9. Standalone Language Specification

Important Note: All the commands described in this section are standalone language commands and are not analogous to ASCII commands. Refer to section 9 for details regarding ASCII commands.

9.1. Standalone Command Set

Command	R/W	Description	Example
;	-	Comment notation. Comments out any text following ; in the same line.	;This is a comment
ABORT	W	Immediately stop all motion for all axis.	ABORT
ABORT[axis]	W	Immediately stop all motion for a single axis	ABORTX ABORTZ
ABS	W	Set the move mode to absolute mode.	ABS X1000 ;move to position 1000

ACC	R/W	Set/get the global acceleration setting. Unit is in milliseconds.	ACC=500
			ACC=V1

ACC[axis]	R/W	Set/get the individual acceleration setting. Unit is in milliseconds.	ACCX=500
			ACCY=V1

AI[1-8]	R	Get the analog input value.	IF AI2>2500
			  DO=1
			ENDIF
			V2=AI1

ARCP[X]:[Y]:[θ]	W	Perform ARC move in CW direction using the X and Y axis.	;move CW to angle 90 degrees ARCP1000:0:90000
ARCN[X]:[Y]:[θ]	W	Perform ARC move in CCW direction using the X and Y axis. [X] and [Y] represent the arc center.	;move CCW to angle 90 degrees ARCN1000:0:90000
BUFOFF	W	Turn the buffer move mode setting off.	BUFOFF
BUFON	W	Turn the buffer move mode setting on.	BUFON
CIRP[X]:[Y]	W	Perform circle move in CW direction using the X and Y axis. [X] and [Y] represent the circle center.	;Perform CW circle around (0,0) CIRP0:0
CIRN[X]:[Y]	W	Perform circle move in CCW direction using the X and Y axis. [X] and [Y] represent the circle center.	; Perform CCW circle around (0,0) CIRN0:0

DEC	R/W	Set/get the global deceleration setting. Unit is in milliseconds.	DEC=500
			DEC=V1

DEC[axis]	R/W	Set/get the individual deceleration setting. Unit is in milliseconds.	DECX=500
			DECY=V1

DELAY	W	Set a delay in milliseconds. Assigned value is a 32-bit unsigned integer or a variable.	DELAY=1000 ;1 second
			DELAY=V1 ;assign to variable

DI	R	Return status of digital inputs. See Table 6.3 for bitwise assignment.	IF DI=0
			  DO=1 ;Turn on DO1
			ENDIF
			V2=DI

DI[1-8]	R	Get individual bit status of digital inputs. Will return [0,1]. See Table 6.3 for bitwise assignment.	IF DI1=0
			  DO=1 ;Turn on DO1
			ENDIF
			V3=DI1

DO	R/W	Set/get digital output status. See Table 6.4 for bitwise assignment.	DO=2 ;Turn on DO2

DO[1-8]	R/W	Set/get individual bit status of digital outputs. Range for the bit assigned digital outputs is [0,1].	DO2=1 ;Turn on DO2
ECLEAR[axis]	W	Clear any motor status errors.	ECLEARX
EO	R/W	Set/get the enable output status. Refer to Table 6.1	EO=3 ;Enable the X and Y motor
EO[1-4]	R/W	Set/get individual bit status of the enable outputs.	EO3=1 ;Enable the Z motor

E[axis]	R/W	Set/get the current encoder position.	EX=1000 ;Set to X enc to 1000
			V1=EU ;Read current U encoder

GOSUB [0-31]	-	Call a subroutine that has been previously stored to flash memory.	GOSUB 0 END
HLHOME[axis][+/-]	W	Home the motor using the home input at low and high speeds in the specified direction. See section 6.13.5 for details.	HLHOMEX+ ;positive X home WAITX ;wait for X home move
HOME[axis][+/-]	W	Home the motor using the home input at high speed in specified direction. See section 6.13.1 for details.	HOMEX- ;negative X home WAITX ;wait for X home move

HSPD	R/W	Set/get the global high speed setting. Unit is in pulses/second.	HSPD=1000
			HSPD=V1

HSPD[axis]	R/W	Set/get the individual high speed setting. Unit is in pulses/second.	HSPDY=1000
			HSPDZ=V1

IF
ELSEIF
ELSE
ENDIF
	-	Perform a standard IF/ELSEIF/ELSE conditional. Any command with read ability can be used in a conditional.
		ENDIF should be used to close off an IF statement.
		Conditions [=, >, <, >=, <=, !=] are available
			IF DI1=0
			DO=1 ;Turn on DO1
			ELSEIF DI2=0
			DO=2; Turn on DO2
			ELSE
			DO=0; Turn off DO
			ENDIF

INC	W	Set the move mode to incremental mode.	INC X1000 ;increment by 1000
JOG[axis][+/-]	W	Move the motor indefinitely in the specified direction.	JOGX+
JOYENA	W	Set the joystick enable setting. See section 6.20 for details.	JOYENA=3 ;enable joystick X,Y

JOYHS[axis]	W	Set the high speed setting for joystick control. See section 6.20 for details.	JOYHSX=2000
			JOYHSZ=5000

JOYDEL[axis]	W	Set the speed change delta for joystick control. See section 6.20 for details.	JOYDELZ=100
			JOYDELU=200

LHOME[axis][+/-]	W	Home the motor using the limit inputs in the specified directions. See section 6.13.2 for details.	LHOMEX+ ;positive home WAITX

LSPD	R/W	Set/get the global low speed setting. Unit is in pulses/second.	LSPD=100
			LSPD=V3

LSPD[axis]	R/W	Set/get the individual low speed setting. Unit is in pulses/second.	LSPDX=100
			LSPDY=V1

MST[axis]	R	Get the current motor status of the motor of an axis. See Table 6.2 for motor status assignment.	
PRG [0-3] END	-	Used to define the beginning and end of a main program. Four standalone programs are available	PRG 0 ;main program END

PSX	R	Get the current motor speed.	V2=PSX ;Set V2 to speed

P[axis]	R/W	Set/get the current motor position.	PX=1000 ;Set to X pos to 1000
			V1=PY ;Read current Y position

SCV[axis]	R/W	Set/get the s-curve enable setting.	SCVX=1 ;enable X s-curve
			V1=SCVY ;read Y s-curve

SLS[axis]	R	Get the current StepNLoop status. See Table 6.14 for details.	V3=SLSX ;Set to status

SL[axis]	W	Enable/disable StepNLoop closed loop mode.	SLX=1 ;Enable StepNLoop
			SLX=0 ;Disable StepNLoop

SR[0-3]	W	Set the standalone control for the specified program. See Table 6.16.	SR0=0 ;Turn off program 0

SSPDM[axis]	W	Set the SSPD mode. Must be done before move command. See Table A.0 for details.	SSPDMX=1 ;Set SSPD mode
			SSPDMX=V2
			JOGX+ ;Jog the motor

SSPD[axis]	W	Perform an on-the-fly speed change. SSPDM[mode] must be set first.	JOGX+ ;Jog the motor
			DELAY=1000 ;Wait 1 second
			SSPDX=1000 ;Change speed
			SSPDX=V1

STOP	W	Stop all motion using a decelerated stop.	
STOP[axis]	W	Stop motion using a decelerated stop for an individual axis.	STOPX STOPY
STORE	W	Store settings to flash.	STORE

SUB [0-31]
ENDSUB
	-	Defines the beginning of a subroutine. ENDSUB should be used to define the end of the subroutine.	SUB 1
			DO=4
			ENDSUB

SYNCFG[axis]	W	Set the sync output configuration. See Table 6.8 for details.	SYNCFGX=2 ;Equal to setting
SYNOFF[axis]	W	Disable the sync output configuration.	SYNOFFX ;Turn off sync
SYNON[axis]	W	Enable the sync output configuration.	SYNONX ;Turn on sync

SYNPOS[axis]	W	Set the sync output reference position.	SYNPOSX=1000 ;Set position
			SYNPOSX=V1 ;Set position

SYNSTAT[axis]	R	Get the current sync output status. See Table 6.9 for details.	V1=SYNSTATX ;Get status
SYNTIME[axis]	W	Set the sync output pulse width time in milliseconds. Maximum of 10 ms.	SYNTIMEX=5 ;set to 5 ms
TOC	W	Sets the communication time-out parameter. Units is in milliseconds.	TOC=1000 ;1 second time-out

TR	R/W	Set/get the timer register.	TR=1000
			IF TR<500
			DO=1
			ENDIF

U[position]	W	If in absolute mode, move the U motor to [position]. If in incremental mode, move the motor to [current position] + [position].	U1000

V[0-99]	R/W	Set/get standalone variables.
		
		The following operations are available:
		[+] Addition
		[-] Subtraction
		[*] Multiplication
		[/] Division
		[%] Modulus
		[>>] Bit shift right
		[<<] Bit shift left
			V1=12345 ;Set V1 to 12345
			V2=V1+1;Set V2 to V1 + 1
			V3=DI ;Set V3 to DI
			V4=DO ;Set V4 to DO V5=~EO
			;Set V5 to NOT EO

		[&] Bitwise AND [|] Bitwise OR [~] Bitwise NOT	
WAIT[axis]	W	Wait for current motion to complete before processing the next line.	X1000 ;move to position 1000 WAITX ;wait for move

WHILE
ENDWHILE
	-	Perform a standard WHILE loop within the standalone program. ENDWHILE should be used to close off a WHILE loop.
		Conditions [=, >, <, >=, <=, !=] are available.
			WHILE 1=1 ;Forever loop
			DO=1 ;Turn on DO1
			DO=0 ;Turn off DO1
			ENDWHILE

X[position]	W	If in absolute mode, move the X motor to [position]. If in incremental mode, move the motor to [current position] + [position].	X1000
X[pos]Y[pos] Z[pos]U[pos]	W	Perform linear interpolated move. This command requires 2 or more axis. If only 1 axis is used, a standard positional move will be performed. If buffer mode is enabled, the move command will be added to the buffer automatically.	X1000Y2000 X1000Y2000U4000 X1000Z3000
Y[position]	W	If in absolute mode, move the Y motor to [position]. If in incremental mode, move the motor to [current position] + [position].	Y1000
ZHOME[axis][+/-]	W	Home the motor using the home input and Z-index. See section 6.13.3 for details.	ZHOMEX+ ;positive X home WAITX
ZOME[axis][+/-]	W	Home the motor using the Z-index only. See section 6.13.4 for details.	ZOMEX- ;negative X home WAITX
Z[position]	W	If in absolute mode, move the Z motor to [position]. If in incremental mode, move the motor to [current position] + [position].	Z1000

Table 9.0

9.2. Example Standalone Programs

9.2.1. Standalone Example Program 1 – Single Thread

Task: Set the high speed and low speed and move the motor to 1000 and back to 0.

```
HSPD=20000        ;* Set the high speed to 20000 pulses/sec
LSPD=1000         ;* Set the low speed to 1000 pulses/sec
ACC=300           ;* Set the acceleration to 300 msec
EO=1              ;* Enable the motor power
X1000             ;* Move to 1000
WAITX             ;* Wait for X-axis move to complete
X0                ;* Move to zero
WAITX             ;* Wait for X-axis move to complete
END               ;* End of the program
```

9.2.2. Standalone Example Program 2 – Single Thread

Task: Move the motor back and forth indefinitely between position 1000 and 0.

```
HSPD=20000          ;* Set the high speed to 20000 pulses/sec
LSPD=1000           ;* Set the low speed to 1000 pulses/sec
ACC=300             ;* Set the acceleration to 300 msec
EO=1                ;* Enable the motor power
WHILE 1=1           ;* Forever loop
      X0            ;* Move to zero
      WAITX         ;* Wait for X-axis move to complete
      X1000         ;* Move to 1000
      WAITX         ;* Wait for X-axis move to complete
ENDWHILE            ;* Go back to WHILE statement
END
```

9.2.3. Standalone Example Program 3 – Single Thread

Task: Move the motor back and forth 10 times between position 1000 and 0.

```
HSPD=20000          ;* Set the high speed to 20000 pulses/sec
LSPD=1000           ;* Set the low speed to 1000 pulses/sec
ACC=300             ;* Set the acceleration to 300 msec
EO=1                ;* Enable the motor power
V1=0                ;* Set variable 1 to value 0
WHILE V1<10         ;* Loop while variable 1 is less than 10
      X0            ;* Move to zero
      WAITX         ;* Wait for X-axis move to complete
      X1000         ;* Move to 1000
      WAITX         ;* Wait for X-axis move to complete
      V1=V1+1       ;* Increment variable 1
ENDWHILE            ;* Go back to WHILE statement
END
```

9.2.4. Standalone Example Program 4 – Single Thread

Task: Move the motor back and forth between position 1000 and 0 only if the digital input 1 is turned on.

```
HSPD=20000                  ;* Set the high speed to 20000 pulses/sec
LSPD=1000                   ;* Set the low speed to 1000 pulses/sec
ACC=300                     ;* Set the acceleration to 300 msec
EO=1                        ;* Enable the motor power
WHILE 1=1                   ;* Forever loop
    IF DI1=1                ;* If digital input 1 is on, execute the statements
        X0                  ;* Move to zero
        WAITX               ;* Wait for X-axis move to complete
        X1000               ;* Move to 1000
        WAITX               ;* Wait for X-axis move to complete
    ENDIF
ENDWHILE                    ;* Go back to WHILE statement
END
```

9.2.5. Standalone Example Program 5 – Single Thread

Task: Using a subroutine, increment the motor by 1000 whenever the DI1 rising edge is detected.

```
HSPD=20000                  ;* Set the high speed to 20000 pulses/sec
LSPD=1000                   ;* Set the low speed to 1000 pulses/sec
ACC=300                     ;* Set the acceleration to 300 msec
EO=1                        ;* Enable the motor power
V1=0                        ;* Set variable 1 to zero
WHILE 1=1                   ;* Forever loop
    IF DI1=1                ;* If digital input 1 is on, execute the statements
        GOSUB 1             ;* Move to zero
    ENDIF
ENDWHILE                    ;* Go back to WHILE statement
END

SUB 1
    XV1                     ;* Move to V1 target position
    WAITX                   ;* Wait for X-axis move to complete
    V1=V1+1000              ;* Increment V1 by 1000
    WHILE DI1=1             ;* Wait until the DI1 is turned off so that
    ENDWHILE                ;* multiple increment are not continuously done
ENDSUB
```

9.2.6. Standalone Example Program 6 – Single Thread

Task: If digital input 1 is on, move to position 1000. If digital input 2 is on, move to position 2000. If digital input 3 is on, move to 3000. If digital input 5 is on, home the motor in negative direction. Use digital output 1 to indicate that the motor is moving or not moving.

```
HSPD=20000              ;* Set the high speed to 20000 pulses/sec
LSPD=1000               ;* Set the low speed to 1000 pulses/sec
ACC=300                 ;* Set the acceleration to 300 msec
EO=1                    ;* Enable the motor power
WHILE 1=1               ;* Forever loop
        IF DI1=1        ;* If digital input 1 is on
                X1000   ;* Move to 1000
                WAITX   ;* Wait for X-axis move to complete
        ELSEIF DI2=1    ;* If digital input 2 is on
                X2000   ;* Move to 2000
                WAITX   ;* Wait for X-axis move to complete
        ELSEIF DI3=1    ;* If digital input 3 is on
                X3000   ;* Move to 3000
                WAITX   ;* Wait for X-axis move to complete
        ELSEIF DI5=1    ;* If digital input 5 is on
                HOMEX-  ;* Home the motor in negative direction
                WAITX   ;* Wait for X-axis home move to complete
        ENDIF
        V1=MSTX         ;* Store the motor status to variable 1
        V2=V1&7         ;* Get first 3 bits
        IF V2!=0        ;* If one of first 3 bits is high (X axis moving)
                DO1=1   ;* Turn on digital output 1
        ELSE            ;* Else if first 3 bits are low (X axis idle)
                DO1=0   ;* Turn off digital output 1
        ENDIF
ENDWHILE                ;* Go back to WHILE statement
END
```

9.2.7. Standalone Example Program 7 – Multi Thread

Task: Program 0 will continuously move the motor between positions 0 and 1000. Simultaneously, program 1 will control the status of program 0 using digital inputs.

```
PRG 0                       ;* Start of Program 0
HSPD=20000                  ;* Set high speed to 20000 pulses/sec
LSPD=500                    ;* Set low speed to 500 pulses/sec
ACC=500                     ;* Set acceleration to 500 msec
WHILE 1=1                   ;* Forever loop
        X0                  ;* Move to position 0
        WAITX               ;* Wait for the move to complete
        X1000               ;* Move to position 1000
        WAITX               ;* Wait for the move to complete
ENDWHILE                    ;* Go back to WHILE statement
END                         ;* End Program 0

PRG 1                       ;* Start of Program 1
WHILE 1=1                   ;* Forever loop
        IF DI1=1            ;* If digital input 1 is triggered
                ABORTX      ;* Stop movement
                SR0=0       ;* Stop Program 1
        ELSE                ;* If digital input 1 is not triggered
                SR0=1       ;* Run Program 1
        ENDIF
ENDWHILE                    ;* Go back to WHILE statement
END                         ;* End Program 1
```

9.2.8. Standalone Example Program 8 – Multi Thread

Task: Program 0 will continuously move the motor between positions 0 and 1000. Simultaneously, program 1 will monitor the communication time-out parameter and triggers digital output 1 if a time-out occurs. Program 1 will also stop all motion, disable program 0 and then re-enable it after a delay of 3 seconds when the error occurs.

```
PRG 0                        ;* Start of Program 0
HSPD=1000                    ;* Set high speed to 1000 pulses/sec
LSPD=500                     ;* Set low speed to 500 pulses/sec
ACC=500                      ;* Set acceleration to 500 msec
TOC=5000                     ;* Set time-out alarm to 5 seconds
EO=1                         ;* Enable motor
WHILE 1=1                    ;* Forever loop
    X0                       ;* Move to position 0
    WAITX                    ;* Wait for the move to complete
    X1000                    ;* Move to position 1000
    WAITX                    ;* Wait for the move to complete
ENDWHILE                     ;* Go back to WHILE statement
END                          ;* End Program 0

PRG 1                        ;* Start of Program 1
WHILE 1=1                    ;* Forever loop
    V1=MSTX&2048             ;* Get bit time-out counter alarm variable
    IF V1 = 2048             ;* If time-out counter alarm is on
        SR0=0                ;* Stop program 0
        ABORTX               ;* Abort the motor
        DO=0                 ;* Set DO=0
        DELAY=3000           ;* Delay 3 seconds
        SR0=1                ;* Turn program 0 back on
        DO=1                 ;* Set DO=1
    ENDIF
    ENDWHILE                 ;* Go back to WHILE statement
END                          ;* End Program 1
```

A: Speed Settings

HSPD value [PPS] [1]	Speed Window [SSPDM]	Min. LSPD value	Min. ACC [ms]	δ	Max ACC setting [ms]
1 - 65K	0,1	1	2	50	((HSPD – LSPD) / δ) × 1000
65K - 130K	2	2	1	100	
130K - 325K	3	5	1	200	
325K - 650 K	4	10	1	800	
650K - 1.3M	5	20	1	1500	
1.3M - 3.2M	6	50	1	3800	
3.2M - 6M	7	100	1	7500	

Table A.0

[1] If StepNLoop is enabled, the [HSPD range] values needs to be transposed from PPS (pulse/sec) to EPS (encoder counts/sec) using the following formula:

EPS = PPS / Step-N-Loop Ratio

A.1. Acceleration/Deceleration Range

The allowable acceleration/deceleration values depend on the **LS** and **HS** settings.

The minimum acceleration/deceleration setting for a given high speed and low speed is shown in Table A.0.

The maximum acceleration/deceleration setting for a given high speed and low speed can be calculated using the formula:

Note: The ACC parameter will be automatically adjusted if the value exceeds the allowable range.

$$\boxed{\textbf{Max ACC} = ((HS - LS) / \delta) \times 1000 \text{ [ms]}}$$

Figure A.0

Examples:

a) If **HSPD** = 20,000 pps, **LSPD** = 10,000 pps:
   a. Min acceleration allowable: **1 ms**
   b. Max acceleration allowable:
      ((20,000 – 10000) / 50) x 1,000 ms = **200,000 ms** (200 sec)

b) If **HSPD** = 900,000 pps, **LSPD** = 9,000 pps:
   a. Min acceleration allowable: **1 ms**
   b. Max acceleration allowable:
      ((900,000 – 9,000) / 1500) x 1000 ms = **594,000** ms (594 sec)

A.2. Acceleration/Deceleration Range – Positional Move

When dealing with positional moves, the controller automatically calculates the appropriate acceleration and deceleration based on the following rules.

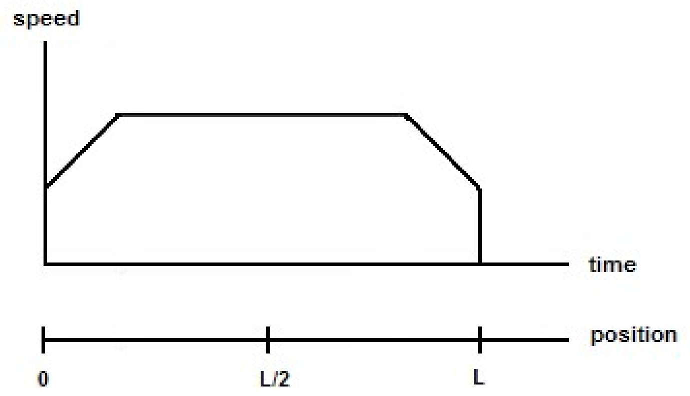

Figure A.1

1) ACC vs. DEC 1: If the theoretical position where the controller begins deceleration is less than L/2, the acceleration value is used for both ramp up and ramp down. This is regardless of the EDEC setting.
2) ACC vs. DEC 2: If the theoretical position where the controller begins constant speed is greater than L/2, the acceleration value is used for both ramp up and ramp down. This is regardless of the EDEC setting.
3) Triangle Profile: If either (1) or (2) occur, the velocity profile becomes triangle. Maximum speed is reached at L/2.