Events and Sequence

General

Events and Sequences are low level elements of QuickControl®. Advanced users will find Events and Sequences useful for the following tasks:

- Prototype Host Level Control
- Advanced Data Collection and Logging

Audience

This application note is for advanced users familiar with QuickSilver's communication protocols and programming with QuickSilver Control's program files (.qcp files). See Communications in SilverLode User Manual for details.

Events

An Event is just another name for a serial communication packet. QuickControl uses Events anytime it communicates with a device (i.e. SilverDust). See the SilverLode Command Reference for a description of all the available commands.

The actual packet sent to the external device will differ from the Event in several ways. Most external devices require an extra “wrapper” or protocol to encapsulate the command message. QuickControl automatically does this for you. QuickControl supports more than one protocol, which are selected during Comm Port setup.

Protocols reformat the command message to conform to specific rules. For example, the QuickSilver 8-Bit ASCII protocol will convert the Data Fields into an ASCII string of space delimited decimal numbers.

```
@15 135 100 2000 2000 0 0 <CR>
```

QuickControl's Data Monitor tool (see Tools menu) provides a view of the QuickControl transmit/receive data in the actual data format.

Events are made up of two other elements, Event Templates and Data Templates. Event Template contains the format of the packet, including ID, command number and command parameter format. Data Templates are used to process the return data. Both of these elements are described in detail below.

Event Templates

The Event Templates define the Event format by storing the redundant command parameters like data labels, data type and data ranges. Many Events can be "based off" the same template. The only difference in these Events would be the command parameter data. Event Templates are created by QuickSilver and are not user editable.

Data Templates

Data Templates complement the Events by allowing application specific information generated by a device to be retrieved. The data templates provide a customizable means of formatting the data received into usable information.

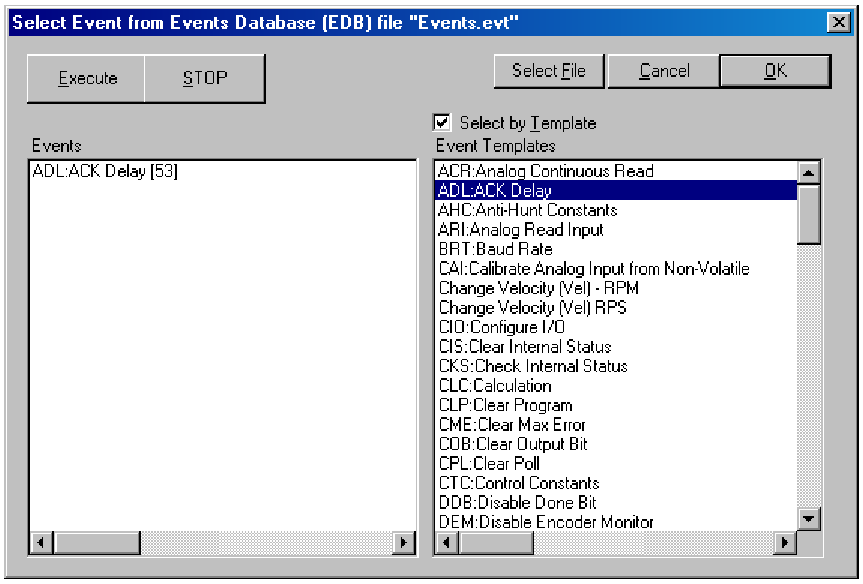
A limited amount of data manipulation using simple math routines like addition, subtraction, multiplication, and division can be performed within the Data Templates and data can be queried to seek specific data values. In addition, Data Templates provide a means of routing data from one Event to another (see Sequence Buffers for more details). If an Event requires dynamic parameters, the parameters can be retrieved from previous Events, mathematically manipulated, and then used in another Event.

Events Database

A collection of Events, Event Templates and Data Templates is called an Events Database (EDB). Each Sequence file (.ctl) and Program File (.qcp) contains an EDB of the Events it uses. There is a generic EDB that is usable for everybody called "Events.evt" and there is a system EDB used only by QuickControl called "System.evt".

Event Details

Select an Event to Edit



Select by Template: ✓

Events:
- ADL:ACK Delay [53]

Event Templates:
- ACR:Analog Continuous Read
- ADL:ACK Delay
- AHC:Anti-Hunt Constants
- ARI:Analog Read Input
- BRT:Baud Rate
- CAI:Calibrate Analog Input from Non-Volatile
- Change Velocity (Vel) - RPM
- Change Velocity (Vel) RPS
- CIO:Configure I/O
- CIS:Clear Internal Status
- CKS:Check Internal Status
- CLC:Calculation
- CLP:Clear Program
- CME:Clear Max Error
- COB:Clear Output Bit
- CPL:Clear Poll
- CTC:Control Constants
- DDB:Disable Done Bit
- DEM:Disable Encoder Monitor

Select an existing Event to edit from the menu:

Edit->Events->Edit Event

The following Select Event dialog box is displayed.

This dialog box is used any time an Event needs to be selected from an Events Database.

The main part of the dialog box is broken up into two list boxes. From left to right, they are a list of Events and Event Templates. The design of this dialog box helps the user navigate through an Events Database with hundreds of Events. The fastest way to select an Event is to first select an Event Template and then select one of the Events that are based off that template.

Select an Event by double clicking on the Event or by selecting the event and pressing OK.

You can bypass selecting the Event Template by unchecking “Select by Template” . This is often a better way of selecting an Event from an Events Database with only 10 or so Events.

The dialog box caption (top of box) displays the Events Database currently being used.

The other dialog box fields are explained below.

Execute
Pressing this button will execute the selected Event.

STOP
Pressing this button will stop all device connected to the PC.

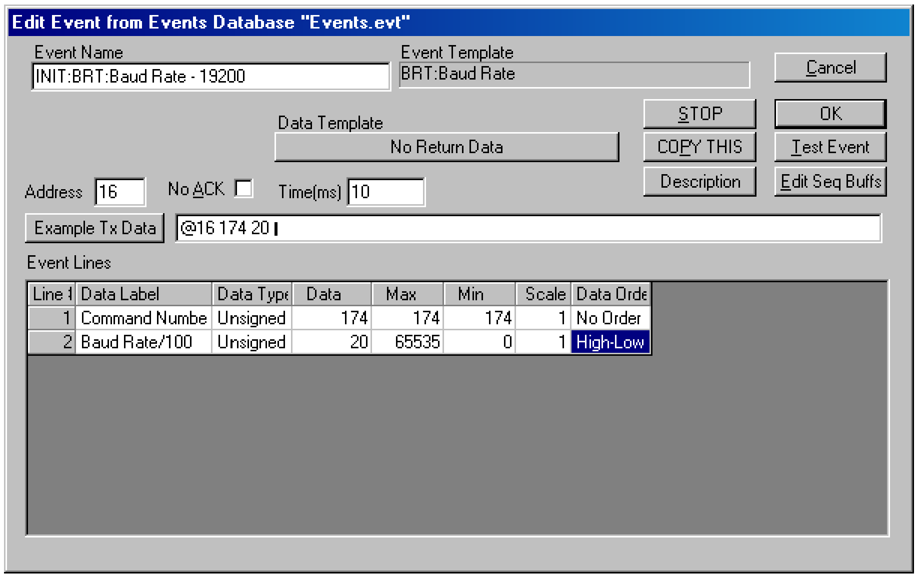
Select File
Press this button if you want to change the Events Database (EDB) file.

Select by Template
When this box is checked, only the Events based off the selected Event Template will be listed.

Tip: Once a list box item is selected, pressing any letter on the keyboard will scroll the list box to the first entry that starts with that letter.

Edit an Event

Select Event to edit (see above) and the Edit Event dialog box will appear.


Edit Event from Events Database "Events.evt"

Event Name: INIT:BRT:Baud Rate - 19200
Event Template: BRT:Baud Rate
Data Template: No Return Data
Address: 16
No ACK: ☐
Time(ms): 10
Example Tx Data: @16 174 20

Event Lines

Line	Data Label	Data Typ	Data	Max	Min	Scale	Data Orde
1	Command Numbe	Unsigned	174	174	174	1	No Order
2	Baud Rate/100	Unsigned	20	65535	0	1	High-Low

The dialog box caption (top of box) displays the Events Database currently being used.

The following fields make up the dialog box.

Event Name
The Event name is used to distinguish one Event from another. The name also appears in list boxes for selecting Events.

Event Template
The name of the Event Template this Event is based off.

Data Template
The name of the Data Template, if any, the Event will use to process the data returned by the device. Press this button to bring up the Edit Data Template Dialog Box where you can select a new Data Template or to remove the current selection.

STOP
Pressing this button will send a Stop command to all devices attached to the PC.

Copy This
The Copy This button creates a new Event that is an exact copy of the current Event except the string “(copy)” is appended to the end of the name.

Description
A description is giving more detail about the Event. It can describe the function of the Event or contain other specifics. The description can be any length.

Test Event
Press this button to “test” the Event. The Event will be packetized and sent to the device.

Edit Sequence Buffers
Press this button to edit the Sequence Buffers (see Sequence Buffers for details).

Address
The address is the Unit ID of the device this Event will be sent to.

No ACK
Check this box when an ACK is not expected. There are some commands that do not send back ACKs (i.e. BRT:Baud Rate).

Time
The time is the expected duration in milliseconds of the Event execution. It is only used in Sequences (see below). If you are going to use the Event in a Sequence, enter a valid value in this field. It must be greater than 10ms and should represent the actual time it takes for the device to complete the command you send it.

Example:
If the Event is a SilverLode command to go 1 revolution at 1 RPS, the Event Time should be about a little over 1 sec.

Example Tx Data
Example Transmit Data displays the actual data packet for this command. This data is sent to the device over the serial port. Press the button to change data formats. As you change data in Event Lines (see below), this field will update. It is useful to construct example commands here and copy them into your application.

Event Lines
The Event Lines make up the majority of an Event. These fields contain the information that defines what the Event is. This data is sent to an external device in the form of a packet.

Each line represents on element of the packet with the line number representing the order of the elements in the packet. Since the packet is a byte stream, any element larger than one byte (i.e. a Decimal Long requires 4 bytes) must be decomposed into bytes as it is packetized.

The only fields in the Event Data that are editable in this dialog box are the Data fields. The other fields are edited by QCI and are read only here.

Data Label (read only)
This is a label that describes the data field

Data Type (read only)
The type of data element can be defined in this field.

Data
This is the current data value for line. Data is editable.

Max (read only)
This field sets the maximum value data can be. This value will be of the same data type defined in the Data Type field.

Min (read only)
This field sets the minimum value data can be. This value will be of the same data type as defined for the Data Type field.

Scale (read only)
The Scale is a floating-point number that is multiplied by the Data to generate the actual data transmitted

Data Order (read only)
This field establishes the order in which the data bytes are transmitted.

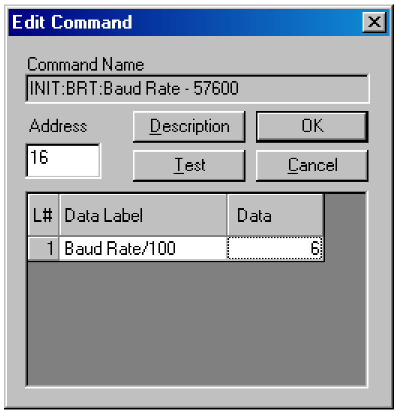
Order	Description
High-Low	High byte first then Low byte
Low-High	Low byte first then High byte
H-MH-ML-L	High byte, Medium-High byte, Medium-Low byte, Low Byte
MH-H-L-ML	Medium-High byte, High byte, Low Byte, Medium-Low byte
L-ML-MH-H	Low Byte, Medium-Low byte, Medium-High Byte, High byte
ML-L-H-MH	Medium-Low byte, Low Byte, High byte Medium-High byte
No Order	No data order is required
Not Used	Not used on this line
Combo	Combine next lines

Transmit Byte Order is the order in which multiple byte data types are transmitted. For example, if a data element is a Decimal Long there are four bytes of data that will be sent. The order that the data is sent is critical to the external device since it must reassemble the four bytes into the required Decimal Long value. If the data is sent in the wrong order the value will be incorrect. In most cases, external devices take data with the High byte sent first followed in order by the next most significant bytes ending with the low byte. The default setting is usually correct.

The Combo Transmit Byte Order selection causes QuickControl to combine the next several lines to make up the indicated data type. For example, if the current Data Type is Unsigned Long, a Combo Transmit Byte Order selection would cause QuickControl to take the next several line (4 bytes total) and combine them into an Unsigned Long.

Lite Edit Event Dialog Box

This small dialog box displays only the essential editable fields and is ideal for fine tuning an Event after it has been created. It is does not take up as much desktop as its big sister and QuickControl allows you to create more than one Event Edit Lite Dialog Boxes at a time.



Command Name: INIT:BRT:Baud Rate - 57600

Address: 16

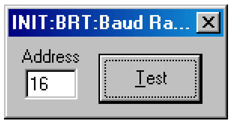
L#	Data Label	Data
1	Baud Rate/100	6

To create an Event Edit Lite Dialog Box select from the main menu,

Edit->Events -> Lite Edit Event.

Execute Event Dialog Box

This small dialog box is ideal for executing utility Events. One or more can be created and moved anywhere on your desktop. They consist of only the Event name and a Test button.



Address: 16

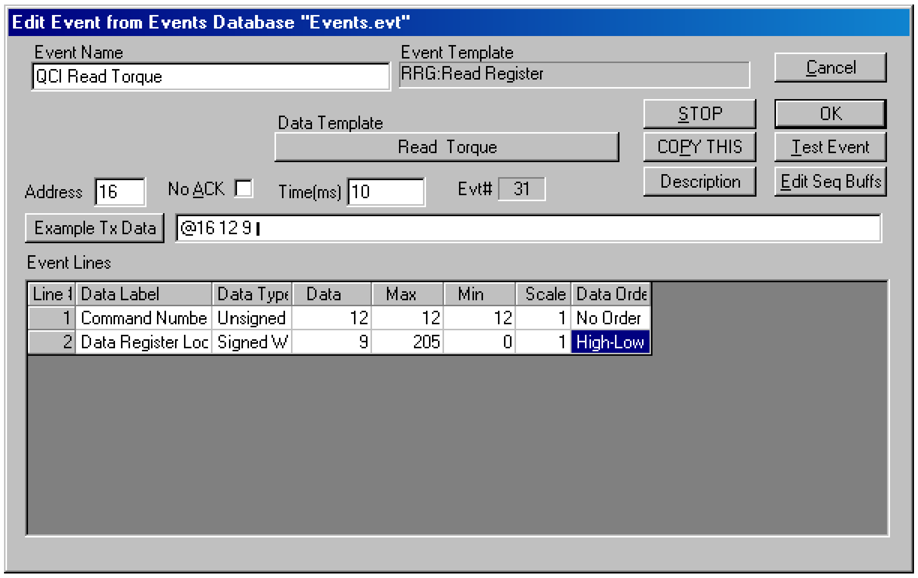
To create an Execute Event Dialog Box select from the main menu,

Edit -> Events -> Execute Event.

Data Template Details

Edit Data Template

A Data Template can be selected from the Event Edit dialog box. For example, from the "QCI Read Torque" Event, a user can edit the "Read Torque" Data Template by pressing "Read Torque"



Event Name: QCI Read Torque

Event Template: RRG:Read Register

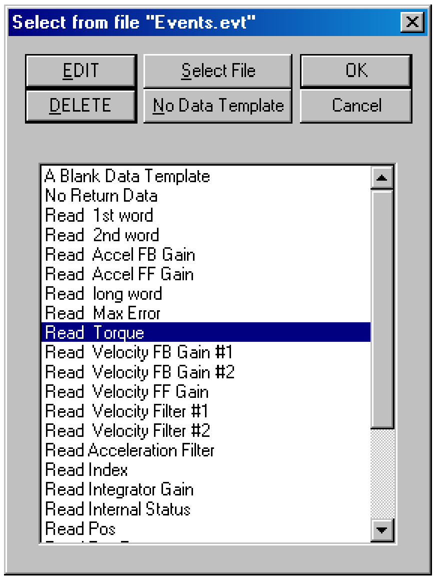
Data Template: Read Torque

Address: 16 No ACK ☐ Time(ms) 10 Evt# 31

Example Tx Data: @16 12 9 I

Line #	Data Label	Data Type	Data	Max	Min	Scale	Data Order
1	Command Numbe	Unsigned	12	12	12	1	No Order
2	Data Register Loc	Signed W	9	205	0	1	High-Low

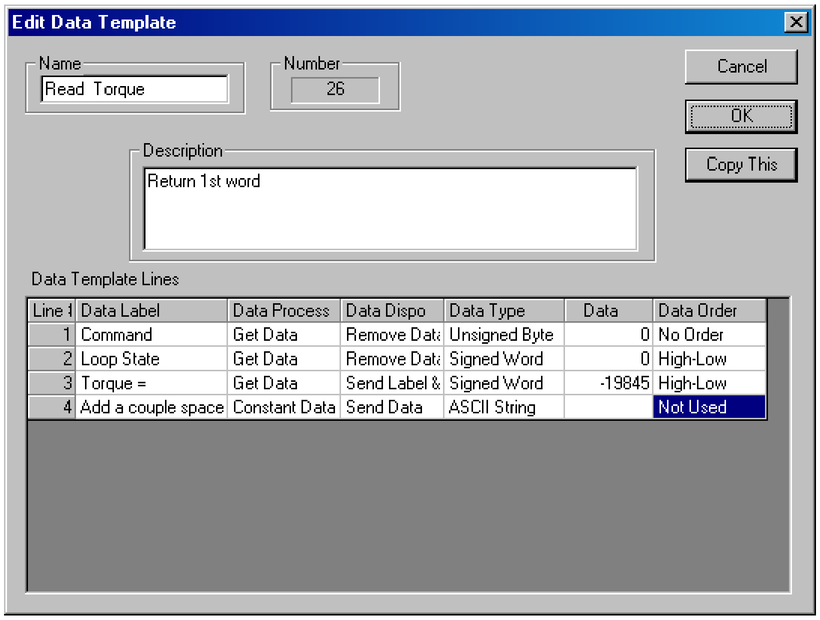
From this dialog box the user can change the Event's Data Template or press the "Edit" button. Press the Edit button to get the following dialog box.



Select from file "Events.evt"

EDIT | Select File | OK
DELETE | No Data Template | Cancel

- A Blank Data Template
- No Return Data
- Read 1st word
- Read 2nd word
- Read Accel FB Gain
- Read Accel FF Gain
- Read long word
- Read Max Error
- Read Torque
- Read Velocity FB Gain #1
- Read Velocity FB Gain #2
- Read Velocity FF Gain
- Read Velocity Filter #1
- Read Velocity Filter #2
- Read Acceleration Filter
- Read Index
- Read Integrator Gain
- Read Internal Status
- Read Pos



Edit Data Template

Name: Read Torque
Number: 26
Description: Return 1st word

Cancel | OK | Copy This

Data Template Lines

Line #	Data Label	Data Process	Data Dispo	Data Type	Data	Data Order
1	Command	Get Data	Remove Dat	Unsigned Byte	0	No Order
2	Loop State	Get Data	Remove Dat	Signed Word	0	High-Low
3	Torque =	Get Data	Send Label &	Signed Word	-19845	High-Low
4	Add a couple space	Constant Data	Send Data	ASCII String		Not Used

The fields for the Edit Data Template dialog box are as follows:

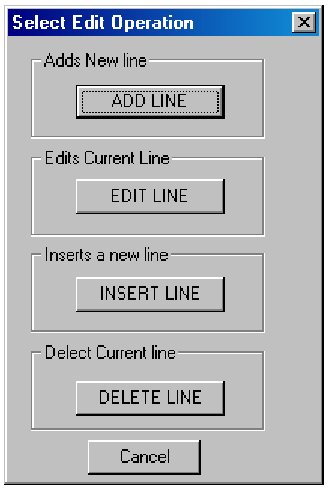
Name
The name for the Data Template.

Number
Index of the Data Template in the Events Database.

Description
The description is constructed to give more detail about the Data Template.

Copy This
Make a copy of this Data Template.

Data Template Lines
These lines define how the returned data is processed. Like a program, processing starts at line 1 and continues through the last line. See below for details. To edit, delete or add a line double click on the line column (first column). The Select Edit Operation dialog box will appear.



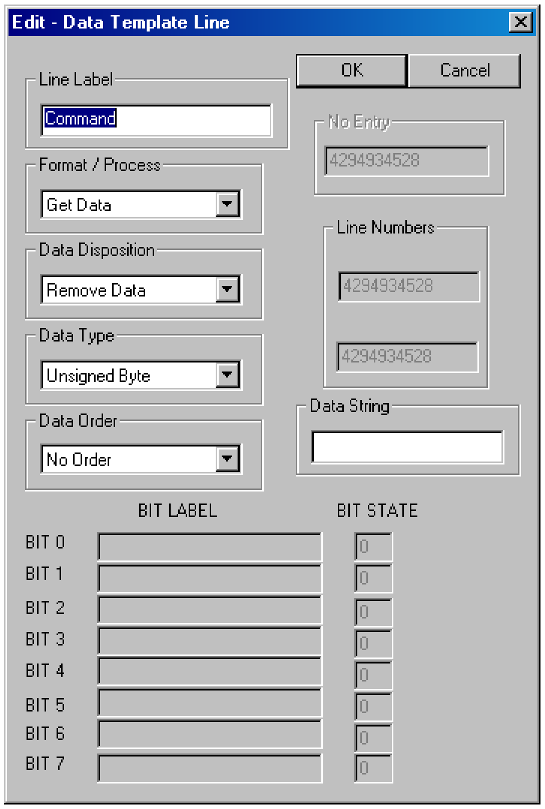
Select Edit Operation

Adds New line: ADD LINE
Edits Current Line: EDIT LINE
Inserts a new line: INSERT LINE
Delect Current line: DELETE LINE
Cancel

Edit Data Template Line Dialog Box

From the Select Edit Operation dialog box, press the EDIT Line button. The Edit Data Template Line dialog box will appear.

Line Label
This label describes the Format or Process field. The label can be somewhat descriptive so as to aid the user in data entry. Labels can be inserted into the data stream to add meaning to the received data and can also be used to signal a condition.



Line Label: Command
Format / Process: Get Data
Data Disposition: Remove Data
Data Type: Unsigned Byte
Data Order: No Order
No Entry: 4294934528
Line Numbers: 4294934528, 4294934528
Data String:

Format/Process
See Data Template Format/Process below for details on this field.

Data Disposition
See Data Template Data Disposition below for details on this field.

Data Type
See Data Template Data Type for details on this field.

Data Order
This field establishes the order in which the data bytes are received from the external device.

Specified Data Fields
Each Process or Format has a different set of parameters that control its activity. For processes that involve using default data, line numbers, buffer numbers or data sizes, three two specified data fields are used to store the data. When a process or format requires specific data, the field(s) will be activated and labeled accordingly.

Data String
Processes that require an ASCII String use this field.

Data Template Format/Process

Get Data
This format routine gets a single data element of the Data Type from the received packet. The data is processed according to the Data Disposition.

Get Data Array
This formatting routine gets an array of data elements of the Data Type defined from the Receive Buffer. The data is then processed according to the Data Disposition. The size of the data array is entered on this line.

Constant Data
Data can be entered into the Data Template in a constant form. The data can then be used in other processes for more complex operations. Data is entered on this line in the form defined by the Data Type. Constant data has some special values for getting other system data such as Sequence time, Current time or Carriage Return and Line Feed.

Repeat to Line Number
This process will cause the operation to repeat to the indicated line. This will allow formatting or processing to repeat itself where needed. This is very useful when a data pattern repeats itself.

Add Line1 to Line2
This process adds the first indicated line to the second indicated line and puts the result on the current line. The resulting data can then be used in other operations. All data is converted to floating point.

Sub Line1 from Line2
This process subtracts the first indicated line from the second indicated line and puts the result on the current line. The resulting data can then be used in other operations. All data is converted to floating point.

Multiply Line1 by Line2
This process multiplies the first indicated line by the second indicated line and puts the result on the current line. The resulting data can then be used in other operations. All data is converted to floating point.

Divide Line1 by Line2
This process divides the first indicated line by the second indicated line and puts the result on the current line. The resulting data can then be used in other operations. All data is converted to floating point.

Clip Data on Previous Line
Clip the data on the previous line to be within the specified limits. If the limits are reversed, dead band is produced. (A zero will be the resultant as long as value is between limits.)

Put in Buffer
This process takes the data from an indicated Data Template line and places it into a Sequence Buffer (see Sequence Buffers for more details). The data that is placed in the buffer is converted to floating point before being moved.

Get From Buffer
This process gets data from a Sequence Buffer on puts it on the current line (see Sequence Buffers for more details). The data can then be used by other applications for more complex processing.

AND Line1 with Line2
This process does a bitwise AND on the first indicated line with the second indicated line and puts the result on the current line. The resulting data can then be used in other operations. All data is converted to floating point. The AND is done on a bit-by-bit bases.

OR Line1 with Line2
This process does a bitwise OR on the first indicated line with the second indicated line and puts the result on the current line. The resulting data can then be used in other operations. All data is converted to floating point. The AND is done on a bit-by-bit bases.

Send Array to Chart
Used by QuickControl's Control Panel and Chart features. Factory use only.

Data Template Data Disposition
This field of the Edit Data Template Line dialog box is used to send data that has been received and formatted or processed to QuickControl's Sequence Monitor screen and to the Device Status Monitor. The same can be done with Data Template line labels and generic data strings. Data can also be removed, stored, or processed. Setting the data disposition will cause the desired process.

Remove Data
Any data on this current line is deleted.

Send Data
This will send the data only to the screen. No carriage return, line feed or spaces are added to the data.

Store Data
Whatever data is on this current line is left there. Data is stored in the Data Template. The last data left on the line may be saved with the Data Template. This data can actually be seen on the Data Template line when editing the Data Template after it has been used.

Send Label
This sends the label only of the current line. This can be used when testing data to send out test information. No other data is sent.

Send Label and Data
This sends out both the line label and the data. This works like Send Data and Send Label together.

Send String
This sends the data string from the current line. This can be used when testing data to send out test information. No other data is sent.

Send String and Data
This works just like Send Label & Data except the data string is sent instead of the label.

Process Only
Same as Store Data except the data is deleted when Data Template processing is complete.

Data Template Data Type
This field of the Edit Data Template Line dialog box is used to define the data element type. These include all normal Data Types and the following additions.
For each Format or Process line, certain data types can be selected. Each format or process has a set of types specific to its operation. When a format or process is selected, only the appropriate data types are made available. The following list shows all the data types used:

ASCII String
This data type is like a common "C" language ASCII character string. The last character in the string is a NULL (0) value that terminates the string. When a string is used as a constant it is entered in the String Data text box. Example of an ASCII String: "The quick gray fox jumps over the lazy brown dog".

Line Number
This is a Data Template line number. The numbers are always displayed as unsigned decimal. The number is range limited by the number of lines in the Data Template and the position of the current line.

Buffer Number
This is a Sequence Buffer number (see Sequence Buffers for more details). The number indicates the buffer location for getting and putting data. Data put into a buffer is always converted to floating point first. Sequence Buffers contain only floating-point data.

Repeat Number
This data type is the number of "repeats" that is used in the Repeat To Line# process. The ranged is limited for storage reasons. If too large a value is used, a time-out in data reception can occur.

Carriage Return
As a convenience this data type is setup for sending an ASCII Carriage Return.

Carriage Return and Line Feed
As a convenience this data type is setup for sending an ASCII Carriage Return and an ASCII Line Feed.

Sequences

A Sequence is a list of Events arranged together so that each is executed in a defined order and at a precise time, making it simply a series of timed Events. Each Event that is added to a Sequence is given an exact execution time. This execution time can be made absolute with respect to the start of the Sequence, or it can be made relative to the execution time of a preceding Event. By connecting the execution time to a preceding Event, the execution times can be easily modified.

What Happens During Sequence Processing?

A number of processes occur during the execution of a Sequence. Each line of the Sequence defines which operation will be performed. The processes occur according to the following:

If a Sequence line is an Event set to execute at time "0", the Event command will be the first operation to take place. The command is sent out, the Event time-out is started and then the Sequence is incremented to the next line.
If the Event was acknowledged, the next step is to process the data. There are a number of different cases that can occur. 1)For commands that do not require any information back, the acknowledge will not contain any data and so the process is complete. 2)If the acknowledge is a communications or protocol error, the error is reported to the status screen. In most cases

this type of error will cause the Sequence to halt. 3)When a response is required, data will be available that can be processed.

Data is processed using the Data Template specified by the executing Event. The template defines a series of operations that will take place. These operations can manipulate the received data, previous data or data constants built into the template.
The Event execution finishes by sending the received data, status information or an error code. The message appears in the Sequence Monitor dialog box (see Sequence Monitor below) and Sequence execution continues.

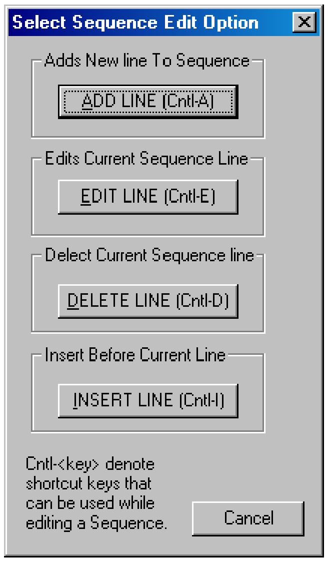
The Sequence continues by updating the Sequence timer and counter. If the Sequence time has expired, the Sequence will end. If the Sequence continues, the next line of the Sequence is checked to see if it will execute on the next timer interrupt. If so, the process that is to take place will be set up for the next interrupt period. If that process is an Event requiring a parameter substitution, the substitution will be done at this time so that the command will be ready for execution.

The Sequence will now wait until the next timer interrupt. Execution of the Sequence will continue until an error occurs, the operators stop it or the Sequence completes normally.

Creating a Sequence

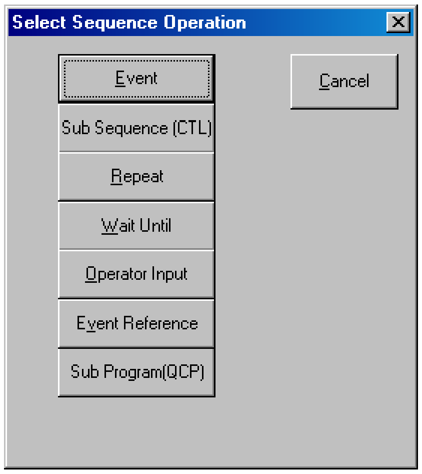
Sequences are created similar to a document in a word processor. From the main menu select,

File ⇒ New Sequence File


Select Sequence Edit Option
Adds New line To Sequence
ADD LINE (Cntl-A)
Edits Current Sequence Line
EDIT LINE (Cntl-E)
Delect Current Sequence line
DELETE LINE (Cntl-D)
Insert Before Current Line
INSERT LINE (Cntl-I)
Cntl-<key> denote shortcut keys that can be used while editing a Sequence.
Cancel

This will open a Sequence document with only one line set to “Not Defined”. New lines can now be added to the Sequence (see Editing a Sequence).

Editing a Sequence

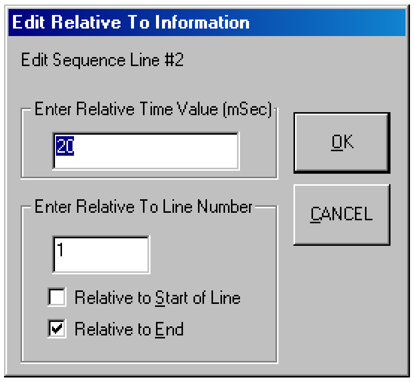


Editing a Sequence or adding a line to a Sequence is done by double-clicking on the Line# & Operation Label column (first column).

When Add a Line, Insert a Line or Edit Line is selected, the Sequence Operation Selection dialog box will appear.

Select the desired operation and following the prompts. See below for details on each operation.

After an operation is selected, you will be prompted to enter the Relative To information (see Sequence Timing).



Edit Sequence Line #2

Enter Relative Time Value (mSec): 20

Enter Relative To Line Number: 1

[ ] Relative to Start of Line

[✓] Relative to End

Deleting a Sequence Line

Double-click on the left column to bring up the Sequence Selection Operation dialog box. A line can only be deleted if there is no other line relative to it.

Executing a Sequence

From the main menu select,

Sequences ⇒ Sequence Exec.

The Sequence Monitor dialog will be displayed from which the Sequence can be set up and executed.

Sequence Operations

Events make up the bulk of the Sequence and are the action part of Sequence execution creating the conditions for other operations to act upon. The other operations allow a Sequence to modify its behavior. To edit a sequence see Editing a Sequence.

The following is a list of Sequence Operations:

Events

See above.

Sub-Sequence

Sequences can become very large and difficult to read. Breaking a Sequence into Sub-Sequences makes it easier to create or manipulate complex Sequences. Sub-Sequences are simply Sequence files that have been created separate from a main Sequence using the same form as a Sequence. When a Sequence commands more than one external device, associating each Sub-Sequence with a particular device helps keep commands distinct. Sub-Sequences are called from a main Sequence and will execute in parallel with the main Sequence.

For Sub-Sequences, the Sequence Event Time field is total time of the Sub-Sequence.

NOTE: Sub-Sequences must be in the same folder as the master sequence.

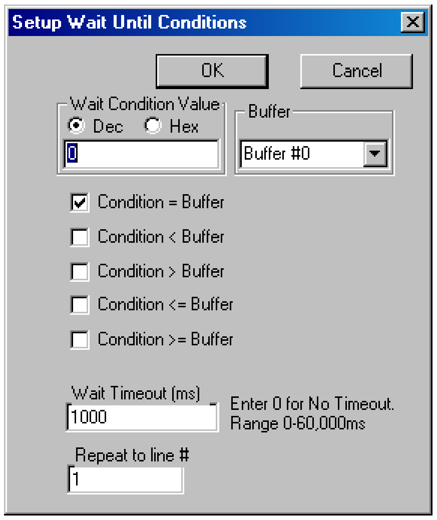
Repeat To

When a block of lines in a Sequence needs to be repeated, instead of writing it over and over again, a Repeat To can be used to loop back any number of times. The user designates a Sequence line to repeat to and sets the number of repeats. Repeat To's can be nested.
For Repeat To, the Sequence Event Time field is the time required to complete all the repeats.

Wait Until

If at some point during execution, the execution must be halted, the Sequence can be told to Wait Until a pre-defined condition is met. The Wait Until can repeat a block of lines like the

Repeat To operation. The line or block of lines will be repeated until the pre-defined condition is reached or until a time-out occurs.


Setup Wait Until Conditions

OK Cancel

Wait Condition Value: [✓ Dec] Hex — 0

Buffer: Buffer #0

[✓] Condition = Buffer
[ ] Condition < Buffer
[ ] Condition > Buffer
[ ] Condition <= Buffer
[ ] Condition >= Buffer

Wait Timeout (ms): 1000 — Enter 0 for No Timeout. Range 0-60,000ms

Repeat to line #: 1

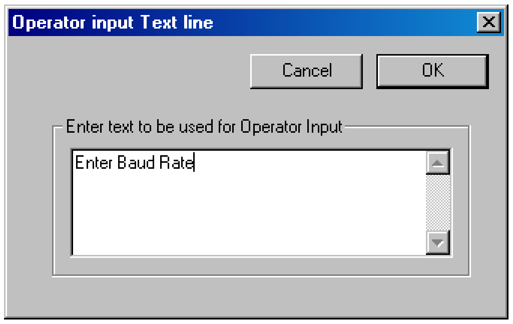
The Wait Until has the following parameters:

- Wait Condition Value
- Numeric value
- Buffer Number
- Condition Equivalence
- Wait Timeout
- Repeat to Line #

The Sequence will loop to the Repeat to Line# until the Wait Condition Value meets the Condition Equivalence to the value stored in the Sequence Buffer # (0-7).
Example:

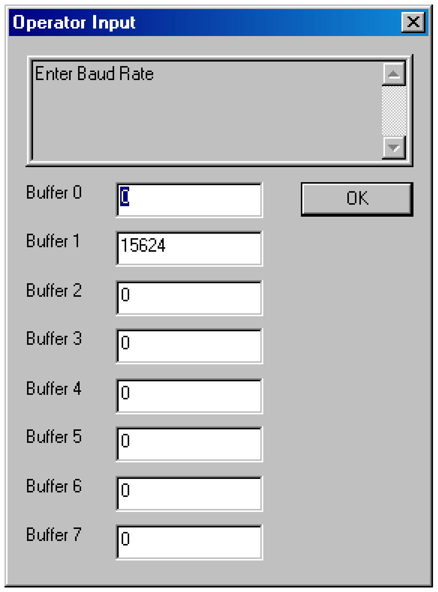
With the above parameters, the Sequence will keep looping to line 4 until 0 < Buffer#1 or until a timeout occurs at 5 seconds.
For Wait Until, the Sequence Event Time field is the time required to complete all the repeats.

Operator Input
During Sequence execution the operator may be required to input a specific piece of information or may simply need to enable continued execution. For this an Operator Input operation can be placed into the Sequence. When this operation is selected, the Operator Input Text Line dialog box allows the developer to enter text for a prompt.


Operator input Text line

Cancel OK

Enter text to be used for Operator Input: Enter Baud Rate


Operator Input

Enter Baud Rate

Buffer 0	0
Buffer 1	15624
Buffer 2	0
Buffer 3	0
Buffer 4	0
Buffer 5	0
Buffer 6	0
Buffer 7	0

OK

When this line is encountered, the Sequence will halt, bring up the Operator Input dialog box and wait until the Operator has entered the requested information.

Data is passed to the Sequence by way of the Sequence Buffers (see Sequence Buffers for details).

The Sequence can then continue execution using the information entered.

Event Reference
Event References are Events stored in an external Events Database file. Normal Events are stored the Sequence file, while Event References are stored in other Sequence files or in the Main Events Database. These are useful when you want more than one Sequence to use the same Event or when you want to use Event Parameter Substitution (see below for details) to modify an Event in another file.

Event References can be thought of as a "window" looking into an Event in another Events Database. You can view and edit these Events just like normal Events, but you must remember that you are actually editing an Event in another file.

Sub-Program
Sequences can download or run program files (QCP files). By default, a program is downloaded and run. This is same as pressing the Run button on the Program Info toolbar. To change the program to download only, double click on the Operation Info field.

NOTE: Sub-Programs must be in the same folder as the master sequence.

Sequence Timing

Sequences execute on a precise time base of 10 milliseconds. Every 10ms a new operation may take place whether it is an Event or another operation. Because data is typically transmitted serially, delays beyond 10ms may occur if the data baud rate is slow or if large amounts of data are being transmitted or received. Sequence timing is set up by establishing timing relationships between Sequence Operations. An operation's time is always relative to either the beginning of the Sequence (Time = 0) or to another operation.

Execution Time
This is the actual time that the operation is executed. This parameter cannot be directly edited by the user. Execution time is calculated from other parameters by the following formula:

For Relative To Start operations:
    Execution Time = Execution Time of the Relative To Line + Relative Time

For Relative To End operations:
    Execution Time = Execution Time of the Relative To Line
                                + Relative To Line's Event Time
                                + Relative Time

Relative To Line
Each Sequence line must be made relative to another line or to line 0. Line 0 is the invisible, starting line of the Sequence and cannot be edited. The time of line 0 is always "0".
This is the line number of the operation that this line should be relative to. If an absolute execution time is required enter line #0. The line number entered cannot exceed the number of lines in the Sequence.

When a Sequence is used as a Sub-Sequence, the start time is the time that the Sub-Sequence operation is executed.

Relative Time
Using the Relative To Line as a reference, this is how long this line will wait before executing.

Relative To Start/End
This specifies whether the line is reference to the start or end of the Relative To Line.

For Example, if an operation is dependent on the completion of a previous operation, it is Relative To End of the previous operation. If an operation is dependent to the time the previous operation starts to execute, it is Relative to Start of the previous operation.
The completion time (or Event Time) of a sequence operation depends on the operation. See Sequence Operations for details.

Sequence Files

QuickControl's Sequence files can be saved and copied just like any other Windows applications, such as a word processor. QuickControl Sequence Files end with the extension "CTL".

Sequence Description

Used for documenting the Sequence.

Sequence Total Time

The total execution time is used during Sequence processing to determine the "timed" end of the Sequence. This is necessary so that if a Sequence is repeated the time can be fixed at a desired length that may not be equal to the actual end of the Sequence line processing. The total time can never be shorter than the execution time of the last line of the Sequence.

Sequence Lines Data

This is all the Sequence timing information required to construct the time ordered list of Sequence Operations.

Sequence Monitor Settings

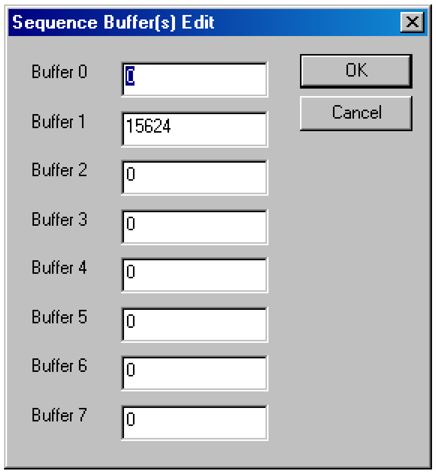
This includes values of all the Sequence Monitor check boxes and Sequence Repetitions.

Event Substitution Data

This includes which Event parameter to substitute and which Sequence Buffers to use (see Sequence Buffers for more details).

Saving and Loading a Sequence

A Sequence file can be loaded and saved like most other Windows applications. From the File Menu select Open, Save or Save As to perform the desired task. Many Sequence files can be opened at one time. Use the Window Menu to select the current file or arrange them for viewing and editing. By default, Sequence files are maximized when starting the application.



Buffer 0	0
Buffer 1	15624
Buffer 2	0
Buffer 3	0
Buffer 4	0
Buffer 5	0
Buffer 6	0
Buffer 7	0

Sequence Buffers

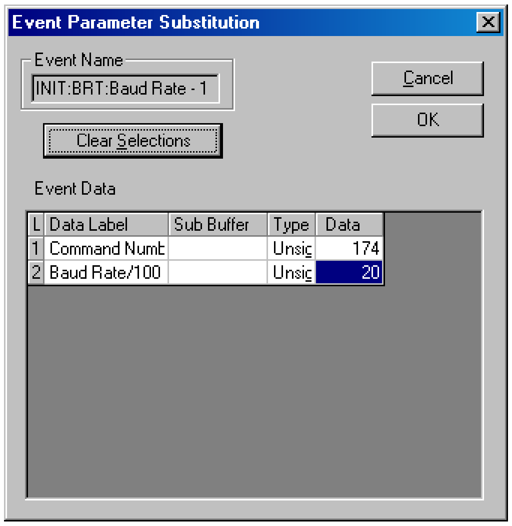
Sequence Buffers are global variables used by developers as temporary storage areas. There are 8 of them numbered 0-7. Each stores a Float data type. The Sequence Buffers keep their values for as long as QuickControl is running. They are great for storing information used in two different Events or Sequences or calculating data in an Event that needs to be used somewhere else.

The Sequence Buffers can be edited from the Sequence

Monitor (see below) or from the menu (Sequences->Edit Sequence Buffers).

Event Parameter Substitution

Event Parameter Substitution is an advanced feature that allows for dynamic substitution of Event parameters.

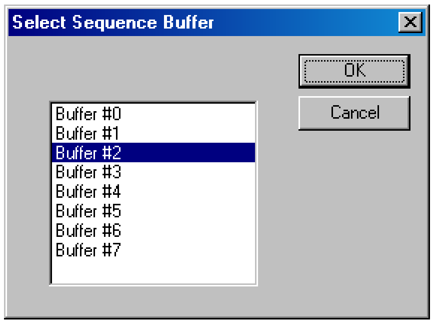


INIT:BRT:Baud Rate - 1

L	Data Label	Sub Buffer	Type	Data
1	Command Numb		Unsig	174
2	Baud Rate/100		Unsig	20

Before an Event is executed, parameters that control that Event's behavior can be dynamically changed. During Sequence editing, a Sequence line may be given parameters to substitute in an Event. For each substitution a Sequence Buffer is associated with the parameter to be changed. When the Sequence is executing, the data in the Sequence Buffer is placed into the parameter location prior to the Event being executed.

After the new Sequence line has been added, the Parameter Substitution can be set. Parameter substitution can only be done on a Sequence containing a command Event. Double-click on the SUB cell. An Event Parameter Substitution dialog box will be displayed showing the parameter lines of the Sequence line Event.



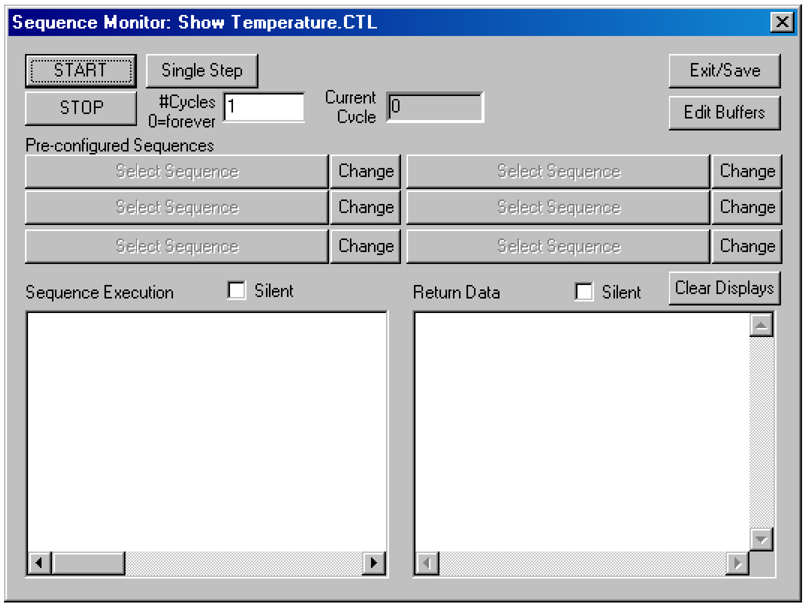
In the above example, if you wanted to have the baud rate set by the contents of Sequence Buffer 2, double click on line 2's Sub Buffer cell.

Select Buffer #2 and press OK.

The next time the Sequence is executed, this Event will get its data from Sequence Buffer #2.

Sequence Monitor

This dialog box is used to execute a Sequence.



#Cycles 0=forever: 1
Current Cycle: 0

It is designed to give the operator control over Sequence execution and provide real time status.
The dialog box is composed of the following elements:

START Sequence

STOP Sequence

Single Step

Each time the Single Step button is pressed the next line in the

Sequence will be executed.

# Cycles
Setting this inputs, the number of times that the entire Sequence will be repeated. Enter a "0" to cycle forever.

Current Cycle
This is a real-time display of the number of times the entire Sequence has been executed.

Exit/Save
Exit and save data.

Edit Buffers
Edit Sequence Buffers.

Buffer data can be viewed at any time during Sequence execution. However, the data is not updated real-time. The data displayed is the data in the buffer at the time the Edit Buffer button was pressed.

Pre-Configured Sequences
These buttons can be configured to execute frequently used sequences. To configure a button, press Change, select the sequence and then press OK. To clear a button, press Change and then Cancel.

Silent check boxes.
Check these boxes to halt the updating of the list boxes.

Clear Displays
This button clears the two list boxes of all text.

Sequence Execution list box
During the execution of the Sequence the Operation Name is displayed when the operation is executed. This allows the user to view the Sequence as it is being executed.

This display is not real-time. There will be a small delay between when the operation is performed and when the operation is displayed.

Return Data list box
This is where Data Templates dump their output.

This display is not real-time. There will be a small delay between when the Event is executed and when the Return Data is displayed.

Editing Sequence Line Shortcuts

QuickControl provides several shortcuts for editing most data on a Sequence line via double clicking in the cell you want to edit. This is different than adding a line in that selected parts of a line can be changed without editing the entire line. The following is a list of columns and the shortcuts that are provided.

Double-click in the following columns to activate the indicated shortcuts.

Operation Name

Sequence Operation	Result of double-clicking
Event	Edit Event Dialog Box.
Event Reference	Edit Event Dialog Box.

Operation Info

Sequence Operation	Result of double-clicking
Event	Select Event Dialog Box
Sub Sequence	Allow Selection of Different Sub Sequence
Repeat To	Edit Repeat To Information
Wait Until	Edit Wait Until Information

Data Template

Sequence Operation	Result of double-clicking
Event	Edit Data Template Dialog Box
Event Reference	Edit Data Template Dialog Box

SUB

Sequence Operation	Result of double-clicking
Event	Event Parameter Substitution Dialog Box
Event Reference	Event Parameter Substitution Dialog Box

Exec Time

No shortcuts.

Rel To ..

Sequence Operation	Result of double-clicking
All	Edit Relative To Information (see Sequence Timing)

Event Time

Sequence Operation	Result of double-clicking
Event	Edit Event Dialog Box.
Event Reference	Edit Event Dialog Box.

The following sequence line shortcuts are also available:

Add Line (Ctrl-A)
Edit Line (Ctrl-E)
Delete Line (Ctrl-D)
Insert Line (Ctrl-I)

Creating a Sequence

This example will demonstrate how to:

- Download & Run. qcp Programs
- Write to Registers
- Read from Registers
- Use the Buffer Registers

Before creating the sequence file, all intended. qcp programs must be created first. Power all devices and poll with QuickControl. Once QuickControl detects all devices, write and save the. qcp programs accordingly to Motor ID. Remember that the active motor is the device number shown to the left screen of QuickControl next to Desc.

Do not save the. qcp program without verifying if it is for the intended motor ID. This will prevent errors when running the sequence file. Save all the. qcp programs in the same location.

Now, create the sequence file, File->New Sequence File. Save and name it right away in the same location as all the. qcp programs.

Line 1 is the default, blank command and may be deleted.

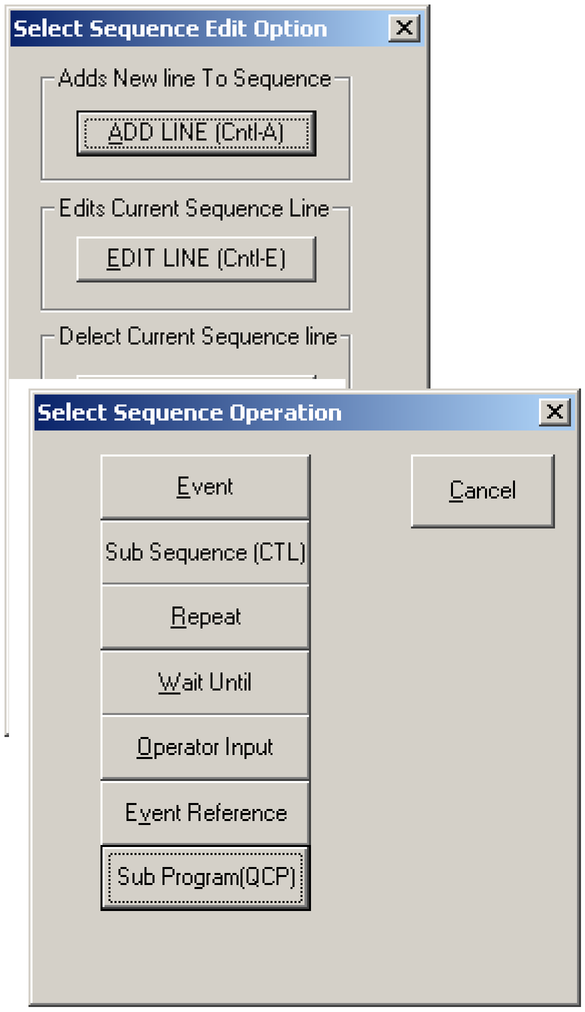
Adding the Sub Program Command Line

Click 'Edit Seq'

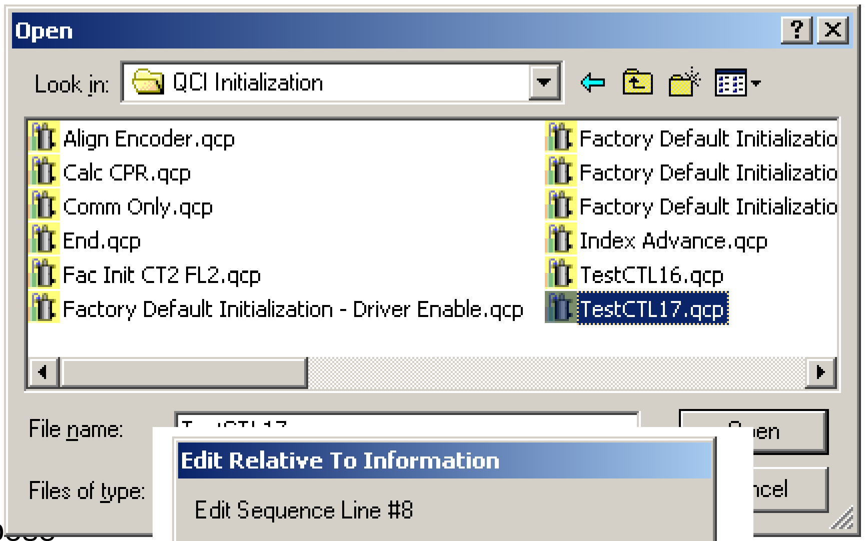
-On the Sequence Edit Option click ADD LINE (Ctrl-A)

These two windows will be your primary keys to accessing features and making modifications. The next section of examples may or may not show these two pictures.

-On the Sequence Operation click Sub Program (QCP)


Select Sequence Edit Option
Adds New line To Sequence
ADD LINE (Cntl-A)
Edits Current Sequence Line
EDIT LINE (Cntl-E)
Delect Current Sequence line
Select Sequence Operation
Event
Cancel
Sub Sequence (CTL)
Repeat
Wait Until
Operator Input
Event Reference
Sub Program(QCP)

-Choose the. qcp program


Open

Look in: QCI Initialization

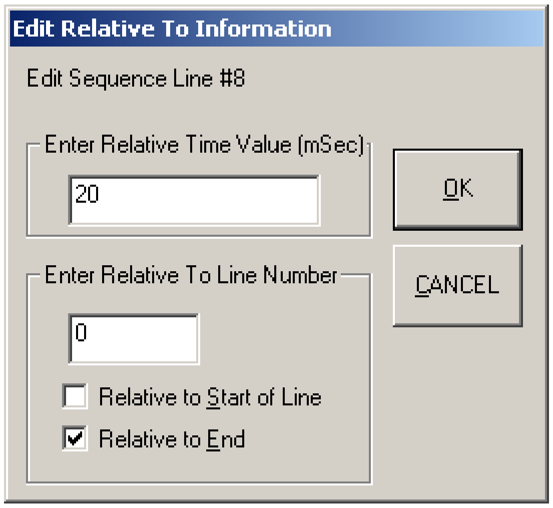
Align Encoder.qcp
Calc CPR.qcp
Comm Only.qcp
End.qcp
Fac Init CT2 FL2.qcp
Factory Default Initialization - Driver Enable.qcp
Factory Default Initializatio
Factory Default Initializatio
Factory Default Initializatio
Index Advance.qcp
TestCTL16.qcp
TestCTL17.qcp

File name:
Files of type:

-Type in the Time and Number. Choose Relative to End.

The Relative Time is the delay time to wait before sending the command.

The Line Number should correspond to the Previous line.


Edit Relative To Information

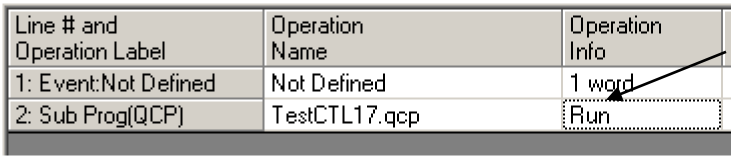
Edit Sequence Line #8

Enter Relative Time Value (mSec): 20

Enter Relative To Line Number: 0

[ ] Relative to Start of Line
[✓] Relative to End

-Double click in line 2, Operation Info column. The Operation must be 'Run' as shown.



Line # and Operation Label	Operation Name	Operation Info
1: Event:Not Defined	Not Defined	1 word
2: Sub Prog(QCP)	TestCTL17.qcp	Run

This completes adding the first sequence command line. You may repeat this section to add more Sub Program command lines as need.

The Edit Relative To Information Window follows every command. Some commands may require more time than others. Increase if necessary.

Check if you have all the Sub Programs corresponding to the correct motor ID.

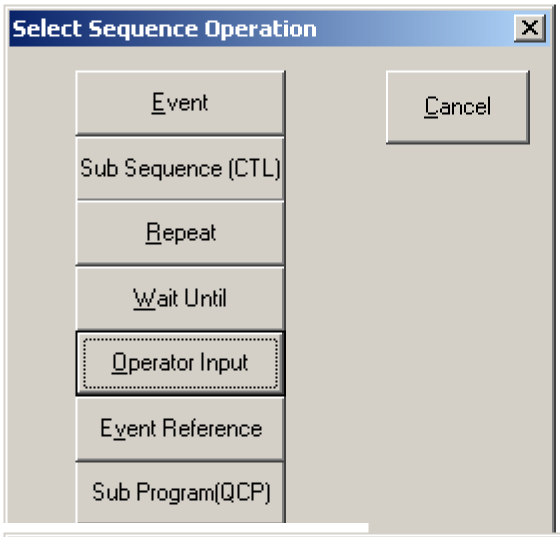
If you double click on the 'Operation Name' column of the Sub Program line, the program will appear with the Desc corresponding to the motor ID (All motor must be powered and QuickControl polling). To return to the sequence program environment, click the close icon.

Adding the Operator Input Command Line

Click 'Edit Seq'

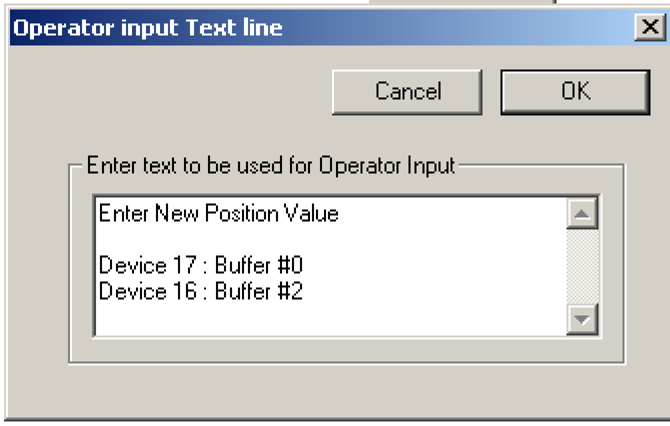
Click Add Line (Ctrl-A)

-Click Operator Input



Operator input pauses the sequence until the operator presses 'OK'. Pressing 'Cancel' will stop the entire sequence.

-Enter the text information desired. Generic information is shown.



Enter New Position Value

Device 17 : Buffer #0
Device 16 : Buffer #2

-Enter time and number, Relative to End. The number line should reference the previous line. Enter if not.

The operator input text area should be informative and productive for the user.

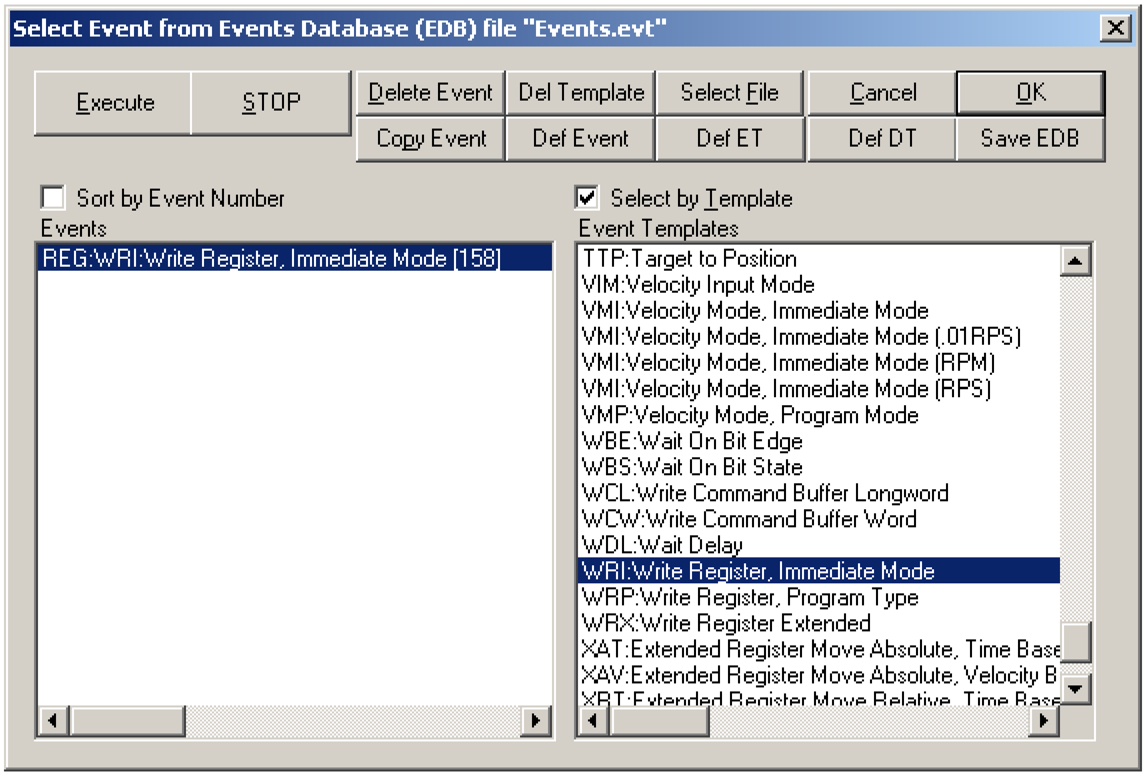
Adding the Write Register Immediate (WRI) Command Line

Click ‘Edit Seq’

Click Add Line (Ctrl-A)

Click Event

-The Events Database appears. Find and select WRI on the right. Then on the left will appear the event, double click the REG:WRI…as shown.



☐ Sort by Event Number

Events

REG:WRI:Write Register, Immediate Mode [158]

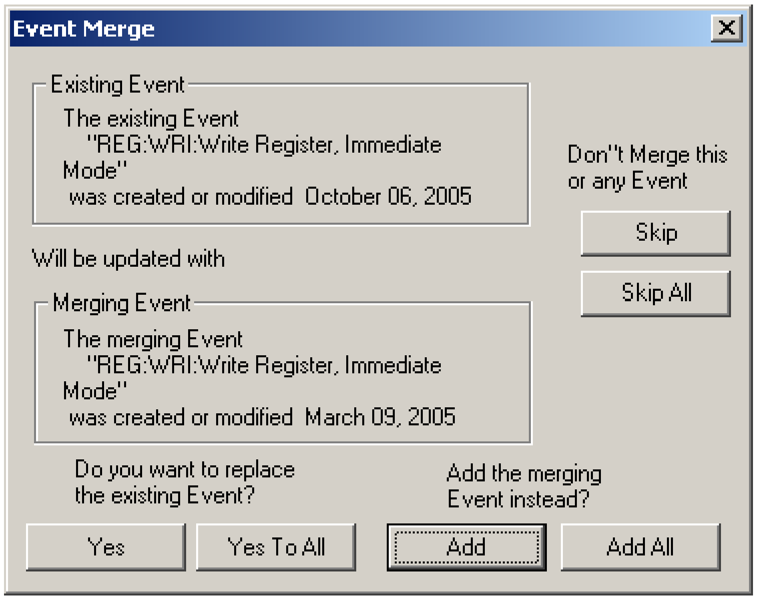
☑ Select by Template

Event Templates

TTP:Target to Position
VIM:Velocity Input Mode
VMI:Velocity Mode, Immediate Mode
VMI:Velocity Mode, Immediate Mode (.01RPS)
VMI:Velocity Mode, Immediate Mode (RPM)
VMI:Velocity Mode, Immediate Mode (RPS)
VMP:Velocity Mode, Program Mode
WBE:Wait On Bit Edge
WBS:Wait On Bit State
WCL:Write Command Buffer Longword
WCW:Write Command Buffer Word
WDL:Wait Delay
WRI:Write Register, Immediate Mode
WRP:Write Register, Program Type
WRX:Write Register Extended
XAT:Extended Register Move Absolute, Time Base
XAV:Extended Register Move Absolute, Velocity B
XRT:Extended Register Move Relative, Time Base

-Enter time and number, Relative to End. The number line should reference the previous line. Enter if not.

When you add two of the same commands, the Event Merge Window will appear. You want to click 'Add' under the phrase "Add the merging Event instead?" This ensures that you do not tie both commands together to reference one Motor ID.


Event Merge

Existing Event
The existing Event
"REG:WRI:Write Register, Immediate Mode"
was created or modified October 06, 2005

Will be updated with

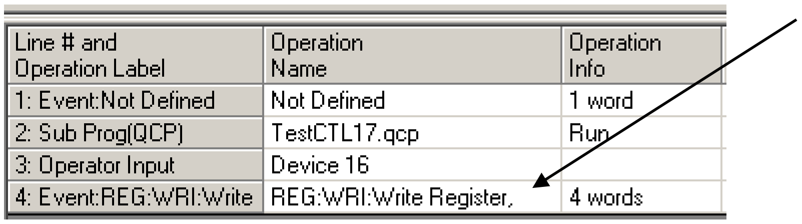
Merging Event
The merging Event
"REG:WRI:Write Register, Immediate Mode"
was created or modified March 09, 2005

Don''t Merge this or any Event

Do you want to replace the existing Event?

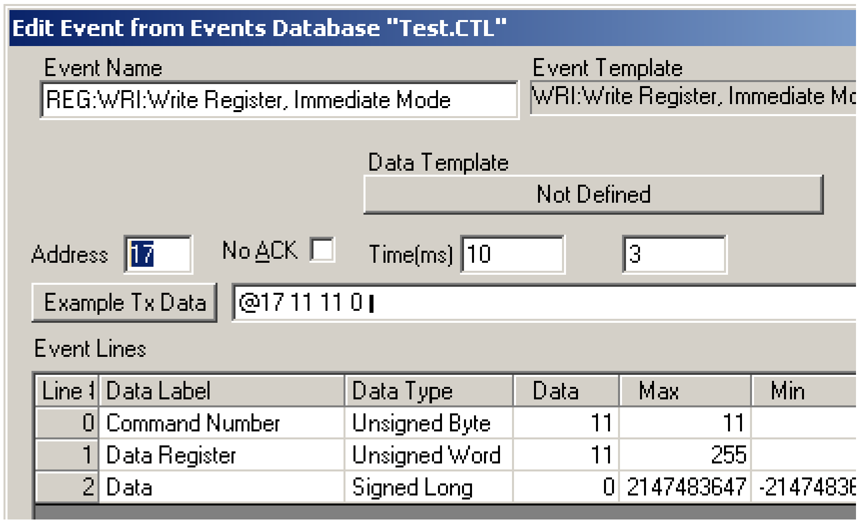
Add the merging Event instead?

Once you have the command showing in the sequence programming window, double click in the column 'Operation Name' of the 'Line #' of the WRI command.



Line # and Operation Label	Operation Name	Operation Info
1: Event:Not Defined	Not Defined	1 word
2: Sub Prog(QCP)	TestCTL17.qcp	Run
3: Operator Input	Device 16	
4: Event:REG:WRI:Write	REG:WRI:Write Register,	4 words

The window below will display. The address reference the which motor to write to; 17 is for motor ID 17.

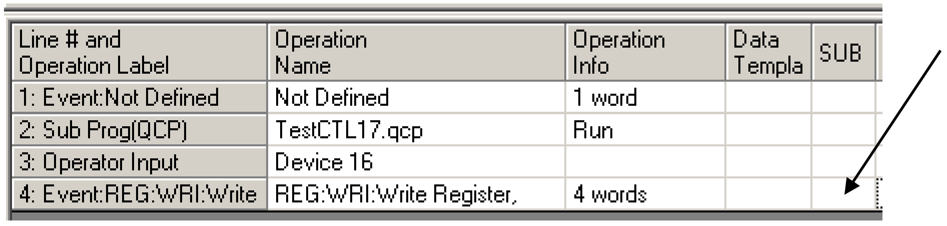

Edit Event from Events Database "Test.CTL"

Event Name: REG:WRI:Write Register, Immediate Mode
Event Template: WRI:Write Register, Immediate Mc
Data Template: Not Defined
Address: 17
No ACK
Time(ms): 10
3
Example Tx Data: @17 11 11 0

Event Lines

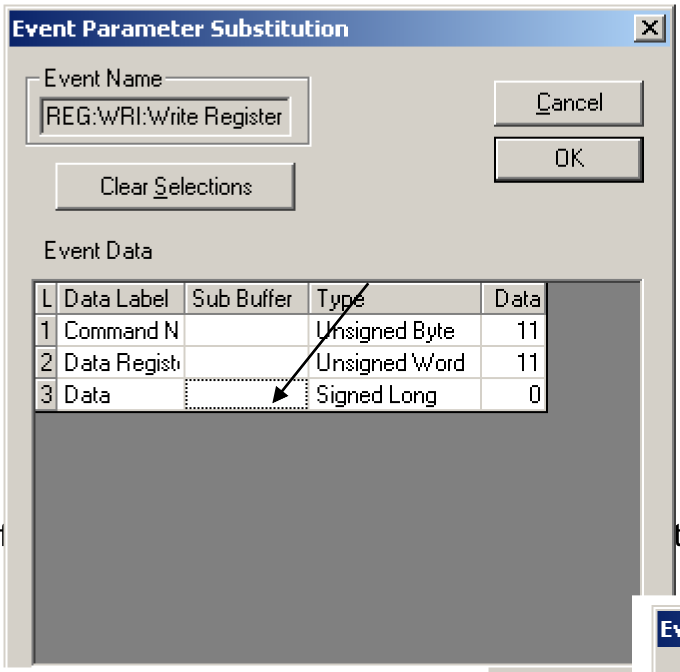
Line #	Data Label	Data Type	Data	Max	Min
0	Command Number	Unsigned Byte	11	11	
1	Data Register	Unsigned Word	11	255	
2	Data	Signed Long	0	2147483647	-2147483

-Click on the ‘SUB’ column of the WRI command line.



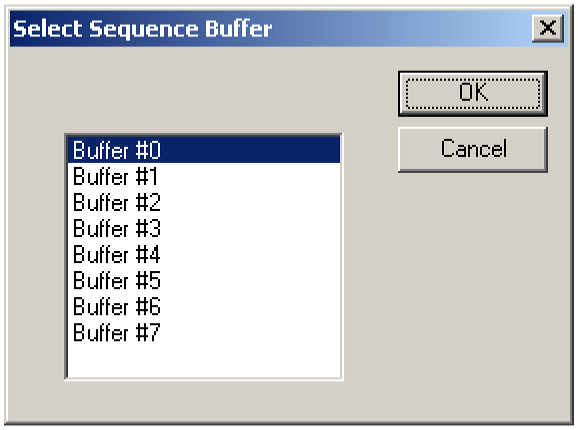
Line # and Operation Label	Operation Name	Operation Info	Data Templa	SUB
1: Event:Not Defined	Not Defined	1 word		
2: Sub Prog(QCP)	TestCTL17.qcp	Run		
3: Operator Input	Device 16			
4: Event:REG:WRI:Write	REG:WRI:Write Register,	4 words		

-The Event Parameter Substitution Window appears. Click in the Sub Buffer column to pull up the Select Sequence Buffer Window. Select desire buffer value.



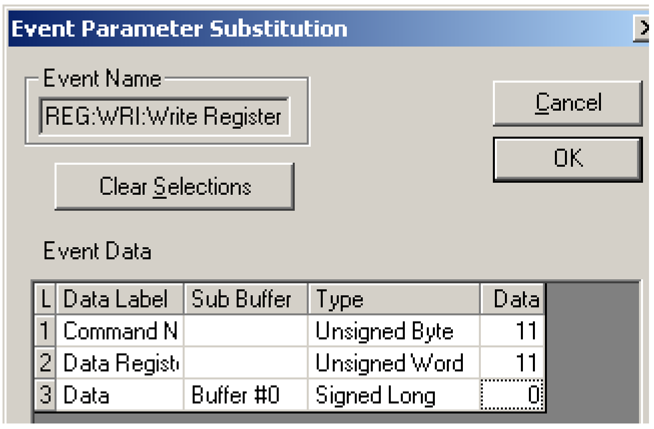
Event Name: REG:WRI:Write Register

L	Data Label	Sub Buffer	Type	Data
1	Command N		Unsigned Byte	11
2	Data Regist		Unsigned Word	11
3	Data		Signed Long	0



Buffer #0
Buffer #1
Buffer #2
Buffer #3
Buffer #4
Buffer #5
Buffer #6
Buffer #7

-Bu the Event Parameter Sub…Window.



Event Name: REG:WRI:Write Register

L	Data Label	Sub Buffer	Type	Data
1	Command N		Unsigned Byte	11
2	Data Regist		Unsigned Word	11
3	Data	Buffer #0	Signed Long	0

The Event Parameter Substitution allows the data entered into Buffer #0 to be used as the data for the WRI command. WRI will send the data in Buffer #0 to the specified register in the motor.

You will need to add as many WRI commands for as many motors you have on the network. You must verify that each WRI added correspond to different motor ID. Repeat section as necessary.

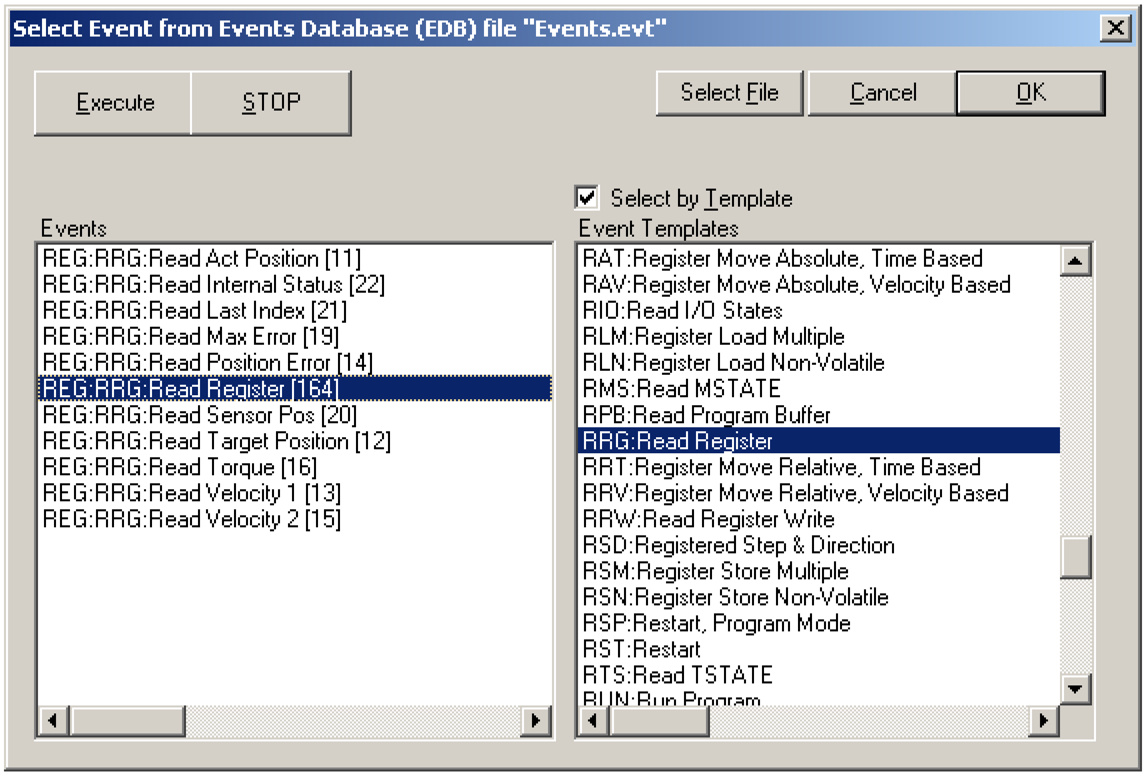
Adding the Read Register (RRG) Command Line

Click 'Edit Seq'

Click Add Line (Ctrl-A)

Click Event

-The Events Database appears. Find and select RRG on the right. Then on the left will appear the event, double click the REG: RRG Read Register…as shown. Notice that there are predefine registers. Read Register reference any registers.



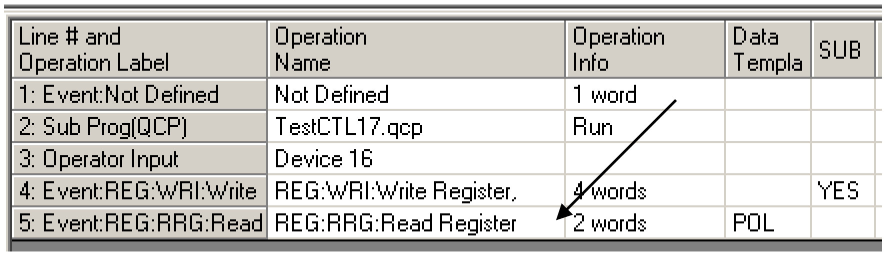
☑ Select by Template

Events
REG:RRG:Read Act Position [11]
REG:RRG:Read Internal Status [22]
REG:RRG:Read Last Index [21]
REG:RRG:Read Max Error [19]
REG:RRG:Read Position Error [14]
REG:RRG:Read Register [164]
REG:RRG:Read Sensor Pos [20]
REG:RRG:Read Target Position [12]
REG:RRG:Read Torque [16]
REG:RRG:Read Velocity 1 [13]
REG:RRG:Read Velocity 2 [15]

Event Templates
RAT:Register Move Absolute, Time Based
RAV:Register Move Absolute, Velocity Based
RIO:Read I/O States
RLM:Register Load Multiple
RLN:Register Load Non-Volatile
RMS:Read MSTATE
RPB:Read Program Buffer
RRG:Read Register
RRT:Register Move Relative, Time Based
RRV:Register Move Relative, Velocity Based
RRW:Read Register Write
RSD:Registered Step & Direction
RSM:Register Store Multiple
RSN:Register Store Non-Volatile
RSP:Restart, Program Mode
RST:Restart
RTS:Read TSTATE
RUN:Run Program

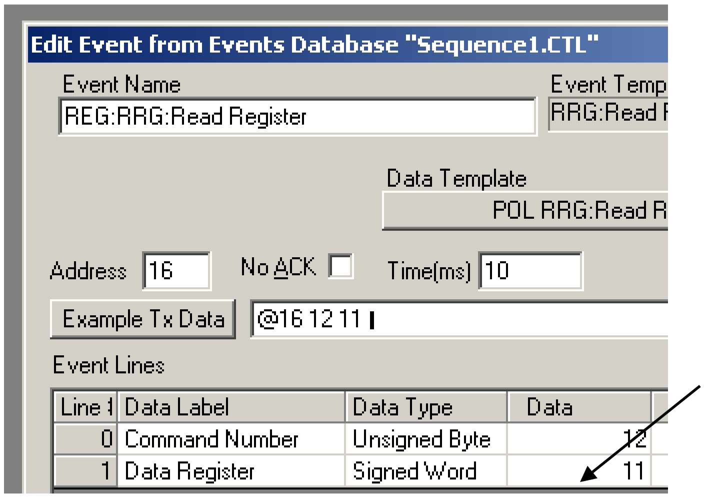
-Enter time and number, Relative to End. The number line should reference the previous line. Enter if not.

-Click the 'Operation Name' of the RRG command line.



Line # and Operation Label	Operation Name	Operation Info	Data Templa	SUB
1: Event:Not Defined	Not Defined	1 word		
2: Sub Prog(QCP)	TestCTL17.qcp	Run		
3: Operator Input	Device 16			
4: Event:REG:WRI:Write	REG:WRI:Write Register,	4 words		YES
5: Event:REG:RRG:Read	REG:RRG:Read Register	2 words	POL	

-Then change the address to reflect the motor ID. Click on the 'Data' column of the Data Register and change the value to the register value you like to read.



Edit Event from Events Database "Sequence1.CTL"

Event Name: REG:RRG:Read Register

Event Temp: RRG:Read R

Data Template: POL RRG:Read R

Address: 16 No ACK ☐ Time(ms): 10

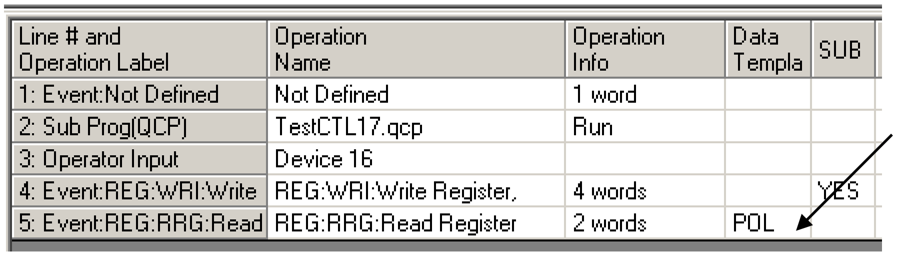
Example Tx Data: @16 12 11

Event Lines

Line #	Data Label	Data Type	Data
0	Command Number	Unsigned Byte	12
1	Data Register	Signed Word	11

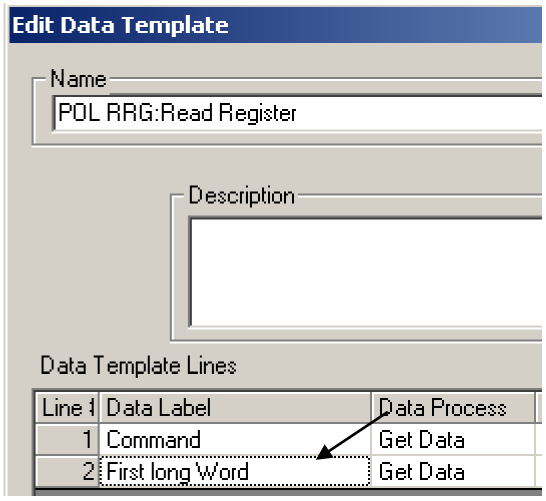
-Click OK to close window.

-Click on 'Data Template' column of RRG command line.



Line # and Operation Label	Operation Name	Operation Info	Data Templa	SUB
1: Event:Not Defined	Not Defined	1 word		
2: Sub Prog(QCP)	TestCTL17.qcp	Run		
3: Operator Input	Device 16			
4: Event:REG:WRI:Write	REG:WRI:Write Register,	4 words		YES
5: Event:REG:RRG:Read	REG:RRG:Read Register	2 words	POL	

-The Edit Data Template opens. Double click on 'First long Word' will open the Select Edit Operation window. Click ADD LINE in Edit Operation.



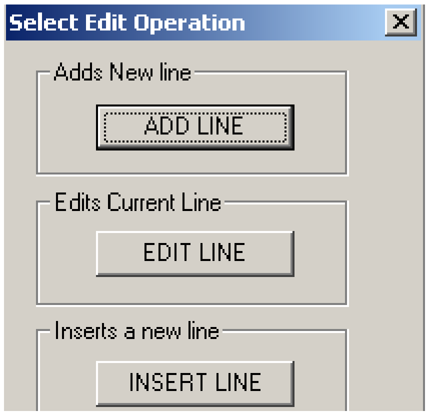
Edit Data Template

Name: POL RRG:Read Register

Description:

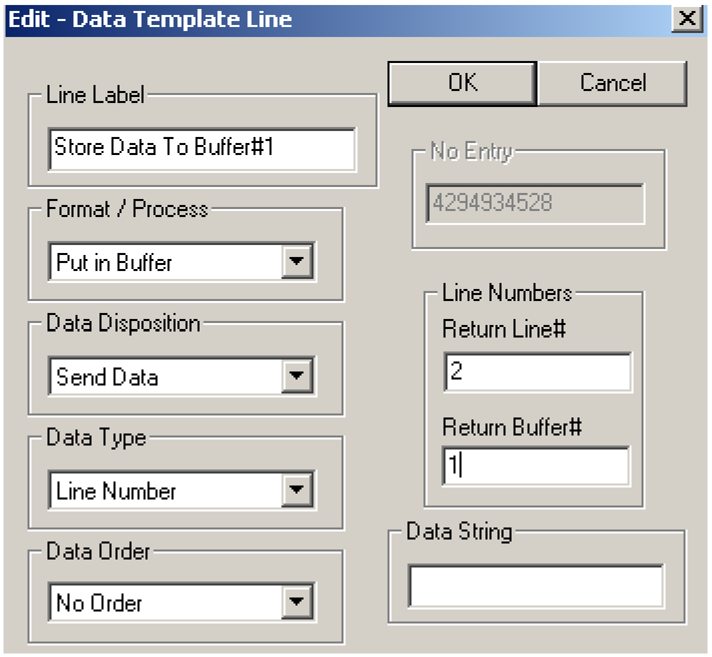
Data Template Lines

Line #	Data Label	Data Process
1	Command	Get Data
2	First long Word	Get Data



Select Edit Operation

-The Edit – Data Template Line window appears. Enter as shown. This tells QuickControl to put the read data into buffer #1.



Line Label: Store Data To Buffer#1
Format / Process: Put in Buffer
Data Disposition: Send Data
Data Type: Line Number
Data Order: No Order
No Entry: 4294934528
Line Numbers
Return Line#: 2
Return Buffer#: 1
Data String:

The line label is entered by you for reference (description).

Put in Buffer puts the data into the buffer.

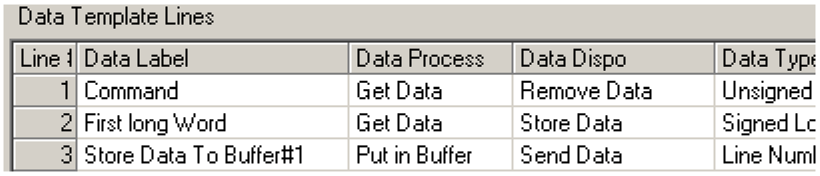
Send Data only sends data.

Line Number is reference to Line Numbers frame at right.

Return Line # refers to the Edit Data Template window, line 2. Line 2 contains the read data; therefore, the Return Line # must reference line 2 to access (obtain) the data.

Return Buffer # refers to the buffer number you want to store the data to.

This is how the commands look.



Data Template Lines

Line #	Data Label	Data Process	Data Dispo	Data Typ
1	Command	Get Data	Remove Data	Unsigned
2	First long Word	Get Data	Store Data	Signed Lc
3	Store Data To Buffer#1	Put in Buffer	Send Data	Line Numl

When the command RRG reads a register, the motor sends back a string of characters (data packet). The data packet must be truncated to obtain only the data.

Line 1 removes the outside frames.
Line 2 obtains and stores the data (handled by QuickControl).
Line 3 copies the data into Buffer# 1.

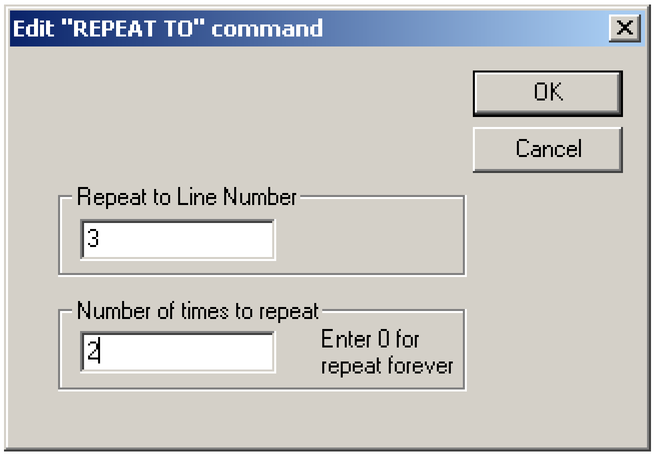
Adding the Repeat To Command Line

Click ‘Edit Seq’

Click Add Line (Ctrl-A)

Click ‘Repeat’

-Type 3 as we want to repeat to the Operator Input and 2 for two loops.

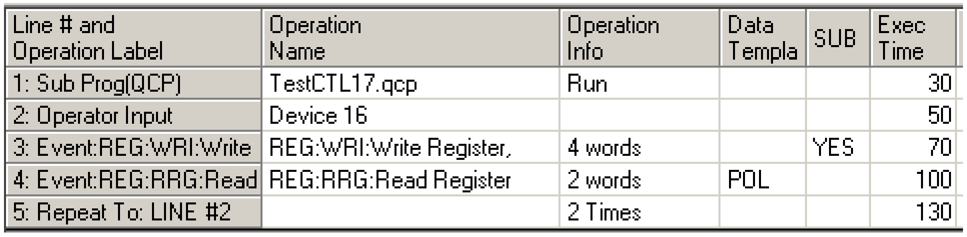


Repeat to Line Number: 3

Number of times to repeat: 2

-Enter time and number, Relative to End. The number line should reference the previous line. Enter if not.

Sequence Complete



Line # and Operation Label	Operation Name	Operation Info	Data Templa	SUB	Exec Time
1: Sub Prog(QCP)	TestCTL17.qcp	Run			30
2: Operator Input	Device 16				50
3: Event:REG:WRI:Write	REG:WRI:Write Register,	4 words		YES	70
4: Event:REG:RRG:Read	REG:RRG:Read Register	2 words	POL		100
5: Repeat To: LINE #2		2 Times			130

Finished Sequence File

Save the sequence and execute.

Note: Real time data acquisition can be done at some level. For example, viewing the actual position of all the devices (motors) can be accomplished if the goal of the sequence is to only read registers. However, since changing data is desired, a write command will need to pause the entire sequence as data are being changed or delayed as each write commands need to be opened and changed. This does not simulate an instantaneous effect, as the delay is too long in between changes.