

VNM SIMULATION., JSC

VNM MOTION FIRMWARE USER MANUAL

Contents

[***1. Introduction*** 2](#_Toc153997600)

[***2. Firmware specs*** 2](#_Toc153997601)

[***3. Configuration App*** 2](#_Toc153997602)

[***4. Master List*** 4](#_Toc153997603)

[***4.1. Servo set*** 4](#_Toc153997604)

[***4.2. Motion controller*** 4](#_Toc153997605)

[***4.3. Other electronic materials*** 4](#_Toc153997606)

[***4.4. VNM Motion controller*** 5](#_Toc153997607)

[***5. Connection*** 5](#_Toc153997608)

[***5.1. Connection Pin of stm32F401R(C/D/E)T*** 5](#_Toc153997609)

[***5.2. STM32-Servo Driver Connection*** 7](#_Toc153997610)

[***5.3. Button Connection*** 8](#_Toc153997611)

[***5.4. VNM Motion Controller*** 8](#_Toc153997612)

[***5.5. Mapping Axes to actuators of each model.*** 9](#_Toc153997613)

[***6. Servo Driver Parameter setup*** 10](#_Toc153997614)

[***6.1. AASD servo driver*** 10](#_Toc153997615)

[***6.2. Other driver sets*** 12](#_Toc153997616)

[***7. Flash firmware*** 12](#_Toc153997617)

[***8. Motion Software and configuration for VNM Motion Controller firmware*** 13](#_Toc153997618)

[***9. How to Testing*** 16](#_Toc153997619)

[9.1. Testing with hercules: 16](#_Toc153997620)

[2. Testing with flyPT & Sim Tools 16](#_Toc153997621)

# ***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 the cheapest price. We can also supply a complete 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, pulse frequency and so on). Almost any servo motor can be used with our firmware.

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

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

# ***2. Firmware specs***

* STM32F401RCT microcontroller with 25 MHz crystal
* Supports up to 9 actuators.
* Supports up to 500 kHz pulse frequency.
* Generates exactly the number of pulses that a servo driver needs to rotate one round.
* Filtering for smooth movement.
* Support belt tensioners without feature on motion software.
* Loop time is 1ms
* Can be used with some motion software like SRS, FlyPt Mover, Simtools, Sim-motion software or similar which support configurable string output.

# ***3. Configuration App***

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

A screenshot of a computer

Description automatically generated

* Select Motion Device: displays the connected VNM Motion Controller.
* Number of actuators: configure the number of actuators of your motion system.
* PWM (Pulse) frequency: this is the output pulse frequency of the controller that can be used as input to the Servo driver. higher pulse frequency, the higher speed of your actuator.
* Step stroke (mm/rev): this is the linear motion of the actuator in mm per revolution of the motor shaft (can be measured with a ruler).
* Stroke length (mm): The distance between the min position and the max position of an actuator (configure for each actuator)
* Park at Pos: choose a position that actuator will park. It is LOW or MID of stroke (configure for each actuator).
* Pulse Per Revolution: it's your encoder's PPR. Normally, one cylce needs Count per spin (CPR = 4\*PPR). This value is important because the firmware will use it to generate the correct number of pulses to drive the actuators.
* Max pulse per loop: How many pulses that the controller will generate each time.
* Deadzone (percentage): It is the actuator offset (default 3%). An actuator will move from offset % to (100-offset) % of your stroke length. It is a spare distance for an error deviation.
* Smothing: Higher value, higher smooth.
* Status: it will show the real-time status of your motion controller:

+ Idling: Motion is working but all actuators are not moving.

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

+ Parking: All actuators are moving to 0 position.

+ Running to Min Position: All actuators are moving to its min position until its torque reach is triggered.

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

* Status Button: requests configuration of the motion controller.
* Run (Stop): soft emergency button.
* Actuator status:

+ Unused: This actuator isn’t used.

+ Not ready: This actuator is used but not yet ready to run.

+ Ready: This actuator is used and ready to run.

* The numbers at each actuator status correspond to actuator stroke ste, actuator stroke length, max pulse (number of pulses required by a servo drive for moving an actuator from offset % to (100-offset) % stroke length).

# ***4. Master List***

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

Any of servo set that supports pulse/dir can use with VNM Motion Controller firmware. It is better if it has torque reach, servo ready output from its driver. If it doesn’t have torque reach output, you must add end stop limit switches at both sides of your actuator. If it doesn’t have servo ready, you must connect fake servo ready to servo ready pin.

## ***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 materials***

- Optocoupler HCPL2631 or 6N137 for pulse/dir connections.

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

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

- Optocoupler PC817 for digital input/output (DI/DO) of servo driver

<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 for pulse/dir

- Isolated power supply 12-24V for DI/DO.

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 that is high quality and includes the following features:

* The anodized Aluminum box reduces noise from other devices.
* Isolates the micro controller from all servo drivers to reduce EMI with very high speed optocoupler and digital isolator.
* Supports up to 9 servos with pulse frequency up to 500khz (configurable).
* The accuracy of pulse generation is almost absolute (see the picture with 100 millions pulse, no pulse is lost).
* Generates exactly the number of pulses that the driver needs so the pulse resolution is very high and smooth movement.
* Automatically correct the position of the actuator in case the servo driver loses pulse causing the actuator reaches the limit.
* Supports up to 2 belt tensioners without belt tensioner feature on a motion software.
* Configure parameters/Monitoring system via VNM Config UI.
* The control loop is 1 ms that is real time with a motion application.
* Signal filtering make the actuator run smoothly in VNM Config UI.
* Unlimited baud rate that mean can run at any speed that a motion software supports.

A screenshot of a computer

Description automatically generated

# ***5. Connection***

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

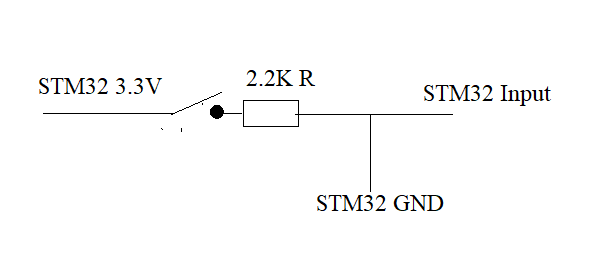
* Pulse: PWM Pin
* Dir: Direction Pin
* TReach: Torque Reach Pin
* SReady: Servo Ready Pin
* Index: corresponding to servo 1,2…8,9
* Calibrate: Button to trigger Calibration Process
* USART6: Future use

|  |  |  |
| --- | --- | --- |
| STM32 Pin | Description | Note |
| PB6 | Pulse 1 | Servo 1 |
| PB5 | Torque reach 1 |
| PB4 | Dir 1 |
| PB3 | Servo ready 1 |
| PB7 | Pulse 2 | Servo 2 |
| PD2 | Torque reach 2 |
| PC12 | Dir 2 |
| PC11 | Servo ready 2 |
| PB8 | Pulse 3 | Servo 3 |
| PC13 | Torque reach 3 |
| PC0 | Dir 3 |
| PC15 | Servo ready 3 |
| PB9 | Pulse 4 | Servo 4 |
| PC3 | Torque reach 4 |
| PC1 | Dir 4 |
| PC2 | Servo ready 4 |
| PA0 | Pulse 5 | Servo 5 |
| PA7 | Torque reach 5 |
| PC4 | Dir 5 |
| PC5 | Servo ready 5 |
| PA1 | Pulse 6 | Servo 6 |
| PB10 | Torque reach 6 |
| PB0 | Dir 6 |
| PB1 | Servo ready 6 |
| PA3 | Pulse 7 | Servo 7 |
| PA4 | Torque reach 7 |
| PA5 | Dir 7 |
| PA6 | Servo ready 7 |
| PA8 | Pulse 8 | Servo 8 |
| PC8 | Torque reach 8 |
| PC9 | Dir 8 |
| PB15 | Servo ready 8 |
| PA9 | Pulse 9 | Servo 9 |
| PB14 | Torque reach 9 |
| PA10 | Dir 9 |
| PB15 | Servo ready 9 |
| PB12 | Calibrate Button |  |
| PA2 | Internal use | Must connect together |
| PA15 |
| PC6 | UART TX | Future use |
| PC7 | UART RX |

## ***5.2. STM32-Servo Driver Connection***

Example of AASD Driver

## ***5.3. Button Connection***



## ***5.4. VNM Motion Controller***

A black rectangular device with many ports

Description automatically generated

|  |  |
| --- | --- |
| **Item** | **Description** |
| CN1->CN7 | Use for actuator only |
| CN8, CN9 | Can be used for actuator or belt tensioner (if active belt tensioner, CN8/CN9 are used for tensioner only. |
| CN10 | Future use |
| DFU button | Click to reset controller, hold few second to change DFU mode. Disconnect USB and reconnect to change Run mode |
| Calibrate Button | Calibrate motion system |
| EMC-Stop GX12-4 | For emergency and servo enable button |
| 24VDC | For 24v DC input. |
| USB type C port | Connect to PC |

|  |  |  |
| --- | --- | --- |
| Pin | Description | Note |
| 14 | Pulse - | Available for CN1->CN7 |
| 3 | Pulse + | Available for CN1->CN7 |
| 4 | Dir + | Available for CN1->CN7 |
| 5 | Dir - | Available for CN1->CN7 |
| 6 | Servo on | Available for CN1-> CN9 |
| 7 | Emergency stop | Available for CN1-> CN9 |
| 9 | 24V output | Available for CN1-> CN9 |
| 11 | Servo Ready | Available for CN1->CN9 |
| 23 | Torque Reach | Available for CN1->CN9 |
| 13 | Torque GND | Only available for CN8/CN9 |
| 21 | Torque Dir | Only available for CN8/CN9 |
| 25 | Torque VREF | Only available for CN8/CN9 |

## ***5.5. Mapping Port to actuators of each model.***

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Model | Port 1 | Port 2 | Port 3 | Port 4 | Port 5 | Port 6 | Port 7 | Port 8 | Port 9 |
| 3 actuators (2 Fronts + 1 Rear) | FL | FR | R |  |  |  |  |  |  |
| 4 actuators (1 Fronts + 2 Rear) | F | RR | RL |  |  |  |  |  |  |
| 4 actuators (2 Fronts + 2 Rear) | FL | FR | RR | RL |  |  |  |  |  |
| 5 actuators (2 Fronts + 2 Rear + TL) | FL | FR | RR | RL | TL |  |  |  |  |
| 5 actuators (2 Fronts + 2 Rear + Surge) | FL | FR | RR | RL | SG |  |  |  |  |
| 6 actuators (2 Fronts + 2 Rear + TL + Surge) | FL | FR | RR | RL | TL | SG |  |  |  |
| 6 actuators (2 Fronts + 2 Rear + 2TL) | FL | FR | RR | RL | TLF | TLR |  |  |  |
| 7 actuators (2 Fronts + 2 Rear + TL+ Surge + Belt) | FL | FR | RR | RL | TL | SG | Belt |  |  |
| 7 actuators (2 Fronts + 2 Rear + 2TL + Surge) | FL | FR | RR | RL | TLF | TLR | SG |  |  |
| 8 actuators (2 Fronts + 2 Rear + 2TL + Surge + Belt) | FL | FR | RR | RL | TLF | TLR | SG | Belt |  |

Whereas

|  |  |  |
| --- | --- | --- |
| F: Front | FL: Font Left | FR: Font Rear |
| R: Rear | RL: Rear Left | RR: Rear Righ |
| TL: Traction Lost | TLF: Traction Lost Front | TLR: Traction Lost Rear |
| SG: Surge |  |  |
| Belt: Belt tensioner |  |  |

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

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

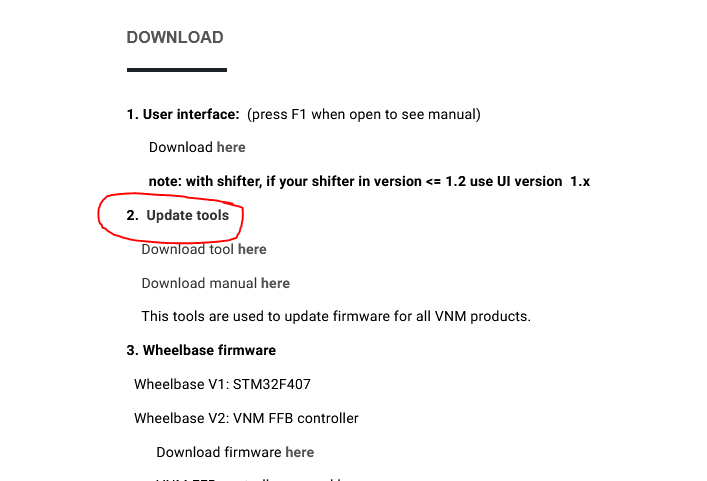
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Parameter / Function** | **Description from User Manual** | **Initial Value** | **VNM Actuator** | **What this parameter/function does and when to change it** |
| Pn001 | Motor code | 3 | 4 | This is the specific code for the motor that came paired 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 | “2” is position mode (see page 47 of the user manual linked above). |
| Pn003 | Servo enable mode | 0 | 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 | 300 | Use as default. Don’t need to change |
| Pn009 | Around inside the torque limit (the CW) | -300 | -300 | Use as default. Don’t need to change |
| Pn024 | Reach the predetermined torque | 100 | 100 | 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 | 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 | 1 | Servo on input. SigIn port function explanation 4.4.1 |
| Pn053 | SigIn2 port functional allocation | 2 | 7 | Emergency stop input. |
| Pn054 | SigIn3 port functional allocation | 19 | 18 | Change direction with speed/torque control. |
| Pn060 | SigOut 1 port functional allocation | 2 | 2 | Servo ready output |
| Pn061 | SigOut 2 port functional allocation | 1 | 6 | Torque reach value at pn024 |
| Pn076 | Emergency stop reset (EMG) | 0 | 1 | Enable EMC |
| Pn096 | Command pulse input mode | 0 | 0 | Set 1 to Forward/reverse pulse |
| Pn097 | Input command direction | 0 | 0 | set pn024 = -150 if you set this param |
| Pn098 | Electronic gear ratio | 1 | 1 | Multiple input pulse with gear ratio |
| Pn109 | Position command Acceleration/deceleration mode | 1 | 1 | smoothing filter |
| Pn110 | Filter constant time | 50 | 30 | in ms |
| Pn113 | The position loop feedforward gain | 0 | 20 | 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 |
| Pn114 | Position loop feedforward filter time constant | 5 | 10 |
| Pn115 | The position controller gain 1 | 100 | 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 |
| Pn083 | Low voltage detection | 200 | 100 | working voltage from >=100V-240V |

## ***6.2. Other driver sets***

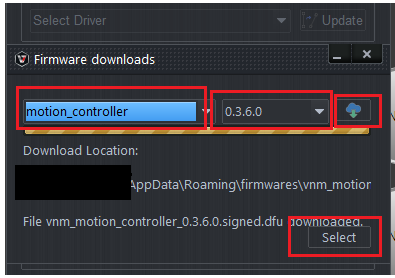
Update later

# ***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 after flashing.

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

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

The data output string is 22 bytes.

0xFF 0xFF b1 b2 b3 b4 b5 b6 b7 b8 b9 b10 b11 b12 b13 b14 b15 b16 b17 b18 LF CR

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

<byte1>< byte2> - 16 bit binary number giving actuator 1

<byte3>< byte4> - 16 bit binary number giving actuator 2

<byte5>< byte6> - 16 bit binary number giving actuator 3

<byte7>< byte8> - 16 bit binary number giving actuator 4

<byte9>< byte10> - 16 bit binary number giving actuator 5

<byte11>< byte12> - 16 bit binary number giving actuator 6

<byte13>< byte14> - 16 bit binary number giving actuator 7

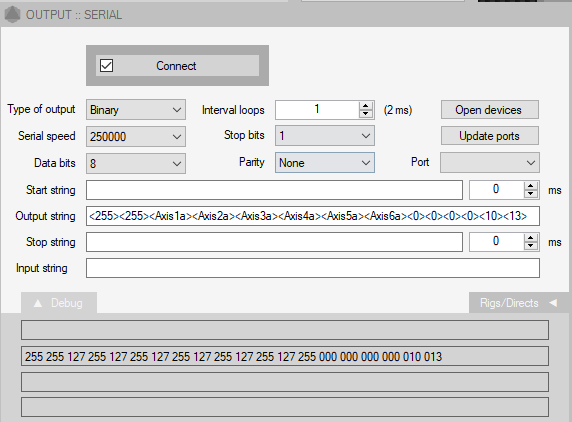
<byte15>< byte16> - 16 bit binary number giving actuator 8

<byte17>< byte18> - 16 bit binary number giving actuator 9

LF - Line Feed character – 0x0A

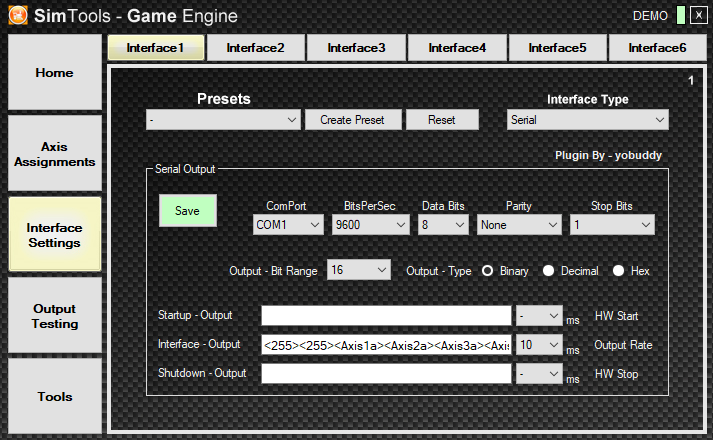
CR - Carriage Return character – 0x0D

Here are example of configuration:



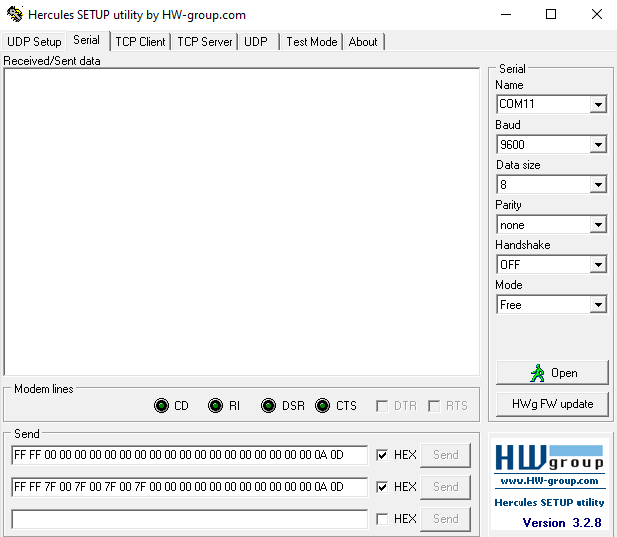
A screenshot of a calculator

Description automatically generated



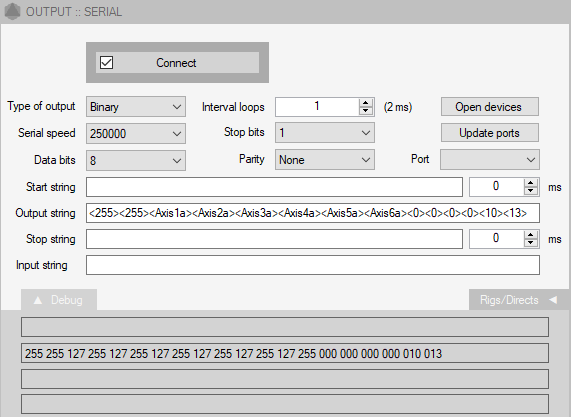
# ***9. How to Testing***

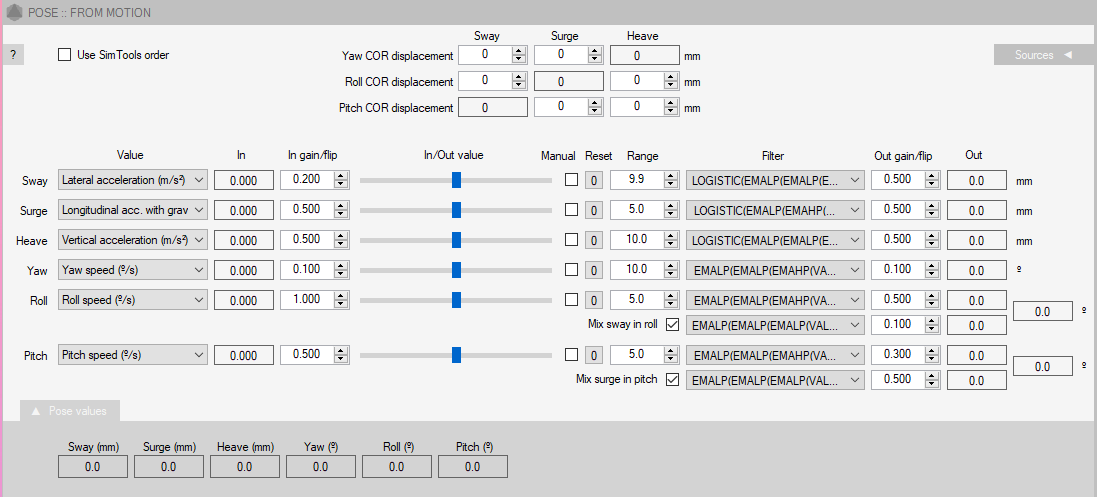
## 9.1. Testing with hercules:

* Connect cable USB to PC.
* Open Hercules/ Serial/ Name (Choose port) and Open
* Send data string like what you set.

## 2. Testing with flyPT & Sim Tools

* Connect cable USB to PC.
* Open flyPT, Sim Tools and open or create new file.
* Connect port.



* Test poses, rigs (or connect with game and play).
* You can set up In gain/Out gain, Range and Filter accordingly as your liking.

