

# Build a High-Performance Inverted Pendulum Robot



## Supplies:

Most of the parts are available and some can probably be found locally. The only components that have only one source are the motors from Servo City and the motor controllers from Pololu.

- Printed Circuit Board (PCB). This must be ordered from a manufacturer. Step 1 explains how to do this. For people with the necessary skills, it is entirely possible to build SixPotatoe using a breadboard or prototyping board.
- [Motor Controllers](#) from Pololu. I use the 24v13 controllers but any of the single channel High Power controllers in this category should work.
- [5V Regulator](#) from Pololu.
- [Planetary gear motors](#) from ServoCity. SixPotatoe is pre-configured to handle five of these motors. These motors are described at the end of this section.

- Encoder cables. If you are using the Yellow Jacket motors, you will need [these cables](#) (2). The HD motors already have the cables attached
- [6mm Clamping D hubs](#) (2) from ServoCity.
- 6-32, 1/2" screws (8) to attach the clamping hubs to the base plate.
- [12mm Hex Wheel Mount](#) (2) from ServoCity.
- [32mm Bore Bottom Tapped Clamping Mount](#)(4) from ServoCity.
- LIPO battery. I use a [Turnigy 1200 mAh battery](#). The battery should have XT60 connectors. Almost any six cell batteries will work. This battery will run for more than an hour before needing to be recharged. Larger batteries give more operating time but also give more weight at the top which helps climbing steep slopes and over rough ground. However, small batteries can give the same performance improvement if mounted high enough.
- [XT60 connectors](#). You will need one male connector to mount on the PCB. You will probably need a male and female connector to make a small battery cable if your battery does not have a long enough cable.
- 16 ga. black and red battery wire. Almost any stranded 14-18 ga. wire will work although I have a strong preference for silicon wire which is available from [Hobby King](#) or [Amazon](#).
- Teensy 4.0 Arduino controller. These are available from a variety of sources including [Sparkfun](#) and [PJRC](#).
- Electrolytic capacitors for the motor controllers (2). I use [330uf from Pololu](#). Pololu recommend larger capacitors. A higher voltage would also be a good idea.
- [6 channel RC receiver and transmitter](#). This is an inexpensive but very capable controller. This Instructable is designed around this controller although other controllers would probably work with a little modification.
- [IMU](#) from Sparkfun. This is an ICM-20948 mounted on a breakout with a Qwiic connector. The Qwiic connector can be used to add other sensors at a later point. You may want to order a short Qwiic cable from Sparkfun to connect the IMU, although this can be easily soldered instead.
- [HDPE sheets](#). Other material can probably be used such as [Phenolic sheets](#), aluminum or even baltic birch plywood. However, the frame needs to be very stiff, especially the base plate! I normally use 1/4" thick HDPE for all the plastic parts although I often use 3/8" for the bumpers. For the bumpers, HPDE is probably the only suitable material

because of its extreme durability. I buy my HPDE from the scrap bin of a local plastics fabricator, but it is widely available from other sources.

- Threaded rod to hold the platforms. I use [M5 rods from Amazon](#) but other sizes work just as well including 6mm, 10-32, 12-24, and 1/4". You will need four lengths that will normally end up with a length of 120 to 160mm. The ones listed from Amazon are 130mm which quite suitable.
- Wheels & Tires. I use [Pro-Line F/R Masher 2.8" A/T Tires](#), but many other tires will work just fine. The wheel must have 12mm hex mounts, which is standard for 1/10 scale vehicles. Also, the wheel should have a "wide offset". Any wheel that fits a 2wd Stampede®/Rustler® Front or 4x4 Stampede®/Rustler® Front and Rear should work. Even wheels with a narrow offset can be made to work by cutting away a bit of the base plate for tire clearance. The softer (more grippy) the rubber, the better. Also, the bigger the tire with more cushioning, the better. On the other hand, if you just plan on running on flat surfaces --- even grassy areas --- wheels like [those from Bainbot](#) work nicely.
- LED momentary contact switch. The PCB is designed to use this [switch from Sparkfun](#). You may find it more convenient to mount a switch on the top plate. There are connection points on the PCB that makes it easy to hook up a cable to do this. For this, I like the LED [switches from Adafruit](#).
- Hardware for mounting the PCBs. There are a number of ways to do this and all of them work just fine. You should choose hardware based on what currently have or what is available locally. Read the description in *Step 8: Mounting the PCBs* to understand the options.
- Mounting hardware for the PCBs. You should read the instructions in Step 8: Mounting the PCBs before deciding what parts to use.
- [Male Header](#)
- [Female headers](#) (2).
- [Schottky Diode](#) (2).
- [Ceramic Capacitor 0.1uF](#)
- [Electrolytic Capacitor 100uF](#)
- [Qwiic cable](#).
- Seven resistors: One each of 150 ohm, 470k ohm and 68k ohm and four 1k ohm resistors.

If you can't find some of the miscellaneous electronic components locally or in your parts bin, they can be purchased from either [Digikey.com](https://www.digikey.com) or [Mouser.com](https://www.mouser.com). Many of the small mechanical parts can be purchased at your local hardware store. If they don't have exactly the part specified, you can usually find something similar there that will work just fine.

The total price for these components should be \$400 - \$500.

Before starting this project, you will probably want download and print out the mechanical drawings in the repository. These can be found in <https://github.com/robertstuart/SixPotatoe-Instructable>, in the "Mechanical Drawings" directory.

## Tools

- Table or band saw to cut the platforms.
- Drill. A drill press is nice but not absolutely necessary.
- Saw to cut the bumpers. The bumpers could be omitted but this is not recommended. A jig saw, coping saw or even a router could be used to fashion the bumpers. I expect that those with good 3D printer skills could make something that would work quite well.
- Soldering iron. A temperature-controlled iron is recommended.
- A vice of some type is needed to hold the parts while drilling or soldering.
- A modest assortment of screwdrivers, wrenches, and pliers

## Choosing a motor

The robot can use any of the following motors, available from Servo City. I have tried all of these motors and I have a slight preference for the HD 612 RPM but there are reasons, including availability, that you might want to choose another one. Here is the complete list:

**Yellow Jacket 435 RPM** - This motor is a little cheaper, is a little less powerful than the equivalent HD motor listed next. The encoders don't have as much precision. Still, the differences would be hard to detect in most situations. Its top safe speed is around 15 KPH (9 MPH).

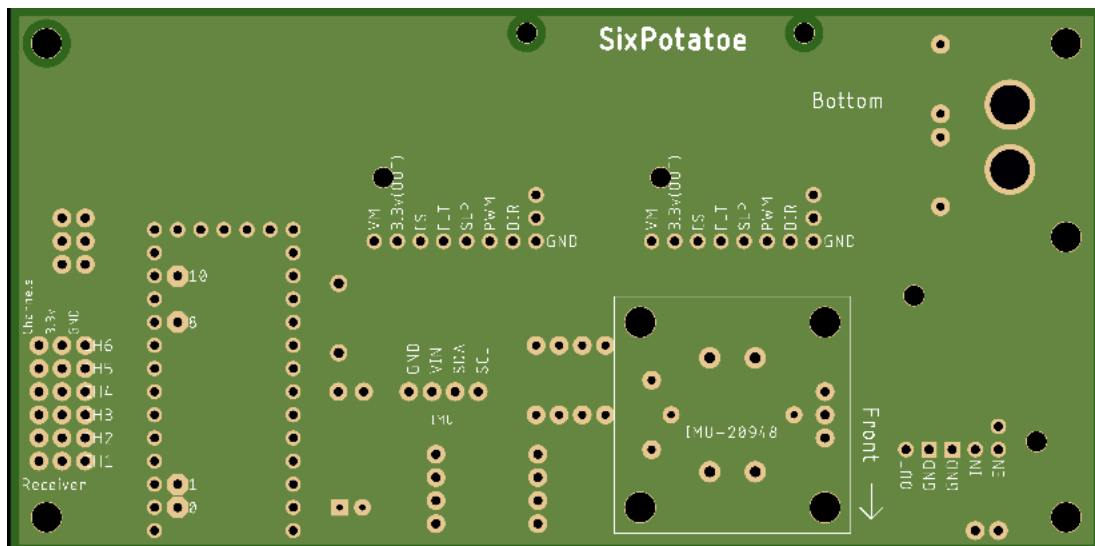
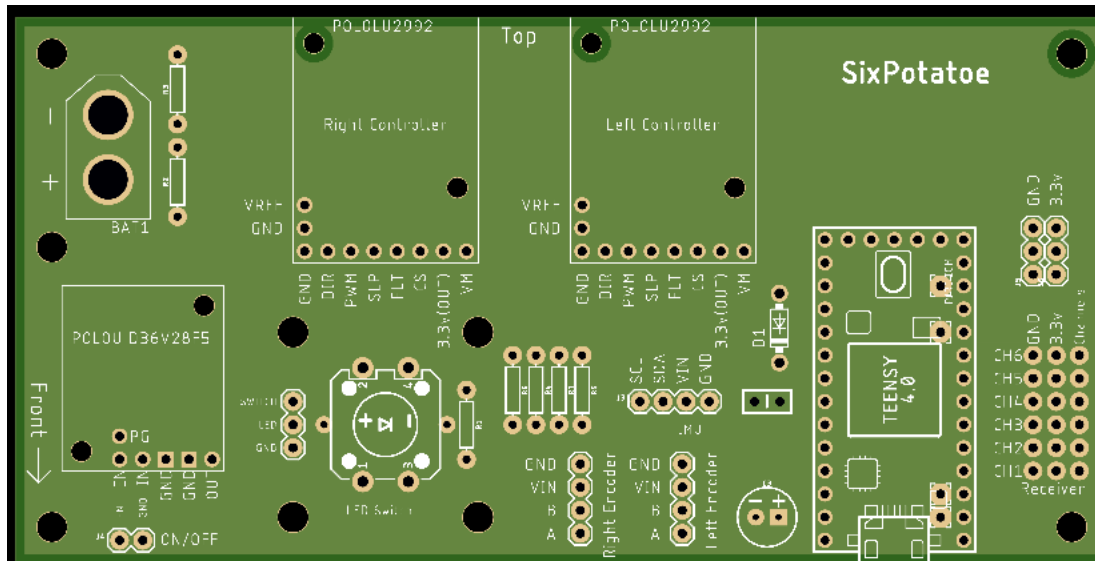
**HD 437 RPM** - For all practical purposes, this motor is equivalent to the Yellow Jacket 435 motor described above. It does have a bit more power and better encoder precision, so in rare instances it will perform a little better. In fact, SixPotatoe used this motor in all of the skate park “dimple” scenes in the YouTube videos.

**HD 612 RPM** - This is the ideal motor for many people. In fact, I delayed the publication of this Instructable until this motor became available again. Its top speed is around 21 KPH (14 MPH) and it manages jumps and rough ground with ease and is plenty fast. It is very stable and runs a long time on one battery charge.

**Yellow Jacket 1150 RPM** - This motor is just like the Yellow Jacket 435 RPM motor but is almost three times as fast. It won't stay upright quite as well over rough ground and jumps, but the speed is very impressive. It is a lot of fun to drive. Its top safe speed is around 40 KPH (25 MPH). In my opinion, this is really faster than you will normally want to go. It can do considerable damage to itself, other things and people when it hits something at that speed. It will appear to be slightly unstable and out-of-control at its top speeds since it will bounce a little bit when encountering slight undulations in the surface. It consumes considerable battery power; especially over rough terrain and the motors will get slightly warm. SixPotatoe was using this motor for most of the high-speed scenes in the YouTube video.

**HD 1621 RPM** - If you are addicted to speed and your personal and property insurance payments are up to date, this is the motor for you. Its top speed is about 55 kph (36 mph). In the skate park, I have made some really spectacular jumps using this motor. (I have been unsuccessful in making it land properly in many cases, so there is often damage. As a result, I have abandoned any further attempts.) Its operation over rough ground is not up to that of the lower RPM motors, but it is still very impressive. The safety, battery power and motor temperature caveats of the previous motor apply even more to this motor. Also, I have not done as much testing with this motor so it may not be as well tuned as it could be.

## **Step 1: Create the PCB**



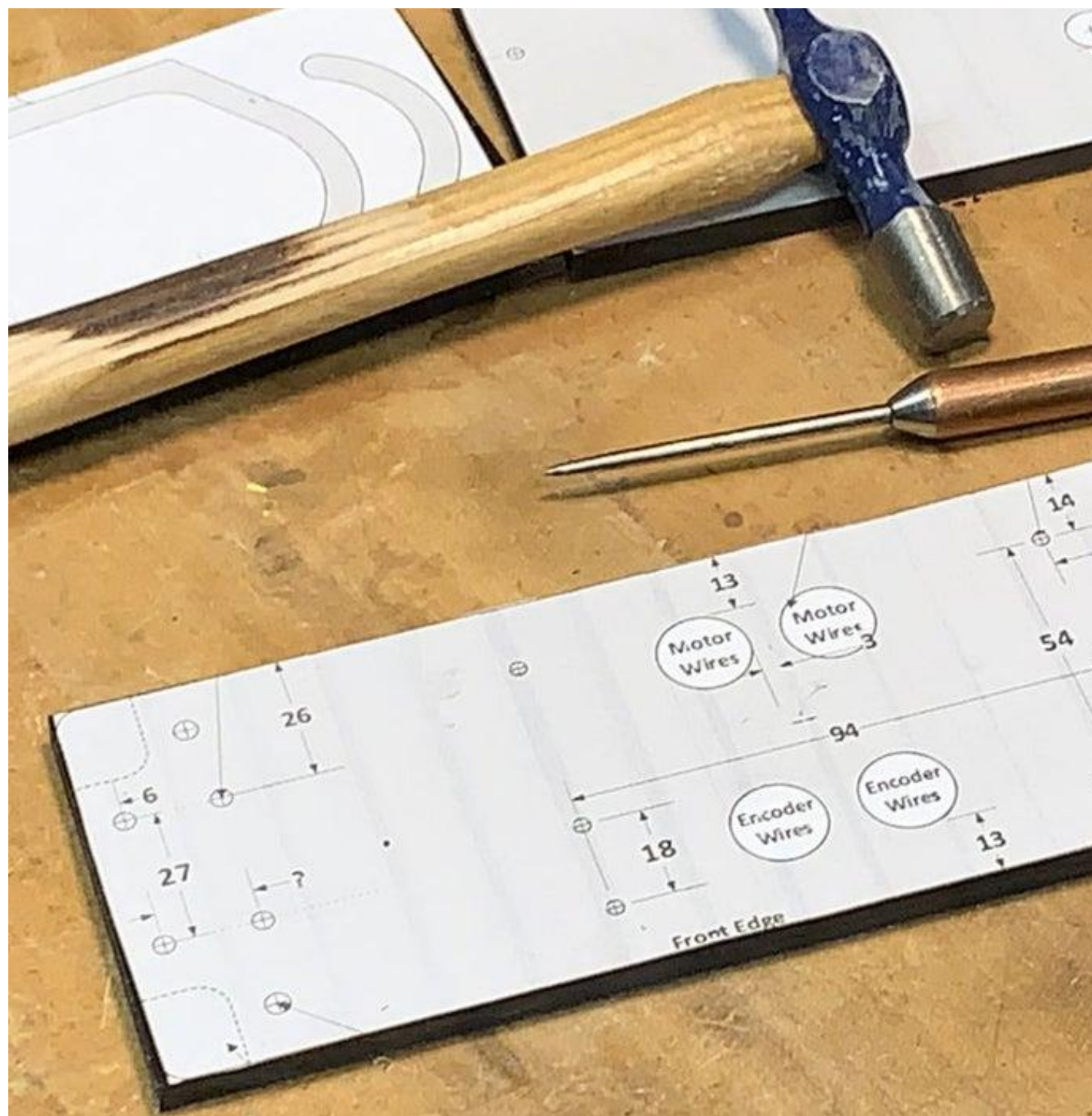
The Gerber file for manufacturing the PCB is included in the *Eagle Files* directory of the [GitHub Repository](#). There are a variety of PCB manufacturers but we have been very happy with the price, quality, and delivery times of [JLCPCB](#). Whatever manufacturer you use, follow the directions from the manufacturer for uploading the Gerber file and ordering your board(s).

Note that it is possible to make this robot using prototyping boards or even breadboards. In fact, early versions of this robot were made using both of these methods. The schematic should give you all of the information you need. The wiring diagram, *Wiring.pdf*, also available from the

GitHub Repository, should also help but be aware that it is doesn't exactly reflect the schematic. Use the schematic and just use the wiring diagram as a general guide!

## **Step 2: Top and Bottom Plates**











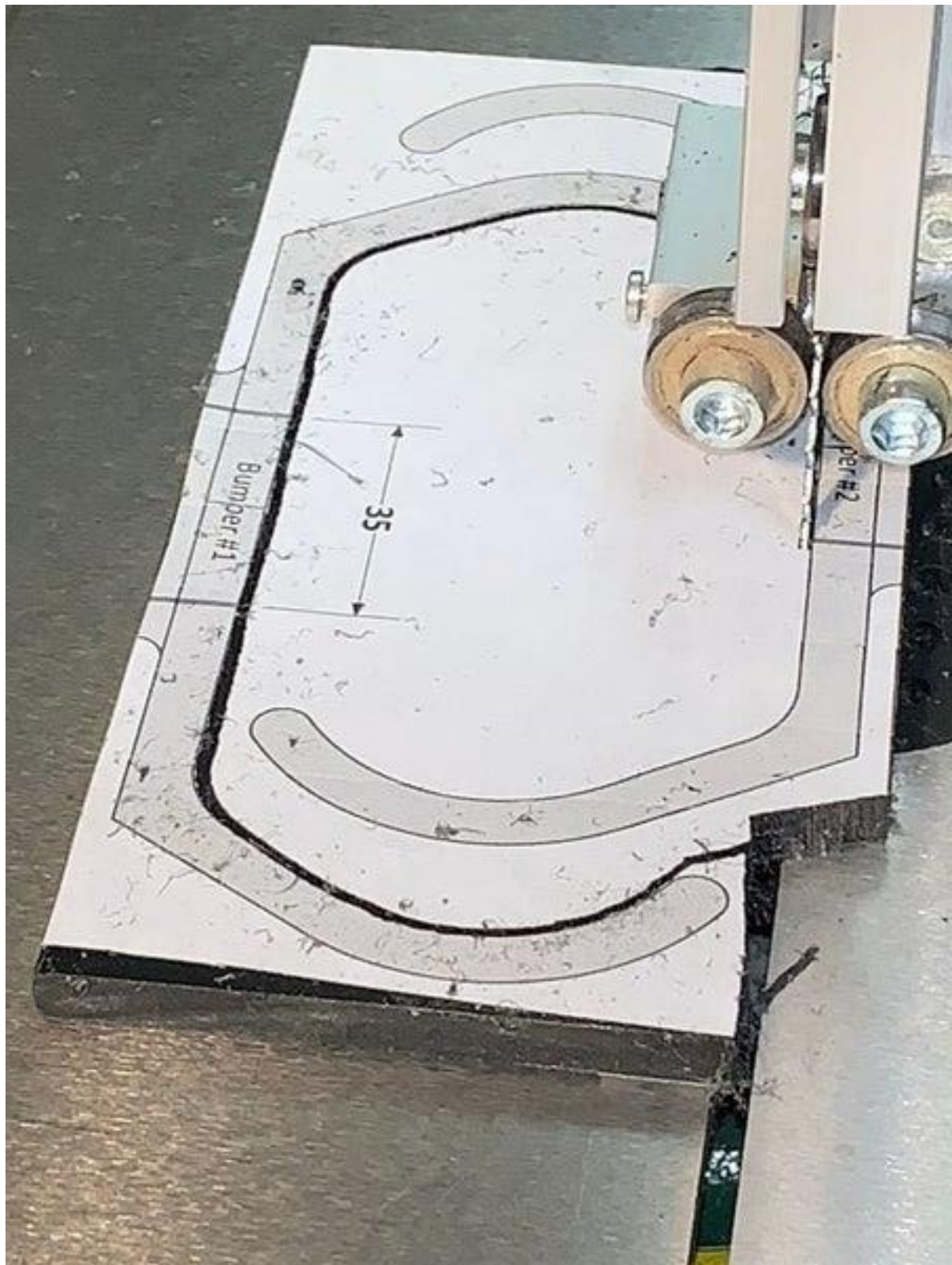
Cut the two HDPE plates to the dimensions shown in the mechanical drawing. Ideally, this can be done on a table saw but a band saw works also. It is a good idea to round the four corners on all the edges. This can be done with a sander. A wood plane also works quite well. The edges can also be carved smooth with a regular knife or an Exacto knife.

To drill the holes, it is usually easiest to make a template by printing out the mechanical drawing. Be sure to print them out using a scale of 100%! Then tape or glue the drawing to the platform. Gluing with rubber cement usually works best. You can then punch the center of each hole with an awl or center punch to give a solid target for the drill so that the bit doesn't wander. The holes can be done with a hand drill although a drill press is a little easier.

Only the motor mounts require any accuracy. To get the spacing between the two motor mounts, install the clamps on your motors and measure the spacing. Normally, you will want to have the gear box protrude a bit from the outside motor mount. I usually adjust the position so that the heads of the screws holding the gearbox end cap are just outside the clamp. The picture shows this relationship.

On the top plate, do not drill the holes for the bumpers. These can be more accurately drilled when the bumpers are completed in the next step.

### Step 3: Bumpers





Again, the easiest way to create the bumpers is to use the printed diagram as a template. There are two base lines that indicate the bottoms of the bumpers where they will be attached to the top plate. Measure the distance between these two lines. Cut out a plate off HDPE so that it has exactly this width. Cut out the printout along these base lines and then glue the template to the plastic, aligning the baselines with the edges. The bumpers can be cut out on a band saw. A simple coping saw also works quite well. After cutting, the edges can be smoothed by carving with a knife or sanding using a medium grade sandpaper.

1/4" HDPE works well for the bumpers but I have a slight preference for 3/8" HDPE since it is a little stronger. The bumpers are the most likely part to get damaged in a driving accident, so you might consider making a couple more at this point. If you are a careful driver, you can just make the required two.

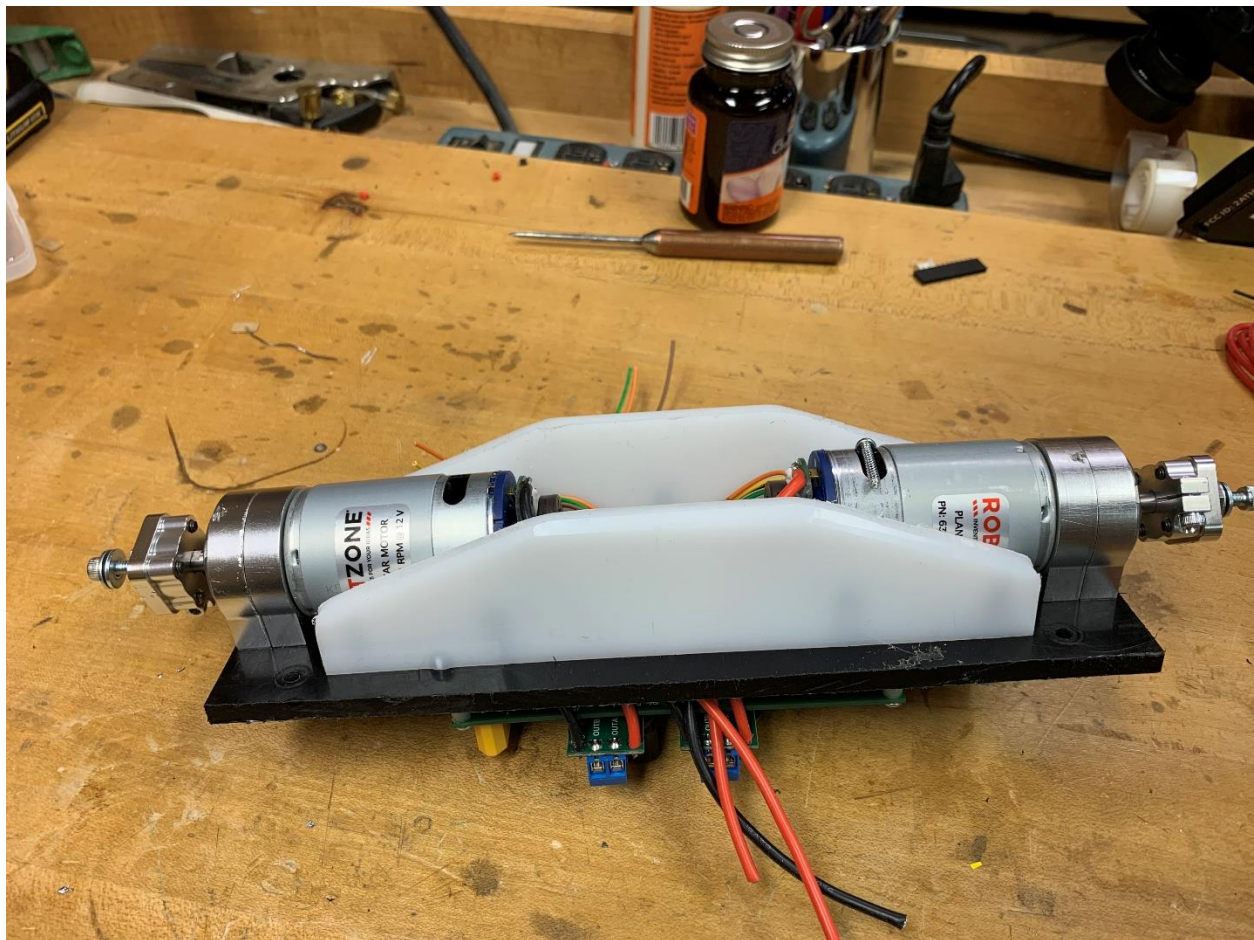
Once the bumpers are cut, you should drill the two mounting holes. This is something you should do very carefully. The bumpers are most likely to break at these mounting holes. They

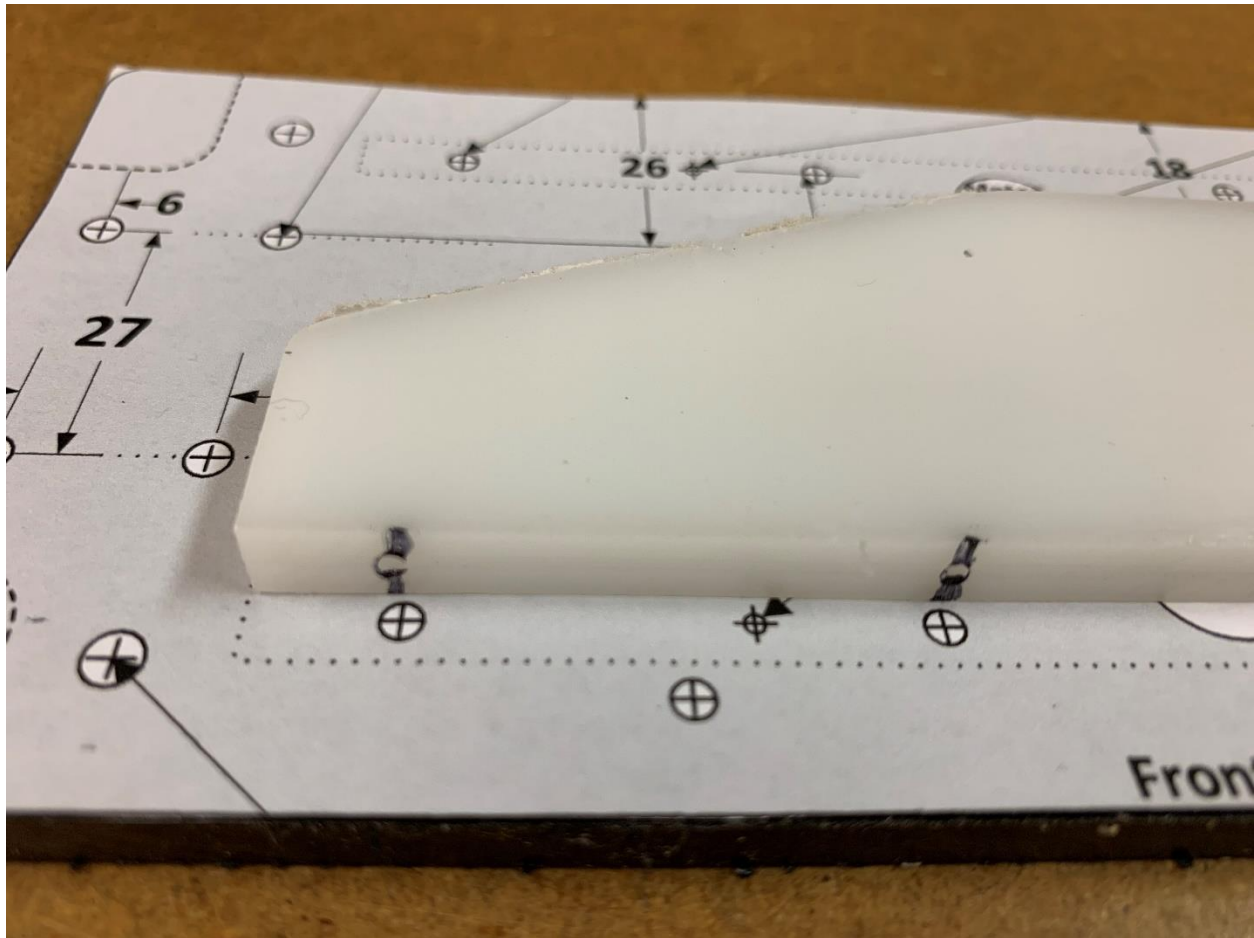


will be strongest if you drill exactly in the center. This hole should be quite small. I normally use M2.5 or 4/40 screws. The holes for the screws should be slightly smaller than the screw. This way you will be able to drive the screw through the hole with a hammer so that the sides of the plastic will bulge out slightly.

After you drill the holes but before you drive in the screws, you should clamp the bumpers to the top plate in their correct position. Using the same drill bit, you can use the bumpers as a drill guide to drill through the top plate. When you un-clamp the pieces, you can then re-drill the hole through the plate so that it has better clearance. When you finally mount the clamp on the top plate, it is a good idea to put a washer underneath the head of the screw. Also, since it is normal to carry SixPotatoe using these bumpers as a handle, it is a good idea to grind off any part of the screw that protrudes from the nut it so that it is comfortable to carry in this manner.

#### Step 4: Encoder Shields





The encoder shields protect the encoders from damage, which can easily happen over rocky ground. They also add stiffness to the entire base plate.

There are five clearance holes in the base plate for each of the two encoder shields. You will mount the encoder shields to the base plate with self-tapping screws. It is advisable to drill small tap holes into the encoder shield to guide the self-tapping screws. Mark the position of these screw holes using the holes in the base plate as a guide. You can see the relationship in the photo. As with the bumpers, it is best to have the holes exactly in the center. I usually drill the holes with a #52 bit but a slightly larger bit would make it easier to drive in the screws. It can be difficult to drill holes that are exactly vertical. The ideal tool to do this is a drill press with a clamp to hold the shield in place but most people should be able to do it with a hand drill and the shield held in place with a vice. In any case, for the shields, accuracy is not important.



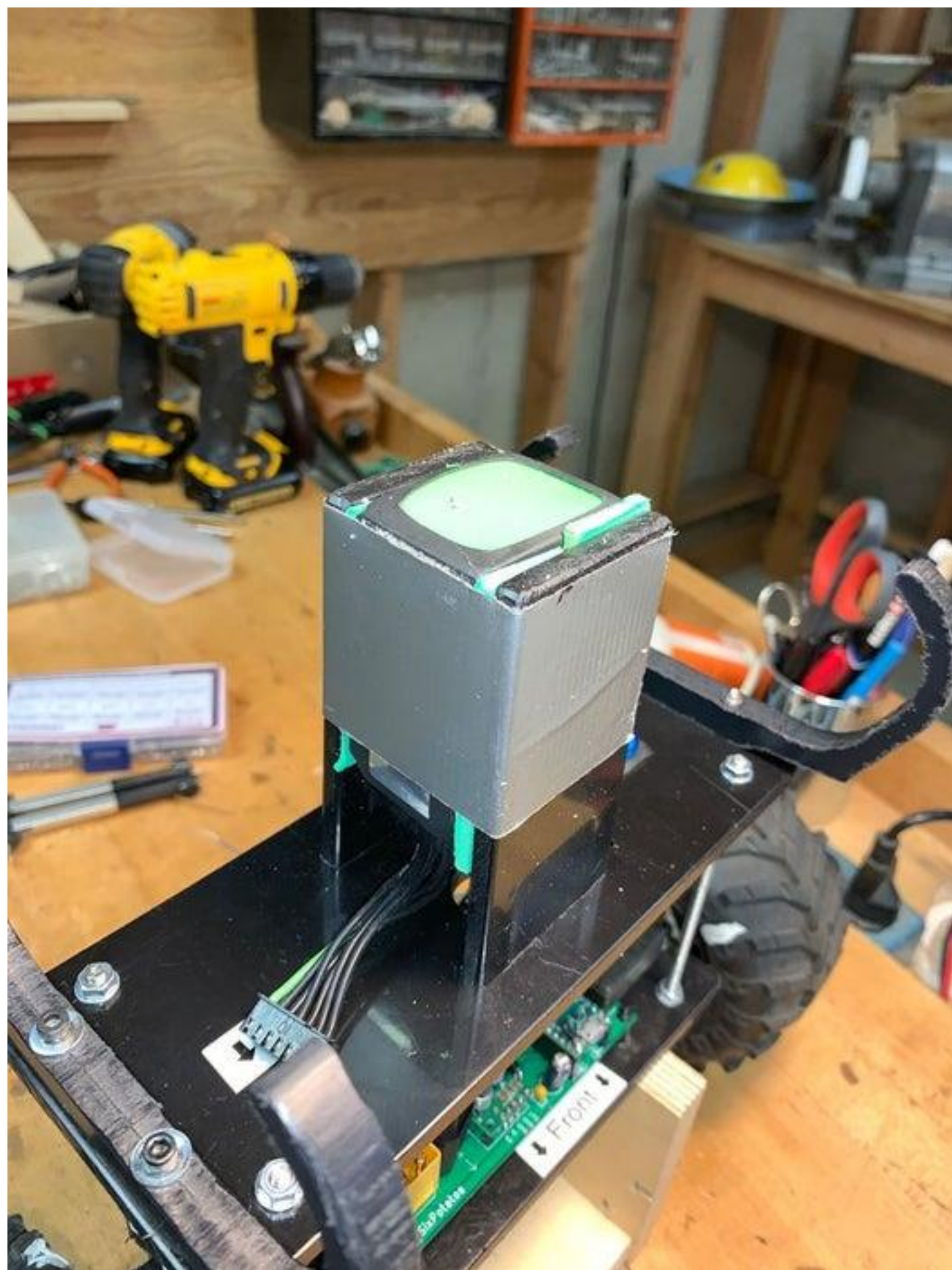
If you do the cutouts as shown in the drawing, this can be done after the tap holes have been drilled. For mounting the shields, I usually use M3 self-tapping screws of about 20mm in length. 4-40  $\frac{3}{4}$ " sheet metal screws are roughly equivalent.

This technique for mounting the shield will also be used in the next step to create the battery tower.

## **Step 5: Battery Tower**







The battery tower is not absolutely required. In fact, all batteries, especially larger batteries, can just be strapped to the top plate so that the battery tower is not needed at all. The battery tower, described here, helps a little bit with performance on steep slopes and rough ground, so it is the preferred method. If you decide to mount the battery flat on the top plate, you might consider lengthening the threaded rods to regain this lost performance. Another consideration is that the battery tower makes it easier to mount a head on the robot.

The battery tower must be custom fit to battery that you have. If the power wires on the battery are long enough, you may be able to plug directly into the PCB. For many batteries, however, you will have to manufacture a short extension so that the battery can be mounted at the proper height. For the specified battery, a height of around 90 mm seems about right. The width should be as wide as the battery.

Once the size is determined and the pieces are cut out, you may have to create some spacers to go between the battery and tower sides. You can see this in one of the pictures above. This is because batteries are often a little wider at the end with the connectors. The spacers prevent any pressure on the connector wiring. Almost any material will work for spacers. Often cardboard is the easiest option.

Sandwich the tower pieces, spacers and battery together placing glue between each of the pieces. Rubber cement or contact cement works best in this situation. Double-sided sticky tape works well also. Lightly clamp the entire sandwich together and set it upright and adjust everything so that the bottoms rest flat on a horizontal surface and the sides are vertical. Allow the glue to dry. After the glue is reasonably dry, remove the clamps and tape the whole thing together. I like duct tape or electrical tape but almost any tape will work.

You must decide what kind of screw you are going to use to attach the tower to the top plate. M3 or M4 self tapping screws of 20-30mm length work well. In the US, #6 sheet metal screws of about 1" seem about right.

Take the battery tower and position it on the top plate so that it is properly centered. Mark the position of the battery tower supports on the top plate using masking tape. Turn the battery tower upside down and mark the position of the guide holes with an awl. These should be exactly centered on the supports and about 8mm from the end. As with the encoder shields, you

need to drill tap holes into the tower sides and, after marking the position on the top plate, drill clearance holes in the top plate.

If you decide to mount an LED switch on the top plate, this is the time to drill the hole for that.

Mount the battery tower guiding the power cable through the top plate.

## **Step 6: Mount the Motors**

The motors should be bolted to the base plate using 6-32 screws. The screws should be as long as possible without bottoming out in the clamp. There are slightly different directions for the HD and Yellow Jacket motors.

### **Yellow Jacket Motors:**

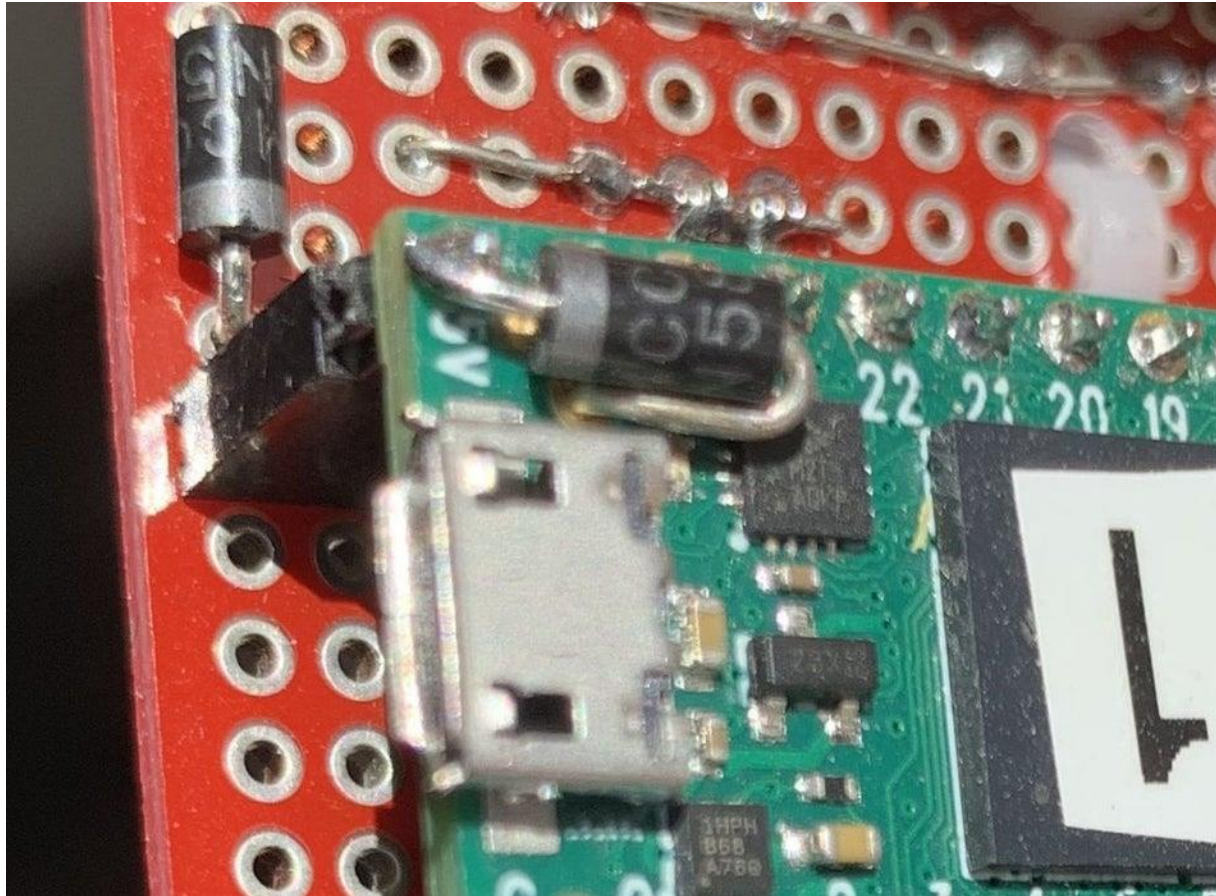
Remove and throw away the plastic shields for the encoders. Wires from the motors need to be kept away from the encoder wires so that spikes from the motor do not interfere with the encoder signals. This interference can cause serious problems in the operation of the robot, particularly during aggressive maneuvers. Completely pull apart the red and black part of the wire pair. Mount the motors so that the encoder wires come out and go directly up to the front of the PCB and the power wires come out the opposite directions and come up the motor controllers

### **HD Motors:**

You should clip off the connector at the end of the six-wire cable. Then you should remove the red and black wires by clipping them off right at the encoder PCB. Cut two lengths of the red 16 Ga. stranded wire to about 130mm. Do the same for two pieces of the black wire. Strip off enough insulation from one end of each of the wires so that it can be soldered to the motor. Solder a red wire to the +M connection and a black wire to the -M connection. Be sure to solder them so that they are leading away from the encoder wires. Mount the motors so that the encoder wires come out and go directly up to the front of the PCB and the power wires come out the opposite directions and come up the motor controllers



## Step 7: Prepare the Teensy

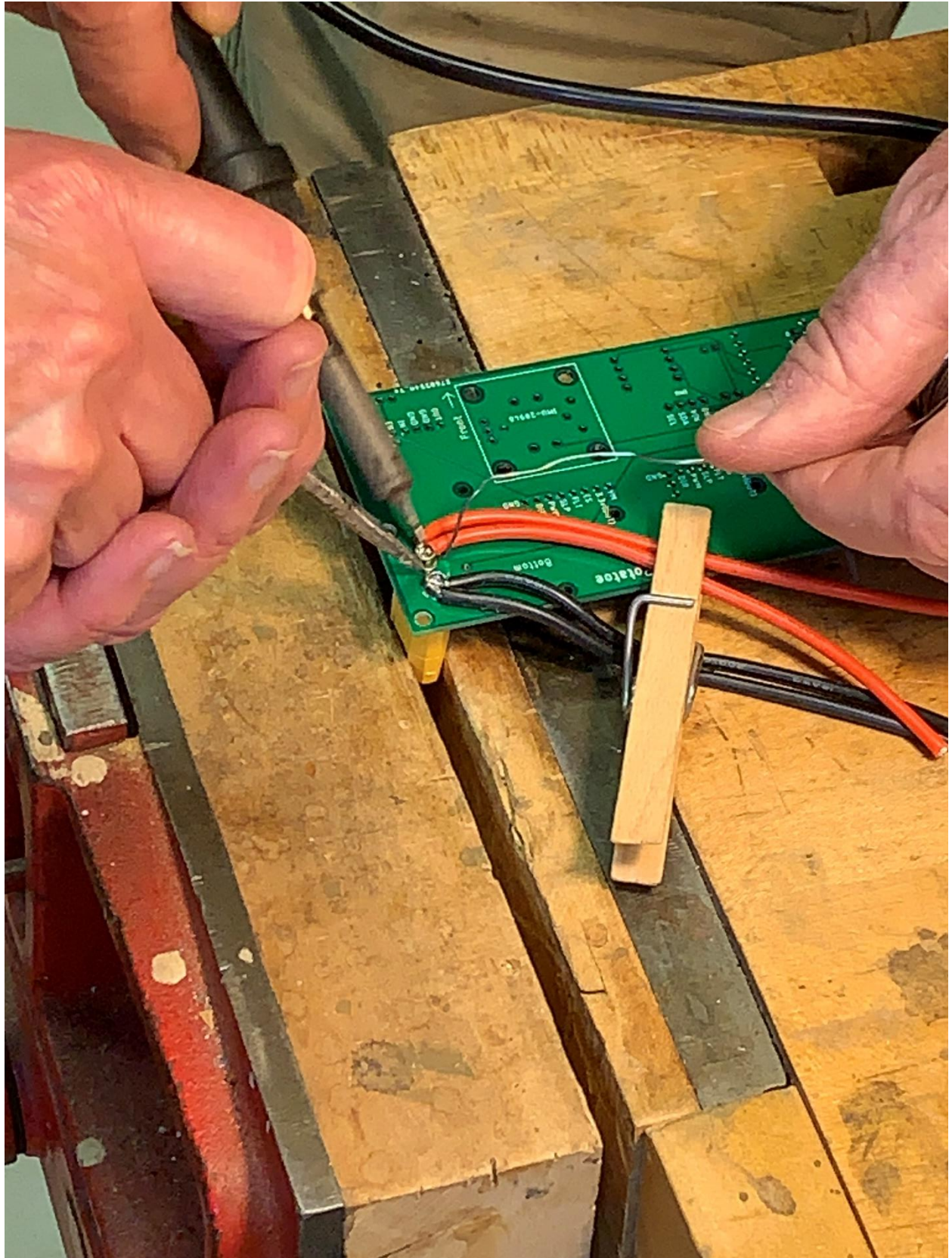


The Teensy 4.0 must be modified so that it can be powered by either the USB or the battery. A trace is cut on the back side of the board and a Schottky diode is added to the front side. The trace is shown on the "Teensy 4.0 Back Side" drawing here: <https://www.pjrc.com/store/teensy40.html>. On the diagram you can see a connection point labeled VUSB. This is the point where the anode side (without the stripe) of the diode must be connected on the top side. After cutting the trace on the back side, place the diode and solder it on the top side so that it connects the VUSB to the 5V (Vin) point as shown in the picture above.

If you have purchased your Teensy without headers, you must solder 14-pin male headers to the bottom side of the board.

## Step 7: Prepare the PCB





By now, your PCBs should have arrived from China.

We will hook up the power connector first since that is easiest and safest to do when there are no other connectors or components on the board. It is best to have one or two people help you with this step since the PCB, the connector, and the wire all need to be held in place when soldering. First of all, cut two pieces of black and two pieces of red 16 gauge wire to a length of about 120mm. Strip about 5mm of insulation from one end of each wire. I also like to insert a female connector (not from a battery!) into the male connector to make it easier to hold on to the connector. Start heating your soldering iron. A good hot iron is needed. I set the temperature to about 400 degrees Celsius

Now comes the hard part. I like to clamp the connector in a vice and then place the board on the clamped connector. This is where you will need someone to hold the board onto the connector. This helper should use something like a screwdriver or an awl to hold the board firmly onto the connector. Then place the two stripped ends of black wires into the negative terminal. and press the wires against the PCB while directing the wires toward where they will eventually connect to the motor controllers. Either have someone else hold the wires against the PCB or use something like a clothes pin to hold the wires.

Stop! Recheck to make sure that: 1) you are mounting the connector on the correct side of the board, 2) the connector + and - are oriented in the correct direction and 3) the black wires are connected to the minus side of connector terminal.

Now you can solder the wires. It will take some time to heat this all up, but once it is hot enough, the solder should flow easily. Apply just enough solder to just fill the connector. Be sure that the solder flows between the connector and the plated through hole on the PCB. Solder the two red wires in the same manner.

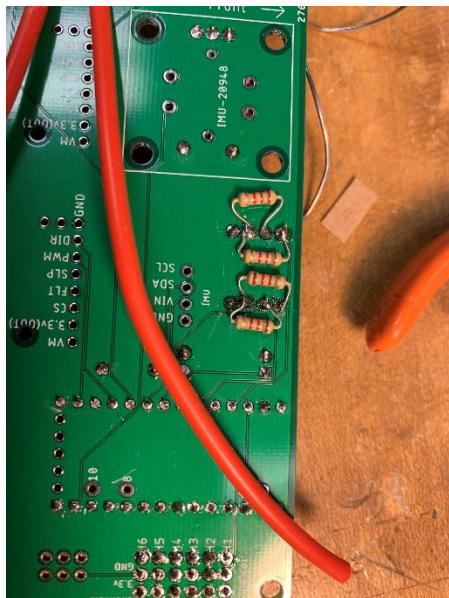
Cut the headers so that you have two 14-pin strips to mount the Teensy and three 6-pin strips to mount the radio receiver. You should cut the strips at a pin, destroying that pin, so that you have the desired number of pins remaining. For the Teensy headers, the ends can be left rough if you want but the three 6-pin strips must be filed off so that they fit into the radio receiver housing. You can do this with a piece of medium sandpaper on a flat surface or with a file. Fingernail files work quite well for this. If you want to put a connector to an LED switch on the top plate, you can

also mount a 3-pin header for this. I also use two 4-pin headers for the encoders although most people will omit this and just solder the encoder wires to the bottom of the PCB.

Solder all the headers to the PCB. This is another situation where it is helpful to have another person hold everything together while doing the soldering.

Solder the resistors, capacitors, and diode to the PCB. The 68k voltage divider resistor is mounted near the top edge by the battery connector and the 470k resistor just below it. The 150-ohm LED resistor is mounted next to the LED switch. The diode, and two capacitors are mounted next to the Teensy as shown on the PCB. Be sure to observe the polarity on the electrolytic capacitor. The four 1k pull-up resistors are mounted in a group next to each other.

If you have an earlier version of the PCB, there will be no mounting holes for the pull-up resistors, so the resistors will have to be mounted on back of the PCB. Each resistor will have one end connected to the 3.3V line and the other end connected to each of the A and B encoder lines. The picture shows one way to do this. This is another case where it is useful to have a helper when soldering the resistors.



## Step 8: Mounting the PCBs

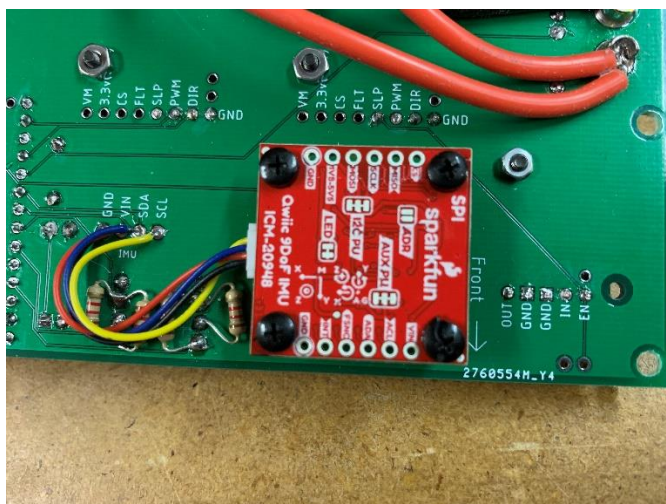
Before you mount the motor controllers, you should add the large electrolytic capacitors to them in the space provided. Be sure to get the polarity correct.



There are four circuit boards that are mounted directly on the main PCB: the two motor control boards, the 5V regulator and the IMU. The usual way to mount these is to use a standoff or spacer to hold the board away from the main PCB and then clamp the sandwich together using a screw and nut. A 6mm or 1/4" spacer works quite well. An M2 or 2-56 screw and nut holds them together. There are probably many other ways to do this. I find that the parts from [this kit from Amazon](#) contains almost everything you need. The IMU, mounted on the bottom, has larger mounting holes so you can use the M2.5 standoffs. In any case, a trip to the hardware store will usually yield a solution.

Mount all the boards into their proper position. All the boards, except the IMU, are mounted on the top of the PCB. Note that the IMU is mounted with the connectors near the main PCB and oriented so that the label "ICM-20948" is nearest the Teensy.

The easiest way to connect the IMU is to hook up a QWIIC connector to the board and solder the wires from the connector to the GND, 3.3V, SDA, and SCL on the main PCB. If you have purchased a connector cable, you can snip off one of the connectors. The black wire will be GND, the red wire 3.3V, the blue wire SDA and the yellow will be SCL. You can also create your own connections using the plated through holes on the side of the IMU board. Note that the QWIIC connectors provide a convenient way to expand the capabilities of SixPotatoe without additional wiring.



Once the boards are mounted, the connections/jumpers can be installed. To do this, run a small solid wire between the board and the main PCB and the solder each end and clip off the ends.

For the motor controllers, only the GND, DIR, and PWM, need to be connected. For the regulator, only the IN, GND, GND, and OUT need to be connected

Once you have mounted all four boards, you need to use the same general technique to mount the main PCB to the bottom plate. I find that 10-15mm or 1/2" spacers are about right. The spacers should be long enough that IMU does not touch the base plate. The screws and nuts can be M2.5 or 4-40. Again, the spacer/standoff kit from Amazon has parts that work very well in this situation.

The last step is to connect the wires from the battery connector to the motor controllers. Train the red and black wires to the GND and VIN connections on the controllers. Cut the wires to the proper length. Before stripping the wires, it is usually easiest if you tin the ends so that the wire doesn't fray when you strip the wire. After you strip the wire, place the red wire into the VIN and the black into the GND and solder them. It will take a fairly hot iron to do this but be careful; the board should not become so hot that other components on the board become unsoldered, thereby ruining the board. Pololu provides screw terminals that can be mounted on the boards. If you wish, you can use these rather than solder the wires.

## **Step 9: Set Up the Arduino Development Environment**

First of all, get the current SixPotatoe code from the GitHub repository, <https://github.com/robertstuart/SixPotatoe-Instructable>, and install it on your system. Again, if you have not used GitHub, there are tutorials and Instructables that explain how to do

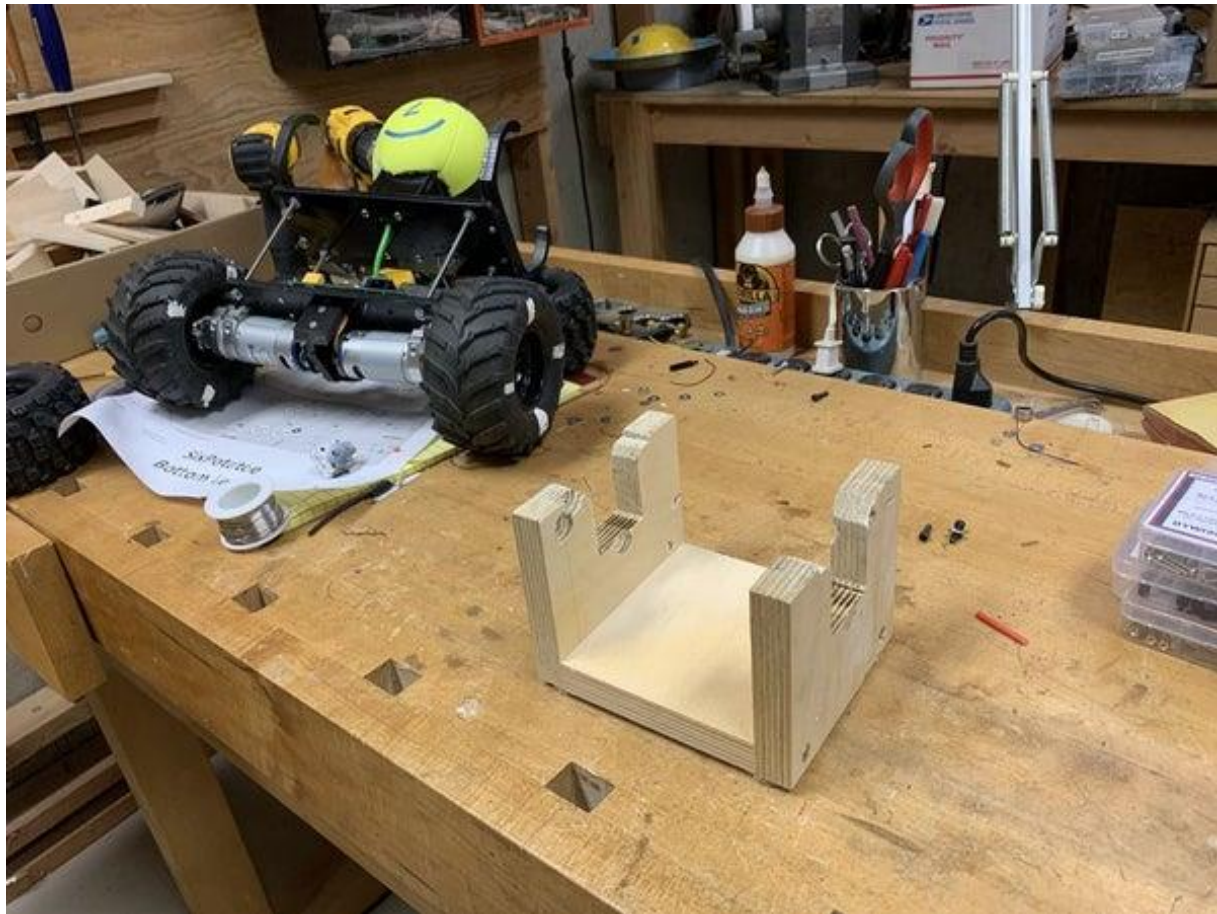


this. I think the easiest way is to click the "Clone or download" button and then download the zip file and put the code wherever you do your development.

You must then set up the Teensyduino environment. Go to this web page, [https://www.pjrc.com/teensy/td\\_download.html](https://www.pjrc.com/teensy/td_download.html), and follow the directions for your environment. Once everything is set up, you will have to start up the Arduino development environment, click on the "Tools" menu and select "Teensy4.0" from the "Board:" menu. Once you are this far, it is a good idea to hook up your Teensy to determine that the SixPotatooe compiles correctly.

## Step 10: System Testing





Before you start testing and running SixPotatoe it is useful to have some sort of test stand. Something like four soda cans will work just fine but I like to make a simple stand out of wood. You can see the cradle that I use in the above pictures. Keeping SixPotatoe in the stand, also keeps the tires from developing "flat spots" when sitting around idle.

After all of the connections have been made, it is a good idea to make some simple checks to be sure that there are no major wiring errors. If you have an ohm meter, measure the resistance between the two battery terminals on the PCB. This should measure in the meg-ohms. You should do the same between the Gnd and Vin on the Teensy. These are the two end pins near the USB connector. This should also be in the meg -ohms.

Next, you must set values in your SixPotatoe.ino file to match the motors that you have installed. There are parameters for several motors already established in the file:

```
RZONE_1621
```

```
RZONE_612
```

```
RZONE_437  
YELLOW_1140  
YELLOW_435
```

One of these values must be specified in first `#define` in the beginning of the *defs.ino* file. The line defining your motor should be uncommented. This will establish appropriate values for each of these motors. If you have other motors, you can probably come up with reasonable values for your motors using the previous definitions as a guide.

There are four system tests:

### System Test #1

This test checks that the IMU is working. Near the bottom of the *defs.h* file there are some lines that activate the system tests. You will need to set the appropriate system test to "true". For this test, you will need to set the lines as follows:

```
// Test defines  
  
const bool IS_TEST1 = true; // Set to be true for the 1st system test.  
  
const bool IS_TEST2 = false; // Set to be true for the 2nd system test.  
  
const bool IS_TEST3 = false; // Set to be true for the 3rd system test.  
  
const bool IS_TEST4 = false; // Set to be true for the 4th system test.
```

Hook up the USB connector on the Teensy to your computer that is running the Arduino development environment. The light in the RC receive should come on. Upload the SixPotatoe sketch and run the serial monitor. The LED on the Teensy should blink and the output on the serial monitor should show the pitch, roll, and yaw, in degrees, of SixPotatoe. Move SixPotatoe around and see if the orientation looks correct. All three values should be near zero when SixPotatoe is upright. The pitch should go positive as SixPotatoe is leaned back and negative as SixPotatoe is leaned forward. Roll will go positive as the left motor is lifted up and Yaw will go positive as SixPotatoe is rotated clockwise.

### System Test #2

This test checks that the Radio Control works. If you haven't already done it, this is the time to bind the transmitter and receiver. Follow the instructions that came with the RC unit to do the binding.

Once the binding is complete set IS\_TEST1 to false, and IS\_TEST2 to true. Upload the program and look at the serial monitor output. Channel 1 value should vary as you turn the steering control, and channel 2 should vary as you manipulate the trigger (throttle). The top left control should change channel 5 and the top right control should change channel 6. Channel 3 is toggled by the push button on the side of the handle and Channel 4 has a 0, 1, and 2 state controlled by the slide switch.

### System Test #3

This test checks the operation of the motor controllers and motor encoders. Also, SixPotatoe should be sitting up on a test stand of some sort. Plug the battery into the connector on the PCB. Unplug the USB from the computer. The LED on the Teensy should continue to blink. Plug the USB connector back in. Set IS\_TEST3 to true and all of the other tests to false. Upload the new SixPotatoe program.

Press the LED push-button. The LED should come on. Toggle the on/off button on the controller. The LED on the switch should turn on and off just like the LED on the controller. Turn the LEDs on. Pull the trigger. Both motors should be operated both forward and backwards by the trigger. The motors should go forward when the trigger is pulled in and backwards with the trigger is pushed out.

The first two columns displayed in the serial output show the right and left wheel speed. The wheel speed should be positive when the trigger is pulled and negative when it is pushed out. The direction will vary depending on the motor encoders and gear box. If the direction is incorrect, this can be easily fixed by simply changing the pin definitions in *SixPotatoe.ino*.

### Configuration A:

```
const int ENC_B_LEFT_PIN  = 20;
const int ENC_A_LEFT_PIN  = 21;
>const int ENC_B_RIGHT_PIN = 22;
```

```
const int ENC_A_RIGHT_PIN = 23;
```

### Configuration B:

```
const int ENC_B_LEFT_PIN  = 21;
```

```
const int ENC_A_LEFT_PIN  = 20;
```

```
const int ENC_B_RIGHT_PIN = 23;
```

```
const int ENC_A_RIGHT_PIN = 22;
```

Use whichever configuration works for the motors you have.

The wheels will turn in opposite directions when the steering control is manipulated.

This is probably the appropriate place to do the initial setup for the adjusting knobs that are underneath the lid on the controller. The bottom left knob adjusts the gain for the steering. This should probably be turned all the way up to 100. The knob above that sets the centering for the steering. This should be set at the center, 0. The right bottom knob sets the gain on the throttle. If you want to run the robot at maximum speed you must set this at 100. However, I recommend that this be set at 50 or lower while you are learning to operate the robot. Also, it is useful to set this at a low value for new users, particularly young children. The knob above that should be set to the center position.. It can be used to set the robot to zero speed. It can be quite useful to stand still on sloping ground.

Both of the switches at the top should be set to the left.

### System Test #4

This test checks the internal control circuitry and software for motor speed control. As before, set IS\_TEST4 to be true and all other tests to false. The motors can be turned on and off, either by pressing the button on SixPotatoe or CH3 on the controller. Again the measured speed will appear in the serial output. If SixPotatoe passes this test, it should work correctly in normal operation.

### Preliminary balancing test

Set all the tests to *false* so that the system is in its normal running mode. Plug in the battery connector and the LED on the Teensy should show its heartbeat. While holding SixPotatoe nearly vertical, press the on/off button once and the LED on the button should come on. Press it again after a second or so, and the light should go off. The motor may have started up while the light was on. If you repeat this while tipping SixPotatoe forward or backwards a few degrees you will see that the motors will start accelerating more forward the more SixPotatoe is leaning forward. You will see the opposite effect as you tip SixPotatoe backwards. If this is what you see, your robot is ready to run!

## **Step 11: Running SixPotatoe**

I think that you will find that SixPotatoe is a lot of fun to drive. If you are not used to the RC controller, it takes a little time to get used to the steering and accelerator – particularly if you are using one of the high RPM motors. Here is some general information to help you get going:

### **Mounting the wheels**

Mount the "D" clamp on the end of the motor shaft. Make sure that the shaft does not protrude past the face of the clamp. Tighten the clamp. Then mount each of the 12mm wheel adapters to the clamp and tighten the four 6-32 screws.

Next, the wheels must be mounted on the wheel adapters. Be careful that the 8-32 wheel-mounting screw does not bottom out on the motor shaft. In the worst case, you can tell that the screw is bottomed out when the wheel is loose after tightening the screw. If the wheel feels solid, it is still sometimes safest to add another washer under the 8-32 screw to be sure that everything is tight.

### **The LEDs**

When you connect the battery, the LED on the Teensy will start blinking at about two times a second. This tells you that it is going through its usual 200 per second cycle. If the light is not blinking, something is very wrong and SixPotatoe will not work.



The system LED on the switch should initially be off. If you hold SixPotatoe in the vertical position and press the switch, the light should come on and SixPotatoe will start balancing. If you press the switch while SixPotatoe is lying down, the LED will flash rapidly indicating that it is ready to go. In fact, you can just start lifting up SixPotatoe and it will continue to get up by itself. With a little manipulation of the throttle, it will easily get up as described below.

## **The controller**

You may have set up the adjustments in the controller but we will do it again here to verify that everything is working. Be sure that the CH4 three-position switch is pulled all the way back. Flip up the lid to the four adjustment knobs. The top left switch should be set to the left, "Nor", position. The top right switch should also be set to the left, "NOR", position. The top left knob should be set to point straight up to the "0" position. This sets the steering bias and there is little need for it with SixPotatoe. The bottom left sets the steering gain. This is normally set at its maximum, 100. Sometimes it is useful to set this at a lower value, particularly for new users or young children who often turn the steering knob too much. A lower value on this adjustment makes the steering less sensitive. The top right adjustment should be centered at the "0" position. This can be adjusted to make SixPotatoe not wander off forward or backward. This can be particularly helpful in making it stand still while on a slope. The bottom right adjustment is the throttle gain. Again, it is useful to turn this back when learning the system and for young children.

There is a RC-Control.pdf file on the repository that has templates for the top of the RC controller that you can cut out and glue to the controller that should help you with these settings.

The CH3 button on the controller turns SixPotatoe on and off and duplicates the function of the switch on SixPotatoe. You should realize that the controller only transmits commands to SixPotatoe and receives no information from SixPotatoe. Therefore, the CH3 button and the button on SixPotatoe may be out of sync if you press the switch on SixPotatoe.

Other than that, control of SixPotatoe is very simple: pulling on the trigger causes SixPotatoe to go forward and pushing on the trigger causes SixPotatoe to go backwards. Similarly, turning the steering knob clockwise causes him to turn right and counterclockwise causes him to go left.

Generally speaking, operation of the throttle will not affect the steering and vice versa. However, SixPotatoe does try to prevent turning a corner so sharply that it tips over. Therefore, very sharp turns will not be possible at high speeds.

## **Getting Up**

SixPotatoe can get up from lying on the ground and it seems to delight everyone who sees it. There is no special code to make this happen; it is just the result of his general ability to stay upright. If SixPotatoe is lying down, you can drive around while lying down. Sometimes this is necessary to get into some open space. SixPotatoe gets up most easily when it is traveling, motors first, and the throttle is rapidly reversed. Usually, just a flick of the throttle will bring him upright. Also, he can be brought upright with a just a little kick from the toe or by "falling" off of a short drop, such as a curb. Getting up requires good traction, so these maneuvers may not be successful on slippery floor, wet grass, loose gravel, etc.

## **Steps and Curbs**

SixPotatoe tries to stay upright when encountering vertical drops. It does pretty well at this but it is not perfect. Yet! When encountering a vertical drop, it is a good idea to have SixPotatoe going at a steady speed and be in a relatively vertical position. Drops are less likely to be successful if SixPotatoe is rapidly accelerating or decelerating.

Also, large drops are potentially harmful. SixPotatoe has no suspension so the small wheels must absorb all of the impact of a large drop. I haven't bent any axles or ruined gear box bearing yet but I expect there is considerable risk of both with larger drops. Jumping off of a chair is about as large a drop as I normally attempt and, even then, I am reluctant to do it. I have successfully jumped off of a tabletop, but this always seems a little risky.

## **Rough Ground**

Traveling over rough ground such as deep grass or rocky areas is one of SixPotatoe's most striking abilities. It is one thing that people are always surprised to see. It is kind of fun to see if you can push SixPotatoe through difficult terrain and I urge to try different kinds of terrain to see how he copes. He has never been near the ocean! If you take him to the beach sometime, please send me a video!

## Autonomous Navigation

SixPotatoe can autonomously navigate a course with surprising accuracy. You can see this in the videos of TwoPotatoe and ThreePotatoe navigating through AVC contest sponsored by Sparkfun. SixPotatoe has much of the same navigation code embedded in it so it is theoretically capable of the same thing. Instructions on how to do this are posted on the [TwoPotatoe web site](#).

## Logging

SixPotatoe can log events so that you can play them back and examine the results. The log can even be put in a spreadsheet and graphed. There is a general explanation of how to do this on the [TwoPotatoe web site](#).

## Help

If you need help, you should probably post a query in the discussion section for this Instructable. You can also reach me by filling out the “Leave a message” form on the [TwoPotatoe web site](#)

## Step 12: Final Caveats







Skate Parks

Skate parks are an ideal place to exercise SixPotatoe's many capabilities. There are a few words of caution, however. The first is general safety considerations. Skate parks are sometimes full of young children. At other times, there are fairly young adults moving very fast and doing very athletic tricks. You should exercise caution at all times while at the skate park. Skate parks are usually relatively free early in the morning and during the school year. SixPotatoe tends to draw a crowd wherever he appears so it is usually best to use the skate park while school is in session or early in the morning.

Also, one of the skate park that I use in Lafayette, Colorado has a sign stating that motorized vehicles are prohibited. A literal interpretation of this suggests that robots like TwoPotatoe are not allowed. No one has ever said anything to me although I did have a police car pull up and the officer got out of the car with a video camera. I turned out that he thought the robot was really cool and wanted to film it to show to his children.

### **Lipo batteries**

Lithium polymer batteries are a known hazard. There are many articles on the internet explaining their proper care. As an example of what can happen, I once accidentally punctured the battery in TwoPotatoe. I managed to get TwoPotatoe outside before any serious damage occurred, but the house filled up with smoke and set the fire alarms off. Be careful!

Also, follow the usual guidelines of Lipo batteries. They should be maintained at the appropriate charge and shouldn't be deep discharged.