AMC-AASD15A servo controller Manual
for Simtools motion software.

This manual is written for firmware v2.09

AMC-AASD15A Interface information

The AMC-AASD15A servo controller allows seamless and fast interface between the PC and the MDBOX servo drives. Using the AMC-AASD15A controller you can interface your linear servomotors to Simtools, X-sim and Ian's 6DOF BFF motion software. The connection to PC is a simple USB connection and the connection to the AASD-15A drives is via straight DB25 cables, one for each drive.

In the LCD menu of the AMC-AASD15A, you can set the following to match your simulator platform (Other settings are not so much important):

- Number of motors → **5axis** (for 4DOF + TL) or **4axis** (if just 4DOF)
- Auto-Park function → **4DOF +TL +Surge** (to disable park on TL axis)
- Actuator Stroke → set to **100mm** (for SFX100 DIY actuators) or **150mm** (for PT-actuator)
- Lead screw → set to **5mm/rev** (for 250mm/s speed SFX100 DIY actuators) or **10mm/rev** for 500mm/s actuators
- Motor Direction →Set to **Inline** (for SFX100 DIY actuators) or **Foldback** (for TL actuators)
- Gear reduction ration→Used mostly foldback actuators that may have different ratio belt drive.
- Platform Check →Blocks the use of the platform with motion data if any of the actuators is not in ready state.
- Pulse Frequency→by default is 200kHz, but some servo drives may require less like the Syntron HS that need 100Khz frequency for the pulses.

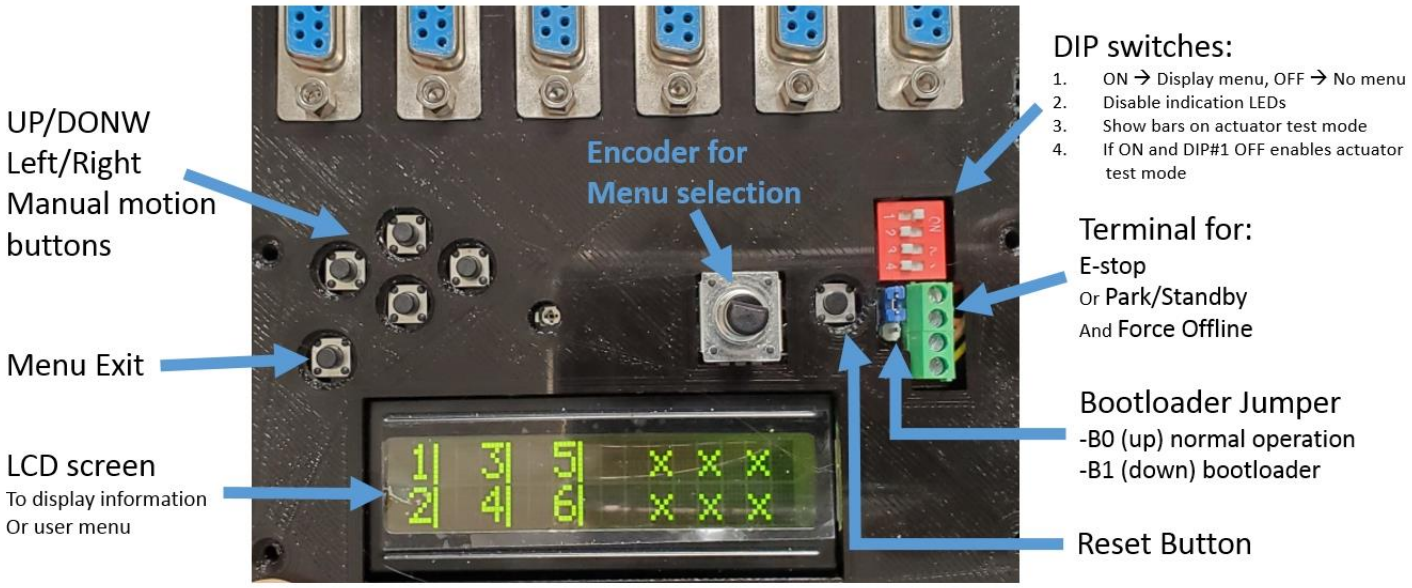
Additionally there are new functions on v2.09 firmware: Rolling Average Filter (smoothing filtering) on the servomotor, to decrease the motor step performance, remove little vibrations from that exist from the use of electronic gearing pulse method and add s-curve smoother accelerations/decelerations, and Spike filter to replace and large spikes in motion (non-linear motion cues) with temporary artificial slower motion until the motion levels relax. The Spike Filter is especially useful to protect the motion platform from large amplitude fast vibrations that translate to full speed swings to the expend of the whole stroke, that are usually generated from the motion software in case of a crash (maxing out force levels). It will also slow down the motors speed in case your run off road to minimize the effects of violent vibration if set to lower level.

The Rolling Average filter is located under the Pulse Freq submenu with title Filter Factor. A good value for minimal vibration and good response is "2-Semi-Hard-Filter". You can disable this the Filter Factor by setting it to "0". The Spike filter is located in a separate menu, where you can set the Level of the spike parameter (distance between two positions, in the scale of 0-65535 values), the Range parameter, which can increase the selection resolution "step" of the values in lower Spike Level values and the option to Enable-Disable the filter. A good level for the Spike filter for normal racing is around "1530".

~


At any point you can restore default parameters by holding the “Menu Exit” button and pressing Reset. Keep holding the “Menu Exit” button until you see the message “Restoring Defaults” appear on the LCD.

For detailed guide on setting up the parameters and options for 4DOF+TL on the AMC-AASD15A controller see this video: https://www.youtube.com/watch?v=HhyF4e7gGWU



When an AASD-15A servo is connected and powered on the AMC-AASD15A servo controller, it will perform calibration seeking for the home position (usually hard stop against the end of travel of the actuator).

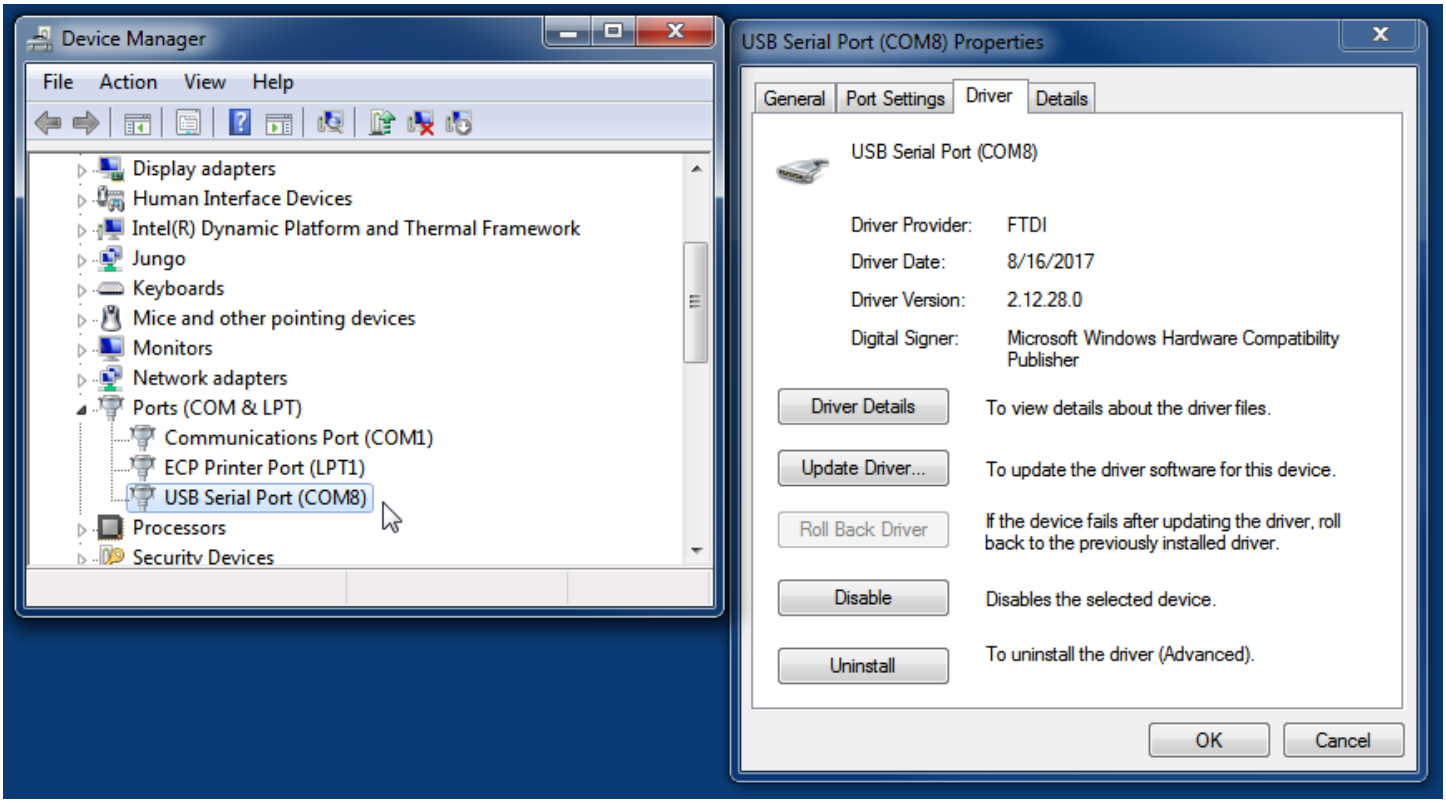
Once the calibration is complete, the motor will head to home position…

…or standby position if motion data from the PC are already present.

If there is no motion data from the PC, the controller will timeout and automatically park the actuator showing a P symbol for the actuator.

The USB Data connection requires FTDI driver that can be downloaded from the FTDI website:
http://www.ftdichip.com/Drivers/VCP.htm

The device appears in the PC Device manager as COM Serial interface device that then can be defined for use with Simtools or any other motion software that provides interface support for the AMC-AASD15A.



If your controller has older firmware you can visit the Github and get the latest firmware to update the controller.

https://github.com/tronicgr/AMC-AASD15A-Firmware

https://github.com/tronicgr/AMC-AASD15A-Firmware/tree/master/Latest_firmware

Make sure to download the bootloader utility:

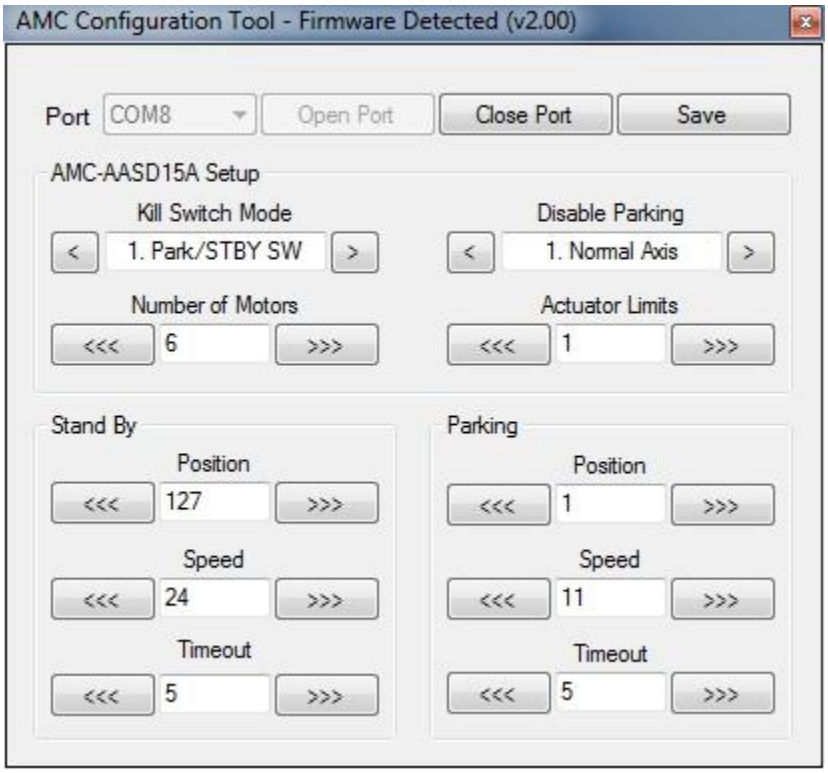
https://github.com/tronicgr/AMC-AASD15A-Firmware/blob/master/Latest_firmware/Firmware%20bootloader%20Utility%2052.zip

Firmware Update procedure video

https://www.youtube.com/watch?v=WkAm-MI0xbo

You can use the AMC config tool to access and modify the parameters in the AMC-AASD15A:

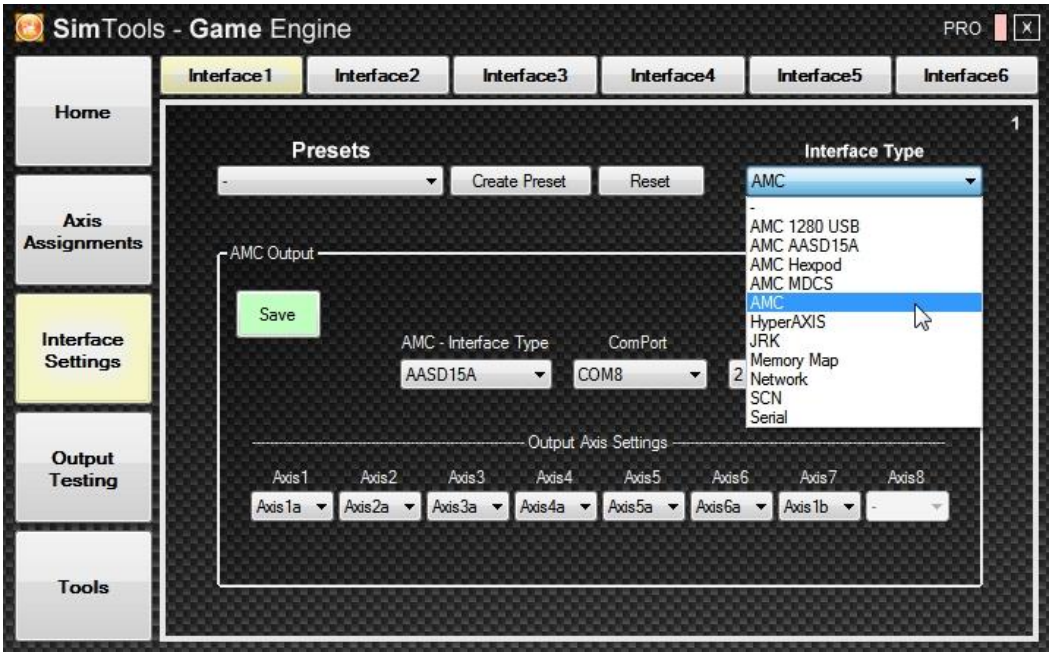
https://github.com/tronicgr/AMC-AASD15A-Firmware/blob/master/Simtools_interface_plugin/AMC_Config_Tool_1_1.zip



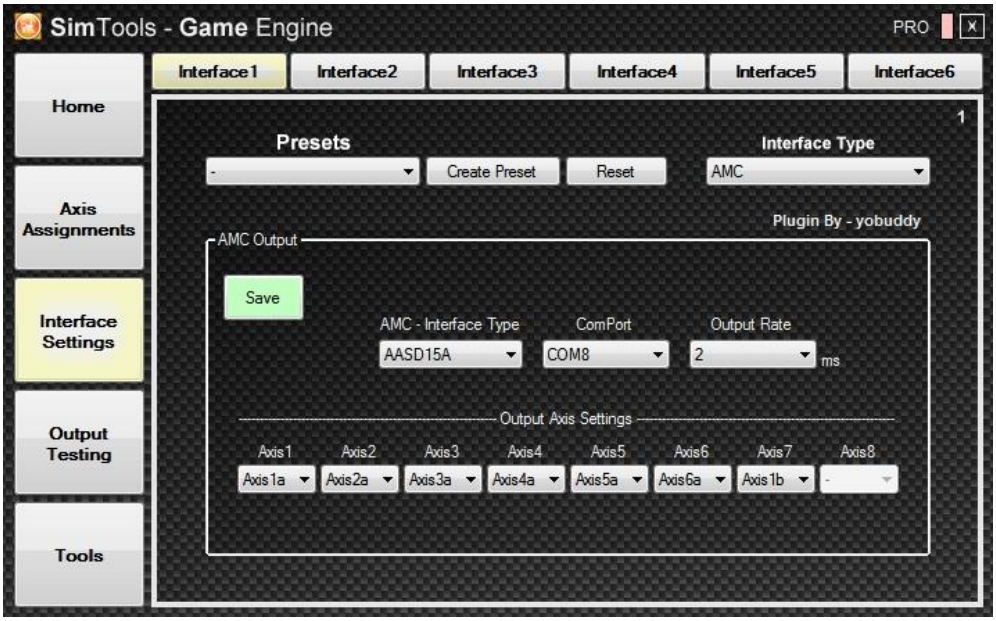
Software Setup

Simtools:

The Simtools v2.4 should already include the AMC interface plugin, if not you can find it on the Github and simply drag and drop the "AMC_InterfacePlugin.dll" into the Simtools PluginUpdater.
Start Simtools, you should see 8axis available now for the AMC1280USB interface plugin.



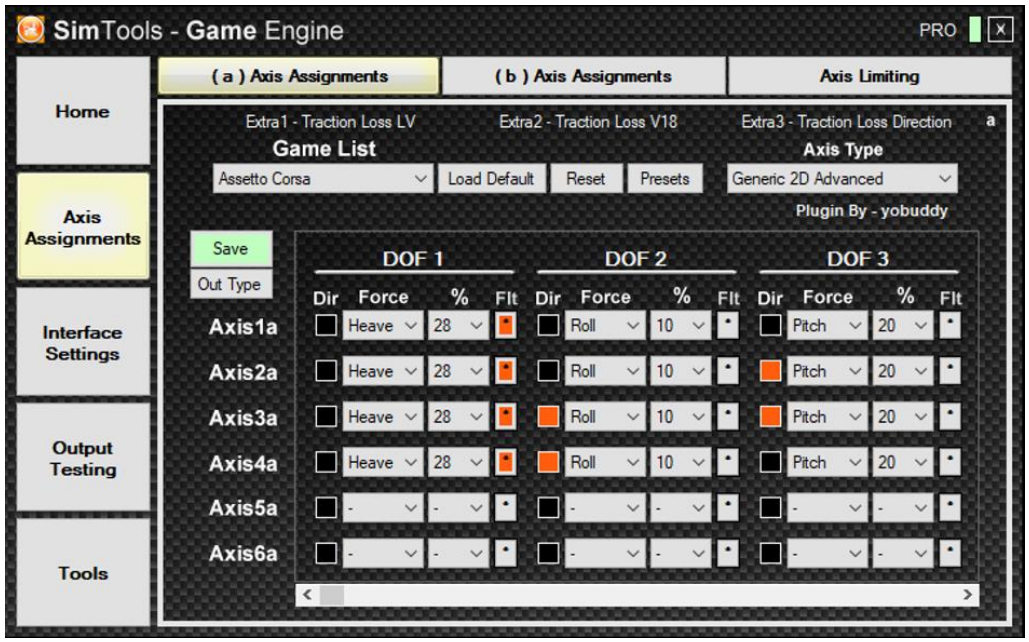
Interfacing the Simtools with direct axis is simple as seen on the below capture. It requires to select the AMC interface plugin, select the AASD15A interface type and select the COM Port that is assigned to the AMC-AASD15A in the PC device manager.

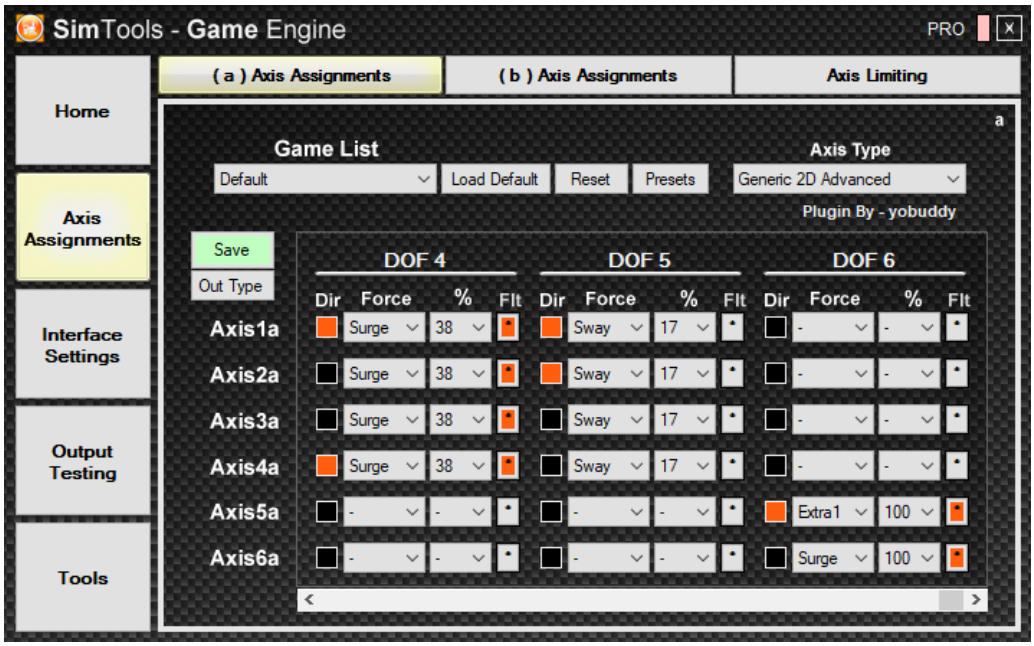


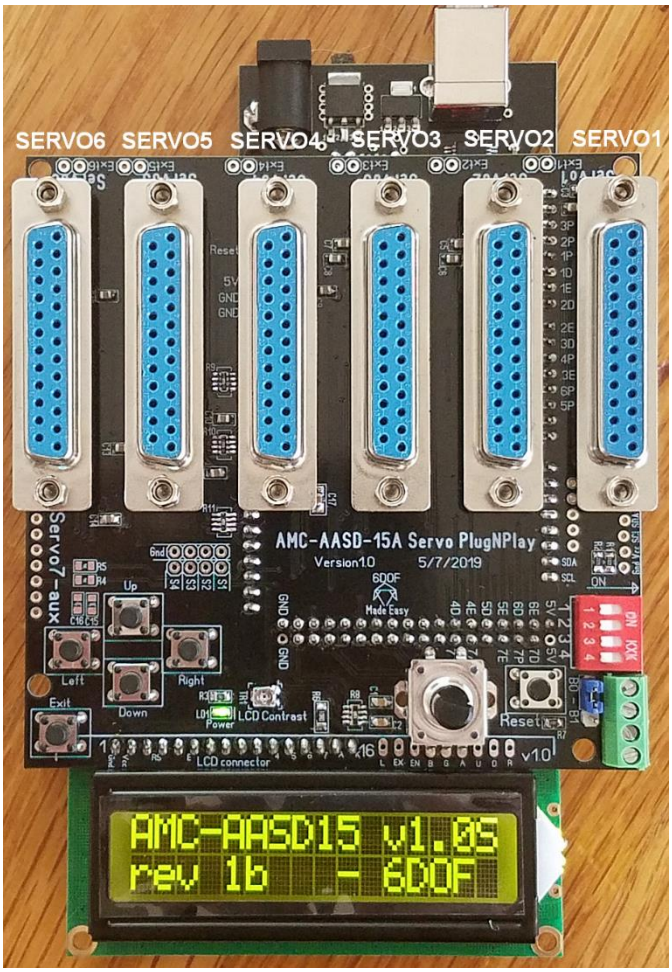
The axis assignments for each DOF provided is up to the use to mix and use as needed. The AMC-AASD15A can be configured to use any of the 3axis, 4axis or 5axis outputs.

To get the desired motion from the computer game to the actuators, you will have to create some profiles that mix the axis information from the game to the axis setup of the actuators. This can be done in the Axis Assignments section of the Game Engine of Simtools. If additional traction loss actuator is used, it can be assigned to Axis5a (extra1 for many games).

Setup example of the Axis assignments with various DOF (degrees of freedom) motion cues data inputs for combined motion. The axis5a on the example uses just the “Extra1” that is traction loss usually and axis6a holds the pure Surge value for the surge actuator:







Physically you will need to arrange the order of connection of each actuator to the AMC-AASD15A controller to correspond to correct order described to the Axis assignments of Simtools. For 5DOF platform the order of connection of each actuator 1-5 is:

Rear left - Servo 1 connector of AMC-AASD15A

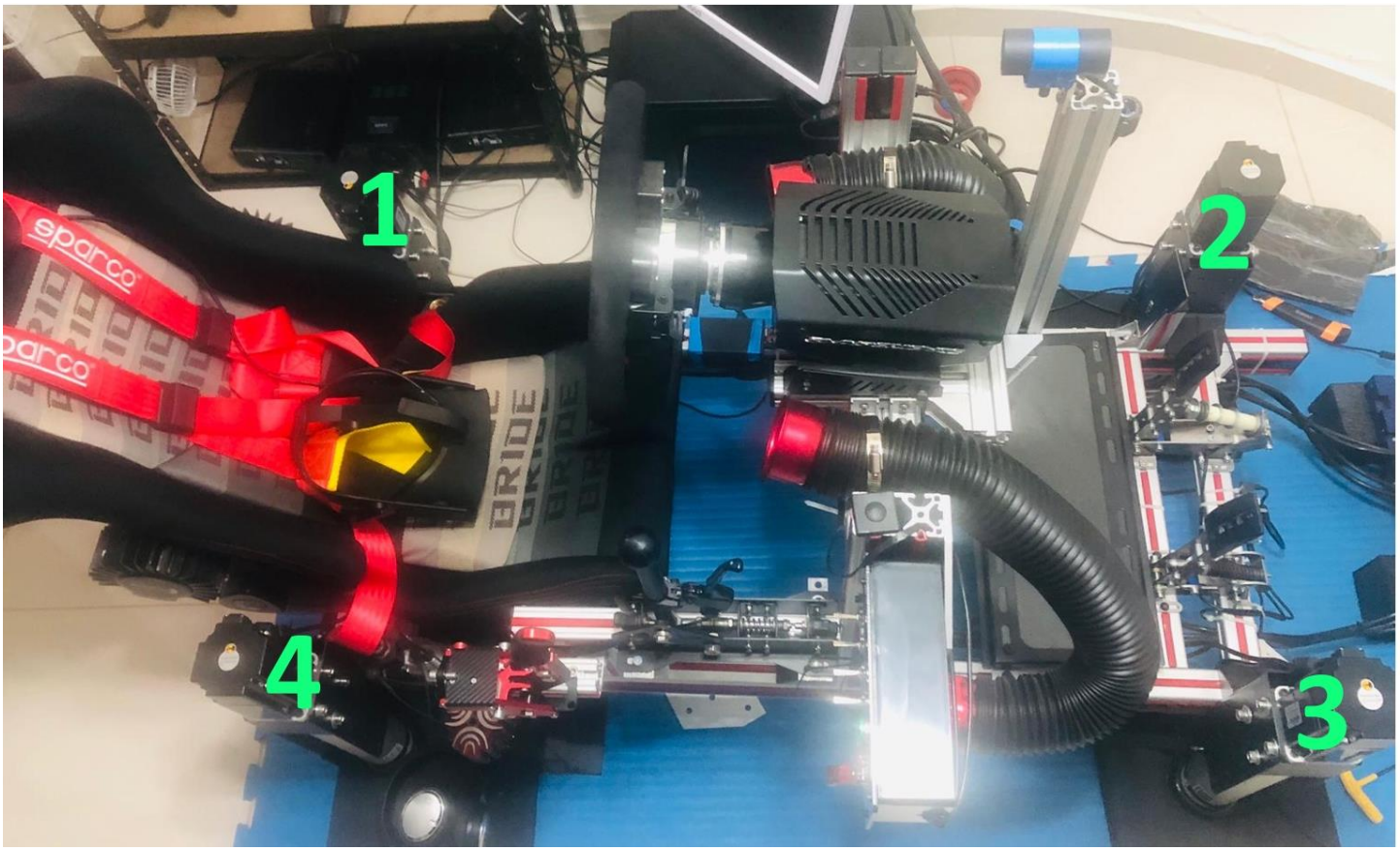
Front left - Servo 2 connector of AMC-AASD15A

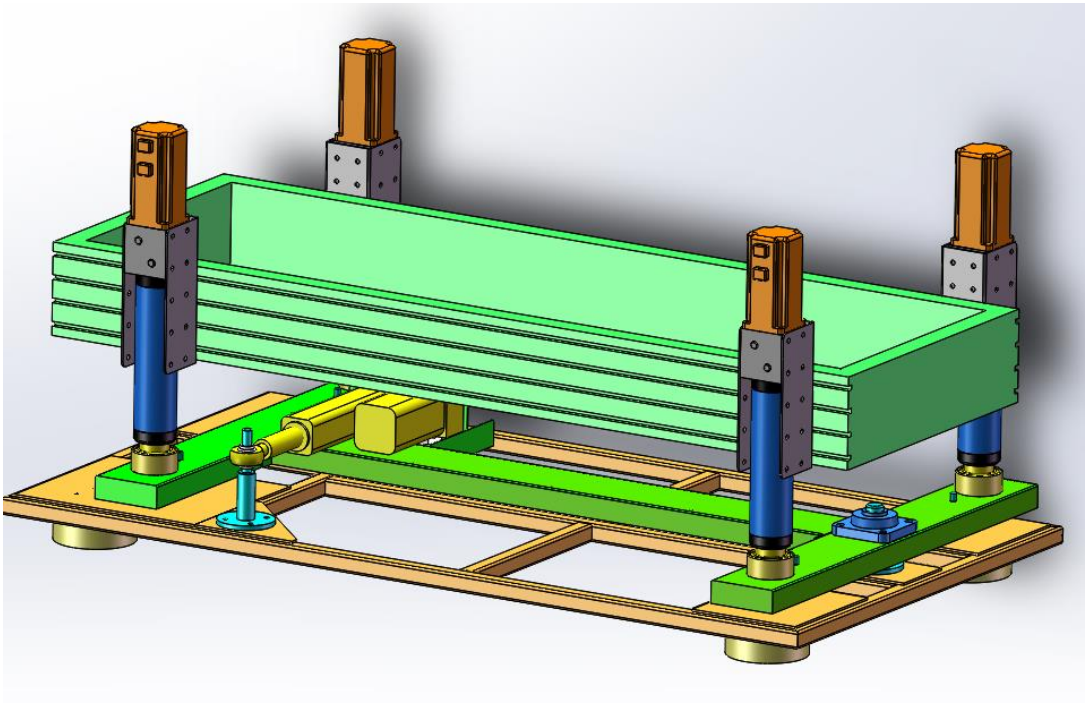
Front right - Servo 3 connector of AMC-AASD15A

Rear right - Servo 4 connector of AMC-AASD15A

Traction Loss - Servo 5 connector of AMC-AASD15A

Surge Actuator - Servo 6 connector of AMC-AASD15A

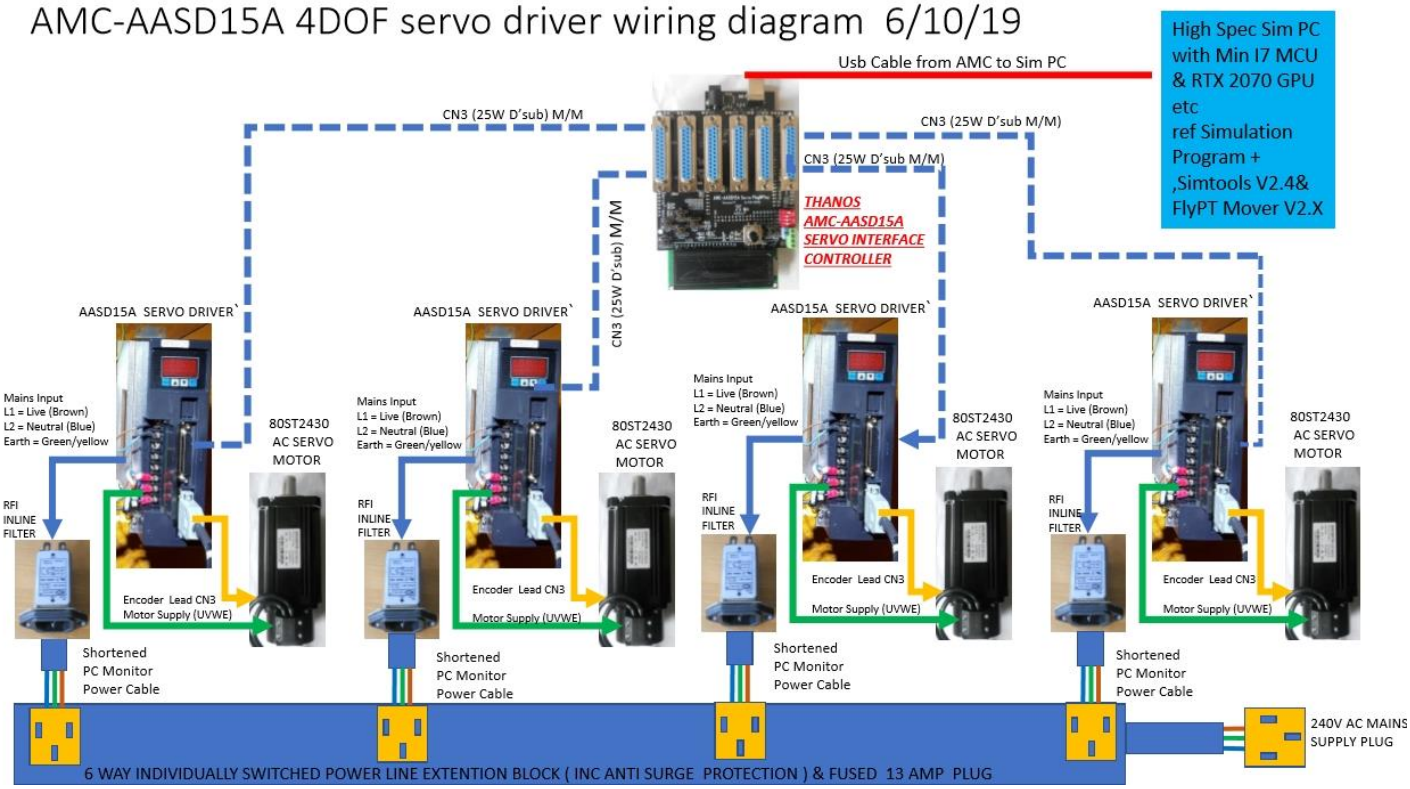




You can see the four actuators mounted vertically and the traction loss actuator (foldback) mounted horizontally

The wiring of the servo drives is not too complicated if you do some basic cable managing

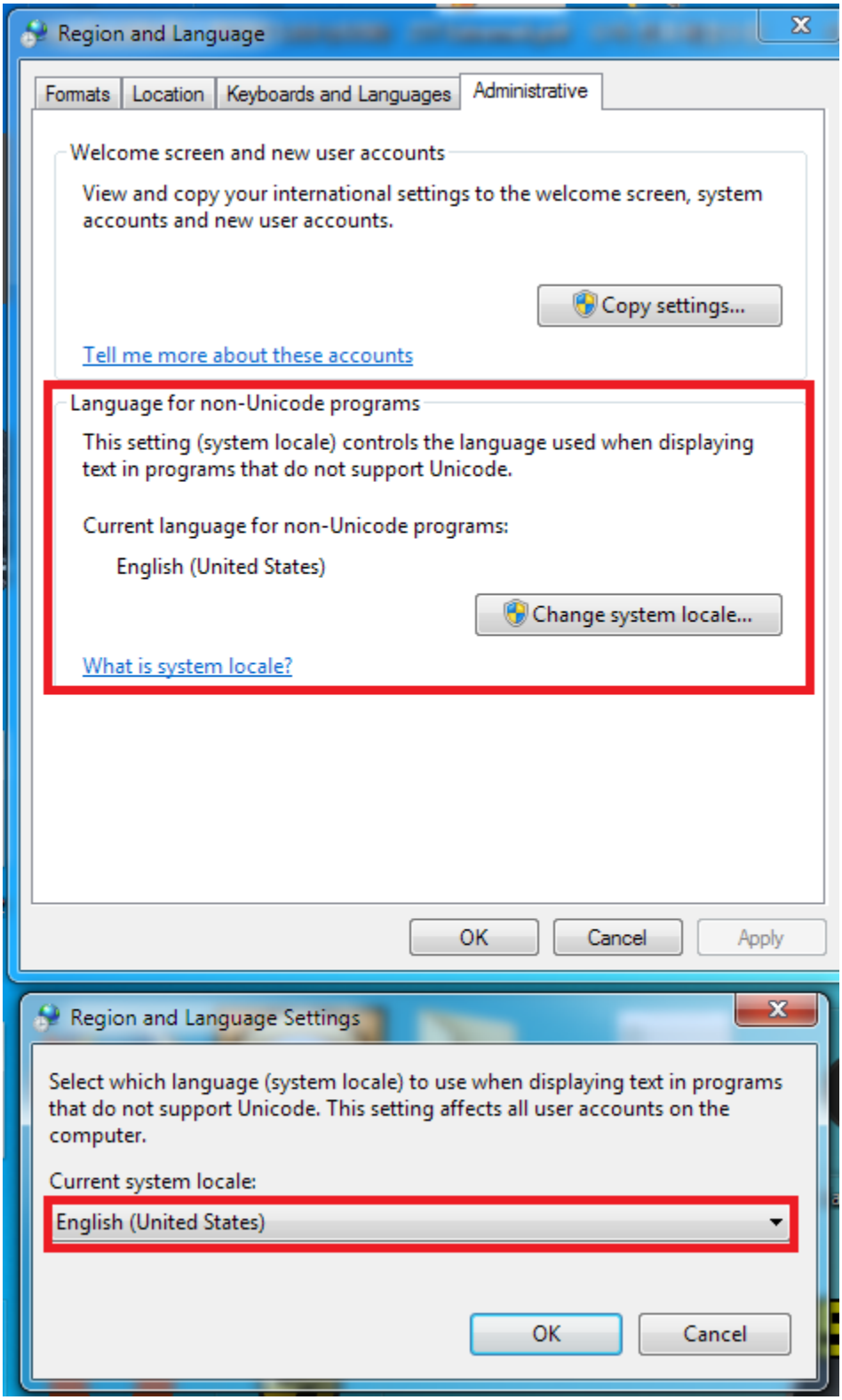
AMC-AASD15A 4DOF servo driver wiring diagram 6/10/19



A better way would be using a small rack to enclose the AASD-15 drives and arrange the wiring. Also including some filtered power strips and chokes on each cable. Such racks often have their own fans that allow cooling of the servo drives while on operation and block much of the noise.

Troubleshooting for Simtools:

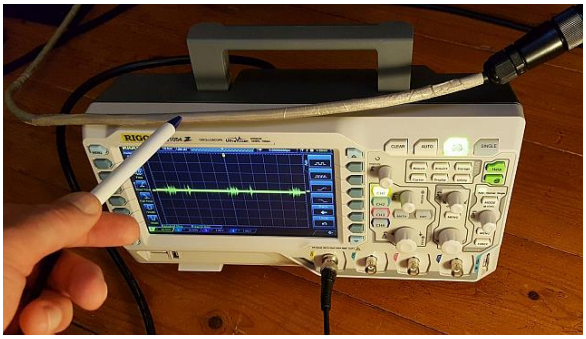
If no there is no motion when you test manually the sliders in Simtools, please change the computer Region and Language settings as below:

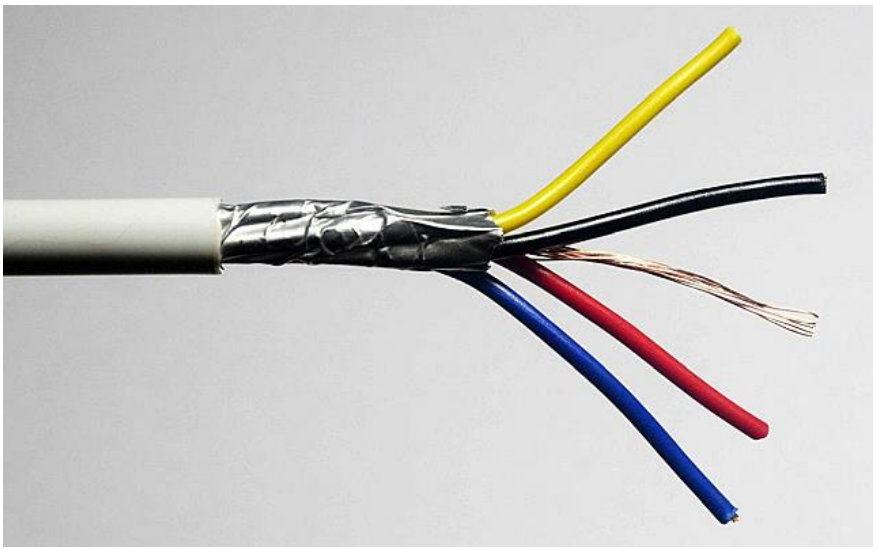


Dealing with EMI issues from the motor cables

EMI and EMC considerations

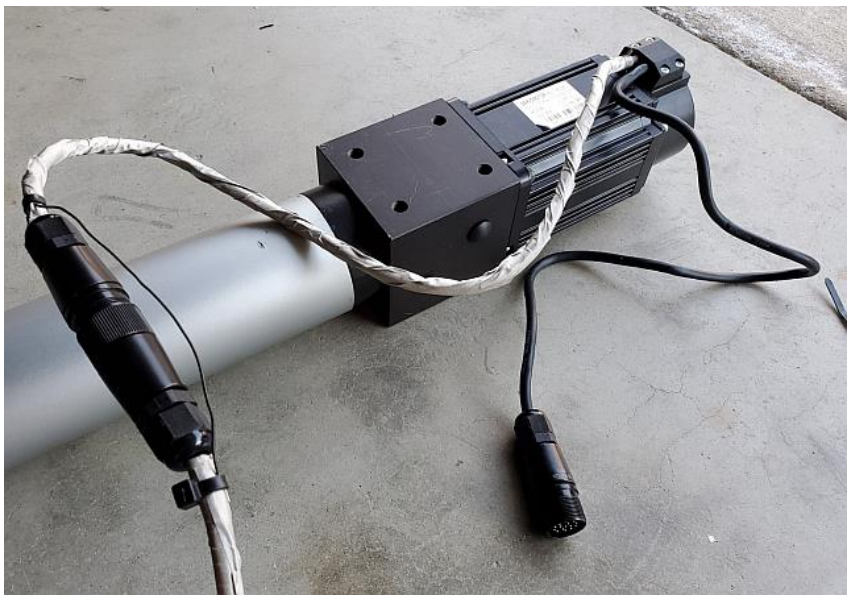
Most of the servomotors are delivered from the factories without proper shielded cable that leaves all other nearby equipment subject to severe EMI interferences. This can be really bad for devices that are really sensitive like many VR equipment that are usually positioned right on top of these interferences. The long unshielded cables of the servomotors can act as antennas for transmitting these EMI interferences several feet away. The source of these is the servo drive itself that generates a 10khz PWM signal on the three phase AC power signals that is used to power and position the servomotors. These signals swing up to 110v in many cases can affect nearby low voltage devices (5v or 3.3v devices). The best way to solve this is to cover these power cables with conductive tape from the servo drive all the way up to the motor. See example capture of the noise on the same cable when it's measured on covered and an uncovered region of the cable:

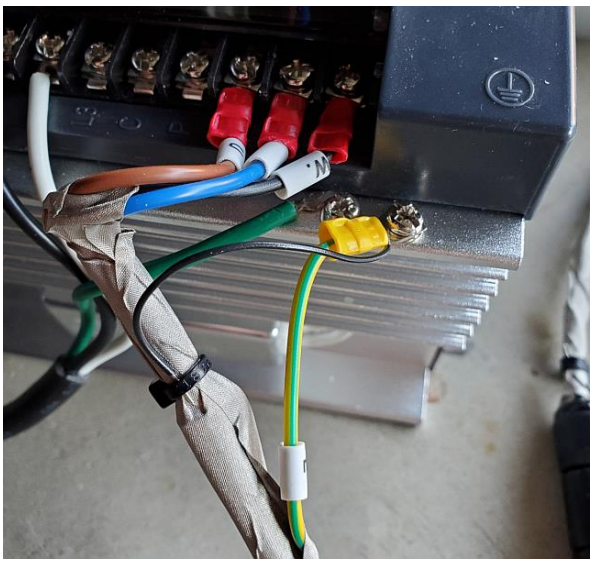




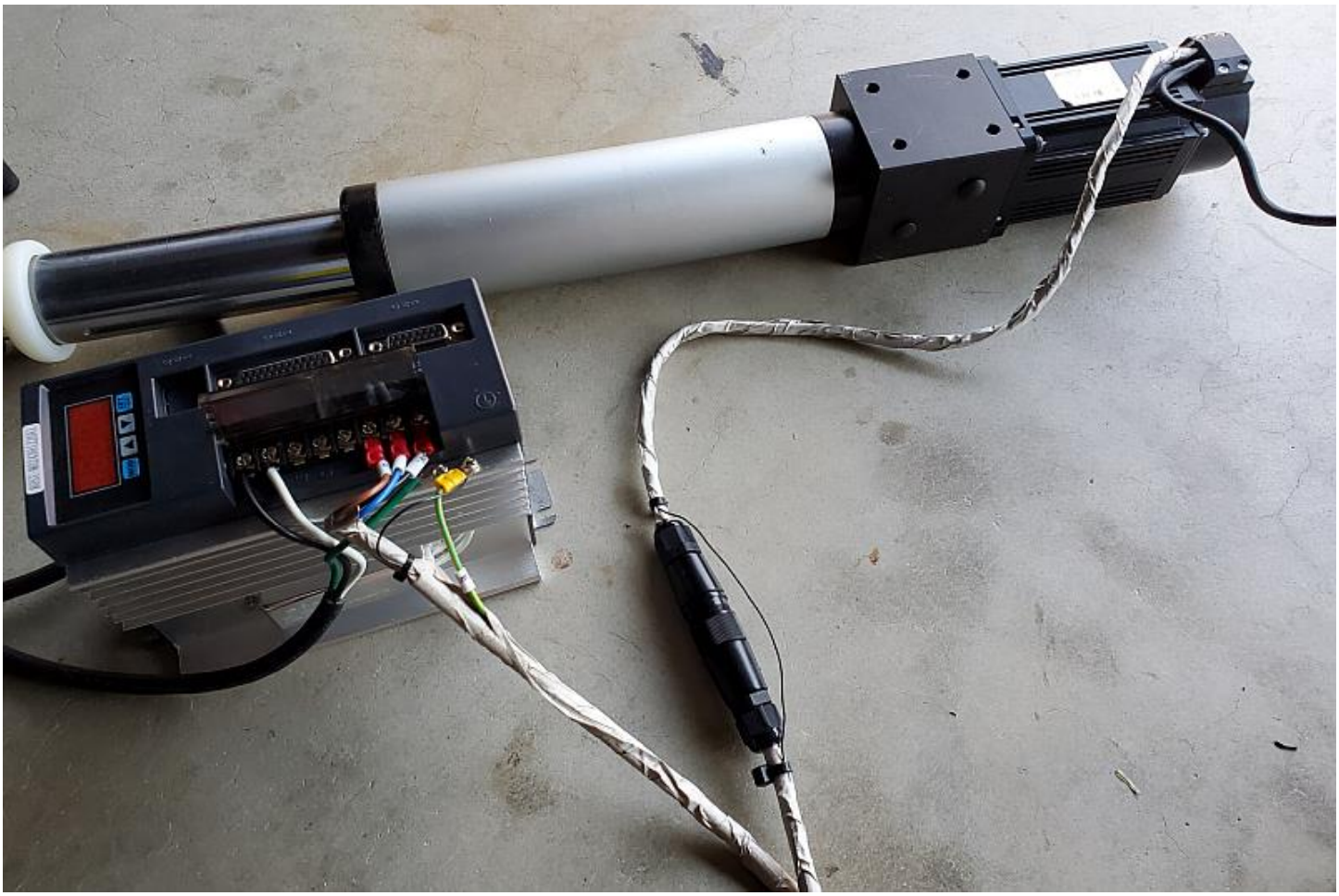
As you can see on the photos above the power cable is not shielded. Next to it, is a shielded cable.

To add the shielding externally you can use the conductive tape. See the following photos with the steps:



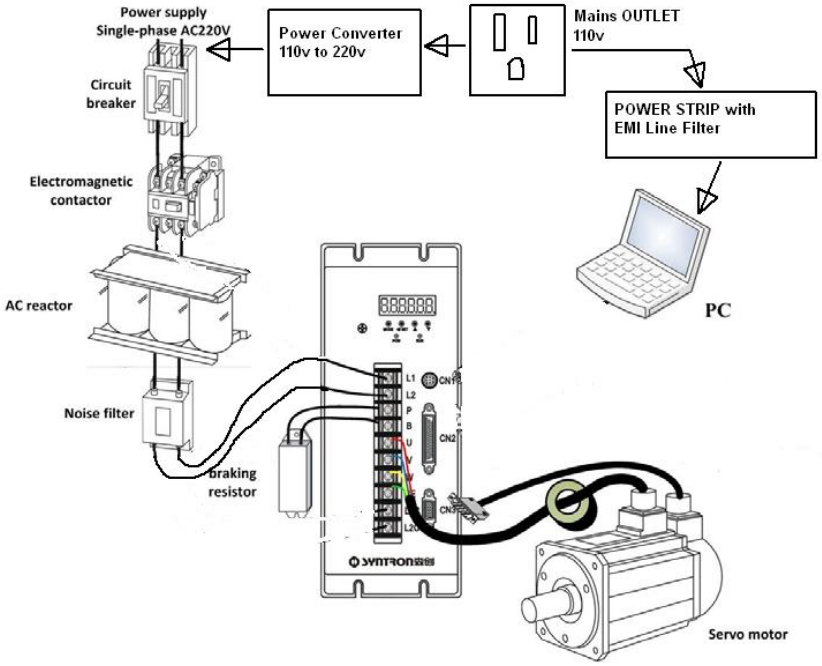


The conductive tape on this power cable needs to be bridged with a simple wire mounted with zip ties as its interrupted by the quick connect plug. This is needed to ensure the whole length of the shield (tape) is grounded, for it to act a faraday cage and eliminate the EMI noise properly. Also the tape has to be connected on the ground bolt (or PE) on the servo drive with another piece of wire and a zip tie.

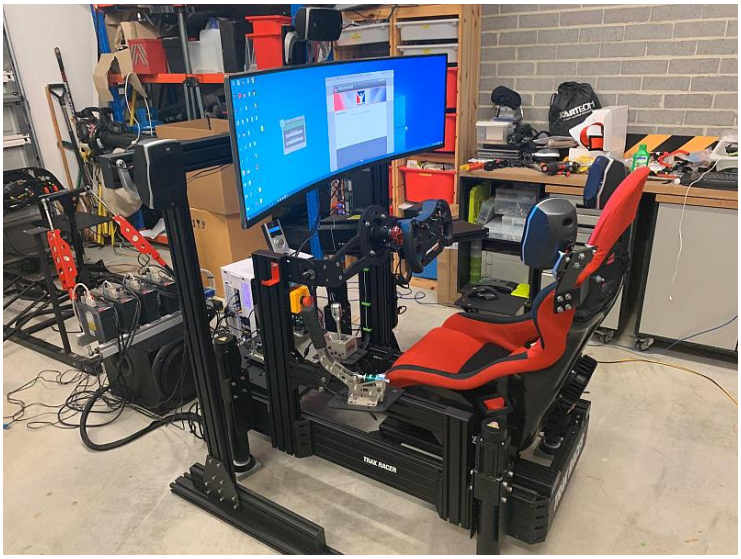


If you like more details this modification watch this video: https://www.youtube.com/watch?v=Vth5iRWEn0o

There is no need to apply the conductive tape on the encoder cable of the servomotor as the signals on it are just 5v level. Also there is no really need to add such shielding on the 110v or 220v power cable that comes from the mains, except by adding some line filter to prevent some of the EMI noise to propagate to other connected devices on same power line like the PC or peripherals:



Here are some examples of platforms for use with the AMC-AASD15A:



Programmer's information:

The data packet string now is 20 bytes long and includes additional spare motion data slots for up to 8axis

The ID is byte values 0xFF + 0xFF
Each Axis is 16bit wide.
LF+CR is required in the end (0x0A + 0x0D)

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

The parameters can be changed via terminal (250000 bps)

---List of commands---

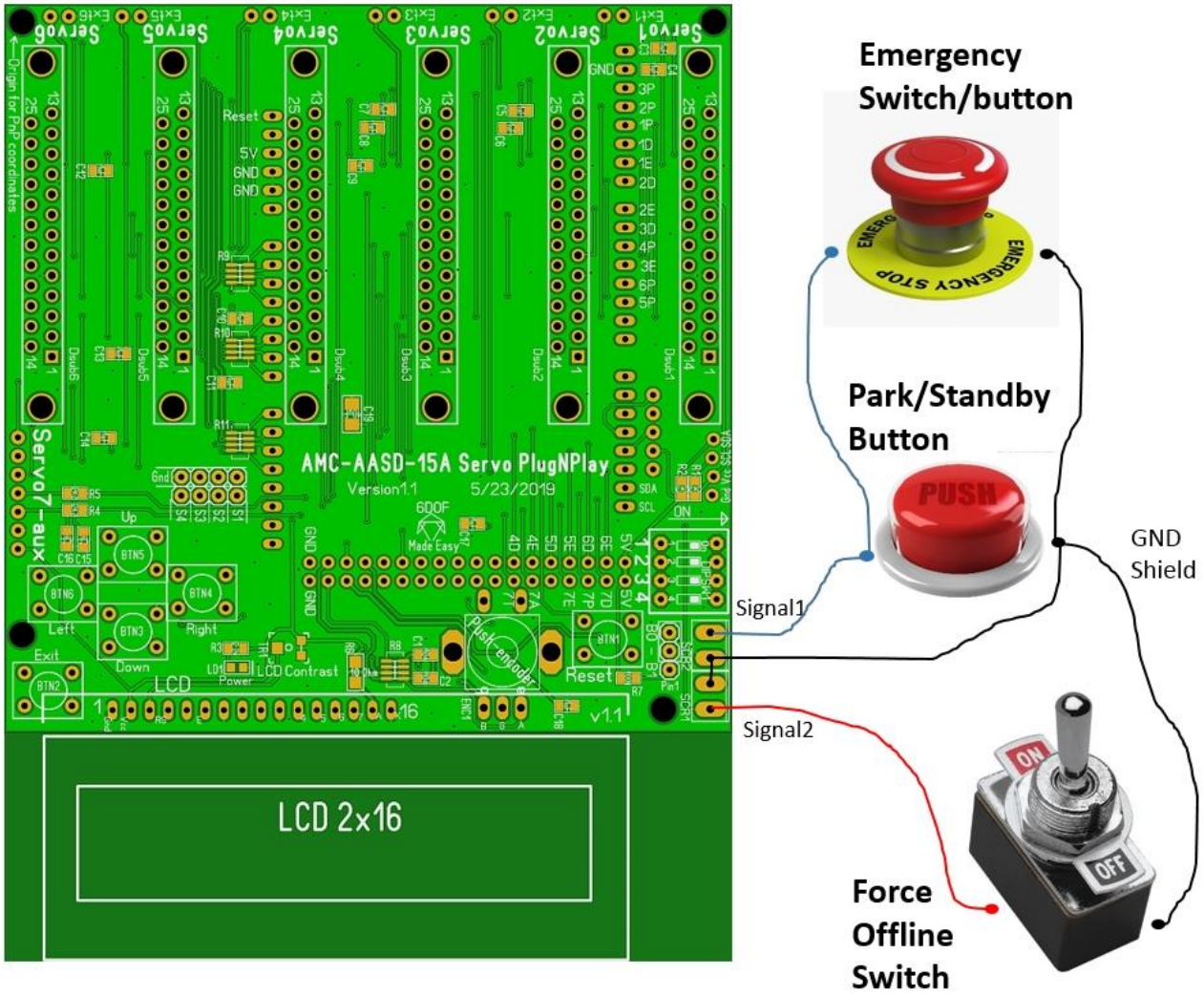
Command Number	Display Parameter	Save Parameter
CMD01	Motornumber:	spv012-spv018
CMD04	Park Position:	spv04001-spv04254
CMD05	Park Move Speed:	spv05001-spv05100
CMD06	Park Move Timeout:	spv0601-spv0690
CMD07	Standby Position:	spv07010-spv07245
CMD08	Standby Speed:	spv08000-spv08100
CMD09	Standby Timeout:	spv0901-spv0990
CMD10	Disable park type:	spv111-spv115
CMD13	Actuator Limits:	spv1300-spv1350
CMD14	Kill switch mode:	spv141-spv142
CMD44	Display all parameters	

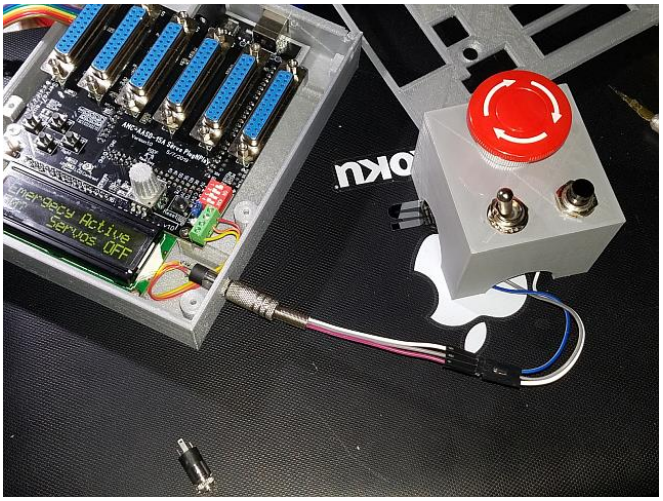
Command Number	Display Parameter	Save Parameter
CMD45	Print this help page	
CMD55	Print delimited parameter list for simtools	
spv45	Saves all parameters at once	
RQM	Displays model,revision and number of motors	
Park	Parks the actuators if in standby mode	

```
  Some Commands may not change value - locked

The CMD$$ displays each parameter, and spv$$### saves each parameter with the value indicated.
To actually store the parameters in the flash memory you need to send "spv45" to save all
parameters at once. The "$$" on the spv is the command number, and the "###" is the value,
Some parameters have single digit value, some two digit value and some 3 digit value.
All values are characters!

Here is a list of the default parameters values you should get when you issue the CMD44 command
(if not like this, you may reset the default parameters via button combination)

01.Motornumber 2-8: 4
04.Park Position 0-254: 1
05.Park_Move_Speed 1-100%: 11
06.Park_Move_Timeout 1-90: 5
07.Standby Position 10-245: 127
08.Standby Speed 0-100%: 24
09.Standby Timeout 1-90: 5
10.Disable park type 1-5: 1
13.Actuator Limits 0-50%: 1
14.Kill switch mode 1-2: 1

“CMD55” returns the following numeric values separated by colon ( : ) punctuation mark:
"data:" <Motornumber> ":" <Parkposition> ":" <Parkmovespeed> ":" <Parkmovetimeout> ":"
<StandbyPosition> ":" <StandbySpeed> ":"<StandbyTimeout> ":" <Disableparktype> ":"
<ActuatorLimits> ":" <Killswitchmode> ":" <Firmwareversion> ":" <AMCModel>
```

Wiring Button – E-STOP switch – Force Offline switch

Note that the switches/buttons must be “NO” type (Normally Open)





You can use a 3.5mm panel mount audio jack to simplify wiring and use ready available shielded audio wire (stereo type)

Wiring DB25 female connector on Servo7-aux pins

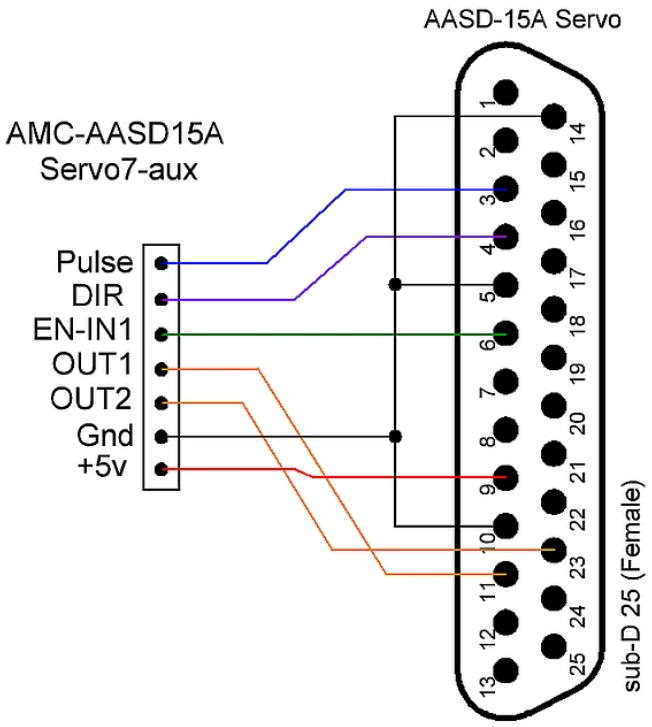
If you need more that 6 axis, you can add a DB25 female connector and plug a 7th servo in the controller. Possible uses are 4DOF+TL+Surge+Belt tension, or 6DOF+rotation axis…

See the videos for details on wiring

https://www.youtube.com/watch?v=CT7M-8LCCwc

https://www.youtube.com/watch?v=pVbQSvhRTq4

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

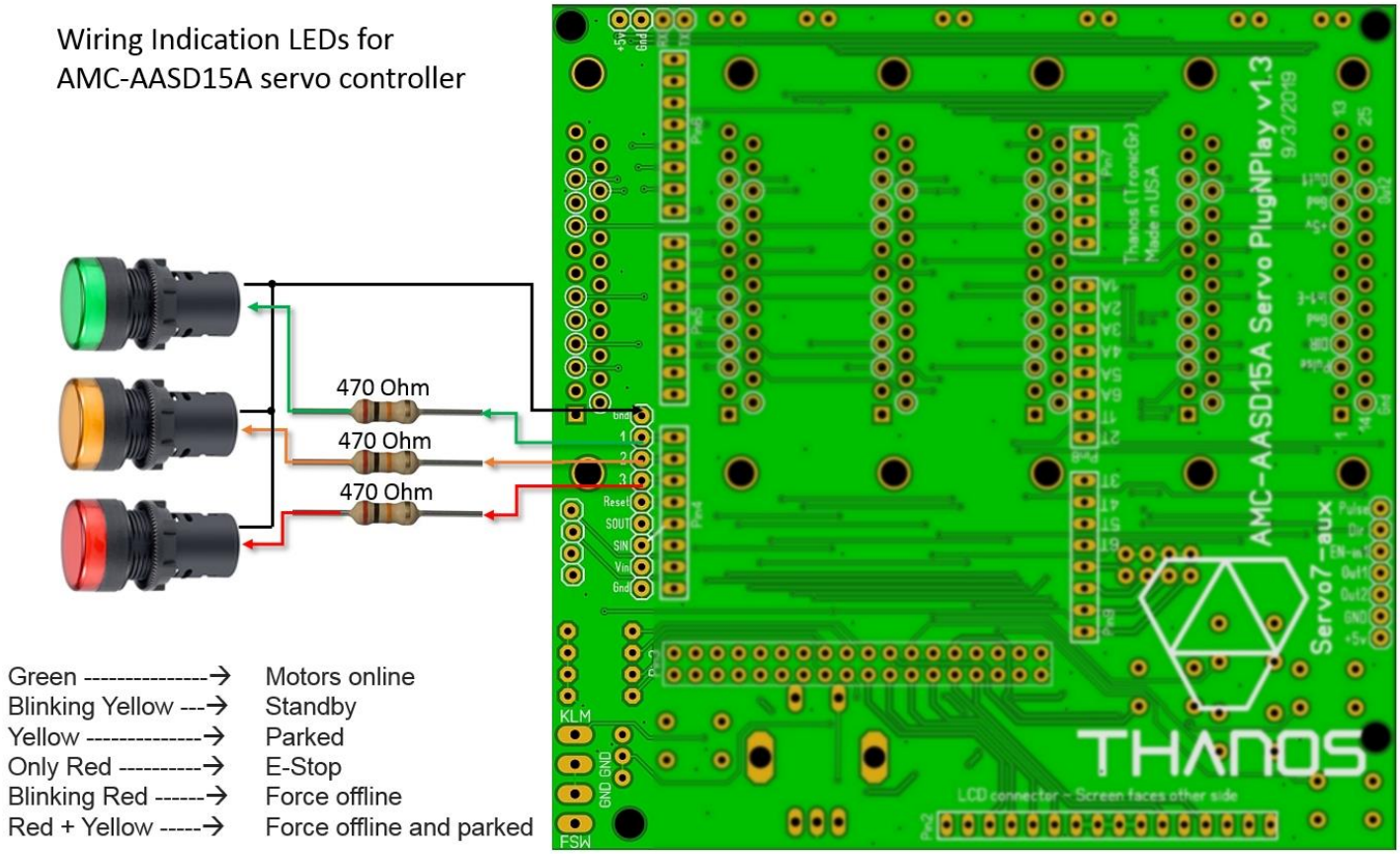


You can now order that allows easy attachment of the BD25 connector without wiring, just a 2.54mm header (male and female): https://oshpark.com/shared_projects/pSGIJpx4

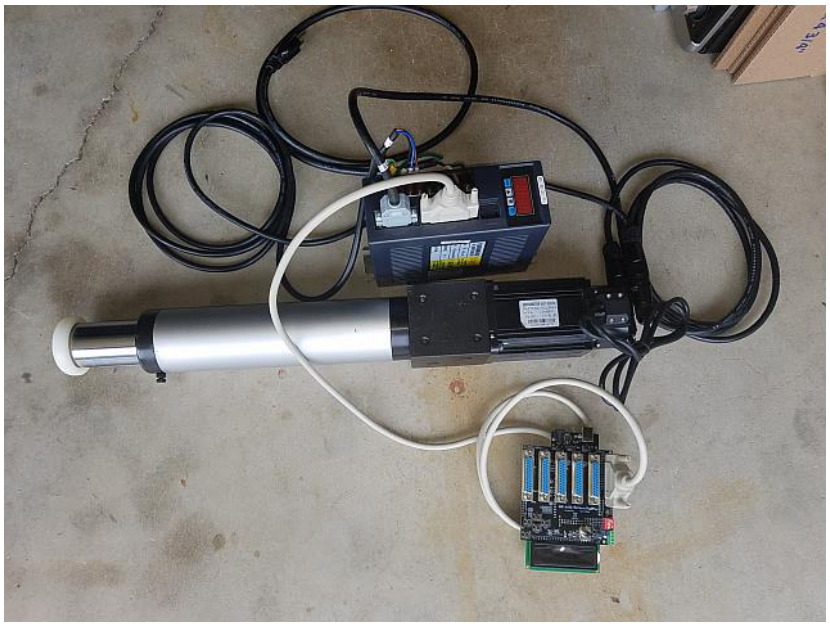
Wiring indication LEDs on AMC-AASD15A

You may need LED indication for the state of the controller from within the cockpit, in that case you can wire some simple LEDs if you don't have access to the LCD display of the controller.

```
Meaning of LED indications depending on the state of the controller:
Green ------------->     Motors online
Blinking Yellow --->     Standby
Yellow ------------>     Parked
Only Red ---------->     E-Stop
Blinking Red ------>     Force offline
Red + Yellow ------>     Force offline and parked
```



AASD-15A Servo drives SETUP



The AMC-AASD15A can be interfaced to all models of AASD that have the DB25 connector and are compatible. Example servo and drive below.

80ST-M02430 220V 0.75W AC Servo Motor 2.4N.M 3000RPM Servo Motor Single-Phase AC Drive Permanent Magnet Matched Driver AASD-15A
https://www.aliexpress.com/item/80ST-M02430-220V-0-75W-AC-Servo-Motor-2-4N-M-3000RPM-Servo-Motor-Single-Phase/32973113245.html

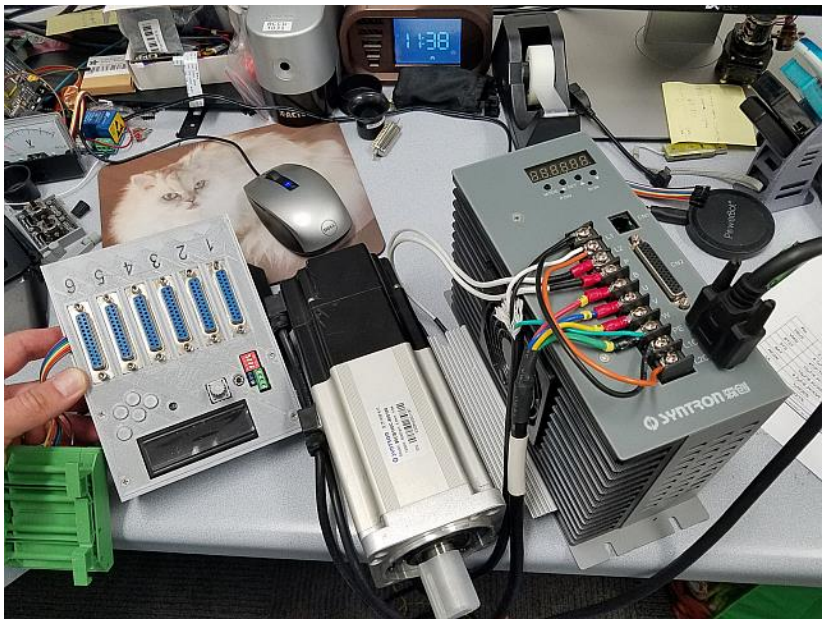
Or visit PT-Actuator for selection of servos on various actuators that may fit your motion simulator type:
http://www.pt-actuator.com/index.asp

The AASD-15A drives need some parameters before they are ready to be used. Most of the parameters are same as SFX100 DIY but some additional one are required.

AASD-15A Servo Settings:
Push MOD until you see Pn000. This enters the parameter mode.
Change and check these settings on all motors:

Pn8 = 300
Pn9 = -300
Pn51 = 3000
Pn98 = 20 - Pulse Multiplier (electronics gear)
Pn109 = 1 - smoothing, 1=fixed smoothing, 2=s-Shaped smoothing
Pn110 = 30 - Smoothing Filter Time
Pn113 = 20 - Feedforward %
Pn114 = 10 - Feedforward Filter Time (ms)
Pn115 = 100 - Gain %
---Extra parameters needed---
Pn24 = 100
Pn52 = 1
Pn60 = 2
Pn61 = 6

Syntron HS Servo drives SETUP

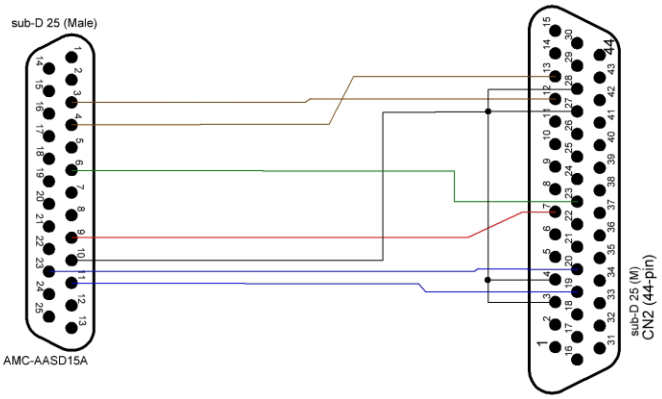


Note: Syntron HS drives require 101Khz Pulse Frequency setting on the AMC-AASD15A servo controller

Example cable between AMC-AASD15A and Syntron HS .

AMC-AASD15A pinout	
9	DC 5V
10	COM (GND)
6	SigIn1 (S-ON)
11	SigOUT1 (Servo ready)
23	SigOUT2 (Treach)
3	PP+ (STEP)
14	PP-
4	PD+ (DIR)
5	PD-

Syntron HS CN2 pinout	
7	DC 5V
23	SigIn1 (S-ON)
19	SigOUT1 (Servo ready)
3	Digital Output 1 (-) (GND)
20	SigOUT2 (Treach)
4	Digital Output 2 (-) (GND)
12	PULS+ (STEP)
27	PULS- (GND)
13	SIGN+ (DIR)
28	SIGN- (GND)



Complete list of values to enter in the Syntron HS:

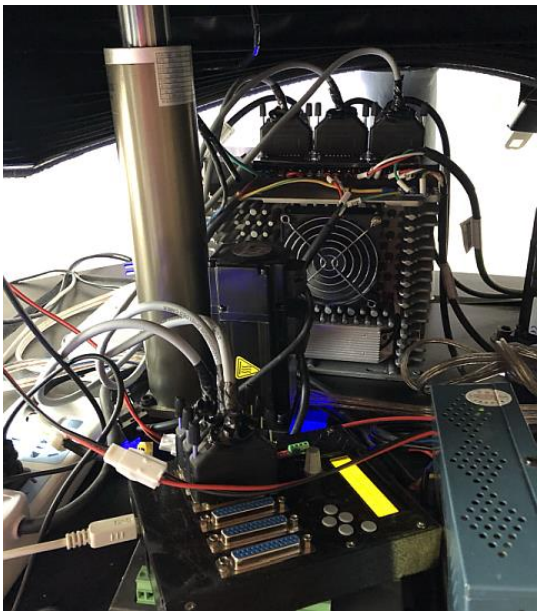
Fn 000 = 1 Hard-wired control port work mode
Fn 001 = 2 External pulse position mode
Fn 010 = -1 Servo Enabling (S-ON)
Fn 030 = -18 Servo Ready (ENA-SRV)
Fn 031 = -6 Target Torque Reached
Fn 038 = 1 Single pulse Positive logic (Pulse/DIR)
Fn 039 = 0 (2.5Mhz filter)
Fn 049 = 100 Torque threshold
Fn 04A = 10 Torque hysteresis
Fn 050 = 20 Electronic gear ratio numerator
Fn 054 = 1 Electronic gear ratio denominator
Fn 004 = 0 Motor direction 0=CCW - 1=CW

Fn 056 = 120 Pulse input command exponent filtering (Acceleration - deceleration)
Fn 057 = 32 Pulse input command moving smoothing filtering
Fn 05A = 0 Position loop speed feed-forward coefficient
Fn 05B = 10 Position-loop speed feedforward low-pass filter's time constant
Fn 05C = 80 Position loop's No.1 proportional gain Kp1
Fn 05D = 80 Position loop's No. 2 proportional gain Kp2
Fn 0DA = 32767 Number of tracking error pulses

Fn 0AC = 3000 (MAX MOTOR RPM)

Monitoning:
DN 00 RPM
DN 01 COMMAND FREQ
DN 02 TORQUE
DN 03 Position Tracking errors
DN 07 Motor speed command value (RPM)

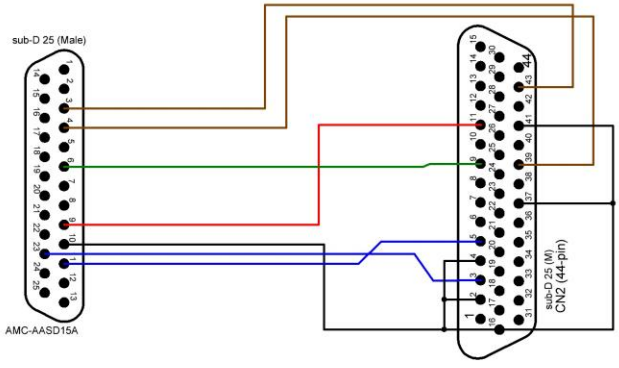
Dorna M1 Servo drives SETUP



Example cable between AMC-AASD15A and Dorna M1

AMC-AASD15A pinout	
9	DC 5V
10	COM
6	SigIn1 (S-ON)
11	SigOUT1 (Servo ready)
23	SigOUT2 (Treach)
3	PP+ (STEP)
14	PP-
4	PD+ (DIR)
5	PD-

Dorna M1 pinout	
11	DC 5V
16	COM SG
9	SigIn1 (S-ON)
5	SigOUT2 (Servo ready)
4	Digital Output 2 (-) (GND)
3	SigOUT3 (Treach)
2	Digital Output 3 (-) (GND)
43	PULS+ (STEP)
41	PULS- (GND)
39	SIGN+ (DIR)
37	SIGN- (GND)



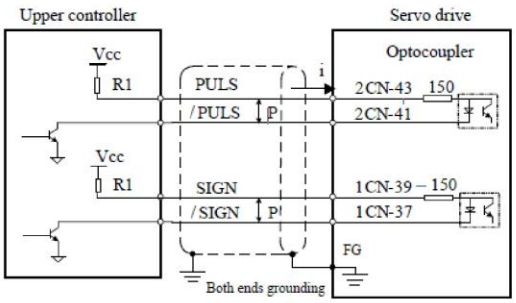
Complete list of values to enter in the Dorna M1:

PA000 = h0000
PA001 = h0000
PA200 = 0000
PA205 = 20
PA206 = 1
PA214 = 5
PA215 = 5
PA216 = 30
PA402 = 120
PA403 = 120
PA404 = 100
PA405 = 100
PA500 = 0
PA508 = b.0000
PA510 = h.0d40
PA511 = b.0000

PA500 = 0 (DEC) (INPUT)
[0] Servo-on (S-ON)

PA510 = 0d40
[4] Servo ready signal (S-RDY): active when servo is in proper status.
[D] Torque Reach signal (Treach): active when load torque reaches PA404/PA405.

- Open collector, option 2 (external 5VDC, 12VDC or 24VDC)



Input current I = 10 ~ 15mA, thus R1 resistance:
If 24VDC, R1=2K Ω;
If 12VDC, R1=510 Ω;
If 5VDC, R1=180 Ω;

Contact for inquiries or questions: Tronicgr@gmail.com