

Dual Axis "Smart" Solar Tracker Instructions

Product Code: 1221504

Brand: Brown Dog Gadgets

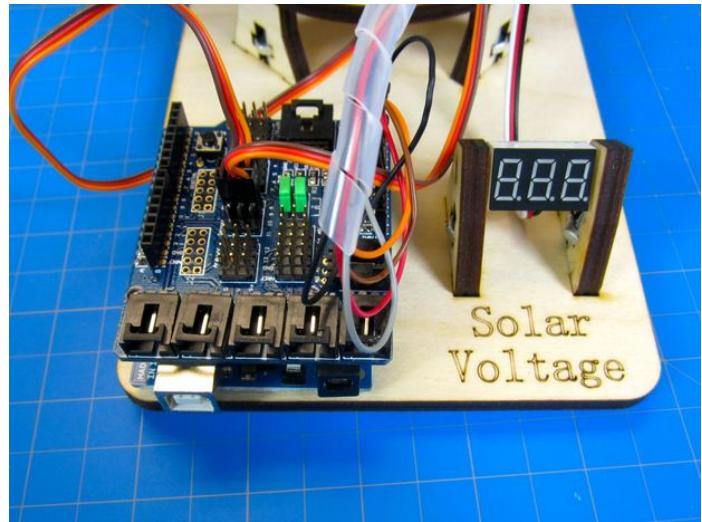
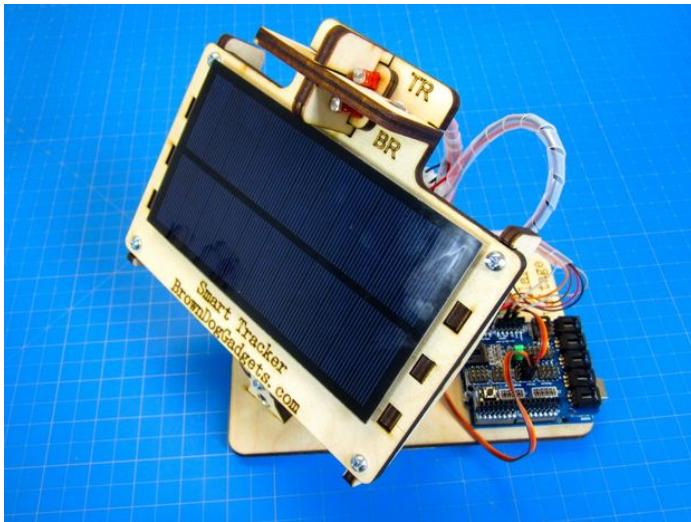
Dual Axis Solar Tracker	1
Intro: Simple Dual Axis Solar Tracker	1
Step 1: Why Track The Sun?	1
Step 2: What You'll Need	2
Step 3: Types of Trackers	3
Step 4: Brains, Sensors, and Servos	4
Step 5: Laser Cutting	4
Step 6: Video Build Guide	5
Step 7: Step By Step Diagram	5
Step 8: Attach the Servos to their Mounts	6
Step 9: Attach the Servo Arms to their Mounts	7
Step 10: Building the Base	8
Step 11: Building the Top	9
Step 12: Build the Center	12
Step 13: Home the Base Servo	13
Step 14: Home the Center Servo	14
Step 15: Double Check Everything	15
Step 16: Attach the Arduino	16
Step 17: Assemble the Sensors	17
Step 18: Hook Up the Terminal Block	18
Step 19: Hook Everything to the Arduino	19
Step 20: The Code	20
Step 21: Common Problems	23
Step 22: Extra Add Ons	23
Step 23: Finished	24

Intro: Simple Dual Axis Solar Tracker

We love using solar energy with our electronics projects. For the most part it's extremely easy to work into small, low voltage, projects.

One frequent question we get from students and hobbyists is "How can I make a solar tracker?"

That's a great question and an even more awesome project, but it's never been overly easy to do. We found ourselves overwhelmed by the "pre made" single axis "dumb" trackers on science education websites (as well as shocked at the \$200 price tags), and overwhelmed by many of the "from scratch" DIY solar trackers. Neither of these options seemed like something we'd want to give to teachers or students so we designed a third option.



Step 1: Why Track The Sun?

It seems you can't walk down the street these days without coming across a solar panel. You can find them lighting up crosswalk signs, mobile power for construction, as well as simple little sidewalk path lights. Solar is easy to use, readily available, and inexpensive.

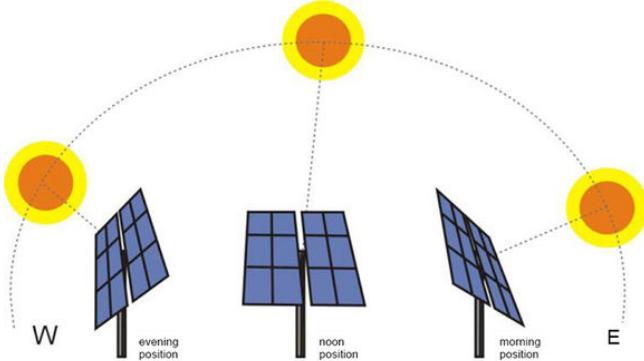
So why aren't we using it to power our homes?

For the most part our common every day solar cells run at an efficiency of 18-20%, meaning they convert 18-20% of the energy they receive into electricity. While this is far better than the 3-6% efficiency that most green plants end up with, it doesn't quite meet our power needs. To bring in enough power we either need to improve the efficiency of our panels or find ways of getting more from our current solar panels.

Every panel you see in your day to day life is in a fixed position, most likely facing south at a 45 degree angle. While this approach is extremely simple and meets the needs of most small applications, it isn't producing as much energy as it could be.

The single most simple way of getting more energy out of a solar panel is to have it track the sun. In fact solar panels that track the sun create around 30% more energy per day than a fixed panel. With that kind of power increase you'd think everyone would be doing it, but there are some good reasons why it's not overly common. First, the initial cost of setup is higher since it requires moving parts. Second, it also requires maintenance and upkeep since they'd be exposed to outdoor conditions year round. Third, you'd need to power this equipment in order to keep it running and moving which then takes away from your output.

For most applications and home use, tracking is overkill. We typically don't see tracking used unless it's in large industrial power generation systems. Though that doesn't mean you can't make your own version at home.



Step 2: What You'll Need

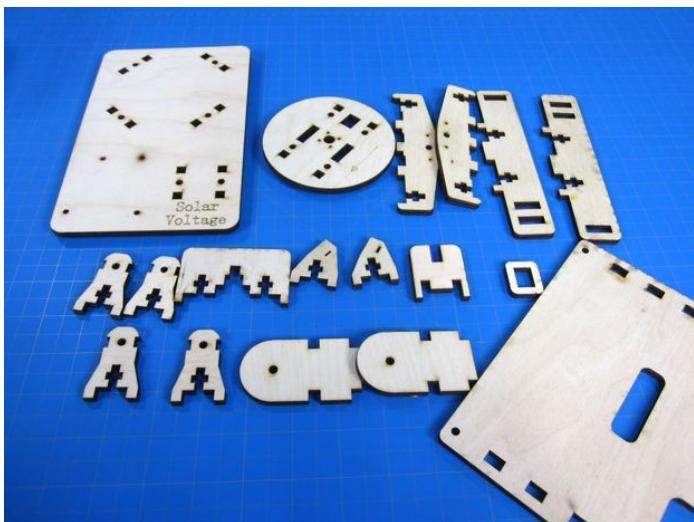
To complete this project you'll need the following tools and parts, as well as access to a laser cutter or CNC router. We understand that not everyone has access to these tools which is why we've put together a kit containing everything you'll need.

Tools Needed:

- Wire Stripper or Cutter
- Several Small Screw Drivers
- Rubber Feet
- Cable Wrap or Twist Ties (highly recommended)

Included Parts:

- Arduino Uno + USB Cable
- Arduino Sensor Shield
- 2 x 9g Metal Gear Servos
- 1 x 5 Port Terminal Block
- 1 x 4 Port Terminal Block
- 4 x 10K Ohm Resistors
- 4 x Light Detecting Resistors
- 4 x JST Socket Connector Cables
- Jumper Wires
- 5.5V Solar Cell
- LED Volt Meter



Step 3: Types of Trackers

There are a couple different types of trackers as well as ways to track the sun.

Single Axis or Dual Axis

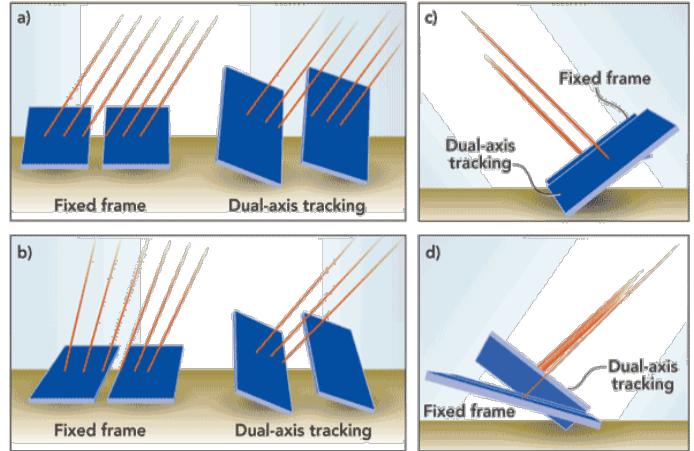
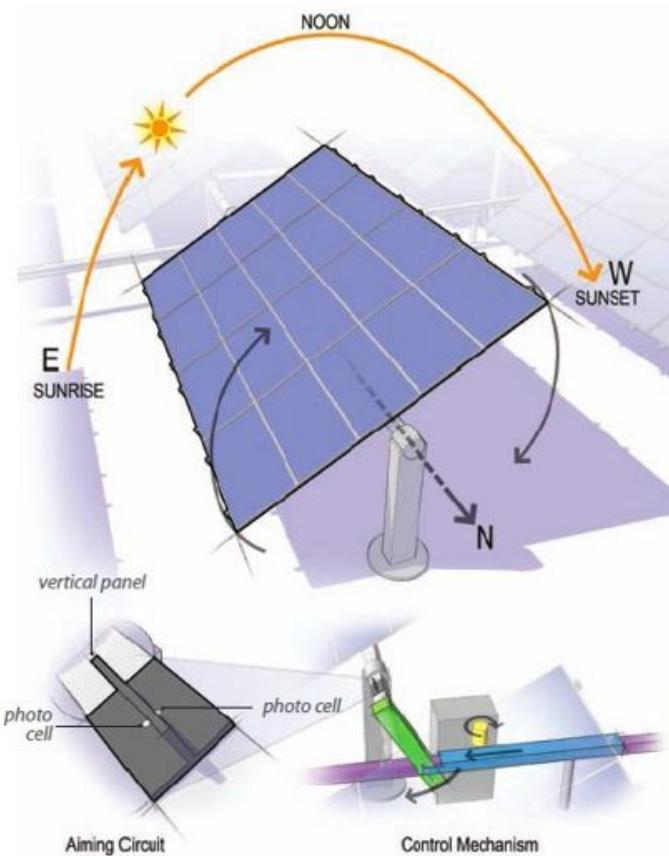
Our tracker is a **dual axis tracker**, meaning it tracks in both X and Y. To put it into even more simple terms, it goes left, right, up, and down. This means once you have your tracker set up you will never need to change or adjust anything, since anywhere the sun moves your tracker will follow. This also impresses people at parties because you can have it track a flashlight around. This method gives the best results for power generation.

If you want to make things a bit more simple you can make a **single axis tracker**, one that does just X or Y. To put it in simple terms again, it'll do just left to right or just up and down. Typically people will make an X axis (left to right) tracker and then just set their panel at 45 degrees for Y. This still gives really high amounts of power generation while at the same time eliminating half the moving parts. You'll frequently find this approach being used in "dumb" trackers that are not computer controlled.

Active Tracking or Scheduled Tracking

Our tracker is an **active tracker** which is controlled by computer program (via an Arduino). This means that we use sensors to find the brightest source of light at all times. If you were to take a flashlight and shine it at the sensors the tracker would follow it around. While this is the most interactive and exciting kind of tracking you can build, it's also overkill for larger setups.

The sun is highly predictable. If you can easily look up the time of every sunrise and sunset for the next 100 years as well as use some simple math to figure out the angle of the sun relative to your location at any time of the year. With this in mind many people end up using a **scheduled tracker**. This system uses a computer program that changes the angle of the panel based on the date, time, and physical location. While not as fancy or exciting as an active tracker, it is in fact far more efficient provided everything is set up properly. You can be sure that your panel is at the mathematically most efficient spot possible even under heavy cloud coverage.



Step 4: Brains, Sensors, and Servos

Since our program is rather simple we've opted to use an Arduino Uno. The Arduino is extremely common for DIY projects as well as quite inexpensive to buy. The Arduino platform is also very easy to for anyone to learn or just modify code using a home computer, something we'll get into later on.

We're using two "micro" servos in the 9g size. We're opting to use ones with metal gears (our kits also include these) since we want our project to last a very long time. The metal gear versions also provide a bit more torque than the plastic geared versions. If you want to use larger servos you can easily modify our laser cut files.

For sensors we're using four light sensitive (detecting) resistors, also known as LDRs. Again, these are super common and you can often find them in outdoor garden lights or indoor night lights. They work by changing their resistance level based on how much light is hitting them. The more light, the less resistance they have.

Our program works by comparing the resistance of the four sensors and moving our servos. How sensitive our sensors are completely depends on our code. The same goes for the servos. We've set up our code so that our servos can only move within a certain predefined area (as to not damage the rest of the project) and at a set speed. These two aspects can also be changed very easily in the code. We'll show you how to do this later on.

To also help things along and remove a bunch of wiring we're using an Arduino Sensor Shield. This is mainly to plug the two Servos into. If you're building this from scratch you could go without, but Sensor Shields are inexpensive and makes life a lot easier.

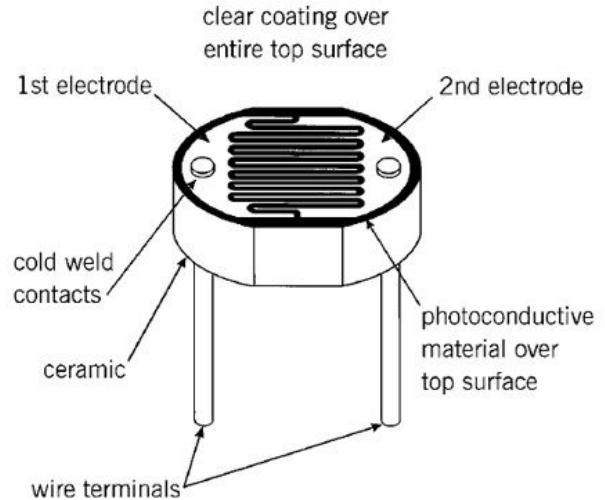
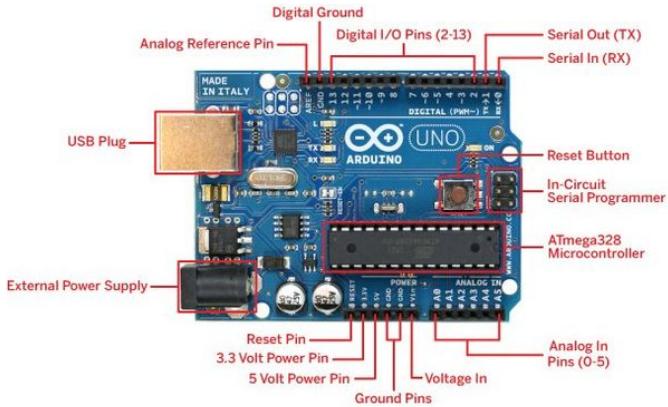


Figure 3
Typical Construction of a Plastic Coated Photocell



Step 5: Laser Cutting

The key to this project is having a functional structure to put it on. To do this you'll need access to a laser cutter and some quarter inch wood or acrylic (or you can glue two 1/8th inch pieces together). If you need a fully cut kit you can grab one from us.

If you don't have access to a Laser Cutter

If you want to free form your tracker you can do so rather easily. The downside is that you really can't mount a solar panel on them. Instructables.com user geo bruce has a nice and simple freehand design. (We also used a modified version of his code.)

You can also 3D print a structure. We really enjoyed this 3D Printed dual axis tracker structure by ThingVerse author OpenSourceClassroom. The downside is that 3D Printing takes a very long time and will probably cost more in plastic than in wood.

We also enjoy this alternative laser cut dual axis tracker design by pdaniel7, though it lacks a setup for the sensors we're using. If you wanted to go this route you could easily make your own sensor mount/ divider from some cardboard.

Tips on using the Laser

We've got all our laser cut files for download. Just ask and we can pass them along.

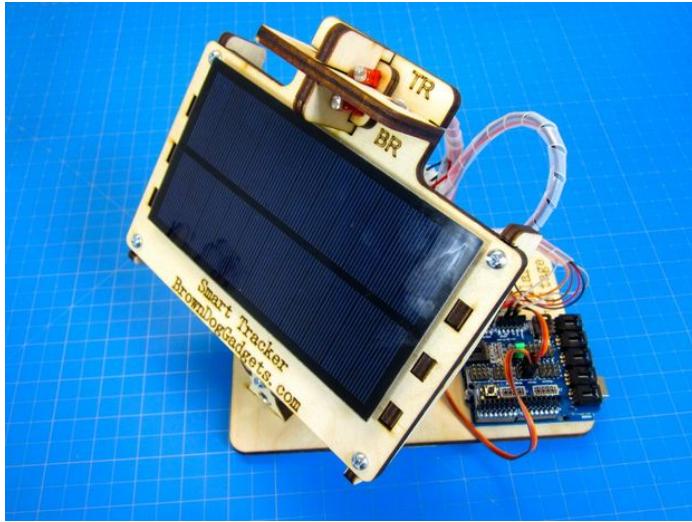
Be sure you're using quarter inch wood or acrylic. Ours measures out to be between 0.25 and 0.26 inches thick. If you use thicker or thinner material you'll need to modify these files, otherwise the parts will not fit together and you'll waste a lot of wood. The same goes for our screws and T connectors since they're all set up for the sizes we're using.

We've also included screw holes for an Arduino as well as a little mounting system for our LED digital volt meter. If you're planning on doing something different you can always modify these parts or remove them completely.

Step 6: Video Build Guide

If you're the kind of person who would rather watch a video instead of read directions, then look no further. You will need to copy and past the code from a later step, but the physical build and wiring is exactly the same.

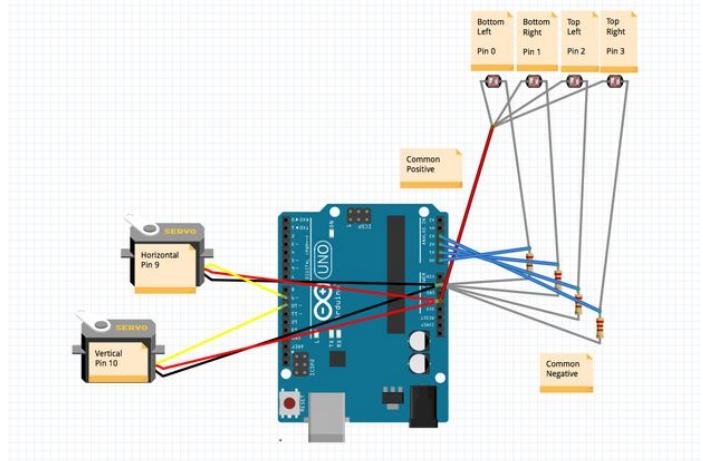
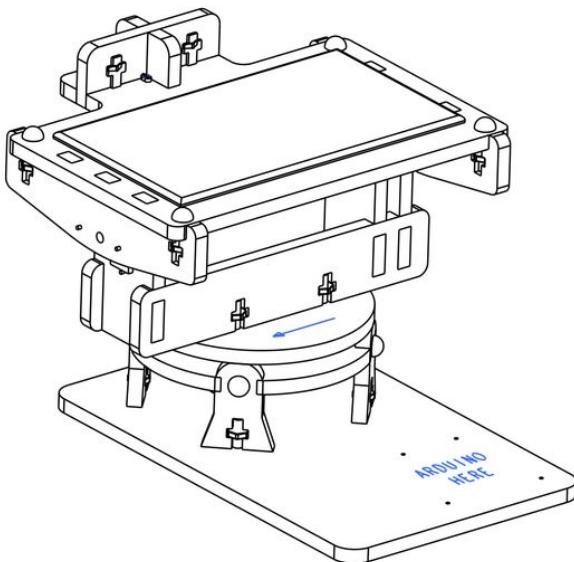
<https://youtu.be/f6FthqPwog>



Step 7: Step By Step Diagram

Sometimes it's easier to use an Ilkea type build diagram instead of pictures.

I've added a diagram of the electronics as well for a better visual reference.



Step 8: Attach the Servos to their Mounts

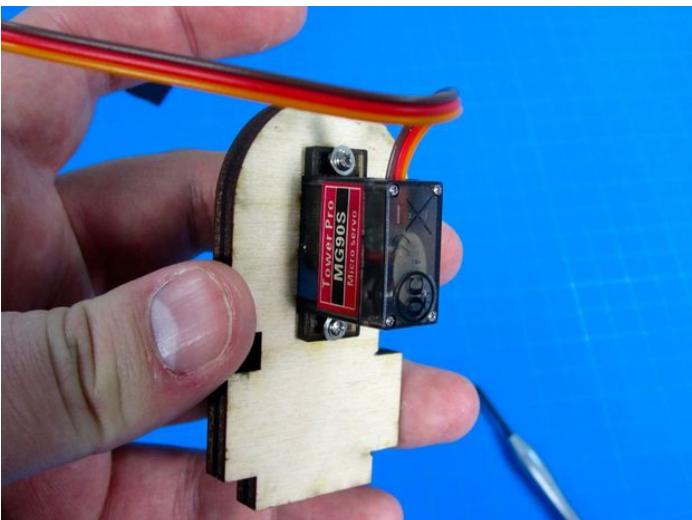
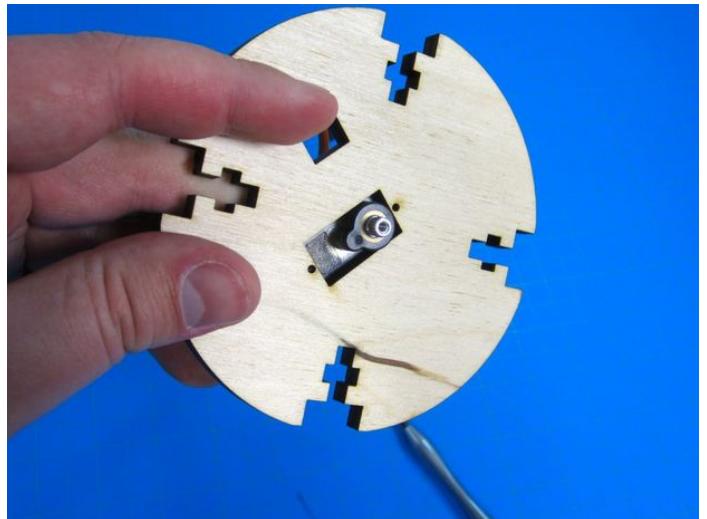
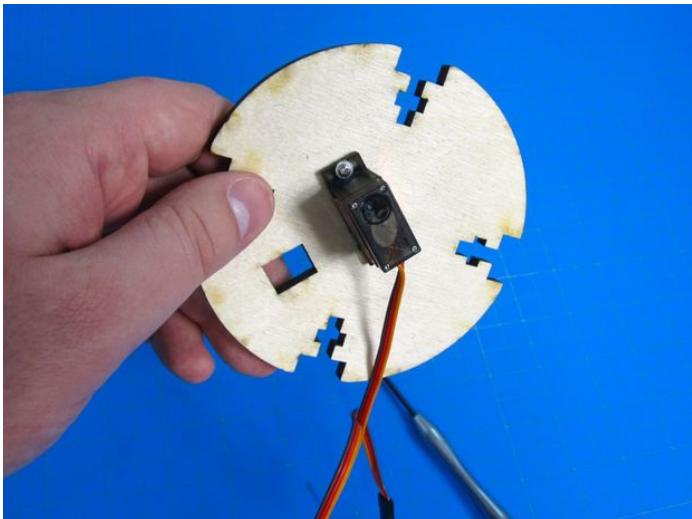
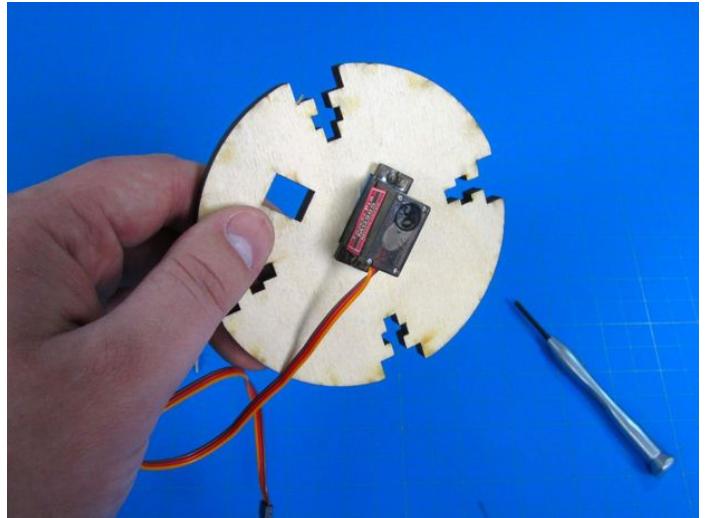
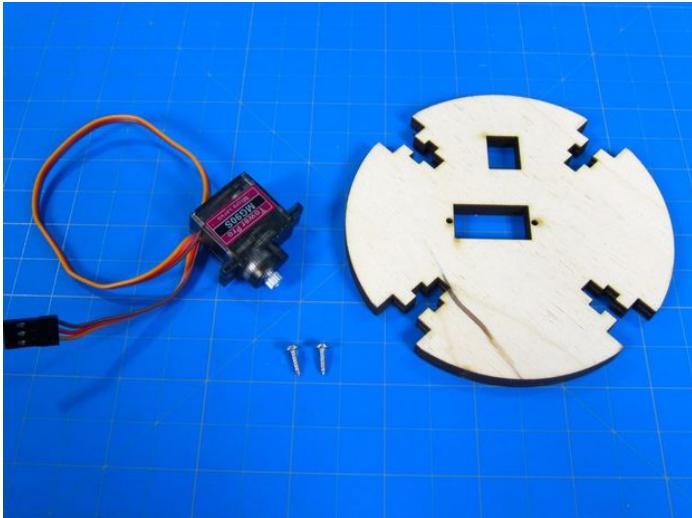
To start out we're going to attach the **Servos** to their mounts, which is actually the second most difficult part of this project.

Open up the bag the **Servo** came in. There will be three screws. Two pointy wood screws with large heads, and one small machine screw. Place the single machine screw to the side, we won't be using it now.

Pick up the large round wooden piece WITHOUT an arrow on it. (For additional confirmation, it only has two very small holes in it, whereas the other round piece has 4 holes.) The **Servo** is mounted on the bottom of the piece.

Line the servo up with the screw holes, and then carefully use the two **Servo** screws to secure it in place. Once in place give it a little tug to make sure it's secure.

Find the second wooden **Servo** mount. We'll be doing the same thing here with our second **Servo**. Mount it on the "back" of the mount with the two screws it came with.



Step 9: Attach the Servo Arms to their Mounts

Now this part is the most difficult part of the project. This is due to the small screws we'll be using and how difficult they can be to get through the wood and the **Servo Arms**. Be sure to go slowly as the **Servo Arms** can crack or break, though extra ones did come with the **Servos**.

Grab the other large round piece of wood. Notice how there are four little holes surrounding a larger center hole.

We'll be attaching the **Servo Arm** to two of these holes using two **Size 2 Wood Screws at 1/4th Inch length**.

Start by laying the round wooden piece on the table with the etched arrow facing down. The arrow side is our "top" side, and we want the **Servo Arm** to be screwed into the "bottom" side.. Be sure that the crown of the Servo Arm is facing up, and not buried in the hole and line up the little holes of the **Servo Arm** with two of the holes in the wood. In our case this resulted in us using the second hole on one side and the third hole on the other.

By hand (without using a screw driver), slowly twist the screw into the desired hole. Once it's in a little bit you can use your screw driver to get it all the way through. Go slowly as to not break the arm.

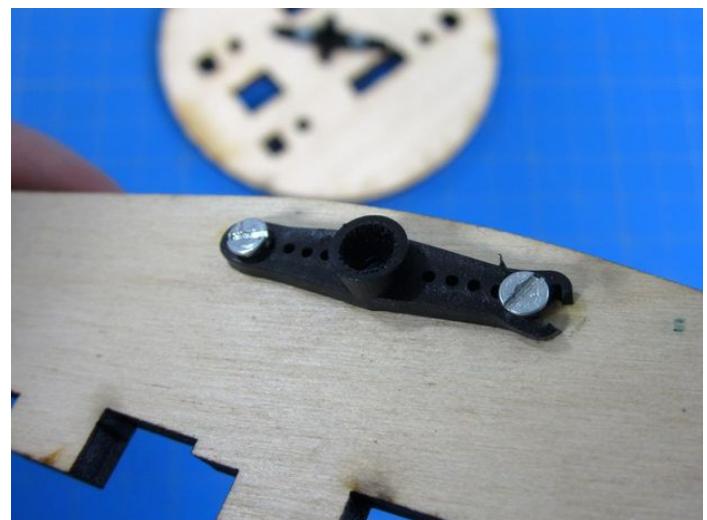
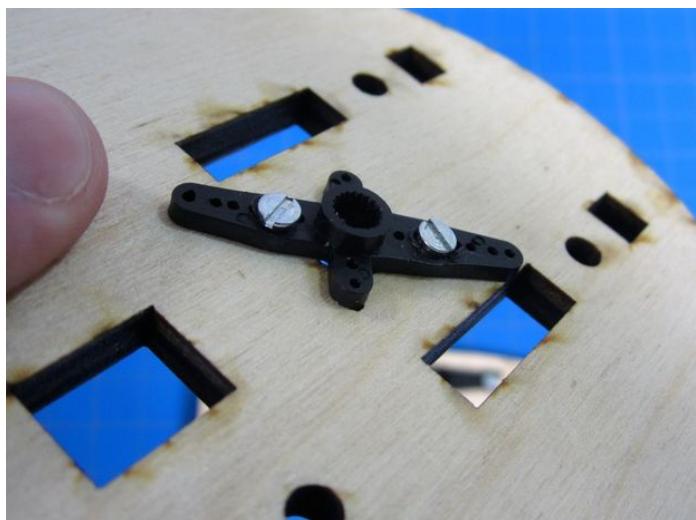
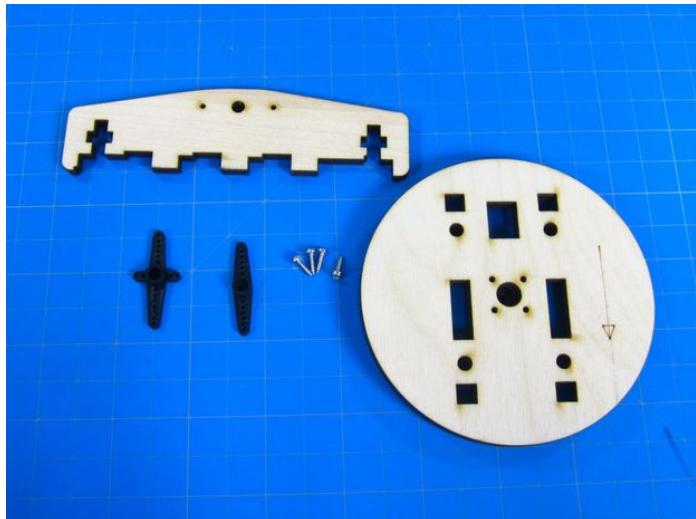
We prefer to get both our screws set a bit into the plastic before we start to screw them into the wood.

Make sure they're both secure in the wood.

We'll be doing the same thing to one of our two triangle shaped braces. They're both identical so it doesn't matter which one you choose.

Follow the same procedure using another **Servo Arm** and two more **Size 2 Wood Screws**. Don't forget that the crown on the **Servo Arm** needs to be facing away from the wood, and not buried in the center hole.

Once this is done, relax. You've done the two hardest parts of the project.



Step 10: Building the Base

Grab the very base plate, the four legs, and the large round piece that now has a **Servo** attached to it. You'll also need 8 of the 6-32 Screws and 8 nuts.

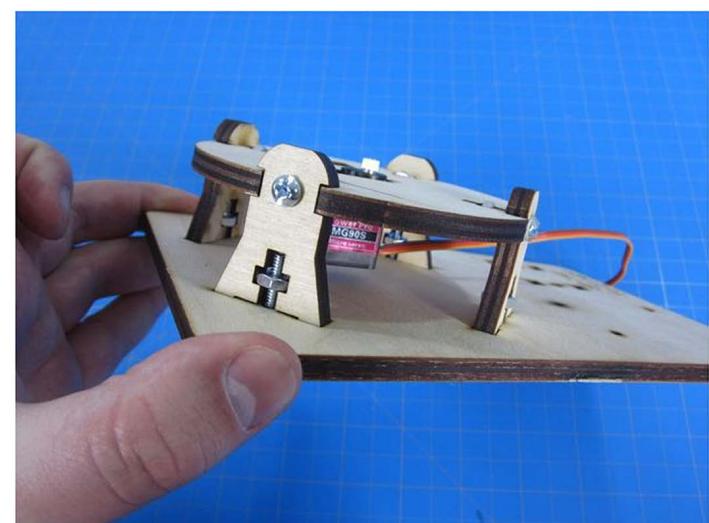
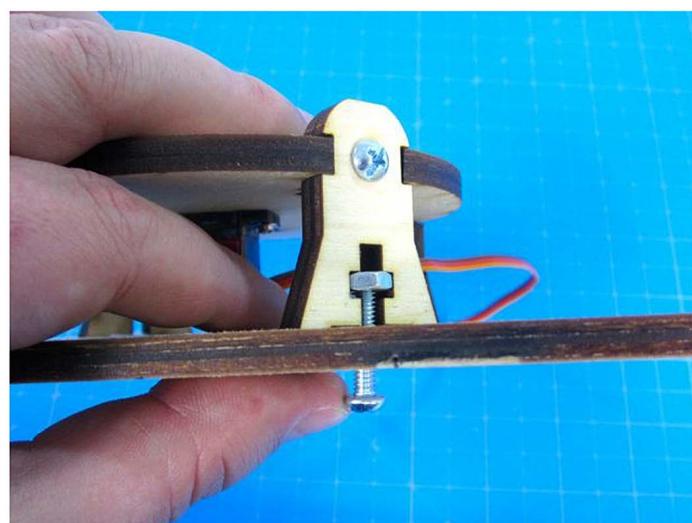
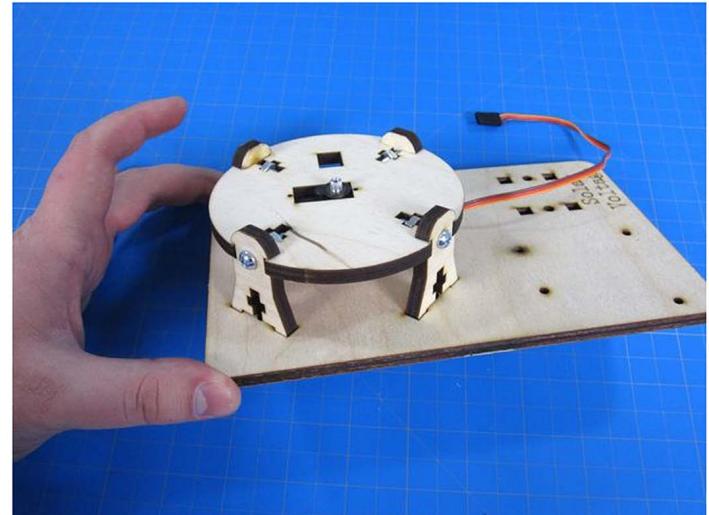
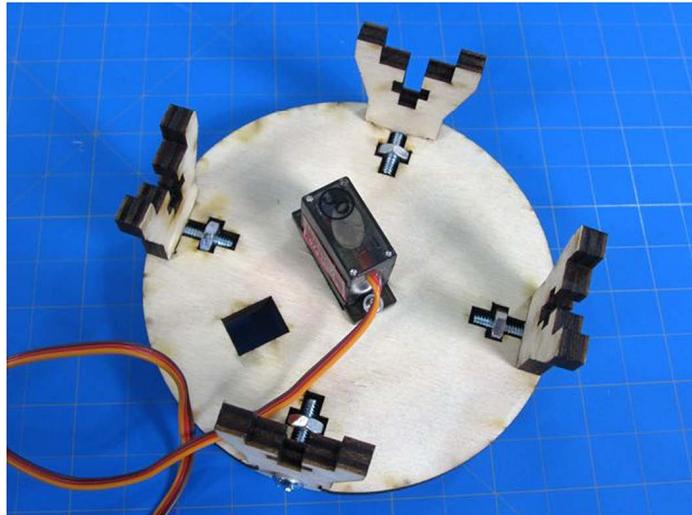
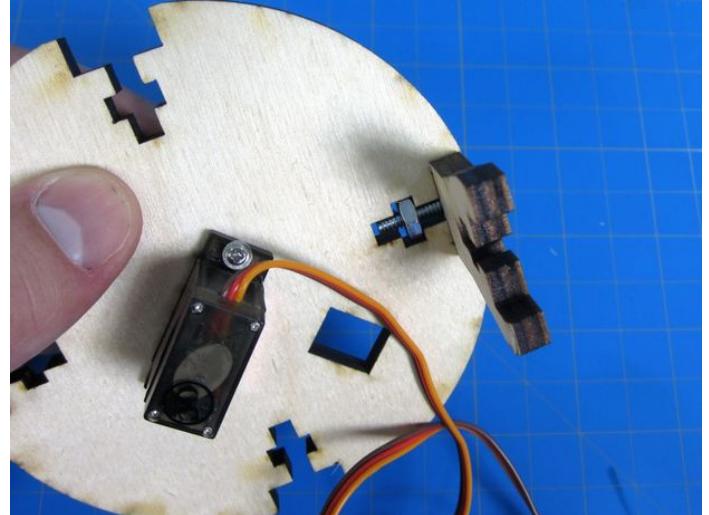
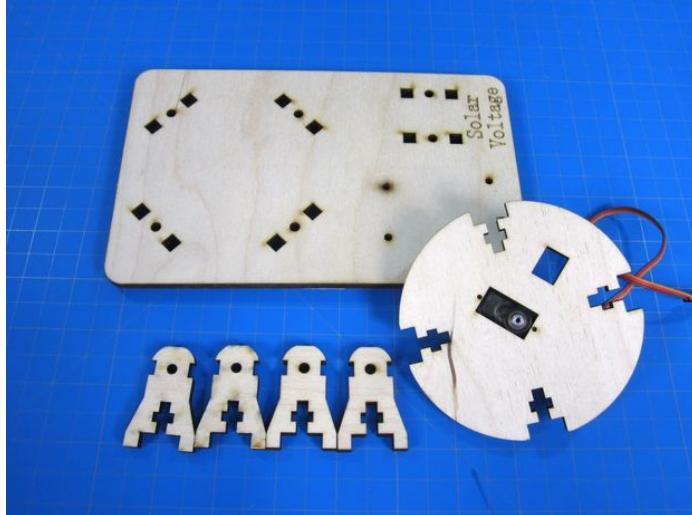
First attach the four legs to the round servo holder. The **Servo** needs to be inside all the legs, between the the base plate and the round servo holder. Don't tighten the screws all the way, leave them a bit loose.

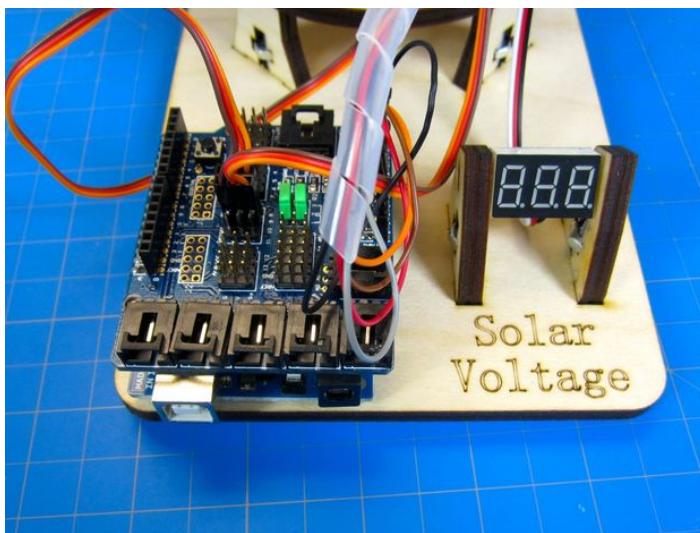
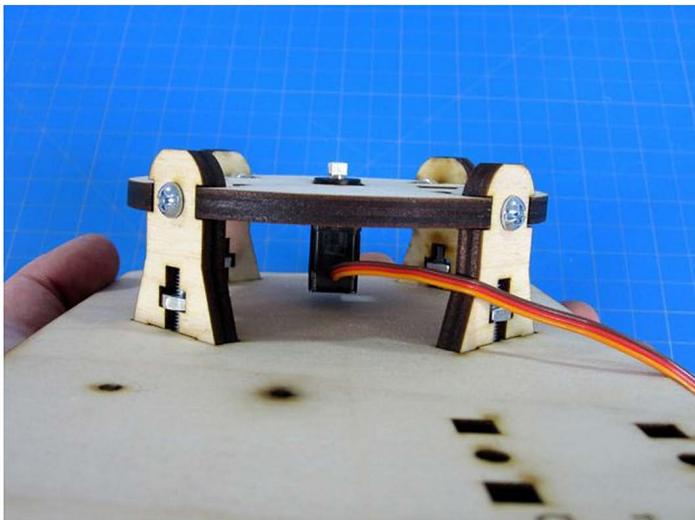
Now fit the four legs into the base plate. Make sure that when you do this the servo wire is positioned so that it's coming out towards the back where all our electronics will be.

Once all our legs are screwed into the base plate go back and tighten the four screws that attach the legs to the round servo holder.

Lastly, put the four rubber feet on the bottom of the base plate so that the screw heads don't scratch up your work surface.

At this time you can also put together the LED display holder. The LED display just fits between the two wooden holders and is secured by two screws and nuts.





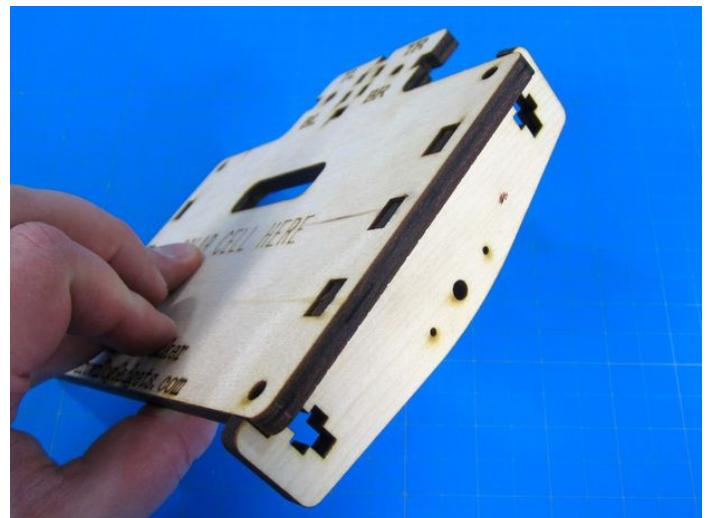
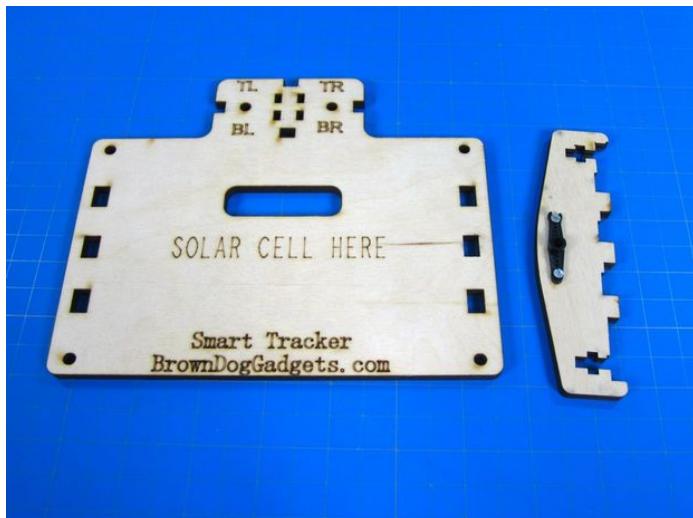
Step 11: Building the Top

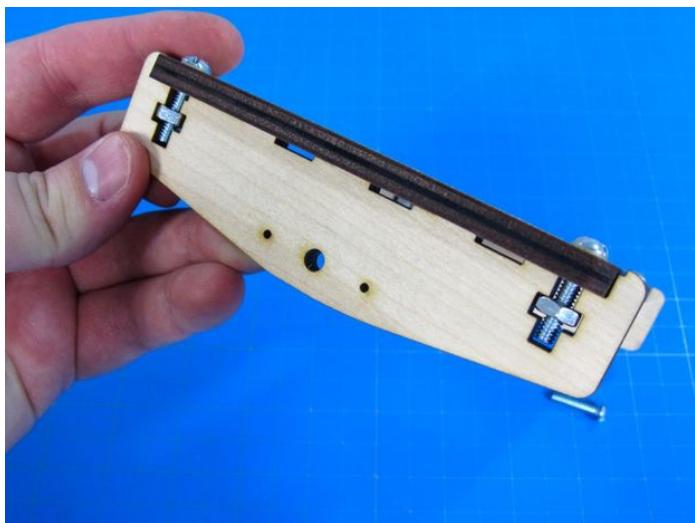
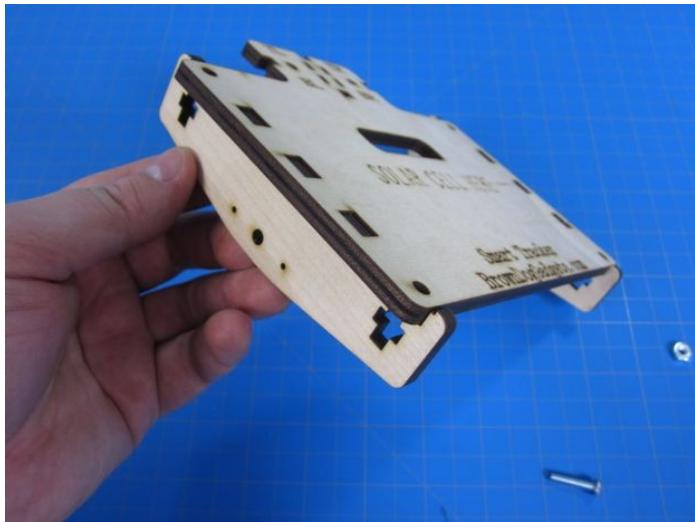
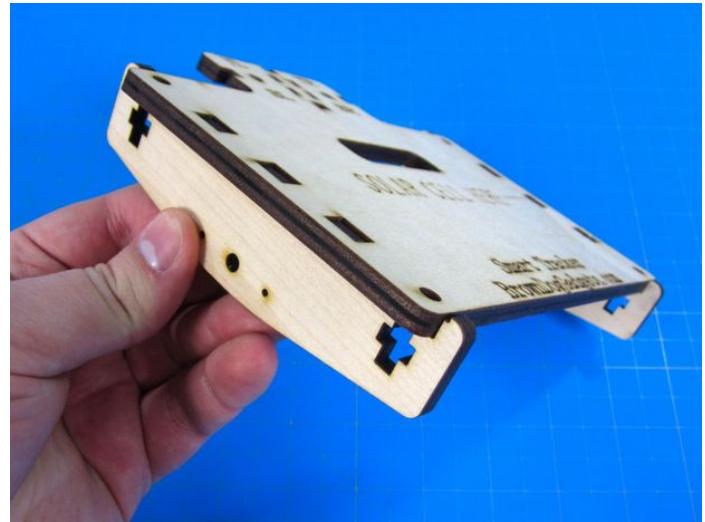
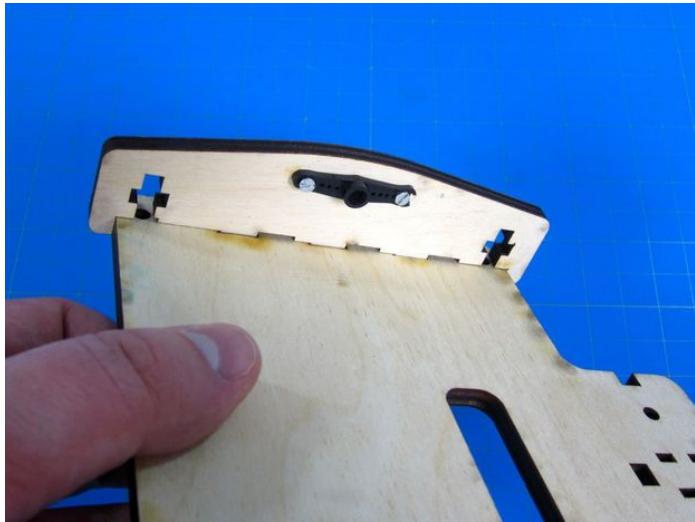
Grab the large solar array face. It's the one that says "Solar Cell Here" on it. We'll also need the two triangle wings, the small rounded corner square piece, and the two small sensor divider pieces. To connect it all we'll need six 6-32 Screws and nuts.

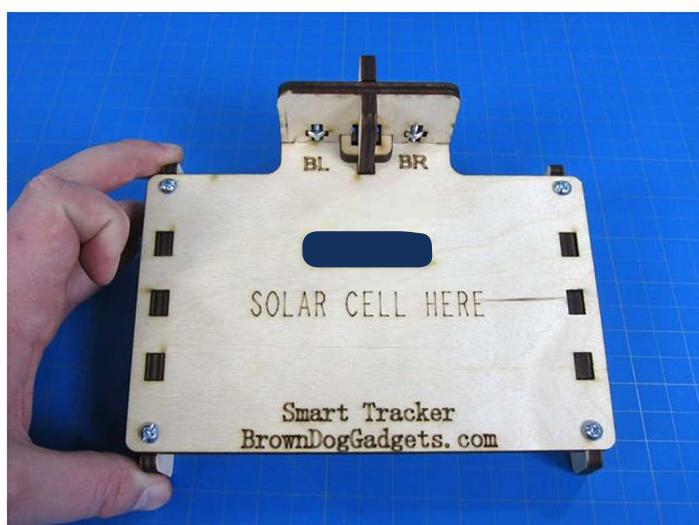
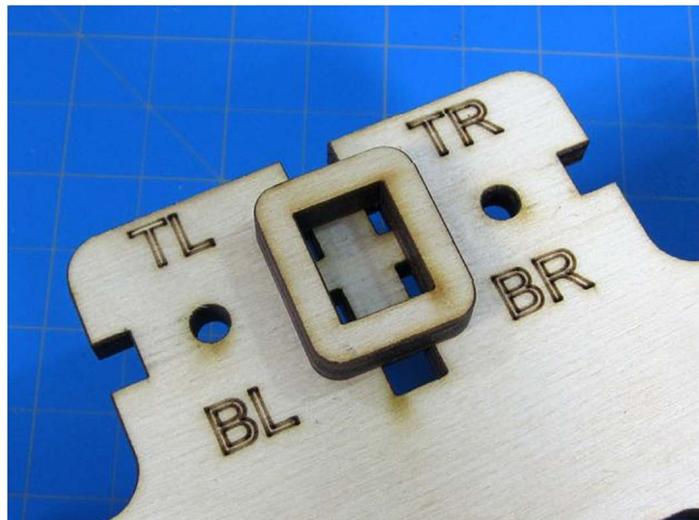
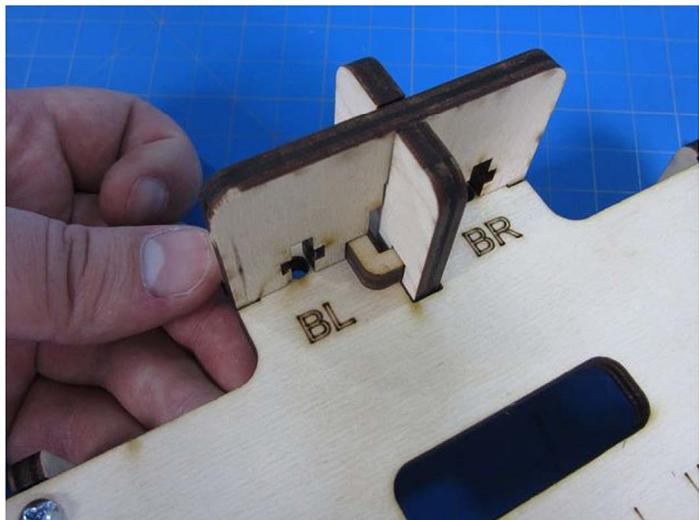
Put the face plate on the table in front of you so that you can read the words. Attach the triangle wing piece with the servo arm on the right side, and the other triangle wing piece onto the left side. We want the plastic servo arm to be facing the inside.

Use four screws and nuts to hold that together.

Now use the three remaining pieces to build the sensor divider. Rounded square first, then the tall skinny piece, and finally the longer piece with the two screw T slots. Once it's all together use two screws and nuts to secure it.







Step 12: Build the Center

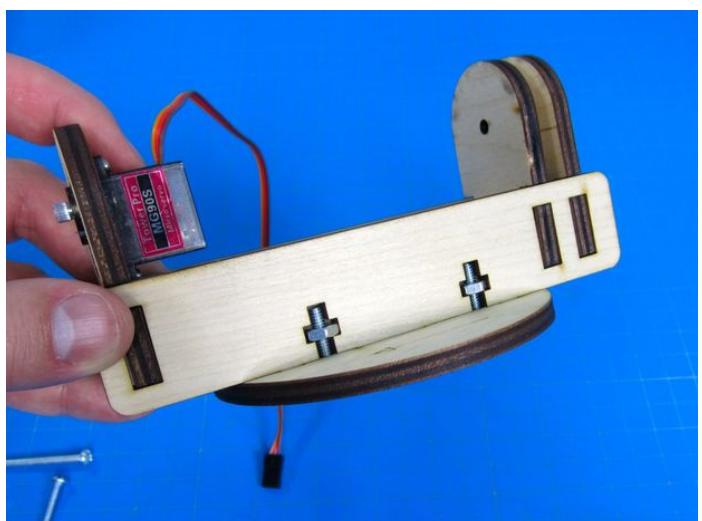
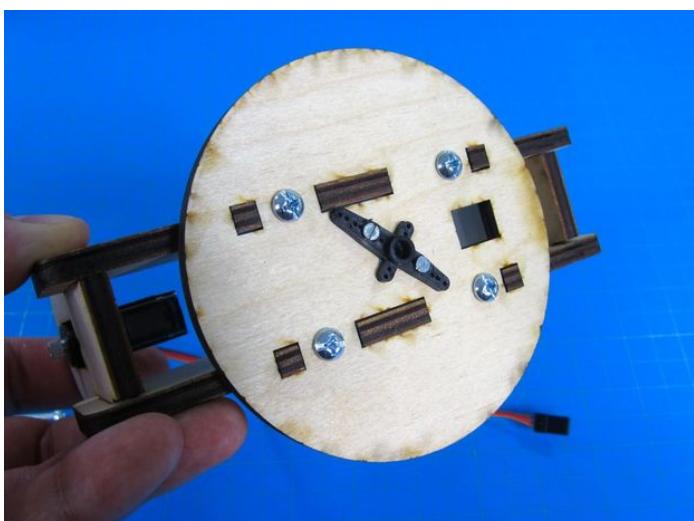
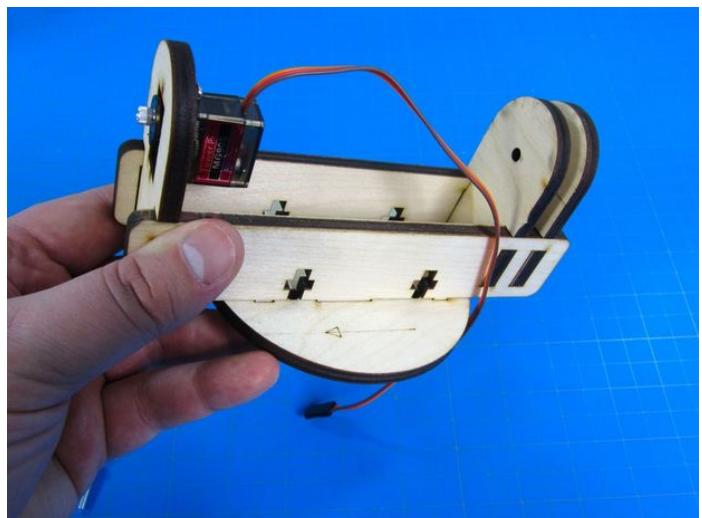
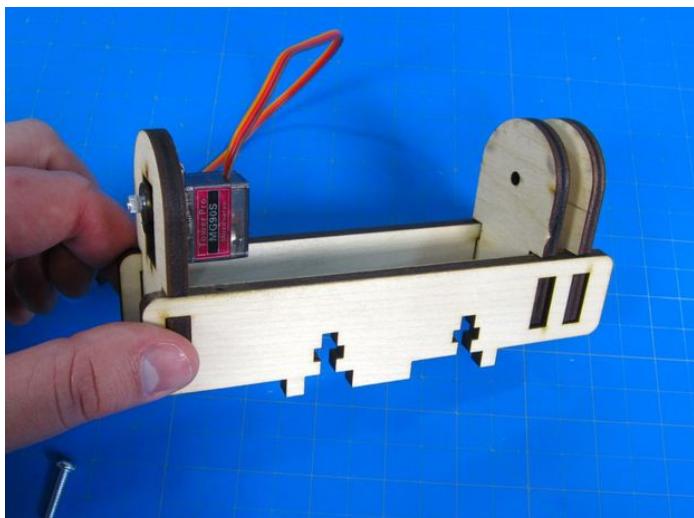
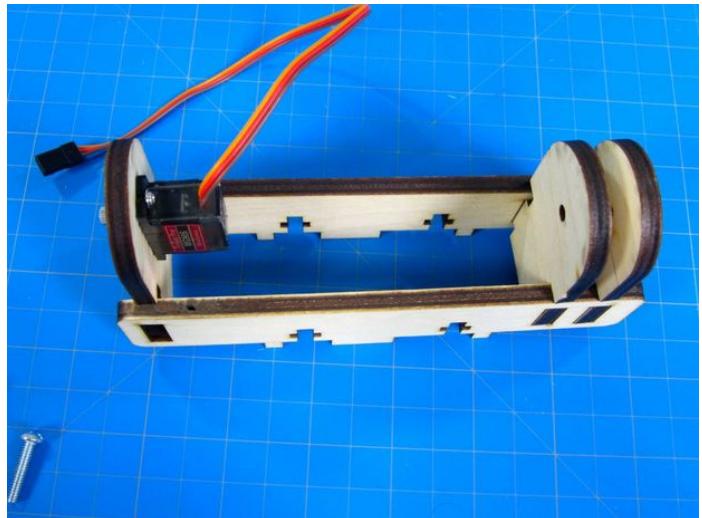
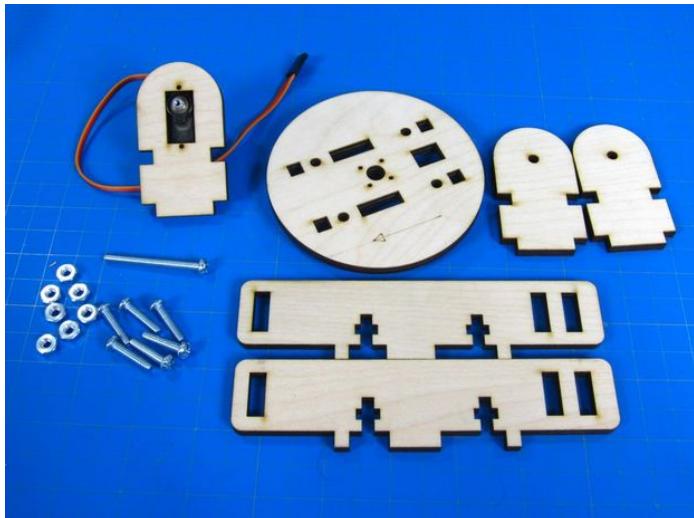
Gather all your remaining pieces.

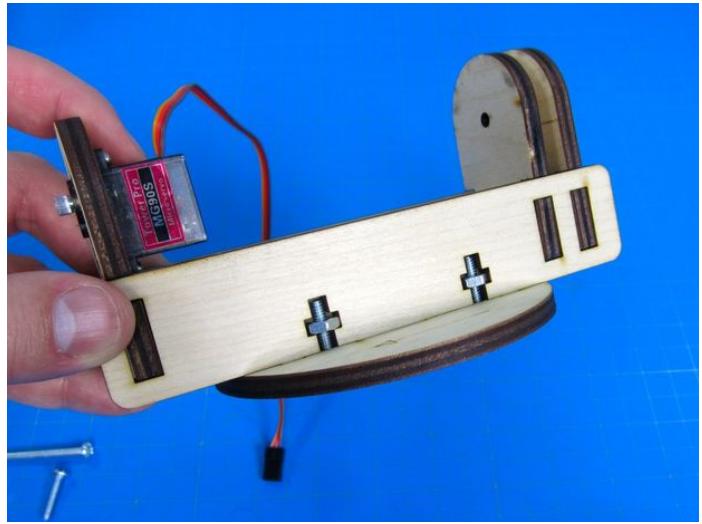
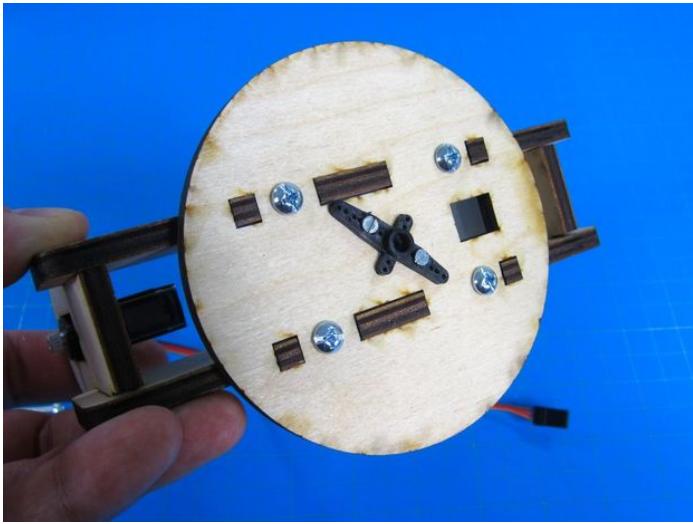
Take the two long pieces, the second servo mount, and the two other pieces that look like your **Servo** mount. Pop them together, and then put them into place on the round board.

Pay attention. The **Servo** should be on the "inside", and the arrow side of the round piece should be facing up.

At this point you should have three wooden structures assembled independently of each other, one very large and long screw, and the two little servo machine screws remaining.

DO NOT attach everything together yet, we need to Home our **Servos** first.





Step 13: Home the Base Servo

IMPORTANT: The Servos must be Homed and in correct positions, otherwise they'll be "backwards" compared to what the software is expecting.

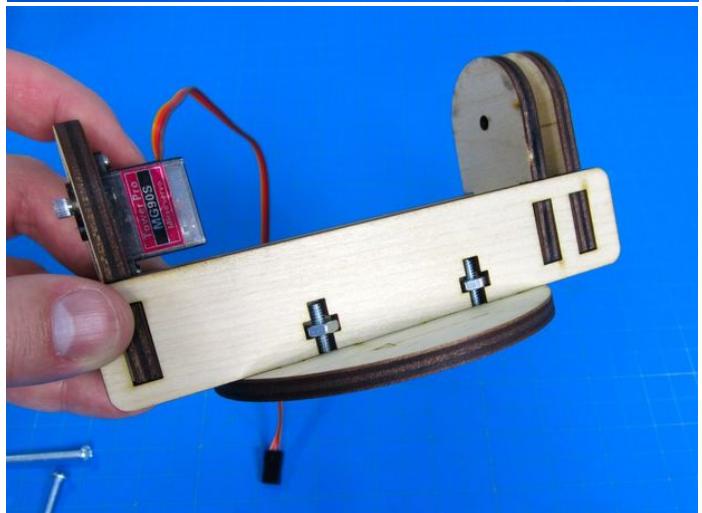
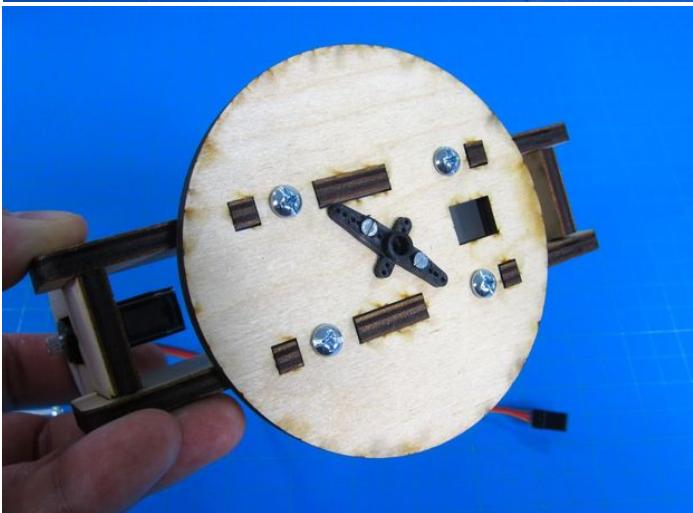
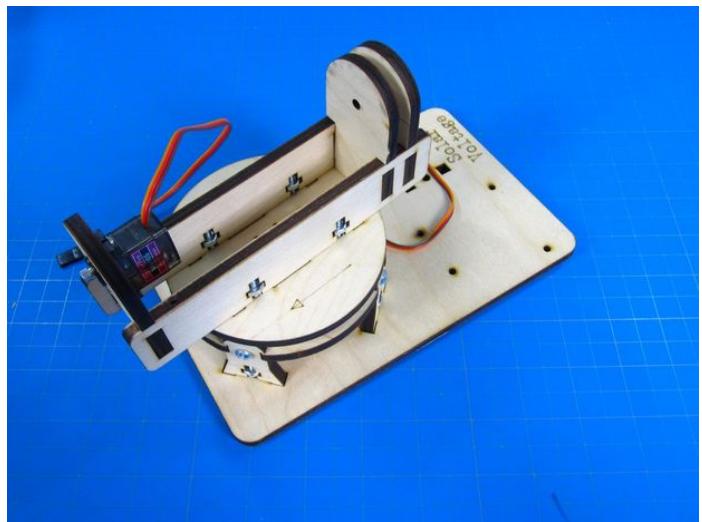
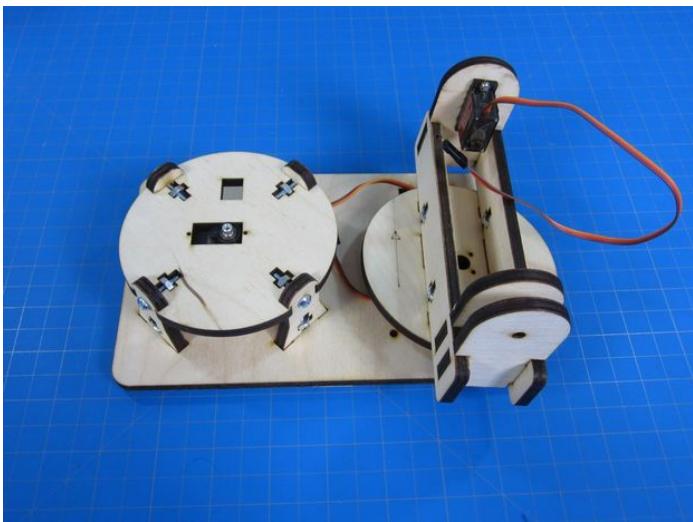
Servos move in 180 degrees. The **Servo** knows where "zero" degree is and where "180" degree is. Since we don't want or need full 180 degree range on our servos (we'd be hitting wood or the electronics) we want to set our "zero" degree to some very specific locations.

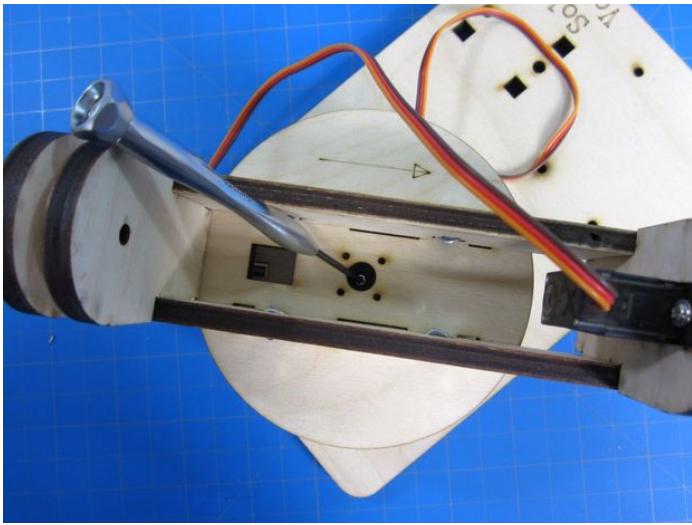
Start with the **Base** plate **Servo**. Without using the little screws (not yet!) push the **Servo Arm** that's attached to the Center into the servo. This may take a little effort, so you may wish to brace the servo with your other hand. Once together, slowly rotate the **Center** counter clockwise until there servo stops. This is "zero" degrees on the **Servo**.

(If you're doing this from scratch with plastic gear servos, be super careful as you can break the internal gears.)

Now take **Center** off your **Servo**. Align **Center** in a similar configuration to the below picture. As you can see, the second **Servo** is near where our **Arduino** will go, and the **Center** is at a 45 degree angle compared to the **Base**.

Use one of the two small **Servo Machine Screws** to secure the **Center** and **Base** together.





Step 14: Home the Center Servo

Same process as before. Push the **Servo Arm** attached to the **Top** into the **Center Servo**.

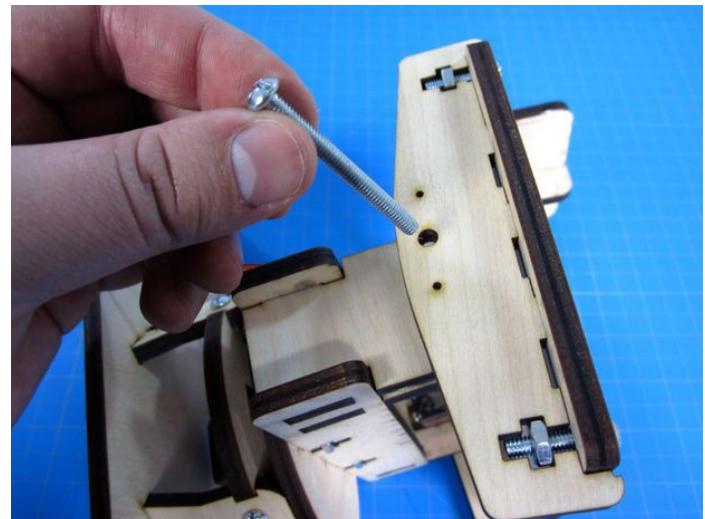
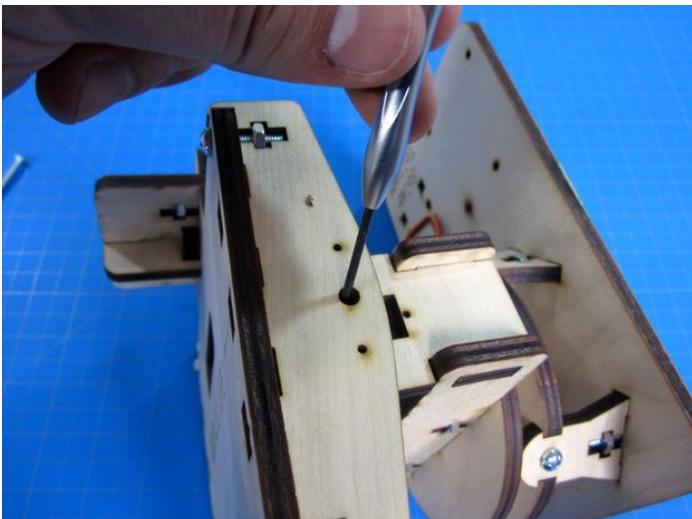
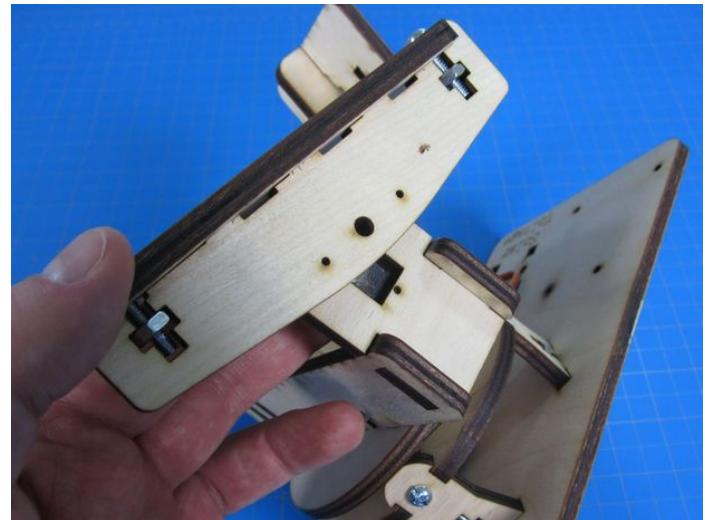
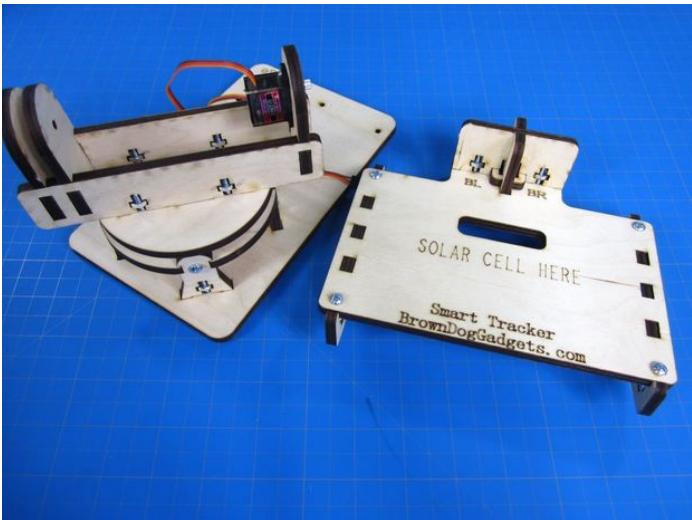
Turn it counter clockwise until it stops.

Disconnect the two.

We want the **Top** to be, more or less, parallel to the **Base**. Attach the **Top** as shown in the below pictures.

Use the final small **Servo Machine Screw** to connect the **Top** to the **Center**.

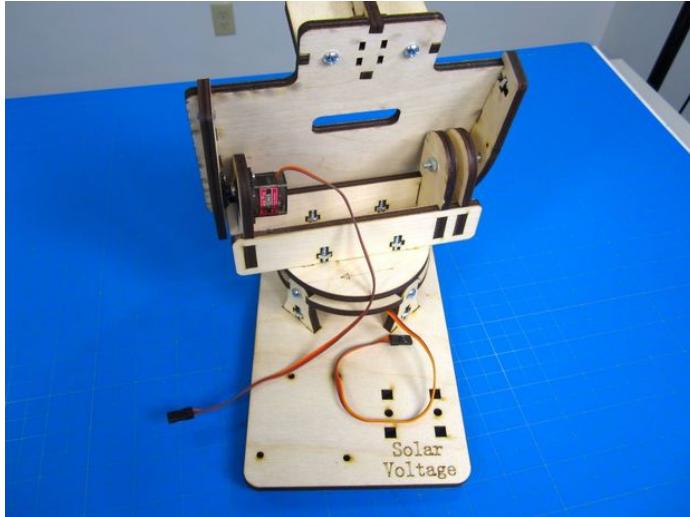
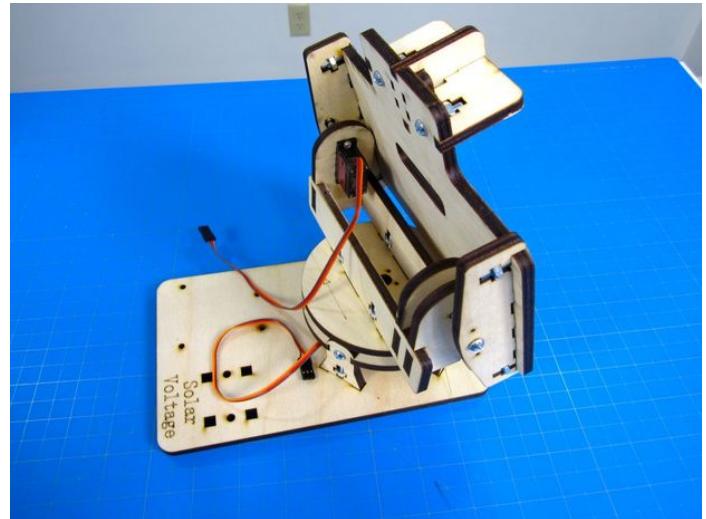
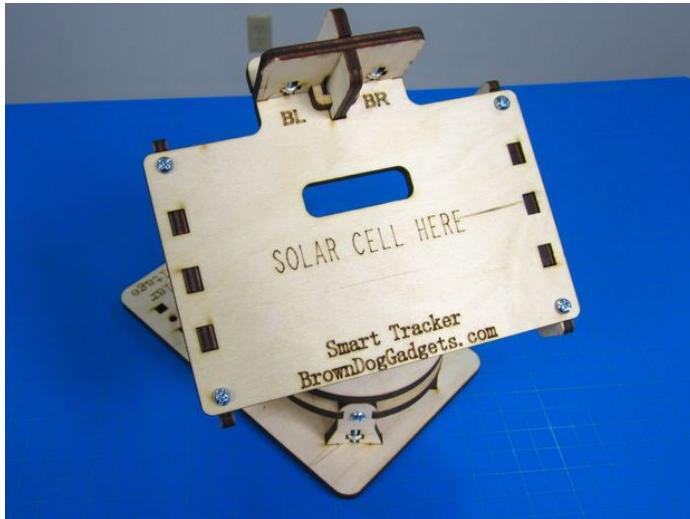
Use the **Long Screw** and final large nut on the other side. The nut isn't overly necessary, but it prevents the **Long Screw** from falling out.



Step 15: Double Check Everything

You're done building the body for this project. Double check the following things.

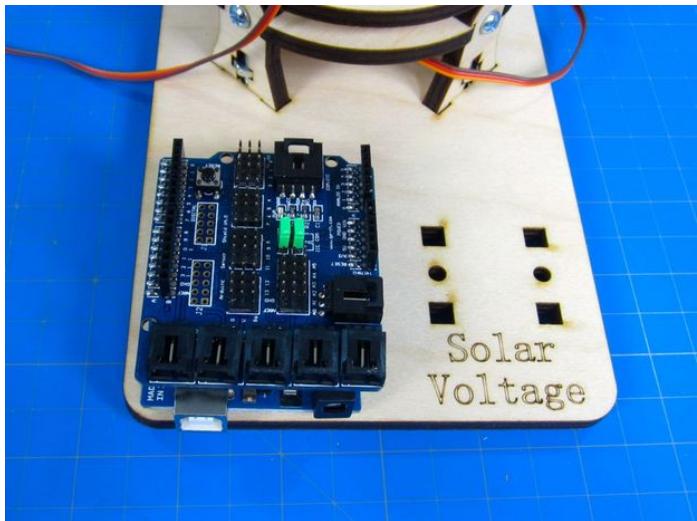
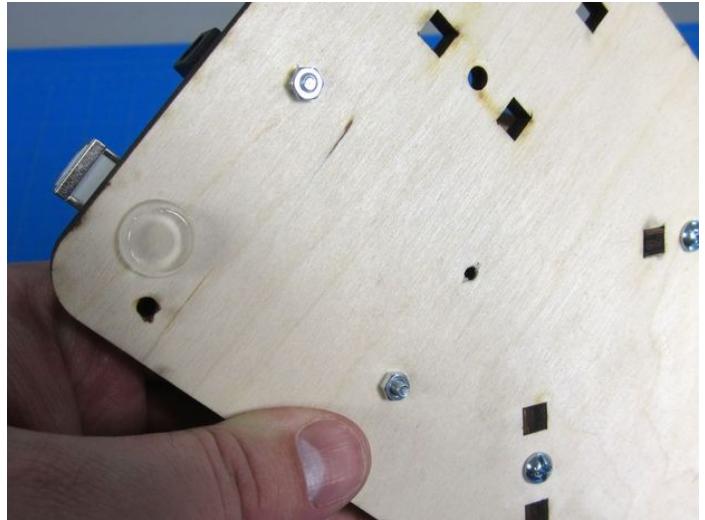
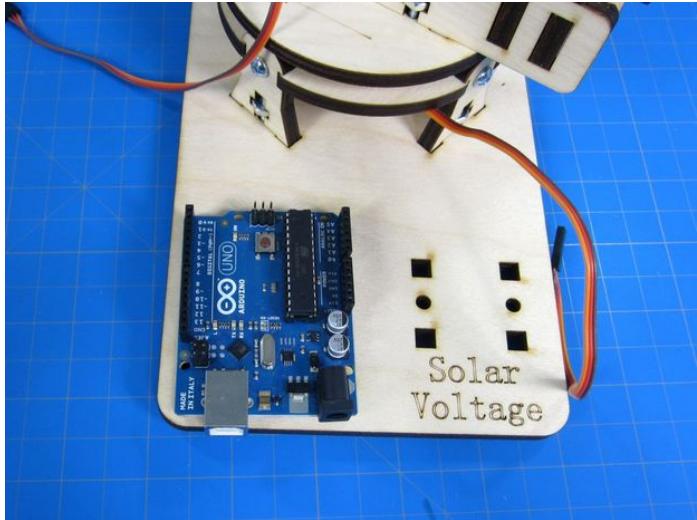
- 1) That your two **Servos** have the appropriate Home (zero) position.
- 2) All your screws are tight enough.
- 3) Things move decently easy. Just slowly move the servos back and forth. Do you hear any weird grinding sounds? Are any wooden pieces oddly rubbing together because you tightened things TOO much?
- 4) Did you put rubber feet on? You really should have by now otherwise you'll scratch up your table.



Step 16: Attach the Arduino

Screw the **Arduino** into place using at least 2 of the M3 Screws and Nuts. You can use all 4 if you'd like, but one in front and one in back should do it.

You can also attach the **Sensor Shield** at this time. Be careful not to break the pins on the Sensor Shield, but it might be necessary to slightly bend them into place.



Step 17: Assemble the Sensors

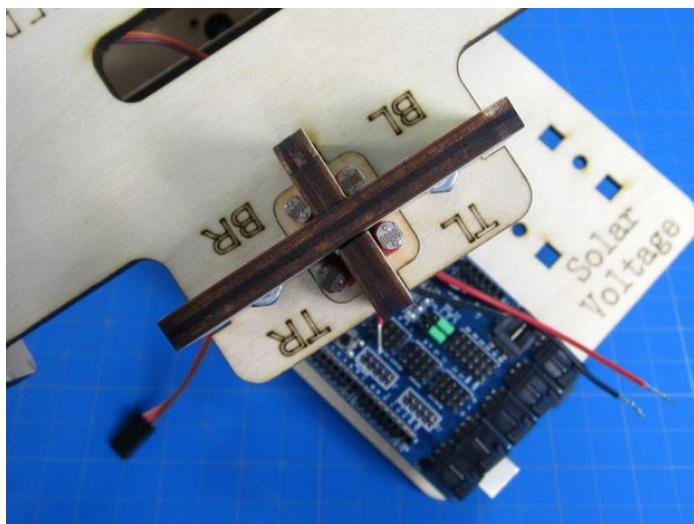
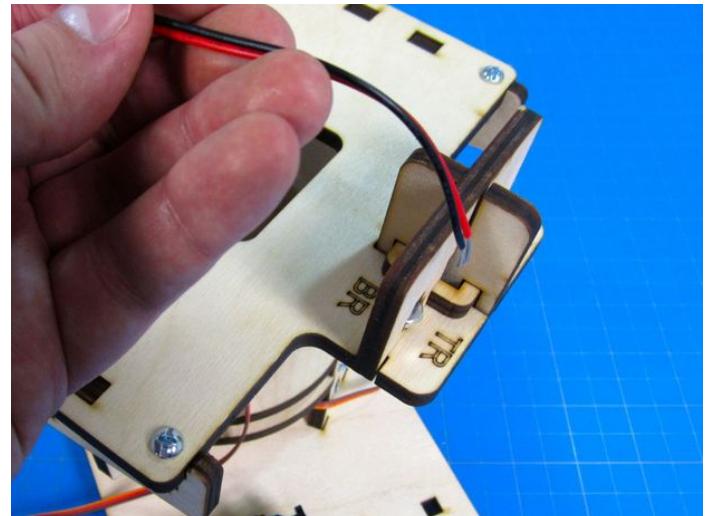
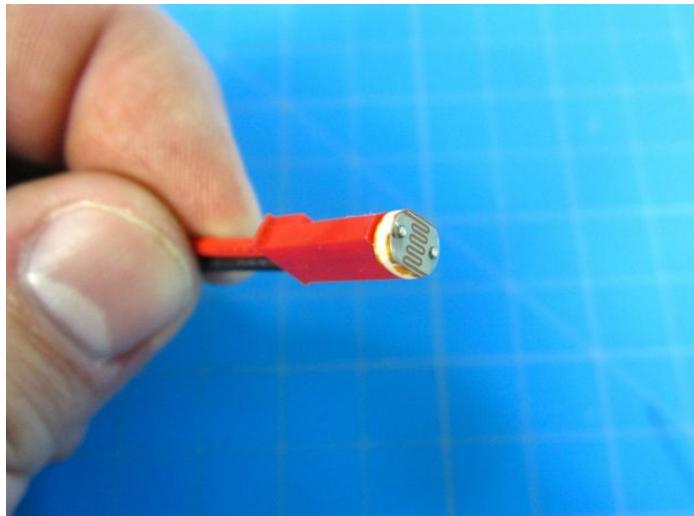
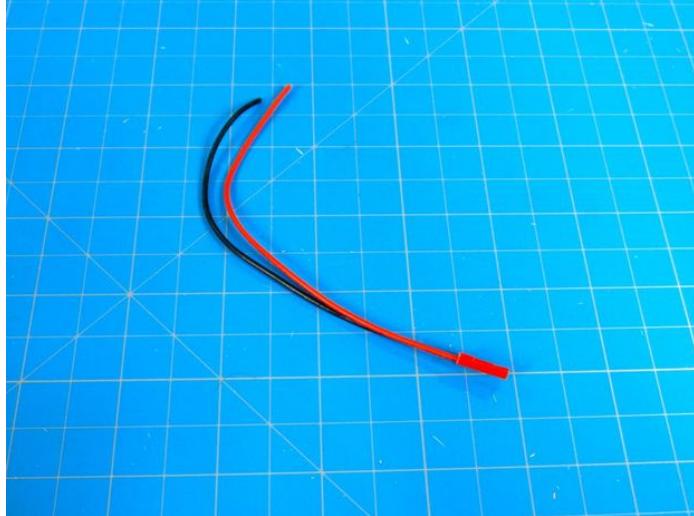
Take your four **Female JST Connectors**. Snip off the ends of the wires and then strip the wires.

(It's often necessary to snip the tips in order to strip the wires properly. We recommend you just do it by default.)

Grab your four **Light Sensitive Resistors**. The legs are way too long. Remove 2/3rds of their legs.

Push one **Light Sensitive Resistor** into each of the four **Female JST Connectors**. They should go in easily. Thread one **Female JST Connector** through each of the four holes around the **Sensor Divider**.

You should now have 8 wires hanging down through the **Top** around the **Sensor Divider**.



Step 18: Hook Up the Terminal Block

Grab your **5 Port Terminal Block**.

Take all four red wires coming from your sensors, twist them together, and put them into the first spot on your Terminal Block. This is the common **Positive**.

Now be careful on this wiring. Take the black wire coming from your **Top Left** sensor (it's labeled TL) and put it into the second Terminal block hole.

Take **Bottom Left** and go into the third hole.

Top Right goes into the fourth hole.

Top Left goes into the fifth hole.

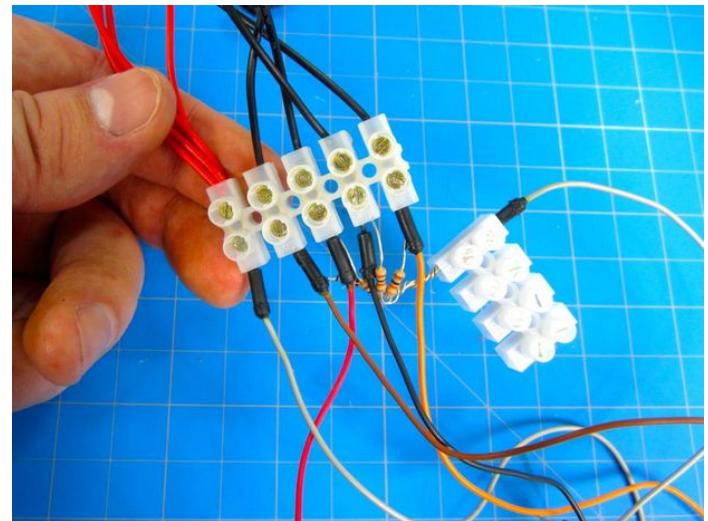
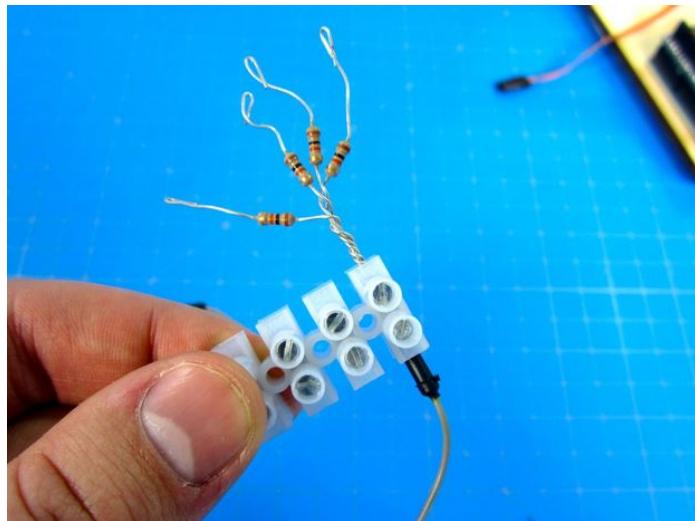
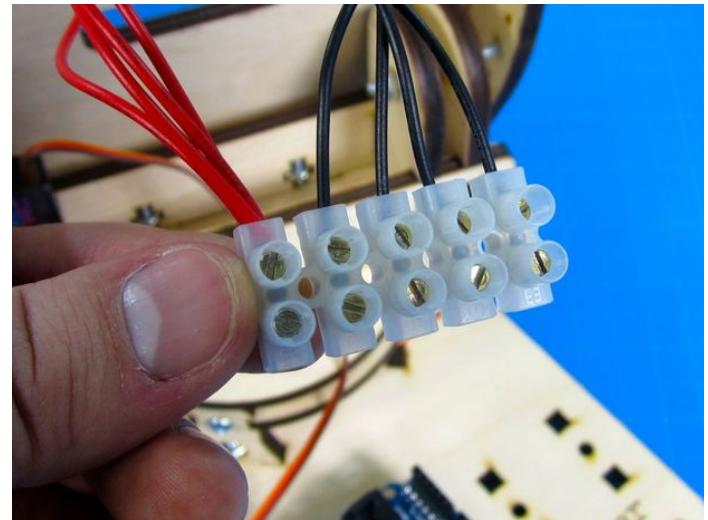
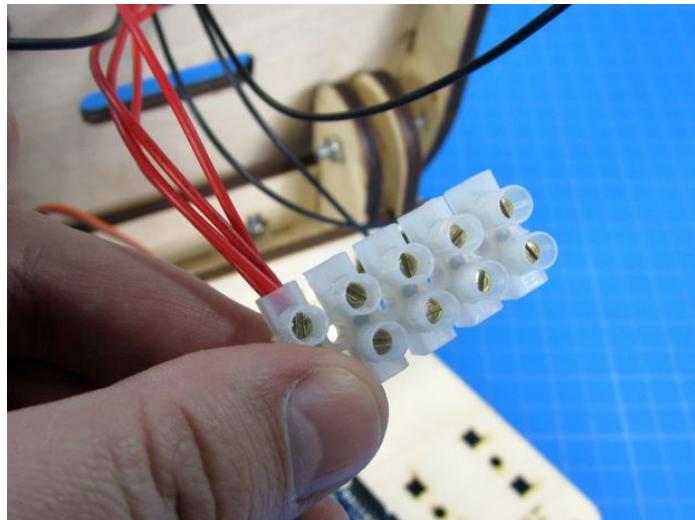
Make sure they're all screwed down tight.

Grab your **4 Port Terminal Block** and your four **10,000ohm Resistors**.

Twist all four **Resistors** together and put them into one of the holes on the **4 Port Block**. In the opposite hole screw in a **Jumper**. That jumper is our common **Negative**.

Now grab four **Jumpers**. Each of these **Jumpers** will share a hole with a **Resistor** on our **5 Port Block**. Stick a **Resistor** end and a Jumper into opposite hole of each of the four Black Wire holes.

You'll also want to use your last remaining **Jumper** in the opposite hole to your **Common Positive** (with all the red wires) in your **5 Port Terminal Block**.



Step 19: Hook Everything to the Arduino

We've also included a nice diagram for wiring if you need help.

Attach the Servos to your Arduino. This is easy to do via Shield's many inner pin rows.

The **Bottom Servo** goes to Pin 10.

The **Top Servo** goes to Pin 9.

Make sure you have them hooked up correctly with Red going to Positive, Black to Negative, and Yellow to Signal.

Now we're going to hook up all six Jumpers coming off our **Terminal Blocks**.

Common Negative (from the Five Port Block) can go to any of the **GND** (Ground) pins.

Common Positive (from the four port block) can go to any of the **5V** pins.

Look at your 5 Port Block. It should be in this order from left to right.

Common Negative. TL. BL. TR. BR.

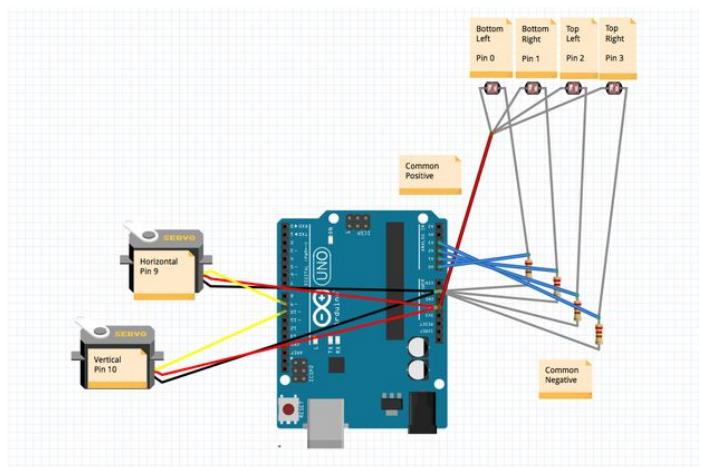
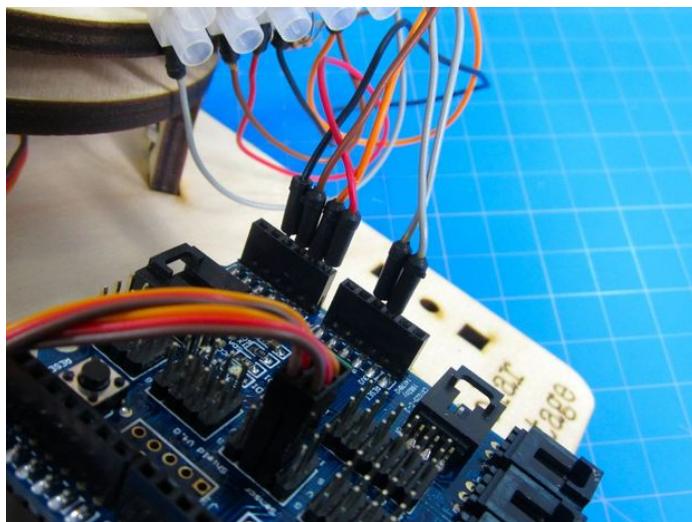
TL goes to Pin 2

BL goes to Pin 0

TR goes to Pin 3

BR goes to Pin 1

You can always change which pin goes where in the Code.



Step 20: The Code

Upload this code to your **Arduino**.

If you've not done so, grab the free Arduino install. When in the program be sure to set your board type to an Uno. (Or whatever version you're using.) You can easily modify the speed and range of the servos, the sensitivity of the sensors, from within the code.

```
#include // include Servo library
// 180 horizontal MAX
Servo horizontal; // horizontal servo
int servoh = 180; // 90; // stand horizontal servo
int servohLimitHigh = 180;
int servohLimitLow = 65;
// 65 degrees MAX
Servo vertical; // vertical servo
int servov = 45; // 90; // stand vertical servo
int servovLimitHigh = 80;
int servovLimitLow = 15;
// LDR pin connections
// name = analogpin;
int ldrlt = 0; //LDR top left - BOTTOM LEFT <--- BDG
int ldrrt = 1; //LDR top right - BOTTOM RIGHT
int ldrld = 2; //LDR down left - TOP LEFT
int ldrdd = 3; //ldr down right - TOP RIGHT
void setup()
{ Serial.begin(9600);
// servo connections
// name.attach(pin);
horizontal.attach(9);
vertical.attach(10);
horizontal.write(180);
vertical.write(45);
delay(3000);
}
void loop()
{ int lt = analogRead(ldrlt); // top left
int rt = analogRead(ldrrt); // top right
int ld = analogRead(ldrld); // down left
```

```

int rd = analogRead(ldrrd); // down right
// int dtime = analogRead(4)/20; // read potentiometers
// int tol = analogRead(5)/4;
int dtime = 10; int tol = 50;
int avt = (lt + rt) / 2; // average value top
int avd = (ld + rd) / 2; // average value down
int avl = (lt + ld) / 2; // average value left
int avr = (rt + rd) / 2; // average value right
int dvert = avt - avd; // check the diffirence of up and down
int dhoriz = avl - avr;// check the diffirence og left and rigt
Serial.print(avt);
Serial.print(" ");
Serial.print(avd);
Serial.print(" ");
Serial.print(avl);
Serial.print(" ");
Serial.print(avr);
Serial.print(" ");
Serial.print(dtime);
Serial.print(" ");
Serial.print(tol);
Serial.println(" ");

if (-1*tol > dvert || dvert > tol) // check if the diffirence is in the tolerance else change vertical angle
{
    if (avt > avd)
    {
        servov = ++servov;
        if (servov > servovLimitHigh)
        {
            servov = servovLimitHigh;
        }
    }
    else if (avt < avd)
    {
        servov= --servov;
        if (servov < servovLimitLow)
        {
            servov = servovLimitLow;
        }
    }
    vertical.write(servov);
}

if (-1*tol > dhoriz || dhoriz > tol) // check if the diffirence is in the tolerance else change horizontal angle { if (avl > avr)

```

```
{
servoh = --servoh;

if (servoh < servohLimitLow){ servoh = servohLimitLow;
}

}

else if (avl < avr)
{
servoh = ++servoh;

if (servoh > servohLimitHigh){ servoh = servohLimitHigh;
}

}

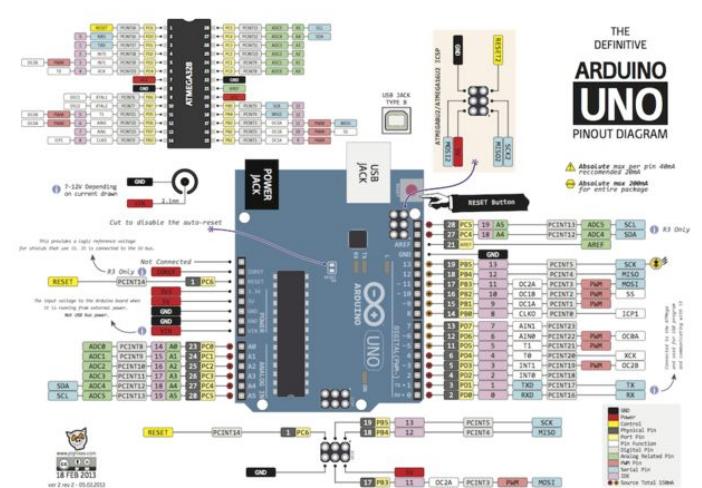
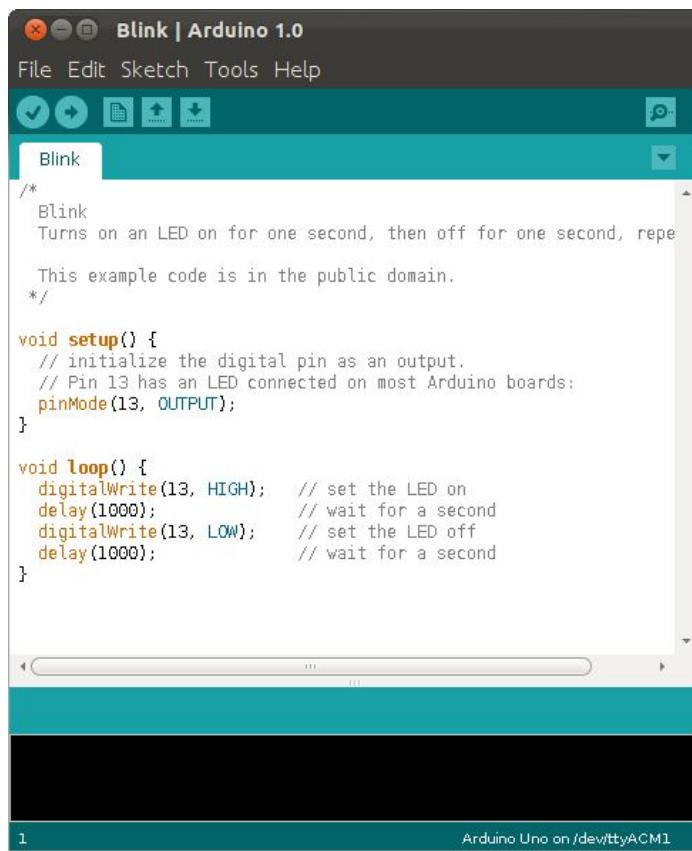
else if (avl = avr)
{
// nothing
}

horizontal.write(servoh);

}

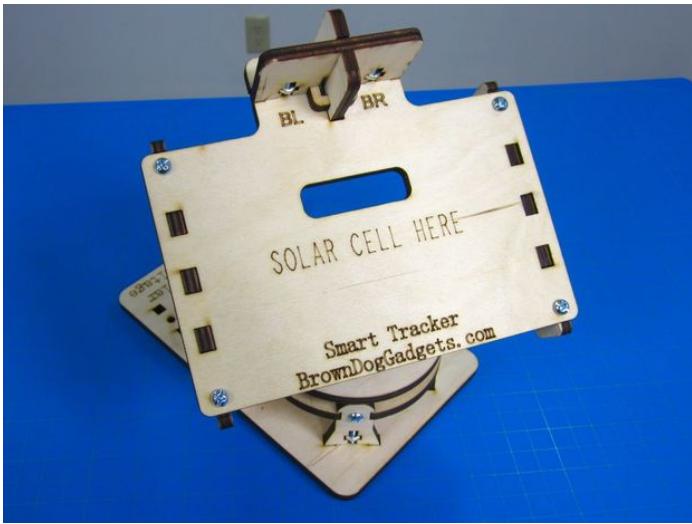
delay(dtime);

}
}
```



Step 21: Common Problems

- 1) **Servos are making a horrible noise or are pressing wood together.** - This is probably because your Servos are not homed properly. You can easily redo the homing process by undoing a couple of screws.
- 2) **Not tracking properly.** - Your jumpers are probably wrong. Make sure the right sensors are going to the right pins on your Arduino. You can also change the sensitivity if need be within the code.
- 3) **Nothing is happening.** - Well, did you hook up a power source? The Arduino Uno runs off USB.
- 4) **Nothing is happening.** - Did you upload your code? Unless you uploaded it the Arduino won't know what to do.
- 5) **I smell smoke.** - Disconnect your power. Something is super wrong.
- 6) **My Servo isn't moving at all, or is making a really bad metal on metal sound.** - While it's rare to have a bad Metal Gear Servo, it can happen. Double check to make sure it isn't the homing process and that the servo isn't doing anything, and then replace the servo.



Step 22: Extra Add Ons

Cable Wrap makes everything better. A couple of lengths will help make everything less annoying. Zip Ties are also another great option.

Adding on a **Solar Cell** and **Volt Meter** makes this project even easier.

You can easily add on a 5.5V 320mA Solar Cell to the top of the Tracker using Foam Tape.

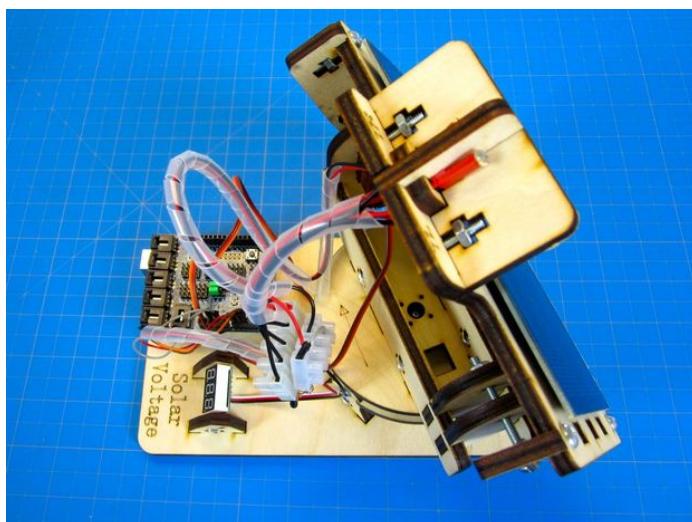
We've included a spot on the Tracker for a small **LED Volt Meter** as well via two wooden mounts.

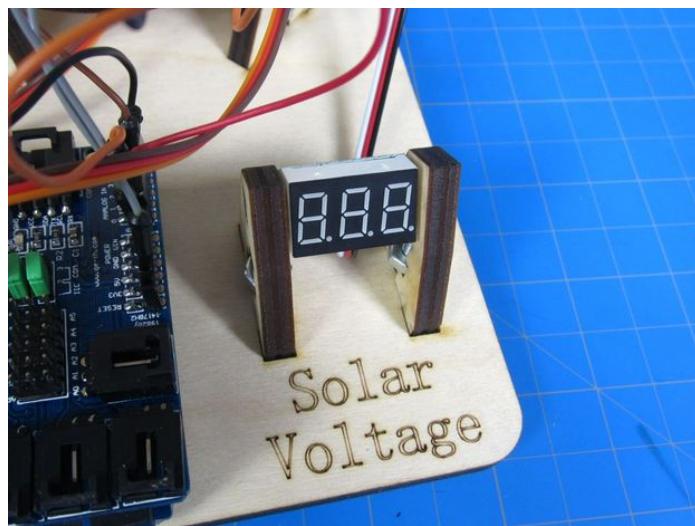
Hook the **Volt Meter** and **Solar Cell** together via the unused spots on the **4 Port Terminal Block**.

The **Volt Meter**'s White and Red Wires connect to the Red (Positive) wire from the Solar Cell. Then connect the Black wire from the **Solar Cell** and **Volt Meter**.

The **Volt Meter** we're using is powered directly from the **Solar Cell**. The upside to this is that you don't need extra batteries or **Jumpers**. The downside is that if the power from the **Solar Cell** dips below 3V the LEDs will turn off.

If this is a problem for you use power from the **Arduino** to run the **LED Volt Meter**. You'll just need two **Jumpers**. White **Volt Meter** wire goes to **Arduino** 3V, and you'll need a **Jumper** to go from an **Arduino** GND and connect to the black wires from the **Solar Cell** and **Volt Meter**.





Step 23: Finished

If everything is working for you, then you're done!

Stick your New Solar Tracker in a window or use a flashlight to see it in action.

While it is super easy to just get all the parts for this kit from various sources, you can also grab a full kit from us as well.

While this build is pretty easy overall, we do understand that many electronics newbies might get a bit overwhelmed with the whole wiring part. We're working on making a simple Arduino Shield for an even easier wiring job.

