Host Command Reference

Q and SCL commands for servo and stepper drives

Includes RS-232, RS-485,
Ethernet UDP, Ethernet TCP/IP, EtherNet/IP,
Modbus RTU and Modbus TCP/IP

Applied Motion Products
A MOONS' COMPANY

APPLIED MOTION PRODUCTS

920-0002 Rev. W
7/6/2021

Contents

SO - Set Output

Compatibility: All drives
See Also: IL, IH, IO commands

Sets an output to the given condition. Care must be taken when using outputs on the main driver board because those outputs are by default programmed for dedicated purposes (Alarm, Brake, Motion). Use the AO, BO and MO commands to reconfigure main driver board output usage to general purpose before using the SO command with those outputs.

Command Details:

Structure	SO(Parameter #1)
Type	BUFFERED
Usage	WRITE ONLY
Non-Volatile	NO
Register Access	None

Parameter Details:

(See Appendix F: Working With Inputs and Outputs)

Examples:

Command	Drive sends	Notes
SO1L	-	Set output 1 low (closed)
SO2H	-	Set output 2 high (open)
SOY1L	-	Set main driver board output 1 low (closed)
SOY2H	-	Set main driver board output 2 high (open)

SP - Set Position

Compatibility: All drives
Affects: FP commands
See Also: EP, FP commands

Sets or requests the motor's absolute position. To ensure that the internal position counter resets properly, use EP immediately prior to sending SP. For example, to set the position to zero after a homing routine, send EP0 immediately followed by SP0.

Note that EP is not supported and, therefore, not required for ST5/10-S and ST5/10-Plus drives.

Command Details:

Structure	SP{Parameter #1}
Type	BUFFERED
Usage	READ/WRITE
Non-Volatile	NO
Register Access	None

Parameter Details:

Parameter #1	Absolute position
- units	encoder counts
- range	+/- 2,147,483,647

Examples:

Command	Drive sends	Notes
SP100	-	Set absolute position offset to 100 encoder counts
SP	SP=100	
EP0	-	(Step 1) reset internal position counter
SP0	-	(Step 2) reset internal position counter

SS - Send String

Compatibility: All drives with RS-232 communication

Instructs drive to respond with the desired character string (up to 4 characters). This command is useful for letting the host system know via the serial port when a sequence of commands has finished executing. Multiple SS commands can be placed into the queue at any time, though care should be taken when using this command to avoid serial data collisions. For example, the host system should avoid sending commands to the drive while expecting a character string (from a previously buffered SS command).

NOTE: Due to the possibility of data collisions related to unscheduled communication from slave devices, this command is nonfunctional for RS-485 drives.

Command Details:

Structure	SS(Parameter #1)
Type	BUFFERED
Usage	WRITE ONLY
Non-Volatile	NO
Register Access	None

Parameter Details:

Parameter #1	String of characters
- units	any printable characters
- range	up to 4 characters

Examples:

Command	Drive sends	Notes
SSdone	done	String “done” sent when SS command is executed

ST - Stop

Compatibility: All drives
See Also: SK, SM commands

Halts the current buffered command being executed, but does not affect other buffered commands in the command buffer. When used to stop a move deceleration rate is controlled by the AM (Max Acceleration) command. If a “D” parameter is used deceleration rate is controlled by either the DE command (with “Feed” moves like FL, FP, and SH) or the JL* command (when jogging).

*Note that setting the JA command also sets the JL command. If distinct JA and JL values are required always set JL after setting JA.

Command Details:

Structure	ST{Parameter #1}
Type	IMMEDIATE
Usage	WRITE ONLY
Non-Volatile	NO
Register Access	None

Parameter Details:

Parameter #1	Deceleration rate
- units	letter
- range	D = deceleration rate set by DE or JA command no parameter = deceleration rate set by AM command

Examples:

Command	Drive sends	Notes
ST	-	Stop motion immediately using the deceleration rate set by the AM command
STD	-	Stop motion immediately using the deceleration rate set by the DE or JA command

TD - Transmit Delay

Compatibility: All drives
Affects: RS-232 & RS-485 Serial Communications
See Also: BR, PB & PR commands

Sets or requests the time delay used by the drive when responding to a command that requests a response. Typically this is needed when using the 2-wire RS-485 interface (Half-duplex). Because the same wires are used for both receive and transmit a time delay is usually needed to allow transition time. The Host device's RS-485 specification must be understood to determine the time delay needed.

Command Details:

Structure	TD{Parameter #1}
Type	BUFFERED
Usage	READ/WRITE
Non-Volatile	Yes
Register Access	None

Parameter Details:

Parameter #1	Time value
- units	milliseconds
- range	0 - 32767
- default	10

Examples:

Command	Drive sends	Notes
TD10	-	Set Tx time delay to 10 milliseconds
TD	TD=10	

TI - Test Input

Compatibility: Q drives only
Affects: Condition Code
See also: QJ, TR commands

Tests a digital or analog input against the given input state. If the input is in the state the condition code is set to “T” (true). If not the condition code is set to “F” (false). The condition code is found in read-only register ‘h’ and is most commonly used in conditional jump (QJ) commands. The input is tested, and the jump is performed only if that input is in a specific state.

Command Details:

Structure	TI(Parameter #1)
Type	BUFFERED
Usage	WRITE ONLY
Non-Volatile	NO
Register Access	None

Parameter Details:

(See Appendix F: Working With Inputs and Outputs)

Examples:

Command	Drive sends	Notes
TI4L	-	Test input 4 for a low state
QJT15	-	Jump to line 15 if the previously tested input is “True”

TO - Tach Output

Compatibility: TSM and SV200 servo drives only
Affects: TSM-P output Y3, TSM-S/Q output Y4, and SV200 output Y6
See also: MO command

Defines and configures the tach pulse output function and resolution for the appropriate digital output on a TSM integrated StepSERVO motor or SV200 servo drive.

Note: This command is only supported on TSM and SV200 models. See corresponding hardware manual and software for additional details.

TO parameter list for TSM-P (output Y3) and TSM-S/Q (output Y4):
3: Output is not used as a Motion Output and can be used for a general purpose output.
4: Output is used as a Tach Output at 100 pulses/rev.
5: Output is used as a Tach Output at 200 pulses/rev.
6: Output is used as a Tach Output at 400 pulses/rev.
7: Output is used as a Tach Output at 800 pulses/rev.
8: Output is used as a Tach Output at 1600 pulses/rev.
12: Output is closed (energized) when motor is not moving(static in-position).
13: Output is open (de-energized) when motor is not moving(static in-position).

TO parameter list for SV200 servo drive (output Y6):
0 = 1 * pole pairs
1 = 2 * pole pairs
2 = 4 * pole pairs
3 = 8 * pole pairs
4 = 16 * pole pairs
5 = 32 * pole pairs
6 = 64 * pole pairs
7 = 128 * pole pairs
For “pole pairs” use a value of 4 for all J series servo motors.

Command Details:

Structure	TO(Parameter #1)
Type	BUFFERED
Usage	READ/WRITE
Non-Volatile	YES
Register Access	None

Parameter Details:

Parameter #1	Output Usage (see above)
- units	integer code
- range	see above

Examples:

For TSM:

Command	Drive sends	Notes
TO12	-	Motion Output will close when the motor is not moving
TO	TO=12	

For SV200:

Command	Drive sends	Notes
TO6	-	Tach Output will output 256 pulses/rev
TO	TO=6”	

TR - Test Register

Compatibility: Q drives only
Affects: All data registers
See also: CR, TI, RI, RD, RM, RL, QJ commands

Tests a data register against a given data value. The result of the test is the setting of the condition code, which can be used for conditional programming (see QJ command).

All conditions codes can be set by this command. See “QJ” command for more details.

Command Details:

Structure	TR(Parameter #1)(Parameter #2)
Type	BUFFERED
Usage	WRITE ONLY
Non-Volatile	NO
Register Access	All data registers

Parameter Details:

Parameter #1	Data register
- units	data register assignment
- range	All data registers
Parameter #2	Test value
- units	integer
- range	+/- 2,147,483,647 (long data registers) +/- 32,767 (short data registers)

Examples:

Command	Drive sends	Notes
TR15	-	Test user-defined register “1” against the value 5

TS - Time Stamp

Compatibility: Q drives only
Affects: Data Register “W”
See also: RC, WD, All register commands

Transfers the oldest Time Stamp value from the time stamp buffer into the “W” data register. The time stamp value is a time value in milliseconds, recorded between two input triggers when using the RC command. Each time a defined input condition is “True” (triggered), the elapsed time from the previous input is stored in the time stamp buffer. The time stamp buffer is 8 words deep and acts as a FIFO buffer. The “I” data register, used by the RC command, records when an input trigger has occurred. Sending the RC command clears the time stamp buffer. Executing the TS command removes the oldest time value from the time stamp buffer and places it in the “W” Data Register where it can be used. With each execution of the TS command a time value is transferred, until the end of the time stamp buffer is reached. If a TS is sent with no time values in the time stamp buffer a “0” is placed in the “W” data register.

Command Details:

Structure	TS
Type	BUFFERED
Usage	WRITE ONLY
Non-Volatile	NO
Register Access	“W” (039)
Units	milliseconds

Example:

This sample Q program illustrates the interaction of the RC and TS commands. After initialization, the program waits for a falling edge event on input X3, at which point a 5 second timer begins counting down. During this delay, the user may trigger X3 an arbitrary number of times. After 5 seconds, the motor will execute a series of 5000-step moves, with the delay between each corresponding to the delay between switch closures on X3. That is, if the user trips X3 four times waiting 1 second between each event, the motor will execute four 5000-step moves with a 1 second dwell between each.

```
                MT      1           Multi-tasking ON
                EG      20000       20,000 steps/rev
                AC      250
                DE      250
                VE      5
                FI      3       200         Filter input 3 for 200 processor ticks
                RX      I       0   Zero the “I” register
                RC      X3F         Setup the “I” register for input X3
                WI      X3F         Wait for input X3
                WT      5.00        Wait 5 seconds >>> trigger inpuxt X3 a few times
                TS                  Throw away first time stamp
LABEL2  RD      I                   Decrement “I” register
                FL      5000        Feed 5000 steps
                TR      I       1   Test “I” against 1
                QJ      L       #LABEL1   Jump to end if “I” less than 1
                TS                  Time stamp
                RM      W       1   Move “W” into “1”
                WD      1           Delay for “1” milliseconds
                QG      #LABEL2     Go to Label 2
LABEL1  NO                          Stop program
```

TT - Pulse Complete Timing

Compatibility: SSM, TSM, TXM, SS, SSAC and SV200 series servo drives

This parameter is used to define a time duration. It is used to determine whether the driver has finished receiving all pluses or not. If the drive does not receive any pluses for the period that is longer than TT defined time, then the pulse input stream to the drive is considered complete.

Command Details:

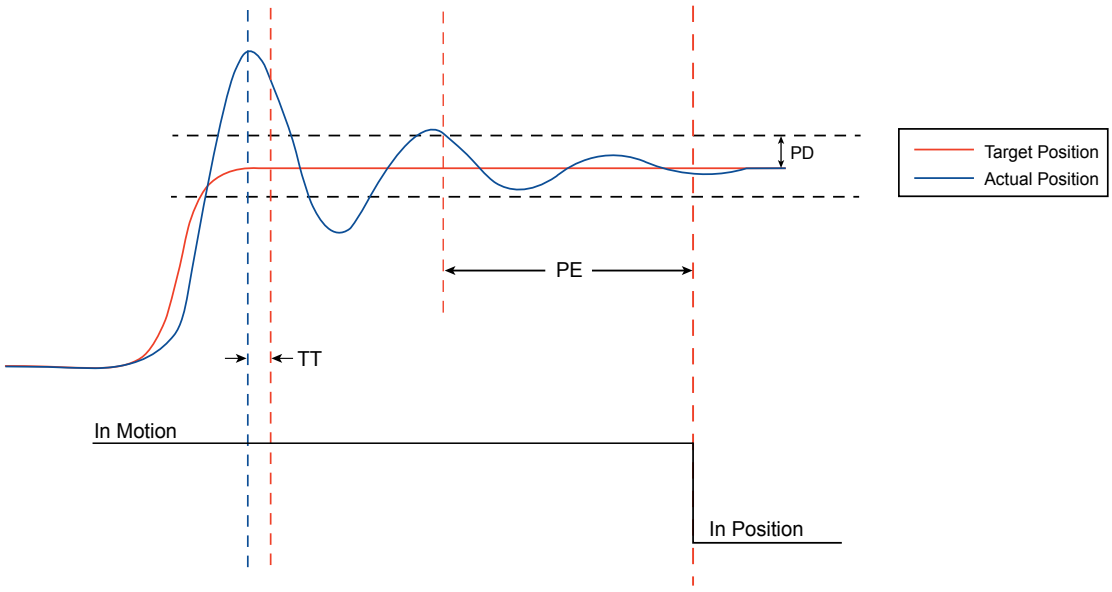
Structure	TT{Parameter #1}
Type	BUFFERED
Usage	READ/WRITE
Non-Volatile	YES
Register Access	None

Parameter Details:

Parameter #1	Pulse Complete Timing
- units	50µsec, (125µsec for SV200)
- range	5~20000
-default	2000

Examples:

Command	Drive sends	Notes
TT1000	-	Sets the pulse complete timing to 50ms for pulse receive complete determination.
TT	TT=1000	



TV - Torque Ripple

Compatibility: SV200 series servo drives only
Affects: Output Y3 in torque mode
See also: MO, PE command

Defines the acceptable torque ripple value around the targeted torque in units of Amperes. If the difference between the actual torque and targeted torque is within the ripple value for the time duration specified by PE command, then the drive defines the actual torque as having met the target torque value. This setting is one of the parameters that controls when the “torque reached” output, which is assigned to digital output Y3, will change state while operating in torque mode.

Command Details:

Structure	TV{Parameter #1}
Type	BUFFERED
Usage	READ/WRITE
Non-Volatile	YES
Register Access	None

Parameter Details:

Parameter #1	See above
- units	Amps
- range	0~4.50
-default	0

Examples:

Command	Drive sends	Notes
TV1	%	Set the torque ripple around the targeted torque to 1Amps
TV	TV=1	The torque ripple around the targeted torque is 1Amps

VC - Velocity Change

Compatibility: All drives
Affects: FC, FD commands

Sets or requests the “change speed” for FC and FD moves..

Command Details:

Structure	VC{Parameter #1}
Type	BUFFERED
Usage	READ/WRITE
Non-Volatile	Yes
Register Access	“U” (037)

Parameter Details:

Parameter #1	Move velocity
- units	rev/sec
- range	BLu, SV, STAC6, ST-Q/Si, ST-S: 0.0042 - 133.3333 (resolution is 0.0042) STM: 0.0042 - 80.0000 (resolution is 0.0042)

Examples:

Command	Drive sends	Notes
VC5	-	Set change velocity to 5 rev/sec
VC	VC=5	

VE - Velocity

Compatibility: All drives
Affects: FC, FD, FE, FL, FM, FS, FP, FY, SH commands

Sets or requests shaft speed for point-to-point move commands like FL, FP, FS, FD, SH, etc.

Command Details:

Structure	VE{Parameter #1}
Type	BUFFERED
Usage	READ/WRITE
Non-Volatile	Yes
Register Access	“V” (038)

Parameter Details:

Parameter #1	Move velocity
- units	rev/sec
- range	BLu, STAC6, : 0.0042 - 133.3333 (resolution is 0.0042) SV200 series, SV7, SVAC3: 0.0042 - 136 (resolution is 0.0042) SSM, TSM, TXM: 0.0042 - 60 (resolution is 0.0042) ST-Q/Si, ST-S, ST-Plus, STM, STAC5, SWM: 0.0042 - 80.0000 (resolution is 0.0042)

Examples:

Command	Drive sends	Notes
VE2.525	-	Set move velocity to 2.525 rev/sec
VE	VE=2.525	

VI - Velocity Integrator Constant

Compatibility: Servo drives only
Affects: Jog commands
See also: VP & JM commands

Sets or requests the velocity-mode (“JM2”) servo control integrator gain term. Gain value is relative: 0 = no gain, 32767 = full gain. VI minimizes steady state velocity errors.

Command Details:

Structure	VI{Parameter #1}
Type	BUFFERED
Usage	READ/WRITE
Non-Volatile	Yes
Register Access	None

Parameter Details:

Parameter #1	Velocity integrator gain value
- units	integer (no specific units)
- range	0 - 32767 (0% - 100%)

Examples:

Command	Drive sends	Notes
VI5000	-	Set velocity integrator gain to 5000
VI	VI=5000	

VL - Voltage Limit

Compatibility: High-voltage Stepper Drives (STAC5, STAC6 only)

Specifies the maximum voltage that will be applied to the motor by the PWM outputs on the drive.

Normally this is set to 100% for modern step motors. Some inexpensive motors are constructed with less robust winding insulation, and require this voltage to be limited. In these rare cases, VL may be lowered. This will directly impact motor performance, but will allow the drive to control a wider variety of motors.

Command Details:

Structure	VL{Parameter #1}
Type	BUFFERED
Usage	READ / WRITE
Non-Volatile	YES
Register Access	None

Parameter Details:

Parameter #1	PWM Duty Cycle
- units	
- range	10 - 1000 (1.0% - 100.0%)

Examples:

Command	Drive sends	Notes
VL1000	-	Maximum voltage applied to the motor: 100.0% (default)
VL	VL = 1000	
VL500	-	Maximum voltage applied to the motor: 50.0%
VL	VL=500	

VM - Maximum Velocity

Compatibility: Servo drives
Affects: Analog Velocity mode
See Also: AM, VC, VE commands

Sets or requests the maximum motor velocity in rev/sec. Used in analog velocity mode to limit the maximum speed of the drive.

Command Details:

Structure	VM{Parameter #1}
Type	BUFFERED
Usage	READ/WRITE
Non-Volatile	YES
Register Access	None

Parameter Details:

BLu, SV

Parameter #1	Move velocity
- units	rev/sec (rps)
- range	0.0042 - 133.3333 (resolution is 0.0042 rev/sec)

Examples:

Command	Drive sends	Notes
VM50	-	Set maximum move velocity to 50 rev/sec
VM	VM=50	

VP - Velocity Mode Proportional Constant

Compatibility: Servo drives only
Affects: Jog commands
See also: VI & JM commands

Sets or requests the velocity-mode servo control Proportional gain term. Gain value is relative: 0 = no gain, 32767 = full gain. VP minimizes velocity error when in velocity mode 2 (see JM command).

Command Details:

Structure	VP{Parameter #1}
Type	BUFFERED
Usage	READ/WRITE
Non-Volatile	Yes
Register Access	None

Parameter Details:

Parameter #1	Velocity mode proportional gain
- units	integer
- range	0 - 32767 (0% - 100%)

Examples:

Command	Drive sends	Notes
VP5000	-	Set velocity mode proportional gain to 5000
VP	VP=5000	

VR - Velocity Ripple

Compatibility: SV200 series servo drives only
Affects: Output Y4
See also: MO, PE commands

Defines the acceptable velocity ripple value around the targeted velocity in units of rev/sec. If the difference between the actual velocity and targeted velocity is within the ripple value, then the drive defines the actual velocity as having met the target velocity value. This setting is one of the parameters that controls when the "velocity reached" output, which is assigned to digital output Y4, will change state while operating in torque or velocity modes.

Command Details:

Structure	VR{Parameter #1}
Type	BUFFERED
Usage	READ/WRITE
Non-Volatile	YES
Register Access	None

Parameter Details:

BLu, SV

Parameter #1	Velocity Ripple (see above)
- units	rps
- range	0~136

Examples:

Command	Drive sends	Notes
VR1	%	Set the velocity ripple around the targeted velocity to 1rps
VR	VR=1	The velocity ripple around the targeted velocity is 1rps

WD - Wait Delay

Compatibility: Q drives only
Affects: None
See also: WI, RX commands

Causes a time delay to occur using a time value from a given data register. The resolution is in milliseconds. Only up to 15 bits of the data register are used, giving a maximum wait time of 32 seconds.

Command Details:

Structure	WD(Parameter #1)
Type	BUFFERED
Usage	WRITE ONLY
Non-Volatile	NO
Register Access	None

Parameter Details:

Parameter #1	Data register
- units	data register assignment
- range	All Read/Write and User-Defined data registers

Examples:

Command	Drive sends	Notes
WD5	-	Wait the number of milliseconds indicated by the value in user-defined data register “5”

WI - Wait for Input

Compatibility: All drives
Affects: Use of “Jog” Inputs
See Also: FI, JE, JD, WD, WM, TI commands

Waits for an input to reach the given condition. Allows very precise triggering of moves if a WI command is followed by a move command. When JE (Jog Enable) is active the drive’s “jog” inputs can be used to jog the motor. JD disables jogging using inputs. (See your drive’s User’s Manual for designation of “jog” inputs).

Command Details:

Structure	WI(Parameter #1)
Type	BUFFERED
Usage	WRITE ONLY
Non-Volatile	NO
Register Access	None

Parameter Details:

(See Appendix F: Working With Inputs and Outputs)

Examples:

Command	Drive sends	Notes
WI3R	-	Wait for input 3 to go high (rising edge) before proceeding to the next command in the queue

WM - Wait on Move

Compatibility: Q drives only
Affects: Queue execution
See also: MT

When in multi-tasking is turned on (see MT command) this command will block execution of subsequent commands until the previously initiated move is complete. This can be any type of move such as “Feeds”, “Jogging” or the “Hand Wheel” (encoder following).

Command Details:

Structure	WM
Type	BUFFERED
Usage	WRITE ONLY
Non-Volatile	NO
Register Access	None

Examples:

Command	Drive sends	Notes
WM	-	Causes queue execution to wait until any move in progress is complete

WP - Wait Position

Compatibility: Q drives only
Affects: Multi-velocity, or complex, move profiles
See Also: FC, DC, VC, MT commands

When multi-tasking is turned on (“MT1”), this command is used in conjunction with the DC command to block program execution until a specific position(s) is reached during a move. When the position(s) specified by the DC command is reached program execution continues.

Common example:
This command is used as a “separator” in changing the motor speed of multi-velocity move profiles created using the FC command. The normal FC command provides for one speed change using values determined by DC and VC commands executed prior to the FC command. Additional speed changes can be added after an FC command is initiated by using the WP command to separate additional DC and VC commands. See the example below.

NOTE: This command, along with the ability to create multi-velocity move profiles with the FC command, is only available in BLu servo drive firmware revisions 1.53C or later. This command is available in all firmware revisions of STAC stepper drives.

Command Details:

Structure	WP
Type	BUFFERED
Usage	WRITE ONLY
Non-Volatile	No
Register Access	None

Examples:

Command	Drive sends	Notes
DI100000	-	Overall move distance set to 100,000 counts
VE10	-	Initial move speed set to 10 rps
DC80000	-	1st change distance set to 80,000 counts
VC15	-	1st change speed set to 15 rps
FC	-	Initiate FC command (complex move)
WP	-	
DC60000	-	2nd change distance set to 60,000 counts
VC9	-	2nd change speed set to 9 rps
WP	-	
DC40000	-	3rd change distance set to 40,000 counts
VC1	-	3rd change speed set to 1 rps
WP	-	
DC20000	-	4th change distance set to 20,000 counts
VC19	-	4th change speed set to 19 rps

WT - Wait Time

Compatibility: All drives
See also: WD command
Causes a time delay in seconds. The resolution is 0.01 seconds with the largest value being 320.00 seconds.

Command Details:

Structure	WT(Parameter #1)
Type	BUFFERED
Usage	WRITE ONLY
Non-Volatile	NO
Register Access	None

Parameter Details:

Parameter #1	Time
- units	seconds
- range	0.00 - 320.00 (resolution is 0.01 seconds)

Examples:

Command	Drive sends	Notes
WT2.25	-	Causes time delay of 2.25 seconds

ZA – Network Communication Time-out (Watchdog) Action

Compatibility: MDX, SSDC, STF, STM23X, TSM14POE, TSM23X, TSM34, TXM24X, TXM34, TXM34XSee also:
Affects: Ethernet Communications
See also: ZE, ZS

Sets or requests the action taken when communication loss has been detected. For eSCL-type drives, the timer starts when ZE is set to 1 (Enabled). For Ethernet/IP-type drives, the timer starts when ZS is non-zero and a valid application-specific message has been received (i.e. opening an implicit connection, sending an explicit message, connecting via configuration/tuning software).
When the drive does not receive an application message (eSCL UDP/TCP commands, Ethernet/IP implicit data, or Ethernet/IP explicit command) for the amount of time specified in ZS, then the drive performs that action set by ZA:
ZA1 to 12: Execute a certain Q program from 1 to 12
ZA13: Immediately stops with the AM value deceleration and is still enabled
ZA14: Immediately stops with DE or JL value deceleration and still enables
ZA15: Immediately stops and disables with the AM value deceleration
ZA16: Immediately stops and disables with DE or JL value deceleration

Command Details:

Structure	ZA{Parameter #1}
Type	BUFFERED
Usage	READ/WRITE
Non-Volatile	YES
Register Access	None

Parameter Details:

Parameter #1	Network Time-out Action
- units	n/a
- range	1 -16 (see above for selections)
- default	13

Examples:

Command	Drive sends	Notes
ZA5	-	Run Q Segment 5 after ZS timeout
ZA	ZA=5	
ZA14	-	Stop drive with DE and keep drive enabled after ZS timeout
ZA	ZA=14	

ZC - Regen Resistor Continuous Wattage

Compatibility: BLuAC5 and STAC6 drives only

Sets or requests the regeneration resistor wattage value. BLuAC and STAC drives dynamically calculate the continuous wattage induced into an external regeneration resistor and must know the continuous wattage rating of the regen resistor to do this effectively.

Command Details:

Structure	ZC{Parameter #1}
Type	BUFFERED
Usage	READ/WRITE
Non-Volatile	Yes
Register Access	None

Parameter Details:

Parameter #1	Continuous wattage value of regen resistor
- units	Watts
- range	1 - 1000

Examples:

Command	Drive sends	Notes
ZC250	-	External regen resistor with value of 250 continuous watts is connected to the drive

ZE - Network Communication Time-Out (Watchdog) Enable

Compatibility: MDX, SSDC, STF, STM23X, TSM14POE, TSM23X, TSM34, TXM24X, TXM34, TXM34X
Affects: Ethernet Communications
See also: ZA, ZS

For eSCL-type drives, setting this to 1 (Enable) will start the Network Communication Watchdog feature. When the drive does not receive a network application message (eSCL UDP/TCP commands) for the time set in ZS, the drive will perform the action set by ZA.
For Ethernet/IP-type drives, this is automatically triggered internally, and should be left at 0 (Disable). To Enable the Network Watchdog feature, set ZS (delay time) to a non-zero value. Otherwise, to Disable, set ZS to 0.

Command Details:

Structure	ZE{Parameter #1}
Type	BUFFERED
Usage	READ/WRITE
Non-Volatile	NO
Register Access	None

Parameter Details:

Parameter #1	Network Time-out Enable
- units	n/a
- range	0 – 1
- default	0

Examples:

Command	Drive sends	Notes
ZE1	-	Enable communication watchdog
ZE	ZE=1	

ZR - Regen Resistor Value

Compatibility: BLuAC5 and STAC6 drives only

Sets or requests the regeneration resistor value. BLuAC and STAC drives dynamically calculate the continuous wattage induced into an external regeneration resistor and must know the value of the regen resistor to do this effectively.

Command Details:

Structure	ZR{Parameter #1}
Type	BUFFERED
Usage	READ/WRITE
Non-Volatile	Yes
Register Access	None

Parameter Details:

Parameter #1	Value of regen resistor
- units	Ohms
- range	25 - 100

Examples:

Command	Drive sends	Notes
ZR50	-	50 ohm external regen resistor connected to drive

ZS- Network Communication Time-out (Watchdog) Delay

Compatibility: MDX, SSDC, STF, STM23X, TSM14POE, TSM23X, TSM34, TXM24X, TXM34, TXM34X
Affects: Ethernet Communications
See also: ZA, ZE
Sets or requests the amount of time before a communication loss is detected. The internal timer is reset when a new network application message (i.e. eSCL UDP/TCP command, Ethernet/IP implicit data, Ethernet/IP explicit message) is received and then proceeds to counts down. When the internal timer reaches zero, the action set by ZA is performed.
For eSCL-type drives, the timer is effective once ZE is set to 1. For Ethernet/IP-type drives, the timer automatically starts when the 1st network application message is received and ZS is non-zero.

Note for Ethernet/IP-type drives: ZS is only read at power-up and will require a power-cycle to be effective.

Command Details:

Structure	ZS{Parameter #1}
Type	BUFFERED
Usage	READ/WRITE
Non-Volatile	YES
Register Access	None

Parameter Details:

Parameter #1	Network Time-out Delay
- units	ms
- range	0 - 10000
- default	500

Example:

Command	Drive sends	Notes
ZS1000	-	Set communication timeout to 1000ms (1s)
ZS	ZS=1000	

ZT - Regen Resistor Peak Time

Compatibility: BLuAC5 and STAC6 drives only

Sets or requests the regeneration resistor time constant. Decides the peak time that the resistor can tolerate full regeneration voltage. When regeneration occurs the full regeneration voltage of 400 volts is applied across the resistor. The peak wattage is typically very high, for example with the built-in 40 ohm resistor the peak wattage is 4000 Watts. Power resistors will tolerate this for only a brief period of time. In the case of the built-in 40 ohm/ 50 Watt regen resistor it is only 0.3125 seconds. The ZT value provides the resistor time constant used to create the “filter” for calculating average wattage in the regen resistor.

Command Details:

Structure	ZT{Parameter #1}
Type	BUFFERED
Usage	READ/WRITE
Non-Volatile	Yes
Register Access	None

Parameter Details:

Parameter #1	Maximum time for peak regen
- units	0.25 milliseconds
- range	1 - 32000

Examples:

Command	Drive sends	Notes
ZT1250	-	Regen resistor peak time set to 0.3125 seconds
ZT	ZT=1250	

Data Registers

Many of the commands listed in this reference function by transferring data to a drive for later use. These data values are stored in data registers within the drive and remain there until new commands change the values or power is removed from the drive. For example, if you send the Velocity command “VE10”, a maximum move speed of 10 rev/sec is placed in the data register for velocity. You can then execute as many FL (Feed to Length), FP (Feed to Position) or FS (Feed to Sensor) move commands as you’d like without sending another VE command: the move speed of 10 rev/sec will remain in the velocity data register until you change it.

In addition to the data register for velocity, there are registers for move acceleration (AC command, “A” register), deceleration (DE command, “B” register) and move distance (DI command, “D” register). There are also registers for limit sensors (DL command), motor current (CC command), encoder resolution (ER command), motor position (SP command) and encoder position (EP command). There are 75 data registers in all. See the following Data Register Assignments section for a complete listing of data registers available in your drive.

Not all commands function by transferring a data value into a register. Conversely, not all data registers are associated with a command. To access data registers that are not associated with a command, you can use a register’s unique character assignment. See the Data Register Assignments on the following pages for a listing of data registers and their character assignments. When accessing a data register using its character assignment you use the RL (Register Load Immediate) or RX (Register Load Buffered) commands. These commands allow you to load data values into a register as well as read back the contents of a data register. For example, we set the move speed to 10 rev/sec in the first paragraph of this page by using the velocity command “VE10”. You can accomplish the same thing by using the RL command and the character assignment for the velocity data register, “V”. By sending “RLV2400” to the drive (see units of “V” register in Data Register Assignments section) you set the move speed to 10 rev/sec.

There are four categories of data registers available with your drive: Read-Only, Read/Write, User-Defined, and Storage. The last two categories, User-Defined and Storage, are only for use with Q drives.

Read-Only data registers

Read-Only data registers are predefined registers that contain information about drive parameters, settings, and states. These include registers for commanded current, encoder position, analog input levels, drive temperature, internal bus voltage, and more. You cannot transfer data values to a Read-Only data register; you can only read the contents of them (see RL and RX commands). Read-Only registers are assigned to lower-case letters.

Read/Write data registers

Read/Write data registers are predefined registers that contain drive and move parameters that can be set by the user. These parameters include acceleration rate, velocity, move distance, continuous current setting, peak current setting, and more. Many of the Read/Write registers are associated with a particular command, so you can read their contents or load data into them with RL, RX, or that parameter’s particular command. Read/Write registers are assigned to upper-case letters.

User-Defined data registers

User-Defined data registers are read/write registers that are not predefined. These registers are only used with Q drives. They allow you to create more flexible and powerful Q programs through math functions, incrementing and decrementing, conditional processing, and more. These registers are assigned to single-digit numbers and other ASCII characters.

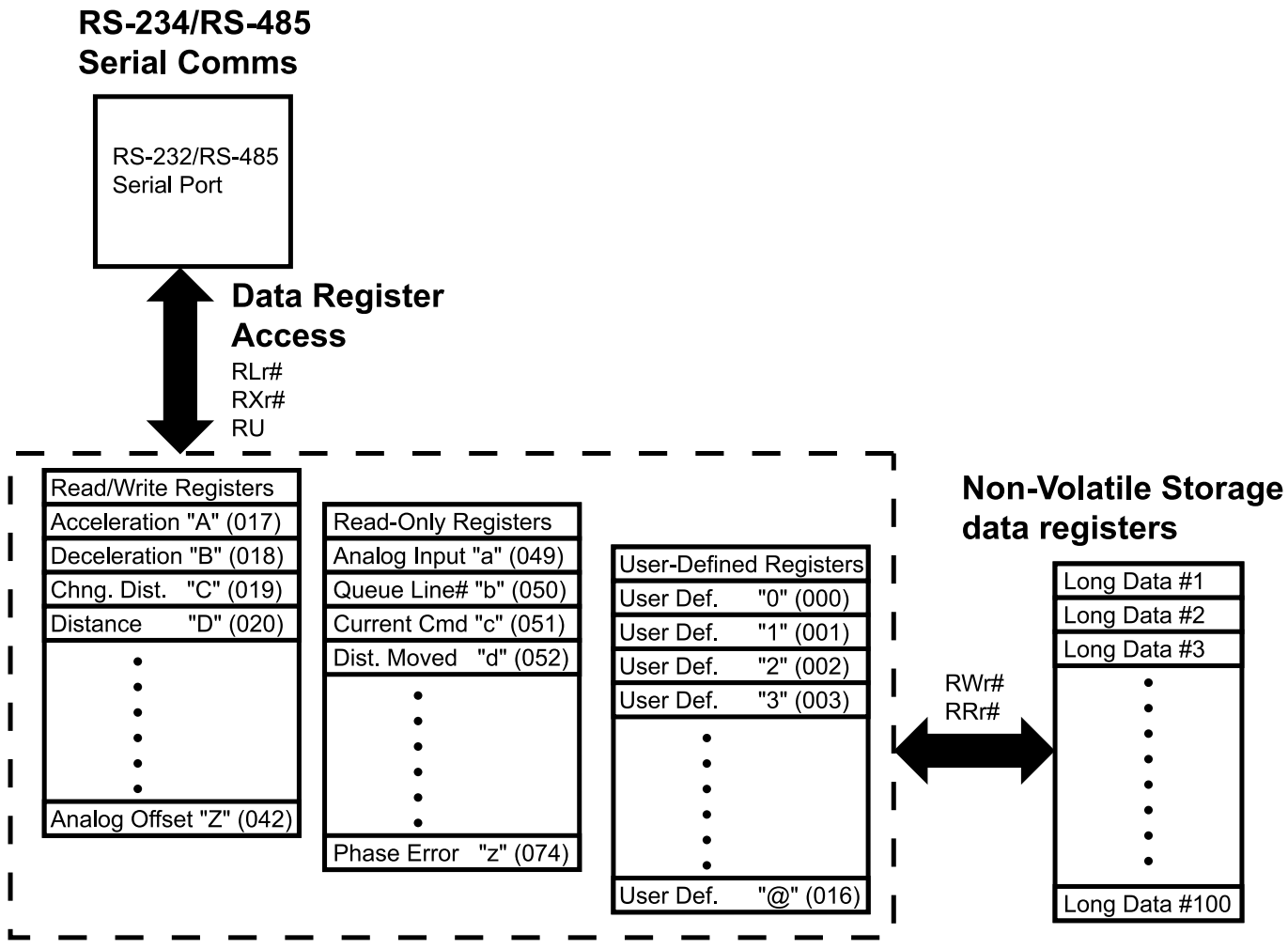
Storage data registers

Each Q drive comes with 100 non-volatile Storage data registers, which can be used to save the contents of other data registers to non-volatile memory. For example, since none of the User-Defined data registers are non-volatile, a user may want to save the values of some of these registers to memory. This can be done by transferring their values to Storage registers (called Writing) before power down of the drive. Then at the next power up, these values can be loaded back into the User-Defined registers from the Storage registers (called

Reading). Each Storage register can save one data register value, and the Storage registers are numbered 1 to 100. See the RR, RW, and SA commands as well as the Appendix for more information on accessing this section of memory.

Using Data Registers

The diagram below shows how a drive's serial port accesses the different volatile (Read-Only, Read/Write, User-Defined) and non-volatile (Storage) data registers within a drive. The user can Load and Upload data register values using the RL, RX, and RU commands via the drive's serial port(s). Read-Only data registers can be uploaded but not loaded. For Q drives only, non-volatile memory is available for data registers in the form of Storage registers. Moving the contents of the volatile data registers back and forth between the non-volatile Storage registers is done with the RW and RR commands. See below for more details.



Loading (RL, RX)

Accessing data registers is done by Loading data into a register, and Uploading data from a register. Loading a data register can be done from a host command line or from a line in a program. To load a register from a host command line use the RL (Register Load) command. This command can be executed at any time, even while a drive is running a program. The RL command is an immediate command. To load a register within a Q program use the RX command, which is a buffered version of Register Load.

Uploading (RL, RU)

Uploading data registers can only be done from a host command line, not within a program. There are two commands available for uploading register values. RL is used to upload one register value at a time, while RU can be used to upload a single register value or an array of register values. Both RL and RU are immediate commands, and therefore can be executed while a program is running. The RU command can request up to 16 data register values in sequence back from the drive. This is great when an array of information is required at one time.

Writing Storage registers (RW) *(Q drives only)*

Writing a data register allows the user to store data register values in non-volatile memory. To write a data register we use the RW (Register Write) command. There are 100 storage locations for data registers in NV memory. Note that the user must keep track of where data registers are stored because the NV memory locations are not associated with any specific data register.

Reading Storage registers (RR) *(Q drives only)*

Reading a data register allows the user to move data previously saved in NV memory into a data register. To read a data register we use the RR (Register Read) command. Reading is typically done in the midst of a Q program.

The following sub-sections describe additional usage of data registers within Q drives only.

Moving data registers (RM) *(Q drives only)*

Data register values can be moved from one register to another. This is done with the RM (Register Move) command. When executing an RM command, the contents of the originating data register are retained. Contents of read-only registers can be moved into read/write registers and user-defined registers. However, as implied by its label, no register values can be moved into read-only registers. Attempting to do so will have no effect and no error code is generated.

Incrementing/Decrementing (RI, RD) *(Q drives only)*

Read/write and user-defined registers can be incremented and decrmented by “1”. Two commands are used for these functions: the RI (Register Increment) and RD (Register Decrement) command. NOTE: Incrementing past the range of a data register will cause the value to wrap around.

Counting (RC, “I” register) *(Q drives only)*

A special data register, the “I” register (Input Counter), is designated for counting input transitions and input state times of a selected digital input. The “I” register is a read/write register that can be used with all other register functions including math and conditional testing.

The RC (Register Counter) command is used to assign digital inputs to register counting. There are four different input states that can be chosen and that have different effects on input counting. When using the “high” or “low” level states the counter acts as a “timer” with a resolution of 100 microseconds (SV servo drives and all stepper drives) or 125 microseconds (BLu servo drives). Edge type states like “falling” or ‘”rising” are used for input counting. (See details of the RC command in the Q Command Reference).

Math & Logic (R+, R-, R*, R/, R&, RI) *(Q drives only)*

Math and logic functions can be performed on data registers. Math is limited to integer values. Some of the math functions are also limited to 16-bit values. When doing math only one operation can be done per instruction. Math and logic results are stored in the Accumulator register, “0”. This register is part of the user-defined register set. Math functions include Add, Subtract, Multiply and Divide. Logic functions include Logical AND and Logical OR.

Conditional Testing (CR, TR) *(Q drives only)*

When constructing complex programs it is usually necessary to do some conditional processing to affect program flow. Two commands are available for evaluating a data register for conditional processing, the TR (Test Register) and CR (Compare Register) commands. The TR command will compare the "First" value of a given data register against a "Second" immediate value. The CR command compares the "First" value of a given data register against the "Second" value of another data register. When using the TR and CR commands an internal "Condition" register is set with the result. The result can be:

"True"	the "First" value is either positive or negative
"False"	the "First" value is not a value (it's zero)
"Zero"	the "First" value equals "0"
"Positive"	the "First" value is "positive"
"Negative"	the "First" value is "negative"
"Greater Than"	the "First" value is more positive than the "Second" value
"Less Than"	the "First" value is more negative than the "Second" value
"Equal to"	the "First" and "Second" values are equal
"Unequal to"	the "First" and "Second' values are not equal

NOTE: The QJ (Queue Jump) command is designed to use the "Condition Codes" above for jumping. The Condition Code can also be accessed via the "h" register.

Data Register Assignments

What follows is a listing of all the data registers available with Applied Motion drives. In the tables below, "Ch." denotes the data register's character assignment, and "Description" gives the name of the data register. The column "3-digit" denotes the register's 3-digit equivalent numerical assignment (see PR command, bit 5); "Data Type" designates whether the data register is a 16-bit word (Short) or a 32-bit word (Long); "Units" shows how a data register's contents are used by the drive; and, "Compatibility" shows which drives can make use of the given register.

NOTE: When programming a Q drive with the Q Programmer software only the character assignment of the register can be used. When communicating to a Q drive via one of its serial ports, either the character assignment or the 3-digit numerical assignment can be used.

Read-Only data registers: a - z

Many of the Read-Only data registers can be read with a specific command. In the tables below, associated commands are shown in parentheses in the "Description" column.

Ch.	Description	3-digit	Data Type	Units	Compatibility
a	Analog Command value (IA)	049	Short	BLu, SV, STAC6, ST-Q/Si: 32760 = +10V; -32760 = -10V ST-S, STM: 16383 = +5V; 0 = 0V*	All drives

*Note that the "a" register is affected by the AV (Analog Offset) command, so the range may vary beyond 0 to 16383.

b	Queue Line Number	050	Short	Line # 1 - 62	Q drives only
c	Current Command (IC)	051	Short	Servo: 0.01 amps RMS Stepper: 0.01 amps, peak-of-sine	All drives

d	Relative Distance (ID)	052	Long	Servo: encoder counts Stepper: steps	All drives

BLu, STAC6

The “d” register (as well as the ID command) contains the relative move distance used in the last move. This means that the “d” register is only updated at the end of every relative move.

SV, ST-Q/Si, ST-S, STM

The “d” register (as well as the ID command) contains the immediate relative distance moved since the start of the last or current relative move. This means the “d” register is updated during relative moves and can therefore be polled during a move to see where the motor is with respect to the overall relative move distance.

e	Encoder Position (IE, EP)	053	Long	encoder counts	Servo drives and stepper drives with encoders

The “e” register can be zeroed by sending the command EP0.

f	Alarm Code (AL)	054	Long	hexadecimal equivalent of binary Alarm Code word (See AL command for details)	All drives
g	Sensor Position	055	Short	Servo: encoder counts Stepper: steps	All drives

The “g” register contains the absolute position of the point at which the input condition is met during moves like FS, FE, SH, and other “sensor-type” moves. It is common practice to use the EP and SP commands to establish known absolute positions within an application or program, which will make the value of the “g” register most meaningful. Otherwise, the absolute position of the motor is zeroed at every power-up of the drive.

h	Condition Code	056	Short	decimal equivalent of binary word (see below)	Q drives only

The response to the “RLh” command will be the decimal equivalent of the condition code’s binary word. Bit assignments and examples are shown below.

Description	Bit #	Decimal Value
TRUE (non-zero)	0	1
FALSE (zero)	1	2
POSITIVE	2	4
NEGATIVE	3	8
GREATER THAN	4	16
LESS THAN	5	32
EQUAL TO	6	64
UNEQUAL TO	7	128

Example:

Command	Drive Sends	Notes
RLh	RLh=149	Bits 7 (UNEQUAL TO), 4 (GREATER THAN), 2

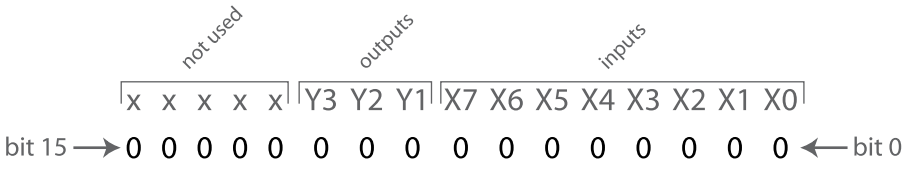
(POSITIVE) and 0 (TRUE) are set. Within a Q program the programmer will often have more than one condition to choose from when using the QJ command. The condition FALSE in Q Programmer is represented by bit 0 = 0 (opposite of TRUE).

i	Driver Board Inputs (ISX)	057	Short	decimal equivalent of binary bit pattern (see below)	All drives

Details when executing the “RLi” command:

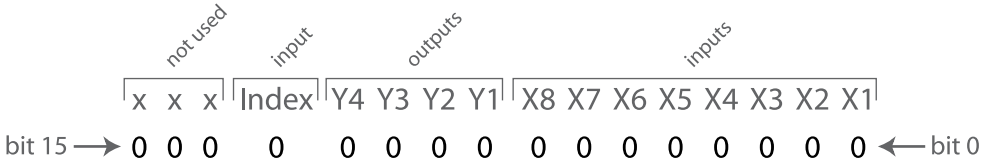
BLu, STAC6

The bit pattern of the “i” register breaks down as follows: bit 0 is the state of the encoder’s index (Z) channel, also known as input X0; bits 1 - 7 represent the states of driver board inputs X1 - X7, respectively; bits 8 - 10 represent the states of driver board outputs Y1 - Y3, respectively; and, bits 11 - 15 are not used. For bits 0 - 7 (inputs X0 - X7), a state of “1” means the optically isolated input is open, and a state of “0” means the input is closed. It is the exact opposite for bits 8 - 10 (outputs Y1 - Y3), for which a state of “1” means the optically isolated output is closed, and a state of “0” means the output is open.



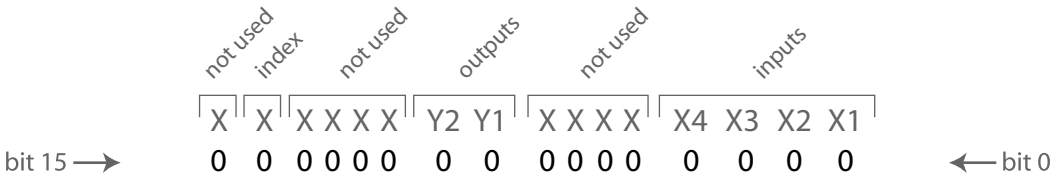
SV, ST-Q/Si

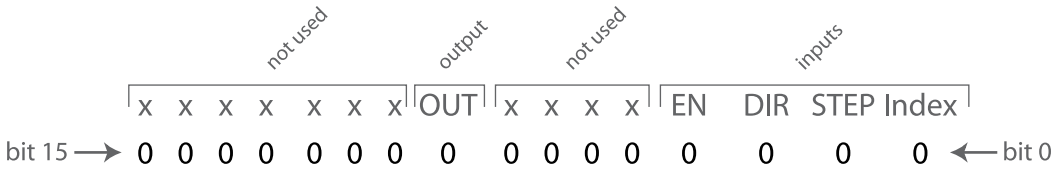
The bit pattern of the “i” register breaks down as follows: bits 0 - 7 represent inputs X1 - X8, respectively; bits 8 - 11 represent outputs Y1 - Y4, respectively; and, bit 12 is the encoder index channel (if present). For bits 0 - 7 and 12 (inputs X1 - X8 and the Index), a state of “1” means the optically isolated input is open, and a state of “0” means the input is closed. It is the exact opposite for bits 8 - 11 (outputs Y1 - Y4), for which a state of “1” means the optically isolated output is closed, and a state of “0” means the output is open.



ST-S, STM

The bit pattern of the “i” register breaks down as follows: bit 0 represents the encoder index channel (if present), bit 1 represents the STEP input, bit 2 the DIR input, and bit 3 the EN input. Bit 8 represents the drive’s single output, OUT. For bits 0 - 3 (Index, STEP, DIR, and EN inputs), a state of “1” means the optically isolated input is open, and a state of “0” means the input is closed.





SVAC3, STAC5

The bit pattern of the "i" register breaks down as follows: bits 0-3 represent inputs X1-X4, respectively; bits 8 and 9 represent outputs Y1 and Y2, and bit 14 represents the encoder index channel (if present). represents the STEP input, bit 2 the DIR input, and bit 3 the EN input. Bit 8 represents the drive's single output, OUT. For bits 0-3 and 14 (X1-X4 and the Index), a state of "1" means the optically isolated input is open, and a state of "0" means the input is closed.

j	Analog Input 1 (IA1)	058	Short	raw ADC counts, 0 - 32760 16383 = 0 volts for BLu, SV, STAC6, ST-Q/Si drives	All drives
k	Analog Input 2 (IA2)	059	Short	raw ADC counts, 0 - 32760 16383 = 0 volts	BLu, SV, STAC6, ST-Q/Si only
l	Immediate Absolute (trajectory) Position	060	Long	Encoder counts (servo), or motor steps (stepper).	All drives
m	Command Mode (CM)	061	Short	Mode #	All drives
n	Velocity Move State	062	Short	State # (see below)	All drives

Response details to the "RLn" command:

Description	Decimal Value	Comment
WAITING	0	In velocity mode waiting for a command
RUNNING	1	Doing a velocity move (jogging)
FAST STOPPING	2	Stopping a velocity move (ST or SK with no parameter)
STOPPING	3	Stopping a velocity move (SJ, STD, or SKD)
ENDING	4	Clean up at end of move (1 PWM cycle, 62 usec)

o	Point-to-Point Move State	063	Short	State # (see below)	All drives

NOTE: The Point-to-Point Move State is only defined during FL, FP, and FS commands.

Details when using "RLo" command:

Description	Decimal Value	Comment
WAITING	0	In position mode waiting for command
WAITING ON BRAKE	1	Waiting for brake to release
CALCULATING	2	Doing the calculations for the move
ACCELERATION	3	Accelerating up to speed
CHANGE VELOCITY	4	Changing the speed (accel or decel)
AT_VELOCITY	5	At the desired speed
DECELERATION	6	Decelerating to a stop
FAST DECELERATION	7	Doing a fast deceleration (ST or SK)
POSITIONING	8	Clean up at end of move (1 PWM cycle, 62 usec)

p	Segment Number	064	Short	Segment # 1 - 12	Q drives only
q	Actual Motor Current (IQ)	065	Short	0.01 Amps	Servo drives only
r	Average Clamp Power	066	Short	Watts	BLuAC5, STAC6

s	Status Code (SC)	067	Short	hexadecimal equivalent of binary Status Code word (See SC command for details)	All drives
t	Drive Temperature (IT)	068	Short	0.1 °C	All drives
u	Bus Voltage (IU)	069	Short	0.1 Volts	All drives
v	Actual Velocity (IV0)	070	Short	0.25 rpm	Servo drives and stepper drives with encoder
w	Target Velocity (IV1)	071	Short	0.25 rpm	All drives*

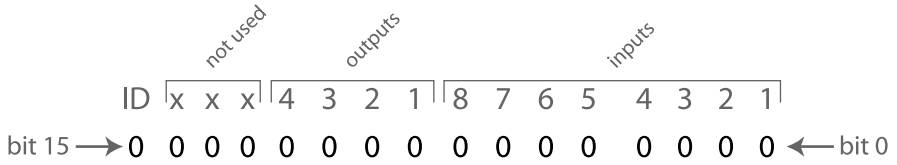
*For stepper drives, the “w” register is only updated when Stall Detection or Stall Prevention is turned on.

x	Position Error (IX)	072	Long	encoder counts	Servo drives and stepper drives with encoder
y	Expanded Inputs (IS)	073	Short	bit pattern	BLu, STAC6

Details when executing the “RLy” command:

BLu, STAC6, SVAC3 and STAC5 drives

The bit pattern of the “y” register breaks down as follows: bits 0 - 7 represent the states of top board inputs 1 - 8, respectively; bits 8 - 11 represent the states of driver board outputs 1 - 4, respectively; and, bits 12 - 15 are not used. For all I/O bits 0 - 11 (inputs 1 - 8 and outputs 1 - 4), a state of “1” means the optically isolated input or output is open, and a state of “0” means the input or output is closed. Bit 15 represents the ID bit, which simply holds a 1 if the IN/OUT2 or screw terminal I/O board is present and a 0 of it’s not. In other words, for SE, QE and Si drives the ID bit will equal 1. For S and Q drives the ID bit will equal 0.



For example, if top board inputs 3 and 5 and top board outputs 1 and 2 were all closed, the response of the drive to the command “RLi” would be “RLi=-29461” (1000 1100 1110 1011). For a more efficient use of the “y” register it is recommended to mask off the ID bit and the other three not used bits. This can be done by using the R& (Register AND) command with the “y” register and a User Defined register set with the value 4095 (0000 1111 1111 1111 1111). Following a register AND operation (&), this will reject the top 4 bits, leaving the rest of the data untouched. For example, the command sequence would look like this.

RL14095	Load User Defined register “1” with the value 4095
R&y1	Register AND the “y” and “1” registers
RL0	Request the value stored in the Accumulator register “0” to which the drive’s response would be RL0=3307.

z	Phase Error	074	Short	encoder counts	Servo drives only

Read/Write data registers: A - Z

Many of the Read/Write data registers are associated with a specific command. In the tables below, associated commands are shown in parentheses in the “Description” column.

NOTE: When using registers pay attention to units. In the case of some Read/Write registers, the units of the register when using the RL and RX command are different than when using the same register’s associated command. For example, the “V” register uses units of 0.25 rpm, but its associated command, VE, uses revs/sec (rps). The reason for this difference is that all registers operate with integer math. On the other hand, when using commands it is often possible to include decimal places which allow for more user-friendly units.

Ch.	Description	3-digit	Data Type	Units	Compatibility
A	Acceleration (AC)	017	Short	10 rpm/sec	All drives

The “A” register units are 10 rpm/sec, which means that the value of the “A” register is equal to 6 times the AC command value. In other words, to achieve an acceleration value of 100 rev/sec/sec send the command RLA600.

NOTE: Take care to ensure that this register is never set to zero. The drive may become stuck in a command mode or program loop and/or refuse to move. See the RL, RM, and RX commands.

B	Deceleration (DE)	018	Short	10 rpm/sec	All drives

The “B” register units are 10 rpm/sec, which means that the value of the “B” register is equal to 6 times the DE command value. In other words, to achieve a deceleration value of 100 rev/sec/sec send the command RLB600.

NOTE: Take care to ensure that this register is never set to zero. The drive may become stuck in a command mode or program loop and/or refuse to move. See the RL, RM, and RX commands.

C	Change Distance (DC)	019	Long	counts	All drives
D	Distance (DI)	020	Long	counts	All drives
E	Position Offset	021	Long	counts	Drives with encoder feedback option

The “E” register contains the difference between the encoder count and the motor position. This value is most useful with servo drives (Blu / SV) where the resolution of the motor and encoder are the same, and this offset can be useful when working with absolute positions. The register contains the difference in counts between the “e” register and the value set by the “SP” command.

F	Other Flags	022	Long	bit pattern (see below)	All drives

BLu

The value of the “F” register is a hexadecimal sum of various drive states, as shown below.

Description	Hex Value	Decimal Value
DISTANCE LIMIT FLAG	0x0001	1
SENSOR FOUND FLAG	0x0002	2
LOWSIDE OVERCURRENT	0x0004	4
HIGHSIDE OVERCURRENT	0x0008	8

Clear flags by sending “RLF0” to the drive.

SV

The value of the “F” register is a hexadecimal sum of various drive states, as shown below.

Description	Hex Value	Decimal Value
DISTANCE LIMIT FLAG	0x0001	1
SENSOR FOUND FLAG	0x0002	2
LOWSIDE OVERCURRENT	0x0004	4
HIGHSIDE OVERCURRENT	0x0008	8
OVER CURRENT READING	0x0010	16
BAD CURRENT OFFSET - Phase A	0x0020	32
BAD CURRENT OFFSET - Phase B	0x0040	64
BAD FLASH ERASE	0x4000	16384
BAD FLASH SAVE	0x8000	32768

Clear flags by sending “RLF0” to the drive.

STAC6

The value of the “F” register is a hexadecimal sum of various drive states, as shown below.

Description	Hex Value	Decimal Value
DISTANCE LIMIT FLAG	0x0001	1
SENSOR FOUND FLAG	0x0002	2
HARDWARE OVERCURRENT	0x0004	4
SOFTWARE OVERCURRENT	0x0008	8
BAD CURRENT OFFSET - Phase A	0x0010	16
BAD CURRENT OFFSET - Phase B	0x0020	32
OPEN WINDING - Phase A	0x0040	64
OPEN WINDING - Phase B	0x0080	128

Clear flags by sending “RLF0” to the drive.

ST-Q/Si, ST-S, STM

The value of the “F” register is a hexadecimal sum of various drive states, as shown below.

Description	Hex Value	Decimal Value
DISTANCE LIMIT FLAG	0x0001	1
SENSOR FOUND FLAG	0x0002	2
LOWSIDE OVERCURRENT	0x0004	4
HIGHSIDE OVERCURRENT	0x0008	8
OVER CURRENT READING	0x0010	16
BAD CURRENT OFFSET - Phase A	0x0020	32
BAD CURRENT OFFSET - Phase B	0x0040	64
OPEN WINDING - Phase A	0x0080	128
OPEN WINDING - Phase B	0x0100	256
LOGIC SUPPLY	0x0200	512
GATE SUPPLY	0x0400	1024
BAD FLASH ERASE	0x4000	16384
BAD FLASH SAVE	0x8000	32768

Clear flags by sending “RLF0” to the drive.

G	Current Command (GC)	023	Short	0.01 Amps	Servo drives only
H	Analog Velocity Gain	024	Short	+/- 32767 ADC counts	BLu servo drives only

The “H” register in BLu servo drives is similar to the AG command in all other drives. The “H” register is used to set the motor speed at a given DC voltage in analog velocity mode. It is recommended to make this setting in *Quick Tuner*, where it is labeled Speed in rev/sec at xx Volts, under the Velocity > Analog Operating Mode.

I	Input Counter	025	Long	counts per edge	Q drives only

J	Jog Velocity (JS)	026	Short	0.25 rpm	All drives

The “J” register units are 0.25 rpm, which means that the value of the “J” register is equal to 240 times the JS command value. In other words, to achieve a jog speed value of 7 rev/sec send the command RLJ1680.

K	RESERVED	027	-	-	-
L	RESERVED	028	-	-	-
M	Max Velocity (VM, servo) Accel/Decel Current (CA, STM Integrated Stepper)	029	Short	Servo: 0.01 amps RMS Stepper: 0.01 amps, peak-of-sine	Servo drives and STM Integrated Steppers
N	Continuous Current (CC, servo) Running Current (CC, stepper)	030	Short	Servo: 0.01 amps RMS Stepper: 0.01 amps, peak-of-sine	All drives
O	Peak Current (CP, servo) Idle Current (CI, stepper)	031	Short	Servo: 0.01 Amps RMS Stepper: 0.01 amps, peak-of-sine	All drives
P	Absolute Position Command	032	Long	counts	All drives
Q	RESERVED	033	-	-	-
R	Steps per Rev*	034	Short	counts	All drives

* Note: R = EG for servo drives. R = EG/2 for stepper drives.

S	Pulse Counter	035	Long	counts	All drives

The “S” register counts pulses coming into the STEP/X1 and DIR/X2 inputs of the drive. This is particularly useful when in Command Mode 7 (see CM command) or executing an FE (Follow Encoder) command. To zero the “S” register send the command RLS0.

T	Total Count	036	Long	(see below)	Q drives only

The “T” register is automatically saved at power down and restored at power up.

U	Change Velocity (VC)	037	Short	0.25 rpm	All drives

The “U” register units are 0.25 rpm, which means that the value of the “U” register is equal to 240 times the VC command value. In other words, to achieve a change velocity value of 7 rev/sec send the command RLU1680.

V	Velocity (VE)	038	Short	0.25 rpm	All drives

The “V” register units are 0.25 rpm, which means that the value of the “V” register is equal to 240 times the VE command value. In other words, to achieve a velocity value of 7 rev/sec send the command RLV1680.

W	Time Stamp	039	Short	0.001 sec	Q drives only

X	Analog Position Gain (AP)	040	Short	Servo: ADC counts/encoder count Stepper: ADC counts/step	All drives
Y	Analog Threshold (AT)	041	Short	raw ADC counts	All drives
Z	Analog Offset (AV)	042	Short	raw ADC counts	All drives

User-Defined data registers: 0 - 9, other characters

Ch.	Description	3-digit	Data Type	Units	Compatibility
0	Accumulator	000	Long	integer	Q drives only

The Accumulator register “0” is, aside from being a User-defined data register, the register in which the result of every register math function is placed. For example, if the drive executes the register addition command “R+D1” the result of this operation (i.e. the sum of the values in data registers “D” and “1”) will be placed in the Accumulator “0” register.

1	User-defined	001	Long	integer	Q drives only
2	User-defined	002	Long	integer	Q drives only
3	User-defined	003	Long	integer	Q drives only
4	User-defined	004	Long	integer	Q drives only
5	User-defined	005	Long	integer	Q drives only
6	User-defined	006	Long	integer	Q drives only
7	User-defined	007	Long	integer	Q drives only
8	User-defined	008	Long	integer	Q drives only
9	User-defined	009	Long	integer	Q drives only
:	User-defined	010	Long	integer	Q drives only
;	User-defined	011	Long	integer	Q drives only
<	User-defined	012	Long	integer	Q drives only
=	User-defined	013	Long	integer	Q drives only
>	User-defined	014	Long	integer	Q drives only
?	User-defined	015	Long	integer	Q drives only
@	User-defined	016	Long	integer	Q drives only
[	RESERVED	043	-	-	-
\	RESERVED	044	-	-	-
]	RESERVED	045	-	-	-
^	RESERVED	046	-	-	-
_	RESERVED	047	-	-	-
`	RESERVED	048	-	-	-

Appendices

The following appendices detail various special topics in working with Applied Motion motor drives.

Appendix A: Non-Volatile Memory in Q drives

The non-volatile memory in Q drives is partitioned into 16 sections. The partitions are dedicated to various elements of a Q drive's data, and are designated as follows:

Partition 1..... Q Program Segment 1
Partition 2..... Segment 2
Partition 3..... Segment 3
Partition 4..... Segment 4
Partition 5..... Segment 5
Partition 6..... Segment 6
Partition 7..... Segment 7
Partition 8..... Segment 8
Partition 9..... Segment 9
Partition 10..... Segment 10
Partition 11..... Segment 11
Partition 12..... Segment 12
Partition 13..... Drive Parameters
Partition 14..... Alarm History
Partition 15..... NV Data Register Storage Locations 1-100
Partition 16..... RESERVED

The separation of these partitions is important in understanding how the drive writes to non-volatile memory. For example, each time the SA command is executed by the drive, all of the Drive Parameters are re-written to non-volatile memory partition 13. Similarly, each time an RW command is executed by the drive, all of the one hundred NV Data Register Storage Locations are re-written in partition 15, even if only one of the locations is being updated with a new data register value.

The significance of these operations becomes clear when we consider that the physical non-volatile memory of the Q drive is limited to approximately 10,000 write cycles. This means that after writing to any one of the 16 partitions 10,000 times, the integrity of the data stored in that memory partition cannot be insured.

For this reason, it is not recommended to use the RW or SA commands in stored Q programs. For example, it might be tempting for a user to include an RW command or two in a stored program in such a manner that allows for various data register values to be written to non-volatile memory on a regular basis. The temptation of this is that there won't be a need to reload register values manually in the case of a power down/up cycle: the register values can simply be loaded back into the program (using RR commands) from non-volatile memory. This is to be avoided, though, because using the RW command (or SA command) in this manner could result in the early failure of the non-volatile memory of the drive. The intended use of the RW command therefore is to be used in the early stages of an application, during startup and programming, to set up a series of non-volatile register locations that can be read into a stored program using the RR command.

The partitions designated for Q Program Segment storage are typically not going to be re-written in a manner similar to the RW and SA commands, as they are only accessed during program/segment downloads during startup and programming of an application.

Appendix B: Host Serial Communications

When a drive is operating in “host mode”, it means that a host device sends commands to the drive (or drives) over a serial connection (or network) and the drive executes the incoming commands. Here are some examples of typical host devices:

- A Windows-based PC running Applied Motion software
- An industrial PC running a custom-built or other proprietary software application
- A PLC with an ASCII module/serial port for sending text strings
- An HMI with a serial connection for sending text strings

The aim of this appendix is to describe the following aspects of operating an Applied Motion Products motor drive in host mode.

- General structure of host serial communications.
- Hardware – wiring and connecting a host device to the serial ports of an Applied Motion drive. (Covered in detail in Appendix C).
- COM Port Settings – UART settings and Bit Rate (Baud) settings.
- Communications Protocol
- Communication Details
- Communication Errors

General structure of host serial communications

Applied Motion’s host serial communications are based on the common ASCII character set transmitted using standard UARTs over an RS-232 or RS-485 hardware interface.

The ASCII character set is used because it is common and well-understood, as well as easy to read. UART (Universal Asynchronous Receiver Transmitter) serial transceivers are available on many types of equipment, including most PCs, and provide a common form of serial communications interface. RS-232 and RS-485 hardware connections are commonly used with UARTs and also provide the easiest and most common form of connectivity.

Hardware

Details on drive terminals and connectors for wiring each of the available hardware configurations are shown in Appendix C. Below is an overview of the three available configurations.

RS-232: This is the easiest method for drive serial communications. Using an Applied Motion supplied adapter/programming cable (one supplied with each Applied Motion drive) a single drive can be connected directly to any PC with a standard 9-pin RS-232 serial port. Here are some RS-232 highlights:

- Easiest to use
- Configuration of choice for using Applied Motion software applications such as Q Programmer, Quick Tuner and STAC6 Configurator
- Short Cable Lengths
- Serial cable provided with each Applied Motion drive
- Susceptible to EMI

RS-422 (4-wire RS-485): RS-422 was originally designed for high reliability communications in point-to-point configurations. It usually requires a special adapter to work with a PC but is common on many types of controllers such as PLCs and HMIs. Our implementation allows for multi-drop communications with a single master (serial network). Here are some RS-422 highlights:

- Relatively easy to use
- NOT supported by Applied Motion software applications such as Quick Tuner or STAC6 Configurator. (Q

Programmer does support RS-422 in a limited fashion).

- Permits longer cable Lengths
- May require special adaptor
- Immune to EMI (when wired properly)

RS-485 (2-wire RS-485): Designed for multi-drop serial networks, provides simple wiring, high reliability, and long cable lengths. Here are some RS-485 highlights:

- More difficult to use
- NOT supported by Applied Motion software applications such as Quick Tuner or Configurator. (Q Programmer does support RS-485 in a limited fashion).
- Permits longest cable lengths: up to 1000 feet at low baud rates
- May require special adaptor
- Fewest wires, smaller cables
- Immune to EMI (when wired properly)

COM Port Settings

UART Settings: We operate our UARTs with the following settings: 1 start bit, 8 data bits, 0 (no) parity bits, and 1 stop bit.

Bit rate (baud) Settings: (BR and PB commands): All AMP drives default to 9600 baud from the factory. In most cases this speed is adequate for setup, configuring, programming, as well as host mode communications. If higher baud rates are required the drives can be configured to operate with a different rate using the BR (Bit rate) or PB (Power-up Bit rate) command. In all cases the drive starts up at the factory rate, 9600, and will remain there if the “power-up packet” is acknowledged by the host (see “Drive Startup” below). When the power-up cycle is complete and if the drive has not received the power-up packet, the drive will activate the new baud rate.

Selecting a baud rate higher than the default 9600 is dependent on the application. If there is a host device operating a number of drives on a network, a higher speed may be required in order to process all the communication needs.

Communications Protocol

In general, the protocol for communications between a host device and a drive is quite simple. The drives do not initiate communications on their own, so drives are normally in a state to receive packets from the host. A communications packet, or packet for short, includes all the characters required to complete a command (host to drive) or response (drive to host) transmission. In other words, a host initiates communication by sending a command packet, and the drive responds to that command (if necessary) by sending a response packet back to the host.

Command Transmission (host to drive): The transmission of characters to the drive requires the host to send all the required characters that form a packet in a limited time frame. At the start of receiving a packet, the drive begins timing the space between characters. Each time a character is received an internal timer is reset to 200 milliseconds. If the timer reaches zero before the next character in the packet is received the drive will terminate its packet parsing (characters will still go into the receive buffer) and may send out an error response packet depending on the protocol setting. The purpose of the time-out feature is to allow the drive to purge its buffers automatically when a bad transmission occurs.

NOTE: This time-out feature limits the usage of host devices such as the Windows application HyperTerminal. We recommend using Applied Motion’s SCL Setup Utility instead. This utility sends out an entire command packet with the minimum delay between characters, and includes the packet’s terminating character (carriage return).

Command packets are terminated by a Carriage Return (ASCII 13).

Response Transmission (drive to host): In response to a command packet from the host a drive can send a response packet. The drive sends out its entire response packet with very limited space between characters.

At 9600 baud the space between characters is less then 1 bit space (0.0001 seconds). The host system must be able to handle this speed. The space between characters can vary depending on the settings of the PR command (see below).

Response packets are terminated by a Carriage Return (ASCII 13).

Protocol Settings (PR Command): The PR (Protocol) command offers users the ability to add various features to the overall communications protocol, i.e. tailor the structure of command and response packets to best fit the needs of the application. In general, when a host device sends a command packet to a drive, the drive will either understand the command or not. If the drive understands the command the drive executes the command. If the drive doesn't understand the command it cannot execute the command. In most cases the host device will want to know whether the drive has understood the command or not, and so the drive can be set to automatically send an Acknowledge (understood) or Negative Acknowledge (not understand) response packet to the host for every command packet received.

Along with Acknowledge/Negative Acknowledge (Ack/Nack), the PR command controls a number of other protocol settings. See Appendix D for details on the PR command. Also, the PR command controls whether or not the drive will respond with error codes in the response packet when communications errors occur.

Communication Details

Transmit Delay: (TD Command): The TD command allows users to define a dwell time in a drive, which is used by the drive to delay the start of transmission of a response packet after the end of reception of a command packet.

When using 2-wire RS-485 networks there are times when a drive's response packet must be delayed until the network is ready for the drive to transmit. Why is this necessary? The answer is because RS-485 networks are by nature "half-duplex", which means you cannot transmit and receive at the same time. Rather, a host must first transmit, stop, then wait to receive. This is because the host and drive transmitters share the same pair of wires. When transmitting, the device that has the transmission rights must assert its transmitter outputs and therefore take control of the pair. At the same time all other devices on the network must de-assert, or open, their transmitters so as not to interfere with the device that has the rights. Transmitters in this scenario have tri-state outputs: the three states are transmit, open, and receive.

Some devices are not as quick in opening their transmitters as others. For this reason it may be necessary for other, faster devices on the network to dwell some time while the slower devices open their transmitters. Applied Motion drives de-assert their transmitters very quickly. Typically it is done within 100 microseconds (.0001 second) after the end of a packet transmission. However it is possible that the host device won't be this fast, and so the TD command allows users to set the time delay that an Applied Motion drive will delay after receiving a command packet before sending a response packet.

Communications Packet: A Communications Packet, or **packet** for short, includes all the characters required to complete a command or response transmission. This can vary depending on the settings of the PR command. See Appendix D for more on the PR command. All packets are terminated by a Carriage Return (ASCII 13).

Drive Startup: At power-up, all Applied Motion drives send out what is called the "power-up packet". This packet notifies a host of the drive's presence. After sending the power-up packet the drive waits for a response from the host. This is one of the rare instances in which a drive will initiate communications with the host. This process is necessary for a number of Applied Motion software applications such as Quick Tuner and STAC6 Configurator. The power-up packet is an exception to the ASCII character rule in that all the characters in the packet are binary value. Even if the character is printable its binary value is what is important. The power-up packet consists of three binary characters with the first character being a binary 255 (255 is not a printable ASCII character). This character designates to the software application that the packet is a power-up packet. The following two characters are the firmware version number and the model number of the drive, respectively.

Power-Up Packet = (255)(F/W Version)(Model No.)

As an example, a BLuAC5-Si with f/w version 1.53 firmware will send out a power-up packet that looks like this: (255)(53)(38). To an ASCII terminal this packet may look like "ÿ5&". The (255) is the power-up packet designator, the (53) actually stands for f/w version 1.53 (the "1" is implied), and the (38) is an internal model

number for the “BLuAC5-Si”

The power-up packet is always sent at 9600 baud, regardless of the bit rate set by the BR or PB command. If an Applied Motion software application is present it will respond to the power-up packet and communications will continue at 9600 baud. If an Applied Motion software application is not present, the drive’s request made by the power-up packet will time-out and the drive will begin communicating at the saved bit rate (BR or PB command), 9600 or otherwise.

Interaction with PM parameter (Power-up Mode): If the drive is currently in power-up modes 1 or 3 (PM1 or PM3), it will be unable to respond to standard SCL commands. In these modes the drive is using a proprietary communication protocol used by Si Programmer (and its interface to the SiNet Hub units) as well as the QuickTuner and Configurator software programs. Standard SCL commands will not be recognized or acted upon by the drive in these modes. If the application requires it, the drive may be temporarily forced into SCL mode through the use of the “double zero.”

Double Zero: When the drive initializes, it will send the power-up packet as detailed above. Typically this packet is used only by Applied Motion Products software, but a host device may also use it to force SCL communication in a drive otherwise not configured to do so.

The host device must recognize the power-up packet and respond with a simple double zero (00). No carriage return is required. Note that this response must occur within 2 seconds of the power-up packet being sent, but must delay at least 2 milliseconds (0.002 sec). This will force the drive into standard SCL mode and enable serial communication without altering the PM setting of the drive.

Communication Errors

During the process of sending communication packets between the host and drive(s), two different types of communication errors can occur.

Hardware errors: Hardware errors are displayed physically by a drive (via either LEDs or a 7-segment display on the drive, see Appendix F), but no response packet is automatically generated from the drive to the host. Therefore it is the responsibility of the host to check for hardware comm errors using the AL, RS, and/or SC commands. See Appendix F for more details on the AL and SC commands. Once the host has determined the presence of a hardware comm error, the nature of the error can be retrieved using the CE command.

Parsing errors: Parsing errors happen when a drive receives a command packet but cannot properly interpret (parse) the command. Parsing errors can automatically generate a response packet from the drive to the host, depending on the settings of the PR command (see Appendix D, PR command, Bit 2).

Appendix C: Host Serial Connections

Introduction

When communicating to a drive over its serial port you will always be using one of the following serial connections: RS-232, 2-wire RS-485, or 4-wire RS-485. Out of the box we suggest starting with RS-232 along with the programming cable and software that was supplied with your Q drive, so that you may be communicating to and familiarizing yourself with your drive as quickly as possible. All software from Applied Motion communicate to a drive via the supplied RS-232 programming cable. These software include:

Quick Tuner ----------used for tuning and configuring servo drives
Configurator----------used for configuring your stepper drives
Q Programmer ----------create and edit stored Q programs, emulate a host
SCL Setup Utility ----------basic host terminal for host emulation

If your project calls for a Q drive (or drives) running stored programs, you will use the supplied RS-232 programming cable along with Quick Tuner or Configurator and Q Programmer to setup, configure, and program your drive(s). If your project calls for your drive(s) only running stored programs, you can read up on the RS-232 sub-section in this section and not read any more about the other serial connections. However, if your application calls for a serial host controller (PC, PLC, HMI, or other serial device that can act as a host) being able to communicate to the drive(s), you will need to choose one of the three available serial connections.

Available Host Serial Connections: RS-232, 2-wire RS-485, 4-wire RS-485

When choosing the best serial connection for your project, the choice may be made for you based on the host controller you plan to use. For example, some devices only communicate via 2-wire RS-485. If you are not restricted by your host controller, here are two guidelines for choosing the best connection.

Single or multi-axis

If your project calls for communicating to only one drive you can consider any of the three options. If your project calls for communicating to more than one drive you should use 2-wire or 4-wire RS-485.

Long communication cables

In many applications, the limitation of 50 feet on RS-232 will be sufficient. In applications where the distance between drive and host controller will be more than 50 feet (up to 1000 feet), you will need to choose 2-wire or 4-wire RS-485.

A Quick Summary of 2-wire and 4-wire RS-485 connections

The 2-wire and 4-wire RS-485 protocols that the drives utilize are based on industry standard RS-485 and RS-422 protocols. Strictly defined, RS-485 is a 2-wire interface that allows multi-node connections limited to half-duplex serial communications. Up to 32 nodes that both transmit and receive can be connected to one network. On the other hand, RS-422 in the strictest definition is a 4-wire point-to-point connection that allows full-duplex serial communications when connected to a single node. RS-422 has one node that is the driver or transmitter and up to 10 nodes that are receivers. RS-422 was not designed for a true multi-node network.

2-wire interfaces require one more significant feature. A network node, master or slave, must be able to tri-state its transmitter to allow other nodes to use the network when required. For high speed baud rates this must be done very quickly to avoid communication collisions.

4-wire interfaces can go beyond simple point-to-point communications and be used in multi-node networks if the slave nodes are capable of tri-stating their transmitters as required in the 2-wire networks. Some RS-485 devices (like Applied Motion drives) are set up to do this and can be used in a 4-wire, multi-node configuration.

The drives are designed to work in a multi-node environment, and so they use both the standard 2-wire RS-485 connection, and a modified RS-422 (4-wire) connection that has been termed “4-wire RS-485”. This is because unlike the standard RS-422, which is designed for single-node connections, the 4-wire RS-485 used by

Applied Motion drives allows multiple nodes.

NOTE: In general we recommend using half-duplex communications with the drives. Even though the 4-wire RS-485 network can support full-duplex, there is now the capability to have multiple nodes and therefore data collisions might occur. For this reason we recommend limiting communications to half-duplex, even with the 4-wire RS-485 connections.

Connecting to your Q drive's serial port(s)

Each drive comes with one or two physical connectors for connecting to a PC or other serial host controller device. One connector is an RJ11 connector (same as a 4-wire phone jack) that is used strictly for RS-232 communications. The second connector is a removable 5-position terminal block for use with 2-wire and 4-wire RS-485 connections.

COM Port Settings

When using software from Applied Motion Products to communicate to a drive there is no need to worry about COM port settings because the software will take care of them. In applications where a host serial controller will be communicating to a drive via one of it's serial ports, the COM port settings should be set as follows: 8 data bits, no Parity, 1 stop bit. The default Baud rate is 9600, though this can be changed (see BR and PB commands).

Connecting to a PC using RS-232

Each drive comes with a programming cable for use with the drive's RS-232 port. This cable is made up of two parts, a 7 foot 4-wire cable (looks just like a 7 foot telephone cord), and an RJ11 to 9-pin DSUB adapter. This adapter allows you to connect to the COM port (serial port) of your PC. Here are the general directions for connecting your drive to your computer.

- Locate your computer within 6 feet of the drive.
- Plug the 9-pin end of the adapter supplied with your drive to the COM1 serial port of your PC. Secure the adapter with adapter's two screws. If the COM1 port on your PC is already used by something else, you may use the COM2 port of your PC. On some PCs, COM2 will have a 25-pin connector rather than a 9-pin. If this is the case with your PC, and you must use COM2, you will have to purchase a 25 to 9 pin serial adapter at your local computer store.

NOTE: If you are using a laptop computer that does not have any COM ports, you will have to use either a USB to Serial adapter or a PCMCIA Serial adapter. There are a variety on the market, and some work better than others. But in general, once you've installed one of the adapters your PC will assign the adapter a COM port number. Remember this number when you go to use your Applied Motion software. Also, if you are having troubles with your adapter, contact Applied Motion for help with recommended adapters.

- Now take the 7 foot cable and plug one end into the adapter you just attached to your PCs COM port, and plug the other end into the RS-232 (RJ11) jack on the drive. If you need to locate your drive farther from the PC, you can replace the 7 foot cable with any 4-wire telephone cord. Do not exceed 50 feet.

WARNING: Never connect an Applied Motion Products drive to a telephone circuit. It uses the same connectors and cords as telephones and modems, but the voltages are not compatible.

Connecting to a host using 4-wire RS-485

An Applied Motion drive's 4-wire RS-485 implementation is a multi-drop network with separate transmit and receive wires. One pair of wires connects the host's TX+ and TX- signals to each drive's RX+ and RX- terminals. Another pair connects the RX+ and RX- signals of the host to the TX+ and TX- terminals of each drive. A common ground terminal is provided on each drive and can be used to keep all drives at the same ground potential. This terminal connects internally to a drive's ground connection, so if all the drives on the 4-wire network are powered form the same supply it is not necessary to connect the logic grounds. You should still connect one drive's GND terminal to the host's signal ground. Before wiring the entire system you'll need to connect each drive individually to the host so that a unique address can be assigned to each drive. (See following sub-section "Before you connect the drive to your system"). Proceed as follows, using the figure below.