

## Motor Temperature Sensing And Control

Associated QuickControl® programs included:  
 Motor Temp Limit.qcp

This application note describes the motor temperature sensing capabilities of the IP65 motor series when coupled to a SilverDust I-Grade controller.

### Hardware

The I-Grade SilverDust controllers have the ability to read the motor temperature sensor located within IP-65 I-Grade Motors. This temperature sensor may be used to monitor motor temperature, to limit motor temperature by causing a kill motor recovery, or to keep the motor above the dew-point temperature to extend its motor life when operating in refrigerators or other cold or potentially cold environments.

### Measuring Motor Temperature

The lower word of Register 241 is updated with the motor temperature several times a second. The temperature sensor provides a digital output of 16 counts per degree Centigrade, with a zero degree output of zero counts. Temperatures below zero result in negative values.

### Limiting Motor Temperature

The upper word of Register 241 defaults to zero (0) disabling the motor over-temperature checking. To enable the over temperature checking, write 16 \* Max\_Temperature to the upper word of Register 241. Bit 4 in IS2 will be set if the motor temperature exceeds the threshold temperature. The **Kill Motor Extended (KMX)** command must also be used, with *the Motor Temperature* bit set to *Fault* to trigger a kill motor shutdown/recovery when the motor temperature exceeds the selected threshold.

Motor Temp Limit.qcp example, to the right demonstrates a 100°C limit with a fault for motor temperature enabled.

Line# Oper	Label	Command
1:REM		IP65 grade motors include a temperature sensor in the motor.  The Temperature reading is available in the lower half of Register 241. It is in degrees C* 16 (1/16 C increments)  The motor temperature shutoff temperature may be set by writing the limit temperature * 16 to the upper half of register 241.
2:REM		Limit motor to 100 C
3:WRP		Write 1600 to "Accumulator[10]" Register
4:CLC		High Word Motor Temp Max   Motor Temp[241] = LO(Accumulator[10])
5:KMX		Kill Motor Conditions Extended: ISw: IS2: If Motor Temp Fault XIO:

## Controlling Minimum Motor Temperature

The operational life of the motor in a high-humidity or potential condensation environment (such as a refrigerator / freezer which may suddenly warm) may be greatly extended by keeping the motor above the dew-point temperature. This feature may also be used to keep a motor warm in cold environments in which the environment would otherwise exceed our lower temperature limits.

Operating the motor consistently above the average dew point prevents the continued build up of condensation which may accelerate corrosion or even flood the electronics section. Note: according to the degree of heating desired, thermal insulation may be required.

The power into the motor is controllable when the motor is not in motion by enabling Anti-Hunt and varying the holding current levels. The windings of the motor are then essentially operating as resistors, with four times as much heat being generated for each doubling of the current. Motion commands automatically cause the system to revert to closed loop and normal operation. Between motions, Anti-Hunt is again operable. If there is sufficient time between moves to supply the needed heat, a thermal control loop may be implemented. This is most easily implemented using the second thread of execution to monitor the motor temperature against the desired temperature, adjusting the holding currents to correct the temperature (we can only add heat). The primary thread may also be used easily if the unit is primarily using a Profile Move Continuous (PMC) mode of operation with position, velocity, and /or acceleration information being written to the appropriate registers by a secondary computer or HMI.

"Reg Temp One Thread.qcp" implements the heating process inside of Thread 1, using the PMC command for motion, as already described. "Reg temp two threads.qcp" uses the second processing thread to control motor temperature, leaving the main motion thread open for the user to run their own motion processing program.

Both programs compare the set point temperature to the measured temperature (17 and 23 frame IP-65 motors used with QCI-D2-IGx SilverDust), setting the open loop torques to 100% if less than the set point, and reducing the holding torque linearly between 100% and 30% over approximately a 5°C interval. This varies the heating from 100% down to 9%, as the heating is a square function of current ( $P=I^2R$ ).

The motor temperature is also available in 0.1 degree increments via register 34 (i.e. 25.9 degrees reads as 259). This allows easy viewing without manual conversion via the Register Watch tool.

**Reg Temp One Thread.qcp**

Line 4 enables Anti-Hunt, which is required to allow the motor to operate in open loop mode (and thus provide controllable heating) when the motor is not actively in motion. Lines 6 through 10 configure initial conditions for the PMC command in line 12 of the program. Line 11 has the EMT command which allows the temperature control loop to operate while the PMC continues to run in the background.

Line 14 sets the desired set point temperature to 30°C. This maintains 100% torque (and power) until the motor temperature sensor reads 30°C, and then reduces it down to 30% torque (9% heating) between 30C and 35C. The torque is not reduced below 30% to allow for effective holding torque for the motion operation.

Line# Oper	Label	Command
1:REM		
2:REM		We will control the motor temperature while running a profile move continuous. Writing a desired position to register 20 will cause the motion with the velocity and accel/decel configured.  Register 30 will hold the desired motor temperature.
3:REM		Make sure antihunt is enabled
4:AHC		Anti-Hunt Constants: Out of Anti-Hunt Error=25 counts Into Anti-Hunt Error=15 counts
5:REM		Set up a profiled move
6:WRP		Write 0 to "User or Profile Move Pos[20]" Register
7:WRP		Write 200000.49 cps/s to "User or Profile Move Acc[21]" Register
8:WRP		Write 120000 cps to "User or Profile Move Vel[22]" Register
9:WRP		Write 200000.49 cps/s to "User or Profile Move Dec[23]" Register
10:WRP		Write 0 to "User or Profile Move Offset[24]" Register
11:EMT		Enable Multi-Tasking
12:PMC		Profile Move Continuous:
13:REM		Set up the desired control temperature in degrees C
14:WRP		Write 30 to "Set Point Temp[30]" Register
15:REM		Raw temperature is in 1/16 degrees, so we will scale up the setpoint
16:CLD	T-LOOP	high res set point[31] = Set Point Temp[30] * 16
17:CLX		high resolution motor temp[32] = LO(Motor Temp Max   Motor Temp[241])
18:CLX		High res error[33] = high resolution motor temp[32] - high res set point[31]
19:REM		We will vary the holding current over a 5 degree interval such that the torque will be 20,000 (100%) at the setpoint and will drop to a 30% holding current (9% power level) over a bit less than a 5 degree interval above the set point temperature.
20:CLD		Accumulator[10] = High res error[33] * 175
21:CLD		Accumulator[10] = 20000 - Accumulator[10]
22:REM		Set the minimum holding current here
23:CLD		Accumulator[10] = Max of Accumulator[10] or 6000
24:REM		Set the maximum heating current here
25:CLD		Accumulator[10] = Min of Accumulator[10] or 20000
26:REM		Use the torque setting for both running and holding open loop
27:CLX		Open Loop Torque Hold   Move[207] = LO(Accumulator[10])<<16 + LO(Accumulator[10])
28:REM		allow motor temp to be displayed in .1 degree increments
29:CLD		Motor temp in .1 C increments[34] = (high resolution motor temp[32] * 40960 )>>16
30:JMP		Jump to "T-LOOP"

**Reg Temp Two Threads.qcp:**

This program consists of two programs. The first sets up a PMC command, with a loop modifying the position, but other motions and code could be easily added here.

Line 6 determines which conditions are excluded from stopping the Thread 2. In this case, the Halt and Stop commands will not kill Thread 2 execution. The second thread is then started up by line 8.

The Thread 2 program measures the temperature and controls the holding torque, just as in the last example.

Line# Oper	Label	Command
1:REM		
2:REM		We will control the motor temperature by using thread 2 running in the background. Any motion or command may be setup in thread 1.
3:REM		Make sure antihunt is enabled so we can vary power by varying open loop currents
4:AHC		Anti-Hunt Constants: Out of Anti-Hunt Error=25 counts Into Anti-Hunt Error=15 counts
5:REM		Allow Thread 2 to continue even with a stop or halt command.
6:T2K		Thread 2 Kill Conditions: Halt Command[Bit 4] Excluded Stop Command[Bit 5] Excluded
7:REM		Now we will start up the temperature control program in the second thread, leaving thread 1 free to do a user motion program.
8:T2S		Start Thread 2 with Program = "Temperature loop"
9:EMT		Enable Multi-Tasking
10:REM		Here we do an example motion program
11:WRP		Write 0 to "User or Profile Move Pos[20]" Register
12:WRP		Write 200000.49 cps/s to "User or Profile Move Acc[21]" Register
13:WRP		Write 120000 cps to "User or Profile Move Vel[22]" Register
14:WRP		Write 200000.49 cps/s to "User or Profile Move Dec[23]" Register
15:WRP		Write 0 to "User or Profile Move Offset[24]" Register
16:PMC		Profile Move Continuous:
17:DLY	LOOP	Delay for 2000 mSec
18:ATR		Add 16000 to "User or Profile Move Pos[20]" Register
19:JMP		Jump to "LOOP"

Line# Oper	Label	Command
1:REM		Set up the desired control temperature in degrees C
2:WRP		Write 30 to "Set Point Temp[30]" Register
3:REM		Raw temperature is in 1/16 degrees, so we will scale up the setpoint
4:CLD	T-LOOP	high res set point[31] = Set Point Temp[30] * 16
5:CLX		high resolution motor temp[32] = LO(Motor Temp Max   Motor Temp[241])
6:CLX		High res error[33] = high resolution motor temp[32] - high res set point[31]
7:REM		We will vary the holding current over a 5 degree interval such that the torque will be 20,000 (100%) at the setpoint and will drop to a 30% holding current (9% power level) over a bit less than a 5 degree interval above the set point temperature.
8:CLD		Accumulator[10] = High res error[33] * 175
9:CLD		Accumulator[10] = 20000 - Accumulator[10]
10:REM		Set the minimum holding current here
11:CLD		Accumulator[10] = Max of Accumulator[10] or 6000
12:REM		Set the maximum heating current here
13:CLD		Accumulator[10] = Min of Accumulator[10] or 20000
14:REM		Use the torque setting for both running and holding open loop
15:CLX		Open Loop Torque Hold   Move[207] = LO(Accumulator[10]) < 16 + LO(Accumulator[10])
16:REM		allow motor temp to be displayed in .1 degree increments
17:CLD		Motor temp in .1 C increments[34] = (high resolution motor temp[32] * 40960) >> 16
18:JMP		Jump to "T-LOOP"
19:REM		