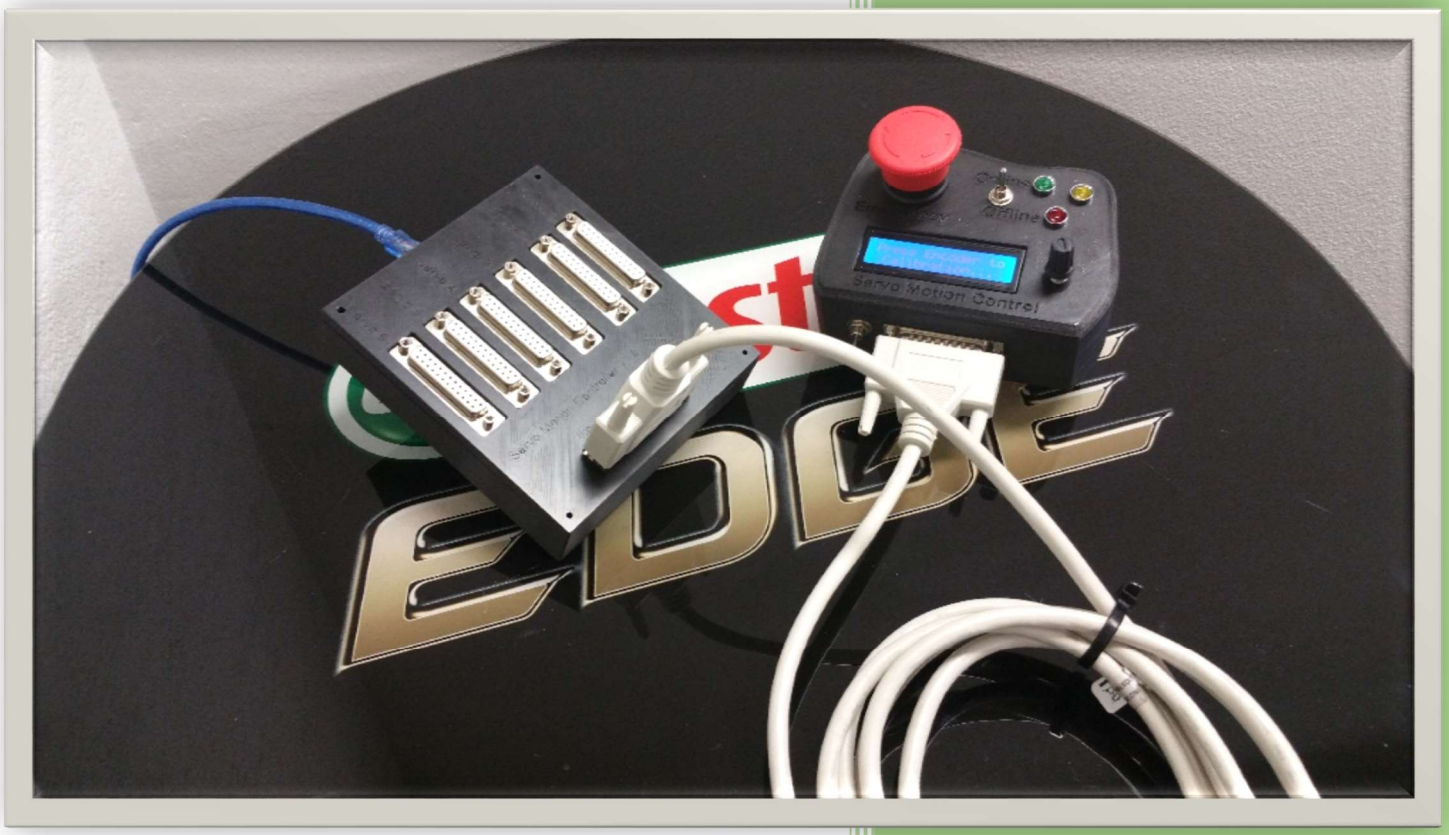
Translated from German to English - www.onlinedoctranslator.com

operation manual

2023

Advanced servo motor control device for r up to 7 engines



Motion4Sim

For AASD Series servo motors

12/12/2023

Table of contents

1 Introduction..... 3
2. Functions..... 4
2.1. Hardware functions 4
2.2. Software functions..... 5
2.3. Motion Cueing Software 6
3. Board and connections 7
PCB Version 1.04 9
PCB Layout V1.04 - Pin assignment 11
PCB Version 1.05 12
PCB Version 1.08 , 1.09 , 1.10..... 14
4. Hardware cabling..... 15
4.1. Servo Motor Amplifier Wiring AASD..... 15
4.2. Wiring Servo 7 connection..... 17
4.3. Cabling online switch and emergency stop switch..... 18
4.4. Cabling of the position limit switches/limit switches..... 19
4.5. Connecting a servo brake..... 20
5. Settings of the servo amplifiers ..... .21
6. Hand-held device..... 22
7. Setting the controller / menu control 23
7.1. Filter Menu 27
7.1.1. Quick guide to setting the Spike Filter..... 28
7.2. Service menu..... 28
7.2.1. Repair – Moving without “Homeing” 29
7.3. Actuators Menu..... 30
7.4. SET UP..... 34
7.5. RIG 39
8. Geometry of kinematics W.I.P..... 41
9. Linear and rotary actuators 45
9.1. Rotating actuators..... 45
9.2. Linear actuators..... 46
9.3. Belt tensioner actuator 47
10. USB interface for motion data..... 48
11. Data transmission 49
11.1. Direct Mode..... 50
11.2. Position Mode 51

11.3. SRS mode –Sim Racing Studio 52
12. LED display 53
13. Factory Reset 54
14. Firmware Update..... 55
14.1. Bootloader Mode 56
14.2. Flashing process..... 58
15. Dashboard App..... 59
15.1. Actuator test functions..... ..... .61
15.2 Motion Cueing System 62
16. Special Serial Commands..... 66

1 Introduction

The M4S-AASD15A servo controller enables the fast and seamless transfer of motion information from the PC simulation to the servo motor. The data exchange takes place between the interface programs such as Simtools, FLyPT Mover, BFF, etc. and the simulation. From Dashboard v3.2 it is also possible to use the dashboard to filter data from Sims and games and send it to the controller. More on this in the Dashboard chapter.

The interface software calculates the motion vectors and passes the values on to the M4S servo controller via the USB interface. This controls the servo motors and also offers the option of calculating and smoothing motion cues yourself. The servo motor output stages are connected with DB25 cables.

All necessary parameters for operating the many actuators and actuator types are freely adjustable and configurable. All common servo motors that support STEP/DIR can be used.

2. Features

The controller was mainly designed for the AASD servo motor output stages. However, other servo motors can also be operated with it. This can be used to operate all servo motors whose amplifiers are operated via the Step/Dir function. Stepper motors that are equipped with Step/Dir also work.

The following properties and functions are provided by the controller:

2.1. Hardware features

1. Control of 6 + 1 servo motors (optional)
2. Pulse speed up to > 550kHz
3. Step/Dir Mode
4. Query and evaluation of torque (for calibration) and error status
5. Display of parameters via LC display 16x2, full menu
6. Operation via rotary encoder with push button
7. Switch for signal reception
8. Emergency stop switch for the servo functions
9. Serial USB connection for data transfer
10. 32 bit processor
11. Saving settings
12. Control of LEDs for status display (external handheld)
13. Housing (diy 3d print)
14. Control part can be installed 3m away (optional)
15. LC display and encoder can also be installed on the mainboard
16. Automatic Home calibration on power up or re-connection without limit switches
17. Limit switch can be optionally installed for home calibration
18. Protection from exceeding physical limits of the actuator
19. Optional Platform Health check, to ensure all actuators are active during gaming
20. Latency as low as 1ms for internal filtering
21. E-stop, Force Offline buttons and switches
22. USB power supply

2.2. SOftware functions

1. Software is menu driven and easy to use
2. All functions can be parameterized to suit your own requirements
3. Monitoring the status of the engines
4. Automatically calibrate the zero position of the servo motors
5. Calibration also possible via limit switches
6. Service function for repairing or checking the engines
7. Manual control of each motor individually, for testing and maintenance
8. Adjustable direction for Inline or Foldback placement of servo motor
9. Adjustable screw lead Pitch advance freely adjustable per revolution
10. Adjustable stroke unlimited
11. selection of belt reduction ratios or gear ratios freely adjustable
12. Rotary and linear actuators are supported
13. Scaling of the input signals (master gain)
14. Real Exponential moving average filter for anti-vibration and smooth pulses on the actuator
15. advanced fully adjustable Spike Filter to automatically eliminate jolts during crashes or unwanted motion cues
16. Offset for each motor (especially for rotating actuators)
17. Variable parking position
18. Actuators individually adjustable (electronic translation, length of act., DOF system)
19. Speeds for calibration, slow speed and fast speed freely adjustable (max. 550 kHz)
20. Inverse kinematics with >1000 calculations per second (for linear and nonlinear actuators)
21. Geometry for steward platforms with 6 DOF and 6 actuators can be individually adjusted
22. Geometry for 3 dof 4 dof and custom rigs is calculated
23. Evaluation of 24 bit input data (Simtools/Mover)
24. Calculation in 32/64 bits for maximum motion smoothness
25. 2 different operation modes: direct or inverse kinematics
26. Actuators can be operated in an infinite loop (like YAW VR)
27. Support for Belt systems
28. Operating hours counter

2.3. Motion cueing software

As an interface between the simulation / games or telemetry data provider, a motion cueing interface is required to control the platform. This software receives the data, converts it as required, filters it and sends this data to the controller, which then calculates and implements the data / position of the motors.

With the Motion 4 Sim Controller it is possible to transfer various data. If you have your own protocols, these can be implemented.

Currently supported motion cueing apps are:

Flypt movers	https://www.flyptmover.com/
Simtools	https://www.xsimulator.net/simtools-motion-simulator-software/
SimMotion	https://eu.sim-motion.com/software/
Sim Racing Studio	https://www.simracingstudio.com/ (not official)
BFF Motion Software	http://bffsimulation.com/6DOF-Motion-Software.php
YAME	https://hexago-motion.com/downloads/
M4S dashboard	https://github.com/motion4sim/AASD15A-Servo-Controller-for-Motion Simrigs/tree/master/Dashboard

Direct access can also be done via a terminal program.

3. Board and connectors

There are several hardware versions that are compatible and run on the same firmware.

Version 1.04 with SAME70 xpld board and breakout board for the interfaces Fig.1.

Version 1.05 with ATSAME70 MCU with FTDI or CH340 USB USART Fig.2

Figure 1 M4S Controller V1.04 with display and encoder

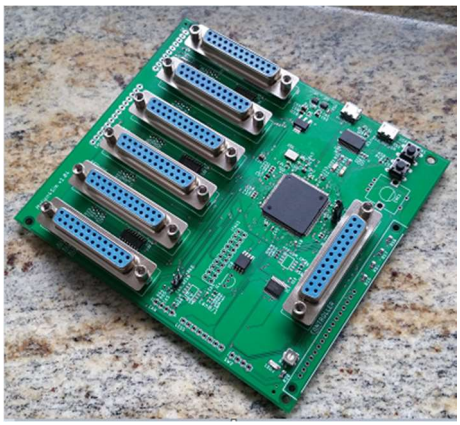


Figure 2 M4S Controller v1.05 FTDI

Version 1.08/ 1.09 /1.10 with ATSAME70 MCU with CH340 USB USART and 7. Servo Connector Fig.3

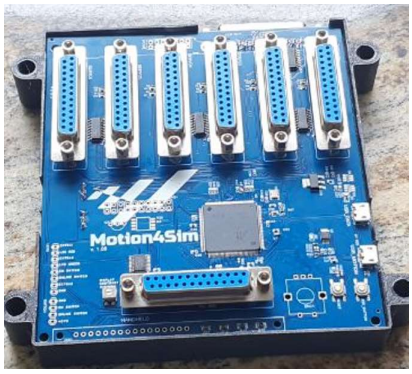


Figure 3 M4S Controller V1.09

There are also additional hardware options. Connector for Brake/Extra01

It is possible to control via a display and encoder directly on the board, see Fig.1. Alternatively, the handheld device can be used for remote control. See Fig.3

The handheld device is connected via a SUBD25 male/male cable. The cable can be up to 3m long. Cables 4.5 and 5m long have also been tested positively.

Figure 3 Handheld Device

There are 6 Sub D25 sockets for connecting 6 AASD servos as well as a Sub-D25 connector for connecting the handheld. Soldering strips are provided for connecting Servo 7 as well as for connecting “homing” switches for each motor. There are also pins for connecting LEDs as well as online switches and emergency stop switches. From PCB version 1.08, 7 servo sockets are installed.

1 7. Servo Connector >pcb V1.08

PCB version 1.04

There are a total of 2 Micro USB connectors on the SAM E70 board and one Micro USB connector on the Motion4Sim breakout board. There is a reset button and a USr button somewhat hidden on the bottom board.

Overview of connections:

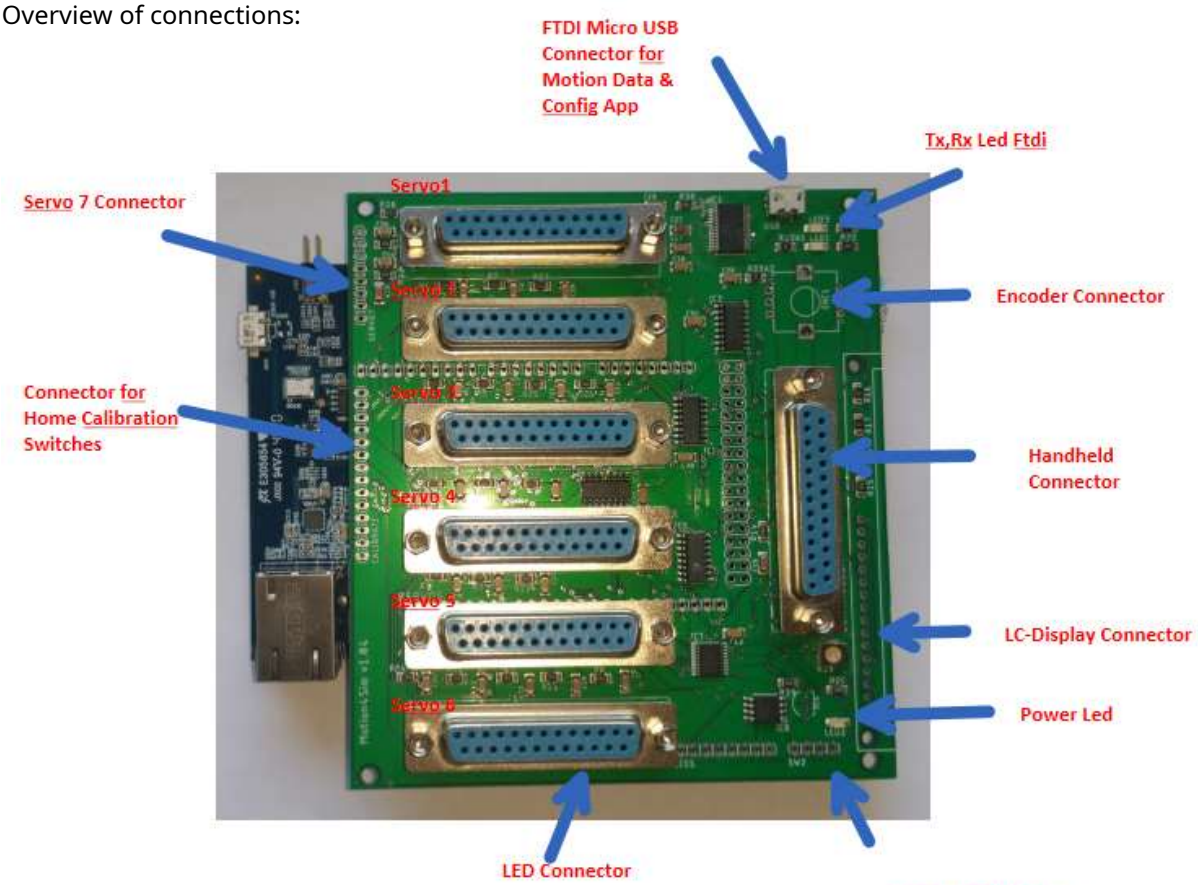
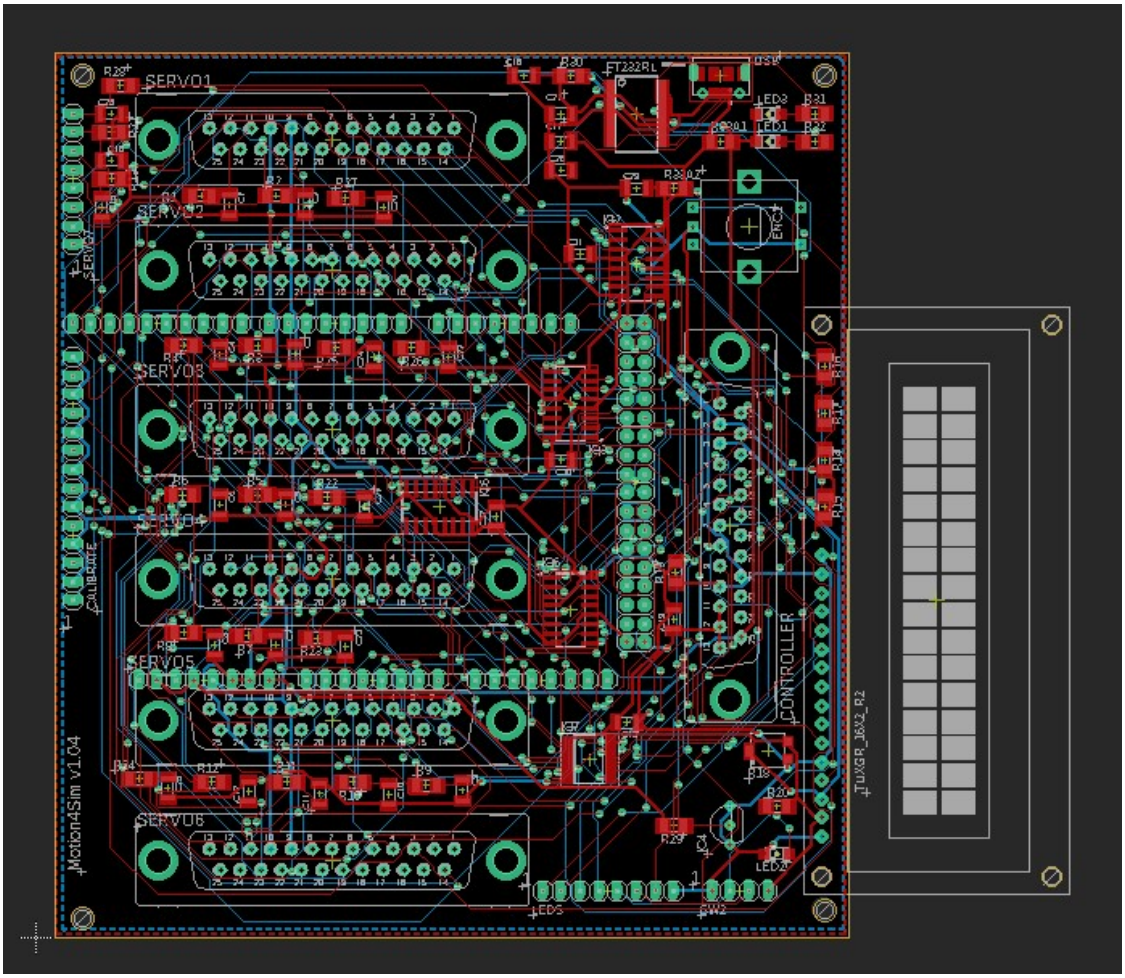
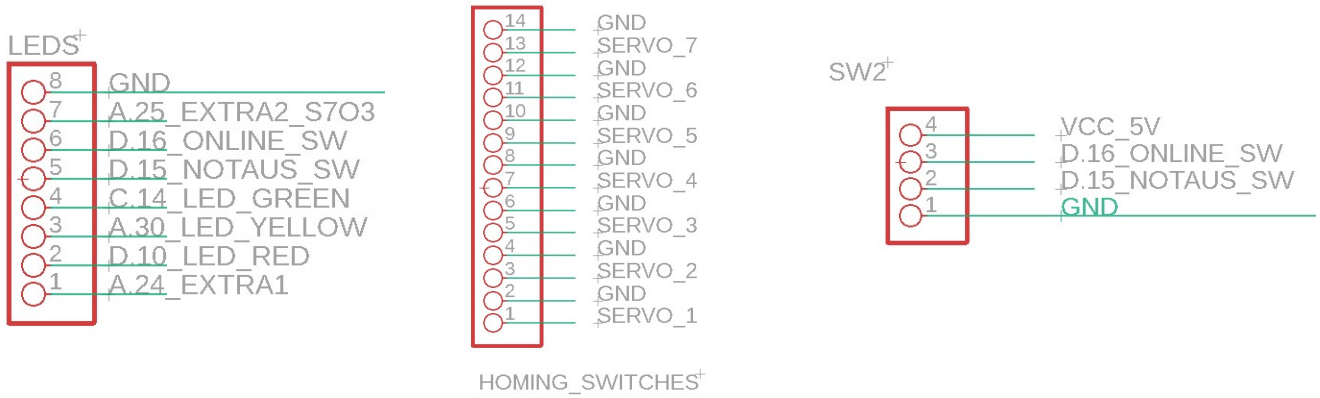


Figure 4 v1.04 Micro Usb port for flashing the firmware


SERVO1
SERVO2
SERVO3
SERVO4
SERVO5
SERVO6
SERVO7
CALIBRATE
Motion4Sim v1.04
FT232RL
CONTROLLER
TUXGR_16X2_R2
LED1
LED2
LED3
SW2

PCB Layout V1.04 - Pinout



LEDS are connected at 3.3 volts without a series resistor. The homing switches switch their signal to GND when in contact. The emergency stop switch is “normally closed” and the online switch is “normally open”. These settings are configurable in the software.

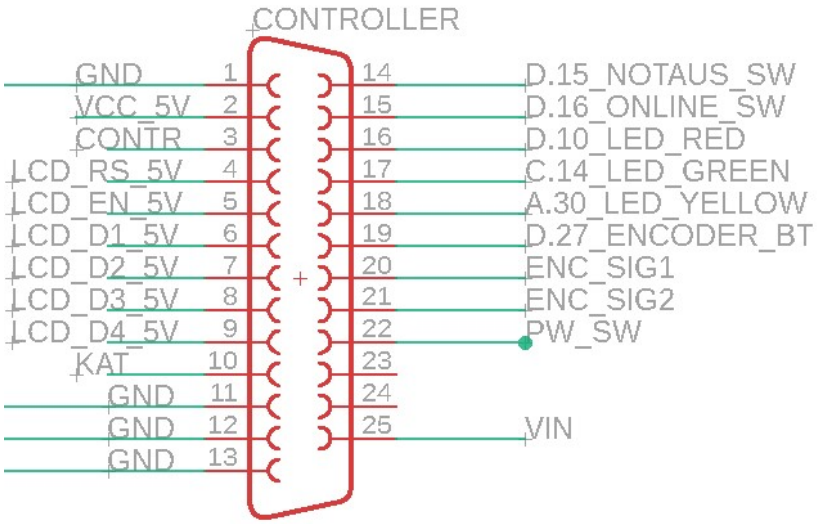


Figure 5 DSub25 handheld pinout

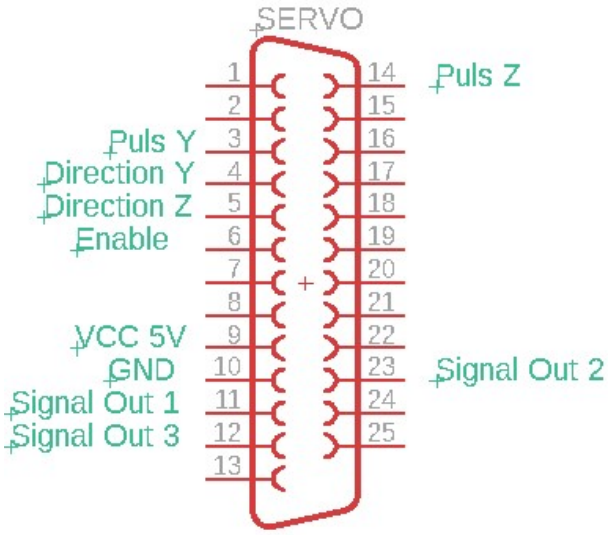


Figure 2 DSub25 Servo Connector Pinout

PCB version 1.05

Overview of connections.

The same connections are available as in PCB version 1.04. There are 6 connections for servo motors, a connection for the handheld device, 2 micro-USB connections, a reset and a USR button as well as various connections for LEDs, extra inputs and outputs as well as soldering pins for a 7th servo connection and its homing switches . To the left of the handheld port there is an adjustable resistor with which you can adjust the brightness of the LCD display.

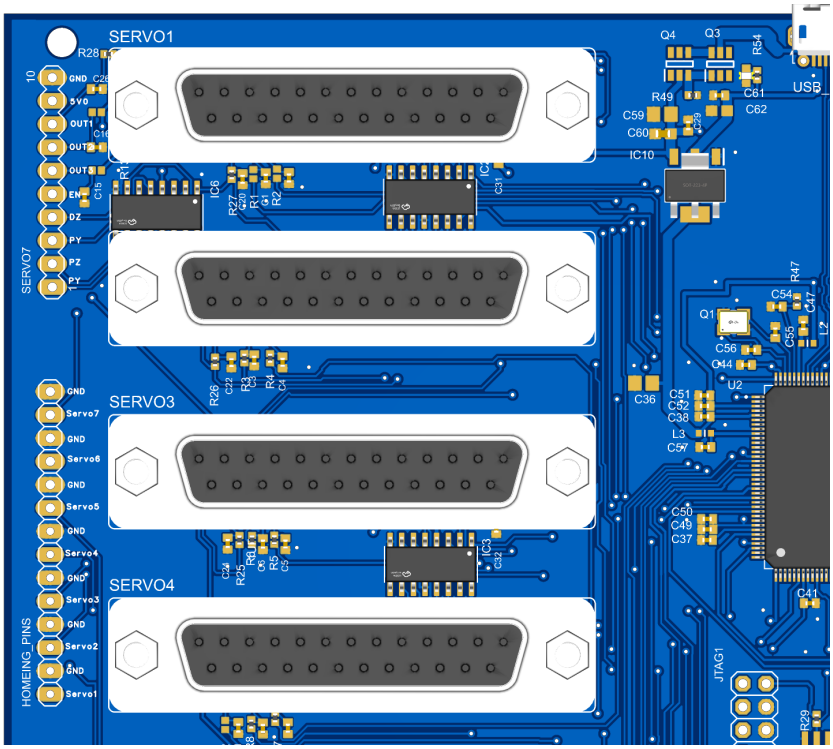


Figure 3 PCB v1.05 overview

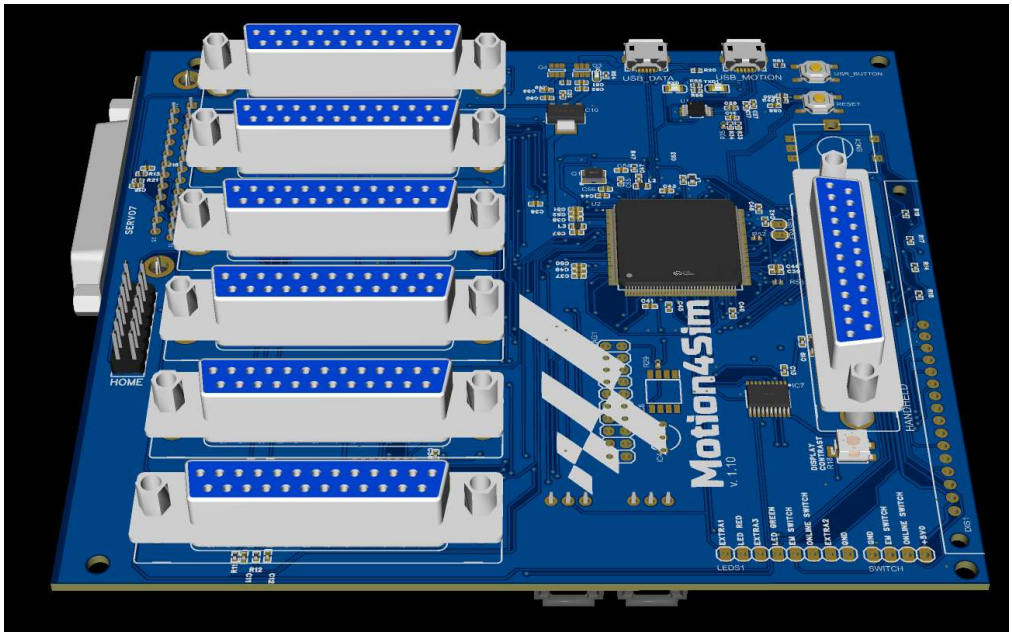
Figure 4 Pinout solder strip PCB v1.05 in detail

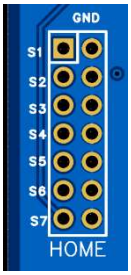
Version 1.05 with display and encoder is delivered with a screw terminal on the connections of SW1. Unfortunately, the labeling of version v1.05 is not implemented as in the visual graphics used here. On some boards delivered, some resistors have to be removed in order to use the Servo 7. As of the publication of these instructions, these have always been removed. These resistors were planned for functions that were not finally implemented in the firmware. If in doubt, contact support.

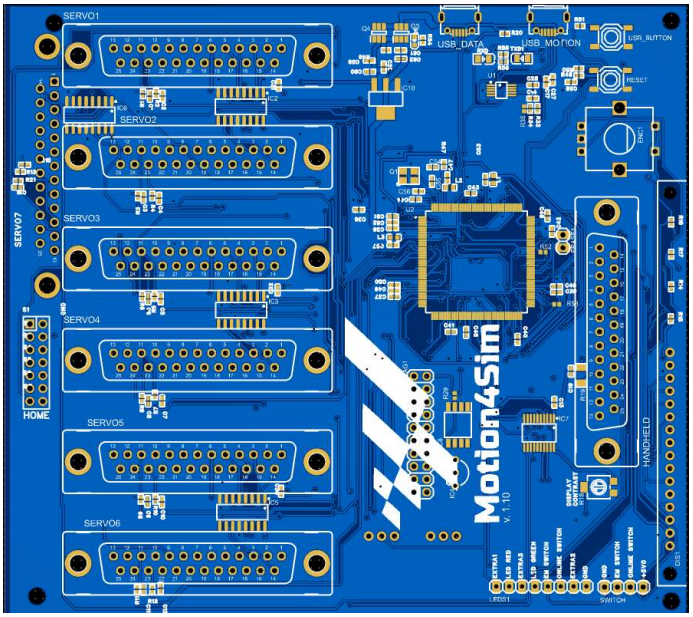
The pin assignment of the servo connections and the handheld are unchanged from the previous HW version. See fig.

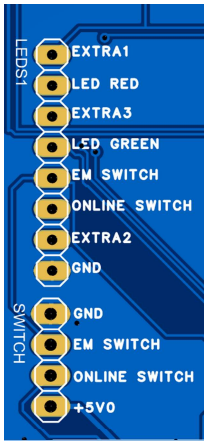
PCB Version 1.08, 1.09, 1.10

These versions have a 7th servo socket installed. The functions are compatible with all previous versions. In addition, the pulse pins of the servo are connected to the hardware timer outputs, which now makes it possible to pulse at higher frequencies. There are also sockets for Extra01 and Extra02. However, they are only installed ex works if requested.









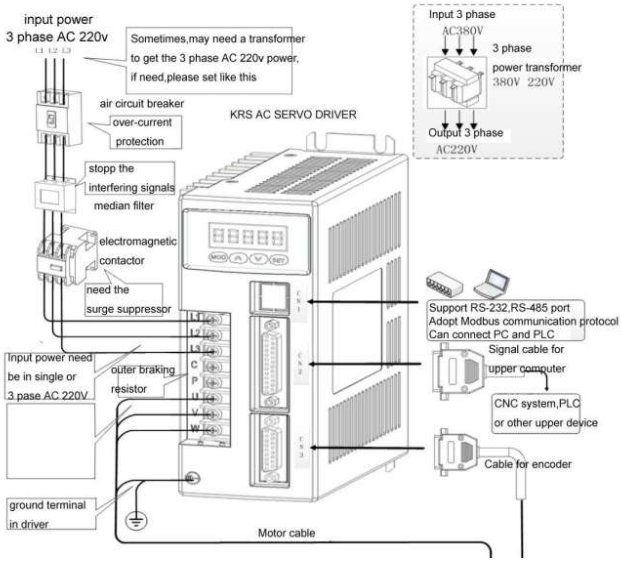
The pin assignment of the servo connections and the handheld are unchanged from the previous HW version. See fig.

To the left of the Dsub socket for the handheld there is a pot that can be used to regulate the brightness of the display.

4. Hardware cabling

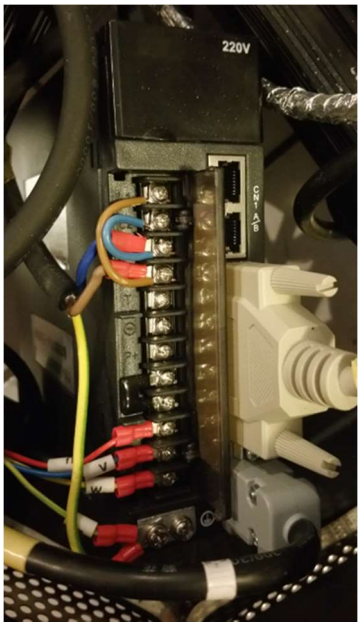
4.1. Servo motor amplifier wiring AASD

2.1.1 Servo driver wiring diagram



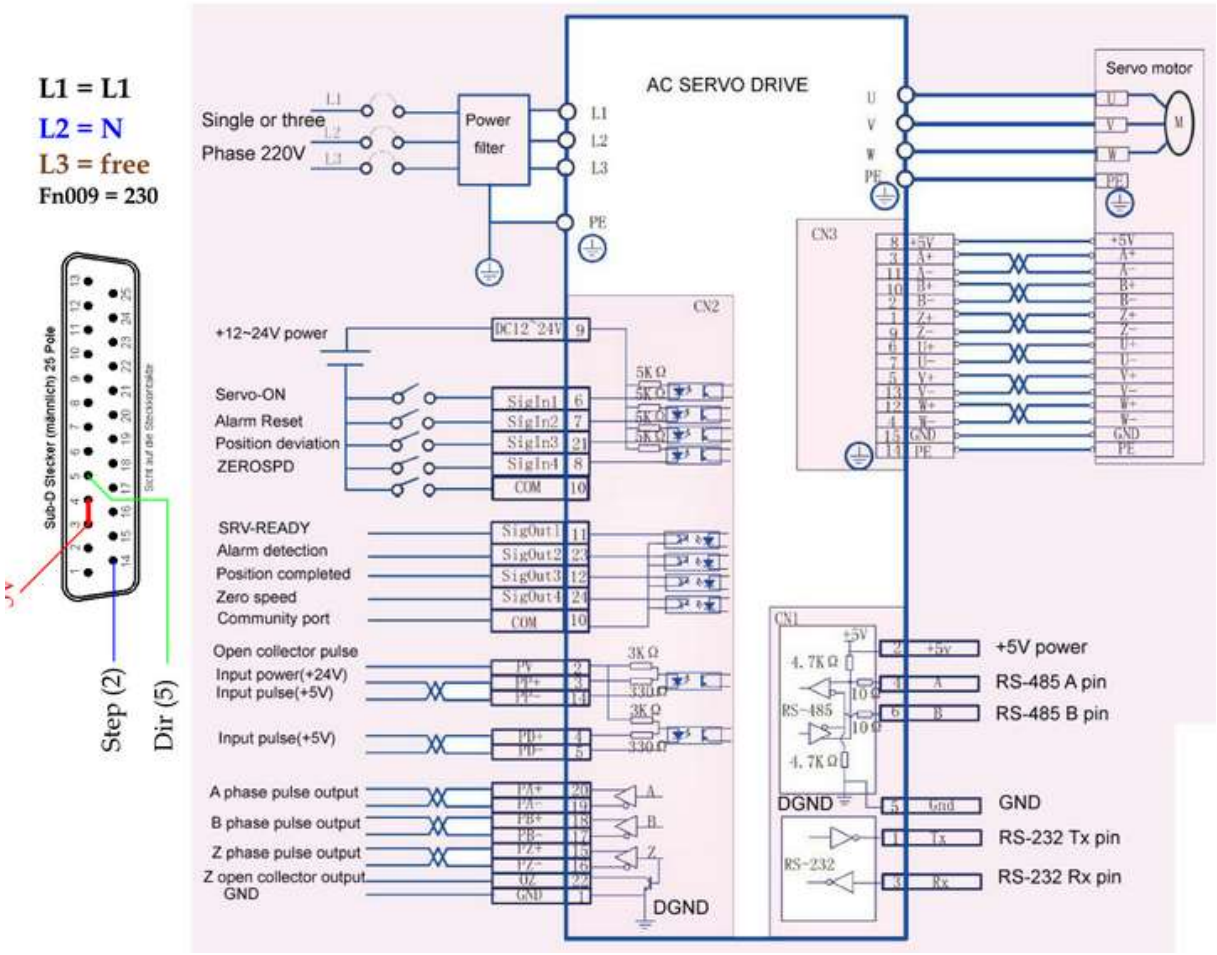
The mains voltage connections are sometimes different. Please always consult the relevant data sheet from the manufacturer. See also, for example:

https://github.com/motion4sim/AASD15A-Servo-Controller-for-Motion-Simrigs/tree/master/Manuals



2.3 Standard connection

2.3.1 Position control wiring diagram



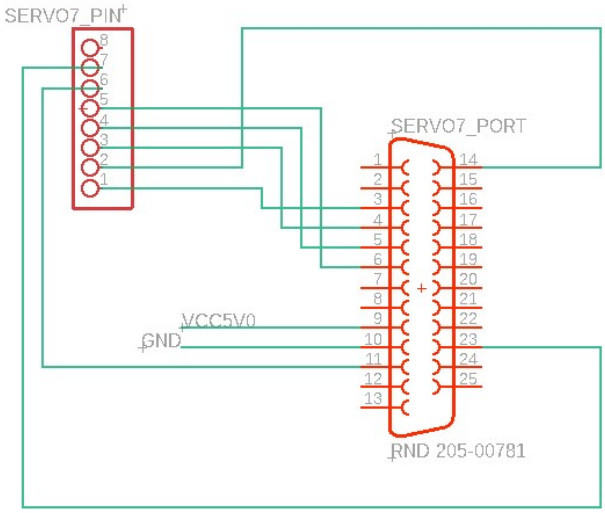
4.2. Wiring servo 7 connection

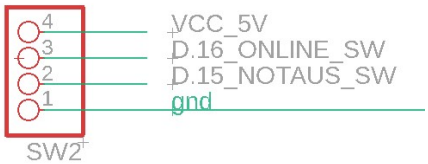
The current version has the 7th servo connection already installed.

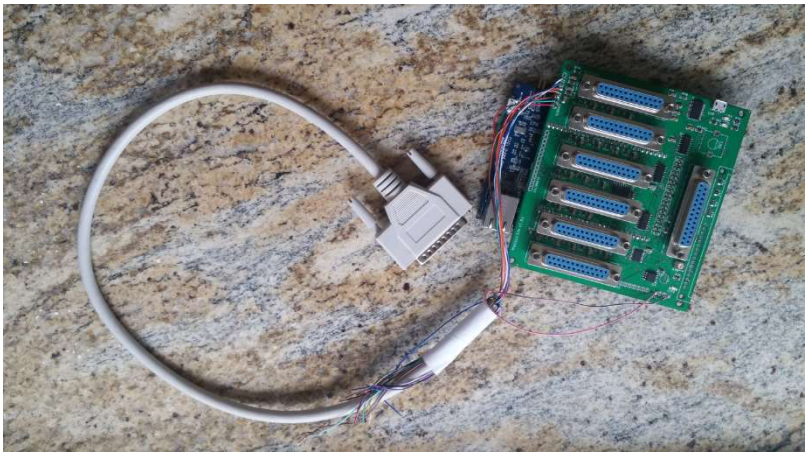
Wiring the pins for the 7th servo motor. It can either be installed directly on a Db25 cable or you can wire a female sub DB25 connector for soldering

In addition, +5 volts and ground are required, which can be used from SW2.

PCB V1.04

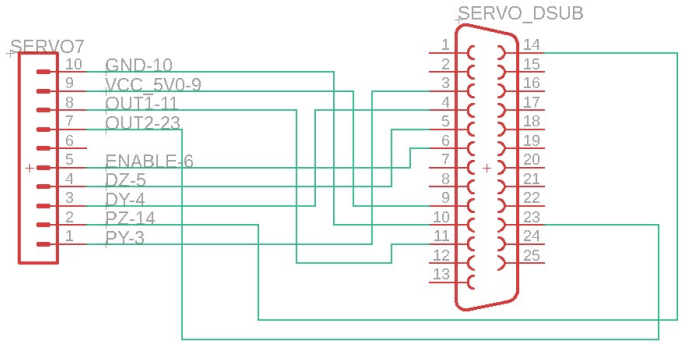




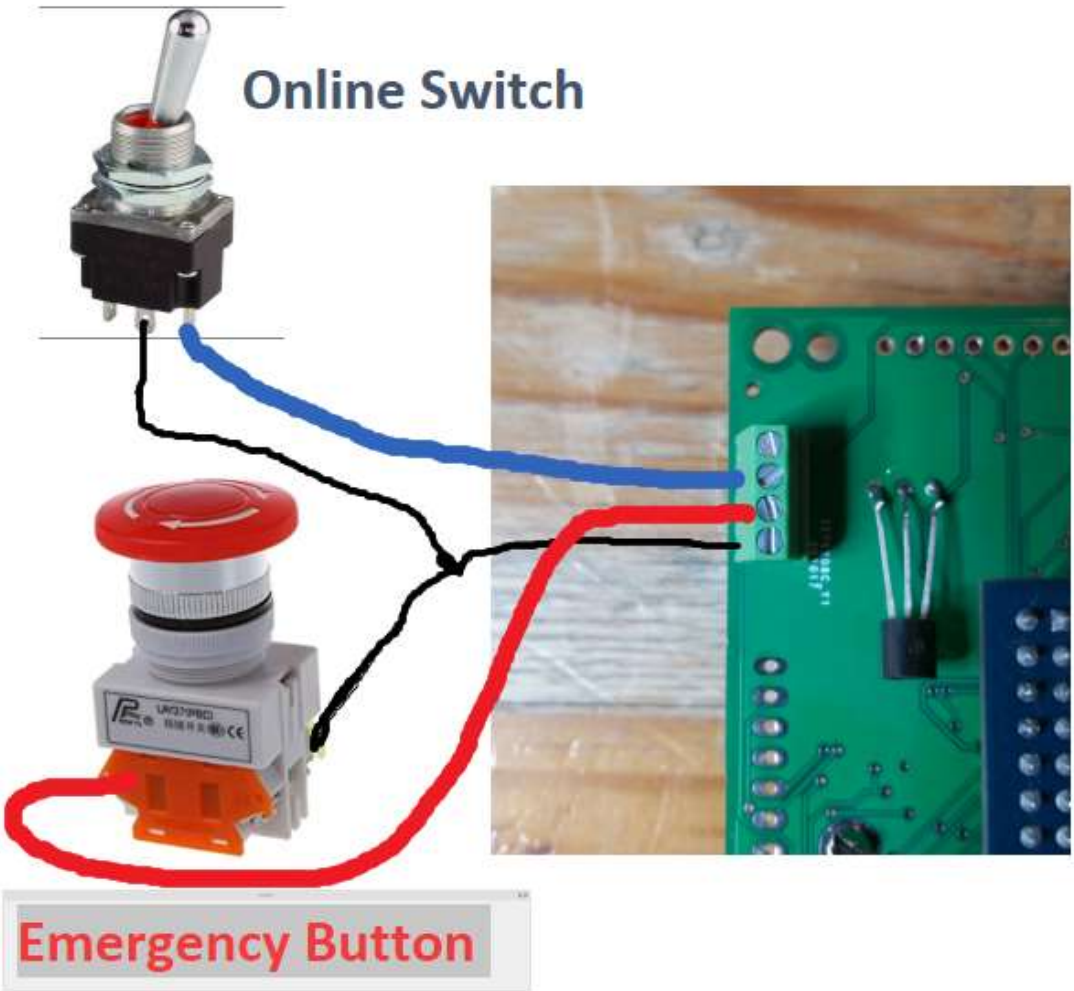


PCB V1.05

For this version, all connections on the pin strips are included.

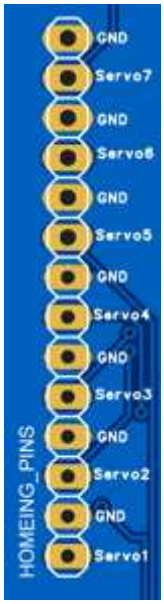


4.3. Wiring online switch and emergency stop switch



4.4. Cabling of the position limit switches/limit switches

The switches are connected via the corresponding pin strip. To trigger the switch, it must be switched to GND. On the board the pins are only labeled “Calibrate”. As can be seen on the left, the assignment takes place from bottom to top.



Inductive proximity switches can also be connected. These only require an additional 5V voltage from the board.

e.g.

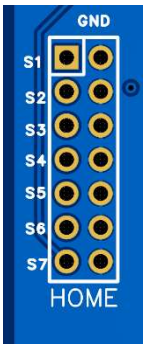
https://wiki.dfrobot.com/5V%20Metal%20Proximity%20Switch%20SKU%3A%20FIT0658

https://www.cytron.io/c-sensor/p-5vdc-inductive-proximity-sensor-npn-no

The brown cable is connected as a signal.

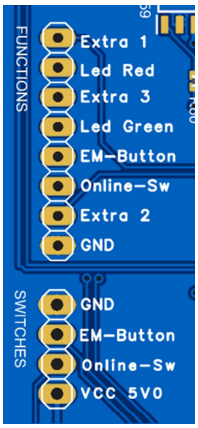
From pcb version> 1.08, the homing switches are to be connected according to the following figure

S1-S7 are the contacts that are connected to the GND right pin.



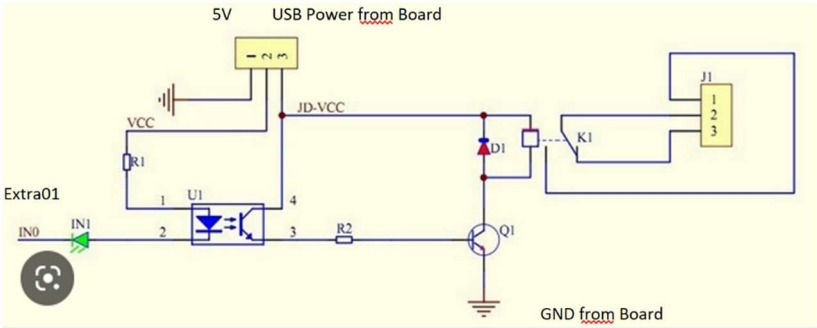
4.5. Connection of a servo brake

Some external functions can be controlled via the extra 1 pin on the LED soldering strip, including a servo brake.



In the software configuration, the corresponding function must be assigned to pin Extra01.

The connection must be made via a relay. The pin can only handle a current of 30 mA and 3.3 volts.



The servo brakes usually have an operating voltage of 24V, but an additional power supply is required. The well-known Arduino relays can therefore also be used.

Additional Extra 01 and Extra02 connection from PCB V >1.08

5. Servo amplifier settings

The controller can be connected to all servos that have a DB25 connection. It is also possible to make your own adapter cables. Ask about compatibility.

AASD-15A Servo Settings:

Push MOD until you see Pn000. This enters the parameter mode.

Change and check these settings on all motors:

FN9 = 230 (230Volts recommend to set this value) Check this value if you get errors

The comparison voltage is measured and saved with FN9.

Pn8 = 300
Pn9 = -300
Pn51 = 3000
Pn98 = 1-20 - Pulse Multiplier (electronics gear) different to Thanos and SFX for higher resolution

Typically on M4S you can set this to PN98 = 2

From pcb v >1.09 the controller supports higher pulse frequencies so it is possible to use PN98=1 with higher frequencies and so the motor also runs full speed.

Pn109 = 1 - smoothing, 1=fixed smoothing, 2=s-Shaped smoothing
Pn110 = 30 - Smoothing Filter Time
Pn113 = 20 - Feed forward %
Pn114 = 10 - Feedforward Filter Time (ms)
Pn115 = 100 - Gain %
Pn24 = 100

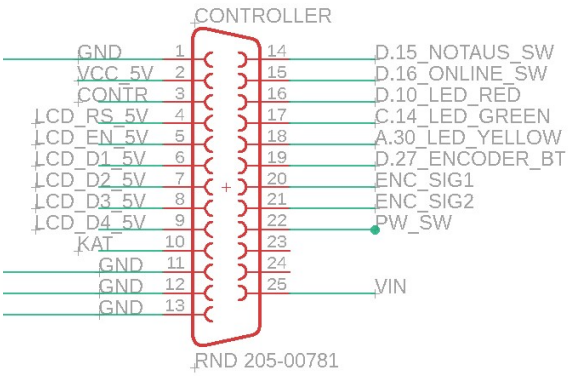
Pn51 = 3000 Motorspeed (2500 or 3000 mainly)
Pn52 = 1 Signal Port 1 Servo enable
Pn60 = 2 Sigout Port 1 Servo ready
Pn61 = 6 Sigout Port 2 Servo Treach

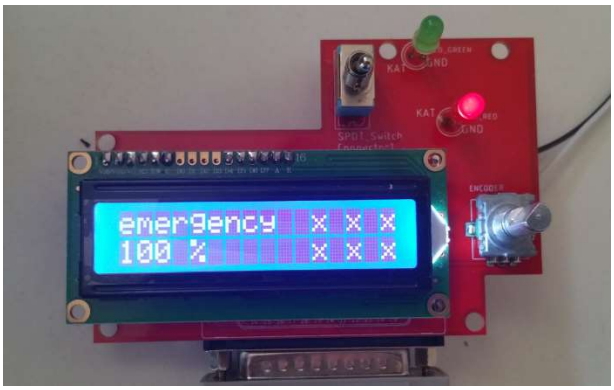
Pn62 = 4 Sigout Port 3 Servo Preach

PN24 = 100 here it shows that you can reduce the torque for the homing to better protect your actuators i set PN24 to 80%

Please also read the AASD servo amplifier manual

6. Handheld device





For further information see also

https://github.com/motion4sim/AASD15A-Servo-Controller-for-Motion-Simrigs/tree/master/handheld

and

https://www.thingiverse.com/thing:4641555

7. Setting the controller / menu control

When you switch on the controller or supply it with USB power, the display activates and loads the last used settings from memory. It then waits for user input. When you press the encoder, automatic calibration/homing is started.

There is also the option of turning the encoder to the left. This enables you to access the service menu without carrying out the calibration. (2nd picture) if you turn the encoder further to the left, there is still the possibility of performing a firmware reset. (Image 3+4) Attention, all settings will be lost and standard settings will be loaded

factory reset for rotating actuators

factory reset for linear actuators

Special factory reset for motionxp rigs

By pressing the encoder you will go to the main screen after pressing the encoder button.
Here the status of the first 6 actuators is shown on the right.

Possible actuator states:

X	is	Offline .. no actuator connected or error
P	is	Preparing (searching for the homeing position)
!	is	Moving to the offsets/safety parking position
R	is	Ready for use

The status of the rig system and which movements are shown in the top line of the display
the machine executes.

preparing..	calibrate and move to offset position
emergency..	Emergency stop switch activated
offline..	Offline switch off, parking position
wait uart..	Movement to home position / in standby position waiting for serial data
on-line..	Serial data is received and movement takes place
act.err	Actuator check is activated and not all actuators are working properly

Functions of the encoder on the main screen:

The status of the master gain control is displayed at the bottom left. (100%)
If you turn the encoder left or right, you reduce or increase the amplification of the pulses in 10% steps. The gain has a range of 10% - 400%. The gain can also be changed or blocked in online mode

If you press the encoder in the main screen you activate the submenu structure. The menu structure is structured as follows.

1. Shutdown	- Park actuators and switch off	
2. Home Belt	- Tighten the belt	
3. Homing	- automatic re-calibration	<- Jump to menu activation
4. Filters	- Output filters	
5. Service	- Menu for manual actuator movements	
6. Actuators	- Actuator settings	
7.Setup	- Controller settings	
8. Rig	- Settings for Stewart geometry	
9. Reboot	- Restart controller	
10. Save and return	- Save current settings and close menu	

You can navigate through the menu structure by turning the encoder left or right. By pressing the encoder you activate the respective menu item.

There is another input feature. By quickly double-clicking the encoder button, further quick access to important value settings can be carried out

The structure is as follows.

-Main screen with gain setting

Double click to go to

- Belt Retreat quick adjustment. Changes are also implemented immediately when the belt is in use.

-Actuator Actuator Filter Quick setting for adjusting the internal EMALP filter

Another double click will take you back to the original screen.

7.1. Filter menu

The following options can be set.

Filter enabled	:	Turn filter on or off
Filter type	:	EMALPExponential Moving Average Low Pass filter (like in Mover) EMALP & SPIKE
Filter Samples	:	Strength of the selected filter. The bigger the Value the smoother the movement
Spike filters:		This filter is designed for excessive unwanted movements. This value influences the smoothing of the pulses if the spike filter was activated by exceeding the limit. The strength increases exponentially (filter ^4) the larger the window exceeds. Video about setting the spike filter: https://youtu.be/j6xVlIVM2L4
Spike window:		If it is exceeded, the filter is activated. 100 roughly corresponds to the pulse distance of half of all possible pulses of the actuator. The key here is to try it out for an optimal setup. The lower the value, the sooner the filter is activated. At 1, all pulses are filtered. _bithalf *RIG.Spikewindow *RIG.Spikewindow * RIG.Spikewindow / 10000000;
Sp.Filter smooth:		The spike filter operates dynamically and changes the smoothing accordingly. That's why another EMA filter was installed for the dynamic change, which is controlled by this value. Small value Spike filter increases quickly and jerkily when triggered and also returns quickly and jerkily after the spike ends. Large value Spike filter increases more gently and smoothly when triggered also returns more smoothly to the initial filter value.

7.1.1. Quick guide to setting the Spike Filter

1. Start in default settings after a factory reset
2. Create a motion loop e.g. in Mover (Saw wave) with a high amplitude
3. Or alternatively, drive the car into a wall in a simulation, which triggers strong acceleration on the actuators.
4. Reduce the value for “Spike Window” until the spike becomes significantly stronger and find a tolerable value.
5. Slowly increase the Spike Filter, Attenuation and Spike Window until you find your setup.
6. “Sp.Filter smooth” is only used for fine-tuning and is actually not of great importance.
   Try it for yourself here.

7.2. Service menu

This menu is intended for testing the actuators and moving the actuators to specific service positions. All actuators can be moved simultaneously or each one individually. After activating the respective submenu, the value of the actuator can be increased or decreased by turning the encoder right or left. The value is changed by the value stored in “Multiplier”. This menu can also be accessed directly when starting the controller without calibrating the actuators. This can be used, for example, to “jack up a Stewart platform” in the event of repairs. When you exit the menu, a calibration is carried out.

This function can also be used to save the offset of individual actuators and all actuators. For Change Offset, set “Yes” with the encoder and confirm with Push.

7.2.1. Repair – Moving without “Homeing”

To maintain the rig or an actuator, it may be necessary to move one or all of the actuators without having carried out “homeing”.

After starting the controller, select and confirm with the “Encoder”.

Move the encoder right or left and select one or all actuators to move.

Please note that a calibration will be carried out after exiting the menu.

If you press the emergency stop button, the servos will be deactivated depending on the setting in the setup.

Note: By turning the encoder, the actuator is always moved by exactly the number of pulses set in the “Multi”. Use a different “Multi” setting to move the actuator faster.

7.3. Actuators Menu

The parameters of the actuators used are set here. Save after changes and restart the controller so that the new values are calculated correctly.

The following values apply to all actuators, changes overwrite the individual values of all actuators.

RIG : Linear or Rotary, Custom

These values are used for the calculations of the inverse kinematics and to calculate the resolution of the actuator.

6dof Rotating Actuator

6dof Linear Actuator

SFX 100

SFX 150

PT100 4Act

PT100 4 Act TL

PT100 4 Act TL SR

PT100 4 Act TL TL SR

PT150 4Act

PT150 4 Act TL

PT150 4 Act TL SR

PT150 4 Act TL TL SR

SFX 150 3 Act 2 front 1 back

SFX 150 3 Act 2 back 1 front

Custom

Encoder PPR : Number of encoder positions depending

Servo motor, at AASD15 10000

Electronics. Gear	:	For calculating the actuator resolution needed PN(98) Value from the servo output stage

These global properties overwrite the values of the individual actuators in the actuator sub-menus.

Gearbox ratio	:	Ratio of used gears. Just reduction possible.
Lead screw pitch	:	Pitch of the trapezoidal thread spindle 5, 10.25 ... in mm per revolution (linear only actuators)
Actuator Length	:	Length of the linear actuator in mm (only linear actuators)
Red.Range	:	“Range” for rotating actuators in degrees. Half a turn is 180 degrees
Lin. Act. Safety	:	Safety distance to actuator limits in mm for linear actuators. Will be bilateral deducted from the actuator length.
Actuators	:	Sub-menu for actuator 1-8 Activate by press the encoder button

(Sub-menu for actuator 1-8)

rotation	:	Direction of rotation of the actuator, CW vs CCW
Park enable	:	If you activate this option the Actuator not parked
Cal. offset	:	Offset in pulses is after calibration set. And as the lower zero point of the Actuator used. (plus parking position only for rotating actuators)
elect. Gear	:	see above
mech.Gear	:	Ratio of used gears. Just reduction possible.
Encoder Pos	:	see above, resolution of the encoder
Parking position	:	Offset from lower zero point (calibration point) of the actuator as Parking position. Offset is total for everyone Actuator individually calculates parking position + Actuator Cal. (rotating actuators only)
LS Length	:	Length of the linear actuator in mm (linear actuators only)
LS Pitch	:	Pitch of the trapezoidal thread spindle 1… 1000 … in mm per revolution (linear actuators only)
LS Safety	:	Safety distance to actuator limits in mm for linear actuators. Will be bilateral deducted from the Actuator Length. Type. 2-5mm
red.Range	:	“Range” for rotating actuators in degrees. Half a turn is 180 degrees
Limitless	:	special function for e.g. 360 ° actuators for Endless YAW. This makes it possible via unscrew the actuator end without the

Having to take a long distance is best

Watch the video on YouTube

Infinite loop on actuators

https://www.youtube.com/watch?v=wvK86g_H9gZg

Belt function : the actuator is used as a belt tensioner.

Here the servo is driven against the resistance for just a few pulses, which causes a moment to build up and the pilot feels the pressure from the belt. The working direction is set with “Rotation”. During homeing the servo is calibrated to maximum torque. The belt then moves back by the pulse rate set in “Belt retreat”. This is supposed to represent the relaxed belt. The value should be adjusted by trying. The torque is set in PN8, PN9 and PN24 for AASD servo motors.

If the rig is switched “offline”, the belt should be further relaxed and runs the pulses set in “Parkpos”. If a loose belt is required, the number in “Parkpos” should be increased

Belt autohoming : the rig will be switched “online”.

At the same time, the homing of the belt is carried out. The driving speed of the servo motors is briefly switched to “home speed”.

Belt retreat : Number of pulses to relax the belt

after homeing.

//End of Actuator Sub menu

Home Speed : Pulse speed for calibration in Hz

Park Speed : Pulse speed for the slow

Speed, procedure to and from

Park/standby position

Online speed	:	operating speed, should be at least 253 kHz with PN98=2
Number Actuators	:	Number of connected motors, Die Number is controlled, checked, if “Actuator check” is activated.
Multi	:	The multiplier is used for value changes used where numbers in high amount need to be changed. So that the Encoder cannot be rotated “1000” times If you have to, you set the multiplier to 1000 and turns just one click. i.e. if multiplier =1, 1 is always added/ subtracted from the value per encoder detent. If the multiplier =500, 500 is added or subtracted. For offset and speeds, parking positions etc.

7.4. SET UP

Emergency Stat	:	Indicates how the engines start after activating the emergency stop Switch can be controlled. Pay careful attention to this setting. Your safety depends on it.
	“power off”	Servos are switched off (RIG falls down) Homing is carried out after switching on again.
	“hold power”	Servos stop but are powered, no movement. Position of the motion platform is maintained. Homing will be performed after switching back on.
	“keep online”	Servos remain energized, no movement. Position of the motion platform is maintained. Homing will be switched on againnotcarried out. The previous movement is picked up and continued. In online mode, a transition to the current movement is carried out. Attention: In the event of an error, homing will be lost; the actuators may move towards their end and damage may occur.

Emergency NC : Normaly open or normaly closed, this is what you choose the switching status of the emergency stop switch.

Actuator check : Checks all actuators when the servo motors are online are. If a fault occurs, there will be no signals more sent to the engines. This should be taken into account for this, all servo connections one after the other and all set motors too must be connected. This means, for example, connection 1-4.

If ports 1-3 and 5 are occupied, this results in one Error if actuator check is “enabled”.

When the error has been corrected the Actuators recalibrated.

Encoder direction : Some encoder types (handheld) deliver like detected the signals incorrect. Should be an encoder You can switch the wrong way round with this option Direction of rotation can be changed.

Extra01 : The function of the pin is set here. There are the following functions:

-nothing	doesn't matter
- servo brake nc	the servo brake switches to normal closed
- servo brake no	the servo brake switches inverted.

- send clock gives a pulse of 35 every 1ms

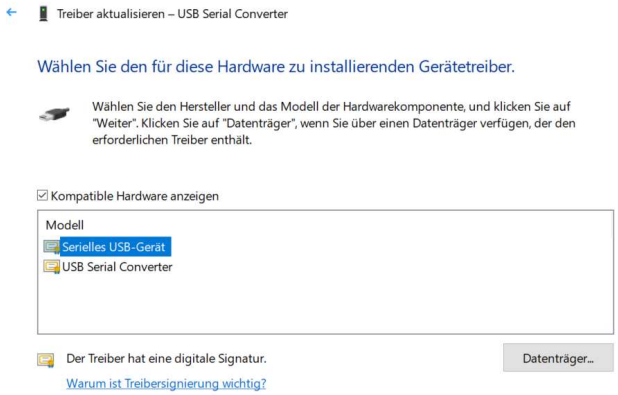
Brake delay	:	sets the delay for switching the brake relay on in [ms] Standard is 50ms.
Servo_enable	:	Changes the pin state from high to low for the servo enable signal.
Master gain max	:	sets the upper value of the master gain.
Master gain enabled	:	ON determines that the master gain is changed Can. OFF prevents this
Online position	:	sets the standby position in percent. I.e. if If you switch online, the system first moves to the standby position, usually the actuator center 50%, and then goes into motion via the transition. If you set the standby position to 0%, you go directly into motion via the transition.
Transition speed	:	Transition mode speed. (Morphing into Motion)
Online time	:	Operating hours counter Can be done by pressing the USR button >12 seconds to be deleted
OnlineCounter	:	ON operating hours counter is not activated when OFF Be careful using it for display problems as the counter is saved in flash. If you use it frequently, the flash memory will also become corrupted
Time-out	:	switches off the servo motors after a standstill period. OFF or 10 min to 120 min
SRS supp.	:	The native USB “Flashport” is emulated as an FTDI USB chip. This makes it possible to automatically connect to SRS. The controller then works in “Thanos mode” and the resolution is significantly reduced.

Please note the following when activating:

The flash port now simulates an FTDI USB chip because SRS only connects to one. However, the FTDI driver does not work with emulation. Therefore, a different driver must be selected manually under Windows. Simply a standard serial driver. To do this, proceed as follows.

1. Enable the setup setting “SRS supp.” To ON
2. Save the setting in the controller
3. Restart the controller…a faulty “USB Serial Converter” is found

4. A new driver is installed manually for this device in the device manager
5. -> Update Driver -> “search for drivers on the computer” -> “select from a list” -> “Serial USB Device”



The dashboard and Flypt Mover also work with these settings

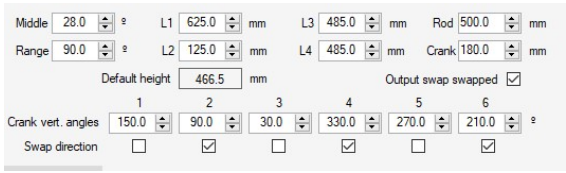
HW pulsing	:	From pcb version 1.09 onwards, the controller can use the hardware timers for pulsing. This makes significantly higher pulse frequencies and therefore higher resolutions of the motors possible at full speed. The AASD motors support pulse frequencies up to 600khz. If the function is activated, this will be displayed when the controller boots.

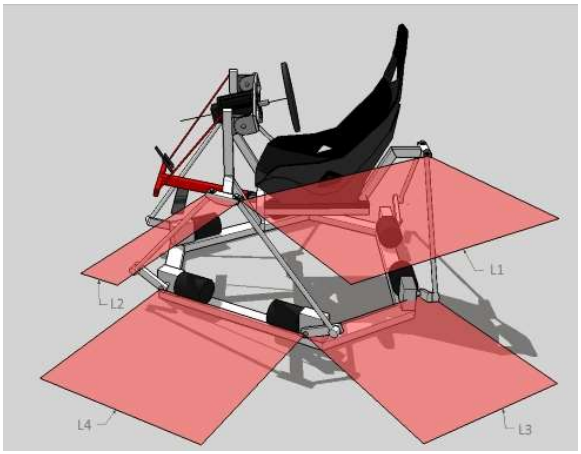
Enter bootloader	:	Starts the bootloader mode for flashing the firmware. Connect the Flash USB port to the computer and execute this option by pressing the encoder twice. The controller automatically starts in bootloader mode and you can install the firmware using the dashboard.
Multi	:	see Multi above

7.5. RIG

Parameters for inverse kinematics. Please configure your rig with Mover to understand how inverse kinematics works. The value entries are based on Mover and can be adopted directly.

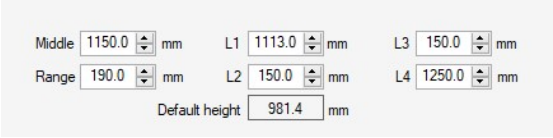
For rotating actuators and 6dof Stewart platforms

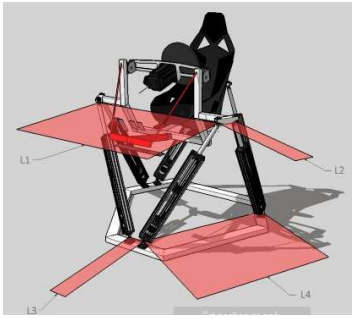




Source: FLypt “Mover” https://www.flyptmover.com/

For linear actuators





Source: FLypt “Mover” https://www.flyptmover.com/

List of values:

Base L3 short	short side of the lower platform in mm
Base L4 long	long side of the lower platform in mm
Rig L1 long	long side of the upper platform in mm
Rig L2 short	short side of the upper platform in mm
Crank/Range len	Length of the crank arm in red. Platforms
	Length of the actuator from the middle position to
	Actuator stop in mm for linear actuators

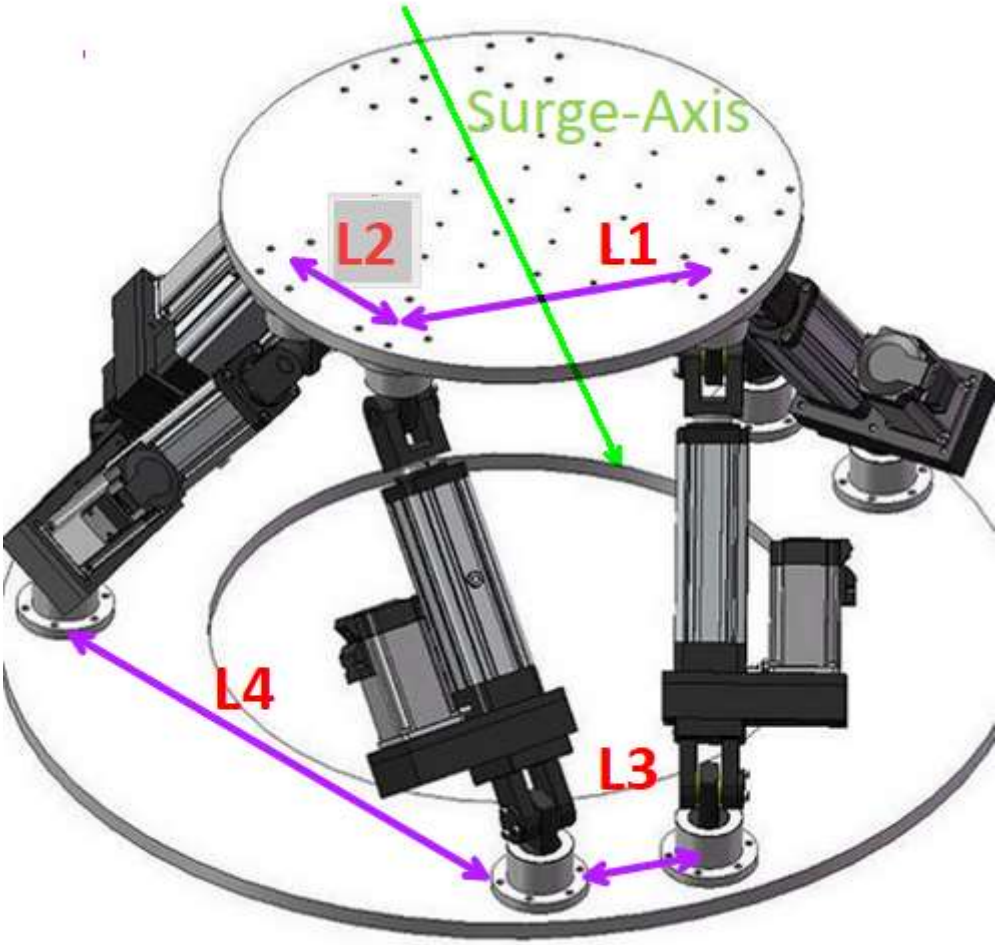
Rod/Middle len	Length of the connecting rod in red. Platforms Length of the actuator in the middle position at lin. Actuators in mm
Middle angle	only for rotating actuators, angle of the Crank arm in standby position of the actuator “determine by measuring”
Crank angle 1	Angle of the motor rotation axes in the base plane The angles are counterclockwise Measured. It's best understood if you paints it on.
Reversed 1	Angle + 180 ° ,
Crank angle 2	
Reversed 2	
Crank angle 3	
Reversed 3	
Crank angle 4	
Reversed 4	
Crank angle 5	
Reversed 5	
Crank angle 6	
Reversed 6	
XY orientation	With this option you can change the axes in the XY Just twist, given in degrees
COR X/sway	X value for the center of rotation calculation in mm from center
COR Y/surge	Y value for the center of rotation calculation in mm from center
COR Z/heave	Z value for the center of rotation calculation in mm from center

8. Geometry of kinematics W.I.P.

For calculations of the inverse kinematics of the Stewart platform, the geometric values of the platform are stored in this menu. These are queried as follows

Basic platform:Length L3 and L4

The basis for this is a regular or irregular hexagon

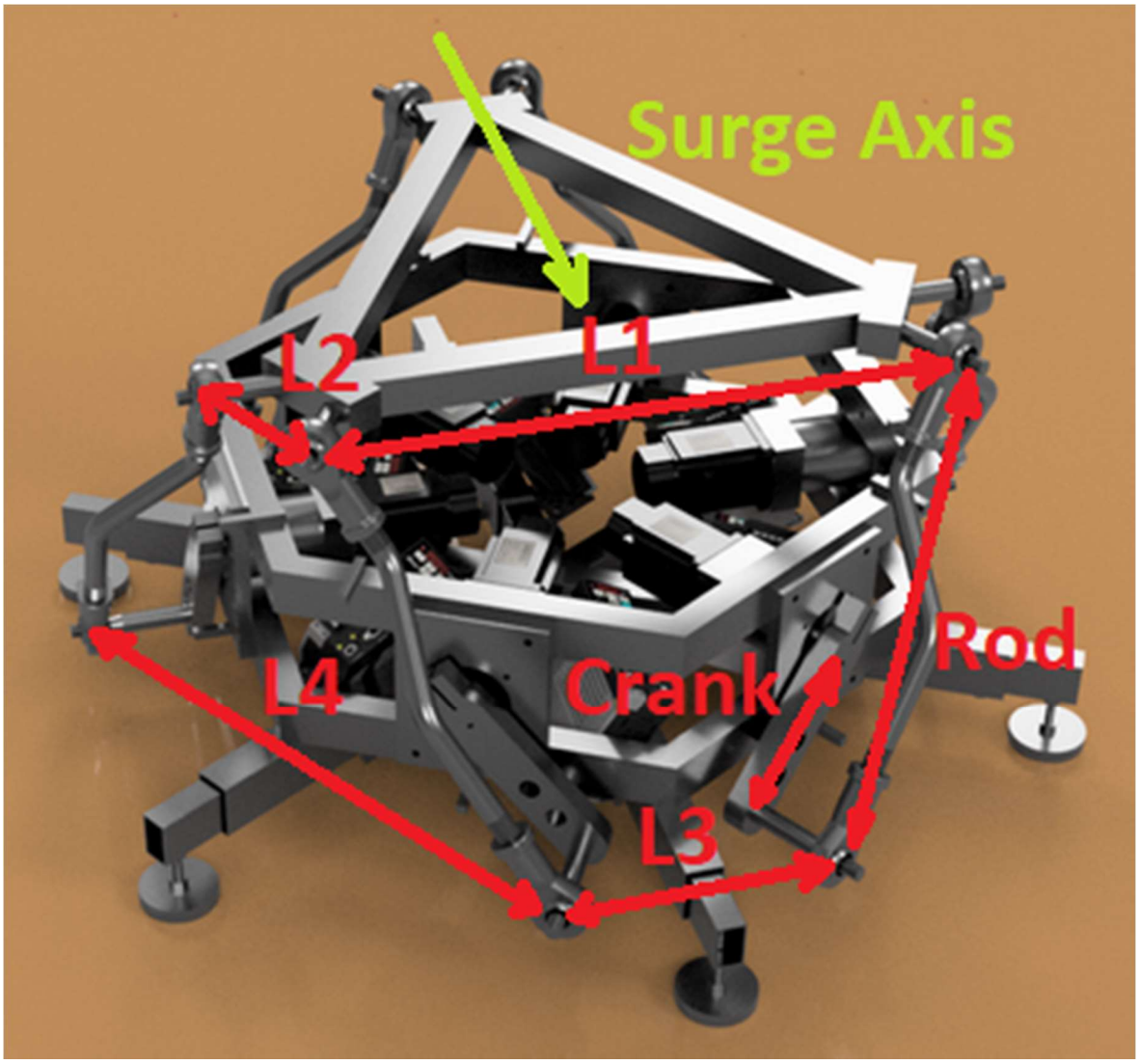


The same geometric basis also applies to the upper platform (the movement platform), but here an unequal-sided hexagon is almost always used as the geometric basis. The corners of which are turned 60 degrees to the base.

Overview of geometry:

Base L3 (shorter side), L4 (longer side)

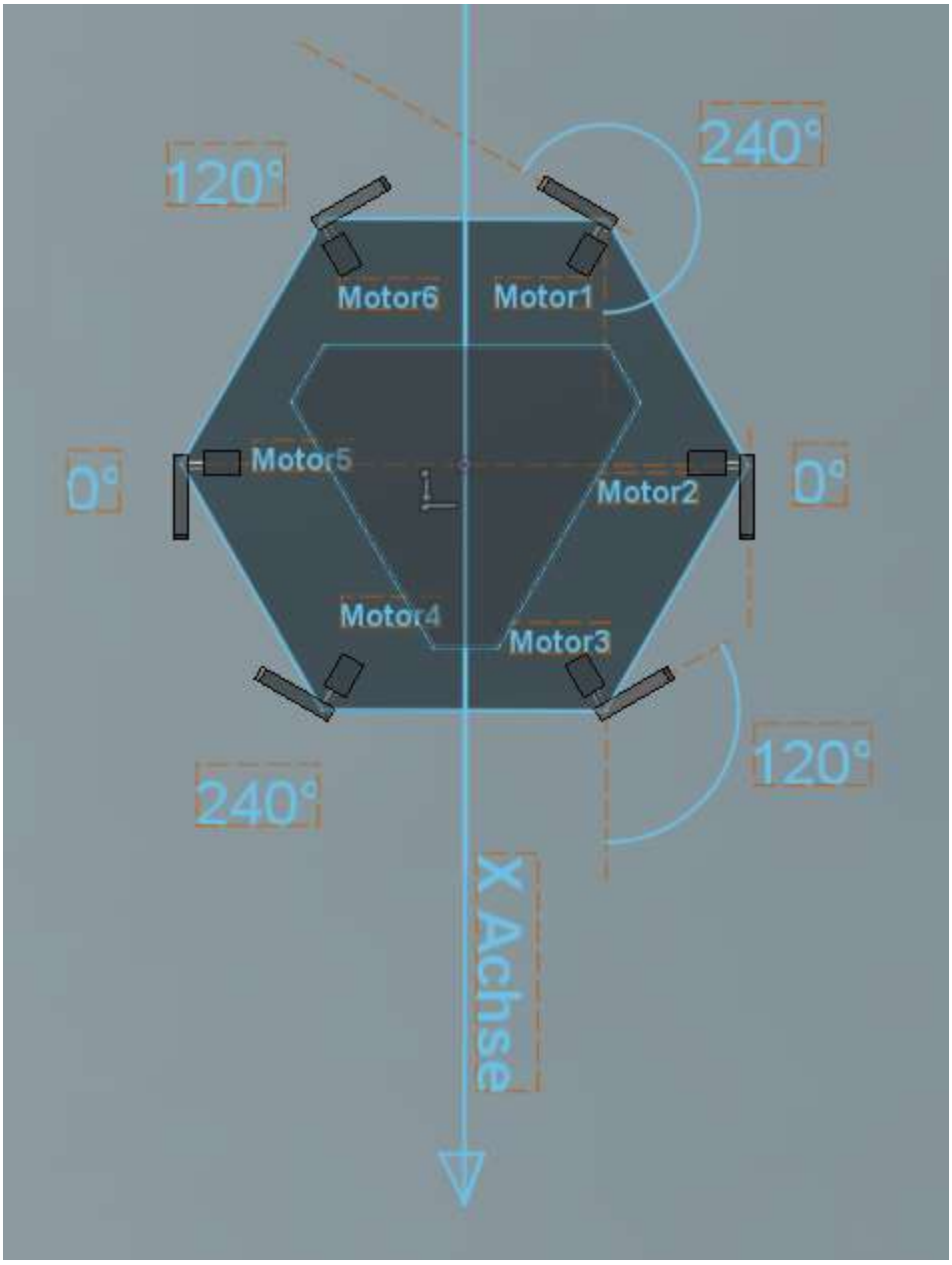
Cockpit L1 (longer side) ,L2 (shorter side)

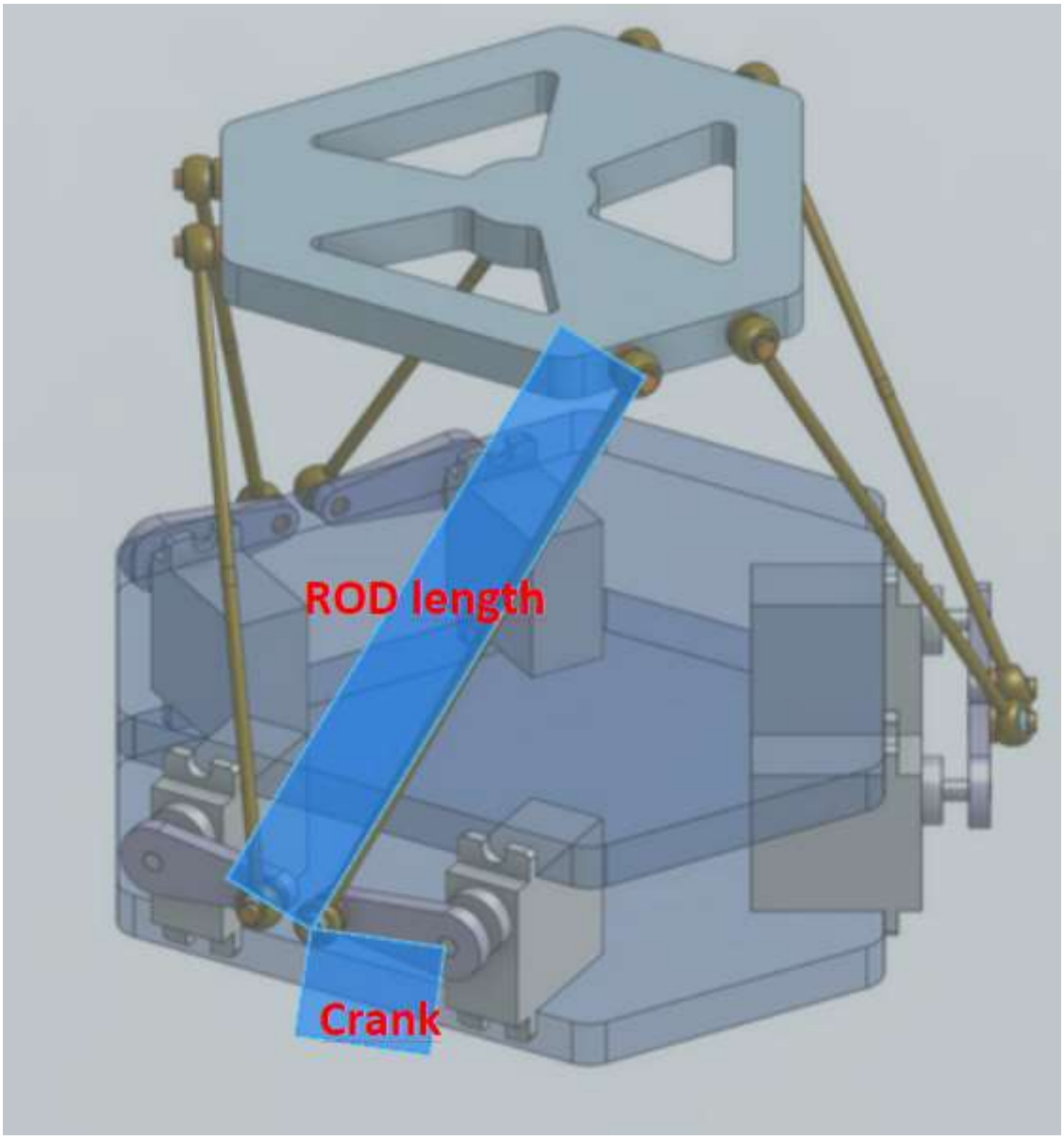


Uneven Base: Linear Hexpod

For the calculations for linear actuators you still need the total length of the actuator in the home position “Rod Length”. The value of the actuator range is taken from the “Actuator” menu. For rotating actuators you still need the angles of the lever arms of the motors in relation to the X-axis according to the following sketch. These are stored in Crank Angel[1..6]. In addition, the length of the connecting rod “Rod Length” and the length of the lever arms “Crank”

To geometrically define the home position you still need the angle of the lever arm in the home position. This can be determined mathematically (18°-29°).


120°
240°
Motor6
Motor1
0°
Motor5
Motor2
0°
Motor4
Motor3
120°
240°
X Achse


ROD length
Crank

9. Linear and rotary actuators

9.1. Rotating actuators

MaxPulse = RIG.ppr *RIG.mechanism_GearRatio_Servo

/RIG.elektrik_GearRatio_Servo

/ (360 /RIG.Range factor);

So for example: (rotating)

AASD15A 10000 ppr encoder

Gearbox 1:50

Range 180°

PN98 = 2

= 10000* 50 / 2 / (360 / 180)

= 125000 positions per half round

Resolution depends on arm length

Cal. Offset:

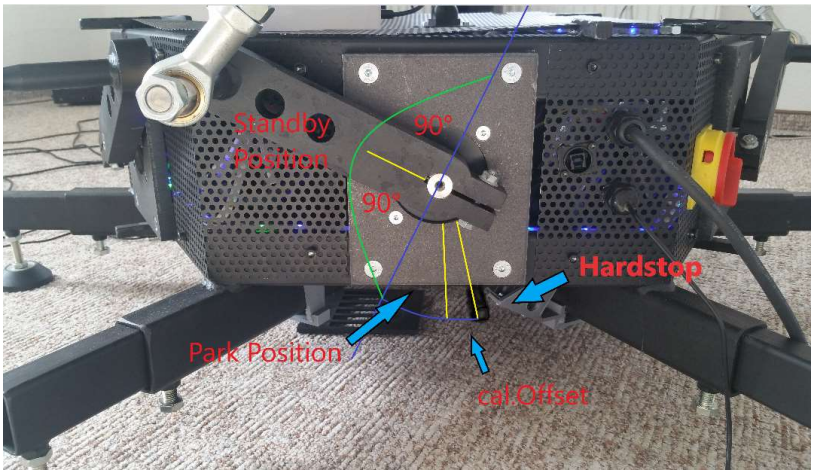
The offset is set and approached individually for each actuator. It indicates the number of pulses that are started after homing. This offset is used to compensate for possible geometric inaccuracies of the hard stop or the position limit switch

Parking position:

The parking position indicates the number of pulses that all actuators travel after calibration. This offset is used to uniformly determine the location of the parking position. Calculation: see above 62500 pulses would be a 90 degree rotation, for example

PS: You can easily calibrate the actuators using the service menu.

Standby position: Calculated from half the range in pulses.



9.2. Linear actuators

MaxPulse =RIG.ppr *RIG.mechanism_GearRatio_Servo / RIG.elektrik_GearRatio_Servo * ((RIG.leadscrew_length -RIG.lin_act_safety * 2) /RIG.leadscrew_pitch

SFX 100 Type

AASD15A 10000 ppr encoder

Gearbox 1:1

PN98 = 2

Lead crew = 5

Leadscrew length = 100 mm

Linear safety = 5 mm

= 10000* 1 / 2 * ((100 -2 * 5) / 5)

= 90000 positions / 100 mm

Resolution = 0.0011mm

9.3. Belt tensioner actuator

The following operating mode applies to these actuators:

To do this, the following parameters must be set on the servo driver according to your own needs.

1. PN98 =1..20 to translate the signal
2. PN24 =100% for the max torque signal in %
3. PN61=6

The belt actuator works according to the following principle. During homing, the motor runs until it reaches maximum torque. This is the end point of the actuator. (Maximum tension) After reaching it, it returns the value set under “Belt Retreat” (in pulses) and thus relaxes the belt (minimum tension and parking point). The belt retreat value defines the number of pulses that can be reached with this actuator. It can be set that when the RIG is switched “online” the belt is automatically homed. When the RIG is switched “offline”, the belt automatically retracts by the number of pulses set in the park position, thereby tensioning the belt. The values that are sent to the belt are 0 to 0xFFFFFF (max. tension) The direct mode is used for the belt, see next chapter.

If position mode is used, the belt can only be connected to servo 7 and is used there in direct mode with 32bit. Please note.

10. USB interface for motion data

The controller contains an FTDI FT232 RL chip or Ch340 which is installed via USB.

From FW 3.20 the “Flashport” can also be used for all USB functions.

The drivers required for this are usually installed automatically under Windows. these can but also about:

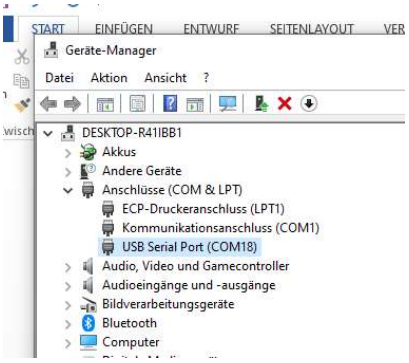
FTDI:

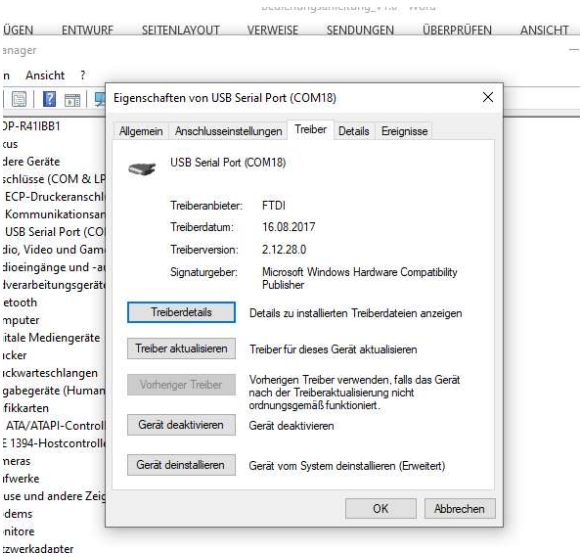
https://www.ftdichip.com/Drivers/VCP.htm or

CH340:

https://www.makershop.de/ch340-341-usb-installieren/

downloaded and installed.





11. Data transfer

Data transfer is carried out via the standard USB interface. There are 2 modes to transfer the data

The data packet string now is 28 bytes long and includes additional spare motion data slots for up to 8axis

The serial interface can be configured as follows

Data output	binary
speed	250,000 bauds
Databits	8 bits
Stop Bits	1
Parity	None

11.1. Direct fashion

This is the direct mode in this mode all actuators are controlled individually. The actuator position is sent bit-coded. The resolution is 24bit i.e. 3 bytes.

This means the following:

Value = 0x00000 (Decimal =0) Actuator at the bottom

Value = 0x7FFFF (Decimal = 8388607) Actuator in the middle

Value = 0xFFFFFF (Decimal= 16777215) Actuator at the top

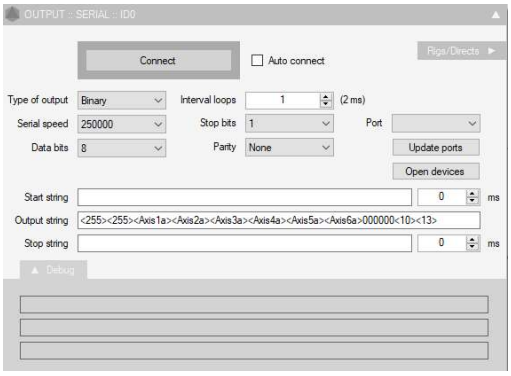
This means that 1,677,7215 positions can be approached per actuator.

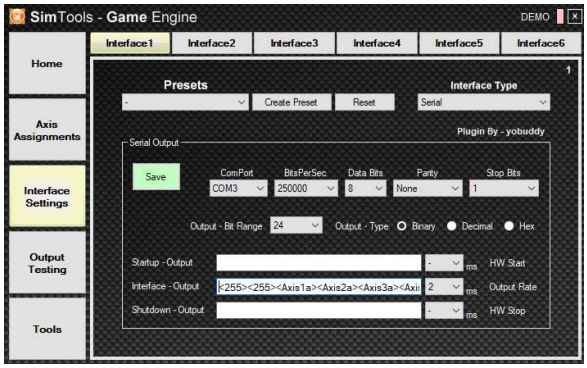
The ID is byte values 0xFF + 0xFF Each Axis is 24 bit wide. (3byte) LF+CR is required in the end (0x0A + 0x0D)

ID AXIS1 AXIS2 AXIS3 AXIS4 AXIS5 AXIS6 AXIS7 AXIS8 LF/CR

Data format

2 start bytes	<255><255>
24 bytes of data	3 bytes (24bit) per channel/actuator 8 channels
2 stop bytes	<10><13>





Danger:

To avoid errors, the following instructions for movers/Simtools.

Characters are entered for each unused actuator so that the string is always the same length (28bytes).

For example, using only one actuator <wind> 21 characters must be sent before/after

<255><255>aaabbbccc<Wind>eeefffggghhh<10><13>

For the actuator <Wind> 24bit must be set in the Rig module or in the Direct module so that this value is also 3bytes long.

11.2. Position Mode

In this mode, positions are sent to the controller in mm for the translational movements (Surge, Sway, Heave) and the angles in degrees for the rotating axes (Yaw, Roll, Pitch)

Floating point numbers are used for this (from Mover 3.4.2). The so-called floats are 32 bits in size and are sent in 4 bytes. This makes the string 32bytes long.

Sway Surge Heave as 4 byte float in mm

Yaw roll pitch as 4byte float in degrees

2 start bytes <252> <252>

8 x 4 byte float

2 stop bytes <10><13>

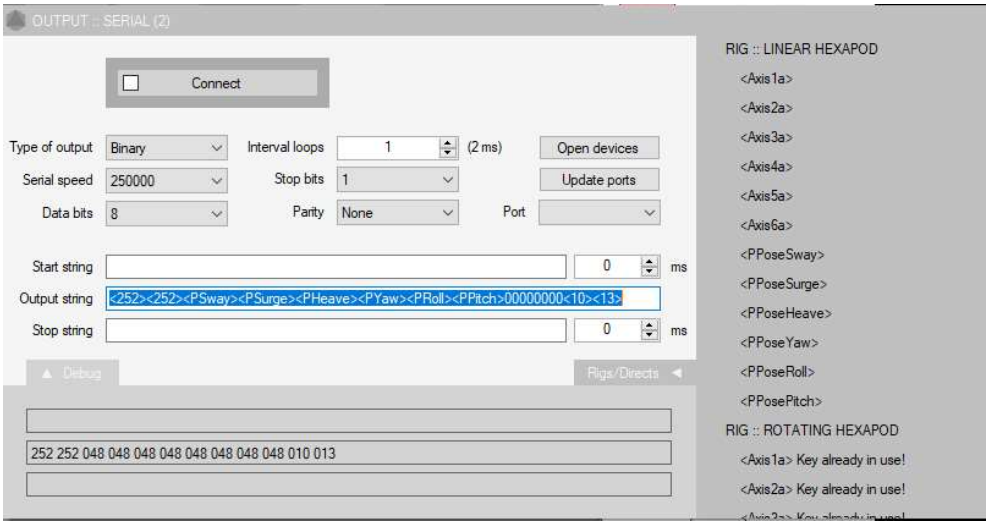
Here too, missing values must be filled in with 4 characters per axis, so that the string is always 36 bytes long.

Example for movers:

<252><252><PSway><PSurge><PHeave><PYaw><PRoll><PPitch>00000000<10><13>

The values of the axes are now 4bytes (32bits) wide, including the fill values for axes 7 and 8, so a total of 36 bytes are transferred.

Settings in the kinematics menu for scaling are no longer required.



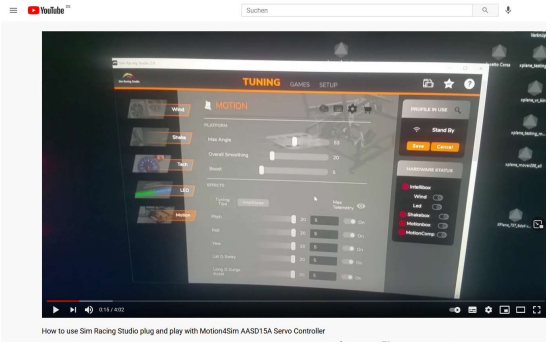
11.3. SRS mode –Sim Racing Studio

To activate, see Setup Menu

There is another mode, this mode provides an interface to automatically connect to SRS - Sim Racing Studio. It works “Plug & Play”. The data transmitted by SRS is expanded from 16 bit to 24 bit.

Video explanation:

https://www.youtube.com/watch?v=BZgPRXhlQf8&t=15s



12. LED display

Meaning of the ads

Green LED shines	Controller standby position or online and receives data
Red LED shines	Controller offline, homeing, parking position
Green flashes and red lights up	Movement to standby or park position
Red flashes irregularly	Spike filter active
Red flashes regularly	Emergency stop mode active

13. Factory reset

Sometimes it is necessary to reset the data in the controller memory to its original state. To do this, use the following instructions

1. Restart the controller
2. Wait until the message “push encoder for homing...” appears on the display
3. Then turn the encoder to the left to open further menu entries

4. Turn until “factory reset” appears in the display. Press the encoder to do this.

From firmware 1.41 there are 2 options for a factory reset. The settings are adapted as a basis for linear and rotating actuators.
The controller restarts itself after execution.

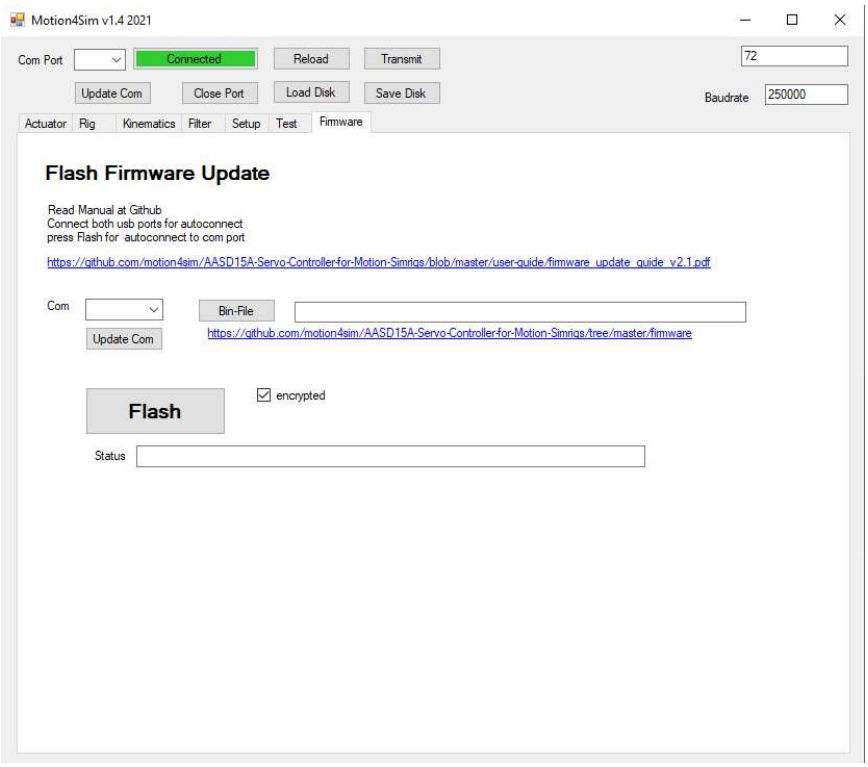
5. The data has been reset. The controller restarts.

14. Firmware update

Regular firmware updates are delivered if necessary. You can find these at

https://github.com/motion4sim/AASD15A-Servo-Controller-for-Motion-Simrigs/tree/master/firmware

Download it and install it on the controller according to the separate instructions. It's very easy.

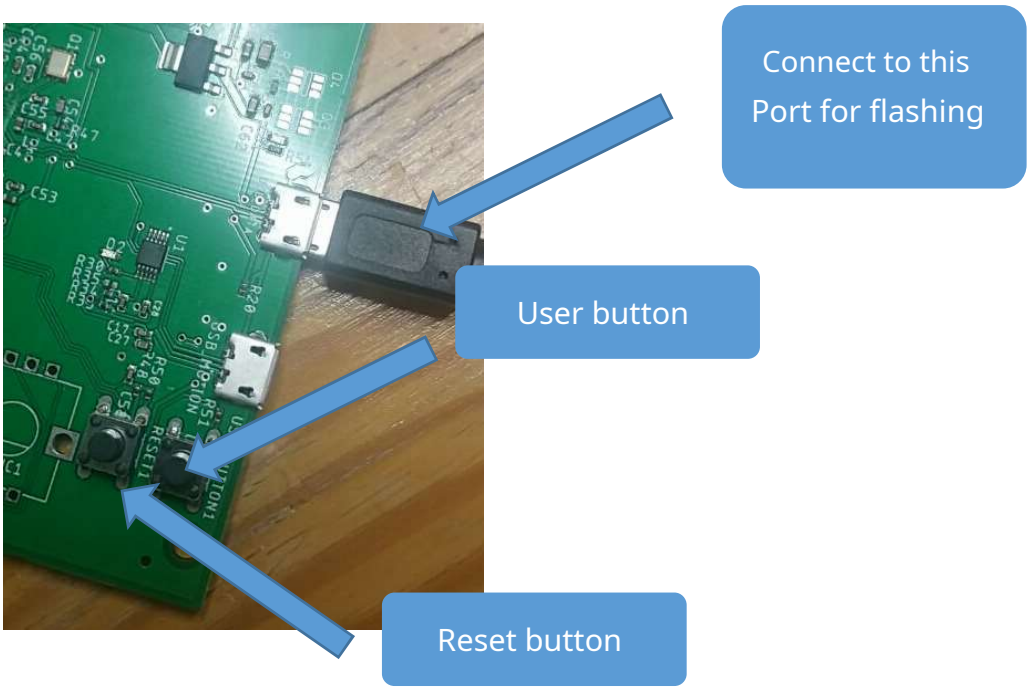


Video instructions:

https://www.youtube.com/watch?v=sPiSgAtpf4k&t=181s

14.1. Bootloader mode

From FW version 3.1 you can easily activate the bootloader mode via the setup menu.



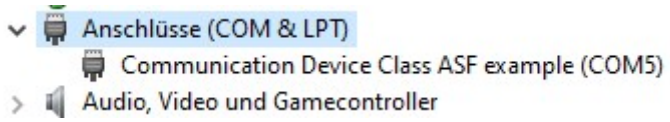
There are 3 ways to get into bootloader mode.

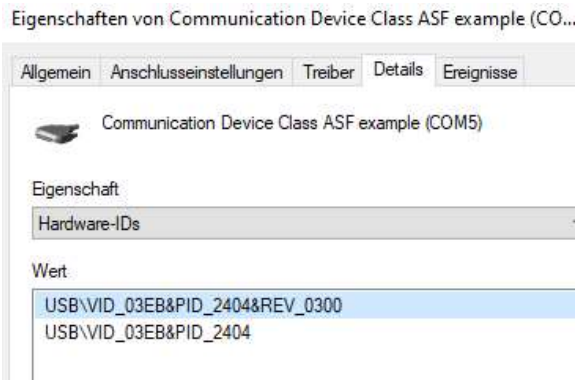
Variant 1. (supported from FW3.1)

1. Connect the controller to the computer using a USB cable as shown in the picture above.
2. The controller starts and navigate to the “Enter Bootloader” option as described in the “SETUP” menu and execute this by pressing the encoder twice.
3. The controller starts in bootloader mode
4. Make sure you have the correct USB port

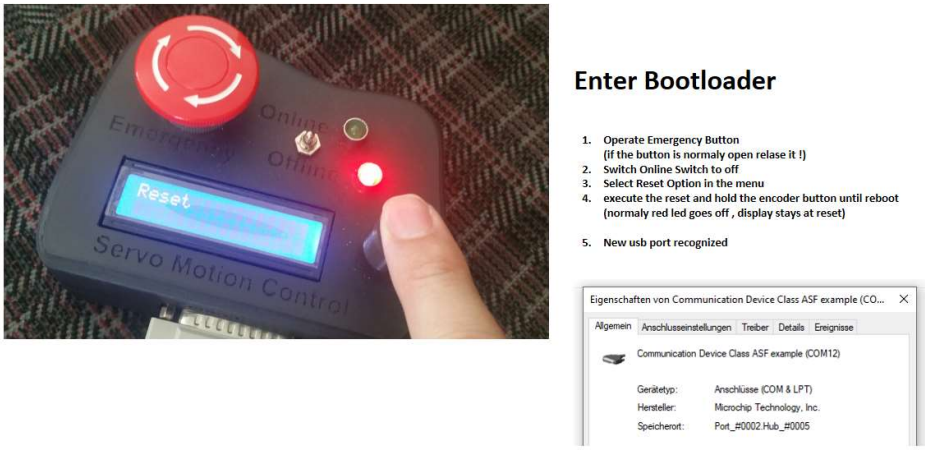
Variant 2.

5. Connect the controller to the computer using a USB cable as shown in the picture above.
6. Press and hold the user button
7. Briefly press the reset button
8. A new USB device should appear in the device manager and the controller is in bootloader mode.





Variant 3. With handheld



The following steps are necessary with the handheld device

1. Press the emergency stop button
2. Switch the online switch to off
3. Navigate to “Reset” in the menu
4. Press the encoder and hold it down (3 s) until a new USB device appears in the device manager. (see variant 1)
5. If that doesn't work, release the emergency stop button and repeat the procedure
6. The controller is in bootloader mode

14.2. Flashing process

1. If you haven't already done so, install the Dashboard app

2. Navigate to the “Firmware” tab

3. Download the latest firmwarehttps://github.com/motion4sim/AASD15A-Servo-Controller-for-Motion-Simrigs/tree/master/firmware

4. Select this file

5. Set the checkbox “encrypted”

6. Do not choose a Com port. Make sure that all unnecessary USB-serial devices unplugged and no Bluetooth serial is activated.

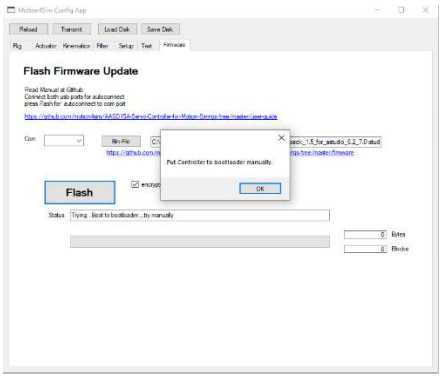
7. The Com port is determined automatically. Should it

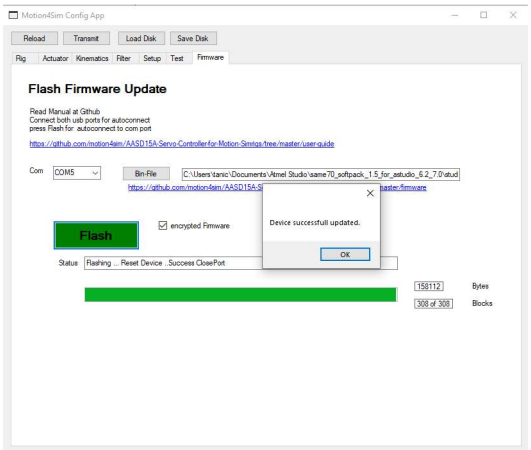
   If you have problems select the Com port manually.

8. Press the Flash button

9. Confirm that the controller is in bootloader mode.

10. The firmware is transferred and the controller restarts.





15. Dashboard app

There is now a new smart remote APP for easy editing of the controller configuration. This allows the most important data to be downloaded from the controller, modified, backed up and written back. It is also possible to test actuators and flash the firmware. Additionally, a handheld device is emulated in software.

Download from Gitub:

https://github.com/motion4sim/AASD15A-Servo-Controller-for-Motion-Simrigs/tree/master/Dashboard

In some cases, the app turns out to be unstable under Windows 11. Please use the flash port as a USB port. This is possible from FW 3.2

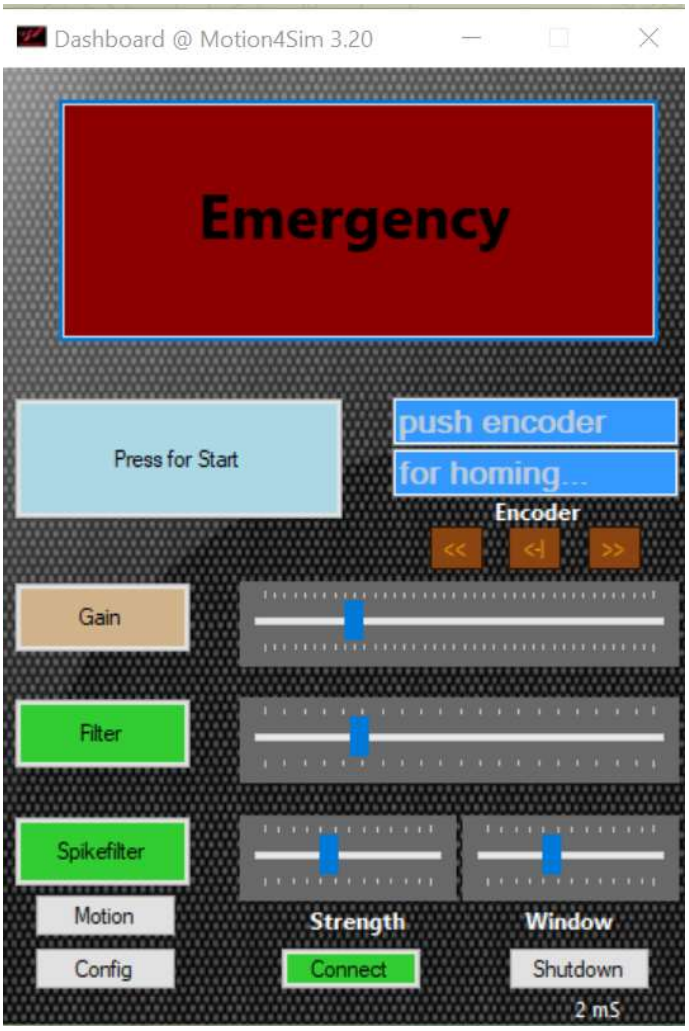
Start the Dashboard app, it will automatically connect to the controller. If a handheld is installed, the app and handheld device can be operated in parallel.

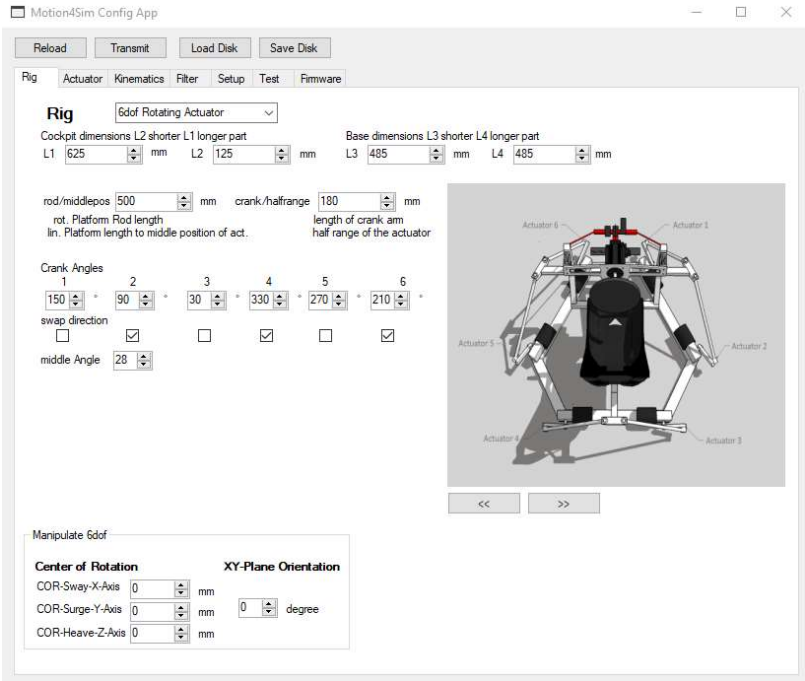
Only the “emergency stop” switch on the handheld device has priority over the app.

Gain and filters can be easily operated via the app.

How it works explained on YouTube.

https://www.youtube.com/watch?v=hUnLz29yCMw&t=469s

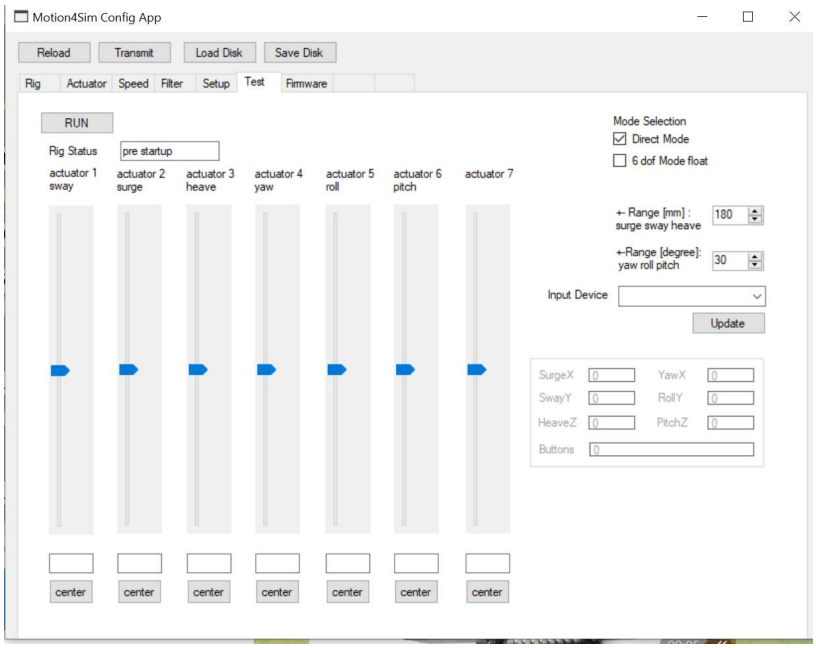

Dashboard @ Motion4Sim 3.20
Emergency
Press for Start
push encoder
for homing...
Encoder
<<
<-|
>>
Gain
Filter
Spikefilter
Motion
Config
Strength
Window
Connect
Shutdown
2 mS



15.1. Actuator test functions

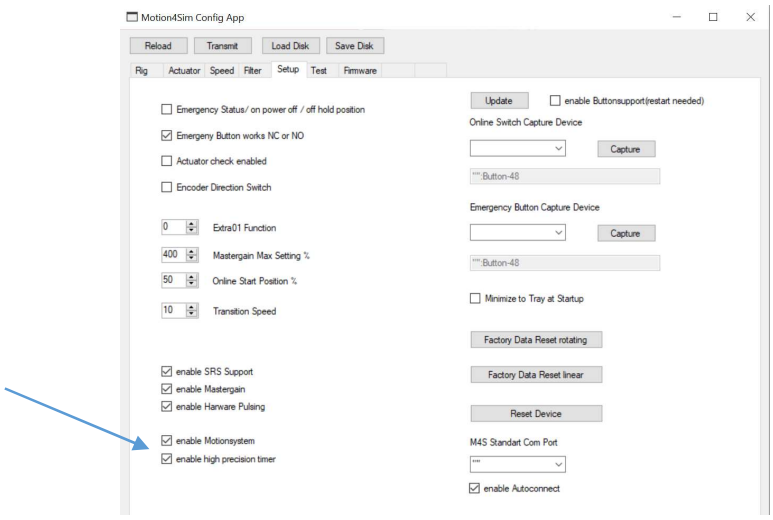
The “test page” allows you to test the actuators. If the controller is “connected” you can switch to the “Test” page tab and press the “RUN” button. The app begins to send the set values for the actuators. The actuators should be calibrated and online.

The specified values for Range are used for the “6dof Mode float”.



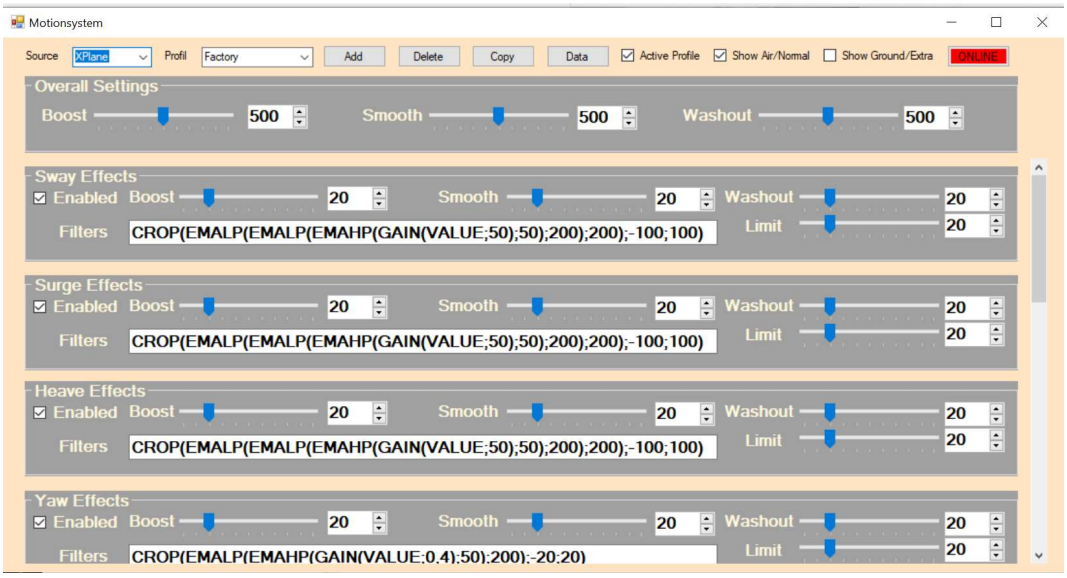
15.2 Motion Cueing System

The motion system must be activated in the config settings. There you can also activate the high precision timer so that the filters are calculated stably with 2ms timing. But it costs more CPU



You can then use the system using the Motion button on the dashboard.

More sources/games/sim will be added.



Functionality:

Attention, the RIG is operated in “Position Mode”. To do this, all geometry settings of the RIG must be stored correctly in the configuration. This is important otherwise wild unpredictable movements can occur. To do this, also test the RIG with the Test tab in the Config section in “6dof float mode”.

Games and Sims are automatically searched for. If a simulation is running, the profile for the rig and the game is automatically loaded. The profile marked “Active Profile” is always loaded. There are 2 profiles (flight simulations) for some Sources. These can be switched in the view with “Show Air/Normal” and “Show Ground/Extra”. Only 2 “Normal” is used for racing games. The transition for the game between air and ground is carried out automatically.

The profiles can be edited using the button, Create -> Add, delete and copy.

Multiple profiles for RIG and games can be created. If profiles for other RIGs are to be edited, the controller must be set for this RIG. Only one can be marked as “Active”. The name of the profile can be changed in the Profile dropdown.

“Data” shows a window with the data possible from the simulation and the timings as well as the calculated values.

The data will only be sent when the RIG is ONLINE. Transitions are built in for all possible scenarios to enable smooth switching between profiles.

Please test the system in peace and start with a gain of only 10% until you are sure that the motion works for your set profile.

When a simulation is active, no other profiles can be edited.

Filter:

The following filters are available

CROP(VALUE;Below Value; Upper Value) The filter cuts off values that are smaller or larger

(the values in mm or degrees)

EMALP(VALUE;filter value) calculates an EMALP filter with the specified value (lowpass)

EMAHP(VALUE;filter value) calculates an EMAHP filter with specified value (high pass)

GAIN(VALUE;Factor) multiplies the value by the factor

MUL(VALUE;Factor) multiplies the value by the factor

LOGISITC(VALUE;Limit,Slope) calculates a logistics filter with the limit value limit (in mm) and the slope

REMAP(VALUE; Input_low;Input_high;Output_low;Output_high) this filter scales / remaps from input to output

ROLLOVER(VALUE;Grad) this filter is for a correct transition for loopings etc.

The filters work like in Mover, so you can develop the filters in Mover and then store them in the profile.

The filters can be nested one inside the other and are then calculated one after the other. To do this, enter the upstream filter for the value “VALUE”.

e.g.

CROP(EMALP(EMALP(EMAHP(GAIN(VALUE;1);100);100);100);-100;100)

The syntax is checked automatically. If this is incorrect the text will be RED.

If you leave the text field with the wrong filter, the old one is loaded.

For each effect, a filter combination that corresponds to classic motion cueing is loaded with “Add”.

For SURGE SWAY and HEAVE these are a combination of high pass and 2x low pass filters

CROP(EMALP(EMALP(EMAHP(GAIN(VALUE;1);100);100);100);-100;100)

And for the rotations YAW ROLL and PITCH from low pass high pass

CROP(EMALP(EMAHP(GAIN(VALUE;1);100);100);-5;5)

3x lowpass are used for tilt coordination

CROP(EMALP(EMALP(EMALP(GAIN(VALUE;1);100);100);100);-5;5)

All of these functions are extended by a crop.

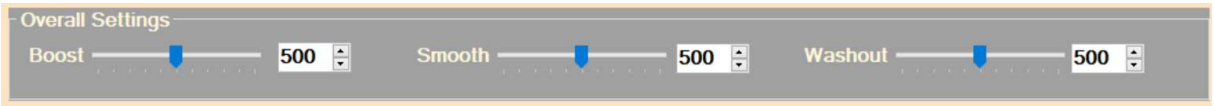
Overall settings:

The sliders change the values in all effects.

Boost sets the maximum value for GAIN

Smooth sets the maximum value for EMALP

Washout sets the maximum value for EMAHP

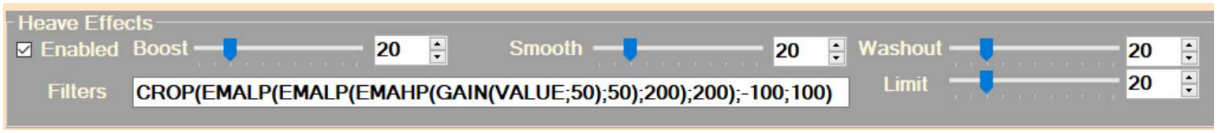


Effects:

There is an effect for each DOF for Sway, Surge and Heave, the accelerations with G are used. The speeds are used for yaw, roll and pitch. There are 2 other effects

The so-called tilt coordination. Sway is calculated in roll and surge in pitch. This means that classic motion cueing can be depicted.

The sliders scale the value set in Overall in % to the respective filter. With “Enabled” the effect can be activated or deactivated.



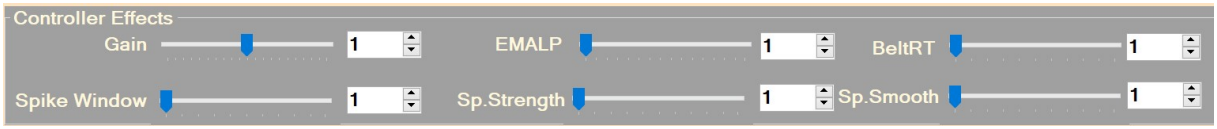
By editing the value in the filter line, the scaling can be changed to a different value.

The Limit Slider manipulates the crop filter to set a maximum value.

All values are saved in the profile when you exit the window.

Controller Effects:

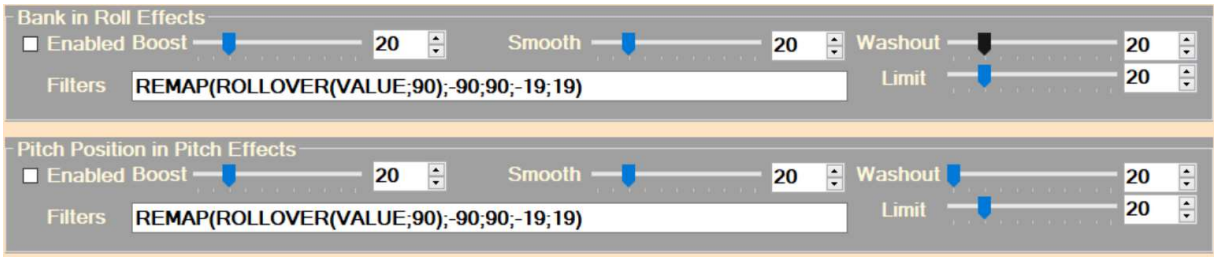
The controller settings for each profile can be set here. These are only sent to the controller once when the profile is loaded. These settings are not saved in the controller, so they will be lost when you restart.



Bank in Roll and Pitch Position Effects:

Here the values of the position of roll and pitch are calculated in roll and pitch. (In degrees) This approach is not entirely correct and is intended more for RIGs with large actuators as a fun effect.

The picture below shows how an aircraft roll is mapped from a maximum of 90 degrees to a 19 degree rig position.



Work is underway to expand the system to include belt effects and motion cancellation.

16. Special Serial Commands

Not only motion data can be sent via a serial terminal as described in Chapter 11. Additional control commands and data queries are described below. This makes it possible to control the controller.

Send:

Syntax:

1. Start bytes	2 bytes <f9><f9>	hex or <249><249> decimal	
2.Command	2byte <0-ffff>	hexor	<0..65535>
3. Value	2byte <0-ffff>	hex or	<0..65535>
4. Stop bytes	2byte <0a><0d> hex or <10><13>		

Total 8 bytes

Reception:

Syntax:

1. Start bytes	2 bytes <f9><f9>	hex or <249><249> decimal	
2.Command	2byte <0-ffff>	hexor	<0..65535>
3. Value	2byte <0-ffff>	hex or	<0..65535>
4. Stop bytes	2byte <0a><0d> hex or <10><13>		

Total 8 bytes

The command part always responds with the requested value.

Commands

send	f9f9 0000 0000 0a0d	(RIG status)
returns	f9f9 0000 XXXX 0a0d	
	xxxx is	
	preparing	0x0000
	parking	0x0001 0x0030
	emergency	0x0002 0x0004
	offline	0x0008 0x0010 0x0020
	ready	0x0031
	standby	0x0032
	wait uart	0x0033 0x0034
	on-line	0x0035
send	f9f9 0100 0000 0a0d	(portid)
returns	motionusbport	= f9f9010001000a0d
	native usb port	= f9f9010002000a0d
send	f9f9 0200 0000 0a0d	(fw_version)
returns	f9f9 0002 0002 0a0d	not implemented ready

send	f9f9 0300 0000 0a0d	(request bootloader mode)
returns	reboots to bootloader mode	

send	f9f9 0400 0000 0a0d	(enables softoverride for handheld app)
returns	f9f9 0004 0002 0a0d	

send	f9f9 0500 0000 0a0d (enable emergency)	
returns	f9f9 0005 00xx 0a0d	(emergenc stat 0 or 1)

send	f9f9 0600 0000 0a0d enable online	
returns	f9f9 0006 00xx 0a0d	(online stat 0 or 1)

send	f9f9 0800 00xx 0a0d 0 = factory reset linear 1= rotary
returns	f9f9 0800 0001 0a0d reboots

send	f9f9 0900 0000 0a0d reboot
returns	

send	f9f9 1000 0000 0a0d	(request status actuators)
returns	f9f9 1000 00xx 0a0d	
	xx is	

0 ='x'; //not connected
1 ='C'; Calibrate
2 ='P'; preparing
4 ='!'; moving
8th='R'; ready

send	f9f9 3000 0000 0a0d	request master gain
return	f9f9 0030 00XX 0a0d	XX 1-40*10 in % 10 is 100%

send	f9f9 3100 00XX 0a0d	set master gain 1-40)
return	f9f9 0031 0002 0a0d	is set

send	f9f9 4000 XXXX 0a0d	set filter to XX or read if xx=0
returns	f9f9 4000 XXXX 0a0d	filter size (emalp)

send	f9f9 4100 XXXX 0a0d	enable/disable filter XXXX = 1 or 0
returns	filterstat if XXXX=2	
send	f9f9 4200 XXXX 0a0d	enable/disable spike filter XXXX = 1 or 0
returns	f9f9 0042 XXXX 0a0d	filterstat if XXXX=2
send	f9f9 4300 XXXX 0a0d	set spikefilter to XXXX or read if XXXX=0
returns	f9f9 4300 XXXX 0a0d	returns spike filter size
send	f9f9 4400 XXXX 0a0d	set spikefilterwindows to XXXX or read if XXXX=0
returns	f9f9 0044 XXXX 0a0d	returns spike filter size
send	f9f9 5000 0000 0a0d	homing
send	f9f9 0060 0000 0a0d	send display frame back, ask for details
	38 bytes f9f9 32bytes Char of display 0a0d	
send	f9f9 0070 0000 0a0d	encoder button pressed
send	f9f9 0071 0000 0a0d	encoder left pressed
send	f9f9 0072 0000 0a0d	encoder rigth pressed
send	f9f9 0080 0000 0a0d	go online after homing without input

Thank you for using Motion4Sim

Have fun

Motion4SIM

Update follows.