APPLICATION NOTE FOR

FULLY INTEGRATED SERVO MOTORS

SMARTMOTOR™ TROUBLESHOOTING WITH STATUS BITS AND SMI TOOLS

Revised: 04/28/2020

DESCRIBES TROUBLESHOOTING METHODS FOR SMARTMOTOR™ SYSTEMS USING STATUS BITS AND SMI TOOLS

www.animatics.com

MOOG ANIMATICS

Copyright Notice

Please let us know if you find any errors or omissions in this manual so that we can improve it for future readers. Such notifications should contain the words "Troubleshooting with Status Bits and SMI Tools Application Note" in the subject line and be sent by e-mail to: animatics_marcom@moog.com. Thank you in advance for your contribution.

Contact Us:

Americas - West
Moog Animatics
2581 Leghorn Street
Mountain View, CA 94043
USA

Tel: 1 650-960-4215

Americas - East
Moog Animatics
1995 NC Hwy 141
Murphy, NC 28906
USA

Tel: 1 828-837-5115
Fax: 1 540-557-6509

Support: 1 (888) 356-0357

Website: www.animatics.com

Email: animatics_sales@moog.com

Table Of Contents

Introduction

Today's electromechanical systems are complex pieces of industrial machinery that use a variety of components, each of which has hours of engineering design work behind it. Therefore, when problems arise, it can be difficult to isolate the root cause of the issue due to the overall complexity of the machine. Fortunately, the onboard diagnostics and data collection capabilities of the SmartMotor™ integrated servo offer the machine designer several ways to help isolate and troubleshoot many issues associated with motion control and machine operation.

The purpose of this document is to provide SmartMotor troubleshooting strategies using its advanced data features along with tools provided in the SmartMotor Interface (SMI) software. It is important to note that the nature of servo motor applications is such that it is impossible to qualify or quantify every possible scenario into a single document. Therefore, the following strategies are generalized and designed to serve as an entry point.

In addition to the information here, see the *Moog Animatics SmartMotor™ Developer's Guide* for further information on the remaining Status Bits.

NOTE: The information in this document applies only to Class 5 and later SmartMotor products.

Troubleshooting - First Steps

When diagnosing any problem, perform these basic first steps to reset and check the equipment as follows:

1. Restart the motor.
2. Restart any associated devices (PLCs, HMIs, PCs).
3. Download a fresh copy of the program (especially if you've been making changes).
4. Check the wiring – make sure all connectors are secure and visually inspect all cables for any signs of wear.

After completing those steps, if the problem still hasn't been resolved, it's time for a deeper investigation to isolate the root cause. If you are unsure of where to start, you can follow the path laid out below. If you are trying to troubleshoot a specific error, please skip the following section.

NOTE: If possible, remove the motor from the machine and connect it to SMI using separate power supplies and cables.

1. Are LEDs on?

   If not, double check your power supply and cables to make sure proper connections are being made. If you are measuring the appropriate voltage at the motor end of your cable, it's likely the motor has been damaged and requires an RMA. If yes, consult the Understand the LEDs topic in your SmartMotor installation guide.

   NOTE: Class 5 D-style motors with the DE option and all M-style motors require separate control and amplifier power. See your SmartMotor installation guide for details.

   - If you are using a -DE motor, please ensure power is being supplied to both pin 15 and A1.
   - If you are stuck in the bootloader, contact Moog Animatics for assistance.
   - If the LEDs indicate you are not stuck in the bootloader, please move to the next step.

2. Can you establish communications?

   If not, make sure your communication settings and hardware are correct. Run the Lockup Wizard in SMI.

   NOTE: If you can only establish communication using the SMI Lockup Wizard, your user program may be setting up the port incorrectly. Clearing the EEPROM will reset the port settings. However, be aware that clearing the EEPROM will also erase the user program.

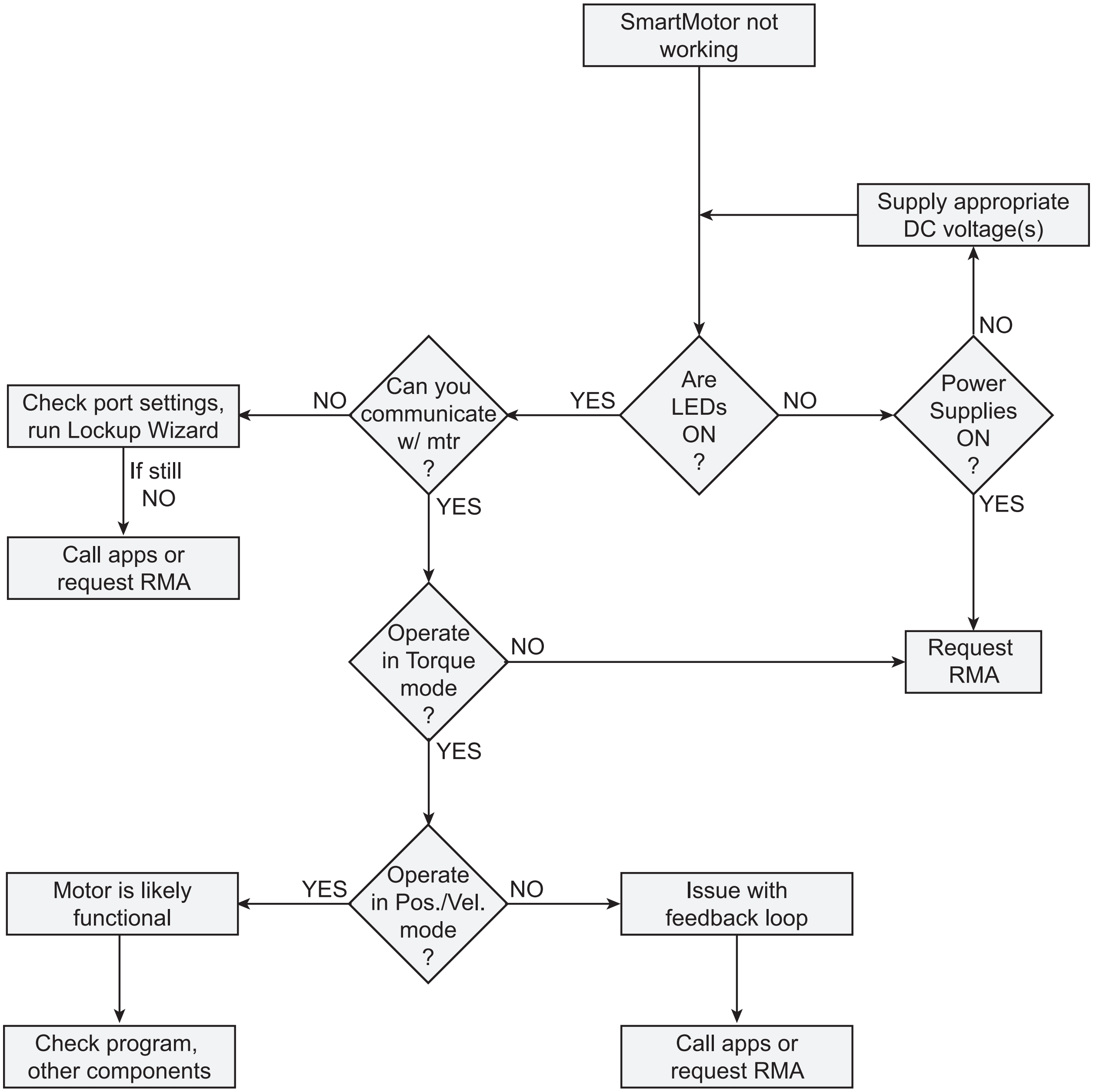
   If you still can't establish communications, then it's possible the motor has been damaged and requires an RMA. If you can, then move to the next step.

3. Will the motor operate in torque mode?

   Open up the SmartMotor Playground in SMI and move the motor in torque mode. Be aware, torque mode is open loop and can cause undesirable motion if precaution is not taken. Please be conscious of safety mechanisms like hardware limit switches, software limits, and E-stops. If not, make sure that there are no errors and the drive ready bit is active. Also, make sure that there is nothing physically impeding the shaft. If you cannot achieve motion, it's likely the motor requires an RMA. If you can achieve motion, then the drive stage is likely functional. Move to the next step.

4. Will the motor operate in position mode?

   Again, use the SmartMotor Playground to command motion in either position mode or velocity mode. If the motor fails to complete a move, it's possible the motor is damaged and requires an RMA. If this step works, it's safe to assume the motor is functional. At this point, it's time to start exploring other components in the system or the specific status bits described in the next section.



SmartMotor Troubleshooting Flowchart

Status Words

The next step should be to consult the Status Words. The Status Words, each made up of 16 Status Bits, contain a variety of information regarding SmartMotor health and operation. Please note:

- Not all Status Bits represent a fault condition
- Not every Status Bit is covered in this section, only the bits most associated with faults are discussed

It is recommended that you consult the Limits and Fault Handling section in the *Moog Animatics SmartMotor™ Developer's Guide* to obtain a clear understanding of what constitutes a fault condition.

Status Word 0: Primary Fault/Status Indicator

Bit 0 - Drive Ready

This condition is probably the most useful indicator available as it encompasses the fault state of the motor. If any of the following faults have occurred, this bit will be low or off. No motion can occur when in this state. The state of the PWR/SERVO LED (LED 0) also reflects the status of this bit.

Status Word	Status Bit
0	3, 5-15
1	12-15
3	1*,2*
6	12*, 13*, 14*
*These bits are model specific.	

NOTE: To simplify overall architecture and minimize bus traffic, Moog Animatics strongly recommends using Status Word 0, Bit 0 as the fault indicator for use by any host device (PLC, HMI or PC). Then the host only has to inquire about the status of a single bit instead of constantly polling for the status of numerous bits. Only when this bit is off should the host then inquire about the other status bits.

Bit 1 - Motor Off

This bit indicates the current state of the drive. By itself, it does not represent a fault condition. The default state of the motor is, in fact, to power up with the drive off. It simply means that the motor has yet to receive any motion commands or has been given an OFF command.

Bit 2 - Trajectory in Progress

This bit indicates the motor has an active trajectory. In general, this bit indicates motion; however, there are a few niche cases where this is not true. Consult the *Moog Animatics SmartMotor™ Developer's Guide* to determine what constitutes an active trajectory. This bit can be used to tell when a move has completed. When coupled with the Drive Ready bit (bit 0), you can learn some very important things about the state of the motor. See the following table.

Trajectory LED

	State	0	1
Drive LED	0	Fault occurred	Impossible case
	1	No trajectory, no faults	Active trajectory

NOTE: Moog Animatics recommends monitoring only Status Word 0, Bits 1 and 2 for detecting normal operation when using a host device (PLC/HMI/PC). Only when the host is not satisfied with the condition of either bit should it inquire for more information. This simplifies overall architecture by encapsulating irrelevant data, when necessary, and also keeps bus traffic to a minimum.

Bit 3 - Servo Bus Voltage Fault

This bit is triggered when either a low bus voltage or high bus voltage threshold is reached. The exact value of this threshold varies depending on the SmartMotor model being used. To see which threshold was met, see Status Word 6: Drive Modes, Bits 13 and 14.

The typical causes of a high voltage fault are regen and back EMF. Back EMF is the voltage produced by the motor as the rotor moves within the stator. It is inherent in all electric motors. Back EMF is commonly associated with speed or velocity. This is true in a constant velocity condition only. Back EMF is actually proportional to the rate of change of magnetic flux (magnetic field strength) inside the stator windings of the motor. The faster the rate of change, the higher the voltage rises. In other words, very large accelerations are capable of create very large voltage spikes. We typically see these accelerations when the motors experience a sudden, dramatic change in load, or encounters a hard stop.

Regen (regeneration) occurs when the motor is back driven. It behaves like a generator and becomes capable of producing a voltage that exceeds the operation range. This can be counteracted by placing a shunt onto the DC bus. The shunt is essentially a large, low resistance component that is capable of dissipating the energy as heat. It will not protect against back EMF due to high accelerations.

Bit 4 - Peak Overcurrent Occurred

In electric motors, current is analogous to torque, and torque is analogous to acceleration. If peak overcurrent situations are being met, there is typically a large torque or acceleration that is being experienced. It is always recommended to use the lowest acceleration values possible. The tradeoff is either higher overall velocities or longer move times.

NOTE: This condition by itself will not cause any change to the drive. If this condition is sustained for a period of time, the drive will limit the commanded power. The behavior varies depending on exact model.

Bit 5 - Real Time Excessive Temperature Fault

The default thermal limit of the SmartMotor is 85°C. When this limit is reached, the drive stage will shut off to prevent overheating and damaging the electrical components. Overheating is indicative of overloading. Therefore, if you find the motor is overheating quickly, it’s likely overloaded in the peak range. Lowering acceleration rates for higher speeds can help. If the motor is overheating slowly, it’s likely overloaded in the continuous range. A motor that is overloaded in this range is likely undersized for the application. Depending on how undersized the motor is, lowering the duty cycle might help. You can clear this bit using the ZS command.

A side note on temperature. The load capabilities of a motor are directly related to heat management. As more heat is removed from the motor, the more power you will be able extract from it. SmartMotor speed torque curves are defined using the motor’s thermal rise from 25°C to 85°C. If your expected ambient temp is higher than 25°C, you must de-rate the performance accordingly.

Bit 6 - Excessive Position Error Fault

This bit is triggered when the value EL is exceeded. This typically means the motor cannot complete the commanded trajectory. This happens when the motor is not capable of moving the load using the given motion parameters. Just because motion parameters are valid values does not mean the motor will be able to complete the trajectory.

Bit 7 - Velocity Limit Fault

This fault occurs when a velocity limit is reached. By default, the value is typically the no load top speed of the specific motor. This fault typically means the motor is being back driven. If the value is changed from the default, care should be taken to ensure the commanded velocity target (VT) does not exceed the new value.

Bit 8 - Real Time Temperature Limit

This fault often occurs in tandem with the Real Time Excessive Temperature Fault (Bit 5). It is true when at or above the temperature limit and has a 5°C cooling off hysteresis before automatically clearing itself. This bit cannot be manually cleared.

Bit 9 - Derivative Error Limit (dE/dt) Fault

This bit occurs when the value DEL is exceeded, by default this value is so large that it is effectively off. This value refers to the rate of change of following error and by nature, is much faster at detecting certain conditions. If this fault frequently occurs, it's likely the DEL value is too low for the application.

Bit 10, 11 - Hardware Limit Enabled

By default, the limit switches are enabled. This is done as a safety to prevent any accidental motion without at least a very basic understanding of SmartMotor operation. To disable, see the EIGN command.

Bit 12, 13 - Historical Limit

These bits indicates that at some point in the past, the limits were asserted. No motion can occur in the direction that corresponds with each bit. If both bits are active, no motion will be allowed in either direction. They can be cleared with the ZS command.

Bit 14, 15 - Limit Asserted

These bits are only valid when the limit switches are active. The hardware is configured so the limits are in the active state when nothing is wired. This is done as a safety to prevent any accidental motion. No motion can occur in the direction that corresponds with each bit. If both bits are active, no motion will be allowed in either direction.

Status Word 1: Index Registration and Software Travel Limits

Bit 12, 13 - Historical Software Overtravel Limit

Similar to the hardware historical limit bits, these indicate that, at some point in the past, the software limits were reached.

Bit 14, 15 - Real Time Software Limits

Software limits function similarly to hardware limits. If the motor ever exceeds the programmed position, motion stops in accordance with the FSA command. See the *Moog Animatics SmartMotor™ Developer's Guide* for a more thorough description of Software Limits.

Status Word 3: PID State, Brake, Move Generation Indicators

Bit 1 - Torque Saturation

This bit indicates that the motor has reached its maximum torque output. This can occur when the value set by the AMPS command is exceeded. The motors will also reduce torque output when trying to perform thermal limiting. This bit is also set when the thermal limiting protection begins reducing torque.

Bit 2 - Voltage Saturation

This occurs when the motor reaches its maximum speed for a given load and bus voltage. In other words, you cannot physically command any additional power from the SmartMotor after it has reached this point.

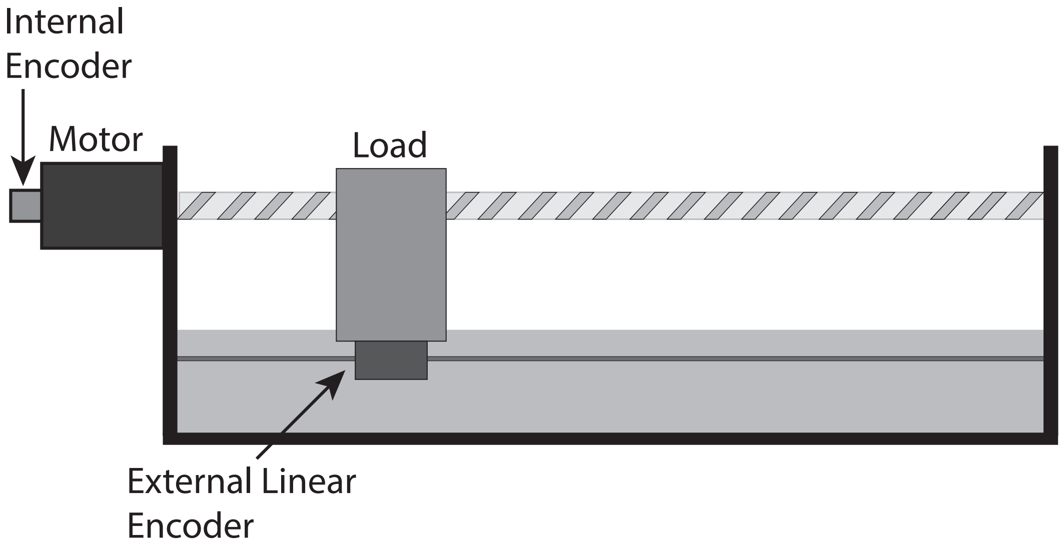
Status Word 6: Drive Modes

Bit 5 - Feedback Fault

Feedback is what makes an ordinary motor into a servo motor. A fault here typically indicates that the drive stage itself is functional but there is an issue with the feedback signal. If you are using the internal encoder (ENC0), this fault typically suggests something is wrong with the hardware itself. Other symptoms that would suggest the hardware needs to be looked at would be: decreased performance in sine mode (MDS), enhanced trap mode (MDE) when compared to standard trap mode (MDT) and failing to settle in the correct physical location, despite the motor reporting the correct location.

If you are using an external encoder (ENC1), there are few things that can go wrong. The first is wiring, make sure the external encoder is wired properly and there is a good clean signal. The other scenario is largely related to the mechanics of the system. It results from having a large mismatch between internal encoder counts and external counts. This is typically not an issue in direct drive scenarios, but it can occur when there is a large gear reduction. As a safety, the SmartMotor firmware sets this fault if it sees a very large number of internal counts and very small number of external counts. This is done to prevent runaway conditions if there is a loose wire or physical slippage.

For example, consider the following system:



Linear System with Internal and External Encoders

If you have an external linear encoder with intervals of 1 µm, then you will see 5000 counts across 5 mm of linear travel. The motor's internal encoder will see 4000 counts. If we introduce a 10:1 gearhead, the motor must then make 10 times as many revolutions and will see 10 times as many counts. We now have a ratio of 5000:40000 or 1:8. You can expect a feedback fault to occur when this ratio reaches 1:250.

Bit 12 - ABS Battery Fault

This fault only applies to motors with absolute encoders having battery backups. This bit is set by default. See the *Moog Animatics Class 5 M-Style SmartMotor™ Multiturn Absolute Encoder Option Guide* for details.

Bit 13 - Low Bus Voltage Fault

If a low voltage fault occurred, check to make sure your power supply is sufficiently sized. In linear power supplies, voltage will drop as the load increases. If it's found that the low voltage fault occurs whenever motion is commanded, it's likely the power supply is not large enough to handle the application.

> **NOTE:** Properly selecting the correct size and type of power supply for your SmartMotor™ application is critical. Not having enough voltage and current may hinder the performance of the SmartMotor, resulting in excessive position errors, overcurrent warnings, etc. For more details, see the *Moog Animatics SmartMotor™ System Best Practices Application Note*.

For example, when dealing with undersized power supplies, it's possible that the motor can cause the power supply to go well below the operating voltage of the motor. When this happens, you will get a low voltage fault, but quickly afterward the motor will power off, the power supply will rebound and turn the motor back on. This can happen so fast that it's tough to tell the motor was ever off – an indicator will be that program flow isn't as expected. For a quick confirmation, look at volatile memory, which is cleared on power loss, and set an arbitrary variable to some value. After the strange behavior or fault occurs, check the value of that variable — if it has been reset to 0, the motor lost power and restarted.

Bit 14 - High Bus Voltage Fault

The typical causes of a high voltage fault are regen and back EMF. Back EMF is the voltage produced by the motor as the rotor moves within the stator. It is inherent in all electric motors. Back EMF is commonly associated with speed or velocity. This is true in a constant velocity condition only. Back EMF is actually proportional to the rate of change of magnetic flux (magnetic field strength) inside the stator windings of the motor. The faster the rate of change, the higher the voltage rises. In other words, very large accelerations are capable of creating very large voltage spikes. We typically see these accelerations when the motors experience a sudden, dramatic change in load, or encounter a hard stop.

Regen (regeneration) occurs when the motor is back driven. It behaves like a generator and becomes capable of producing a voltage that exceeds the operation range. This can cause damage to the power source or the SmartMotor itself.

To protect against Back EMF and regen, place a shunt onto the DC bus. The shunt is essentially a large, low resistance component that is capable of dissipating the energy as heat. Shunts offered by Moog Animatics are automatically engaged when the bus voltage exceeds 49.5 VDC. It is important to keep this in mind when evaluating a power system.

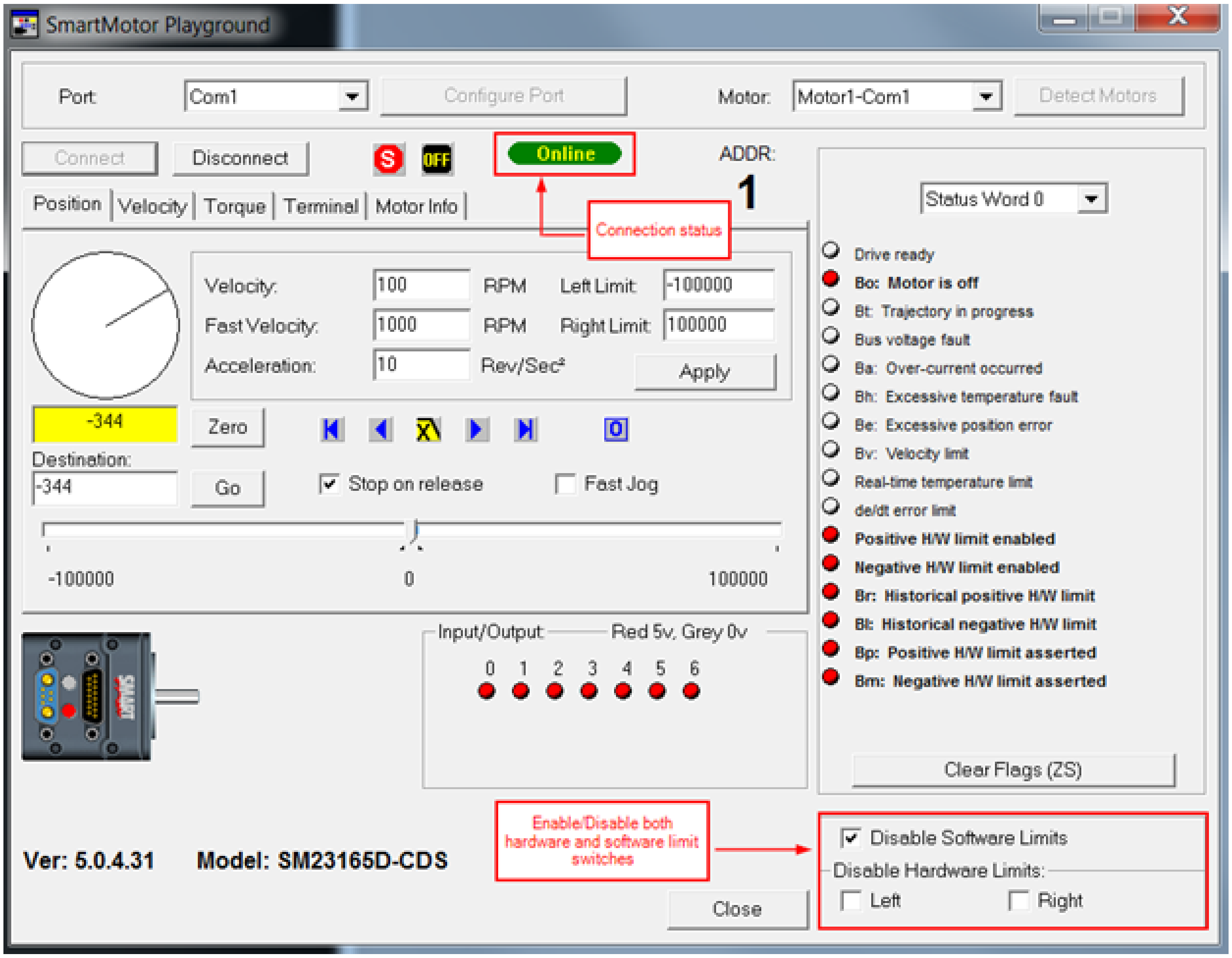
SMI Troubleshooting Tools

The free SmartMotor Interface (SMI) software, available on the Moog Animatics website, contains a set of graphical tools that make development and troubleshooting much easier. The key troubleshooting tools are outlined in the following sections.

NOTE: Random data in the SMI Terminal window is caused by one or more of the other SMI tools being open while trying to communicate with the motor through the Terminal window – the Terminal window is capturing the communications between that other tool and the SmartMotor. To prevent this, stop the tools from polling or disable PRINT commands using the SILENT command. See the SMI software online help for details.

SmartMotor Playground

The SmartMotor Playground is a graphical interface that displays information and allows you to control the motor. It is designed for simple jogging or verifying basic functionality like toggling I/O. The SmartMotor Playground should not be used for advanced troubleshooting and is not recommended for any development purposes.

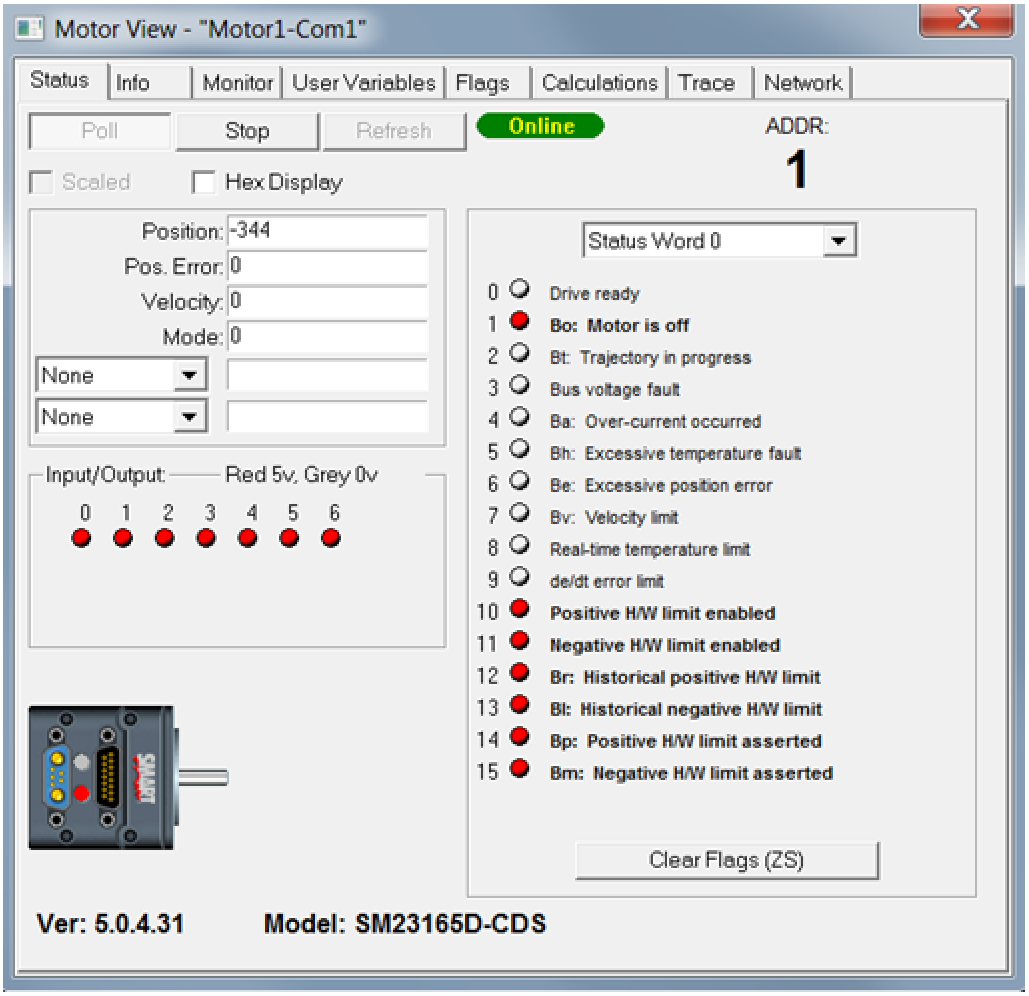


SmartMotor Playground

Motor View

The Motor View is the recommended diagnostic tool for most applications. It is similar to the SmartMotor Playground but does not allow direct control of the motor. It lets you track variables and view the state of the Status Words.

Note the various tabs in the Motor View – these allow you to see specific sets of information, thereby reducing the communication load on the bus. In situations where data throughput becomes critical, those tabs provide an easy mechanism for doing that. The other tabs also include a trace tool and a calculator that converts between real world units (RPM, RPS, RPSS) and motor units (VT, ADT).



Motor View

Chart View

The Chart View allows you to dynamically track the various conditions and values in the motor as the program runs. Anything that can be polled from the terminal can be tracked from the Chart View. To access the Chart View, open SMI and select Tools > Chart View. The following window opens. To begin polling, add a new chart item and then start recording.

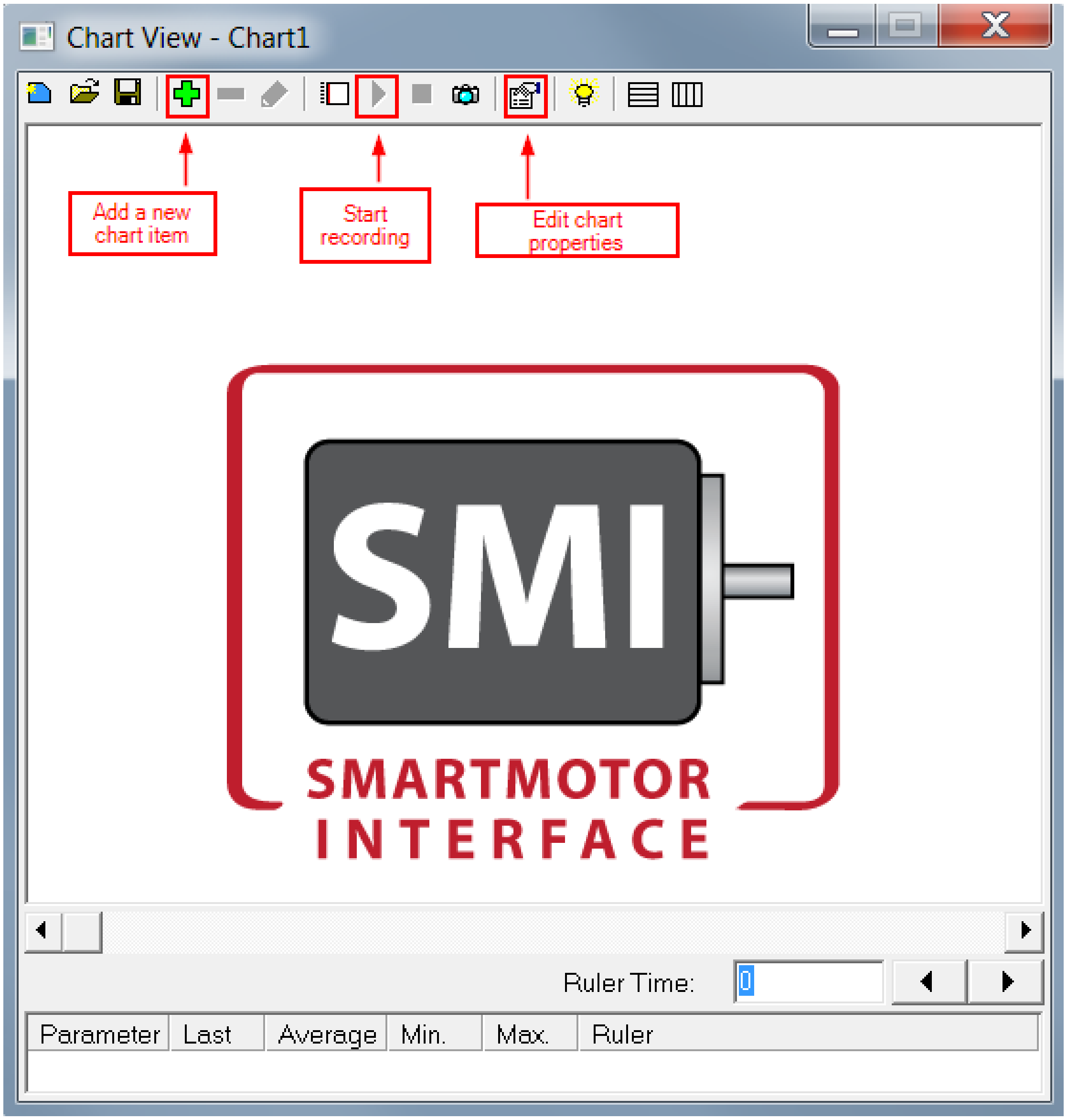


Chart View

Note the following:

- The Chart View is a sampling tool, which means it is possible certain events can occur within a single cycle and be missed by the tool. If this is suspected, adjust the sample rate in the Chart Properties.

- The samples taken by the Chart View are not obtained in real time, so there will be a slight deviation. This is especially noticeable at faster sampling rates. For a more accurate representation, sample the internal clock (CLK) on the motor and use that as a reference.

The Chart View is powerful because it allows users to see when specific events (including errors) occur, which can help in the troubleshooting process. For example,

- Getting an overcurrent error during the acceleration phase of a move suggests that the commanded acceleration is too high for the given loading.
- Getting an over current error during the slew phase of a move suggests the motor is undersized, or perhaps there are some inefficiencies in the mechanics of the system.

To get a better understanding of how this works in a real world application, see the Case Study: Using SmartMotor™ Data to Diagnose Linear Actuator Performance Problems.

NOTE: A useful feature of the Chart View is the ability to export the data in text format for detailed analysis in other applications. To do this, simply right click anywhere in the chart area and select "Copy Data". It can then be pasted in a spreadsheet, such as Microsoft® Excel, or any text editor.

Relevant Troubleshooting Commands

This section highlights some commonly overlooked commands that can be useful when trying to troubleshoot applications. For a full description of these commands, including examples, please refer to the *Moog Animatics SmartMotor™ Developer's Guide*.

ECS – Encoder Count Shift: When using an external encoder, this command allows you to simulate encoder counts.

CAUTION: The effect of ECS is immediate and can cause extremely large accelerations when gearing or camming.

Therefore, it is recommended that only small values are used. If larger values are needed, issue the command multiple times with small values. For example, if a shift of 50 counts is needed, use a value of 5 and issue the command 10 times with a small delay (e.g., 1-10ms) between each one. If this creates jerky motion or causes overcurrent warnings, increase the delay time and/or decrease the shift size.

NOTE: It is possible to get motor faults with the ECS command. Consider the following application example:

- The application is gearing, compensation for slipping is done with ECS
- Normal operation runs a following error of 150 counts (i.e., REA reports an average of 150)
- EL is set to 300

ECS=150 is issued and the motor faults out immediately because EA+ECS (150+150) reaches the error limit (EL=300).

FSA – Fault Stop Action: This command gives you some degree of control over how faults and limit switches are handled. FSA(0,0) is the default configuration, which sets the fault action of all types of faults to result in Mode Torque Brake (MTB).

MDT, MDE, MDS, - Commutation Mode: The default and available modes vary depending on model. One of the largest differences between these modes is the feedback method used for commutation (not position/velocity).

- In MDT, the hall sensors are used for feedback. This provides a rough estimation of rotor location and applies current in a trapezoidal profile. This results in some slight cogging at low speeds.
- In MDS, the encoder is used for commutation feedback and provides a much more precise estimation of rotor location. In this mode, current is applied in a sinusoidal profile and reduces cogging.
- MDE relies on the encoder but still uses a trapezoidal profile. Trajectory Overshoot Braking (TOB) is a feature that requires a more precise knowledge of the rotor position and, therefore, requires the motor to use MDE/MDS.

MINV – Mode Inverse: This allows users to reverse the direction convention of the motor.

O – Origin: This command, commonly used during homing routines, allows you to set/reset the current position to any value.

OSH – Origin Shift: This command allows users to perform a position offset on the fly without changing the current position orientation. This can be useful if the application requires multiple coordinate systems.

S, X – Stop, Decelerate to Stop: During normal operation, the X command should be used for stopping, which will cause the motor to decelerate to a stop. The S command will cause the motor to stop as fast as physically possible. Depending on the application, improper use of the S command can cause large spikes in back-EMF and subject the mechanics to very high torques.

Revision History

October 2018

- Initial release

November 2018

- Improved linear system diagram in Status Word 6
- Minor surface edits

April 2020

- Modified covers and fonts to Moog IG format.

NOTES

TAKE A CLOSER LOOK

Moog solutions for a wide variety of applications, including medical, office automation, packaging, industrial, aerospace and defense are only a click away. Visit our worldwide web site for more information.

Americas - West
Moog Animatics
2581 Leghorn Street
Mountain View, CA94043
United States

Tel: +1 650-960-4215
www.animatics.com

Americas - East
Moog Animatics
1995 NC Hwy 141
Murphy, NC 28906
United States

Tel: +1 828-837-5115
www.animatics.com

Europe
Moog GmbH
Allgaeustr. 8a
87766 Memmingerberg
Germany

Tel: +49 (0) 8331 98 480-0
www.animatics.de

Asia
Moog Animatics
Kichijoji Nagatani City Plaza 4F
1-20-1, Kichijojihoncho
Musashino-Shi, Tokyo 180-0004
Japan

Tel: +81 (0) 422 201251
www.animatics.jp

For product information, visit **www.animatics.com**

For more information or the office nearest you, contact us online, **animatics_sales@moog.com**

Moog Animatics Troubleshooting with Status Bits and SMI Tools Application Note, Revised: 04/28/2020
PN: SC801000??-001