

T-Vibe Instructions

Tempest, 4-11-21

The T-Vibe, or T-Code Vibrator controller, is an open source motor controller for sex toys. It is a simple, low cost, reliable way to control a small vibrator directly via a USB connection from your computer.



The control protocol used, T-Code, is supported by several media players and thus compatible with standardised scripting languages. This means that it's possible to hook up any wired vibrator of your choice and have it sync to the media file that you are watching or listening to. Full speed, off, and any speed in between.

The T-Vibe is designed to be put together with no soldering required and with a minimal understanding of electronics and programming. These instructions will walk you through the process step by step. The box is a two-piece 3D printed design and all of the other parts used are easily available online.

Finally, for those of you who are a bit more confident and want to build, tinker and explore, the T-Code infrastructure exists so that you can do just this! The code is Arduino based and can equally well be run on an Uno with a motor controller shield, for example. If you want to design and build your own computer controlled toy T-code gives you an easy toolkit to do just that. Those of you who are really adventurous might want to check out my robot designs, which I will mention later.

For now though let's keep things simple and get your vibrator hooked up.

Parts List

Vibrator







Needless to say you will need some kind of vibrator. There are countless generic wired battery powered vibrating bullets available at very low cost. What you want is a “dumb” device with no clever electronics of its own. Literally a battery pack with a switch, wired to a motor.

As a general rule the Romeo’s motor controllers should be able to handle anything that runs off AA batteries or equivalent. If you want to use anything more powerful (eg 12V motors from a massage cushion or a Hitachi magic wand) this is completely doable but you will need some more robust electronics that are beyond the scope of this guide.

Off-the-shelf parts

The components listed here should be easily available to purchase online or from your local hardware store.

	Part	Qty	Description
	Romeo BLE Mini v2 or v1.1	1x	An Arduino based microcontroller that can read incoming T-Code commands and interpret them. The T-Vibe is designed around the Romeo BLE mini. This is because it is compact and avoids the need for soldering.
	Micro USB cable	1x	To connect to your computer.
	Bolts: M3x6mm	3x	Metric threaded bolts: 3mm wide by 6mm long. These hold the Romeo in place. I use Hex Socket Cap Head bolts with an Allen Key, but any type is fine.
	Cable ties	3x	Small cable ties Used to secure the wires to the box.

3D printed parts



These components are not strictly necessary to get the device working, but they do hold it all together and surround it in a protective case. I have deliberately not branded the box with “HELLO WORLD! I’M A SECRET VIBRATOR!”, so it should be fine to print this through a friend, family member or maker space without too many questions. If you don’t have access to a 3D printer there are printing services available online.

These should be printed in PLA or ABS. The part files are provided in the print orientation and are designed to be printed with supports turned off.

	Part	Qty	Description
	T-Vibe Box	1x	This is the main enclosure that holds the Romeo and associated wiring.
	T-Vibe Lid	1x	A cover that clips into place over the box.

Optional extras

If you want to save money on batteries you can use a mains power supply to power your vibrator.

	Part	Qty	Description
	DC power supply with voltage selection	1x	<p>Small DC power supplies with a selectable voltage output are easily available these days. Select the same voltage that your vibrator uses.</p> <p>2x AA batteries = 3V 3x AA batteries = 4.5V 4x AA batteries = 6V</p> <p>Even small power supplies these days are capable of supplying 2 amps (2A) and this is more than enough to power a small vibrator.</p>
	5.5x2.1mm female barrel jack connector	1x	<p>A female connector with a set of trailing leads. This is easier and more convenient than cutting the wires on your power supply.</p> <p>Exact details are not important as long as it is compatible with your power supply.</p>

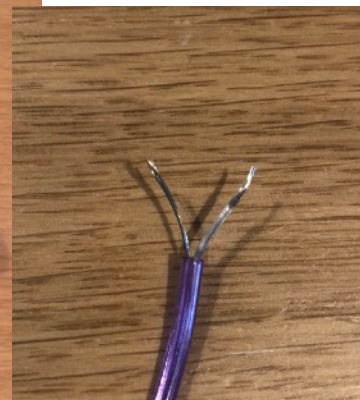
Assembly instructions

For assembly you will need your vibrator, the listed components, and the appropriate screw drivers and/or Allen key, depending on which bolt type you're using. You will also need a craft knife or wire stripper.



1) You will have to cut the cable between the battery pack and the vibrator egg, separating the two devices. This should be done with thought and care because it can't easily be un-done if you do it in the wrong place. My advice is that you leave most of the cable on the vibrator, with about 10cm of cable left on the battery pack.

2) Once you have cut the cable set the battery pack aside for a moment. Take the cable end that connects to the vibrator egg, separate the cable into two wires and strip the two ends.

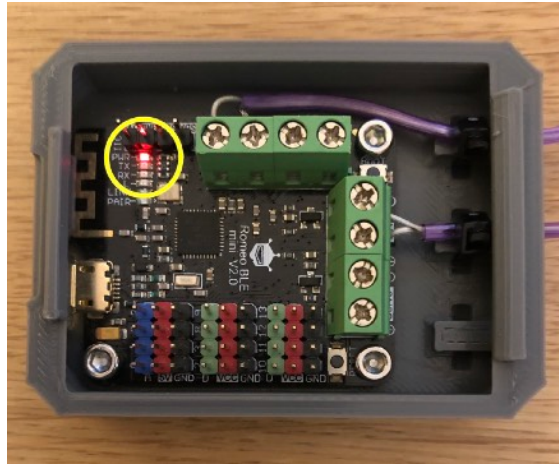


3) Insert the cable through one of the holes in the box, outside to inside, and then install the two wire ends into the screw terminals on the Romeo marked "MA1 and MA2". Which way around you insert them will determine the direction that the vibration motor will spin. Usually this doesn't matter!

4) Take the battery pack and separate and strip the wires in the same way as above. Alternatively if you want to use a power jack adaptor you can use that instead for the next steps.

5) Prepare to attach the battery pack to the screw terminals marked "VIN". This time it does matter which way around the wires are installed. The positive (+) and negative (-) wires must go to the appropriate terminals.

Usually the wires are not marked. By far best way to tell which is which is to turn the power pack on and use a multimeter, if you have one. If you don't have a multimeter what you can do is turn on the power pack on the lowest setting and tap the wires against the VIN screw terminals. If the wires are the right way around there is a red LED on the board that will light up dimly. If they are the wrong way around the LED will not come on. Do not hold or connect the wires for more than just an instant if the LED isn't coming on!



6) Once you have determined the correct way to attach the battery pack wires insert the cable through one of the holes in the box, outside to inside, and then screw the wires into the “VIN” screw terminals. Check that the red LED comes on when you turn on the battery pack. If it doesn't, turn the power off immediately!

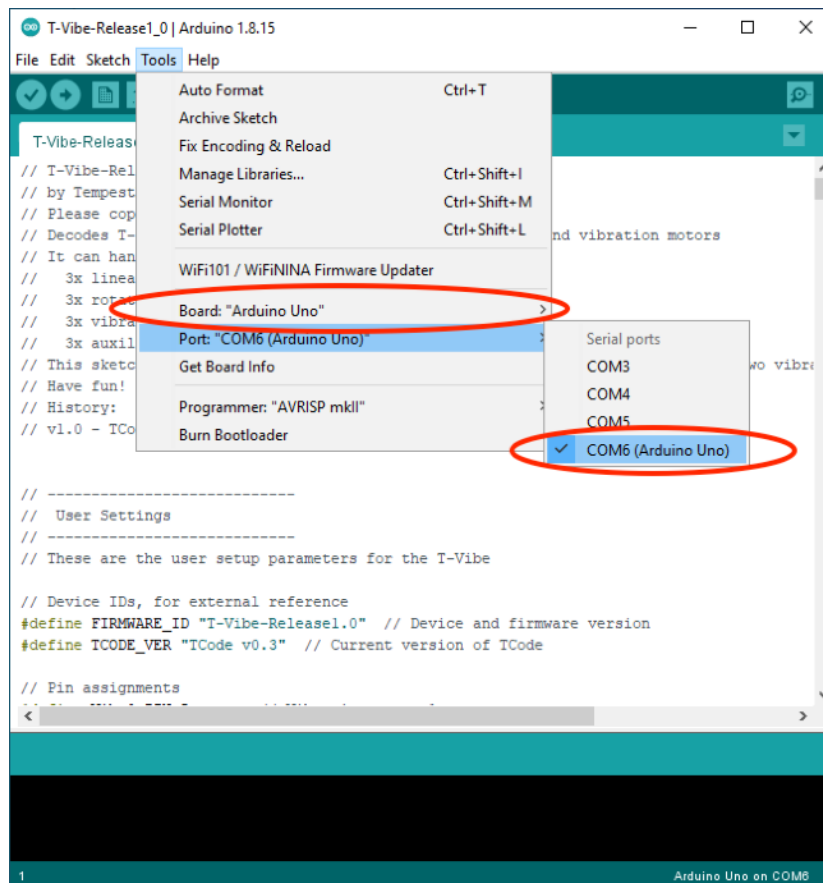
7) With the wires now connected, insert the Romeo into the box and secure it in place with the 3x M3x6 screws, which screw straight into the plastic.

8) Secure the cables in place inside the box using cable ties. This protects the wires from being accidentally pulled out of the screw terminals.



9) Plug the Romeo into a USB port on your computer via the micro USB cable.

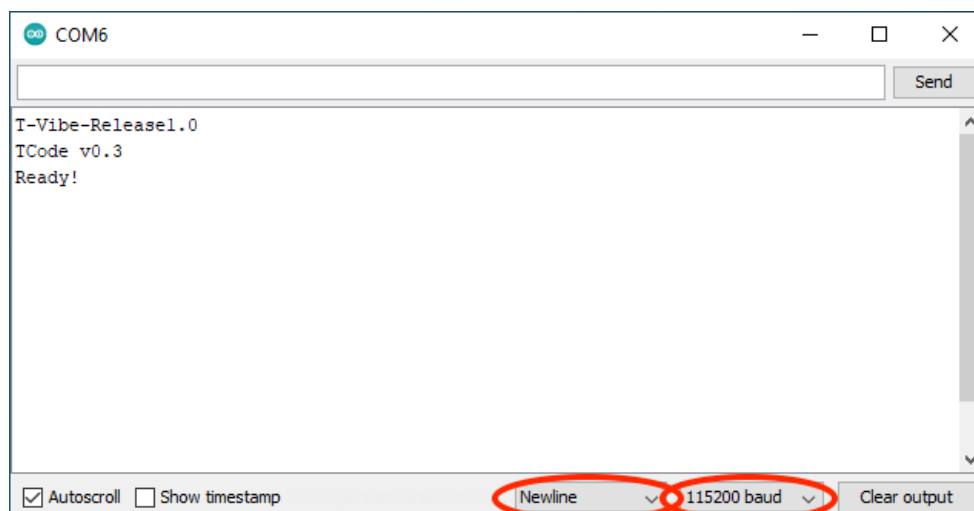
10) Open the Arduino IDE and load the T-Vibe sketch (.ino file format). From the Tools menu check that the Board is set to “Arduino Uno” (exact text can vary depending on your OS), and check that the Port selected is the one that is connected to the Romeo. Then click the upload arrow in the top left.



You should see lights flashing on the Romeo as the upload occurs and after about 5 seconds you should see a message saying “Done uploading”.

If the upload process is unsuccessful the first time try the upload button again. Ensure that you have the Board and Port set correctly.

11) In the Arduino IDE Open the Serial Monitor from the Tools menu. In the new window that opens ensure that “Newline” is set and “115200 baud” is selected. This should result in three messages beginning with “T-Vibe” appearing in the window. This tells you that the sketch has been uploaded successfully to the Romeo, and that it is talking to your computer.



If you do not see anything, or the message is garbled, try unplugging the Romeo and closing the Serial Monitor, plug the Romeo back in and re-open the Serial Monitor.

12) Turn on the battery power to full power, or attach the power supply. Try entering some T-Code commands into the text box and hitting "Send":

- "V0999" should switch on the vibrator at full power.
- "V0500" should switch on the vibrator at half power.
- "V0000" should switch the vibrator off.

Note that the vibrator will also time out and switch off after an interval of time, which can be changed by changing a parameter at the top of the T-Vibe sketch and re-uploading.

13) Put the lid on the box. The T-Vibe is now ready for use.

Using the T-vibe with a script player.



The T-Vibe will work with any application that can send T-Code commands via a serial connection.

The current standard format for sex toy scripting is .funscript. Note that often the .funscript files that accompany hypnosis files (e.g. Shibby's files) may instead come with a .txt file extension, but the data format used is actually the same. To use them with a script player you will have to give the script file an identical name to the media it is associated with, and change the file extension to ".funscript".

There are currently two T-Code script players available that can drive the T-Vibe using .funscript files to accompany the audio or video:

[Cross-platform T-Code Player \(XTP\)](#) by [Khrull](#)

[MultiFunPlayer \(MFP\)](#) by [Yoooi](#)

The creators of these superb little apps have made them available to the community for free. If you like their work you should definitely consider supporting them on Patreon.

Cross-platform T-Code Player (XTP)

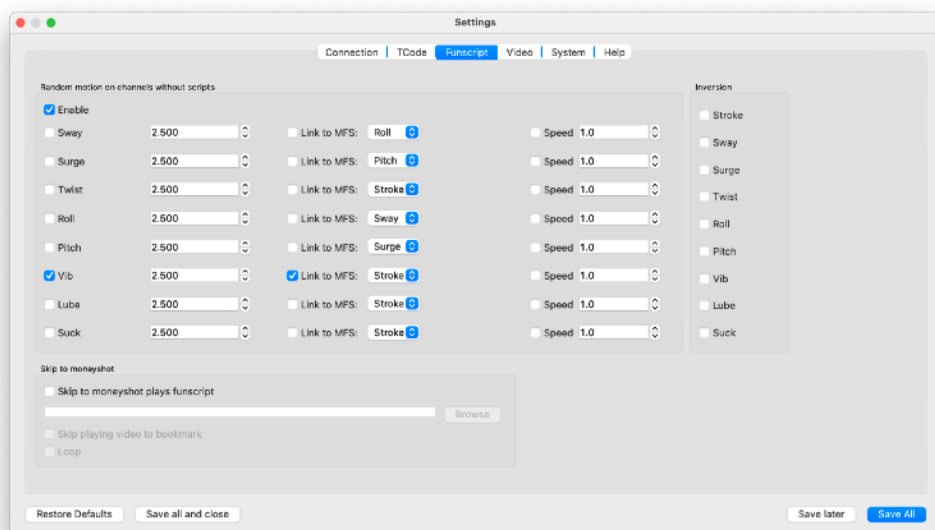
XTP is produced by [Khrull](#) and can be downloaded from his [Patreon](#) page.

This program is just what it says: a T-Code player that will work on Windows, MacOS, and Linux platforms. It's a neat little player that will list your content library (video and audio) and play it back with your T-Vibe in sync.



XTP is designed to work with multi axis stroker robots, but it works just fine for vibrator-only scripts. All you need to do is make a couple of changes in the settings menu.

The app assumes that every scripted media file will have a ".funscript" file with the same filename, and that this file controls the up-down ("Stroke") movement of a stroker robot. To output the appropriate commands on the vibration channel it is necessary to click to "Enable" motion on unscripted channels, click the "Vib" channel, and to link it to the "Stroke" channel. This will duplicate the stroke channel commands sent to your T-Vibe (which it will ignore) with the appropriate Vibration commands.

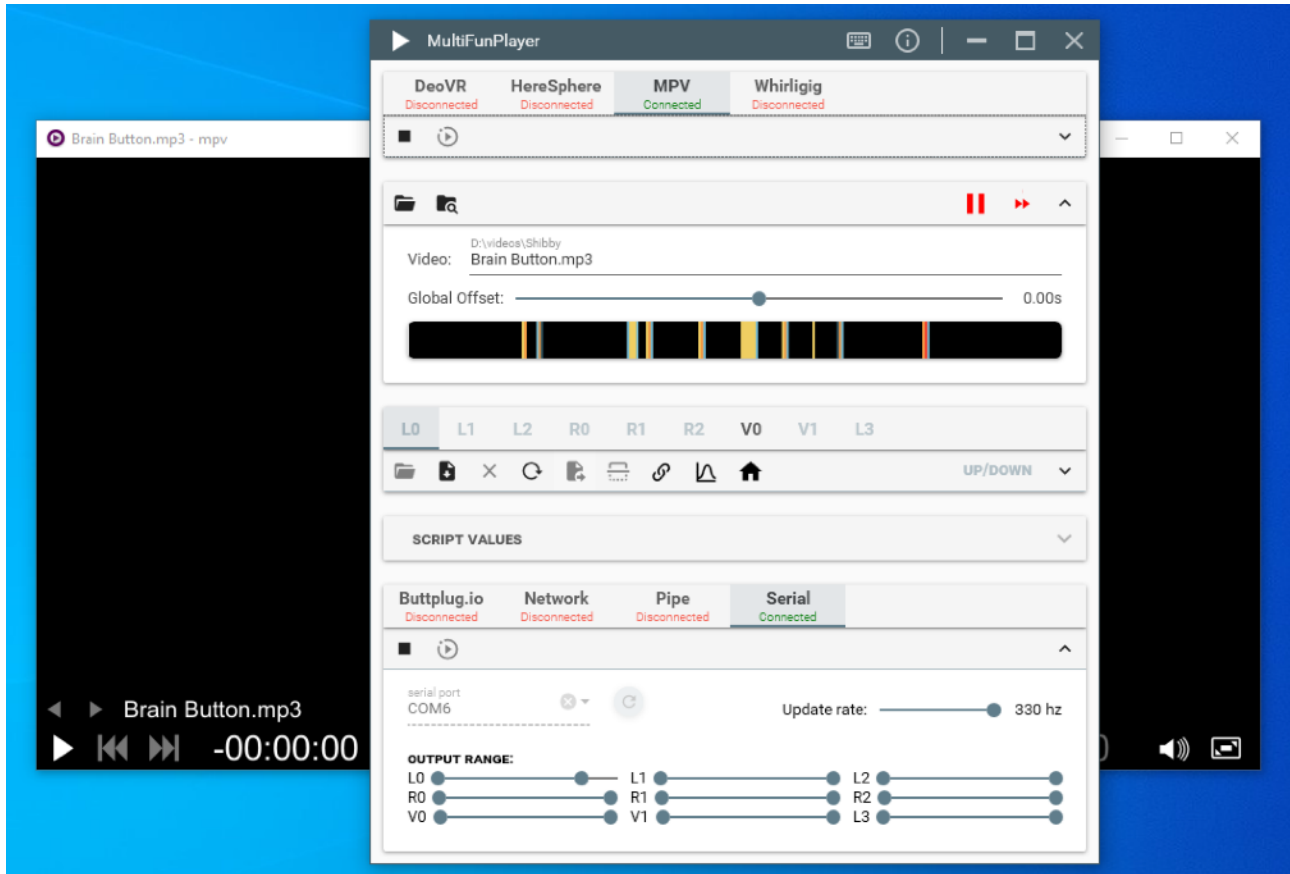


(Version of XTP shown is v0.265b)

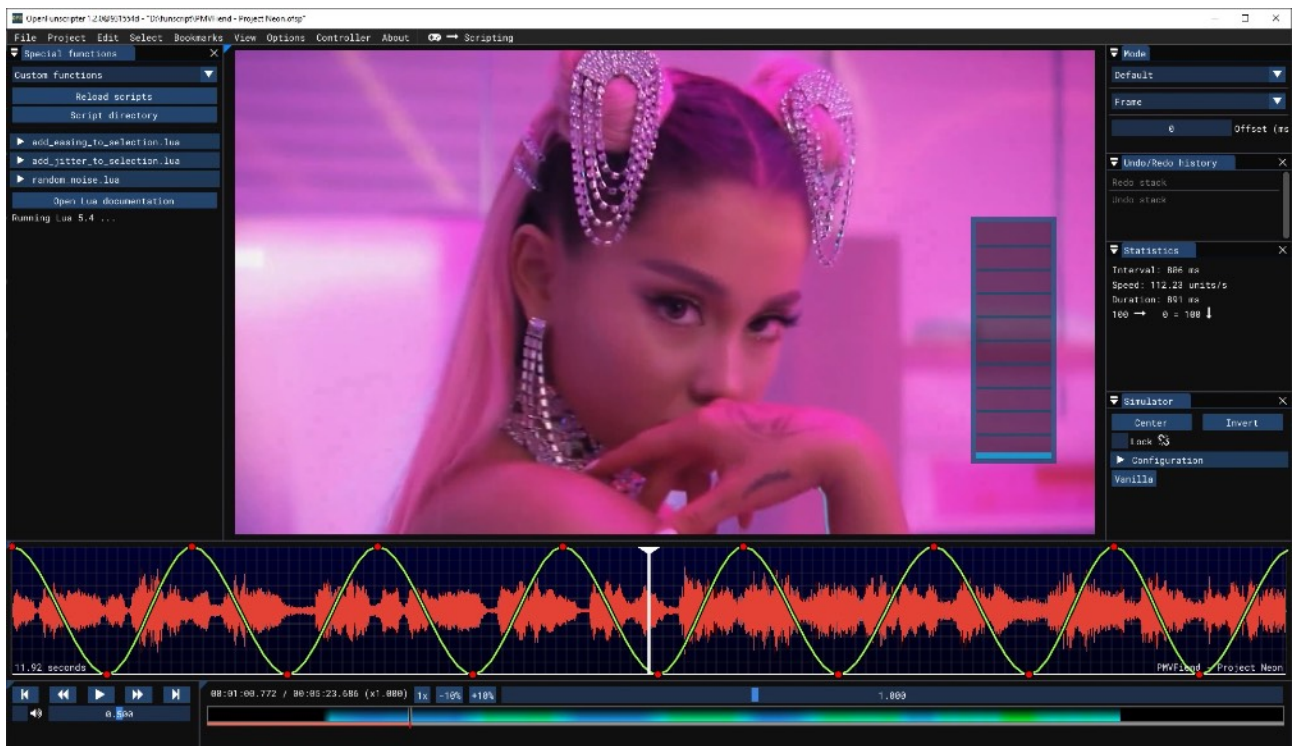
MultiFunPlayer (MFP)

MFP is produced by [Yoooi](#) and the latest version can be downloaded from [Github](#).

This player is PC only but it has a lot of cool options and interoperability.



Scripting - OpenFunscripiter



If you want to create your own script files the software to do this is available for free in the form of OpenFunscripiter. This is the work of eroscripts.com user [gagax123](#) You can download the latest version from [Github](#).

This is an incredibly versatile tool and a superb resource, well worth a look if you have any interest in creating scripted content. Sadly I'm not aware of the creator having a Patreon or similar.

OSR2 and SR6

Finally a word about me and what I do.

I'm Tempest and I create multi-axis stroker robots. My ethos is that rather than wait for somebody to create the ultimate sex toy, why not build it yourself. I create homebuild toys, or rather designs for toys that can be built, maintained, repaired and upgraded by anybody who can follow step-by-step instructions. I keep my designs simple and avoid the need for special maker skills or tools.

I have two main designs: OSR2 and SR6.



The **OSR2** (left) is an **O**pen source **S**roker **R**obot (**2**-axis). This is a machine that is designed to wield a fleshlight or similar adult toy in sync to adult content, such as .funscript videos, or adult VR games such as Virt-a-Mate. Control is by T-Code commands via USB Serial, which is far more reliable than bluetooth.

The device is assembled using only 3D printed and off-the-shelf parts. The driving force comes from two standard 20kg.cm servos, which provide more than enough strength for long and powerful up-and-down strokes. The basic arrangement can also roll left-to-right, which adds to the range of sensations possible, and a pitch module can be added to enable pitching forward and backward. Other upgrades such as a twist module and suction valve are also available.

The **SR6** (right) is a **S**roker **R**obot (**6**-axis). This machine is the OSR2's big brother, more complex but much more capable, with six degrees of freedom. That's full 3D movement: Up-down, left-right, forward-backward, pitch, roll and twist.

The T-Vibe is an easy re-purposing of technology created for my robots. I have made it available for free because I have always received a lot of help and support from the online community and I would like to give something back. I also love it when people in our space innovate and try new things, and I think a lot of cool devices could be built off the back of some of the tech presented here.

The original OSR2 design is available for free on [Thingiverse](#). If you like my work you can find more of it on [Patreon](#) or [SubscribeStar](#). My patrons are always the first to see and get their hands on my latest designs.

Have fun, and happy building!

Tempest