Controlling Animatronic Mechanisms with Various Interfaces

Included Files:

- QCI-AN079_0to10v_Interface.qcp
- QCI-AN079_DMX_Interface.qcp
- QCI-AN079_CAN_Interface.qcp

This application note is to demonstrate the programming flexibly of QuickSilver Controls' SilverMax™ X-series fully integrated hybrid servo motors and their conventional non-integrated SilverSterling™ series servo controller/drivers in animatronic and show applications.

SilverMax X-series servo motors and SilverSterling series servo controllers both support various interfaces common in the animatronic and show industries. With multiple interface options, the user is free to select the desired servo motor controller interface and not be constrained by product capabilities.

This application note provides example programs for controlling a rotary axis with the following common animatronic/show interfaces:

1. 0 to 10v Analog Input*
2. DMX-512
3. CANopen

*SilverMax X-series servo motors include circuitry to directly interface with a 0 to 10v analog input. For SilverSterling series controller/driver, contact factory for 0 to10v analog interface options.

Mechanism

In this application note example, the mechanism consists of a NEMA 23 SilverMax X-series servo motor (not shown) driving a 50:1 gearhead with a home limit switch. The NEMA 23 SilverMax X-series has an encoder resolution of 8,000 counts/revolution. The clear blue piece is used to illustrate an animatronic rotary joint, such as an elbow, shoulder, knee, etc.

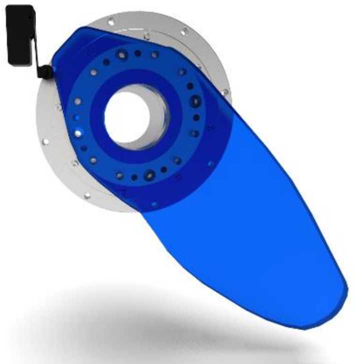


Figure 1 - Axis is homed

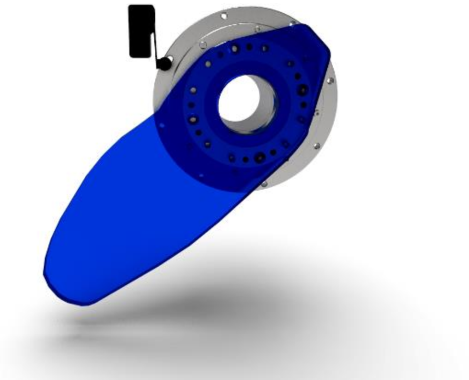


Figure 2 – Axis rotated 90 degrees CW from home

Alternatively, in applications with size and/or weight constraints, the integrated SilverMax X-series servo motor and gearhead may be replaced with a more compact 3rd party geared motor, such as a Harmonic Drive 3-phase geared servo motor with position feedback, paired with a SilverSterling servo controller/driver. Contact Factory for options.

Homing to Sensor

All program examples provided in this application note include a homing routine that homes the motor to a sensor wired into I/O #1. The home sensor is assumed to be active LOW.

Additional homing methods are discussed in Application Note AN001 - Homing Techniques.

On power-up, the motor is programmed to find the home sensor. If the home sensor is active on power-up, logic is included to move motor away from the home sensor until inactive before reversing direction towards home sensor.

After the sensor is found, software stop limits are implemented as a safe guard to only allow commanded motions within a safe travel range.

Once homed, the servo motor program will enter one of the following three interfaces described in this application note:

1. 0 to 10v Analog Input
2. DMX512
3. CAN Bus

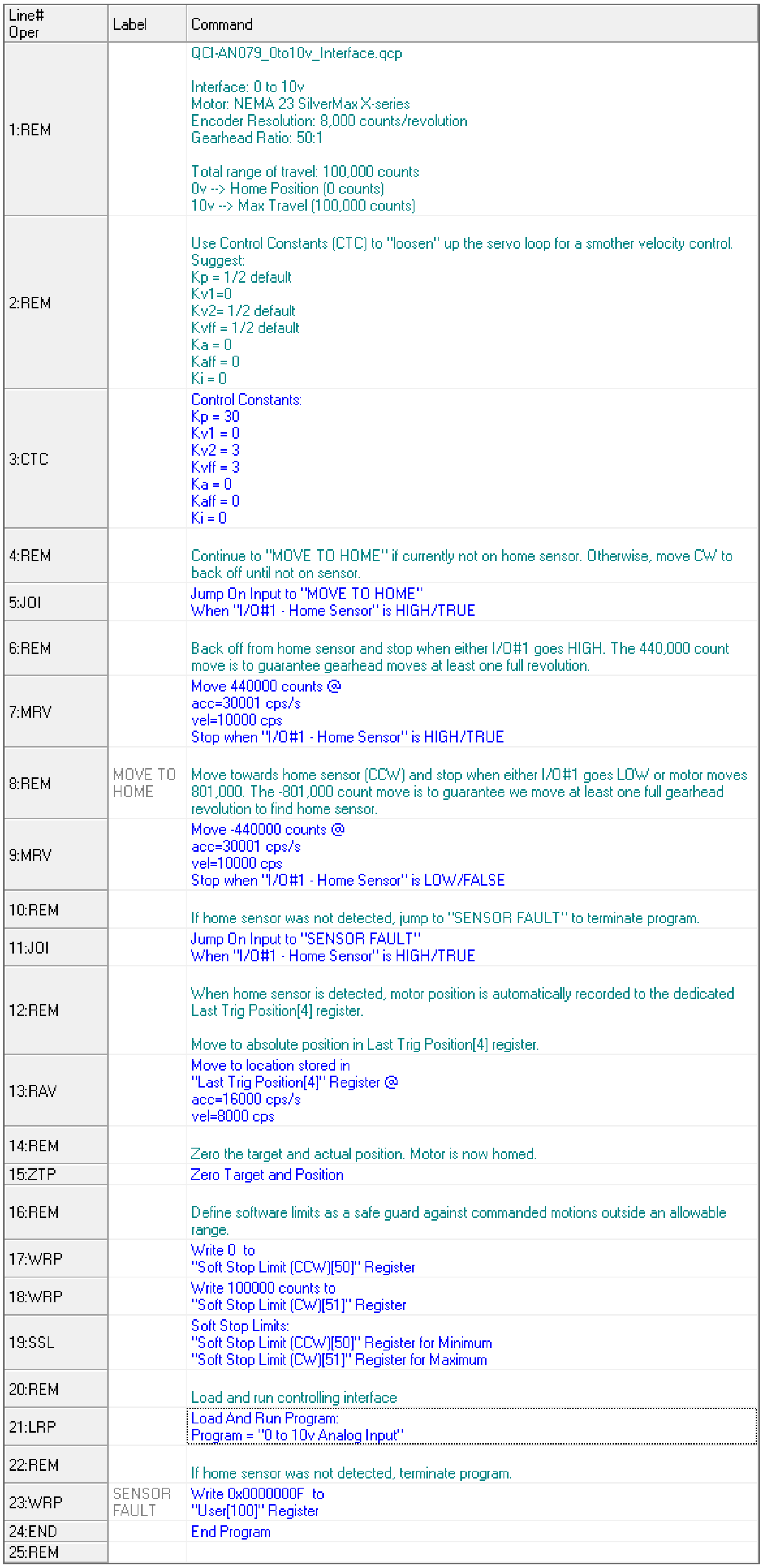
Total Range of Travel

For the NEMA 23 SilverMax X-series servo motor, from the home sensor, the total range of travel is ¼ turn (90°) of the 50:1 gearhead output shaft. This equates to:

*50 revs * ¼ * 8,000 counts/rev*

Total Range of Travel = 100,000 counts

The maximum interface input (10v for analog input and 65535 for DMX/CAN 16-bit input data) will be scaled to the total range of travel, which is 100,000 counts in our application.



Line# Oper	Label	Command
1:REM		QCI-AN079_0to10v_Interface.qcp Interface: 0 to 10v Motor: NEMA 23 SilverMax X-series Encoder Resolution: 8,000 counts/revolution Gearhead Ratio: 50:1 Total range of travel: 100,000 counts 0v --> Home Position (0 counts) 10v --> Max Travel (100,000 counts)

2:REM		Use Control Constants (CTC) to "loosen" up the servo loop for a smother velocity control.
		Suggest:
		Kp = 1/2 default
		Kv1=0
		Kv2= 1/2 default
		Kvff = 1/2 default
		Ka = 0
		Kaff = 0
		Ki = 0

3:CTC		Control Constants:
		Kp = 30
		Kv1 = 0
		Kv2 = 3
		Kvff = 3
		Ka = 0
		Kaff = 0
		Ki = 0

4:REM		Continue to "MOVE TO HOME" if currently not on home sensor. Otherwise, move CW to back off until not on sensor.
5:JOI		Jump On Input to "MOVE TO HOME" When "I/O#1 - Home Sensor" is HIGH/TRUE
6:REM		Back off from home sensor and stop when either I/O#1 goes HIGH. The 440,000 count move is to guarantee gearhead moves at least one full revolution.

7:MRV		Move 440000 counts @
		acc=30001 cps/s
		vel=10000 cps
		Stop when "I/O#1 - Home Sensor" is HIGH/TRUE

8:REM	MOVE TO HOME	Move towards home sensor (CCW) and stop when either I/O#1 goes LOW or motor moves 801,000. The -801,000 count move is to guarantee we move at least one full gearhead revolution to find home sensor.

9:MRV		Move -440000 counts @
		acc=30001 cps/s
		vel=10000 cps
		Stop when "I/O#1 - Home Sensor" is LOW/FALSE

10:REM		If home sensor was not detected, jump to "SENSOR FAULT" to terminate program.
11:JOI		Jump On Input to "SENSOR FAULT" When "I/O#1 - Home Sensor" is HIGH/TRUE
12:REM		When home sensor is detected, motor position is automatically recorded to the dedicated Last Trig Position[4] register. Move to absolute position in Last Trig Position[4] register.

13:RAV		Move to location stored in
		"Last Trig Position[4]" Register @
		acc=16000 cps/s
		vel=8000 cps

14:REM		Zero the target and actual position. Motor is now homed.
15:ZTP		Zero Target and Position
16:REM		Define software limits as a safe guard against commanded motions outside an allowable range.
17:WRP		Write 0 to "Soft Stop Limit (CCW)[50]" Register
18:WRP		Write 100000 counts to "Soft Stop Limit (CW)[51]" Register
19:SSL		Soft Stop Limits: "Soft Stop Limit (CCW)[50]" Register for Minimum "Soft Stop Limit (CW)[51]" Register for Maximum
20:REM		Load and run controlling interface
21:LRP		Load And Run Program: Program = "0 to 10v Analog Input"
22:REM		If home sensor was not detected, terminate program.
23:WRP	SENSOR FAULT	Write 0x0000000F to "User[100]" Register
24:END		End Program
25:REM		

Figure 3 – Home to Sensor

0 to 10v Analog Input

The 0 to 10v analog input is scaled from 0 counts (home position) to 100,000 counts (Total Range of Travel), respectively.

To achieve this, the Position Input Mode (PIM) command function is used to automatically control and scale motor position proportional to the analog input.

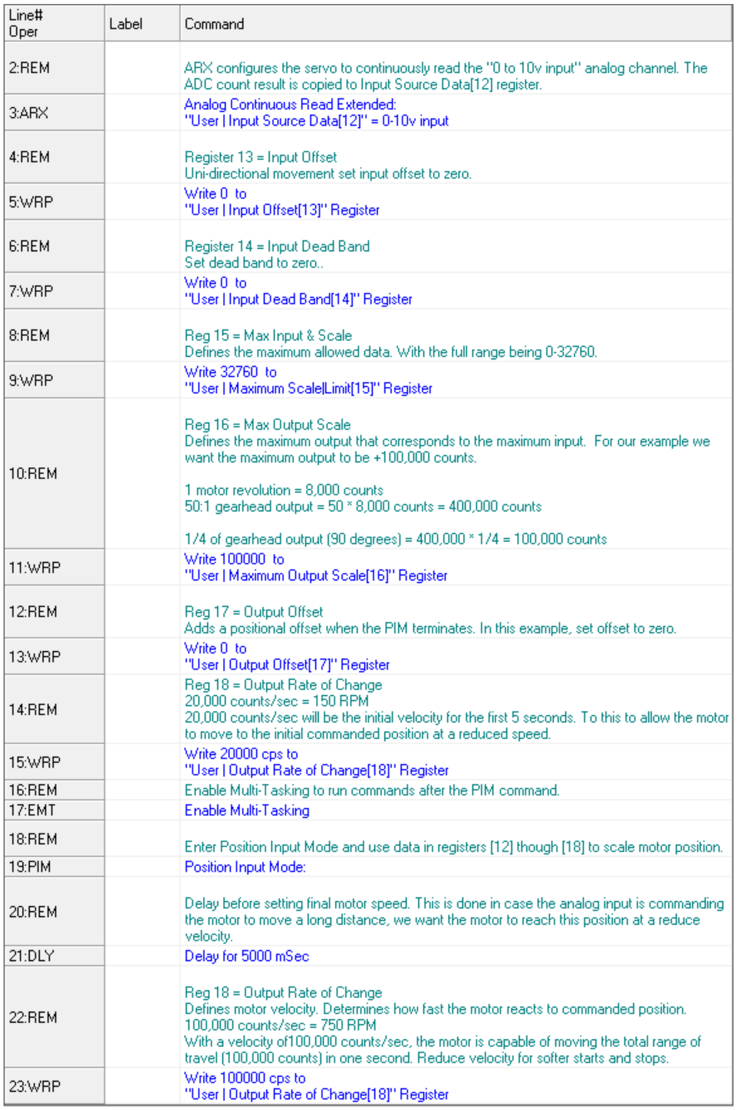
Line 3: ARX command configures the ADC continuously convert the 0 to 10v analog input and copy the ADC result to Input Source Data[12] register.

Lines 5-15: Define registers used by PIM command:

Input Offset [13]
Input Dead Band[14]
Maximum Scale[15]
Maximum Scale Output[16]
Output Offset[17]
Output Rate of Change[18]

Line 11: Sets total range of travel. If modifying this line, note to also modify the software stop limits in the homing program.

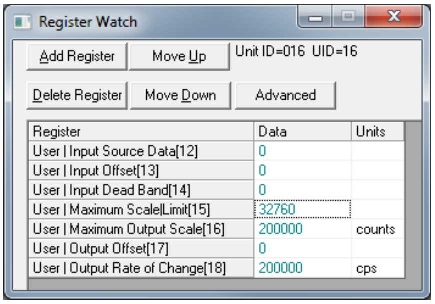
Line 15: Sets motor velocity to control how fast motor can respond to changes in commanded position. For immediate motor response to commanded position, increase velocity. For softer starts and stop, reduce velocity. Note: Initially, motor velocity is at a reduced valued for 5 seconds to allow motor to wind-up to target position. After a brief delay, velocity is increased.



Line# Oper	Label	Command
2:REM		ARX configures the servo to continuously read the "0 to 10v input" analog channel. The ADC count result is copied to Input Source Data[12] register.
3:ARX		Analog Continuous Read Extended: "User | Input Source Data[12]" = 0-10v input
4:REM		Register 13 = Input Offset Uni-directional movement set input offset to zero.
5:WRP		Write 0 to "User | Input Offset[13]" Register
6:REM		Register 14 = Input Dead Band Set dead band to zero..
7:WRP		Write 0 to "User | Input Dead Band[14]" Register
8:REM		Reg 15 = Max Input & Scale Defines the maximum allowed data. With the full range being 0-32760.
9:WRP		Write 32760 to "User | Maximum Scale|Limit[15]" Register
10:REM		Reg 16 = Max Output Scale Defines the maximum output that corresponds to the maximum input. For our example we want the maximum output to be +100,000 counts. 1 motor revolution = 8,000 counts 50:1 gearhead output = 50 * 8,000 counts = 400,000 counts 1/4 of gearhead output (90 degrees) = 400,000 * 1/4 = 100,000 counts
11:WRP		Write 100000 to "User | Maximum Output Scale[16]" Register
12:REM		Reg 17 = Output Offset Adds a positional offset when the PIM terminates. In this example, set offset to zero.
13:WRP		Write 0 to "User | Output Offset[17]" Register
14:REM		Reg 18 = Output Rate of Change 20,000 counts/sec = 150 RPM 20,000 counts/sec will be the initial velocity for the first 5 seconds. To this to allow the motor to move to the initial commanded position at a reduced speed.
15:WRP		Write 20000 cps to "User | Output Rate of Change[18]" Register
16:REM		Enable Multi-Tasking to run commands after the PIM command.
17:EMT		Enable Multi-Tasking
18:REM		Enter Position Input Mode and use data in registers [12] though [18] to scale motor position.
19:PIM		Position Input Mode:
20:REM		Delay before setting final motor speed. This is done in case the analog input is commanding the motor to move a long distance, we want the motor to reach this position at a reduce velocity.
21:DLY		Delay for 5000 mSec
22:REM		Reg 18 = Output Rate of Change Defines motor velocity. Determines how fast the motor reacts to commanded position. 100,000 counts/sec = 750 RPM With a velocity of100,000 counts/sec, the motor is capable of moving the total range of travel (100,000 counts) in one second. Reduce velocity for softer starts and stops.
23:WRP		Write 100000 cps to "User | Output Rate of Change[18]" Register

Figure 4 – QCI-AN079_0to10v_Interface.qcp

Once program is downloaded, registers may be modified via the Register Watch tool (Tools → Register Watch) with changes to registers being applied on the fly. This allows the programmer to test register parameters without having to redownload program.



Unit ID=016 UID=16

Register	Data	Units
User | Input Source Data[12]	0	
User | Input Offset[13]	0	
User | Input Dead Band[14]	0	
User | Maximum Scale|Limit[15]	32760	
User | Maximum Output Scale[16]	200000	counts
User | Output Offset[17]	0	
User | Output Rate of Change[18]	200000	cps

Figure 5 – Register Watch Tool

Suggestions

If Input Source Data[12] ADC result doesn't zero out with a 0v input, this offset can be cancelled by adjusting the Input Offset[13] register. On the opposite end, if motor doesn't reach the full length of travel for 10v input, adjust the Maximum Scale|Limit[15] register.

For velocity "jog" control, rather than positional control, the Velocity Input Mode (VIM) command function is available. For more information on Input Mode command functions, refer to Application Note AN047 - Input Mode.pdf

DMX512

The DMX protocol transmits data in bursts (DMX packet). A DMX packet begins with a start code which identifies the data type followed by up to 512 individual bytes of data called slots or channels. Each slot carries 8-bits of information and can hold a value from 0 to 255. QCI's devices can map a single 8-bit slot, up to four contiguous 8-bit slots, providing 32-bits of DMX data.

The application program will map 2 contiguous 8-bit slots (16-bits) for a total incoming DMX data range of 0 to 65535. DMX slots 10 and 11 are mapped to user register [41], then the contents of register [41] is copied to the Input Source Data [12] register. The program uses the same PIM command function as the 0 to 10v application example above. However, instead converting analog input to ADC counts into Input Source Data[12] register, the program will map DMX data into Input Source Data[12] register.

Lines 3-5: Configures communication for DMX protocol.

Line 7: DMRM command defines a single register map. The DMRM command can define up to 6 separate register maps to map additional DMX slots within the same DMX packet.

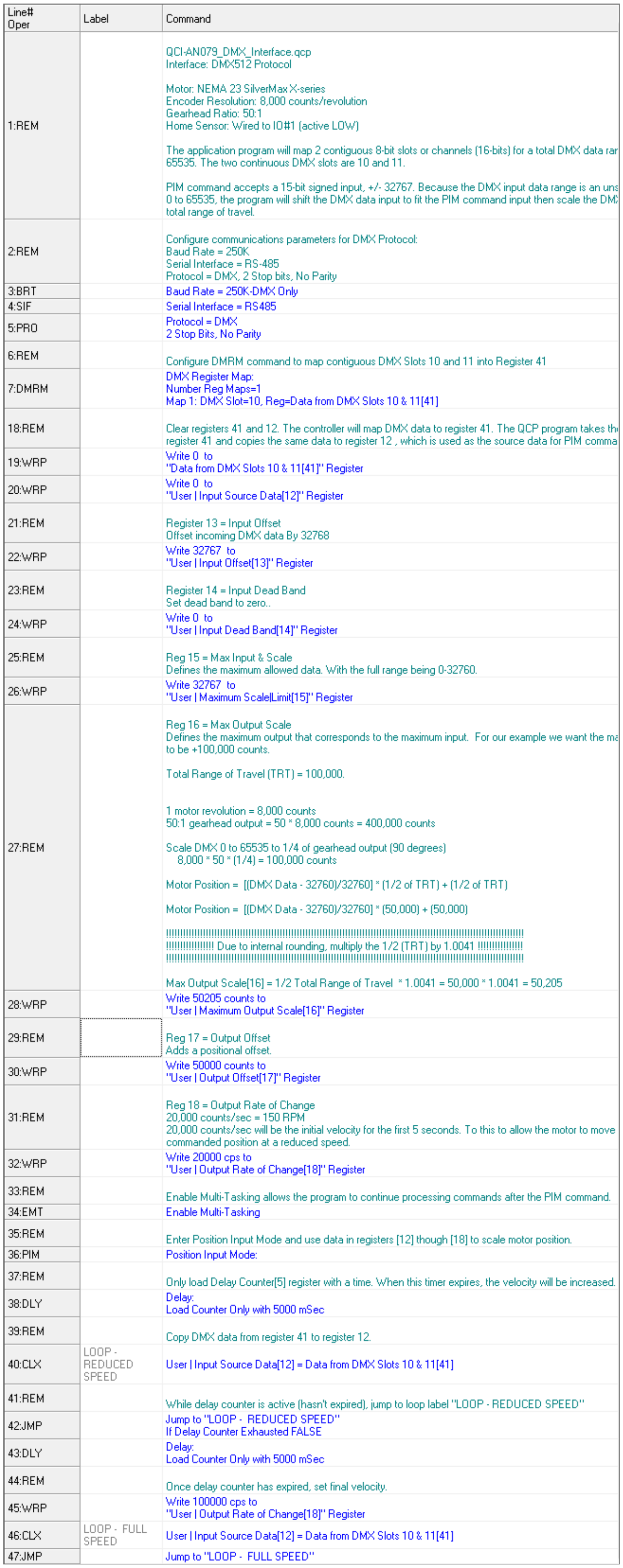
Lines 19-32: Define registers used by PIM command:

Input Offset [13]
Input Dead Band[14]
Maximum Scale[15]
Maximum Scale Output[16]
Output Offset[17]
Output Rate of Change[18]

Line 32: Sets motor velocity to control how fast motor can respond to changes in commanded position.

Line 46: Copy mapped DMX data from register [41] into Input Source Data[12].

Refer to Application Note AN045 – DMX512 Protocol.pdf for more information on advanced DMX-512 configuration options.



Line# Oper	Label	Command
1:REM		QCI-AN079_DMX_Interface.qcp Interface: DMX512 Protocol Motor: NEMA 23 SilverMax X-series Encoder Resolution: 8,000 counts/revolution Gearhead Ratio: 50:1 Home Sensor: Wired to IO#1 (active LOW) The application program will map 2 contiguous 8-bit slots or channels (16-bits) for a total DMX data rar 65535. The two continuous DMX slots are 10 and 11. PIM command accepts a 15-bit signed input, +/- 32767. Because the DMX input data range is an uns 0 to 65535, the program will shift the DMX data input to fit the PIM command input then scale the DM total range of travel.
2:REM		Configure communications parameters for DMX Protocol: Baud Rate = 250K Serial Interface = RS-485 Protocol = DMX, 2 Stop bits, No Parity
3:BRT		Baud Rate = 250K-DMX Only
4:SIF		Serial Interface = RS485
5:PRO		Protocol = DMX 2 Stop Bits, No Parity
6:REM		Configure DMRM command to map contiguous DMX Slots 10 and 11 into Register 41
7:DMRM		DMX Register Map: Number Reg Maps=1 Map 1: DMX Slot=10, Reg=Data from DMX Slots 10 & 11[41]
18:REM		Clear registers 41 and 12. The controller will map DMX data to register 41. The QCP program takes th register 41 and copies the same data to register 12 , which is used as the source data for PIM comma
19:WRP		Write 0 to "Data from DMX Slots 10 & 11[41]" Register
20:WRP		Write 0 to "User | Input Source Data[12]" Register
21:REM		Register 13 = Input Offset Offset incoming DMX data By 32768
22:WRP		Write 32767 to "User | Input Offset[13]" Register
23:REM		Register 14 = Input Dead Band Set dead band to zero..
24:WRP		Write 0 to "User | Input Dead Band[14]" Register
25:REM		Reg 15 = Max Input & Scale Defines the maximum allowed data. With the full range being 0-32760.
26:WRP		Write 32767 to "User | Maximum Scale|Limit[15]" Register
27:REM		Reg 16 = Max Output Scale Defines the maximum output that corresponds to the maximum input. For our example we want the ma to be +100,000 counts. Total Range of Travel (TRT) = 100,000. 1 motor revolution = 8,000 counts 50:1 gearhead output = 50 * 8,000 counts = 400,000 counts Scale DMX 0 to 65535 to 1/4 of gearhead output (90 degrees) 8,000 * 50 * (1/4) = 100,000 counts Motor Position = [(DMX Data - 32760)/32760] * (1/2 of TRT) + (1/2 of TRT) Motor Position = [(DMX Data - 32760)/32760] * (50,000) + (50,000) !! !!!!!!!!!!!!!!!!! Due to internal rounding, multiply the 1/2 (TRT) by 1.0041 !!!!!!!!!!!!!!!! !! Max Output Scale[16] = 1/2 Total Range of Travel * 1.0041 = 50,000 * 1.0041 = 50,205
28:WRP		Write 50205 counts to "User | Maximum Output Scale[16]" Register
29:REM		Reg 17 = Output Offset Adds a positional offset.
30:WRP		Write 50000 counts to "User | Output Offset[17]" Register
31:REM		Reg 18 = Output Rate of Change 20,000 counts/sec = 150 RPM 20,000 counts/sec will be the initial velocity for the first 5 seconds. To this to allow the motor to move commanded position at a reduced speed.
32:WRP		Write 20000 cps to "User | Output Rate of Change[18]" Register
33:REM		Enable Multi-Tasking allows the program to continue processing commands after the PIM command.
34:EMT		Enable Multi-Tasking
35:REM		Enter Position Input Mode and use data in registers [12] though [18] to scale motor position.
36:PIM		Position Input Mode:
37:REM		Only load Delay Counter[5] register with a time. When this timer expires, the velocity will be increased.
38:DLY		Delay: Load Counter Only with 5000 mSec
39:REM		Copy DMX data from register 41 to register 12.
40:CLX	LOOP - REDUCED SPEED	User | Input Source Data[12] = Data from DMX Slots 10 & 11[41]
41:REM		While delay counter is active (hasn't expired), jump to loop label "LOOP - REDUCED SPEED"
42:JMP		Jump to "LOOP - REDUCED SPEED" If Delay Counter Exhausted FALSE
43:DLY		Delay: Load Counter Only with 5000 mSec
44:REM		Once delay counter has expired, set final velocity.
45:WRP		Write 100000 cps to "User | Output Rate of Change[18]" Register
46:CLX	LOOP - FULL SPEED	User | Input Source Data[12] = Data from DMX Slots 10 & 11[41]
47:JMP		Jump to "LOOP - FULL SPEED"

Figure 6 – QCI-AN079_DMX_Interface.qcp

CANopen

The CANopen program example configures an Receive Process Data Object (RPDO) to have the unit to receive 16-bits of data transmitted from a remote CAN device with CAN ID of 1. The data is assumed to be transmitted and received via CAN Channel #1. The data is mapped to Input Data Source [12] register, where the PIM command uses to scale motor position.

Lines 3-5: Configures CAN baud rate and CAN ID.

Lines 7-18: Configures the unit to map CAN data to its local register, Input Source Data[12] register.

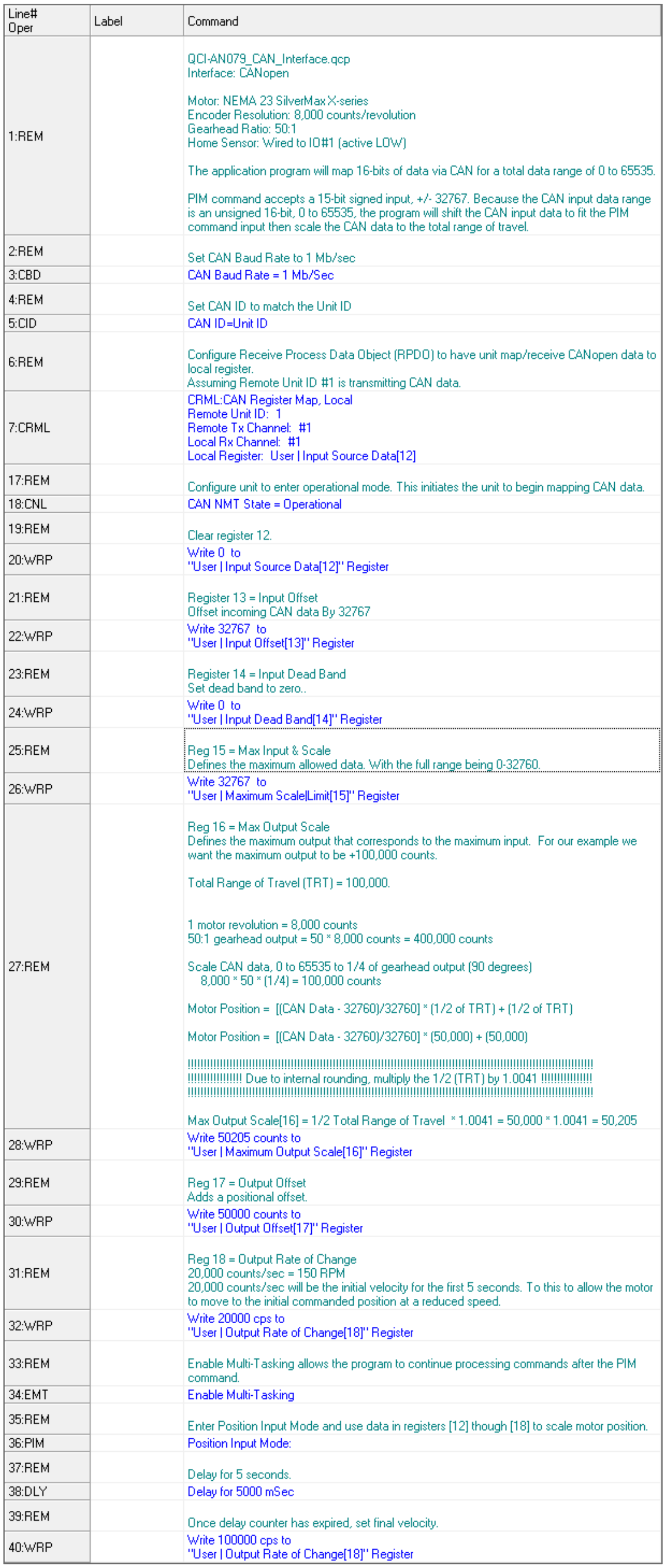
Lines 20-32: Define registers used by PIM command:

Input Offset [13]
Input Dead Band[14]
Maximum Scale[15]
Maximum Scale Output[16]
Output Offset[17]
Output Rate of Change[18]

Line 32: Sets the initial motor velocity. The initial motor velocity is reduced to prevent the motor from taking off at a high speed when first entering PIM.

Line 40: After an initial delay of 5 seconds, the final velocity is set.

Refer to SilverLode CANopen User Manual for more information on CANopen.



```
Line#
Oper      Label   Command
1:REM             QCI-AN079_CAN_Interface.qcp
                  Interface: CANopen

                  Motor: NEMA 23 SilverMax X-series
                  Encoder Resolution: 8,000 counts/revolution
                  Gearhead Ratio: 50:1
                  Home Sensor: Wired to IO#1 (active LOW)

                  The application program will map 16-bits of data via CAN for a total data range of 0 to 65535.

                  PIM command accepts a 15-bit signed input, +/- 32767. Because the CAN input data range
                  is an unsigned 16-bit, 0 to 65535, the program will shift the CAN input data to fit the PIM
                  command input then scale the CAN data to the total range of travel.
2:REM             Set CAN Baud Rate to 1 Mb/sec
3:CBD             CAN Baud Rate = 1 Mb/Sec
4:REM             Set CAN ID to match the Unit ID
5:CID             CAN ID=Unit ID
6:REM             Configure Receive Process Data Object (RPDO) to have unit map/receive CANopen data to
                  local register.
                  Assuming Remote Unit ID #1 is transmitting CAN data.
7:CRML            CRML:CAN Register Map, Local
                  Remote Unit ID:  1
                  Remote Tx Channel:  #1
                  Local Rx Channel:  #1
                  Local Register:  User | Input Source Data[12]
17:REM            Configure unit to enter operational mode. This initiates the unit to begin mapping CAN data.
18:CNL            CAN NMT State = Operational
19:REM            Clear register 12.
20:WRP            Write 0  to
                  "User | Input Source Data[12]" Register
21:REM            Register 13 = Input Offset
                  Offset incoming CAN data By 32767
22:WRP            Write 32767  to
                  "User | Input Offset[13]" Register
23:REM            Register 14 = Input Dead Band
                  Set dead band to zero..
24:WRP            Write 0  to
                  "User | Input Dead Band[14]" Register
25:REM            Reg 15 = Max Input & Scale
                  Defines the maximum allowed data. With the full range being 0-32760.
26:WRP            Write 32767  to
                  "User | Maximum Scale|Limit[15]" Register
27:REM            Reg 16 = Max Output Scale
                  Defines the maximum output that corresponds to the maximum input.  For our example we
                  want the maximum output to be +100,000 counts.

                  Total Range of Travel (TRT) = 100,000.


                  1 motor revolution = 8,000 counts
                  50:1 gearhead output = 50 * 8,000 counts = 400,000 counts

                  Scale CAN data, 0 to 65535 to 1/4 of gearhead output (90 degrees)
                     8,000 * 50 * (1/4) = 100,000 counts

                  Motor Position =  [(CAN Data - 32760)/32760] * (1/2 of TRT) + (1/2 of TRT)

                  Motor Position =  [(CAN Data - 32760)/32760] * (50,000) + (50,000)

                  !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
                  !!!!!!!!!!!!!!! Due to internal rounding, multiply the 1/2 (TRT) by 1.0041 !!!!!!!!!!!!!!!
                  !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!

                  Max Output Scale[16] = 1/2 Total Range of Travel  * 1.0041 = 50,000 * 1.0041 = 50,205
28:WRP            Write 50205 counts to
                  "User | Maximum Output Scale[16]" Register
29:REM            Reg 17 = Output Offset
                  Adds a positional offset.
30:WRP            Write 50000 counts to
                  "User | Output Offset[17]" Register
31:REM            Reg 18 = Output Rate of Change
                  20,000 counts/sec = 150 RPM
                  20,000 counts/sec will be the initial velocity for the first 5 seconds. To this to allow the motor
                  to move to the initial commanded position at a reduced speed.
32:WRP            Write 20000 cps to
                  "User | Output Rate of Change[18]" Register
33:REM            Enable Multi-Tasking allows the program to continue processing commands after the PIM
                  command.
34:EMT            Enable Multi-Tasking
35:REM            Enter Position Input Mode and use data in registers [12] though [18] to scale motor position.
36:PIM            Position Input Mode:
37:REM            Delay for 5 seconds.
38:DLY            Delay for 5000 mSec
39:REM            Once delay counter has expired, set final velocity.
40:WRP            Write 100000 cps to
                  "User | Output Rate of Change[18]" Register
```

Figure 7 – QCI-AN079_CAN_Interface.qcp