

The following few pages are actual examples of tech support request. The questions are specific in nature, but may be similar to other cases that may come up. The answers have been tweaked to give a more general market aid in customer troubleshooting.

Q If I limit the SmartMotor with the AMPS commands, would this prevent an over current occurred fault?
(clarification: they are referring to the peak over current bit since few know the Bh bit is also RMS over current)

A The AMPS command is strictly to limit Continuous equivalent RMS current and has no effect on peak. There is no direct means to limit peak current short of lowering commanded acceleration or moment of inertia mismatch.

Smartmotors are a True fully digital PWM drive. This means the voltage from the supply is truly switched hard on or hard off. The PWM tells the duration of the time it is switched on (hence equivalent RMS current) , but while switched on, full bus voltage reaches the windings. There is no means to reduce the full peak that can be made available due to true digital switching.

Simply put:

If you turn on a light switch , the bulb gets 120VAC. If you turn it off, it gets nothing. No matter how short the duration of the "on-time", the bulb still gets the full 120VAC. This is a fundamental and often over looked aspect of true 100% digital PWM control.

More detail on this:

If you increase load inertia, then the motor takes longer to get up to speed. As a result, it takes longer to produce Back EMF. Back EMF is the ONLY real means to reduce peak current in an installed system.

The only means to get more back EMF is to get to a higher speed.

If the load prevents you from speeding up then peak current will be sustained longer while trying to get up to speed. If you command a higher acceleration, then you are commanding a longer on-time PWM cycleWHILE STILL at lower speed, hence more Peak current on-time.

By lowering commanded acceleration, the on-time is reduced, so the peak current does not exist for as much time. But in both cases above, the peak current is still there, it is just there for a different amount of time. One last note: Some will ask why they get peak over current when "we were not telling it to go that fast".

As stated above, Back EMF limits peak current. the slower you go, the higher the peak current goes. It is a basic property of inductive loads.

Q Why will my Smart Motor not run smoothly at very low speeds.

A There are a few reasons, for this. The basic means to control commutation is one limit. SmartMotors use 6 step trapezoidal commutation. As a result, within a single phase, the motor gives no specific shaft angle back to the drive. In so being, the controller will give linear PWM rates within that phase, vice the more natural sinusoidal Back EMF that it must work against. This tends to show itself as cogging torque at lower speed.

Resolvability with regards to speed detection is another factor in speed control. The SmartMotors typically scan shaft position every ~250 microseconds or about 4000 times per second. Lets suppose you need to move at a rate of 1RPM and have a 4000 count per rev feedback. That equates to 1 encoder count per every 250 microseconds or 1 count per servo sample. At this low of a speed, even a slight change in load will have a major effect on the rate of change per sample. The controller expects to see 1 count per sample, but if it gets behind the slighted bit, it may not see any counts for 2 or 3 samples, the result is a quick increase in commanded PWM to the drive. The motor then jerks forward only to cause multiple counts per the next PID sample. This is seen as speed ripple which cannot be overcome without some change in the application design. The two methods to gain higher resolvability are higher resolution encoders or additional gear ratio. If higher resolution feedback is available , this is a better choice. The reason being is, it does not limit over all top speed of the servo motor. Adding gear ratio does add multiply torque, but also takes away from top speed.

Q Why are the values wrong or corrupt in the Motor View polling window?

A This may occur if a user program has Report commands or PRINT command and is running those commands while polling from the SMI software. The reason for this is as follows: The SMI software polls the motors by sending non-synchronous commands to the RS-232 port. If the motor program happens to execute a Report command at the same time the software sends a command, then an unexpected result may occur because of the timing of the motor response. As a result, the polling window or motor view will show spurious characters or unexpected results. In some cases the SMI software will actually throw up an error and ask for input.

While polling with the SMI software, either disable or take out your print statements in the motor program. It may be possible to just issue the "SILENT" command. This will disable motor responses generated internally.

Q I can't get communications through multiple motors on a serial daisy chain.

A There may be a few reason for this. First, check for proper cable connections and wiring.

Once done, make sure there is power (LED's lit up) on each motor. Finally, make sure each motor has no program running in it that may be printing data or reporting data. If this is the case, the SMI software will be unable to establish communications. Also, if there are multiple motors with differences in baud rate due to downloaded programs, then SMI will not be able to establish communications at all. IF you are unsure of this, it is best to connect to each motor individually and remove or modify individual programs to have default baud rate.

Q I have gotten my smart motor into some kind of a loop where it will not communicate anymore with my programming laptop. Everything was fine during setup. The smart motor ran a demo program from the manual, so I know that it is my program that is bad. Is there a way to reset or delete a program in the smart motor when it will not communicate with my laptop. The green led on the right is lit the led on the left is off.

A Use Communications Lockup recovery. This will prevent the downloaded program from running at power-up thereby allowing communications to be established.

Q We have a tool down in the field for a few days now and need your advice. The SmartMotor keeps sending the Fault 2294 to our controller. The alignment on the z-axis assembly is good and we re-calibrated the SmartMotor using the "autoteach". Do you have any ideas before we replace the SmartMotor.

A Customer used complete custom data mode program. We had no means to troubleshoot. In a case like this, we would need the complete program and full instruction on how it was intended to be used. The time it takes in such cases is sometimes cost prohibitive.

Q Hi, I have a question about the SMI. In our application, we need the motor to be controlled by a time parameter. For example, we need keep the motor running for 60 seconds. How can we do that using SMI? If not, what should we do to make that task done?

A SMI was being used as a control tool for a task, It was suggested to write a user program with a move profile and WAIT command to achieve the proper delay.

Example:
MV
V=123456
A=1234
G
WAIT=60*4069 (wait for ~60 seconds)
X (decelerate to a stop)

Q I am getting errors that I trap and kick me out of my program via the Bp Real-time positive travel-limit and once in a while, Be Excessive position error. I am using Port A as an analog input for speed, Port B as a digital input for enabling the drive and just recently added port C as another analog input for tapering, I believe the native uses for the port inputs are causing the issue, i.e. limit switch function. How can I disable native port assignments so I can use the ports for what I want.

Issue UCI to disable Left Travel Limit, Note: SMI Playground DOES change the way limit switches work and may cause issues when testing.

The subject motor is used in our auto-throttle application to drive the aircraft throttles to a position commanded by the Automatic Flight Control System to maintain an airspeed selected by the pilot. On several occasions, the motor will sense an overheat condition and will shutdown. The motor is powered by a 28 volt power supply and the current limit (AMPS) is set to 500 mA. Nominal operating temperature of the motor (as reported by the motor's TEMP variable) is between 32 and 34 deg C. When the problem occurs, the motor temperature seems to rise above 70 deg C. within a fraction of a second. We've also encountered occasions where the motor will report an overheat condition to our application software (at which case the motor shuts down) and then returns to normal operation within a 1 to 2 seconds. We have the following questions:

1. Is it possible for the motor's temperature to increase 38 deg in a fraction of a second?
2. What could cause this rapid temperature rise?
3. What would cause the motor to report an overheat condition but then return to normal operations within a couple of seconds?

Submit: Submit Form

A The Bh bit is a shared status bit with RMS over current. Most likely you are not actually reaching 70Deg.C. You are more than likely reaching the continuous RMS over current condition.

Two ways to tell:

1. You monitor temperature immediately after the fault and find it <65.
2. You are unable to start until it is less than 65.

In case 1: The motor never reached 70. Therefore it is RMS over current.

In Case 2: It did reach 70. The processor will not allow a restart unless the motor cools down to 65.

Note: Loss of power just after Bh bit will bypass the 5 degree point because the processor will have lost knowledge of the overheat condition.

Q While jogging the motors in Playground, they seemed to work fine. But after exiting the Playground, the motors would not run their program properly. Why?

A The playground changes travel limit settings and programmable software limits as well. Be sure to check for travel limit faults this is typically the cause.

Q I am using the 2310D motor and trying to program through SMI. I want the motor to run at an exact speed. The motor has a 2000 line/rev. encoder and I need to use a decimal conversion factor through SMI to get the required speed. It looks like the format is ASCII and will not allow a decimal factor for this. Is there another way to do this. i.e. the conversion factor required for velocity on the encoder is 542.29 and I am only able to round to the nearest integer 542 or 543. That equates to a speed inaccuracy of 67 RPM while trying to drive the motor at 6000 RPM. Can you help with this?

A (actual email answer)

To be sure I did not misunderstand, could you tell me where the 542 and 543 came from?

As I understand it, you need to run at 6000 RPM. This equates to a commanded velocity of: $V=3221258$ Even though you cannot send a floating point number, the total value is very large. +/- 1 on that number would be: $\sim .000031\%$ deviation. This means the commanded speed would be within a very small fraction of $<0.001\text{RPM}$. I am not sure where +/-67RPM and 542 came from. But maybe I missed something from our automated email system.

Q We have several SM2315DTmotors. How do we disable the electronic thermostat? On the new SM2316DT can we control the overload protection circuits. We only operate in the overload condition for about 80seconds. We are not concerned with long life.

A The 2316DT has the newer PLS firmware. The older SM2315D does not. As a result, you would not be able to do so. Also you may note the 2315Da d 2315DT trip at 70Deg.C. The new SM2316 series trip at 85Deg. C. The TH command controls the trip point i can only be lowered below max design value, not raised any higher. The values are 70 or 85 as above.

The THD command tells the number of CPU cycles the motor can be at RMS over current prior to tripping. It's max value can be 65535. The default it 12000. That will give you some room.

The SM2316 series additionally has another RMS control that can be totally disabled. It would allow about 15 seconds at full rated peak before reaching the thermal limit and shutting down at 85 either way.

Q I have upgraded from 2315 series to 2316 series and now my machine doesn't work. Why Not?

A The 2316's all ship with PLS version firmware. As a result, the travel limit switches default to Active high to stop vice Active Low. If you are not using limit switches in your old application, then the new motors will fault on travel limit immediately and not allow motion,

If this has been taken care of, the PLS firmware also has force-reset on fault meaning any protection fault will issue "END" and not allow another G (go) until the fault status bit is cleared.

Insure proper fault handing routines are used.

Q Motor stopped working, no longer responds to commands

A Check LED's, If LED's are not lit up, check cable and power supply connectors. Use voltmeter and check sully at connector pins at cable. If still no LED's the controller may be damaged. It should be sent back for RMA repairs.

Q The motor is responsive to commands but will not move. It gets a position error immediately. What is wrong?

A First, establishing communications, see if manual rotation of the shaft is detected. If position updates as would be expected, then the encoder is working, if not, then the encoder may be at fault. If that is the case, typically, the motor shaft will jerk for a moment when a move is commanded. If the encoder is working, test the motor in Torque Mode (open Loop)

Issue MT and T=100 or so to see if the shaft moves. If it does not, insure there is no excessive load on the shaft (test unloaded). If the shaft still does not move, then see if a Peak Over current bit (Bs) is on. If so, the current-sense resistor may be damaged. To check this, command T=500 or so. Look for Ba bit and if it is a 1 and the shaft did not move, manually rotate the shaft back and forth, If it freely spins in both directions and yet the motor still reports being in Torque Mode, then the current sense resistor is blown open.

Q After downloading the program, The SMI software always ops up a window that asked for RUN RESET or CLOSE. When I click RESET, I lose motor communications. Why?

A The RESET option issues “Z” which is the same as doing a down power reset. As a result, if the polling windows is up and running, the software will lose communications when the CPU resets. It is better to choose “RUN” or CLOSE if the polling window is being used.

Q I have a motor with both Brake Option and DE. I cannot get the brake to release. Why?”

A The DE option means you must have both drive and control power to allow any motion. The brake option gets it's power form the drive side, but is controlled by the control power. If drive power is not available, the brake will be re-engaged and the shaft will be locked.

Q I keep getting loss of communication s randomly .

A Without going into the vast details of why this may be happening, here are things to consider:

Is it a USB to RS-232 converter? If so, go into Windows Device Manager, find your specific com port. Go into advanced settings and reduce buffers and timeout delays to the smallest value possible. The reason for this is the way the windows operating system interrupts programs to tell them messages have been received. The larger the buffer size, the longer a delay before software gets interrupted.

Check all cables, grounding and shielding. One common issue is when the PC and machine are on separate grounds, The difference between them can cause noise or completely suppress the communications signal.

Long PRINT commands or transmitted command strings can cause buffer overflows in the motors. Be sure to add delays between long strings to allow process time in each motor.

Q My application requires continuous index moves in one direction all day long,. For no apparent reason it error out after many ours of operation. The only means to recover seems to be a down power reset. Why?

A Most likely the motor got a wrap status error meaning the shaft position exceeded $\pm 2^{31}$ in magnitude. When it passed through the maximum count, it would fail on position error. A quick means to fix this in indexing applications is to reset the counter after a TWAIT after each move.

Q My program runs fine, but after a few cycles, it just stops and becomes unresponsive.

A Look for the potential of stack overflows and math errors. If the number of cycles it runs is 6 and then stops, then the stack overflow is the most likely error. If so, check for recursive subroutine calls. This is the most likely reason. If it appears to run a very long time and then errors out, it may be a math overflow error. If so, then check for math multiples that exceed the bit width of the variables being used.