Ensemble Notes

IP address: 134.79.218.154

Subnet Mask: 255.255.252.0

Default Gateway: 134.79.219.1

IP address change:

1. Control panel
2. View network status and tasks
3. Click on ethernet connection
4. Properties
5. Internet Protocol Version 4(TCP/IPv4) properties
6. Change IP address to 134.79.218.--- (like 100)
7. Subnet mask to 255.255.252.0
8. No need for default gateway

What to do?

- Retrieve parameters
- save them on a file on the pc
- Address the questions of the SmartMotor on the controller
- Figure out how to connect the motors to the controller

Aerotech BMS60-A-D25-FLB-E1000ASH-15DM (BRUSHLESS MOTOR):

The configuration shown in Figure 2-5 is an example of a typical brushless motor connection.

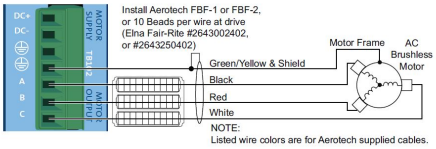


Figure 2-5: Brushless Motor Configuration

Table 2-9: Wire Colors for Aerotech Supplied Cables (Brushless)

Pin	Wire Color Set 1 (1)	Wire Color Set 2	Wire Color Set 3	Wire Color Set 4
⏚	Green/Yellow & Shield (2)	Green/Yellow & Shield	Green/Yellow & Shield	Green/Yellow & Shield
A	Black	Blue & Yellow	Black #1	Black & Brown
B	Red	Red & Orange	Black #2	Red & Orange
C	White	White & Brown	Black #3	Violet & Blue

(1) Wire Color Set #1 is the typical Aerotech wire set used by Aerotech.
(2) "&" (Red & Orange) indicates two wires; " / " (Green/White) indicates a single wire

fig 1: brushless motor connections

Initiating Ensemble:

- Connect to LAN cable
- Connect motors as shown in pic on the left
- Controller connect to connect
- Enable or write ENABLE @0 in the Execute Command

Notes:

- PositionFeedback I assume is feedback from encoders (cnts)
- ProgramPositionFeedback is feedback in (deg)
- current command and current feedback to monitor current
- How is the direction of motion defined and units
  - positive is CW

Command motions to keep track of:

- **LINEAR** X 20 F 100 (works without F)
  - Goes distance 20 at speed 100 on axis X. Linear motion
- prog_dist= **CMDPOS** (X)
  - Gets the command position (enc counts)
- temp_dist= **EXTPOS** (X)
  - Gets external position of axis
- temp_val= **GETPARM**(X, GainKpos)
  - retrieves a single parameter for specific axis or task
- **HOME** X Y
  - Return to their reference positions (home)
- temp_current = **ICMD** (X)
  - return current command in amps
- **FREERUN** X 200 Y 300
  - Keep running without a specified distance, just speed
- **OSCILLATE** X, 5000.0, 1.0, 2, 3
  - OSCILLATE <Axis>, <Distance>, <Frequency>, <Cycles> , <NumFreqs> (last not necessary)
  - Generate sinusoidal oscillations on an axis
- temp_dist= **PCMD**(X)
  - Return the position commanded to the servo in (user units)
- **MOVEABS** X 5 XF 10

How I fixed the task error: not activated feature:

- FeatureNotEnabled

  FeatureNotEnabled

  You tried to use a feature that has not been activated on the controller.

  You are trying to use a feature that is not activated on the controller. This error can occur when you try to set one of the following parameters.

  Axis → Dynamic Controls Toolbox → Threshold Gain Scheduling → ThresholdScheduleSetup

  Axis → Dynamic Controls Toolbox → Dynamic Gain Scheduling → DynamicScheduleSetup

  Axis → Dynamic Controls Toolbox → Command Shaping → CommandShaperSetup

  Axis → Dynamic Controls Toolbox → Harmonic Cancellation → HarmonicCancellationSetup

  Axis → Enhanced Tracking Control → EnhancedTrackingSetup

  If this error occurs while you are sending a parameter file to the controller, use **Configuration Manager** to set the values of the parameters listed above to the default values. To change parameters that are not included in your controller license, you must set the **Show Only Licensed Parameters** preference to **False** in **Configuration Manager**.

- Refer to License Options for more information on which features are available with a given license.
- Set them to default values, or 0
- Or basically: Axis Enhanced Tracking Control EnhancedTrackingSetup was set to 1 changed to 0

Table 2-12: Motor Feedback Connector Pin Assignment (J103)

Pin#	Description	In/Out/Bi
1	Chassis Frame Ground	N/A
2	Motor Over Temperature Thermistor	Input
3	+5V Power for Encoder (500 mA max)	Output
4	Reserved	N/A
5	Hall-Effect Sensor B (brushless motors only)	Input
6	Encoder Marker Reference Pulse -	Input
7	Encoder Marker Reference Pulse +	Input
8	Analog Input 0 -	Input
9	Reserved	N/A
10	Hall-Effect Sensor A (brushless motors only)	Input
11	Hall-Effect Sensor C (brushless motors only)	Input
12	Clockwise End of Travel Limit	Input
13	Optional Brake - Output	Output
14	Encoder Cosine +	Input
15	Encoder Cosine -	Input
16	+5V Power for Limit Switches (500 mA max)	Output
17	Encoder Sine +	Input
18	Encoder Sine -	Input
19	Analog Input 0 +	Input
20	Signal Common for Limit Switches	N/A
21	Signal Common for Encoder	N/A
22	Home Switch Input	Input
23	Encoder Fault Input	Input
24	Counterclockwise End of Travel Limit	Input
25	Optional Brake + Output	Output

Connector

Mating Connector	Aerotech P/N	Third Party P/N
25-Pin D-Connector	ECK00101	FCI DB25P064TXLF
Backshell	ECK00656	Amphenol 17E-1726-2

fig 2: feedback connections

fig 3: specs

Parameters:

- FaultMask Parameter:
  - Use it to define the axis fault conditions detected by the controller
  - You can configure what the axis does for each fault
  - Recovering the axis fault :
    - correct the condition that caused the controller to generate an axis fault
    - Click acknowledge all button or issue an ACKNOWLEDGEALL command to clear all axis faults
- Feedback
- Motor (Motortype):
  - Use it to configure various settings related to the motor connected to the axis
- Units:
  - Use the CountsPerUnit parameter to specify the number of encoder counts per primary programming unit
    - always use a positive value

Questions to answer:

1. Identify and document how to look at faults and how to clear them
2. Identify and document how to read motor position readback
   a. On screen (Program Position Feedback
3. Identify and document how to engage/disengage brake
   a. Enable/Disable axis
   b. ENABLE @0
4. Identify and document how to enable/disable amplifier
5. Identify and document how to set motor speed and acceleration
   a. units for speed and pos can be modified in the parameters->units
6. Identify and document how to jog motor in both directions
   a. Note which direction is the pos and which is neg
7. Identify and document how to stop motion
   a. **ABORT**

- Look at parameters and understand how the controller handles faults
- tasks: what they mean, and how to run them with autorun

Parameters to change:

- [ ] Prepare parameter file for test stand
- [x] Configure the faults to make sure it reacts by immediately stopping the motion, limit switches, estop position error, overrents
- [ ] Change the motor parameters (based on the type of motor (stepper motor)) (running current as well)
- [ ] Look up in the help how to set up a stepper motor (both in the ensemble software and cp drive)
- [x] set up the encoder, set up feedback parameters to use the right parameters
- [ ] Figure out conversion from motor turn to linear motion
- [ ] Conversion from encoder counts to primary unit
- [ ] microstepping (stepper motor generally 200 counts per rev)
- [ ] benefit of stepper(less fancy) over servo(fancy)
- [ ] Faults, motors, feeedback (its a resolute encoder)
- [ ] stepperholdingcurrent, stepperresolution

8. Identify and document how to perform absolute vs incremental moves
    a. Absolute:
        i. **MOVEABS** *<Axis> <Distance>* [*<Axis> <Distance>* ...]
        ii. **MOVEABS** *<Axis> <Distance>* [*<Axis> <Distance>* ...] *<Axis>* F*<Speed>* [*<Axis>* F*<Speed>* ...]
        iii. Example: **MOVEABS** X 5 XF 10
    b. Incremental:
        i. **MOVEINC** <Axis> <Distance> [<Axis> <Distance> ...]
        ii. **MOVEINC** <Axis> <Distance> [<Axis> <Distance> ...] <Axis> F<Speed> [<Axis> F<Speed> ...]
        iii. **MOVEINC** X 5 XF 10
9. Connect Renishaw encoder to Aerotech. Determine pinout and wire using breakout boards. Configure drive to just read encoder. Confirm that can see position readback change when encoder is moved manually. (Ask Alex for this)
10. Identify and document how to set system resolution so that target position can be commanded and read back in user units

Table: Axis Fault Bit Descriptions

Bit	AeroBasic Constant	Name	Description
0	AXISFAULT_PositionError	Position Error Fault	The absolute value of the difference between the position command and the position feedback exceeded the threshold specified by the PositionErrorThreshold parameter.
1	AXISFAULT_OverCurrent	Over Current Fault	The average motor current exceeded the threshold specified by the AverageCurrentThreshold and AverageCurrentTime parameters.
2	AXISFAULT_CwEOTLimit	CW/Positive End-of-Travel Limit Fault	The axis encountered the clockwise (positive) end-of-travel limit switch.
3	AXISFAULT_CcwEOTLimit	CCW/Negative End-of-Travel Limit Fault	The axis encountered the counter-clockwise (negative) end-of-travel limit switch.
4	AXISFAULT_CwSoftLimit	CW/High Software Limit Fault	The axis was commanded to move beyond the position specified by the SoftwareLimitHigh parameter.
5	AXISFAULT_CcwSoftLimit	CCW/Low Software Limit Fault	The axis was commanded to move beyond the position specified by the SoftwareLimitLow parameter.
6	AXISFAULT_AmplifierFault	Amplifier Fault	The amplifier for this axis exceeded its maximum current rating or experienced an internal error.
7	AXISFAULT_PositionFbk	Position Feedback Fault	The drive detected a problem with the feedback device specified by the PositionFeedbackType and PositionFeedbackChannel parameters.
8	AXISFAULT_VelocityFbk	Velocity Feedback Fault	The drive detected a problem with the feedback device specified by the VelocityFeedbackType and VelocityFeedbackChannel parameters.
9	AXISFAULT_HallFault	Hall Sensor Fault	The drive detected an invalid state (all high or all low) for the Hall-effect sensor inputs on this axis.
10	AXISFAULT_MaxVelocity	Maximum Velocity Command Fault	The commanded velocity is more than the velocity command threshold. Before the axis is homed, this threshold is specified by the VelocityCommandThresholdBeforeHome parameter. After the axis is homed, this threshold is specified by the VelocityCommandThreshold parameter.

Parameters changed from default:

Fault:

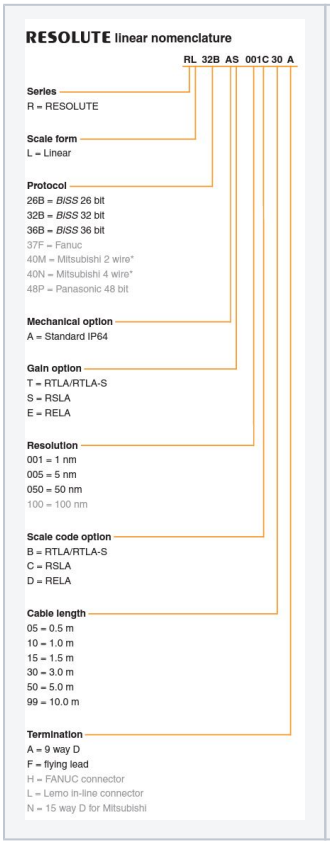
- FaultMask:

- Encoder Fault (previously off, now on)
    - FaultMaskDecel:
- Position Error Fault (previously off, now on)
    - FaultMaskDisable:
- Velocity Feedback Fault (previously on, now off)

Feedback:

- Resolute Encoder part number: RL26BAE050D30A
    - RL: Resolute Linear
    - 26B: BISS 26 bit
    - A: Standard IP64
    - S: RSLA
    - 050: Resolution 50 nm
    - D: RELA
    - 30: cable length 3.0m
    - A: 9 way D
- AbsoluteFeedbackOffset= -30mm
- PositionFeedbackType set to 9 (for resolute encoder)
- Resolute encoder:
    - ResoluteEncoderResolution set to 26 (from BISS 26 bit)

Motor:



Part Number:	HT23-601C
Frame Size:	NEMA 23
Motor Type:	High torque
Part Number w/Double Shaft:	HT23-601DC
Part Number w/Encoder:	HT23-601DC-ZAA or HT23-601DC-CAA
Part Number w/Encoder & Cover:	HT23-601DC-ZAC
Encoder Feedback:	Optional
Encoder Resolution:	2,000 lines (8,000 counts quadrature)
Motor Length:	3.20 inches
Number of Lead Wires:	8
Lead Wire Configuration:	shielded cable, no connector
Lead Wire/Cable Length:	10 feet
Lead Wire Gauge:	22 AWG
Unipolar Holding Torque:	191 oz-in
Bipolar Holding Torque:	269 oz-in
Step Angle:	1.8 deg
Bipolar Series Current:	2.12 A/phase
Bipolar Series Resistance:	2.2 Ohms/phase
Bipolar Series Inductance:	6.8 mH/phase
Bipolar Parallel Current:	4.24 A/phase
Bipolar Parallel Resistance:	0.7 Ohms/phase
Bipolar Parallel Inductance:	1.7 mH/phase
Unipolar Current:	3.00 A/phase
Unipolar Resistance:	1.1 Ohms/phase
Unipolar Inductance:	1.7 mH/phase
Rotor Inertia:	6.51E-03 oz-in-sec^2
Integral Gearhead:	No
Weight:	2.2 lbs
Storage Temperature:	-30 +70 °C
Operating Temperature:	-20 +50 °C
Insulation Class:	B
Shaft Diameter:	6.35 mm (0.250 inch)

- CountsPerRev set to 100000 (how many encoder counts per 1mm of movement) (assuming 5mm/rev)
    - since we have a 50 nm resolution encoder, 50nm=1count, 1mm/50nm= 20000 counts
- MaxCurrentClamp set to 10 (standard) (Currently set at running current 4.24A)
- MotorType set to 3 (for stepper)
- Stepper:
    - StepperDampingCutoffFrequency/Gain /StepperHoldingCurrent set from other parameter file
    - StepperResolution (microstepping) set to 40000 (doubled the CountsPerUnit for more accuracy & control)
    - StepperRunningCurrent set to 4.24 (from the datasheet)

Units:

- CountsPerUnit: 20000 (20000 counts of the encoder for 1mm 20000 counts/mm)
- UnitsName: mm

Notes:

- look at CyclesPerRev (electrical cycles)
- look at encoder counts
- If no gear reducer available, **CountsPerRev** are external encoder counts per one rev of motor
- **StepperResolution** is something I choose, generally bigger than CountsPerRev * gear_reducer