Analog Inputs

Included Files:
- ACR Velocity.qcp
- ACR Torque.qcp
- ACR Pos Error.qcp

Many example program files can be found in the folder:
"…/QuickControl/QCI Examples/Analog Inputs"

Overview

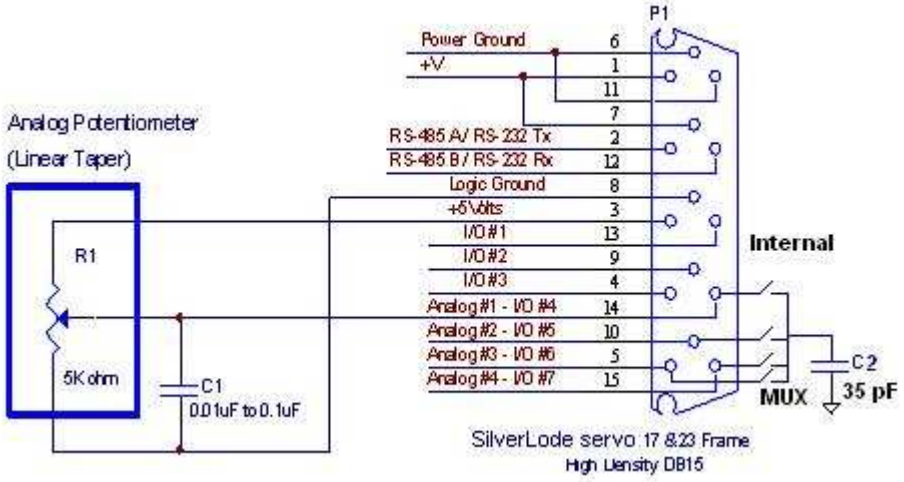
The controllers of SilverLode product family use one of two analog input voltage ranges as follows:

SilverNugget	0 to +5.0 V
SilverDust	0 to +3.3 V
SilverSterling	0 to +3.3 V

When activated as single channels, analog channels 1 through 4 are assigned to I/O lines 4 through 7, respectively. When activated as differential channels, analog channels 5 and 6 are assigned to I/O lines 4 - 5 and 6 - 7, respectively. SilverSterling Analog Channel 1 is assigned to I/O 4. the other I/O are available on the internal 60 pin header. The remaining channels are reserved for internal signals as detailed below.

Hardware

For accurate results, the analog input I/O lines should be driven from a low impedance source (100 ohm or less) or should be shunted with a capacitance across the input to provide a low impedance source for the internal Analog to Digital Converter (See Figure 1). C1 is typically 0.01uF. C2 is an internal capacitor, which is the primary signal filter.



QuickControl® and QCI® are Registered Trademarks of QuickSilver Controls, Inc.
SilverLode™, SilverNugget™, SilverDust™, SilverSterling™, PVIA™, QuickSilver Controls™, and AntiHunt™ are trademarks of QuickSilver Controls, Inc..

Typical wiring of I/O lines for analog input

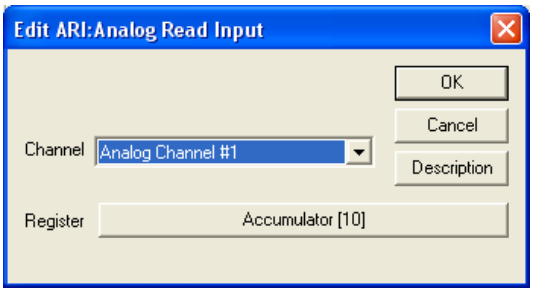
The internal analog to digital converter (ADC) provides 10-Bit resolution of the signal; the SilverSterling uses a 12-Bit resolution converter. This value is scaled up by a factor of 32 (8 for SilverSterling) and filtered via a digital filter. Due to the averaging of multiple signals in the presence of sufficient noise, this usually yields another 2-3 bits of dynamic range (effectively 12-13 bits; 14-15 bits for SilverSterling). Analog signals with a range of 0 to +5 VDC will result in 4.88 mV (SilverNugget) per internal ADC count. Analog signals with a range of 0 to +3.3 VDC will result in 3.22mV (SilverDust) per increment internal ADC count. For the SilverSterling this is 0.805 mV/ count. Including the x32 (x8) scaling factor, this is :

(2^{10} bits –1) * 32 = 32736 counts, 5 volts / 2^15 =0 .152 millivolts/increment (after the filter)
(2^{10} bits –1) * 32 = 32736 counts, 3.3 volts / 2^15 = 0.10 millivolts/increment (after the filter)

(Note: without the effects of filtering and noise induced dithering, the output would change in increments of 32 counts due to the basic 10 bit resolution of the ADC. With a proper input source impedance, typical noise is on the order of 3 counts RMS indicating an effective resolution of better than 12 bits.)

The inputs are taken with respect to the common voltage return ground. For high resolution readings, it is suggested either to use differential input methods (described below) to minimize any ground shift effects with varying loads, or to reference the input sensor to the logic ground connection, avoiding all other ground references to prevent ground loops.

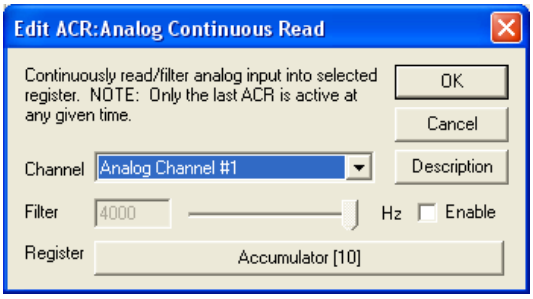
SilverLode Commands for Reading Analog Inputs



Analog Read Input (ARI): ARI does a single read of a selected analog input and places the value in a selected user data register. See SilverLode Command Reference for details.

Single channel ADC readings are scaled and filtered to produce an unsigned 15-Bit value before being stored in a designated user data register. Single channel readings have a data range from 0 to 32652 (approximately 0 to [2^{15} – 1]) across the analog input voltage range. For example, on the SilverDust; a reading of 3.3V would nominally correspond to a register value of 32652. The slight reduction in range from the expected 32767 is due to the 10 bit ADC as well as some rounding in the filter (The maximum value from the ADC is 1023 * 32 = 32736; Truncation due to finite resolution in the filter accounts for the remainder.)

Differential channel readings are performed by simultaneously sampling the related channels, converting and then taking the difference (see below for details) are converted to signed 15-Bit values with a range from –32562 to +32652 counts.



Analog Continuous Read (ACR): ACR puts the SilverLode servo into a continuous read cycle that takes an analog reading every servo cycle (120 usec.). Putting the SilverLode servo into analog continuous read has no effect on other operations; the servo performs the readings in a "background routine" that is designed to work without affecting the controller performance. When continuously reading an analog input the analog value is placed in the selected user data register after each reading (every 120 usec). This provides a continuous update of the analog signal allowing for better tracking. See SilverLode Command Reference for details.

NOTE: Analog continuous read can only be set up for one channel at a time. The command can be re-issued at any time to switch to a different channel or redirect the data to a different user data register.

ADC readings taken using the Analog Continuous Read (ACR) command are automatically filtered with a low pass filter having a roll off frequency of 30 Hz. SilverDust units with a software version of 36 or higher have the option of additional read channels as well as an optional secondary filter.

Analog Channel Definitions

Analog Channel	External I/O Line(s)	Single or Differential	Description
#1	#4	Single	10-bit resolution scaled to 15-bit value
#2	#5	Single	10-bit resolution scaled to 15-bit value
#3	#6	Single	10-bit resolution scaled to 15-bit value
#4	#7	Single	10-bit resolution scaled to 15-bit value
#5	#4 - #5	Differential	11-bit resolution scaled to 16-bit value
#6	#6 - #7	Differential	11-bit resolution scaled to 16-bit value
V+ (Main Bus Voltage)	Internal	N/A	Raw ADC of main bus voltage
Temperature	Internal	N/A	Raw ADC of processor temperature
V+ Scale Factor	Internal	N/A	Main bus voltage scale factor
Processor V+	Internal	N/A	Raw ADC of processor voltage
Driver Temp	Internal	N/A	Raw ADC of driver circuit temperature
Processor V+ Cal	Internal	N/A	Processor voltage calibration factor
Velocity	Internal	N/A	Raw Velocity measurement
Torque	Internal	N/A	Raw Torque measurement
Position Error	Internal	N/A	Raw Position Error measurement

Single Channel Inputs

The single channel analog input is the simplest way to input an analog signal. For single channel usage, the analog signal is referenced to the servo logic ground. An input capacitor may help minimize external noise pickup and will provide a low impedance source for the sampling ADC built into the DSP. The low impedance is needed to minimize the voltage variations induced when the ADC samples the input - momentarily connecting its small internal sampling capacitor across the input.

Differential Channel Inputs

Analog “channels” #5 and #6 read the differential analog input for analog channels #1-#2 and #3-#4 respectively. The differential channel analog input provides noise rejection of the analog signals. In addition, the differential readings also double the resolution of the ADC reading from 10-Bit to 11-Bit if both inputs are varied differentially. The value read in by the analog inputs digitally scale from -32562 to +32562. To achieve the full range, both inputs must be active. If on the SilverNugget, for example, channels #1 & #2 are used in differential mode, +32562 will be read when #1 is at +5 volts and #2 is at 0 volts, -32562 will be read when #1 is at 0 volts and #2 is at +5 volts. The two inputs are sampled (essentially) simultaneously, so common mode noise and ground voltage offsets may be effectively canceled.

Special Internal Analog Channels

Several internal signals are also measured via the ADC and may be selected.

V+ Main Bus Voltage: This power supply voltage (and processor voltage is no separate processor input voltage is present) is scaled down by an internal voltage divider. The resulting non-calibrated V+ supply ADC count may be converted to volts using the V+ Scale Factor. The scale factor is a calibration value determined at the factory and stored to non-volatile memory. The voltage is calculated as follows:

V(volts) = (raw ADC) / (V+Scale Factor)

See below for V+ Scale Factor.

Processor Temperature: This is the non-calibrated processor temperature of the SilverLode servo. The temperature sensor is mounted to processor PCB for the SilverNugget, or to processor/driver board for the SilverDust. It measures the component temperature which is above the ambient temperature.

SilverNugget: T(C˚) = (raw ADC – 9011) / 147 + 3
SilverDust: T(C˚) = (raw ADC – 4960) / 99

V+ Scale Factor: This is not really an analog channel; it reports a voltage calibration value for the V+ Main Bus that is stored in the non-volatile memory (NVM).

Processor V+ (not available on all SilverLode products): This is the non-calibrated processor voltage. It requires a calibration factor in order to get a usable result in volts.

V(volts) = (raw ADC) / (V+Scale Factor)

Driver Temp (not available on all SilverLode products): This is the non-calibrated driver temperature of the SilverLode servo amplifier. It is measured via a thermistor bridge and requires a calibration equation for a usable temperature result.

SilverNugget: T(C˚) = (raw ADC –9011) / 147 + 3

Processor V+ Cal (not available on all SilverLode products): This is not really an analog channel; it reports a voltage calibration value for the V+ Processor that is stored in the non-volatile memory (NVM).

Velocity (SilverDust Revision 36 and higher): This data reads from the upper word of register 7. Normally, reading the actual velocity may be somewhat noisy due to the wide bandwidth needed by the velocity signal as used by the control loop coupled with the incremental nature of the encoder from which it is derived. Filtering the velocity signal provides a better view of the actual velocity of the load.

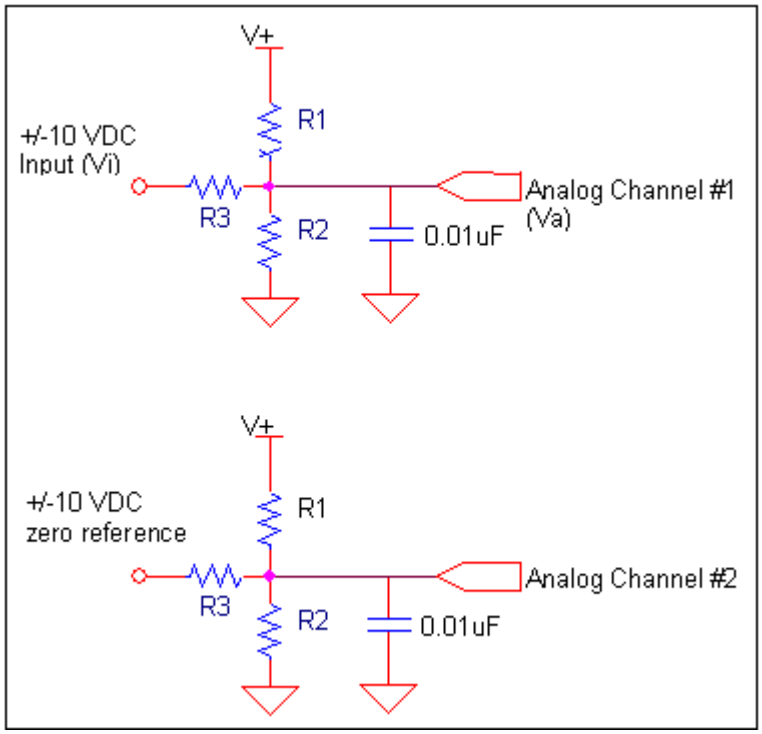
Torque (SilverDust Revision 36 and higher): This data reads from the lower word of register 9. This signal may be filtered (if selected) to remove higher frequency components present due to the incremental nature of the encoder as well as the action of the control loop. This produces a better estimate of the short term average torque than may be determined from a single reading of the raw value.

Position Error (SilverDust Revision 36 and higher): This data reads from the lower word of register 6. The raw error signal is highly dynamic. This function allows the signal to be filtered, thus removing rapidly changing artifacts.

Connecting +/-10 VDC to SilverLode Analog Inputs

Tthe following circuit will convert a +10 to -10 V analog input signal to the 0 to 5V or 0 to 3.3V signal level required by SilverNugget or SilverDust respectively.

Note: The two resistor networks have the same schematic. The +/-10V is read into analog channel #1 through the network as Va. The following equation was used to calculate the resistor values in the table below. V+ is the 5v reference voltage provided via the SMI interface.



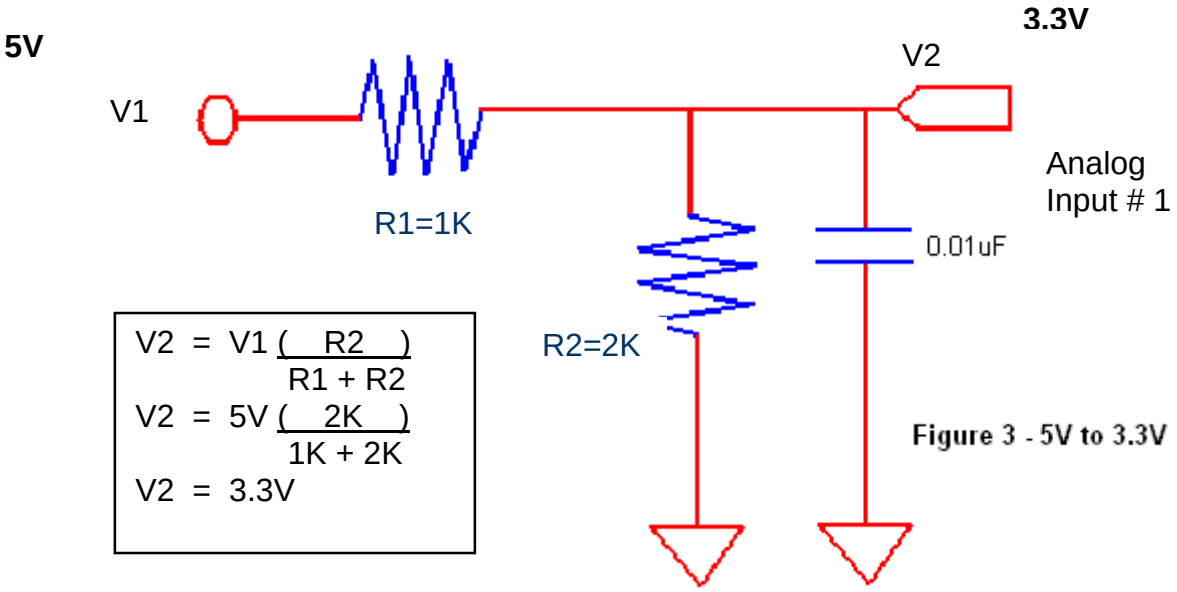
(Vi-Va)/R3 + (V+ - Va)/R1 = Va/R2

SilverLode	V+	R1	R2	R3	Va (Vi=10)	Va (Vi=0)	Va (Vi=-10)
SilverNugget	5V	1K	2K	2K	5.00V	2.50V	0.00V
SilverDust	5V	1K	667	2K	3.33V	1.67V	0.00V

Although the channel #1 could be read by itself, QCI recommends reading channel #1 - #2. Feeding the +/-10V zero reference into channel #2 compensates for differences in the controller's zero reference (logic ground) and the +/-10V zero reference. QCI recommends using 1% resistors.

Converting a 5V Signal to a 3.3V Signal

To convert a 5V signal to a 3.3V signal, use a voltage divider.



$$V2 = V1 \left(\frac{R2}{R1 + R2} \right)$$

$$V2 = 5V \left(\frac{2K}{1K + 2K} \right)$$

$$V2 = 3.3V$$

Figure 3 - 5V to 3.3V

Note the above voltage divider is ideal for converting the SilverMax/SilverNugget 0-5V to the SilverDust 0-3.3V.

A similar conversion can be achieved between using just software. To go from 3.3V to 5V, multiplying the analog value by (5V / 3.3V). To go from 5V to 3.3V, multiply the analog value by (3.3V / 5V).

Examples:	SilverNugget/SilverMax	->	SilverDust
	3.3V = 21,626.22	21,626.22 * (5 / 3.3) =	32,767 = 3.3V
	SilverDust	->	SilverNugget/SilverMax
	2.0V = 19,858.79	19858.79 * (3.3 / 5) =	13,106.8 = 2.0V

Convert 5V to 3.3V.qcp

This program uses a special CLD multiply command to, in software, convert 5V to 3.3V. See Technical Document "QCI-TD026 Calculation Commands" for details.

Line# Oper	Label	Command
1:ARI		Analog Read Input: "Accumulator[10]" = "Analog Channel #1"
2:REM		Multiply by 3.3 / 5 Scale factor is (3.3 / 5) * 65536 = 43254
3:CLD		Converted ADC reading[11] = (Accumulator[10] * 43254)>>16

Example: V+ Main Bus Voltage

The V+ main bus voltage channel is read using ARI or ACR. In the example shown, a single read of the channel is done and placed in the accumulator (Data Register #10). A single read of the V+ Scale Factor is also taken and placed in Data Register #11. The accumulator is divided by Register #11 to achieve the calibrated voltage. This is stored in Data Register #11 for future use.

Voltage Calibration Equation

RV = Raw V+ Main Bus Voltage
CF = V+ Scale Factor

V+ Buss Voltage (vdc) = RV / CF

Line# Oper	Label	Command
1:REM		This Program reads the current V+ power supply voltage converts it to a usable value and stores it in User Data Register #11
2:REM		Read in the "raw" V+ voltage value into the accumulator
3:ARI		Analog Read Input: "Accumulator[10]" = "V+ (Main Buss Voltage)"
4:REM		Read the V+ scaling factor form Non-Volatile memory and store it in User Data Register #11
5:ARI		Analog Read Input: "User[11]" = "V+ Scale Factor"
6:REM		Divide the V+ raw value by the scaling factor to give a reading in "Volts"
7:CLC		"Accumulator[10]" = "Accumulator[10]"/"User[11]"
8:REM		Write the calibrated voltage value to Data Register #11
9:CLC		"User[11]" = "Accumulator[10]"

Reading & Calibrating Main Bus voltage

Example: ACR Velocity.qcp

The Velocity channel can be read using ARI or ACR. Using ARI to read Velocity is the same as reading the lower half of register 7. Using ACR allows the typically noisy velocity signal to be filtered. In the example shown, an ACR is used to read Velocity into User[25] through a 50 Hz low pass filter.

Lines 5 through 7 are some simple moves to allow us to analyze the low pass filter effect.

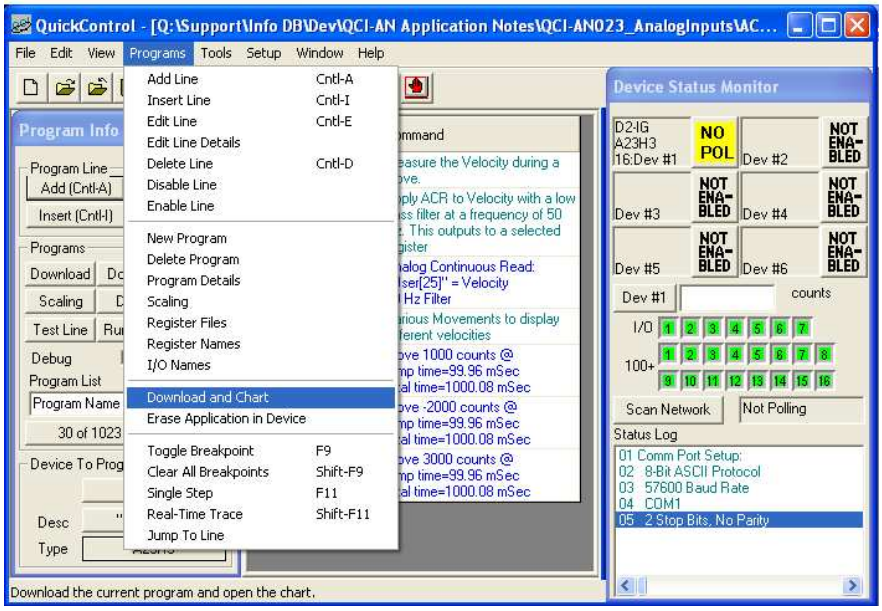
Line# Oper	Label	Command
1:REM		Measure the Velocity during a move.
2:REM		Apply ACR to Velocity with a low pass filter at a frequency of 50 Hz. This outputs to a selected register
3:ACR		Analog Continuous Read: "User[25]" = Velocity 50 Hz Filter
4:REM		Various Movements to display different velocities
5:MRT		Move 1000 counts @ ramp time=99.96 mSec total time=1000.08 mSec
6:MRT		Move -2000 counts @ ramp time=99.96 mSec total time=1000.08 mSec
7:MRT		Move 3000 counts @ ramp time=99.96 mSec total time=1000.08 mSec

Charting register 25 verses the unfiltered Velocity is a good way to understand the effects of the filter.

To access the charting program:

Programs->
Download and Chart
(see figure).

This will download the current program and bring up the strip chart dialog box.



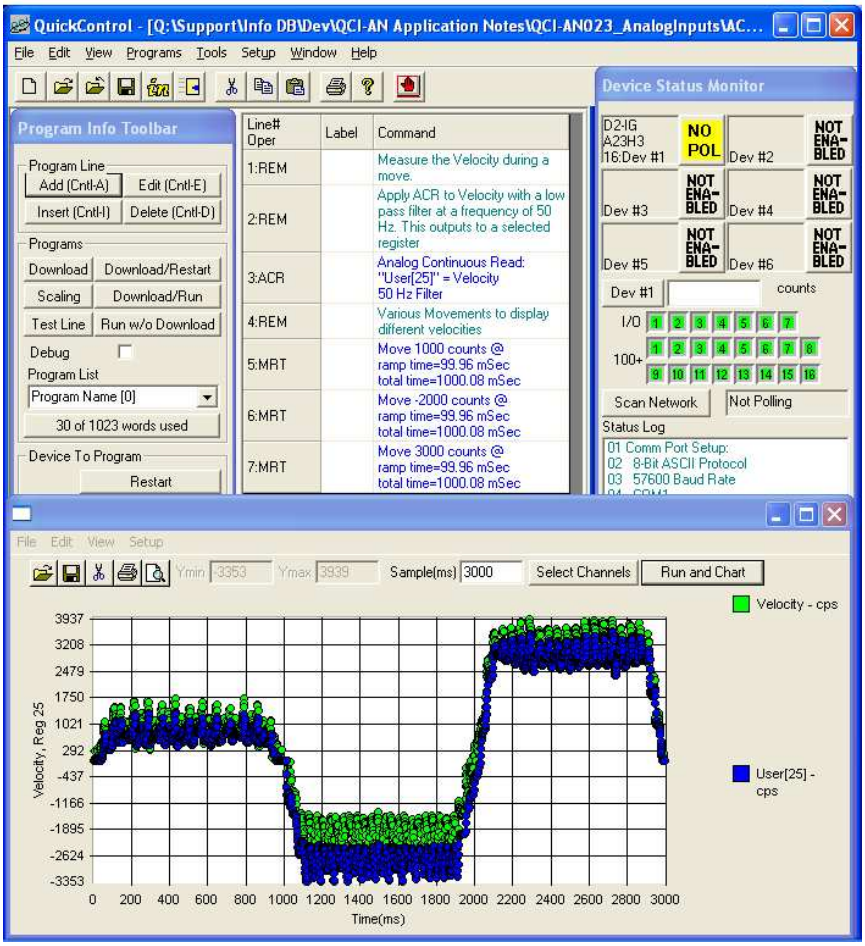
Click ‘Select Channels’ and select Velocity and User 25.

Click ‘Run and Chart’.

The program will execute and upon completion, the strip chart data will display.

Notice how the 50Hz filter smoothes out the raw velocity.

Figure 6 - Strip Chart Dialog Box



Example: ACR Torque.qcp

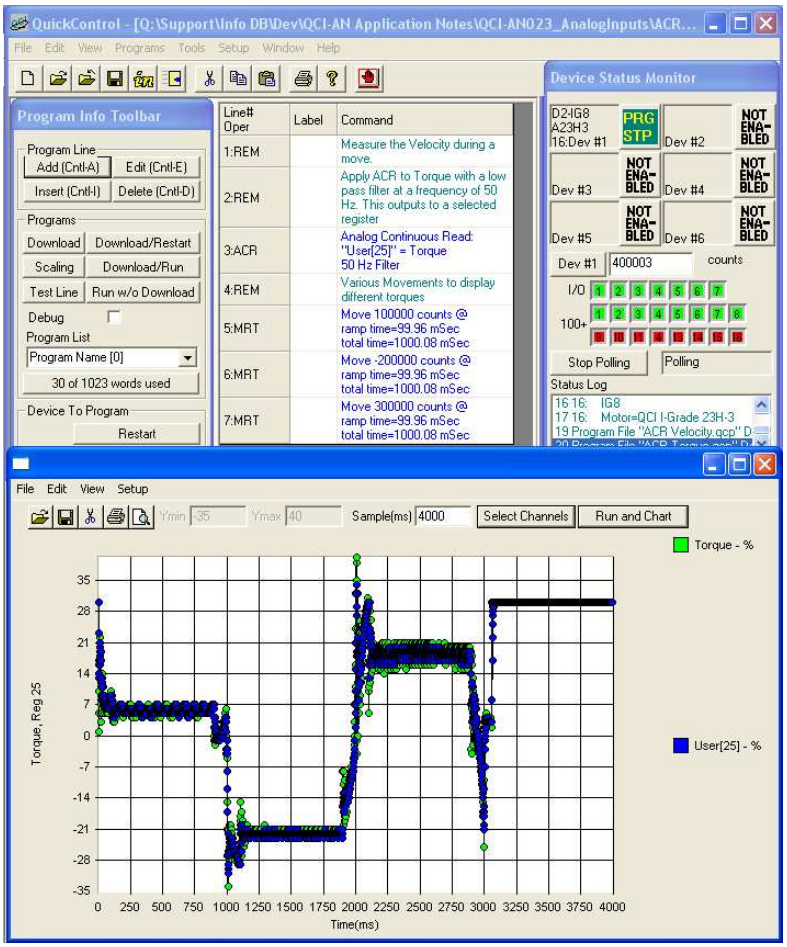
The Torque channel can be read using ARI or ACR. In the example shown, an ACR is used to read Torque into User[25] through a 50 Hz low pass filter.

ACR Torque.qcp

Line# Oper	Label	Command
1:REM		Measure the Velocity during a move.
2:REM		Apply ACR to Torque with a low pass filter at a frequency of 50 Hz. This outputs to a selected register
3:ACR		Analog Continuous Read: "User[25]" = Torque 50 Hz Filter
4:REM		Various Movements to display different torques
5:MRT		Move 100000 counts @ ramp time=99.96 mSec total time=1000.08 mSec
6:MRT		Move -200000 counts @ ramp time=99.96 mSec total time=1000.08 mSec
7:MRT		Move 300000 counts @ ramp time=99.96 mSec total time=1000.08 mSec

This program configures User[25] to store the torque data. This data runs through a 50Hz low pass filter. Without the filter, the raw torque data contains more high frequency components associated with the operation of the control system. The filtered data may be seen in the chart.

Download and Chart

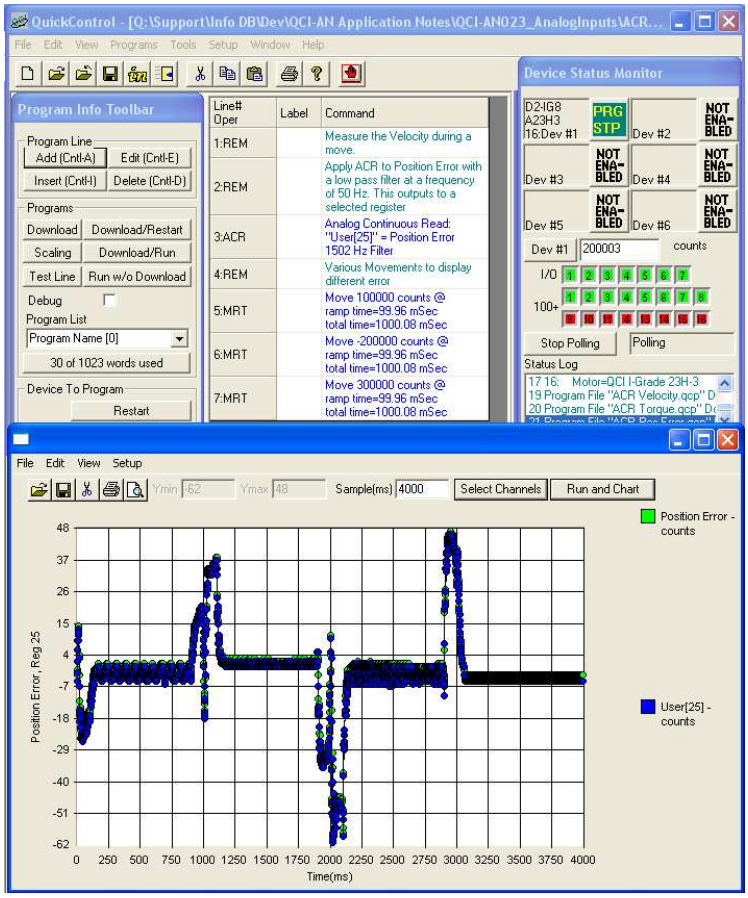


Example: ACR Pos Error.qcp

The Position Error channel can be read using ARI or ACR. The ACR provides the option of using a filter to remove high frequency components associated with the use of an incremental encoder. In the example shown, a continuous read of the channel is used, with the filtered output placed in User[25].

ACR Position Error.qpc

Line# Oper	Label	Command
1:REM		Measure the Velocity during a move.
2:REM		Apply ACR to Position Error with a low pass filter at a frequency of 1500 Hz. This outputs to a selected register
3:ACR		Analog Continuous Read: "User[25]" = Position Error 1502 Hz Filter
4:REM		Various Movements to display different error
5:MRT		Move 100000 counts @ ramp time=99.96 mSec total time=1000.08 mSec
6:MRT		Move -200000 counts @ ramp time=99.96 mSec total time=1000.08 mSec
7:MRT		Move 300000 counts @ ramp time=99.96 mSec total time=1000.08 mSec

.

Download and Chart