Red Lion HMI

Associated Files:
- Profile Move Continuous.qcp: Example QuickControl Program File
- SilverDust.cd2: Example Red Lion Crimson 2.0 Program File
- Red LionQCIMasterDriverInfoSheet.pdf: Red Lion's documentation on interfacing to QCI servos.

Overview

The Human Machine Interface (HMI) creates a powerful interactive communication system between human operators and machines. This application note describes how to successfully achieve this type of interface using a SilverDust Controller and Red Lion Controls G306 HMI. Note, although the example is written for a SilverDust IGB, the general concept and wiring is pertinent to all SilverLode servo controllers.

Depending on the Red Lion HMI model used, there may be additional settings or options required. The basic description in this document should allow for successful operation. The following application will describe how to use the Profile Move Continuous command with the Red Lion HMI as the host. For support on the Red Lion devices please contact:techsupport@Red Lion.net. For support on the SilverLode servo controller contact:support@QuickSilverControls.com. It is assumed the reader is familiar with programming QuickSilver's servo controllers. If not familiar with our controllers, read chapter 1 of the SilverLode User Manual.

Hardware and Software Setup

Only three lines are required for communications to the HMI. The three lines have different meanings depending on whether using RS-232 or RS-485. The Red Lion G306 device uses a RJ-12 cable for RS-232 communication and a RJ-45 cable for RS-485 communication.

RS-232 RJ-12 cable connection

Signal Type	SMI Port* DB15	D9 Port	RJ-12 cable from G306*
Rx	Pin 12	Pin 3	Pin 2 (Tx)
Tx	Pin 2	Pin 2	Pin 1 (Rx)
Logic Ground	Pin 8	Pin 5	Pin 5 (Gnd)

Note, the SilverDust and SilverNugget have the same wiring for RS-232.
*Pin 1 on left when RJ-12 facing you with tang up.
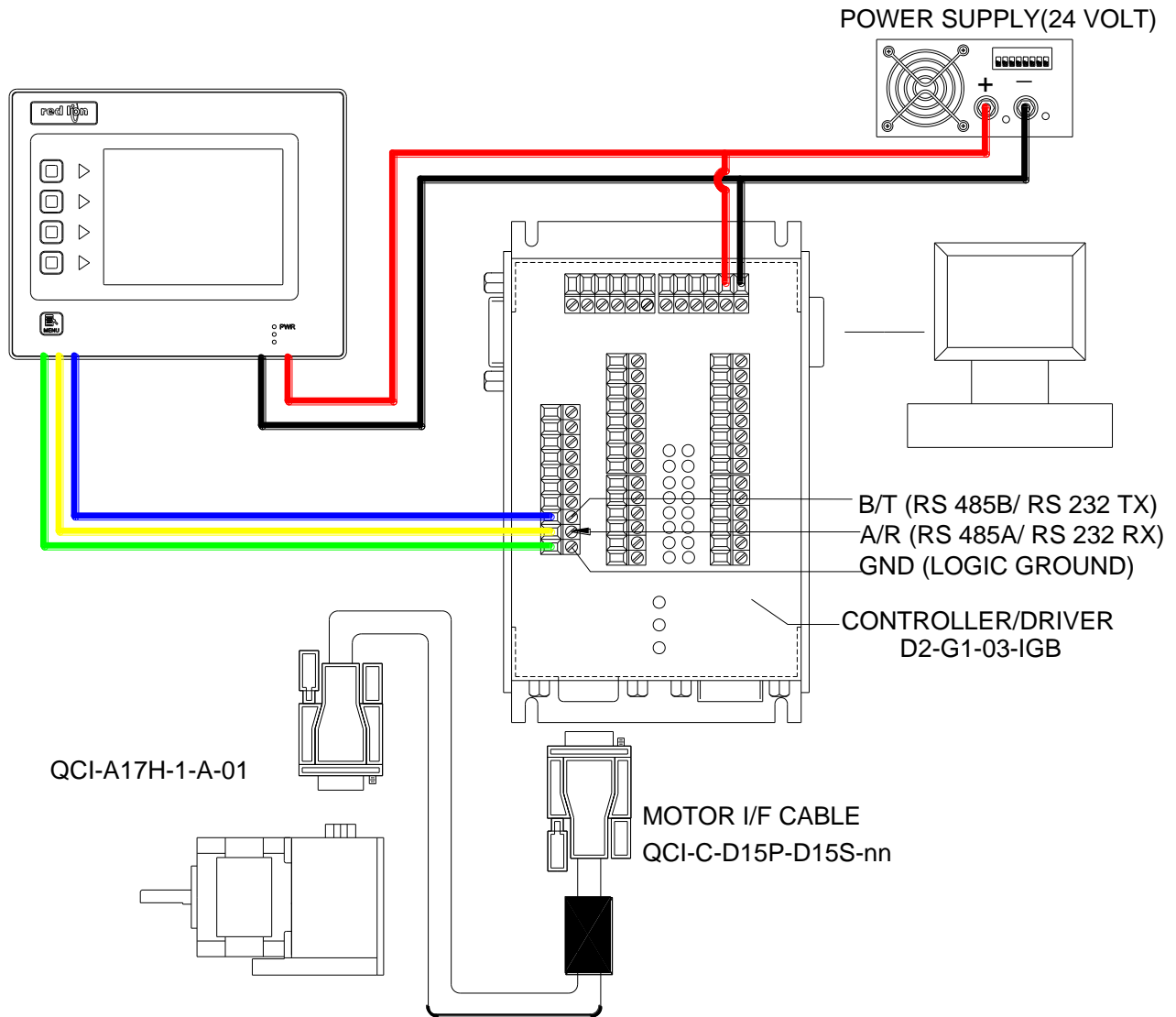
RS-485 RJ-45 cable connection

Signal Type	SilverDust SMI Port*	SilverNugget SMI Port*	SilverDust D9 Port	SilverNugget D9 Port	RJ-45 cable from G306
A	Pin 12	Pin 2	Pin 3	Pin 2	Pin 2 or 8
B	Pin 2	Pin 12	Pin 2	Pin 3	Pin 1 or 7
Logic Ground	Pin 8	Pin 8	Pin 5	Pin 5	Pin 6

*The communications signals are available on all SilverLode controllers SMI ports. Breakouts like the QCI-BO-B are useful to wire the signals to the HMI. In addition, some controllers, like the IGB, have built in breakouts for these signals. Refer to controller's datasheet for details.

Note that for RS-232 communication, the RX/TX lines are attached in null modem style. Note Logic Ground needs to be attached for both RS-232 and RS-485.

Please refer to Red Lion and SilverLode manuals for more information.

The following wiring diagram shows the SilverDust IGB attached to the HMI all powered off a single 24V supply.

POWER SUPPLY(24 VOLT)

B/T (RS 485B/ RS 232 TX)
A/R (RS 485A/ RS 232 RX)
GND (LOGIC GROUND)

CONTROLLER/DRIVER
D2-G1-03-IGB

QCI-A17H-1-A-01

MOTOR I/F CABLE
QCI-C-D15P-D15S-nn

Setting Up the SilverDust Servo Controller

Use QuickControl 4.51 or greater – available from the QCI website (www.quicksilvercontrols.com) – to program the controller. Initialize the controller to the appropriate communication protocol of RS-232 or RS-485. Our example assumes the servo has been setup for Unit ID 16, ASCII Protocol (8 data bits 2 stop bits), RS-232, and 57600 baud.

Once the servo is initialized, download the "Profile Move Continuous.qcp" program.   This program uses the PMC command to do simple clockwise/counter clockwise movements. See Command Reference for more details.

Note, the PMC command uses register 22 for velocity, which the HMI will change (see below).

Line# Oper	Label	Command
1:REM		This program puts the servo into a continuous Profile Move. Multi-Tasking is enabled to allow the internal program to move the motor back and forth
2:EMT		Enable Multi-Tasking
3:REM		Writes "0" to the Position Data Register #20
4:WRP		Write 0 counts to "User or Profile Move Pos[20]" Register
5:REM		Writes to the Acceleration Data Register #21
6:WRP		Write 99999 cps/s to "User or Profile Move Acc[21]" Register
7:REM		Writes to the Velcity Data Register #22
8:WRP		Write 10000 cps to "User or Profile Move Vel[22]" Register
9:REM		Writes to the Deceleration Data Register #23
10:WRP		Write 99999 cps/s to "User or Profile Move Dec[23]" Register
11:REM		Put the servo into a Continuous Profile Move. This will not end until a Stop, Halt or Velocity command is used to Over-ride
12:PMC		Profile Move Continuous:
13:REM		Go into a loop that changes the Postion every second.
14:WRP	AGAIN	Write 20000 counts to "User or Profile Move Pos[20]" Register
15:DLY		Delay for 5000 mSec
16:WRP		Write 0 counts to "User or Profile Move Pos[20]" Register
17:DLY		Delay for 5000 mSec
18:JMP		Jump to "AGAIN"

Setting Up the HMI

Crimson 2.0 software-available from Red Lion controls- is used to program the Red Lion G306 HMI. Bundled with the Application Note is the example Red Lion program "Silverdust.cd2". Run Crimson and open the file "Silverdust.cd2".
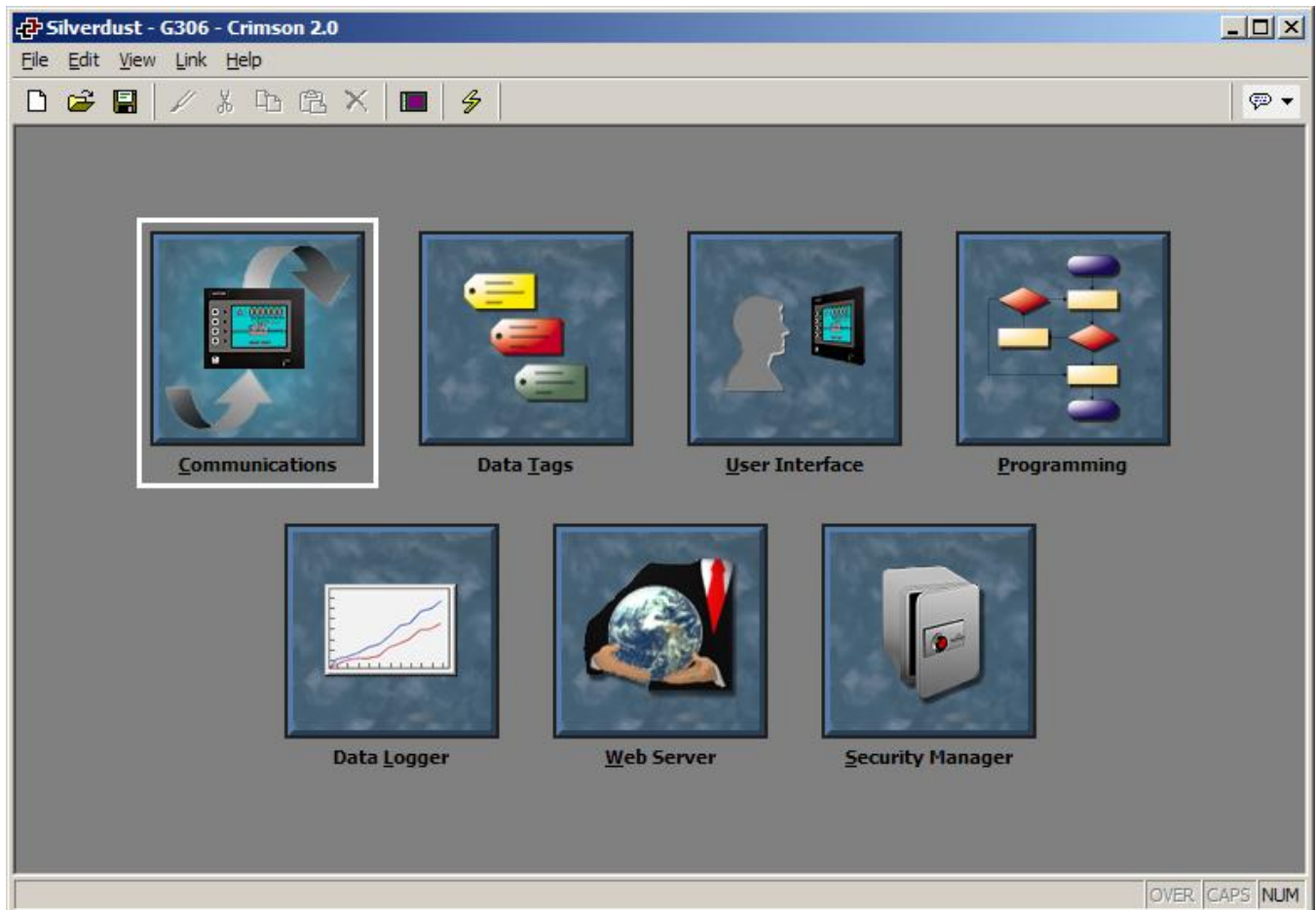
The program file, Silverdust.cd2, already has communications setup between the servo and the HMI for Unit ID 16, RS-232, and 57600 baud.   This can be changed by pressing the Communications button.
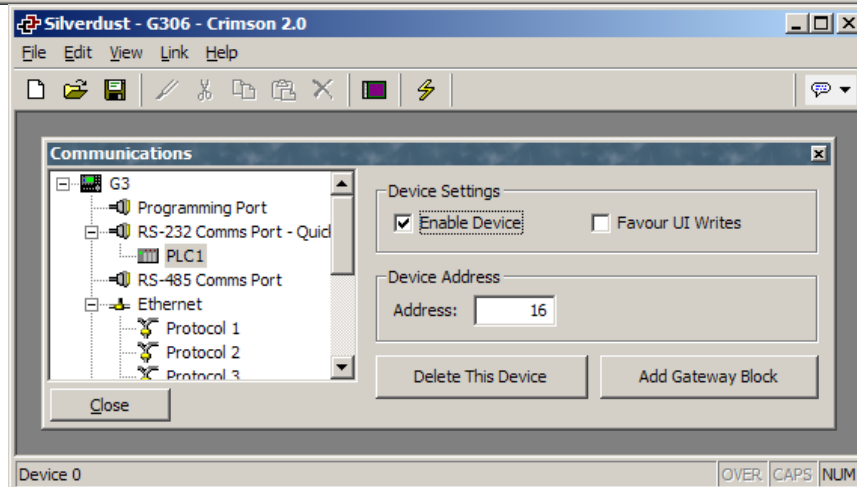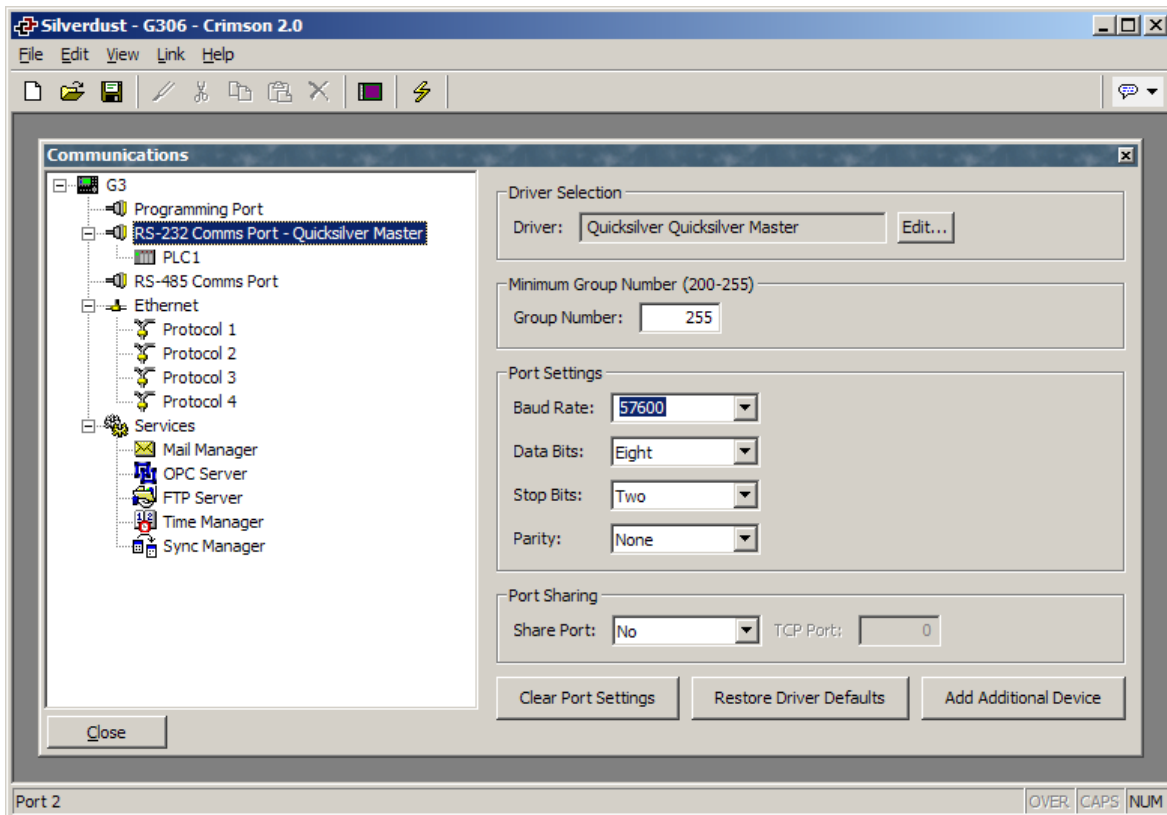
Red Lion Data Tag

The Red Lion data tags are used to store internal information or to access data from a remote location. For this application, there are three data tags used.

RPM
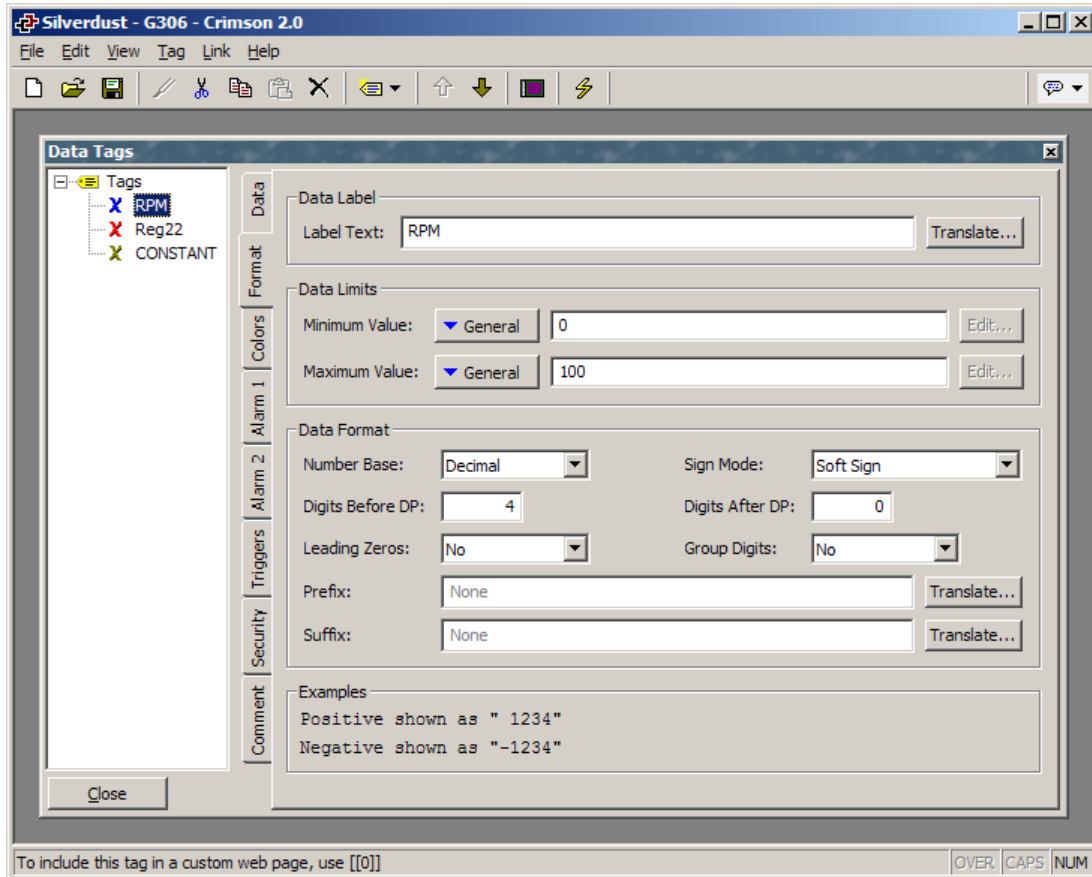
The integer variable tag "RPM" is used for the requested RPM. The Data Limits are set to 0 and 100. (Note, the max RPM for the servo is 4000RPM but this example only uses 100RPM to keep the speeds slow).

Reg22
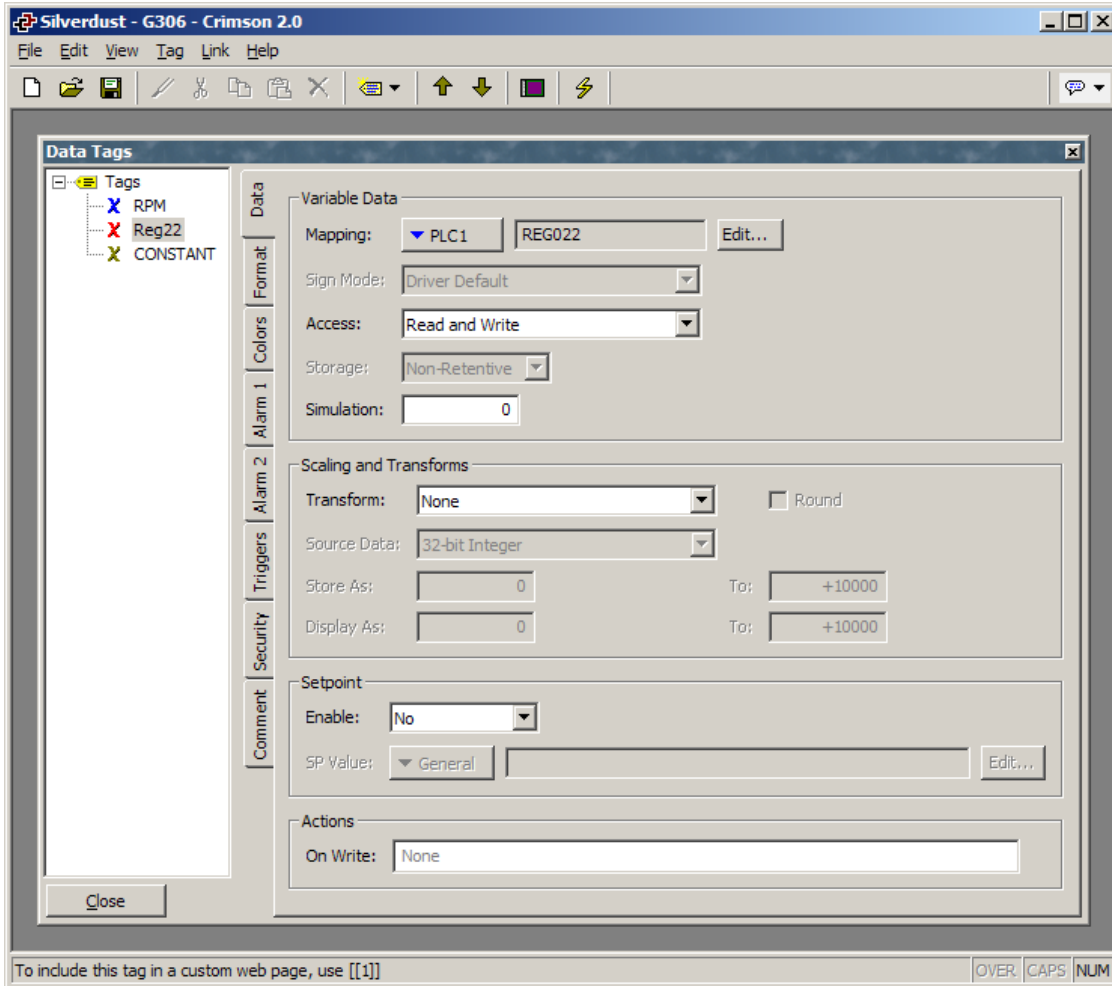
Reg22 is mapped to register 22 of the controller. Changes to this data tag will result in writes (WRI) to the controller. Viewing the contents of this data tag will result in reads (RRG) from the controller. Press Edit if you want to change the register number.

CONSTANT

The formula integer tag "CONSTANT" is used to represent the constant value of 536871. This and RPM are used to convert the RPM entry into SilverLode Velocity Unit (SVU). The formula used to calculate SVU is the following:

SVU = CONSTANT x RPM
SVU= 536871 x RPM

Red Lion User Interface
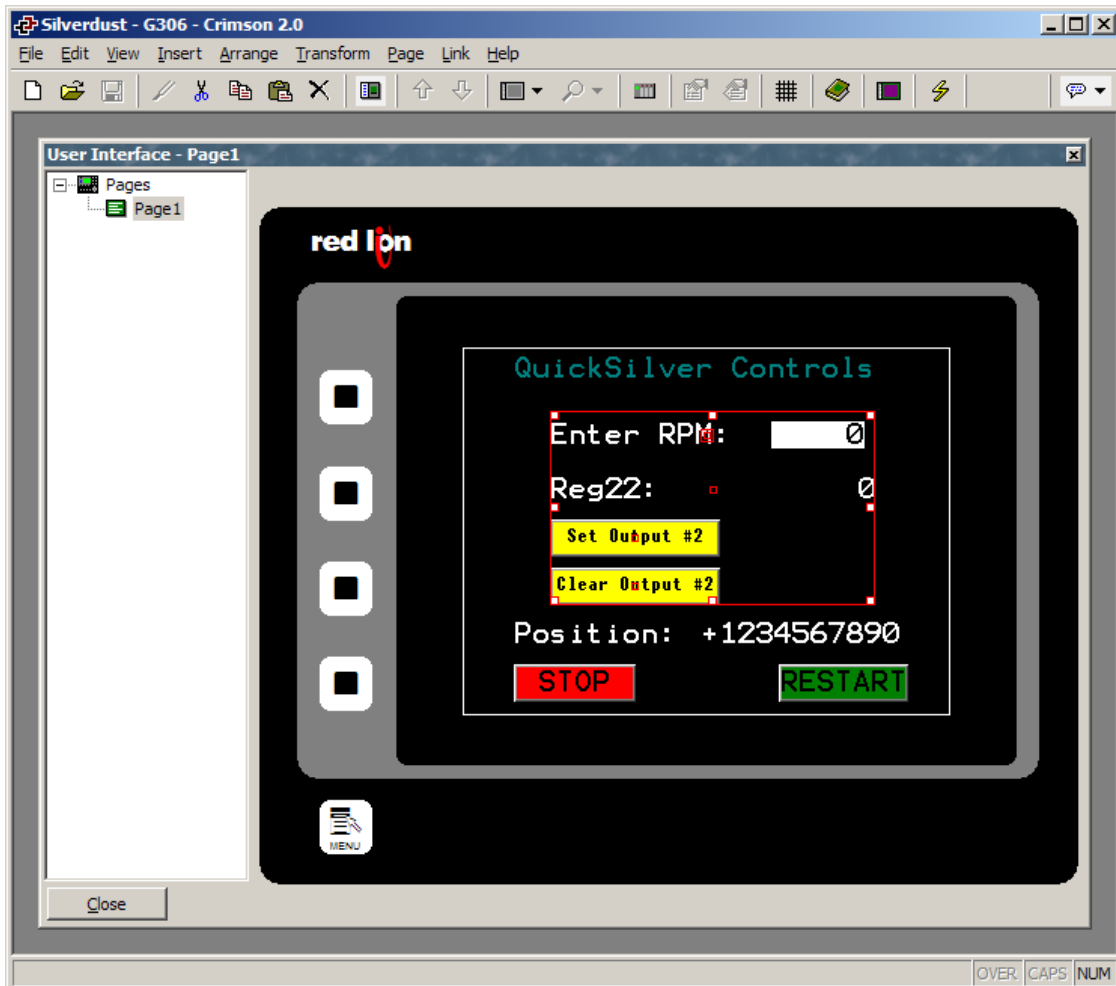
The Red Lion user interface screen displays a page that the user interacts with. The page defines the action taken when the screen is pressed, released or held down. The user interface screen will display the actual position register and display the SVU (SilverLode Velocity Units) being sent to the velocity register of the controller.

Enter RPM
The "Enter RPM" text integer displays the RPM data tag. RPM can be used to enter a value of 0 to 100 RPM by pressing the field.

Reg22
Reg22 displays the servo's register 22.

Set Output #2/Clear Output #2
Pressing "Set Output #2" and "Clear Output #2" will set and clear the output as long as the program is not running (see STOP below). To make these buttons work while the program is running, change them to Red Lion's CII command as follows. Note, CII will not work with the SilverNugget.

These are examples of using SilverLode commands without a data tags.
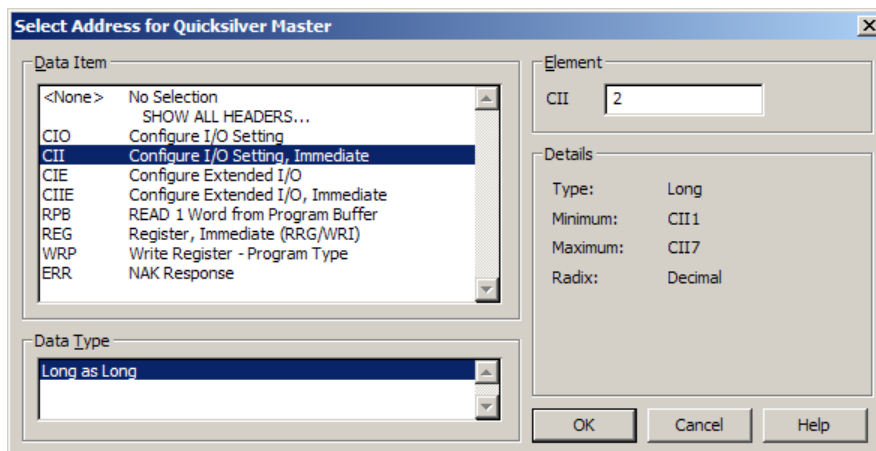
Position
Position displays the actual position register (reg 1) from the servo.

STOP/RESTART
Press STOP to send a Stop (STP) command to the servo. The Stop command stops the servo and ends the "Profile Move Continuous.qcp" program. New RPM entries will be ignored until the RESTART button is pressed which sends a Restart (RST) command

Troubleshooting Tips

Make sure both the HMI and the SilverDust controller have been setup for the same communication parameters. See the "Hardware and Software Setup" section of this document for the recommended settings. When the HMI communicates with the SilverDust controller the red LED on the SilverDust should flash constantly verifying communication. If this is not the case, double-check the wiring. The SilverDust controller can only communicate with one device at a time, make sure not to communicate with the QuickControl software and have the Red Lion HMI simultaneously.

To verify the SilverDust controller is working, disconnect the Red Lion HMI and use QuickControl's Register Watch feature to write to and read from the registers instead of the HMI.