R2-X_2 2-3-2 System

Table of Contents

Foreword:

Building a 2-3-2 droid can be challenging for a variety of reasons and building an aluminum 2-3-2 droid is even more challenging due to the weight of the droid. So you may be asking why exactly is building a 2-3-2 droid such a difficult task? Well let's take a moment to examine the reasons why it is so difficult. Let me start with the big one, COST! With sufficient funding you can pretty much make anything you want however, most builders are on a budget and most don't have 5 years and 100 thousand dollars to devote to a full development cycle. Droids are not commercial endeavors (at least not yet) so there isn't a research and development department with people that have highly specialized skills working on the designs to make them light, strong, efficient and reliable like with other commercial products such as cars, airplanes, or appliances. We are simple hobbyists with a passion and in my opinion have done some amazing work given the time and budgets our droids have been built with. Cost limits us to mostly off the shelf parts or parts we can make ourselves. For some critical parts of 2-3-2 systems some have turned to a machine shop to build custom parts but again this gets expensive very quickly and is not something most builders will be able to afford just to add the capability of 2-3-2 to their droid. Another reason 2-3-2 is difficult is that Building a 2-3-2 mechanism requires a high degree of mechanical precision and to achieve that level of precision requires a set of fairly sophisticated tools that most builders don't have in their garage. Some of these tools also require some degree of experience to use them effectively. Again most builders are not machinists and do not have extensive experience with machine shop tools such as lathes, & milling machines. Finally, you need to think about the efficient use of space inside your droid. A builder might achieve a nicely working 2-3-2 mechanism only to find that they have used up most of the space inside the droid and don't have room for the rest of the electronics or other mechanical parts they want to add to their droid. The trick to building a successful 2-3-2 mechanism is not to just build a working mechanism but build one that is light, strong, fast, reliable, quiet, efficient and takes up as little space as possible. Those are some rather demanding requirements for the home hobbyist on a small budget. Admittedly my own design falls short on a number of these requirements but my design (like all of them so far) is a compromise of cost, time, materials, and the tools available to me.

Before you decide to take on the task of building a 2-3-2 droid think carefully about your droid's characteristics. These characteristics will largely determine the types of materials you will be able to use when constructing the 2-3-2 mechanism. This in turn will determine the cost of the materials and the tools you will need to construct the mechanism which in turn will determine the skills you will need and the time it will take. If you have a styrene droid or a plastic droid your options for materials to use will be much wider. For a heavy aluminum droid the options are more limited since you will need to build a mechanism that can handle the weight and stress of the 2-3-2 transition. You will also need to think about the power necessary to perform the transition. For the heavier aluminum droid you will need significantly more power with bigger motors and higher voltages or utilize higher gear ratios.

As with any engineering endeavor there is no one right way to construct a 2-3-2 mechanism and this has been proven in our group by the fact that all of the 2-3-2 droids have utilized different techniques to make the transition happen. This article will focus on my own design as the other 2-3-2 droid builders will focus on their designs in separate articles.

Center Foot Lift/Lower Mechanism:

Below is a photo of all the center foot lift mechanism parts laid out in approximately their assembled positions. Since this photo was taken I have replaced some of the parts seen in the photo. The motor was replaced with a more powerful Pittman motor, the timing gears on the tops of the lead screws were replaced with larger plastic timing gears and the custom Acetal parts that hold the timing belt were replaced so that I could achieve a better traction between the center timing pulley and the timing belt. The updated timing gears and belt wrapping pattern are seen in Figure 18.

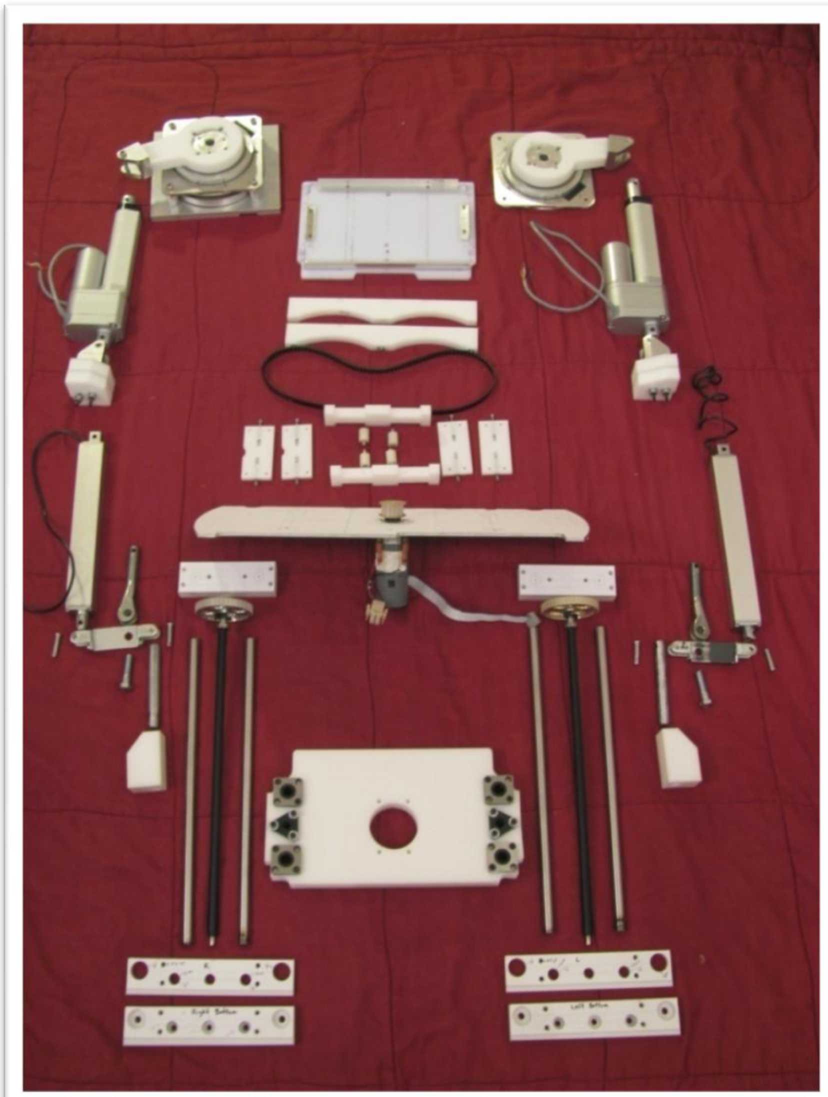


Figure 1: 2-3-2 mechanism parts layout

The concept of the lifting/lowering platform for the center foot is similar to the Mike Lambert design in that there are 4 vertical round steel shafts that guide the platform on its travel up and down. In my design I have replaced the plastic sliders Mike Lambert used with linear bearings to achieve a smoother motion along the rails. The linear bearings can also compensate for the steel rails being slightly out of perfect vertical alignment. I used ¾" Acetal plastic for the lifting platform. It is strong and light and easy to make accurate hole placement for the bearings and the lead nuts. Acetal plastic is also easy to tap, so you are able to mount parts to it using standard screws. To cut the shape and to drill the holes I used a template that I drew using a CAD program. I use the free program downloadable from E-MachineShop.com. I then printed the template on a piece of 8.5 x 11 label paper. I peeled the backing off of the label and placed it directly on the Acetal plastic. This allowed me to accurately cut the Acetal plastic with a band saw (using a fine metal cutting blade) and drill the holes with a drill press (using Forstner bits). I then did some fine touchup work with a Dremel sanding bit. I tapped the holes in the Acetal plastic to mount the linear bearing and the lead nuts to the lift plate with hex head cap screws. The hole in the center of the lift plate allows the center motor to pass through the lift plate when the center foot is retracted. The cavity of the center ankle actually envelops the motor when the foot is retracted thus this design is a great space saver. Figure 2 shows the bottom of the center foot lift platform with the lead screws through the 2 lead nuts, the 4 linear bearings and the center foot motor passing through the lift plate.

Figure 2: Center foot lift platform

The lead screws that do the actual lifting are also mounted at the top and bottom with custom Acetal plastic parts that I machined on a drill press. To allow the lead screws to turn smoothly I use ball bearings mounted in the bottom and top Acetal plastic mounts. I was able to mount the ball bearings by using slightly different size end mills to create a "shelf" to sit the ball bearing lip on. This took a bit of practice and several attempts to develop a technique for precisely aligning the different end mills using just my drill press.

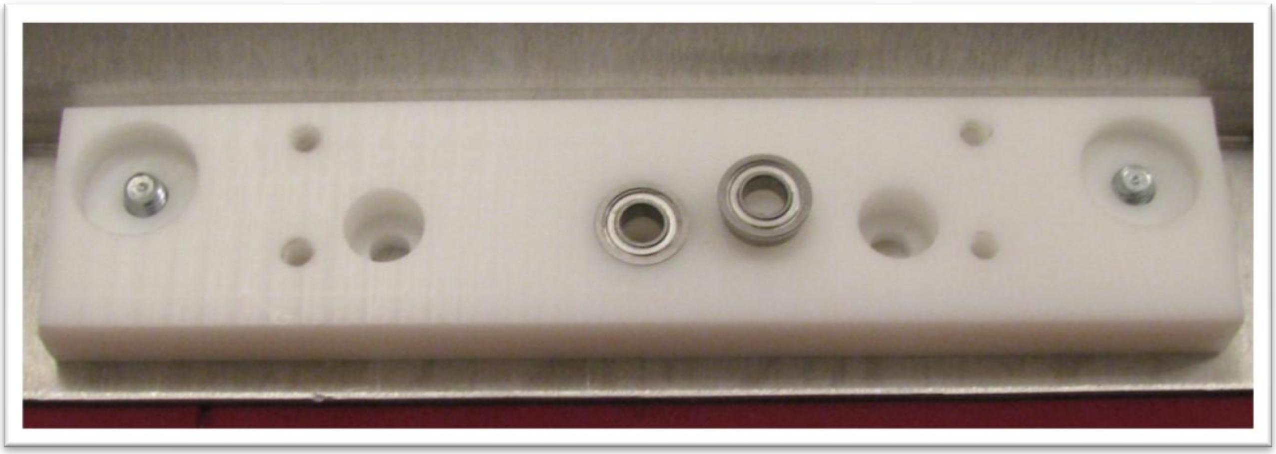


Figure 3: Lead screw ball bearing mounted

Figure 3 shows the ball bearing mounted in the Acetal plastic bottom mount for the steel rods and the lead screws. This bottom Acetal plastic mount is attached to the bottom skirt using the ¾" aluminum posts that came with the JAG frame (See Figure 4).

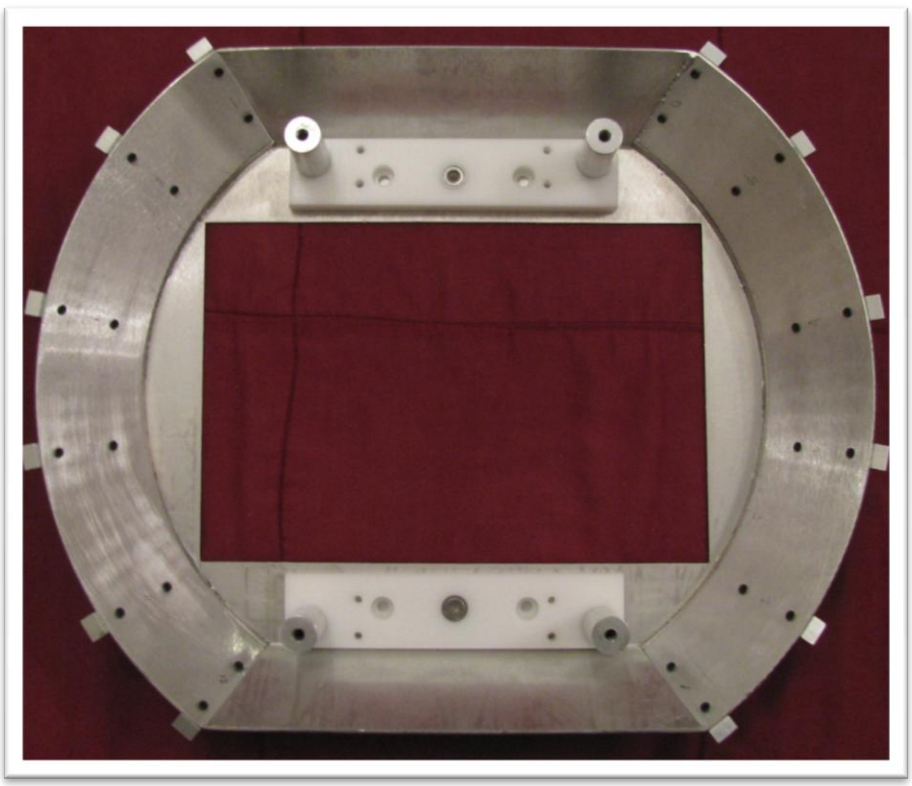


Figure 4: Both ball bearings mounted in Acetal plastic mounts

Once the ball bearings were mounted to the skirt a cap piece of Acetal plastic was screwed on top to keep the ball bearings in place.

Figure 5: Ball bearing cap - not mounted yet

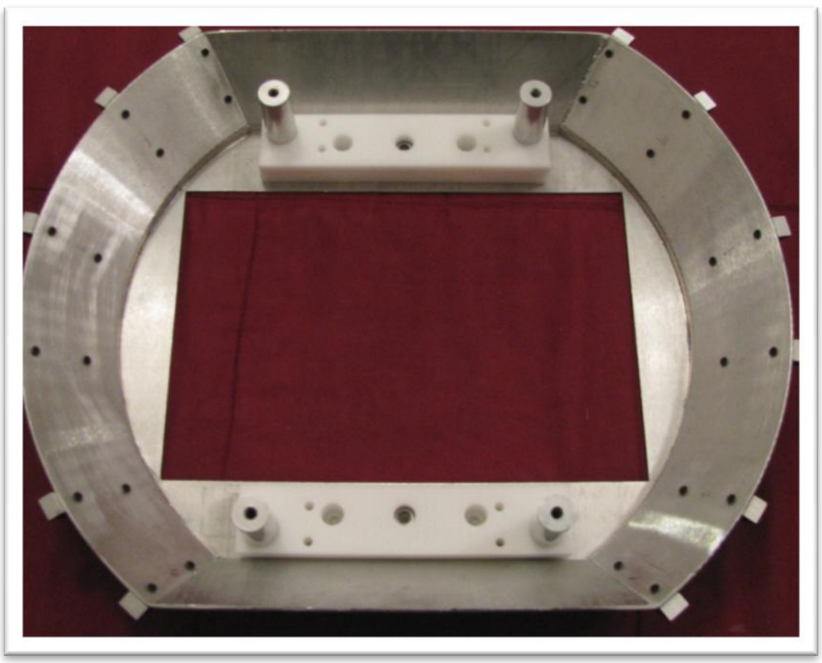


Figure 6: Ball bearing caps mounted

Figure 6 shows the bottom Acetal plastic mounting pieces with the cap pieces that hold the ball bearing in place mounted

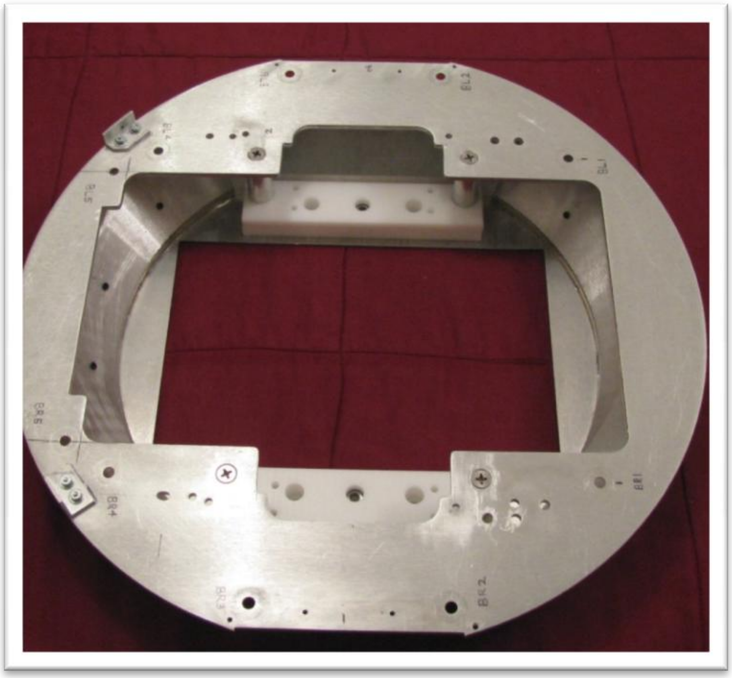


Figure 7: Skirt with bottom frame ring mounted

Figure 7 shows the bottom frame ring mounted. Notice I had to widen the opening front and back from where the steel rods are placed. This is for the lifting/lowering plate to pass through.

At the top of the steel shafts another set of Acetal plastic mounting blocks uses a second set of ball bearings for the lead screws. These are mounted directly to the cross frame plate of the JAG 4 frame. Since the ball bearing sit against the aluminum plate no cap piece is needed to keep them in place.

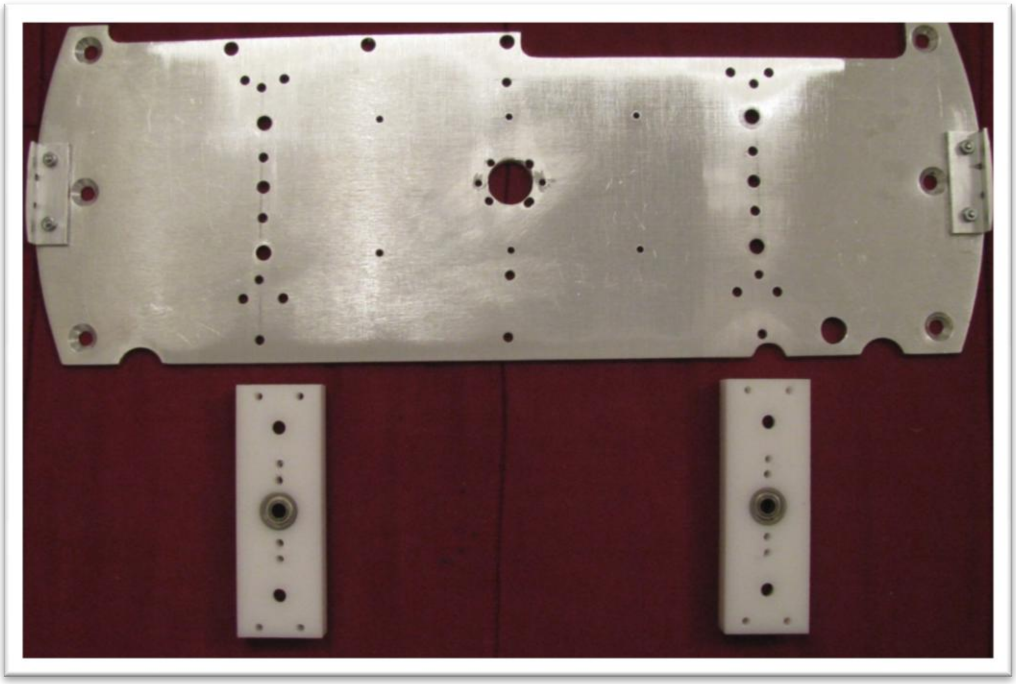


Figure 8: Top ball bearing mounts

Figure 8 shows the Acetal mounting blocks upside down, they get flipped over so that the ball bearings are mounted toward the aluminum plate.

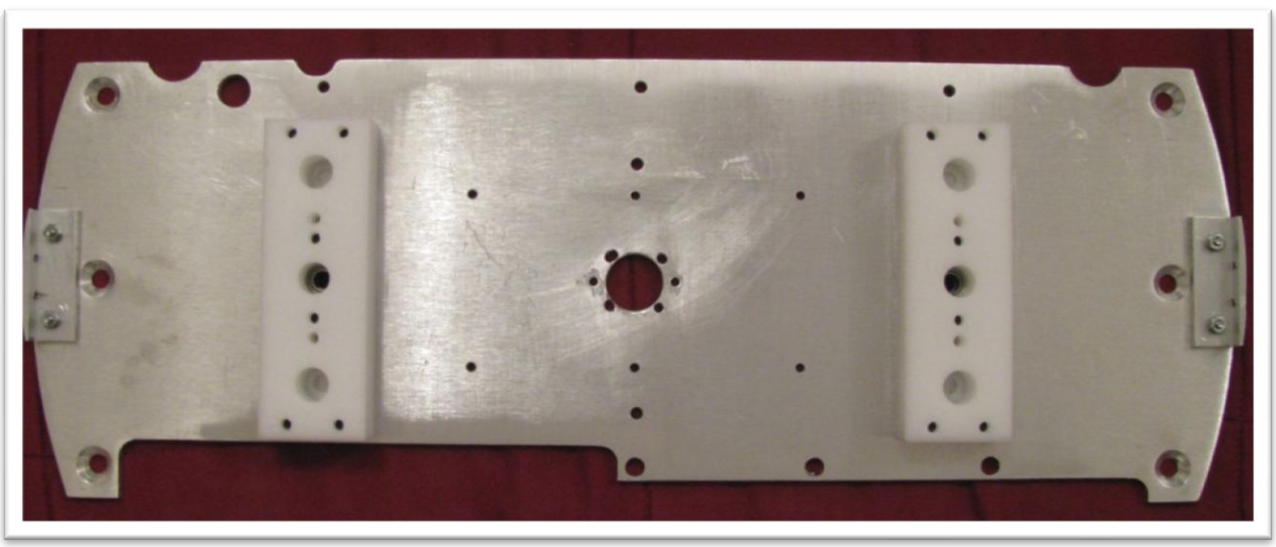


Figure 9: Top ball bearing mounts fastened in place

Figure 9 shows the top ball bearing mounts mounted to the aluminum cross frame plate. These mounting blocked are attached to the aluminum cross frame plate with cap head screws by drilling and tapping the Acetal plastic mounts.

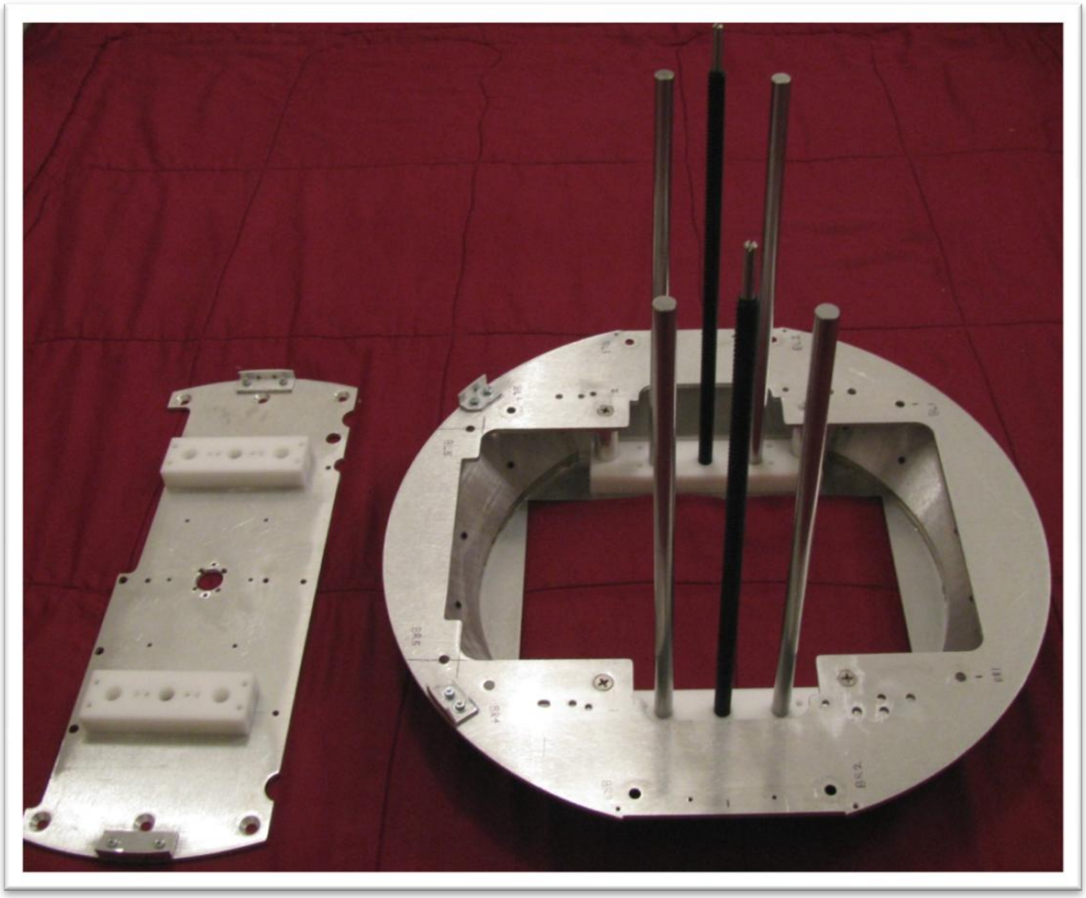


Figure 10: Steel rods and lead screws inserted

Figure 10 shows the steel bearing rods and the lead screws inserted into the lower Acetal plastic mounts and the aluminum cross frame plate with the top Acetal plastic mounts to the left.

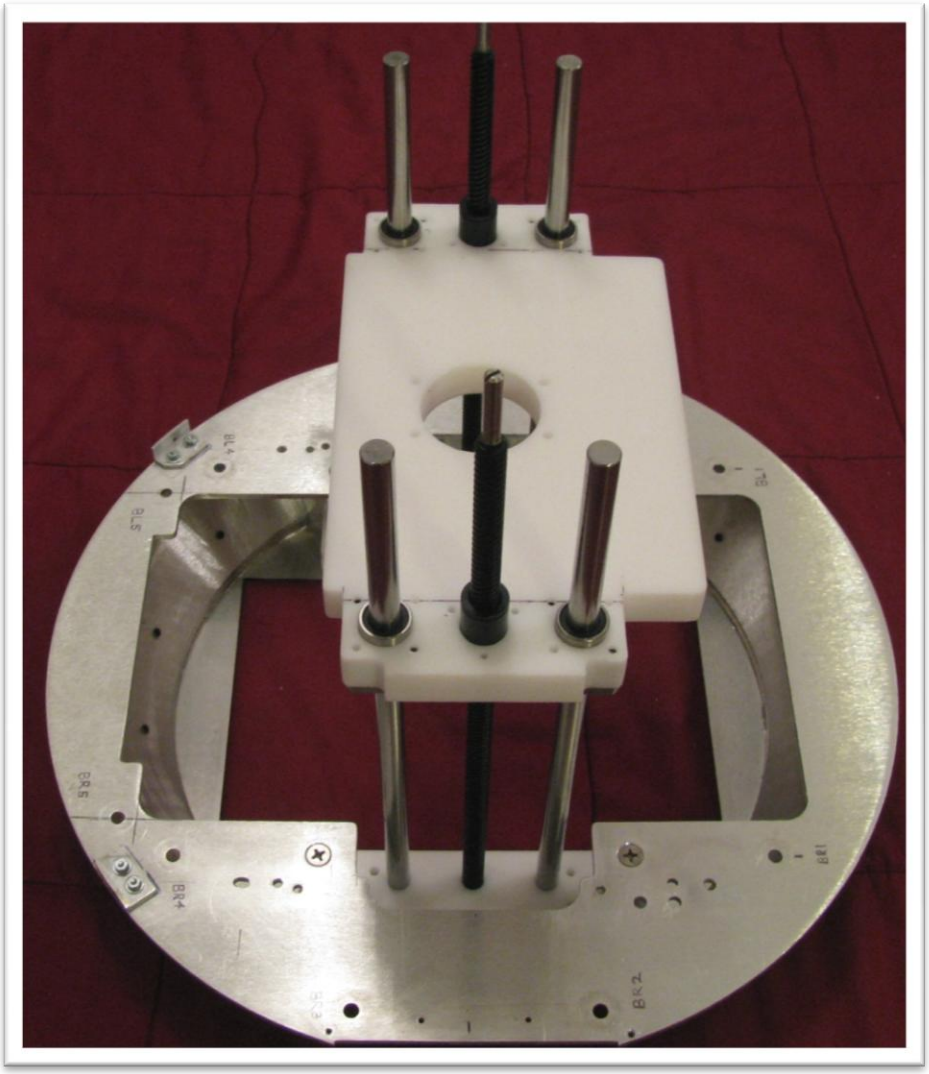


Figure 11: Center foot lifting plate mounted on rods

Figure 11 shows the center foot lifting/lowering plate mounted on the steel guide rods and the lead screws.

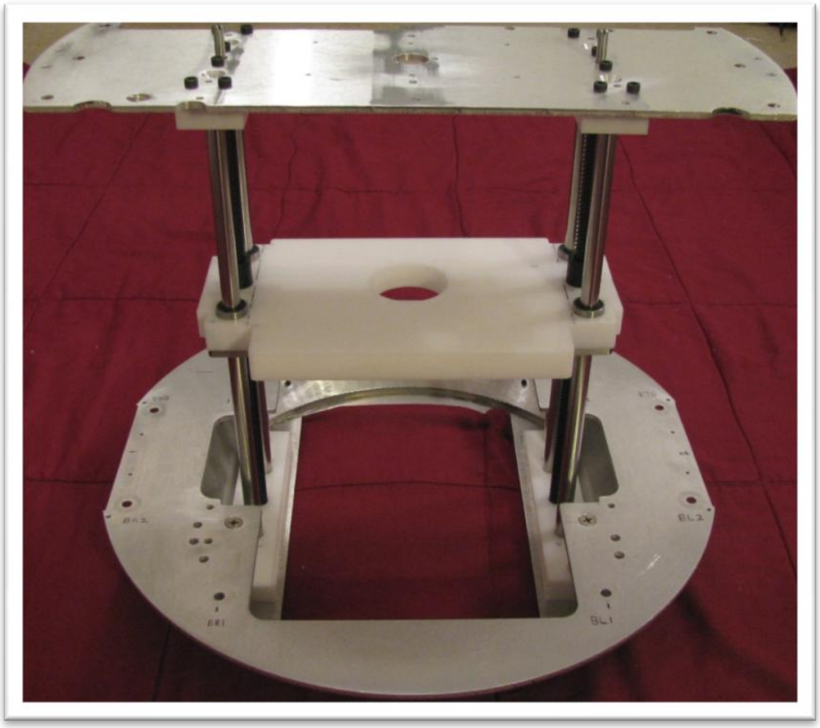


Figure 12: Aluminum cross frame plate mounted

Figure 12 shows the aluminum cross frame plate mounted on top of the steel guide rods and the lead screws. Note the lead screws protrude through the aluminum plate for mounting the timing gears that turn the lead screws.

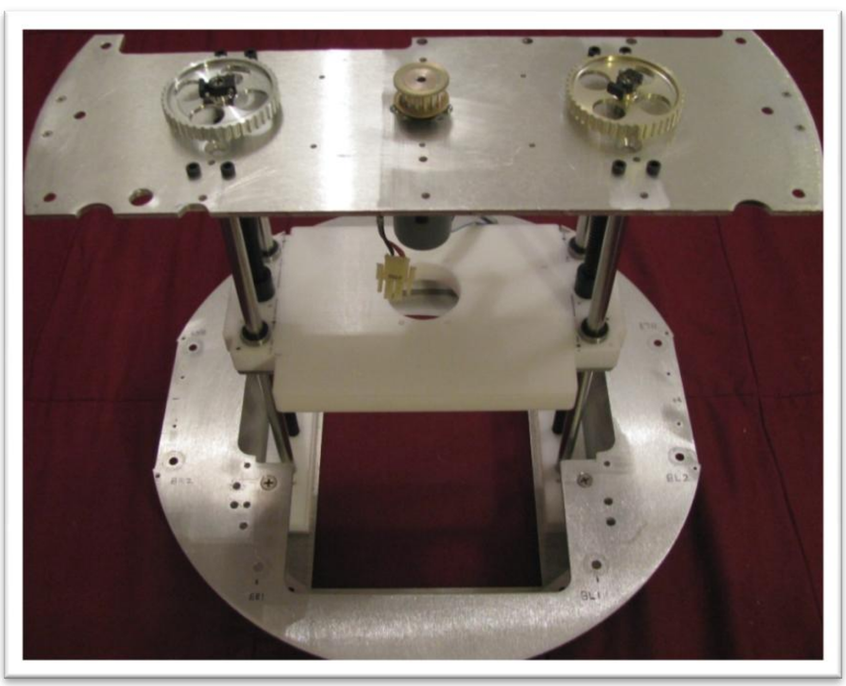


Figure 13: Motor and lead screw timing gears mounted

Figure 13 shows the timing gears mounted to the lead screws and the motor mounted to the aluminum cross frame plate. Note: this photo was of an earlier assembly and since then the timing gears were replaced with larger timing gears and the motor was replaced with a more powerful Pittman motor. Also note the hole in the center of the lifting/lowering plate. The hole allows the motor to pass through the lifting plate as it moves toward the top. The center motor turns the two outer timing gears via a timing belt. However, the timing belt needs a set of idler pulleys to accomplish two things. First the timing belt needs to have enough contact with the timing pulley mounted on the center motor and second since It would be nearly impossible to find an off the shelf timing belt to exactly fit the configuration of these timing gears so the timing belt needs to have an adjustment mechanism to set the tension of the belt.

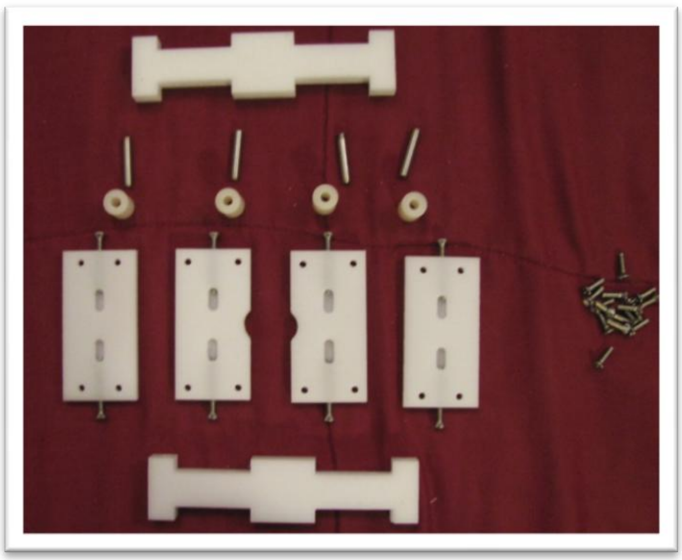


Figure 14: Original timing belt idler mounts

Figure 14 shows the Acetal plastic parts I machined with a drill press and end mills to mount the idlers for the belt to make sure there is enough contact with the center timing pulley. It also shows the adjustment screws used to set the belt tension.

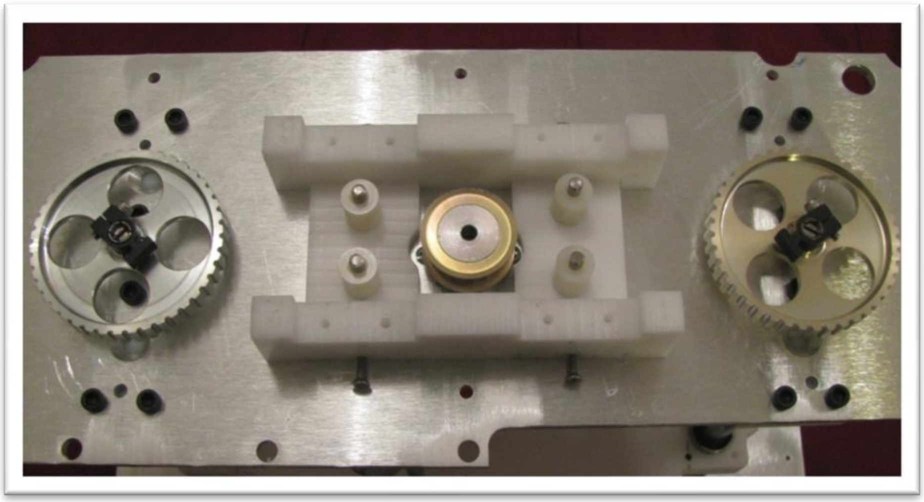


Figure 15: Timing belt idler mounts partially assembled

Figure 15 shows the timing belt idler mounts partially assembled. The idlers are mounted on steel rods and are simple nylon rollers. These roll amazingly well for not having any bearings. The tension set screws visible at the bottom push up against the bass of the steel rods on which the rollers are mounted. There are a second set of tension set screws in the top mounting parts (not shown here) that adjust the top position of the steel rods. These screws allow the steel rods to be moved in or out to adjust the belt tension.

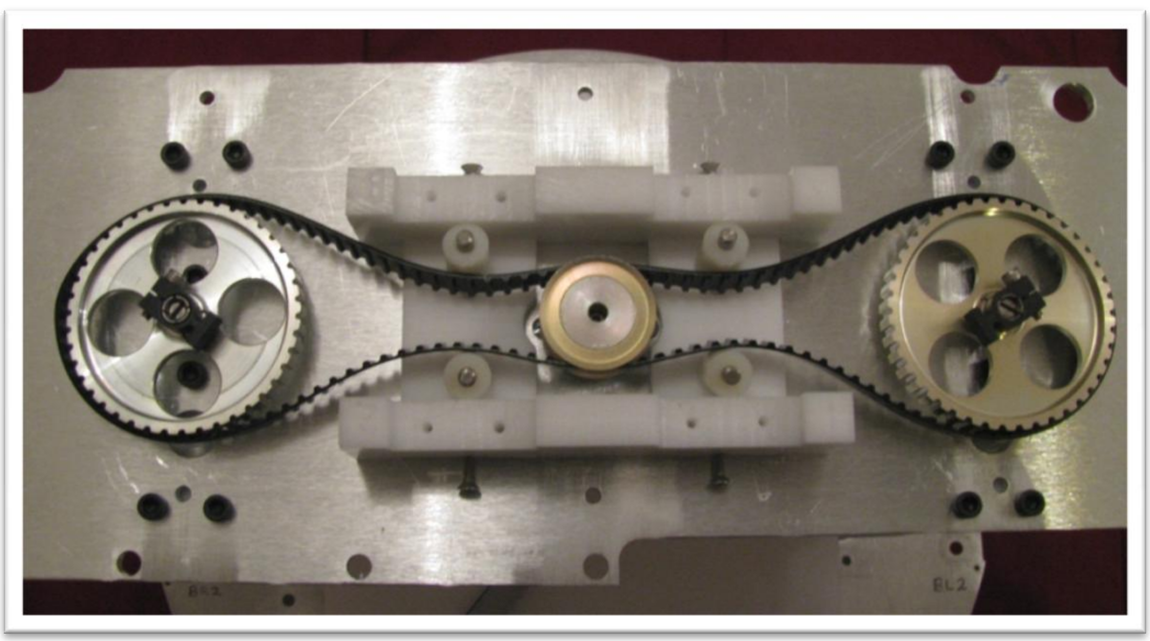


Figure 16: Timing belt mounted loose

Figure 16 shows the timing belt in place loosely around the timing gears.

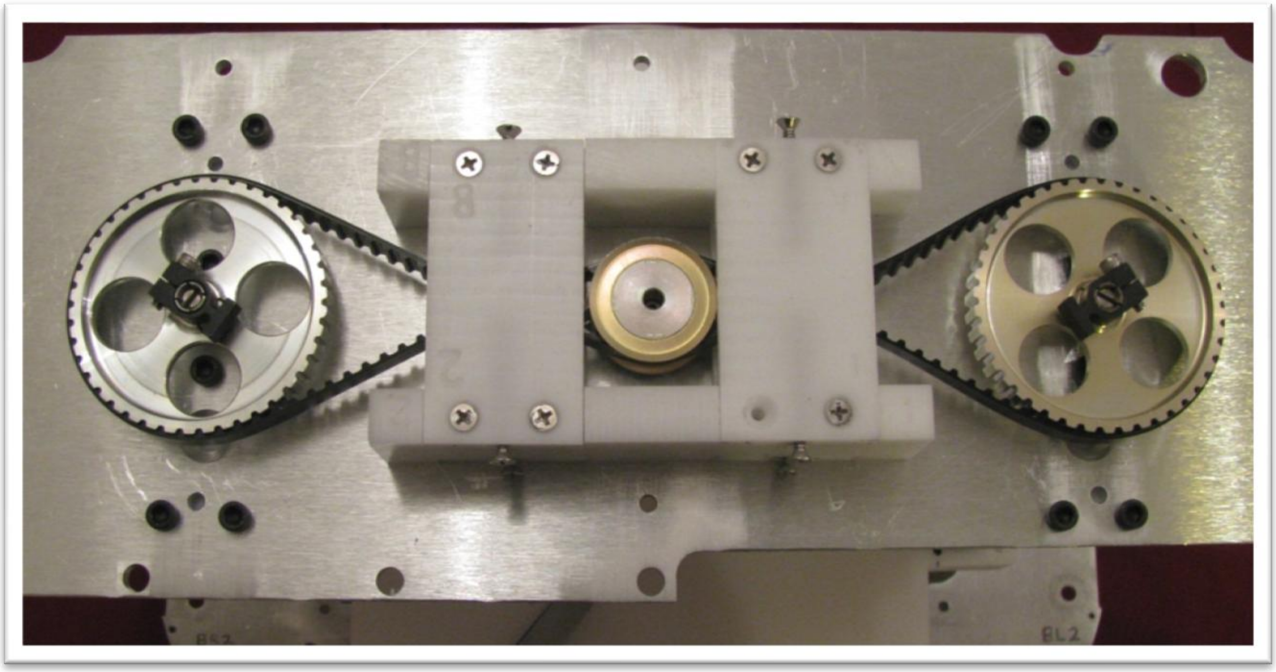


Figure 17: Timing belt mounted with tension

Figure 17 shows the top idler mounting parts in place and the timing belt with tension applied. Note the tension screws on the top idler mounting plates and the set position of the screws to apply the tension.

So it looks good but after testing this design there was a problem with the center timing gear not having enough contact length with the timing belt. I used this series of photos in this document to give you the basic idea of how the lifting mechanism works and the new design works very much the same way except I was able to loop the timing belt around the center timing gear giving it more contact with the timing belt. Unfortunately I do not have good photos of the new design but you can see it fairly well in the following images. With the new design the timing gears have been replaced with larger black plastic ones to give the system more torque as well as the change to the belt pattern.

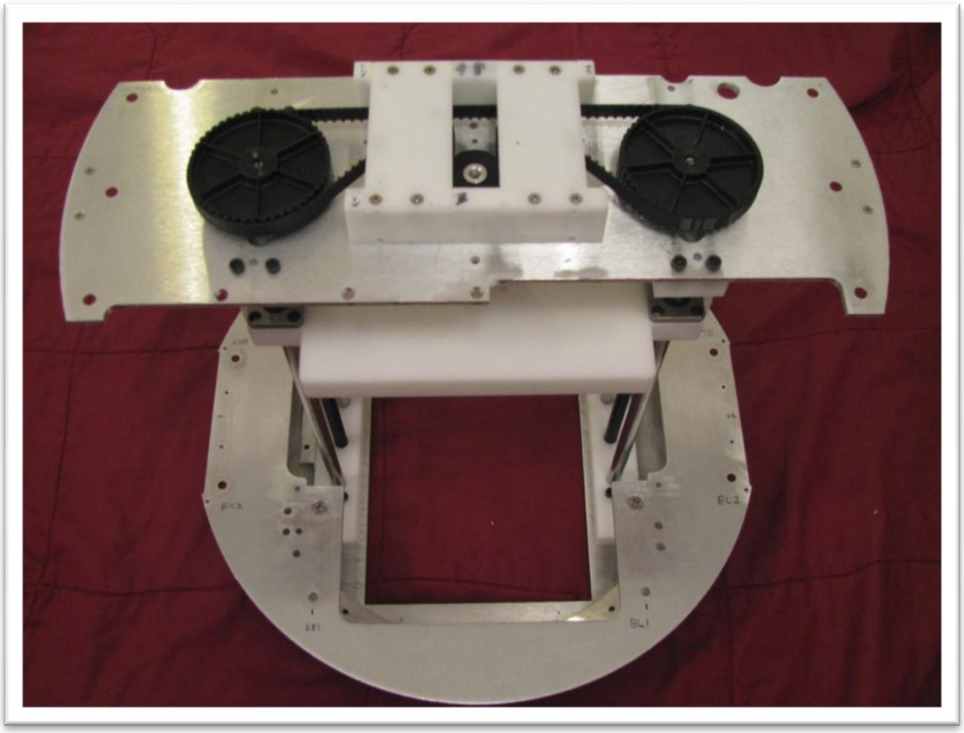


Figure 18: New design top timing gears

There is video of this mechanism being tested on the Astromech site in my galleries and on YouTube.com.

http://www.youtube.com/user/BManncsci?feature=mhum#p/u/11/LhqDIaZOa2I

http://www.youtube.com/user/BManncsci?feature=mhum#p/u/12/WDO4u-5FJEw

Shoulder Rotation Mechanism:

When undergoing a 2-3-2 transition the body of the droid and the legs rotate to form a stable tripod stance. In a 2-3 transition the legs of the droid rotate backwards and the rotation is centered around the shoulder joint. The rotation is accomplished by two linear actuators that push on a lever arm that is attached to the shoulder rotation bearing. During the reverse 3-2 transition the same linear actuators pull on the shoulder lever arms to rotate the legs back into a vertical stance.

The shoulder rotation lever was designed to replace the satellite motor assembly for rotating the shoulders. The satellite motors were reported to have some problems when the weight of the droid was above 150 pounds. The satellite motors use very fine toothed steel gears with a high gear ratio to rotate the shoulders and under heavy stress tend to shear off the gears. However, since those early reports it has been shown by Bob Ross that the issue was due less to the high stress of performing the

2-3-2 transitions but one of driving the droid while in 3 legged mode. The jarring that a droid endures while moving on rough surfaces is the cause of the gear stripping. The solution Bob implemented was to add a locking mechanism to the leg so the gear teeth are not taking the impacts while driving the droid.

Figure 19: Fine toothed satellite motor gears

However, even with this the gear problem resolved I felt that the satellite motors took up quite a bit of space in the center of the droid that could be put to better use so I designed the shoulder lever mechanism. This design underwent a number of prototypes to come up with the final design. I drew the shoulder lever in a 3D CAD program and I had originally intended for it to be made of aluminum. However, the cost of these made from aluminum was just too high for my budget and the design languished for almost 2 years. The idea to make the shoulder lever out of Acetal plastic came to me after see some of the work Daren Murrer was doing with his 300 mm dome. He uses an Acetal plastic ring inside the dome. But honestly I had real doubts that it would be strong enough to handle the 2-3-2 transition of my heavy droid. I started working with Acetal plastic for some other parts of my droid and I got some experience working with it and at some point decided to give the shoulder lever arm a try in plastic. I redesigned the arm to make it a bit larger around the ring to compensate for the use of a weaker material. To make the levers I used my template technique of printing it on an 8.5 x 11 label paper and sticking it directly to the Acetal plastic. I was then able to use a band saw and my drill press with end mills and Forstner bits to make the final part. I finished it off with some sanding of the rough edges with a Dremel sanding wheel.

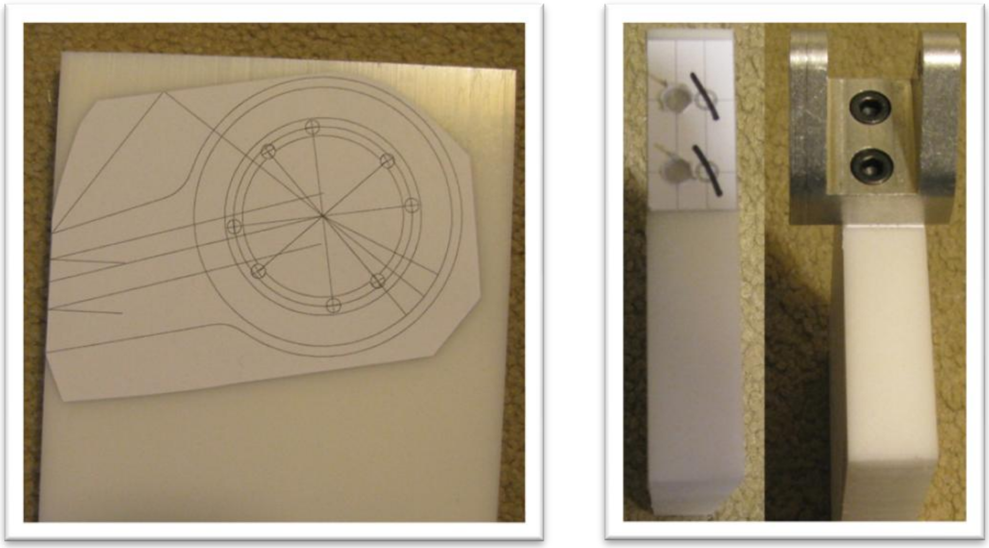


Figure 20: Shoulder lever template and tapped end with clevis mounted

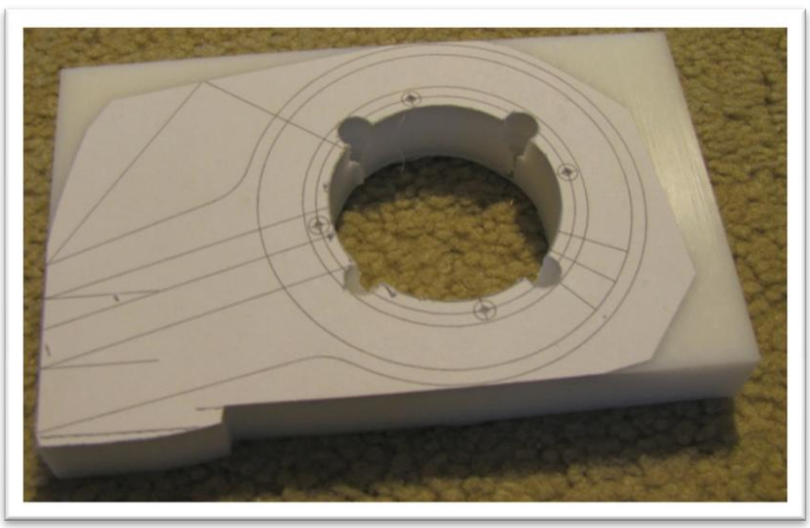
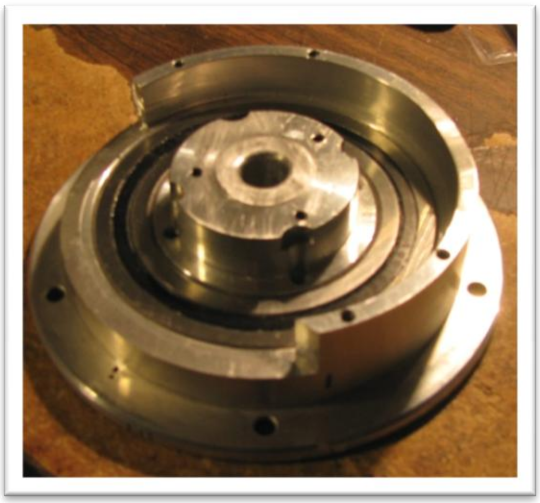


Figure 21: Shoulder lever partially cut and milled

Creating a shoulder lever was only part of the solution for the shoulder rotation mechanism. I always felt that the satellite motors offered a nice solution to the shoulder rotation but I wanted to do away with the actual motor and gears. The bearing in the satellite motors is very nice and provides that prefect rotation mechanism as well as a way to pass wires through a rotating shoulder to power my foot drive. The issue I had was that the satellite motor assembly had a large lip around most of the bearing that was interfering with the rotation of the shoulder lever. Removing this lip was going to be difficult. I could have a machine shop put it in a milling machine to have the lip removed but again that was too expensive for my budget. Using a Dremel would have taken weeks and probably destroyed the Dremel in the process. At one point I tried using a band saw and quickly determined it was way too dangerous. Then I saw a new tool called the Fein Multimaster that could cut along a flat surface. Again this tool was too costly for my budget but then I saw that Dremel had copied this tool with their own version for about $100 so I decided to give it a try. I bought some extra metal cutting bits and started having at it. It was slow and painful but it was doing the job. It took about 3 hours to cut the lip off of one housing. The intense vibration from the Dremel Multimax left my hand numb for about 2 days but it had one the job.



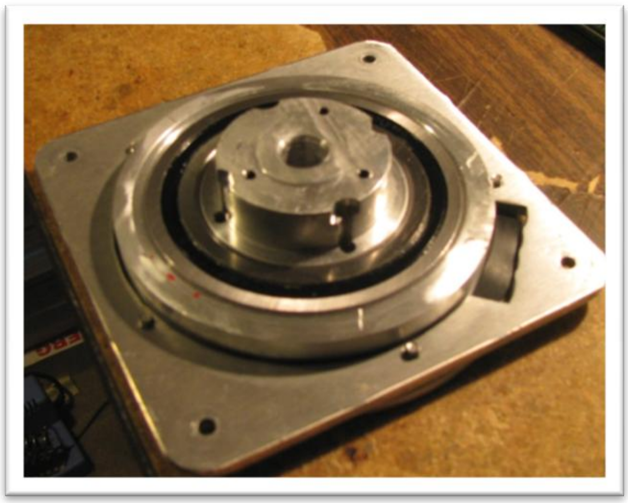


Figure 22 a & b: Satellite bearing lip, and bearing lip removed

Figure 23: Satellite bearing lip cut off pieces – I can't feel my hand!

I finished the other satellite bearing housing the same way and then I was able to then mount the levers to the satellite bearings without any rotation of the lever being blocked.

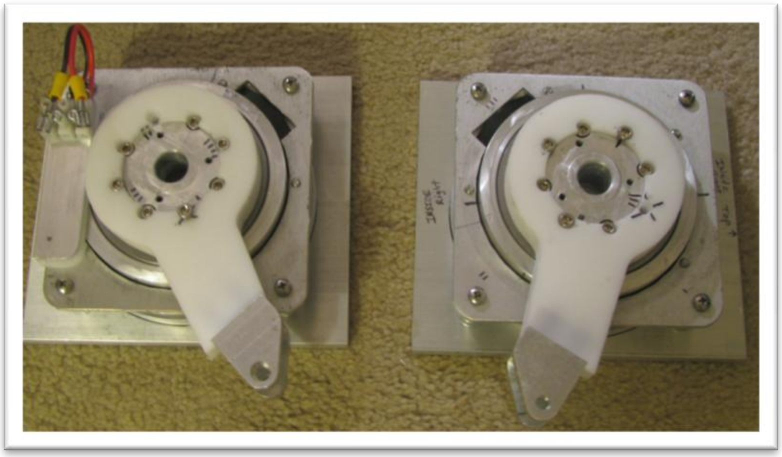


Figure 24: Completed shoulder rotation lever assembly

Figure 25: Completed shoulder rotation assembly top view

To make the shoulders rotate I purchased 2 SPAL - 4" linear actuators. These actuators each came with 2 clevises to mount the base and the end of the extender piston. The clevis allows the base and the end to pivot while the actuator is extending or retracting. I mounted one clevis to the end of the shoulder lever arm as seen in Figure 24. In order to use the internal limit switch of the linear actuator I needed to mount the base of the actuator above the bottom frame plate such that the fully retracted linear actuator connects to the lever arm while the shoulder is rotate to the 2 legged position. This way when executing the 3-> 2 transition I can drive the linear actuator until it is fully retracted and not have an extra limit switch for that end of the motion. This simplifies my electronic control of the shoulder rotation as I only need to control the amount of extension with an external limit switch. In order to get the linear actuator at the right height I made 2 mounting blocks out of Acetal plastic and mounted them to the bottom frame ring. Then the base clevis of the linear actuator is mounted to the Acetal plastic mount.

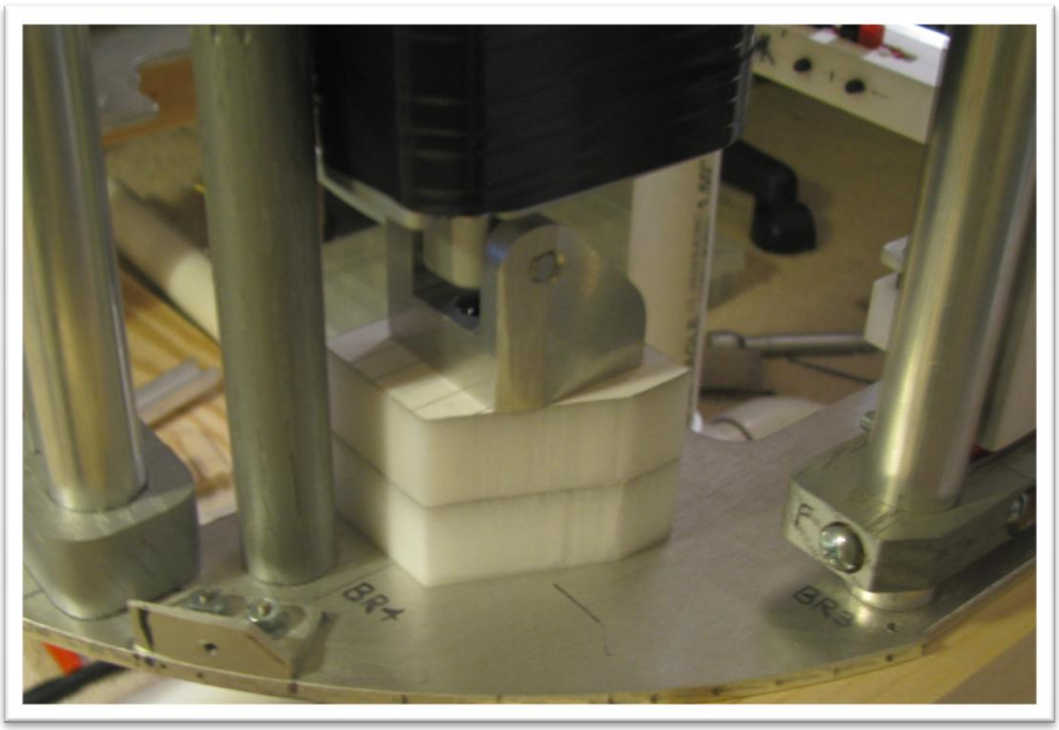


Figure 26: Acetal plastic linear actuator mounting base

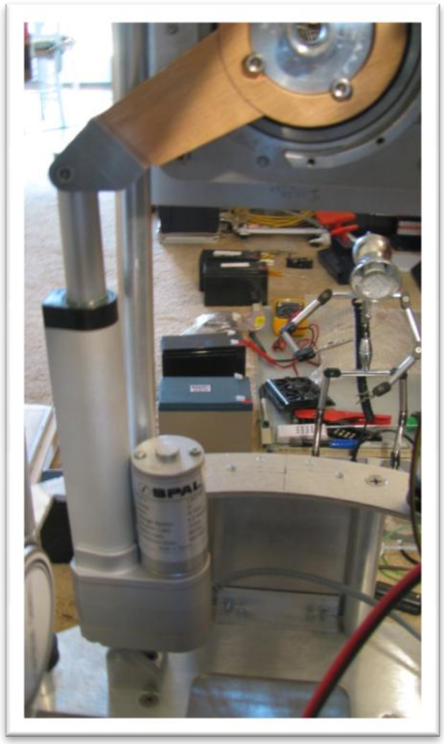


Figure 27: Linear actuator mounted to shoulder lever

Unfortunately, Figure 27 is the best photo I have of how the linear actuator works with the shoulder lever. It is an old photo where I was using the second wooden prototype I made and there is no mounting block for the base clevis to set the correct height. It is also mounted in the front of the droid and I have since changed the position of the linear actuators to the back of the droid so that I have room to mount other devices behind the front skin panels such as the mechanical grasper arm and the computer probe arm. However, it does nicely show how the linear actuator works with the shoulder rotation lever.

Ankle Joint Rotation:

In addition to the lifting and lowering of the center foot and the shoulder rotation the outer ankle joints of the droid also must rotate during a 2-3-2 transition so that the foot drive units stay level with the surface. The mechanical joints between the ankle and the foot shells are free rotating joints. However, if they were allowed to rotate freely while the droid is standing upright the droid would fall either forward or backward. So a mechanism is needed to keep the droid standing upright while standing in two legged mode. A few years back Darren Murrer created an ankle pin mechanism for his droid that was actually contained within the ankle joint itself. Basically he drilled a hole in the bolt that passes through the ankle joint and installed a fixed rod in it such that the rod would control the ankle joint motion. This was an ingenious mechanism but very difficult to construct and I was worried about the strength of the joint since it required a fairly large hole to be drilled right through the main ankle bolt. I had had another idea that I thought would work pretty well but it would not be completely invisible like Daren's solution. My solution to keep the droid from falling forward would have a pin extend from the bottom of the ankle and that would press against the top of the steel foot shell to keep the droid standing upright. To keep the droid from falling backwards would be easier by simply using a fixed block of material wedged between the top of the steel foot and the underside of the aluminum ankle. Again both of these would be visible if you got down and looked at the ankle level but it was a much more stable solution and much easier to implement than Daren's solution. To implement my solution I needed a mechanism that would make the pin extend and retract but was contained within the leg and ankle. The pin was only required to have a very short stroke of movement (about 3/4") so I had several ideas of how to implement the mechanism. One potential idea had a worm gear like solution similar to a cork remover for removing corks from a bottle of wine. The mechanism lets you clamp on to the bottle at the top and then drive the cork screw into the cork with a downward motion of the handle then remove the cork from the bottle by moving the handle in the opposite direction. The cork screw moves in a vertical up and down motion while the handle move in an arc. This would have worked but I could not find a mechanism made of steel which would be strong enough for the application in a droid. Another potential solution came from an old style hand water pump as seen below in Figure 28. This mechanism allows the piston that operates the pumping action to move vertically up and down while the handle moves in an arc. It achieves this motion using an extra pivoting linkage that is attached to the pump handle. It also uses a lever to amplify the force placed on the handle of the pump making it much easier to pump water. This would be very helpful in my application since the linear actuator that I am using which fits inside the aluminum leg does not have a lot of force when acting alone. The lever would help amplify the force of my linear actuator when restoring the droid to a two legged stance. All I had to do was turn this mechanism over and I had exactly what I needed. Actually it would work the way it is oriented in the picture but there is nowhere to mount the end of the linkage rod inside the ankle unless you move the linkage to the top of the lever arm. Then I could use a hanger to mount the end of the linkage.

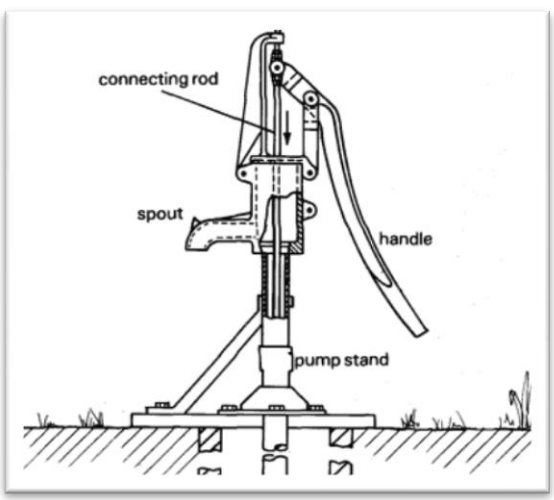


Figure 28: Old hand water pump

Figure 29 shows the prototype I created to test my mechanism. I cut an outline of the leg (to scale) out of plywood and mounted the linear actuator and lever arm. The steel pin guide block was made from wood and super glued to the plywood. I made the lever arm from a piece of strip steel that I drilled and tapped both ends and used machine screws to fasten the two end clevises with. The pivoting linkage is a clevis with a threaded rod body that I purchased from an automotive supply magazine. To my surprise it worked nearly perfectly the first time.

Figure 29: Prototype ankle pin mechanism

So with a working prototype I was inspired to build the actual mechanism that would be installed in the aluminum legs. Little did I realize how much work it was going to take to go from a simple working prototype to an installable mechanism.

The installable mechanism was going to need a way to mount the linear actuator, the pivoting linkage, and the ankle pin guide block inside the aluminum leg. The ankle would also need to have a hole drilled in the bottom to allow the pin to pass through. Frankly the thought of screwing up my aluminum ankles and legs by drilling holes in them had me pretty worried and held me back for a long time while I pondered if I should really go through with it. I also worried that a second hole in the bottom of the ankle (one was already there for foot drive wires) would weaken the ankle to the point of making it unusable in a heavy droid. So far those thoughts have been proven incorrect.

At some point the courage to drill into the aluminum legs and ankles came to me and I fired up the drill press! However, before I did any actual drilling I planned it well. I had drawn the whole thing in a CAD program exactly to scale and I made sure it all fit. Figure 30 shows the CAD drawing with the ankle pin mechanism mounted inside the aluminum leg. To make sure the holes were properly aligned on the aluminum leg with the mounting blocks I had the CAD drawing printed full scale on an Architectural drawing plotter. Before drilling the holes in the aluminum leg I cut out 2 of the printed drawings and taped them to the aluminum leg on both sides. This allowed me to accurately align and drill the holes in the aluminum legs for the Acetal plastic mounting blocks.

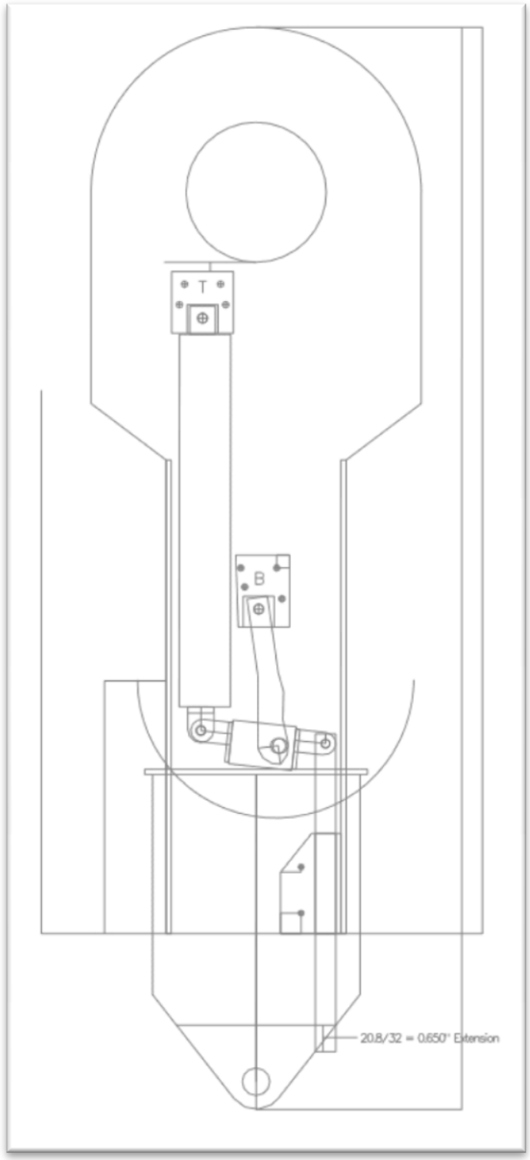


Figure 30: Ankle pin mechanism CAD drawing

Figure 30 shows the mounting brackets I used to mount the linear actuator, the vertical linkage for the lever arm and the ankle pin guide block. To create the mounting bracket for the top of the linear actuator I First milled a block of Acetal plastic. Basically this is a cube with four screw holes on each side for mounting it inside the leg. It also has the bottom of the cube milled out to allow the top mounting bracket of the linear actuator to be mounted with a steel pin through it. The steel pin allows the body of the linear actuator to swing as the piston extends and retracts. The linear actuator only swings slightly on the steel pin but it must be allowed to move or the mechanism will bind. What is not shown in Figure 30 is that the bottom of the mounting block is also beveled upward to allow the linear actuator to swing. The mounting block for the lever's fulcrum mounts the same way. It also has a steel pin that passes through the clevis's shaft and allows the linkage to swing as the linear actuator piston extends and retracts. The ankle pin guide block has a number of tapped holes to mount it in place. An additional hole needed to be drilled through the ankle pin guide block to allow the aluminum bolts that secure the ankle to the leg to pass through the guide block.

Figure 31: Ankle pin linear actuator

Just like the shoulder linear actuators I wanted to take advantage of the internal limit switch inside the linear actuator for the ankle pin mechanism. This meant mounting the top end of the linear actuator precisely such that when completely retracted the ankle pin would be in the fully extended position and be pressing against the top of the foot shell. By doing this I only have to control the extended position of the linear actuator with a second external limit switch. To do that I mounted a limit switch in the bottom of the ankle pin guide block. An Acetal plastic cantilever piece that slides over the ankle pin and is fixed in place with a set screw will trigger the limit switch as the pin is retracted (seen in Figure 31).

Figure 32 shows the bottom of the ankle with two holes drilled in the ankle. The center hole is for wires to pass through to power the foot drive. The hole to the left is the hole the ankle pin passes through. The 4 smaller holes are to mount an Acetal plastic guide inside the ankle. This helps stabilize the ankle pin through its travel. While it is difficult to see in the photo there is also a block of Acetal plastic mounted to the bottom right side of the ankle that keeps the droid from falling backward while standing in two legged mode. Since it is colored black it is virtually invisible when looking at the fully assembled droid.

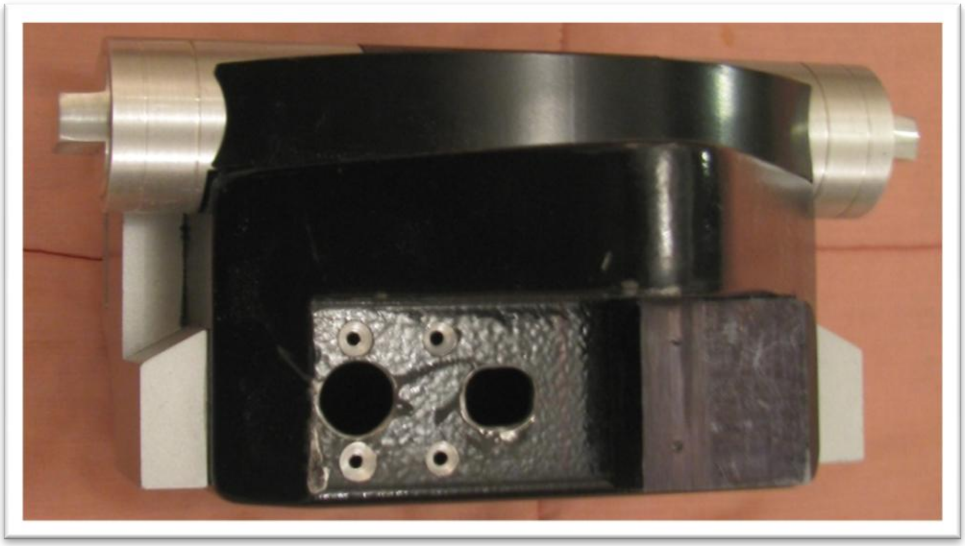


Figure 32: Bottom of aluminum ankle

Figure 33 & 34 shows close ups of the lower and upper components of the ankle pin mechanism.

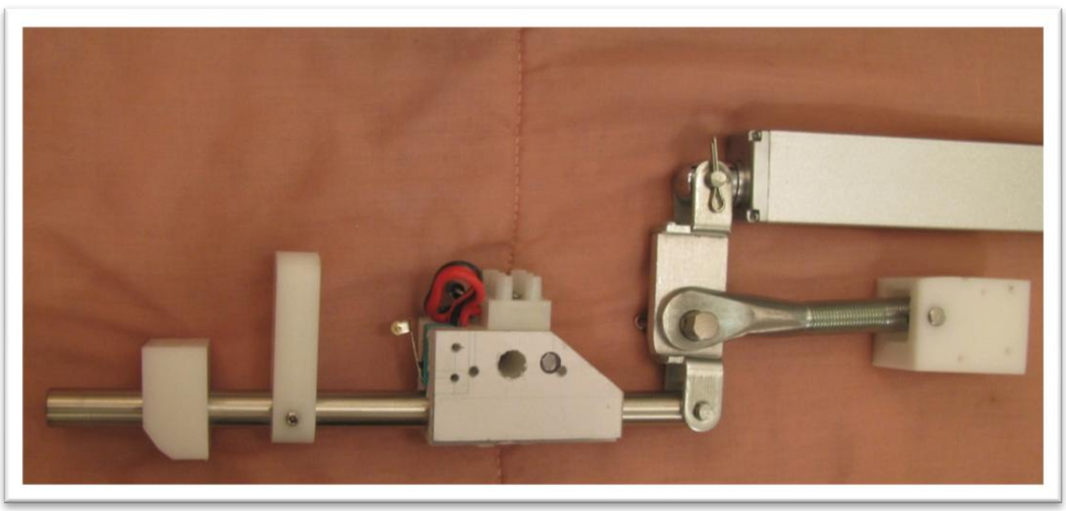


Figure 33: Ankle pin mechanism lower components

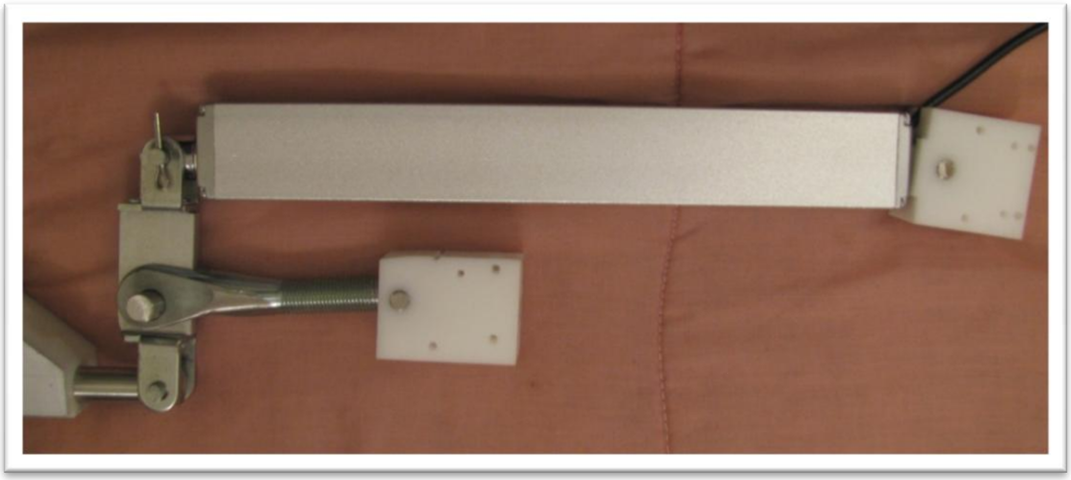


Figure 34: Ankle pin mechanism upper components

The lever is made from plate aluminum that was drilled and tapped on its ends. The two clevises that came with the linear actuator are mounted to the ends of the aluminum plate with machine screws and Loctite is used to keep the screws in place. The hole in the lever is placed off center to amplify the force of the linear actuator.

Electronic Control:

To make the 2-3-2 transitions happen all of the previously described mechanisms must be coordinated to turn on and off with proper timing. To do this the whole system uses a micro controller. The R2-X2 eXperimental 2-3-2 electronics design uses a PicAxe 18x project board and 2 Dimension Engineering SyRen motor speed controllers to control all of the start and stop motor motions for the 2-3-2 action. It also uses 3 opto-electrical switches to sense the position of the center foot. The motion of the shoulder rotation and leg pin actuator motors are triggered by the position and direction of movement of the center foot using opto-electrical switches. The start of the 2 cycles (2->3 & 3->2) are triggered by two different buttons of an RF relay remote. Figure 35 shows that whole electronics wiring diagram for the 2-3-2 system and Figure 36 is an enlarged section of the connections to the Picaxe 18x project board. While I am using a Picaxe 18X project board I'm certain this could also be done using one of the other micro controller boards such as the Arduino if you like one of them better than the Picaxe. The SyRen motor speed controller will accept input from any of these micro controllers as long as the baud rate is correct. The communications protocol for talking to the SyRen speed controller is 8N1 – that is 8 data bits, No parity, 1 stop bit.

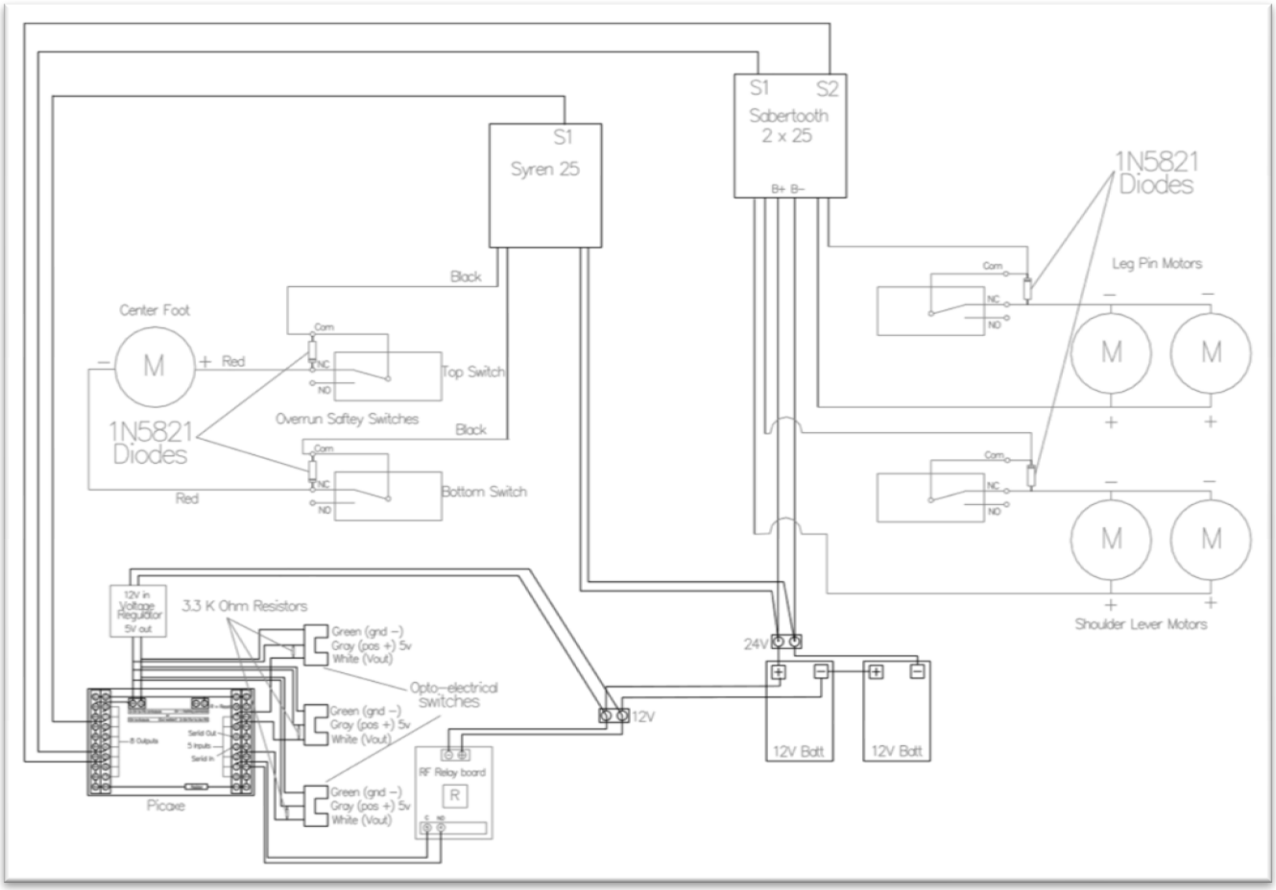


Figure 35: 2-3-2 electronics wiring diagram

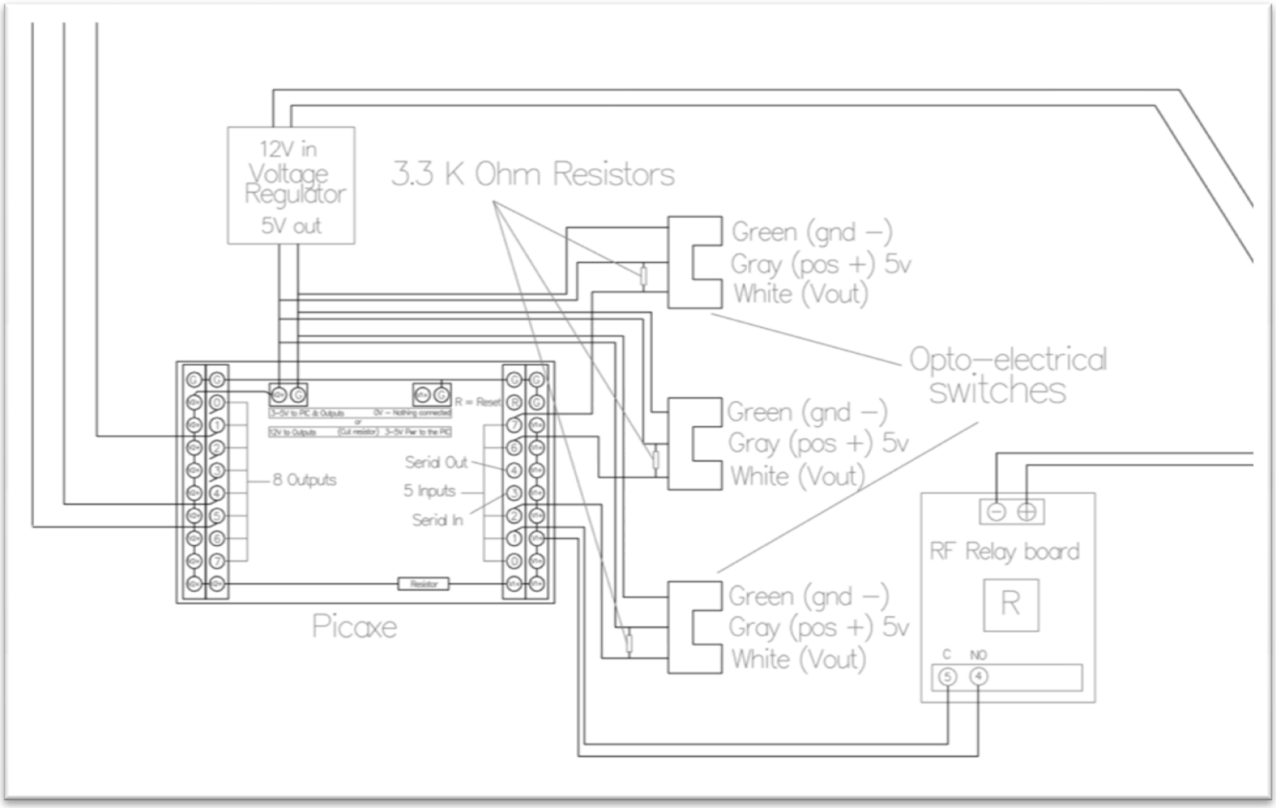


Figure 36: Enlarged section 2-3-2 Picaxe 18x micro controller

Dimension Engineering has built into their motor speed controllers a way to control them from a micro controller. They call it the simplified serial mode and basically the micro controller hooks up to the speed controller with a single TX wire and sends 1 byte of data that contains the speed and direction you want to set. You can continually send new bytes of data thus continually adjusting the speed of the motor. By using these motor speed controllers I can easily adjust the power I want to apply to each motor thus affecting its speed and therefore its timing. If I need to do timing adjustments to fine tune the motions I can easily do it by simply changing a values in the software and reloading it to the micro controller. In fact I can even create software routines to have the motors accelerate or decelerate as part of the 2-3-2 motion.

Before deploying the whole electronics setup in my droid I decided it would be a good idea to set up a test apparatus and use it to get my software working and fine tuned. This turned out to be one of the best things I could have done since it took about 4 days of work to get my software working correctly. I am not a micro processor software developer and it took a bit of research to understand the way micro processors need to be programmed before things fell into place. Figure 37 shows the test apparatus I built to do the preliminary testing of my 2-3-2 software. At the top right of the photo you will see 3 motors. Basically I used these 3 motors to simulate the center foot lift motor and the 2 sets of linear actuators (shoulder rotation and ankle pin linear actuators). I could do this since each set of linear actuators is turned on and off at the same time and for testing purposes could be considered a single motor. At the bottom right of the photo I mounted the 3 opto-interrupter switches (small black squares) which controls the on/off of the motors via the software in the Picaxe 18X. The opto-interrupter switches are small horseshoe shaped devices that have a beam of light that passes from one side of the horseshoe to the other. A photo receptor on one side can sense if the beam of light is on or off. If something passes through the horseshoe the beam of light is “interrupted” and this event can be used as a switch in the micro controller software to do something. In my case I use it to stop or start different motors of the 2-3-2 system.

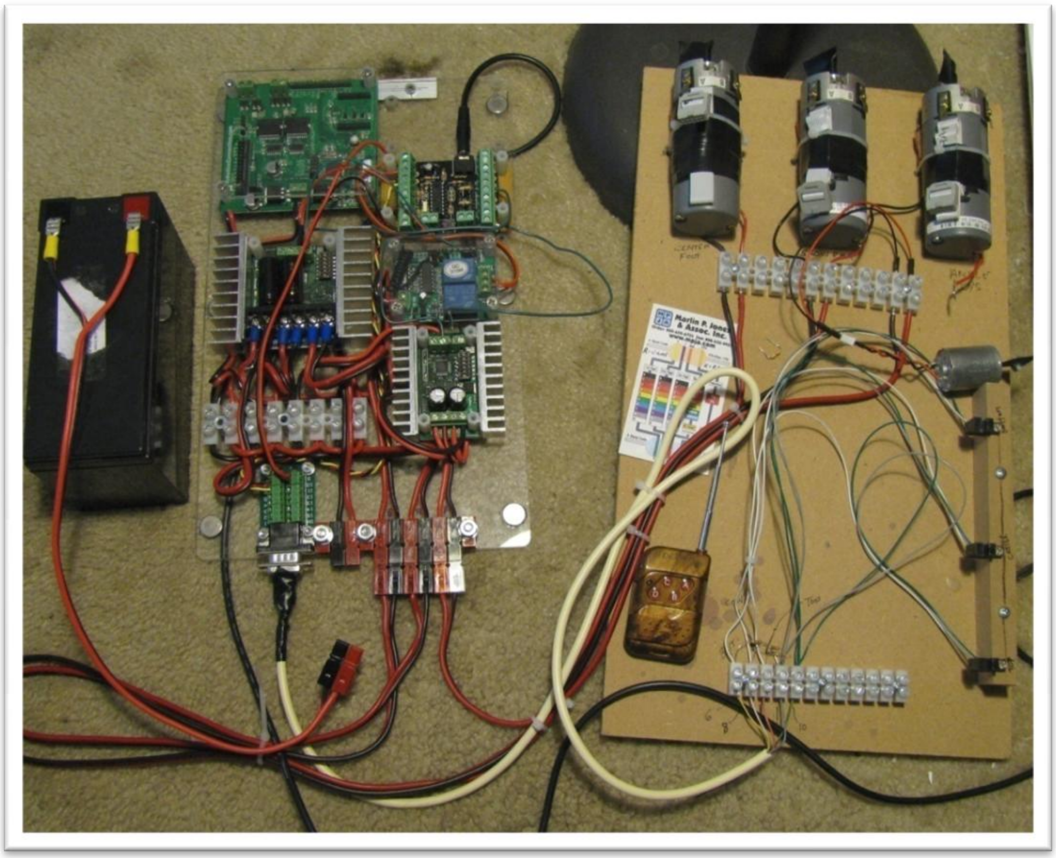


Figure 37: 2-3-2 Software test apparatus

In the actual droid I have a bar of Acetal plastic that I mounted vertically right next to the center foot lift/lower plate. On the lift/lower plate I mounted a flat piece of aluminum that cantilevers out past the edge of the plate. The cantilevered portion of the aluminum rides inside a groove within the Acetal plastic bar. The opt-interrupter switches to control the 2-3-2 motion are mounted in that Acetal plastic bar and therefore when the center foot moves up and down the aluminum cantilevered plate passes through each of the opt-interrupter switches triggering some part of the 2-3-2 motion coordination in the Picaxe software. Figure 38 shows the Acetal plastic guide bar with 2 of the 3 opto-interrupter switches installed. Figure 39 shows the opto-interrupter switch guide bar mounted in the droid next to the center foot lift plate. You can also see in Figure 39 (near the bottom of the guide bar) the cantilevered piece of aluminum mounted to the lifting platform that rides inside the groove of the Acetal plastic bar.

Figure 38: Acetal plastic guide bar for opto-interrupter switches (only 2 switches yet installed)

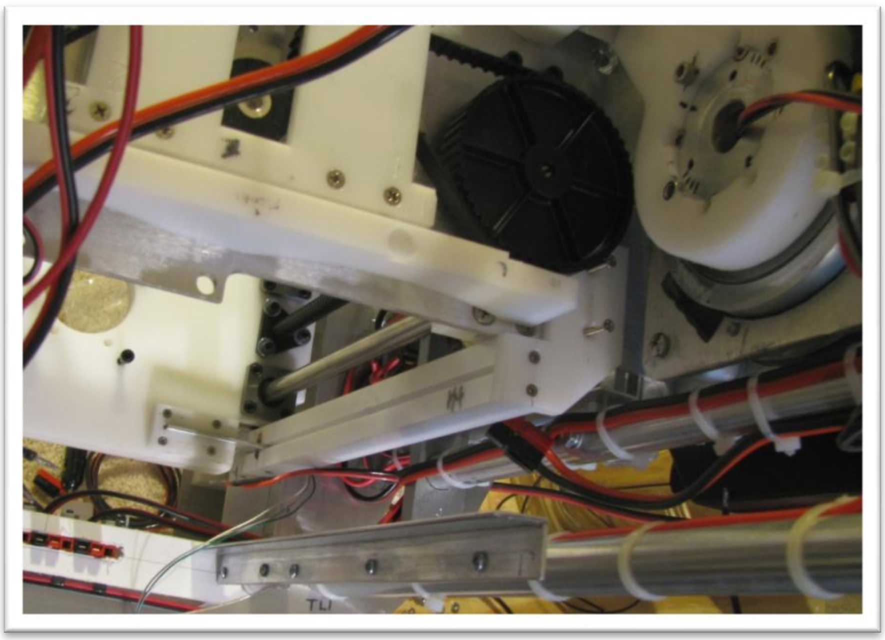


Figure 39: Acetal plastic guide bar installed

In the center of Figure 37 is the removable electronics panel I created for all of the animatronics of my droid. You can see on this panel I have the 2 SyRen motor speed controllers (one single motor speed controller and one dual motor speed controller), the 2 channel RF relay switch for triggering the 2-3-2 transitions and the Picaxe 18X project board that contains the software. Power for this board comes in from the bottom Anderson Power pole connectors when the board in plugged into the droid but for testing purposes a 12V battery is directly attached to the board (but disconnected in the photo).

The 2 motor speed controllers also have power pole connectors to drive the center foot motor and the 2 sets of linear actuators (seen connected in the photo). The control wires for the opto-interrupter switches are connected via a 15 channel sub D connector mounted on the bottom left of the animatronics board. Connections to the opt-interrupter switches in the droid are also established when the animatronics panel is plugged into the droid and the receiver for the sub D connector is mated with the end mounted on the animatronics panel. The top left circuit board is a Serializer board that is not yet being use but I do have future plans for it to control servos in the droid.

Picaxe 18X Micro Controller Software:

Below is the program logic I am using in the Picaxe software for each phase of both 2-3-2 transitions. Appendix A however, has the full Picaxe software I wrote and successfully test on my 2-3-2 system.

2-3 transition:

1. RF relay triggers the beginning of the 2-3 transition subroutine of the program.
2. The top opt-interrupter switch must be triggered in order to continue. This ensures the center foot is in the top position ready to move down.
3. Center foot begins travels down until center foot reaches the middle opt-interrupter switch.
4. When the middle opt-interrupter switch is triggered the shoulders to begin rotating to completion.
5. X seconds later ankle pins begin retracting to fully retracted position. X is a program variable that can easily be set.
6. The center foot continues to its fully extended position.
7. When the center foot reaches the bottom opt-interrupter switch the 2→3 transition is complete.

3-2 transition:

1. RF relay triggers the beginning of the 3-2 transition subroutine of the program.
2. The bottom opt-interrupter switch must be triggered in order to continue. This ensures the center foot is in the bottom position ready to move up.
3. Center foot begins retracting.
4. Simultaneously with (number 3) shoulders begin rotating to upright position.
5. Simultaneously with (number 3) the ankle pins begin extending to their fully extended position.
6. When the center foot reaches the top opt-interrupter switch the 3→2 transition is complete.

The actual eXperimental Picaxe 18x source code for the 2-3-2 system is listed in the Appendix A of the manual.

Unfortunately, just getting the software working was not enough. Since I am using a simple RF relay to trigger my 2-3-2 transitions I need to have a safety mechanism to prevent stray RF signals from potentially triggering a 2-3-2 transition at an inopportune time. The first line of defense was a switch that kills power to the animatronics board. While switched off no 2-3-2 transitions can happen. However, that means that none of the other animatronics can work either. As I am writing this I am thinking that what I really need to do is add a switch that kills the power to the RF relay board which would allow me to selectively turn off the 2-3-2 capability of my animatronics board. The second line of defense is software. In the Picaxe software I added routines to require that a certain sequence of buttons of the RF remote be pressed before the system become "armed" to perform the 2-3-2 transitions. In fact each transition requires an arming sequence before it can be triggered. Appendix A includes the software with this arming capability.

Future 2.4 GHz RF control via a web interface:

While having several RF devices to control the sound system, periscope up and down, 2-3-2 motions and a host of other devices works fine it would really be nice to be able to consolidate all of them into a single configurable interface device for sending commands to the droid. This is precisely what I am currently working on. The XBee RF modules have become very popular in the robotics community and for good reason. They are small, light, easy to set up, reliable and consume very little power. What's even more impressive is that the ZigBee communication protocol that is implemented is actually a networking protocol that let's each device communicate with any other device on the network. The hardware setup in the droid calls for each micro controller to have an XBee module that can receive serial commands. The Arduino boards already have support for XBee modules and Picaxe has release a new board with an XBee socket on it as well. Using the API (packet) mode a single "coordinator" or master XBee module can send serial commands to any of the other XBee modules on the network.

My concept is to create a user configurable web-based interface that will run from a tablet PC (think iPad running Windows). Dell is coming out soon with a tablet netbook PC (Inspiron Duo) that runs Windows 7. Since it runs Windows 7 it will have IIS (Internet Information Service) which is the Windows service for running websites. This will allow the user to set up a local host website with my droid interface. The website will be a configurable AJAX enabled tabbed interface that the user can configure anyway they like with whatever buttons they like. Each button the user creates can then be assigned to kick off a serial command script. The script runs locally on the tablet PC and sends the serial commands to the droid via the "coordinator" XBee module that is attached to the tablet PC via a USB port. The XBee module is extremely small and light so that it can be attached to the bottom of the table PC with Velcro.

Appendix A:

Picaxe 2-3-2 Control Code (Experimental):

```
' ELECTRONICS SETUP:
'
' PICAXE 18X:
' 1. Remove the Darlington driver IC from the PicAxe and jumper the pins you plan to use
'    with 330 Ohm resistors or replace it with a 16 pin 8x330 Ohm resistor DIL pack.
' 2. Output 1 of the PicAxe gets connected to S1 of the SyRen 25 motor speed controller.
' 3. Output 4 of the PicAxe gets connected to S1 of the Sabertooth 2x25 motor speed controller.
' 4. Output 5 of the PicAxe gets connected to S2 of the Sabertooth 2x25 motor speed controller.
'
' OPTO_INTERUPTER SWITCHES:
' 1. All 3 green leads from the Opto switches get connected together then the
'    green leads of the Opto switch gets connected to the G on the input side of the Picaxe.
```

```
' 2. All 3 of the gray leads from the Opto switches get connected together then the
'    gray leads of the Opto switchs gets connected to the PIC V+ on the input side of the Picaxe.
' 3. You must have at least a 1K or up to 3.3K Ohm resistor installed between all of the gray & white leads
'    of the Opto switches.
' 4. The 3 white leads of the Opto switchs get connected to pins (2, 6, 7) on the input side of the PicAxe.
'
' TWO CHANNEL RF RELAY:
' 1. The COM of the 2 CH RF relay gets connected to PIC V(+) on the input side of the PicAxe.
' 4. Relay connection 1 gets connected to 0 (pin0) on the input side of the PicAxe.
' 5. Relay connection 2 gets connected to 1 (pin1) on the input side of the PicAxe.
'
' MOTOR SPEED CONTROLLERS:
' 1. You must set the DIP switches on the SyRen 25 and Sabertooth 2x25 for "Simplified Serial" mode.
'    see the SyRen and Sabertooth manuals for DIP switch settings.
' 2. BAUD rate is set by DIP switches 4 & 5 on the motor spped controllers. I am using 2400 BAUD.
' 3. SyRen:                          0   = full reverse, 127 = stop,  255 = full forward.
' 4. Sabertooth: Motor 1:            1   = full reverse, 64  = stop,  127 = full forward.
'                Motor 2:            128 = full reverse, 192 = stop,  255 = full forward.
' 5. serout x is used to comtrol both motors on the Sabertooth.
'    tx line S1 is used to controll both motors just send a different range of numbers
'    tx line S2 is for a slave mode not the second motor.
'    Sending a value of 1-127 will command motor 1
'    Sending a value of 128-255 will command motor 2.
' 6. When powering 3 motors (2 on the SaberTooth & 1 on the SyRen) you need to have an extra ground
'    wire between the 2 motor speed controllers as a return path for the signal circuit.
'    When powering the PicAxe from the one or the other motor speed controller the ground wire is
'    present (i.e. the 0V wire from the 5V regulator on the speed controller). To add a common
'    signal ground run a wire from the 0V on one speed controller to the 0V on the other speed controllers
'    5V regulator (i.e. the 0V of the 5V power source provided on the speed controller).

pause 100
b0 = 0
main:

	' Main loop looks for pre-defined code sequences (ABC & CAB).
	' If found run a subroutine for 2 minutes then return to the main loop.
	' CODES:
	' 2->3 or 3->2 armed:			ABC = pin0, pin1, pin2 = CODE 1
	' Discrete motions armed:	CAB = pin2, pin0, pin1 = CODE 2
	'
	' Use variable b10, b11, b12 to indicate the correct sequence is being entered.
	do
		' Check for first number
		if pin0=1 then
			b10=1
		elseif pin1=1 then
			b10=10
		elseif pin2=1 then
			b10=2
		endif

		pause 1000
		if b10>0 then
			' Turn on first LED - indicates first button press was recorded
			high 1
			gosub	subCheckCode2
		endif
```

```
	loop while b0=0

goto main

subCheckCode2:

	' Check for 2nd number
	do
		if pin0=1 then
			b11=2
		elseif pin1=1 then
			b11=1
		elseif pin2=1 then
			b11=10
		endif

		pause 1000
		if b11>0 then
			' Turn on second LED - indicates second button press was recorded
			high 2
			gosub	subCheckCode3
		endif

	loop while b0=0

subCheckCode3:

	' Check for 3rd number
	do
		if pin0=1 then
			b12=10
		elseif pin1=1 then
			b12=2
		elseif pin2=1 then
			b12=1
		endif

		pause 1000
		if b12>0 then
			' Turn on third LED - indicates third button press was recorded
			high 3
			gosub	subCheckValidCode
		endif

	loop while b0=0

subCheckValidCode:

	' If either 3 letter code was entered reset the variables and enter armed mode.
	if b10=1 and b11=1 and b12=1 then
		b0=0
		b10=0
		b11=0
		b12=0
		gosub exe23or32
	elseif b10=2 and b11=2 and b12=2 then
		b0=0
		b10=0
```

```
				b11=0
				b12=0
				gosub exeDiscrete
		endif

		b0=0
		b10=0
		b11=0
		b12=0

		' Turn off LEDs
		low 1
		low 2
		low 3
		low 4

goto main


exe23or32:

		' Turn on 2-3-2 mode armed LED
		high 4

		' loop until a subroutine is called or pin6 = 1 for 3 seconds.
		' use pin6 since that is the center opto switch and not likely to be in an interrupted state.
		do

				if pin0 = 1 then				' Testing for CH1 of the RF relay to be closed - Activates 2 -> 3 cycle.
						gosub subTwoToThree
						exit
				endif

				if pin1 = 1 then				' Testing for CH2 of the RF relay to be closed - Activates 3 -> 2 cycle.
						gosub subThreeToTwo
						exit
				endif

				if pin6 = 1 then				' Push and hold for 3 seconds to exit the armed mode
						pause 3000
						if pin6 = 1 then		' If pin6 is still = 1 after 3 seconds exit the armed condition.
								b0 = 1
						endif
				endif

				pause 200

		loop while b0=0
		b0=0

return


exeDiscrete:

		' loop until a subroutine is called or pin6 = 1 for 3 seconds
		' use pin6 since that is the center opto switch and not likely to be in an interrupted state.
		do
```

```
' blink the LED for armed in discrete mode
if b1=0 then
		b1=1
		high 4
else
		b1=0
		low 4
end if

if pin0=0 and pin1=1 and pin2=1 and pin6=1 then ' Testing for ony pin0=1

		do
				' Move center foot motor in a 2->3 direction (extend)
				serout 1,T2400_4,(255)		' Start center foot motor (PicAxe output 1)
											' SyRen ESC so use (255) as the full forward command.
				pause 100
		loop while pin0=0
		serout 1,T2400_4,(127) ' Stop the center foot motor

elseif pin0=1 and pin1=0 and pin2=1 and pin6=1 then ' Testing for ony pin1=0

		do
				' Move shoulders in a 2->3 direction (push up)
				serout 4,T2400_4,(127)		' Start shoulder lever linear actuators (PicAxe output 4)
								' Sabertooth motor 1 so use (127) as the full forward command.
				pause 100
		loop while pin1=0
		serout 4,T2400_4,(64) ' Stop the shoulder lever linear actuators

elseif pin0=1 and pin1=1 and pin2=0 and pin6=1 then ' Testing for ony pin2=0

		do
				' Move ankle pins in a 2->3 direction (retract)
				serout 4,T2400_4,(128)		' Start ankle pin linear actuators (PicAxe output 4)
								' Sabertooth motor 2 so use (128) as the full forward command.
				pause 100
		loop while pin2=0
		serout 4,T2400_4,(192) ' Stop the ankle pin linear actuators

elseif pin0=0 and pin1=1 and pin2=1 and pin6=0 then ' Testing for ony pin0 and pin6=0

		do
				' Move center foot motor in a 3->2 direction (retract)
				serout 1,T2400_4,(0)		' Start center foot motor (PicAxe output 1)
											' SyRen ESC so use (0) as the full reverse command.
				pause 100
		loop while pin0=0 and pin6=0
		serout 1,T2400_4,(127) ' Stop the center foot motor

elseif pin0=1 and pin1=0 and pin2=1 and pin6=0 then ' Testing for ony pin1 and pin6=0

		do
				' Move shoulders in a 3->2 direction (pull down)
				serout 4,T2400_4,(1)		' Start shoulder lever linear actuators (PicAxe output 4)
								' Sabertooth motor 1 so use (1) as the full reverse command.
				pause 100
		loop while pin1=0 and pin6=0
```

```
				serout 4,T2400_4,(64) ' Stop the shoulder lever linear actuators

		elseif pin0=1 and pin1=1 and pin2=0 and pin6=0 then ' Testing for ony pin2 and pin6=0

				do
					' Move ankle pins in a 3->2 direction (extend)
					serout 4,T2400_4,(128)		' Start ankle pins linear actuators (PicAxe output 4)
								' Sabertooth motor 2 so use (128) as the full reverse command.
					pause 100
				loop while pin2=0 and pin6=0
				serout 4,T2400_4,(192) ' Stop the ankle pin lever linear actuators

		endif

		if pin6 = 1 then				' Push and hold for 3 seconds to exit the armed mode
				pause 3000
				if pin6 = 1 then			' If pin6 is still = 1 after 3 seconds exit the armed condition.
					b0 = 1
				endif
		endif

		pause 200

	loop while b0=0
	b0=0
	b1=0

return

subTwoToThree:

	do
		if		pin2 = 1 and pin6 = 1 and pin7 = 0 then ' Only the top	opto sensor pin7 is triggered
				goto subTwoToThree1stPhase
		elseif pin2 = 1 and pin6 = 0 and pin7 = 1 then ' Only the middle opto sensor pin6 is triggered
				goto subTwoToThree2ndPhase
		elseif pin2 = 0 and pin6 = 1 and pin7 = 1 then ' Only the bottom opto sensor pin2 is triggered
				goto subTwoToThreeComplete
		else
				exit
		endif

		subTwoToThree1stPhase:
		do
				serout 1,T2400_4,(255)		' Start center foot motor (PicAxe output 1)
								' SyRen ESC so use (0) as the full forward command.

				if pin0 = 0 then			' Test for CH1 of the RF relay to be opened - stop all motors
					gosub	subStopAllMotors
				endif

		loop until pin2 = 1 and pin6 = 0 and pin7 = 1		' Center opto sensor pin6 has been reached.

		subTwoToThree2ndPhase:
		do
				serout 4,T2400_4,(127)		' Start shoulder lever linear actuators (PicAxe output 4)
								' Sabertooth motor 1 so use (127) as the full forward command.
```

```
                    pause 1500                              ' Wait until foot extends more before releasing ankle pins

                    serout 4,T2400_4,(255)        ' Start ankle pin linear actuators             (PicAxe output 4)
                                                  ' Sabertooth motor 2 so use (255) as the full forward command.


                    if pin0 = 0 then              ' Test for CH1 of the RF relay to be opened - stop all motors
                              gosub     subStopAllMotors
                    endif

          loop until pin2 = 0 and pin6 = 1 and pin7 = 1       ' Bottom opto sensor pin2 has been triggered.

          subTwoToThreeComplete:
          gosub     subStopAllMotors
          b0 = 1

     loop while b0=0

return

subThreeToTwo:

     do
          if        pin2 = 0 and pin6 = 1 and pin7 = 1 then ' Only the bottom opto sensor pin2 is triggered
                    goto subThreeToTwoStart
          elseif pin2 = 1 and pin6 = 1 and pin7 = 0 then ' Only the top    opto sensor pin7 is triggered
                    goto subThreeToTwoComplete
          else
                    exit
          endif

          subThreeToTwoStart:
          do
                    serout 1,T2400_4,(0)          ' Reverse center foot motor.
                                                  ' SyRen ESC so use (254) as the full reverse command.

                    serout 4,T2400_4,(1)          ' Reverse shoulder lever linear actuators (PicAxe output 4)
                                                  ' Sabertooth motor 1 so use (1) as the full reverse command.

                    serout 4,T2400_4,(128)        ' Reverse ankle pin linear actuators (PicAxe output 4)
                                                  ' Sabertooth motor 2 so use (128) as the full reverse command.

                    if pin1 = 0 then              ' Test for CH2 of the RF relay to be opened - stop all motors
                              gosub     subStopAllMotors
                    endif

          loop until pin2 = 1 and pin6 = 1 and pin7 = 0       ' Top opto switch has been reached.

          subThreeToTwoComplete:
          gosub     subStopAllMotors
          b1 = 1

     loop while b1=0

return

subStopAllMotors:
```

```
        ' Stop all motors
        serout 1,T2400_4,(127)        ' SyRen ESC so use (127) as the full stop command.
        serout 4,T2400_4,(64)         ' Sabertooth motor 1 so use (64) as the full stop command.
        serout 4,T2400_4,(192)        ' Sabertooth motor 2 so use (192) as the full stop command.

return
```