

SilverSterling S2-IGH and S3-IGH Using BO-BP8 Memory

Overview

The S2-IGH and S3-IGH controllers are designed to allow reading from the 1-wire™ devices. 1-wire memories are included in the BO-BP8 backplane (one memory per slot), and in QCI I-Grade motors. An optional 1-wire temperature sensor is available in QCI I-Grade motors; these devices may be added externally to 3rd party motors. (note “1-wire” is trademark of Maxim Semiconductor.) The controllers also have their own internal non-volatile memory used for calibration as well as for storage of initialization and user programs; Firmware is also present in the controllers for the code that runs the DSP.

The three available 1-wire devices each serve different purposes. The motor memories are included with I-Grade motors, but which can also be added to third party motors. These store motor serial number, motor type, and any motor calibration and alignment information to allow these motors to be identified to the controller automatically, and to be swapped without having to re-align them to a particular controller. The temperature sensors allow the actual motor temperature to be monitored. This temperature can also be used to trigger a controller kill-motor if the threshold is exceeded. The third 1-wire memory is located in the BO-BP8 backplane. This card cage memory is used to hold the axis specific information (show parameters) for a system. This includes motor type, motor configuration, tuning information, homing information, CAN ID, serial ID, desired maximum travel and travel distance, as well as homing information. A 64-bit cumulative motion counter is also stored, which is updated each time the units' fault or are powered down. The controllers, when used in this way, are all programmed with the same QCP (QuickSilver Controls Program), and so may be interchanged for troubleshooting without reprogramming and without the axis information being lost.

This shared program also has provisions for the advent that new axes may eventually require a later version of the card program, or of later firmware versions. The data written to the backplane memory thus includes a minimum revision date (YYYYMMDD format as a number) which the card must have for that axes to operate properly. The minimum firmware version is also checked against a card cage based requirement. This allows an error code to be reported via CAN and also flashed to the green LED if the program revision or firmware revision is not sufficient. This allows the spares to be updated to the latest versions so they can be used anywhere, while not requiring all controllers to be updated every time a change is needed.

For applications that do not use the BO-BP8 backplane, refer to Application Note:QCI-AN082. This application note demonstrates how to configure a SilverSterling S2 & S3 controller, SilverNugget X-series controller, and SilverMax X-series integrated servos to read show parameter data from either internal EEPROM or external 1-wire non-volatile memory on power-up. Once show parameter data is read, the controller configures itself to interface with a Weigl device via CANopen.

The rest of this technical document describes all show parameters in detail and steps through programming backplane the 1-wire memory.

Programming the Backplane Memory

The first operation in setting up this system is to configure the motor characteristics into the backplane memory. This is done using a spare S2-IGH (or S3-IGH if preferred for the bigger card cages). This controller can later be reprogrammed for normal usage once the backplane has been initialized.

The first step in this process is to define the function of each slot. It is suggested that a separate subdirectory is used for each 8 slot card cage, as the data that will be used to configure the memory is imported from a file "Table Axis 1-8.csv" in the same directory as the import directive file "Import Register Files.txt" which describes how to use the .csv file, and the QCP program "Write to card cage memory 20190906.qcp" used to do the actual writing of the data. The data is automatically downloaded from the .csv file into the controller when the QCP is downloaded. The card is then moved from slot to slot in the card cage, using QuickControl to instruct the controller to initialize the card cage memory. These steps will be covered in detail below.

Into a new directory associated with the selected 8- slot card cage, copy "Write to card cage memory 20190906.qcp", "Import Register Files.txt", "List of parametersMAA 20200121.xlsx", and "Table Axis 1-8.csv". Next, open "List of parametersMAA 20200121.xlsx", and select the Axis 1-8 data tab on the bottom. This is the user editable page. This has all of the current motor types currently configured. (Contact the factory to have additional entries added.) Move and/or copy the desired motor type starting from row 5 "Motor type" down to the last used row in that column to the desired slot location (Axis 1 .. Axis 8).

The light peach highlighted values are automatically calculated from the values below the table. The Light blue values are customizable fields for tuning and torque and speed, the white ones should not be modified (except to copy with the whole column for motor type). The Yellow one is the checksum for the column and is automatically calculated. The light green fields are spare and do not need to be modified. You will notice the last column to the right of Axis 8: this contains the register number which holds this information in the runtime "Run from card cage memory 20190920.qcp" program. The user can use this information to quickly modify the values and resave them. This will be detailed below.

Once each column in the first page has been modified, the whole file should now be saved with a new descriptive name and date in .xlsx format. Next, select the tab on the bottom which is labeled "Export 1-8". This second page has all of the numerical information from the first page formatted to be saved in csv format. To write this information, select **File -> SaveAs** and select "CSV UTF-8 (Comma delimited) (*.csv)" from the scroll down bar, and select "Table Axis 1-8.csv" (no quotes) as the file save name. Excel will ask you if you want to overwrite – say yes. Next it will give a warning that csv does not support multiple sheets and only the active sheet will be saved. Select "OK". Now close the spread sheet (X-out) WITHOUT saving. The file should have already been saved above with the new name. Saving again at this point, excel will try to overwrite the original file with a .csv version missing the first page!

The data for the whole 8 cage slot has now been assembled into the same directory. The next step is to download this information into the spare S2-IGH. It is suggested that all of the controllers be removed from the card cage so that there will not be any other controller with the same ID (16) as the S2-IGH being used to configure the card cage. Place the spare controller into any slot (or into the specific slot

to be updated if only one slot is being updated. The serial connection should not be connected to any other card cages with controllers that will be powered up when doing this process.

Start **QuickControl** and configure the proper serial port for your USB-485 converter **Setup=> Comm Port**. Our stand USB-485 converter typically defaults to COM3. On the top line of QuickControl, select **Tools=> Unknown Wizard** and then follow the steps (power on the cage, continue, select RS-485 as protocol). You should see axis 16 in the status window. Go to **File=> Open** "Write to card cage memory 20200213.qcp" program from the directory assigned (by you) for this card cage. This will wipe the initialization for any motors and will download all of the information for the eight-axes for this card cage. The download will take a little over a minute. QuickControl should pop-up a box indicating a successful download. Hit OK to clear the pop-up box.

Go to **Tools => Register Watch** click "Add Register" and scroll down to register 29 and hit "Add". Next select "Long" as display type. Repeat with register 30.

To initialize the card cage memory for the backplane slot into which the controller has been plugged, click on the "data" column for register 30 (in the register watch box) and type the slot number (1-8) corresponding to that slot. When the value in register 30 goes back to 0, the programming is complete for that slot. Power down the card cage, move the controller to the next slot. Power up and enter the slot number into register 30 for that slot, wait for completion and repeat for as many slots in that block as need initialization or updating.

Register 29 will show the status after the operation has completed. A value of 0=no error, a value of -1 indicates the selected axis is not defined, a value of -2 indicates the checksum is not consistent with the data.

Repeat this process for each backplane, with the data copied to a fresh directory for each backplane.

Spreadsheet data

The spreadsheet fields names are to the left of the actual data, with the actual register location to where this data is read into the runtime program the left column. The data can be modified within the runtime program and saved again without needing to run the full process above, but it is highly suggested that the spread sheet for the card cage be updated as this is being done to keep the two in synchronization. The process for this will be described in detail below. The description for each of the fields is described below:

Minimum Revision Date Code: This is the date code for the minimum qcp revision needed by axis

Minimum Firmware Revision: this is the minimum firmware revision for the card

Serial Unit ID: RS-485 serial address for this slot

CAN Unit ID: CANopen unit ID for this slot

Motor ID [high word decimal [53]: this is an index into the motor type carried in the deviceXXX files. If the motor has a motor memory, this value is confirmed, and then the alignment calibration is read from the motor memory.

MC1, MC2, MC3, MC4, MC5, MC6, MC7, MC8: These constants define the motor characteristics and cable length. They can be determined by initializing the motor with a

default initialization for that motor type, and copying the values from the MCT command into the table. Normally copy the entire column from the desired motor type, and then modify the user items. Motor constants vary with motor type, initialization voltage, and cable length (not overly critical).

PAC1, PAC2, PAC3: these hold the motor phase advance information. Normally copy the entire column. The Phase advance constants vary with the motor type and the initialization voltage.

Step4 | Step_shft [51]: These are a function of the number of motor poles and the encoder resolution. Copy from the table for that motor type.

Encoder Gain 1 | Encoder Gain 2 [52]: These are used to scale the encoder and top speed. Copy from the table for that motor type.

Scaling: This is used to multiply the physical encoder resolution to a higher value to make the number of counts per electrical cycle of the motor into an integer value. Copy from the table for that motor type.

Index Width | encoder alignment [50]: The index width varies with the motor type, 0 or 1 for a conventional encoder, a specific value for various QuickSilver I-grade motors, according to the encoder resolution. The Encoder alignment is used to offset the commutation value either as a function of the particular motor (for I-grade motors, the value here indicates to read the motor memory for the factory calibrated alignment values, for 3 phase motors, this value is STEP4/12, automatically calculated in the spread sheet. It is advised to use the values provided for the motor type being used.

Motor Style: This selects whether the motor is a stepper, Mosolver, Hybrid servo, DC brush, or a 3 phase motor (different settings for S2-IGH and S3-IGH).

Homing Torque (native) : Torque values range from 0 to 30,000. The value corresponding to 100% may be found in the devAux file for the motor. A maximum intermittent value and the 100% continuous values are contained in that file. Homing Torque allows a reduced current to be selected when homing to a hard stop so that the force is constrained.

Running Torque (native): torque value used to allow maximum motion dynamics. The range of motion is automatically constrained by the calibration distance and homing offset, which should prevent the axis from reaching hard limits.

Delay before Homing (ms): This value is used to delay homing operations, particularly if the automatic homing option is selected.

SPARE: The various SPARE locations were reserved for future updated to allow back compatibility. For this reversion they are not currently in use, and are normally set to 0.

Homing Speed (native SVU): The homing speed allows a reduced speed for the homing operation

Homing Offset (counts): this is the offset from the home location to the “zero” location.

Homing Method: 0 is the default home against a hard stop. There are also options to home to a sensor.

Maximum travel: The sign of this term picks the direction of the home and of the motion. The home direction is opposite the direction of the maximum travel and is 1.5* larger in scale to allow the home to be found. A positive value will pick a clockwise motion as the command value via CAN is increased from 0 to 65535. A negative value will pick a counterclockwise motion (of the motor) as the command value is increased.

Initial Velocity (native SVU): When the show mode is first started, the velocity should be set

to a lower value to allow the figure to gracefully get to its starting position, rather than going at maximum speed.

Delay to Show Velocity (ms): This is the delay after starting show mode that the Initial Velocity will limit the speed of the motion. After this the Show Velocity will be used.

Show Velocity (native SVU): This is the maximum speed to be used within the show for this axis.

Show Method: This is a place holder for future options. The default follow the CAN input is "0"

Max Position Error: a positive value will cause a motion error if exceeded, requiring the axis to be restarted. A negative value will set a warning flag and will keep the target from getting further ahead of the motion than this value so that if an axis gets behind it will more gracefully recover.

Maximum Motor Temperature C*16 : Each motor may be equipped with a 1-wire temperature sensor located on the motor. This is wired to Motor Memory and Logic Ground connections. This sensor (if present) will update the temperature readings several times per second. This is reported via the CAN frames. The Maximum Motor Temperature limit shuts down the motor and forces an error to be reported if the temperature exceeds this setting. The value is 16 times the maximum temperature Centigrade, so 100C would be limited by setting this value to 1600.

Minimum Start Voltage * 100: This setting delays start of the motor initialization until the power supply has stabilized to reach at least this setting. Attempting to initialize when the voltage is too low can cause the power supplies to go into fold back, and can also lead to a sub-optimal initialization. This value is scaled by a factor of 100, so 22v would be set by writing 2200 to this line.

Maximum Voltage: exceeding this voltage will cause the drive to shut down and report an error.

OEM KEY: the current value of 4671568 is 0x474850. Ascii codes 0x47 0x48 0x50 decode to GHP. This shall be changed for other vendors.

Max Homing Position Error: For a home to hard stop, the motor is rotated until the difference between the commanded position and the actual position exceed this limit. It should be large enough to not accidentally be triggered, but not so large that it cannot be reached when the axis is started from its furthest distance from home. Typical values are in the table.

Closed loop holding torque: This is the torque allowed when no motion is present. This is also used to set the maximum average torque allowed when the thermal limit is enabled. The peak torque can be set higher, but will eventually lower to the holding torque limit if the axis starts jamming. If the motor has a temperature sensor and limit, this can be set higher as the actual temperature will then be used.

Auto Recover: If set to 1, the axis will automatically reboot when an error is detected and attempt to continue the show. By default, if a higher function show computer is monitoring the axes, this should be set to 0 to allow the show computer to govern recovery.

Enable torque thermal limit: If set to a 1, the maximum torque will be reduced if the motor usage is such that overheating of the motor may occur. Setting to 0 will turn off this function.

Checksum: This is a checksum value (checksum and all the previous values with a 32 bit modulo will equal zero).

Accumulated Motion (HI) | Accumulated Motion(LO): The controller program reads this 64

bit value at power up into a special register in the SilverSterling.

Programming Run Program into S2 and S3 cards

Make sure a copy of devAux03000 3 phase HD 20190906.txt (or a later date) is in the directory “**QCI Initialization**” . Older versions of QuickControl put this under

C:\ProgramData\QuickSilver Controls Inc\QuickControl

Starting with version 6.33, this has been moved to resolve issues with Windows 10 changing permissions for ProgramData. The data for QuickControl is now in:

C:\Users**USER**\AppData\Roaming\QuickSilver Controls Inc\QuickControl

where **USER** is your user name on the computer.

After the slots have been configured, with power off, insert one S2-IGH or one S3 IGH into any slot. Disconnect the motor if it has a Motor Memory. Power the slot on. A fresh card will default to ID16. “Scan Network” (right side panel) and detect the controller. Open **Tools=> Initialization Wizard** and open

“Run from card cage memory 20190920.qcp”

Hit “Download file to device” button. A screen will pop up indicating no motor memory has been found. Indicate the device has no memory (if a motor actually does have memory, the motor type and encoder alignment of 0x8001 – will trigger the program to read the information from the motor memory when this program runs).

If the controller is an S2-IGH, then select “Auto 2” as your motor type, for an S3-IGH, select “Auto 3” as your motor type. This will help you to view the motor current in native units on the control panel.

Power down the card cage, insert the next controller, power up, and repeat from the “Download file to device” stage for all controllers.

Testing Motors

You may choose if you want to connect all motors and cards at the same time or to take them one by one. With power off, connect the motors to the card cage and insert the controllers with the “Run from card cage memory ...” program into the slots for the motors you wish to work with at this time. If the auto-start has been enabled, then the motor will attempt to home and then will transition to the show mode. If auto-start has not been enabled, then it

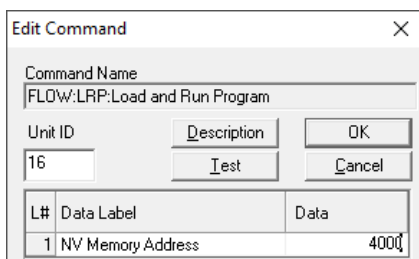
will be necessary to change the NMT state to operational for that drive. QuickControl can be used to determine range of motion by using the control panel. This information can be used to limit the actual range of travel of the axis as configured in the motor memory.

Modifying the Card Cage Memory on the Fly

The card cage memory may be modified by going back to the excel file method above. However, this is time consuming when commissioning many axes, so another approach has also been provided.

Make sure to open “Run from card cage memory...” program so that it is visible in QuickControl. Click in the left panel where it says “Desc” (right below the “Restart” button) and select the desired axis. Open **Tools => Register Watch** and open the registers for the parameters you wish to view and edit. The register numbers are in the rightmost column of the first page of the spread sheet. The value in the register may be modified. To save all values, go to **Edit-> Event -> Lite Edit Event**

In the right column of the page that pops up, scan down and then select “LRP:Load and Run Program. In the left column you will see FLOW:LRP:Load and Run Program. Double click on this in the left column and a new small box will open. The larger “Select Events from Events Database” box may now be closed.



Click in the box under Unit ID and enter the ID of the (here we are assuming 16) controller you are working with. Under Data type 4000.

Click on “Test” (NOT “OK or the box will just close). If it says device is busy, hit the **stop** button (red hand at the top of the screen) and repeat the “Test”.

The controller will run a program to save all of the values from registers 101-164 into the card cage

memory for the slot with the selected controller, and then reboots the controller.

This “modification on the fly” can be useful to adjust range of motion, speed, control constants, and even the “auto-start” state to allow easy commissioning and then this state can be set back to off for normal show operation. NOTE: it is highly suggested that the Excel file be kept open so that any updates made to the card cage memory also get updated in the Excel file so that the configuration remains properly documented.

Program Section Descriptions

The runtime program “Run from card cage memory 20190920.qcp” is broken into the following sub-programs. These automatically run in sequence, selecting the chosen home method as configured in the card cage memory.

Setting ID

The program checks to see if the backplane memory has been configured by checking if the serial id is between 1 and 254 and if the CAN id is between 1 and 127. These may be used even if the overall checksum is bad, so that the checksum error can be reported. If either of these values is out of range, the program status will be set to -1, the serial address will be set to 254 and the CAN ID will be set to 127. The EMGCY message will be set with bit 31 high "Slot memory not initialized". The Green led will flash "S" in morse code – 3 short flashes and a delay.

Verify Checksum

Status is set to 2 while checking the checksum. An invalid checksum will cause the STATUS word to be set to -2, bit 20 in the EMCY word will be set high and EMCY sent, and the Green LED will flash "SOS" 3 short, three long, 3 short flashes and a delay.

Verify Revision

The date of the required revision versus the program revision are next checked. An invalid revision will set STATUS to -5. Bit 29 in the EMCY word will be set, and the Green LED will flash "SOSI" - 3 short, 3 long, 3 short, 2 short and a pause.

Configure Startup Conditions

Enable the normal reporting of EMCY messages for other sources. Drive is normally disabled, and position errors are only valid once the drive is activated, so these bits are masked out to prevent non-existent errors from being reported. STATUS is set to 3.

EMCY using default address, relevant bits configured - don't turn on everything here until the driver has been enabled to prevent spurious error bits from being reported as the configuration continues.

We start thread 2 with the torque limit program, which will only function if enabled. This Thread 2 program also assembles the messages for TPDO3 and TPDO4 into registers accessible to CAN, to then be sent via CANopen for drive status. This thread also is used to clear bits from object 2002 from the controller, if so desired. This thread also clears any clear bits received in RPDO4.

TPDO2: Motor target and Motor Measured Positions - sent once per second

TPDO3: Combined message:

- Motor Temperature 16 bits, C*16;
- Processor Temperature 16 bits C*10;
- Power Supply voltage 16 bits V*100,
- Status state of power up - still being defined

TPDO4: combined message:

- 32 bits: Copy of CAN Errors reported 2002

32 bits: Spare, not currently in use, may be 2 x 16 registers

RPDO1: Target position - 16 bits

RPDO4: clear bits in object 2002 (32 bits)

The CAN messages are then configured. If auto-startup is enabled, the messages NMT will be automatically set to operational. If auto-startup is not enabled, the drive will wait until the show controller sets NMT for the axis to operational (able to process PDO).

The kill motor recovery is configured to catch any errors defined in the kill motor extended command, and the drive waits for the input voltage to reach the minimum voltage.

STATUS=4 is written once the minimum voltage has been reached.

The accumulated motion registers (64038:64039) are loaded with the value stored in the backplane memory. These will automatically be updated and will be saved automatically when the unit is powered down (voltage drops to 18v) which leaves enough power for the write operations to update backplane memory with the final count.

Configure the Motor

The registers associated with configuring the motor are copied from the values read from the backplane memory. The motor constants and phase advance information are also configured from the values stored in the backplane memory. If the slot has been configured to read calibration from the motor memory (currently just QCI motors), STATUS=5 and the data is read, and the checksum calculated. The motor type is also compared to what is specified in the card cage memory for that slot. STATUS=8.

An error will set bit 27 – wrong motor attached. LED will flash “SOSM” 3 short, 3 long, 3 short, 2 long flashes and a delay.

The system next sets up the controller for the particular motor type, 2-phase or 3-phase, configuring the hall sensors to be from IO 2,3,4, or setting up a DC motor if so selected.

The overvoltage limit is set from the backplane memory for this slot.

Configure Servo Tuning

The control constants and filter constants are configured from the values in the backplane memory.

Configure Motion

The motor is started up to configure commutation, according to the motor type. Closed loop is enabled, and the torque limits are set.

Configure Limits

STATUS=11; Kill motor recover program is configured, and faults for temperature, and overvoltage are configured. Motion errors are disabled until the home has been completed. The driver temperature limits are configured, and the motor temperature limit from the backplane memory is enabled.

The homing error limit (home threshold for homing against a hard stop) are set from card cage memory.

Note that positive values for the limit will generate a moving error and a kill condition. Negative values will not produce moving errors.

Configure Misc

STATUS=12, wait homing delay time (from card cage memory) before starting home.

Select the homing method (hard stop=0, home to sensor high or home to sensor low) an invalid home type will set STATUS=-3 and stop the program.

Home to Hard Stop

The moving and holding error kill motor bits are disabled so that a motion error will not cause the drive to fault. The current is reduced to the homing torque, and a motion in the opposite direction to the motion (and 1.5 times larger than the maximum motion distance) is started while monitoring for a moving error. Upon finding a moving error, the command motion stops, and the target is set to the measured position (removing the error between them). The axis is then moved the home offset count in the selected motion direction. The motor is move home offset distance in the direction of the motion from the hard stop and then this location is set as "zero".

Soft travel limits are set to keep the motion between the zero and the maximum distance (or between a negative distance an zero for CCW motions).

The desired show method is then started. Invalid show methods will set STATUS=-4

Home to Sensor

Moves opposite to the direction of motion (up to 1.5 * the normal travel distance) until IO1 is goes to the desired state (IO1 goes high for homing method =1, IO1 goes low for homing method=2). Full current is used for the home, and error limits are active as for a normal motion.

The axis is then moved off the sensor to the homing distance (in the direction of normal motion) and this is set as "zero".

Soft travel limits are set to keep the motion between the zero and the maximum distance (or between a negative distance an zero for CCW motions).

The desired show method is then started. Invalid show methods will set STATUS=-4

Position Input Mode

This is the only implemented show method. The value sent to the unit as a 16 bit unsigned number is scaled to move between 0 and the travel distance (either positive or negative) defined in the backplane memory. The “initial velocity” is used initially until the timer for delay to show velocity has counted down, and then the full show velocity is enabled. The target position is a calculated filtered value based on the commanded position (16 bits) and the allowed velocity. This will continue until either a fault or until the show controller disables or resets the drive.

Kill Motor Recovery

This saves the accumulated motion, sets STATUS=-21, disables the motor and flashes a slow pattern. If auto-recovery is enabled, then the system will restart when the condition that caused the kill motor has ended. The faults that caused the kill motor should automatically be reported by the EMCY word.

Power Low Recovery

Recovery for power dropping starts with saving the accumulated motion. STATUS = -20. If Auto-recovery is enabled, the controller will restart when the power comes up again. If auto-recovery is not enabled, then the program will stop here until the show controller restarts the axis. The low voltage should automatically be reported by the EMCY word.