

What is Noxy-RED Standalone, what does it do for me?

There are three Parts on the way to understand what we are going to do:

1. Our goal is to Control Toys. Those can be a stroker, vibrator or custom Toys, either off the shelf toys or custom-made ones.
2. We want a logic that is going to introduce interactivity, giving us the option to create patterns, loops, etc. that will drive toys and control them. In the example it's going to be an edging logic with either voice or button input.
3. We want the ability to have connectivity not only for output, but also for input to control the logic. As Noxy-RED Standalone does not have the LLM/AI control part, we do this for example via a button in the web interface or Voice Control over programs such as Voice Commander etc.

Noxy-RED is a bundle that will enable you do those 3 Points.

What Software are we using?

MFP / MultiFunPlayer:

Multifunplayer is very versatile to drive all kinds of toys, compatible with both Buttplug.io and Tcode Devices alike, giving you the options to use both, motion pattern as well as scripted pattern, we are going to use it, by sending emulated keystrokes to the program to control the action.

On top of this, it also allows you to add more toys directly, and triggering them over the additional channels you are not using for your stroker or vibrator. For Example, you could add an additional Funscript, that triggers in an existing Movie at strategic points the estim. Either via UDP or Websocket.

You will find a MultiFunPlayer.config.json in the docs folder that you can use as a starting point:
Mapped are in that case: CTRL + 0 until CTRL + 8.

0 is Stop, 1-3 are assigned to Slow Strokes, 4-6 are considered Medium and 6 – 8 would be fast enough to make the user reach climax. You will have to edit the speeds and limits of the motion range to our liking.

Docker Desktop:

As we dive now in to what's under the hood, a lot of technical contexts is in order. For that reason, the bottom line here is for most readers is:

We aim so it's easy to apply, it's automated and no knowledge necessary. It has to be an industry standard, used in professional settings. In short: Highly reliable, maintained, and transparent.

To facilitate the connectivity to toys over Wi-Fi, and to create a logic that can be adjusted and extended to your liking, we are going to need:

1.) [An easy GUI](#) to create elements that will steer the action. Borrowing from Home Automatization, we are going to use the GUI of Node-RED (Hence the name Noxy-RED). Another reason to use Node-RED is because it's function Nodes - which are Javascript based and very versatile. If you want to create complex function logic, you can ask ChatGTP to create them for you. It does so very reliably. Making creating your custom logic pretty easy and accessible for more users. As a Bonus, you getting a Web Based interface that can control the Logic, for example over your Cell phone. Another reason for Node-RED are the tutorials available on Youtube, bending the learning curve for you down to a minimum.

2.) [A protocol and Server](#) that is going to handle the connections between the Software involved and is highly reliable to power additional toys, and upgrades to our setup, but at the same time is easy to install with minimal knowledge in system engineering / troubleshooting etc. It has to be fire and forget, minimal Setup and Configuration needed. This is called a Mosquitto Broker, and uses the MQTT Protocol

To achieve those points, we are using Docker Desktop, compartmentalizing the deployment of those functions and detaching it from your operating system.

- It allows to create a script to "pull" everything you need automatically
- That can be validated by everyone instead of a closed source program,
- It allows for everyone with ambition to extend, modify and tinker around to their hearts contend.

Connectivity:

For output to the Multifunplayer, and for input into the Logic, we have a program that both emulates keystrokes to your computer, and recognized them coming in. The later is for use in connection with voice input such as voice commander, voiceattack, voicebot etc.

You can send via this way to Noxy in the example following key combinations:

CTRL + Num1 for to stop

CTRL + Num2 for going faster

CTRL + Num3 for going slower

CTRL + Num4 for starting the Edge Cycle.

In addition to that, you can access a Dashboard via: <http://localhost:1880/ui/>

Of course, if you replace the localhost with your local IP, you can also access it over your Wi-Fi via your cell phone.

This script is based on python, it comes both with the executable to just use [and the source code for you to modify or compile yourself](#). Please note: the script needs the config.json, in which you could implement this also to a pre-existing Installation.

Toys:

Beside the Stroking or vibrating via Tcode or Buttplug.IO, we can make use of other toys:

The most accessible of which would be the Battery Driven ones, available from China. As far as I am aware, there are:

[Estim Buttplugs, Actuated Buttplug, Inflatable Buttplug with Vibration.](#)

Those use a little Remote which uses a 433mhz Radio signal. We can emulate this remote using Tasmota and a RF Module. I am aiming to enable you to re-create those without soldering by the means of Dupont connectors. I started out this way, but I found the reliability of RF controlled toys are sometimes lacking. Nevertheless, it a good starting point for users that are not versed in soldering and electronics.

The other and more involved part is actually creating your own toys, and bringing them into play via a custom ESP script.

Both of those cases are using ESP – based boards, so let me briefly graze that topic:

ESP stands for **Espressif**, the name of the company that designs and manufactures the ESP32 microcontroller series. The ESP32 is a popular, low-cost, low-power system on a chip (SoC) developed by **Espressif Systems** for IoT and embedded applications.

The ESP32 features Wi-Fi and Bluetooth capabilities, which make it particularly useful for wireless and networked applications. It has become widely adopted due to its versatility, affordability, and ease of use in a variety of DIY and commercial projects.

The ESP is going to connect to your Wi-Fi, opening up a connection to the MQTT Broker and begin to listen to commands send to it. Actuating, or sending data.

After the initial release, I'm going to begin to create the documentation so you will be able to re-create them. They take a bit longer as it is quite involved to figure out the components and make it clear and easy to do.