Tel +1 (717) 767-6511
Fax +1 (717) 764-0839
www.redlion.net

QUICKSILVER MASTER Driver

Information Sheet for Crimson v2.0

Compatible Devices

- Quicksilver Silvermax
- Quicksilver SilverLode Servos

Verified Devices

- Silvermax Model 23-4
- Silverdust D2-G1-03-IGB

Driver Option

When the Quicksilver is selected, the programmer has an option of setting the lowest address that will be considered a "Group". Read operations attempted on a device with an address equal to, or higher than, the Group address will be ignored, and a value of 0 will be returned.

Device Option

The programmer sets the target address of the Quicksilver to be accessed.

NOTE: The programmer is responsible for assigning only those commands supported by the device being configured.

NOTE: New Quicksilver Commands – REG and ERR

Previous versions of the driver had used mnemonic **REG** for reading and writing registers via the Quicksilver commands **RRG** and **WRI**. Additionally, **ERR** was defined as the driver's mnemonic for the value of a NAK error return. Therefore, new Quicksilver commands, **REG** for Registered Electronic Gearing, and **ERR** for Error Limits, Remote, have been assigned driver mnemonics **RGG** and **ELR**, respectively, in order to maintain compatibility with previous databases.

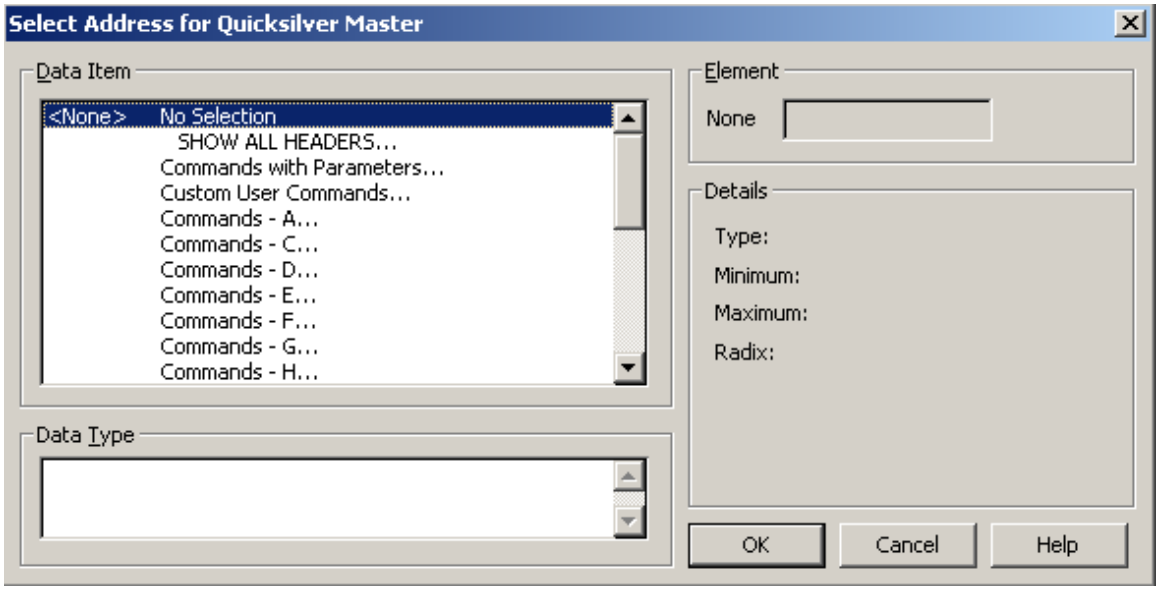
Custom User Strings

The driver now has selections that permit the programmer and operator to enter command strings and process the response. There are 4 pairs that will be sent and read when encountered (must not be used for command/write operations), and 8 pairs that require a number from 1 to 8 to be entered in the "_Send" selection. Please read Appendix B at the end of this document before using.

COMMAND SELECTION: The initial command layout comprises a list of headers. Selecting "SHOW ALL HEADERS" will also display the entire list of headers. Selecting any other header displays a list of commands in that category. The first header is "Commands with Parameters…", which displays "Configure I/O", "Register", and "Program Buffer" access commands. The programmer selects the appropriate Element value for the desired operation, i.e. the I/O point, Register number, or Program Buffer position. This list also includes the internal value ERR, used to display the command and error numbers if the device cannot perform the requested operation.

The second header is "Custom User Commands…" which permits selections for defining a command string. Please read Appendix B before using.

The other headers, beginning with "Commands – A", through "Commands – Z", display all commands beginning with that letter.



Reading Data:
READ commands comprise:
POL – Read Polling Status Word
RPB – Read 1 Word from Program Buffer
REG – Read a Register
RIO – Read I/O Status
RIS – Read Internal Status Word

RPB and **REG** require an address. They are represented as:
RPBnnn and **REG**nnn, where nnn is the desired address.
REG is either Read or Write. Others are Read-Only.

There are two special READ ONLY operations:
PLG is READ Programmed Lowest Group Number. This indicates the lowest Unit Address that will be considered part of a group. This value is selected in the configuration of the unit, and is made available as a reference for the IDT command.
PUN is READ Programmed Unit Number. This indicates the Unit Address for the device it is assigned to. This value is selected in the configuration of the unit, and is made available as a reference for the IDT command.
Both of these values are internal values only.

Custom User Commands
Either the programmer or operator may enter strings that perform either read or write operations. Please read and understand Appendix B before implementing these functions.

ERR is an internal Read/Write value. When read, the high word is the command number that generated a NAK. The low word is the code describing the error. Any write clears the value.

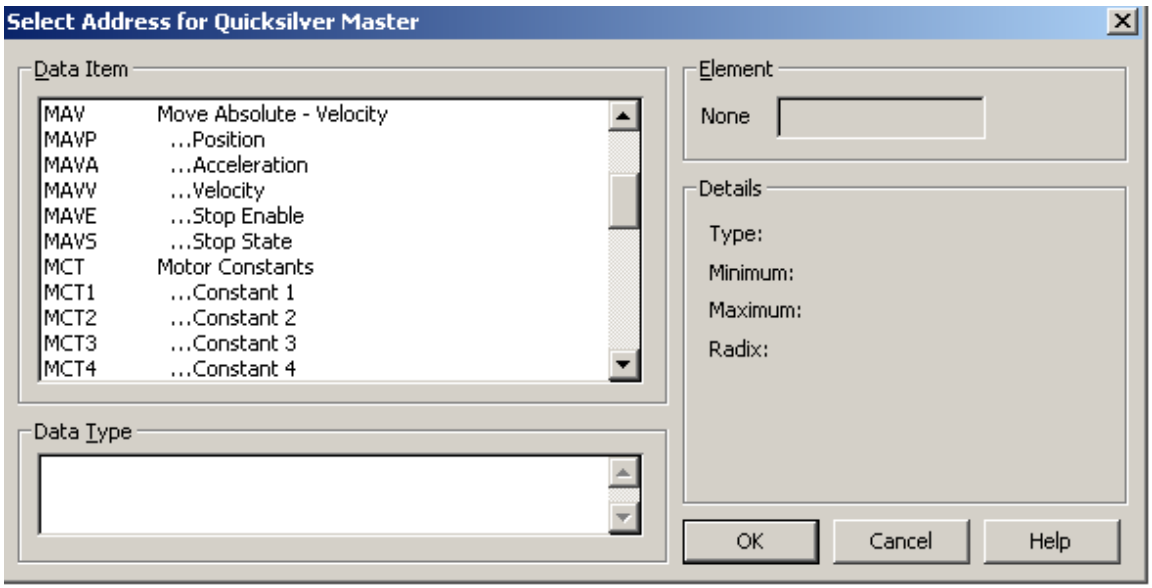
REGnnn**,** which is R/W, is the only other instruction that can be read.

Sending Commands and Writing Data:
The programmer must be familiar with the operation of each instruction used. Some three-letter instructions are commands, requiring only to perform a write instruction. **TTP** (Set Target to Position) is an example. The programmer would set up an action, VariableTTP = 1, in order to execute the command.

Some three-letter instructions require a single data value to be sent. **OVT** (Over Voltage Trip) is an example. The programmer would set up an action VariableOVT = 12, to set the Trip point to 12. The instruction **WRP** (Write Register – Program Type) requires a numeric address set to the desired register. WRP030 will write a value to Program Register 30.
Note: Some of these commands require that only certain values be sent. It is the responsibility of the programmer to limit data values to those supported by the model being used.

The rest of the instructions are four letter commands representing one of multiple data to be sent upon a write to the corresponding three-letter command. For example:
MAV - Move Absolute, Velocity Based is the command that sends:
MAVP - Position
MAVA - Acceleration
MAVV – Velocity
MAVE - Stop Enable
MAVS - Stop State

The programmer must fill the 5 values with data applicable to the operation, then execute the **MAV** instruction using an action similar to VariableMAV = 1. Any value is permissible, including 0. Every instruction of four or more letters, (except **CIIE)**, is marked as being a "Cached Value" in the register list, and the first three letters identify the command that sends those values.



Read commands performed on cached values (…Position, for example) return the most recent value written to them.

Commands that are write-only, and for which 0 is a valid value, return the value 409600000 when read. This value allows the display of 00000 in decimal mode, or of 0000 in hexadecimal mode, while permitting the write of the value 0.

Read operations performed on any other write-only instructions will return 0.

Cable Information

SILVERLODE Servos

G3 RS-232 Port	G3 RS-485 Port	Quicksilver (DB9-F)
5 (TxD)	1 or 7 (B)	3 (232-Rx / 485-B)
2 (RxD)	2 or 8 (A)	2 (232-Tx / 485-A)
4 (0V)	6 (0V)	5 (Gnd)

SILVERMAX

G3 RS-232 Port	Quicksilver (DB9-F)
5 (TxD)	3 (Rx)
2 (RxD)	2 (Tx)
4 (0V)	5 (Gnd)
G3 RS-232 Port	**Quicksilver (15 way Female)**
5 (TxD)	12 (Rx)
2 (RxD)	2 (Tx)
4 (0V)	8 (Gnd)

APPENDIX A – Command List

PREFIX	Description
	Commands with Parameters…
CIO	Configure I/O Setting
CII	Configure I/O Setting, Immediate
CIE	Configure Extended I/O
CIIE	Configure Extended I/O, Immediate
RPB	READ 1 Word from Program Buffer
REG	Register, Immediate (RRG/WRI)
WRP	Write Register - Program Type
ERR	NAK Response
	Custom User Commands…
_CW1	Command String 1 (Cmd Number data data…)
_RW1	String 1 Response
_CW2	Command String 2 (Cmd Number data data…)
_RW2	String 2 Response
_CW3	Command String 3 (Cmd Number data data…)
_RW3	String 3 Response
_CW4	Command String 4 (Cmd Number data data…)
_RW4	String 4 Response
_CW5	Command String 5 (Cmd Number data data…)
_RW5	String 5 Response
_CW6	Command String 6 (Cmd Number data data…)
_RW6	String 6 Response
_CW7	Command String 7 (Cmd Number data data…)
_RW7	String 7 Response
_CW8	Command String 8 (Cmd Number data data…)
_RW8	String 8 Response
_SEND	Send Command String N
_CRA	Continuous Read-String A (Cmd# data data…)
_RRA	Read String A Response
_CRB	Continuous Read-String B (Cmd# data data…)
_RRB	Read String B Response
_CRC	Continuous Read-String C (Cmd# data data…)
_RRC	Read String C Response
_CRD	Continuous Read-String D (Cmd# data data…)
_RRD	Read String D Response
	Commands – A…
ACR	Set Analog Continuous Read
ACRC	...Analog Channel Number
ACRD	...Data Register
ADL	ACK Delay
ADX	ACK Delay Extended
AHC	Set Anti-Hunt Constants
AHCOC	...Out: Open->Closed
AHCCO	...Into: Closed->Open
AHD	Anti-Hunt Delay
AHM	Anti-Hunt Mode
ARI	Analog Read Input
ARIC	...Analog Channel Number
ARID	...Data Register
ATR	Add to Register

ATRA	...Data Register
ATRD	...Data To Add
	Commands – C…
CER	Command Error Recovery
CIS	Clear Internal Status
CKS	Check Internal Status
CKSE	...Condition Enable
CKSS	...Condition State
CLCA	Calculation (Not SilverMax)
CLC1	...Operation
CLC2	...Data Register
CLC	Calculation (SilverMax Only)
CLCO	...Operation
CLCD	...Data Register
CLD	Calculation, Extended with Data
CLDR	...Register
CLDD	...Data
CLDO	...Operation
CLDA	...Result Register
CLM	Control Loop Mode
CLX	Calculation, Extended
CLX1	...Register 1
CLX2	...Register 2
CLCO	...Operation
CLXA	...Result Register
CME	Clear Max Error
COB	Clear Output Bit
CPL	Clear Poll
CTC	Control Constants
CTC1	...Velocity 1 Feedback Gain
CTC2	...Velocity 2 Feedback Gain
CTCV	...Velocity Feedforward Gain
CTCB	...Acceleration Feedback Gain
CTCF	...Acceleration Feedforward Gain
CTCP	...Proportional Gain
CTCI	...Integrator Gain
C2T	Control Constants 2
C2T1	...Velocity 1 Feedback Gain
C2T2	...Velocity 2 Feedback Gain
C2TV	...Velocity Feedforward Gain
C2TB	...Acceleration 1 Feedback Gain
C2TC	...Acceleration 2 Feedback Gain
C2TF	...Acceleration Feedforward Gain
C2TP	...Proportional Gain
C2TI	...Integrator Gain
	Commands – D…
DEM	Disable Encoder Monitor
DDB	Disable Done Bit
DIF	Digital Input Filter
DIFL	...I/O Line Number
DIFF	...Filter Constant
DIR	Set Direction
DLC	Dual Loop Control

DLY	Delay
DMD	Disable Motor Driver
DMT	Disable Multi-Tasking
	Commands – E…
EDH	Enable Done High
EDL	Enable Done Low
EEM	Enable Encoder Monitor
EMD	Enable Motor Driver
EMN	Encoder Monitor
EMNM	...Mode
EMNI	...Index State
EMNR	...Reserved
EMT	Enable Multi-Tasking
END	End Program
ERL	Error Limits
ERLM	...Moving limit
ERLH	...Hold Limit
ERLD	...Delay to Holding
ELR	Error Limits, Remote (ERR)
ELRP	...Remote Position Register
ELRE	...Maximum Starting Error
ETN	End of Travel, Negative
ETN1	...Enable ISW
ETN2	...State ISW
ETN3	...Enable IS2
ETN4	...State IS2
ETN5	...Enable XIO
ETN6	...State XIO
ETP	End of Travel, Positive
ETP1	...Enable ISW
ETP2	...State ISW
ETP3	...Enable IS2
ETP4	...State IS2
ETP5	...Enable XIO
ETP6	...State XIO
	Commands – F…
FLC	Filter Constants
FLC1	...Velocity 1 Feedback
FLC2	...Velocity 2 Feedback
FLCA	...Acceleration Feedback
FL2	Filter Constants 2
F2LD	...Kd: Damping
F2LS	...Ksi: Stiffness Per Inertia
F2LA	...Kaa: Anticipated Acceleration
F2LV	...Velocity 2 Feedback
F2L1	...Acceleration 1 Feedback
F2L2	...Acceleration 2 Feedback
	Commands – G…
GCL	Go Closed Loop
GOC	Gravity Offset Constant
GOP	Go Open Loop
	Commands – H…
HLT	Halt

HSM	Hard Stop Move
	Commands – I…
IDT	Identity
IDTG	...Group Number
IDTU	...Unit Number
IMQ	Interpolated Move Queue Clear
IMS	Interpolated Move Start
IMW	Interpolated Move Write Queue
IMWT	...Time
IMWP	...Position
IMWA	...Acceleration
IMWV	...Velocity
	Commands – K…
KDD	Kill Disable Driver
KED	Kill Enable Driver
KMC	Kill Motor Conditions
KMCE	...Condition Enable
KMCS	...Condition State
KMR	Kill Motor Recovery
KMX	Kill Motor Conditions, Extended
KMX1	...Condition Enable ISW
KMX2	...Condition State ISW
KMX3	...Condition Enable IS2
KMX4	...Condition State IS2
KMX5	...Condition Enable XIO
KMX6	...Condition State XIO
	Commands – L…
LPR	Load Program
LPRA	...NV Memory Address
LPRC	...Count
LRP	Load and Run Program
LVP	Low Voltage Processor Trip
LVT	Low Voltage Trip
	Commands – M…
MAT	Move Absolute - Time
MATP	...Position
MATA	...Acceleration Time
MATT	...Total Time
MATE	...Stop Enable
MATS	...Stop State
MAV	Move Absolute - Velocity
MAVP	...Position
MAVA	...Acceleration
MAVV	...Velocity
MAVE	...Stop Enable
MAVS	...Stop State
MCT	Motor Constants
MCT1	...Constant 1
MCT2	...Constant 2
MCT3	...Constant 3
MCT4	...Constant 4
MCT5	...Constant 5
MCT6	...Constant 6

MCT7	...Constant 7
MCT8	...Constant 8
MDC	Modulo Clear
MDS	Modulo Set
MDSC	...Count
MDSE	...Encoder Source
MDSF	...Output Format
MDT	Modulo Trigger
MRT	Move Relative - Time
MRTD	...Distance
MRTA	...Acceleration Time
MRTT	...Total Time
MRTE	...Stop Enable
MRTS	...Stop State
MRV	Move Relative - Velocity
MRVD	...Distance
MRVA	...Acceleration
MRVV	...Velocity
MRVE	...Stop Enable
MRVS	...Stop State
MTT	Maximum Temperature Trip
	Commands – O…
OLP	Open Loop Phase
OVT	Over Voltage Trip
	Commands – P…
PAC	Set Phase Advance Constants
PAC1A	...Phase Advance-P_adv
PAC2A	...Phase Advance-P2_adv
PACL	...Phase Advance-P_limit
PCG	Pre-Calculate Go
PCI	Program Call on Input
PCIL	...Program Buffer Location
PCIE	...I/O Enable
PCIS	...I/O State
PCL	Program Call
PCLL	...Program Buffer Location
PCLE	...Condition Enable
PCLS	...Condition State
PCM	Pre-calculated Move
PCP	Position Compare
PIM	Position Input Mode
PIMF	...Filter Constant
PIME	...I/O Exit Enable
PIMS	...I/O Exit State
PLG	DISPLAY Lowest Group Number
PLR	Power Low Recovery
PMC	Profile Move Continuous
PMCE	...Stop Enable
PMCS	...Stop State
PMO	Profile Move Override
PMOE	...Stop Enable
PMOS	...Stop State
PMV	Profile Move

PMVE	...Stop Enable
PMVS	...Stop State
PMX	Profile Move Exit
POL	READ Polling Status Word
PRI	Program Return on Input
PRIE	...I/O Enable
PRIS	...I/O State
PRT	Program Return
PRTE	...Condition Enable
PRTS	...Condition State
PUN	DISPLAY Unit Number
PUP	Protect User Program
PUPC	...Lockout Code
PUPA	...First Memory Address
PVC	Profile Velocity Continuous
PVCM	...Mode
PVCR	...Starting Data Register
PVCE	...Stop Enable
PVCS	...Stop State
PWO	PWM Output
PWOR	...Register
PWOM	...Mode
	Commands – R…
RAT	Register Move Absolute - Time
RATD	...Data Register
RATA	...Acceleration Time
RATT	...Total Time
RATE	...Stop Enable
RATS	...Stop State
RAV	Register Move Absolute - Velocity
RAVD	...Data Register
RAVA	...Acceleration
RAVV	...Velocity
RAVE	...Stop Enable
RAVS	...Stop State
RGG	Registered Electronic Gearing (REG)
RGGM	...Mode
RGGR	...Starting Data Register
RGGC	...Cycle Count
RIO	READ I/O Status
RIS	READ Internal Status Word
RLM	Register Load Multiple
RLMN	...Number of Registers
RLMD	...Starting Data Register
RLMA	...NV Memory Address
RLN	Register Load from non-volatile
RLND	...Data Register
RLNA	...NV Memory Address
RRT	Register Move Relative - Time
RRTD	...Data Register
RRTA	...Acceleration
RRTT	...Total Time
RRTE	...Stop Enable

RRTS	...Stop State
RRV	Register Move Relative - Velocity
RRVD	...Data Register
RRVA	...Acceleration
RRVV	...Velocity
RRVE	...Stop Enable
RRVS	...Stop State
RRW	Read Register, Write
RRWO	...Operation
RRWR	...Register
RRWD	...Data
RSD	Registered Step and Direction
RSM	Register Store Multiple
RSMN	...Number of Registers
RSMD	...Starting Data Register
RSMA	...NV Memory Address
RSN	Register Store to non-volatile
RSND	...Data Register
RSNA	...NV Memory Address
RSP	Restart, Program Mode
RST	Restart, Immediate
RUN	Run Program
	Commands – S…
SCF	S-Curve Factor
SEE	Select External Encoder
SEEI	...Index State
SEES	...Index Source
SEEE	...Encoder Style
SEF	Select Encoder Filter
SIF	RS-232 = 0, RS-485 = 1
SLC	Single Loop Control
SOB	Set Output Bit
SPR	Store Program
SSD	Scaled Step and Direction
SSE	Single Step Exit
SSI	SSI Port Mode
SSIM	...Mode
SSIR	...Resolution
SSIO	...Options
SSL	Soft Stop Limits
SSP	Single Step Program
STP	Stop
	Commands – T…
T1F	Thread 1 Force
T2K	Thread 2 Kill Exclusions
T2S	Thread 2 Start
T2SA	...NV Memory Address
T2SZ	...Program Buffer Size
TIM	Torque Input Mode
TIMF	...Filter Constant
TIME	...Stop Enable
TIMS	...Stop State
TQL	Set Torque Limits

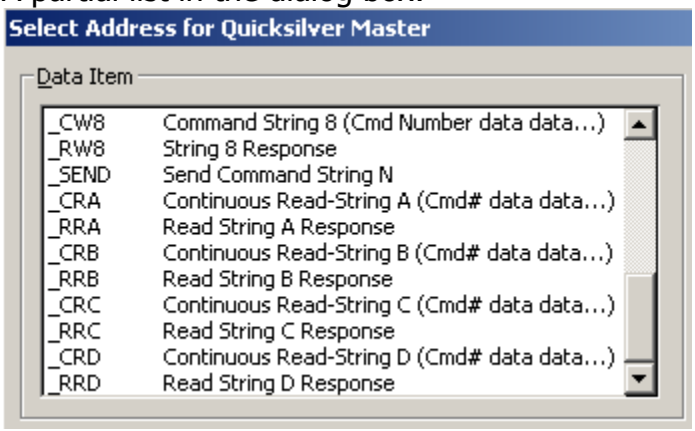
TQLCH	...Closed Loop Holding
TQLCM	...Closed Loop Moving
TQLOH	...Open Loop Holding
TQLOM	...Open Loop Moving
TRU	Torque Ramp Up
TRUF	...Final Value
TRUI	...increment
TTP	Set Target to Position
	Commands – V…
VIM	Velocity Input Mode
VIMF	...Filter Constant
VIME	...I/O Exit Enable
VIMS	...I/O Exit State
VLL	Velocity Limits
VLLM	...Moving Limit
VLLH	...Holding Limit
VMI	Velocity Mode - Immediate. Mode
VMIA	...Acceleration
VMIV	...Velocity
VMIE	...Stop Enable
VMIS	...Stop State
VMP	Velocity Mode - Program Type
VMPA	...Acceleration
VMPV	...Velocity
VMPE	...Stop Enable
VMPS	...Stop State
	Commands – W…
WCL	Write Command Buffer, Long
WCLR	...Register
WCLA	...Program Buffer Address
WCW	Write Command Buffer, Word
WCWR	...Register
WCWA	...Program Buffer Address
WDL	Wait Delay
WRF	Write Register File
WRFR	...Register
WRFD	...Data
WRX	Write Register Extended
WRXO	...Operation
WRXR	...Register
WRXD	...Data
	Commands – X…
XAT	Ext. Register Move Abs. - Time
XATD	...Starting Data Register
XATE	...Stop Enable
XATS	...Stop State
XAV	Ext. Register Move Abs. - Velocity
XAVD	...Starting Data Register
XAVE	...Stop Enable
XAVS	...Stop State
XRT	Ext. Register Move Rel. - Time
XRTD	...Starting Data Register
XRTE	...Stop Enable

XRTS	...Stop State
XRV	Ext. Register Move Rel. - Velocity
XRVD	...Starting Data Register
XRVE	...Stop Enable
XRVS	...Stop State
	Commands – Z…
ZTG	Zero Target
ZTP	Zero Target and Position

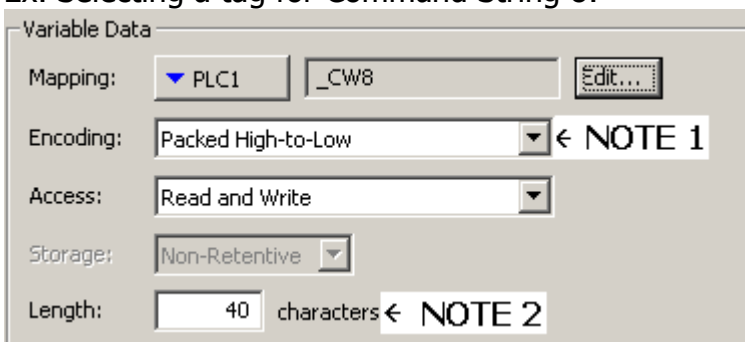
Appendix B – Custom User Commands

Configuring Custom User Commands in Data Tags:

A partial list in the dialog box:



Ex. Selecting a tag for Command String 8:



NOTE 1: Always set Encoding to Packed High-to-Low
NOTE 2: The selected Length must exceed the size of any string programmed. The maximum number of characters in a command, or response, is 40. Only the characters up to the end of the string will be sent.
These properties must be set correctly for all _CW, _RW, _CR, _RR selections.

_SEND is to be configured as an integer tag. The driver will not execute, without indication, any entry that is not in the range 1 through 8.

Configuring Custom User Commands in User Interface:

Configure the tags as any other item would be configured.

Runtime:

The driver automatically prefixes the entered string with "@ dd ", i.e. the start of the command string indicator '@', a space, the device address "dd", and a space. The ending Carriage Return is automatically appended.
All tags _Cxx comprise the command number, and any further parameters separated by the space character.

NOTE: NEVER ASSIGN WRITE COMMANDS TO _CR SELECTIONS. THE DRIVER CANNOT KNOW IF THE COMMAND IS A READ OR A WRITE, AND WILL CONTINUALLY WRITE THE VALUE.

Example: Register 0 is to be read continuously using CR3/RR3:
Read Register command number is 12.
Enter "12 0" into CR3.
RR3 will then display the string:
10 000C hhhh hhhh
where '#' indicates a good read, 10 is the address in hex, 000c is the command number in hex, and hhhh hhhh is the hex value of the 32 bit result.

NOTE: The protocol requires decimal numbers in the command, but returns hexadecimal values in the response.

Example: Register 10 is to be set to 1234 using CW5/RW5:
Write Register Immediate command number is 11.
Enter "11 10 1234" into CW5.
Nothing will happen until _SEND is set to 5. Then RW5 will display the string:
"* 10", where '*' is the successful write indicator, and 10 is the device address.

Example: Register 123 is to be read only upon request using CW7/RW7:
Enter "12 123" into CW7.
Nothing will happen until _SEND is set to 7. Then RW7 will display the string:
10 000C hhhh hhhh

Other considerations:

The response string comprises only those characters that are received up to, but not including, the Carriage Return.

The driver will not send any string that does not begin with a digit 1 through 9. Disable a _C string by writing any other character at the beginning. This is the ONLY check that the driver performs on the entered string. The programmer is responsible for ensuring only valid and complete strings can be entered.

_RW and _RR can be cleared by any write.

It is the responsibility of the programmer to parse the response string for its acknowledge (# or *), negative acknowledge (!), and the data. The Find(), Left(), Right(), Mid(), and TextToInt() commands will be helpful in this regard. For example, one way to extract the data from the response RR1 =
"# 10 000C 1234 5678"
HighWord = TextToInt(Right(RR1, 9), 16)
LowWord = TextToInt(Right(RR1, 4), 16)
Data = (HighWord << 16) + LowWord
The programmer could combine these into a single expression:
Data = (TextToInt(Right(RR1,9) , 16) << 16) + TextToInt(Right(RR1,4) , 16)

The ERR value will not be set by a negative acknowledge returned from a User Command.