

Downloading and Uploading Programs

Why will SMI2 not let me download to a SmartMotor™ that is moving or running a program?

Due to safety concerns, the new SMI2 software forces you to turn off the SmartMotor™ holding current or stop a running program prior to download to prevent possible unexpected motion. Let's suppose you have a program that places the SmartMotor™ in Velocity mode or Torque Mode and then you try to download without first tuning off the motor. During travel, it will crash into the end stop while downloading. For safe operation we want to ensure the SmartMotor is in the OFF condition.

When I start a download, the motor stops everything it was doing. Why?

When SMI starts a download of a new program, it issues the END command to stop prior code from running. This is to prevent processor memory pointer errors while the EEPROM header portion is being re-written. It is also for safety reasons.

Is there a way to prevent someone from uploading a program?

Yes. You can protect your program from being seen or copied by sending LOCKP to the motor via serial port. The LOCKP command is a means of locking the program. It does not prevent a user from downloading a new program, but it does prevent them from seeing the program you have downloaded.

I/O Handling

Since the I/O is non-isolated 5 VTTL, are there any options for 24 VDC I/O?

Yes, there are a few options Animatics provides cables with built in 5 V to 24 V isolated logic circuits right in the connector hood. This allows the user to have a choice of either 4 Inputs and 3 Outputs or 5 Inputs and 2 Outputs at 24 VDC. They can be set as sinking or sourcing. The CBLIO-ISO1 series cable can be purchased in lengths from 3 to 10 meters. The DINIO7 is a DINrail mount breakout board that also provides a means of isolation using Industry Standard Opto-22, Gordos, or Grahill I/O modules such as ODC, IDC, OAC and IAC series. The DINIO7 also allows interconnection to other motors and their I/O via a built-in back plane.

Are there any Expanded I/O options?

Yes. Each Class 5 SmartMotor has the-AD1 option providing an additional 10 channels of isolated 24V sourcing I/O. This option will appear as an additional 12 pin 12mm connector where 24VDC must be supplied to make it work. Each channel may be configured as Sourcing Input, Sourcing Output, or 10 bit analog input. As analog inputs, they may be read over the range of 0-5VDC or zero to full supply rail. The Analog reading returns either raw binary value or milivolts directly.

All I/O pins are reverse voltage, over current, and short circuit

protected. A fault on an I/O pin requires a down power reset.

What type of electrical protection/isolation does each I/O pin have?

Each I/O pin has a 100 Ohm series current-limit resistor tied to a 5.6 VDC over-voltage limiting zener diode. The user ties into the 100 Ohm resistor directly. The connection between the 100 Ohm resistor and zener diode ties in directly to the CPU pin. This is why the motors are limited to 5 VTTL I/O logic levels only. Animatics does, however, provide 24 VDC I/O adapters and adapter cables for converting the 5 VDC I/O to optically isolated 24 VDC logic for connection to PLCs and other equipment. Please visit the Animatics website and search under Cables and Accessories.

Power Supplies

Which is better, Linear or Switcher Supplies?

Linear supplies are better suited for inductive loads. SmartMotor™ servos are inductive loads. Linear supplies can handle high current surges typically caused by starting and stopping of servo motors.

However, linear supplies have what is known as voltage droop.

This is characteristic of voltage dropping down with an increase in load. Typically, unregulated toroidal transformer supplies will drop 4 to 7% and E-Core types (the big square transformers) are >10%. Switchers have no voltage droop until they reach maximum load. Then they just drop completely to zero volts.

However, since they maintain a tight control over voltage up to the trip point, they can typically aid greatly in reaching maximum speed and acceleration of a given servo. However, the switching supply must be sized for the maximum expected peak current draw of the motor system. A linear supply only needs to be sized for continuous load. Linear supplies have a large capacitance to supply much higher current surges when needed, so this is more of an application specific question.

What concerns are there with maximizing voltage on the supply?

The higher the voltage, the faster the SmartMotor can move and the faster it can accelerate. This is a good thing. However, in conjunction with this, the higher the voltage, the closer you get to a peak voltage that can cause over-voltage breakdown of the controller. Also, the higher the voltage, the faster a rate of change of current can occur. It is a risk with any application to get faster response by moving towards a higher voltage.

Typically speaking, it is the dynamics of sudden changes that increases risk by a "x^2" factor whereas the continuous load risk is only a direct ratio increase. This is because rate-of-change in current is proportional to acceleration which is the square of velocity, i.e. x^2. For safety sake, a 42 VDC supply for a 48 VDC system gives good margin with little speed losses.

How do I size Power Supplies?

The quick answer is "more is better".

First, be sure you have the correct SmartMotor for the job. Once that has been done, take the nominal power rating for that motor and you should size a LINEAR supply to provide about 10% more power to allow for longer sustained current loads. Any LINEAR supply will typically provide more than enough peak current. This is where sizing gets tricky with Switch-Mode Power supplies. "SWITCHERS" typically come with some rated voltage and current.

For example: 48 VDC at 6 Amps.

Well, that is just it...this can provide 48 VDC nailed to the wall all the way up to 6 Amps, but if any more current is applied, the power supply will drop out to zero VDC and typically reset. Any time you wish to use switchers, you need to take the peak expected load of the motor and size the switcher's continuous rating for that contingency.

As a rule of thumb, any 23 NEMA Frame SmartMotor™ can pull as much as 12 Amps instantaneous. Most 23 Frame SmartMotor servos will not pull more than about 9 Amps instantaneous. A 10 Amp switcher can supply any 23 frame SmartMotor for MOST applications. With a 34 frame SmartMotor will require a 20 Amp continuous rating to be sure you will not get a tripped power supply. 34 frame SmartMotor can pull as much as 40 Amps or more for a few milliseconds, so as originally stated: "Bigger is better," especially when it comes to switchers.

Back EMF

How do I protect against Back EMF?

Short Answer: Don't back-drive the SmartMotor.

The problem is back-driving the SmartMotor isn't the only means to produce Back EMF. Read the FAQ on "What is Back EMF and where does it come from?" The best way to protect against Back EMF is to use a shunt such as the active 48 VDC 100 Watt shunt supplied by Animatics. It drops a 100 Watt load onto the bus any time bus voltage exceeds 49.5 VDC. It removes the load when bus voltage goes back down below 46 VDC.

It will work with Switching or Linear supplies as long as no-load voltage does not stay above 48 VDC. Otherwise the shunt will be on all the time. Another method of protection is to use a mechanical brake controlled by the brake commands in the motor. The SmartMotor can respond to a fault and send a signal to the brake within 250µseconds to help hold the shaft from back driving. None of these ideas help against hitting a hard stop. Please read the next FAQ for more.

What is Back EMF and where does it come from?

Back EMF is the voltage generated when a rotor is moving within the stator of any motor. It is literally the motor acting as

a generator. There is a common rule that Back EMF or voltage generated is proportional to 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, RPM of the motor shaft does not have to be that high to have very high voltages created.

Take any relay coil or solenoid valve coil in a 24 VDC system for example. When it is energized, the magnetic field pulls in the contactor or pilot valve. The magnetic flux reaches saturation and a DC electromagnet is then formed. When the power is removed from the coil, the magnetic flux rapidly collapses because there is no forward voltage to maintain it. Since the circuit is not electrically open, there is nothing to prevent the magnetic flux from collapsing rapidly at a hyperbolic rate. The result is something called "inductive-kick".

The "The inductive-kick" or spike in voltage for a 24 VDC coil can reach very high voltages and currents on the order of 100 times that of the original applied voltage, i.e. 2400 VDC! This is why it is very common to place reverse polarity diodes across relay coils and solenoid valve coils. It protects the system from high voltage spikes. The same thing occurs when a SmartMotor hits a hard stop. Suddenly, the rate of change of magnetic flux in the stator windings skyrockets upward because the rotor stopped moving. This sudden change causes an excessive voltage and current spike in the controller and can damage components.

What can we do about it? Practically speaking, not much. We could design the drive stage to be able to take the hit of a fast hard stop, but the drive stage would be very large. The controller would have a lot more components in it and the practicality of it would be diminished. The SmartMotor would grow in size for the same torque output to three times larger. This is just not practical.

I was told the SmartMotor™ failed due to over-voltage, but I never back drove it or ran it fast...

HARD STOP CRASHES:

The best recommendation for preventing damage to the SmartMotor/controller in the case of hitting a hard stop is to place a limit switch near the hard stop that trips the motor off line just prior to hitting the stop. The best way to prevent it beyond that is to prevent the cause of hitting the hard stop in the first place.

If this is due to jogging the SmartMotor in Velocity Mode and not letting the jog switch in time, then jog in Position Mode instead and use the "X" or "S" command to stop the SmartMotor when the jog switch is released. In any case, ***much care should be taken to be sure the SmartMotor is not intentionally or unintentionally allowed to hit a hard stop while under normal speeds and load conditions.***

 **Warning: Improper Power Supply Sizing may result in Motor Position Error Faults, Motor Resets, and Machine Faults** 

Communications

I have a SmartMotor that will not communicate no matter what I try. What should I do?

It is possible to unintentionally write and download a program that will lock up the CPU or prevent serial communications. If you power it up and there is one or more LED lit up, then try the following: Isolate the SmartMotor such that you have a single motor power and communications cable between the SmartMotor and the PC. Connect the motor to the PC serial port, but DO NOT power up the SmartMotor. In other words, have the power supply either disconnected or turned off. Then start up the SMI software and click on the TOOLS drop-down menu.

Under TOOLS, click on "Communications Lock-up Recovery" and a pop-up window will tell you to do what is described above. Then click NEXT. The Lock-up recovery utility will begin transmitting multiple "E" characters to the motor. It will tell you to power-up the SmartMotor at this point.

Then it will wait about one second and attempt to establish communications. If it does establish communications, it will tell you and then prompt you to either clear the program or upload a new program. At this point, it is advisable to clear the EEPROM to down-power reset the motor and reestablish communications normally. This way you will know if it was just a program issue or a hardware issue.

If you apply power and there are no LEDs lit up on the motor, then there is a major problem with it electrically. It will have to be sent back for repair.

Occasionally I lose communications and don't know why. What could be causing this?

Here are some general things to consider:

- Shielding and grounding must be done properly to insure good signal integrity.
- Long character strings should be avoided. The receive buffer is only 16 bytes long.
- Never use the shield as the ground reference connection for RS-232 or RS-485.
- Make sure there are no non-terminating strings being transmitted. If a string is not followed by a carriage return or space character, the SmartMotor will hang indefinitely while waiting on the terminating character. It is done this way to allow priority to the serial ports over any downloaded program execution.

How far can I transmit on RS-232 or RS-485?

"RS" in the RS-232 and RS-485 specification means "Recommended Standard." Not every company or chip manufacturer actually meets the "RS." The IEEE specification states that RS-232 single-ended signal is SUPPOSED to be +/-12 VDC or a 24 VDC swing from logic zero to logic 1. The specification for RS-485 is +/-5 VDC differential. By voltage levels, RS-232 will logically be able to transmit much further.

By noise immunity, a differential signal should be able to reach its destination "cleaner". The reality is similar to the following analogy: a clean whisper can't be heard a mile away, but a loud guttural voice can. RS-232 can transmit further but RS-485 transmits cleaner.

Also note that the higher the voltage level, the higher the induced noise must be to overcome the signal, so RS-232 isn't so bad after all. As far as actual distance goes, there are applications in the field running 250 feet on RS-232. RS-485 can't drive the cables well enough to go beyond 100 feet without experiencing loading problems. Also, RS-485 is a parallel bus. The more motors you add, the shorter the overall practical distance due to bus loading. RS-232 is serial. Therefore, one motor transmits directly to only one "receive" buffer. This means compounded bus loading does not occur, another reason RS-232 can actually transmit further.

I'm using Profibus between multiple SmartMotors and the communication sometimes drops out or does not initialize in the first place. What might be the cause?

First, Check your baud rate and cable lengths between nodes:

To use baud rates at or above 1.5 Mbps, special connectors ARE required. These connectors must have built in inductors in order to run these higher rates over Profibus. Additionally, the minimum cable length between two node MUST BE 1M (3 feet). If the cable is too short, reflected impedance can cause loss of communications data packets and spurious node errors. Branch lines are not permitted when using Baud rates greater than 1.5 Mbps. For more about Profibus and the SmartMotor, see page 23.