

**VNM SIMULATION., JSC**

VNM MOTION CONTROLLER USER MANUAL

Contents

[**VNM SIMULATION., JSC** 1](#_Toc126415364)

[***1. Introduction*** 2](#_Toc126415365)

[***2. Firmware specs*** 2](#_Toc126415366)

[***3. Configuration App*** 2](#_Toc126415367)

[***4. Master List*** 4](#_Toc126415368)

[***4.1. Servo set*** 4](#_Toc126415369)

[***4.2. Motion controller*** 4](#_Toc126415370)

[***4.3. Other electronic material*** 5](#_Toc126415371)

[***4.4. VNM Motion controller*** 5](#_Toc126415372)

[***5. Connection*** 6](#_Toc126415373)

[***5.1. Connection Pin of stm32F401R(C/D/E)T*** 6](#_Toc126415374)

[***5.2. Pulse/Dir Connection*** 7](#_Toc126415375)

[***5.3 Torque reach and servo ready connection*** 8](#_Toc126415376)

[***5.4. Button Connection*** 8](#_Toc126415377)

[***6. Servo Driver Parameter setup*** 8](#_Toc126415378)

[***6.1. AASD servo driver*** 8](#_Toc126415379)

[***6.2. Mige servo driver set*** 10](#_Toc126415380)

[***7. Flash firmware*** 10](#_Toc126415381)

[***8. Config motion controller in VNM UI*** 12](#_Toc126415382)

[***9. Motion Software and configuration for VNM Motion Controller firmware*** 12](#_Toc126415383)

[***10. How to Testing*** 14](#_Toc126415384)

# ***1. Introduction***

VNM Simulation., JSC is a sim racing manufacturer based in Vietnam. We strive to design and produce high quality sim racing equipment. Our firmware/software is available to DIY community and unlocks the possibility to make hardware for equipment like ffb wheelbase, pedal set, shifter, steering wheel rim and a motion rig at thecheapest price. We can also supply a completed solution for a car simulator.

VNM Motion Controller firmware is an STM32F401RCT-based firmware package with an accompanying Windows configurator app. The configurator app is used to make setup changes to motion controller (i.e actuator specs, pwm control frequency and so on). Almost any servo motor can be used with our firmware.

Connect to the [Discord channel](https://discord.gg/gtXNWHnz)

Connect to the [Facebook group](https://www.facebook.com/groups/2303223886447618/)

# ***2. Firmware specs***

* STM32F401RCT microcontroller with 25 MHz crystal
* Supports up to 8 actuators
* Supports up to 250 kHz PWM frequency with <= 6 actuators and 200 kHz with <= 8 actuators
* Input for soft Emergency stop, Servo ready, Torque reach (stroke limit detection)
* Can be used with some motion software like FlyPt Mover, Simtools or similar which support configurable string output.

# ***3. Configuration App***

VNM UI is central configuration software for all VNM Products like VNM Wheelbase, Direct Drive, Shifter, Handbrake, Steering wheel rim, Pedal set, Motion controller…

In Motion tab:

* Select Motion Device: displays the connected VNM Motion Controller Firmware version
* Number of actuators: configure number of actuators of you motion system.
* PWM frequency: this is the output PWM frequency of the controller that can be used as an input for the Servo driver. With higher PWM frequency the speed of your actuators will be higher. PWM frequency can be set to 25, 50, 75, 100, 125, 150, 200 or 250 kHz.
* Stroke step (mm/round): this is a linear movement of the actuator in mm per 1 motor shaft rotation (can be measured with a ruler). Frequency selection depends on your taste :D
* Stroke length (mm): The full stroke length of your actuator.
* Pulse Per Revolution: it is the PPR of your encoder. Usually, one round need Count Per Revolution (CPR = 4\*PPR). This value is very important because firmware will use it to create exact number of pulses to control actuator.
* Deadzone (percentage): it is the actuator offset (default 10%). Actuator will move within offset% to (100-offet)% of your stroke length. It is a spare distance for error deviation.
* Status: it will show realtime status of your motion controller:

+ Idling: Motion is working but actuator is not moving.

+ Running: Motion controller is working, and actuators are moving.

+ Parking: Actuator are moving to middle position.

+ Running to Min Position: Actuator is moving to its lowest position until its torque reach is triggered.

+ Calibrating: Motion Controller is in Calibration mode to detect min max position.

+ Running to Offset Position: actuator is moving to offset position.

* Calibrate button: triggers calibration process.
* Status Button: requests configuration of motion controller.
* Run (Stop): soft emergency button
* Park: requests parking process.
* Max Pulse: displays number of pulses required by motion controller for moving an actuator from offset% to (100-offset)% stroke length
* Actuator status:

+ Unused: don’t use this actuator.

+ Not ready: Actuator is armed but not yet ready to run.

+ Ready: Actuator is armed and ready to run.

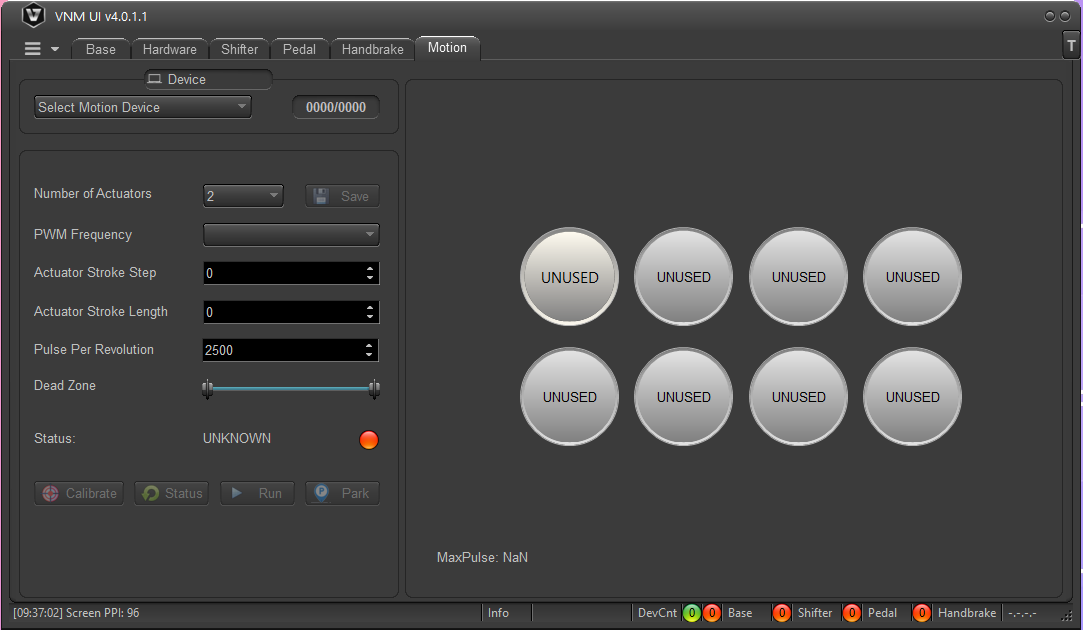


Figure 1 VNM application interface

# ***4. Master List***

## ***4.1. Servo set***

Can be any of servo set that supports pulse/dir. It is better if it has torque reach, servo ready output from its driver. Most every servo has servo ready signal. If it doesn’t have torque reach output, you have to add an endstop limit switch at both sides of your actuator stroke. If it doesn’t have, you have to connect it to trigger (describe later) to detect servo ready.

## ***4.2. Motion controller***

It is stm32f401RxT (RCT, RDT, RET) kit with 25mhz crystal.

Example :

* STM32F401RET Nucleo with 25mHz crystal

<https://www.digikey.ca/en/products/detail/stmicroelectronics/nucleo-f401re/4695525>

<https://www.digikey.com/en/products/detail/ecs-inc/ECS-250-18-4X-F/827533>

## ***4.3. Other electronic material***

- Optocoupler HCPL2631 or 6N137

<https://www.digikey.com/en/products/detail/rochester-electronics-llc/HCPL2631/11558973>

https://www.digikey.com/en/products/detail/liteon/6N137/1969175

- Optocoupler PC817

<https://www.digikey.com/en/products/detail/american-bright-optoelectronics-corporation/BPC-817-B-BIN/9678142>

Resistor 1.5kohm->2.2kohm, 200ohm->330 ohm, capacitor 100nF.

- Isolated power supply 5V (like your mobile phone charger)

you can buy material from other manufacturer or shop.

To be continued…

## ***4.4. VNM Motion controller***

If you don’t have time for DIY controller or just want to support VNM Simulation, you can buy VNM Motion Controller. Motion simulator circuit by VNM Simulation is high quality and includes the following features:

* Supports up to 8 actuators.
* PWM Frequency up to 250kHz with 6 actuator and 200kHz with 8 actuators.
* High resolution PWM make the actuator run smoother (generate 250000 pulse per second so you dont need to set gear ratio higher than 1).
* MCU is isolated completely from servo drivers (prevents EMI noise issues from servo driver).
* Baud rate is unlimited.

Ảnh có chứa văn bản, thiết bị điện tử, mạch

Mô tả được tạo tự động

Figure 2 Motion Printed Circuit Board

# ***5. Connection***

## ***5.1. Connection Pin of stm32F401R(C/D/E)T***

* Pulse: PWM Pin
* Dir: Direction Pin
* TReach: Torque Reach Pin
* SReady: Servo Ready Pin
* Index: corresponding to servo 1,2,…8
* On/off: Button to turn on/off motion controller
* Park: Button to trigger Parking Process
* Calibrate: Button to trigger Calibration Process
* USART1/USART6: Future use

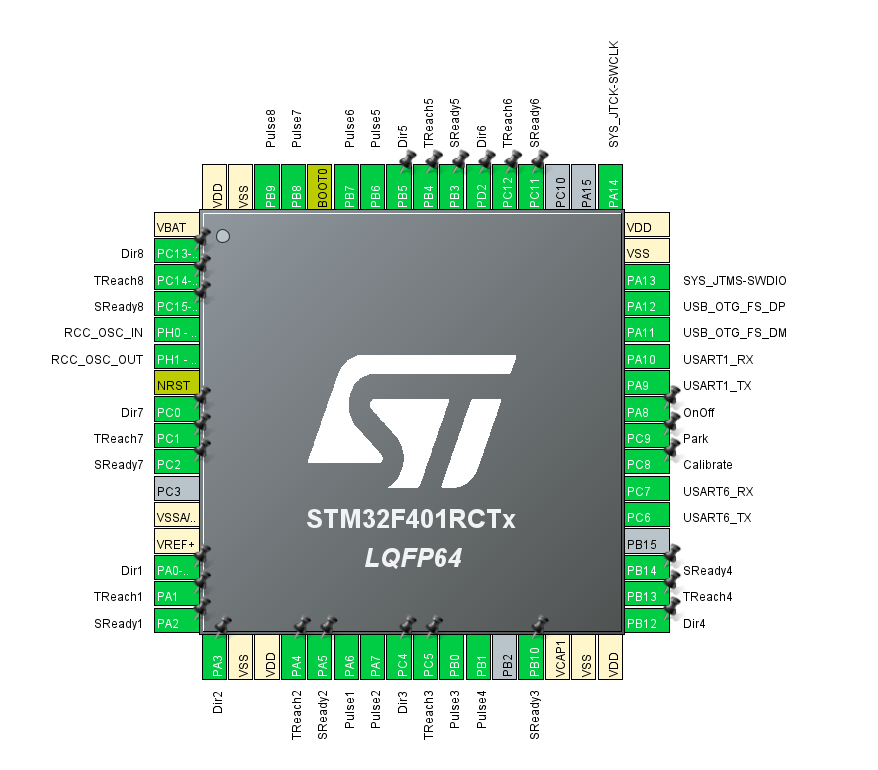


Figure 3 Pinout chip STM32

## ***5.2. Pulse/Dir Connection***

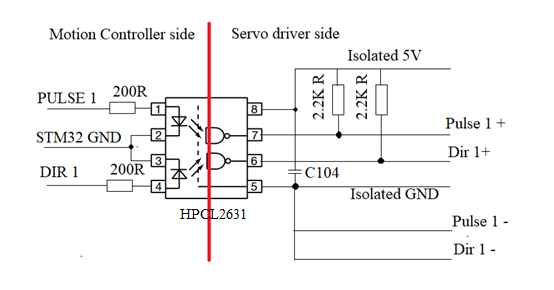


Figure 4 Isolation signal pulse and direction with HPCL2631

Repeat for other servos.

## ***5.3 Torque reach and servo ready connection***

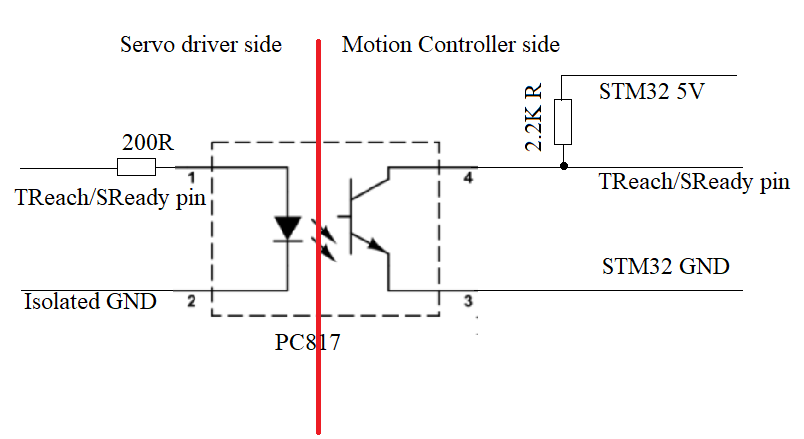


Figure 5 Isolation signal torque reach and servo ready with PC817

Repeat for other servo.

## ***5.4. Button Connection***

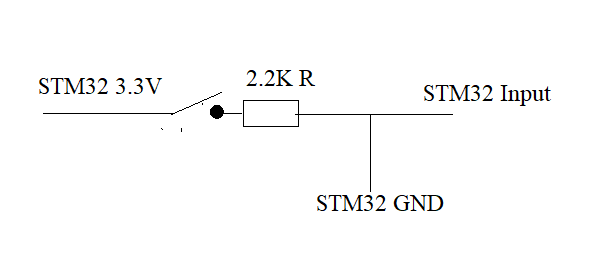


Figure 6 Button diagram

# ***6. Servo Driver Parameter setup***

## ***6.1. AASD servo driver***

|  |  |  |  |
| --- | --- | --- | --- |
| **Parameter / Function** | **Description from User Manual** | **Initial Value** | **What this parameter/function does and when to change it** |
| Pn001 | Motor code | Don’t change | This is the specific code for the motor that came paired with your driver. If you need to set this value (you should not have to), refer to page 46 of the user manual linked above to find the motor code of your motor. |
| Pn002 | Control mode | 0 | “2” is position mode (see page 47 of the user manual linked above). You should not need to change this parameter. |
| Pn003 | Servo enabled | 0 | This setting means that the servo motor requires external input to be operated (input from the controller). If you don’t have servo on button, set it to 1. If you have servo on button set it to 0 |
| Pn008 | Internal around are torque limit (CCW) | 300 | Use as default. Don’t need to change |
| Pn009 | Around inside the torque limit (the CW) | -300 | Use as default. Don’t need to change |
| Pn024 | Reach the predetermined torque | 100 | Don’t need to change. Torque reach is triggered when the value of torque is greater than or equal to Pn024 set |
| Pn051 | The motor running top speed limit | 3000 | This is the rated top speed of the motor based on the motor specs (see page 46 of the user manual linked above). Depends on your preference how to adjust this value :D. |
| Pn052 | SigIn1 port functional allocation | 1 | Don’t need to change. Let it is servo on.  SigIn port function explanation 4.4.1 |
| Pn053 | SigIn1 port functional allocation | 2 | Change it to 7. Let it is Emergency stop.  SigIn port function explanation 4.4.1 |
| Pn060 | SigOut 1 port functional allocation | 2 | Don’t need to change. Let it is servo ready  SigOut port function explanation 4.4.2 |
| Pn061 | SigOut 2 port functional allocation | 1 | Change to 6, let it is torque reach |
| Pn098 | Electronic gear ratio | 1 | Don’t need to change |
| Pn109 | Position command deceleration mode | 1 | Don’t need to change, set it is smoothing filter |
| Pn113 | The position loop feedforward gain | 0 | position feedforward directly on the speed instruction, can reduce the position tracking error, improve the response. If the feedforward gain is too big, can lead to speed overshoot => Not yet tested |
| Pn114 | Position loop feedforward filter time constant | 5 | position feedforward directly on the speed instruction, can reduce the position tracking error, improve the response. If the feedforward gain is too big, can lead to speed overshoot=> Not yet tested |
| Pn115 | The position controller gain 1 | 100 | In mechanical systems do not produce under the premise of vibration or noise, increase the position loop gain value, to speed up the reaction rate, shorten the positioning time=> Not yet tested |

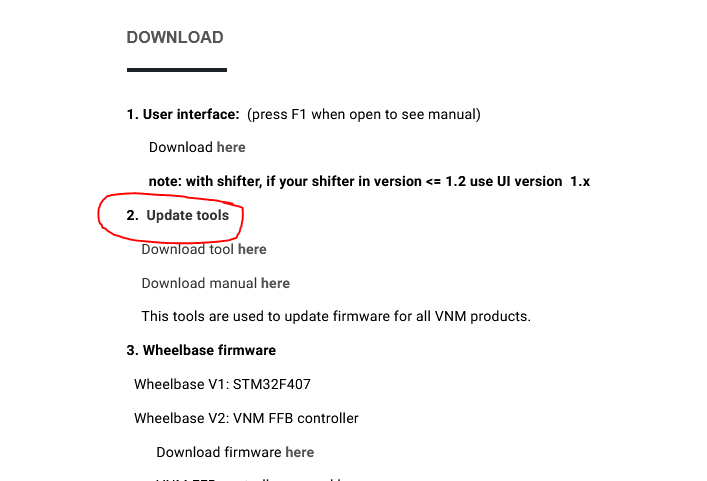
Will research more to add.

## ***6.2. Mige servo driver set***

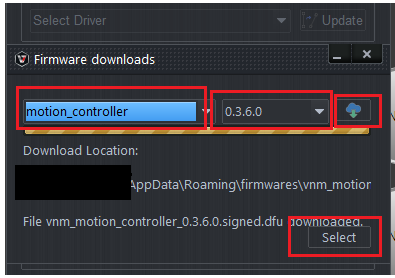
Will be added later

# ***7. Flash firmware***

Download VNM Flash and its manual from <https://vnmsimulation.com/download>



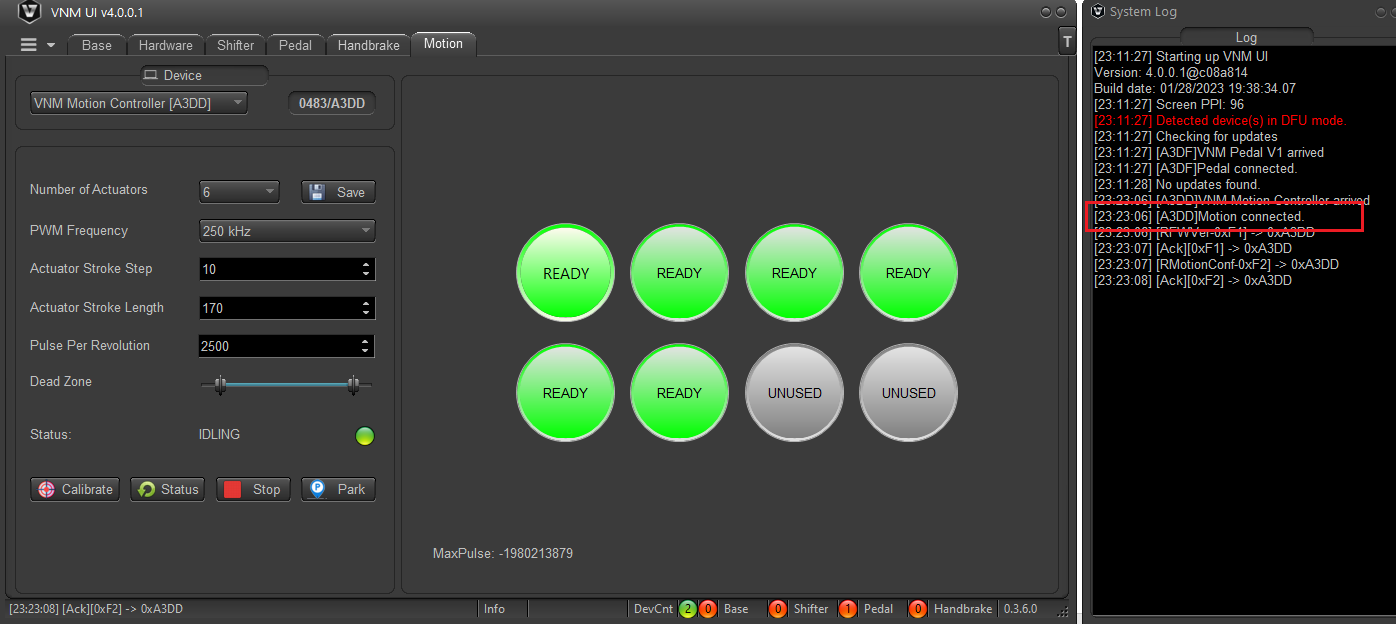
Choose lastest version of motion controller and download then click select.



Flashing the firmware follows the VNM Flash manual. Get the controller out of the bootloader mode and reconnect the usb cable.

# ***8. Config motion controller in VNM UI***

In UI version >= 4.0.0.1, you will see motion controller connects to PC.



Config your motion system information follow session 3 and click “save” button.

# ***9. Motion Software and configuration for VNM Motion Controller firmware***

Currently we just tested with FlyPT Mover but it should work with other software that support binary and configurable string output like simtools,…

The data output string is 20 bytes.

0xFF 0xFF b1 b2 b3 b4 b5 b6 b7 b8 b9 b10 b11 b12 b13 b14 b15 b16 LF CR

0xFF 0xFF - start of data identifier for the receiving micro controller

<byte1>< byte2> - 8 bit binary number giving actuator 1

<byte3>< byte4> - 8 bit binary number giving actuator 2

<byte5>< byte6> - 8 bit binary number giving actuator 3

<byte7>< byte8> - 8 bit binary number giving actuator 4

<byte9>< byte10> - 8 bit binary number giving actuator 5

<byte11>< byte12> - 8 bit binary number giving actuator 6

<byte13>< byte14> - 8 bit binary number giving actuator 7

<byte15>< byte16> - 8 bit binary number giving actuator 8

LF - Line Feed character – 0x0A

CR - Carriage Return character – 0x0D

Here are example of configuration:

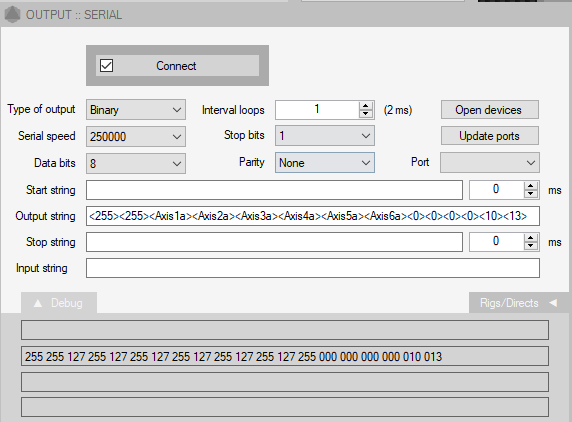


Figure 7 Set output string

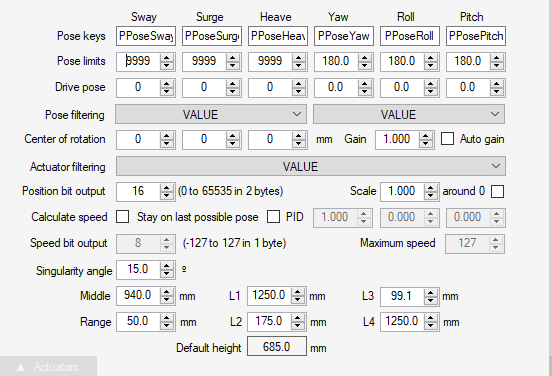


Figure 8 Set position bit output and system parameters

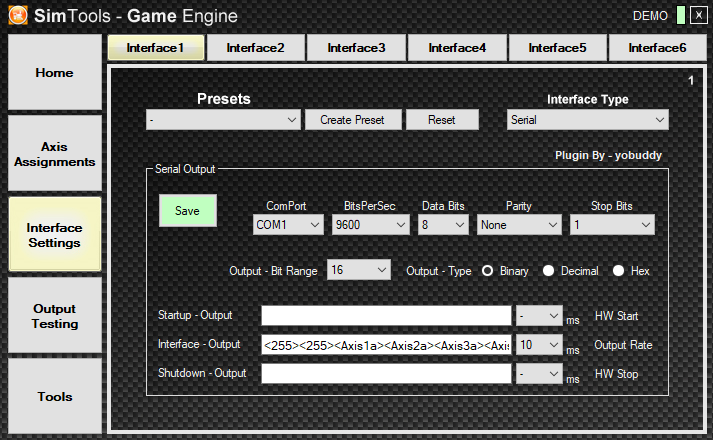


Figure 9 With SimTools

# ***10. How to Testing***

Will be added later

* Testing:

1/ Testing with hercules:

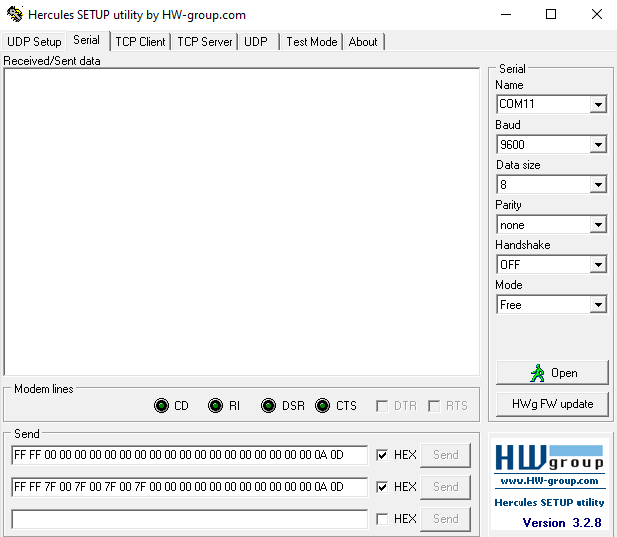
* Connect cable USB type-C from PC to motion circuit.
* Open Hercules/ Serial/ Name (Choose port)/ Open

Figure 10 Hercules interface

* Send data string like what you set.

2/ Testing with flyPT & Sim Tools

* Connect cable USB type-C from PC to motion circuit.
* Open flyPT, Sim Tools and open or create new file.
* Connect port.

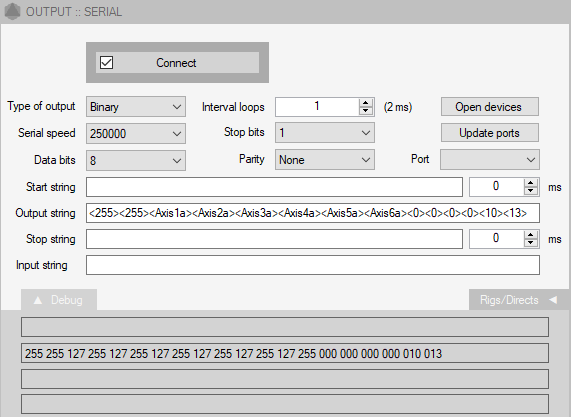


Figure 11 Connect port

* Test poses, rigs (or connect with game and play).
* If you want a good experince, you need set up In gain/Out gain, Range and Filter accordant

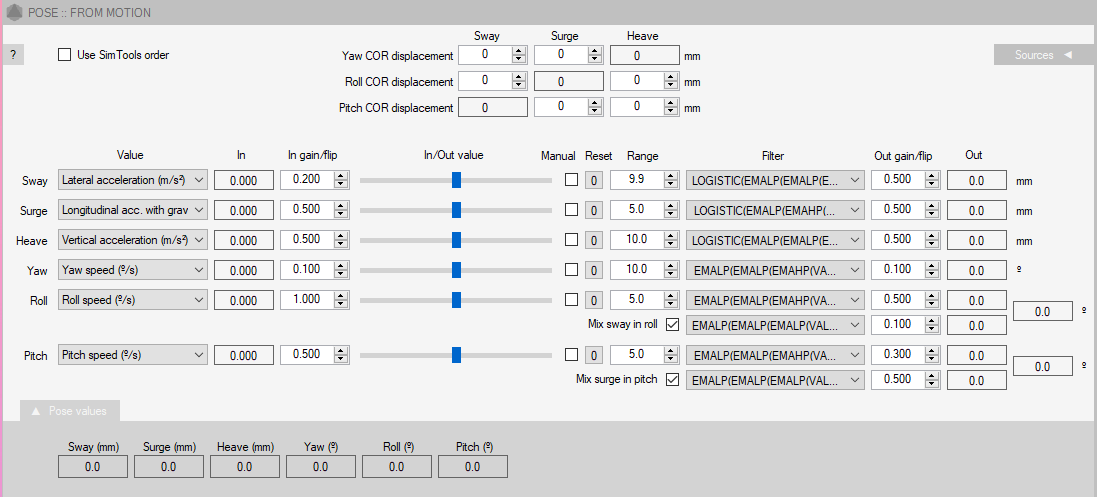


Figure 12 Test poses

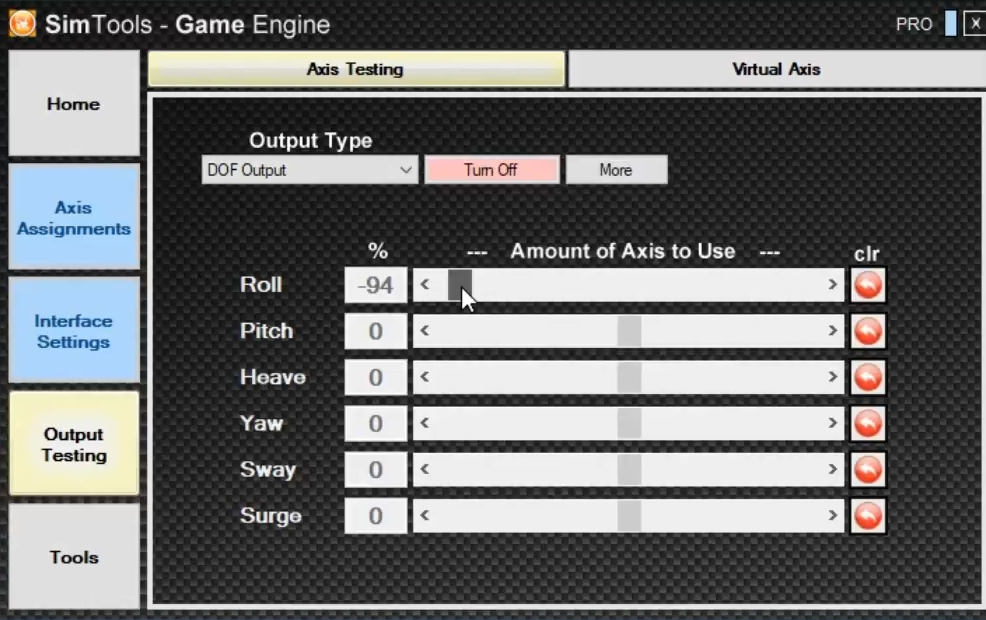


Figure 14 With SimTools