August

Hello Benefactors. Here's what I've been working on in August.

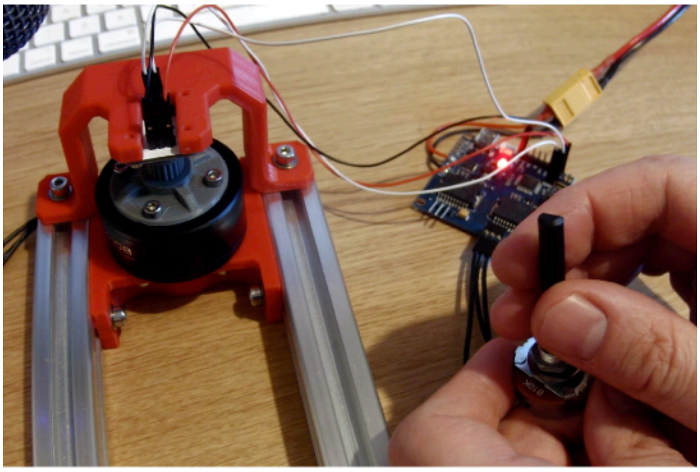
If I'm honest, this last month has been a perfect example of how my creative impulses are a law unto themselves. Anyone who looks back at my public update at the beginning of August will be able to see that my plan for the month was to release an updated random stroker VaM plugin. Immediately after I had posted that I felt compelled to finally get the organically shaped receiver that I teased in January finished.

Whilst the receiver has taken the lion's share of my effort I have also continued to tinker with and expand my knowledge of brushless motors. I've also had another quick look at squeezer mechanisms, at crimzzon's request, and shared my existing concept design.

Brushless Motors update

Over the last month or so I've been working through some of the SimpleFOC tutorials and I have to say so far I'm very impressed. You can check these out at the SimpleFOC website.

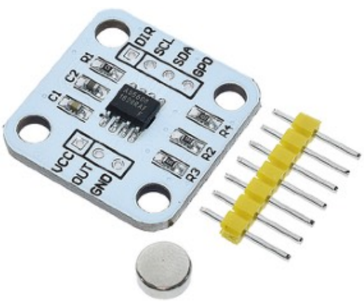
To use the technique called "Field Oriented Control" you need a suitable motor, some kind of feedback sensor, and a motor controller, which may or may not be integrated into a board with a microcontroller. The motor controller has proven to be the most problematic of those three parts. SimpleFOC offer a selection of Arduino shields and breakout boards, but frustratingly none of these are available at the moment due to current massive global supply chain issues.



TESTING THE GBM4108H-120T MOTOR WITH A BGC3.1 BOARD AND AN AS5600 MAGNETIC ENCODER

After a lot of trawling I was able to get hold of some BGC3.1 gimbal controllers. They seem to work okay, within certain parameters, so these might prove to be a suitable candidate for future homebuild designs.

The GBM4108H-120T brushless motor I have been doing my testing with seems to be reasonably available, so it might be a candidate for future build tutorials. It should also have enough power to take the place of a servo in an OSR2-like device, assuming I can build a functional 3D-printed gearbox. I have also been working with AS5600 and AMT103 encoders, which are also widely available. These represent two different ways of doing the same thing, which is providing position feedback, and both have strengths and weaknesses.



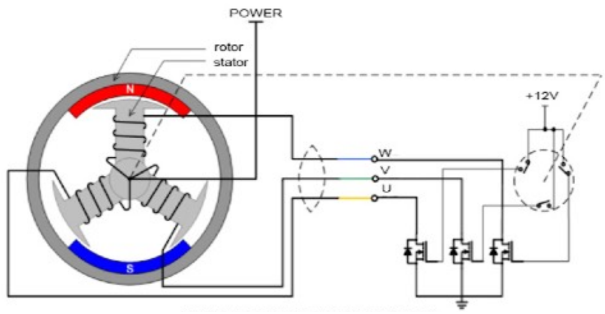
AS5600 MAGNETIC ENCODER

AMT103 ENCODER

The most significant progress that I've been able to make has been getting the motor to operate in a closed-loop mode using the SimpleFOC libraries. This has been the usual story when it comes to working with someone else's code, which is that it's great to have a powerful pre-existing toolkit at your disposal, but on the other hand it can take a while to learn how to use it, how it works, and what its idiosyncrasies are.

As I've discussed in previous dispatches, Brushless motors use sophisticated control electronics to activate the windings in the motors in such a way as to very precisely control the orientation and strength of the magnetic field inside the motor. This is why it's called Field Oriented Control.

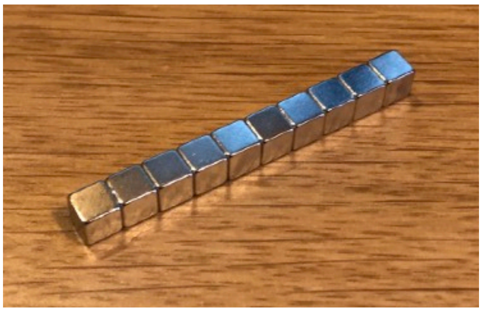
One way to drive these motors is to simply perform a dumb rotation of the field, which will cause the rotor, the moving part of the motor, to turn with it. This is what I have been able to achieve before this last month. The problem is that there's no way of knowing if the rotor is following the field, and so there's no way to gauge the appropriate field strength. This means that the field will inevitably be too strong, creating excessive heat, or too weak, causing the rotor to lose its position.



BRUSHLESS DC MOTOR WORKING PRINCIPLE

Closed loop operation is when a sensor is used to feed back the orientation of the rotor. This means that the field can be reoriented hundreds or thousands of times per second by the control system. This allows for extremely efficient control of the motor, only ever applying as much power to the field as is required. It also allows for accurate, absolute position control, which is exactly what we want.

In order to achieve closed loop control a sensor is required and this has been the main source of my problems. I started out using an AS5600 magnetic sensor. This uses hall effect sensors to detect the orientation of the motor via a magnet attached to the end of the rotating shaft. It interfaces with the microcontroller using an I2C connection and doesn't have the best written documentation. I struggled for weeks to get this to work without success.



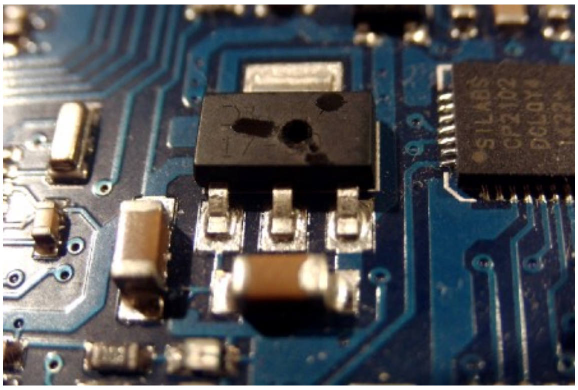
5MM NEODYMIUM MAGNETS

The fact that the AS5600 is very inexpensive and easily acquired off Amazon or eBay fuelled my paranoia that I was dealing with cheap Chinese knockoffs. I tried buying another from a more reputable source at greater expense, but to my frustration I saw exactly the same error with the new component! In the end I realised that the angles being reported by the sensor were uneven as I rotated the shaft through a full turn. This led me to suspect that the problem was with the magnets. Sure enough, when I replaced the cheap chinesium magnets that came with the AS5600 with some 5mm cube neodymium magnets the sensor instantly began to work perfectly.

My second problem with the AS5600 appeared when I tried to drive the motor at speed. When I did this I started to hear an electronic buzzing noise; frustrating given that one of the main attractions of these motors is how quiet they can be. It turned out that this is an artefact of the way the I2C control loop works. Every cycle of the motor control loop the microcontroller sends a request to the AS5600 for the current position of the rotor, which it duly sends. This process slows down the control loop so that each cycle takes about 2 milliseconds, which sounds fast but 500Hz (500 cycles per second) is right in the middle of the human hearing range. This results in a "staircase" effect on the updates to the magnetic field, which doesn't matter when the rotor is moving at low speed, but creates an audible vibration at higher speeds when the rotor has moved a more considerable fraction of a field cycle in 2ms.

The alternative to the AS5600 is something like the AMT103 encoder. This is a very nice piece of kit, though it is larger and more expensive. This device functions as a traditional encoder, with two digital output channels that go low-high-low for each step of rotation, and the number of steps per rotation (e.g. 1000) can be defined by flipping tiny switches inside the device. This means that there's no delay in communicating with the device, instead interrupt pins on the microcontroller are used to monitor rotation. This eliminates the electronic buzz. The downside of this device is that the absolute position of the rotor isn't known.

I discovered one of the limitations of the BGC3.1 very abruptly about a week or so ago. The GBM4108H-120T motor is rated to run at 20v, so I tried turning up the power supply to 16v to see how this affected the performance of the motor. It all appeared to work just fine, with the motor running noticeably faster, right up until the point when there was a loud bang and I saw a glowing spark fly up into the air. I quickly realised that this was the destruction of the 5v voltage regulator, which powers the microcontroller in the absence of a USB connection, and must be rated up to 12 volts. I have subsequently replaced this blown out component, but sadly the board appears to have died. In short, if we are going to use the BGC3.1 for future devices we will be limited to 12v.



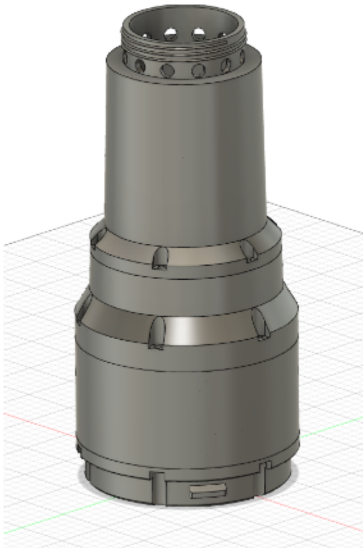
THE 5V VOLTAGE REGULATOR WENT OUT WITH QUITE A BANG, LEAVING A SMALL CRATER

At the time of writing I am continuing my research by building a custom breakout board for a L6234D. This should help with my experimentation, though it's obviously a non-starter for a future homebuild design.

Squeezer mechanisms

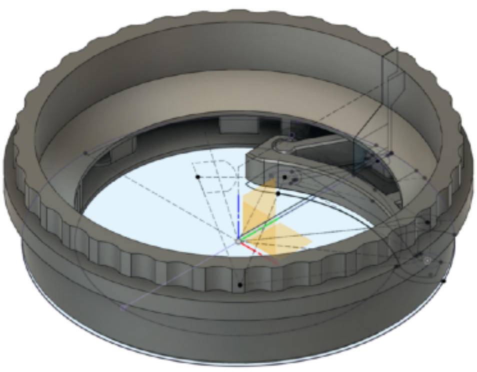
Earlier in the month I was approached by crimzzon who is interested in developing a squeezing mechanism. He asked me if I would be willing to share the Fusion360 files for my squeezer concept designs.

I originally developed these two concept demonstrators over a year ago. The first is based on the Realtouch orifice mechanism, and the second is inspired by the grip mechanism from the Space Shuttle's CanadaArm. They've been on my to-do list for a long time, but I've held off developing them since I came up with the new twist because of the extremely high torque that the new 1:1 drive mechanism can exert. I want to be sure I'm not creating a penile chinese burn-o-matic.



MODCASE

Add-on mechanisms like this are one of the reasons that I developed the modular case or "ModCase" as a substitute for the standard fleshlight case. The .STEP files for this have been available since I released it.



THE MK1 SQUEEZER CONCEPT

I have shared the F3D files for the squeezer mechanisms in the devstream on Discord. crimzzon is definitely one of the more innovative members of our little community, so I'm looking forward to seeing what he comes up with.

The Contour Receiver

Back in January I talked about what a next level machine from an OSR2 or SR6 might look like. I think that this might be a machine where the moving parts look like a female lower half, or maybe something like a disembodied pair of cycling shorts that straddles and rides the user. Essentially a human female shape from the waist to just above the knees.

In a world where homebuild aeroplanes are a real phenomenon, more complex sex robots are definitely a possibility.

We're a little way off the day when I release the plans for such a machine, but I find the concept useful because it gives me ideas for technologies to explore. This is an incremental process after all.

The contour receiver that I released last month is an example of this kind of exploration. My idea was to mount something a bit more human shaped onto an SR6.

THE SHAPE OF THINGS TO COME?

AN .OBJ FILE OUTPUT FROM VIRT-A-MATE

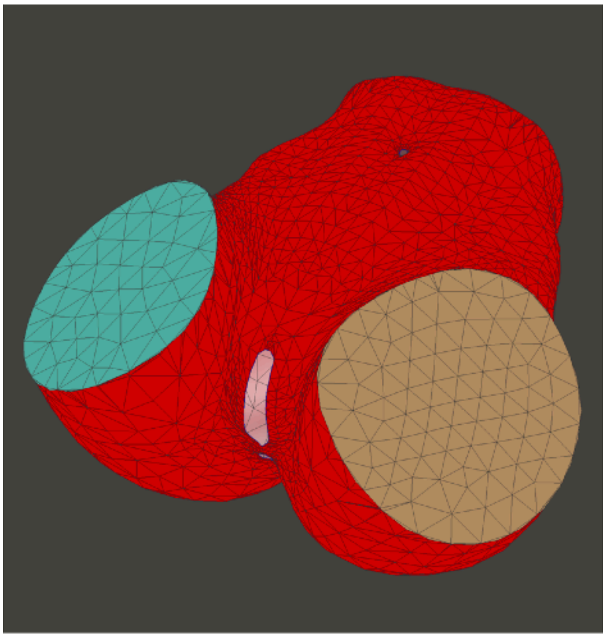
I was talking about next-gen devices in January because at that time I was investigating how easy it was to take objects out of Virt-a-Mate and 3D print them. It is certainly possible to export characters from VaM and these come out in the .obj file format. It isn't possible to output hair, or any items of clothing that the character might be wearing.

Making use of the model is where things become a lot more difficult, because a 3D model designed for computer graphics is far from being a solid object, or the design for one. It's certainly not simply a case of dropping the model into slicing software and hitting the "print" button.

The first thing to contend with is that the model is not a solid, but rather a collection of surfaces. There are holes in the model where, for example, the eyes are. There are surfaces that exist purely as 2D cosmetic features, such as the eyelashes. There are also disconnected models that are independent of the main model, such as the eyes.

In order to get on top of this it is first necessary to load the model into a program like Autodesk MeshMixer. This is an unwieldy but powerful tool that makes it possible to edit the mesh down to something that's a lot more manageable.

In the case of the receiver I wanted to build I only needed part of the model, so I could happily remove large parts of the model, including the head with all of its complexities. Meshmixer allows you to cut and remove parts of a mesh using a plane, so that made it easy to get rid of the legs and everything above the waist. Less easy was the necessity to remove the complex geometry around the genital area and the bellybutton.

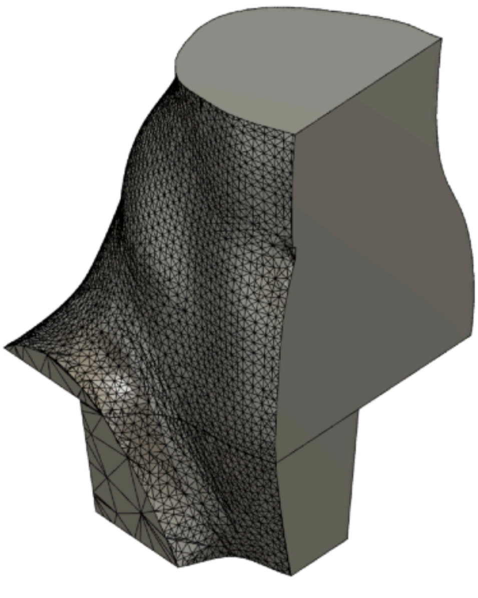


THE CUT-DOWN MESH AS SEEN IN MESHMIXER

For a 3D model created for computer graphics the mesh tends to have minimum number of surfaces possible because it makes rendering easier. It's possible to cheat and use textures and normals, etc, to hide that the apparently curved object is actually made up of flat surfaces. This doesn't work when you take the same object and make it physical. My previous attempt at printing this particular mesh gave me what I called "Stealth Midriff", because it was made of obvious flat surfaces.

Fun fact: The F-117 Nighthawk Stealth Fighter, that monster made out of triangles that looks like it shouldn't be able to fly, is the shape that it is because of the limited power of computers in the 1980s. They had a bunch of equations that allowed them to estimate radar returns from flat surfaces, but the computer simply couldn't perform the calculations in a sensible amount of time if they used a shape with more surfaces.

Meshmixer includes a function that can be used to smooth out a mesh, which it does by subdividing existing surfaces over a certain size and spline-interpolating the nodes. Thankfully I wasn't interested in designing to confound soviet air defence radar with this model, so I was able to increase the number of surfaces to the point where hopefully the surface would look and feel smooth but wouldn't crash Fusion360. To give you an idea of how the number of triangles increased, the mesh subdivision increased the file size from 63KB to 2.1MB.



THE IMPORTED MESH CONVERTED TO A SOLID BODY AND CUT INTO TWO PIECES

With this done I could insert the modified mesh into Fusion360, cross my fingers, and then use the "convert mesh" function to turn it into a solid body. When you use this feature F360 uses an automated process to fill any empty holes in the mesh. The results of this aren't optimal, but apart from the belly button the affected areas wouldn't feature on the finished design, so this wasn't an issue.

Once I had the 3D shape I wanted in SolidWorks as a solid body it was simply a case of using regular 3D CAD design methods to turn it into a finished design. The same rules applied: for example, if your design is symmetrical only design half of it and mirror it.

I decided that I wanted to produce the design as effectively a regular SR6 receiver modified to carry the midriff shape as an add-on section on the top. The top section would have to be in two halves to fit on most 3D printers, so I called these sections "wings". I chopped my starting shape into appropriate pieces and got to work.

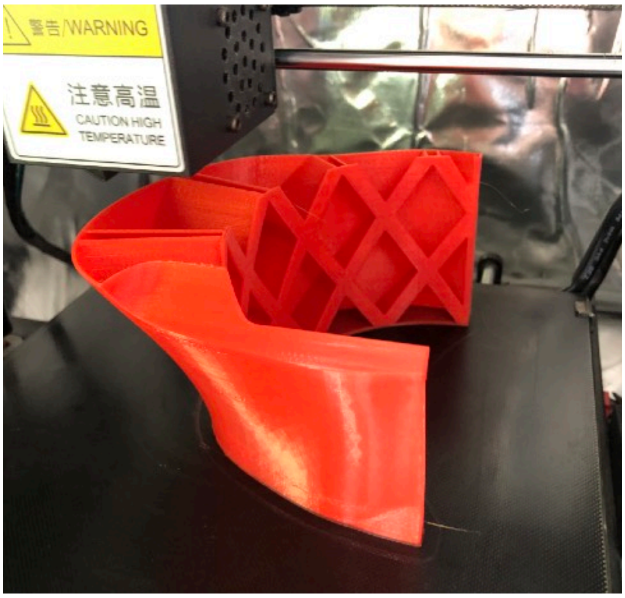
For the shaped parts of the design I wanted to use the same hollow lightweight structure approach as I used on the ModCase design. This actually presented the most challenging part of the CAD design. When I tried using the “shell” feature F360 ceased all activity and went into the corner to huddle up and cry to itself.

I had to come up with another way of achieving the same result. I needed to create a copy of the solid body, scale it down in both horizontal directions, and then use a boolean operation to subtract it from the original. I have definitely found myself increasingly using such body operations more and more in recent times.

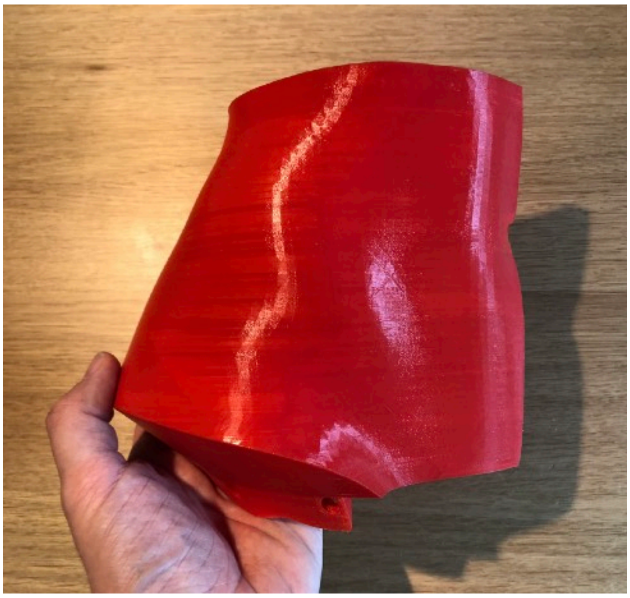
By the time that I had finished the design of the upper and lower parts of the contour receiver it was clear to me that these were some of the most intricate parts that I have yet designed. Part of this was just the number of hours I had to sink into the design itself.

Another issue was how long it took to print the parts. The wings are relatively light, at less than 150 grams each but because most of that is perimeter, and I tend to use low printing speeds for perimeters in order to get a good surface finish, the print time for each wing was 30 hours. As you can imagine, iterating the design wasn’t exactly a rapid process!

THE STRUCTURE OF THE CONTOUR RECEIVER



ONE OF THE PROTOTYPE WINGS ON THE 3D PRINTER



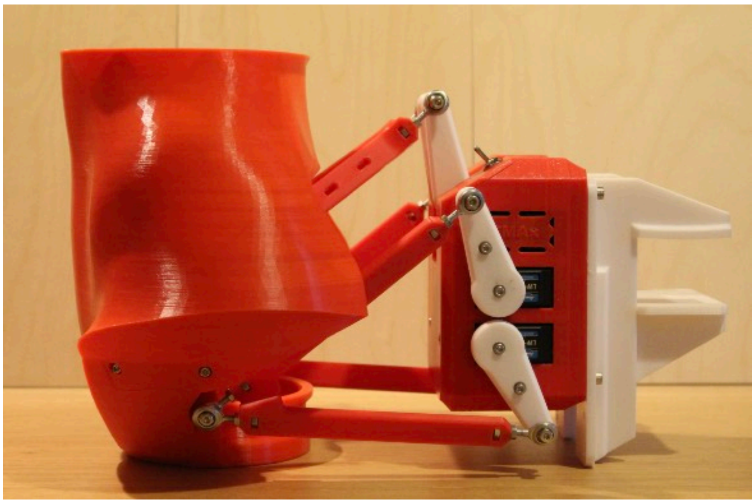
The first finished wing

I have to say though, I was very pleased when I finally got to put together the three parts and I was holding the first finished contour receiver in my hands. The facets are still clearly visible on the surface, but crucially they can’t be felt. I found it a very satisfying shape to feel in my hands, and all the more surreal for being something that previously had only existed virtually.

Is it, I mused to myself, cheating if you get defined abs by 3D printing them?

The final test of course was to check that it would be able to perform the full range of movements for the SR6 without the various links fouling on the wings. In the end I had to adjust a few parameters from the initial design, but no reworking of the model was required.

ALL THREE PARTS FIT TOGETHER VERY WELL



THE CONTOUR RECEIVER MOUNTED ON AN SR6

Very soon after printing and assembly I had the contour receiver running on an SR6 and mounted to my desk using the 3D printed VESA clamp.

Feeling it move was actually one of the most compelling experiences I've had since I started developing these machines. I had two thoughts: one was expected, the other surprised me.

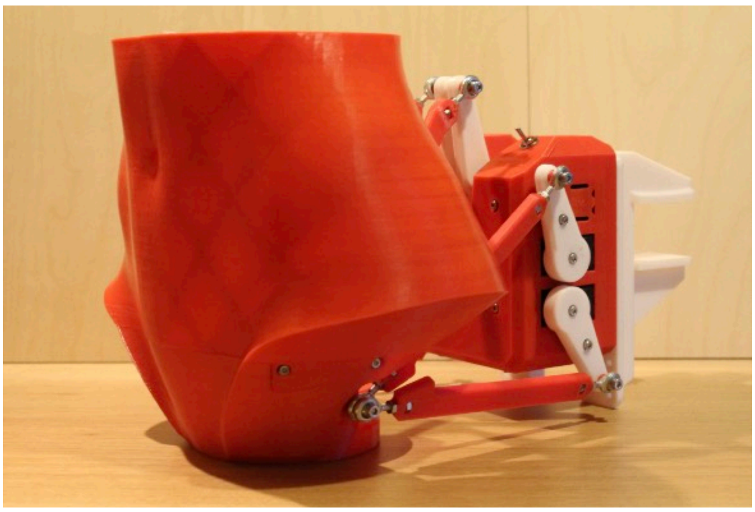
The first thought was that resting your hands on a pair of hips can bring a whole new level to the feeling of a multi-axis movement. The Kayla model I used actually has quite prominent hip bones, which adds to this experience.

The second thing, which I didn't expect, is that there's a different, more peripheral, feeling when there's a body moving in front of you, instead of just a receiver and a fleshlight. One can only imagine that this could be magnified further by potentially adding thighs and replacing the hard plastic with something softer, but those are design challenges for another day.

Developing this receiver has definitely been an interesting experiment and I'm looking forward to seeing the community reaction. I can innovate all I like, but at the end of the day it’s my userbase who really have the final say on what I work on.

The final thing to say is that I’m aware that “contour receiver” is perhaps quite a dull name for this project and I should have thought of something better. For example, Mrs Tempest wanted to me to call it the T-ummy.

This is why Mrs Tempest doesn’t get to name things.



THE CONTOUR RECEIVER MOUNTED ON AN SR6

Finally...

I’d like to say a big thanks to you guys for the support you give me. I’m actually enjoying sharing this glimpse into my creative process with you, so I hope that it’s interesting to see and read. I would like to invite you guys to engage more on the discord server, especially if you see something I’ve posted and you have any questions or suggestions. I’m also definitely open to organising livestreams and hangouts on there too if there’s anything particular you’d like to see more of. Let me know.

Thanks again!
Tempest