2-3-2: An adventure in amateur engineering!

Ted Koenig

Call me Ishmael. *AHEM* Ok not really, call me Ted ☺ Anyways, the important part is that this should be a more thorough summary of the process of building my 2-3-2 droid R4-I9 than was contained in my build log. Many photos will likely be the same, but here I can include more of that boring engineering stuff I love to drone on about. JOY! I will explain a bit more about the choices/tradeoffs I made for certain things, and why. I will examine each major system involved in detail, and hopefully this will be of use to other considering a similar project ☺

OVERVIEW

There were several goals I had in mind for my 2-3-2, and these informed all of my choices during the build:

Keep it as "cheap" as possible.

I knew I wouldn't be able to do it as cheaply as my first basic RC build, and certainly no droid build is really an inexpensive undertaking, but I wanted this build to come in well under the cost of many basic RC builds. There were several choices I made to facilitate this goal…

- Use nearly all resin detail parts. It saved gobs of money to do so, since there is so many details on a droid! Also, they help save weight!
- Buy broken/damaged parts on the junkyard and see if I can restore them. Sometimes people mess up, and want to start over on a part. I'm pretty good with bondo and sanding, so often I'm able to undo the damage and make use of that part anyways ☺
- Use wood legs. A full set of just the structural parts of a pair of aluminum legs is something like $1500. The ATL wood legs I used in particular can bolt on the same way a aluminum ones would, are much cheaper, and quite sturdy.
- Scooter drives. Scooter motors are way cheaper than NPCs, and have quite good power.
- Styrene dome. Now to be fair, aluminum R4 domes have never been offered, but nonetheless, styrene domes of any kind look great once completed, weigh less, cost less, and are in general a fantastic deal.
- Keep the weight down. Lower weight means smaller actuators, thinner metal parts, lower power, and less money. My target weight was <100 lbs. For lack of a good way to measure the final weight I'm not certain what it came out to exactly, but it feels close ☺

Keep the weight as low as reasonable

As said above, lower weight helps with cost, but it's a goal in and of itself as well. A lower weight makes the droid easier to transport. It causes less damage to my back to lift and move around the shop while building ☺ Also, it makes it easier to find motors, actuators and other parts that will be able to handle the loads required. There were some choices made to make this happen:

- Use bracing rather than thickness for extra load areas. A thicker part with more complex machining to lighten it can accomplish similar results, but bracing is simpler and cheaper as well.
- Use resin and styrene parts. Again, they don't just save money, they often weight less too. This is less of a thing with the most recent super diet versions of many aluminum parts, but solid resin still often comes in lighter than diet aluminum, if only by a few ounces ☺
- Don't overkill. Sure using a half inch thick leg plate means it will never ever bend, but it's also probably over twice the thickness required for the load ☺ This goes for motors, battery, everything. Test the capacity of things, and try to determine more closely the real strength required. Overkill is the sworn enemy of low cost and weight. The only place I endorse overkill is on wire thickness ☺

Keep it simple!

Simplicity helps with both lower weight and lower cost, but again, it has its own advantages. Simple designs are more certain to work, easier to fix, and easier to debug if they indeed go wrong. The simplest solution is usually the correct one, and this applies to engineering as well☺. In my case there was the additional motivation that honestly I couldn't afford to be wrong too much. This meant I had to go with designs that I was nearly certain would work the first time, and simpler designs carry lower risk. So here are a few things I did to keep things simple:

- Use proven designs where possible. I read every other 2-3-2 build log I could find, and the excellent write-ups done by Blake and CuriousMarc, and I borrowed the ideas that suited me.
- When a new design is needed, go with the simplest thing you think will work. This helps minimize (but not eliminate!) re-doing things, and the waste that results from it.
- Compromise where required. If getting that "last 10%" makes the cost increase or the weight increase more than I felt was acceptable, I chose not to chase it.
- Minimize machining. Extra machining adds cost, time and complexity. Where possible, I avoided it.

Problem #1: Shoulders!

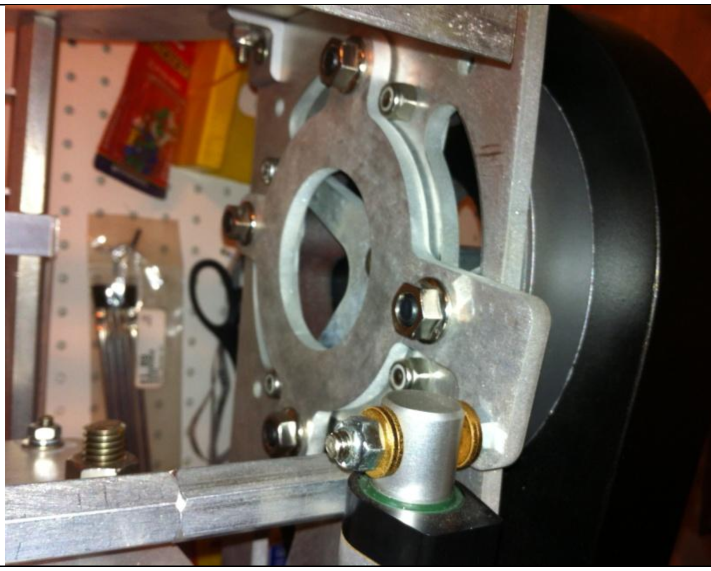
The basic assembly

Here's how the shoulder assembly goes together:

The sort of wobbly x shaped thing I call a "spider". It's purpose is to provide an anchor point for the top end of the leg rods that will control the ankle angle. The beefy aluminum standoff shown provides the spacing a the proper distance to make the leg rod run down the center of the leg. In the production version of the frame this spaces is steel, for extra strength, though it's not necessarily required, it adds minimal weight or cost so why not ☺

The 4 curved slots are where the leg bolts ride, they are 5/8" wide roughly, and the ½" bolts pass through flanged bronze bushings to move smoothly.

Here I have some bolts (not the shoulder bolts I actually use now, old pic) passing through my leg-to-body hubs, then the flanged bushings, to the other side. You can see how the spider's goofy shape keeps it out of the way of the travel of the hub as it would pivot. This prototype hub was made for me by joymonkey out of CNC cut acrylic, but again the production ones are beefier, made of aluminum instead of plastic.



Here you can see the inside of the assembly. The bolts pass through the shoulder plate, riding in their bushings, and then hidden behind the lever are 1/16" bronze thrust washers, which help it to move smoothly across the inside of the shoulder plate. The shoulder plate then meets the bolts right at the end of their shoulder, and has 3/8" sized holes for the threaded portion at the end, then I used flanged nuts with serrated back to lock them in place. This all makes for a very rigid shoulder assembly, though admittedy it's a little tricky to assemble. The clearances also are tight to minimize any play in the mechanism.

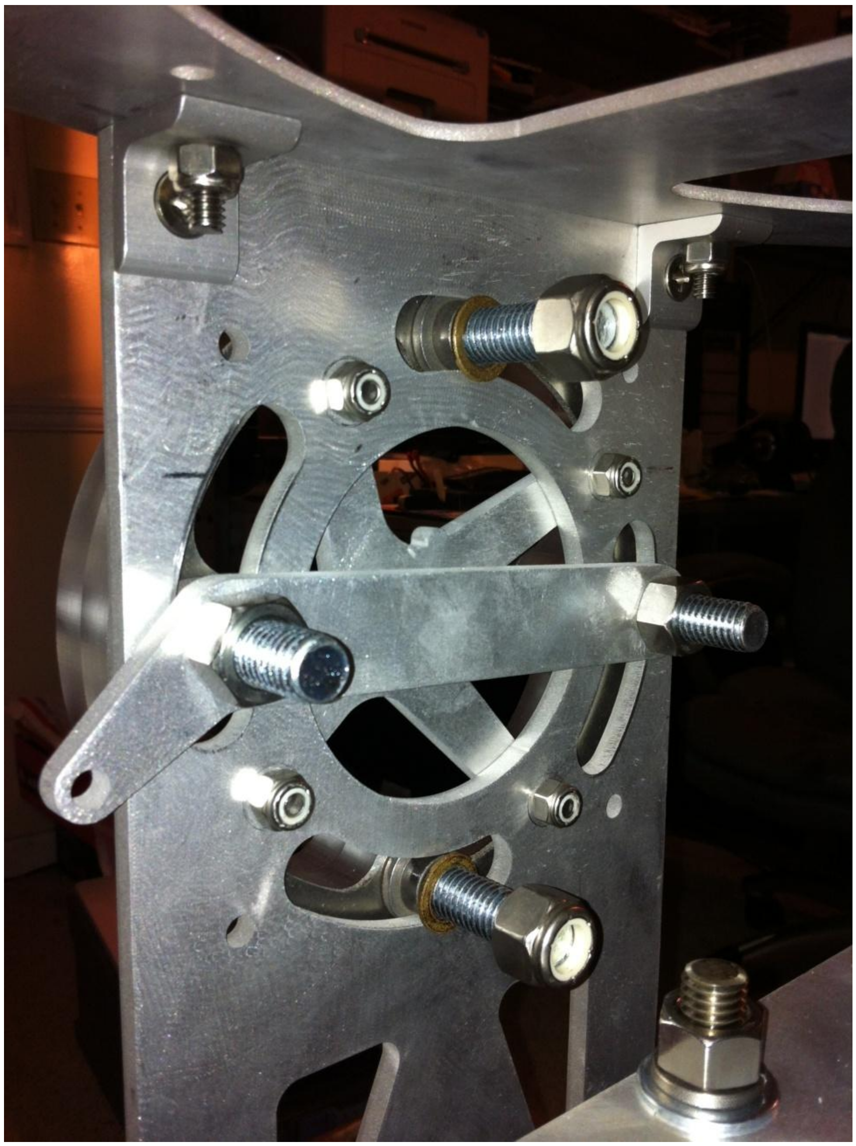
The actuator then attaches at the top with a ¼-20 screw to the lever, with a plain nut to hold it square, then a pair of ¼" thrust washers on each side, then a nylon locknut to keep it on there.

Similarly at the bottom, a screw, the nut to square it, then a bushing for spacing and smoothness, then a locknut.

The Levers

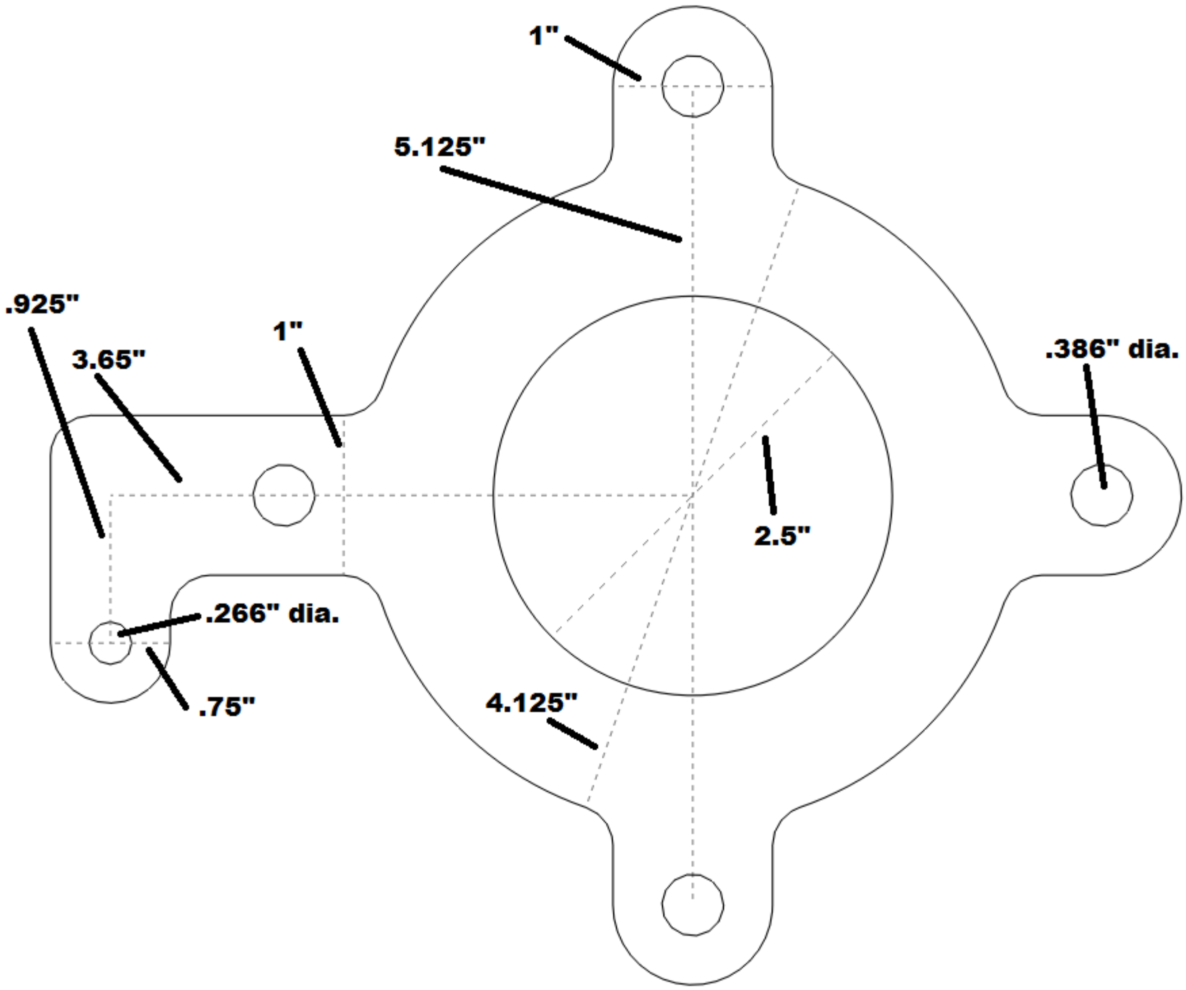
My shoulder design is based heavily on Marc's. His write up included the dimensions of his levers for the shoulder pivot, which I used for my version. This was one area where my first try didn't work as desired, and I had to do a redesign.

This was the original version:



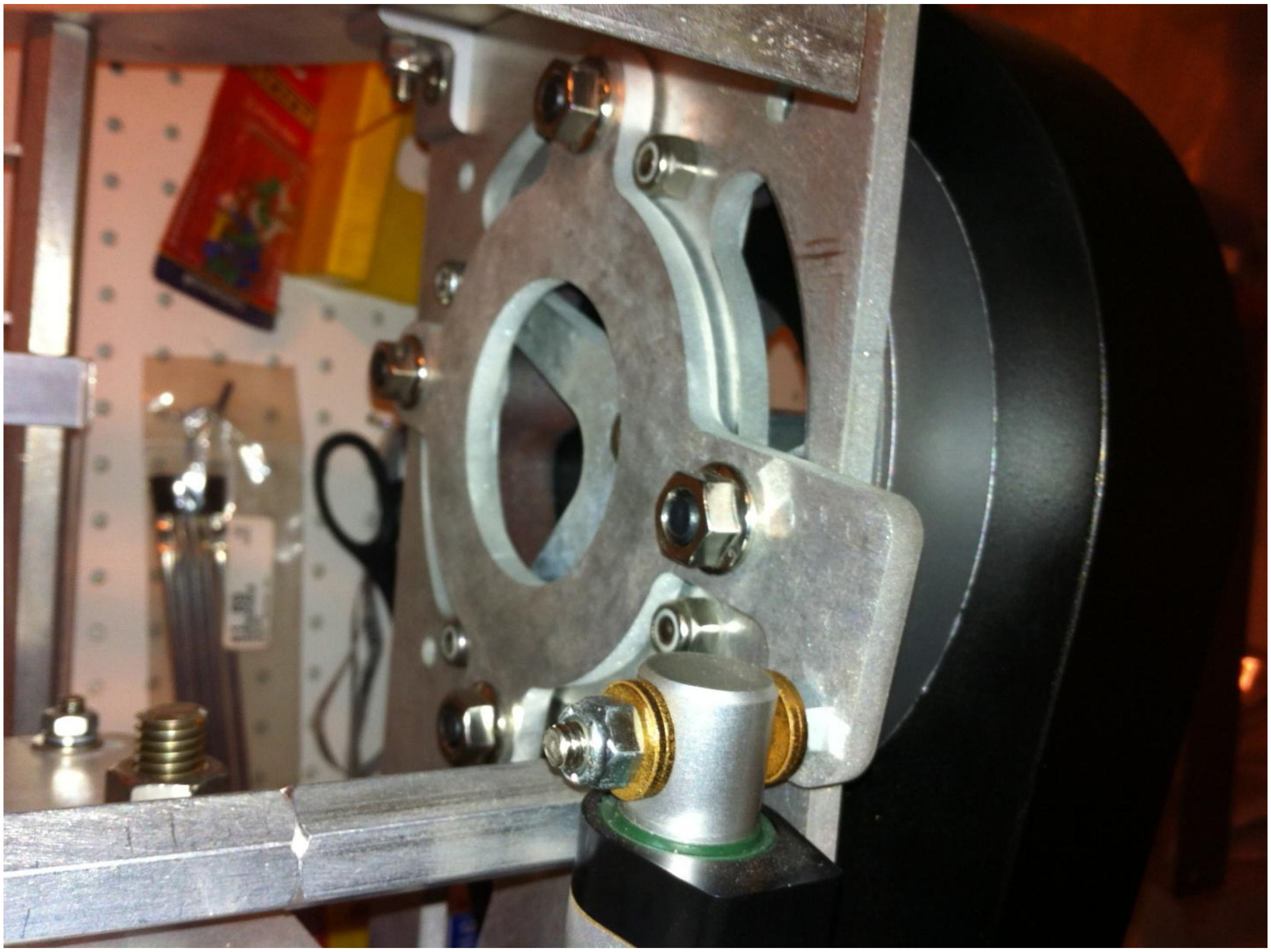
There was a few things wrong with this. It only attached to two of the bolts, so on the others they were sort of dragged along by the leg-to-body hub. Also, the bolts I used were wrong. Using simple hex bolts was cheap, but allows the bolts to wiggle in the assembly, since there is nothing but their tension to hold them still. All this conspired to cause a lot of wiggle and inaccuracy in shoulder movement. If I tightened it more to try and alleviate it, it just meant that my shoulders were too tight to move at all.

Here is the design I came up with to fix these issues:



This has several advantages. All four bolts are shoulder bolts, locked into this plate to keep all four in precise relative position to each other. It pushes all four at once, so the toque goes to the whole thing, making the leg-to-body hub little more than a big spacer, and all of this means that the leg is much more securely locked to this plate, making for much more precise shoulder movement, and almost no play.

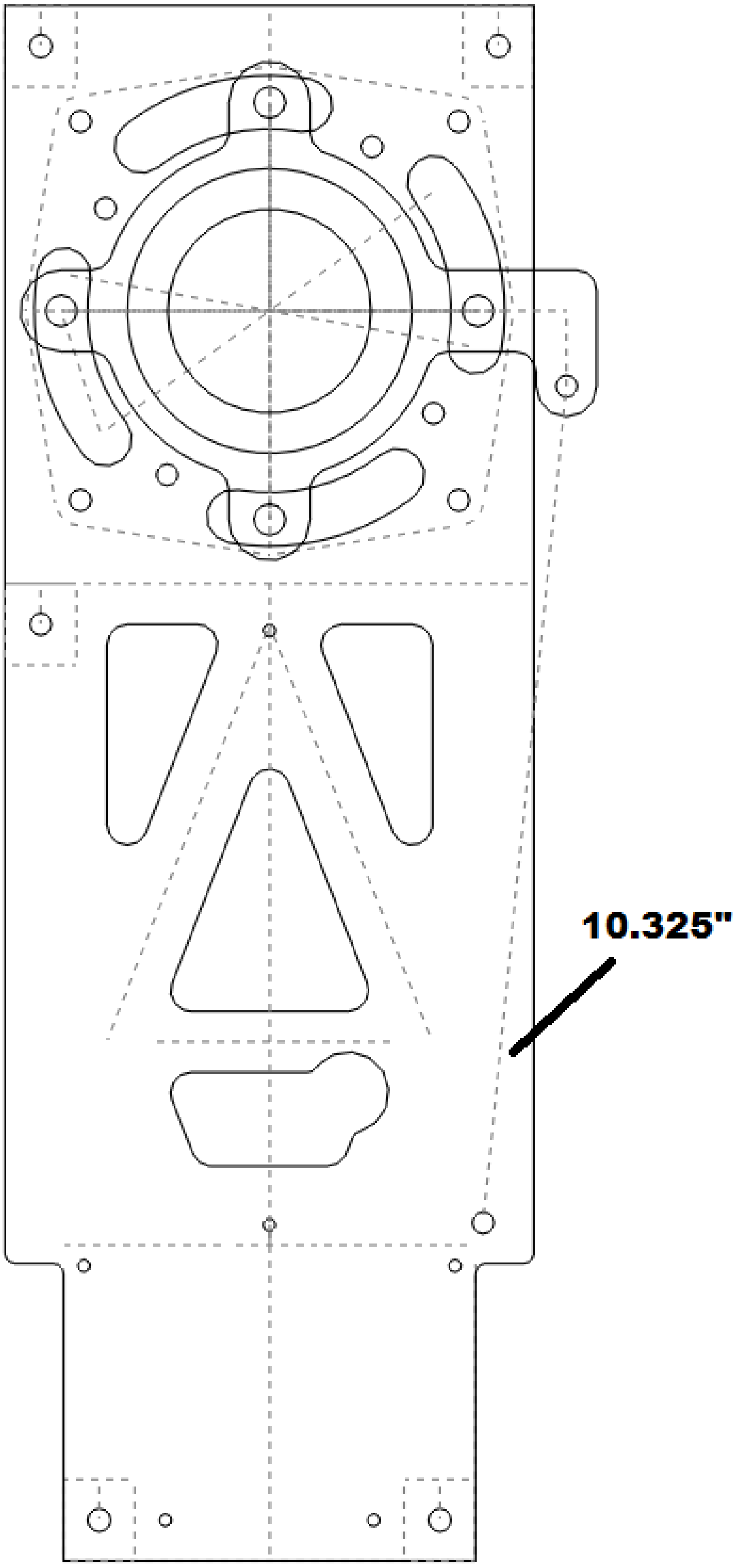
Here it is installed:



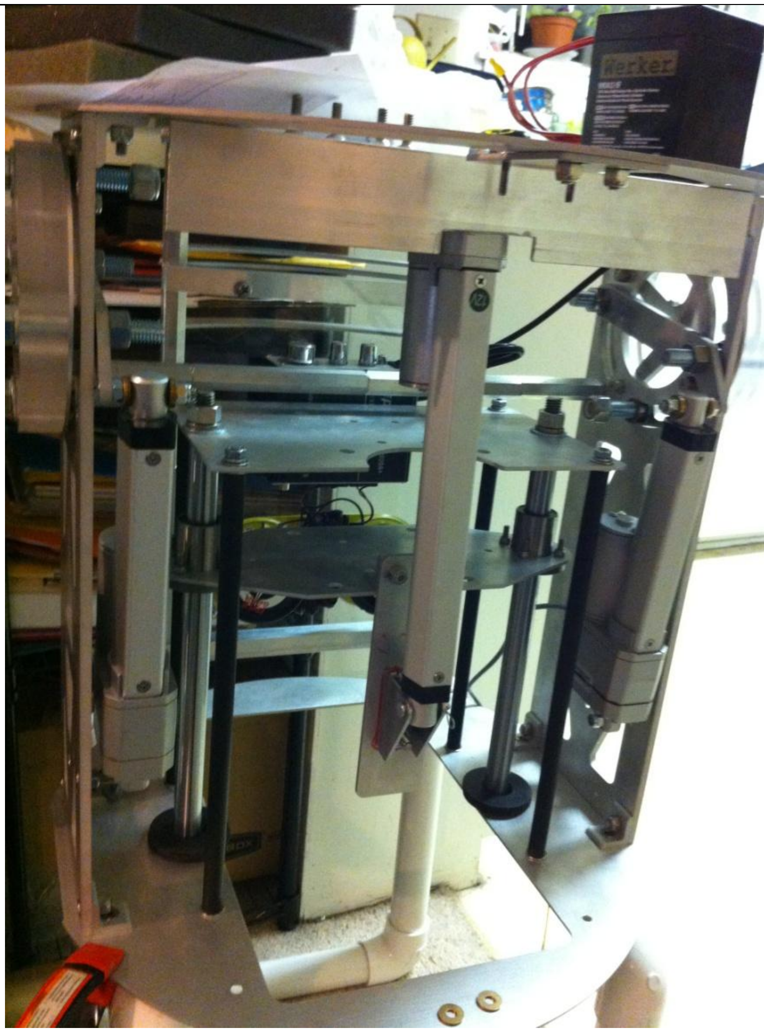
The important dimensions here are that the round part in the middle must fit inside the track of the 4 leg bolts, the lever for the actuator must be short enough to not run into the skins, and the range of motion of the lever must fit within the travel of the actuator used.

The Actuators

<table>
<tr>
<td>
</td>
<td>This is the actuator I chose for the shoulders:
http://www.pololu.com/catalog/product/2305
It has a 4” range of travel, and it is 9.7” between the mounting holes fully retracted, 13.6” full extension.</td>
</tr>
</table>



I liked the idea of making the shoulder mechanism a more or less contained unit, based on the structural support of the leg plate, and the dimensions of this actuator make that possible. I placed the lower mounting hole inside the range of the actuator, such that I could lean forward ten degrees if I wanted to.

	This shows how the actuators all bolt in more or less. Didn't take a pic like this with the skins off and the new levers installed, but you get the idea I hope ☺

Control & Synchronization

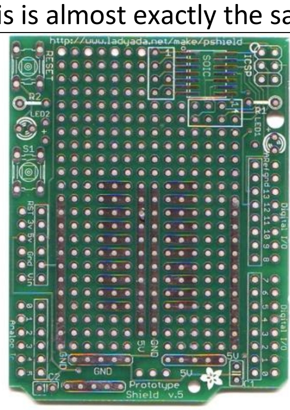
Well in addition to the regular stuff any RC droid has, like the foot drive and dome motor, with 2-3-2 you have the center leg motor, and in my case 2 more, each of which controls the position of an outer leg. The center leg actuator is relatively simple, since the only valid positions for it really are all the way up, and all the way down. Both these positions go right to the end of travel, so I allow the limit switches built into the actuator to stop it at either end of it's travel, and a spare channel on my second sabretooth which controls the dome motor to drive it. Really a simple relay would do, and I may replace it with that and use that other motor channel for something else one day ☺

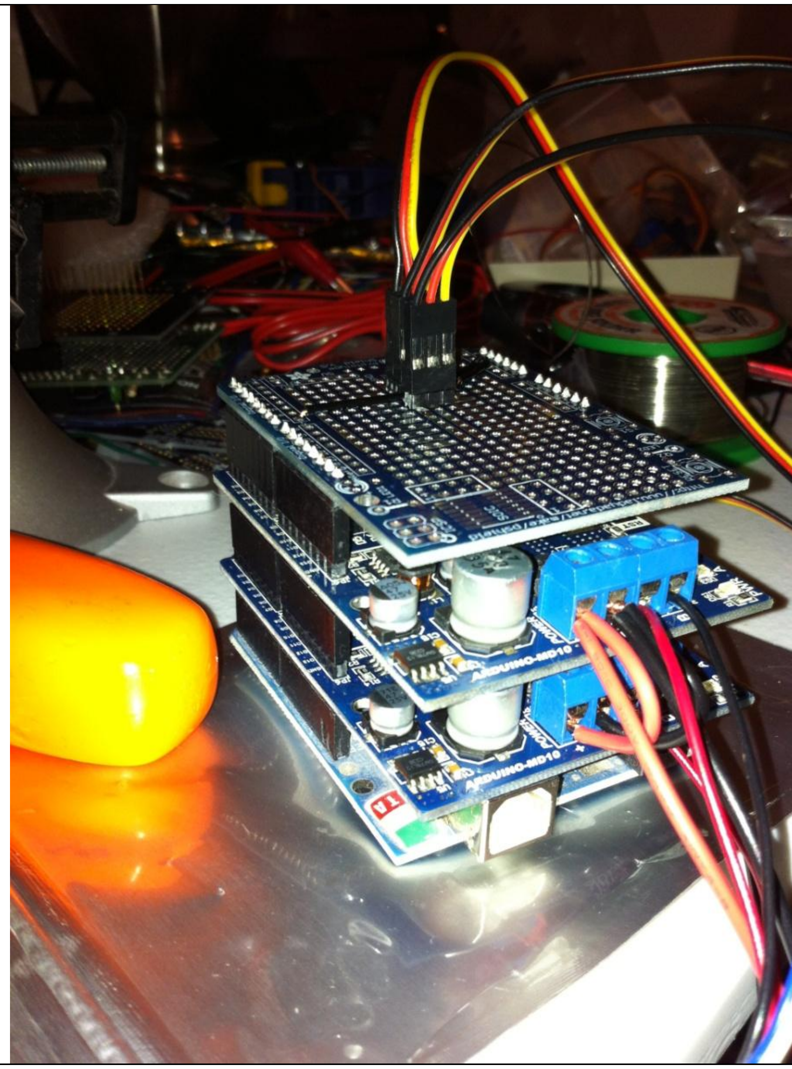
The outer legs however, present a more challenging problem. Again, borrowing a bit from CuriousMarc's build for my levers, I have then of course borrowed incidentally the travel required for the legs, which is roughly 3". I have 4" travel actuators. That means I have to "soft" stop them on either end, before their mechanical limit. If I don't they will run into the end of the slots and possibly break. Also, I wan tthose soft stop positions to correspond to the legs being straight vertical, and swept back 36 degrees per specs. Additionally, they must both stay more or less in sync with each other while in motion, otherwise things might get unstable, and he could fall over during transition.

The actuators I chose include a potentiometer built into them, to read the position of the actuator by it's resistance, and this is the feedback mechanism I use to control them. Since my shoulder mechanism with the special levers is so rigid, and has so little play, this system has pretty good accuracy, despite not reading the shoulder angle directly.

SO, how to control the motors? Well there are commercial motor controllers that could do the job, such as the pololu JRK series. However, since I already have so many arduinos, am quite comfortable programming them, and the motor controller shields were so cheap, I chose to roll my own control mechanism for them ☺

Here's what I used:

	http://www.robotshop.com/productinfo.aspx?pc=RB-Cyt-116&lang=en-US This is a cheap, stackable 10A motor driver shield fitting any standard sized arduino. I set the pin selection with the jumpers, stacked up a pair of them on top of my uno, and Bob's your uncle.
http://www.sainsmart.com/arduino-compatibles-1/sainsmart-uno-r3-atmega328-au-development-board-compatible-with-arduino-uno-r3.html Arduino UNO compatible. This is the sainsmart knockoff, super cheap, it works, what more could you want. They are even cheaper on ebay, and at this point, entirely ubiquitous. I happened to have a very old arduino diecimila from 2008 or so lying about and actually used that, but this is almost exactly the same thing.	
	http://www.adafruit.com/products/55 This is the bare PCB for the adafruit proto shield. I just got this since I didn't need the rest of the gubbins that come with the full proto shield set. I just wanted a blank shield to put a few headers and jumpers on to simplify hookup.

And this odd monstrosity is what I built. I stuck a few headers on the outer edges to stack it onto the motor shield below it, a few along the power rails in the center with 3 pin sets for connecting the two pot inputs, and the PWM input from the RC receiver, and soldered in small jumpers, connecting the relevant pins for those inputs to the 3-pin header’s signal pins. With the right software, this could control my pair of actuators ☺	

So, that’s the hardware more or less, as for the software….

(You can find this code in the SVN for the r-series project btw, and that may be the best source. Someday the I2C connection might not just be for debug, it might actually connect to the r-series system and be controlled differently, so the version in SVN is what you really want, though I imagine the overall gist of its operation will remain.)

```
#include <Wire.h>
#include <Servo.h>

const int leftSensorPin = A2;
const int rightSensorPin = A3;
const int servoInputPin = 7;

int pwm_a = 10;  //PWM control for motor outputs 1 and 2 is on digital pin 10
int pwm_b = 11;  //PWM control for motor outputs 3 and 4 is on digital pin 11
```

```
int dir_a = 12;  //direction control for motor outputs 1 and 2 is on digital pin 12
int dir_b = 13;  //direction control for motor outputs 3 and 4 is on digital pin 13

const int lowStopValue = 215;
const int highStopValue = 730;

const int pulseMin = 1090;
const int pulseMax = 1900;

const int hysterisis = 10;

int goalValue = 0;
int leftValue = 0;
int rightValue = 0;
int x = 0;

void setup() {
  // put your setup code here, to run once:
    Serial.begin(9600);
    Serial.println("R-Series Shoulder motor synchronizer and Controller.");
    pinMode(servoInputPin, INPUT);

    pinMode(pwm_a, OUTPUT);  //Set control pins to be outputs
    pinMode(pwm_b, OUTPUT);
    pinMode(dir_a, OUTPUT);
    pinMode(dir_b, OUTPUT);

    analogWrite(pwm_a, 0);
    analogWrite(pwm_b, 0);

    Wire.begin(7);              // join i2c bus with address #7
    Wire.onReceive(receiveEvent); // register event
    Serial.println("done startup!");
}

void loop() {
  // put your main code here, to run repeatedly:
    readPots();
    readGoalValue();
    motorTick(dir_a, pwm_a, leftValue);
    motorTick(dir_b, pwm_b, rightValue);
    delay(10);
}

void motorTick(int motorpindir, int motorpinspd, int potVal){
    if(closeEnough(potVal, goalValue)){
      digitalWrite(motorpindir, LOW);  //Set motor direction, 1 low, 2 high
      analogWrite(motorpinspd, 0);
```

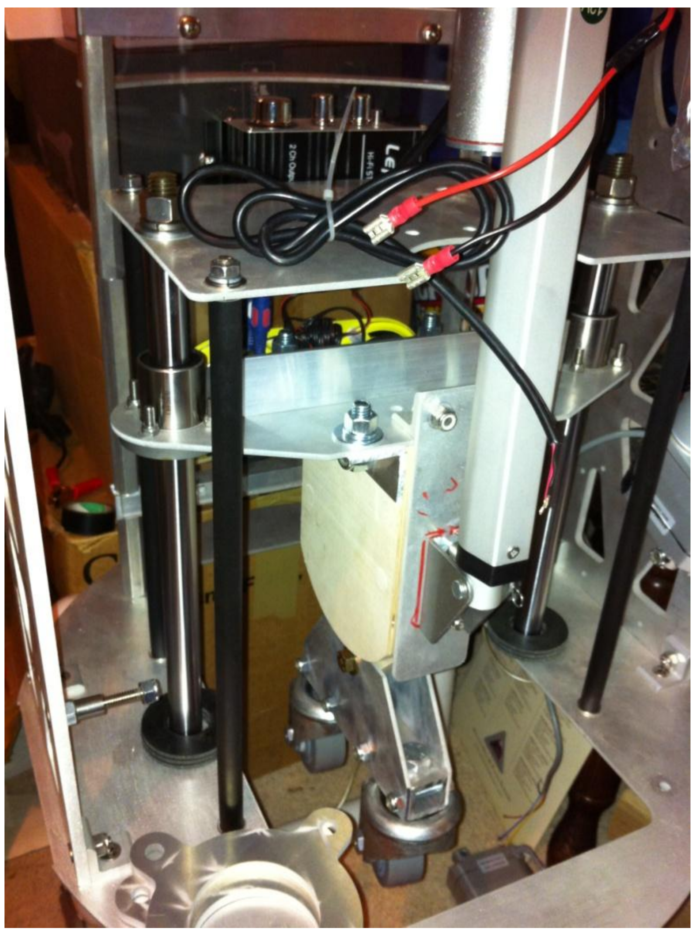
```
  }else if(potVal < goalValue){
    digitalWrite(motorpindir, HIGH);
    analogWrite(motorpinspd, 255);
  }else{
    digitalWrite(motorpindir, LOW);
    analogWrite(motorpinspd, 255);
  }
}

boolean closeEnough(int testVal, int targetVal){
  if( abs(testVal - targetVal) <= hysterisis ){
     return true;
  }
  return false;
}

void receiveEvent(int howMany) {
  x = Wire.read();    // receive byte as an integer
  Serial.println(x);        // print the integer - DEBUG CODE
}

void readPots(){
  leftValue = analogRead(leftSensorPin);
  rightValue = analogRead(rightSensorPin);
}

void readGoalValue(){
  int pw = pulseIn(servoInputPin, HIGH); // read pulse
  pw = constrain(pw, pulseMin, pulseMax);
  goalValue = map(pw, pulseMin, pulseMax, lowStopValue, highStopValue); // scale to match
pot range
}
```

So, what's all that do? Well the real magic is that last function, "readGoalValue()". That reads the input PWM pin, makes sure the input is within the known range pulseMin <= pw <= pulseMax, then maps that value, across the range (lowStopValue, highStopValue). Put simply, it translates the input signal from the RC receiver into a potentiometer value that the motorTick then tries to move the motors to match. MotorTick just runs all the time, every 10ms, so it's just dumbly sitting there waiting for its goal to change, and when it does, it tries to move to match it, as long as that goal has shifted by some meaningful amount (hysteresis). That last bit, the hysteresis, is a simplistic way of dealing with the inevitable electrical "noise" which is endemic in any analog device, such as a potentiometer. This means that the actuator position always, as a ratio within its high and low stop positions, matches the position of the control stick on the radio. Basically I just turned the actuators into a pair of really big servos☺. By sweeping that value at some rate that both motors can match, they keep themselves in sync too, since they are both trying to match the same value. Compared to my first attempt, this is surprisingly simple, and in practice, it's been very effective so far ☺

The values I used for the high and low stop, hysteresis, and pwm values were all determined experimentally. So I wrote little sketches that just printed the values read in from the PWM, or from the pots, on the serial console, and then manually moved the actuators to see what values meant 2 leg position, what meant 3-leg position, etc. These would likely be identical or very similar for the same brand actuators, in the same frame and position, but for your setup, if it varies, you would need to determine these as well and tune them to the right values if you use this sketch.

Problem #2 – Center leg lift!

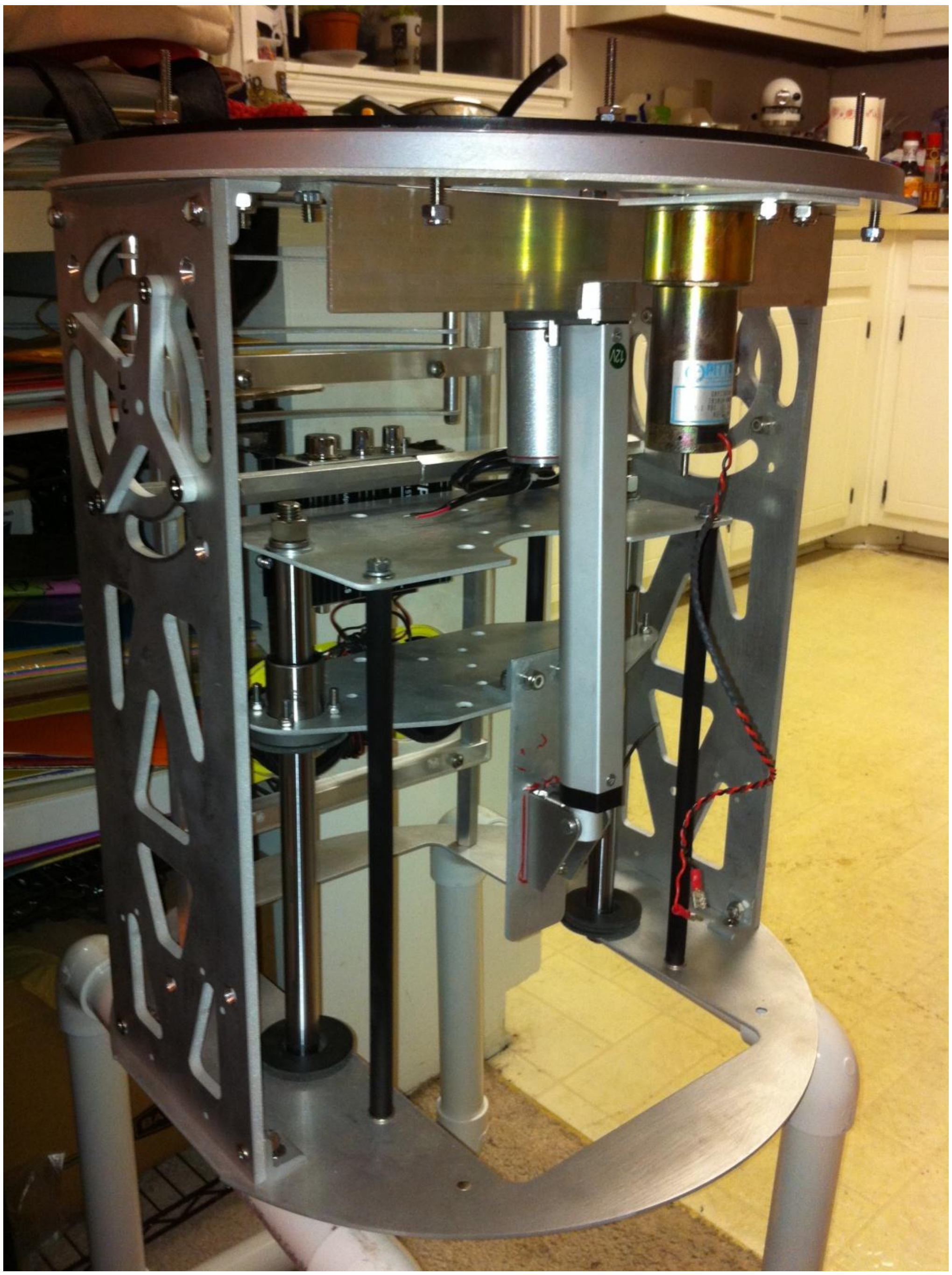
Well the most obvious part of any 2-3-2 setup is that there needs to be a mechanism for retracting and extending the center leg. I went for the simplest mechanism I could think of, which was to use a pair of linear rods centered in the frame, and to have the center leg ride on a plate that moves up and down along them. In order for the center leg to retract fully into the body of a club spec droid, it has to have a travel of approximately 10 inches. Commodity linear actuators rarely come in 10" travel, but 8" and 12" are available. 12" ones would have extended too far down the back of the ankle, or alternately into the dome, so I went with 8". This means that the bottoms of the caster wheels are visible below the skirt when fully retracted, but this turns out to be not noticeable at all when looking at the droid from eye height.



Assembly

I used 20mm hollow linear shafts, which conveniently are just large enough inside for a ½” threaded rod to pass through them snugly, which makes a nice cheap way to secure them to the bottom and top of the leg lift structure. These rods are 12” long, precut, available from VXB. VXB also sells the linear ball bearing I used, which are flanged with 4 m5 holes, which I use to bolt them to the leg mounting plate/carriage. The flanged bearing are something like1.65” high, which added to the 3 rubber washers I use to adjust for proper bottom stop, and to cushion it’s lower position, Means that I end up with about 10” of possible travel in the mechanism. Since I only can use 8” due to the actuator, you can see it stops 2” from the top. Across the center of the mounting plate, there is a piece of angle aluminum that acts as a brace. This vastly increases the stiffness and strength of the leg plate, making the mounting more than able to stand up to the rigors of supporting the center foot ☺. In the down position, there is no additional locking mechanism, the actuator serves as a pillar, transmitting the forces exerted on the center leg, to the top plate via a beefy cross bar that mounts at the top. Depending on experience, a locking mechanism may be a later addition, but in the very small tests so far in my garage, it has performed adequately. I chose it based on an estimate of the capacity needed.

Here you can see the center leg actuator top to bottom. (ignore the pittman, I don't use that anymore and have in fact sold it, I use just pololu motors for the dome drive now). On the production version the top plate is more solid, more beefed up, and the top cross brace isn't angle aluminum, but instead C-Channel, so the whole thing is even tougher. Also it uses the regular mounting bracket at the top, rather than the custom one I used here. The bottom mount point you can probably see connects to a plate which hangs down behind the center ankle, and is hidden by it from the front. This way the actuator actually can extend somewhat outside the body of the droid in the down position, and that's how I get it to fit. It is connected to the center leg mounting plate/carriage by an large angle bracket and screws.

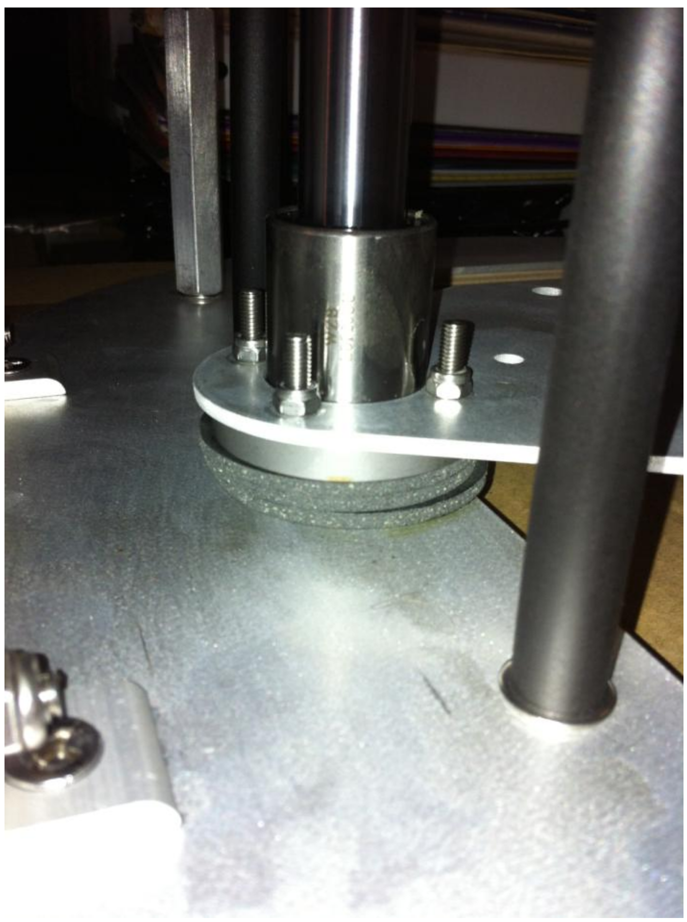


Loads and capacities

Acting as a pillar how it does, the center leg actuator must support the entire weight of the droid and more during stopping when the weight shifts forward onto that foot. Also, whenever it hits a bump, that actuator is going to take the shock. Clearly, it needs to be up to it. I found various sources stating that the shock load on a structure should be rated for 3x or 5x the weight of the robot, and lacking any formal mechanical engineering education myself, I went with the larger of those two, trying to be cautious. While I was unable to find formally stated static load ratings for the center leg actuator, its construction is identical nearly to the ones on the outer legs and those say 500lbs. Again lacking the will to do destructive testing, I went with my gut and figured it would work.

The 20mm rods are steel, so I had little worry about them, and as for the linear bearings, they would not take weight of the droid, or even much side load really, so I more or less just got some that looked reasonably beefy.

Here's a quick view of how the bearing, rod, and center leg mounting plate connect on each side, note the three rubber washers at the bottom to cushion it, and provide proper spacing:



Actuator

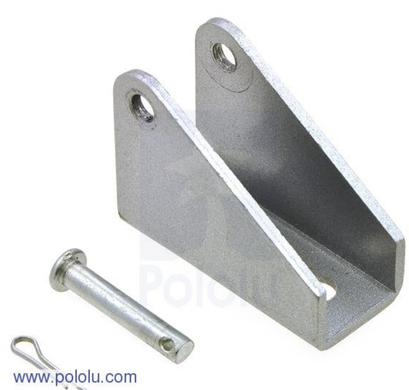
As mentioned before, I estimated that for a 100lb droid, I would want an actuator capable of withstanding 500 lbs or so of load. This is just an estimate really, but so far the one chosen seems OK.

http://www.pololu.com/catalog/product/2348

It seemed comparable to the name brand ones I saw that had a 500lb static load rating, identical in both construction and capability, so I figured well, if it does somehow break, it's as cheap as possible to replace. Now honestly how did I justify that? Well 5g is about how much an average person can handle before losing consciousness, so 5x the force of gravity. So 100 lbs becomes 500 lbs. That seemed good enough to me ;-). Seriously, I'm not trained as a mechanical engineer, I have just my High school physics to work with, so I just made what I figured were reasonable estimates. Yup….anyways!

This actuator is cheap, quiet, smooth, reasonably fast, and at least so far has held up to light driving. Seems OK for now, and if not its really cheap, and easy to replace, so not much loss if it fails, except for the embarrassment of a possible face plant ☺

It connects to the frame at the top and bottom with the pololu-recommended brackets. (actually on my prototype, the top uses a custom bracket, but the production frame uses the standard brackets)

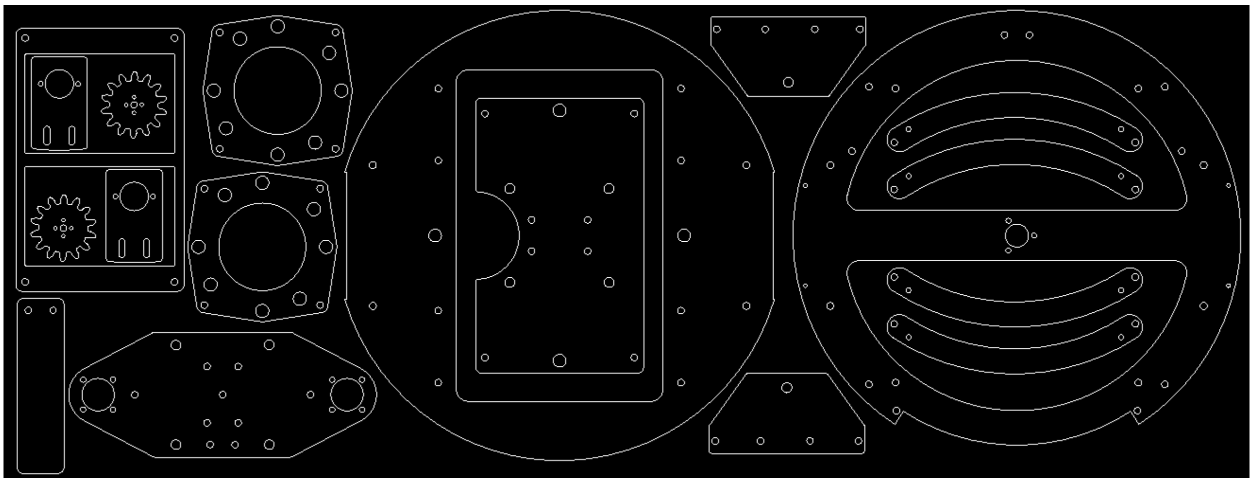


http://www.pololu.com/catalog/product/2355

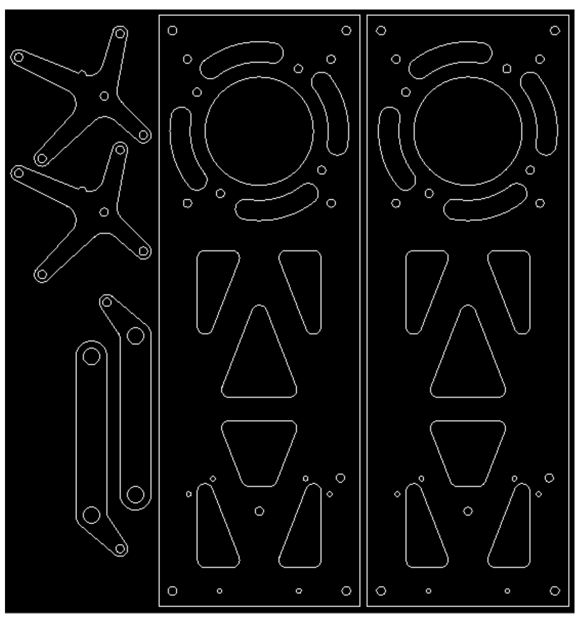
Problem #3 – Frame

Well, I made a completely custom frame! I didn't have to per se, I could have modified a com8-B perhaps, and still hit my weight goals, but I relished the challenge, and rolling my own from scratch allowed me to entirely do it my own way.

These bits were 1/8" 6061 aluminum:



And these bits were ¼" 6061 aluminum:

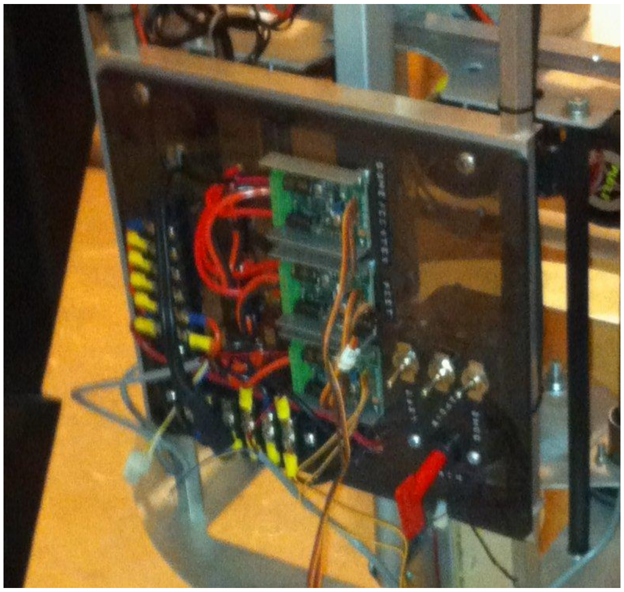


That picture above was the initial design. There are a few extra parts on there; I made mounting plates and gears for a pittman dome motor, shortly before they went out of stock. I ended up selling those or just having them hang about. I made a set of center foot plates, and some shoulder locking plates and a battery tray, more or less just in case this whole 2-3-2 thing didn't work I could reconfigure the frame to use for a regular 3-legged droid (I even planned ahead for failure :-P). The levers for the shoulder of course subsequently changed as discussed earlier. Many of the holes get countersunk, but otherwise other than water jetting those parts pictured from the applicable thicknesses of aluminum, there is not additional machining required. The leg plates attach via angle brackets, the utility arm holder via spacers and long bolts, and the whole frame has long riser rods, made from long aluminum standoffs connected via short threaded rods. For production we will make precisely cut and threaded rods to length on the lathe, but for the prototype it's all just stuff from McMaster-Carr!

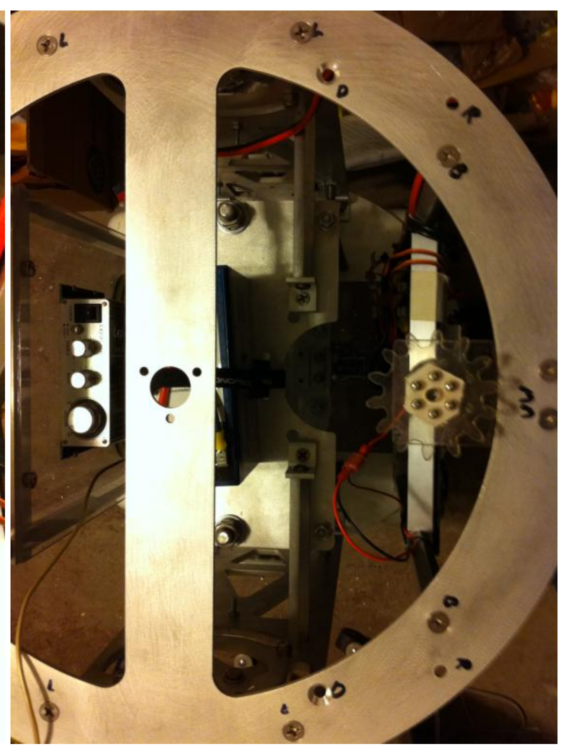
Here is how it ended up going together:

The frame is made to club spec, or R1.0, whatever you would like to call it now ☺. So the 19.350" total height is a little bit of an odd one. I have 1/8" thick top and bottom plates, so the riser rods needed to be 19.15" long. Well I made them 19" long, and then later added a pair of 1/16" thick braces to each set. Initially I just used washers, and that got me close enough.

The braces added stiffness, and also gave me a great place to mount plastic plates which I could mount my switches, electronics, speakers, amp, and etc to.




Eventually, I also tied the side plates into the plate at the top of the center leg lift, again to add stiffness.

For production I've modified the top plate of the center leg lift to connect itself, so it doesn't require the standoff there. After that last modification, all perceivable twist or play or flex was eliminated from the frame. Everything is mutually supported, so nothing flexes relative to each other. With the thin plates, there was bound to be some flex, but with some bracing and a few simple modifications, I got rid of it ☺

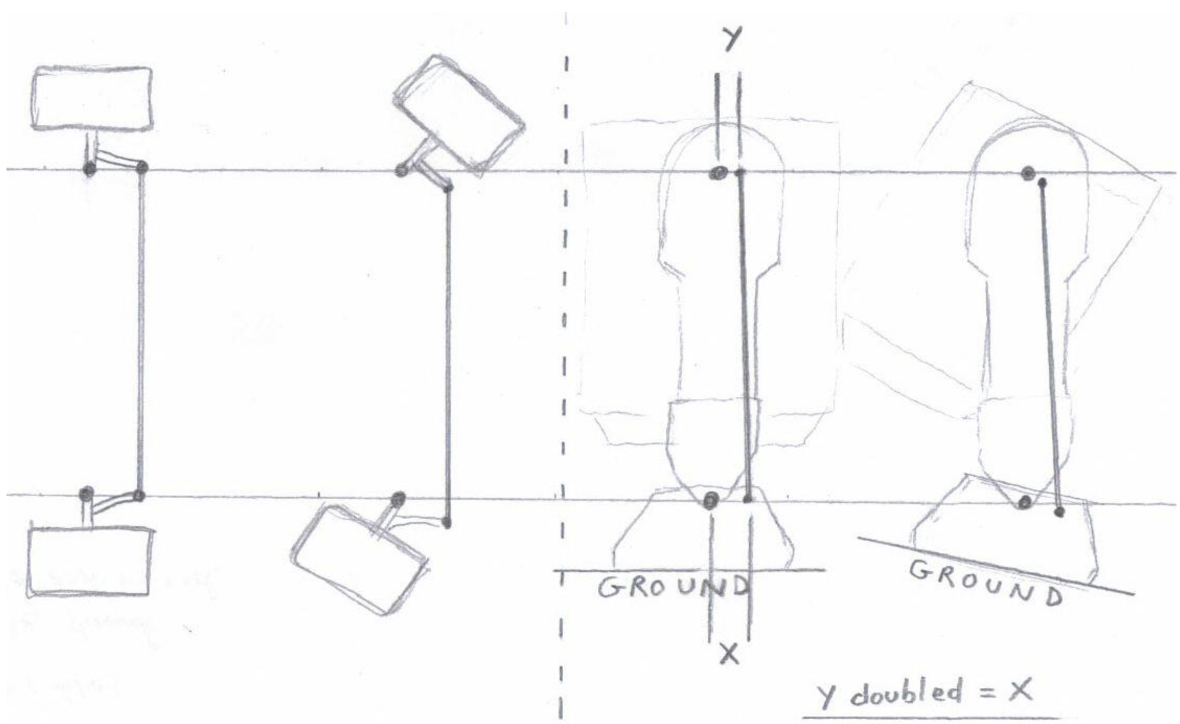
As you can see I'm sure, I borrowed a little from common frame designs. JAG's PDFs of the JAG v5 were very helpful to me in establishing basic things, like the vertical placement of the shoulder slots, the basic pattern for them to be compatible with regular legs, and the shape of the top and bottom plates are based loosely on the JAG V5. The utility arms carriage is pretty unique, I haven't really seen it don't this way before, but it's sort of an obvious method, so it likely has just not in a log I read or some such.

In a nutshell, I went and made a pretty unique, super lightweight frame to suit my 2-3-2 idea. I tried to build it in such a way that it could be reproduced, and even run, so hopefully that will happen.

Problem #4 – Leg rods and Ankles

So the last problem set in making 2-3-2 work, Is how to control the ankles, so that in 2 leg mode your droid doesn't fall over, and in 3 leg mode they are able to move to the right angle relative to the body. There is lots of ways to accomplish this; separate actuator inside the legs, many variations on ankle locking mechanisms, and the simplest perhaps, is leg rods. I went with leg rods, for their simplicity, durability, and the fact that they lock the ankles in 3 leg mode as well, which I find desirable for driving.

I don't remember the original source, but this diagram explains the general concept:



So by placing the top of the rod, one distance from center at the shoulder, and the bottom twice that distance from center at the ankle, the ankles would always be at half the angle of the body. The body is 36 deg per spec, and the ankle 18, so it works out. Given the width of a leg drawn up in CAD, the widest distance I could fit in round numbers in a pair of ATL wood legs, was .75” at the shoulder, and 1.5” at the ankle. I wanted the widest distance I could fit, since a wider distance, meant the actual pivot points could be somewhat less accurate to achieve minimal play in the joints.

At each end the rod is secured with a grade 8 bolt, at the top it goes into a spacer connecting to a special plate bolted to the frame inside the leg to body hub, meant to simply provide this mounting point. At the bottom it bolts to conveniently available holes included in my ATL foot drives ☺ The rod itself is just a length of ¼-28 threaded rod, and two steel ball ends. This way it is easy to adjust for proper fit and angle.

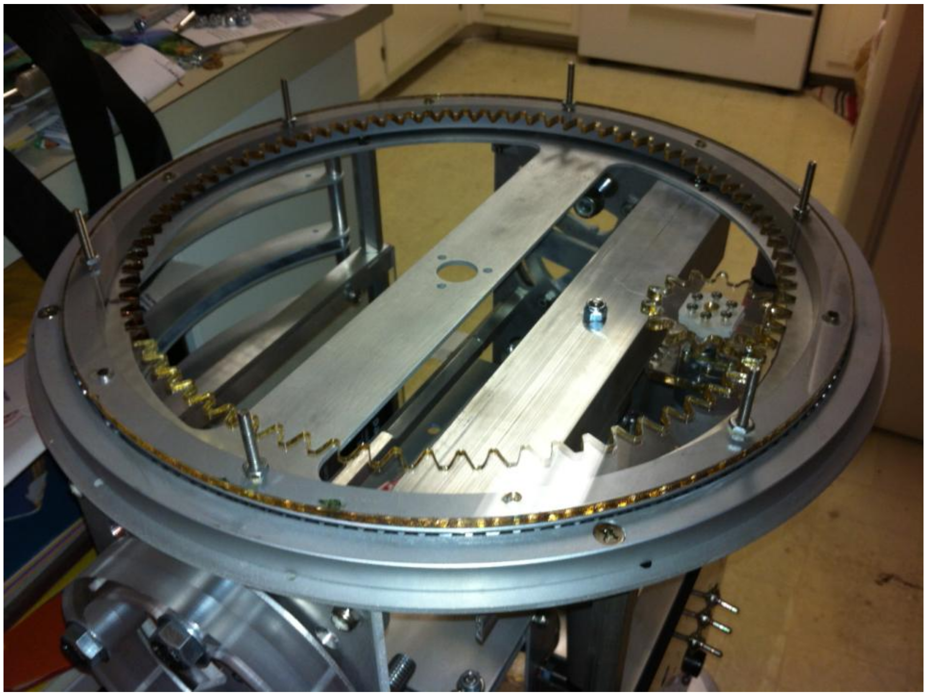
That’s pretty much my whole story on the 2-3-2, hope it was an informative and fun read! What follows will be more pictures, with some descriptions ☺

The “spider” inside the leg to body hub, which provides the mounting point t the shoulder for the leg rods.

New shoulder lever plate installed, with wired to foot drive passing through the middle. That’s what the holes are for ☺

Here’s R4, skinless, but operable ☺

Skin snaps and aluminum skins are a good fit for this frame. The production version has pre-drilled mounting holes for darren's skin snaps. Aluminum skins are good because the frame provides minimal support to the skins to save weight, so it needs the skins to hold their shape on their own.



I used a custom gear drive for the dome on this droid. An aluminum version may be offered as an option if we run the frame.

Servo operated doors and utility arms are definitely possible with the frame. My utility arms are open-able, but I haven't installed servos yet.