for punching the aluminum bars:

<http://www.youtube.com/watch?v=7ghroz5BXOg&list=UULPI4grwwb-bSrqgJTe70kA&index=3&feature=plcp>

12. Punching the aluminum bars:

<http://www.youtube.com/watch?v=OaYJc1bgfhl&list=UULPI4grwwb-bSrqgJTe70kA&index=1&feature=plcp>

13. Testing the DC motors and gearbox with Mecanum wheels:

<http://www.youtube.com/watch?v=6CHvov9WS2s&list=UULPI4grwwb-bSrqgJTe70kA&index=2&feature=plcp>

XVII. Reference

- [1] Adăscălipei, Florentina and Ioan Doroftei. "Practical Applications for Mobile Robots based on Mecanum Wheels -a Systematic Survey". MECAHITECH'11, vol.3, 2011.
- [2] MIT Electric Vehicle Team, December 2008
- [3] , [4] <http://www.andymark.com/product-p/am-0083.htm>
- [5], [6] <http://banebots.com/pc/P60K-S5/P60K-444-0004>
- [7] <http://www.andymark.com/product-p/am-0077.htm>
- [8] <http://www.trossenrobotics.com/p/sabertooth-dual-12A-motor-driver.aspx>