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

Yaskawa Universal SMC – Serial Port / TCP

Information Sheet for Crimson v2.0

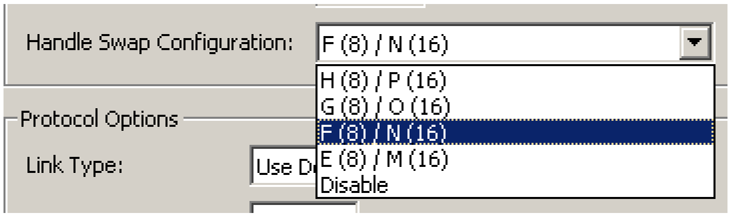
Compatible Devices

- SMC 3010, 4000, 2000

Verified Device

- 4040, 3010

Device Options (TCP Only): REMOVED - Version 1.50



Handle Swap Configuration: F (8) / N (16)

Handle Swap Configuration has been added for distributed control systems. Use ‘Disable ”for all other systems. Select the default handle to which the G3 is to be set. M-P are for 3010, 16 handle systems, E-H are for all others. The G3 will read the type and exchange its assigned handle for the one selected. If the selected one is unavailable, the driver will attempt the next higher. If none of the top 4 handles are available – being assigned to other peripheral devices, the G3 will be unable to communicate. H (8) / P (16) will usually be suitable.

Version Information:

Version 1.50 – Removed Automatic Handle Swap. It has been replaced by a command HDS, which allows the programmer to determine when, and to which handles, the distributed controller’s handles are to be assigned.
Version 1.40 – Added TCP Distributed Control functions. 1) Automatic handle switch to put the G3 above the handles of the axis, and the Tell Inputs (TIA) maximum has been increased to 16 to access inputs in 8 axis systems.
See Device Options for Handle Swap Configuration.
Version 1.30 – Added capability for user defined strings to be sent, and the read responses interpreted as integer, real, or string. Added capability to send write-only user defined strings.

Version 1.21 –Added Device Control for Custom Variable Names. See Note 2. Added Access Control for V, D, and R, User Arrays. See Note 3.

Accessible Data

ABBR	ACTION	ACTION TYPE
AB	Abort Motion and Program	Command
ABA	Abort Motion Only	Command
AC	Acceleration	Integer Read/Write
AE	Absolute Encoder	Integer Write
AF	Enable Digital Feedback	Send 0 to AXIS
AFA	Enable Analog Feedback	Send 1 to AXIS
AL	Arm Latch	Command
AO	Analog Output	Real Write
BG	Begin Motion (Axis Pattern)	Command
BL	Reverse Software Limit	Real Read/Write
BN	Burn	Command
BP	Burn Program	Command
BV	Burn Variables	Command
CB	Clear Bit (Bit Number)	Integer Write
CD	Contour Data	Integer Read/Write
CE	Configure Encoder	Integer Read/Write
CM	Contour Mode (Axis Pattern)	Integer Write
DC	Deceleration	Integer Read/Write
DE	Dual (Auxiliary) Encoder Position	Real Read/Write
DP	Define Position	Real Read/Write
DT	Delta Time	Integer Read/Write
DV	Dual Velocity Enable (Dual Loop)	Integer Read/Write
EA	ECAM Master Axis (Axis)	Integer Write
EB	ECAM Mode Enable	Integer Read/Write
EC	ECAM Counter	Integer Read/Write
EG	ECAM Engage	Integer Read/Write
EM	ECAM Cycle	Integer Read/Write
EQ	ECAM Quit	Integer Read/Write
ER	Error Limit	Real Read/Write
FA	Acceleration Feed Forward	Real Read/Write
FE	Find Edge (Axis Pattern)	Integer Read/Write
FI	Find Index (Axis)	Integer Read/Write
FL	Forward Software Limit	Real Read/Write
FV	Velocity Feed Forward	Real Read/Write
GR	Gear Ratio	Real Read/Write

HM	Home (Axis Pattern)	Integer Read/Write
HX	Halt Program	Integer Read/Write
IA	IP Address (TCP-Read Only)	Integer Read-Write (Serial)
IL	Integrator Limit	Real Read/Write
IP	Increment Position	Real Read/Write
IT	Independent Time Constant	Real Read/Write
JG	Jog	Real Read/Write
KD	Derivative Constant	Real Read/Write
KI	Integrator	Real Read/Write
KP	Proportional Constant	Real Read/Write
MM	Master Modulus	Integer Read/Write
MO	Motor Off (Axis Pattern)	Status/Integer Read/Write
OB	Output Bit	Integer Write
OE	Off on Error - Enable/Disable	Integer Read/Write
OF	Offset	Real Read/Write
OP	Output Port	Integer Read/Write
PA	Position Absolute	Real Read/Write
PR	Position Relative	Real Read/Write
RP	Reference Position	Real Read
RS	Reset	Command
SB	Set Bit (Bit Number)	Integer Read/Write
SC	Stop Code	Integer Read
SH	Servo Here (Axis Pattern)	Integer Write
SP	Speed	Real Read/Write
ST	Stop (Axis Pattern)	Integer Write
TB	Tell Status Byte	Integer Read
TC	Tell Error Code	Integer Read
TD	Tell Dual Encoder	Real Read
TE	Tell Error	Real Read
TI	Tell Inputs (Axis 1 Only)	Integer Read
TIA	Tell Inputs (Any Axis)	Integer Read
TL	Torque Limit	Real Read/Write
TM	Time Command	Integer Read/Write
TP	Tell Position	Real Read
TS	Tell Switches	Integer Read
TT	Tell Torque	Real Read
TV	Tell Velocity	Integer Read
TW	Time Wait	Integer Read/Write
UV	User Variables UV000-UV999	Real Read/Write
V0	Array Variables V0000-V0999	Word Read/Write-Note 1
V1	Array Variables V1000-V1999	Word Read/Write-Note 1
V2	Array Variables V2000-V2999	Word Read/Write-Note 1
V3	Array Variables V3000-V3999	Word Read/Write-Note 1

V4	Array Variables V4000-V4999	Word Read/Write-Note 1
V5	Array Variables V5000-V5999	Word Read/Write-Note 1
V6	Array Variables V6000-V6999	Word Read/Write-Note 1
V7	Array Variables V7000-V7999	Word Read/Write-Note 1
D0	Array Variables D0000-D0999	DWord Read/Write-Note 1
D1	Array Variables D1000-D1999	DWord Read/Write-Note 1
D2	Array Variables D2000-D2999	DWord Read/Write-Note 1
D3	Array Variables D3000-D3999	DWord Read/Write-Note 1
D4	Array Variables D4000-D4999	DWord Read/Write-Note 1
D5	Array Variables D5000-D5999	DWord Read/Write-Note 1
D6	Array Variables D6000-D6999	DWord Read/Write-Note 1
D7	Array Variables D7000-D7999	DWord Read/Write-Note 1
R0	Array Variables R0000-R0999	Real Read/Write-Note 1
R1	Array Variables R1000-R1999	Real Read/Write-Note 1
R2	Array Variables R2000-R2999	Real Read/Write-Note 1
R3	Array Variables R3000-R3999	Real Read/Write-Note 1
R4	Array Variables R4000-R4999	Real Read/Write-Note 1
R5	Array Variables R5000-R5999	Real Read/Write-Note 1
R6	Array Variables R6000-R6999	Real Read/Write-Note 1
R7	Array Variables R7000-R7999	Real Read/Write-Note 1
XQ	Execute Program	Command
AW	V,D,R Array Access Control	Internal Setting –Note 3
	NEW FEATURE IN VERSION 1.40	
WO	Write Specific Output Bit	Bit Write –Note 5
	NEW FEATURES IN VERSION 1.30	
USR	Device Control Fn. 1 Response	User Defined –Note 4
	NEW FEATURE IN VERSION 1.50	TCP Only
HDS	TCP Handle Swap <data> = A ’– P ’	Byte Write –Note 6

NOTE: Certain commands might not be supported by a particular drive. The programmer must verify the selections via the Controller Manual.

ACTION TYPE:

1. COMMAND –A write to these instructions will execute the command.
2. INTEGER READ/WRITE –Reads or writes an integer value.
3. INTEGER READ –Reads an integer. Writes are ignored.
4. INTEGER WRITE –Writes an integer. Returns without indication if a read is attempted.
5. REAL READ/WRITE –Reads or writes floating point values. For some instructions, the drive sets the fractional part to 0 on a read.
6. REAL READ –Read a floating point value. Writes are ignored.
7. REAL WRITE –Writes a floating point number. Returns without indication if a read is attempted.

8. STATUS/INTEGER R/W –Returns the motor status when read. For writing, send the Axis Pattern for the motors that are to turn off.

Not every combination of fractional digits is attainable with real numbers. In some cases a read may not return exactly the same value that was written.

In the Serial driver, Echo is set to off. Also, Position Format and Variable Format are always set to 10.4, 10 digits to the left of the decimal point, and 4 to the right. These are not set in the TCP driver.

Description Comments

(Axis Pattern) means a binary bit pattern is written to indicate which axes are to be affected. Bit 0 is axis X, bit 1 is axis Y, etc. E.g. a value of 5 (0000 0101) written to SH would turn on axis X and axis Z.

(Axis) means a numeric value is written which corresponds to the axis affected. 1 is axis X, 2 is Axis Y, 3 is Axis Z, and 4 is Axis W. E.g. writing 2 to FI will find the Index on axis Y.

(Bit Number) means the number of the bit is written. E.g. writing a 3 to CB will clear bit 3.

(TCP-Read Only). The IA command (IP Address) is read-only using Ethernet. It is Read and Write using Serial.

NOTE 1: User Variables and Arrays

1000 User Variables (UV000 through UV999) are selectable. Each UVnnn is an independent data item, using a single request/response from the drive.

V0…V7, D0…D7, R0…R7: The programmer will allocate arrays V0[],V1[],…V7[], as necessary in the initial SMC program. The driver can access items 000-999 in each of those arrays. All selections Vxnnn, Dxnnn, and Rxnnn, access the same V arrays. Only the data handling is different. For 16 bit Words use the Vxnnn selections. For 32 bit Dwords use Dxnnn. For Real numbers use Rxnnn.

IMPORTANT: Array commands are NOT VALID when the motor is running. Use individual UV variables to read and write while running. The programmer must be aware of the quantity limits for arrays and user variables in the SMC used.

NOTE 2: Custom User Variable Names

Version 1.21 added support for the Device Control Function. This can be used to assign custom variable names to User Real Variables UV968 through UV999.

To use Device Control program VarOK = DevCtrl(D, A, T), where VarOK is an integer that will be set to 1 if the assignment was successful, 0 otherwise.
‘D’ is the device number (not the drop number). Highlight the driver in the Communications Section – the device number is in the status bar.
‘A’ is the UV number to be mapped, and must be in the range 968 through 999.
‘T’ is the name, and can be from 0 to 8 characters.

In the example sequence that follows, assume UV999 is selected.
VarOK = DevCtrl(0, 999, ‘XYZ’) maps UV999 to Drive Variable XYZ.
VarOK = DevCtrl(0, 999, ‘ABX’) remaps UV999 to Drive Variable ABX.
VarOK = DevCtrl(0, 999, “”) remaps UV999 to Drive Variable UV999.

DevCtrl() can be used at any time to change the name mapped to the tag.

It is recommended to first run DevCtrl() using “On Select” in the Page/Properties of the first page.

NOTE 3: AW - V, D, R Array Access Control

AW permits control of the method of accessing V, D, and R, array values. When this selection is 0, Array Downloads and Uploads (QD/QU) commands are executed. These commands are not permitted when the drive is in motion, however, the array values can be accessed individually by setting AW to anything other than 0.

NOTE 4: USR – Device Control Command String

A custom command string may be sent using DevCtrl() functions 1 or 2. Executing DevCtrl(DeviceNumber, DesiredFunction, CommandString) will send CommandString (plus a CR) to DeviceNumber. If DesiredFunction is 1, USR will display the first 32 characters of the string returned, if any. If DesiredFunction is 2, USR will not be changed.

IMPORTANT: The programmer is responsible for constructing the correct string, and for parsing the response if needed.

Mandatory Assignments: Assign USR to a string tag. Set Encoding to Packed High-To-Low, and Length to a maximum of 32 characters.

Writing anything to USR will clear it.

Example: DevCtrl(0, 1, “XQ #Label, 1 ”) will send “XQ #Label, 1 ”and a <CR>. If the response received is ‘[CR][LF]>: ”USR will display the symbols for carriage return and line feed, followed by ‘>: .’

NOTE 5 – WO – Write Specific Output bit

Allows the programmer to assign an output bit to a specific controller in a distributed system. The programmer must calculate the proper number using the Yaskawa specification, in order to send the bit status to the proper device and bit number.

NOTE 6 – HDS – TCP Handle Swap

Only for use with distributed control via TCP.
Allows the programmer to assign a handle to a specific controller. On a read, the current handle value is returned. On a write, the driver executes a handle swap function to assign the distributed controller to the correct handle. In general, the programmer will monitor the current handle, and if it changes because of a power-down of either the controller or the G3, correct it before writing any other data.

TCP Information

The G3 Ethernet Port must be manually configured to the IP of the Drive. The programmer uses the Boot P utility of the Yterm software to assign the IP.
Use a crossover cable for direct connection to G3.

Serial Cable Information

G3 RS232 Port	SMC 232 Port
TxD - 5	RxD – 3
RxD - 2	TxD – 2
CTS – 6	CTS-OUT 4
RTS – 1	CTS-IN 1
0V – 3 and/or 4	5