SR6 Build Instructions

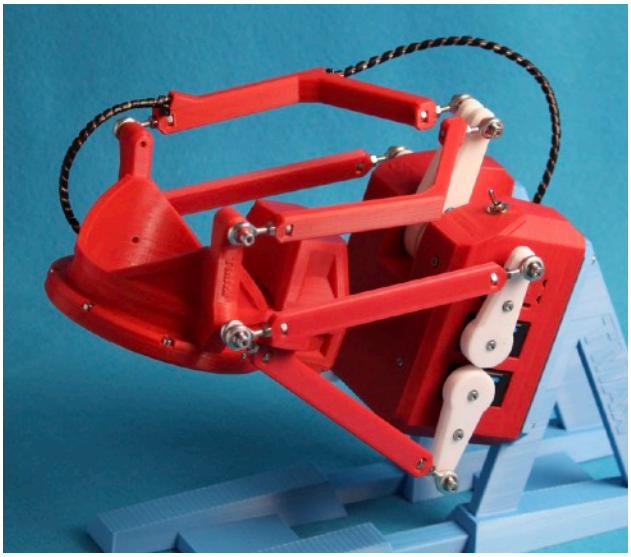
TEMPEST
TMAx
MULTI-AXIS

An assembly guide for the Stroker Robot, 6-Axis

June 2022

Introduction

SR6 ALPHA DEC 2020



SR6 BETA JUNE 2022

The SR6 is the machine I always wanted to make. I first considered the possibility of a 6-axis machine back when I was still playing with laser cut plywood back in the OSR3 days. Anyone familiar with the OSR2, and OSR2+, will appreciate the extra nuance that comes from adding each new movement axis to a stoker robot. Having all six: up/down, left/right, forward/backward, pitch, roll and twist is therefore the obvious destination of that train of thought!

The first SR6 was a set of loose brackets mounted to a plate that I used to work out the arm configuration. This led to the Alpha-release, which was a functional but very user unfriendly design. This design, Beta, is a clean-sheet redesign of the SR6 structure that offers a lot of advantages over the older designs. It doesn't just look a lot sleeker, the biggest improvement is the new frame-based structure, which makes disassembly and repairs a lot easier.

SR6 is not intended as a replacement for OSR2, rather I see it as an enthusiast's machine; it's for users who want that little bit extra. I intend to continue to support both, and the fact that the control protocol (T-Code) is the same for both should hopefully mean that this is also true for the growing number of software developers who support open source stroker robots.

This document contains everything you will need to know to put together an SR6 Beta: a full list of parts, step-by-step build instructions, and setup notes.

The SR6 is a more complicated machine than the OSR2, and it requires a greater level of confidence and skill to put together. I therefore highly recommend that anybody who is new to this kind of project build an OSR2 first before progressing to an SR6. The OSR2's relative simplicity, and its modularity, makes it a much better first time project. it also uses a lot of the same off-the-shelf parts, so those can be re-used later on in an SR6 build.

Lastly I want to say thanks to all of my supporters. A lot of time, energy and resources went into the creation of this design and I could not have made it a reality without your help.

Happy building!

Tempest (31/01/2022)

Table of contents

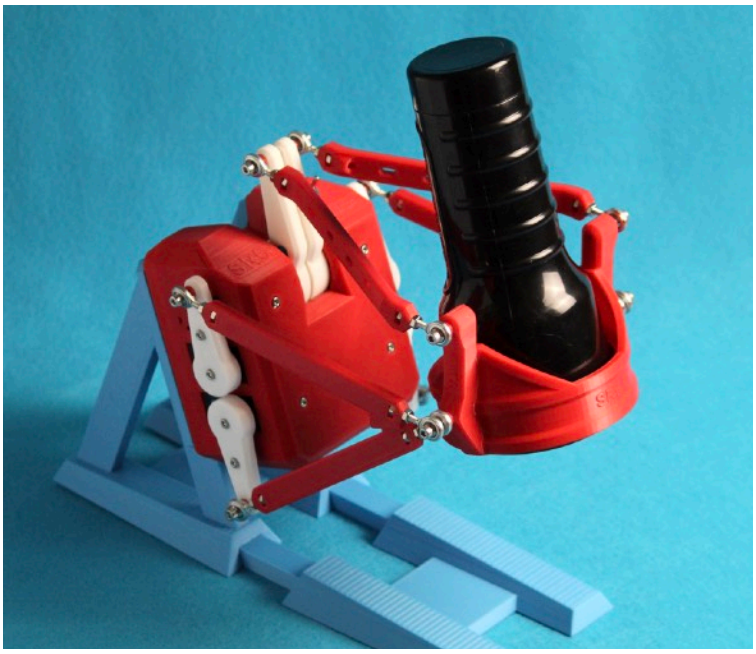
Record of changes

Issue	Date	Changes
SR6 Beta 1	31-1-2022	First issue
SR6 June ‘22	9-6-2022	Document updated: • New swanky cover page added • Twist receiver hardware and instructions updated to T-wist mk4 Added to document: • Table of contents • Record of changes • Trays & shields section

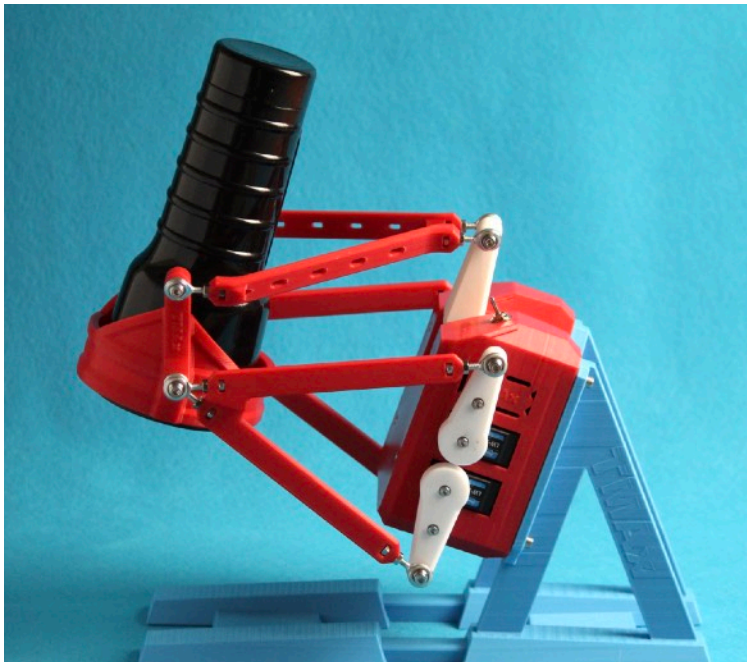
Description

The SR6 is a Multi-Axis Stroker Robot (MAxSR) designed to hold and move a standard fleshlight or similar toy with six degrees of freedom. That means that it can move up and down, left and right, forward and backward, and it can rotate to roll left and right, pitch forward and backward, and with the twist receiver upgrade it can also rotate around the up-down axis.

It is constructed from a combination of 3D printed plastic and commercial off-the-shelf components. It is designed to be home built, modifiable, and user-repairable.



The design is based around a modified Stewart platform with six servos driving six arms, which connect to the receiver via six linkages. The arrangement gives a theoretical total movement range of 120mm in the up-down direction and 60mm in the left-right and forward-backward directions. The receiver can simultaneously also rotate in roll and pitch by approximately ±30 degrees.

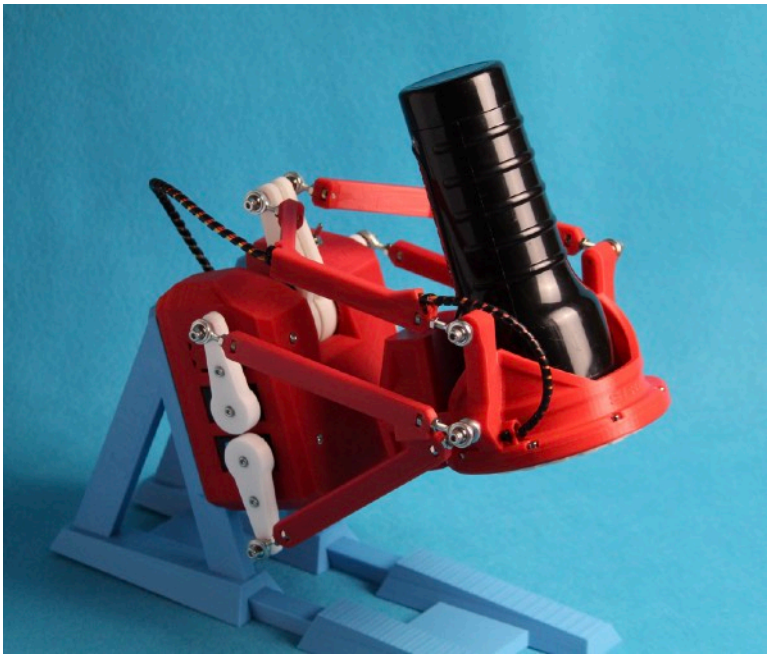


The receiver itself is not set up to twist via the main servos as this would only achieve very small angles. Instead, an optional twist receiver can be installed, which is driven by an additional servo. This can rotate the toy mounted in the receiver up to approximately 270° of rotation (±135°) around the up-down axis.

SR6 TWIST RECEIVER

Control is via a generic open source USB serial protocol called T-Code. The SR6 uses an ESP32 microcontroller to receive these simple commands and calculate the appropriate control angles for each of the servos to achieve the desired position for the receiver.

Like OSR2 the SR6 is a hands-free machine that mounts to a standard VESA 100mm hole pattern using 4x M4 bolts. This means that an enormous range of mounting options, mostly in the form of display and monitor stands, are commercially available. There are also a couple of 3D-printable desk and chair clamp options available on my Patreon/SubscribeStar/Discord platforms.



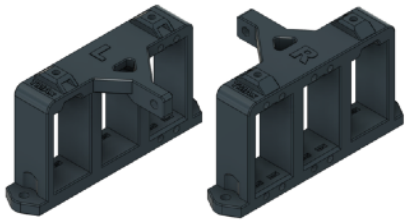
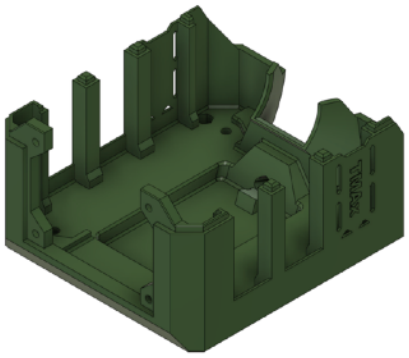
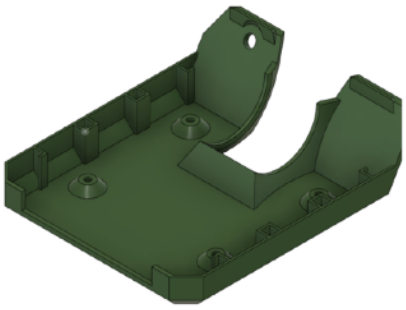
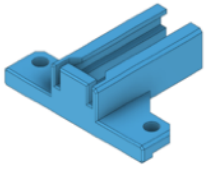
FULL SR6 WITH TWIST

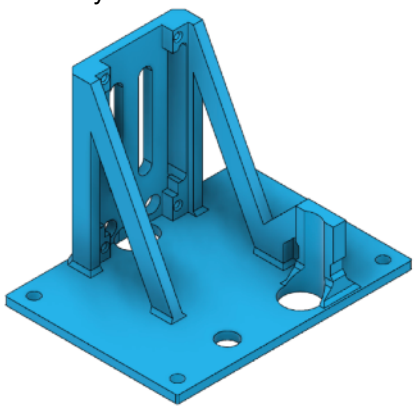
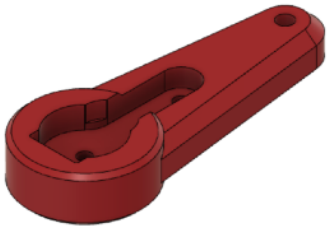
SR6 parts list

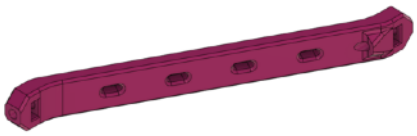
3D printed parts list

This is a list of the 3D printed parts required to build a basic SR6 Beta. See the separate section for the parts to build the twist receiver.

The 3D printed parts are supplied in .STL format; you can print them yourself or order them from an online 3D printing service. Unless otherwise specified the STL files are provided in the intended build orientation and are designed to be printed without supports.

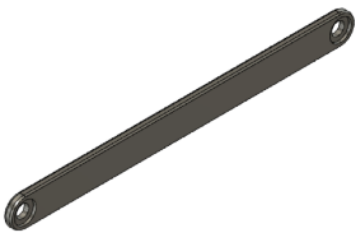
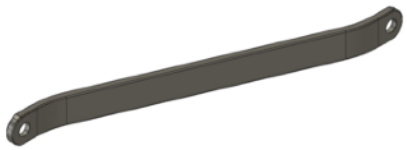
Part	Qty	Description
SR6 Frame - (Left & Right) 	1x	This is a two-part structure that holds the SR6's six main servos. The two halves are different, and marked "L" and "R".
SR6 Base 	1x	This part forms the majority of the enclosure for the SR6's electronics and servos. It also forms the main structural interface between the frame and the VESA mount.
SR6 Lid 	1x	The cover that is screwed into place to complete the main enclosure. This part includes a mounting hole for an on/off switch.
Power Bus Holder 	1x	This is a small clip that is used to hold the power bus in position on top of the ESP32.

SR6 Tray 	1x	This is a removable part that screws into the bottom of the main enclosure. It holds the electronics and the default version is configured to mount an ESP32 DevKit v1, a female jack plug and a toggle switch. This part is intended to make customisation of the SR6 electronics easy, and as such there are various alternative community-designed tray options available on the discord server. Previous versions of this component were designed around the Romeo BLE mini microcontroller.
SR6 Receiver 	1x	The piece that is manipulated by the arms and holds your toy. The default configuration is designed to hold a full sized standard fleshlight case.
SR6 Main Arm 	4x	The main arms that are driven by the servos. They are designed to fit around Futaba 25T M3 metal servo horns.
SR6 Pitcher Arms (Left & Right) 	1x	The two arms driven by the upper servos. They are designed to fit around Futaba 25T M3 metal servo horns. The Left and Right arms are different; you will need one of each.

SR6 Bearing Main Link 	4x	These link the main arms to the lower mounts on the receiver. These parts are designed to be used with a 4mm male rod end bearing on each end.
SR6 Bearing Pitcher Link 	2x	These link the main arms to the lower mounts on the receiver. These parts are designed to be used with a 4mm male rod end bearing on each end.
4x3mm Spacer 	12x	These are small 3D printed rings that are used to maximise the range of movement of the rod-end bearings. They are not needed with the grommet-based links.

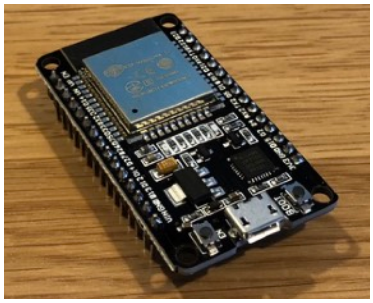
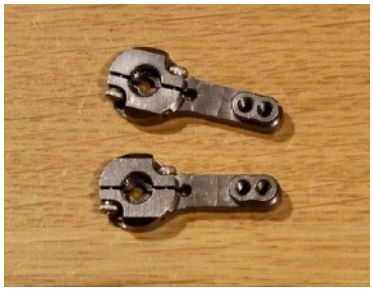
Alternate 3D printed parts

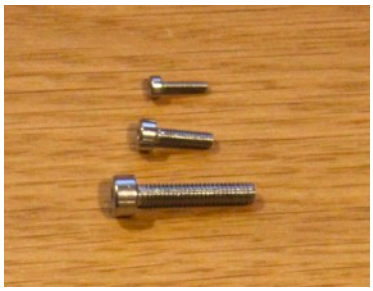
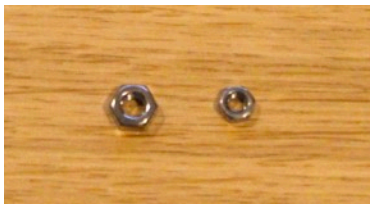
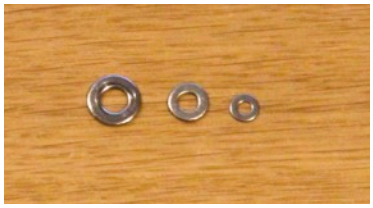
The link parts below are available as an option if you are unable to acquire the 4mm male rod end bearings required for the standard links. There is no need to use 4x3mm spacers with these parts.

SR6 Main Link (Alpha 1) 	4x	These link the main arms to the lower mounts on the receiver. These parts are designed to be used with a pair of 6mm wiring grommets each.
SR6 Pitcher Link (Alpha 1) 	2x	These link the pitcher arms to the upper mounts on the receiver. These parts are designed to be used with a pair of 6mm wiring grommets each.

Off-the-shelf parts list

These are the off-the-shelf components that you will need to assemble a basic SR6. Note that the power bus materials are listed separately. Likewise the twist receiver upgrade requires additional parts, not listed here.

	Part	Qty	Description
	ESP32 DevKit v1	1x	The SR6 Beta 1 uses the ESP32 DevKit v1 microcontroller.
	Micro USB cable	1x	To connect to your computer.
	Power supply	1x	A power supply capable of providing at least 6A (preferably 10A) at 5V or 6V.
	Standard size servo (20kg.cm or more)	6x	A matched set of six seem to work best.
	Futaba 25T M3 metal servo horn	6x	Be sure to purchase a set if your servos don't come with them.

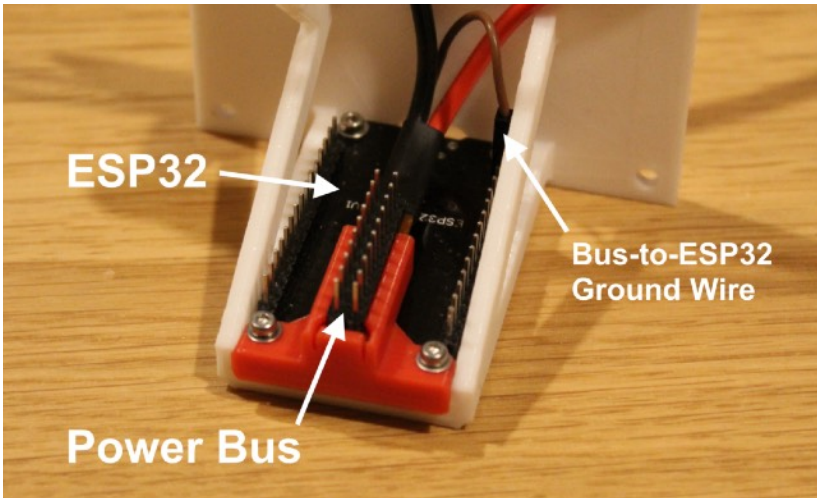
	Cable tie	1x	A normal cable tie. Used to secure the power bus barrel jack in position
	Bolts: M4x16mm M4x20mm M4x25mm M4x30mm M3x8mm M3x10mm M2x8mm M4x20mm*	 5x 6x 2x 2x 4x 44x 4x 4x*	These are metric threaded bolts. Any type is fine, but I recommend hexagon socket cap head bolts. (*Bolts for VESA mount)
	Nuts: M3 M4	 32x 33x	Metric threaded nuts.
	Washers: M2 M3 M4 M4*	 4x 36x 5x 4x*	Metric washers, Form A. (*Washers for VESA mount)
	4mm Rod end bearings	12x	Sometimes called “rose joints” or “fish eye bearings”. The type used have a threaded shaft with an M4 metric thread, and a 4mm hole. The wiring grommets below, with the alternative links, are an acceptable substitute if you have trouble sourcing these parts.
	**OR** 6mm (1/4”) wiring grommet	12x	These are an alternative way to create the flexible joints between the arms, linkages and receiver. Quality can vary, so I recommend buying more than you need. 6mm is the Outer Diameter (OD). The Inner Diameter (ID) should be 4mm.
	1x1 DuPont 0.1” Crimp Connector Housings	9x	Covers for the crimp connectors on the end of the servo signal wire ends. This prevents metal-to-metal contact between adjacent pins when they are plugged into the ESP32.

Assembly Steps

Power bus

On the OSR2 the external power bus is optional, on SR6 it is a must because of the amount of power drawn by the servos. In addition the ESP32 microcontroller does not have the same plug-and-play arrangement of servo power pins in addition to the signal pins.

A power bus is a neat arrangement that that allows you to plug the power and ground leads of the servos into a common power supply. The servo signal leads meanwhile can be plugged in to the appropriate pins on the ESP32. Note that the bus ground must also be connected to ground on the ESP32 to complete the circuit.



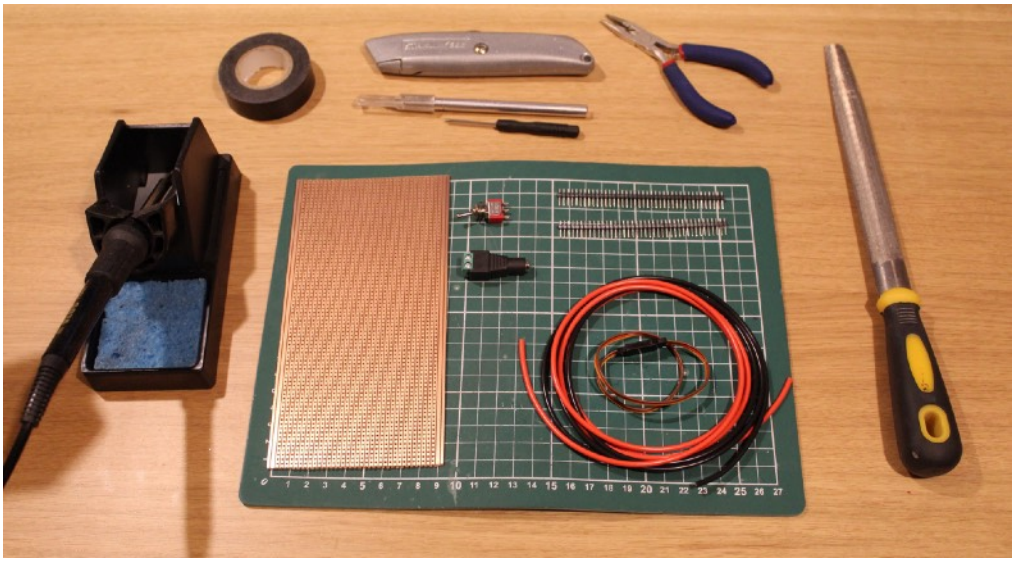
Creating a power bus is probably the most challenging part of assembling an SR6 for the uninitiated, which is why I have made it the first stage in these instructions. It requires a little soldering, but it's not too difficult and the parts used are inexpensive so it's quite acceptable to make several attempts if you have to.

Regarding alternatives: 3rd party power bus options, including some pre-made by members of the community, are available. Some members of the community also swear by Wago connectors. I suggest that you check out the discord community if you want to consider one of these other solutions.

These instructions on how to build an SR6 power bus are also available as a video tutorial. You can find this on my Patreon/SubscribeStar page or on the Tempest discord server.

To assemble a power bus you will need:

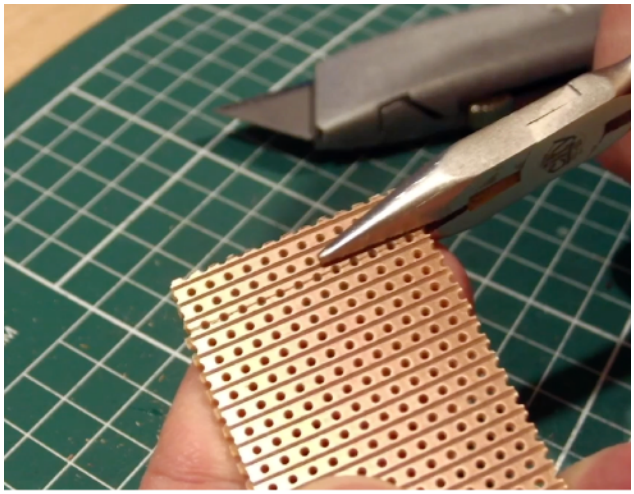
- 0.1” copper stripboard.
- 0.1” header pins
- A barrel jack power connector - compatible with your power supply.
- A toggle switch
- Red and Black multi-stranded cable (rated to a higher amperage “A” than your power supply!)
- Servo extension cable.



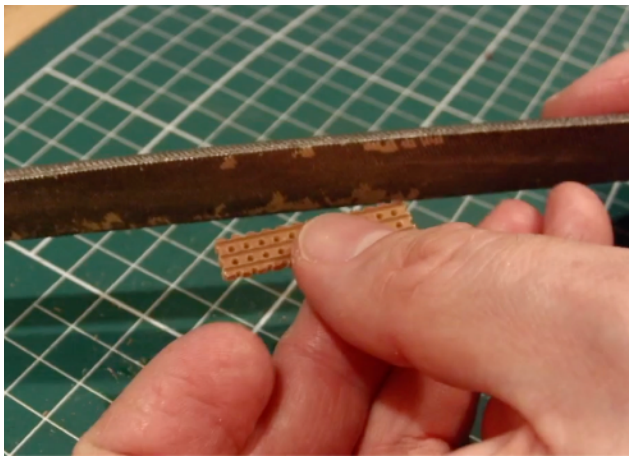
The tools that you will need are a soldering iron, a craft knife/knives, a file, pliers and electrical tape. You will also need a screwdriver if any of your components use screw connectors.

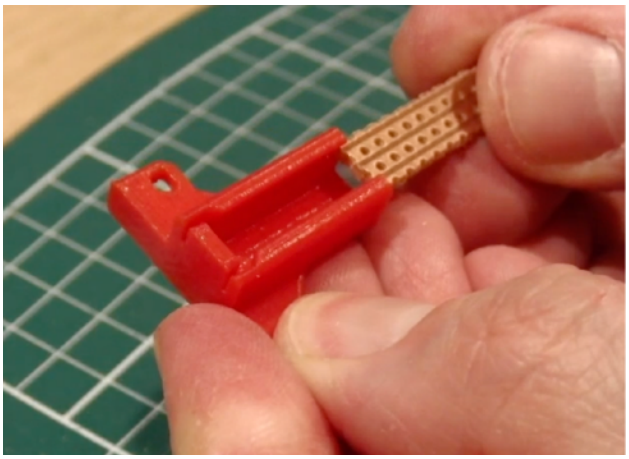
1. Take the copper strip board and cut out a section with two strips running lengthways with length to accommodate holes for pins for all of the servos or other devices you will need to power, plus two more holes. In this example I am going to assume 10 power pins (6x main servos, twist, valve, 2x future-proofing) so the strip should be 12 holes long.

To cut the board to size I recommend that you use a knife and score from hole to hole, then use pliers to break the board off. If you have scored the board well, only minimal force should be needed with the pliers, rocking back and forth.

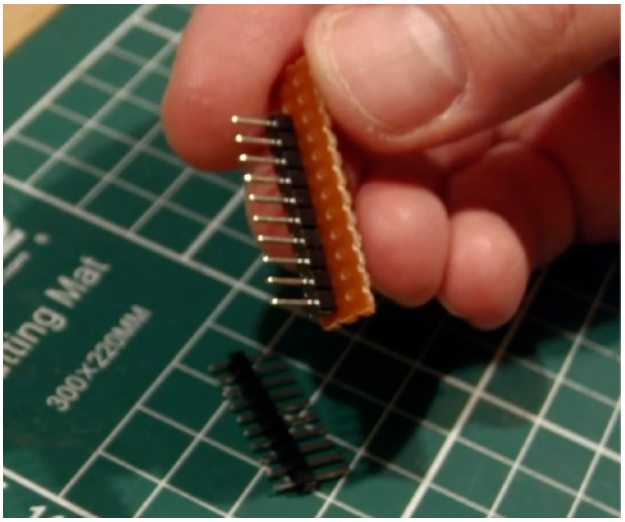


2. Check the broken off board against the end of the 3D printed Power Bus Holder part. If it doesn’t, use a file to shave off material from the sides and make the board narrower until you are able to slide it into the holder with a snug fit.

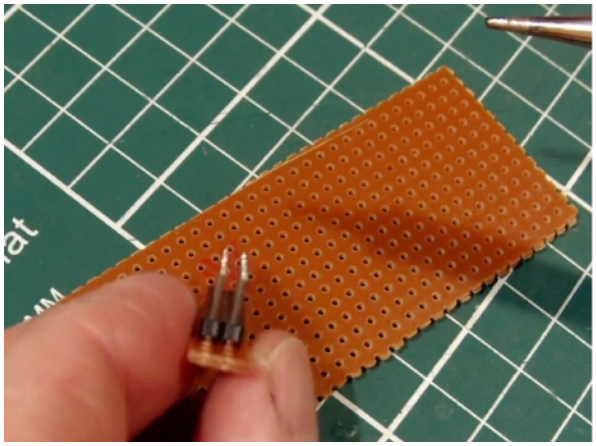




3. Break off two rows of 10 header pins and insert them into the board. This should leave a 2x2 section of strip with no pins at one end.

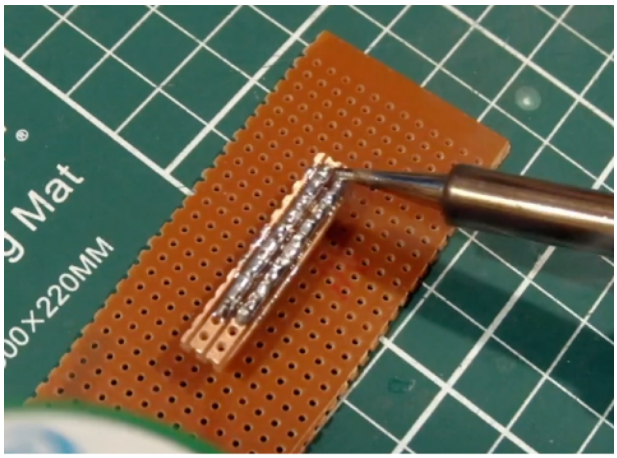


4. Use solder to secure one pin from each row to the board. A middle pin is best. Check the alignment of the pins and that they are fully settled onto the board. Using this method means that if you have to adjust the position of the row you only have the melt the solder on one pin to do this.



A useful tip here is to use a piece of stripboard as a jig to hold the parts in place during soldering.

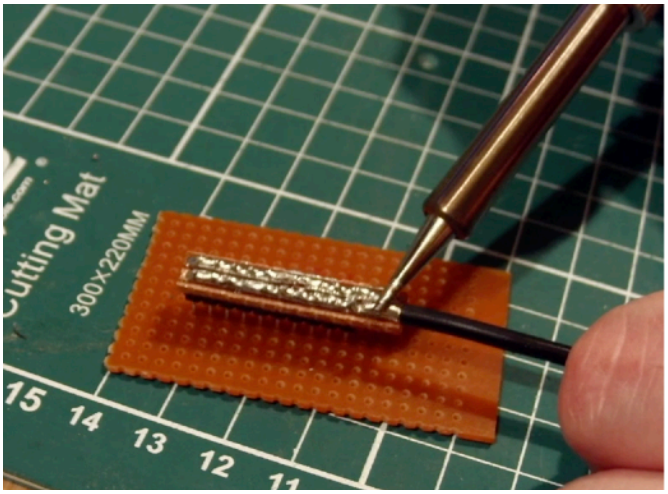
5. Once you are satisfied that all of the pins are in position you can now solder all of them to the board. Use plenty of solder, but be sure that there is no metal bridging the gap between the two lines of pins: they must remain electrically isolated from each other!



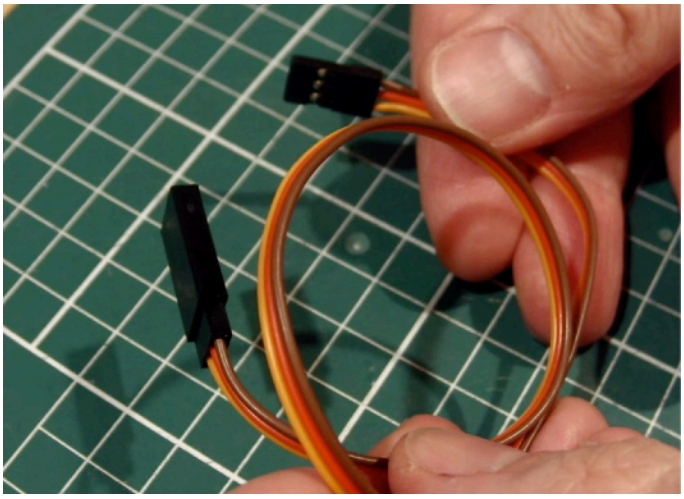
This has created the rack of pins that will power your servos. Next we need to do some wiring!

6. Take the thick black multi-stranded wire and cut off a length of about 10cm. This will be the main ground wire for the servos. Strip the insulation off 5-10mm of each end.

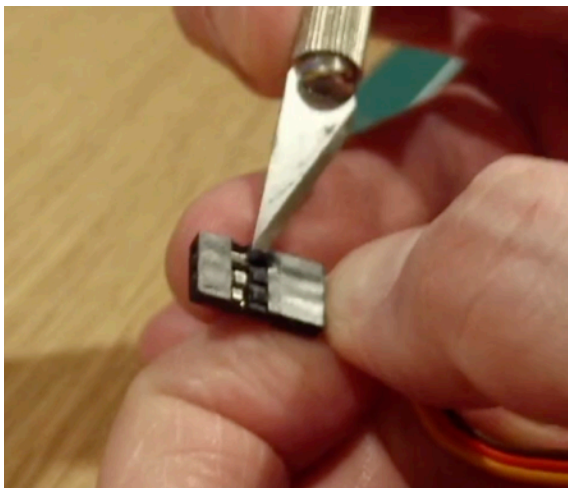
7. Solder one end of the ground wire to the end of one of the rows of pins on the board. It doesn't matter which one, just make sure that there is no metal bridging the gap between the two rows.

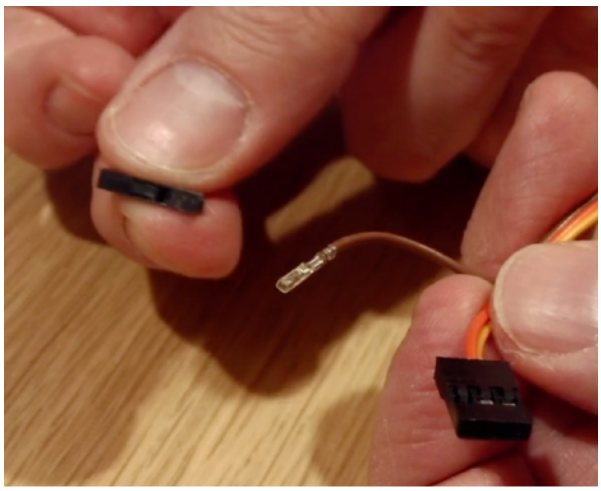


8. For the servos to be able to receive signals from the ESP32 via the signal wires the power bus must also be connected to a ground pin on the ESP32 to complete the circuit. For this we need a short length of thin multi-stranded wire with a female crimp connector on the end. The simplest way to get one of these, without the need to buy in a special crimping kit and tool, is to cut it out of a servo wire extension. These are inexpensive and easily available.

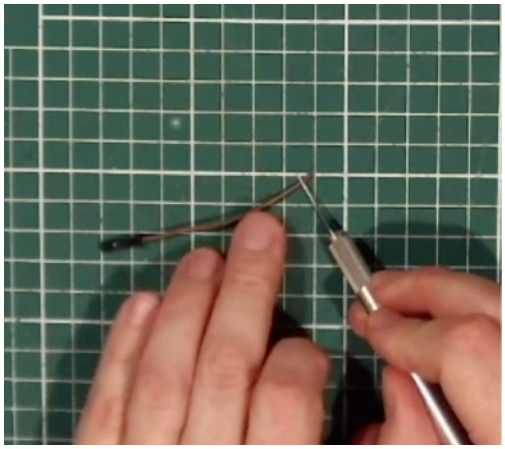


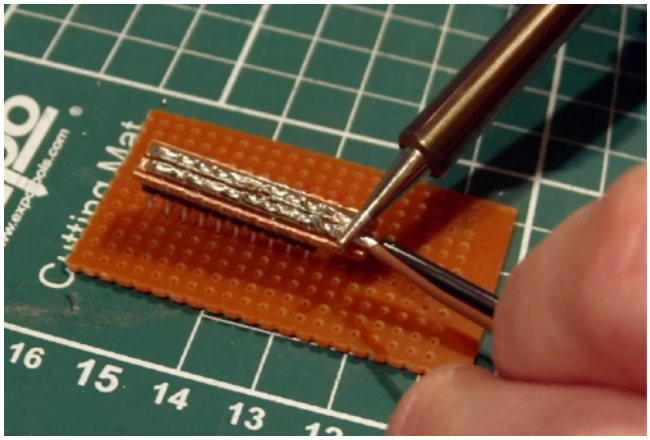
Remove the ground wire (brown or black) from the housing on the female end of the wire. You can do this by using a craft knife to lift up the plastic clip that holds it in place. This will expose the metal crimp connector on the end. Place a 1x1 connector housing over the top.





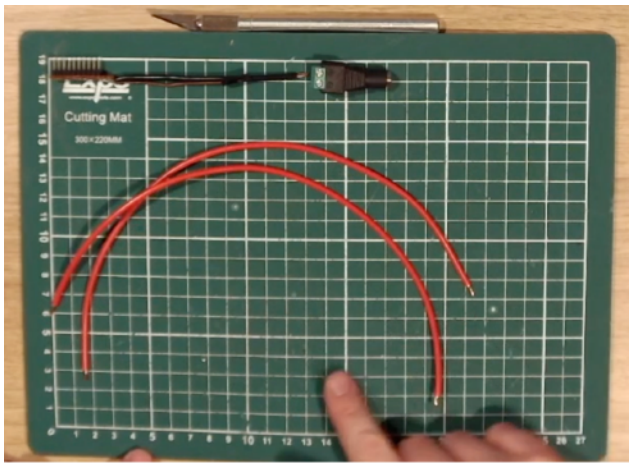
9. Pull the ground cable away from the other two servo wires and cut it to a length of about 7cm. Strip off about 5mm of insulation from the cut end, and solder this to the same place on the board that you soldered the main ground wire.

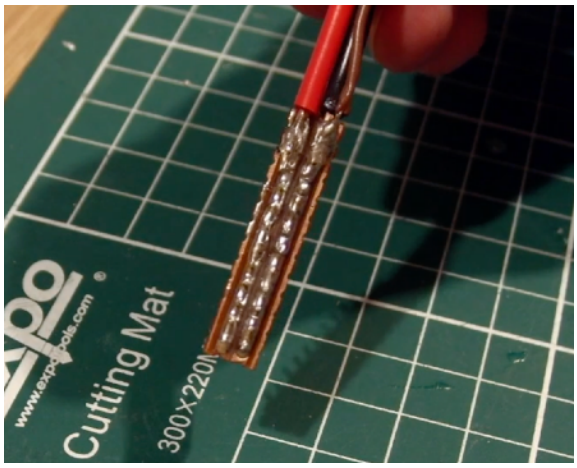




10. The Take the thick red multi-stranded wire and cut off two lengths of it 30cm long. Strip off 5-10mm of insulation off each end of both lengths.

11. Solder one of the lengths of thick red multi-stranded wire to the other track on the board. Ensure that there is no metal bridging the gap between the two tracks! One way to be confident is to use a craft knife to scrape the flux residue out of the gap, down to the plastic and blow out all of the dust. Use a multimeter if you have one to double check that there is no electrical connection between the two tracks.





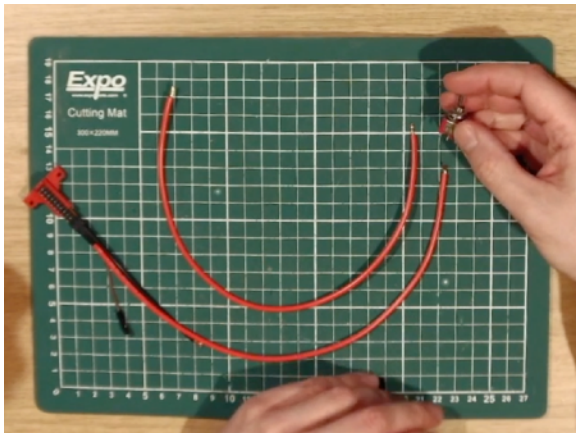
12. Wrap the joint between the board and wires tightly in electrical tape (or heat shrink if you have it). This helps secure the wires in place, and keeps everything nice and tidy.

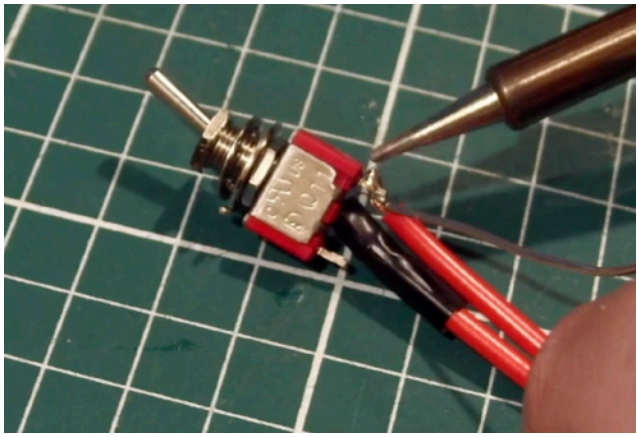
13. The toggle switch will be used as an on/off switch to connect and disconnect the power to the positive pins on the board, and thus the servos. To do this on a two-way toggle switch you should solder the free end of the red wire from the board to one of the outer connectors on the switch.

14. Solder one end of the second length of wire to the middle connector on the switch.

Note if your switch is different from the 2-way toggle switch illustrated your wiring may differ from that described above. Ultimately it just needs to be wired as a simple on/off switch.

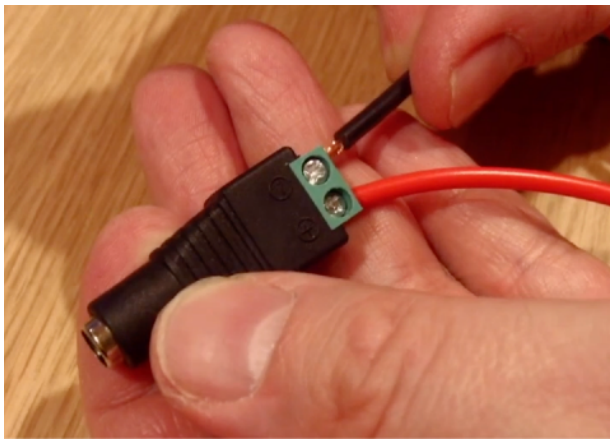
15. Wrap the joints with electrical tape (or heat shrink).



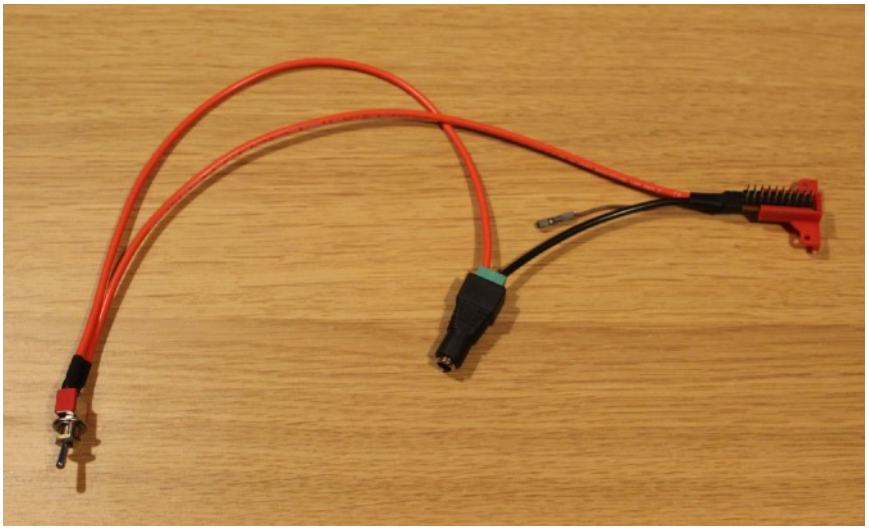


16. Finally, connect up the free red and black wire ends to the positive and negative terminals respectively on your power connector. This may be by soldering or by screw connector as illustrated. If you are unsure which terminal is which the best way to check is to plug the connector into your power supply and use a multimeter.

(Obviously: red = positive = power; black = negative = ground.)



You should now have a rack of two rows of pins (power and ground), joined to a connector and switchable on and off by a switch on the positive wire. There should be an extra little wire linked to the ground pins, which can be plugged into the ESP32. This is the finished power bus.

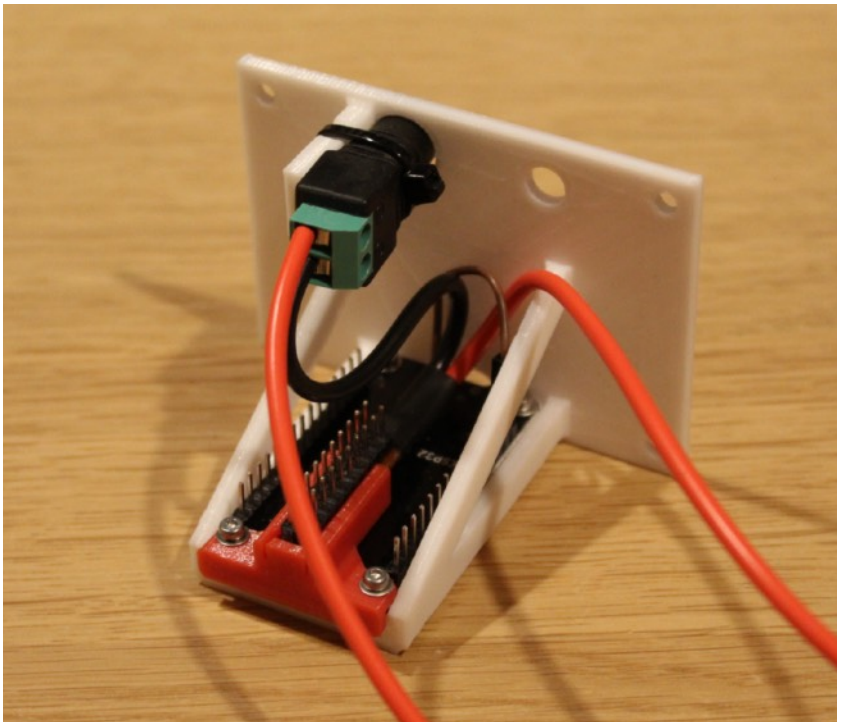


Congratulations you have completed the hardest part of the build! We can now move on to assembling the SR6 itself.

Tray

The tray is a removable part of the main enclosure that mounts the microcontroller and power bus and gives them access to the outside. It is designed to make the SR6's electronics easily user customisable.

Different trays may have different configurations. These instructions will describe the default version, which at the time of writing uses the "Alpha 3" tray part.



1. Install the ESP32 face down into the tray, with the micro-USB port facing out through the hole. Secure it in place using 2x M2x8 screws, with M2 washers, at the connector end.

2. Install the power bus board into the power bus holder. Install the power bus holder onto the top of the ESP32 and secure both in place using 2x M2x8 screws with M2 washers.

3. Mount and secure the power connector using a cable tie.

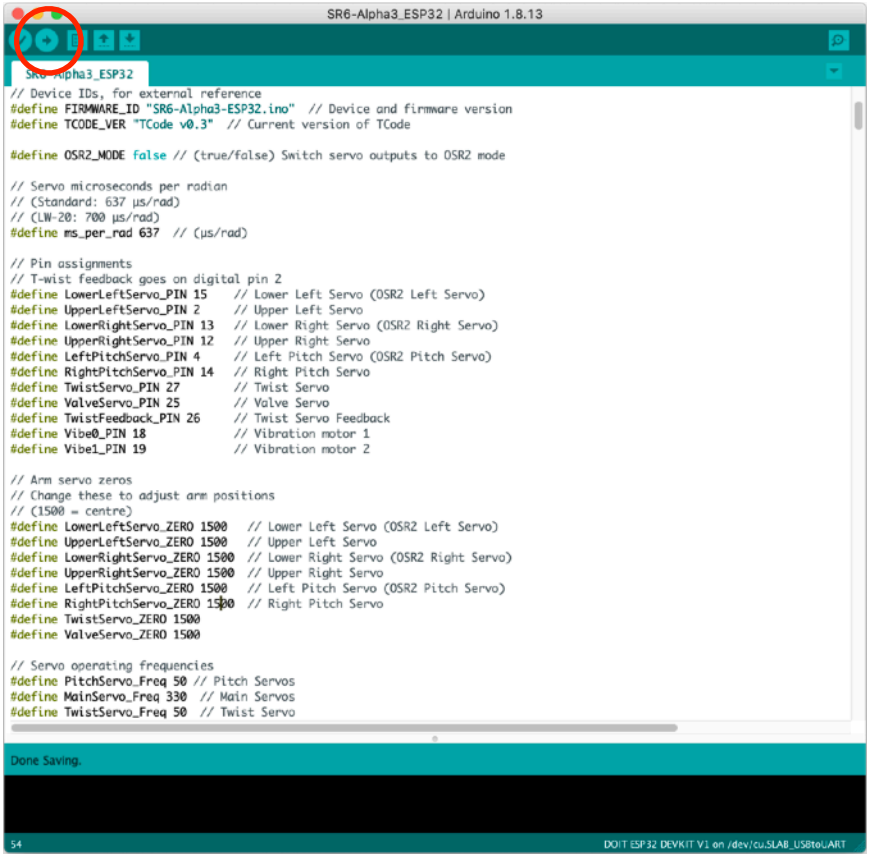
4. Plug the ground wire from the power bus into a "GND" pin on the ESP32.

Install the ESP32 firmware

At the time of writing the current firmware for the SR6 is the SR6 Alpha 3 ESP32 firmware (SR6-Alpha3-ESP32.ino). This is written for an ESP32 DevKit v1 microcontroller, as opposed to the Romeo BLE mini that was used for previous versions of the SR6 and the OSR2. it offers significant performance improvements and T-Code v0.3 support.

You will need to have the Arduino IDE installed (www.arduino.cc), and you will also need to install the ESP32 add-on. You can find a useful set of instructions for this here or here. This enables you to upload the SR6 firmware to an ESP32 module in exactly the same way as you would a Romeo BLE mini or Arduino Uno.

To upload the firmware to the ESP32 you will need open the .ino file in the Arduino IDE. In the "Tools" menu select "DOIT ESP32 DEVKIT V1" as the Board and select the appropriate COM port (this might be obvious, or you might have to guess!). Click the "upload" button (this looks like a right arrow) and after a few seconds you should see a message saying "done uploading".



It's worth nothing that this firmware can also be used to control the 3-servo setup of an OSR2 instead of an SR6. All you need to do is change the "OSR2_MODE" line to read "true" instead of "false".

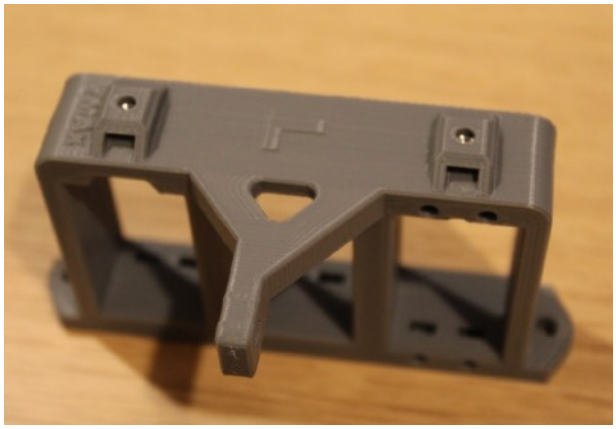
This firmware version supports PWM pin control of two vibration channels, or a vibration channel and a lube motor with control button. The ESP32 has no motor speed controllers built in, so these features would require additional hardware.

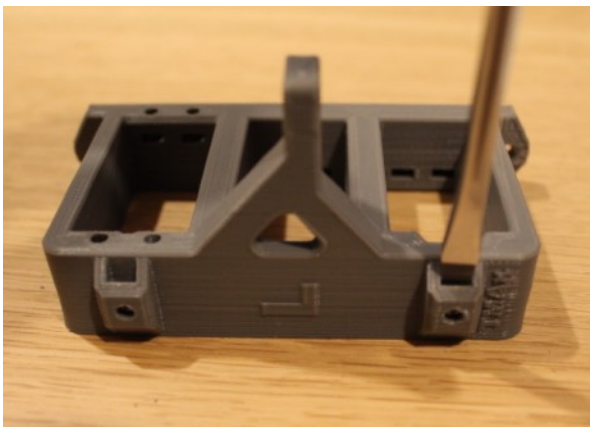
This release is USB only, as it does not currently use the ESP32's bluetooth or WiFi capabilities. It also does not support an external display screen. It is expected that these features will be added in future updates.

Frame

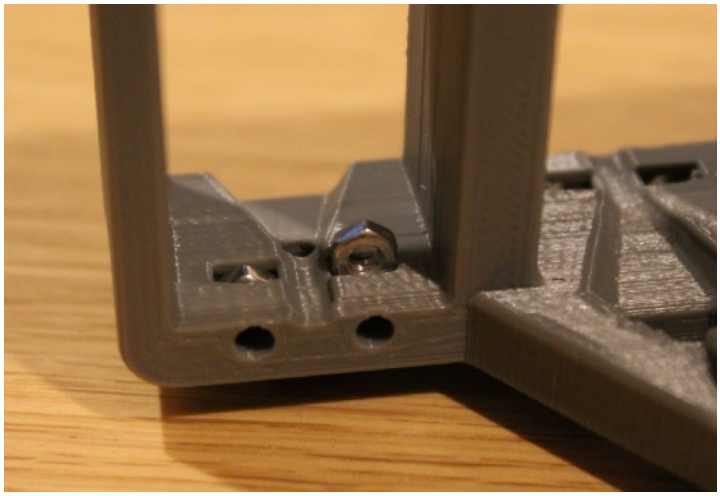
The SR6 Beta is built around a core structure called the frame, to which the six servos are mounted. The frame comes in two halves, left and right, and broadly speaking these are a mirror image of one another. The two halves of the frame are marked “L” and “R”, and as always left and right are from the user’s point of view.

1. For each of the two frames insert an M3 nut into each of the two slots on the top of the part. Push them in all the way so that they line up with the circular holes. If necessary use a screwdriver or similar tool to move them into position.

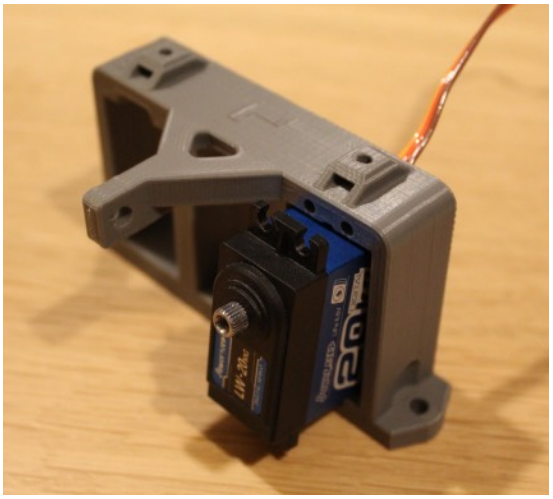


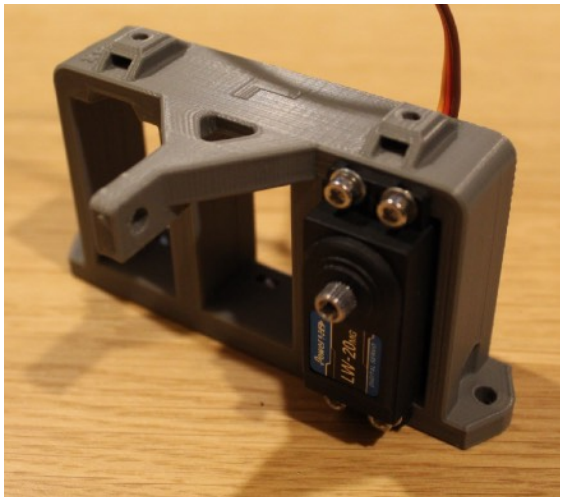


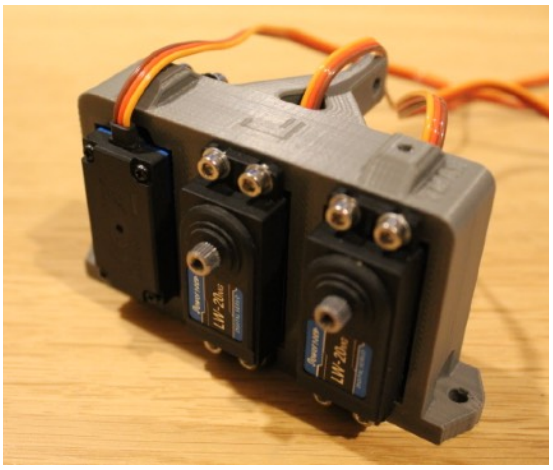
2. For each frame insert an M3 nut into each of the 12 inside slots so that they line up with the circular holes.

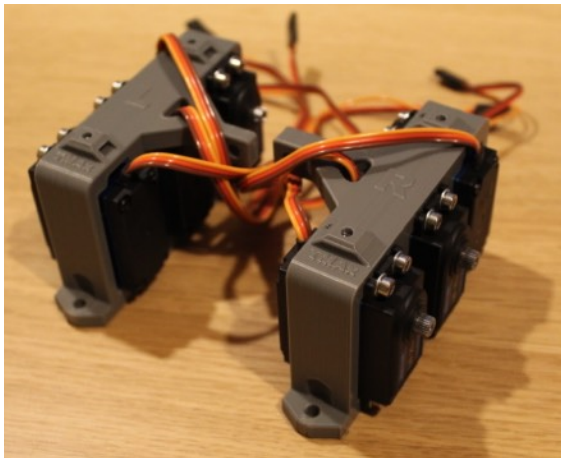


3. For each frame, install the three servos and secure each of them with 4x M3x10 bolts and M3 washers. Note that the servos should be arranged so that the servo axles, and servo leads, should be nearest the top of the frame. The 2 pitch servos point inward, and the 4 main servos point outward.

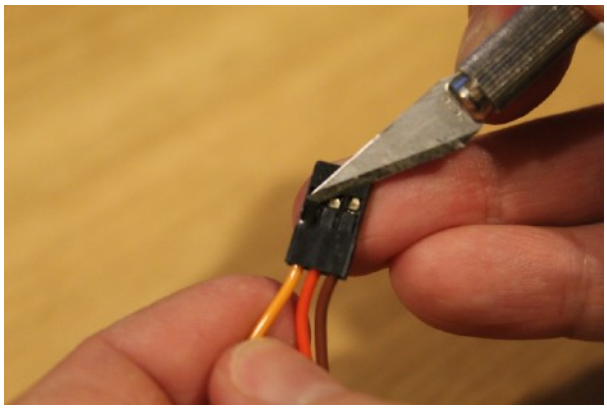


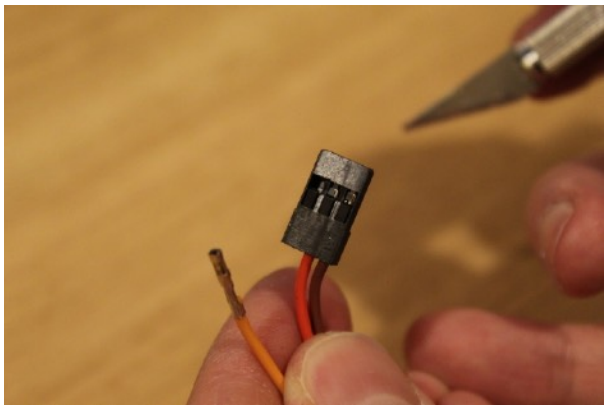




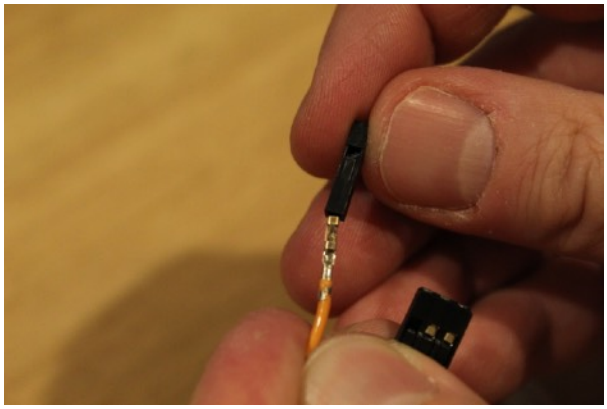


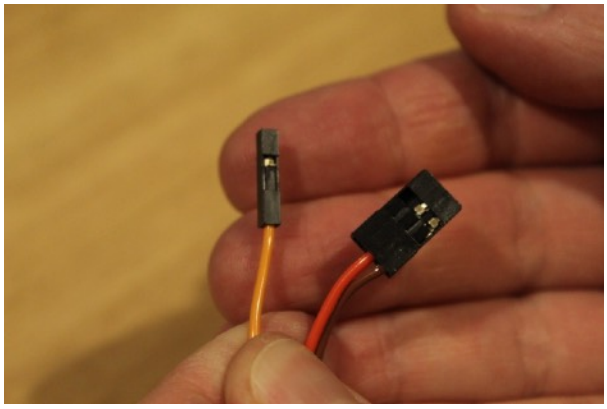
4. For each servo remove the signal wire (yellow or white) from the connector housing. Leave the power (red) and ground (brown or black) wires in place.





5. Cover each naked signal connector with a single connector housing, or alternatively wrap with electrical tape.

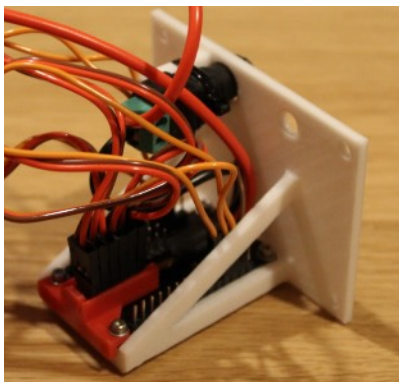


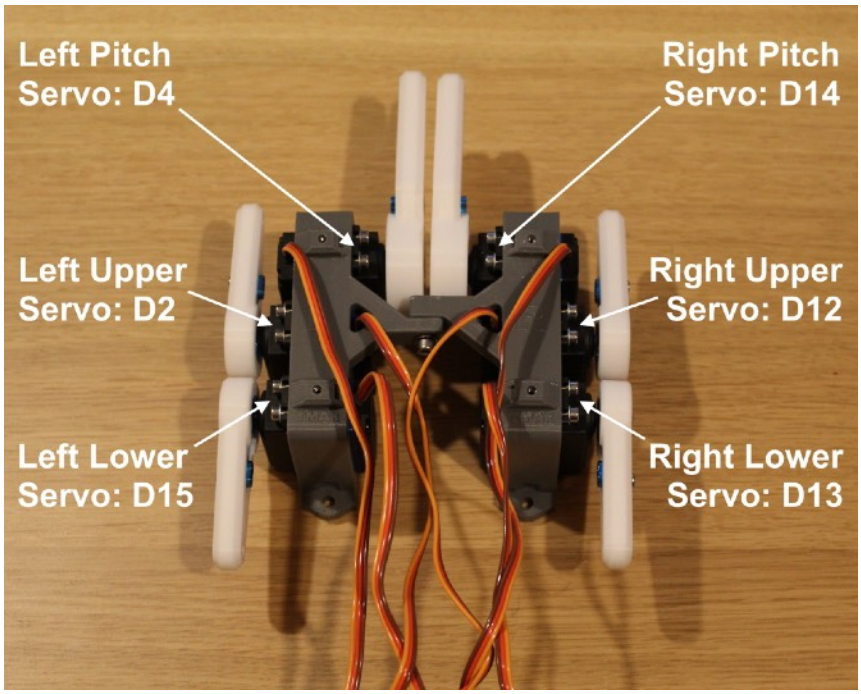


6. Plug the servo power and ground leads into the power bus. Plug the servo signal leads into the digital pins on the ESP32 as follows:

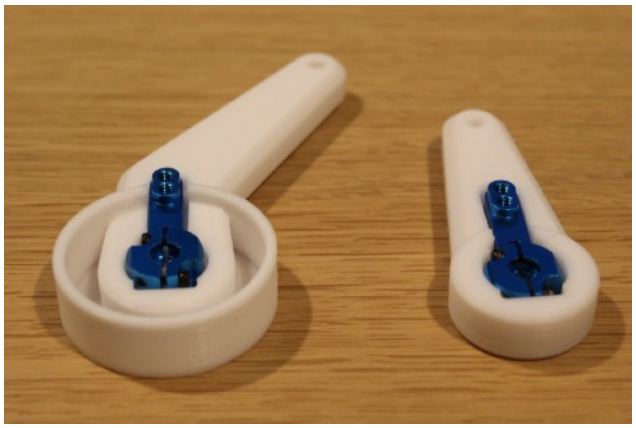
Left Pitch Servo: D4
Left Upper Servo: D2
Left Lower Servo: D15
Right Pitch Servo: D14
Right Upper Servo: D12
Right Lower Servo: D13

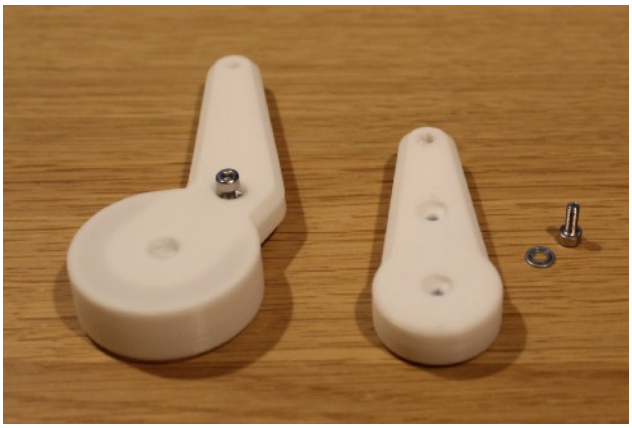
Ensure that the ground on the power bus is connected to a ground pin on the ESP32.





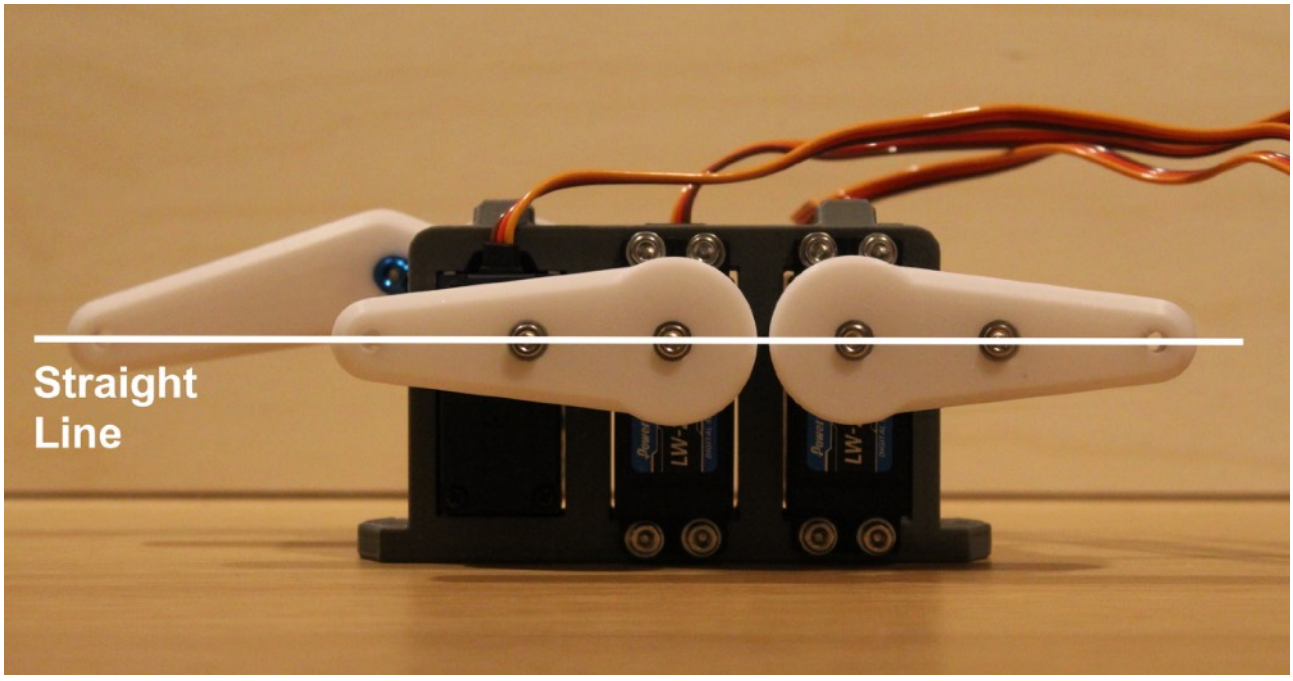
7. For each of the arms, 4 main, L-pitcher and R-pitcher, install a Futaba 25T servo horn into the shaped slot and secure in place using an M3x8 bolt with an M3 washer. Do not tighten these bolts fully at this stage, just enough to keep the servo horn loosely in place inside the arm slot.





8. Connect the ESP32 to a USB cable so that the board is powered. Connect the power bus up to your power supply and move the power switch to the “on” position. You should hear the servos move into their default startup positions.

9. For each frame attach two main arms and one pitcher arm onto the servos. The arms should be positioned so that the holes on the end of the arms and the holes that line up with the servo axles are all on a straight line that’s parallel with the base of the frame. Note that the hole on the end of the pitcher arm should be on this line, and that the angled kink in the pitcher arm should point upwards.



Servo horns can only be inserted onto the servo axle at one of 25 possible angles so a perfect alignment may not be possible at first. With the servo power on, install the arms as close as possible to the desired position.

Fix each arm in place using an M3x10 bolt with M3 washer. Also tighten the arm’s other M3 bolt so that the arm is now securely fixed to the servo horn.

To verify that the servos are powered and in the correct positions, move the power switch to the off position, move the arms to new positions, and then return the switch to the on position. The arms should return to their set positions.

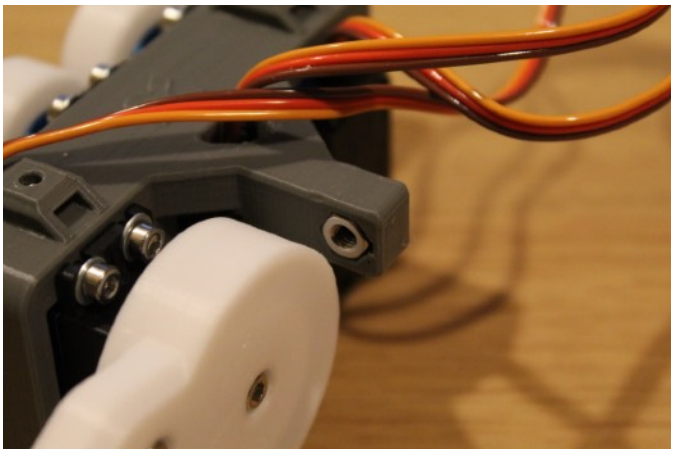
10. It is possible to fine tune the “zero” position of each servo in the firmware. For each servo if you want to adjust the starting position you can change the zero value in the firmware sketch. To do this you must change the number in the sketch (the default for each will be 1500) and re-upload the sketch to the ESP32.

```
// Arm servo zeros
// Change these to adjust arm positions
// (1500 = centre)
#define LowerLeftServo_ZERO 1550
#define UpperLeftServo_ZERO 1550
#define LowerRightServo_ZERO 1600
#define UpperRightServo_ZERO 1550
#define LeftPitchServo_ZERO 1400
#define RightPitchServo_ZERO 1600
#define TwistServo_ZERO 1500
#define ValveServo_ZERO 1500
```

As a guide + or - 160 to the 1500 corresponds to the same angle as moving the servo horn from one step to the next on the servo axle (ie 1/25 of a full turn). In theory therefore your numbers shouldn’t need to be outside the range 1420 to 1580 if the horns were already on the closest positions possible. The best policy to find the right number is to try small increments (20 to 40), observe the change, and then adjust from there.

Note that you can be sure the ESP32 has restarted by pressing the enable button “EN”.

11. Finally, install an M4 nut into the back of the bracket on the right frame.



Links

The links are the parts that connect the arms to the receiver. The recommended option for the SR6 beta is bearing links, which use male M4 Rod End Bearings. There is an alternate option, the Alpha release links that use wiring grommets instead of the rod end bearings, which will also be described here.

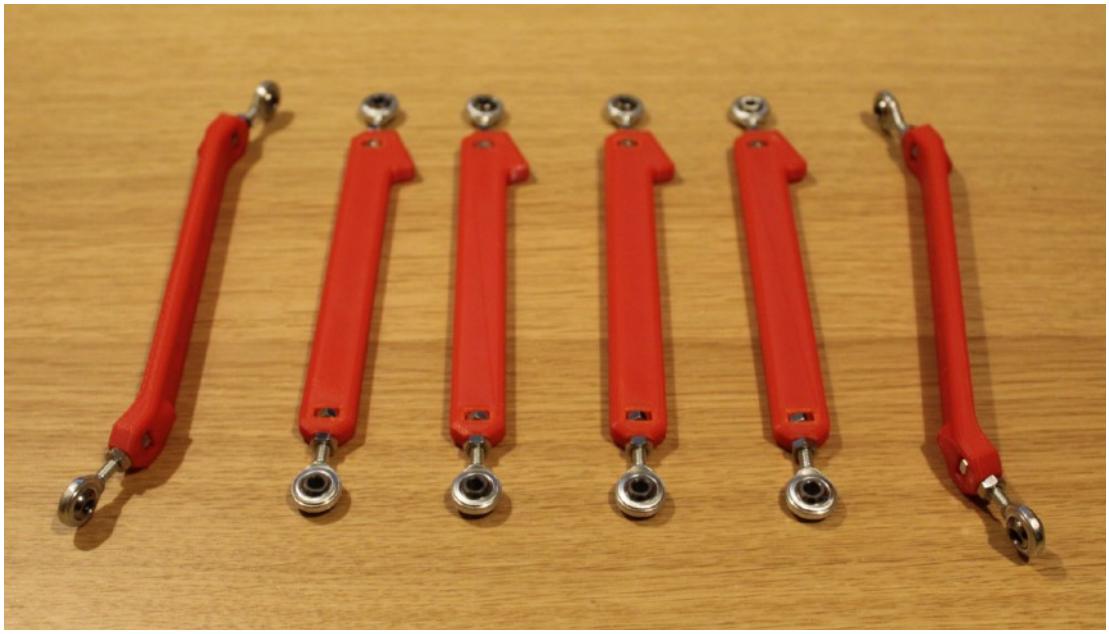
There are 6x links: 4x main links, 2x pitcher links. Each of these requires 2x M4 rod end bearings, and 4x M4 Nut.

The procedure for both ends of each link is the same:

1. Insert first M4 nut into the slot in the link.
2. Screw second M4 onto the shaft of the M4 rod end bearing
3. Screw the rod end bearing into end of the link.
4. Use pliers or a spanner (wrench) to tighten the second nut up against the end of the link, fixing the bearing in place

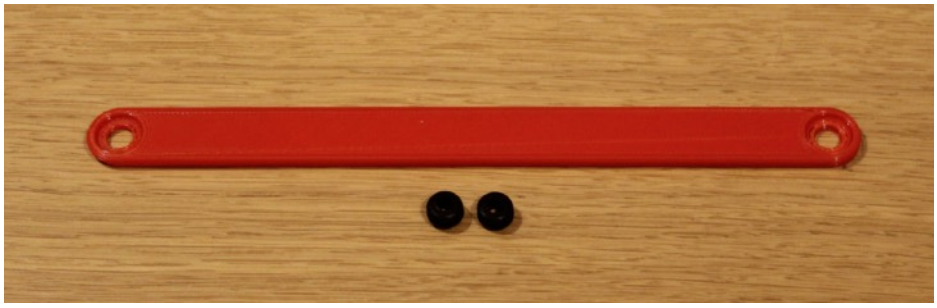
On each finished link the centres of the eyes of the two bearings should be 175mm apart, with the bearing shafts protruding from each end an equal amount. The bearing should be aligned with the flat sides of the link.

Check the six links against each other, particularly the straight main links, and ensure that they are all the same length. The two pitcher links should also be the same length as each other.

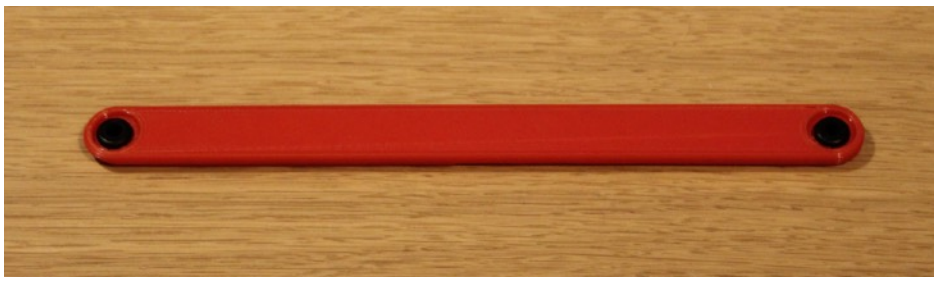


Alternative links

The SR6 Alpha used links that had flexible joints made using wiring grommets, which are still compatible with SR6 Beta should rod end bearings be unavailable.

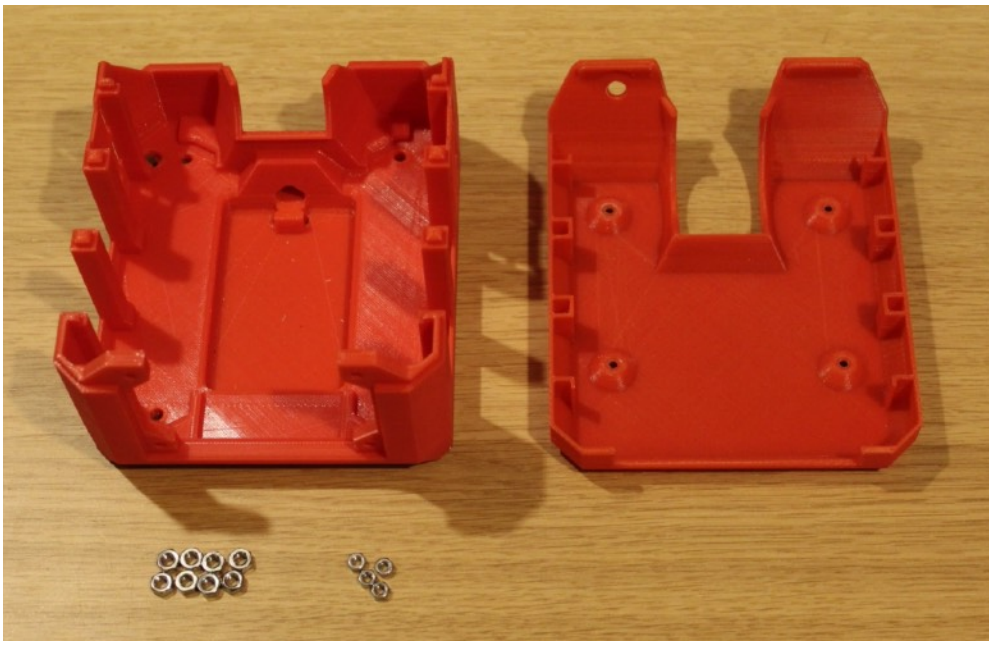


Each link (4 main, 2 diagonal) is used with 2x 6mm wiring grommets. One grommet is inserted into each 6mm hole, on opposite ends of the link.



Enclosure

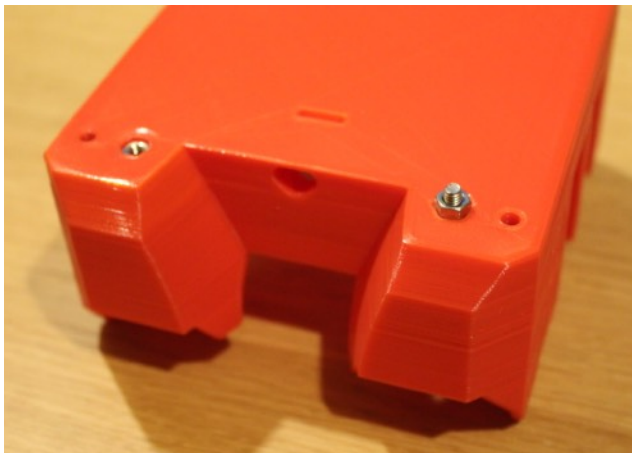
The SR6 enclosure is a box that sits around the outside of the working parts of the device. it comprises mostly of a base and a lid. The base includes the four-hole interface to the VESA 100 mount and the mounting holes for the internal frame. It also mounts the tray. The lid completes the enclosure, and includes a mounting hole for the power bus on/off switch.

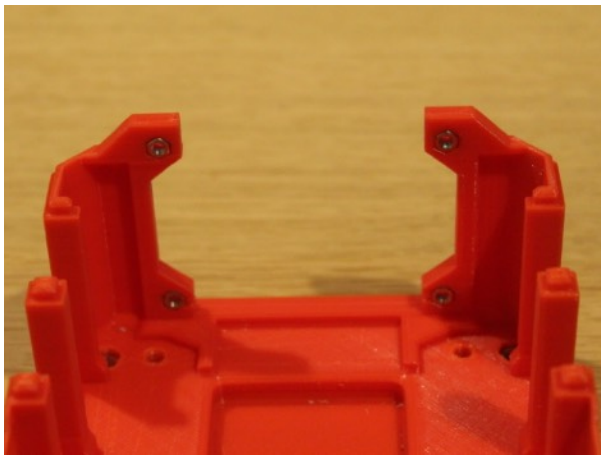


1. To prepare the base for the installation of the frame install 8x M4 nuts into the hexagonal holes: 4x inside, 4x outside.

2. To prepare the base for the installation of the tray, install 4x M3 nuts into the hexagonal holes in the enclosure wall.

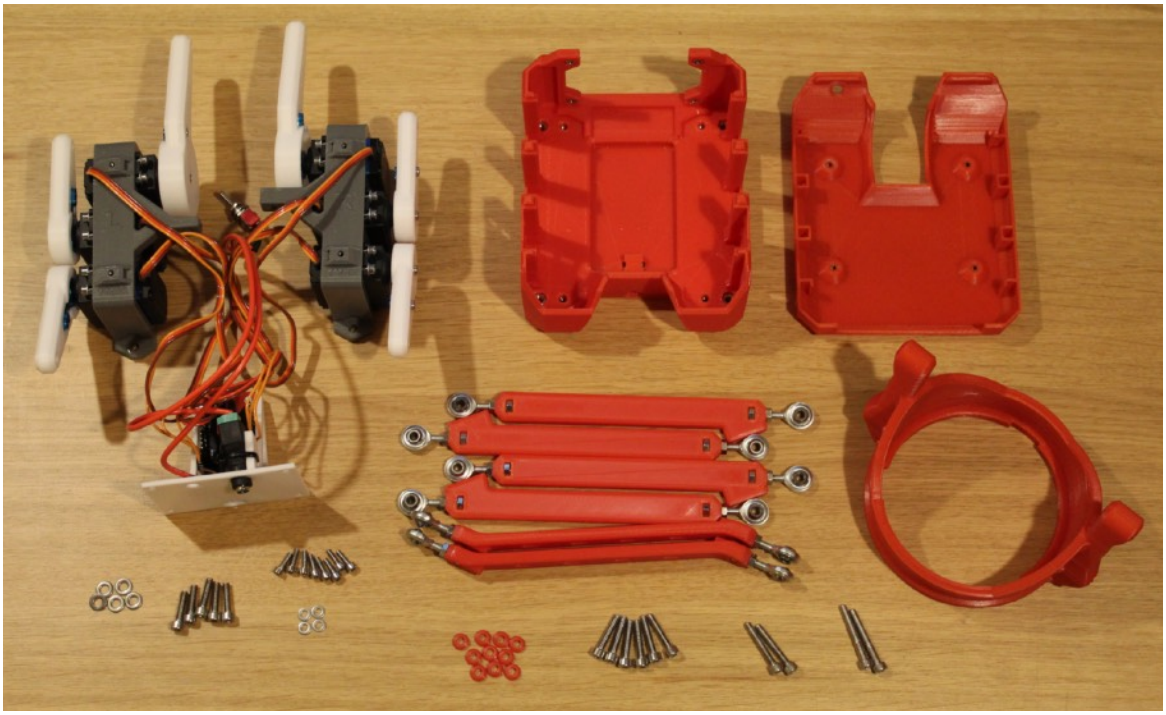
Depending on the tightness of the fit, sometimes it is necessary to use a bolt and washer on the other side to pull the nut into position. On the contrary, if the fit is loose a dab of superglue is sometimes useful to keep the nuts in place.





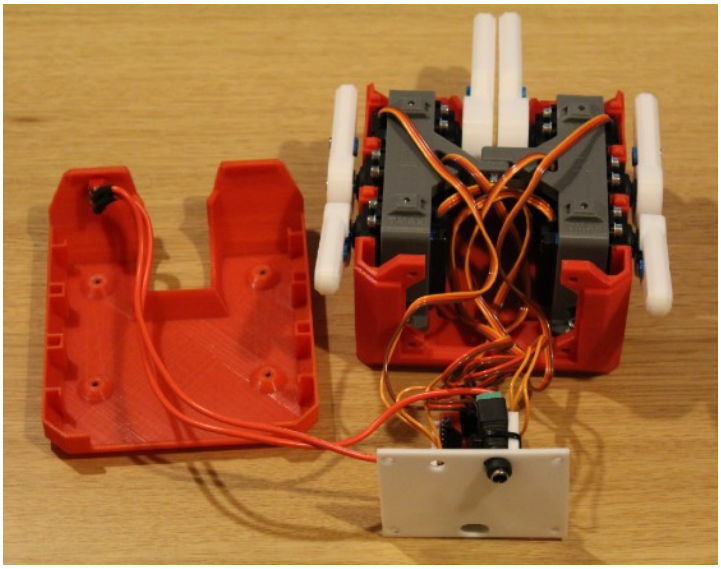
Final Assembly

Now all of the constituent parts have been assembled it's a relatively straightforward ask to put them all together.

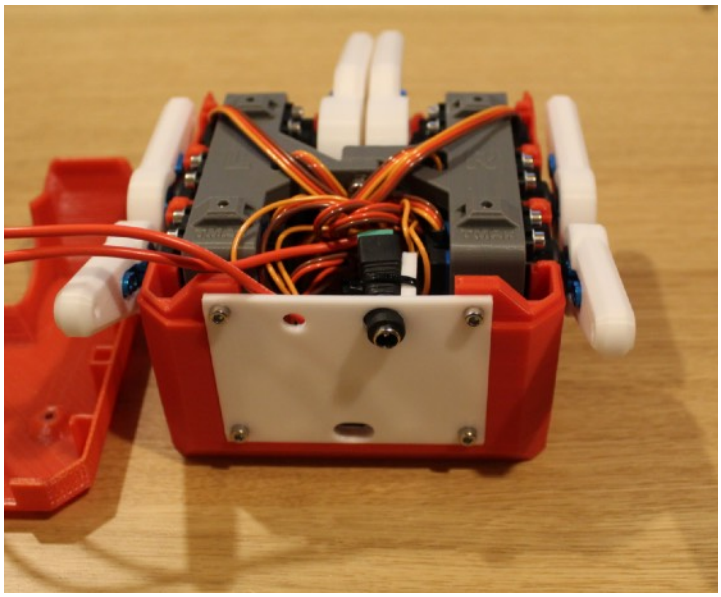


1. Install the power bus switch into the lid.

2. Install both halves of the frame into the base and secure them in place with 4x M4x16 bolts with M4 washers. Also install the 1x M4x16 bolt, with M4 washer, that links the two halves of the frame together.



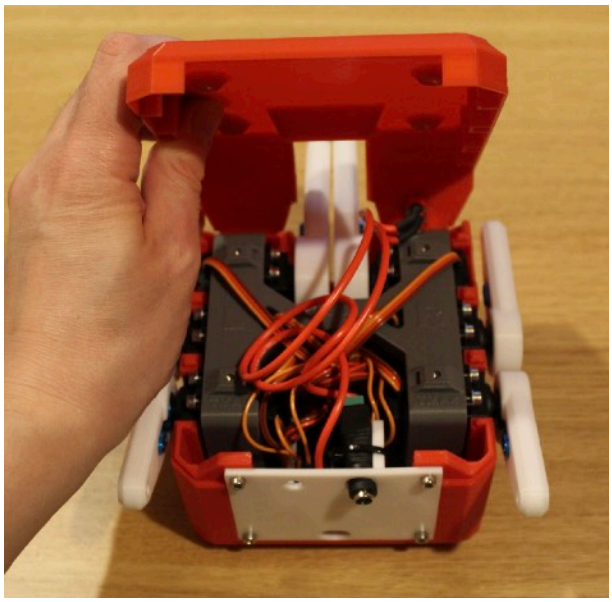
3. Install the tray into the base and secure it in place with 4x M3x10 bolts, with M3 washers. Tuck all of the servo wires into the space between the frame and the tray.

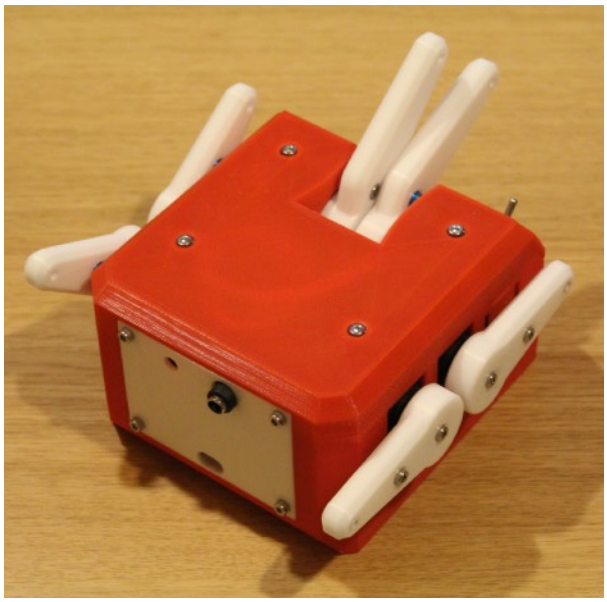


4. Install the lid onto the top of the enclosure and secure it in place with 4x M3x10 bolts, with M3 washers. Take care over the routing of the wires to the power bus switch, and be sure that they're not pinched between the base/frame and lid.

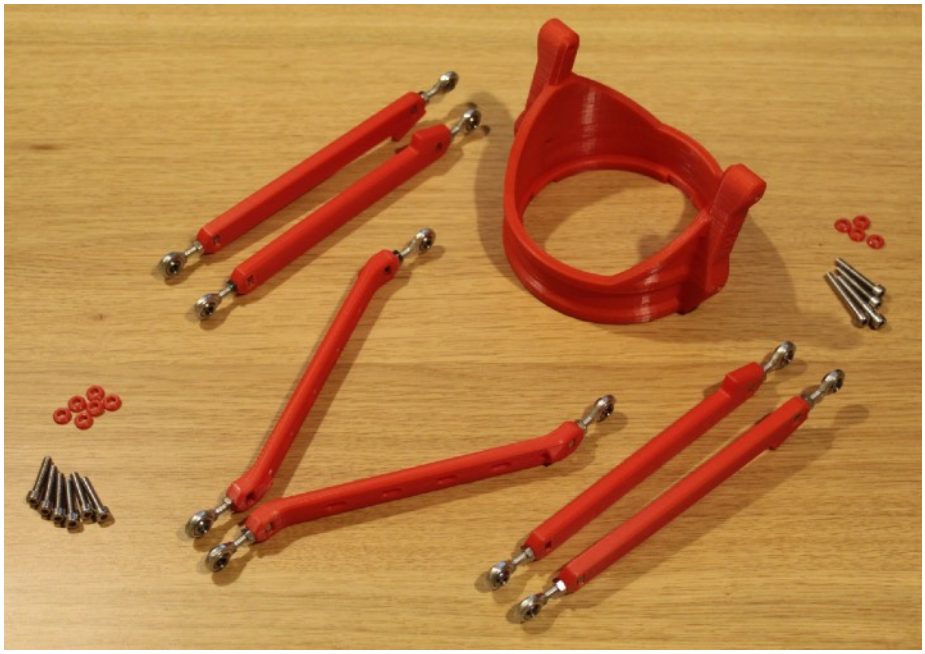
Tightening the lid bolts should be easy. If you encounter excessive resistance do not fight it! Usually resistance means that either a wire has become pinched, or that the nut holes in the frame are not in alignment with the lid holes. Damaging something at this stage of assembly can be really frustrating, so if in doubt take the lid off and check!

If the frame holes are not in alignment a good solution is to remove the tray and loosen the bolt connecting the two halves of the frame. This should resolve the alignment problem. Once the lid is screwed on you can re-tighten the frame bolt through the gap where the tray was attached. Now remove the lid again and go back to step 3.



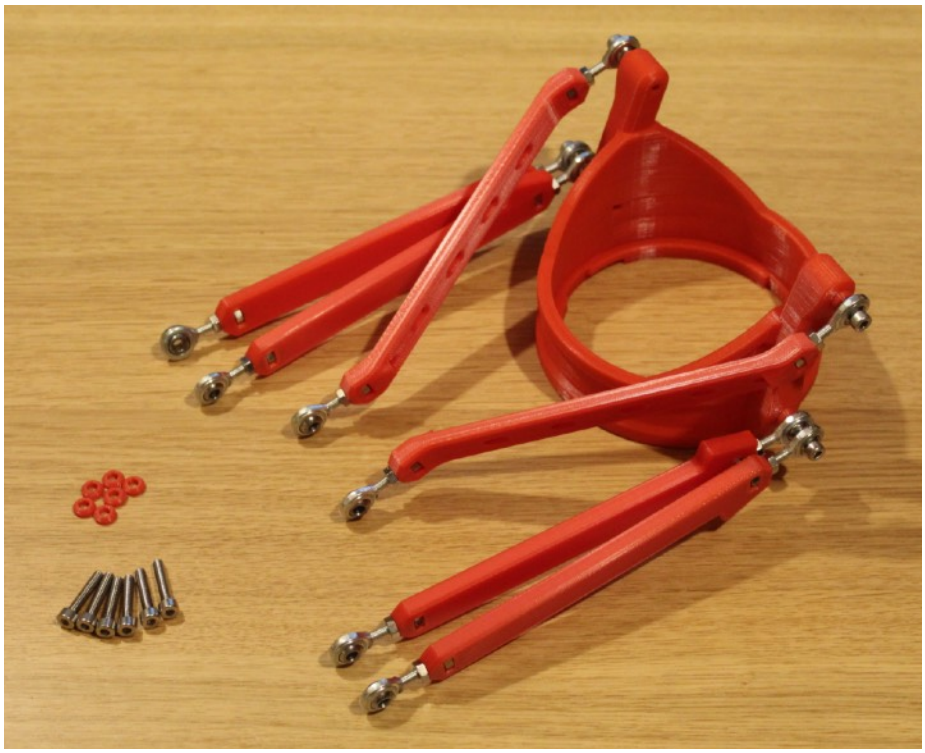


Now it’s time to connect up the linkages.

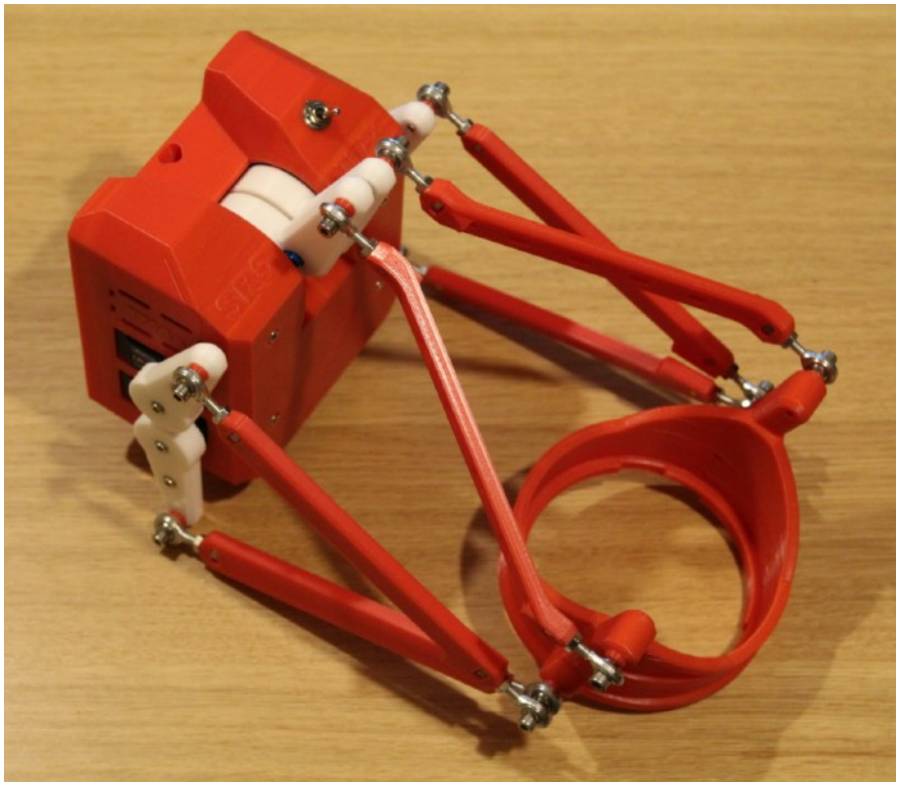


5. Connect the pitcher links to the upper mounting points on the receiver using 2x M4x25 bolts. Use a 4x3mm spacer with each bolt, located between the receiver and the link, with the narrow end pointed toward the bearing. The links should be mounted so that they point diagonally inward.

6. Connect the four main links to the lower mounting points on the receiver using 2x M4x30 bolts, with two links on each bolt. As above, use a 4x3mm spacer with each bolt. The links should be mounted to the receiver by the end with the protruding tab, with the tabs pointing one up, one down.



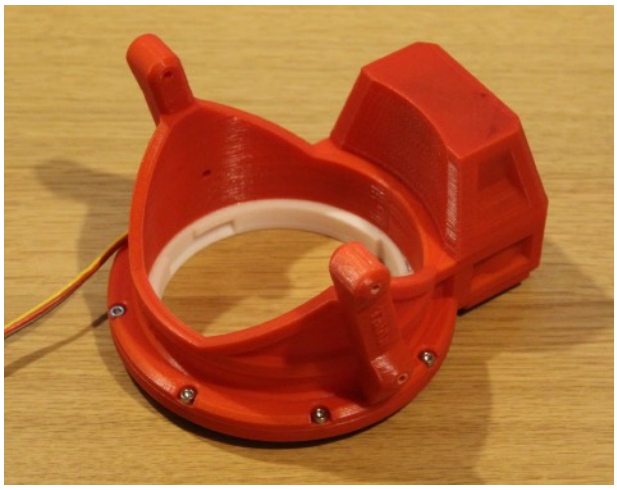
7. Attach the six linkages to the six arms on the SR6 using 6x M4x20 bolts, with 4x3mm spacers. As above, the narrow end of the spacers should point toward the bearings. The pairs of main arms should be spread so that the tabs on the links point toward each other. This simple feature creates a permanent overlap and prevents the links from sitting on top of one another.



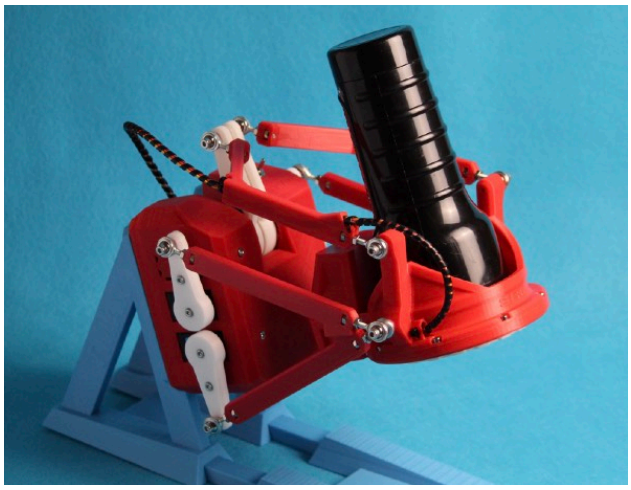
This completes the basic SR6 build!

T-wist Module Assembly for SR6

The T-wist is a modular upgrade for the SR6 that adds an additional movement axis. The new axis is 180° or 270° rotation (±90° or ±135°) of the fleshlight case around the long axis.



T-WIST 4 RECEIVER



SR6 WITH T-WIST 4

The twist movement is achieved by means of a rotating ring mount inside the receiver, which is driven by a standard sized servo. This is made possible by the use of a simple gearbox that transfers the servo movement to the ring with a 1:1 ratio.

Previous versions of the T-wist have used a continuous rotation servo and depended on some kind of feedback system. Versions 1 and 2 used a potentiometer located at the top of the fleshlight case in a T-valve housing; version 3 used a specific servo type, the Parallax 360, which has a feedback wire. Version 4 therefore represents a welcome simplification!

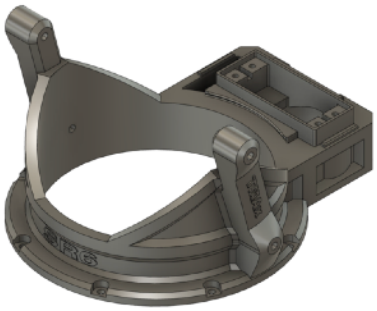
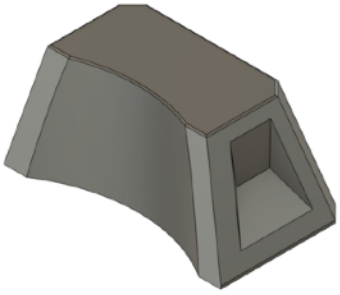
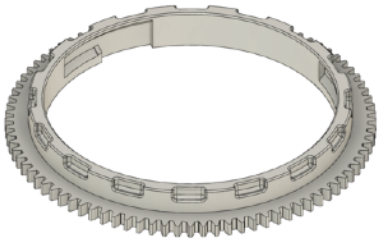
T-WIST 3 RECEIVER

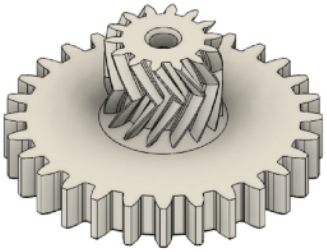
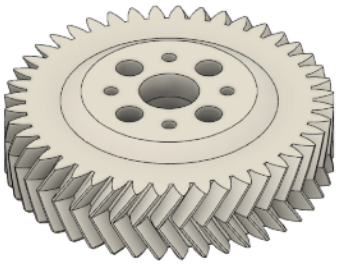
T-wist parts list

This is a complete list of the parts that comprise the T-wist.

The 3D printed parts are supplied in .STL format; you can print them yourself or order them from an online 3D printing service. Unless otherwise specified the STL files are provided in the intended build orientation and are designed to be printed without supports.

The off-the-shelf parts should be easily available through ebay, amazon, local hardware or hobby shops, etc.

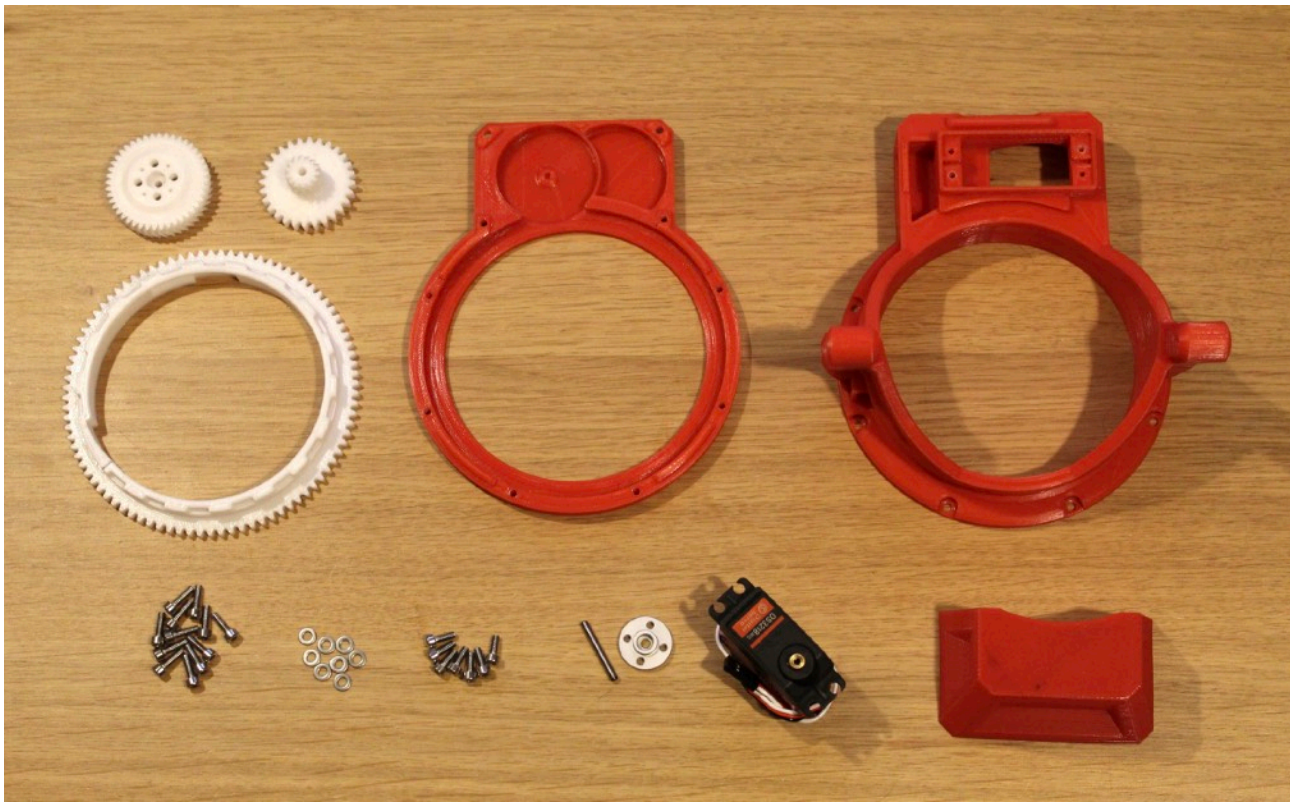
Part	Qty	Description
T-wist4 SR6 Receiver Base 	1x	This is the base of the receiver. The corners of the gear enclosure are supported during the print by small break-off tabs.
T-wist4 SR6 Receiver Body 	1x	This is the main receiver body with attachment points that match the standard OSR2 receiver. There is a break-off support under the gear enclosure that ensures that the transfer gear axle hole forms correctly.
T-wist4 Receiver Lid 	1x	This is the clip-on lid that sits over the servo on the receiver body.
T-wist Ring Gear 	1x	This is the rotating mount for the fleshlight case. This part is backwards compatible, ie it is interchangeable with the clip ring from the T-wist 3.

T-wist 4 Transfer Gear 	1x	This is the intermediate gear that transfers power from the drive gear to the ring gear I advise printing this part in high resolution for the best performance.
T-wist 4 Drive Gear 	1x	This is the gear that is mounted on the servo horn. It has 4x counterbored mounting holes for M3 bolts to mount to a metal servo horn, and 4x holes for M2 bolts for mounting to a plastic servo horn. I advise printing this part in high resolution for the best performance.
Servo 	1x	The T-wist 4 works with a standard sized servo. I recommend that you use a 270° servo for best range of movement, but a 180° servo will also work. A 20kg.cm servo has been pictured, however this is likely overkill and a servo in the 5-10kg.cm torque range is probably more than sufficient. I look forward to the community feedback on this!
Circular Metal 25T Servo Horn 	1x	I recommend that you use a metal servo horn. These will almost certainly not come with your servo and will have to be sourced separately. If you can't get hold of a circular metal servo horn you can substitute the circular plastic one that often comes with standard sized servo.
M3x25 Dowel Pin 	1x	This is a metal pin that is used as the axle for the transfer gear. An M3x25 dowel pin (3mm diameter, 25mm length) is a standard part that should be easy to get hold of. If you have trouble sourcing one a length of 3mm metal rod or tube cut to length will work just fine.

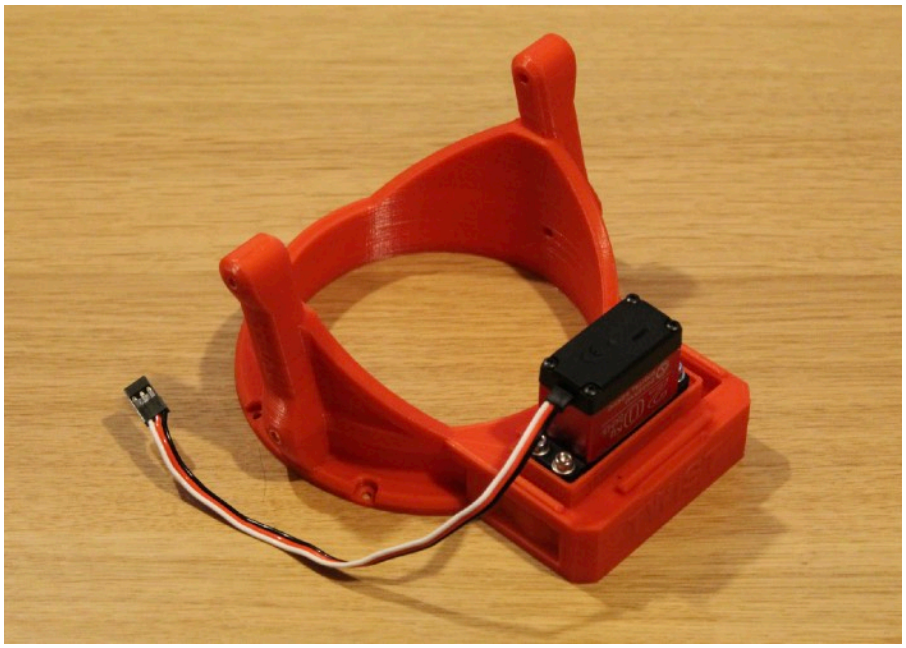
Bolts, washers, etc 	7x M3x8 Bolt 12x M3x10 Bolt 9x M3 Washer *4x M2x6 Bolt *4x M2 Washer *1x M3x16 Bolt	The two parts of the receiver are held together using (6x) M3x8 and (4x) M3x10 bolts, which screw directly into the plastic. Likewise the servo is held in place with (4x) M3x10 bolts, with M3 washers. 4x M3x10 bolts with M3 washers mount the metal servo horn to the drive gear. An M3x8 bolt and washer mounts the horn to the servo. *If you are using a plastic servo horn you will instead need (4x) M2x6 bolts and washers, and a single M3x16 to attach the drive gear to the servo.
Servo lead extension 	1x	This is to connect the servo back to the OSR2 body via the arms/links. The optimal length of extension from the T-wist back to the main enclosure is 600mm
Spiral cable binding & cable ties 	-	These are used to tidy up the cables once you have everything working.

You will also want your SR6 to hand so that you can check that the mechanism is working.

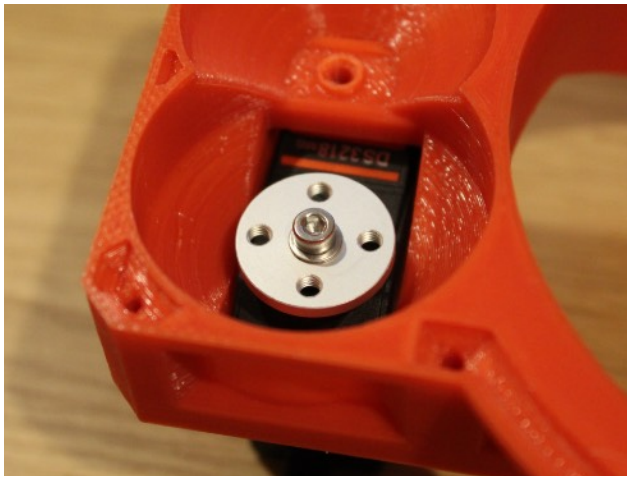
T-wist assembly steps

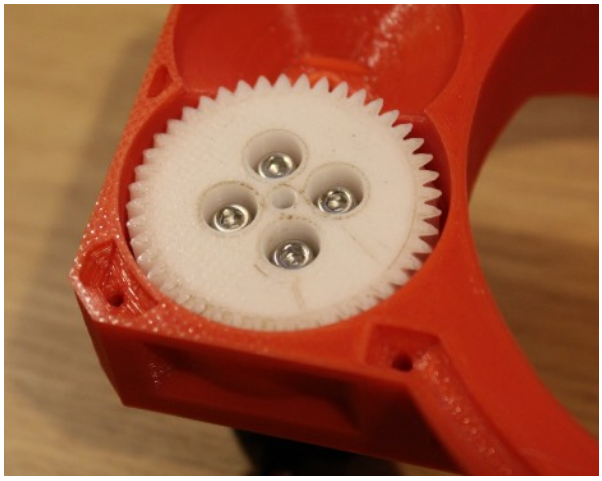


1) Mount the servo into the receiver body and attach it using 4x M3x10 bolts and M3 washers. Do not fully tighten the bolts at this stage: allow some freedom of movement to the servo.

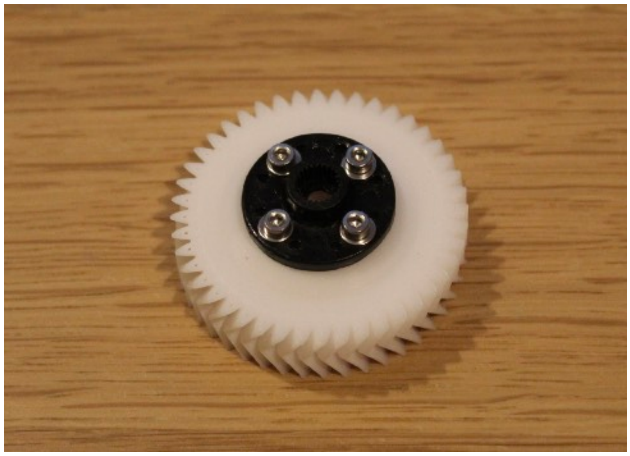


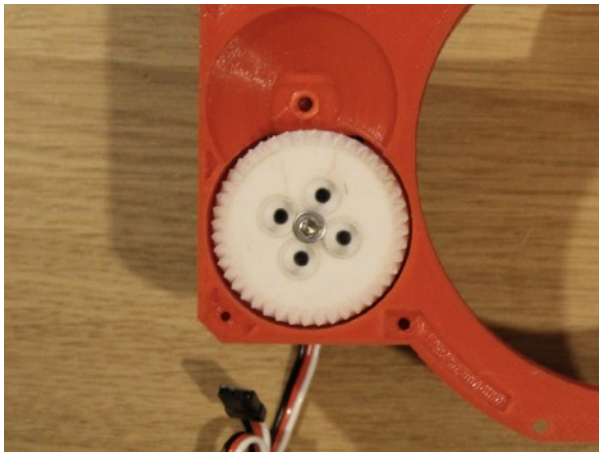
2) If you are using a metal servo horn (recommended), place it onto the servo and secure it in place with an M3x8 bolt and M3 washer. Then install the drive gear onto the servo horn using 4x M3x10 bolts and M3 washers.





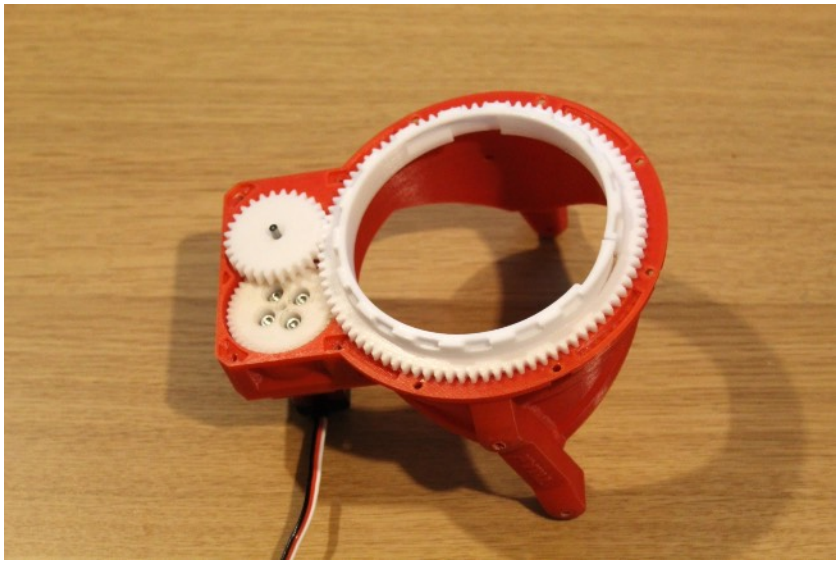
2A) If you are using a plastic servo horn mount the servo horn to the drive gear first using 4x M2x6 bolts and M2 washers. You may need to enlarge the holes in the plastic servo horn to do this. Next install the horn and gear onto the servo using the M3x16 bolt and an M3 washer.



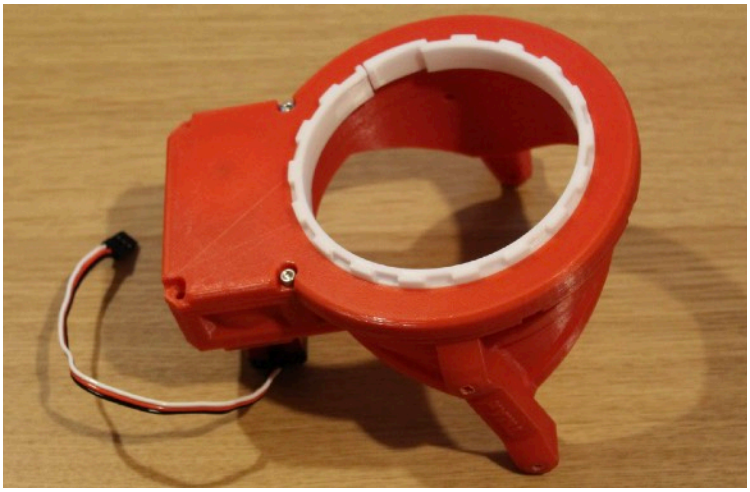


3) Install the transfer gear into the receiver body and insert the M3x25 dowel through the gear so that it sits in the hole on the receiver body.

4) Place the ring gear on the receiver so that it intermeshes with the transfer gear.



5) Place the receiver base onto the receiver body, making sure that the M3x25 dowel sits in the appropriate hole in the receiver base. Temporarily secure the receiver base in place using 2x M3x10 bolts.

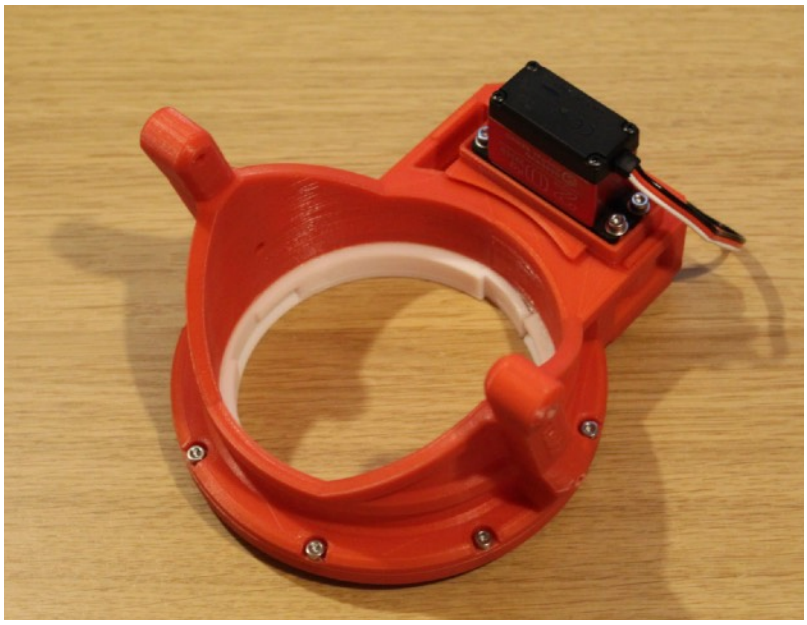


6) Check that the mechanism is working correctly by turning the ring gear by hand. You should be able to feel and hear the servo rotating in train with the gears as you do this.

7) Move the servo into a position where you can feel the drive gear smoothly engaging with the transfer gear and tighten the 4x M3x10 bolts to fully secure the servo in place.

8) At this point an optional but highly recommended step is to remove the base and liberally coat all of the mechanism contact surfaces with vaseline. This makes an enormous difference to how smoothly (and quietly!) the mechanism will run.

9) Close up the two halves of the receiver fully using 6x M3x8 bolts and 4x M3x10 bolts, screwing directly into the plastic. This completes the assembly of the T-wist mechanism.



10) Make sure your SR6's ESP32 is running *SR6-Alpha4_ESP32.ino* or later.

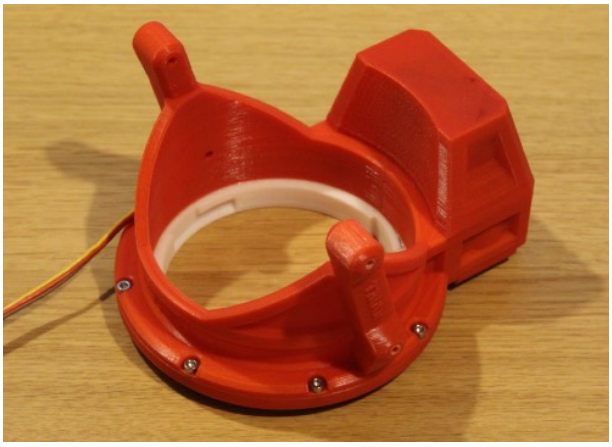
11) Hook up the servo to D27 on the ESP32 and the power bus in the same manner as all of the other servos in the SR6.

12) When you power on the SR6 the T-wist should now respond to R0 T-Code commands. You can test this by issuing these commands using something like the Arduino Serial monitor or the MOSA web app. https://trymosa.netlify.app/.

With the T-wist functioning you can now install the receiver onto the SR6 in place of the old receiver.

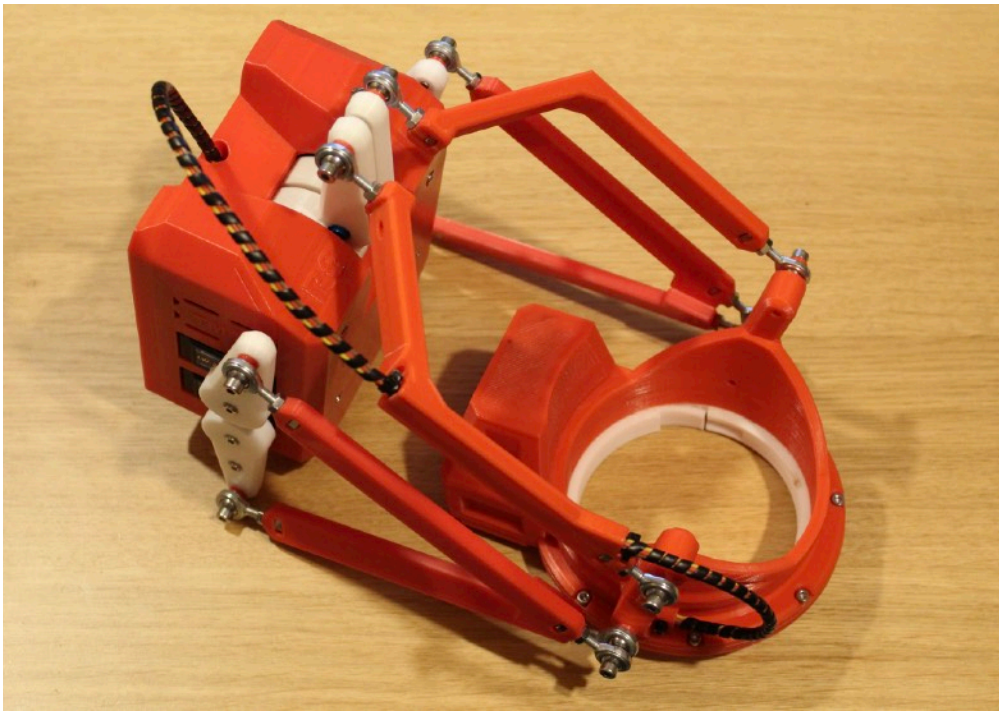
13) Feed the servo cable extension down the conduit on the side of the receiver and connect it up to the servo.

14) collect up the spare cable and tuck it inside the lid, which you can then clip down.



15) Install the twist receiver onto your SR6 in place of the original receiver. Replace the standard pitcher arms with the L-shaped T-wist pitcher arms.

16) Use cable ties and spiral cable binding to route the cable back to the SR6 enclosure. Use gentle curves to avoid putting excessive strain on the cable during use.

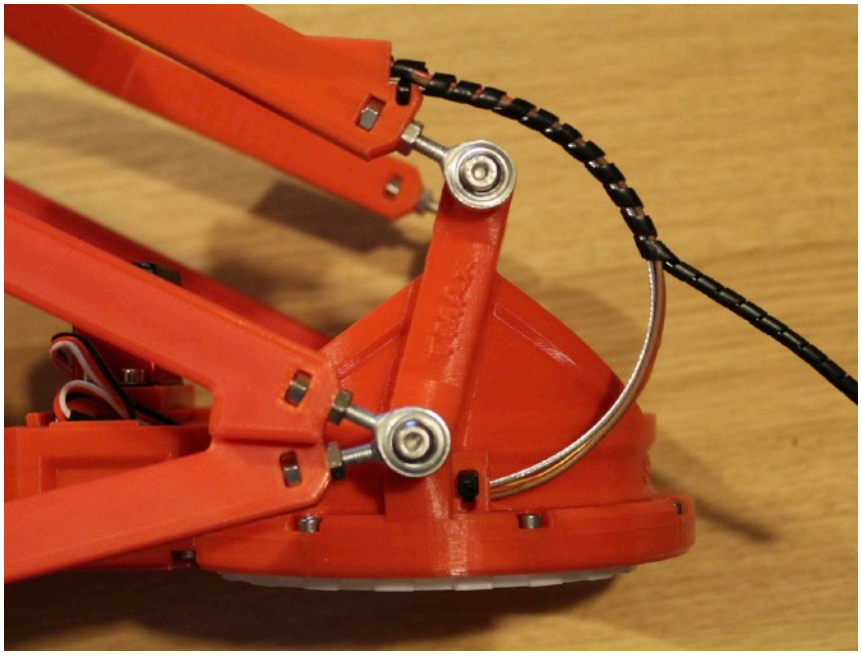


This completes the T-wist assembly process.

T-wist cable routing

A frequently occurring problem with previous versions of the T-wist was wires going to the servo on the receiver becoming damaged by the movement of the device. This was compounded by the usual point of failure being where the wires met the servo itself, with the specialised parallax servo being particularly expensive and difficult to get hold of.

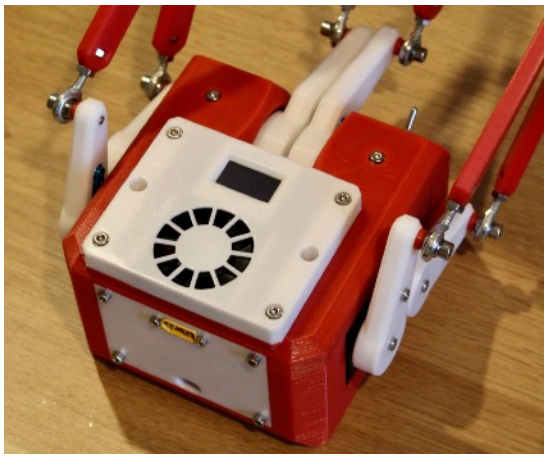
This new version of the twist receiver addresses this problem in two ways. First, it uses a generic servo and the servo wires themselves do not see any bending. It's just the servo extension cable that bends and this component is a lot less expensive to replace.

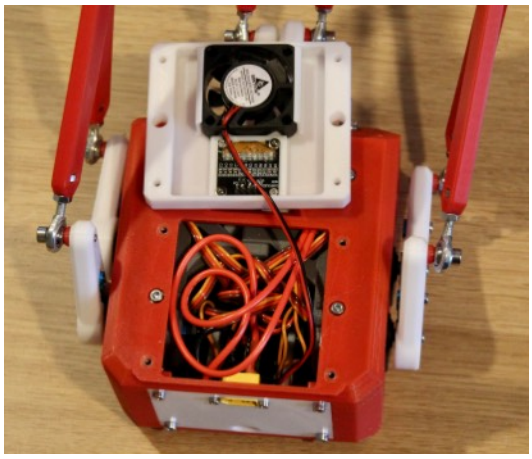


Secondly, the new receiver routes the extension cable out of the front of the receiver. This is so that the bend radius for the cable is nice and gentle and therefore in theory it should not see any harsh localised bending. Added to this the best way I have found to protect the extension is to use spiral cable binding to wrap it up with a tougher piece of wire, for example a length of bicycle brake cable.

SR6 Shields & Trays

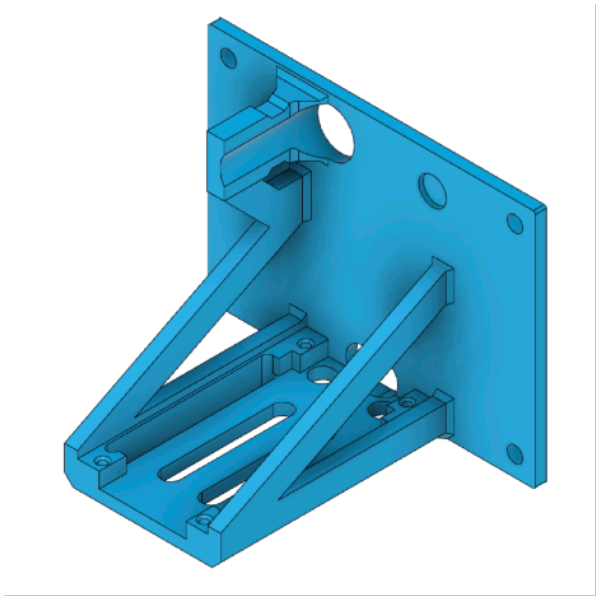
The SR6 has been designed to make custom modifications easy without having to make changes to the primary structure of the machine. The parts designed for customisability are the Tray, which holds the SR6's electronics, and Shields, which are an optional part that can be installed onto a modified lid. This pack includes several different options for each.

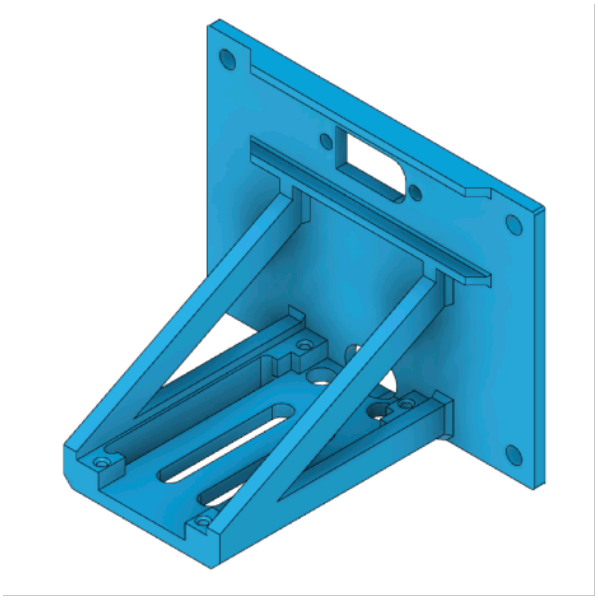




Trays

The SR6's electronics are mounted to a component called the Tray. The design of this part dates back to the SR6 Alpha, so shields from the SR6 Alpha can be used on the SR6 Beta. The current standard tray holds an ESP32 and has a tab that a female jack plug can be attached to using a cable tie. There is also still the outline of the power bus switch hole, which is covered over now that this fixture has been moved to the upper corner of the lid.

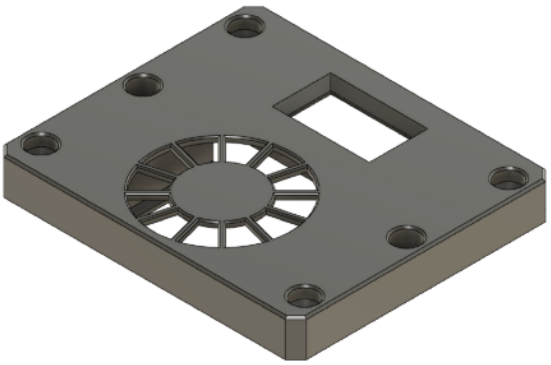


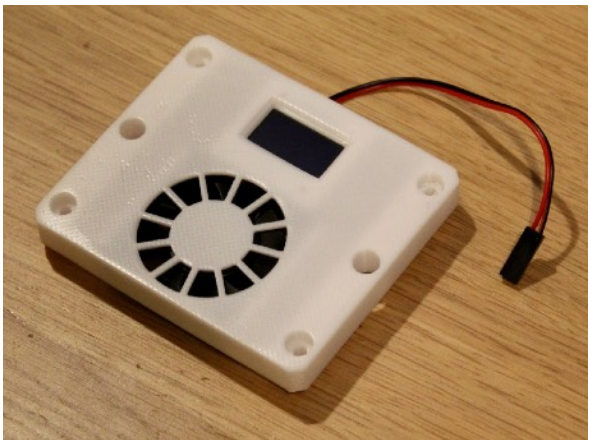


Included with this pack are a couple of alternative versions of this part. There is a hole-mounted jack plug variant, and a variant that has a hole for an XT60 connector (XT60E1-M). The XT60 connector is used extensively on applications like LiPo batteries for drones and it creates a solid connection well suited to carrying large currents. I think it is therefore ideal for your SR6, as long as you have the confidence to modify your power supply cable.

Shields

This pack includes an alternate version of the lid that features a mounting window for a shield. A shield is a customised panel that can be used to mount additional accessories you might wish to add to your SR6 build. For example, a cooling fan or a small OLED display.





Included in the pack is a tray designed to hold a 40mm fan. There are also two additional versions of this shield with a mount for a 0.96" OLED display. The display is not yet supported by the official firmware but it is popular with SR6 modders.

Members of the community are encouraged to create their own shields and share their designs on the Discord server. For this reason a STEP file of a blank shield is included in this release, which you can import into whatever CAD package you are working with and cut down to whatever height you might need.

A word about personal safety

The SR6 is an adult toy and that means you need to be an adult and make sure that you play safe. This is not a finished consumer product, this is a piece of experimental tech that you put together yourself. Electrics can overheat, motors can exert a lot of force, materials can be sharp, microelectronics can be unpredictable, etc, so be careful!

I suggest you take some time to understand how the SR6 can move before use. Make sure that you remove any sharp edges that may be created by your 3D printer with a craft knife or file. Make sure that the wire that you used for the power bus is thick enough to handle the current being used. I also highly recommend that you don't leave any hardware unattended whilst powered up, especially if you are using "budget" components. With servos six it's a lot more likely that you might have a bad apple.

Overall I take no responsibility for any actions that you may take after reading this document. It is your responsibility, and yours alone, to use any information that you have received here in a way that does not risk your personal safety.

Have fun; be safe!

And finally...

These notes contain everything you need to get started and build an SR6.

If you're having issues with your build I recommend getting yourself on the Tempest Discord server for help. I am on there a lot, as are a lot of other helpful enthusiasts. I must say I also get a real kick out of seeing the build pictures in the #show-and-tell channel. If nothing else a bit of feedback goes a long way.

I hope you get a lot of enjoyment out of your SR6 Beta. Do stay tuned for further developments as I am constantly looking for ways to improve my designs and I think there are many updates for this machine yet to come.

All the best!

Tempest