PiCPro™

Appendices A-S

Version 16.1

G & L Motion Control Inc.

NOTE

Progress is an on going commitment at G & L Motion Control Inc. We continually strive to offer the most advanced products in the industry; therefore, information in this document is subject to change without notice. The illustrations and specifications are not binding in detail. G & L Motion Control Inc. shall not be liable for any technical or editorial omissions occurring in this document, nor for any consequential or incidental damages resulting from the use of this document.

DO NOT ATTEMPT to use any G & L Motion Control Inc. product until the use of such product is completely understood. It is the responsibility of the user to make certain proper operation practices are understood. G & L Motion Control Inc. products should be used only by qualified personnel and for the express purpose for which said products were designed.

Should information not covered in this document be required, contact the Customer Service Department, G & L Motion Control Inc., 672 South Military Road, P.O. Box 1960, Fond du Lac, WI 54936-1960. G & L Motion Control Inc. can be reached by telephone at (920) 921-7100 or (800) 558- 4808 in the United States or by e-mail at glmotion.support@danahermotion.com.

Release 200822

Catalog No. (Order No.) M.1300.7592

Printed Version Part Number (Volume 1 and 2) M.3000.0421

Electronic Version Part Number (Volume 1) M.3000.0420 - Rev 1

Electronic Version Part Number (Volume 2) M.3000.0418 - Rev 1

Table Of Contents

APPENDIX A - Quick Reference to In-Out Function Data

Function Input/Output Data Type Reference

Function Input/Output Type Reference

This is a quick reference giving the data types of all the inputs and outputs of the functions and function blocks used in PiCPro. The input or output is on the left and the data type is on the right.

If an input/output is listed more than once, it has more than one data type depending on the function it comes from. In that case, the functions are listed in the third column. All the various IN inputs and OUT outputs are covered in the tables that follow.

Data Type Reference

Numeric	**Data**	**Function**
4mA	BOOL	
10ms	BOOL	
100ms	BOOL	
A	**Data Type**	**Function**
ACC	LREAL	
ACCL	UDINT	
ACT	DINT	PID
ACT	INT	IPREAD, IPRECV, IPWRITE, NETRCV, NETSEND, READ, WRITE
ACTV	WORD	
ANGL	REAL/LREAL	
AXIS	USINT	
B	**Data Type**	**Function**
BEG	BOOL	
BIPO	BOOL	
BKPR	BOOL	
BTVL	DINT	
BUFR	ARRAY, STRUCT, STRING	
C	**Data Type**	**Function**
CAM	ARRAY OF STRUCT	
CD	BOOL	
CFG	STRING	
CFGZ	STRING	
CHAN	USINT	
CLRC	UINT	
CLSD	BOOL	
CMD	UINT	
CNFG	STRUCT	

CNT	INT	
CNTL	UINT	
COMN	STRUCT	
CONF	STRUCT	
COS	REAL/LREAL	
CSTP	BOOL	
CU	BOOL	
CV	INT	
D	**Data Type**	**Function**
DABL	BOOL	
DATA	DINT	
DATA	ARRAY	IO_CFG
DATA	STRUCT	SCA_RECV, SCA_SEND, SCS_RECV, SCS_SEND
DAY	UINT	
DBUF	ARRAY, STRUCT, STRING	
DCNT	INT	
DECL	UDINT	
DEN	INT	
DERR	INT	
DEST	ARRAY OF STRUCT	
DID	USINT	
DIFF	IN Input Reference	Same as IN0
DIM	DINT	
DIR	STRING	
DIST	DINT	DISTANCE, FAST_QUE
DIST	UDINT	REGIST
DVND	NUMERIC or TIME	
DONE	BOOL	
DROP	DINT	
DTST	STRING	
DVSR	Same NUMERIC as DVND or DINT if DVND = TIME	
E	**Data Type**	**Function**
ELEM	UINT	
EN	BOOL	
ERR	BOOL	TME_ERR?, ANLG_OUT

ERR	INT	ASSIGN, CLOSE, CONFIG, COORD2RL, DELFIL, FB_STA, FRESPACE, IPACCEPT, IPCLOSE, IPCONN, IPHOSTID, IPLISTEN, IPNAM2IP, IPREAD, IPRECV, IPSEND, IPSOCK, IPWRITE, OI_SER, NETOPEN, NETRCV, NETSND, OPC_ENET, OPEN, READ, RENAME, RESUME, SCA_ACKR, SCA_CLOS, SCA_CTRL, SCA_ERST, SCA_PBIT, SCA_RCYC, SCA_RECV, SCA_REF, SCA_SEND, SCA_WCYC, SCR_ERR, SCS_ACKR, SCS_CTRL, SCS_RECV, SCS_REF, SCS_SEND, SEEK, STATUS, TUNEWRIT, WRITE
ERR	SINT, UINT	IO_CFG, SC_INIT
ERR	USINT	A_INCHIT, A_INCHRD, A_INMDIT, ARTCHIT, ARTDCHRD, ARTDMDIT, ATMPCHIT, ATMPCHRD, ATMPMDIT, STRTSERV, CAPTINIT, SCR_CONT, SERVOCLK
ERRS	WORD	
ESTP	BOOL	
ET	TIME	
F	**Data Type**	**Function**
FAHR	BOOL	
FAIL	BOOL	
FAST	USINT	
FU	DINT	
FUNC	USINT	
G	**Data Type**	**Function**
G	BOOL	
H	**Data Type**	**Function**
HILT	BOOL	
HNDL	INT	CLOSE, CONFIG, OPEN, READ, SEEK, STATUS, WRITE
HNDL	UINT	IPACCEPT, IPCLOSE, IPCONN, IPLISTEN, IPREAD, IPRECV, IPSEND, IPSOCK, IPWRITE
HOSZ	STRING	
I	**Data Type**	**Function**
I147	WORD	
IDN	UINT	
IGNR	UDINT	
INDX	USINT	
INPS	BOOL	
INs	IN Input Reference	
IPZ	STRING	
IST	STRUCT	

J	Data Type	Function
JERK	LREAL	
K	**Data Type**	**Function**
K	NUMERIC	
L	**Data Type**	**Function**
L	INT	
LD	BOOL	
LEN	INT	
LGTH	UDINT	
LOG	REAL/LREAL	
LOLT	BOOL	
LN	REAL/LREAL	
LU	DINT	
M	**Data Type**	**Function**
MAIN	STRUCT	
MAN	BOOL	
MAST	USINT	
MAX	Same as MIN	
MCND	NUMERIC or TIME	
MDST	DINT	
µsec	UINT	
MIN	Any	
MODE	INT	
MOVE	STRUCT	
MPLR	Same NUMERIC as MCND or DINT if MCND = TIME	
MSTR	DINT	
N	**Data Type**	**Function**
N	USINT	
NAME	STRING	
NAMZ	STRING	
NODE	USINT	
NUM	INT	RATIOSCL
NUM	NUMERIC	NUM2STR, STR2NUM
NUM	REAL/LREAL	EXP, LOG, LN
NUM	USINT	IO_CFG, STR2USI, USIN2STR,
O	**Data Type**	**Function**
OFF	DINT	
OFST	UINT	
OIER	INT	
OIFL	BOOL	

OK	BOOL	
ON	DINT	
ONCE	BOOL	
ONLI	BOOL	
OPTN	USINT	SC_INIT, SCA_CTRL, SCS_CTRL
OPTN	WORD	FAST_REF, LAD_REF, RATIOCAM, RATIOSCL, RATIOSLP, RATIO_RL, SCA_REF, SCS_REF
ORG	INT	
OUT	OUT output reference	
P	**Data Type**	**Function**
P	NUMERIC	DELETE, INSERT, MID, REPLACE
PHAS	USINT	
PLUS	BOOL	
PNT	USINT	
PNUM	USINT	
PORT	STRING	OI_SER
PORT	UINT	IPCONN, IPSEND, OPC_ENET
POS	DINT	
POSN	ARRAY OF STRUCT	
PRB	USINT	
PRI	BOOL	
PROD	Same as MCND	
PROT	UINT	
PT	TIME	
PTR	ARRAY OF STRUCT	
PV	INT	
Q	**Data Type**	**Function**
Q	BOOL	
QAVL	BOOL	
QTY	DINT	
QTY	USINT	BIO_PERF, PLS
QUE	USINT	
QUOT	Same as DVND	
R	**Data Type**	**Function**
R	BOOL	
RACK	USINT	
RATE	UDINT	
RATE	TIME	SERVOCLK
RDNE	BOOL	
REAL	ARRAY OF STRUCT	
REFD	DINT	

REM	Same as DVND	
REQ	BOOL	
RETR	BOOL	
REV	BOOL	
ROOT	Same as SQR	
RNGE	USINT	
RPER	USINT	
RPTP	BOOL	
RSCD	STRUCT	
RSLT	DINT	
RSMD	BOOL	
S	**Data Type**	**Function**
SDIR	BOOL	
SDST	DINT	
SEG1	STRUCT	
SERR	UINT	
SET	BOOL	
SID	USINT	
SIN	REAL/LREAL	
SIZE	DINT	
SIZE	USINT	CAPTINIT
SLOT	USINT	
SLPE	ARRAY OF STRUCT	
SLV	UINT	
SPT	DINT	
SQR	UDINT, UINT, USINT, or constant	
SR	STRUCT	
SRCE	ARRAY OF STRUCT	
SRS	STRUCT	
SSTR	DINT	
STAT	DWORD	FB_STA
STAT	INT	NETMON, SCA_REF, SCS_REF, STATUS
STAT	WORD	SCA_STAT, SCS_STAT, STATUSSV, STEPSTAT
STOP	BOOL	
STR	STRING	
STRC	STRUCT	
STRT	BOOL	
SUM	IN input reference	Same as IN1

T	Data Type	Function
TAN	REAL/LREAL	
TASK	STRUCT	
TBUF	ARRAY, STRUCT, STRING	
TCNT	INT	
TIME	UINT	
TOLR	UDINT	
TYPE	USINT	
V	**Data Type**	**Function**
VALU	INT	
VAR	SINT	
VARS	STRUCT	
W	**Data Type**	**Function**
WEEK	BOOL	

IN Inputs Reference

[Inputs for extensible functions are followed with (ext).]

Input	Data Type	Function
IN	any	SIZEOF
	BITWISE	NOT, ROL, ROR, SHL, SHR
	BOOL	TOF, TON, TP
	DATE	DATE2STR
	DATE_AND_TIME	CLOCK, DT2DATE, DT2STR, DT_2_TOD
	NUMERIC	ABS, NEG
	STRING	DELETE, LEFT, MID, RIGHT
	TIME	TIME2STR
	TIME_OF_DAY	TOD2STR
	Same as MIN	LIMIT
IN0	NUMERIC OR TIME	SUB
	any except STRUCT	SEL
IN0 (ext)	any except STRUCT	MUX

Input	Data Type	Function
IN1	any except BOOL or STRUCT	NE
	BOOL	SCA_CTRL, SCS_CTRL
	DATE	D_TOD2DT, S_D_D
	DATE_AND_TIME	A_DT_T, S_DT_DT, S_DT_T

	STRING	FIND, INSERT, REPLACE
	TIME_OF_DAY	A_TOD_T, S_TOD_T, S_TOD_TO
	same as IN0	SEL, SUB
IN1 (ext)	any	MOVE
	any except BOOL or STRUCT	EQ, GE, GT, LE, LT, MAX, MIN
	BITWISE	AND, OR, XOR
	NUMERIC or TIME	ADD
	STRING	CONCAT
	same as IN0	MUX

Input	**Data Type**	**Function**
IN2	BOOL	SCA_CTRL, SCS_CTRL
	DATE	S_D_D
	DATE_AND_TIME	S_DT_DT
	TIME	A_DT_T, A_TOD_T, S_DT_T, S_TOD_T
	TIME_OF_DAY	D_TOD2DT, S_TOD_TO
	same as IN1	NE
	STRING	FIND, INSERT, REPLACE
IN2 (ext)	same as IN1	ADD, AND, EQ, GE, GT, LE, MAX, MIN, OR, XOR
	STRING	CONCAT, REPLACE
IN3	BOOL	SCA_CTRL, SCS_CTRL

OUT Output Reference

Output	**Data Type**	**Function**
OUT	BOOL	CAM_OUT, EQ, GE, GT, LE, LT, NE, PLS
	DATE	DT2DATE
	DATE_AND_TIME	A_DT_T, CLOCK, D_TOD2DT, S_DT_T
	same as IN	ABS, NEG, NOT, ROL, ROR, SHL, SHR
	same as IN0	MUX, SEL
	same as IN1	AND, OR, XOR
	same as MIN	LIMIT
	NUMERIC	FIND
	UINT	SIZEOF
	TIME	S_D_D, S_DT_DT, S_TOD_TO
	TIME_OF_DAY	A_TOD_T, DT2TOD, S_TOD_T
OUT1	same as IN1	MAX, MIN, MOVE
OUT—OUT	STRING	CONCAT, DATE2STR, DELETE, DT2STR, INSERT, LEFT, MID, REPLACE, RIGHT, TIME2STR, TOD2STR

Note:

There are also INs/OUTs on all the data type conversion functions. In those functions, the IN is always the data type you are converting from and the OUT is always the data type you are converting to.

APPENDIX B - Errors

Overview: Errors

When errors occur they are reported to you in the information window or in a message dialog. PiCPro errors include:

- Compile Error Messages (Ladder)
- Custom Motor Error Messages
- Dependency List Error Messages
- Fieldbus Error Messages
- Function Error Codes
- Function Errors
- Hardware Declaration Error Messages
- Ladder Error Messages
- Library Error Messages
- SERCOS Error Messages
- Servo C-Stop, E-Stop and Programming Error Codes
- Servo Setup Error Messages
- TCP/IP Error Codes

Compile Error Messages (Ladder)

Compile Error Messages (Ladder)

If the compile process is unsuccessful, errors will be reported in the information window. Typically, you must re-edit the ladder to correct the errors and recompile.

These errors can be the following types:

1. **Fatal** - indicates a severe problem that prevents the compile from being completed.
2. **Error** - indicates a program syntax error.
3. **Warning** - provides an informational message. A warning does not prevent the compile command from being completed, but it is recommended that the situation that caused the warning be corrected.

If a compile error does occur, you can double-click on it in the information window and it will navigate you to the place in the ladder where the error occurred. You can get help on an error by single clicking on the error in the information window and pressing the **F1** key.

1000 BINARY: BUILD

A file open exception has occurred. Error opening file "__".

PiCPro has attempted to open the indicated file (usually a BIN or temporary file) and was unsuccessful.

Tip: Ask these questions

- Is the disk drive full?
- Is the disk drive write protected?
- Is the TEMP/TMP environment variable pointing to a valid drive?

1002 BINARY: OUT OF MEMORY

Out of control application memory space.

The ladder code that is being compiled requires more memory than is available in the processor specified in the hardware declarations.

1003 BINARY: FUNCTION NOT FOUND

"_" not found in libraries.

There is a function required by your application that cannot be found.

Tip: Check to make sure your library paths are correct.

1004 BINARY: ARRAY MISMATCH

External data typing error, array mismatch, "_" task "_".

The indicated external variable declared in the indicated task was found in the main module, but one is declared as an array and the other is not. They cannot be different.

Tip: Either declare both as an array or not.

1005 BINARY: EXTERNAL UNDEFINED

External "_" in task "_" has no source in the main module.

A variable marked with the external attribute in the software declarations table in the task ladder was not declared in the main ladder as required.

Tip:

Any variable used in a task that has been marked as external must also be declared in the main ladder whether or not the main ladder uses it. It must never be marked as external in the main ladder, only in the task ladder.

1006 BINARY: TYPE MISMATCH

External data typing error, "_" in task "_", "_".

The indicated external variable in the indicated task was found in the main module, but the variable types are different as indicated. The types must match.

Tip:

Be sure that the variable type of the external variable in a task has been declared in the main ladder with the same variable type.

1007 BINARY: OUT OF PATCHES

More than 100 patches. A scan stopped, full module download is required.

The control provides for 100 patches, which have been used up.

Tip:

- Do a full download to incorporate all the existing patches in the application.
- After a full download, the patch area is again available for another 100 patches.

- As you work with on-line edit, check the resources available summary in the information window to see how many patches you have left.

1008 BINARY: TOO MANY PATCHES

Too many patch operations at once. A scan stopped, full module download is required.

There is a limit of 40 internal operations in any one patch download, depending on memory available. Every time a network is modified it is considered one patch, but it could include several internal operations. This error simply means you are trying to do too much at one time.

Tip: Do a full download and proceed.

1009 BUILDER: NAME CHANGE

Windows filename "_" changed to "_" in the control.

The control only supports the original DOS 8.3 filename format. The filename supplied does not comply with that format and has been modified as indicated for use in the control. This is an informational message.

Tip:

Use DOS format for a maximum of 8 characters, followed by a period, then 3 characters when naming your files.

1010 OLE: HARDWARE CHANGED

Hardware declarations have changed.

Changes to existing hardware declarations will prevent a patch download.

Tip:

If you make changes to the hardware declarations table, you must perform a compile and download before attempting to patch the ladder.

1011 OLE: UDFB BIT MEMORY

Out of function block bit memory. A remake of this library function is required, along with a full compile and download.

There is at least one new BOOLEAN variable for which there is no room reserved in the control data memory. Either a debug version of the UDFB is not being used, or there have already been 40 additional bits used since the last remake and full download.

Tips:

Remake the UDFB to incorporate any new variables and download the entire application.

OR

Delete the additional variable(s) and all the places they are used.

Note:

If you plan on adding variables you must have a debug version of the UDFB downloaded. When working with on-line edit, check the resources available summary in the information window.

1012 OLE: UDFB BYTE MEMORY

Out of function block byte memory. A remake of this library function is required, along with a full compile and download.

There is at least one new variable for which there is no room reserved in the control data memory. Either a debug version of the UDFB is not being used or there have already been 80 additional bytes used since the last remake and full download.

Tips:

Remake the UDFB to incorporate any new variables and download the entire application.

OR

Delete the additional variable(s) and all the places they are used.

Note:

If you plan on adding variables you must have a debug version of the UDFB downloaded. When working with on-line edit, check the resources available summary in the information window.

1013 OLE: UDFB LINKS

Out of local label/function links.

There is at least one new function or network label for which there is no room reserved in this UDFB code memory. Either a debug version of the UDFB is not being used, or there have already been 20 additional network labels of functions used since the last remake and full download.

Tips:

Remake the UDFB to incorporate the new labels and download.

OR

Delete the additional labels and functions.

Note:

If you plan on adding variables you must have a debug version of the UDFB downloaded. When working with on-line edit, check the resources available summary in the information window.

1014 OLE: GLOBAL LINKS

Too many functions/labels. Out of global link table space.

PiCPro establishes a link table for labels and/or function/blocks added during on-line editing. There is a limit of 26 links. Every time you do a full download, this link area becomes available again.

Tip: Do a full download when you get this message and the link area will become available.

1015 OLE: LABEL UNDEFINED

Network label "_" is undefined.

Network labels are required on Jump to Label and Jump to Subroutine commands. If you enter a label that you have not assigned to a network, you will get this error.

Tip:

Ensure that any network you want to jump to has a label assigned to it. If you get this error after entering a label with the jump command, make sure the label exists and/or check the spelling of the label to ensure it matches the label of the destination network.

1072 COMPILE: BUILD ERROR OBSOLETE CPU

Displayed when the attempt is made to compile (bin, hex, task, UDFB) a ladder that specifies an obsolete CPU in Hardware Declarations.

Tip: Edit Hardware Declarations and select a proper CPU.

1073 APP: OUT OF RETAINED DATA MEMORY

Data memory has been exceeded. The limit is 24K.

Custom Motor Error Messages

CUSTOM MOTOR ERROR: CANNOT ALIGN STEGMANN ENCODER

Attempt made to enter alignment mode while connected to a drive with a motor that has a Stegmann encoder. PiCPro cannot align a motor with a Stegmann encoder.

CUSTOM MOTOR ERROR: CUSTOM MOTOR FILE NOT USED

A custom motor file name has also been found in the installed motor file.

Tip: Rename the custom motor file with a name that is not in the installed motor file.

CUSTOM MOTOR ERROR: FILE NOT USED INVALID CHECKSUM

A custom motor has been found with an invalid checksum. The file may have been edited outside of PiCPro.

Tip: Recreate the file using PiCPro.

CUSTOM MOTOR ERROR: FILE NOT USED TOO MANY ENTRIES

A custom motor file has been found with more than one custom motor entry. The file must have been edited outside of PiCPro.

Tip: Recreate the file using PiCPro.

CUSTOM MOTOR ERROR: FILE UNKNOWN PARAMETERS NOT USED

A custom motor file has been found that contains information for motor parameters that are not supported by the firmware versions supported in this version of PiCPro. If changes are made to the custom motor and the file is saved in this version of PiCPro, the unsupported motor parameters will be lost.

CUSTOM MOTOR ERROR: INVALID CHECKSUM

An attempt was made to open a custom motor file containing an invalid checksum. The file may have been edited outside of PiCPro.

Tip: Recreate the file using PiCPro.

CUSTOM MOTOR ERROR: INVALID CUSTOM MOTOR FILE LOCATION

An attempt has been made to read the custom motor files when the custom motor directory does not exist. Select **View | Options** and click the **Drive Setup** tab to change the location of custom motor files.

CUSTOM MOTOR ERROR: TOO MANY ENTRIES

A custom motor file has been found with more than one custom motor entry. The file must have been edited outside of PiCPro.

Tip: Recreate the file using PiCPro.

Dependency List Error Messages

1016 DEPEND: FILE EXCEPTION

This error can occur when building a dependency list.

1018 DEPEND: FILE READ EXCEPTION

This error can occur when reading a file during the building of a dependency list.

1019 DEPEND: FILE WRITE EXCEPTION

This error can occur when writing a file during the building of a dependency list.

1020 DEPEND: SRVFILE NOT FOUND

The servo file could not be found.

1021 DEPEND: SCRFILE NOT FOUND

The SERCOS file could not be found.

Fieldbus Error Messages

9071 FIELDBUS: LIBRARY NOT AVAILABLE

The library file for storing control fieldbus interface information could not be located in the library directories.

Check that you have indicated the correct directory for the libraries. Select **File | PiCPro Libraries…** from the menu.

9072 FIELDBUS: UNRECOGNIZED FORMAT

Unrecognized fieldbus information file format.

Your ladder's UCT file does not have the correct format. The most likely cause is manual editing of the UCT file. Re-create this file using the Danaher Motion DeviceNet Configurator.

9073 FIELDBUS: UNDEFINED TAG

Fieldbus information file line '_____'. '_____' undefined Tag Name.

The tag name specified in the error message cannot be found in the software declarations for the associated ladder. The tag name may have been spelled wrong when it was entered in the G & L DeviceNet™ Configurator. If so, run the Configurator and correct the problem. Or the variable really does need to be added to your ladder's software declarations. If this is the case, go to software declarations and add a variable with the same name and type as declared in the Configurator.

9074 FIELDBUS: INVALID LOCATION

Fieldbus information file line '_____'. '_____' invalid IRAM Location.

The tag name specified in the error message has an invalid IRAM location configured for it. The most likely cause of this error is manual editing of the UCT file. Re-create the file using the Danaher Motion DeviceNet Configurator.

9076 FIELDBUS: INVALID MASK

Fieldbus information file line '_____'. '_____' invalid Bit Mask.

The tag name specified in the error message has an invalid bit mask associated with it. The valid values are 0 through 7. The most likely cause of this error is manual editing of the UCT file. Re-create the file using the Danaher Motion DeviceNet Configurator.

9077 FIELDBUS: INVALID TYPE

Fieldbus information file line '_____'. '_____' invalid Data Type.

The tag name specified in the error message has a data type that does not match the corresponding variable's type in the ladder's software declarations. Run the Danaher Motion DeviceNet™ Configurator and change the data type for this tag name to match the type for the same variable in software declarations – Or – edit the software declarations for your ladder and change the variable's type.

9078 FIELDBUS: INVALID SIZE

Fieldbus information file line '_____'. '_____' invalid Size.
The tag name specified in the error message has an invalid size associated with it. The most likely cause of this error is manual editing of the UCT file. Re-create the file using the Danaher Motion DeviceNet Configurator.

9079 FIELDBUS: INVALID UPDATE

Fieldbus information file line '_____'. '_____' invalid Update.

The tag name specified in the error message has an invalid update type associated with it. The currently supported values are "Polled Input" (bit 0=1) and "Polled Output" (bit 1=1). The most likely cause of this error is manual editing of the UCT file. Re-create the file using the Danaher Motion DeviceNet Configurator.

9080 FIELDBUS: INVALID FORMAT

Fieldbus information file line '_____'. Invalid format. '_____'

The specified line in your UCT is invalid. The most likely cause of this error is manual editing of the UCT file. Re-create the file using the Danaher Motion DeviceNet Configurator.

9081 FIELDBUS: INVALID SLOT

Fieldbus information file line '_____'. Invalid Slot information.

The specified line in your UCT has an invalid slot number associated with it. The most likely cause of this error is incorrect manual editing of the UCT file. Make sure the Slot number is within the valid range of 3 through 13.

Function Error Codes

Function Error Codes

The categories of function errors are:

- General function errors
- I/O block function block errors
- String function errors
- Stepper errors
- Start servo function errors

General Function Errors

For all functions, the output variables will have unpredictable values and the output at OK, DONE, or Q will not be energized whenever the following occurs.

- An output variable does not have enough bits to hold the result.
- An output variable is an unsigned integer and the result is negative.
- The operation attempts to divide a number by zero.
- The input data is invalid.

I/O Function Block Error Codes

If an error occurs when the I/O functions execute, the following occurs.

- The output at DONE is not energized.
- The output at FAIL is energized.
- The ERR type of output (e.g. ERR, FERR, OERR, RERR etc.) holds one of the error numbers listed below.

Error #	Error Description
01	**opcode_error** Invalid device open mode specified for function (e.g. 16#602)
02	**handle_error** The handle to a function is invalid. Possible reasons include: • Using I/O functions with an unopened handle • Attempting to do a READ on a handle opened for WRITE ONLY • Attempting to do a WRITE to a handle opened for READ ONLY
03	**already_open** Reserved for Danaher Motion. Internal GLOS error
04	**open_error** OPEN mode (READ/WRITE/APPEND) specified is invalid.
05	**device_empty** Reserved for Danaher Motion

06	**read_error** Depending on the device you are using, this error will be one of those listed below. • DISK driver detects a checksum error on the DISK. Media error on DISK reading disk FCB (File Control Block). • A parity, overrun, or framing error exists in the data in the input buffer of the COM port.
07	**write_error** Error writing to the device
08	**write_protect** Reserved for Danaher Motion
09	**device_error** Caused by any of the following: • The Seek origin is not A00, A01, or A02. • Improper device name (Choices are PICPRO:, RAMDISK:, FMSDISK:, or USER:) • Attempting to do a STATUS on a file • Attempting to do a file operation on a non-DOS device • Bad parity in the configuration string or a hardware error in configuring the port • Attempting to do a second or subsequent operation before the first one was completed
10	**out_of_handles** Too many files open. A maximum of 10 handles can be used. READ/WRITE or APPEND operations use two handles. READ ONLY or WRITE ONLY operations use one handle.
11	**invalid_device** Improper device name (Choices are PICPRO:, RAMDISK:, FMSDISK:, USER:, or any name you have assigned at the NAMZ input of a function block)
12	**invalid_file_name** Filename format is wrong. Filename is too large.
13	**too_many_drivers** Reserved for Danaher Motion. Internal GLOS error
14	**too_many_connections** Reserved for Danaher Motion. Internal GLOS error
15	**read_write_error** Error reading or writing a disk file
16	**device_error** Reserved for Danaher Motion
17	**file_not_open** Attempting a READ, WRITE, or SEEK on a file that is not open.

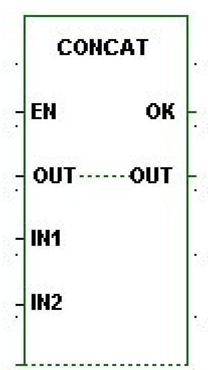
18	**invalid_access** Caused by any of the following: • Attempting to CLOSE a file that was not open • SEEK has a problem seeking a new location. • Attempting to write to a file opened for read only
19	**file_not_in_dir** Caused by one of the following: • Filename not found in directory • Volume name not found in directory
20	**file_already_open** Attempting to OPEN a file that is already open
21	**write_protect** Error occurs when attempting to delete a READ ONLY file from DISK.
22	**mode_error** Caused by one of the following: • Invalid mode inputted to OPEN • Invalid mode inputted to SEEK
23	**out_of_handles** A maximum of 10 handles are available in the control. This error means there are no more handles available (too many files open).
24	**end_of_dir** No more directory entries
25	**function_locked** Reserved for Danaher Motion. Internal GLOS error.
26	**rename_new_exists** Reserved for Danaher Motion
27	**rename_old_open** Reserved for Danaher Motion
28	**rename_no_old** Reserved for Danaher Motion
29	**attrib_invalid** Reserved for Danaher Motion. Internal GLOS error.
30	**handle_too_large** Reserved for Danaher Motion
31	**disk_error** General error on disk drive
32	**device_parity_error** Reserved for Danaher Motion
33	**end_of_devices** Device driver for the device specified not found
34	**io_dev_aborted** Reserved for Danaher Motion

35	**duplicate_filename** Reserved for Danaher Motion
36	**no_memory** Reserved for Danaher Motion
37	**no_buffers** No memory to get more file buffers
38	**dir_not_empty** Reserved for Danaher Motion
39	**dir_not_found** Reserved for Danaher Motion
40	**out_of_fats** No more entries available in the FAT (File Attribute Table). Reduce the number of files in this directory.
41	**sdir_exists** Reserved for Danaher Motion
42	**filespec_too_long** Reserved for Danaher Motion
43	**no_default_device** Reserved for Danaher Motion
44	**parameter_error** Occurs if the value entered at the CNT input of the READ function block is larger than the declared size of a string used as the BUFR input.

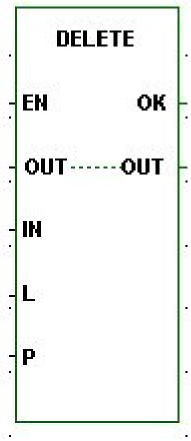
String Function Errors

If an error occurs when the string functions execute, the following occurs.

- The output at OK is not energized.
- The STRING variable output will be null (have a length of zero).

CONCAT Function	**An error occurs if**
	The length of IN1 > the length of OUT The length of IN2 > the length of OUT The length of IN1 + the length of IN2 > the length of OUT IN2, IN3, IN4…IN17 = OUT

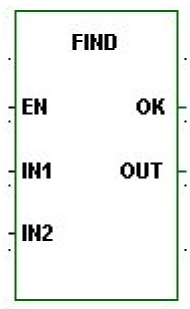
DELETE Function



An error occurs if

P = 0
P > 255
P > length of IN
L > 255
The length of IN – L > the length of OUT

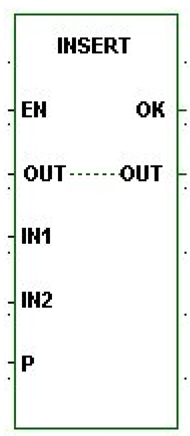
FIND Function



An error occurs if

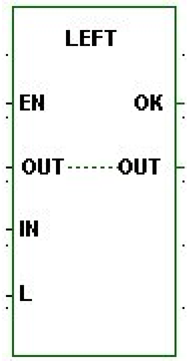
The length of IN1 = 0
The length of IN2 = 0
The length of IN2 > the length of IN1

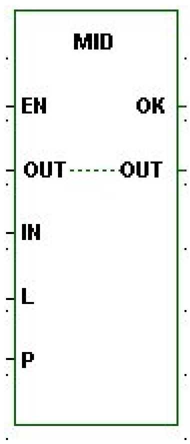
INSERT Function

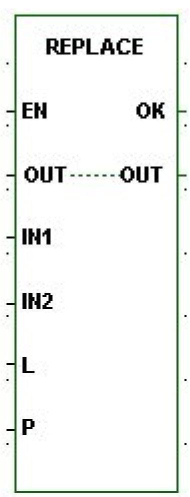


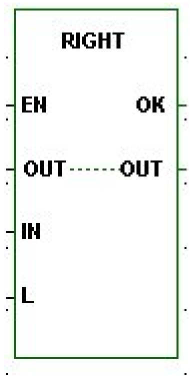
An error occurs if

P = 0
P > 255
P > length of IN
IN2 = OUT
The length of IN1 + the length of IN2 > the length of OUT

LEFT Function	An error occurs if
	L > 255 L > the length of OUT

MID Function	An error occurs if
	P = 0 P > 255 P > length of IN L > 255 L > the length of OUT

REPLACE Function	An error occurs if
	P = 0 P > 255 P > length of IN1 L > 255 IN1 = OUT IN2 = OUT The length of IN1 + the length of IN2 > the length of OUT

RIGHT Function	An error occurs if
	L > OUT L > 255

Function Errors

6008 FUNCTION: NO DEFINITION FOR FUNCTION

The definition for the function _ was not found. Changes to this file cannot be saved unless this function stub is removed.

PiCPro cannot locate the library containing the function. Check that you are pointing at the correct library. Set the path correctly and close and reopen your file.

Hardware Declaration Error Messages

Hardware Declaration Error Messages

The following errors or messages can be received when working with hardware declarations.

4000 HWD: REMOTE TO NONE

You have selected a master rack only hardware configuration. Your hardware is currently configured for remote I/O. All expansion rack information will be deleted and irretrievable.

4001 HWD: PiC9 BOARDS

Selecting a PiC9 CPU automatically puts a Servo Encoder declaration in slot 3 of the master rack and an In/Out 24V DC declaration in slot 4 of the master rack. If you currently have modules declared in these slots, they will be replaced.

4002 HWD: REMOTE TO BLOCK

You have selected a block I/O hardware configuration. Your hardware is currently configured for remote I/O. All expansion rack information will be deleted and irretrievable if you continue.

4003 HWD: BLOCK TO NONE

You have selected a master rack only hardware configuration. Your hardware is currently configured for remote I/O. All block I/O information will be deleted and irretrievable.

4004 HWD: BLOCK TO REMOTE

You have selected a remote I/O hardware configuration. Your hardware is currently configured for block I/O. All block I/O information will be deleted and irretrievable.

4006 HWD: DELETE BLOCK

When you delete a block I/O module, block I/O modules after the deleted module are shifted down and renumbered accordingly. If you want to remove the module without shifting, change the module to an Empty module.

4008 HWD: MAXIMUM RACKS EXCEEDED

The maximum number of expansion racks allowed exists. PiCPro currently supports a maximum of seven expansion racks.

4009 HWD: INSERT BLOCK WARNING

When you insert a block I/O module, all block I/O modules after the inserted module are shifted up and renumbered accordingly.

4010 HWD: MAXIMUM BLOCKS EXCEEDED

PiCPro currently supports 77 block I/O modules. You cannot execute an Insert After or Paste Insert After on block module 77.

4011 HWD: NO CLIPBOARD

The workstation system clipboard could not be opened for the requested function (copy, cut). This indicates a problem with your workstation. Make sure another application does not currently have the clipboard open.

4013 HWD: INVALID PASTE

The keys <**Ctrl + V**> initiate a Paste command. Either nothing has been previously copied or the command is invalid for the currently selected item.

4014 HWD: RESOLVER EXPANSION

There is an Input Resolver (2 or 4 ch) module on the clipboard. You cannot use either of these modules in an expansion rack.

4015 HWD: NOT BLOCK CPU

The CPU currently specified in the Master Rack does not support block I/O.

4016 HWD: NOT REMOTE CPU

The CPU currently specified in the Master Rack does not support remote I/O.

4017 HWD: LAST BLOCK NOT EMPTY

The 77th block module is not an Empty module. Performing any Insert function will remove this module.

4019 HWD: TOO MANY RACKS

This version of PiCPro supports only seven expansion racks. The ladder you are loading currently has eight expansion racks.

The eighth expansion rack will not be included in hardware declarations. Saving a ladder in this condition will not save the eighth rack and render the hardware declarations for the eighth rack as irretrievable.

4021 HWD: INVALID COPY

The keys <**Ctrl + C**> initiate the Copy command. That command is invalid for the currently selected item.

4022 HWD: INVALID CUT

The keys <**Ctrl + X**> initiate the Cut command. That command is invalid for the currently selected item.

4023 HWD: INVALID PASTE/INSERT

The keys <**Ctrl + A**> initiate the Paste Insert After command. Either nothing has been previously copied or the command is invalid for the currently selected item.

4024 HWD: INVALID DELETE

The **Delete** key initiates the delete command. That command is invalid for the currently selected item.

4025 HWD: INVALID INSERT

The **Insert** key initiates the Insert After command. This command is invalid for the currently selected item.

4026 HWD: I/O CONFLICT

The current module in the specified slot of the master rack does not match the hardware declaration information that has been downloaded to the control.

4027 HWD: NO SOFT BIT

The binary file being downloaded to the control requires a CPU in the control that can handle software bit memory. The CPU currently declared in the hardware declarations is not capable of handling software bit memory.

4029 HWD: CONFIRM SWITCH PIC TO STANDALONE MMC

When you switch from a PiC CPU to a standalone MMC CPU, this confirmation dialog will appear asking if you want to continue. If you do, slots 1 and 2 will hold modules for the chosen MMC configuration, slots 3 and 4 are emptied, and slots 5 to 13 and remote I/O are removed. You will want to check the I/O points in your software declarations table to ensure they are mapped to the correct locations. If they are not, make the necessary edits before you proceed.

4030 HWD: CONFIRM SWITCH STANDALONE MMC TO PIC

When you switch from an MMC CPU to a PiC CPU, a confirmation dialog will appear asking if you want to continue. If you do, slots 3 and 4 will be emptied, a CSM appears in slot 1, the chosen CPU appears in slot 2, and slots 5 to 13 are added to the master rack. The Remote I/O option is made available. You will want to check the I/O points in your software declarations table to ensure they are mapped to the correct locations. If they are not, make the necessary edits before you proceed.

Refer to: *Changing standalone MMC CPU*

4031 HWD: CONVERT INVALID CPU FORMAT

If you attempt to open a ladder with a PiC CPU type in the MMC-Limited Edition of PiCPro, you can choose to convert the PiC CPU to an MMC CPU. However, all the I/O in the main and remote racks will be removed in the conversion to an MMC CPU. Software Declarations should be checked to ensure the I/O is mapped to the correct locations after the conversion. See Conversion References in this section.

4032 HWD: PASTE CONFIRM SWITCH MMC TO PIC

You are warned of the consequences of pasting the CPU type from one hardware declarations table to another with a different CPU type.

Refer to: *Changing standalone MMC CPU*

4033 HWD: PASTE CONFIRM SWITCH PIC TO MMC

You are warned of the consequences of pasting the CPU type from one hardware declarations table to another with a different CPU type.

4034 HWD: MMC EDITION DOWNLOAD2MMC ONLY

When working with PiCPro MMC-Limited Edition, you can only download ladders to a standalone MMC control.

4035 HWD: DOWNLOAD FAILURE CPU TYPE MISMATCH

The CPU type you have selected in your ladder program does not match the CPU type in your control. Edit your hardware declarations so that the CPU type in your ladder matches the CPU type in your control.

4036 HWD: OUT OF DATA MEMORY

Out of Data Memory. Out of memory error. Your program requires more memory than is available in the Control CPU.

This is a compile time error of a main ladder. This can occur when the data memory required for retained variables (retentive) and declared hardware exceeds 64K.

This error can also occur on older style CPUs that have hardbit memory, when the number of retained BOOL's and declared hardware data memory exceeds 8K.

The only corrective action in either case is to reduce the amount of hardware or retained variables. Contact Danaher Motion Tech Support for help in optimizing data memory usage.

4037 HWD: LADDER CONFIGURABLE I/O

You have chosen to enable Ladder Configurable I/O. This allows the ladder to continue scanning even if there has been a failure in the remote or block I/O connected to your system. I/O in the main rack will continue to work unless there has been a failure in the main I/O. If that occurs, all I/O in the system becomes non-operational. If the CPU is an analog MMC for PC CPU, ladder configurable I/O applies only to Block I/O. If there has been a failure in ASIU I/O, the system becomes non-operational.

Warning:

Since this will allow the control to continue to scan even if there are I/O errors, the function block IO_CFG must be used to allow the ladder to react to I/O.

Tip:

If you need information on IO_CFG, open Function Block Help by selecting **Help | Function/Function Block Help** from the menu. Select the Index tab and then scroll down until you see **IO_CFG**. Double-click on it. A description of this function block is then displayed on the right.

4038 HWD: CONVERT INVALID MMC FOR PC CPU

The ladder you are trying to open has an MMC for PC type CPU and you are using PiCPro MMC-Limited Edition software.

You are asked if you would like to convert the existing ladder to use a standalone MMC CPU instead.

Note:

Master Rack and ASIU I/O will be removed because the standalone MMC CPU does not support this kind of I/O. Software Declarations should be checked to ensure the I/O is mapped to the correct locations after the conversion. See Conversion References in this section.

Refer to: *Changing MMC for PC CPU*

4039 HWD: LAST ASIU MODULE IS NOT EMPTY

ASIU 8 is not empty. An empty ASIU will be inserted below the ASIU with focus. All ASIUs below the one currently selected will be moved down, and what was ASIU 8 will be removed.

4040 HWD: INSERT ASIU WARNING

An empty ASIU will be inserted below the ASIU currently selected and all subsequent units will be shifted down.

4041 HWD: MAXIMUM ASIU MODULES EXIST

The maximum number of ASIU I/O slots already exist and therefore, no other modules can be inserted after ASIU 8.

4042 HWD: CONFIRM SWITCH MMC TO MMC FOR PC

When you switch from a standalone MMC to an MMC for PC CPU, this confirmation dialog will appear asking if you want to continue. If you do, slot 1 will hold the chosen MMC for PC module, slot 2 is emptied and slots 3 and 4 are removed.

Note: The I/O points defined in Software Declarations should be checked to ensure they are correct.

Refer to: *Changing standalone MMC CPU*

4043 HWD: CONFIRM SWITCH MMC FOR PC ANALOG CPU TO DIGITAL CPU

When you switch from an MMC for PC Analog Servo CPU to a Digital Servo CPU, all of the ASIUs will be removed. You are asked to confirm this action.

Note: The I/O points defined in Software Declarations should be checked to ensure they are correct.

4044 HWD: CONFIRM SWITCH PIC TO MMC FOR PC

When you switch from a PiC CPU to an MMC for PC CPU, slot 2 will be emptied and slots 3-13 will be removed. All remote I/O will also be removed. You are asked to confirm this action.

Note: The I/O points defined in Software Declarations should be checked to ensure they are correct.

4045 HWD: CONFIRM DELETE ASIU

When you delete an ASIU module, all of the subsequent module locations will be shifted. You are asked to confirm this action.

4046 HWD: PASTE NON-MMC FOR PC ANALOG SERVO

When you attempt to paste a non-MMC for PC Analog Servo master rack on an existing MMC for PC Analog Servo master rack, all ASIU I/O will be removed. You are asked to confirm this action.

4047 HWD: CONFIRM SWITCH MMC FOR PC ANALOG SERVO TO PIC

When you switch from an MMC for PC Analog Servo CPU to a PiC CPU, a CSM will be put into slot 1, the selected PiC CPU will be put in slot 2, and slots 3 – 13 will be added to the master rack. All ASIU I/O will be removed. You are asked to confirm this action.

Note:

The I/O points defined in Software Declarations should be checked to ensure they are correct. See Conversion References in this section.

Refer to: *Changing MMC for PC CPU*

4048 HWD: CONFIRM SWITCH MMC FOR PC DIGITAL SERVO TO PIC

When you switch from an MMC for PC Analog Servo CPU to a PiC CPU, a CSM will be put into slot 1, the selected PiC CPU will be put in slot 2, and slots 3 – 13 will be added to the master rack. All ASIU I/O will be removed. You are asked to confirm this action.

Note:

The I/O points defined in Software Declarations should be checked to ensure they are correct. See Conversion References in this section.

Refer to: *Changing MMC for PC CPU*

4049 HWD: CONFIRM SWITCH MMC FOR PC ANALOG SERVO TO STANDALONE MMC

When you switch from an MMC for PC Analog Servo to a standalone MMC, the contents of slots 1 and 2 will be replaced, and slots 3 and 4 will be added to the master rack. All ASIU I/O will be removed. You are asked to confirm this action.

Note: The I/O points defined in Software Declarations should be checked to ensure they are correct.

Refer to: *Changing MMC for PC CPU*

4050 HWD: CONFIRM SWITCH MMC FOR PC DIGITAL SERVO TO STANDALONE MMC

When you switch from an MMC for PC Digital Servo to a standalone MMC, the contents of slots 1 and 2 will be replaced, and slots 3 and 4 will be added to the master rack. You are asked to confirm this action.

Note: The I/O points defined in Software Declarations should be checked to ensure they are correct.

Refer to: *Changing MMC for PC CPU*

4057 HWD: LADDER CONTINUE SCAN WITH ASIU

WARNING
This will allow the control to continue to scan with I/O errors in the Master Rack, Block I/O or ASIUs. The function block IO_CFG must be used to allow the ladder to react to I/O errors.

This message is displayed when the "**Reconfigurable Block I/O and continue to scan with Master Rack, Block or ASIU failures**" radio button is selected.

Click **OK** to close the message box and return to the dialog to continue editing. The radio button is selected.

Click **Cancel** to close the message box and return to the dialog to continue editing. The radio button is not selected.

4058 HWD: TASK I/O ERROR

This message is displayed when the attempt is made to compile a task with Remote Rack or Block I/O modules declared.

4059 HWD: DOWNLOAD FAILURE CPU TYPE MISMATCH OBSOLETE CPU

A ladder that specifies a valid CPU has been compiled but the CPU in the control is obsolete.

- PiCTurbo2, part number 502-3814-00 with a 486DX processor is not compatible with PiCPro Version 13.0 or later. Use PiCPro Version 12.x or earlier
- All other PiC CPU models with a 186 or 486SX processor, are not compatible with PiCPro version 11.0 or later. Use PiCPro version 10.x or earlier.

4060 HWD: DOWNLOAD FAILURE MMC/MMC PLUS CPU TYPE MISMATCH

An attempt has been made to compile and download a ladder with a MMC PLUS and one or more MMC Analog Servo modules in slots 3 – 6 into a control, which has a MMC CPU.

4061 HWD: CONFIRM SWITCH PIC TO MMC PLUS

When you switch from a PiC CPU to a MMC PLUS CPU, slot 1 will be filled with MMC 2 or 4 axis Analog Servo Module. Slot 2 will be filled with a MMC PLUS In/Out 24V DC Module. Slots 3-6 will be emptied. Slot 7 and up will be removed. All remote I/O will also be removed. Any Block I/O defined will remain. You are asked to confirm this action.

Note: The I/O points defined in Software Declarations should be checked to ensure they are correct.

4062 HWD: CONFIRM SWITCH MMC FOR PC ANALOG SERVO TO MMC PLUS

When you switch from a MMC for PC Analog Servo to a MMC PLUS, slot 1 will be filled with a MMC Analog Servo Module. Slot 2 will be filled with a MMC PLUS In/Out 24V DC module. Slots 3 – 6 will be added. All ASIU I/O will be removed. Any Block I/O defined will remain. You are asked to confirm this action.

Note: The I/O points defined in Software Declarations should be checked to ensure they are correct.

4063 HWD: CONFIRM SWITCH MMC FOR PC DIGITAL SERVO TO MMC PLUS

When you switch from a MMC for PC Analog Servo to a MMC PLUS, slot 1 will be filled with a MMC Analog Servo Module. Slot 2 will be filled with a MMC PLUS In/Out 24V DC module. Slots 3 – 6 will be added. Any Block I/O defined will remain. You are asked to confirm this action.

Note: The I/O points defined in Software Declarations should be checked to ensure they are correct.

4064 HWD: CONFIRM SWITCH MMC PLUS TO MMC

When you switch from a MMC PLUS to a MMC, slot 1 will be filled with a MMC Analog Servo Module. Slot 2 will be filled with a MMC In/Out 24V DC module. Slots 3 and 4 will be emptied if they contain MMC Analog Servo Modules. Slots 5 and 6 will be removed. Any Block I/O defined will remain. You are asked to confirm this action.

Note: The I/O points defined in Software Declarations should be checked to ensure they are correct.

4065 HWD: CONFIRM SWITCH MMC PLUS TO MMC FOR PC

When you switch from a MMC PLUS to a MMC for PC, slot 1 will be filled with a MMC for PC CPU. Slot 2 will be emptied. Slots 3 - 6 will be removed. Any Block I/O defined will remain. You are asked to confirm this action.

Note: The I/O points defined in Software Declarations should be checked to ensure they are correct.

4066 HWD: CONFIRM SWITCH MMC PLUS TO PIC

When you switch from a MMC PLUS to a PiC, slot 1 will be filled with a CSM module. Slot 2 will be filled with a PiC CPU. Slots 3 - 6 will be emptied. Slots 7 – 13 will be added. Any Block I/O defined will remain. You are asked to confirm this action.

Note: The I/O points defined in Software Declarations should be checked to ensure they are correct.

Ladder Error Messages

5000 COMPILE: INVALID TYPE

Data type mismatch. ‘Data type’ supplied. ‘Data type’ required.

The data type of the variable does not match the data type required.

Tip: Enter the correct data type.

5001 COMPILE: DOUBLE DEFINED

'Label': has already been defined as a Label, cannot use again.

The specified label you have entered is already defined in another network. Each label must be unique.

Tip: Delete this label and enter a new label.

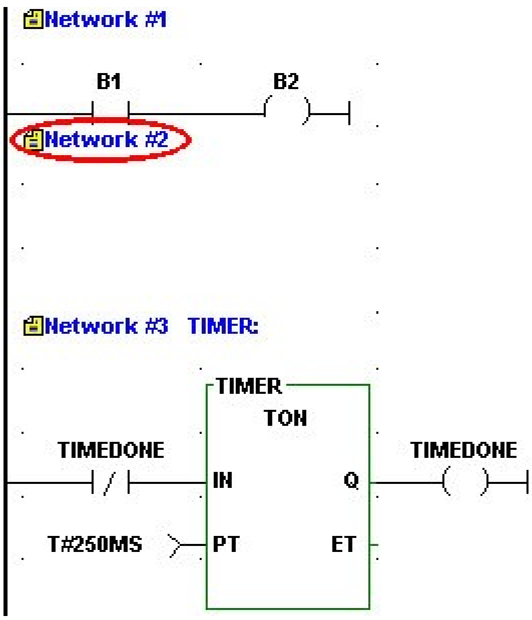
5002 COMPILE: EMPTY NETWORK

Empty network not allowed.

An empty network is not allowed. Delete the empty network if it is not needed.

Example of Error

Network 2 is empty.



Tips:

- Whenever an empty network is included in a ladder, an error message will appear.
- Delete the empty network if it is not required.
- Enter ladder logic.

Note:

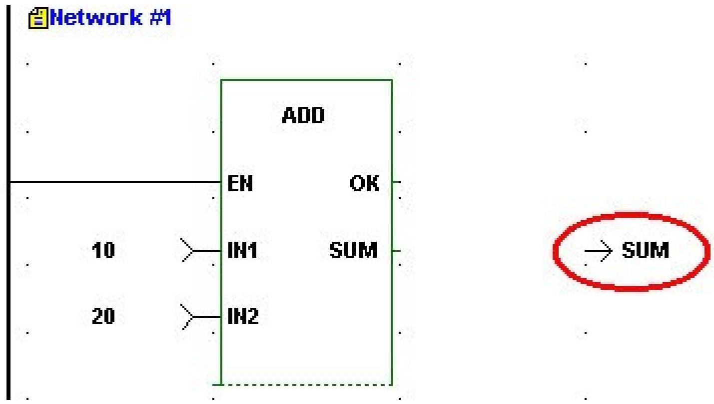
If a network has a label assigned to it but no logic to execute (empty), the label will be ignored and may cause an undefined label error message to appear. If you need the network (i.e. when designing UDFBs), you can prevent this error by simply adding a horizontal wire in the empty network. The error will be avoided and you will be able to successfully compile your UDFB.

5003 COMPILE: CONNECTION ERROR

DATA IN, DATA OUT, or CONSTANT must be directly connected to a function/function block.

Example of Error

The **SUM** DATA OUT variable is not connected to the ADD function.



Tips:

To correct this error, do one of the following:

- Move the cell containing the DATA IN, DATA OUT, or CONSTANT to the function/function block input or output.
- Delete the cell containing the DATA IN, DATA OUT, or CONSTANT.

5004 COMPILE: FUNCTION ON LEFT RAIL

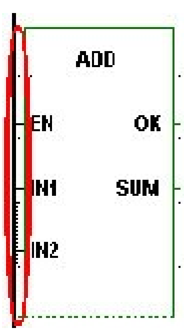
'Function': Function or Function Block cannot be directly connected to left rail.

You cannot connect a function/function block directly to the left power rail.

Example of Error

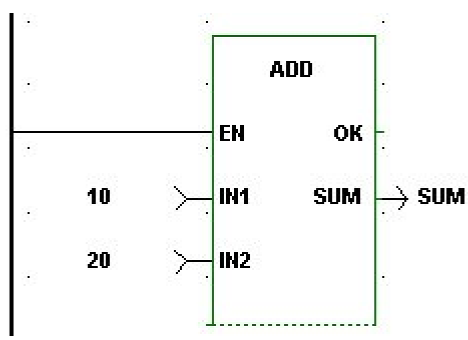
Incorrect

Function incorrectly placed at left power rail.



Correct

Function correctly positioned.



Tip:

Function/function blocks can never be placed in the first column of your network. Reposition the function/function block to column two or greater and make the appropriate connections.

5005 COMPILE: DELETED NETWORK CONTAINS TASK

Deleted Network contains a task and cannot be patched. A scan stopped, full module download is required.

You cannot use the patch feature when you delete a network that contains a task.

Tip:

Whenever you delete a network that contains a task, you must stop the scan and perform a full download.

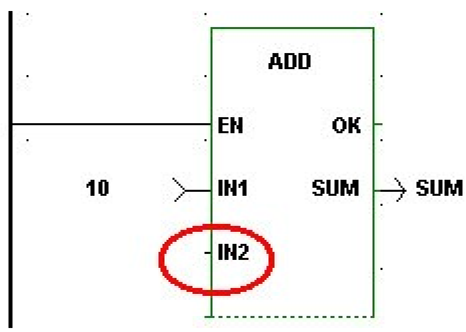
5006 COMPILE: NOT ENOUGH INPUTS

'Function Name': Function requires '#' inputs, only '#' supplied.

You have not supplied the correct number of required inputs for the function you are using.

Example of Error

An input at IN2 is required.



Tip: Ensure that the correct number of inputs has been supplied to the function.

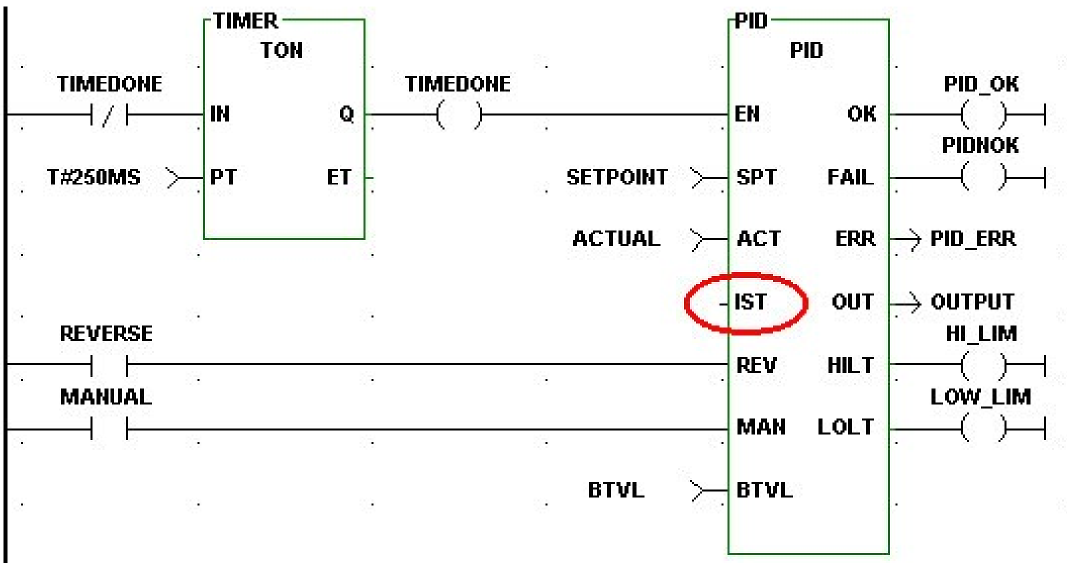
5007 COMPILE: INPUT REQUIRED

Input required

Some function blocks have inputs that are required.

Example of Error

The IST input on the PID function block requires an input and none was entered.



Tip: Enter the required function block input.

5008 COMPILE: NETWORK CONTAINS TASK

Modified network contains a task and cannot be patched. A scan stopped, full download is required.

You cannot use the patch feature when you modify a network that contains a task.

Tip:

Whenever you make changes to a network that contains a task, you must stop the scan and perform a full download.

5009 COMPILE: FUNCTION CONNECTION ERROR

Connection Error. Function or Function Block must be connected to each other by wires.

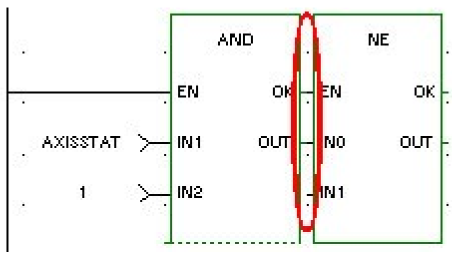
When connecting function/function blocks, you must leave at least one empty column between them and connect the outputs of the first to the inputs of the second with wires or contacts/coils.

Example of Error

Do not place a second function/function block directly next to an existing one.

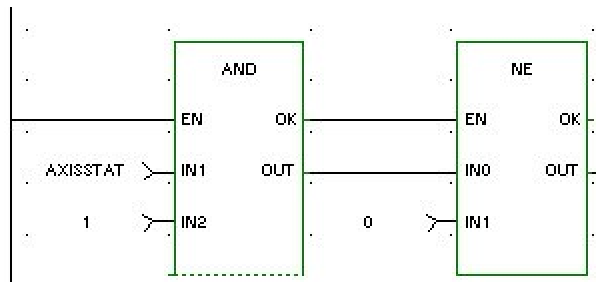
Incorrect

NE function incorrectly placed directly next to the AND function.



Correct

NE function correctly positioned and connected.



Tip:

Function/function blocks cannot be directly connected to each other. Leave a column between them and use wires or contacts/coils to make the required connections.

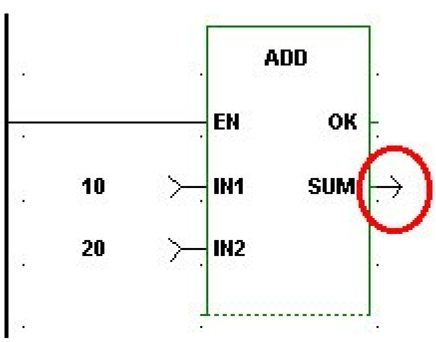
5010 COMPILE: VARIABLE REQUIRED

Variable name required.

DATA IN, DATA OUT, CONTACTS, and COILS all require variables or constants.

Example of Error

The name of the variable at the SUM output is missing.



Tip:

You must enter a variable name declared with the appropriate data type to the DATA IN, DATA OUT, or the CONTACT/COIL location. The DATA IN may have a constant entered in place of a variable name. The data type for the contact/coil variable is always a boolean.

Note:

Under the **View | Options** menu in User Preferences, you can choose to turn on Force Declarations. PiCPro will then prompt you to declare a variable each time you are required to enter one.

5011 COMPILE: DATAOUT CONNECTION ERROR

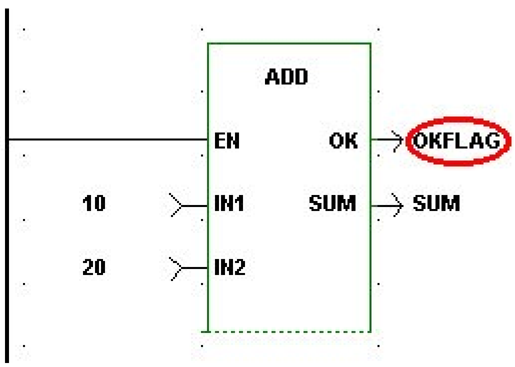
Connection Error. DATA OUT cannot be connected to first output.

DATA OUT cannot be connected to the first output on a function/function block. The first output can only be a contact/coil or a wire or left disconnected.

Example of Error

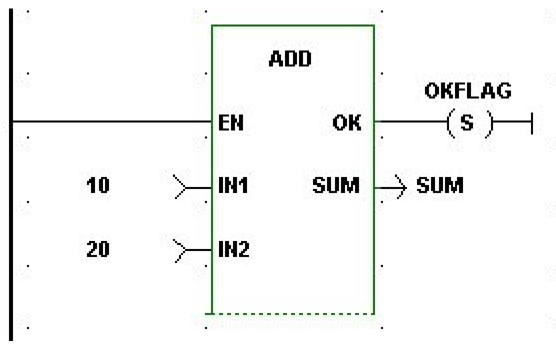
Incorrect

The DATA OUT variable OKFLAG is incorrect.



Correct

The set coil with variable name OKFLAG is an example of a correct connection to the first output of the ADD function.



Tip:

A wire or a contact/coil can be connected to the first output (i.e. the OK) of a function/function block if desired. You may choose not to make any connection to the first output if you do not need to receive the output data.

5012 COMPILE: INVERT DATA TYPE WARNING

Data inversion not allowed on non-boolean data types. Input will not be inverted.

You can only invert Boolean inputs on functions or function blocks. If you attempt to invert a non-boolean input this warning will appear and the input will be treated as a Data In, not as a Data Inverted.

Tips:

- Change input to be a boolean constant, variable, or wire.
- Make the data input a Data In instead of Data Inverted.
- Consider changing the data type of the non-boolean variable to boolean.

5013 COMPILE: BOOLEAN REQUIRED

Incorrect data type specified, boolean required.

A contact or coil has been assigned a variable name that is some other data type than the boolean required.

Tips:

- Enter a valid boolean variable.
- Declare variable as a boolean in the software declarations table.

5014 COMPILE: LABEL REQUIRED

Label required.

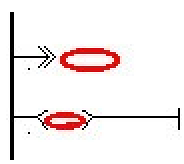
The label of the network you want to jump to with the Jump to Label or Jump to Subroutine commands must be entered with the jump command.

Example of Error

The Jump to Label and the Jump to Subroutine entries require a label of the network the jump is going to.

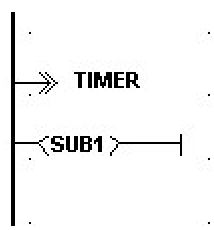
Incorrect

No labels entered for the Jump to Label or Jump to Subroutine commands.



Correct

Labels have been correctly entered with the Jump to Label and Jump to Subroutine commands.



Tips:

Ensure that you have assigned a label to the network you want to jump to.

OR

Remove the Jump to Label or Jump to Subroutine entry.

5015 COMPILE: LABEL UNDEFINED

Label is not defined.

Network labels are required on Jump to Label and Jump to Subroutine commands. If you enter a label that you have not assigned to a network, you will get this error.

Tips:

- Ensure that any network you want to jump to has a label assigned to it.
- If you get this error after entering a label with the jump command, make sure the label exists and/or check the spelling of the label to ensure it matches the label of the destination network.

5016 COMPILE: NPX REQUIRED

Requires a CPU with an NPX processor.

A warning appears when a variable name or constant has been defined with a data type that requires a CPU with an NPX processor. The CPU currently declared in the Hardware Declarations table does not have an NPX processor.

Tips:

Declare a CPU with an NPX processor in the Hardware Declarations table.

OR

Change the data type from REAL, LREAL, LINT, etc. to a data type that does not require a NPX processor.

5017 COMPILE: INVALID BRANCH

Invalid branch.

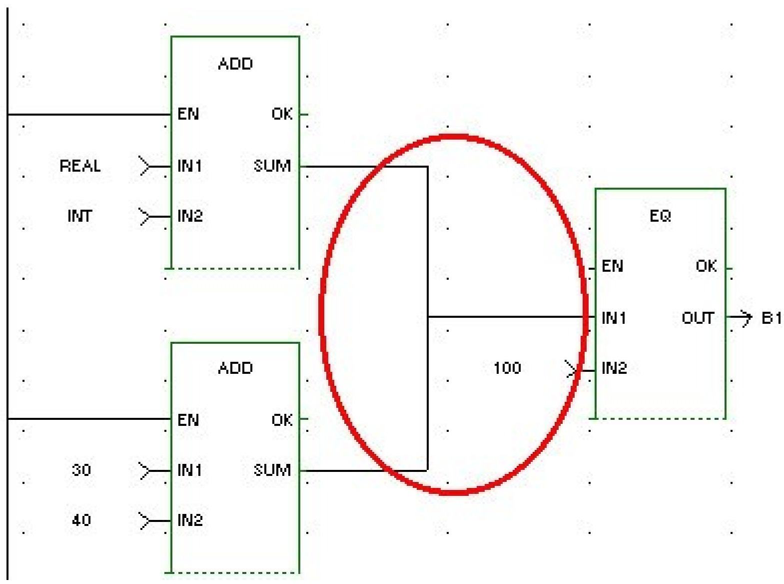
An invalid branch has been attempted. Two or more wires/rungs may form a branch only if the data types being passed on the wire are Boolean.

Example of Error

Incorrect

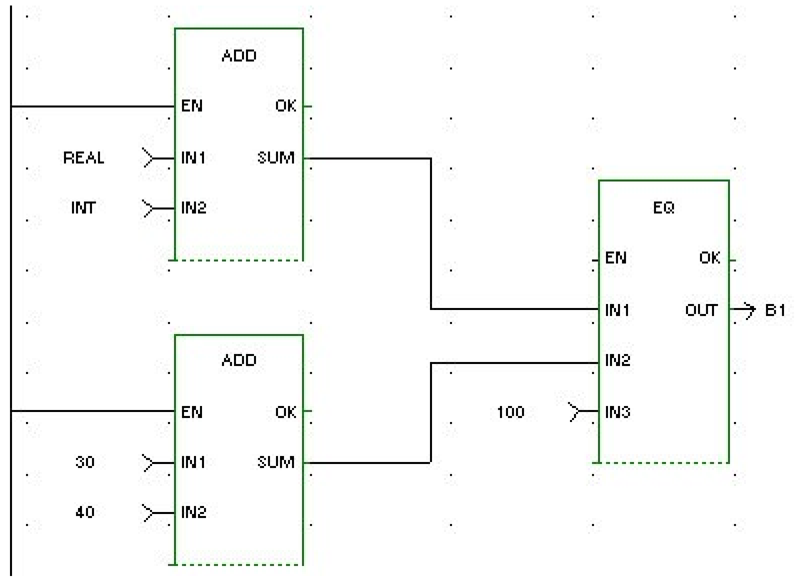
An invalid branch is formed by the wires from the SUM outputs of the ADD functions going to the IN1 input of the EQ function. The SUM outputs are non-boolean data types and cannot be branched this way.

Note: The OK outputs from the ADD functions could form a valid branch since they are both Boolean data types.



Correct

The correct method of branching the above example is shown below.



Tip:

It is valid to branch as shown in the "incorrect" example above if the two branch sources are both boolean data types.

5018 COMPILE: INTERNAL ERROR

An internal error has occurred. Please contact Danaher Motion Tech Support.

Tips:

If you receive an internal error, take the following steps.

- Write down the error number and the exact wording of the error message.
- Note the version of software you are using.
- Save your ladder file.
- Send the above information, your ladder file and all related files to Danaher Motion.

5019 COMPILE: INTERNAL LABEL NOT FOUND

An internal error has occurred. Please contact Danaher Motion Tech Support.

Tips:

If you receive an internal error, take the following steps.

- Write down the error number and the exact wording of the error message.
- Note the version of software you are using.
- Save your ladder file.
- Send the above information, your ladder file and all related files to Danaher Motion.

5020 COMPILE: INTERNAL EMPTY LIST

An internal error has occurred. Please contact Danaher Motion Tech Support.

Tips:

If you receive an internal error, take the following steps.

- Write down the error number and the exact wording of the error message.
- Note the version of software you are using.
- Save your ladder file.
- Send the above information, your ladder file and all related files to Danaher Motion.

5021 COMPILE: INTERNAL UNKNOWN ELEMENT

An internal error has occurred. Please contact Danaher Motion Tech Support.

Tips:

If you receive an internal error, take the following steps.

- Write down the error number and the exact wording of the error message.
- Note the version of software you are using.
- Save your ladder file.
- Send the above information, your ladder file and all related files to Danaher Motion.

5022 COMPILE: INTERNAL NO TEMPLATE

An internal error has occurred. Please contact Danaher Motion Tech Support.

Tips:

If you receive an internal error, take the following steps.

- Write down the error number and the exact wording of the error message.
- Note the version of software you are using.
- Save your ladder file.
- Send the above information, your ladder file and all related files to Danaher Motion.

5023 COMPILE: INTERNAL NO DESTINATION

An internal error has occurred. Please contact Danaher Motion Tech Support.

Tips:

If you receive an internal error, take the following steps.

- Write down the error number and the exact wording of the error message.
- Note the version of software you are using.
- Save your ladder file.
- Send the above information, your ladder file and all related files to Danaher Motion.

5024 COMPILE: INTERNAL INVALID OUTPUT COUNT

An internal error has occurred. Please contact Danaher Motion Tech Support.

Tips:

If you receive an internal error, take the following steps.

- Write down the error number and the exact wording of the error message.
- Note the version of software you are using.
- Save your ladder file.
- Send the above information, your ladder file and all related files to Danaher Motion.

5025 COMPILE: INTERNAL LDO ERROR

An internal error has occurred. Please contact Danaher Motion Tech Support.

Tips:

If you receive an internal error, take the following steps.

- Write down the error number and the exact wording of the error message.
- Note the version of software you are using.
- Save your ladder file.
- Send the above information, your ladder file and all related files to Danaher Motion.

5026 COMPILE: INTERNAL INVALID DATA TYPE

An internal error has occurred. Please contact Danaher Motion Tech Support.

Tips:

If you receive an internal error, take the following steps.

- Write down the error number and the exact wording of the error message.
- Note the version of software you are using.
- Save your ladder file.
- Send the above information, your ladder file and all related files to Danaher Motion.

5027 COMPILE: NO NETWORKS

Ladder does not contain any networks.

The ladder you are attempting to compile does not contain any networks.

Tips:

- Load a ladder that does contain networks.
- Program network(s) into your ladder and compile again.

5028 COMPILE: INVALID ARRAY INDEX

Array index must be either UINT or USINT.

The data type of the index for an array must be either a USINT or UINT.

Tips:

- Use a variable or constant that is a USINT or UINT.
- Change the variable's data type to USINT or UNIT.

5029 COMPILE: PARAMETER ERROR

Parameter error

The parameter does not meet the necessary requirements for this function or function block.

Tip:

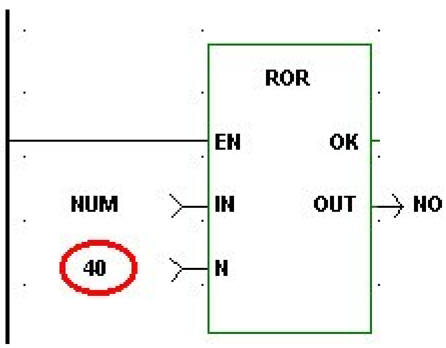
Review the documentation associated with the function/function block you are working with and correct the input.

5030 COMPILE: INVALID CONSTANT

Invalid constant. Constant value must be in the range '#' to '#'.

Example of Error

The constant 40 at the NUM input (declared as a WORD) of the ROR function is out of range for this input. The valid range is from 0 to 31.



Tip:

Ensure that the value you enter as a constant is within the required range. The range is based on the number of bits specified for the data type.

Range if IN data type is less than 8 bytes (BYTE, WORD, or DWORD)	Range if IN data type = 8 bytes (LWORD)
0 - 31	0 - 63

5031 COMPILE: TASK IN FUNCTION

Tasks may not be called from within functions.

You have attempted to call a task from within a UDFB.

Tip: Program a task within the main ladder only.

5032 COMPILE: TOO MANY TASK INPUTS

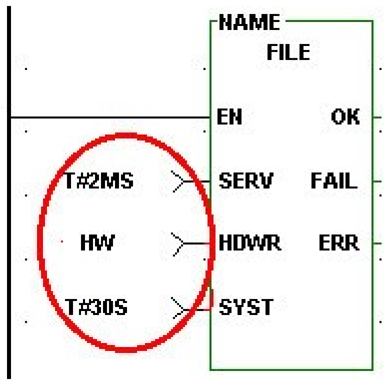
Too many task inputs specified. Task inputs SERV, HDWR, and SYST are mutually exclusive. Only one may be specified.

Task inputs SERV, HDWR, and SYST are mutually exclusive. You may specify only one of these inputs for a task.

Example of Error

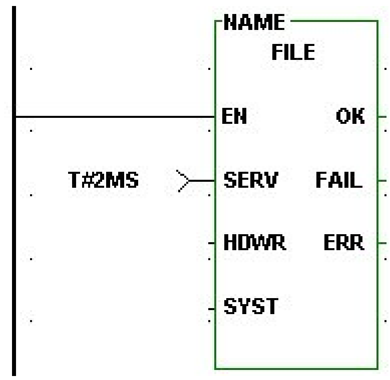
Incorrect

All three TASK inputs (SERV, HDWR, and SYST) have been entered.



Correct

Only one TASK input has been entered.



Tip: Select only one of the three inputs for a task function block.

5033 COMPILE: TASK INPUT REQUIRED

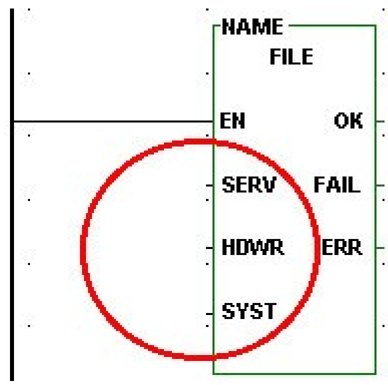
An input is required at one of the task inputs.

You have not connected any input at any of the task inputs of SERV, HDWR, or SYST.

Example of Error

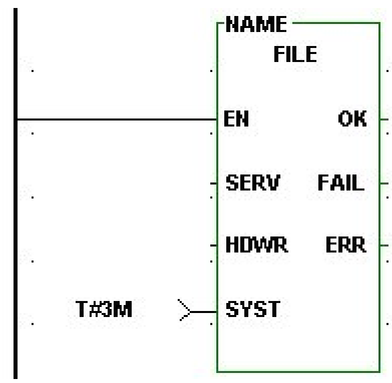
Incorrect

None of the three TASK inputs (SERV, HDWR, and SYST) has been entered.



Correct

One TASK input must be entered.



Tip:

Connect one input on the task function block depending on whether your task is a servo, hardware, or system interrupt task.

5034 COMPILE: DATA IN REQUIRED

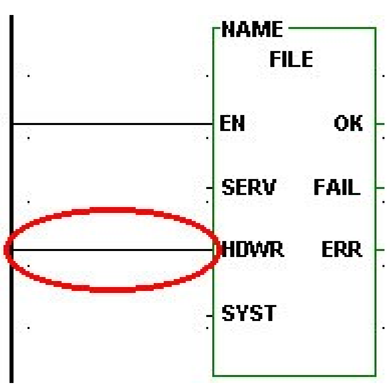
Input must be DATA IN or DATA INVERTED.

When you want to use an I/O point to trigger a hardware interrupt task, the HDWR input of the task function block must be data in or data inverted. Never program a wire or contact for this input.

Example of Error

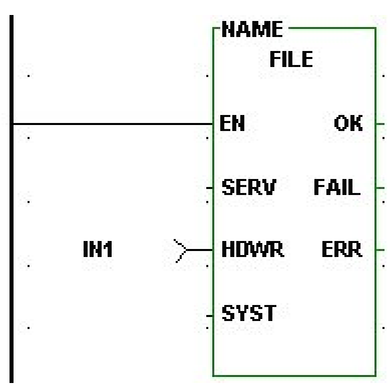
Incorrect

A wire has been connected to the HDWR input of the TASK function block.



Correct

Data In and the variable IN1 have been entered at the HDWR input of the TASK function block.



Tips:

Enter DATA IN or DATA INVERTED for the HDWR input of the task function block.

OR

Convert the wire or contact to a DATA IN or DATA INVERTED.

5035 COMPILE: MAX IO EXCEEDED

The number of function block inputs or outputs exceed 64.

An internal error has occurred. Please contact Danaher Motion Tech Support.

Tips:

If you receive an internal error, take the following steps.

- Write down the error number and the exact wording of the error message.
- Note the version of software you are using.
- Save your ladder file.
- Send the above information, your ladder file and all related files to Danaher Motion.

5036 COMPILE: BAD TASK SERV

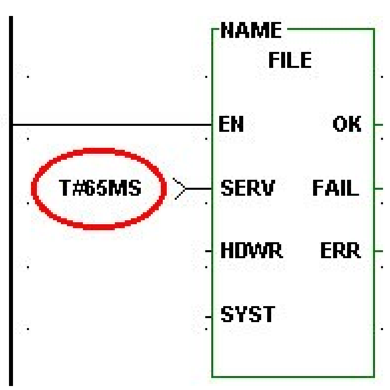
Input must be one of the following constants: T#1MS, T#2MS, T#4MS, T#8MS or T#16MS.

When you want to trigger a servo interrupt task, the SERV input of the task function block must be one of these servo time tick constants: T#1MS, T#2MS, T#4MS, T#8MS or T#16MS.

Example of Error

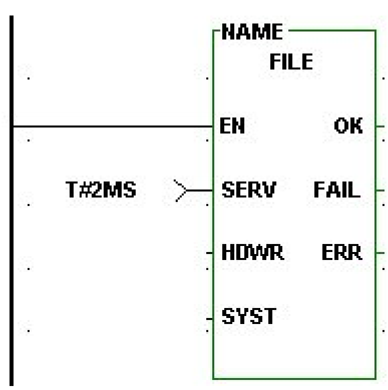
Incorrect

An invalid constant has been entered for the SERV input of the TASK function block.



Correct

One of the valid servo time ticks has been entered at the SERV input of the TASK function block.



Tip:

Enter T#1MS, T#2MS, T#4MS, T#8MS or T#16MS at the SERV input of the task function block.

5037 COMPILE: MAIN RACK ONLY

I/O point must be from main rack only.

Tasks can only access I/O modules located in the main rack.

Tip: Only enter an I/O point from an I/O module in the main rack to trigger a hardware interrupt task.

5038 COMPILE: IO POINT REQUIRED

I/O point required.

When you want to use an I/O point to trigger a hardware interrupt task, you must enter the I/O point at the HDWR input of the task function block.

Tip: Enter an I/O point at the HDWR input of the task function block.

5039 COMPILE: INPUT REQUIRES BOOLEAN VARIABLE

Boolean variable required.

A constant was entered as an input and a boolean variable is required.

Tip: Enter any boolean variable instead of a constant.

5101 COMPILE: INVALID USE OF KEYWORD

Keyword cannot be used as a variable name.

A keyword variable was found in a Structured Text statement. Keywords are not allowed as variable names and cannot be used as variable names in Structured Text statements. Change the name of this variable in software declarations.

Tip: Replace keyword with a variable that is not a keyword.

Refer to: *Reserved Keywords*

5102 COMPILE: SYNTAX ERROR

Syntax Error.

The structured text syntax entered does not match the rules for structured text statements. Verify the statement or expression where the error occurred to make sure that it is syntactically correct.

Tip:

Double click on the error message and examine the statement were the error has occurred. If in an expression make sure that:

- a ")" is not missing,
- an operator is not missing.

If in a structured text statement make sure that an extra keyword or that a keyword was not used out of context.

Correct the syntax error and then recompile.

5103 COMPILE: INVALID ASSIGNMENT

Missing or Invalid assignment statement.

The assignment statement used to form a FOR loop is not syntactically correct.

Tip: Enter a correct assignment statement in the FOR loop.

5104 COMPILE: EXIT OUTSIDE LOOP

EXIT statement used outside of loop.

Exit statements can only be used within an iteration statement.

Tip: Delete the EXIT statement or move within an iteration statement.

5105 COMPILE: NO MATCHING WHILE

END_WHILE without a matching WHILE statement.

An END_WHILE statement was encountered that does not have a matching WHILE statement.

Tip: Delete the END_WHILE statement.

5106 COMPILE: NO MATCHING REPEAT

UNTIL or END_REPEAT without a matching REPEAT statement.

An UNTIL or END_REPEAT statement was encountered that does not have a matching REPEAT statement.

Tip: Delete the UNTIL or END_REPEAT statement.

5107 COMPILE: NO MATCHING FOR

END_FOR without a matching FOR statement.

An END_FOR statement was encountered that does not have a matching FOR statement.

Tip: Delete the END_FOR statement.

5108 COMPILE: NO MATCHING IF

ELSIF, ELSE, or END_IF without a matching IF statement.

An ELSIF, ELSE or END_IF statement was encountered that does not have a matching IF statement.

Tip: Delete the ELSIF, ELSE, or END_IF statement.

5109 COMPILE: UNEXPECTED KEYWORD

Unexpected Keyword found - TO, THEN, DO, NOT, OR, XOR, AND.

Keyword is used out of context.

Tip: Delete the keyword entered or fix the syntax error associated with this keyword.

5110 COMPILE: NESTING ERROR 1

Nesting error - Expecting UNTIL-END_REPEAT before END_WHILE.

Structured Text statements can be nested inside other structured statements. A statement that is nested within another statement must be entirely within the outer most statement. No overlapping of statements are allowed.

Example of Error:

Incorrect:

```
WHILE A DO
   REPEAT
END_WHILE;
UNTIL B END_REPEAT;
```

Correct:

```
WHILE A DO
   REPEAT
   UNTIL B END_REPEAT;
END_WHILE;
```

Tip:

Ensure that the REPEAT statement is entirely within the WHILE statement as the Correct example illustrates above.

5111 COMPILE: NESTING ERROR 2

Nesting error - Expecting END_FOR before END_WHILE.

Structured Text statements can be nested inside other structured statements. A statement that is nested within another statement must be entirely within the outer most statement. No overlapping of statements are allowed.

Example of Error:

Incorrect:

```
WHILE A DO
   FOR I=0 TO 5 DO
END_WHILE;
END_FOR;
```

Correct:

```
WHILE A DO
   FOR I=0 TO 5 DO
   END_FOR;
END_WHILE;
```

Tip:

Ensure that the FOR statement is entirely within the WHILE statement as the Correct example illustrates above.

5112 COMPILE: NESTING ERROR 3

Nesting error - Expecting ELSIF, ELSE, or END_IF before END_WHILE.

Structured Text statements can be nested inside other structured statements. A statement that is nested within another statement must be entirely within the outer most statement. No overlapping of statements are allowed.

Example of Error:

Incorrect:

```
WHILE A DO
   IF B THEN
END_WHILE;
END_IF;
```

Correct:

```
WHILE A DO
   IF B THEN
   END_IF;
END_WHILE;
```

Tip: Ensure that the IF statement is entirely within the WHILE statement as the Correct example illustrates above.

5113 COMPILE: NESTING ERROR 4

Nesting error - Expecting END_WHILE before UNTIL-END_REPEAT.

Structured Text statements can be nested inside other structured statements. A statement that is nested within another statement must be entirely within the outer most statement. No overlapping of statements are allowed.

Example of Error:

Incorrect:

```
REPEAT
   WHILE A DO
UNTIL B END_REPEAT;
END_WHILE;
```

Correct:

```
REPEAT
   WHILE A DO
   END_WHILE;
UNTIL B END_REPEAT;
```

Tip:

Ensure that the WHILE statement is entirely within the REPEAT statement as the Correct example illustrates above.

5114 COMPILE: NESTING ERROR 5

Nesting error - Expecting END_FOR before UNTIL-END_REPEAT.

Structured Text statements can be nested inside other structured statements. A statement that is nested within another statement must be entirely within the outer most statement. No overlapping of statements are allowed.

Example of Error:

Incorrect:

```
REPEAT
   FOR I := 0 TO 5 DO
UNTIL B END_REPEAT;
END_FOR;
```

Correct:

```
REPEAT
   FOR I := 0 TO 5 DO
   END_FOR;
UNTIL B END_REPEAT;
```

Tip:

Ensure that the FOR statement is entirely within the REPEAT statement as the Correct example illustrates above.

5115 COMPILE: NESTING ERROR 6

Nesting error - Expecting ELSIF, ELSE, or END_IF before UNTIL-END_REPEAT

Structured Text statements can be nested inside other structured statements. A statement that is nested within another statement must be entirely within the outer most statement. No overlapping of statements are allowed.

Example of Error:

Incorrect:

```
REPEAT
   IF B THEN
UNTIL B END_REPEAT;
END_IF;
```

Correct:

```
REPEAT
   IF B THEN
   END_IF;
UNTIL B END_REPEAT;
```

Tip:

Ensure that the IF statement is entirely within the REPEAT statement as the Correct example illustrates above.

5116 COMPILE: NESTING ERROR 7

Nesting error - Expecting END_WHILE before END_FOR

Structured Text statements can be nested inside other structured statements. A statement that is nested within another statement must be entirely within the outer most statement. No overlapping of statements are allowed.

Example of Error:

Incorrect:

```
FOR I := 0 TO 5 DO
   WHILE A DO
END_FOR;
END_WHILE;
```

Correct:

```
FOR I := 0 TO 5 DO
   WHILE A DO
   END_WHILE;
END_FOR;
```

Tip:

Ensure that the WHILE statement is entirely within the FOR statement as the Correct example illustrates above.

5117 COMPILE: NESTING ERROR 8

Nesting error - Expecting UNTIL-END_REPEAT before END_FOR.

Structured Text statements can be nested inside other structured statements. A statement that is nested within another statement must be entirely within the outer most statement. No overlapping of statements are allowed.

Example of Error:

Incorrect:

```
FOR I := 0 TO 5 DO
   REPEAT
END_FOR;
UNTIL B END_REPEAT;
```

Correct:

```
FOR I := 0 TO 5 DO
   REPEAT
   UNTIL B END_REPEAT;
END_FOR;
```

Tip:

Ensure that the REPEAT statement is entirely within the FOR statement as the Correct example illustrates above.

5118 COMPILE: NESTING ERROR 9

Nesting error - Expecting END_IF before END_FOR.

Structured Text statements can be nested inside other structured statements. A statement that is nested within another statement must be entirely within the outer most statement. No overlapping of statements are allowed.

Example of Error:

Incorrect:

```
FOR I := 0 TO 5 DO
   IF A THEN
END_FOR;
END_IF;
```

Correct:

```
FOR I := 0 TO 5 DO
   IF A THEN
   END_IF;
END_FOR;
```

Tip:

Ensure that the IF statement is entirely within the FOR statement as the Correct example illustrates above.

5119 COMPILE: NESTING ERROR 10

Nesting error - Expecting END_WHILE before END_IF.

Structured Text statements can be nested inside other structured statements. A statement that is nested within another statement must be entirely within the outer most statement. No overlapping of statements are allowed.

Example of Error:

Incorrect:

```
IF A THEN
   WHILE B DO
END_IF;
END_WHILE;
```

Correct:

```
IF A THEN
   WHILE B DO
   END_WHILE;
END_IF;
```

Tip:

Ensure that the WHILE statement is entirely within the IF statement as the Correct example illustrates above.

5120 COMPILE: NESTING ERROR 11

Nesting error - Expecting UNTIL-END_REPEAT before END_IF.

Structured Text statements can be nested inside other structured statements. A statement that is nested within another statement must be entirely within the outer most statement. No overlapping of statements are allowed.

Example of Error:

Incorrect:

```
IF A THEN
   REPEAT
END_IF;
UNTIL B END_REPEAT;
```

Correct:

```
IF A THEN
   REPEAT
   UNTIL B END_REPEAT;
END_IF;
```

Tip:

Ensure that the REPEAT statement is entirely within the IF statement as the Correct example illustrates above.

5121 COMPILE: NESTING ERROR 12

Nesting error - Expecting END_FOR before END_IF

Structured Text statements can be nested inside other structured statements. A statement that is nested within another statement must be entirely within the outer most statement. No overlapping of statements are allowed.

Example of Error:

Incorrect:

```
IF A THEN
   FOR I := 0 TO 5 DO
END_IF;
END_FOR;
```

Correct:

```
IF A THEN
   FOR I := 0 TO 5 DO
   END_FOR;
END_IF;
```

Tip:

Ensure that the FOR statement is entirely within the IF statement as the Correct example illustrates above.

5122 COMPILE: MISSING CASE

END_CASE without a matching CASE statement.

An END_CASE statement was encountered that does not have a matching CASE statement.

Tip: Delete the END_CASE statement.

5123 COMPILE: NESTING ERROR 13

Nesting error - Expecting END_CASE before END_WHILE.

Structured Text statements can be nested inside other structured statements. A statement that is nested within another statement must be entirely within the outer most statement. No overlapping of statements are allowed.

Example of Error:

Incorrect:

```
WHILE A DO
   CASE A OF
   1..10:
      SPEED := SPEED + 10;
END_WHILE;
END_CASE;
```

Correct:

```
WHILE A DO
   CASE A OF
   1..10:
      SPEED := SPEED + 10;
   END_CASE;
END_WHILE;
```

Tip:

Ensure that the CASE statement is entirely within the WHILE statement as the Correct example illustrates above.

5124 COMPILE: NESTING ERROR 14

Nesting error - Expecting END_CASE before UNTIL-END_REPEAT.

Structured Text statements can be nested inside other structured statements. A statement that is nested within another statement must be entirely within the outer most statement. No overlapping of statements are allowed.

Example of Error:

Incorrect:

```
REPEAT
   CASE A OF
   1..10:
      SPEED := SPEED + 10;
UNTIL A END_REPEAT;
END_CASE;
```

Correct:

```
REPEAT
   CASE A OF
   1..10:
      SPEED := SPEED + 10;
   END_CASE;
UNTIL A END_REPEAT;
```

Tip:

Ensure that the CASE statement is entirely within the REPEAT statement as the Correct example illustrates above.

5125 COMPILE: NESTING ERROR 15

Nesting error - Expecting END_CASE before END_FOR.

Structured Text statements can be nested inside other structured statements. A statement that is nested within another statement must be entirely within the outer most statement. No overlapping of statements are allowed.

Example of Error:

Incorrect:

```
FOR I:=0 TO 5 DO
   CASE A OF
   1..10:
      SPEED := SPEED + 10;
END_FOR;
END_CASE;
```

Correct:

```
FOR I:=0 TO 5 DO
   CASE A OF
   1..10:
      SPEED := SPEED + 10;
   END_CASE;
END_FOR;
```

Tip:

Ensure that the CASE statement is entirely within the FOR statement as the Correct example illustrates above.

5126 COMPILE: NESTING ERROR 16

Nesting error - Expecting END_CASE before END_IF.

Structured Text statements can be nested inside other structured statements. A statement that is nested within another statement must be entirely within the outer most statement. No overlapping of statements are allowed.

Example of Error:

Incorrect:

```
IF A THEN
   CASE A OF
   1..10:
      SPEED := SPEED + 10;
END_IF;
END_CASE;
```

Correct:

```
IF A THEN
   CASE A OF
   1..10:
      SPEED := SPEED + 10;
   END_CASE;
END_IF;
```

Tip:

Ensure that the CASE statement is entirely within the IF statement as the Correct example illustrates above.

5127 COMPILE: NESTING ERROR 17

Nesting error - Expecting END_WHILE before END_CASE or CASE selector.

Structured Text statements can be nested inside other structured statements. A statement that is nested within another statement must be entirely within the outer most statement. No overlapping of statements are allowed.

Example of Error:

Incorrect:

```
CASE A OF
   1..10:
   WHILE B DO
   SPEED := SPEED + 10;
END_CASE;
END_WHILE;
```

Correct:

```
CASE A OF
   1..10:
   WHILE B DO
      SPEED := SPEED + 10;
   END_WHILE;
END_CASE;
```

Tip:

Ensure that the WHILE statement is entirely within the CASE statement as the Correct example illustrates above.

5128 COMPILE: NESTING ERROR 18

Nesting error - Expecting UNTIL-END_REPEAT before END_CASE or CASE selector.

Structured Text statements can be nested inside other structured statements. A statement that is nested within another statement must be entirely within the outer most statement. No overlapping of statements are allowed.

Example of Error:

Incorrect:

```
CASE A OF
   1..10:
   REPEAT
      SPEED := SPEED + 10;
END_CASE;
UNTIL A END_REPEAT;
```

Correct:

```
CASE A OF
   1..10:
   REPEAT
      SPEED := SPEED + 10;
   UNTIL A END_REPEAT;
END_CASE;
```

Tip:

Ensure that the REPEAT statement is entirely within the CASE statement as the Correct example illustrates above.

5129 COMPILE: NESTING ERROR 19

Nesting error - Expecting END_FOR before END_CASE or CASE selector.

Structured Text statements can be nested inside other structured statements. A statement that is nested within another statement must be entirely within the outer most statement. No overlapping of statements are allowed.

Example of Error:

Incorrect:

```
CASE A OF
   1..10:
   FOR I:=0 TO 5
      SPEED := SPEED + 10;
END_CASE;
END_FOR;
```

Correct:

```
CASE A OF
   1..10:
   FOR I:=0 TO 5
      SPEED := SPEED + 10;
   END_FOR;
END_CASE;
```

Tip:

Ensure that the FOR statement is entirely within the CASE statement as the Correct example illustrates above.

5130 COMPILE: NESTING ERROR 20

Nesting error - Expecting END_IF before END_CASE or CASE selector.

Structured Text statements can be nested inside other structured statements. A statement that is nested within another statement must be entirely within the outer most statement. No overlapping of statements are allowed.

Example of Error:

Incorrect:

```
CASE A OF
   1..10:
   IF A THEN
      SPEED := SPEED + 10;
END_CASE;
END_IF;
```

Correct:

```
CASE A OF
   1..10:
   IF A THEN
      SPEED := SPEED + 10;
   END_IF;
END_CASE;
```

Tip:

Ensure that the IF statement is entirely within the CASE statement as the Correct example illustrates above.

5131 COMPILE: MISSING SELECTOR

Missing or expecting CASE selector.

A CASE selector must immediately follow the OF in the case statement.

Example of Error:

Incorrect:

```
CASE A OF
   SPEED := SPEED + 10;
END_CASE;
```

Correct:

```
CASE A OF
   1..10:
   SPEED := SPEED + 10;
END_CASE;
```

Tip: Add missing CASE selector.

5132 COMPILE: MISSING INTEGER

Missing Integer.

In the example below an integer number should appear after the "," in the CASE Selector.

Example of Error:

Incorrect:

```
CASE A OF
   1,:
   SPEED := SPEED + 10;
END_CASE;
```

Correct:

```
CASE A OF
   1,2:
   SPEED := SPEED + 10;
END_CASE;
```

Tip: Either eliminate the extra comma or add and integer number to the CASE selector.

5133 COMPILE: MISSING PARAMETER

Missing Function/Function Block Parameter.

Two commas right next to each other as illustrated in the example below is a syntax error and implies that a function/function block parameter is missing.

Example of Error:

```
MAX(IN1 := A,, IN2 := B);
```

Tip: Remove the extra comma or add another function/function block parameter between the commas.

5134 COMPILE: IMPROPERLY ENTERED CONSTANT

Invalid Hex, Octal or Binary Constant.

An invalid character was used in a Hex, Octal, or a binary constant. See the manual for the proper syntax of these constants.

Example of Error:

```
16#G1AF is an invalid Hex constant because it contains the letter G.
```

Tip:

- Verify that the hex number only contains the digits 0-9 and the letters A-F.
- Verify that the octal number only contains the digits 0-7.
- Verify that the binary number only contains the digits 0-1.

Make appropriate changes.

5135 COMPILE: EXPRESSION SYNTAX ERROR

Expression syntax error.

The structured text syntax entered does not match the rules for structured text statements. Verify the expression where the error occurred to make sure that it is syntactically correct.

Tip:

Double-click on the error message and examine the statement were the error has occurred. Ensure that:

- a ")" is not missing.
- an operator is not missing.

Correct syntax error and then recompile.

5136 COMPILE: MISSING CLOSE COMMENT

Missing closing comment indicator "*)".

A comment start "(*" was encountered that does not have a matching comment end "*)".

Tip: Find the end of comment and add "*)".

5137 COMPILE: EXPECTING INTEGER

Expecting Integer.

Non-Integer constant value entered were an Integer constant value is expected. In the example below, B should be an integer constant. Only integer constants are allowed in a CASE selectors.

Example of Error:

```
CASE A OF
   1, B:
   SPEED := SPEED + 10;
END_CASE;
```

Tip: Enter only an Integer constant or delete the characters that caused the error.

5138 COMPILE: INVALID CASE EXPRESSION

Invalid CASE expression.

The structured text syntax for the entered CASE expression does not match the rules for structured text statements. Verify the expression where the error occurred to make sure that it is syntactically correct.

Tip:

Double click on the error message and examine the statement were the error has occurred. Ensure that:

- a ")" is not missing.
- an operator is not missing.

5139 COMPILE: EXPECTING VARIABLE

Expecting variable to the left of ":=" or "=>".

Example of Error:

```
FOR := 0 TO 5
;
END_FOR;
```

In the above example the assignment variable is missing.

Tip: Add the missing variable or parameter name to the left of the ":=" or "=>".

5140 COMPILE: MISSING FOR ASSIGNMENT

Missing assignment statement that initializes the iteration variable in a FOR loop.

As illustrated in the example below the assignment statement used to initialize the loop counter (LOOP vs. LOOP := 0) is missing or incomplete.

Example of Error:

Incorrect:	Correct:
`FOR LOOP TO 5`	`FOR LOOP:=0 TO 5`
`;`	`;`
`END_FOR;`	`END_FOR;`

Tip: Enter a complete assignment statement for the loop counter.

5141 COMPILE: FUNCTION BLOCK RESYNC

Syntax error in Function/Function block parameter.

The structured text syntax entered does not match the rules for structured text Function /Function Block parameters. Verify the statement or expression where the error occurred to make sure that it is syntactically correct.

Tip:

Double-click on the error message and examine the statement were the error has occurred. The offending characters will be highlighted.

- Ensure that "," follows the Function/Function Block parameter.
- If error occurs right after an expression, verify the expression for accuracy.

Correct syntax error and then recompile.

5142 COMPILE: INVALID ASSIGNMENT

Invalid expression assignment statement.

An internal error has occurred. Please contact Danaher Motion Tech Support.

Tip:

If you receive and internal error, take the following steps:

1. Write down the error number and the exact wording of the error message.
2. Note the version of the software you are using.
3. Save your ladder file.
4. Send the above information, your ladder file and all related files to Danaher Motion.

5143 COMPILE: FUNCTION BLOCK PARAMETER ERROR

Expression must be assigned to a Function/Function Block input or output parameter name using ":=" or "=>".

The input or output parameter used in a function or function block is incomplete. The parameter name and ":=" or "=>" are missing.

Tip: Enter the complete function or function block parameter (i.e., IN0:=A+1 or OUT1=>B).

5144 COMPILE: STATEMENT SYNTAX ERROR

Statement syntax error.

The structured text syntax entered does not match the rules for structured text statements. Verify the statement or expression where the error occurred to make sure that it is syntactically correct.

Tip:

Double-click on the error message and examine the statement were the error has occurred. If in an expression make sure that:

- a ")" is not missing.
- an operator is not missing.

If in a structured text statement, that user makes, an extra keyword does not exist or that a keyword was not used out of context

Correct syntax error and then recompile.

5145 COMPILE: MISSING EXPRESSION

Missing or incomplete expression.

A required expression is missing in a ST statement. In example #1 the assignment statement does not have an expression. In example #2 the MAX function is missing an expression after "IN1 :=".

Example of Error #1

```
A := ;
```

Example of Error #2

```
MAX(IN1:=, IN2:=A, OUT1=>B);
```

Tip: Enter an expression.

5146 COMPILE: NEGATIVE SUBSCRIPT

Array subscripts cannot be negative.

Array subscripts cannot be negative and must be in the range 0 and 998, inclusive. Therefore, the negative sign is not permissible on an array subscript.

Tip: Ensure that the array subscript is between 0 and 998.

5147 COMPILE: NEGATIVE CONSTANT

Unary operator ("-") not allowed on this constant.

The unary operator "-" is not valid on constants that begin with 16# (Hex), 8# (Octal), 2# (Binary), TOD# (Time of Day), DT# (Date and Time), and T# (Time).

Tip: Enter a valid constant value by removing the unary operator "-".

5148 COMPILE: POWER OPERATOR NOT SUPPORTED

The operator Raise to Power ("") is not currently supported.**

The operator Raise to Power ("**") is not currently supported in PiCPro but may be supported in a future release of PiCPro.

Tip: Replace Raise to Power ("**") operator with an expression that performs that same calculation.

5160 COMPILE: NO PARSER INTERFACE

INTERNAL ERROR: Failed to initialize/obtain parse engine interface.

An internal error has occurred. Please contact Danaher Motion Tech Support.

Tip:

If you receive and internal error, take the following steps:

1. Write down the error number and the exact wording of the error message.
2. Note the version of the software you are using.
3. Save your ladder file.
4. Send the above information, your ladder file and all related files to Danaher Motion.

5161 COMPILE: NO GRAMMAR FILE

Cannot continue with compile. Missing internal system file "filename".

An internal error has occurred. Please contact Danaher Motion Tech Support.

Tip: If you receive and internal error, take the following steps:

1. Write down the error number and the exact wording of the error message.
2. Note the version of the software you are using.
3. Save your ladder file.
4. Send the above information, your ladder file and all related files to Danaher Motion.

5162 COMPILE: PARSER SYNCH ERROR

INTERNAL ERROR: Parser is out of sync with parse tree in routine **.**

An internal error has occurred. Please contact Danaher Motion Tech Support.

Tip: If you receive and internal error, take the following steps:

1. Write down the error number and the exact wording of the error message.
2. Note the version of the software you are using.
3. Save your ladder file.
4. Send the above information, your ladder file and all related files to Danaher Motion.

5163 COMPILE: BY 0 NOT ALLOWED

Constant value of 0 not allowed after BY. Value produces infinite loop.

Specifying a constant value of 0 after the BY in a FOR loop is not allowed. Allowing this would cause an infinite loop that would scan loss the control.

Tip: Enter a non-zero constant value after the BY in the FOR loop.

5164 COMPILE: DUPLICATE SELECTOR VALUE OR RANGE

Duplicate selector value or selector range overlaps with a previous selector.

The selector values or range of values used in a CASE statement must be unique and cannot overlap.

Example of Error:

Incorrect:

```
CASE A OF
   2:
   SPEED := SPEED + 20;
   1..10:
   SPEED := SPEED + 10;
END_CASE;
```

Correct:

```
CASE A OF
   2:
   SPEED := SPEED + 20;
   1, 3..10:
   SPEED := SPEED + 10;
END_CASE;
```

Tip: Ensure that no selector values overlap and change accordingly.

5165 COMPILE: SELECTOR RANGE OUT OF ORDER

Selector range values should be from lowest to highest.

When specifying a selector range, the range must be lowest value first then highest value.

Example of Error:

Incorrect:

```
CASE A OF
   10..1:
   SPEED := SPEED + 10;
END_CASE;
```

Correct:

```
CASE A OF
   1..10:
   SPEED := SPEED + 10;
END_CASE;
```

Tip:

Ensure that the range is lowest value first then highest value. Typically, all that needs to be done is to reverse the range values.

5166 COMPILE: FUNCTION MUST RETURN VALUE

Function must return a value to be used in an expression.

Functions that do not return a result, such as STEPCNTL, cannot be called directly from an expression. Functions that fall into this category must be invoked as independent standalone ST instructions that have similar syntax to a function block call.

Tip:

Remove function call from expression and use function block syntax to make function call. Results from function call can then be used in the expression.

5167 COMPILE: TOO MANY FUNCTION OUTPUTS

Functions that contain multiple output values cannot be used in an expression.

Functions such as MOVE and BY2TBOOL have multiple outputs. These functions cannot be called directly from any expression. Functions that fall into this category must be invoked as independent standalone ST instructions that have similar syntax to a function block call. In this case, function output parameters are typically specified.

Tip:

Remove function call from expression and use function block syntax to make function call. Results from function call can then be used in the expression.

5168 COMPILE: FUNCTION UNDEFINED

Function, Function Block, or Function Block title is not defined.

The specified Function, Function Block, instance name or title is not defined.

Tip:

- Check the spelling of the Function, Function Block, instance name or title.
- Ensure that the specified Function or Function Block is in a library (.lib) that is defined in the PiCPro Libraries path.

5169 COMPILE: FUNCTION BLOCK TITLE MISMATCH

"Function Block Title" syntax cannot be used with Functions. Ignoring......

The syntax "Function Block instance name:Function Block title" is only valid with Function Blocks. Specifying this syntax with a Function is an error.

Tip: Remove ":Function Block Title" from the specified function call.

5170 COMPILE: FUNCTION BLOCK IN EXPRESSION

Function Blocks cannot be used within an expression.

Only Functions can be called from an expression. Function Blocks cannot be called from an expression. Function Block calls are independent standalone statements in ST. The results from a Function Block call can be used in any expression.

Tip:

Remove Function Block call from expression and make call to Function Block a standalone ST statement. If necessary, use results from Function Block call in an expression.

5171 COMPILE: FUNCTION PARAMETER ALREADY SPECIFIED

Input/Output parameter was already specified.

A Function or Function block parameter input or output can only be specified once per Function or Function Block. The specified parameter name was already specified for this Function or Function block.

Tip: Remove either of the duplicated parameter names.

5172 COMPILE: FUNCTION I/O PARAMETER NOT DEFINED

Input/Output parameter is not defined for this Function or Function Block.

The specified input or output parameter name is not defined for this Function. Verify this name against the definition.

Tip:

- Check the spelling of input or output name.
- Replace the specified input or output name with a valid input or output name.

5173 COMPILE: FUNCTION OUTPUT NOT ALLOWED

Cannot specify an output parameter on a function used within an expression.

When a Function is specified in an expression, output parameters cannot used. The results from the function call are directly used in the expression.

Tip: Remove all output parameters from this Function call.

5174 COMPILE: NOT DEFINED AS FUNCTION BLOCK

Not defined as a Function Block.

The specified variable is not defined as a function block in software declarations.

Tip:

- Check the spelling of the variable name.
- Replace specified variable with one that is defined as a function block.

5175 COMPILE: BAD FUNCTION BLOCK TITLE

Function Block Title entered does not match definition. Ignoring......

The specified function block title does not match the function block title defined in software declarations for the specified function block instance name.

Tip:

- Change Function Block title so it matches definition in software declarations.
- Ensure that the right function block instance name is specified.

5176 COMPILE: VARIABLE REQUIRES SUBSCRIPT

Variable is defined as an array and requires an array subscript.

The specified variable is defined as an array and requires an array subscript.

Tip:

- Add array subscript to variable.
- Verify spelling of variable name.

5177 COMPILE: STRUCTURE UNDEFINED

Variable is not defined or data type of defined variable is not a structure.

The variable is not defined as a structure in software declarations or is not defined at all.

Tip:

- Check the spelling of the variable name.
- Define the specified variable as a structure.
- Replace specified variable with one that is defined.

5178 COMPILE: VARIABLE UNDEFINED

Variable is not defined.

The specified variable is not defined in software declarations.

Tip:

- Check the spelling of the variable name.
- Define the specified variable.
- Replace specified variable with one that is defined.

5179 COMPILE: VARIABLE NOT ARRAY

Variable is not defined as an array.

The specified variable is used as an array but is not defined as an array in software declarations.

Tip:

- Check the spelling of the variable name.
- Remove the array subscript
- Define the specified variable as an array.
- Replace specified variable with a defined array name.

5180 COMPILE: STRUCTURE ELEMENT UNDEFINED

Structure Element is not defined.

The specified structure element name is not defined in software declarations.

Tip:

- Check the spelling of the element name.
- Define the specified element name.
- Change the specified element name to one that is already defined.

5181 COMPILE: SUBSCRIPT OUT OF RANGE

Constant value for array subscript is out of range. Range values are 0 to 998.

Array constants must be between 0 and 998.

Tip: Ensure that the constant value specified for the array subscript is between 0 and 998.

5182 COMPILE: REQUIRED INPUT NOT SPECIFIED

"Input Parameter Name" - Required input not specified.

There are certain input parameters in a function or function block that are required and are not optional. These parameters must be specified when calling this function or function block.

Tip: Enter the required function or function block parameter.

5183 COMPILE: MISUSED RELATIONAL OPERATOR

A logical operator (OR, XOR, or AND) must be used to separator expressions containing different relational operators.

Expressions such as:

```
A < B < C < D
```

Is legitimate in our implementation of ST in PiCPro, while the expression:

```
A < B < C > D
```

Is invalid in our implementation of ST in PiCPro. To correct, the second expression should be rewritten as:

```
A < B < C AND C > D
```

The basic rule of thumb is all operands in a relational expression must use the same relational operator. If more than one relational operator is required to form the expression then a logical expression must be used.

Tip: Separate into two or more relational expressions and separate with a logical operator.

5184 COMPILE: BAD CONSTANT

Invalid constant; value too large or improperly formed.

The constant entered is either syntactically incorrect or is too large. For more information about constant syntax rules see the manual.

Tip: Ensure that the constant entered is syntactically correct and within the range of the data type.

5185 COMPILE: CANNOT INTERMIX OPERATORS

Cannot intermix logical operators with arithmetic operators. Resultant data types do not match.

Example of Error:

```
D := A + 5 AND B + C;
```

If A and B are defined as integers, then this statement does not make logical sense and results in this error.

Tip:

Ensure that logical operators and arithmetical operators do not get intermixed in the same expression.

5186 COMPILE: INTERNAL PARSER ERROR

INTERNAL ERROR: In first Pass of Structure Text. Report error code - ##

An internal error has occurred. Please contact Danaher Motion Tech Support.

Tip:

If you receive and internal error, take the following steps:

1. Write down the error number and the exact wording of the error message.
2. Note the version of the software you are using.
3. Save your ladder file.
4. Send the above information, your ladder file and all related files to Danaher Motion.

5187 COMPILE: MISSING TERM

Missing "term"

In parsing a ST network the expected next term was not present.

Example:

```
WHILE DO
```

Note:

In the example above that the expression in the WHILE is missing and the END_WHILE statement is missing. This example will generate 2 errors: Missing "expression" and Missing "END_WHILE".

Tip: Review the syntax of the statement that has the error and then add in the missing term.

5188 COMPILE: BAD ERROR NUMBER

INTERNAL ERROR: Error number out of range: %s.

An internal error has occurred. Please contact Danaher Motion Tech Support.

Tip:

If you receive and internal error, take the following steps:

1. Write down the error number and the exact wording of the error message.
2. Note the version of the software you are using.
3. Save your ladder file.
4. Send the above information, your ladder file and all related files to Danaher Motion.

5189 COMPILE: NETWORK CODE SIZE EXCEEDED

Exceeded network code size limit. Limit is 4095, Actual is %s.

The binary code generated for any given network within a ladder is limited to 4095 bytes. If you receive this error the offending network must be split into 2 or more networks.

Tip: Split network into 2 or more networks.

5190 COMPILE: VARIABLE NAME TOO LONG

Variable names are limited to 63 characters.

A variable name as defined in software declarations is limited to 63 characters. The variable name entered contains more than 63 characters. See the manual for the syntax rules of a variable name.

Tip: Reduce the size of the variable name entered to 63 characters or less.

5191 COMPILE: FUNCTION NAME TOO LONG

Function name or Function Block title is limited to 8 characters.

Function names and function block titles are limited to 8 characters. The name entered contains more than 8 characters.

Tip: Reduce the size of the Function name or Function Block title to 8 characters or less.

5192 COMPILE: INVALID FUNCTION BLOCK NAME

Function Block instance name contains an invalid character.

Function Block instance names can only consist of characters that are alphanumeric or contain an "_" (underscore). Characters not part of this set cause this error. For more information about Function Block instance name syntax rules see the manual.

Tip: Delete the invalid characters and verify that the Function Block instance name is not misspelled.

5193 COMPILE: UNDEFINED FUNCTION BLOCK INSTANCE NAME

Function Block variable/instance name is not defined.

The specified Function Block instance name is not defined in software declarations.

Tip:

Verify that Function Block instance name is not misspelled. If misspelled correct the spelling otherwise define the Function Block instance name in software declarations.

5194 COMPILE: FUNCTION INPUT NOT ALLOWED

Function input parameter "EN" is not allowed in a ST network function call.

When invoking a Function in a ST network, the input parameter enable (i.e., EN) should not be specified. Functions in a ST network always execute unless there is conditional code around the Function call.

Example:

```
IF INPUT1 THEN
   B := MAX(IN1:=C, IN2:=D);
END_IF;
```

Tip:

Delete this parameter from the Function call and if necessary add a conditional statement around the Function call to conditionally execute the function.

5195 COMPILE: COUNTDOWN FOR NOT ALLOWED

Countdown FOR loop is not allowed when data type is unsigned.

Countdown FOR loops are not permissible when the data type specified for the counter is variable USINT, UINT, UDINT, or ULINT.

Tip:

- Change the data type used for the counter in the FOR loop to be SINT, INT, DINT, or LINT.
- Change the conditions of the FOR loop to count up versus counting down.
- Change the constant after BY to be positive.
- Consider using a WHILE loop instead of a FOR loop. Most FOR loops can be easily converted into a WHILE loop.

7002: FUNCTION REQUIRES NPX

Function _ requires numeric coprocessor

A warning appears when a variable name or constant has been defined with a data type that requires a CPU with an NPX processor to process and the CPU declared in the Hardware Declarations table does not have an NPX processor.

Tips:

- Declare a CPU with an NPX processor in the Hardware Declarations table.
- Change the data type from REAL, LREAL, LINT, etc. to a data type that does not require a NPX processor.

7003: LIBRARY NOT FOUND

"_" was not found.

When PiCPro scanned the libraries, it found a library that it cannot now open to retrieve functions for compiling. Possibly the library has been deleted or renamed with Windows Explorer while PiCPro was running. The library has to be found for the compile to be successful.

Tips:

- Check the library paths.
- Go to the libraries dialog and click OK to rescan the libraries.
- Do not make any changes to the library paths or directories when PiCPro is running.

7004: TASK IO CONFLICT

Task "_" contains an I/O board conflict.

The I/O used by this task cannot be different than the I/O specified by the main module.

Tip:

Compare the hardware declarations for this task module and for the main module and ensure that the I/O declared is the same.

9001 COMPILE: NPX REQUIRED

Incompatible Variable _/_/_. Use of _ type requires a CPU with an NPX processor.

A warning appears when a variable name or constant has been defined with a data type that requires a CPU with an NPX processor to process and the CPU declared in the Hardware Declarations table does not have an NPX processor.

Tips:

- Declare a CPU with an NPX processor in the Hardware Declarations table.
- Change the data type from REAL, LREAL, LINT, etc. to a data type that does not require a NPX processor.

9002 COMPILE: NEW STRUCTURE MEMBER

'__.__' is a new structure member that cannot be properly assigned. A scan stopped, full module download is required.

If you add a new member to an existing structure, you must perform a full download with the scan stopped in order for PiCPro to recognize the new member.

9003 COMPILE: MODIFIED STRUCTURE MEMBER

'__.__' is an existing structure member that has been modified. A scan stopped, full module download is required.

If you modify an existing structure member, you must perform a full download with the scan stopped in order for PiCPro to recognize the modification.

9004 COMPILE: FUNCTION MODIFIED

'__' is an existing function block instance that has been modified. A scan stopped, full module download is required.

If you modify a function block, you must perform a full download with the scan stopped in order for PiCPro to recognize the modification.

9005 COMPILE: INVALID ATTRIBUTE

Invalid symbol attribute in '__'.

There is an invalid symbol attribute in the software declarations table.

Tip: Ensure that the attributes you assign to any variable in the software declarations table is valid.

9006 COMPILE: SYMBOL MODIFIED

'__' is an existing variable that has been modified. A scan stopped, full module download is required.

If you modify an existing variable, you must perform a full download with the scan stopped in order for PiCPro to recognize the modification.

Tip: Always perform a full module download with the scan stopped when you modify an existing variable.

9007 COMPILE: NEW RETAINED

'_____' is a new RETAINED variable that cannot be properly assigned. A scan stopped, full module download is required.

If you enter a new variable with the retained attribute, you must perform a full download with the scan stopped in order for PiCPro to recognize the new retained variable.

9008 COMPILE: FUNCTION INITIALIZATION REQUIRED

'_____' calls for initialization. A remake of this library function is required, along with a full module download.

If you make changes that require initialization i.e. change initial values, add function/blocks, declarations, (strings always require initialization), you must recompile the function block and perform a full module download.

Tip: Whenever a change is made that requires initialization, you must:

- Recompile the function block.
- Download the module.

9009 COMPILE: NEW/MOVED INPUT/OUTPUT

A new/moved input/output '____' cannot be added in a patch. A remake of this library is required, along with a full download.

You have attempted to patch the module after adding a new or moving an existing input or output. This requires that you recompile the module and perform a full download.

Tip: Select **Compile | Bin File | Compile & Download** from the menu.

9010 COMPILE: I/O NOT AN INPUT

'_____' calls for an input point '____' in rack '____' slot '____' which is not configured.

The hardware configuration defined by the hardware declarations table is compared to the software configuration when a module is compiled. PiCPro will detect an error if you attempt to define an input point at an output location.

Tip: Ensure that discrete I/O points declared in software match the hardware declarations.

9011 COMPILE: I/O NOT AN OUTPUT

'_____' calls for an output point '____' in rack '____' slot '____' which is not configured.

The hardware configuration defined by the hardware declarations table is compared to the software configuration when a module is compiled. PiCPro will detect an error if you attempt to define an output point at an input location.

Tip: Ensure that discrete I/O points declared in software match the hardware declarations.

9012 COMPILE: CANNOT FIND FUNCTION

Unable to locate function block '____' in the function/block libraries.

The function block cannot be found in the function/block libraries.

Tip: Compile the function block and designate the library it should be stored in.

9013 COMPILE: INSUFFICIENT DATA MEMORY

Out of data memory. Out of memory error. Your program requires more memory than is available in the Control CPU.

You have reached the memory limits on your current system.

9014 COMPILE: INPUT NOT ALLOWED

'_____', input not allowed in TASK.

The input you have entered is not allowed in a TASK.

Tip: Do not mark any variables in the software declarations table with the Variable In attribute in a task LDO.

9015 COMPILE: OUTPUT NOT ALLOWED

'_____', output not allowed in task.

The output you have entered is not allowed in a task.

Tip: Do not mark any variables in the software declarations table with the Variable Out attribute in a task LDO.

9016 COMPILE: FIRST IN IS NOT A BOOLEAN

'_____', the first input, is not a BOOLEAN.

UDFBs require that the first input be a Boolean.

Tip: Enter the BOOL data type in the software declarations table for the first input to your UDFB.

9017 COMPILE: FIRST OUTPUT IS NOT A BOOLEAN

'_____', the first output, is not a BOOLEAN.

UDFBs require that the first output be a Boolean.

Tip: Enter the BOOL data type in the software declarations table for the first output to your UDFB.

9018 COMPILE: INVALID FUNCTION INPUT

'_____' is an invalid function INPUT.

Inputs to UDFB can be any data type except function blocks.

9019 COMPILE: INVALID FUNCTION OUTPUT

'_____' is an invalid function output.

Outputs to UDFBs can be any data type except function blocks, structures, arrays, and strings.

9020 COMPILE: EXTERNAL INITIAL VALUE IGNORED

Initial value for EXTERNAL '_____' is ignored.

An initial value was entered for a variable with an EXTERNAL attribute. PiCPro ignores that value.

Tip:

Since UDFBs cannot have any variables marked EXTERNAL, do not apply this attribute to any UDFB variable, nor enter an initial value for it.

9022 COMPILE: NO BOOLEAN INPUTS

No UDFB Variable In attributes.

The first input to the UDFB must be a boolean and it must be assigned the Variable In attribute in the software declarations table.

Tip: When creating an UDFB, ensure in the software declarations table that the first input:

- Is a boolean.
- Is assigned the Variable In attribute.

9023 COMPILE: TOO MANY INPUTS

Too many UDFB Variable In attributes.

The total number of inputs and outputs for any UDFB is 64. You have exceeded that number.

Tip:

It is recommended that you keep the number of inputs and outputs to the UDFB to a minimum (under 16). More can be declared if necessary, but transferring all the inputs and outputs to and from a function block does use scan time. There is also the constraint of the 255-element matrix to consider. If you have a large number of inputs, you may want to enter them as a structure using just one input.

9025 SWD: INSERT NEW SYMBOL

If you insert a new symbol into your ladder that has not previously been declared in the software declarations table, a message will ask if you want to add the new symbol now. If you choose **Yes**, the software declarations table appears and you can insert the new symbol. If you choose **No**, you will not be able to add the undeclared symbol to your ladder until it has been declared.

9026 SWD: AUTO INSERT

This symbol cannot be inserted into a structure.

When you have entered a variable name in your ladder and are prompted to enter it in the software declarations table, you cannot insert the new variable in software declarations if the focus is on a structure member or on END_STRUCT.

Tip: Move the focus anywhere else in the table or to End List and press the **Insert** key.

9027 SWD: DIRECT I/O

Direct I/O points cannot be assigned in the software declarations table in the following situations:

- If the variable is a member of a structure
- If the variable has an initial value

9028 SWD: INITIAL VALUE

Initial values cannot be assigned in the software declarations table in the following situation:

- If the variable has an I/O point assigned to it.

9029 SWD: VAR IN

You have attempted to assign the Variable In attribute to a symbol that cannot accept it. The Variable In attribute cannot be assigned to the following:

- To the member of a structure (Attributes can only be defined for the entire structure)
- To a variable with the function block type
- To a variable with an I/O point assigned

9030 SWD: VAR OUT

You have attempted to assign the Variable Out attribute to a symbol that cannot accept it. The Variable Out attribute cannot be assigned to the following:

- To the member of a structure (Attributes can only be defined for the entire structure.)
- To a variable with the function block, structure, string, or array type
- To a variable with an I/O point assigned

9031 SWD: FUNCTION

You have attempted to assign the Function data type to a symbol that cannot accept it. The Function data type cannot be assigned to the following:

- To the member of a structure

9032 SWD: STRUCTURE

You have attempted to assign the Structure data type to a symbol that cannot accept it.

9033 SWD: ARRAY

You have attempted to assign the Array data type to a symbol that cannot accept it.

9034 SWD: TYPE CHANGE

If you attempt to change the data type of a symbol with an initial value to an incompatible data type, an error message will appear.

9035 SWD: RETENTIVE

You have attempted to assign the Retentive attribute to a symbol that cannot accept it. The Retentive attribute cannot be assigned to the following:

- To a variable with an I/O point assigned

9036 SWD: NAME FORMAT

You have either left a variable unnamed or assigned a duplicate name. Every variable entered in your ladder must have a unique name assigned to it in the software declarations table.

9037 SWD: DIRECT I/O FORMAT

Direct I/O must be entered in the software declaration table as Boolean data type. The address format must follow the conventions for master rack, expansion rack, block I/O, or ASIU I/O as listed below.

- General PiC CPU
- Master Rack, PiC CPU
- Expansion Rack I/O - PiC CPU
- Master Rack Standalone MMC CPU
- ASIU I/O for MMC for PC CPU
- Block Expansion Rack I/O
- Blown Fuse Status
- Short Circuit Detection
- Fast Inputs - PiC CPU
- Fast Inputs - Standalone MMC CPU
- Fast Inputs - MMC for PC CPU
- MMC Digital I/O Points

9038 SWD: ARRAY FORMAT

Only variables in the software declarations table that are not yet referenced in the ladder can be made into arrays. The size of an array must be between 2 and 999 elements. Function block data type cannot be made into an array.

9039 SWD: COMPLEX NAME FORMAT

If the format of a complex name is incorrect, an error message appears.

9042 SWD: FIND NAME FORMAT

When using the Find/Find Next command in the software declarations table, the following applies:

- When searching by Name, you can enter the structure or member name, but not an element of an array or an array of structures name.
- When Whole Name match is selected, an entry is required in the Name field.\

9044 SWD: INITIAL VALUE LIMITS

You have exceeded the limits on initial values in the software declarations table. The limits are:

Data Type	Minimum Value	Maximum Value
BOOL	0	1
BYTE	0	255
DATE	D#1988-1-1	D#2051-12-31
DATE_AND_TIME	DT#1988-1-1-00:00:00	DT#2051-12-31-23:59:59
DINT	-2,147,483,648	2,147,483.647
DWORD	0	4,294,967,295
FUNCTION BLOCK	N/A	N/A
INT	-32,768	32,767
LINT	-9,223,372,036,854,775,808	9,223,372,036,854,775,807
LREAL	*	*
LWORD	0	18,446,744,073,709,551,615
REAL	*	*
SINT	-128	127
STRING	0	255 ASCII characters
STRUCT	N/A	N/A
TIME	0	T#49d17h2m47s294ms T#1193h2m47s294ms T#71582m47s294ms T#4294967s294ms T#4294967294ms
TIME_OF_DAY	TOD#00:00:00	TOD#23:59:59
UINT	0	65,535
UDINT	0	4,294,967,295
ULINT	0	9,223,372,036,854,775,807
USINT	0	255
WORD	0	65,535

*Validation done for invalid characters.

9045 SWD: STRING FORMAT

You cannot change the length of a string in the software declarations table to a value that is shorter than the length of any initial values entered.

9046 SWD: SAVE CHANGES

All your changes to the software declarations table will be lost if you do not save before exiting.

9047 SWD: FORCE LIST ENTRY LIMITATIONS

Entries in the Force List must be explicit. For example, to enter an element of an array, enter name (3), not name (index).

9048 SWD: STRING LENGTHS EXTENDED

If you edit the initial values for an array of strings and any of the strings are now longer than the declared string length, these string lengths will be automatically extended.

9049 SWD: STRING LENGTH EXTENDED

If you edit the initial values for a string and the string is now longer than the declared string length, the string length will be automatically extended.

9051 COMPILE: OI LIBRARY NOT AVAILABLE

The ASFB file for storing control Operator Interface information could not be located in the library directories.

If you are using the Operator Interface feature, you must have the Operator Interface ASFBs installed. They are in the opinter.lib supplied by Danaher Motion.

Tip: Check the following:

- The Operator Interface ASFBs have been installed on your workstation.
- The path to the opinter.lib containing the ASFBs is defined.

9052 COMPILE: INVALID STRING SPECIFIED

The string you have specified is invalid.

9053 COMPILE: UDFB DISCRETE IO

Discrete I/O cannot be declared in UDFB.

You cannot declare discrete I/O points in the software declarations table of a UDFB.

Tip:

Even though you cannot use discrete I/O in the finished UDFB, you may need to add discrete I/O in order to test your UDFB. If you do this, be sure to remove all discrete I/O before you compile the UDFB.

9054 COMPILE: NO BOOLEAN OUTPUTS

No UDFB Variable Out attributes.

The first output to the UDFB must be a Boolean and it must be assigned the Variable Out attribute in the software declarations table.

Tip: When creating a UDFB, ensure in the software declarations table that the first output:

- Is a boolean.
- Is assigned the Variable Out attribute.

9055 COMPILE: TOO MANY OUTPUTS

Too many UDFB Variable Out attributes.

The total number of inputs and outputs for any UDFB is 64. You have exceeded that number.

Tip:

It is recommended that you keep the number of inputs and outputs to the UDFB to a minimum (under 16). More can be declared if necessary, but transferring all the inputs and outputs to and from a function block does use scan time. There is also the constraint of the 255-element matrix to consider. If you have a large number of inputs, you may want to enter them as a structure using just one input.

9056 COMPILE: BLOCK I/O NOT INPUT

'_____' calls for an input point '_____' in block '_____' which is not configured.

The hardware configuration defined by the hardware declarations table is compared to the software configuration when a module is compiled. PiCPro will detect an error if you attempt to define an input point at an output location.

Tip: Ensure that discrete I/O points declared in software match the hardware declarations.

9057 COMPILE: BLOCK I/O NOT OUTPUT

'_____' calls for an output point '_____' in block '_____' which is not configured.

The hardware configuration defined by the hardware declarations table is compared to the software configuration when a module is compiled. PiCPro will detect an error if you attempt to define an output point at an input location.

Tip: Ensure that discrete I/O points declared in software match the hardware declarations.

9058 COMPILE: INVALID UDFB ATTRIBUTE

Invalid symbol attribute in '_____'.

There is an invalid symbol attribute in the software declarations table for the UDFB.

Tip:

Ensure that the attributes you assign to any variable in the software declarations table is valid. For variables that will be inputs or outputs to the UDFB, attributes may not be external, retained, global, or discrete I/O.

9059 SWD: MODIFY NAME USED IN LADDER

When you attempt to modify the name of a symbol used in your ladder, this warning/confirmation message appears. Changing the name of a symbol means that every occurrence of the name in your ladder will be changed.

9060 SWD: DELETE

You can delete any selected item from the software declarations table that is not used in your ladder. If you want to delete a structure, you must be sure to select the entire structure.

9061 SWD: STRUCTURE ATTRIBUTES

You have attempted to add an attribute to a member of a structure.

9062 SWD: PASTE IN STRUCTURE

You cannot insert a function block, structure, or a symbol with a direct I/O point into a structure.

9063 SWD: SYMBOL EXISTS TYPE INVALID

You cannot name a ladder element with the name of an existing symbol whose data type is invalid for this ladder element.

9064 SWD: INVALID ARRAY COUNT EDIT

You cannot enter a name in your ladder that is an array without entering the array index. Conversely, you cannot enter a name with an array index if the array does not exist.

9065 SWD: INVALID CONSTANT EXPRESSION

The constant you entered is invalid.

9066 SWD: INVALID CONSTANT TYPE

The constant you entered is valid but the data type of the constant is not the data type required.

9067 SWD: FUNCTION BLOCKS UNAVAILABLE

The function block menu cannot be displayed from within the software declarations table.

9082 SWD: TOO BIG FOR SEGMENT

STRUCT or STRING too big, must be less than 64K.

A structure or string variable declared in Software Declarations exceeds the maximum size allowed (64K bytes).

Double clicking on the error in the information window will reposition you to the declaration in software declarations that is causing the problem. Edit the structure or string variable in Software Declarations and recompile.

9083 SWD: INSUFF DATA MEMORY

Out of Data Memory. A scan stop and a full compile and download is required.

This error occurred while patching in new software declarations variables and extended data memory is being used for this ladder.

There are a total of 3 data segments. They are only used as needed. Patching of declarations can only occur until the current segment is full. If the current segment becomes full when you patch in new declarations, you'll get this message. Then you must do a full compile and download of the ladder. If a full compile and download does not fix the problem, all three segments are full and you must optimize your usage of data memory by consulting with Danaher Motion Tech Support.

9086 SWD: INVALID NAME SPECIFIED

" Variable name " cannot be inserted. It is not a valid variable name.

Variable names are limited to 63 characters where the first character is an alpha character and the remaining characters are alphanumeric or underscores.

9090 COMPILE: DUPLICATE INPUTS NOT ALLOWED

Input: The first four characters must be different from other inputs.

When compiling a UDFB, the input names are truncated to four characters. The truncated input must be different from the other truncated inputs. Double-click on the error in the information window to display that variable in software declarations. Rename the variable to make the first four characters of the variable name different from the other inputs.

9091 COMPILE: DUPLICATE OUTPUTS NOT ALLOWED

Output: The first four characters must be different from other outputs.

When compiling a UDFB, the output names are truncated to four characters. The truncated output must be different from the other truncated outputs. Double-click on the error in the information window to display that variable in software declarations. Rename the variable to make the first four characters of the variable name different from the other outputs.

Library Error Messages

1033 BINARY: PIC LIBRARY DIFFERENCE

Function/block LEVEL 1 version difference.

If you attempt to patch a UDFB network after completing a full download and then editing and recompiling the UDFB, you will get this error.

Tip:

You must perform a full download whenever you want to change a network containing a UDFB that has been edited and recompiled since the previous full download.

SERCOS Error Messages

Ring Error Codes

The ring errors listed below appear at the ERR output of the SCR_ERR function and will appear on the **Ring Error State** line in the **Ring State** dialog.

ERR#	Description	What to do/check
3	Axis is not initialized, is not a SERCOS axis, or the slot/ring/slave specification is incorrect.	• SERCOS board in correct slot • SR structure members correct
17	The SERCOS module did not receive an expected AT response. Cable could be disconnected.	• Check connection
20	Phase 0 detected that the ring is not complete.	• Check connection • Ensure drive is turned on

65	Error occurred calculating when MDT should occur.	• Too many slaves on one ring • One or more drives cannot accommodate required MDT
66	Error occurred calculating when drive data valid.	• Too many slaves on one ring • One or more drives cannot accommodate command times
67	Error occurred calculating when feedback data valid.	• Too many slaves on one ring • One or more drives cannot accommodate feedback capture times
68	Error occurred calculating total time required for communication cycle	• Too many slaves on one ring • Cyclic data on slaves too long • Update rate too fast
69	Error occurred calculating cyclic data memory for SERCON processor.	• Too many slaves on one ring • Cyclic data on slaves too long
70	Error occurred calculating cyclic data memory for internal memory map.	• Too many slaves on one ring • Cyclic data on slaves too long
71	Error occurred calculating service channel memory map.	• Too many slaves on one ring • Cyclic data on slaves too long
74	CPU on SERCOS module has too many tasks during update.	• Too many slaves on one ring • Cyclic data on slaves too long
128	Slave error occurred. Read SERR output to identify error. The SLV output indicates the slave number.	• SLV output contains slave number • IDN output contains the IDN transfer that caused the error • SERR output contains the drive generated error number • Read Drive diagnostic IDN 95
136	Individual slave will not respond. The SLV output indicates the slave number.	• Address switch on drive does not match slave number • Baud rate switch on drive does not match rate in ring definition • SLV output contains slave number that does not respond

144	Individual slave cannot carry out a Procedure Command Function. The SLV output indicates the slave number.	• SLV output contains slave number • IDN output contains the Procedure Command Function that caused the error • For IDN = 127, read IDN 22 to read list of IDNs still required by the drive • For IDN = 128, read IDN 23 to read list of IDNs still required by the drive • Read Drive diagnostic IDN 95

Slave Error Codes

The slave errors listed below appear at the SERR output of certain slave SERCOS functions and will appear on the **Slave Error** line in the **Ring State** dialog.

SERR #	Description
4097	This IDN does not exist.
4105	The data for this IDN may not be accessed.
8193	The name does not exist.
8194	The name transmission is too short.
8195	The name transmission is too long.
8196	The name may not be changed.
8197	The name is write-protected.
12290	The attribute transmission is too short.
12291	The attribute transmission is too long.
12292	The attribute is write-protected at this time.
16385	The units do not exist.
16386	The units transmission is too short.
16387	The units transmission is too long.
16388	The units may not be changed.
16389	The units are write-protected at this time.
20481	The minimum value does not exist.
20482	The minimum value transmission is too short.
20483	The minimum value transmission is too long.
20484	The minimum value may not be changed.
20485	The minimum value is write-protected.
24577	The maximum value does not exist.
24578	The maximum value transmission is too short.
24579	The maximum value transmission is too long.
24580	The maximum value may not be changed.
24581	The maximum value is write-protected.

28674	The data is too short.
28675	The data is too long.
28676	The data may not be changed.
28677	The data is write-protected at this time.
28678	The data is smaller than the minimum value.
28679	The data is larger than the maximum value.
28680	The bit pattern for this IDN is invalid.

11895 SERCOS: SLAVE NUMBER GREATER THAN 16

This error message is displayed when using Save As to save an MMC for PC SRV file that specifies a SERCOS axis with a slave number greater than 16 in a format readable by PiCPro V12.0.

12002 SERCOS: IDN VALUE FORMAT

An invalid value has been specified. The acceptable ranges are listed below.

Range:

Two byte value = -32768 to 32767

Four byte value = -2,147,483,648 to 2,147,483,647

12003 SERCOS: CUT DELETE

A confirmation prompt asking if you are sure you want to cut or delete this ring or slave appears. You may choose to proceed or cancel.

12004 SERVOS: FATAL ERROR CREATING TEMP FILE

The slave data could not be edited because a temporary file could not be created.

12005 SERCOS: INVALID SELECTION

You have selected more than eight slaves to be cut/copied. The limit is eight since that is the maximum that can be pasted on a ring. You may select rings or slaves; you cannot select both. If both are selected, an error will occur.

12006 SERCOS: RING INFO UNREADABLE

The ring data is corrupted.

12007 SERCOS: SLAVE INFO UNREADABLE

The slave data is corrupted.

12008 SERCOS: NO RING

No rings have been defined. Define a ring in SERCOS setup and proceed.

12009 SERCOS: NO SLAVE

No slaves have been defined in Slot/Ring __. Define slaves in SERCOS setup.

12010 SERCOS: DUPLICATE

There is a duplicate slot definition at ___. Slot definitions cannot be duplicated.

12011 SERCOS: DUPLICATE SLAVE

There are duplicate slave numbers entered. Each slave must have a unique number.

12012 SERCOS: NOT SEQUENTIAL

The slave numbers must be sequential and begin with 1.

Refer to: *Inserting / Editing SERCOS Slaves*

12013 SERCOS: CLEAR DATA VALUE

A confirmation prompt asking if you are sure you want to clear the selected data. You may choose to proceed or cancel.

12014 SERCOS: INTERNAL ERROR

An internal error of flag byte out of range has occurred. Report this error to Danaher Motion.

12015 SERCOS: NOT INITIALIZED

SERCOS is not initialized in your ladder. The SC_INIT function block was not successfully called in your ladder.

12016 SERCOS: SERVER QUEUE IS FULL

The server queue is full. Too many requests have been made. The requested information exceeds the available buffer space in the control.

12017 SERCOS: SERCOS BOARD DOES NOT SUPPORT ANIMATION

The SERCOS board you have does not support animation. Contact Danaher Motion Tech Support for new SERCOS firmware.

12018 SERCOS: ERR 3

The axis is not a SERCOS axis, is not initialized, or slot/ring/slave specification is incorrect.

- Check that your SERCOS module is in the correct slot.
- Check that the SC_INIT function block in your ladder was called successfully to initialize SERCOS.

12019 SERCOS: DRIVE ERROR

This is a drive error. Refer to your drive documentation for information on it.

12020 SERCOS: UNDEFINED INTERNAL ERROR

An undefined error has been detected. Write down the error message and the error number and contact Danaher Motion Tech Support.

12021 SERCOS: NO ATTRIBUTE IN FILE

The attribute is missing for <IDN> in <IDN filename>.

All IDNs must have an attribute.

If an attribute is missing from your drive IDN file, contact your drive manufacturer and then contact Danaher Motion Tech Support. We may be able to help you with a work-around.

If an attribute is missing from your system IDN file, contact Danaher Motion Tech Support.

12025 SERCOS: DOCUMENT ALREADY OPEN

Unable to save document. A document with this name is already open, please save to a different name.

You cannot perform a "Save As" to a file that is already open. Either "Save As" to a different name or cancel this "Save As", close the conflicting open document, and do the "Save As" again.

12026 SERCOS: VERSION MISMATCH

The version of PiCPro specified for the open project does not match the version of PiCPro that is running. Compilers are different for different versions of PiCPro. You cannot compile this function in this version of PiCPro while the project is open.

You cannot compile in this situation. You can either:

- Launch the correct version of PiCPro (that is specified in the project)
- Change the version of PiCPro in the project to the one you are currently running
- Close the project. Open the SERCOS Setup file and compile.

12027 SERCOS: INVALID RING SPECIFIED

A ring in your file has a slot or ring value set to zero. You must specify a valid slot or ring value before you can save this file in a previous version format (10.2 – 11.0) or compile the function.

12029 SERCOS: INVALID SLAVE SPECIFIED

A slave in your file has slave number of zero. You must specify a valid slave number before you can save this file in a previous version format (10.2 – 11.0) or compile the function.

12030 SERCOS: TOO MANY RINGS HAVE BEEN SPECIFIED

The file has too many rings specified for the CPU type. The ring(s) must be deleted, or the CPU type changed before the file can be saved in a previous version format (10.2 – 11.0) or the function compiled.

12031 SERCOS: TOO MANY SLAVES HAVE BEEN SPECIFIED

The file has too many slaves specified for the CPU type. The slave(s) must be deleted, or the CPU type changed before the file can be saved in a previous version format (10.2 – 11.0) or the function compiled.

12435 SERCOS: CANNOT REMOVE FROM CYCLIC DATA

<IDNs in cyclic data> cannot be removed from cyclic data unless the <corresponding type> checkbox is unchecked.

The specific cyclic data mentioned in this message cannot be removed from cyclic data unless the corresponding checkbox is unchecked.

Edit the slave data (close the Cyclic Data dialog first) and uncheck the specified checkbox. The IDN you were trying to remove will automatically be removed from cyclic data.

Servo C-Stop, E-Stop and Programming Error Codes

Servo C-Stop, E-Stop, and Programming Errors Codes

There are four types of errors that can occur when working with servo control. The first three apply to individual axes. The fourth, timing error, is connected to the entire system.

1. C-stop (controlled-stop) errors
2. E-stop (emergency stop) errors
3. P (programming) errors
4. Timing errors

Servo C (controlled) - Stop Errors

When a C-stop (Controlled-stop) error occurs on an individual servo axis, the following happens:

- The axis remains in servo lock and the axis is brought to a controlled stop at the rate specified by the controlled stop ramp in servo setup.
- The active and next queues are cleared.
- The FAST_QUE mode is canceled when the C-stop is reset.

Bit Location (low byte)	Error Description	Hex* Value (decimal in ladder)
7	**Part reference error** Move was in progress when a part reference or a part clear function was called.	0x8080 (32896)
6	**Part reference dimension error** When the dimension for the part reference was converted to feedback units, it was too big to fit into 29 bits.	0x8040 (32832)
5	**Distance or position move dimension error** When the dimension for the move was converted to feedback units, it was too big to fit into 31 bits.	0x8020 (32800)
4	**Feedrate error** When the feedrate for the move was converted to feedback units per servo update, it was too big to fit into 32 bits or it exceeded the velocity limit entered in setup. **Note:** This error can occur with feedrate override, new feedrate, position, distance, velocity, or machine reference moves.	0x8010 (32784)
3	**Machine reference error** When the dimension for the machine reference was converted to feedback units, it was too big to fit into 29 bits.	0x8008 (32776)
2	**User-defined C-stop** When this bit is set, a user-defined C-stop has occurred.	0x8004 (32772)

1	**Negative software limit exceeded** The command position exceeded the user-defined negative software end limit.	0x8002 (32770)
0	**Positive software limit exceeded** The command position exceeded the user defined positive software end limit.	0x8001 (32769)

When more than one error occurs, the hex values are OR'd. For example, if 8001 and 8004 occur, the result is 8005 hex (32773 decimal).

Servo E (emergency) - Stop Errors

When an E-stop (Emergency-stop) error occurs on an individual servo axis, the following happens:

- The system is out of servo lock.
- A zero voltage is sent to the analog outputs.
- The active and next queues are cleared.
- The FAST_QUE mode is canceled when the E-stop is reset.

EXCEPTION:

If a User-Set or Excess Error E-Stop occurs while Resumable E-Stop Allow is set (Servo Setup or WRITE_SV/READ_SV Variable 63), a Resumable E-Stop will occur and the following happens:

- The system is out of servo lock.
- A zero voltage is sent to the analog outputs.
- The moves in the active and next queues remain intact.
- The axis' Normal Interpolator remains running.
- The axis goes into Resume Mode, the axis will follow the Resume Interpolator. The Resume Interpolator will output zero velocity until the RESUME function is called. The RESUME function can be called after the Resumable E-Stop has been reset and the servo loop has been closed. The axis remains in Resume Mode until the RESUME function brings back onto the Normal Interpolator's path or until a non-resumable E-Stop cancels Resume Mode.

Bit Location	Error Description	Hex* Value (decimal in ladder)
8 (high byte)	**Digital Drive Communication Error** Two consecutive CRC errors were detected in the data transferred between the MMCD and the digital drive. This E-stop cannot be reset with E_RESET. The ladder must call DSTRTSRV again to restart communication and reset this E-stop.	0x8100 (33024)
7	**Digital Drive Fault** A Drive fault was reported from the digital drive.	0x8080 (32896)
6	**ASIU Update Error** The MMC-for-PC did not receive the servo update data from the ASIU in time.	0x8040 (32832)

5	**SERCOS error** Cyclic data synchronization error.	0x8020 (32800)
4	**SERCOS error** SERCOS drive E-Stop - Status word bits 15, 14, and 13 are not equal to 1 1 0 respectively.	0x8010 (32784)
3	**User-set** An E-stop on a servo axis has occurred which was called in the ladder using the ESTOP function.	0x8008 (32776)
2	**Overflow error** A slave delta overflow during runtime has occurred. This problem is most likely to occur if you are moving at a high rate of speed and/or the slave distance is very large compared to the master distance. There are two conditions that can set this bit. In FU, if the master moved position times the slave distance entered is greater than 31 bits. In FU, if the master moved times the SDIS divided by the MDIS > 16 bits.	0x8004 (32772)
1	**Excess error** When an excess following error has occurred, the axis has exceeded the limit entered in the Servo setup program as the following error limit. This represents the maximum distance the commanded axis position can be from the actual axis position.	0x8002 (32770)
0	**Loss of feedback** A loss of feedback from the feedback device has occurred. Available for servo and digitizing axes.	0x8001 (32769)

When more than one error occurs, the hex values are OR'd. For example, if 8001 and 8004 occur, the result is 8005 hex (32773 decimal).

Servo P (programming) - Errors

P- (Programming) errors occur during master/slave moves or a FAST_QUE call. P-errors may prevent:

- The move from being placed in the queue (or if the move is in the queue, abort the move)

  OR

- The OK on the function from being set

Bit Location (low byte)	Error Description	Hex* Value (decimal in ladder)
7	**The FAST axis in the FAST_QUE function moved too far in the wrong direction** The axis traveled more than 65,535 FU in the opposite direction of the value entered in DIST of the FAST_QUE function.	0x8080 (32896)

6	**Profile number not found** Data for a profile move is not valid.	0x8040 (32832)
5	**Master axis not available** This error can occur when using the FAST_QUE function or the functions for master/slave moves (RATIO_GR, RATIOSYN, or RATIOPRO). The conditions that can set this bit include: Master axis or fast axis not initialized Interrupt rates different for axes Axis at slave input is the same as axis at master input in master/slave moves The master/slave move has requested to use the master's command position and the master axis is not a servo axis. Choosing to use the master's command position is achieved with the OPTN input for RATIOCAM, RATIOSLP, and RATIO_RL or WRITE_SV Variable 59 for RATIO_GR, RATIOSYN, and RATIOPRO.	0x8020 (32800)
4, 3, 2, 1	Not Used	---
0	**Master start position for lock on** When the dimension for the lock position was converted to feedback units, it was too big to fit into 32 bits.	0x8001 (32769)

Bit Location (high byte)	Error Description	Hex* Value (decimal in ladder)
7	A programming error has occurred. Set at least one other bit in the word.	0x8000 (32768)
6, 5, 4	Not Used	---
3	Master axis beyond start point The master axis is beyond its starting point for a ration synchronization (RATIOSYN) move.	0x8800 (34816)
2	Slave axis beyond start point The slave axis is beyond its starting point for a ratio synchronization (RATIOSYN) move.	0x8400 (33792)
1	Master distance not valid When the master distance is converted to feedback units, it is greater than 16 bits.	0x8200 (33280)
0	Slave distance not valid When the slave distance is converted to feedback units, it is greater than 16 bits.	0x8100 (33024)

When more than one error occurs, the hex values are OR'd. For example, if 8100 and 8200 occur, the result is 8300 hex (33536 decimal).

Servo Timing Errors

All the servo calculations for one interrupt must be completed in the time frame selected by you in servo setup before the next interrupt begins. If they are not completed, a timing error occurs. The timing error is connected to the entire system. This error is monitored in the ladder program with the TME_ERR? function. If the boolean output at ERR is set, a timing error is occurring. Depending on the system, this can affect performance.

IMPORTANT
Always set an E-stop on all axes when a timing error occurs.

Servo Setup Error Messages

11001 SERVO: INVALID AXIS CUT COPY

The axis data you were attempting to cut or copy has been corrupted. Close the program without saving any changes and try again.

11002 SERVO: INVALID AXIS PASTE

The axis data you are attempting to paste has been corrupted. Close the program without saving any changes and try again.

11003 SERVO: INVALID AXIS LABEL

You must enter a unique axis label for each axis you are entering before proceeding. This label can be up to eight characters in length.

11004 SERVO: TUNE NO PARENT

When you attempt to view or force variables within servo setup, the servo setup file must be opened from within the parent ladder by choosing Servo function from the View menu. If you open the .SRV file using the Open command, viewing and forcing will be disabled.

11005 SERVO: TUNE NO MAIN

If you attempt to activate servo viewing and forcing when the path to the main ladder has not been defined, an error will occur.

11007 SERVO: NO MORE DATA

This is an internal software condition. Please note error number and consult factory.

11008 SERVO: AXIS INFO UNREADABLE

The axis data has been corrupted.

11019 SERVO: NO AXIS DEFINED

No axes have been defined for this servo setup function. Insert one or more axes into the servo setup program.

11020 SERVO: SOFTWARE UPPER LIMIT CALCULATION ERROR

Overflow calculating the Software Upper Limit. Check your inputs for Axis # "__".

In servo setup you have entered scaling data for feedback units (FU) and ladder units (LU) and the software upper limit. The software uses the formula shown to convert this information into feedback units and checks that the result is within the acceptable range.

$$\frac{\text{software upper limit} * \text{FU}}{\text{LU}} = \text{N (where N must be within range of -536870912 to 536870911 FU)}$$

Tip:

Adjust the axis data information so that the result of the conversion calculation will fall within the acceptable range.

11021 SERVO: SOFTWARE LOWER LIMIT CALCULATION ERROR

Overflow calculating the Software Lower Limit. Check your inputs for Axis # "__".

In servo setup you have entered scaling data for feedback units (FU) and ladder units (LU) and the software lower limit. The software uses the formula shown to convert this information into feedback units and checks that the result is within the acceptable range.

$$\frac{\text{software lower limit} * \text{FU}}{\text{LU}} = \text{N (where N must be within range of -536870912 to 536870911 FU)}$$

Tip:

Adjust the axis data information so that the result of the conversion calculation will fall within the acceptable range.

11022 SERVO: FOLLOWING ERROR LIMIT CALCULATION ERROR

Overflow calculating Excess Error Limit. Check your inputs for Axis # "__".

In servo setup you have entered scaling data for feedback units (FU) and ladder units (LU) and the excess error limit. The software uses the formula shown to convert this information into feedback units and checks that the result is within the acceptable range.

$$\frac{\text{excess error limit} * \text{FU}}{\text{LU}} = \text{N (where N must be within range of 0 to 536870911 FU)}$$

Tip:

Adjust the axis data information so that the result of the conversion calculation will fall within the acceptable range.

11023 SERVO: IN POSITION BAND CALCULATION ERROR

Overflow calculating the In Position Band. Check your inputs for Axis # "__".

In servo setup you have entered scaling data for feedback units (FU) and ladder units (LU) and the in position band. The software uses the formula shown to convert this information into feedback units and checks that the result is within the acceptable range.

$$\frac{\text{in position band * FU}}{\text{LU}} = N \text{ (where N must be within range of 0 to 536870911 FU)}$$

Tip:

Adjust the axis data information so that the result of the conversion calculation will fall within the acceptable range.

11024 SERVO: ROLLOVER POSITION CALCULATION ERROR

Overflow calculating the Rollover Position. Check your inputs for Axis # "__".

In servo setup you have entered scaling data for feedback units (FU) and ladder units (LU) and the rollover position. The software uses the formula shown to convert this information into feedback units and checks that the result is within the acceptable range.

$$\frac{\text{rollover position * FU}}{\text{LU}} = N \text{ (where N must be within range of 0 to 536870911 FU)}$$

Tip:

Adjust the axis data information so that the result of the conversion calculation will fall within the acceptable range.

11025 SERVO: PLUS INTEGRAL LIMIT CALCULATION ERROR

Overflow calculating Plus Integral Error Limit. Check your inputs for Axis # "__".

In servo setup you have entered scaling data for feedback units (FU) and ladder units (LU) and the plus integral error limit. The software uses the formula shown to convert this information into feedback units and checks that the result is within the acceptable range.

$$\frac{\text{plus integral error limit * FU}}{\text{LU}} = N \text{ (where N must be within range of 0 to 536870911 FU)}$$

Tip:

Adjust the axis data information so that the result of the conversion calculation will fall within the acceptable range.

11026 SERVO: MINUS INTEGRAL LIMIT CALCULATION ERROR

Overflow calculating Minus Integral Error Limit. Check your inputs for Axis # "__".

In servo setup you have entered scaling data for feedback units (FU) and ladder units (LU) and the minus integral error limit. The software uses the formula shown to convert this information into feedback units and checks that the result is within the acceptable range.

$$\frac{\text{minus integral error limit * FU}}{\text{LU}} = N \text{ (where N must be within range of -536870912 to 0 FU)}$$

Tip:

Adjust the axis data information so that the result of the conversion calculation will fall within the acceptable range.

11027 SERVO: VELOCITY LIMIT CALCULATION ERROR

Overflow calculating Velocity Limit. Check your inputs for Axis # "__".

In servo setup you have entered scaling data for feedback units (FU) and ladder units (LU), the update rate, and the velocity limit (entered in LU/min). The software uses the formula shown to convert this information into feedback units per iteration and checks that the result is within the acceptable range.

$$\frac{\text{velocity limit} * \text{FU} * 8 * \text{update rate}}{\text{LU}} = \text{N (where N must be within range of 0 to 32767 FU/ iteration)}$$

Tip:

Adjust the axis data information so that the result of the conversion calculation will fall within the acceptable range.

Note:

When using a formula in any of the Servo error calculations calling for an update rate entry, use the number of milliseconds for an update rate value, i.e., for a 2 ms update rate, use "2".

11028 SERVO: INTEGRAL GAIN CALCULATION ERROR

Overflow calculating Integral Gain. Check your inputs for Axis # "__".

In servo setup you have entered scaling data for feedback units (FU) and ladder units (LU), the update rate, and the integral gain. The value entered in setup for integral gain represents the ladder units per minute per ladder unit of following error (FE) times minutes. The software uses the formula shown to convert this information into feedback units and checks that the result is within the acceptable range.

$$\frac{\text{integral gain value} * 3253 * \text{update rate}}{\text{FU/min-volt}} = \text{N (where N must be within range of 0 to 32767)}$$

Tip:

Adjust the axis data information so that the result of the conversion calculation will fall within the acceptable range. A typical value for integral gain entered in setup is zero. If required, up to 32,000 LU/min/LUFE * min. can be entered.

Note:

When using a formula in any of the Servo error calculations calling for an update rate entry, use the number of milliseconds for an update rate value, i.e., for a 2 ms update rate, use "2".

11029 SERVO: PROPORTIONAL GAIN CALCULATION ERROR

Overflow calculating Proportional Gain. Check your inputs for Axis # "__".

In servo setup you have entered scaling data for feedback units (FU) and ladder units (LU) and the proportional gain. The value entered in setup for proportional gain represents the ladder units per minute for each ladder unit of following error. The software uses the formula shown to convert this information into feedback units and checks that the result is within the acceptable range.

$$\frac{\text{proportional gain value} * 762601}{\text{FU/min-volt}} = \text{N (where N must be within range of 0 to 32767 FU)}$$

Tip:

Adjust the axis data information so that the result of the conversion calculation will fall within the acceptable range. Typical values for proportional gain entered in setup are from 1,000 to 5,000 LU/min/LUFE.

11030 SERVO: DERIVATIVE GAIN CALCULATION ERROR

Overflow calculating Derivative Gain. Check your inputs for Axis # "__".

In servo setup you have entered scaling data for feedback units (FU) and ladder units (LU), the update rate, and the derivative gain. The value entered in setup for derivative gain represents the ladder units per minute for each ladder unit of following error per minute. The software uses the formula shown to convert this information into feedback units and checks that the result is within the acceptable range.

$$\frac{\text{derivative gain value} * 178734}{\text{FU/min-volt} * \text{update rate}} = N \text{ (where N must be within range of 0 to 32767 FU)}$$

Tip:

Adjust the axis data information so that the result of the conversion calculation will fall within the acceptable range. A typical value for derivative gain entered in setup is zero. If required, up to 500 AU/min/AUFE/min can be entered.

Note:

When using a formula in any of the Servo error calculations calling for an update rate entry, use the number of milliseconds for an update rate value, i.e., for a 2 ms update rate, use "2".

11031 SERVO: FEED FORWARD CALCULATION ERROR

Overflow calculating Feed Forward Factor. Check your inputs for Axis # "__".

In servo setup you have entered scaling data for feedback units (FU) and ladder units (LU), the update rate, and the feed forward percent. The software uses the formula shown to convert this information into feedback units and checks that the result is within the acceptable range.

$$\frac{\text{feed forward percent} * 457560436}{\text{FU/min-volt} * \text{update rate}} = N \text{ (where N must be within range of 0 to 524272)}$$

Tip:

Adjust the axis data information so that the result of the conversion calculation will fall within the acceptable range.

Note:

When using a formula in any of the Servo error calculations calling for an update rate entry, use the number of milliseconds for an update rate value, i.e., for a 2 ms update rate, use "2".

11032 SERVO: CSTOP RAMP CALCULATION ERROR

Overflow calculating Controlled Stop Ramp. Check your inputs for Axis # "__".

In servo setup you have entered scaling data for feedback units (FU) and ladder units (LU), the update rate, and the controlled stop ramp in ladder units/minute/second. The software uses the formula shown to convert this information into feedback units and checks that the result is within the acceptable range.

$$\frac{\text{cstop ramp * FU * update rate * update rate}}{\text{LU * 937500}}$$ = N (where N must be within range shown below)

If SERCOS, then N must be within the range of 1 to 536870911.0 FU
If Encoder, Resolver or TTL, then N must be within the range of 1 to 67108864.0 FU
If Analog Output or Stepper, then N must be within the range of 1 to 262144.0 FU

Tip:

Adjust the axis data information so that the result of the conversion calculation will fall within the acceptable range. Typical values entered in setup are 10,000 to 10,000,000 LU/min/sec, not to exceed 1023 FU/update/update.

Note:

When using a formula in any of the Servo error calculations calling for an update rate entry, use the number of milliseconds for an update rate value, i.e., for a 2 ms update rate, use "2".

11033 SERVO: ACCEL RAMP CALCULATION ERROR

Overflow calculating Acceleration Ramp. Check your inputs for Axis # "__".

In servo setup you have entered scaling data for feedback units (FU) and ladder units (LU), the update rate, and the acceleration ramp in ladder units/minute/second. The software uses the formula shown to convert this information into feedback units and checks that the result is within the acceptable range.

$$\frac{\text{acceleration ramp * FU * update rate * update rate}}{\text{LU * 937500}}$$ = N (where N must be within range shown below)

If SERCOS, then N must be within the range of 1 to 536870911.0 FU
If Encoder, Resolver or TTL, then N must be within the range of 1 to 67108864.0 FU
If Analog Output or Stepper, then N must be within the range of 1 to 262144.0 FU

Tip:

Adjust the axis data information so that the result of the conversion calculation will fall within the acceptable range. Typical values entered in setup are 10,000 to 10,000,000 LU/min/sec not to exceed 1023 FU/update/update.

Note:

When using a formula in any of the Servo error calculations calling for an update rate entry, use the number of milliseconds for an update rate value, i.e., for a 2 ms update rate, use "2".

11034 SERVO: DECEL RAMP CALCULATION ERROR

Overflow calculating Deceleration Ramp. Check your inputs for Axis # "__".

In servo setup you have entered scaling data for feedback units (FU) and ladder units (LU), the update rate and the deceleration ramp in ladder units/minute/second. The software uses the formula shown to convert this information into feedback units and checks that the result is within the acceptable range.

$$\frac{\text{deceleration ramp * FU * update rate * update rate}}{\text{LU * 937500}}$$ = N (where N must be within range shown below)

If SERCOS, then N must be within the range of 1 to 536870911.0 FU
If Encoder, Resolver or TTL, then N must be within the range of 1 to 67108864.0 FU
If Analog Output or Stepper, then N must be within the range of 1 to 262144.0 FU

Tip:

Adjust the axis data information so that the result of the conversion calculation will fall within the acceptable range. Typical values entered in setup are 10,000 to 10,000,000 LU/min/sec, not to exceed 1023 FU/update/update.

Note:

When using a formula in any of the Servo error calculations calling for an update rate entry, use the number of milliseconds for an update rate value, i.e., for a 2 ms update rate, use "2".

11035 SERVO: BUILDER FEED OVERRIDE

This is an internal error. Please make a note of the error number and consult factory.

11036 SERVO: SLOW FILTER CALCULATION ERROR

Overflow calculating Slow Velocity Filter. Check your inputs for Axis # "__".

In servo setup you have entered scaling data for feedback units (FU) and ladder units (LU), the update rate, the slow filter, and the slow velocity filter in milliseconds. When the slow filter is non-zero, the software uses the formula shown to convert this information into feedback units and checks that the result is within the acceptable range.

$65535 * (1-e^{(-Y)})$ **= N** (where **N** must be within range of 0 to 65535 FU)

And where:

-Y = -(update rate / slow filter)

Tip:

Adjust the axis data information so that the result of the conversion calculation will fall within the acceptable range.

Note:

When using a formula in any of the Servo error calculations calling for an update rate entry, use the number of milliseconds for an update rate value, i.e., for a 2 ms update rate, use "2".

11037 SERVO: FAST FILTER CALCULATION ERROR

Overflow calculating Fast Velocity Filter. Check your inputs for Axis # "__".

In servo setup you have entered scaling data for feedback units (FU) and ladder units (LU), the update rate, the fast filter, and the fast velocity filter in milliseconds. When the fast filter is non-zero, the software uses the formula shown to convert this information into feedback units and checks that the result is within the acceptable range.

$65535 * (1-e^{(-W)})$ **= N** (where **N** must be within range of 0 to 65535 FU)

And where:

-W = -(update rate / fast filter)

Tip:

Adjust the axis data information so that the result of the conversion calculation will fall within the acceptable range.

Note:

When using a formula in any of the Servo error calculations calling for an update rate entry, use the number of milliseconds for an update rate value, i.e., for a 2 ms update rate, use "2".

11038 SERVO: FILTER VELOCITY THRESHOLD CALCULATION ERROR

Overflow calculating Slow/Fast Velocity Threshold. Check your inputs for Axis # "__".

In servo setup you have entered scaling data for feedback units (FU) and ladder units (LU), the update rate, and the slow/fast velocity threshold in ladder units/minute. The software uses the formula shown to convert this information into feedback units and checks that the result is within the acceptable range.

$$\frac{\text{velocity threshold * FU * update rate}}{\text{LU * 60000}} = \text{N (where N must be within range of 1 to 65535 FU)}$$

Tip:

Adjust the axis data information so that the result of the conversion calculation will fall within the acceptable range. Typical value entered in setup is zero, not to exceed 4095.

Note:

When using a formula in any of the Servo error calculations calling for an update rate entry, use the number of milliseconds for an update rate value, i.e., for a 2 ms update rate, use "2".

11039 SERVO: FILTER LAG CALCULATION ERROR

Overflow calculating Velocity Filter Lag Error. Check your inputs for Axis # "__".

In servo setup you have entered scaling data for feedback units (FU), and ladder units (LU), the update rate, the velocity threshold, and the fast velocity filter in milliseconds. When the fast filter is non-zero, the software uses the formula shown to convert this information into feedback units and checks that the result is within the acceptable range.

V * FU* update rate)/ (LU *65535 *(1-e $^{(-Y)}$)) **= N** (where **N** must be within range of 0 to 65535 FU)

And where:

V = velocity threshold

-Y = -(update rate / slow filter)

Tip:

Adjust the axis data information so that the result of the conversion calculation will fall within the acceptable range.

Note:

When using a formula in any of the Servo error calculations calling for an update rate entry, use the number of milliseconds for an update rate value, i.e., for a 2 ms update rate, use "2".

11040 SERVO: UPDATE SCALING CALCULATION ERROR

Overflow calculating Servo Update Rate. Check your inputs for Axis # "__".

The selected update rate entered in milliseconds is out of range.

Tip:

Adjust the axis data information so that the result of the conversion calculation will fall within the acceptable range.

4ms is adequate for most applications. Lower values consume more CPU processing time. If too many axes have too low an update rate, the CPU may not have enough processing time available to run the user program.

11041 SERVO: D/A OFFSET CALCULATION ERROR

Overflow calculating Analog Output Offset. Check your inputs for Axis # "__".

The D/A offset is out of range.

Tip:

Adjust the axis data information so that the result of the conversion calculation will fall within the acceptable range.

11042 SERVO: BUILDER NOT FOUND

If the filename is invalid when a function is made, this message appears. Please make a note of the error number and consult the factory.

11043 SERVO: BUILDER NEW FILE

If the filename is not filled in when a function is made, this message is displayed. Please make a note of the error number and consult the factory.

11044 SERVO: UPDATE DIFFERENCES CALCULATION ERROR

Invalid ratio between slowest/fastest update rates with Axis # "__" and Axis # "__".

The ratio between the updates of the fastest and slowest is greater than 16.

Tip:

Adjust the update rate in the position loop data of axis data so that the result of the conversion calculation will fall within the acceptable range.

11099 SERVO: NEWER VERSION SERVO FILE

This file was made with a newer version of PiCPro than the one you are currently using. You must open the file in the PiCPro version it was created with. The file can then be saved as for the older version of PiCPro.

11102 SERVO: OVERFLOW CONSTANT JERK ERROR

There was an overflow calculating Constant Jerk for Move Accel/Decel. Check your inputs for Axis number percentages.

In Servo Setup you have entered scaling data for feedback units (FU) an ladder units (LU) and Ladder Units to Axis Units (LU2AU), the update rate, and the Constant Jerk for Move Accel/Decel in ladder units/minute/second. The software uses the formula shown to convert this information into feedback units and checks that the result is within the acceptable range.

$$\frac{\text{Constant Jerk} * \text{FU} * \text{update rate} * \text{update rate} * \text{update rate} * \text{LU2AU}}{\text{LU} * 60000 * 1000 * 1000}$$ = N (where N is > 0 and < 67108863.0)

Tip:

Adjust the axis data information so that the results of the conversion calculation will fall within the acceptable range.

Note:

When using a formula in any of the Servo error calculations calling for an update rate entry, use the number of milliseconds for an update rate value (e.g. for a 2 ms update rate, use 2).

11103 SERVO: OVERFLOW MAX ACCELERATION

There was an overflow calculating Max Acceleration for Move Accel/Decel. Check your inputs for Axis number percentages.

In Servo Setup you have entered scaling data for feedback units (FU) an ladder units (LU) and Ladder Units to Axis Units (LU2AU), the update rate, and the Max Acceleration for Move Accel/Decel in ladder units/minute/second. The software uses the formula shown to convert this information into feedback units and checks that the result is within the acceptable range.

$$\frac{\text{Max Acceleration} * \text{FU} * \text{update rate} * \text{update rate} * \text{LU2AU}}{\text{LU} * 60000 * 1000 * 1000}$$ = N (where N is > 0 and < 67108863.0)

Tip:

Adjust the axis data information so that the results of the conversion calculation will fall within the acceptable range.

Note:

When using a formula in any of the Servo error calculations calling for an update rate entry, use the number of milliseconds for an update rate value (e.g. for a 2 ms update rate, use 2).

11104 SERVO: OVERFLOW CALCULATING CONSTANT JERK FOR CONTROLLED STOP DECEL

There was an overflow calculating Constant Jerk for Move Accel/Decel. Check your inputs for Axis number percentages.

In Servo Setup you have entered scaling data for feedback units (FU) an ladder units (LU) and Ladder Units to Axis Units (LU2AU), the update rate, and the Constant Jerk for Controlled Stop Decel in ladder units/minute/second/second. The software uses the formula shown to convert this information into feedback units and checks that the result is within the acceptable range.

$$\frac{\text{Constant Jerk} * \text{FU} * \text{update rate} * \text{update rate} * \text{update rate} * \text{LU2AU}}{\text{LU} * 60000 * 1000 * 1000}$$ = N (where N is > 0 and < 67108863.0)

Tip:

Adjust the axis data information so that the results of the conversion calculation will fall within the acceptable range.

Note:

When using a formula in any of the Servo error calculations calling for an update rate entry, use the number of milliseconds for an update rate value (e.g. for a 2 ms update rate, use 2).

11105 SERVO: OVERFLOW CALCULATING MAX ACCEL FOR CONTROLLED STOP DECEL

There was an overflow calculating Max Acceleration for Controlled Stop Decel. Check your inputs for Axis number percentages.

In Servo Setup you have entered scaling data for feedback units (FU) an ladder units (LU) and Ladder Units to Axis Units (LU2AU), the update rate, and the Max Acceleration for Controlled Stop Decel in ladder units/minute/second. The software uses the formula shown to convert this information into feedback units and checks that the result is within the acceptable range.

$$\frac{\text{Max Acceleration * FU * update rate * update rate * LU2AU}}{\text{LU * 60000 * 1000}} = \text{N (where N is > 0 and < 67108863.0)}$$

Tip:

Adjust the axis data information so that the results of the conversion calculation will fall within the acceptable range.

Note:

When using a formula in any of the Servo error calculations calling for an update rate entry, use the number of milliseconds for an update rate value (e.g. for a 2 ms update rate, use 2).

11116 AXIS: PASTE INVALID AXIS NUMBER ERROR

"Axis is not a valid axis number for this CPU type. Axis will be inserted as axis number X. Do you want to continue?"

This dialog is displayed in the MMC-Limited Edition when copying an axis with an invalid axis number. The file was created or modified using the Professional Edition.

11171 AXIS: PASTE IMAGE AXIS ERROR

A DLS axis can only be pasted if the target CPU is an MMC for PC.

An attempt has been made to paste a digitizing DLS axis into a servo setup where the target CPU is not an MMC for PC.

11800 AXIS: CUT

Check that you want to cut the selected axis.

11801 AXIS: DELETE

Check that you want to delete the selected axis.

11802 AXIS: MAXIMUM ALLOWED

You have exceeded the maximum number of axes and cannot proceed with pasting your selection.

11803 AXIS: PASTE DUPLICATE ERROR

Axis # already exists. The axis will be inserted as axis # _. You can choose to continue or cancel.

11804 AXIS: PASTE INCOMPATIBLE TYPE

When you paste an axis over another axis, both axes must be the same type: servo or digitizing.

11805 AXIS: PASTE DIFFERENT I/O

The axis data you are pasting is from an axis with an input type of _ and an output type of _. You can choose to continue or cancel.

11806 AXIS: PASTE DIFFERENT INPUT TYPE

The axis data you are pasting is from an axis with an input type of _. You can choose to continue or cancel.

11807 FORCE: UPDATE CONFIG DATA WITH FORCE VALUES

Forcing values have changed. You can choose to update the axis configuration data with the current forcing values or cancel.

11808 FORCE: ENTRY ERR PGAIN

The value entered in the Force List data for proportional gain is out of the acceptable range.

11809 FORCE: ENTRY ERR IGAIN

The value entered in the Force List data for integral gain is out of the acceptable range.

11810 FORCE: ENTRY ERR DGAIN

The value entered in the Force List data for derivative gain is out of the acceptable range.

11811 FORCE: ENTRY ERR FEEDFWD

The value entered in the Force List data for feed forward percentage is out of the acceptable range.

11812 FORCE: ENTRY ERR SFILTER

The value entered in the Force List data for the slow speed filter is out of the acceptable range.

11813 FORCE: ENTRY ERR DAOFF

The value entered in the Force List data for the D/A offset is out of the acceptable range.

11850 SERVO: MMC AXES NOT SAME TYPE

This error message is displayed when attempting to compile a standalone MMC or MMC for PC SRV file that contains more than one axis type. All axes must be either D/A Encoder or SERCOS axes, but not both.

11853 SERVO: DOCUMENT ALREADY OPEN

Unable to save document. A document with this name is already open, please save to a different name.

You cannot perform a "Save As" to a file that is already open. Either "Save As" to a different name or cancel this "Save As", close the conflicting open document, and do the "Save As" again.

11854 SERVO: COMPILE ABORTED

The version of PiCPro specified for the open project does not match the version of PiCPro that is running. Compilers are different for different versions of PiCPro. You cannot compile this function in this version of PiCPro while the project is open.

You cannot compile in this situation.

You can either:

- Launch the correct version of PiCPro (that is specified in the project)
- Change the version of PiCPro in the project to the one you are currently running
- Close the project. Open the Servo Setup file and compile.

11856 SERVO: UNKNOWN CPU TYPE

This message is displayed if the CPU in the file is not a PiC, standalone MMC, or an MMC for PC.

11857 SERVO: SAVE AS FAILED

This error message is displayed when using Save As with a PiC or standalone MMC SRV file to an earlier version of PiCPro.

One of the following may not be supported in the earlier version of PiCPro:

- There may be too many axes
- Invalid axes numbers are specified
- An invalid axis number or axis type is specified
- Ladder feedback units exceed 65535
- A slot, channel, ring, or slave number is invalid

You can highlight the error number in the Information Window and press **F1**, or you can access more information about the error number from the index on the **Help** menu.

11858 SERVO: SAVE AS FAILED

This error message is displayed when using Save As with a PiC or standalone MMC SRV file to an earlier version of PiCPro.

One of the following may not be supported in the earlier version of PiCPro:

- There may be too many axes
- Invalid axes numbers are specified
- An invalid axis number or axis type is specified
- Ladder feedback units exceed 65535
- A slot, channel, ring, or slave number is invalid

You can highlight the error number in the Information Window and press **F1**, or you can access more information about the error number from the index on the **Help** menu.

11859 SERVO: INVALID PIC AXIS

This error message is displayed when using Save As to save a PiC file in a previous PiCPro version (10.0 through 11.0) which contains one or more axes with a slot, channel, ring, or slave value of zero. Also displayed when attempting to compile a PiC SRV file where any of the axes has a 0 value for slot, channel, ring, or slave.

11860 SERVO: INVALID MMC DIGIT AXIS

This error message is displayed when using Save As with a standalone MMC SRV file which contains a digitizing axis with a slot, channel, ring or slave value of zero to a previous version of PiCPro (10.2 through 11.0). It is also displayed when compiling a standalone MMC SRV file where any of the axes has a 0 value for slot, channel, ring, or slave.

11861 SERVO: INVALID MMC FOR PC DIGIT AXIS

This error message is displayed when saving or compiling an MMC for PC SRV file that contains a digitizing axis with a slot, channel, ring, or slave value of zero.

11862 SERVO: INVALID MMC SERVO AXIS

This error message is displayed when using Save As with a standalone MMC SRV file which contains a servo axis with a slot, channel, ring, or slave value of zero to previous version of PiCPro (10.2 through 11.0). It is also displayed when compiling a standalone MMC SRV file where any of the axes has a 0 value for slot, channel, ring, or slave.

11863 SERVO: INVALID MMC FOR PC SERVO AXIS

This error message is displayed when saving or compiling an MMC for PC SRV file, which contains a servo axis with a slot, channel, ring, or slave value of zero.

11864 SERVO: MMC TOO MANY AXES

This error message is displayed when using Save As with a standalone MMC SRV file which contains more axes than allowed in previous versions of PiCPro (10.2 through 11.0). It is also displayed in the MMC-Limited Edition when compiling a standalone MMC SRV file, which contains more axes than are allowed.

11866 SERVO: INVALID MMC DIGIT AXIS NUMBER

This error message is displayed when using Save As with a standalone MMC SRV file which contains a digitizing axis with an invalid axis number or axis type combination to a previous version of PiCPro (10.2 through 11.0). It is also displayed in the MMC-Limited Edition when compiling a standalone MMC SRV file, which contains a digitizing axis with an invalid axis number or axis type combination.

11867 SERVO: INVALID MMC SERVO AXIS NUMBER

This error message is displayed when using Save As with a standalone MMC SRV file which contains a servo axis with an invalid axis number or axis type combination to a previous version of PiCPro (10.2 through 11.0). It is also displayed in the MMC-Limited Edition when compiling a standalone MMC SRV file, which contains a servo axis with an invalid axis number or axis type combination.

11868 SERVO: INVALID AXIS FU OR LU EXCEEDS 65535

This error message is displayed when using Save As to save an SRV file into a previous version format (10.2 – 11.0) when file contains an axis whose feedback units or ladder units exceed 65535.

11869 SERVO: UNABLE TO ANIMATE

This error message is displayed when downloading a ladder which does not contain the STRTSRV function, viewing the servo setup function in the ladder, displaying a servo view or force list, and selecting "Viewing On" or "Forcing On".

11870 SERVO: INVALID AXIS FU/LU RATIO EXCEEDS 65535

This error message is displayed when compiling a servo setup function which contains an axis whose Feedback Units to Ladder Units ratio is larger than 65535.

11873 AXIS: DUPLICATE AXIS NUMBER

This message is displayed when attempting to change the axis number to a number that is already in use by another axis. The axis number can be changed when inserting or editing the axis properties.

11877 SERVO: WARNING INPUT POLARITY REVERSED

This message is displayed anytime an axis or axes is converted or pasted between PiC and standalone MMC or MMC for PC CPU types. To change the polarity, select the axis, select **Edit | Axis Data**, select the **Position Loop Data** tab, and change the input polarity.

Warning: If the input polarity is not changed, unexpected motion may occur.

11885 SERVO: INVALID MMC EDITION SERVO AXIS

This error message is displayed in the MMC-Limited Edition when using Save As with a standalone MMC SRV file which contains a servo axis with a slot, channel, ring, or slave value of zero to a previous version of PiCPro (10.2 through 11.0). It is also displayed when compiling a standalone MMC SRV file where any of the axes has a 0 value for slot, channel, ring, or slave.

11886 SERVO: INVALID MMC EDITION DIGIT AXIS

This error message is displayed in the MMC-Limited Edition when using Save As with a standalone MMC SRV file which contains a digitizing axis with a slot, channel, ring, or slave value of zero to a previous version of PiCPro (10.2 through 11.0). It is also displayed when compiling a standalone MMC SRV file where any of the axes has a 0 value for slot, channel, ring, or slave.

11915 SERVO: TOO MANY MASTER AXES

More than one axis is specified as a DLS Communications Master axis. Only one Communications Master is allowed.

Tip:

Determine which axis should be the Communications Master axis. Edit the other Communications Master axes and uncheck the Communications Master checkbox.

11916 SERVO: TOO MANY IMAGE AXES

More than one axis is specified as a digitizing DLS axis. Only one digitizing axis is allowed.

Tip:

Determine which axis should be the digitizing DLS axis. Edit the other digitizing DLS axis and select a different Input Type.

11917 SERVO: MASTER AND IMAGE AXIS INVALID

Both a DLS Communications Master axis and a digitizing DLS axis have been specified. A servo setup function can include one or the other but not both.

TCP/IP Error Codes

TCP/IP ERR Error Codes

The following errors can be reported out of the ERR output on the IPXXXX function blocks.

ERR#	Description
-35	Detected hardware failure
-34	Can’t find a reasonable interface
-33	Can’t find a reasonable next IP hop
-32	Bad header at upper layer (for upcalls)
-31	No ARP for a given host
-30	Send to net failed at low layer
-24	TCP layer timeout error
-23	TCP layer state error
-22	Ran out of other queue-able resource
-21	Ran out of free packets
-20	Malloc or calloc failed
0	No error
1	Not owner
2	No such file or directory
3	No such process
4	Interrupted system call
5	I/O error
6	No such device or address
7	Arg list too long
8	Exec format error
9	Bad file number
10	No children
11	No more process
12	Not enough core
13	Permission denied
14	Bad address
15	Directory not empty
16	Mount device busy
17	File exists
18	Cross-device link

19	No such device
20	Not a directory
21	Is a directory
22	Invalid argument
23	File table overflow
24	Too many files open
25	Not a typewriter
26	File name too long
27	File too large
28	No space left on device
29	Illegal seek
30	Read-only file system
31	Too many links
32	Broken pipe
33	Resource deadlock avoided
34	No locks available
35	Unsupported value
36	Message size
37	Argument too large
38	Result is too large
40	Destination address required
41	Protocol wrong type for socket
42	Protocol not available
43	Protocol not supported
44	Socket type not supported
45	Operation not supported on socket
46	Protocol family not supported
47	Address family not supported
48	Address already in use
49	Can't assign requested address
50	Socket operation on non-socket
51	Network is unreachable
52	Network dropped connection on reset
53	Software caused connection abort
54	Connection reset by peer
55	No buffer space available
56	Socket is already connected
57	Socket is not connected
58	Can't send after socket shutdown
59	Too many references: can't splice

60	Connection timed out
61	Connection refused
62	Network is down
63	Text file busy
64	Too many levels of symbolic links
65	No route to host
66	Block device required
67	Host is down
68	Operation now in progress
69	Operation already in progress
70	Operation would block
71	Function not implemented
72	Operation canceled
1000	There is a non-zero terminated string that requires zero termination or a zero length string.
1001	There is a CNT input that is too large
1002	The SLOT number requested does not contain an Ethernet board
1003	Either the firmware does not support TCP/IP or there is no Ethernet board in the rack
1004	The IPZ buffer is too small
1005	TCP/IP stack failure. Socket no longer valid.
10000 and up	Any error code above 10,000 that occur while using these function blocks was generated by the Microsoft operating system. Please check their documentation.

APPENDIX C - PiCPro Reference: Errors, Variables, Keyboard Shortcuts

CAPTINIT Error Codes

Error numbers and descriptions for the CAPTINIT function ERR output are:

0	No Error
1	The CAPSTAT function has not stopped capturing data from a previous data capture initialization
2	An axis number in the structure is invalid
3	The limit of eight variables in the array of structures has been exceeded
4	Parameter number in the structure is out of range
5	The CAPTINIT function was called before the STRTSERV function was called

SERCOS Errors Reference Card

Theses errors can appear at the ERR output of certain SERCOS functions/function blocks.

Error #	Description
0	No error
1	IDN queue was busy when called.
2	Quantity specified in the .AVAIL structure member is not large enough for received data
3	Axis is not initialized, is not a SERCOS axis, or the slot/ring/slave specification is incorrect.
4	Invalid data in DATA input structure
5	Error reset function could not be completed.
6	SERCOS ring 1 busy
7	SERCOS ring 2 busy
8	SERCOS ring 1 configuration size error
9	SERVOS ring 2 configuration size error
10	Function block enabled while already in process
11	Bit 3 or bit 8 set in the procedure command acknowledgment (data status) Either operation data invalid or procedure command error
12	Not enough pool memory available
13	Change bit in status word was zero after reference complete.
14	The IDN queue was cleared during an IDN transfer, typically caused by calling the SC_INIT function while an IDN is being read or written.
15	SERCOS module is unavailable for IDN transfer because the phase-to-phase transition in progress is between phase 2 and phase 4.
16	Slave response timed out

17	The SERCOS module did not receive an expected AT response. SERCOS cable may be disconnected.
18	Number of SERCOS slots equals zero.
19	The SERCOS module did not receive an expected MDT response. SERCOS cable may be disconnected.
20	Phase 0 detected that the ring is not complete. The optic cable could be open or drive turned off.
21	The SERCOS module firmware is outdated for the features requested from a newer version of the motion library.
22	The SERCOS module firmware is a newer version and the motion library is outdated and unable to interface.
23	The version of PiCPro used to create the SERCOS setup data is outdated for the features requested from the library or the SERCOS module firmware.
24	The version of PiCPro used to create the SERCOS setup data is a newer version and the library is unable to interface.
25	A two-ring SERCOS module was specified in SERCOS setup but the module is a one-ring SERCOS module.
26	Invalid PRB input on the SCA_RFIT function blocks or invalid OPTN input on the SCA_RFIT function block.
27	The SERCOS setup data is configured for a different CPU (PiC, MMC, or MMC for PC).
28	The SERCOS ring is not currently halted in phase 2, SERCOS setup may not have specified "Pause after Phase 2".
29	The axis is in Resume Mode or Resumable E-stop Allow (REAC_SV/WRITE_SV Variable 63) is set
30	The drive status word (bit 13=1) indicates an error.
31	An E-stop condition exists for this axis in the PiC900.
32	Incorrect phase number, contact Danaher Motion Tech Support.
33	Incorrect address error, contact Danaher Motion Tech Support.
34	Incorrect AT number error, contact Danaher Motion Tech Support.
35	Variable 48 is set to 1 and you have attempted to close the loop
36	OPTN input is invalid.
37	The quantity specified in the .AVAIL structure member is not large enough for the received data. The actual size of the received data is returned in the .ACTUAL structure member. This error is reported by the motion library software.
38	Open loop was requested while SCA_CLOS was in progress.
48	Service channel not ready when attempt to send/receive non-cyclic data
49	No data to send or receive
50	The value of the .SIZE member of the TASK input structure does not match the byte count in the SERCOS module.
51	The value of the .SIZE member of the MAIN input structure does not match the byte count in the SERCOS module.
65	Error occurred calculating when MDT should occur.
66	Error occurred calculating when drive data valid.
67	Error occurred calculating when feedback data valid.

68	Error occurred calculating total time required for communication cycle.
69	Error occurred calculating cyclic data memory for SERCON processor.
70	Error occurred calculating cyclic data memory for internal memory map.
71	Error occurred calculating service channel map.
72	Incorrect ring error, contact Danaher Motion Tech Support.
73	Incorrect AT count error, contact Danaher Motion Tech Support.
74	CPU on SERCOS module has too many tasks during update.
128	Slave error occurred. Read SERR output to identify error. The SLV output indicates the slave number.
136	Slave will not respond in phase 1. The SLV output indicates the slave number.
155	Procedure command error – The slave number can be viewed at the SLV output and the IDN number at the IDN output.
152	CRC error. The bit pattern received by the SERCOS receiver is corrupted.

Servo Error Reference Table

E-stop Error	Bit Location Bits 0-7 are in low byte - Bit 8 is in high byte									Hex value
	8	7	6	5	4	3	2	1	0	(in ladder)
Digital Drive Fault	E									0x8100 (33024)
Digital Drive Fault		E								0x8080 (32896)
ASIU Update Error			E							0x8040 (32832)
SERCOS synchronization error				E						0x8020 (32800)
SERCOS drive E-stop					E					0x8010 (32784)
User-set						E				0x8008 (32776)
Overflow error							E			0x8004 (32772)
Excess error								E		0x8002 (32770)
Loss of feedback									E	0x8001 (32769)

C-stop Error	Bit Location (low byte)								Hex value
	7	6	5	4	3	2	1	0	(in ladder)
Part reference error	E								0x8080 (32896)
Part reference dimension error		E							0x8040 (32832)
Distance or position move dimension error			E						0x8020 (32800)
Feedrate error				E					0x8010 (32784)
Machine reference error					E				0x8008 (32776)
User-defined C-stop						E			0x8004 (32772)
Negative software limit exceeded							E		0x8002 (32770)
Positive software limit exceeded								E	0x8001 (32769)

Programming Error	Bit Location (high byte)								Hex value
	7	6	5	4	3	2	1	0	(in ladder)
Set whenever a P error occurs	X								0x8000 (32768)
Master axis beyond start point					E				0x8800 (34816)
Slave axis beyond start point						E			0x8400 (33792
Master distance not valid							E		0x8200 (33280)
Slave distance not valid								E	0x8100 (33024)

Programming Error	Bit Location (low byte)								Hex value
	7	6	5	4	3	2	1	0	(in ladder)
The FAST axis in the FAST_QUE function moved too far in wrong direction	E								0x8080 (32896)
Profile number not found		E							0x8040 (32832)
Master axis not available			E						0x8020 (32800)
Master start position for lock on								E	0x8001 (32769)

Slave Error Codes

The slave errors listed below appear at the SERR output of certain slave SERCOS functions and will appear on the **Slave Error** line in the **Ring State** dialog.

SERR #	Description
4097	This IDN does not exist.
4105	The data for this IDN may not be accessed.
8193	The name does not exist.
8194	The name transmission is too short.
8195	The name transmission is too long.
8196	The name may not be changed.
8197	The name is write-protected.
12290	The attribute transmission is too short.
12291	The attribute transmission is too long.
12292	The attribute is write-protected at this time.
16385	The units do not exist.
16386	The units transmission is too short.
16387	The units transmission is too long.
16388	The units may not be changed.
16389	The units are write-protected at this time.
20481	The minimum value does not exist.
20482	The minimum value transmission is too short.
20483	The minimum value transmission is too long.
20484	The minimum value may not be changed.
20485	The minimum value is write-protected.
24577	The maximum value does not exist.

24578	The maximum value transmission is too short.
24579	The maximum value transmission is too long.
24580	The maximum value may not be changed.
24581	The maximum value is write-protected.
28674	The data is too short.
28675	The data is too long.
28676	The data may not be changed.
28677	The data is write-protected at this time.
28678	The data is smaller than the minimum value.
28679	The data is larger than the maximum value.
28680	The bit pattern for this IDN is invalid.

STRTSERV Function Errors

#	STRTSERV Function Errors
0	No error
1	Bad user function data
2	Not enough low memory
3	Feedback module(s) not found
4	Analog module(s) not found
5	Update rate (SERCOS axis only)
6	Incorrect CPU type
7	Wrong CPU
8	Incompatible drive firmware
9	Invalid firmware
10	Address not found
11	Duplicate address
12	Digital drive firmware too old
13	Digital drive firmware too new
14	No digital drives found
15	Cyclic data sizes not identical
16	Cyclic failed to start
17	Deallocate servo data memory failure
18	Communication error reading feedback
19	This control does not support the number of axes declared in servo setup.
23	Outdated servo setup data
24	Newer servo setup data

Word Output from STATUSSV Function

Word output from STATUSSV function

Characteristic	Binary value	Hex
Move started	00000000 00000001	0001
Fast input occurred	00000000 00000010	0002
Fast input on	00000000 00000100	0004
Good mark detected	00000000 00001000	0008
Bad mark detected	00000000 00010000	0010
DIST + TOLR exceeded	00000000 00100000	0020
Fast input rising	00000000 01000000	0040

Keyboard Shortcuts

Generic keyboard shortcuts applicable to PiCPro windows that can perform the functions below	
Context Sensitive Help	F1
Copy Item	Ctrl-C
Cut Item	Ctrl-X
Paste Item	Ctrl-V
Insert Item	Insert Key
Delete Item	Delete Key
Find/Find Next	F3
Select All	Ctrl-A

Disk Operations Dialog – Ram and Flash Disks	
Create New Folder	Ctrl-N
Paste List	Ctrl-L
Refresh Display	Ctrl-F
Rename	Ctrl-R

Drive List View – Analog and Digital	
Download	Alt-Shift-D
Edit Drive Data	Ctrl-Shift-D
Edit Drive Name	Ctrl-Shift-N
Enable Drive	Ctrl-Shift-E
Reset Faults	Ctrl-Shift-F
Reset Peaks	Ctrl-Shift-P

Drive Data View – Analog and Digital	
Edit Drive Name	Ctrl-Shift-N
Move to First Cell in Column	Ctrl-Home or Ctrl-Up Arrow
Move to Last Cell in Column	Ctrl-End or Ctrl-Down Arrow
Move to First Cell in Row	Ctrl-Left Arrow
Move to Last Cell in Row	Ctrl-Right Arrow
Move to Next Cell	Tab
Move to Previous Cell	Shift-Tab
Multiple Cell Selection	Shift-Cursor Up/Down

Hardware Declarations	
Paste Insert	Ctrl-A
Select Block I/O	Ctrl-B
Select Master Rack Only	Ctrl-M
Select Remote I/O	Ctrl-R
Select Continue to Scan with Master Rack, Block or ASIU failures	Ctrl-U
Select Continue to Scan with Master Rack, Remote Rack or Block I/O failures	Ctrl-K
Select Extend Pool Memory	Ctrl-P
Save/Close	F10

Ladder Editor	
Go to End of Network	Ctrl-End
Go to End of Ladder	Ctrl-End, Ctrl-End
Go to Top of Network	Ctrl-Home
Go to Top of Ladder	Ctrl-Home, Ctrl-Home
Delete Network	Shift-Delete
Find/Replace	Alt-F3
Insert Coil De-Energize	Ctrl-Shift-D
Insert Coil Energize	Ctrl-Shift-E
Insert Coil Reset (Unlatch)	Ctrl-Shift-R
Insert Coil Set (Latch)	Ctrl-Shift-S
Insert Contact NC	Ctrl-Shift-C
Insert Contact NC, Negative Transition	Ctrl-Shift-A
Insert Contact NC, Positive Transition	Ctrl-Shift-I
Insert Contact NO	Ctrl-Shift-O

Insert Contact NO, Negative Transition	Ctrl-Shift-N
Insert Contact NO, Positive Transition	Ctrl-Shift-P
Insert Data In	Ctrl-Shift-F
Insert Data In Inverted	Ctrl-Shift-W
Insert Data Out	Ctrl-Shift-Q
Insert Wire (both)	Ctrl-Shift-B
Insert Horizontal Wire	Ctrl-Shift-W
Insert Vertical Wire	Ctrl-Shift-V
Insert Network	Shift-Insert
Insert Structured Text	Ctrl-Insert
Insert Jump	Ctrl-Shift-J
Insert Return	Ctrl-Shift-T
Insert Subroutine	Ctrl-Shift-U
Move Horizontal Wire Left	Ctrl-J
Move Horizontal Wire Right	Ctrl-K
Move Vertical Wire Down	Ctrl-M
Move Vertical Wire Up	Ctrl-I
Structured Text - Check Syntax	Ctrl-Alt-S
Structured Text – Declare All Unknown Variables	Ctrl-Alt-D

Ladder – Structured Text Editor	
Cut Line	Ctrl-L
Delete Line	Ctrl-Shift-L
Delete from Cursor to End of Word	Ctrl-Delete
Delete from Cursor to Start of Word	Ctrl-Backspace
Go to Start of Line	Home
Go to End of Line	End
Go to Top of ST Network	Ctrl-Home
Go to End of ST Network	Ctrl-End
Move Cursor to Next Word	Ctrl-Right Arrow
Move Cursor to Prev Word	Ctrl-Left Arrow
Toggle Insert/Overstrike	Insert
Toggle White Space View (Displays tabs and spaces)	Ctrl-Shift-8
Select Word	Ctrl-Shift-W

PiCPro Main Window	
New	Ctrl-N
Open	Ctrl-O
Print	Ctrl-P
Save	Ctrl-S
Switch to Next Window	Alt-F6
Switch to Previous Window	Shift-Alt-F6

Software Declarations	
Insert Symbol	Insert Key
Make an Array	Alt-A
Popup Attributes Menu	Alt-M
Purge Unused	Alt-P

Data Capture

Data Capture Variables

There is a set of variables that can be captured using the data capture functions. The variable number is entered as the VAR member of a structure in the array of structures at the SRCE input of CAPINIT function.

VAR 1 - Actual Position

VAR 2 - Fast Input Occurred

VAR 3 - Commanded Position

VAR 4 - Position Error

VAR 5 - Slow Velocity Error

VAR 6 - Command Change

VAR 7 - Position Change

VAR 8 - Feedback Position

VAR 9 - Prefilter Commanded Position

VAR 10 - Prefilter Command Change

VAR 11 - Remaining Master Offset

VAR 12 - Remaining Slave Offset

VAR 13 - Command Change

VAR 14 - Position Change

VAR 15 - Prefilter Command Change

Data Capture Reference Card

Axis variables that can be captured on a servo interrupt basis with the CAPTINIT function.

VAR	Description	Type
1	Actual position	DINT
2	Fast input	BYTE
3	Commanded position	DINT
4	Position error	DINT
5	Slow velocity filter error	INT
6	Command change	INT

7	Position change	INT
8	Feedback position	DINT
9	Prefilter commanded	DINT
10	Prefilter command change	INT
11	Remaining master offset	DINT
12	Remaining slave offset	DINT
13	Command change	DINT
14	Position change	DINT
15	Prefilter command change	DINT

Data Capture Variable 01 - Actual Position

Enter at VAR 1

Data Capture Variable 1 Actual Position is the same as READ_SV variable 1. Captures the actual position of the device with reference reset applied.

UNITS	TYPE
FU	DINT

Data Capture Variable 02 - Fast Input Occurred

Enter at VAR 2

Data Capture Variable 2 Fast Input Occurred is the same as bit 00000010 out of STATUSSV. It is bit 00001000 of this byte. It is on for one interrupt

UNITS	TYPE
FU	BYTE

Data Capture Variable 03 - Commanded Position

Enter at VAR 3

Data Capture Variable 3 commanded position is the same as READ_SV variable 3. It is the commanded position sent to the servo upgrade.

UNITS	TYPE
FU	DINT

Note: This is the same as actual for a digitizing axis.

Data Capture Variable 04 - Position Error

Enter at VAR 4

Data Capture Variable 4 position error is the same as READ_SV variable 4. It is the error between the filtered output and the actual.

UNITS	TYPE
FU	DINT

Data Capture Variable 05 - Slow Velocity Filter Error

Enter at VAR 5

Data Capture Variable 5 position error is the same as READ_SV variable 5. It represents the accumulated value in the slow velocity filter.

UNITS	TYPE
FU	DINT

Data Capture Variable 06 - Command Change

Enter at VAR 6

Data Capture Variable 6 command change is the same as READ_SV variable 6. It represents the command delta for this interrupt after filter.

UNITS	TYPE
FU/upgrade	INT

Data Capture Variable 07 - Position Change

Enter at VAR 7

Data Capture Variable 7 position change is the same as READ_SV variable 7. It represents the change in actual position for this upgrade.

UNITS	TYPE
FU/upgrade	INT

Data Capture Variable 08 - Feedback Position

Enter at VAR 8

Data Capture Variable 8 position change is the same as READ_SV variable 8. It represents the 24 bit counter from the hardware. The top byte is always 0.

UNITS	TYPE
FU	DINT

Data Capture Variable 09 - Prefilter Commanded Position

Enter at VAR 9

Data Capture Variable 9 prefilter commanded position represents the commanded position prior to the filter.

UNITS	TYPE
FU	DINT

Note: This is the same as actual for a digitizing axis.

Data Capture Variable 10 - Prefilter Command Change

Data Capture Variable 10 prefilter command change represents the command delta for this interrupt before the filter.

UNITS	TYPE
FU	INT

Data Capture Variable 11 - Remaining Master Offset

Data Capture Variable 11 remaining master offset represents the accumulated master offset.

UNITS	TYPE
FU	DINT

Data Capture Variable 12 - Remaining Slave Offset

Data Capture Variable 12 remaining slave offset represents the accumulated slave offset.

UNITS	TYPE
FU	DINT

Data Capture Variable 13 - Command Change

Data Capture Variable 13 command change is the same as READ_SV variable 6 and is the command delta for this interrupt after filter.

UNITS	TYPE
FU	DINT

Data Capture Variable 14 - Position Change

Data Capture Variable 14 position change is the same as READ_SV variable 7 and represents the change in actual position for this upgrade.

UNITS	TYPE
FU	DINT

Data Capture Variable 15 - Prefilter Command Change

Data Capture Variable 15 prefilter command change is the command delta for this interrupt before filter.

UNITS	TYPE
FU	DINT

READ_SV and WRITE_SV Functions

Overview: Read/Write Servo Variables

The READ_SV and WRITE_SV functions allow you to read and write variables for a specified axis. Depending on the variable, the axis may be a servo, digitizing, time, or SERCOS.

READ / WRITE Reference Table

Variables used with the READ_SV (**Read** column) and WRITE_SV (**Write** column) functions. These variables are used with servo (**S**), time (**T**), and/or digitizing (**D**) axes.

V#	Variable Description	Read	Write
1	Actual Position	S, T, D	T
2	Move Type 11 – position move, 12 – distance move, 14 – velocity start 16 – fast or ladder referenced, 18 – ratiopro, 20 – ratiosyn or ratiogr, 22 – ratiocam, 23 – ratioslp, 24 - ratioreal	S	
3	Command Position	S, D	
4	Position Error	S	
5	Slow Velocity Filter Error	S	
6	Command Velocity	S, T	T
7	Position Change	S, D	
8	Feedback Last	S, D	
9	Fast Input Position	S, D	
10	Reg/Ref Position Change	S, D	
11	Consecutive Bad Marks	S, D	S, D
12	Rollover on Position	S, T, D	S, T, D
13	Slave Offset Incremental	S	S
14	Master Offset Incremental	S	S
15	Slave Offset Absolute	S	S
16	Master Offset Absolute	S	S
17	Slave Offset Filter		S
18	Master Offset Filter		S
19	Fast Input Direction		S, D
20	Fast Input Distance	S, D	
21	Reversal Not Allowed	S	S
22	Fast Input Position (software)	S, D	
23	Position (software) of Axis 1 with Fast Input On Axis 2	S, D	S, D
24	Registration Switch	S, D	S, D
25	Fast Queuing	S	S
26	Synchronized Slave Start	S, T, D	S, T, D

27	Backlash Compensation	S	S
28	TTL Feedback	S, D	S, D
29	Reference Switch Position	S	
30	Filter Time Constant	S	S
31	Filter Error Limit	S	S
32	Velocity Compensation Flag	S	S
33	Filter Lag	S	
34	Position Change Over Several Interrupts	S, D	S, D
35	Part Reference Offset	S, D	
36	Software Upper Limit	S	S
37	Software Lower Limit	S	S
38	Commanded Position (before slow velocity filter)	S, D	
39	Following Error Limit	S	S
40	In-Position Band	S	S
41	Current Segment Number	S	
42	Slave Distance into Segment	S	
43	Master Distance into Segment	S	
44	Set User Iteration Command	S	S
45	User Iteration Command	S	S
46	Set User PID Command	S	S
47	User PID Command	S	S
48	Disable Servo Software	S	S
49	(Reserved)		
50	Override Endlimit Check	S	S
51	SERCOS Command Position	S	
55	Queued Move Type 11 – position, 12 – distance, 14 – velocity start, 16 – fast/ladder reference, 18 – ratiopro, 20 ratiosyn/gear, 22 – ratiocam, 23 – ratioslp and 24 - ratioreal	S	
58	SERCOS Modulo Value	S	
59	Command Position Based Master/Slave	S	S
60	Servo Axis S-curve Interpolation	S	S
61	SERCOS Velocity Compensation	S	S
62	Velocity Compensation Filter	S	S
63	Resumable E-Stop Allow	S	
64	Resume Distance	S	
65	Velocity Compensation Factor	S	S
66	Superimposed Move Axis Assignment	S	S
67	Digital Drive Status Word	S, D	
68	Digital Drive Faults	S	
69	Digital Drive Warnings	S	

70	Digital Drive Analog Input	S	
71	Digital Drive Inputs	S	
72	Digital Drive Outputs	S	S
73	Digital Drive Current	S	
74	Digital Drive Average Current	S	
75	Digital Drive Plus Current Limit	S	S
76	Digital Drive Minus Current Limit	S	S
77	Digital Drive Motor Temperature	S	
78	Digital Drive Position Loop Proportional Gain	S	S
79	Digital Drive Position Loop Feedforward	S	S
80	Digital Drive Velocity Loop Proportional Gain	S	S
81	Digital Drive Velocity Loop Integral Gain	S	S
82	Digital Drive Velocity Loop Integrator Inhibit	S	S
83	Digital Drive Velocity Loop Integrator Hold	S	S
84	Digital Drive Current Plus Enable	S	S
85	Digital Drive Current Minus Enable	S	S
86	Digital Drive Ignore Plus Travel Limit	S	S
87	Digital Drive Ignore Minus Travel Limit	S	S
88	Digital Drive Position Loop I-Gain	S	S
89	Digital Drive Predicted Command Velocity	S	
90	Digital Drive Absolute Reference Position		S, D
91	Digital-Drive-to-Control-Communications Errors	S	
92	Control-to-Digital-Drive Communications Errors	S	
93	Virtual Axis Feedback Source Switch	S	S
94	Virtual Axis Feedback		S
95	Distance Between the Last Two Good Marks	S,D	
96	Registration Compensation	S,D	
97	Consecutive Good Marks	S,D	S,D
98	Master Axis Number	S	
99	Actual Velocity	S,D	
100	Fast Input Response Time	S,D	S,D

Variable 01 - Actual Position

Enter at VAR 1

Variable 1 allows you to read the actual position of the axis in ladder units (feedback units with READ_SVF or WRIT_SVF).

With a time axis, it allows you to write the position.

Applies to these types of axes:

READ	WRITE
Servo	Time
Digitizing	
Time	

Range:

For the time axis, the range is +2,147,483,647 to -2,147,483,648 ladder units.

Variable 02 - Move Type

Enter at VAR 2

Variable 2 allows you to read the active move type indicated by one of the following numbers.

11	position move
12	distance move
14	velocity start
16	fast or ladder reference
20	ratiosyn or ratiogr
22	Ratiocam
23	Ratioslp
24	Ratioreal

Applies to these types of axes:

READ	WRITE
Servo	---

Variable 03 - Command Position

Enter at VAR 3

Variable 3 allows you to read the commanded position of the axis in ladder units (feedback units with READ_SVF).

Applies to these types of axes:

READ	WRITE
Servo	---
Digitizing	

Variable 04 - Position Error

Enter at VAR 4

Variable 4 allows you to read the position error in ladder units (feedback units with READ_SVF).

Applies to these types of axes:

READ	WRITE
Servo	---

Note:

With SERCOS where the actual position error is in the drive, internal calculations approximate the position error and bring the approximation out to variable 4. This approximation may vary by the distance moved in one or two updates from the actual position error read from the drive via the service channel.

Variable 05 - Slow Velocity Filter Error

Enter at VAR 5

Variable 5 returns the error of the slow velocity filter in ladder units (feedback units with READ_SVF).

Applies to these types of axes:

READ	WRITE
Servo	---

Variable 06 - Command Velocity

Enter at VAR 6

Variable 6 shows the velocity ramping up and down with the move in ladder units/minute for a servo axis and in ladder units /second (counts/second) for a time axis (feedback units/update with READ_SVF or WRIT_SVF).

Applies to these types of axes:

READ	WRITE
Servo	Time*
Time	

*Do not write a command velocity when running S-curve velocity profiles.

Range:

For the time axis, the range is +2,000,000,000 to -2,000,000,000 ladder units/second.

Variable 07 - Position Change

Enter at VAR 7

Variable 7 reads the distance moved during one interrupt in ladder units/minute for a servo axis and in ladder units/update for a digitizing axis. (Feedback units/update with READ_SVF).

To read the position change over several interrupts, use variable 34.

Applies to these types of axes:

READ	WRITE
Servo	---
Digitizing	

Variable 08 - Feedback Last

Enter at VAR 8

Variable 8 reads the latest feedback position directly from the feedback module in feedback units. (Feedback units with READ_SVF also).

Applies to these types of axes:

READ	WRITE
Servo	---
Digitizing	

Range for various devices

Encoder/Resolver	Counts from 0 to 16,777,215 FU then rolls over. The number returned will count according to the feedback polarity specified in setup.
Analog input	0 to 4095 unipolar; -2048 to 2047 bipolar
TTL	Depends on the number of bits used for position data

Variable 09 - Fast Input Position - Hardware

Enter at VAR 9

Variable 9 reads the axis position when the fast input occurs in feedback units. (Feedback units with READ_SVF also). The module must have been set up to respond to fast inputs through one of these functions:

- FAST_QUE
- FAST_REF
- REGIST
- MEASURE

Applies to these types of axes:

READ	WRITE
Servo	---
Digitizing	

Variable 10 - Registration/Referencing Position Change

Enter at VAR 10

Variable 10 reads the distance position changed in ladder units (feedback units with READ_SVF) due to registration or the last machine reference. This number can be used to allow the ladder to synchronize axes if a slave axis started before registration ever ran.

Applies to these types of axes:

READ	WRITE
Servo	---
Digitizing	

Variable 11 - Consecutive Bad Marks

Enter at VAR 11

Variable 11 reads the number of consecutive bad marks since the last good mark when using registration. You can also write any positive number into variable 11 to set the number of consecutive bad marks. Typically, 0 would be entered to initialize the counter.

When a good mark occurs, this number will be reset to 0. If the number of bad marks exceeds 2,147,483,647, the number returned will "roll over" to -2,147,483,648 and start counting toward 0.

Applies to these types of axes:

READ	WRITE
Servo	Servo
Digitizing	Digitizing

When using READ_SVF or WRIT_SVF functions, the units are feedback units.

Variable 12 - Rollover On Position

Enter at VAR 12

Variable 12 reads the rollover position or writes a rollover position in ladder units (feedback units with READ_SVF or WRIT_SVF). The written rollover position overwrites the one entered in setup.

Applies to these types of axes:

READ	WRITE
Servo	Servo
Digitizing	Digitizing
Time	Time

Range

The range is 1 to 536,870,912 FU. Entering a 0 turns ‘Rollover On Position’ off. Negative values cannot be entered. The OK on the WRITE_SV function will not be set.

Note:

Without rollover on position when 2,147,483,647 is reached, the next number will be -2,147,483,648. The count continues to 0 and back up to 2,147,483,647, etc.

Variable 13 - Slave Offset Incremental

Enter at VAR 13

Variable 13 reads the total remaining slave offset in slave ladder units (feedback units with READ_SVF or WRIT_SVF). It can also write an incremental slave offset. The total incremental offset entered is applied each time the WRITE_SV function is called. The offset cannot be canceled.

Applies to these types of axes:

READ	WRITE
Servo	Servo

Variable 14 - Master Offset Incremental

Enter at VAR 14

Variable 14 reads the total remaining master offset in slave ladder units (feedback units with READ_SVF or WRIT_SVF). It can also write an incremental master offset. The total incremental offset entered is applied each time the WRITE_SV function is called. The offset cannot be canceled.

Applies to these types of axes:

READ	WRITE
Servo	Servo

Variable 15 - Slave Offset Absolute

Enter at VAR 15

Variable 15 reads the absolute slave offset in slave ladder units (feedback units with READ_SVF or WRIT_SVF). It can also write an absolute slave offset. Each time the WRITE_SV function is called with an absolute offset, an offset is applied which equals the difference between the last call and this call. An absolute offset can be canceled by entering an absolute offset of 0.

Applies to these types of axes:

READ	WRITE
Servo	Servo

Variable 16 - Master Offset Absolute

Enter at VAR 16

Variable 16 reads the absolute master offset in slave ladder units (feedback units with READ_SVF or WRIT_SVF). It can also write an absolute master offset. Each time the WRITE_SV function is called with an absolute offset, an offset is applied which equals the difference between the last call and this call. An absolute offset can be canceled by entering an absolute offset of 0.

Applies to these types of axes:

READ	WRITE
Servo	Servo

For examples of incremental and absolute offsets (variables 13, 14, 15, 16), refer to the PiCPro Function/Function Block Reference Guide.

Variable 17 - Slave Offset Filter

Enter at VAR 17

Variable 17 allows you to write a rate in the range of +1 to +101 or -1 to -10001. This range represents the percentage the velocity will increase or decrease to apply the offset. At +101 or -10001, the offset is applied as a step function, which in effect is no filter. This is the default if nothing is entered in WRITE_SV variable 17.

Applies to these types of axes:

READ	WRITE
---	Servo

Variable 18 - Master Offset Filter

Enter at VAR 18

Variable 18 allows you to write a rate in the range of +1 to +101 or -1 to -10001. This range represents the percentage the velocity will increase or decrease to apply the offset. At +101 or -10001, the offset is applied as a step function, which in effect is no filter. This is the default if nothing is entered in WRITE_SV variable 17.

Applies to these types of axes:

READ	WRITE
---	Servo

Variable 19 - Fast Input Direction

Enter at VAR 19

Variable 19 allows you to enter one of the following numbers so that the fast input will be written according to the description given.

#	Description
0	written only on a low to high (rising) transition
1	written only on a high to low (falling) transition
2	written alternating rising and falling beginning with a low to high transition.
3	written alternating falling and rising beginning with a high to low transition

Applies to these types of axes:

READ	WRITE
---	Servo
	Digitizing

Variable 20 - Fast Input Distance

Enter at VAR 20
Variable 20 allows you to read the distance in ladder units between the most recent fast input and the previous fast input. This allows the ladder to measure the distance between two fast inputs. When this variable is used with the MEASURE and REGISTRATION functions, the function must be called first and then the variable read.

This distance can be on of four distances depending on how the direction was defined in variable 19. For examples and more information, see the PiCPro Function/Function Block Reference Guide.

Applies to these types of axes:

READ	WRITE
Servo	---
Digitizing	

Variable 21 - Reversal Not Allowed

Enter at VAR 21
Variable 21 allows you to turn on or off the feature of the slave following the master when the master reverses direction to be turned on or off for the RATIO_GR and RATIOSYN functions. A "0" (the default) allows the slave to follow the master in the reverse direction. A "1" does not allow the slave to follow the master in the reverse direction. WRITE_SV must always be called before the move function. The state of reversal cannot be changed after the move has started.

An overflow E-stop error will occur if the reversed distance exceeds 536,870,912 units in either the plus or minus direction.

Applies to these types of axes:

READ	WRITE
Servo	Servo

Variable 22 - Fast Input Position - Software

Enter at VAR 22
Variable 22 allows you to read the actual software position of the axis in ladder units. Things like the reference value and rollover on position determine this position value. The module must have been set up to respond to fast inputs through the FAST_QUE, FAST_REF, REGIST, or MEASURE functions.

Note: This differs from the variable 9 fast input position that is the hardware latch position.

Applies to these types of axes:

READ	WRITE
Servo	---
Digitizing	

Variable 23 - Position Of Axis 1 With Fast Input On Axis 2 - Software

Enter at VAR 23

Variable 23 allows you to read the position in feedback units of axis 1 when a fast input occurs on axis 2 Both WRITE_SV and READ_SV functions are required to use this variable. The module must have been set up to respond to fast inputs through the FAST_QUE, FAST_REF, REGIST, or MEASURE functions.

Enter the number of the fast input axis (servo or digitizing axis) at the AXIS input of both functions. Enter the number of the axis (servo, digitizing, or time axis) whose position you want to read in the DATA input of the WRITE_SV function. The position is read at the RSLT output of the READ_SV function.

The position of a servo, digitizing, or time axis can be read.

Applies to these types of axes:

READ	WRITE
Servo	Servo
Digitizing	Digitizing

Variable 24 - Registration Switch

Enter at VAR 24

Variable 24 allows you to turn registration on or off for the master or slave axis (bit 0, 1) and allows you to choose whether or not the registration calculations will change the axis position (bit 2).

Set bit 0 to turn off registration compensation for the slave axis. Set bit 1 to turn off registration compensation for the master axis.

Note:

Bit 0 and bit 1 of variable 24 deal with master/slave compensation due to registration. It is important to remember that this compensation affects the master/slave relationship, not the individual axes. The master axis is accessed through the slave axis. The number of the slave axis is entered at the AXIS input of the READ_SV and WRITE_SV functions.

Set bit 2 so that the registration calculations do not change the axis position.

Note:

This bit can be used with a servo axis or a digitizing only axis. When used with a digitizing only axis, bit 0 and bit 1 must be set to zero.

Variable 10 can be read to see how much change there would have been if bit 2 were not set.

Writing a zero to variable 24 returns the registration calculations to normal.

Applies to these types of axes:

READ	WRITE
Servo	Servo
Digitizing	Digitizing

Variable 25 - Fast Queing

Enter at VAR 25

Variable 25 allows you to turn on fast queuing by entering a 1. A move start, abort move, or a fast queue event will now start within one interrupt. When it is set to zero, these activities can take up to eight interrupts to begin. Fast queuing makes your axis more responsive, but there is a trade-off in that the execution time is increased.

When one or more axis is slaved to a master axis that is starting and stopping using distance moves (normally with the SCURVE function), you must also set Fast queuing for each slave axis. This ensures that the slave distances will be reached before the master axis stops.

When doing a synchronized slave start, see the **IMPORTANT** note at variable 26.

Applies to these types of axes:

READ	WRITE
Servo	Servo

Variable 26 - Synchronized Slave Start

Enter at VAR 26

Variable 26 allows you to tell a master axis which of its slave axes must be queued up before any of them begin their move. Each slave axis you want to synchronize is identified by setting a bit in a DINT using the lower 16 bits where the LSB = axis 1 and the MSB = axis 16. When the last "set" axis has been queued, all the slave axes will begin their move on the next interrupt.

WRITE_SV must be called before the move. It can be called again when you want to identify a different set of synchronized slave axes. Change the bits only after the slave axes identified in the first WRITE_SV have started to move.

Writing a zero to variable 26 clears all identified axes.

READ_SV reads the number of the slave axes being synchronized.

IMPORTANT:

Always use fast queuing (variable 25) with this variable. This ensures that the slave axes will be checking for the synchronized slave start flag every interrupt, not just on the next interrupt. Remember that the synchronized slave start variable 26 is set on the master axis and fast queuing variable 25 is set on each slave axes.

Applies to these types of axes:

READ	WRITE
Servo Digitizing Time	Servo Digitizing Time

Variable 27 - Backlash Compensation

Enter at VAR 27

Variable 27 allows you to write a backlash compensation value. Enter the value in ladder units. The amount is added or subtracted from the command whenever the commanded direction is reversed. The value written should equal the amount of mechanical backlash in the gears between the servo motor and the desired motion.

Note:

Because the backlash value is added or subtracted after the commanded position is calculated, the distance moved will not be reflected in variable 3 (commanded position). It will, however, be reflected in variable 1 (actual position).

It is important at power on to ensure that the PiC will compensate for backlash correctly. The PiC assumes that the most recent move is in the positive direction. Program a positive move to "wind up" the backlash in a positive direction before writing to variable 27. Once the initial positive direction has been established, the PiC will compensate for backlash as described above whenever the commanded value changes direction.

READ_SV reads the backlash compensation value in ladder units. (0 - 32767 feedback units) default = 0

Applies to these types of axes:

READ	WRITE
Servo	Servo

Variable 28 - TTL Feedback

Enter at VAR 28

Variable 28 allows you to read the position of the feedback axis by returning the state of 24 TTL inputs to the DINT at the RSLT output of READ_SV. The 24 inputs are the low 24 bits.

Depending on the hardware, the 24th TTL input can be used as an indicator of valid data. When it is used to indicate valid data, then you must monitor a waiting flag at the MSB of the DINT at RSLT.

The waiting flag will be low until the hardware sends valid data to the TTL inputs. Do not attempt to close the loop while the waiting flag is low. When valid data is received, the waiting flag goes high and you can then successfully close the loop.

You can write to the eight TTL outputs using the eight LSBs of the DINT at the DATA input on the WRITE_SV function.

Applies to these types of axes:

READ	WRITE
Servo	Servo
Digitizing	Digitizing

Variable 29 - Reference Switch Position

Enter at VAR 29

Variable 29 allows you to read the position representing the distance between the reference switch and the index mark in feedback units when using encoder feedback.
When using resolver feedback, the position represents the absolute position of switch closure in feedback units. When using analog input or TTL feedback, the position represents the absolute position when referencing occurred.

Note:

The number returned in variable 29 always counts in the same direction regardless of the feedback polarity specified in setup.

This measurement could be in error up to the distance traveled in eight updates. You can reduce that error to no more than the distance traveled in one update by setting fast queuing variable 25 using the WRITE_SV function.

Applies to these types of axes:

READ	WRITE
Servo	---

Variable 30 - Filter Time Constant

Enter at VAR 30

Variable 30 allows you to define a first order filter on the master axis. In some applications it is necessary to filter the master delta to control variations that can occur in master axis travel. There are 10 approximate filter values:

2	64
4	128
8	256
16	512
32	1024

The time constant has a fine resolution at low values and a coarse resolution at high values.

Identify the slave axis at the AXIS input of READ_SV or WRITE_SV.

Related master filter variables: 31, 32, 33.

(1 - 1023, 0 disables filter)

Applies to these types of axes:

READ	WRITE
Servo	Servo

Variable 31 - Filter Error Limit

Enter at VAR 31

Variable 31 allows you to limit the amount of lag introduced by the filter. When this limit is reached, the filter will no longer be in effect. This allows you to implement a large filter at low velocities when resolution problems are more pronounced and still limit the following error effects at high velocities when filtering is not required. A positive number is entered using WRITE_SV. It applies to both positive and negative errors.

Identify the slave axis at the AXIS input of READ_SV or WRITE_SV.

Related master filter variable: 30, 32, 33.

(1 - 32767 feedback units)

Applies to these types of axes:

READ	WRITE
Servo	Servo

Variable 32 - Velocity Compensation Flag

Enter at VAR 32

Variable 32 allows you to turn off the default velocity compensation feature by entering a 1. Turning it off will result in the slave axis lagging the master axes by the amount traveled by the master axis in one interrupt.

Note: Velocity compensation works independent of the filter.

Identify the slave axis at the AXIS input of READ_SV or WRITE_SV.

Related master filter variable: 30, 31, 33.

(0, 1)

Applies to these types of axes:

READ	WRITE
Servo	Servo

Variable 33 - Filter Lag

Enter at VAR 33

Variable 33 allows you to read the filter following error.

Identify the slave axis at the AXIS input of READ_SV or WRITE_SV.

Related master filter variable: 30, 31, 33.

(-32768 - 32767 feedback units)

Applies to these types of axes:

READ	WRITE
Servo	---

Note:

Normally, the filter time constant and error limit will be established prior to the move call. If they are changed after the slave axis is locked to the master axis, keep the following in mind:

- If the filter lag is already at the filter error limit and the error is increased, the new limit will be reached at the rate defined by the filter and master axis velocity.
- If the filter lag is already at the filter error limit and the error is decreased, the excess will be dumped into the slave axis command in one update.
- If the filter lag is already at the filter error limit, changing the time constant will have no effect.
- If the filter time constant is set to zero, any lag will remain.

Variable 34 - Position Change Over Several Interrupts

Enter at VAR 34

Variable 34 allows you to read the change in position over several interrupts. Variable 7 reads the change in position in a single interrupt. However, it can be difficult to get an accurate reading in one interrupt especially if an axis is moving slowly.

Write at the DATA input of WRITE_SV the number of interrupts (0 to 255) over which the change in position will be summed. Writing a zero to the DATA input turns the feature off.

Read with READ_SV the distance moved over several interrupts in ladder units for a servo or digitizing axis. The value is not necessarily changed every interrupt. It changes only after the number of interrupts designated with WRITE_SV have occurred since the last value was read. **Note**: A non-zero value must be written with WRITE_SV before you call READ_SV or the READ_SV OK will not be set.

An overflow can occur if the axis is moving fast and the number of interrupts selected is large. If an overflow occurs, the OK of READ_SV will not be set. Write to variable 34 to clear an overflow error condition.

Applies to these types of axes:

READ	WRITE
Servo	Servo
Digitizing	Digitizing

Variable 35 - Part Reference Offset

Enter at VAR 35

Variable 35 allows you to read the part reference offset in ladder units. The offset represents the distance that would have to be subtracted from the current position to remove the part reference.

Applies to these types of axes:

READ	WRITE
Servo	---
Digitizing	

Variable 36 - Software Upper Limit

Enter at VAR 36

Variable 36 allows you to read or write in ladder units the upper end limit for a servo axis. Exceeding the end limit will generate a C-stop.

The range is -536870912 to 536870911 feedback units.

Applies to these types of axes:

READ	WRITE
Servo	Servo

Variable 37 - Software Lower Limit

Enter at VAR 37

Variable 37 allows you to read or write in ladder units the lower end limit for a servo axis. Exceeding the end limit will generate a C-stop.

The range is -536870912 to 536870911 feedback units.

Applies to these types of axes:

READ	WRITE
Servo	Servo

Variable 38 - Commanded Position

Enter at VAR 39

Variable 38 allows you to read the commanded position before the slow velocity filter is applied to a servo axis. If the slow velocity filter is not in effect, it returns the same commanded position as variable 3 returns.

Applies to these types of axes:

READ	WRITE
Servo	---
Digitizing	

Variable 39 - Following Error Limit

Enter at VAR 39

Variable 39 allows you to read or write in ladder units the following error limit for a servo axis. This overrides the following error limit entered in servo setup.

The range is -536870912 to 536870911 feedback units.

Applies to these types of axes:

READ	WRITE
Servo	Servo

Variable 40 - In-Position Band

Enter at VAR 40

Variable 40 allows you to read or write in ladder units the in-position band for a servo axis. This overrides the in-position band entered in servo setup.

The range is -536870912 to 536870911 feedback units.

Applies to these types of axes:

READ	WRITE
Servo	Servo

Variable 41 - Current Segment Number

Enter at VAR 41

Variable 41 allows you to read the segment number from a RATIOCAM, RATIOSLP, or RATIO_RL move currently being executed. The first segment is number 1. This matches the array element number in the profile. If one of the three above moves is not being executed, the OK of READ_SV will be clear.

Applies to these types of axes:

READ	WRITE
Servo	---

Variable 42 - Slave Distance Into Segment

Enter at VAR 42

Variable 42 allows you to read the distance the slave axis is into the segment identified in variable 41 from a RATIOCAM, RATIOSLP, or RATIO_RL move. If one of the three moves is not being executed, the OK of READ_SV will be clear.

The units are feedback units.

Applies to these types of axes:

READ	WRITE
Servo	---

Variable 43 - Master Distance Into Segment

Enter at VAR 43

Variable 43 allows you to read the distance the master axis is into the segment identified in variable 41 from a RATIOCAM, RATIOSLP, or RATIO_RL move. If one of the three moves is not being executed, the OK of READ_SV will be clear.

The units are feedback units.

Applies to these types of axes:

READ	WRITE
Servo	---

Variable 44 - Set User Iteration Command

Note:

Refer to the section on background information for variables 44 through 48 found in the PiCPro Function/Function Block Reference Guide.

Enter at VAR 44

Variable 44, when set to one, allows you to use the User Iteration Command before the slow velocity filter. The User Iteration Command is written with variable 45. A valid value should be written to variable 45 before variable 44 is set to one.

0 = use Servo Iteration Command (default)

1 = use User Iteration command before PID calculations

Applies to these types of axes:

READ	WRITE
Servo	Servo

Variable 45 - User Iteration Command

Note:

Refer to the section on background information for variables 44 through 48 found in the PiCPro Function/Function Block Reference Guide.

Enter at VAR 45

Variable 45 allows you to read the result of the Servo Iteration Command and write the User Iteration Command to the input of the next PID calculations when variable 44 is set to one.

To zero the command, a zero must be written with variable 45. Otherwise, the most recent write value will be in effect.

The range is -32768 to 32767 FU/update.

Applies to these types of axes:

READ	WRITE
Servo	Servo

Variable 46 - Set User PID Command

Note:

Refer to the section on background information for variables 44 through 48 found in the PiCPro Function/Function Block Reference Guide.

Enter at VAR 46

Variable 46, when set to one, allows you to use the User PID Command after the PID calculation and before the D/A command. You can then write a User PID Command with variable 47. A valid PID command should be written to variable 47 before variable 46 is set to one.

0 = use Servo PID Command (default)

1 = use User PID Command

Applies to these types of axes:

READ	WRITE
Servo	Servo

Variable 47 - User PID Command

Note:

Refer to the section on background information for variables 44 through 48 found in the PiCPro Function/Function Block Reference Guide.

Enter at VAR 47

Variable 47 allows you to read the output of the Servo PID Command that is to be sent to the D/A and write a User PID command when variable 46 is set to one. To zero the PID command, a zero must be written with variable 47. Otherwise, the most recent write value will be in effect.

The range is -32768 to 32767 D/A bits.

Applies to these types of axes:

READ	WRITE
Servo	Servo

Variable 48 - Disable Servo Software

Note:

Refer to the section on background information for variables 44 through 48 found in the PiCPro Function/Function Block Reference Guide.

Enter at VAR 48

Variable 48, when set to one, allows you to control the D/A command with the ANLG_OUT function instead of the servo software. The most recent value from the servo software or the ANLG_OUT function remains in effect regardless of any E-stop or other fault conditions.

0 = use servo software (default)

1 = disable servo software (use ANLG_OUT function for D/A command)

Applies to these types of axes:

READ	WRITE
Servo	Servo

Variable 50 - Override End Limit Check

Enter at VAR 50

Variable 50 allows you to disable end limit checking whether referencing has occurred or not. It is used primarily when you want to ignore end limits even though referencing has occurred.

0 = end limit check (default)

1 = ignore end limit check even if reference has occurred.

Applies to these types of axes:

READ	WRITE
Servo	Servo

Variable 51 - SERCOS Command Position

Enter at VAR 51

Variable 51 allows you to read or write the SERCOS position in servo software.

The value is in feedback units.

Applies to these types of axes:

READ	WRITE
Servo	---

Variable 55 - Queued Move Type

Enter at VAR 55

Variable 55 allows you to read the number of the move type in the queue.

11	position move
12	distance move
14	velocity start
16	fast or ladder reference
20	ratiosyn or ratiogr
22	ratiocam
23	ratioslp
24	ratioreal

Applies to these types of axes:

READ	WRITE
Servo	---

Variable 58 - SERCOS Modulo Value

Enter at VAR 58

Variable 58 tells the control what the SERCOS drive's modulo value is for its SERCOS digitizing axis. Writing to this variable will not change the modulo value in the SERCOS drive. This only tells the control what modulo value the SERCOS drive is using so the control can account for the rollover in the feedback value it's reading from the drive.

The rule for using this variable is:

```
If the axis is a SERCOS digitizing axis AND
 Bit 7 (modulo format) of IDN 76 (position data scaling type) is set AND
IDN 103 (modulo value) is non-zero
THEN
 Write the modulo value to variable 58 via WRITE_SV
```

Writing a value of zero to this variable tells the control that modulo format is not being used in the SERCOS drive. Writing variable 58 is only required for a SERCOS digitizing axis. The modulo flag and value for a SERCOS servo axis are read from the drive when the SCA_CLOS function block is executed.

Reading this variable with READ_SV will not read the modulo value from the drive. It will only read what is currently stored in variable 58.

Applies to these types of axes:

READ	WRITE
Servo	Servo
Digitizing	Digitizing

Variable 59 - Command Position Based Master/Slave

Enter at VAR 59

Variable 59 reads/writes if RATIO_GR, RATIOSYN, and RATIOPRO will base their slave axis motion on the master axis' actual position or command position. The slave axis number is entered at the AXIS input. When writing this variable, the WRITE_SV function must be executed prior to the execution of the RATIO_GR, RATIOSYN, or RATIOPRO function. Velocity compensation should be inhibited (Variable 32 = 1) when using this feature.

0 = Use master's actual position (default)

1 = Use master's command position

Note:

This variable must be 0 if the master axis is a time or digitizing axis, otherwise a "Master axis is not available" P-error will occur when the RATIO move is attempted.

Applies to these types of axes:

READ	WRITE
Servo	Servo

Variable 60 - Servo Axis S-Curve Interpolation

Enter at VAR 60

Variable 60 indicates/selects whether S-Curve Interpolation or Linear Ramp Interpolation will be used when the axis is accelerating and decelerating. Writing this variable is only allowed if the axis "Enable S-Curve" checkbox is checked in Servo Setup.

0 = the current acceleration ramp and deceleration ramp will be used to accelerate and decelerate the axis

1 = the current S-Curve will be used to accelerate and decelerate the axis

Applies to these types of axes:

READ	WRITE
Servo	Servo

Variable 61 - Multiple Interrupt Velocity Compensation

Enter at VAR 61

Variable 61 selects whether multiple interrupt velocity compensation or normal velocity compensation will be used for master/slave moves. The slave axis number should be specified at the AXIS input.

Valid Range = [0,1]

0 = Use normal single-interrupt velocity compensation (default)

1 = Use multiple-interrupt velocity compensation (default for digital drive axes)

Normal velocity compensation compensates for the inherent one-interrupt position lag that occurs between master and slave axes when the master's actual position is used to command the slave axis. With SERCOS and digital drive axes, there is a multiple-interrupt lag that occurs.

With SERCOS axes, the "Position Error Cyclic Update Offset" specified in Servo Setup is used to determine the correct number of interrupts. The offset that provides the correct position error in the control will also be the offset that provides the correct amount of velocity compensation.

With digital drive axes, the correct number of interrupts is 3. These defaults are established when STRTSERV or DSTRTSRV executes.

Due to the larger amount of velocity compensation, SERCOS and digital drive axes will default to using a velocity compensation filter to avoid an overly sensitive slave axis. This will cause some lag to occur between the master and slave positions during master acceleration and deceleration. This filter can be adjusted or eliminated with Variable 62. See Variable 62 for a description of the velocity compensation filter.

If Variable 32 Velocity Compensation Flag = 1, all velocity compensation will be inhibited regardless of the state of variable 61.

Applies to these types of axes:

READ	WRITE
Servo	Servo

Variable 62 - Velocity Compensation Filter

Enter at VAR 62

Variable 62 specifies the number of servo interrupts in which a given amount of velocity compensation will be applied to a slave axis

Valid Range = [1, 20]

1 = the amount of velocity compensation calculated for a given interrupt will be applied in 1 interrupt (i.e. no filter)

2 = the amount will be divided up and applied over the next 2 interrupts

3 = the amount will be divided up and applied over the next 3 interrupts

:

:

20 = the amount will be divided up and applied over the next 20 interrupts

Increasing this value will reduce the sensitivity of the slave to changes in the master's velocity but will increase the amount of master/slave position lag that will occur during master acceleration and deceleration. Reducing this value will reduce or eliminate the amount of master/slave position lag that will occur during master acceleration and deceleration but will increase the sensitivity of the slave to changes in the master's velocity. Note that this master/slave position lag only occurs during acceleration and deceleration of the master axis.

Default Value = 1, for analog interface servo axes
= 3, for digital drive axes
= Position Error Cyclic Update Offset + 6 for SERCOS axes

Applies to these types of axes:

READ	WRITE
Servo	Servo

Variable 63 - Resumable E-Stop Allow

Enter at VAR 63

Variable 63 selects whether the User-Set E-Stop (E_STOP function) and the Excess Following Error E-Stop will be resumable.

When this variable is 1, the E_STOP function and the Excess Following Error E-Stop will execute a Resumable E-Stop. When a Resumable E-Stop occurs, the following happens:

1. The servo loop is opened
2. Zero voltage is sent to the analog outputs.
3. The moves in the active and next queues remain intact.
4. The axis' Normal Interpolator remains running.
5. The axis goes into Resume Mode. In Resume Mode, the axis will follow the Resume Interpolator. The Resume Interpolator will output zero velocity until the RESUME function is called. The RESUME function can only be called after the Resumable E-Stop has been reset and the servo loop has been closed. The axis remains in Resume Mode until the RESUME function brings it back on path or until a non-resumable E-Stop occurs and cancels Resume Mode.

When this variable is 0, the E_STOP function and the Excess Following Error E-Stop will execute a normal E-Stop (i.e. open the servo loop, zero voltage to the Analog outputs, and clear the active and next queues).

This variable is initialized by STRTSERV based on the selection in Servo Setup.

Note:

The E_STOP function and the Excess Following Error E-Stop are the only types of E-Stops that are resumable. All other types of E-Stops will execute normally regardless of the state of this variable.

Also see READ_SV Variable 64, RESMODE?, and RESUME.

Applies to these types of axes:

READ	WRITE
Servo	Servo

Variable 64 - Resume Distance

Enter at VAR 64

Variable 64 reads the signed distance between the Resume Interpolator's command position and the Normal Interpolator's command position in ladder units. This value determines the direction and distance of a RESUME move. If this value is positive, the RESUME function will cause the axis to move in the positive direction. If this value is negative, the RESUME function will cause the axis to move in the negative direction. This value is only valid when the axis is in Resume Mode.

Also see READ_SV Variable 63, RESMODE?, and RESUME.

Applies to these types of axes:

READ	WRITE
Servo	---

Variable 65 - Velocity Compensation Factor

Enter at VAR 65

Variable 65 specifies the value used to multiply the change in the master axis feedback delta when calculating the slave axis's velocity compensation. Refer to Variable 61 for a description of Multiple Interrupt Velocity Compensation. The slave axis number should be specified at the AXIS input.

For most applications, the default value will be correct and this variable should not be changed. However, the exception to this rule is described in the following SERCOS NOTE.

SERCOS Note:

For SERCOS axes, this value defaults to [Position Error Cyclic Update Offset + 6]. The Position Error Cyclic Update Offset is entered in Servo Setup. It is used to correctly calculate the position error of a SERCOS axis. In most cases, this default value will be the correct value to eliminate any position lag between SERCOS master and slave axes. However, if the master and slave axes are different types, (i.e. one analog interface and the other SERCOS) or the master and slave SERCOS drives are different (i.e. different manufacturer), it may be necessary to change this value to eliminate position lag between the master and slave axes. If this value is changed, it may also be desirable to change the velocity compensation filter. Refer to Variable 62 for a description of the velocity compensation filter.

Valid Range = [1, 20]

1 = The change in the master axis feedback is not multiplied prior to calculating the slave axis's velocity compensation. In other words, it will operate exactly like normal velocity compensation.

2 = The change in the master axis feedback delta will be multiplied by 2 when calculating the slave axis's velocity compensation.

:

:

20 = The change in the master axis feedback delta will be multiplied by 20 when calculating the slave axis's velocity compensation.

Default Value = 1, for analog interface servo axes
= 3, for digital drive axes
= Position Error Cyclic Update Offset + 6 for SERCOS axes

Applies to these types of axes:

READ	WRITE
Servo	Servo

Variable 66 - Superimposed Move Axis Assignment

Enter at VAR 66

Variable 66 activates or cancels the Superimposed Move feature. This feature allows the ladder to add a move on top of an axis's current move.

Writing a valid servo axis number to this variable turns on the Superimposed Move feature by assigning that axis (the Superimposed Move axis) to the axis specified at the AXIS input. (the Receiving axis). After this variable is written, any move executed by the Superimposed Move axis will be added on top of the current move of the Receiving axis. Internally, this is performed by adding the iteration command of the Superimposed Move axis to the iteration command of the Receiving axis. The Receiving Axis will only accept the additional command while it is executing a RATIO_ move or a VEL_STRT move. A typical application will specify a virtual axis for this variable. Writing a value of 0 cancels the Superimposed Move axis assignment. Both the Receiving axis and the Superimposed Move axis must have the same servo update rate.

Reading this variable will return the axis number of the Superimposed Move axis. A returned value of 0 indicates there is no Superimposed Move axis assigned.

Valid Range = [0, 16] and [101, 116]
Default Value = 0

Notes:

If Variable 44 (User Iteration Command) is set to 1, Superimposed Moves will not be applied.

Reading Variable 45 (Iteration Command) will not return the combined iteration command of the specified axis and the assigned Superimposed Move axis. It will return only the iteration command of the specified axis.

Applies to these types of axes:

READ	WRITE
Servo	Servo

Variable 67 - Digital Drive Status Word

Enter at VAR 67

Variable 67 reads the following digital drive states. Each bit represents a state.

0000 0001H	Startup Commutation Complete
0000 0002H	At Zero Speed
0000 0004H	In Speed Window
0000 0008H	Up to Speed
0000 0010H	At Plus Current Limit
0000 0020H	At Minus Current Limit
0000 0040H	Drive Bus Charged
0000 0080H	Drive Enabled
0000 0100H	Drive Ready
0000 0200H	Release Brake
0000 0400H	Drive Fault
0000 0800H	Drive Warning
0000 1000H	220V Shunt on 440V Drive
0000 2000H	Drive Ready and Bus Charged
0100 0000H	Hardware Enable Line
0200 0000H	Auxiliary Feedback Loss-Of-Feedback

All other bits are reserved

Applies to these types of axes:

READ	WRITE
Servo	---
Digitizing	

Variable 68 - Digital Drive Faults

Enter at VAR 68

Variable 68 reads any faults currently active in the digital drive. Each bit represents a fault.

0000 0000H	No fault
0000 0001H	Drive Memory Fault
0000 0002H	Drive Bus Over Voltage Fault
0000 0004H	Drive PM1 Over Current Fault
0000 0008H	Drive Bus Under Voltage Fault
0000 0010H	Motor Temperature Fault
0000 0020H	Continuous Current Fault
0000 0040H	Drive Heatsink Temperature Fault

0000 0080H	Drive F2 Feedback Fault
0000 0100H	Drive F1 Feedback Fault
0000 0200H	Drive Ambient Temperature Fault
0000 0400H	Motor Calculated Temperature Fault
0000 0800H	Drive Timing Fault
0000 1000H	Drive Interface Fault
0000 2000H	User Set Fault
0000 4000H	Drive F1 Communication Fault
0000 8000H	Over Speed Fault
0001 0000H	Over Current Fault
0002 0000H	Control Panel Disconnect Fault
0004 0000H	Drive Power Module Fault
0008 0000H	Feedback Type Mismatch Fault
0010 0000H	(reserved)
0020 0000H	Drive Relay Fault
0040 0000H	Drive PM2 Over Current Fault
0080 0000H	Drive PM Temperature Fault
0100 0000H	Motor Ground Fault
0200 0000H	Drive AC Input Over Voltage Fault
0400 0000H	Overtravel Plus Fault
0800 0000H	Overtravel Minus Fault
1000 0000H	Digital Link Communication Error
2000 0000H	Invalid Switch Setting Fault
4000 0000H	(reserved)
8000 0000H	(reserved)

Applies to these types of axes:

READ	WRITE
Servo	---

Variable 69 - Digital Drive Warnings

Enter at VAR 69

Variable 69 reads any warnings currently active in the digital drive. Each bit represents a warning.

0000 0000H	No warnings
0000 0001H	Drive Heatsink Temperature Warning
0000 0002H	Drive Ambient Temperature Warning
0000 0004H	Motor Temperature Warning
0000 0008H	Motor Calculated Temperature Warning
0000 0010H	Loss of Feedback on Aux Warning

All other bits are reserved.

Applies to these types of axes:

READ	WRITE
Servo	---

Variable 70 - Digital Drive Analog Input

Enter at VAR 70

Variable 70 reads a value representing the voltage at the digital drive's analog input. The value is in the range [-8192, 8191] where 8192 counts = 10 volts.

For example:

```
8191 = 10V
4096 = 5V
0 = 0V
-4096 = -5V
-8192 = -10V
```

The following formula can be used to calculate the voltage:

```
Voltage = Variable70 * 10V / 8192
```

Applies to these types of axes:

READ	WRITE
Servo	---

Variable 71 - Digital Drive Inputs

Enter at VAR 71

Variable 71 reads the states of the digital drive inputs. Each bit represents the state of one input.

```
Bit 0 = Input 1
Bit 1 = Input 2
Bit 2 = Input 3
Bit 3 = Input 4
Bit 4 = Input 5
Bit 5 = Input 6
Bit 6 = Input 7
Bit 7 = Input 8
Bits 8 through 31 are undefined
```

0 means the input if OFF
1 means the input is ON

Note:

"MMC Application Input" must be checked as an Input Assignment in PiCPro for each input that is to be read here. If not, the bit representing that input will always be 0.

Applies to these types of axes:

READ	WRITE
Servo	---

Variable 72 - Digital Drive Outputs

Enter at VAR 72

Variable 72 reads the states of the digital drive outputs. Each bit represents the state of one output.

```
Bit 0 = Output 1
Bit 1 = Output 2
Bit 2 = Output 3
Bit 3 = Output 4
Bit 4 = Output 5
Bits 5 through 31 are undefined
```

0 means the output if OFF

1 means the output is ON

Writing this variable will set or reset the digital drive outputs. Write a bit to 0 to turn OFF an output and write a bit to 1 to turn ON an output. If other states are assigned to an output by PiCPro, the following logic is used to determine if the output should be turned ON or OFF:

The output will be turned ON if this bit is ON OR any of the other assigned states are ON.

The output will be turned OFF if this bit is OFF AND all other assigned states are OFF.

Note:

"MMC Application Output" must be checked as an Output Assignment in PiCPro for each output that is to be written by this variable. If not, writing this variable will have no effect on the digital drive output.

Applies to these types of axes:

READ	WRITE
Servo	Servo

Variable 73 - Digital Drive Current

Enter at VAR 73

Variable 73 reads the current in the range [-25500, 25500] where the units represent .01 amps.

For example:

```
25500 = 255.00 amps
165 = 1.65 amps
-14554 = -145.54 amps
```

Applies to these types of axes:

READ	WRITE
Servo	---

Variable 74 - Digital Drive Average Current

Enter at VAR 74

Variable 74 reads the average current in the range [0, 25500] where the units represent .01 amps.

For example:

```
25500 = 255.00 amps
165 = 1.65 amps
```

Applies to these types of axes:

READ	WRITE
Servo	---

Variable 75 - Digital Drive Plus Current Limit

Enter at VAR 75

Variable 75 reads/writes the plus current limit in the range [0, 25500] where the units represent .01 amps.

For example:

```
25500 = 255.00 amps
165 = 1.65 amps
```

The digital drive will limit this value to the lesser of the Motor Maximum Current and the Drive Maximum Current.

Applies to these types of axes:

READ	WRITE
Servo	Servo

Variable 76 - Digital Drive Minus Current Limit

Enter at VAR 76

Variable 76 reads/writes the minus current limit in the range [0, 25500] where the units represent .01 amps.

For example:

```
25500 = 255.00 amps
165 = 1.65 amps
```

The digital drive will limit this value to the lesser of the Motor Maximum Current and the Drive Maximum Current.

Applies to these types of axes:

READ	WRITE
Servo	Servo

Variable 77 - Digital Drive Motor Temperature

Enter at VAR 77

Variable 77 returns the motor temperature. If the motor has a thermistor, the temperature is returned in degrees C. If the motor has a thermal switch, 0 is returned if the switch is open and 1 is returned if the switch is closed. If the motor has neither a thermistor nor a thermal switch, the calculated temperature is returned in degrees C.

Applies to these types of axes:

READ	WRITE
Servo	---

Variable 78 - Digital Drive Position Loop Proportional Gain

Enter at VAR 78

Variable 78 reads/writes the position loop proportional gain in the range [0, 32767]. The units are:

`feedback units / minute / feedback units of following error`

Applies to these types of axes:

READ	WRITE
Servo	Servo

Variable 79 - Digital Drive Position Loop Feedforward

Enter at VAR 79

Variable 79 reads/writes position loop feedforward in the percentage of feedforward applied to the digital drive's position loop. The range is [0%, 100%].

Applies to these types of axes:

READ	WRITE
Servo	Servo

Variable 80 - Digital Drive Velocity Loop Proportional Gain

Enter at VAR 80

Variable 80 reads/writes velocity loop proportional gain in the range [0, 32767] representing values in the range [0.0, 3276.7]

Applies to these types of axes:

READ	WRITE
Servo	Servo

Variable 81 - Digital Drive Velocity Loop Integral Gain

Enter at VAR 81

Variable 81 reads/writes velocity loop integral gain in the range [0, 32767].

Applies to these types of axes:

READ	WRITE
Servo	Servo

Variable 82 - Digital Drive Velocity Loop Integrator Inhibit

Enter at VAR 82

Variable 82 enables/disables ‘velocity loop integrator inhibit’:

0 = do not inhibit the digital drive velocity loop integrator

1 = inhibit the digital drive velocity loop integrator

Applies to these types of axes:

READ	WRITE
Servo	Servo

Variable 83 - Digital Drive Velocity loop Integrator Hold

Enter at VAR 83

Variable 83 enables/disables ‘velocity loop integrator hold’:

0 = do not hold the digital drive velocity loop integrator

1 = hold the digital drive velocity loop integrator

Applies to these types of axes:

READ	WRITE
Servo	Servo

Variable 84 - Digital Drive Current Plus Enable

Enter at VAR 84

Variable 84 enables/disables ‘current plus enable’:

0 = disable digital drive plus current

1 = enable digital drive plus current (default)

Applies to these types of axes:

READ	WRITE
Servo	Servo

Variable 85 - Digital Drive Current Minus Enable

Enter at VAR 85

Variable 85 enables/disables ‘current minus enable’:

0 = disable digital drive minus current

1 = enable digital drive minus current (default)

Applies to these types of axes:

READ	WRITE
Servo	Servo

Variable 86 - Prevent Digital Drive Overtravel Plus Fault

Enter at VAR 86

Variable 86 enables/disables ‘ignore plus travel limit’:

0 = enforce the digital drive plus travel limit

1 = ignore the digital drive plus travel limit

Applies to these types of axes:

READ	WRITE
Servo	Servo

Variable 87 - Prevent Digital Drive Overtravel Minus Fault

Enter at VAR 87

Variable 87 enables/disables ‘ignore minus travel limit’:

0 = enforce the digital drive minus travel limit

1 = ignore the digital drive minus travel limit

Applies to these types of axes:

READ	WRITE
Servo	Servo

Variable 88 - Digital Drive Position Loop I-Gain

Enter at VAR 88

Variable 88 reads/writes the integral gain value to be used in the digital drive's position loop. The units are: `((FU / min) * 1000) / (FUFE * min).`

Applies to these types of axes:

READ	WRITE
Servo	Servo

Variable 89 - Digital Drive Predicted Command Velocity

Enter at VAR 89

Variable 89 reads the command velocity that can be used to perform a smooth transition when switching the digital drive to Velocity Mode. See DVELCMD. The units are RPM, motor revolutions / min.

Applies to these types of axes:

READ	WRITE
Servo	---

Variable 90 - Digital Drive Absolute Reference Position

Enter at VAR 90

Variable 90 specifies the position value to be assigned to the current position of a digital drive axis with an absolute feedback device. This is a one-time setup operation. This value is sent to the digital drive and the drive will retain this reference position through power cycles. When this value is sent to the digital drive, the current rollover position is also sent. The digital drive uses the rollover position to properly calculate the absolute position on subsequent power cycles. (Therefore, if the rollover position is ever changed by the ladder with WRITE_SV Variable 12, this absolute reference will need to be performed again.) The function REF_DNE? will indicate that the absolute reference is complete. Events that will clear this reference are:

- Drive scaling changed
- Loss-of-feedback occurred
- Location of F1 or F2 feedback has changed
- PiCPro's "Clear Absolute Reference" was selected by the user
- Motor is changed

The valid range is [0, 2147483647].

Also, after feedback scaling is applied in the digital drive, the result must be in the range [0, 4294967295].

Applies to these types of axes:

READ	WRITE
---	Servo
	Digitizing

Variable 91 - Digital-Drive-to-Control Communication Errors

Enter at VAR 91

Variable 91 reads the number of communication errors detected in messages sent from the digital drive to the MMCD.

Applies to these types of axes:

READ	WRITE
Servo	---

Variable 92 - Control-to-Digital-Drive Communication Errors

Enter at VAR 92

Variable 92 reads the number of communication errors detected in messages sent from the MMCD to the digital drive.

Applies to these types of axes:

READ	WRITE
Servo	---

Variable 93 - Virtual Axis Feedback Source Switch

Enter at VAR 93

Variable 93 selects the source of the virtual axis's feedback.

- 0 - The control automatically provides a feedback value (default)
- 1 - The ladder provides a feedback value via WRITE_SV variable 94.

Applies to these types of axes:

READ	WRITE
Servo	Servo

Variable 94 - Virtual Axis Feedback

Enter at VAR 94

Variable 94 allows you to provide the feedback value for a virtual axis. The control assumes the value is in feedback units in the range -8388608 to 8388607. This will only have an effect when WRITE_SV variable 93 is 1.

Applies to these types of axes:

READ	WRITE
---	Servo

Variable 95 - Distance between the Last Two Good Marks

Enter at VAR 95

Variable 95 returns the distance, in ladder units, between the last two good registration marks. READ_SVF returns this value in feedback units.

Applies to these types of axes:

READ	WRITE
Servo	---
Digitizing	

Variable 96 - Registration Compensation

Enter at VAR 96

Variable 96 returns the most recent registration compensation value in ladder units. READ_SVF returns this value in feedback units.

Applies to these types of axes:

READ	WRITE
Servo	---
Digitizing	

Variable 97 - Consecutive Good Marks

Enter at VAR 97

Variable 97 returns the number of consecutive good registration marks since the last bad registration mark. When a bad mark occurs, this value will be reset to 0.

Any number can be written to this value via WRITE_SV. Typically, 0 would be written to initialize this value.

Applies to these types of axes:

READ	WRITE
Servo	Servo
Digitizing	Digitizing

Variable 98 - Master Axis Number

Enter at VAR 98

Variable 98 returns the axis number of the master axis. When the axis specified at the AXIS input is executing a RATIO_ move, this variable will return the master's axis number. If no move is active or the active move is not a RATIO_ move, this variable will return 0.

Applies to these types of axes:

READ	WRITE
Servo	---

Variable 99 - Actual Velocity

Enter at VAR 99

Actual Velocity is sampled every 256 msec and is returned in ladder units/minute. If the actual velocity exceeds the range -2147483648 lu/min to 2147483647 lu/min, the OK output will be reset and the RSLT output will return 0.

Applies to these types of axes:

READ	WRITE
Servo	---
Digitizing	

Variable 100 - Fast Input Response Time

Enter at VAR 100

Variable 100 specifies the observed response time of the fast input in µsec. When this value is non-zero, the control will compensate for this time delay by calculating and applying adjustments to the latched positions based on the current velocity of the axis. The valid range for this variable is 0 µsec to 32767 µsec.

Applies to these types of axes:

READ	WRITE
Servo	Servo
Digitizing	Digitizing

APPENDIX D - Stepper Reference Card

Stepper Reference Card

#	Profile Commands	Range
1	Distance Move	± 2,147,352,575 steps
2	Position Move	± 2,147,352,575 steps
3	Velocity Move	± 1,000,000 steps/sec
4	Set Maximum Velocity	1-1,000,000 steps/sec
5	Set acc/dec Rate	1-16,777,215 steps/sec/sec
6	Set Reference	± 2,147,352,575 steps
7	Pause	N/A

#	Control Words
1	Enable Profile
2	Pause Profile
3	Continue Profile
4	E-Stop
5	C-Stop
6	Step/direction mode (default)
7	CW/CCW mode

Word output from STEPSTAT function

Characteristic	Binary value	Decimal	Hex
Profile Enabled	00000000 0000000x(1)	1	0001
Profile Paused	00000000 000000x(1)0	2	0002
At Velocity	00000000 00000x(1)00	4	0004
Que Empty	00000000 0000x(1)000	8	0008
Que Full	00000000 000x(1)0000	16	0010
Control Word Not Processed	00000000 00x(1)00000	32	0020

Errors displayed in the .ERROR member of stepper structure	#
No Error	0
Invalid rack number or remote rack not available	1
Invalid slot number	2
Module not found at rack and slot location OR not enough channels on module	3
Invalid command number	4
Invalid data for the command	5
Invalid control number	6
Stepper function called before STEPINIT function called	7

APPENDIX E - Control Diagnostic LED Error Codes

Diagnostic LED Error Codes

The DIAG LED on the CPU module (status software with MMC for PC) will indicate an error condition using two different methods of turning the LED on and off. If the LED is continuously pulsing from bright to dim, it is indicating one of the following:

- The Servo Setup Function used in the ladder was compiled with a PiCPro version prior to 16.0. Recompile the Servo Setup Function, then compile and download the ladder with PiCPro 16.0 or greater.
- The application has accessed an array element beyond the defined array boundary. Avoid this practice.
- A UDFB is not preceded by the recommended Enable/OK lines. Precede UDFBs with Enable/OK lines.
- An internal software error has occurred. Consult the factory.

Otherwise, if there is an error, the LED will flash a three-digit code signal. For example, if there is a long pause-flash-pause-flash-flash-pause-flash-flash-flash-long pause, the code is 123. The table below lists the possible blink codes.

Code	Error	Description
122	**No math coprocessor**	Attempted to perform floating-point operation with no math coprocessor installed on the CPU.
123	**Scan too long**	A ladder scan loss has occurred because the CPU takes more than 200 ms to scan the application program. Whenever the scan light is out, the discrete outputs go to the OFF state and the analog outputs are zeroed.
124	**Excessive overhead**	The system overhead update time is excessive.
125	**Insufficient memory**	There is insufficient memory installed on the CPU to run the current program.
126	**No hardware bit memory**	There is no bit memory installed on the CPU and the program requires it.
127	**No software bit memory**	There is no bit memory capability via software and the program requires it.
222	**Driver error**	No driver support on the CPU for the I/O module. Update your system EPROMs.
22_	**Master rack error**	The I/O modules in the master rack do not match what was declared in the hardware master declaration table. The number of flashes in the third digit (_) identifies the slot number that is in error.
231*	**No daughter board**	There is no communications daughter board installed on the CPU when attempting to do expansion I/O communications.
232*	**Communications error**	A failure has occurred in remote I/O communications.
233*	**Number of racks error**	The number of expansion racks in the system does not match the number of expansion racks declared in the expansion hardware declarations table.
24_	**Master rack error**	The I/O modules in the MMC for PC rack do not match what was declared in the hardware master declaration table. The number of the third digit (_) identifies the slot number that is in error.

25_	**ASIU Error**	The ASIU modules found do not match what was declared in the hardware master declaration table. One or more ASIU modules have the same address switch setting. The number in the third digit indicates the ASIU address.
3_ _*	**Expansion rack error**	The I/O modules in the expansion rack(s) or the block I/O modules do not match what was declared in the expansion hardware declaration table. For rack expansion: The number of flashes in the third digit indicates the slot number. For block I/O modules: The number of flashes in the second and third digit indicates the block I/O module (01 through77). The second digit will flash a 1 - 7, 10 for 0. The third digit will flash a 1 - 9, 10 for 0. For example, if the second digit flashes 3 times and the third digit flashes 10 times, the module is 30.
260	**ASIU Data Overrun Error**	The specified number of axes at the specified update rate exceeds the throughput capacity of CPU to ASIU communications.
621	**Low Battery**	Applies to Digital MMC only. The battery for the control is low. Replace the battery with a new, fully charged battery.

* Errors connected with I/O expansion. Refer to the I/O Driver Module write-up in the PiC900 Hardware Manual for more information.

Drive LED Error Codes

When the drive is running, the DIAG LEDs will flash a two-digit code signal if there is an error. For example, if there is a long pause-flash-pause-flash-flash-long pause, the code is 12.

Code	Error	Description
01	**Drive Heatsink Temp. Warning**	Drive Heatsink Temp. exceeds Drive Heatsink Temp. Warning limit
02	**Drive Ambient Temp. Warning**	Drive Ambient Temp. exceeds Drive Ambient Temp. Warning limit
03	**Motor Temp. Warning**	Motor – Thermal Device is a thermistor and Motor Thermistor Temp. exceeds Motor Temp. Warning limit
04	**Motor Calculated Temp. Warning**	Motor Calculated Temp. exceeds the Motor Calculated Temp. Warning limit
11	**Drive Memory Fault**	Drive’s non-volatile memory is not functioning properly
12	**Drive Bus Over Voltage Fault**	Drive Bus Voltage exceeds Drive Bus Over Voltage Fault limit
13	**Drive PM1 Over Current Fault**	Current – Feedback exceeds Drive Over Current Fault limit

14	**Drive Over Power Fault**	Drive current and voltage output, in combination with the heatsink temp., indicate that the power output required by the drive would damage the power section
15	**Motor Temp. Fault**	Motor overheats
16	**Continuous Current Fault**	Current exceeds the continuous motor current rating for an extended period of time
17	**Drive Heatsink Temp. Fault**	Drive Heatsink Temp. exceeds Drive Heatsink Temp. Fault limit
22	**Drive F1 Feedback Fault**	Error is detected in the motor feedback
23	**Drive Ambient Temp. Fault**	Drive Ambient Temp. exceeds the Drive Ambient Temp Fault limit
24	**Motor Calculated Temp. Fault**	Motor Calculated Temp. exceeds the Motor Calculated Temp. Fault limit
25	**Drive Timing Fault**	Timing error is detected in the execution of the control algorithms performed by the drive’s digital signal processor
26	**Drive Interface Fault**	Communication error is detected in the transmission of information between the drive’s digital signal processor and the drive’s power section
27	**User Set Fault**	Select the PiCPro Set User Fault command or activate the Control Panel mode while the drive is enabled
31	**Drive F1 Communication Fault**	Communication error is detected in the transmission of information between the drive and a high resolution or multi-turn absolute feedback device
32	**Over Speed Fault**	Motor Velocity exceeds the Over Speed Fault limit
33	**Over Current Fault**	Current – Average exceeds the Over Current Fault limit
34	**Drive Communication Fault**	Communication error occurs while drive control is being performed using the PiCPro Control Panel tools
35	**Drive Power Module Fault**	Drive’s power section detects a fault condition
36	**Drive Setup Data Fault**	Drive software determines that the configuration data has been corrupted
41	**Drive Relay Fault**	Drive’s power section detects an error during power up
42	**Drive PM2 Over Current Fault**	Current – Feedback exceeds the Drive Over Current Fault limit
43	**Drive PM Over Temp. Fault**	Drive power module temp. exceeds the Drive Power Module Temp. Fault limit
44	**Motor Ground Fault**	Ground fault has occurred
45	**Drive AC Input Over Voltage Fault**	Incoming AC voltage is too high

APPENDIX F - IBM ASCII Chart

ASCII Chart

Dec	Hex	Char	Dec	Hex	Char	Dec	Hex	Char	Dec	Hex	Char
00	0x00	NUL	32	0x20	SPC	64	0x40	@	96	0x60	`
01	0x01	?	33	0x21	!	65	0x41	A	97	0x61	a
02	0x02	?	34	0x22	½	66	0x42	B	98	0x62	b
03	0x03	©	35	0x23	#	67	0x43	C	99	0x63	c
04	0x04	®	36	0x24	$	68	0x44	D	100	0x64	d
05	0x05	â	37	0x25	%	69	0x45	E	101	0x65	e
06	0x06	™	38	0x26	&	70	0x46	F	102	0x66	f
07	0x07	BEL	39	0x27	'	71	0x47	G	103	0x67	g
08	0x08	BS	40	0x28	(	72	0x48	H	104	0x68	h
09	0x09	HT	41	0x29	)	73	0x49	I	105	0x69	i
10	0x0A	LF	42	0x2A	*	74	0x4A	J	106	0x6A	j
11	0x0B	VT	43	0x2B	+	75	0x4B	K	107	0x6B	k
12	0x0C	FF	44	0x2C	,	76	0x4C	L	108	0x6C	l
13	0x0D	CR	45	0x2D	-	77	0x4D	M	109	0x6D	m
14	0x0E	?	46	0x2E	.	78	0x4E	N	110	0x6E	n
15	0x0F	¼	47	0x2F	/	79	0x4F	O	111	0x6F	o
16	0x10	Þ	48	0x30	0	80	0x50	P	112	0x70	p
17	0x11	ÿ	49	0x31	1	81	0x51	Q	113	0x71	q
18	0x12	þ	50	0x32	2	82	0x52	R	114	0x72	r
19	0x13	!	51	0x33	3	83	0x53	S	115	0x73	s
20	0x14	¶	52	0x34	4	84	0x54	T	116	0x74	t
21	0x15	§	53	0x35	5	85	0x55	U	117	0x75	u
22	0x16	-	54	0x36	6	86	0x56	V	118	0x76	v
23	0x17	þ	55	0x37	7	87	0x57	W	119	0x77	w
24	0x18	|	56	0x38	8	88	0x58	X	120	0x78	x
25	0x19	Ø	57	0x39	9	89	0x59	Y	121	0x79	y
26	0x1A	Æ	58	0x3A	:	90	0x5A	Z	122	0x7A	z
27	0x1B	¨	59	0x3B	;	91	0x5B	[	123	0x7B	{
28	0x1C	¿	60	0x3C	<	92	0x5C	\	124	0x7C	|
29	0x1D	⬬	61	0x3D	=	93	0x5D	]	125	0x7D	}
30	0x1E	s	62	0x3E	>	94	0x5E	^	126	0x7E	~
31	0x1F	t	63	0x3F	?	95	0x5F	=__	127	0x7F	›

Dec	Hex	Char	Dec	Hex	Char	Dec	Hex	Char	Dec	Hex	Char
128	0x80	Ç	160	0xA0	à	192	0xC0	¿	224	0xE0	‡
129	0x81	ü	161	0xA1	í	193	0xC1	¡	225	0xE1	·
130	0x82	é	162	0xA2	ó	194	0xC2	¬	226	0xE2	‚
131	0x83	â	163	0xA3	ú	195	0xC3	Đ	227	0xE3	„
132	0x84	ä	164	0xA4	ñ	196	0xC4	ƒ	228	0xE4	‰
133	0x85	à	165	0xA5	Ñ	197	0xC5	Ý	229	0xE5	Â
134	0x86	å	166	0xA6	ª	198	0xC6	ý	230	0xE6	Ê
135	0x87	ç	167	0xA7	°	199	0xC7	«	231	0xE7	Á
136	0x88	ê	168	0xA8	¿	200	0xC8	»	232	0xE8	Ë
137	0x89	ë	169	0xA9	©	201	0xC9	…	233	0xE9	È
138	0x8A	è	170	0xAA	™	202	0xCA	þ	234	0xEA	¾
139	0x8B	ï	171	0xAB	´	203	0xCB	À	235	0xEB	Î
140	0x8C	î	172	0xAC	¨	204	0xCC	Ã	236	0xEC	÷
141	0x8D	ì	173	0xAD	¡	205	0xCD	Õ	237	0xED	ý
142	0x8E	Ä	174	0xAE	«	206	0xCE	Œ	238	0xEE	Œ
143	0x8F	Å	175	0xAF	»	207	0xCF	œ	239	0xEF	«
144	0x90	È	176	0xB0	Õ	208	0xD0	–	240	0xF0	½
145	0x91	Æ	177	0xB1	±	209	0xD1	—	241	0xF1	±
146	0x92	Æ	178	0xB2	ð	210	0xD2	"	242	0xF2	Š
147	0x93	ô	179	0xB3	Š	211	0xD3	"	243	0xF3	ð
148	0x94	ö	180	0xB4	¥	212	0xD4	‘	244	0xF4	Ù
149	0x95	ó	181	0xB5	µ	213	0xD5	’	245	0xF5	ª
150	0x96	û	182	0xB6	¹	214	0xD6	÷	246	0xF6	³
151	0x97	ù	183	0xB7	²	215	0xD7	þ	247	0xF7	Ý
152	0x98	ÿ	184	0xB8	³	216	0xD8	ÿ	248	0xF8	°
153	0x99	Ö	185	0xB9	¼	217	0xD9	Ÿ	249	0xF9	•
154	0x9A	Ü	186	0xBA	½	218	0xDA	/	250	0xFA	Ž
155	0x9B	¢	187	0xBB	ª	219	0xDB	¤	251	0xFB	Đ
156	0x9C	£	188	0xBC	°	220	0xDC	‹	252	0xFC	‚
157	0x9D	¥	189	0xBD	¾	221	0xDD	›	253	0xFD	"
158	0x9E	û	190	0xBE	æ	222	0xDE	?	254	0xFE	ž
159	0x9F	ü	191	0xBF	ø	223	0xDF	?	255	0xFF	

APPENDIX G - Time Axes

Using a Time Axis

The time axis feature allows a servo axis to be slaved to time instead of a physical master position transducer. All the master/slave functions can be used with a time axis.

If time axes are going to be used they must be defined in Servo Setup. There are four axis numbers reserved for a time-based master; 25, 26, 27, and 28. This is the number used to identify the master time axis on the input to a master/slave function i.e. RATIO_GR.

The S_CURVE or ACC_DEC functions or the command velocity variable 6 can be used to move a time axis. The time axis can be manipulated with variables 1, 12, and 26. Use the WRITE_SV and READ_SV functions to work with these variables.

Referencing a Time Axis

Actual Position (Variable 1)

The actual position variable allows you to read the position of the time axis or change the current position by writing a value with the WRITE_SV function.

Range: +2,147,483,647 to -2,147,836,648 counts

Controlling Time Axis Velocity

You can use either the S-CURVE or ACC_DEC functions or the command velocity variable 6 to move a time axis.

S_CURVE and ACC_DEC Functions

When using the S_CURVE or ACC_DEC functions with a time axis, you can use the distance, position, or velocity moves to move the axis. The S_CURVE or ACC_DEC functions must be called first when using these moves. See the S_CURVE and ACC_DEC descriptions in the Function/Function Block Reference Guide.

Command Velocity (Variable 6)

If you are not using the S_CURVE or ACC_DEC functions, the command velocity variable can be used to define how fast the time axis will travel. It is programmed in counts per second. When the WRITE_SV function is called with variable 6, the time axis will step to the programmed velocity.

For example, if a value of 1000 were programmed as the number of counts per second for the velocity, then the time axis would move one count in one millisecond. If the master distance (MDST) was set at 1000 and the slave distance (SDST) was set at 2000 in the RATIO_GR function, it would take the slave axis one second to move 2000 units.

By entering a zero, the time axis is stopped. This provides the ability to synchronize multiple slave axes. You call all the moves you want to synchronize and then write a non-zero value to variable 6. All the axes will begin motion at the same time.

Note:

In order for all slave axes to start at the same time, the master start position of any master/slave move with a MSTR input would have to have the same value (or zero) at its MSTR input. If the option to ignore master start is selected, the slave axes will start when the master axis begins to move.

An alternative method for synchronizing slave starts is to use variable 26.

Range: +/-2,000,000 counts/sec

Rollover Position with a Time Axis

The rollover on position variable allows you to select where the time will reset to zero. The variable is entered in ladder units.

Note:

Without rollover on position, when 2,147,483,647 is reached, the next number will be -2,147,483,648. The counts continue to zero and back up to 2,147,483,647, etc.

Range: 1 to 536,870,911 counts (Entering a zero, turns rollover position off.)

Synchronizing Slave Axes with a Time Axis

The synchronized slave start variable allows you to tell the time axis which of its slave axes must be queued up before any of them begin their move. Each slave axis you want to synchronize is identified by setting a bit in a DINT using the lower 16 bits where the LSB = axis 1 and the MSB = axis 16. When the last set axis has been queued, all the slave axes will begin their move on the next interrupt.

The WRITE_SV function with variable 26 must be called before the move. It can be called again when you want to identify a different set of synchronized slave axes. Change the bits only after the slave axes identified in the first WRITE_SV function have started to move.

Writing a zero to variable 26 clears all identified axes. The READ_SV function can be used to read the number of slave axes being synchronized.

APPENDIX H - Stepper Axis Module Notes

Overview: Stepper Axis Module

Stepper motors can be controlled by either of the following methods:

- The stepper motor control module (SMCM) using the PiCPro stepper functions.
- The stepper axis module (SAM) using servo setup and the move types available in the PiCPro motion library. It can be a master or a slave in the application.

This appendix covers the stepper axis module. Any move type from the motion library can be used to perform motion control with the stepper except those move types requiring a fast input. There is no feedback from a stepper axis module.

Servo setup is used to set up the stepper axis module and create a start servo function. Once all the setup data has been entered, compile the servo function. This function will be stored in the servo library and can then be called in your ladder program to initialize the setup data for your application.

Notes on using Motion Library Functions and Variables with the Stepper

This section summarizes things you should be aware of when using the stepper and the motion library of PiCPro.

READ_SV/WRITE_SV Functions

These **READ_SV** and **WRITE_SV** variables cannot be used when using a stepper on an axis.

Var #	Name
4	Position error
9	Fast input position (Hardware)
10	Registration/referencing position change
11	Consecutive bad marks
19	Fast input direction
20	Fast input distance
24	Registration switch
27	Backlash compensation
28	TTL feedback
29	Reference switch position
46	Set user PID command
47	User PID command
48	Disable servo software

All other **READ_SV** and **WRITE_SV** variables can be used with a stepper.

Note: Feedback units are stepper units. Ladder units may still be used.

TUNEREAD/TUNEWRIT Functions

The filter variable is the only one that can be read and written by these functions when using a stepper axis. The remaining **TUNEREAD/TUNEWRIT** variables cannot be used by a stepper axis.

CLOSLOOP, OPENLOOP, REGIST, and **MEASURE** Functions

These functions cannot be used on a stepper axis.

Reference-Related Functions

The reference-related functions in PiCPro on the left below cannot be used with a stepper axis. The functions on the right can be used with a stepper axis.

Not Available with Stepper	Available with Stepper
FAST_REF	PART_REF
LAD_REF	PART_CLP
REF_DNE	
REF_END	

STRTSERV Function

Call **STRTSERV** to initialize the stepper axis and begin the stepper motion.

ERRORS

There is no loss of feedback or excess error. If an E-stop error occurs, the command to the stepper will be zeroed.

APPENDIX I - Toolbar Buttons

Overview: Toolbars

The following toolbars are available in PiCPro.

- Advanced Operations
- Analog Drive List
- Basic Online Operations
- Compiler
- Custom Motor Toolbar
- Digital Drive List
- Function/Function Blocks
- Insert New Network
- Ladder
- Servo and SERCOS
- Standard
- Structured Text Tools
- View Navigator

Advanced Operations Toolbar

 Toggle Animation

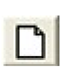
 Toggle Forcing

 Group Enable

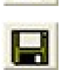
 Update Force Values

 Abort last patch

Analog Drive List Toolbar

 New Document

 Open Document

 Save Document

 Cut

 Copy

 Paste

 Undo

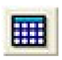
 Redo

 Print

 About

 Help

 Options

 Display Drive Data

 Rename Drive

 Drive Maintenance

 Download to Drive

 Reset Faults

 Reset Peaks

 Restore Factory Defaults

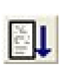
 Set User Fault

 Cold Restart

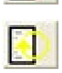
 Control Panel

Basic Online Operations Toolbar

 Stop the Scan

 Run One Scan

 Hot Restart

 Warm Restart

 Cold Restart

 Backup User Program

 Backup RAMDISK

 Restore User Program

 Reset Power

 Control Status

 Display Drive Maintenance Dialog

Compiler Toolbar

 Compile Bin File

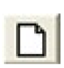
 Compile and Download Bin File

 Dump a Hex-86 File

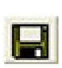
 Build a Task

 Build a UDFB

Custom Motor Toolbar

 New Document

 Open Document

 Save Document

 Cut

 Copy

 Paste

 Undo

 Redo

 Print

 About

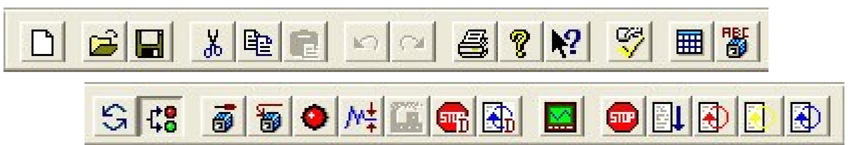
 Help

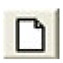
 Options

 Restore Factory Defaults

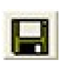
 Choose Default Motor Values

Digital Drive List Toolbar



 New Document

 Open Document

 Save Document

 Cut

 Copy

 Paste

 Undo

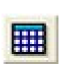
 Redo

 Print

 About

 Help

 Options

 Display Drive Data

 Edit name and address

 Refresh All Drive Data

 Show Status

 Drive Maintenance

 Download File Data

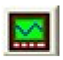
 Reset Faults

 Reset Peaks

 Restore Factory Defaults

 Set User Fault

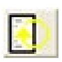
 Cold Restart

 Control Panel

 Stop Scanning

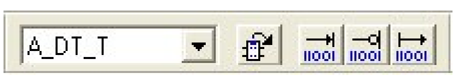
 Run One Scan

 Hot Restart

 Warm Restart

 Cold Restart

Function/Function Blocks Toolbar



Function/Function Block List

 Place the Selected Function

 Data In

 Data In Inverted

 Data Out

Insert New Network Toolbar

Insert Ladder Network

Insert Structured Text Network

Ladder Toolbar

Pointer

Normally Open Contact

Normally Closed Contact

Normally Open Positive Transition Contact

Normally Closed Positive Transition Contact

Normally Open Negative Transition Contact

Normally Closed Negative Transition Contact

Energized Coil

De-energized Coil

Set Coil

Reset Coil

Horizontal Wire

Vertical Wire

Combination Wire

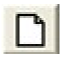
Point to Point Wire

Jump to Label

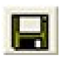
Jump to Subroutine

Return from Jump

Servo and SERCOS Toolbar

 New Document

 Open Document

 Save Document

 Cut

 Copy

 Paste

 Print

 About

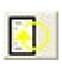
 Help

 Options

 Stop the Scan

 Run One Scan

 Hot Restart (Red Arrow)

 Warm Restart (Yellow Arrow)

 Cold Restart (Blue Arrow)

 Backup User Program

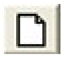
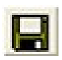
 Restore Use Program

 Reset Power

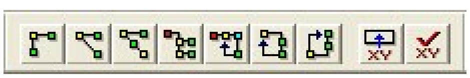
 Control Status

 Drive Maintenance

Standard Toolbar

-  New Document
-  Open Document
-  Save Document
-  Cut
-  Copy
-  Paste
-  Print
-  About
-  Help
-  Options
-  Lock./Unlock ldo

Structured Text Toolbar



-  IF-THEN
-  IF-THEN-ELSE
-  ELSEIF-THEN-ELSE
-  CASE
-  FOR-DO
-  WHILE-DO
-  REPEAT-UNTIL
-  Insert Variable
-  Check ST Syntax

View Navigator Toolbar

 Display Software Declarations

 Displays Hardware Declarations

 Displays View List

 Update Forcing List

 Toggles Long Name Display

 Zoom Display Out

 Zoom Display In

APPENDIX J - Module Filenames

Module Filenames

The following tables list the files that are created during the programming process.

Extensions	Type of File	Created/Updated Upon
ADV Analog Interfaced Drive System File	Drive parameters from an analog interfaced drive and associated motor parameters	Save from MMC Smart Drive Setup
BAK Library Backup	Backup copy of library file – the version created with the last save	Save
BIN Binary	Compiled (PiC language) ladder file – cannot be converted back into uncompiled state for animating, etc.	Compile or Compile and Download
DDV Digital Interfaced Drive System File	Drive parameters from a digital interfaced drive and associated motor parameters	Save from MMC Smart Drive Setup
DPL Dependency List	List of all files the currently loaded module depends upon	Build dependency list
FRC Forcing	List of variables that can be forced and their values	Save, if forced variables are designated
G&L G&L Motion Control	Compresses the project file created in Project Manager	New project
HEX Hexadecimal	Hex representation of the ladder file	Hex compile
HTM HTML	HTML Help file you create to define your UDFBs	Edit Function Block Help
LBK Ladder Backup	Backup copy of the ladder file – the version previous to the version created with the last save	Save
LDO Ladder Diagram Object	Ladder or network logic file	Save
LIB Library	User-defined functions/function blocks to be used in the ladder	Make function, Compile UDFB/Task
LST List	Print file containing ladder, REM and cross reference, formatted for printing	Print
MAP MAP	Produces a readable symbol map file	Compile when Generate Symbol Map is checked in Settings
OID Operator Interface Definition	Used to support the operator interface or Ethernet module	Compile when Construct Data File is checked in Settings
PPR PiCPro Restore	Used to restore program to the control	Save to File is chosen in the PiC Restore dialog

PRJ Project	Contains a project generated from Project Manager	New project
RBK Remarks Backup	Backup copy of the remarks file	Save
REM Remarks	The documentation or comments for all networks in a module	Save
RTD Real Time Display	List of variables in the View Variables List	Edit view list
SCP Scope Template	Contains the control panel and oscilloscope settings	Save Scope Template
SCT SERCOS View / Tune	Viewing and tuning data for the SERCOS axes used in application	Save
SRC SERCOS Setup	SERCOS setup data for SERCOS axes used in application	Save
SRV Servo Setup	Setup data for all servo axes used in application	Save
SVT Servo View / Tune	Viewing and tuning data for the servo axes used in application	Save
TXT Trace Text	Contains the oscilloscope trace	Save Trace
UCP Universal Communications Personality	Used by the configuration tool when downloading the personality file to the DeviceNet scanner module	Save from the Danaher Motion DeviceNet Configuration Tool
UCT Universal Communications Text	Used by PiCPro when downloading your ladder	Save from the Danaher Motion DeviceNet Configuration Tool

APPENDIX K - Updating PiCPro - Service Pack Installation

Service Pack Installation

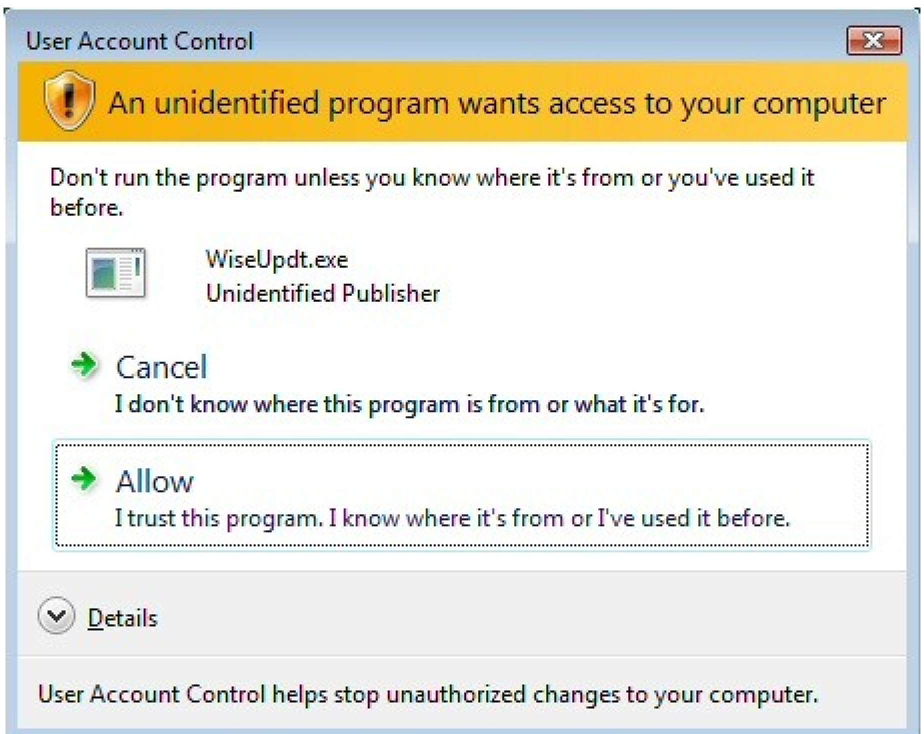
The Wise™ Update installation update program determines if any PiCPro Service Packs are available on the Internet, specific to the version of PiCPro you are currently running. If so, you are notified and then have the option to install the service pack. This program is automatically installed when PiCPro is installed.

Note:

Connection to the Internet is required to use this program.

This update program can be used on either on a manual basis or be set to work automatically, where an automatic check is initiated every so many days.

Under Windows Vista, you will be prompted to run the update program. Choose 'Allow' to enable the update to run. This prompt will also be displayed when the update is scheduled automatically.



Service Pack Update Installation - Manual Method

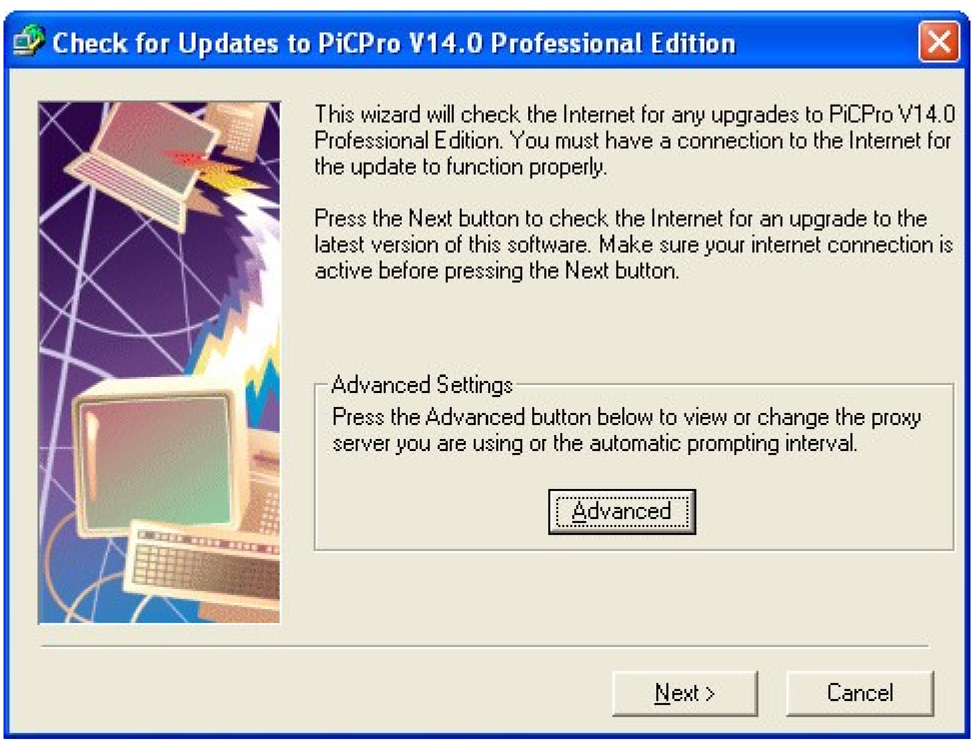
The update program can be started manually in one of three ways:

- From the Windows Start menu select **Start | Programs | PiCPro | Check for PiCPro Vxx Updates**. Where **xx** represents the version of software installed.
- From within PiCPro, using **Help | Check for Updates**. If the updated program is started from the PiCPro **Help** menu, it can be run at the same time as PiCPro
- After PiCPro installation is complete by using a check box as shown below.

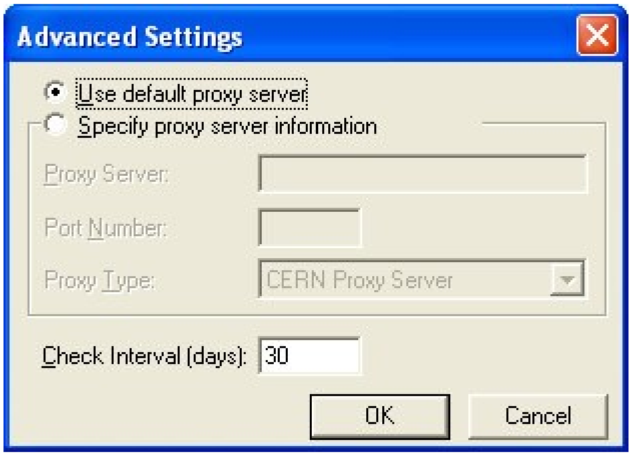
- Check the box for Yes, I want to check the Internet for updates.
- Click **Finish**.

The update program will now be started.

1. A window will be displayed as follows:

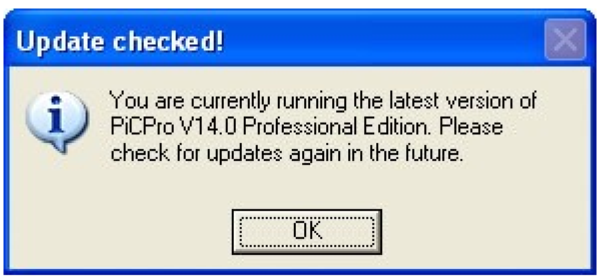


2. Clicking on the **Advance** button will bring up a dialog that allows you to change the proxy server information and the time period between update checks.



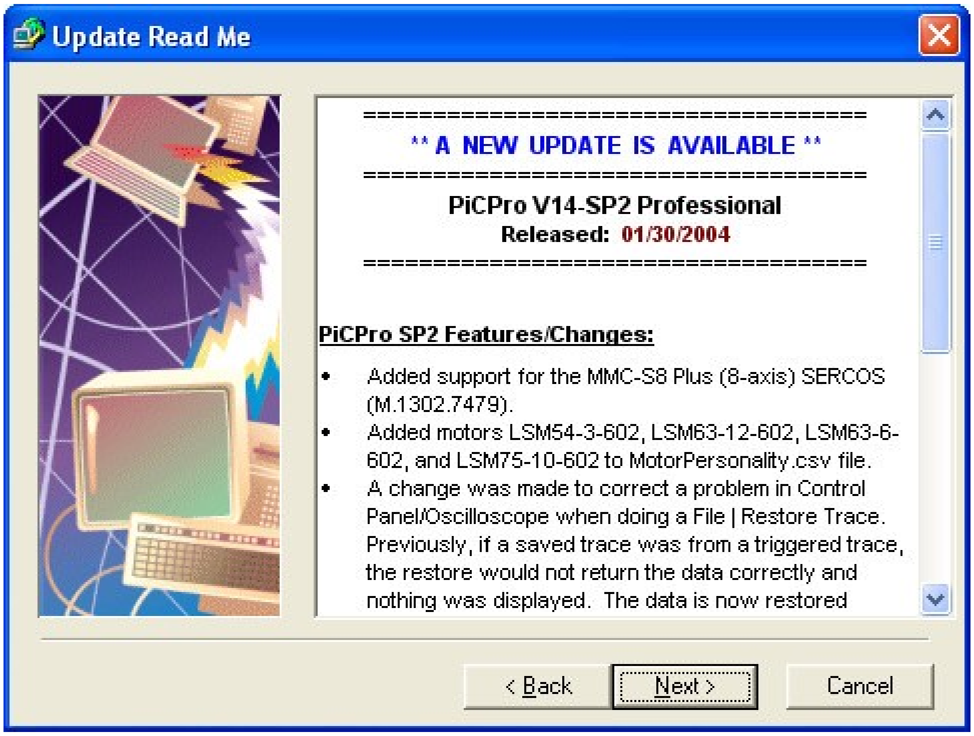
3. Make the desired changes in the **Advanced Settings** dialog and click **OK**.
4. Click **Next** and the update program will check for any PiCPro updates.

If you already have the latest PiCPro Service Pack, a message dialog will be displayed stating the following:

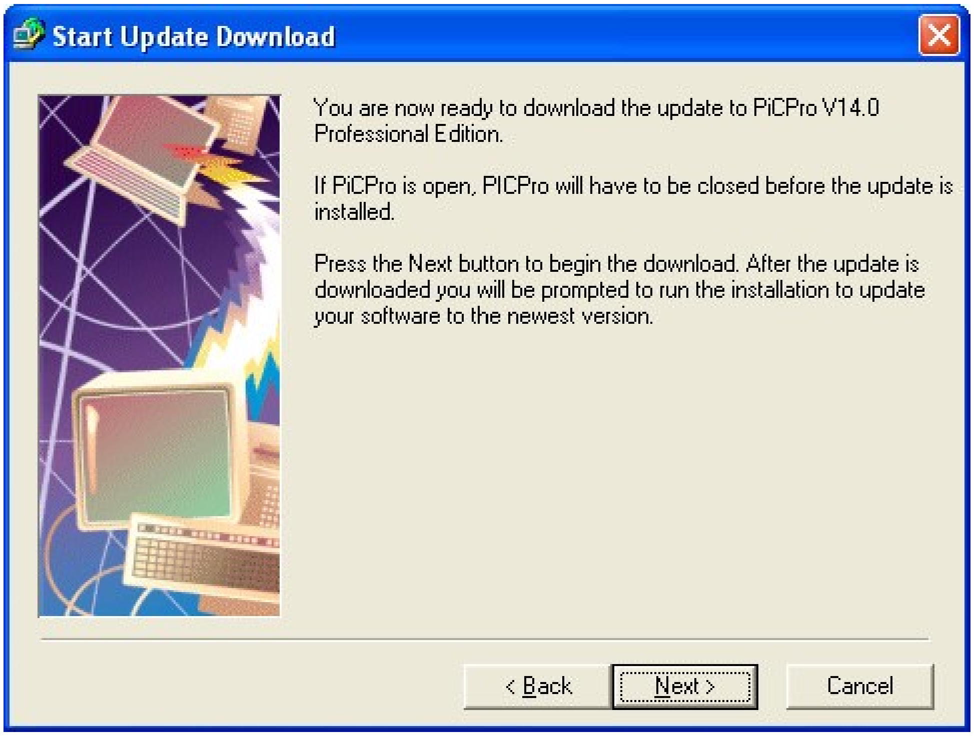


Pressing **OK** will exit the update program.

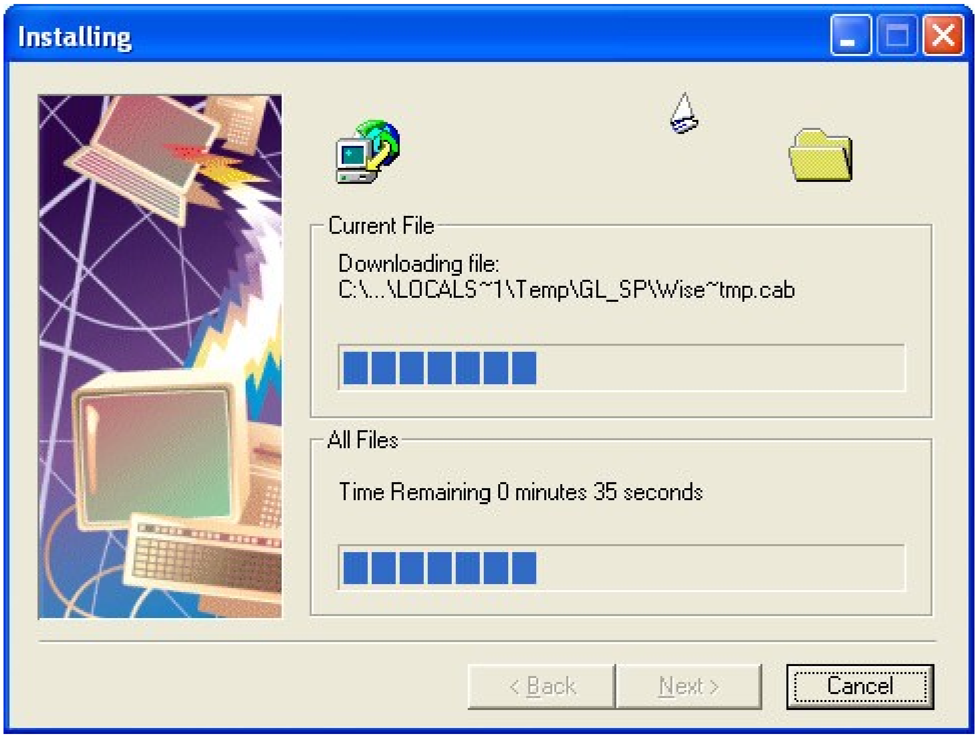
5. If a new PiCPro Service Pack exists, a dialog will be displayed stating the following:



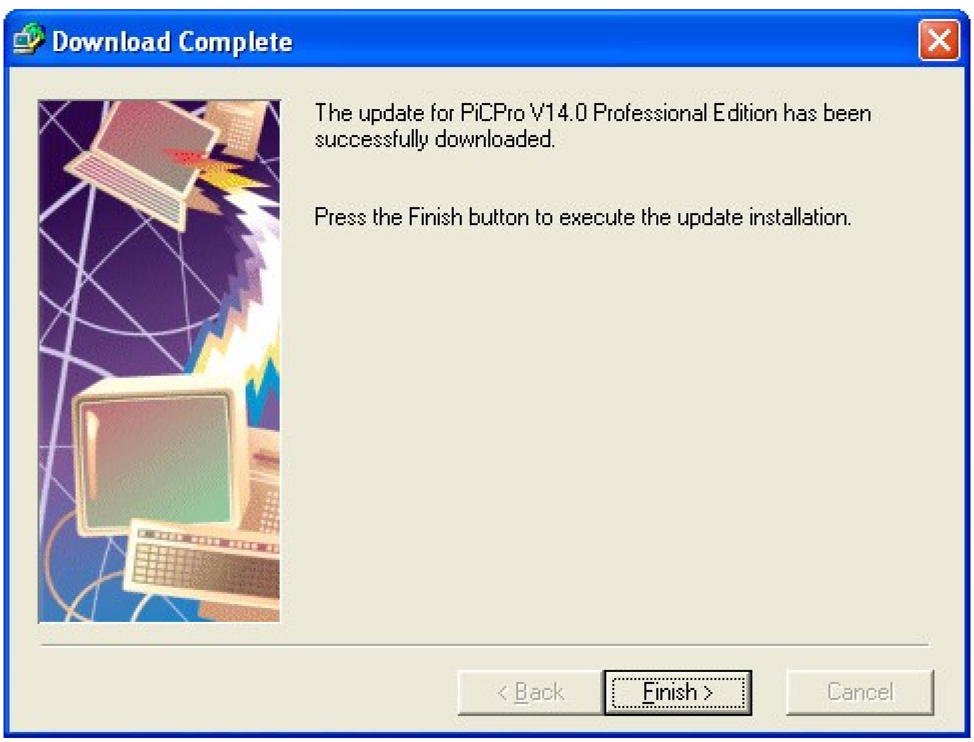
Click on the **Next button** to continue.



6. Click on the **Next button** to download the update
7. The update will be downloaded. The approximate time to download the update will be displayed in the dialog.



8. After the download is complete, a dialog will be displayed stating the following:



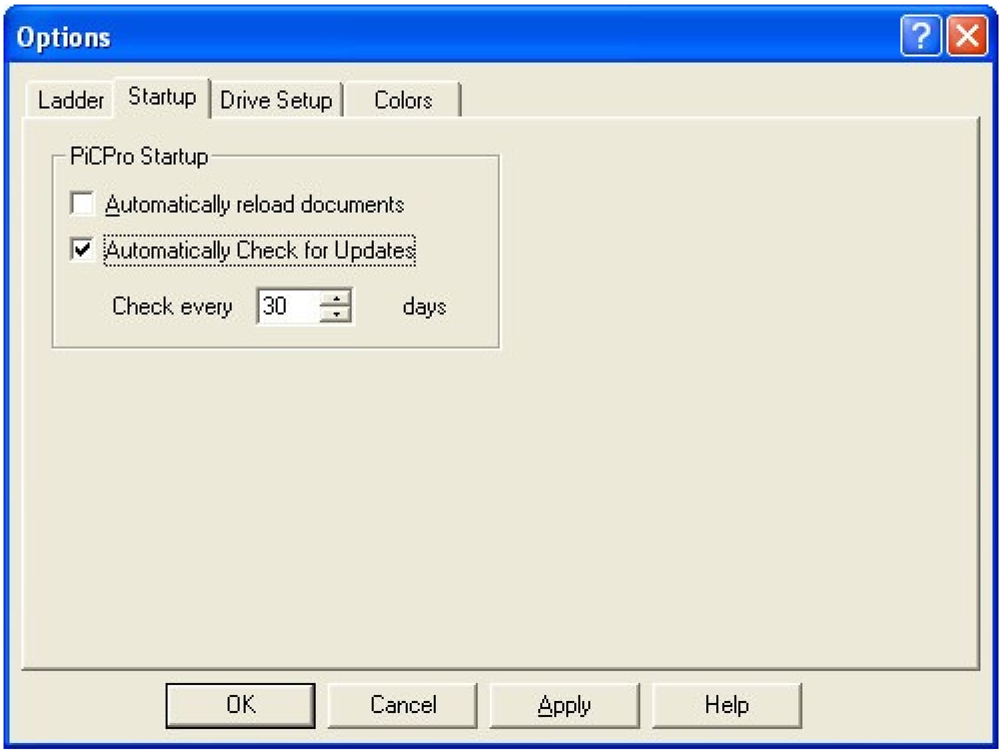
9. Click the **Finish** button to execute the update installation.

Service Pack Installation - Automatic Mode

When run in automatic mode, the update program is added to the start of PiCPro and is executed every time PiCPro is started.

The update program is set up for automatic mode by doing the following:

1. From the PiCPro menu, choose **View | Options**. Click on the **Startup** tab.



2. Check the box for Automatically Check for Updates.
3. Specify the time period between updated checks in the **Check every** ___ **days** field.
4. Click **OK**.
5. Follow steps 1 through 5 from the section titled Service Pack Update Installation - Manual Method.

APPENDIX L - Virtual Axis

Virtual Axis

Any servo axis can be defined as a virtual axis (both input and output types are virtual). This axis can be used for simulation or a time base for complex moves.

A virtual axis is a servo axis with no feedback hardware and no command output hardware. It can perform all the same functions as a normal servo axis: motion functions, master/slave functions, registration, referencing, etc. All WRITE_SV/READ_SV variables that are available for servo axes will be available for virtual servo axes.

The control will use the virtual axis' command position to generate a simulated feedback position. The feedback will be a 32-bit value. If rollover position = 0, the range of feedback will be –2,147,483,648 to 2,147,483,647.

APPENDIX M - Firmware

Firmware

To determine whether you need to update your control firmware file (*.hex) select **Online | Control Operations | Status…**, view **Control EPROM Version** and then compare it with the date shown for the firmware in the firmware folder for your control.

To determine whether you need to update your drive firmware file (*.hex), do the following:

1. Open the **Drive List View** and select your drive from the **Drive List**. See Open an Existing Analog Interfaced Drive System File or Open an Existing Digital Interfaced Drive System File.
2. Open the **Drive Data View** (Access Drive Data view from Drive List view or Access Drive Data view from digital Drive List View)) and view **Firmware Revision** and **Firmware Date/Time Stamp**.
3. Compare the date shown in **Firmware Date/Time Stamp** with the date shown for the firmware in the firmware folder for your drive.

Typically, the most current version of firmware should be used.

Notes:

- RMMmm is the version number of the firmware. It will appear in the Drive Data View window or the Online | Status window. MM represents a 2-digit major number and mm represents a 2 digit minor number.
- RMMmm Beta – Is used for Beta firmware *.hex files.
- Rev MM.mm is the version number that appears in the Drive Firmware window.
- dd/mm/yyyy is the Date that appears in the **Online | Controls | Status** window.

Firmware for the Analog MMC Smart Drive

Location in PiCPro:

C:\G&L Motion Control Data\PiCPro V*xx.x Edition Name*\Firmware\Analog MMC Smart Drive

Previous Name	Current Name	Description
DRB####.hex	MMC-SD-xxx Analog MMC Smart Drive RMMmm.hex	Analog MMC Smart Drive Firmware

Firmware for the Digital MMC Smart Drive and Digital MMC Controls

Location in PiCPro:

C:\G&L Motion Control Data\PiCPro Vxx.x Edition Name \Firmware\Digital MMC Smart Drive

Note: The location is the same for all Digital MMC Smart Drive *.hex files.

Previous Name	Current Name	Description
N/A	MMC-Dx Resident Control RMMmm.hex	Drive Resident Digital MMC Control Firmware
N/A	MMC-Dx Standalone Control RMMm.hex	Standalone Digital MMC Control Firmware

N/A	MMC-SD-xxx-D Digital MMC Smart Drive RMMmm.hex	Digital MMC Smart Drive Firmware
N/A	Clear Flash MMC-Dx Resident Control.hex	Clears Drive Resident Digital MMC control’s Flash Memory
N/A	Clear Application MMC-Dx Resident Control.hex	Clears Drive Resident Digital MMC control’s Application Memory
N/A	Clear Application MMC-Dx Standalone Control.hex	Clears Standalone Digital MMC control’s Application and Flash Memory

Firmware for the MMC for PC

Location in PiCPro:

C:\G&L Motion Control Data\PiCPro V*xx.x Edition Name* \Firmware\MMC for PC

Note: The location is the same for all MMC for PC *.hex files.

Previous Name	Current Name	Description
MMCPCS##.hex	MMC for PC-Sx SERCOS RMMmm.hex	MMC for PC with a SERCOS CPU
MMCPCA##.hex	MMC for PC Analog RMMmm.hex	MMC for PC with an Analog CPU
Clrflash.hex	Clear Flash – MMC for PC.hex	Clears MMC for PC Flash memory
Clrapp.hex	Clear Application MMC for PC.hex	Clears MMC for PC Applications memory
CON768.hex	768K Application Memory Configuration.hex	768K Application Memory
CON704.hex	704K Application Memory Configuration.hex	704K Application Memory
CON640.hex	640K Application Memory Configuration.hex	640K Application Memory
Con512.hex	512K Application Memory Configuration.hex	512K Application Memory
Con384.hex	384K Application Memory Configuration.hex	384K Application Memory

Firmware for the PiC900-PiC90 Controls

Location in PiCPro:

C:\G&L Motion Control Data\PiCPro V*xx.x Edition Name* \Firmware\PiC900 - PiC90

Note: The location is the same for all PiC900-PiC90 *.hex files.

Previous Name	Current Name	Description
P904x##.hex	PiC90 904x CPU RMMmm.hex	PiC90 with a 486 processor (9041, 9043)
P94X##.hex	PiC90 94x CPU RMMmm.hex	PiC900 with a 486 processor (941, 943, 945, 947)

Serco##.hex	SERCOS RMMmm.hex	SERCOS module firmware
TCPIP##.hex	Ethernet TCPIP RMMmm.hex	Ethernet module firmware
CON768.hex	768K Application Memory Configuration.hex	768K Application Memory
CON704.hex	704K Application Memory Configuration.hex	704K Application Memory
CON640.hex	640K Application Memory Configuration.hex	640K Application Memory
Con512.hex	512K Application Memory Configuration.hex	512K Application Memory
Con384.hex	384K Application Memory Configuration.hex	384K Application Memory

Firmware for the Standalone MMC Control

Location in PiCPro:

C:\G&L Motion Control Data\PiCPro Vxx.x Edition Name \Firmware\Standalone MMC

Note: The location is the same for all Standalone MMC *.hex files.

Previous Name	Current Name	Description
MMC##	MMC-Sx SERCOS MMC RMMmm.hex	Standalone MMC with a SERCOS CPU
MMC##	MMC-Ax Analog MMC RMMmm.hex	Standalone MMC with an Analog CPU
Serco##.hex	SERCOS RMMmm.hex	SERCOS module firmware
TCPIP##.hex	Ethernet TCPIP RMMmm.hex	Ethernet module firmware
Clrflash.hex	Clear Flash - MMC.hex	PiC90 with a 486 processor (9041, 9043)
Clrapp.hex	Clear Application - MMC.hex	PiC900 with a 486 processor (941, 943, 945, 947)

APPENDIX N - Converting an Application from One CPU Type to Another

Converting an Application from One CPU Type to Another

- **IPSTAT**
  The Ethernet functions for a PiC or standalone MMC do not require the IPSTAT function because those CPU models have an external Ethernet Module. An MMC for PC ladder with Ethernet functions will require the IPSTAT function for its connection to the host operating system. **Note:** IPSTAT will always return OK for a PiC or standalone MMC. OK will be returned whether the reset input is on or off.
- **A_INCHRD** or **A_IN_MMC**
  A PiC or MMC for PC ladder with an A/D input requires A_INCHRD.
  A standalone MMC with an A/D input can use A_IN_MMC or A_INCHRD.
- **I/O Points**
  All non Block I/O points should be checked. (Block I/O points are not affected by a change in CPU type.)
- **Servo and SERCOS Setup**
  Any servo or SERCOS setup functions used must be opened, converted to the new CPU type, and compiled.
- **Serial Port Functions**
  When converting a ladder to an MMC for PC, all serial port functions must be removed. (For example, OI_SER, etc.)

Conversion References

Standalone MMC CPU to PiC CPU

Below is a list of the I/O Point Labels for the general connector on the MMC module CPU and for the axis and auxiliary connectors on the analog module. In the following table, "#" represents the slot number.

The PiC I/O point column is what will appear in the software declarations table if you change your CPU from a Standalone MMC to a PiC or to an MMC for PC. Always check the I/O points after changing CPU types to be sure the I/O mappings match your system.

Discrete point	Declared Name	Software Declaration I/O assignment (MMC I/O Point)	PiC I/O Point
16 general DC Inputs	GENI1–GENI16	IGEN.1 – IGEN.16	I3.1 – I3.16
16 general DC Outputs	GENO1–GENO16	OGEN.1 – OGEN.16	O3.1 – O3.16
2 short circuit Inputs	SHORT1-SHORT2	ISGEN.1 – ISGEN.2	I3.17 – I3.18
6/12 auxiliary DC Inputs	AUXI#_1-AUXI#_12	IAUX#.1 – IAUX#.12	I4.1 – I4.12
Axis 1 DC input	AX1RDY#	I#A1.1 (Slot #, Axis 1, Input 1)	I4.13
Axis 2 DC input	AX2RDY#	I#A2.1	I4.14
Axis 3 DC input	AX3RDY#	I#A3.1	I4.15
Axis 4 DC input	AX4RDY#	I#A4.1	I4.16
Axis 1 DC output	AX1EN#	O#A1.1 (Slot #, Axis 1, Output 1)	O4.1
Axis 2 DC output	AX2EN#	O#A2.1	O4.3
Axis 3 DC output	AX3EN#	O#A3.1	O4.5

Axis 4 DC output	AX4EN#	O#A4.1	O4.7
Axis 1 DC output	AX1RES#	O#A1.2 (Slot #, Axis 1, Output 2)	O4.2
Axis 2 DC output	AX2RES#	O#A2.2	O4.4
Axis 3 DC output	AX3RES#	O#A3.2	O4.6
Axis 4 DC output	AX4RES#	O#A4.2	O4.8

The following I/O points can be **manually entered** to software declarations if desired:

Discrete MMC Point	Declared Name	Software Declaration I/O assignment (MMC I/O Point)	PiC I/O Point*
Axis 1 fast input	AX1FIN#	IFAUX#.1	I4.17
Axis 2 fast input	AX2FIN#	IFAUX#.2	I4.18
Axis 3 fast input	AX3FIN#	IFAUX#.3	I4.19
Axis 4 fast input	AX4FIN#	IFAUX#.4	I4.20
Axis 49 fast input	DIGFIN#	IFAUX#.49	I4.21

MMC for PC CPU to PiC CPU

The following table describes the MMC for PC, Analog Servo Interface Units (ASIU). The ASIU number is included in both the declared name and the I/O assignment. In the following table, "#" represents the ASIU number.

The PiC I/O point column is what will appear in the software declarations table if you change your CPU from an MMC for PC to a PiC or to a standalone MMC. (In column 4, the "s" indicates the slot number which is the sum of the ASIU # and 5. I.e. ASIU2 would show as I7.1, when converted.) Always check the I/O points after changing CPU types to be sure the I/O mappings match your system.

Discrete point	Declared Name	Software Declaration I/O assignment	PiC I/O Point
16 general DC Inputs	GENI#_1–GENI#_16	IGEN#.1 – IGEN#.16	Is.1 – Is.16
16 general DC Outputs	GENO#_1–GENO#_16	OGEN#.1 – OGEN#.16	Os.1 – Os.16
2 short circuit Inputs	SHORT#_1-SHORT#_2	ISGEN#.1 – ISGEN#.2	Is.17 – Is.18
6/12 auxiliary DC Inputs	AUXI#_1-AUXI#_12	IAUX#.1 – IAUX#.12	Is.19 – Is.30
Axis 1 DC input	AX1RDY#	I#A1.1 (ASIU #, Axis 1, Input 1)	Is.31
Axis 2 DC input	AX2RDY#	I#A2.1	Is.32
Axis 3 DC input	AX3RDY#	I#A3.1	Is.33
Axis 4 DC input	AX4RDY#	I#A4.1	Is.34
Axis 1 DC output	AX1EN#	O#A1.1 (ASIU #, Axis 1, Output 1)	Os.17
Axis 2 DC output	AX2EN#	O#A2.1	Os.19

Axis 3 DC output	AX3EN#	O#A3.1	Os.21
Axis 4 DC output	AX4EN#	O#A4.1	Os.23
Axis 1 DC output	AX1RES#	O#A1.2 (ASIU #, Axis 1, Output 2)	Os.18
Axis 2 DC output	AX2RES#	O#A2.2	Os.20
Axis 3 DC output	AX3RES#	O#A3.2	Os.22
Axis 4 DC output	AX4RES#	O#A4.2	Os.24

Declarations for the Fast Inputs available for each axis **must be manually entered** into software declarations. They cannot be copied to the clipboard from hardware declarations.

Axis 1 fast input	AX1FIN#	IFAUX#.1	Is.35
Axis 2 fast input	AX2FIN#	IFAUX#.2	Is.36
Axis 3 fast input	AX3FIN#	IFAUX#.3	Is.37
Axis 4 fast input	AX4FIN#	IFAUX#.4	Is.38
Axis 49 fast input	DIGFIN#	IFAUX#.49	Is.39

APPENDIX O - Function, Function Block and Library Overview

Overview: Arithmetic Library Groups

The arithmetic library contains the ARITH, DATETIME, and TRIG groups of functions.

ARITH Group

The ARITH group of functions perform the familiar operations of addition, subtraction, multiplication, division, modulo (remainder), square root, and negate (opposite) value.

C A U T I O N

If an underflow or overflow error occurs when one of these arithmetic functions executes, the output at OK will not energize. The value at OUT will be unpredictable.

The ARITH group includes the following:

Function	Description
ABS	Gives the absolute value of a number.
ADD	Adds from 2 to 17 numbers.
DIV	Performs the division operation and returns the quotient.
MOD	Performs the division operation and returns the remainder.
MUL	Multiplies from 2 to 17 numbers.
NEG	Returns the opposite value of a number.
SQRT	Determines the square root of a number.
SUB	Performs the subtraction operation on two numbers.

DATETIME Group

The DATETIME group of functions are used to add or subtract TIME duration and/or TIME_OF_DAY type variables or constants. The D#, T#, TOD#, and DT# characters are part of the result in the output variables with the exception of STRINGS.

When one of these functions executes, if an error occurs, the output at OK does not energize, and the value of the variable at OUT will be:

TIME duration:	T#0
TIME_OF_DAY:	TOD#0:0:0
DATE:	D#1988-01-01
DATE_AND_TIME:	DT#1988-01-01-00:00:00STRING: null (length 0)

For every output variable, its value cannot exceed the largest value allowed for the largest time increment, and it cannot be less than zero for the smallest time increment. Other values "roll over".

For example, if the largest increment is days, the output value must not exceed 49. If the smallest increment is seconds, the output value must not be less than 0 seconds. However, 24 hours becomes one day for a DATE_AND_TIME variable whose largest increment is years.

The DATETIME group includes the following:

Function	Description
A_DT_T	Adds DATE_AND_TIME to TIME and outputs a DATE_AND_TIME sum.
A_TOD_T	Adds TIME_OF_DAY to TIME and outputs a TIME_OF_DAY sum.
S_DT_DT	Subtracts a DATE_AND_TIME from a DATE_AND_TIME and outputs a TIME duration value.
S_DT_T	Subtracts TIME from a DATE_AND_TIME and outputs DATE_AND_TIME.
S_D_D	Subtracts a DATE from a DATE and outputs a TIME duration value.
S_TOD_T	Subtracts TIME from TIME_OF_DAY and outputs TIME_OF_DAY.
S_TOD_TO	Subtracts TIME_OF_DAY from TIME_OF_DAY and outputs a TIME duration value.

TRIG Group

The TRIG group of functions perform trigonometric or transcendental functions.

The TRIG group includes the following:

Function	Description
ACOS	Calculates the arc cosine.
ASIN	Calculates the arc sine.
ATAN	Calculates the arc tangent.
COS	Calculates the cosine.
EXP	Calculates the exponent.
LN	Calculates the natural log.
LOG	Calculates the log.
SIN	Calculates the sine.
TAN	Calculates the tangent.

Overview: Binary Library Functions

The binary library functions perform two types of operations:

1. Logical or boolean operations
2. Bit shifting and rotating operations

Logic Functions

The logic functions evaluate the input values on a bit by bit basis, and place the results for each bit into the corresponding bit of the output variable. In general, bit x of every input variable is evaluated and a result is put into bit x of the output variable.

Bit Shifting and Rotating Functions

The bit shifting and rotating functions "move" the values of bits. The values are shifted or rotated to the left or right.

Function	Description
AND	Performs the boolean AND operation on from 2 to 17 numbers.
NOT	Complements the bits of a number.
OR	Performs the boolean inclusive OR operation on from 2 to 17 numbers.
ROL	Rotates n bits from left to right (most significant to least significant positions).
ROR	Rotates n bits from right to left (least significant to most significant positions).
SHL	Shifts all bits of a number n positions to the left, discarding n bits of the left (most significant, and inserting n 0s on the right (least significant).
XOR	Performs the boolean exclusive OR operation on from 2 to 17 numbers.

Overview: Counters Library Function Blocks

The Counters library function blocks serve as counters.

Function Block	Description
CTD	Counts down from a specified value and then energizes an output.
CTU	Counts up to a specified value and then energizes an output.
CTUD	Counts up or down from a specified value and then energizes the appropriate output.

Overview: Datatype Library Groups

The Datatype library contains all the functions that convert one data type to another data type.

BOOL2BYT Group

The BOOL2BYT group converts a Boolean data type.

Function	Description
BOOL2BYT	Changes the data type from boolean to byte.

BYTECONV Group

The BYTECONV group converts byte data types.

Function	Description
BYT2BOOL	Changes the data type from byte to boolean.
BYTE2DW	Changes the data type from byte to double word.
BYTE2LW	Changes the data type from byte to long word.
BYTE2 SI	Changes the data type from byte to short integer.
BYTE2 USI	Changes the data type from byte to unsigned short integer.
BYTE2 WO	Changes the data type from byte to word.

DINTCONV Group

The DINTCONV group converts double integer data types.

Function	Description
DINT2DW	Changes the data type from double integer to double word.
DINT2INT	Changes the data type from double integer to integer.
DINT2LI	Changes the data type from double integer to long integer.
DINT2RE	Changes the data type from double integer to real.
DINT2SI	Changes the data type from double integer to short integer.
DINT2UDI	Changes the data type from double integer to unsigned double integer.

DWORDCONV Group

The DWORDCONV group converts double word data types.

Function	Description
DWOR2BYT	Changes the data type from double word to byte.
DWOR2DI	Changes the data type from double word to double integer.
DWOR2LW	Changes the data type from double word to long word.
DWOR2RE	Changes the data type from double word to real.
DWOR2UDI	Changes the data type from double word to unsigned double integer.
DWOR2WO	Changes the data type from double word to word.

D_TCONV Group

The D_TCONV group converts date and time data types.

Function	Description
DATE2STR	Changes the DATE value to a STRING value.
DT2DATE	Changes the DATE from a DATE_AND_TIME value.
DT2STR	Change the DATE_AND_TIME value to a STRING value.
DT2TOD	Outputs the TIME_OF_DAY from a DATE_AND_TIME value.
D_TOD2DT	Concatenates DATE and TIME_OF_DAY values and outputs a DATE_AND_TIME.
TIME2UDIN	Changes the data type from TIME to unsigned double integer.
TIME2STR	Changes a TIME duration value to a STRING value.
TOD2STR	Changes a TIME_OF_DAY value to a STRING value.

INTCONV Group

The INTCONV group converts integer data types.

Function	Description
INT2DINT	Changes the data type from integer to double integer.
INT2LINT	Changes the data type from integer to long integer.
INT2SINT	Changes the data type from integer to short integer.
INT2UINT	Changes the data type from integer to unsigned integer.
INT2WORD	Changes the data type from integer word.

LINTCONV Group

The LINTCONV group converts long integer data types.

Function	Description
LINT2DI	Changes the data type from long integer to double integer.
LINT2INT	Changes the data type from long integer to integer.
LINT2LR	Changes the data type from long integer to long real.
LINT2LW	Changes the data type from long integer to long word.
LINT2SI	Changes the data type from long integer to short integer.
LINT2ULI	Changes the data type from long integer to unsigned long integer.

LREALCONV Group

The LREALCONV group converts long real data types.

Function	Description
LREA2LI	Changes the data type from long real to long integer.
LREA2LW	Changes the data type from long real to long word.
LREA2RE	Changes the data type from long real to real.
LREA2ULI	Changes the data type from long real to unsigned long integer.

LWORDCNV Group

The LWORDCNV group converts long word data types.

Function	Description
LWOR2BYT	Changes the data type from long word to byte.
LWOR2DW	Changes the data type from long word to double word.
LWOR2LI	Changes the data type from long word to long integer.
LWOR2LR	Changes the data type from long word to long real.
LWOR2ULI	Changes the data type from long word to unsigned long integer.
LWOR2WO	Changes the data type from long word to word.

NUM2STR Group

The NUM2STR group converts numeric data type.

Function	Description
NUM2STR	Changes the data type from numeric to string.

REALCONV Group

The REALCONV group converts real data types.

Function	Description
REAL2DI	Changes the data type from real to double integer.
REAL2DW	Changes the data type from real to double word.
REAL2LR	Changes the data type from real to long real.
REAL2UDI	Changes the data type from real to unsigned double integer.

SINTCONV Group

The SINTCONV group converts short integer data types.

Function	Description
SINT2BYT	Changes the data type from short integer to byte.
SINT2DI	Changes the data type from short integer to double integer.
SINT2INT	Changes the data type from short integer to integer.
SINT2LI	Changes the data type from short integer to long integer.
SINT2USI	Changes the data type from short integer to unsigned short integer.

SIZEOF Group

The SIZEOF group contains one function.

Function	Description
SIZEOF	Reports the size in bytes of the variable name listed at the IN input.

STRCONV Group

The STRCONV group converts string data types.

Function	Description
STR2D_T	Changes the data type from string to date and time.
STR2NUM	Changes the data type from string to numeric.
STR2USI	Changes the data type from string to unsigned short integer (ASCII code).

UDINTCNV Group

The UDINTCNV group converts unsigned double integer data types.

Function	Description
UDIN2DI	Changes the data type from unsigned double integer to double integer.
UDIN2DW	Changes the data type from unsigned double integer to double word.
UDIN2RE	Changes the data type from unsigned double integer to real.
UDIN2TIM	Changes the data type from unsigned double integer to time.
UDIN2UI	Changes the data type from unsigned double integer to unsigned long integer.
UDIN2ULI	Changes the data type from unsigned double integer to unsigned long integer.
UDIN2USI	Changes the data type from unsigned double integer to short integer.

UINTCONV Group

The UINTCONV group converts unsigned integer data types.

Function	Description
UINT2INT	Changes the data type from unsigned integer to integer.
UINT2UDI	Changes the data type from unsigned integer to unsigned double integer.
UINT2ULI	Changes the data type from unsigned integer to unsigned long integer.
UINT2USI	Changes the data type from unsigned integer to unsigned short integer.
UINT2WO	Changes the data type from unsigned integer to word.

ULINTCNV Group

The ULINTCNV group converts unsigned long integer data types.

Function	Description
ULINT2LI	Changes the data type from unsigned long integer to long integer.
ULINT2LR	Changes the data type from unsigned long integer to long real.
ULINT2LW	Changes the data type from unsigned long integer to long word.
ULINT2UDI	Changes the data type from unsigned long integer to unsigned double integer.
ULINT2UI	Changes the data type from unsigned long integer to unsigned integer.
ULINT2USI	Changes the data type from unsigned long integer to unsigned short integer.

USINTCNV Group

The ULINTCNV group converts unsigned long integer data types.

Function	Description
USINT2BYT	Changes the data type from unsigned short integer to byte.
USINT2SI	Changes the data type from unsigned short integer to short integer.
USINT2STR	Changes the data type from unsigned short integer (ASCII code) to the first character in string.
USINT2UDI	Changes the data type from unsigned short integer to unsigned double integer.
USINT2UI	Changes the data type from unsigned short integer to unsigned integer.
USINT2ULI	Changes the data type from unsigned short integer to unsigned long integer.

WORDCONV Group

The WORDCONV group converts word data types.

Function	Description
WORD2BYT	Changes the data type from word to byte.
WORD2DW	Changes the data type from word to double word.
WORD2INT	Changes the data type from word to integer.
WORD2LW	Changes the data type from word to long word.
WORD2UI	Changes the data type from word to unsigned integer.

Overview: Evaluate Library Functions

The Evaluate library functions compare numbers. The comparisons are:

- equal to (=)
- not equal to (≠)
- greater than (>)
- less than (<)
- greater than or equal to (≥)
- less than or equal to (≤)

Function	Description
EQ	Compares from 2 to 17 numbers and energizes an output if all numbers are equal to each other.
GE	Compares from 2 to 17 numbers and energizes an output if all numbers are greater than or equal to successive numbers.
GT	Compares from 2 to 17 numbers and energizes an output if all numbers are greater than successive numbers.
LE	Compares from 2 to 17 numbers and energizes an output if all numbers are less than or equal to successive numbers.

LT	Compares from 2 to 17 numbers and energizes an output if all numbers are less than successive numbers.
NE	Compares from 2 to 17 numbers and energizes an output if they are not equal to each other.

Note on String Evaluations:

Strings are compared character by character based on the ASCII value of the characters. Therefore, String 1 in the example would be greater than String 2 because 9 is greater than 3.

Example: If String 1 = 1 2 9 and String 2 = 1 2 3 4, than String 1 is greater than String 2.

If two strings have different lengths and the characters in the shorter string match the characters in the longer string, then the shorter string is less than the longer one.

Example: If String 1 = 1 2 3 and String 2 = 1 2 3 4, than String 1 is less than String 2.

With ASCII values in a string, the value of upper case letters is less than the value of lower case letters.

Example: If String 1 = TIME and String 2 = Time, than String 1 is less than String 2.

Overview: Fbinter Library Function Blocks

The function/function blocks in the Fbinter library allow you to interface with fieldbus communications via the appropriate fieldbus hardware module.

Function	Description
FB_CLS	Closes communications with the field bus.
FB_OPN	Opens communications with the field bus placing the DeviceNet module in the RUN mode.
FB_RCV	Receives all data from the configurator file indicated by Tag names.
FB_SND	Sends data indicated by Tag names in the configurator file.
FB_STA	Allows you to check if the DeviceNet module is communicating with the nodes and to check field bus information.

Overview: Filter Library Functions

The Filter library functions act as filters or sorters. They move the value of one of the inputs into an output variable.

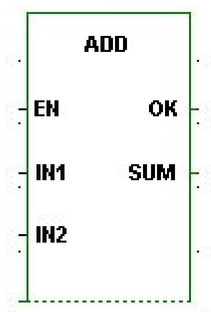
Function	Description
LIMIT	Evaluates a number and outputs the number if it is within specified limits, or outputs the upper or lower limit if the number is greater than or less than the limit, respectively.
MAX	Compares from 2 to 17 numbers and outputs the largest number.
MIN	Compares from 2 to 17 numbers and outputs the smallest number.
MOVE	Places from 1 to 17 numbers into output variables of the same type(s).

MUX	Evaluates from 2 to 17 numbers and outputs one of the numbers based on the value of an independent number.
SEL	Evaluates two numbers and outputs one of them based on the state of a boolean input.

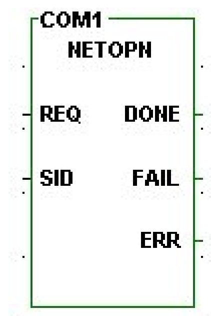
Overview: Functions and Function Blocks

Functions and function blocks are similar to what is called a subroutine in other programming languages. They perform operations on data. They perform arithmetic operations, evaluate bits, read and write data, move axes, etc. You specify input values and provide variables into which the PiC inserts the output values. The execution of functions and function blocks is always triggered by Boolean logic.

Function



Function Block



Functions and function blocks are similar to each other and look similar on your screen. The one important difference between them is that function blocks may take more than one scan to complete an operation, whereas functions must complete an operation synchronously with the scan. This requires that function blocks have internal storage for their variables from scan to scan until they complete the operation. You must declare function blocks and assign each one a unique name in software declarations. Then the PiC can reserve memory for them.

To use functions/function blocks in your application, you place them in the network of your LDO file and make the appropriate connections.

Note:

The function ADD belongs to a group of functions that are "extensible" indicated by the dashed line on the bottom. Extensible functions can be "extended" to accommodate up to 17 inputs. Place the focus on the lower left-hand corner of the function for inputs and choose Data In to add an input.

Overview: I/O Library Groups

The I/0 library functions initialize and send/receive data from:

- Analog input module
- Analog and 4-20mA output modules
- Controls, ports, files, devices, serial communications module
- J-K thermocouple module
- PID loops
- Encoder module (background read)

- RTD module
- Stepper module

The I/O functions have an ERR output that will hold the number of the error that occurred when execution of the function fails.

ANLGIN group

The ANLGIN group contains functions that work with the analog input module.

Function	Description
A_INCHIT	Initializes a channel on an analog input module.
A_INCHRD	Reads or samples the voltage or current occurring at a channel on an analog input module.
A_INMDIT	Initializes an analog input module.
A_IN_MMC	Outputs the digital value of an analog input for the MMC.

ANLGOUT group

The ANLGOUT group contains functions that work with the analog or 4-20mA output module.

Function	Description
ANLGINIT	Initializes an analog or 4-20mA module.
ANLG_OUT	Sends a value (to be converted to voltage or current) to a channel on an analog or 4-20mA output module.

BAT_OK? Group

The BAT_OK? group contains one function that allows you to check the battery of the control from the ladder.

Function	Description
BAT_OK?	Checks the battery from the ladder.

BIO_PERF group

The BIO_PERF group contains two function/function blocks: one that allows you to check the performance of the block I/O modules in your system and one that initializes the configuration of the block system.

Function	Description
BIO_PERF	Checks the performance of the block I/O modules.
IO_CFG	Initializes the block I/O configuration, checks the status, and inhibits the block system when clocks are added or removed.

COMM group

The COMM group contains function blocks that are used to transfer (read/write) data between the devices on the left and the data types on the right.

Device	Data Type
User Port on the PiC PiC RAMDISK Files PiC FLASHDISK Files Workstation Files Serial Communications Module	Strings, Arrays, Structures

Function	Description
ASSIGN	Sets up the channels on the serial communications module to work like the User Port for communications.
CLOSE	Closes the communication channel between the LDO and workstation file, RAMDISK file, FLASHDISK file, User Port, or a serial communications channel.
CONFIG	Establishes protocol between the LDO and User Port or a serial communications channel. Must execute after OPEN and before READ, WRITE, or STATUS.
DELFIL	Deletes files from the PiC RAMDISK or PiCPro.
DIRECT	Reads PiC RAMDISK or FLASHDISK directory information.
FRESPACE	Checks the amount of available disk space there is on the PiC RAMDISK or FLASHDISK.
OPEN	Opens the communication channel between the LDO and a DOS file, RAMDISK file, FLASHDISK file, User Port, or a serial communications channel. Must execute before CONFIGURE, READ, WRITE, STATUS, or SEEK.
READ	Reads data from a workstation, RAMDISK, or FLASHDISK file, User Port, or a serial communications channel and places it into a string, array, structure, array element, or structure member.
RENAME	Renames a file on the PiC RAMDISK or PiCPro.
SEEK	Positions a pointer in a RAMDISK or FLASHDISK file before a read/write is performed.
STATUS	Outputs the number of bytes in the input buffer of User Port or a serial communications channel.
WRITE	Writes data from a memory area to a workstation file, RAMDISK file, User Port, or a serial communications channel.

JKTEMP group

The JKTEMP group contains functions that work with the J-K thermocouple module.

Function	Description
ATMPCHIT	Initializes a channel on a J-K thermocouple module.
ATMPCHRD	Reads or senses the temperature occurring at a channel on a J-K thermocouple module.
ATMPMDIT	Initializes a J-K thermocouple module.

NETWORK group

The NETWORK function blocks are used to perform communication operations among networked PiCs.

Function	Description
NETCLS	Closes the communication channel between the PiC in which it is executed and all other networked PiCs.
NETFRE	Used after data from a transaction has been received (NETRCV) to clear the input buffer.
NETMON	Monitors network activity for diagnostic purposes.
NETOPN	Opens the communication channel between the PiC in which it is executed and all other networked PiCs.
NETRCV	Receives or reads data that was sent by another PiC.
NETSND	Sends data to another PiC or all PiCs in the network.
NETSTA	Reports how many bytes are in the input buffer to be received by one or more NETRCVs.

PID group

The PID group contains one function that performs PID control.

Function	Description
PID	Performs proportional, integral, and derivative control.

READFDBK group

The READFDBK group contains one function that reads an encoder or 12-channel resolver module on a scan time basis (background).

Function	Description
READFDBK	Performs background read on encoder module, 12-channel resolver module, or block resolver or encoder modules.

RTDTEMP group

The RTDTEMP group contains functions that work with the RTD module.

Function	Description
ARTDCHIT	Initializes a channel on a RTD module.
ARTDCHRD	Reads or senses the temperature occurring at a channel on a RTD module.
ARTDMDIT	Initializes a RTD module.

SOCKETS group

The SOCKETS group contains function blocks that are used to communicate from application to application using Danaher Motion's implementation of the BSD socket interface.

Function	Description
IPACCEPT	Used by the TCP server to accept incoming connect requests.
IPCLOSE	Used by an application to terminate a communication session for the socket specified at HNDL.
IPCONN	Used by a client application to connect to a remote server by specifying the remote endpoint address for a socket.
IPHOSTID	Optional and not required to be used.
IPIP2NAM	Allows the application to obtain the host name when you supply the IP address.
IPLISTEN	Used to make a socket passive.
IPNAM2IP	Allows the application to obtain an IP address when you supply the host name.
IPREAD	Allows you to read input data sent between a client function and a remote server.
IPRECV	Used to get a packet of data sent between a client function and a remote server.
IPSEND	Used to send data between client function and remote servers.
IPSOCK	Used to obtain a data structure and assign it to a specific communication resource.
IPSTAT	Used to obtain the status of the resources provided by the Windows™ operating system.
IPWRITE	Used to send data between client function and remote servers.

Note:

There are a maximum number of sockets that can be open at one time. This number is based on the control type as shown in the table below:

Control	Max number of open sockets
PiC900	64
MMC	64
MMC for PC	64
MMC Digital	40

STEPPER group

The STEPPER group contains functions that work with the stepper module.

Function	Description
STEPCNTL	Sends a control word to the stepper motion control module (SCMS).
STEPINIT	Initializes an axis as a stepper axis.
STEPSTAT	Reads the data on the status of the stepper axis.
STEP_CMD	Sends a profile command and its related data to the command queue of the SMCM to run a step profile.
STEP_POS	Reads the position of a stepper axis.

Overview: Motion Library Groups

The Motion library contains functions that allow you to perform motion control tasks. In addition to the standard motion function, there are two servo functions that can be created by you with Servo setup and SERCOS setup.

I M P O R T A N T

For parameters in these functions such as feedrates, accelerations, decelerations, position, distance, etc., you must enter ladder units (LU). You defined ladder units for your application in the scaling data section of setup.

When you have ladder units equal to feedback units (FU) in setup, then you are entering feedback units in the ladder.

Often a range of values in FU is listed with function inputs. If ladder units are not equal to feedback units, be sure to convert LU to FU to check that you are in range.

DATA group

The DATA group allows you to read, write, or check the status of certain variables and characteristics within motion control.

Function	Description
CAPTINIT	Initializes what data is to be captured each servo interrupt and where it is to be stored.
CAPTSTAT	Provides the ability to start and stop the capturing of data from the ladder.
COORD2RL	Calculates profile segments used for circular/linear interpolation. Used with the RATIO_RL function.
FU2LU	Converts feedback units to ladder units.
LU2FU	Converts ladder units to feedback units.
READ_SV	Allows you to read the variables in your ladder.
READ_SVF	Allows you to read any of the READ_SV variable at a faster rate. All values that involve velocity or distance are in feedback units and updates rather than ladder units and minutes.

SCA_CTRL	Writes control bits to the MDT for a servo axis
SCA_RCYC	Reads cyclic data from the AT for a servo axis
SCA_RECV	Allows you to receive information from the service channel section of SERCOS communication for a servo axis.
SCA_SEND	Allows you to send information to the service channel section of SERCOS communication for a servo axis.
STATUSSV	Allows you to check the status of the following characteristics from the word output of the STATUSSV function. Move started Fast input occurred Fast input on Good mark detected Bad mark detected DIST + TOLR exceeded Fast input rising
TUNEREAD	Provides the ability to read tuning parameters from the ladder.
TUNEWRIT	Provides the ability to write tuning parameters from the ladder.
WRITE_SV	Allows you to write variables from your ladder.
WRIT_SVF	Allow you to write any of the WRITE_SV variables at a faster rate. All values that involve velocity or distance are in feedback units and updates rather than ladder units and minutes.

ERRORS group

The ERRORS group contains controlled stop (C-stop), emergency stop (E-stop), programming, and timing errors that can occur within motion control. C-stop, E-stop, and programming errors can all be viewed in tuning when running the Servo Setup program.

C-Stop Errors

When a C-stop occurs, the following events occur:

- The axis remains in servo lock and the axis is brought to a controlled stop at the rate specified by the controlled stop ramp in setup.
- The active and next queues are cleared.
- The FAST_QUE mode is canceled when the C-stop is reset.

Function	Description
C_STOP	Set a controlled stop on the axis.
C_ERRORS	Indicates what C-errors have occurred at the word output
C_RESET	Resets a C-stop error.
C_STOP?	Asks if there is a C-stop in effect for the designated axis.

E-Stop Errors

When an E-stop occurs, the following events occur:

- The system is out of servo lock.
- Zero voltage is sent to the analog outputs.
- The active and next queues are cleared.
- The FAST_QUE mode is canceled when the E-stop is reset.
- If it is a loss of feedback E-stop error, then the machine must be redone.

In most respects, you are in a condition immediately following initialization with the exception of the queue number. The queue number does not start over but continues from where it left off when the E-stop occurred.

Remember that the queue number is assigned by the software from 1 to 255. When 255 is reached, it rolls over to 1.

Function	Description
E_STOP	Set an emergency stop on the axis.
E_ERRORS	Indicates what E-errors have occurred at the word output
E_RESET	Resets a E-stop error.
E_STOP?	Asks if there is a E-stop in effect for the designated axis.
SCA_RST	Resets internal E-errors for a SERCOS system.

Programming Errors

These errors occur during master/slave moves or a FAST_QUE call. They may prevent the move from being placed in the queue (or if the move is in the queue, abort the move) or they may prevent the OK from being set.

Function	Description
P_ERRORS	Indicates what programming errors have occurred at the word output
P_RESET	Resets a programming error.

Timing Error

The timing error is connected to the entire system. It is monitored by the TME_ERR? function. All the servo calculations for one interrupt must be completed

All the servo calculations for one interrupt must be completed in the time frame selected by you in setup before the next interrupt can perform its calculations. If they are not, this timing error occurs and the ERR output of the TME_ERR? function is set.

I M P O R T A N T
Always set an E-stop when a timing error occurs.

Function	Description
TME_ERR?	Asks if the time required to carry out the servo calculations exceeds the allotted interrupt time.

INIT group

The INIT group allows you to initialize the servos and be ready for motion commands from the ladder.

Function	Description
CLOSLOOP	Closes the position loop for the designated axis.
CLSLOOP?	Asks if the position loop for the designated axis is closed.
OPENLOOP	Opens the position loop for the designated axis.
SCA_CLOS	Closes the position loop in a SERCOS system.
STRTSERV	Used with the user-defined setup function to initialize setup data.

MOVE group

The MOVE group cause motion to begin or end. The moves are not master/slave moves.

The other functions that can cause motion are found in the RATIOMOVE and REF groups. They are master/slave moves and the fast input (FAST_REF) and ladder (LAD_REF) reference functions used to perform a machine reference.

Function	Description
POSITION	Moves an axis at a specified feedrate to an endpoint.
DISTANCE	Moves an axis a specified distance at a specified feedrate.
VEL_STRT	Moves an axis at a specified feedrate and direction.
VEL_END	Ends a velocity start move.

MOVE_SUP group

The MOVE_SUP group functions allow you to make adjustments to the moves.

Function	Description
ACC_DEC	Allows you to change the acc/dec rates entered in setup from the ladder.
CAM_OUT	Allows you to turn on discrete I/O points for a specified distance during the rollover on position cycle.
HOLD	Stops the iteration of the current move.
HOLD_END	Resumes the move that was halted with the HOLD function.
IN_POS?	Asks if the active move is in position
MEASURE	Enables the fast input response when not using registration or referencing.
NEWRATIO	Allows you to change the ratio of a RATIO_GR or RATIO_SYN move or the default ratio of the RATIOSLP move.
NEW_RATE	Allows you to change the feedrate of the moves in the queue.

RATIOSCL	Allows you to scale the slave and/or master axis in RATIOCAM, RATIOSLP, and the master axis in RATIO_RL moves.
REGIST	Sets the axis position to a defined value when a fast input occurs.
R_PERCEN	Allows you to change the feedrate by a percentage for all moves connected to an axis.
SCA_PBIT	Initializes the SERCOS fast input.
SCURVE	Allows a master time axis to follow an s-curve velocity profile minimizing the amount of jerk that can occur in a trapezoidal velocity profile.

QUE group

There are two queues used by the servo software to manage moves for an axis. One is the active queue, which holds the move that is currently active. The other is the next queue, which is the move that is ready and waiting to proceed when the active queue move is completed. The QUE group functions affect the moves in the queues.

The servo software assigns a queue number to any motion function that has a QUE output. The numbers are assigned sequentially from 1 to 255. When 255 is reached, the number rolls over to 1.

Function	Description
ABRTMOVE	Aborts the move identified by the number entered in its QUE input.
ABRTALL	Aborts the moves in both queues.
FAST_QUE	Manages the queues based on the occurrence of a fast input.
Q_NUMBER	Gives the number of the move that is in the active queue.
Q_AVAIL?	Asks if a queue is available for the specified.

RATIOMOV group

The RATIOMOV group functions cause motion to begin or end. They involve master/slave ratio moves.

Function	Description
GR_END	Ends a RATIO_GR nor RATIOSYN move.
RATIOCAM	A master/slave move where each segment of the profile has a constant ratio.
RATIOSLP	A master/slave move where the ratio in each segment of the profile can vary linearly.
RATIOSYN	A master/slave move where the slave axis will follow the master axis at a constant ratio and a positional relationship between the master and slave axes is established.
RATIO_GR	A master/slave move where the slave axis will follow the master axis at a constant ratio.
RATIO_RL	A master/slave move where the slave axis will follow the master axis in a profile that can be a trigonometric function or a polynomial using floating point variables.
REP_END	Ends a repeating master/slave profile move.
SYN_END	Ends a RATIOSYN move by specifying a drop point for the slave axis.

REF group

The REF group functions allow you to do machine or part referencing. A machine reference provides position information to the PiC with respect to the machine. It is a fixed dimensional reference used to establish a repeatable point of reference between servo initializations. The PiC bases its position calculations on this position information. Motion may occur when performing a machine reference.

A part reference is a floating dimensional reference. It establishes a position based on the location of a part, not the machine. No motion occurs when performing a part reference. The axis has been moved into position before the reference occurs.

Function	Description
FAST_REF	Performs a machine reference based on a fast input.
LAD_REF	Performs a machine reference from the ladder.
PART_CLR	Cancels the part reference dimension supplied by the PART_REF function.
PART_REF	Performs a part reference on the designated axis.
REF_DNE?	Asks if the machine reference cycle is complete.
REF_END	Ends the ladder machine reference.
SCA_ACKR	Acknowledges the reference cycle for a servo SERCOS axis.
SCA_REF	Runs a reference cycle on a SERCOS axis.
SCA_RFIT	Initializes the fast input on a SERCOS drive and monitors the reference switch or index mark position.

SERC_SLV group

The SERC_SLV group functions allow you to work with the SERCOS slave function/function blocks.

Function	Description
SCS_ACKR	Acknowledges the SERCOS reference cycle.
SCS_CTRL	Controls the bits in the MDT control word.
SCS_RECV	Receives information from the service channel section of the SERCOS communication.
SCS_REF	Runs a reference cycle on the SERCOS slave axis.
SCS_SEND	Sends information to the service channel section of the SERCOS communication.
SCS_STAT	Monitors the ready-to-operate drive mode, diagnostic trouble shooting, or two real-time data bits returned from the drive.

SERC_SYS group

The SERC_SYS group functions allow you to work with SERCOS rings and to start the SERCOS system.

Function	Description
SCR_CONT	Allows you to continue through SERCOS phases if you have halted after phase 2 to send additional IDNs.
SCR_ERR	Identifies ring errors that can occur during the transfer of IDNs.
SCR_PHAS	Identifies the current SERCOS phase.
SC_INIT	Copies the initialization data into all interface boards.

Overview: String Library Functions

The String library contains functions that operate on variables that have a STRING data type. Most of these functions return a STRING as an output. The variable assigned to receive this output STRING must be specified as an input variable on the left side of the function. Assigning the variable on the right side is optional, but, if used, it must be the same variable as the input variable. This characteristic is unique to all functions that have a STRING as an output, including functions not in this group.

The output at OK will not energize and the output STRING will be null (have length zero) if an error occurs.

Function	Description
CONCAT	Concatenates two STRINGs.
DELETE	Deletes characters from a STRING.
FIND	Searches for a STRING within another STRING and, if found, outputs its location.
INSERT	Inserts a STRING into another STRING.
LEFT	Place a specified number of characters from the left side of a STRING into a variable.
LEN	Returns the length of a STRING.
LWR_CASE	Converts all the characters in a string to lower case characters.
MID	Places a specified number of characters from the middle of a STRING into a variable.
REPLACE	Places a STRING within another STRING, replacing one or more characters.
RIGHT	Places a specified number of characters from the right side of a STRING into a variable.
UPR_CASE	Converts all the characters in a string to upper case characters.

Overview: Timers Library Function Blocks

The Timer library contains function blocks that are used to energize and de-energize outputs (contacts/coils) after a duration of time. The time, as it elapses, can be viewed on the monitor with real time animation. The elapsed time value can be used elsewhere in the module but its value cannot be reset.

Function Block	Description
TOF	De-energizes an output after a duration of time.
TON	Energizes an output after a duration of time.
TP	Energizes an output for a duration of time.

Overview: Xclock Library Functions

The Xclock library functions are used for clock or calendar functions.

Function Block	Description
CLOCK	Outputs from the PiC the current time and date, or sets the time and date on the PiC.
GETDAY	Outputs the number of the day of the week or day of the year.
SERVOCLK	Allows a task to run on the servo clock when no servos are running.

APPENDIX P - Analog Drive Parameters

220V Shunt on 440V Drive (Analog)

Category:	Limits & Indicators
Method of Selection:	Read-Only
Units:	n/a
Defaults:	n/a
Range:	n/a

Description:

Monitor the bus voltage while using a 220V motor with a 440V drive. The indicator is ON when the bus voltage exceeds 415V while using a 220V motor. The indicator will then go OFF when the bus voltage goes below 400V.

AccDec Limits Active (Analog)

Category:	Command
Method of Selection:	Dropdown List
Units:	n/a
Defaults:	Not Active
Range:	Not Active, Active

Description:

Select Active to apply rate-of-change limits to the Velocity Command. Set to Not Active when used with an MMC for closed loop servo applications.

Accel Limit (Analog)

Category:	Command
Method of Selection:	Edit
Units:	RPM / Sec
Defaults:	20000
Range:	16 to 1048560

Description:

Enter the limit of the rate-of-change of the Velocity Command when it is increasing and the AccDec Limits are Active.

Active Operating Mode (Analog)

Category:	Basic/System/Command/Tuning
Method of Selection:	Read-Only
Units:	n/a
Defaults:	Velocity
Range:	Velocity, Torque, Autotune, Alignment

Description:

Monitor the drive control mode. In Velocity mode, the ±10 volt Analog Command is converted to the Velocity Command. In Torque mode, the ±10 volt Analog Command is converted to the Current Command.

Alignment Mode (Analog)

Category:	Motor
Method of Selection:	Read-Only
Units:	n/a
Defaults:	n/a
Range:	n/a

Description:

Defines the Startup Commutation.

Analog Command (Analog)

Category:	Basic/System/Command
Method of Selection:	Read-Only
Units:	Millivolts (mV)
Defaults:	n/a
Range:	-10000 to 1,000

Description:

Monitor the voltage on the command input of the drive I/O connector. In Velocity mode, the ±10 volt Analog Command is converted to the Velocity Command. In Torque mode, the ±10 volt Analog Command is converted to the Current Command.

At Minus Current Limit (Analog)

Category:	Tuning
Method of Selection:	Read-Only
Units:	n/a
Defaults:	n/a
Range:	n/a

Description:

Monitor the state of the At Minus Current Limit indicator. ON when the Current Command equals the Minus Current Limit. The At Minus Current Limit indicator can be assigned to a drive output.

At Plus Current Limit (Analog)

Category:	Tuning
Method of Selection:	Read-Only
Units:	n/a
Defaults:	n/a
Range:	n/a

Description:

Monitor the state of the At Plus Current Limit indicator. ON when the Current Command equals the Plus Current Limit. The At Plus Current Limit indicator can be assigned to a drive output.

At Zero Speed (Analog)

Category:	Limits & Indicator
Method of Selection:	Read-Only
Units:	n/a
Defaults:	n/a
Range:	n/a

Description:

Monitor the At Zero Speed indicator. The indicator is ON when the absolute value of the Motor Velocity is less than or equal to the Zero Speed Rate. The indicator can be assigned to a drive output.

Brake Apply Time (Analog)

Category:	Motor
Method of Selection:	Edit
Units:	msec
Defaults:	0
Range:	-32768 to 32767

Description:

Enter the time value used to define the Release Brake and Drive Enabled controls when Drive Enable changes from ON to OFF while Fault is OFF. Release Brake ON allows motion. Release Brake OFF holds the load. Brakes internal to servomotors are meant to prevent a stationary load from moving, not to decelerate the load.

Drive Enable	Fault	Brake Apply Time	Release Brake	Drive Enabled
on to off	off	= 0	off immediately	off immediately
on to off	off	> 0	off after Brake Apply Time	off immediately
on to off	off	< 0	off immediately	off after absolute value of Brake Apply Time
-	on	-	off immediately	off immediately

Enter a negative value for the Brake Apply Time to allow the brake to take hold before power to the motor is removed.

Brake Release Time (Analog)

Category:	Motor
Method of Selection:	Edit
Units:	msec
Defaults:	0
Range:	-32768 to 32767

Description:

Enter the time value used to define the Release Brake and Drive Enabled controls when Drive Enable changes from OFF to ON. Release Brake ON allows motion. Release Brake OFF holds the load. Brakes internal to servomotors are meant to prevent a stationary load from moving, not to decelerate the load.

Drive Enable	Drive Ready	Fault	Brake Release Time	Release Brake	Drive Enabled
off to on	on	off	= 0	on immediately	on immediately
off to on	on	off	> 0	on after Brake Release Time	on immediately
off to on	on	off	< 0	on immediately	on after absolute value of Brake Release Time

Enter a positive value for the Brake Release Time to allow the drive to take control of the motor before the brake is released.

Continuous Current (Analog)

Category: Motor

Method of Selection: Read-Only

Units: Amps

Defaults: n/a

Range: n/a

Description:

The current the motor can accept continuously without overheating.

Cold Restart Required (Analog)

Category: Basic/System

Method of Selection: Read-Only

Units: n/a

Defaults: n/a

Range: n/a

Description:

Monitor the need to Cold Restart the drive after changing a major drive configuration parameter such as the Motor Model. An OFF-to-ON transition of Drive Cold Restart initiates the equivalent of a drive control power-on sequence. The Drive Cold Restart function can be assigned to a drive input. Select the PiCPro Drive Cold Restart command to initiate a Cold Restart in the ONLINE drive.

Custom Motor (Analog)

Category:	Basic/System/Motor
Method of Selection:	Read-Only
Units:	n/a
Defaults:	n/a
Range:	Custom Motor

Description:

Monitor the type of motor after changing motor model. Custom motor data can be modified by editing the custom motor file found in the custom motor directory. Standard motor data cannot be modified.

Current - Average (Analog)

Category:	Basic/System/Tuning
Method of Selection:	Read-Only
Units:	Amps
Defaults:	n/a
Range:	n/a

Description:

Monitor the drive average actual current output.

Current - Feedback (Analog)

Category:	Tuning
Method of Selection:	Read-Only
Units:	Amps
Defaults:	n/a
Range:	n/a

Description:

Monitor the output current as measured by the drive current feedback sensing circuit.

Current - Minus Enable (Analog)

Category:	Command
Method of Selection:	Read-Only
Units:	n/a
Defaults:	n/a
Range:	n/a

Description:

Monitor the status of the Current – Minus Enable function. ON enables the drive to respond to a negative Velocity or Current Command. OFF disables the drive from responding to a negative Velocity or Current command. Current – Minus Enable can be assigned to a drive input. If not assigned to an input, it is always ON.

Current - Peak Minus (Analog)

Category:	Tuning
Method of Selection:	Read-Only
Units:	Amps
Defaults:	n/a
Range:	n/a

Description:

Monitor the maximum negative current output. Use the PiCPro Reset Peaks command to reset this value to zero.

Current - Peak Plus (Analog)

Category:	Tuning
Method of Selection:	Read-Only
Units:	Amps
Defaults:	n/a
Range:	n/a

Description:

Monitor the maximum positive current output. Use the PiCPro Reset Peaks command to reset this value to zero.

Current - Plus Enable (Analog)

Category:	Command
Method of Selection:	Read-Only
Units:	n/a
Defaults:	n/a
Range:	n/a

Description:

Monitor the status of the Current – Plus Enable function. ON enables the drive to respond to a positive Velocity or Current Command. OFF disables the drive from responding to a positive Velocity or Current command. Current – Plus Enable can be assigned to a drive input. If not assigned to an input, it is always ON.

Current Command (Analog)

Category:	System/Command/Tuning
Method of Selection:	Read-Only
Units:	Amps
Defaults:	n/a
Range:	-Current Limit to +Current Limit

Description:

Monitor the command to the current loop. When the Active Operating Mode is Velocity, this is the output of the velocity loop. When the Active Operating Mode is Torque, this is the converted and scaled Analog Command.

Decel Limit (Analog)

Category:	Command
Method of Selection:	Edit
Units:	RPM / Sec
Defaults:	20000
Range:	16 to 1048560

Description:

Enter the limit of the rate-of-change of the Velocity Command when it is decreasing and the AccDec Limits are Active.

Default Mode (Analog)

Category:	Basic/System
Method of Selection:	Dropdown List
Units:	n/a
Defaults:	Velocity
Range:	Velocity, Torque

Description:

Select the mode that will be the Active Operating Mode when Override Mode Select is OFF. When using an MMC control, select Velocity.

Drive Ambient Temp. Fault Limit (Analog)

Category:	Limits & Indicators
Method of Selection:	Read-Only
Units:	Degrees C
Defaults:	n/a
Range:	n/a

Description:

If the Drive Ambient Temperature exceeds this limit, the drive will fault. This protects from erratic operation of the drive due to excessive heat. Typically, this occurs if the drive enclosure temperature rises due to excessive power dissipation. This fault only applies to MMC-SD-x-460 systems.

Drive Ambient Temp. Warning Limit (Analog)

Category:	Limits & Indicators
Method of Selection:	Read-Only
Units:	Degrees C
Defaults:	n/a
Range:	n/a

Description:

If the Drive Ambient Temperature exceeds this limit, the warning indicator will be set. The warning indicator can be assigned to a drive output to provide an early warning before a fault limit is exceeded. This warning only applies to MMC-SD-x-460 systems.

Drive Bus Charged (Analog)

Category:	System
Method of Selection:	Read-Only
Units:	n/a
Defaults:	n/a
Range:	n/a

Description:

Monitor the status of Drive Bus Charged indicator. Drive Bus Charged is ON when the Drive Bus Voltage level is sufficient for motor control. Drive Bus Charged can be assigned to a drive output.

Drive Bus Over Voltage Fault Limit (Analog)

Category:	Limits & Indicators
Method of Selection:	Read-Only
Units:	Volts
Defaults:	n/a
Range:	n/a

Description:

If the Drive Bus Voltage exceeds this limit, the drive will fault. This protects the drive’s power section from voltage levels that would damage it. If a motor is rated for 220/230 volts and is used with a 440/460 volt drive, this limit will protect the motor from voltage levels that would damage it. Typical causes of this fault include line voltage that is too high or back EMF generated by the motor during deceleration requiring an external shunt system to dissipate the energy.

Drive Bus Under Voltage Fault Limit (Analog)

Category:	Limits & Indicators
Method of Selection:	Edit
Units:	Volts
Defaults:	n/a
Range:	n/a

Description:

If the Drive Bus Voltage falls below this limit, the drive will fault. This protects the application from the drive’s inability to control the motor when the bus voltage is inadequate. A typical cause of this fault is an incoming line voltage that is too low.

Drive Bus Voltage (Analog)

Category:	System
Method of Selection:	Read-Only
Units:	Volts
Defaults:	n/a
Range:	0 to 1000

Description:

Monitor the voltage of the Drive Bus.

Drive Cold Restart (Analog)

Category:	System
Method of Selection:	Read-Only
Units:	n/a
Defaults:	n/a
Range:	n/a

Description:

Monitor the status of the Drive Cold Restart function. An OFF-to-ON transition of the Drive Cold Restart initiates the equivalent of a drive control power-on sequence. The Drive Cold Restart function can be assigned to a drive input. Select the PiCPro Drive Cold Restart command to initiate a Cold Restart in an ONLINE drive.

Drive Enable (Analog)

Category:	System
Method of Selection:	Read-Only
Units:	n/a
Defaults:	n/a
Range:	n/a

Description:

Monitor the status of Drive Enable. This is typically the state of the drive input that the Drive Enable function is assigned to. During normal operation while Drive Enable is ON and Fault is OFF, the drive's power section will be ON as indicated by Drive Enabled.

Drive Heatsink Temp. (Analog)

Category:	System
Method of Selection:	Read-Only
Units:	Degrees C
Defaults:	n/a
Range:	0 to 180

Description:

Monitor the temperature of the Drive Heatsink.

Drive Heatsink Temp. Fault Limit (Analog)

Category:	Limits & Indicators
Method of Selection:	Read-Only
Units:	Degrees C
Defaults:	n/a
Range:	n/a

Description:

If the Drive Heatsink Temperature exceeds this limit, the drive will fault. This protects the drive's power section from damage due to overheating. This occurs when the heatsink temperature rises due to excessive power output, excessive ambient temperature or inadequate airflow.

Drive Heatsink Temp. Warning Limit (Analog)

Category:	Limits & Indicators
Method of Selection:	Read-Only
Units:	Degrees C
Defaults:	n/a
Range:	n/a

Description:

If the Drive Heatsink Temperature exceeds this limit, the warning indicator will be set. The warning indicator can be assigned to a drive output to provide an early warning before a fault limit is exceeded.

Drive Inputs (Analog)

Category:	Basic/System
Method of Selection:	Read-Only
Units:	n/a
Defaults:	n/a
Range:	n/a

Description:

Monitor the state of the drive inputs. An input number will be displayed when the input is ON. For example, if inputs 1, 4 and 8 are ON , "1..4…8" will be displayed.

Drive Model (Analog)

Category:	Basic/System
Method of Selection:	Dropdown List
Units:	n/a
Defaults:	n/a
Range:	Extensive, see dropdown list for available choices

Description:

Displays the drive model. The drive model is selected when the drive is being inserted into the drive list. To change, in the File column, select the Drive Model parameter and pick the desired model from the dropdown list. The Drive Model parameter value in the actual column is read from the drive and must match the selected drive model. When the drive model is selected, all drive specific parameters are set to factory default values.

Drive Outputs (Analog)

Category:	Basic/System
Method of Selection:	Read-Only
Units:	n/a
Defaults:	n/a
Range:	n/a

Description:

Monitor the state of the drive outputs. An output number will be displayed when the output is ON. For example, if outputs 1, 3 and 5 are ON, "1.3.5" will be displayed.

Drive Over Current Fault Limit (Analog)

Category:	Limits & Indicators
Method of Selection:	Read-Only
Units:	Amps
Defaults:	n/a
Range:	n/a

Description:

If the Current – Feedback exceeds this limit, the drive will fault. This protects the drive's power section from damage due to excessive current output.

Drive Status (Analog)

Category:	Basic/System
Method of Selection:	Read-Only
Units:	n/a
Defaults:	n/a
Range:	n/a

Description:

Monitor the Drive Status Message indicating the state of the Drive Ready, Bus Charged, Drive Enabled, Fault and Warning indicators.

English/Metric Units (Analog)

Category:	System
Method of Selection:	Dropdown List
Units:	n/a
Defaults:	Metric
Range:	Metric, English

Description:

Select the display units for Motor – Kt, Inertia and Max Torque.

External Brake Release (Analog)

Category:	Motor
Method of Selection:	Read-Only
Units:	n/a
Defaults:	n/a
Range:	n/a

Description:

Monitor the status of the External Brake Release function. When the External Brake Release function is assigned to a drive input, the Release Brake output will follow the state of the External Brake Release input and the Brake Apply Time and the Brake Release time will be ignored. This allows an external control to coordinate operation of the brake.

External Brake Release Control (Analog)

Category:	Motor
Method of Selection:	Read-Only
Units:	n/a
Defaults:	n/a
Range:	n/a

Description:

ON when the External Brake Release is assigned to an input indicating that the Release Brake output will follow the External Brake Release input. When the External Brake Release function is assigned to an input, the Brake Apply Time and the Brake Release Time are ignored.

Fault Reset (Analog)

Category:	System
Method of Selection:	Read-Only
Units:	n/a
Defaults:	n/a
Range:	n/a

Description:

Monitor the status of the Fault Reset function. Pulse the Fault Reset ON to reset faults. Fault Reset can be assigned to a drive input. Select the PiCPro Reset Faults command to reset the faults in an ONLINE drive.

Feedback Type (Analog)

Category:	Motor
Method of Selection:	Read-Only
Units:	n/a
Defaults:	n/a
Range:	n/a

Description:

Presents the type of feedback device provided by the selected motor.

Firmware Date/Time Stamp (Analog)

Category:	Basic/System
Method of Selection:	Read-Only
Units:	n/a
Defaults:	Date/Time Stamp specified in personality file.
Range:	n/a

Description:

Provides the date and time of the firmware build that constitutes the firmware revision selected in parameter **Firmware Revision**.

Format: mm/dd/yyyy hh:mm AM/PM

Firmware Revision (Analog)

Category:	Basic/System
Method of Selection:	Main Menu
Units:	Major.Minor
Defaults:	Revision specified in personality file
Range:	1.00 to 99.99

Description:

Select a new version of firmware when the drive firmware is being changed. PiCPro will default to the latest version of firmware when inserting new drives.

Hall Offset (Analog)

Category:	Motor
Method of Selection:	Read-Only
Units:	Degrees
Defaults:	n/a
Range:	n/a

Description:

The angular offset between the Hall Effect sensors and the motor windings.

Index Offset (Analog)

Category:	Motor
Method of Selection:	Read-Only
Units:	Degrees
Defaults:	n/a
Range:	n/a

Description:

The angular offset between the index mark and the motor windings.

Inductance (Analog)

Category:	Motor
Method of Selection:	Read-Only
Units:	milliHenry
Defaults:	n/a
Range:	n/a

Description:

Phase-to-phase motor inductance.

In Speed Window (Analog)

Category:	Limits & Indicators
Method of Selection:	Read-Only
Units:	n/a
Defaults:	n/a
Range:	n/a

Description:

Monitor the In Speed Window indicator. When the absolute value of the Motor Velocity is greater than or equal to the Speed Window – Lower Limit and less than or equal to the Speed Window – Upper Limit, the In Speed Window indicator will be ON. The indicator can be assigned to a drive output.

Inertia (Analog)

Category:	Motor
Method of Selection:	Read-Only
Units:	Metric: kg-cm^2 English: in-lbs^2
Defaults:	n/a
Range:	n/a

Description:

The inertia of the motor rotor.

Inertia Ratio (Analog)

Category:	Tuning
Method of Selection:	Edit
Units:	Ratio of Application Inertia to Motor Inertia
Defaults:	1
Range:	0.00 to 500.0

Description:

Enter the ratio of the application inertia to the motor inertia.

Input (Analog)

Category:	I/O
Method of Selection:	Read-Only
Units:	n/a
Defaults:	n/a
Range:	n/a

Description:

This parameter displays the resultant state of the inputs assigned to it.

Input Assignment (Analog)

Category:	I/O
Method of Selection:	Dropdown Check List
Units:	Millivolts (mV)
Defaults:	Input1: Drive Enable Input 2: Fault Reset
Range:	See list below

Description:

Use Input Assignment to assign drive functions to drive inputs. When a drive input is ON, all functions assigned to it will be ON. Multiple functions can be assigned to a single input and a function can be assigned to multiple inputs. If a function is assigned to multiple inputs and any of the inputs are ON, the function will be ON.

Input Function Assignments	Monitors in these Categories
Drive Enable	System
Fault Reset	System
Current – Plus Enable	Tuning
Current – Minus Enable	Tuning
Override Mode Select	System
External Brake Release	Motor
Velocity Loop Integrator Inhibit	Tuning
Velocity Loop Integrator Hold	Tuning
Drive Cold Restart	System

Drive Enable – ON requests that the drive power output be enabled. Drive power will be enabled while Drive Enable is ON unless an overriding condition such as Fault is ON. Drive Enable OFF disables the drive power section. Drive Enable OFF will immediately disable drive power unless the Brake Apply Time is set to a negative value to delay power section shutdown. Drive Enable is always ON if it is not assigned to an input.

Fault Reset – Pulse ON to reset the Fault indication. Fault Reset should be OFF during normal operation.

Current – Plus Enable – ON allows the drive to output positive current. OFF prevents the drive from outputting positive current. Current – Plus Enable is always ON if it is not assigned to an input.

Current – Minus Enable – ON allows the drive to output negative current. OFF prevents the drive from outputting negative current. Current – Minus Enable is always ON if it is not assigned to an input.

Override Mode Select – ON causes the Override Mode to be the Active Operating Mode. OFF causes the Default Mode to be the Active Operating Mode.

External Brake Release – Assign External Brake Release to an input to use an external system to control the state of the Release Brake output. When the External Brake Release input is ON, the Release Brake output is turned on to close the relay and release the brake. When the External Brake Release input is OFF, the Release Brake output is turned off to open the relay and engage the brake. The Brake Apply Time and Brake Release Time are ignored when External Brake Release is assigned to an input.

Velocity Loop Integrator Inhibit – When ON, I Gain is zero and the integrator is zeroed. When OFF, I Gain is the velocity loop I Gain and the integrator functions normally.

Velocity Loop Integrator Hold – When ON, the integrator value is held at its present value. When OFF, the integrator functions normally.

Drive Cold Restart – An OFF-to-ON transition of Drive Cold Restart initiates the equivalent of a drive control power-on sequence.

Input Offset (Analog)

Category:	Command
Method of Selection:	Edit
Units:	Millivolts (mV)
Defaults:	0
Range:	-10000 to 10000

Description:

Enter an offset to be subtracted from the Analog Command. This is used to cancel the offset between the external control and the drive so a zero output from the control is seen as an Analog Command of zero. Select the PiCPro Remove Input Offset command to automatically calculate and enter the Input Offset for an ONLINE drive.

Input Polarity (Analog)

Category:	Command
Method of Selection:	Dropdown List
Units:	n/a
Defaults:	Forward
Range:	Forward, Reverse

Description:

Select the direction to move the motor in response to a positive input command. The motor will move in the direction that causes increasing feedback position when Forward is selected. The motor will move in the direction that causes decreasing feedback position when Reverse is selected.

Ke (Analog)

Category:	Motor
Method of Selection:	Read-Only
Units:	Volts / 1000 RPM
Defaults:	n/a
Range:	n/a

Description:

Motor Back EMF voltage parameter.

Kt (Analog)

Category:	Motor
Method of Selection:	Read-Only
Units:	Metric: n-m/Amp English: in-lb/Amp
Defaults:	n/a
Range:	n/a

Description:

Motor torque parameter.

Last/Previous Fault Message (Analog)

Category:	Fault History
Method of Selection:	Read-Only
Units:	n/a
Defaults:	n/a
Range:	n/a

Description:

These parameters are a record of the faults as they occur with the most recent on top.

Lines Per Rev (Analog)

Category:	Motor
Method of Selection:	Read-Only
Units:	Lines
Defaults:	n/a
Range:	n/a

Description:

The number of feedback device lines per motor revolution.

Low Pass Filter Bandwidth (Analog)

Category:	Tuning
Method of Selection:	Edit
Units:	Hz
Defaults:	150
Range:	1 to 32767

Description:

Enter the bandwidth of the Low Pass Filter. The lower the bandwidth, the greater the filter. This is used to eliminate noise due to mechanical resonance or feedback jitter.

Low Pass Filter Enable (Analog)

Category:	Tuning
Method of Selection:	Dropdown List
Units:	n/a
Defaults:	Active
Range:	Active, Not Active

Description:

Select Active to enable the Low Pass Filter.

Magnetic Offset (Analog)

Category:	Motor
Method of Selection:	Read-Only
Units:	Degrees
Defaults:	n/a
Range:	n/a

Description:

When using resolvers, it is the signed value in feedback counts of the distance between the resolver null and Magnetic zero.

Maximum Current (Analog)

Category:	Motor
Method of Selection:	Read-Only
Units:	Amps
Defaults:	n/a
Range:	n/a

Description:

The peak current the motor can accept.

Maximum Motor Temp (Analog)

Category:	Motor
Method of Selection:	Read-Only
Units:	Degrees C
Defaults:	n/a
Range:	n/a

Description:

The maximum allowable motor winding temperature.

Max Speed (Analog)

Category:	Motor
Method of Selection:	Read-Only
Units:	RPM
Defaults:	n/a
Range:	n/a

Description:

The maximum motor operating speed.

Minus Current Limit (Analog)

Category:	Basic/Tuning
Method of Selection:	Edit
Units:	Amps
Defaults:	Value of Maximum Current (lesser of Motor Max Current and Drive Max Current)
Range:	0.01 to Maximum Current (lesser of Motor Max Current and Drive Max Current)

Description:

Enter a negative output current limit to protect the application. If the drive and motor system can deliver more torque than the application can accept, use this limit to protect it.

Motor Calculated Temp. (Analog)

Category:	Motor
Method of Selection:	Read-Only
Units:	Degrees C
Defaults:	n/a
Range:	n/a

Description:

Monitor the calculated motor temperature. The Motor Calculated Temperature is the anticipated motor temperature based on the amount of current being delivered over time.

Motor Calculated Temp. Fault Limit (Analog)

Category:	Limits & Indicators
Method of Selection:	Read-Only
Units:	Degrees C
Defaults:	n/a
Range:	n/a

Description:

If the Motor Calculated Temperature exceeds this limit, the drive will fault. This protects the motor from damage due to excessive heat. The Motor Calculated Temperature is the anticipated motor temperature based on the amount of current being delivered over time.

Motor Calculated Temp. Warning Limit (Analog)

Category:	Limits & Indicators
Method of Selection:	Edit
Units:	Degrees C
Defaults:	110
Range:	0 to Maximum Motor Temp

Description:

If the Motor Calculated Temperature exceeds this limit, the warning indicator will be set. The Motor Calculated Temperature is the anticipated motor temperature based on the amount of current being delivered over time. The warning indicator can be assigned to a drive output to provide an early warning before a fault limit is exceeded.

Motor Data Status (Analog)

Category:	Basic/System/Motor
Method of Selection:	Read-Only
Units:	n/a
Defaults:	n/a
Range:	n/a

Description:

This parameter indicates the current status of all of the motor data. There are three states that the motor data can be:

- **Same** – The current motor data is the same as the information in the motor database file on the workstation.
- **Different** – The current motor data is different than the information in the motor database file on the workstation.
- **Not Found** – The current motor data was not found in the motor database file on the workstation.

Motor Feedback - A (Analog)

Category:	Motor
Method of Selection:	Read-Only
Units:	n/a
Defaults:	n/a
Range:	n/a

Description:

ON when the A signal of quadrature feedback is ON.

Motor Feedback - B (Analog)

Category:	Motor
Method of Selection:	Read-Only
Units:	n/a
Defaults:	n/a
Range:	n/a

Description:

ON when the B signal of quadrature feedback is ON.

Motor Feedback - I (Analog)

Category:	Motor
Method of Selection:	Read-Only
Units:	n/a
Defaults:	n/a
Range:	n/a

Description:

ON when the Z signal of quadrature feedback is ON.

Motor Feedback - S1 (Analog)

Category:	Motor
Method of Selection:	Read-Only
Units:	n/a
Defaults:	n/a
Range:	n/a

Description:

ON when the Hall Effect alignment signal S1 is ON.

Motor Feedback - S2 (Analog)

Category:	Motor
Method of Selection:	Read-Only
Units:	n/a
Defaults:	n/a
Range:	n/a

Description:

ON when the Hall Effect alignment signal S2 is ON.

Motor Feedback - S3 (Analog)

Category:	Motor
Method of Selection:	Read-Only
Units:	n/a
Defaults:	n/a
Range:	n/a

Description:

ON when the Hall Effect alignment signal S3 is ON.

Motor Model (Analog)

Category:	Basic/System/Motor
Method of Selection:	Dropdown List
Units:	n/a
Defaults:	n/a
Range:	Extensive, see dropdown list for available choices.

Description:

Select the Model of the motor being used. The motor model is selected when the drive is being inserted into the drive list. When a motor model is selected, all motor specific parameters are set to that motor's factory default values.

Motor Position (Analog)

Category:	Basic/System/Motor
Method of Selection:	Read-Only
Units:	Counts
Defaults:	n/a
Range:	-2,147,483,648 to +2,147,483,647

Description:

Monitor the accumulated motor feedback.

Motor Temp. (Analog)

Category:	System
Method of Selection:	Read-Only
Units:	see below
Defaults:	n/a
Range:	See below

Description:

Monitor the motor temperature.

Motor – Thermal Device	Motor Temp. display	Range
none	Motor Calculated Temp.	0 – 180 Degrees C
Thermal Switch	Closed – ok, Open – motor is overheated	
Thermistor	Measured Motor Temperature	0 – 180 Degrees C

Motor Temp. Fault Limit (Analog)

Category:	Limits & Indicators
Method of Selection:	Read-Only
Units:	Degrees C
Defaults:	n/a
Range:	n/a

Description:

If the Motor Thermistor Temperature exceeds this limit, the drive will fault. This protects the motor from damage due to excessive heat. This typically occurs when the motor is undersized for the application and excessive output current causes the motor temperature to rise. This may also occur when the motor heatsink is inadequate or the ambient temperature around the motor is too high. This only applies when the Motor – Thermal Device is a thermistor.

Motor Temp. Warning Limit (Analog)

Category:	Limits & Indicators
Method of Selection:	Edit
Units:	Degrees C
Defaults:	110
Range:	0 to Maximum Motor Temp

Description:

If the Motor Thermistor Temperature exceeds this limit, the warning indicator will be set. The warning indicator can be assigned to a drive output to provide an early warning before a fault limit is exceeded. This only applies when the Motor – Thermal Device is a thermistor.

Motor Thermal Switch (Analog)

Category:	Motor
Method of Selection:	Read-Only
Units:	n/a
Defaults:	n/a
Range:	n/a

Description:

Monitor the state of the motor thermal switch when the Motor – Thermal Device is set to Thermal Switch. Open when the motor has overheated, closed when the motor is not overheated.

Motor Thermistor Temp. (Analog)

Category:	Motor
Method of Selection:	Read-Only
Units:	Degrees C
Defaults:	n/a
Range:	n/a

Description:

Monitor the temperature of the motor thermistor when the Motor – Thermal Device is set to Thermistor.

Motor Velocity (Analog)

Category:	Basic/System/Tuning/Motor
Method of Selection:	Read-Only
Units:	RPM
Defaults:	n/a
Range:	n/a

Description:

Monitor the actual motor velocity.

Operation Time Elapsed (Analog)

Category:	System/Fault History
Method of Selection:	Read-Only
Units:	d(days) h(hours) m(minute) s(seconds)
Defaults:	n/a
Range:	0 to 49710d 6h 28m 16s

Description:

Monitor the time that the drive has been powered.

Output (Analog)

Category:	I/O
Method of Selection:	Read-Only
Units:	n/a
Defaults:	n/a
Range:	n/a

Description:

This parameter displays the resultant state of the outputs assigned to it.

Output Assignment (Analog)

Category:	I/O
Method of Selection:	Dropdown Check List
Units:	n/a
Defaults:	Output 3: 220 V Shunt on 440v Drive Output 4: Drive Ready Output 5: Brake Release
Range:	n/a

Description:

Use Output Assignments to assign drive status indicators to drive outputs. When a status indicator is ON, the output it is assigned to will be ON. A status indicator can be assigned to multiple outputs and multiple status indicators can be assigned to a single output. When multiple status indicators are assigned to a single output, the output will be ON, when any of the status indicators are ON.

Output Status Assignments	Monitors in these Categories
Drive Ready	System (Drive Status)
Drive Enabled	System (Drive Status)
At Plus Current Limit	Tuning
At Minus Current Limit	Tuning
Fault	System (Drive Status)
Warning	System (Drive Status)
Drive Bus Charged	System
Drive Ready and Bus Charged	System (Drive Status)
Release Brake	Motor
At Zero Speed	Limits & Indicators
In Speed Window	Limits & Indicators
Up to Speed	Limits & Indicators
Startup Commutation Complete	Motor
220V Shunt on 440V Drive	Limits & Indicators

Drive Ready – ON indicates the drive power section will be enabled when Drive Enable is ON and the bus is charged. Drive Ready is ON when Fault is OFF, Drive Bus Charged is ON in a 460V drive and Cold Restart Required is OFF. Drive Ready is OFF when Fault is ON or Drive Bus Charged is OFF in a 460V drive or Cold Restart Required is ON.

Drive Enabled – ON when Drive Enable is ON and the drive power section is enabled. OFF when the drive power section is disabled. See parameter Drive Status for a detailed description.

At Plus Current Limit – ON when the Current Command equals the Plus Current Limit. OFF when the Current command is less than the Plus Current Limit.

At Minus Current Limit – ON when the Current Command equals the Minus Current Limit. OFF when the Current command is greater than the Minus Current Limit.

Fault – ON when a fault condition has been detected. OFF when no faults are present. Select the PiCPro Reset Faults command or pulse the Reset Faults input to reset the Fault indicator.

Warning – ON when any warning limit is being exceeded. OFF when no warning limits are being exceeded.

Drive Bus Charged – ON when the Drive Bus Voltage is above the minimum level required to safely operate a motor. OFF when the Drive Bus Voltage is below the minimum level require to safely operate a motor.

Drive Ready and Bus Charged – ON indicates the drive power section will be enabled when Drive Enable is ON. Drive Ready and Bus Charged is ON when Fault is OFF. Drive Bus charged in ON and Cold Restart Required is OFF. Drive Ready and Bus Charged is OFF when Fault is ON or Drive Bus Charged is OFF or Cold Restart Required is ON.

Release Brake – ON to close the brake relay and release the brake. OFF to open the brake relay and engage the brake. See Motor - Brake Apply Time and Motor - Brake Release Time parameters for a detailed description of Release Brake control.

At Zero Speed – ON when the absolute value of the Motor Velocity is less than or equal to the Zero Speed Rate.

In Speed Window – ON when the absolute value of the Motor Velocity is greater than or equal to the Speed Window – Lower Limit and less than or equal to the Speed Window – Upper Limit.

Up to Speed – ON when the absolute value of the Motor Velocity is greater than or equal to the Up to Speed Rate.

Startup Commutation Complete – ON when the drive has completed startup and the motor is being controlled normally.

220V Shunt on 440V Drive – ON when bus voltage exceeds 415V while using a 220V motor. OFF when the bus voltage then goes below 400V. **Note:** This indicator can only be assigned to Output 3. If assigned to Output3, '220V Shunt on 440V Drive' must be the only indicator assigned. If other items are assigned, a message will result and the other items will be cleared. This output cannot be overridden; if this indicator is assigned. If this indicator is assigned on a 460 volt motor or a 230 volt MMC Smart Drive, it will be ignored. **Note also:** The drive will disable PWM with an over voltage fault, when the motor is a 220 volt motor and if the bus voltage exceeds 440 volts (regardless of whether this indicator is enabled).

Over Current Fault Limit (Analog)

Category:	Limits & Indicators
Method of Selection:	Edit
Units:	Amps
Defaults:	Maximum Current (lesser of Motor Max Current and Drive Max Current)
Range:	0.01 to Maximum Current (lesser of Motor Max Current and Drive Max Current)

Description:

If the Current – Average exceeds this limit, the drive will fault. This is used to protect the application from excessive torque. If the drive and motor can deliver more torque than the application can accept, use this limit to protect it.

Over Speed Fault Limit (Analog)

Category:	Limits & Indicators
Method of Selection:	Edit
Units:	RPM
Defaults:	120% of Motor Maximum Speed
Range:	1 to 120% of Motor Maximum Speed

Description:

If the Motor Velocity exceeds this limit, the drive will fault. This typically occurs when an unexpected event causes system instability resulting in improper motor control. If the Motor – Maximum Speed is greater than that required by the application, this limit can be set to create a fault at a lower speed.

Override Mode (Analog)

Category:	System
Method of Selection:	Dropdown List
Units:	n/a
Defaults:	Velocity
Range:	Velocity, Torque

Description:

Select the mode that will be the Active Operating Mode when Override Mode Select is ON.

Override Mode Select (Analog)

Category:	System
Method of Selection:	Read-Only
Units:	n/a
Defaults:	n/a
Range:	n/a

Description:

Monitor the Override Mode Select function. The Active Operating Mode equals Default Mode when Override Mode Select is OFF. The Active Operating Mode equals Override Mode when Override Mode Select is ON. Override Mode Select can be assigned to a drive input.

Plus Current Limit (Analog)

Category:	Basic/Tuning
Method of Selection:	Edit
Units:	Amps
Defaults:	Value of Maximum Current (lesser of Motor Max Current and Drive Max Current)
Range:	0.01 to Maximum Current (lesser of Motor Max Current and Drive Max Current)

Description:

Enter a positive output current limit to protect the application. If the drive and motor system can deliver more torque than the application can accept, use this limit to protect it.

Poles Per Rev (Analog)

Category:	Motor
Method of Selection:	Read-Only
Units:	n/a
Defaults:	n/a
Range:	n/a

Description:

Number of poles per motor revolution

Rated Voltage (Analog)

Category:	Motor
Method of Selection:	Read-Only
Units:	Volts
Defaults:	n/a
Range:	n/a

Description:

Voltage rating of the motor.

Release Brake (Analog)

Category:	Motor
Method of Selection:	Read-Only
Units:	n/a
Defaults:	n/a
Range:	n/a

Description:

Monitor the status of the Release Brake indicator. By default, Release Brake is assigned to drive output 5, a relay output. When Release Brake is ON, the relay is closed. When Release Brake is OFF, the relay is open.

Reset Motor Position (Analog)

Category:	Motor
Method of Selection:	Edit
Units:	n/a
Defaults:	n/a
Range:	n/a

Description:

When set to a value greater than zero, the Motor Position will be reset to zero when it exceeds that value.

Resistance (Analog)

Category:	Motor
Method of Selection:	Read-Only
Units:	Ohms
Defaults:	n/a
Range:	n/a

Description:

Phase-to-phase motor resistance.

Speed Window - Lower Limit (Analog)

Category:	Limits & Indicators
Method of Selection:	Edit
Units:	RPM
Defaults:	20
Range:	0 to 65535

Description:

Enter the lower limit for the In Speed Window indicator. When the absolute value of the Motor Velocity is greater than or equal to the Speed Window – Lower Limit and less than or equal to the Speed Window – Upper Limit, the In Speed Window indicator will be ON.

Speed Window - Upper Limit (Analog)

Category:	Limits & Indicators
Method of Selection:	Edit
Units:	RPM
Defaults:	200
Range:	0 to 65535

Description:

Enter the upper limit for the In Speed Window indicator. When the absolute value of the Motor Velocity is greater than or equal to the Speed Window – Lower Limit and less than or equal to the Speed Window – Upper Limit, the In Speed Window indicator will be ON.

Startup Commutation Complete (Analog)

Category:	Motor
Method of Selection:	Read-Only
Units:	n/a
Defaults:	n/a
Range:	n/a

Description:

ON after the drive has completed the motor startup commutation process.

Thermal Capacitance (Analog)

Category:	Motor
Method of Selection:	Read-Only
Units:	Watt-sec / deg C
Defaults:	n/a
Range:	n/a

Description:

Thermal capacitance from the winding to ambient.

Thermal Device (Analog)

Category:	Motor
Method of Selection:	Read-Only
Units:	n/a
Defaults:	n/a
Range:	n/a

Description:

The type of thermal sensor embedded in the motor windings.

Thermal Resistance (Analog)

Category:	Motor
Method of Selection:	Read-Only
Units:	Deg C / Watt
Defaults:	n/a
Range:	n/a

Description:

Describes the head dissipation properties of the motor.

Torque Mode Command Scale (Analog)

Category:	Command
Method of Selection:	Read-Only
Units:	ma / Volt
Defaults:	1
Range:	1 to 10000

Description:

Enter the scale factor used to convert incoming Analog Command voltage to a Current Command when the Active Operating Mode is Torque. Enter the Current Command in milliamps for an Analog Command of 1 volt.

Up To Speed (Analog)

Category:	Limits & Indicators
Method of Selection:	Read-Only
Units:	n/a
Defaults:	n/a
Range:	n/a

Description:

Monitor the Up to Speed indicator. The indicator is ON when the absolute value of the Motor Velocity is greater than or equal to the Up to Speed Rate. The indicator can be assigned to a drive output.

Up To Speed Rate (Analog)

Category:	Limits & Indicators
Method of Selection:	Edit
Units:	RPM
Defaults:	300
Range:	0 to 32767

Description:

Enter the Up to Speed indicator rate. When the absolute value of the Motor Velocity is greater than or equal to the Up to Speed Rate, the Up to Speed indicator will be ON.

Velocity Command (Analog)

Category:	System/Command/Tuning
Method of Selection:	Read-Only
Units:	RPM
Defaults:	n/a
Range:	Max Speed of motor to + Max Speed of motor

Description:

Monitor the command to the velocity loop when the Active Operating Mode is Velocity.

Velocity Command Limit (Analog)

Category:	Command
Method of Selection:	Edit
Units:	RPM
Defaults:	120% of Max Speed of motor
Range:	1 to Max Speed of motor

Description:

Enter the limit of the Velocity Command when the Active Operating Mode is Velocity.

Velocity Error (Analog)

Category:	Tuning
Method of Selection:	Read-Only
Units:	RPM
Defaults:	n/a
Range:	n/a

Description:

Monitor the difference between Velocity Command and Motor Velocity.

Velocity Loop D Gain (Analog)

Category:	Tuning
Method of Selection:	Edit
Units:	n/a
Defaults:	0
Range:	0.000 to 32.000

Description:

Enter the Velocity Loop Derivative Gain. This can be used to compensate for mechanical resonance. Typically, this is set to zero.

Velocity Loop I Gain (Analog)

Category:	Basic/Tuning
Method of Selection:	Edit
Units:	n/a
Defaults:	2000
Range:	0 to 32767

Description:

Enter the Velocity Loop Integral Gain. Increasing the I gain increases the stiffness of the system. Excessive I gain can cause instability.

Velocity Loop Integrator Hold (Analog)

Category:	Tuning
Method of Selection:	Read-Only
Units:	n/a
Defaults:	n/a
Range:	n/a

Description:

Monitor the Velocity Loop Integrator Hold function. When ON, the integrator value is held at its present value. When OFF, the integrator functions normally. Velocity Loop Integrator Hold can be assigned to a drive input.

Velocity Loop Integrator Inhibit (Analog)

Category:	Tuning
Method of Selection:	Read-Only
Units:	n/a
Defaults:	n/a
Range:	n/a

Description:

Monitor the Velocity Loop Integrator Inhibit function. When ON, I gain is zero and the integrator is zeroed. When OFF, the I gain is the Velocity Loop I Gain and the integrator functions normally. Velocity Loop Integrator Inhibit can be assigned to a drive input.

Velocity Loop P Gain (Analog)

Category:	Basic/Tuning
Method of Selection:	Edit
Units:	n/a
Defaults:	50
Range:	0.0 to 32767.7

Description:

Enter the Velocity Loop Proportional Gain. Increasing the P gain improves response time. Excessive P gain can cause instability. An extremely low P gain can cause poor servo response.

Velocity Mode Command Scale (Analog)

Category:	Command
Method of Selection:	Edit
Units:	RPM/Volt
Defaults:	0.1
Range:	0.1 to 6499.9

Description:

Enter the scale factor used to convert the Analog Command voltage to a Velocity Command when the Active Operating Mode is Velocity. Enter a Velocity Command in RPM for an Analog Command of 1 volt.

Zero Speed Rate (Analog)

Category:	Limits & Indicators
Method of Selection:	Edit
Units:	RPM
Defaults:	10
Range:	0 to 32767

Description:

Enter the At Zero Speed indicator rate. When the absolute value of the Motor Velocity is less than or equal to the Zero Speed Rate, the At Zero Speed indicator will be ON.

APPENDIX Q - Analog Drive Faults and Warnings

(01) Drive Heatsink Temp. Warning (Analog)

Category: Faults & Warnings

Method of Selection: Read-Only

Description:

This warning is set if the Drive Heatsink Temperature exceeds the Drive Heatsink Temperature Warning limit. The warning indicator can be assigned to a drive output to provide an early warning before a fault limit is exceeded.

(02) Drive Ambient Temp. Warning (Analog)

Category: Faults & Warnings

Method of Selection: Read-Only

Description:

This warning is set if the Drive Ambient Temperature exceeds the Drive Ambient Temperature Warning limit. This warning only applies to MMC-SD-x-460 systems. The warning indicator can be assigned to a drive output to provide an early warning before a fault limit is exceeded.

(03) Motor Temp. Warning (Analog)

Category: Faults & Warnings

Method of Selection: Read-Only

Description:

This warning is set if the Motor – Thermal Device is a thermistor and Motor Thermistor Temperature exceeds the Motor Temperature Warning limit. The warning indicator can be assigned to a drive output to provide an early warning before a fault limit is exceeded.

(04) Motor Calculated Temp. Warning (Analog)

Category: Faults & Warnings

Method of Selection: Read-Only

Description:

This warning is set if the Motor Calculated Temperature exceeds the Motor Calculated Temperature Warning limit. The Motor Calculated Temperature is the anticipated motor temperature based on the amount of current being delivered over time. The warning indicator can be assigned to a drive output to provide an early warning before a fault limit is exceeded.

(11) Drive Memory Fault (Analog)

Category: Faults & Warnings

Method of Selection: Read-Only

Description:

This fault is set if the drive's non-volatile memory is not functioning properly. If this fault occurs, contact Danaher Motion Tech Support.

(12) Drive Bus Over Voltage Fault (Analog)

Category: Faults & Warnings

Method of Selection: Read-Only

Description:

This fault is set if the Drive Bus Voltage exceeds the Drive Bus Over Voltage Fault limit. This protects the drive's power section from voltage levels that would damage it. If a motor is rated for 220/230 volts and is used with a 440/460 volt drive, this limit will protect the motor from voltage levels that would damage it. Typical causes for this fault include an incoming line voltage that is too high or back EMF generated by the motor during deceleration requiring an external shunt system to dissipate energy.

(13) Drive PM1 Over Current Fault (Analog)

Category: Faults & Warnings

Method of Selection: Read-Only

Description:

This fault is set if the Current-Feedback exceeds the Drive Over Current Fault limit. This protects the drive's power section from damage due to excessive current output.

(15) Motor Temp. Fault (Analog)

Category: Faults & Warnings

Method of Selection: Read-Only

Description:

This fault is set if the motor overheats. This protects the motor from damage due to excessive heat. This typically occurs when the motor is undersized for the application and excessive output current causes the motor temperature to rise. This may also occur when the motor heatsink is inadequate of the ambient temperature around the motor is too high.

Motor – Thermal Device	Motor Temp. Fault
None	-
Thermal Switch	Set if Motor – Thermal Switch is open
Thermistor	Set if Motor Temp. > Motor Temp. Fault Limit

(16) Continuous Current Fault (Analog)

Category: Faults & Warnings

Method of Selection: Read-Only

Description:

This fault indicates the current has exceeded the continuous motor current rating for an extended period of time.

(17) Drive Heatsink Temp. Fault (Analog)

Category: Faults & Warnings

Method of Selection: Read-Only

Description:

This fault is set if the Drive Heatsink Temperature exceeds the Drive Heatsink Temperature Fault limit. This protects the drive's power section from damage due to overheating. This occurs when the heatsink temperature rises due to excessive power output, excessive ambient temperature or inadequate airflow.

(22) Drive F1 Feedback Fault (Analog)

Category: Faults & Warnings

Method of Selection: Read-Only

Description:

This fault is set if an error is detected in the motor feedback. Typical causes for this fault include a faulty feedback cable, electrical noise on the feedback signal due to improper feedback cable routing or failure of the feedback device in the motor.

(23) Drive Ambient Temp. Fault (Analog)

Category:	Faults & Warnings
Method of Selection:	Read-Only

Description:

This fault is set if the Drive Ambient Temperature exceeds the Drive Ambient Temperature Fault limit. This protects from erratic operation of the drive due to excessive heat. This fault only applies to MMC-SD-x-460 systems. Typically, this occurs if the drive enclosure temperature rises due to excessive power dissipation.

(24) Motor Calculated Temp. Fault (Analog)

Category:	Faults & Warnings
Method of Selection:	Read-Only

Description:

This fault is set if the Motor Calculated Temperature exceeds the Motor Calculated Temperature Fault limit. This protects the motor from damage due to excessive heat. The Motor Calculated Temperature is the anticipated motor temperature based on the amount of current being delivered over time.

(25) Drive Timing Fault (Analog)

Category:	Faults & Warnings
Method of Selection:	Read-Only

Description:

This fault is set if a timing error is detected in the execution of the control algorithms performed by the drive's digital signal processor. If this fault occurs, contact Danaher Motion Tech Support.

(26) Drive Interface Fault (Analog)

Category:	Faults & Warnings
Method of Selection:	Read-Only

Description:

This fault is set if a communication error is detected in the transmission of information between the drive's digital signal processor and the drive's power section. If this fault occurs, contact Danaher Motion Tech Support.

(27) User Set Fault (Analog)

Category: Faults & Warnings

Method of Selection: Read-Only

Description:

This fault is set by selecting the PiCPro Set User Fault command or when activating the Control Panel mode while the drive is enabled.

(31) Drive F1 Communication Fault (Analog)

Category: Faults & Warnings

Method of Selection: Read-Only

Description:

This fault is set if a communication error is detected in the transmission of information between the drive and a high resolution or multi-turn absolute feedback device. Typical causes for this fault include a faulty feedback cable, electrical noise on the feedback signal due to improper feedback cable routing or failure of the feedback device in the motor.

(32) Over Speed Fault (Analog)

Category: Faults & Warnings

Method of Selection: Read-Only

Description:

This fault is set if the Motor Velocity exceeds the Over Speed Fault limit. this typically occurs when an unexpected event causes system instability resulting in improper motor control.

(33) Over Current Fault (Analog)

Category: Faults & Warnings

Method of Selection: Read-Only

Description:

This fault is set if the Current – Average exceeds the Over Current Fault limit. This is used to protect the application from excessive torque. If the drive and motor can deliver more torque than the application can accept, use this limit to protect it.

(34) Drive Communication Fault (Analog)

Category: Faults & Warnings

Method of Selection: Read-Only

Description:

This fault is set if a communication error occurs while drive control is being performed using the PiCPro Control Panel tools. A typical cause for this fault is disconnecting the cable between the drive and the workstation while in Control Panel mode.

(35) Drive Power Module Fault (Analog)

Category: Faults & Warnings

Method of Selection: Read-Only

Description:

This fault is set by the drive's power section if it detects a fault condition. This fault is typically set when the power section detects a demand for power that would cause damage to the power section.

(36) Drive Setup Data Fault (Analog)

Category: Faults & Warnings

Method of Selection: Read-Only

Description:

This fault is set if the drive software determines that the feedback type found is not the same as the feedback type specified in the motor data. Down load the configuration data if this occurs. If this fault persists, contact Danaher Motion Tech Support.

(41) Drive Relay Fault (Analog)

Category: Faults & Warnings

Method of Selection: Read-Only

Description:

This fault is set if the drive’s power section detects an error during power up. If this fault occurs, contact Danaher Motion Tech Support.

(44) Motor Ground Fault (Analog)

Category:	Faults & Warnings
Method of Selection:	Read-Only

Description:

This fault indicates a ground fault has occurred. Specifically, the current in the motor ground is excessive.

Note: This fault only applies to MMC-SD-x-460 systems.

(42) Drive PM2 Over Current Fault (Analog)

Category:	Faults & Warnings
Method of Selection:	Read-Only

Description:

This fault is set if the Current – Feedback exceeds the Drive Over Current Fault limit. This protects the drive's power section from damage due to excessive current output. This fault applies to MMC-SD-x-460 drives only.

(43) Drive PM Over Temp Fault (Analog)

Category:	Faults & Warnings
Method of Selection:	Read-Only

Description:

This fault is set if the drive power module temperature exceeds the Drive Power Module Temperature Fault limit. This protects the drive's power section from damage due to overheating. This occurs when the power module temperature rises due to excessive power output, excessive ambient temperature or inadequate airflow.

(45) Drive AC Input Over Voltage Fault (Analog)

Category:	Faults & Warnings
Method of Selection:	Read-Only

Description:

This fault indicates the incoming AC voltage is too high.

Note: This fault only applies to MMC-SD-x-460 systems.

(53) Hardware Failure Fault (Analog)

Category: Faults & Warnings

Method of Selection: Read-Only

Description:

This fault indicates an unrecoverable hardware error.

(xy) Drive Bus Under Voltage Fault (Analog)

Category: Faults & Warnings

Method of Selection: Read-Only

Description:

This fault is set if the Drive Bus Voltage falls below the Drive Bus Under Voltage Fault limit. This protects the application from the drive's inability to properly control the motor when the bus voltage is inadequate. A typical cause of this fault is an incoming line voltage that is too low.

APPENDIX R - Digital Drive Parameters

2-Pole Filter Enable (Digital)

Category:	Advanced Tuning
Method of Selection:	Dropdown List
Units:	n/a
Defaults:	Not Active
Range:	Active, Not Active

Description:

Select Active to enable the 2-Pole Filter. The Notch Filter and the 2-Pole filter may not be enabled at the same time.

2-Pole Filter Frequency (Digital)

Category:	Advanced Tuning
Method of Selection:	Edit
Units:	Hz
Defaults:	100
Range:	10 to 600

Description:

Enter the cut off Frequency of the 2-Pole, low pass filter. The damping ratio of the filter is fixed at 0.707.

220V Shunt on 440V Drive (Digital)

Category:	Limits & Indicators
Method of Selection:	Read-Only
Units:	n/a
Defaults:	n/a
Range:	n/a

Description:

Monitor the bus voltage while using a 220V motor with a 440V drive. The indicator is ON when the bus voltage exceeds 415V while using a 220V motor. The indicator will then go OFF when the bus voltage goes below 400V. 220V Shunt on 440V Drive can be monitored using READ_SV variable 67 AND (16#1000).

Absolute Reference Complete (Digital)

Category:	Feedback and Scaling
Method of Selection:	Read-Only
Units:	n/a
Defaults:	n/a
Range:	n/a

Description:

Monitor the state of the Absolute Reference Complete indicator. ON if an absolute reference has been performed. OFF if an absolute reference has not been performed or if the absolute reference has been cleared.

Accel Limit (Digital)

Category:	Command
Method of Selection:	Edit
Units:	RPM/Sec
Defaults:	20,000
Range:	16 to 1048560

Description:

Enter the limit of the rate-of-change of the Velocity Command when it is increasing and the AccDec Limits are Active.

AccDec Limits Active (Digital)

Category:	Command
Method of Selection:	Dropdown List
Units:	n/a
Defaults:	Not Active
Range:	Not Active, Active

Description:

Select Active to apply rate-of-change limits to the Velocity Command. Set to Not Active when used with an MMC for closed loop servo applications.

Acceleration Feedforward (Digital)

Category:	Tuning
Method of Selection:	Dropdown list
Units:	n/a
Defaults:	On
Range:	On, Off

Description:

The purpose of Acceleration Feedforward is to be responsive to velocity command changes by applying additional current based on inertial load on the motor. Turning Acceleration Feedforward ‘On’ will cause the drive to generate an additional current command when the velocity command changes. The magnitude of the additional current is dependent on the value of Inertia Ratio and the amount of velocity command change. Turning Acceleration Feedforward ‘Off’ will prevent this additional current command.

Active Operating Mode (Digital)

Category:	Basic/System/Command/Tuning
Method of Selection:	Read-Only
Units:	n/a
Defaults:	Position
Range:	Velocity Control, Torque Control, Position, Autotune, Alignment Control

Description:

Monitor the drive control mode. In Position Control mode when the drive is enabled it closes the position, velocity and current loops and follows the Digital Link Position Command. In Control Panel Velocity Control mode the Active Operating Mode is Velocity Control. In Control Panel Current Control mode the Active Operating Mode is Torque Control. In control panel Autoune mode, the Active Operating Mode is Autotune. In Control Panel Alignment Mode the Active Operating Mode is Alignment Control.

Alignment Mode (Digital)

Category:	Motor
Method of Selection:	Read-Only
Units:	n/a
Defaults:	n/a
Range:	n/a

Description: Defines the Startup Commutation.

Analog Command (Digital)

Category:	Basic/System/Command
Method of Selection:	Read-Only
Units:	Millivolts (mV)
Defaults:	n/a
Range:	-10000 to 10000

Description:

Monitor the voltage on the command input of the drive I/O connector. In Velocity mode, the ±10 volt Analog Command is converted to the Velocity Command. In Torque mode, the ±10 volt Analog Command is converted to the Current Command.

Analog Input (Digital)

Category:	I/O
Method of Selection:	Read-Only
Units:	Millivolts (mV)
Defaults:	n/a
Range:	-10000 to 10000

Description:

Monitor the voltage on the analog input of the drive I/O connector. Analog input can be monitored using READ_SV variable 70.

At Minus Current Limit (Digital)

Category:	Tuning
Method of Selection:	Read-Only
Units:	n/a
Defaults:	n/a
Range:	n/a

Description:

Monitor the state of the At Minus Current Limit indicator. ON when the Current Command equals the Minus Current Limit. The At Minus Current Limit indicator can be assigned to a drive output. READ_SV variable 67 AND (16#00000020) to monitor At Minus Current Limit.

At Plus Current Limit (Digital)

Category:	Tuning
Method of Selection:	Read-Only
Units:	n/a
Defaults:	n/a
Range:	n/a

Description:

Monitor the state of the At Plus Current Limit indicator. ON when the Current Command equals the Plus Current Limit. The At Plus Current Limit indicator can be assigned to a drive output. READ_SV variable 67 AND (16#00000010) to monitor At Plus Current Limit.

At Zero Speed (Digital)

Category:	Limits & Indicators
Method of Selection:	Read-Only
Units:	n/a
Defaults:	n/a
Range:	n/a

Description:

Monitor the At Zero Speed indicator. The indicator is ON when the absolute value of the Motor Velocity is less than or equal to the Zero Speed Rate. The indicator can be assigned to a drive output. At Zero Speed can be monitored using READ_SV variable 67 AND (16#2).

Brake Apply Time (Digital)

Category:	Motor
Method of Selection:	Edit
Units:	msec
Defaults:	0
Range:	-32768 to 32767

Description:

Enter the time value used to define the Release Brake and Drive Enabled controls when Drive Enable changes from ON to OFF while Fault is OFF. Release Brake ON allows motion. Release Brake OFF holds the load. Brakes internal to servomotors are meant to prevent a stationary load from moving, not to decelerate the load.

Drive Enable	Fault	Brake Apply Time	Release Brake	Drive Enabled
on to off	off	= 0	off immediately	off immediately
on to off	off	> 0	off after Brake Apply Time	off immediately
on to off	off	< 0	off immediately	off after absolute value of Brake Apply Time
-	on	-	off immediately	off immediately

Enter a negative value for the Brake Apply Time to allow the brake to take hold before power to the motor is removed.

Brake Release Time (Digital)

Category: Motor

Method of Selection: Edit

Units: msec

Defaults: 0

Range: -32768 to 32767

Description:

Enter the time value used to define the Release Brake and Drive Enabled controls when Drive Enable changes from OFF to ON. Release Brake ON allows motion. Release Brake OFF holds the load. Brakes internal to servomotors are meant to prevent a stationary load from moving, not to decelerate the load.

Drive Enable	Drive Ready	Fault	Brake Release Time	Release Brake	Drive Enabled
off to on	on	off	= 0	on immediately	on immediately
off to on	on	off	> 0	on after Brake Release Time	on immediately
off to on	on	off	< 0	on immediately	on after absolute value of Brake Release Time

Enter a positive value for the Brake Release Time to allow the drive to take control of the motor before the brake is released.

Cold Restart Required (Digital)

Category:	Basic/System
Method of Selection:	Read-Only
Units:	n/a
Defaults:	n/a
Range:	n/a

Description:

Monitor the need to Cold Restart the drive. ON when user has changed Motor Model, F1 Polarity, F2 Polarity or User Defined Position Scaling. An OFF-to-ON transition of Drive Cold Restart initiates the equivalent of a drive control power-on sequence. The Drive Cold Restart function can be assigned to a drive input. Select the PiCPro Drive Cold Restart command to initiate a Cold Restart in the connected drive. The application ladder scan must be stopped to perform a Cold Restart.

Continuous Current (Digital)

Category:	Motor
Method of Selection:	Read-Only
Units:	Amps
Defaults:	n/a
Range:	n/a

Description:

The current the motor can accept continuously without overheating.

Control Communication Errors (Digital)

Category:	Basic/System
Method of Selection:	Read-Only
Units:	n/a
Defaults:	n/a
Range:	0 to 65535

Description:

Monitor the number of Digital Link errors detected by the control. Two consecutive errors will cause a Digital Drive Communication Error to be reported by E_Errors. Control Communication Errors is reset to zero each time DSTRTSRV is called.

Current - Average (Digital)

Category:	Basic/System/Tuning
Method of Selection:	Read-Only
Units:	Amps
Defaults:	n/a
Range:	n/a

Description:

Monitor the drive average actual current output. Current - Average can be monitored using READ_SV variable 74. The average current is the actual current, filtered by a 0.5 second time constant.

Current - Feedback (Digital)

Category:	Tuning
Method of Selection:	Read-Only
Units:	Amps
Defaults:	n/a
Range:	n/a

Description:

Monitor the output current as measured by the drive current feedback sensing circuit.

Current - Minus Enable (Digital)

Category:	Tuning
Method of Selection:	Read-Only
Units:	n/a
Defaults:	n/a
Range:	n/a

Description:

Monitor the status of the Current – Minus Enable function. ON enables the drive to respond to a negative Position, Velocity or Current Command. OFF disables the drive from responding to a negative Position, Velocity or Current command. Current – Minus Enable can be assigned to a drive input. Current - Minus Enable can be monitored and controlled by READ_SV/WRITE_SV variable 85. If not assigned to an input, it defaults to ON.

Current - Peak Minus (Digital)

Category:	Tuning
Method of Selection:	Read-Only
Units:	Amps
Defaults:	n/a
Range:	n/a

Description:

Monitor the maximum negative current output. Use the PiCPro Reset Peaks command to reset this value to zero in the connected drive.

Current - Peak Plus (Digital)

Category:	Tuning
Method of Selection:	Read-Only
Units:	Amps
Defaults:	n/a
Range:	n/a

Description:

Monitor the maximum positive current output. Use the PiCPro Reset Peaks command to reset this value to zero in the connected drive.

Current - Plus Enable (Digital)

Category:	Tuning
Method of Selection:	Read-Only
Units:	n/a
Defaults:	n/a
Range:	n/a

Description:

Monitor the status of the Current – Plus Enable function. ON enables the drive to respond to a positive Position, Velocity or Current Command. OFF disables the drive from responding to a positive Position, Velocity or Current command. Current – Plus Enable can be assigned to a drive input. Current - Plus Enable can be monitored and controlled by READ_SV/WRITE_SV variable 84. If not assigned to an input, it defaults to ON.

Current Command (Digital)

Category:	System/Command/Tuning
Method of Selection:	Read-Only
Units:	Amps
Defaults:	n/a
Range:	-Current Limit to +Current Limit

Description:
Monitor the command to the current loop. When the Active Operating Mode is Position Velocity or Velocity Control this is the output of the Velocity Loop. When using the Control Panel Current control mode this is the user specified current. The Current Command can be monitored using READ_SV variable 73.

Custom Motor (Digital)

Category:	System/Motor
Method of Selection:	Read-Only
Units:	n/a
Defaults:	n/a
Range:	Standard, Custom

Description:

Display whether the selected motor is Standard or Custom. Standard motor table data is supplied by the factory and cannot be modified. Custom motor data allows control of third-party motors and can be defined by the user.

DC Power Input (Digital)

Category:	Limits and Indicators
Method of Selection:	Dropdown List
Units:	n/a
Defaults:	Off
Range:	Off, On

Description:

Enables DC power input feature. Disables AC power check. Contact technical support for more information.

Decel Limit (Digital)

Category:	Command
Method of Selection:	Edit
Units:	RPM/Sec
Defaults:	20000
Range:	16 to 1048560

Description:

Enter the limit of the rate-of-change of the Velocity Command when it is decreasing and the AccDec Limits are Active.

Digitized Actual (Digital)

Category:	Basic/System
Method of Selection:	Read-Only
Units:	FU
Defaults:	n/a
Range:	n/a

Description:

Monitor the position of the Digitized feedback device. Unless otherwise specified in Feedback and Scaling, the Digitized Feedback device is the F2 feedback. Digitized Actual can be monitored using READ_SV variable 1 for the Digitized axis associated with this drive.

Digitized Feedback Assignment (Digital)

Category:	Feedback and Scaling
Method of Selection:	Feedback Assignment Dialog
Units:	n/a
Defaults:	F2
Range:	F1, F2

Description:

Identifies the feedback to be used for digitized feedback.

Refer to: *Feedback Assignment Dialog*

Digitized FU/Load Rev (Digital)

Category:	Feedback and Scaling
Method of Selection:	Edit
Units:	n/a
Defaults:	Number of counts per motor rev
Range:	n/a

Description:

Enter the number of desired feedback units for each rev of the load. Double-click to edit in dialog box.

Digitized Latched (Digital)

Category:	System
Method of Selection:	Read-Only
Units:	FU
Defaults:	n/a
Range:	n/a

Description:

Monitor the latched position of the Digitized feedback device. Unless otherwise specified in Feedback and Scaling the Digitized Feedback device is the F2 feedback. To enable latching of F2 the Capture F2 Position checkbox for Drive Input 8 must be checked. Position Latched is available via READ_SV variable 22 for the Digitized axis associated with this drive.

Digitized Load Rev (Digital)

Category:	Feedback and Scaling
Method of Selection:	Edit
Units:	n/a
Defaults:	1
Range:	n/a

Description:

Used in conjunction with Digitized Motor Rev. Enter the number of revs the load turns for a given machine distance. In Digitized Motor Rev, enter the number of revs the Motor turns for the same machine distance. Double-click to edit in dialog box.

Digitized Motor Rev (Digital)

Category:	Feedback and Scaling
Method of Selection:	Edit
Units:	n/a
Defaults:	1
Range:	n/a

Description:

Used in conjunction with Digitized Load Rev. Enter the number of revs the motor turns for a given machine distance. In Digitized Load Rev, enter the number of revs the Load turns for the same machine distance. Double click to edit in dialog box.

Drive Ambient Temp. Fault Limit (Digital)

Category:	Limits & Indicators
Method of Selection:	Read-Only
Units:	Degrees C
Defaults:	n/a
Range:	n/a

Description:

If the Drive Ambient Temperature exceeds this limit, the drive will fault. This protects the drive’s power section from damage due to overheating. This occurs when the drive temperature rises due to excessive ambient temperature.

Drive Ambient Temp. Warning Limit (Digital)

Category:	Limits & Indicators
Method of Selection:	Read-Only
Units:	Degrees C
Defaults:	n/a
Range:	n/a

Description:

If the Drive Ambient Temperature exceeds this limit, the warning indicator will be set. The warning indicator can be assigned to a drive output to provide an early warning before a fault limit is exceeded. This warning only applies to MMC-SD-x-460 systems.

Drive Bus Charged (Digital)

Category:	System
Method of Selection:	Read-Only
Units:	n/a
Defaults:	n/a
Range:	n/a

Description:

Monitor the status of Drive Bus Charged indicator. Drive Bus Charged is ON when the Drive Bus Voltage level is sufficient for motor control. Drive Bus Charged can be assigned to a drive output.

READ_SV variable 67 AND (16#00000040) indicates the state of Drive Bus Charged.

Drive Bus Over Voltage Fault Limit (Digital)

Category:	Limits & Indicators
Method of Selection:	Read-Only
Units:	Volts
Defaults:	n/a
Range:	n/a

Description:

If the Drive Bus Voltage exceeds this limit, the drive will fault. This protects the drive’s power section from voltage levels that would damage it. If a motor is rated for 220/230 volts and is used with a 440/460 volt drive, this limit will protect the motor from voltage levels that would damage it. Typical causes of this fault include line voltage that is too high or back EMF generated by the motor during deceleration requiring an external shunt system to dissipate the energy.

Drive Bus Voltage (Digital)

Category:	System
Method of Selection:	Read-Only
Units:	Volts
Defaults:	n/a
Range:	0 to 1000

Description:

Monitor the voltage of the Drive Bus.

Drive Cold Restart (Digital)

Category:	System
Method of Selection:	Read-Only
Units:	n/a
Defaults:	n/a
Range:	n/a

Description:

Monitor the status of the Drive Cold Restart function. An OFF-to-ON transition of the Drive Cold Restart initiates the equivalent of a drive control power-on sequence. The Drive Cold Restart function can be assigned to a drive input but is ignored while the application ladder is scanning. Select the PiCPro Drive Cold Restart command to initiate a Cold Restart in the connected ONLINE drive. When working ONLINE the application ladder scan must be stopped to perform a Cold Restart.

Drive Communication Errors (Digital)

Category:	Basic/System
Method of Selection:	Read-Only
Units:	n/a
Defaults:	n/a
Range:	n/a

Description:

Monitor the number of Digital Link errors detected by the drive. Two consecutive errors will cause a Digital Link Error fault shutdown. Drive Communication Errors is reset whenever cyclic data is started.

Drive Enable (Digital)

Category:	Basic/System
Method of Selection:	Read-Only
Units:	n/a
Defaults:	n/a
Range:	n/a

Description:

Monitor the status of Drive Enable. ON when the drive input that Hardware Enable Input is assigned to is on. Always ON when Hardware Enable Input is not assigned to any drive input. OFF when the drive input that Hardware Enable Input is assigned to is off.

Drive Status will indicate Drive Enabled when Drive Enable is ON, CLOSLOOP is ON and Fault is OFF. Drive Enabled indicates that the drive power section is on and the drive control loops are closed. READ_SV variable 67 AND (16#01000000) indicates the state of Drive Enable.

Drive Heatsink Temp. (Digital)

Category:	System
Method of Selection:	Read-Only
Units:	Degrees C
Defaults:	n/a
Range:	0 to 180

Description:

Monitor the temperature of the Drive Heatsink.

Drive Heatsink Temp. Fault Limit (Digital)

Category:	Limits & Indicators
Method of Selection:	Read-Only
Units:	Degrees C
Defaults:	n/a
Range:	n/a

Description:

If the Drive Heatsink Temperature exceeds this limit, the drive will fault. This protects the drive's power section from damage due to overheating. This occurs when the heatsink temperature rises due to excessive power output, excessive ambient temperature or inadequate airflow.

Drive Heatsink Temp. Warning Limit (Digital)

Category:	Limits & Indicators
Method of Selection:	Read-Only
Units:	Degrees C
Defaults:	n/a
Range:	n/a

Description:

If the Drive Heatsink Temperature exceeds this limit, the warning indicator will be set. The warning indicator can be assigned to a drive output to provide an early warning before a fault limit is exceeded.

Drive Heatsink Temp. Warning Limit (Digital)

Category:	Limits & Indicators
Method of Selection:	Read-Only
Units:	Degrees C
Defaults:	n/a
Range:	n/a

Description:

If the Drive Heatsink Temperature exceeds this limit, the warning indicator will be set. The warning indicator can be assigned to a drive output to provide an early warning before a fault limit is exceeded.

Drive Inputs (Digital)

Category:	Basic/System
Method of Selection:	Read-Only
Units:	n/a
Defaults:	n/a
Range:	n/a

Description:

Monitor the state of the drive inputs. An input number will be displayed when the input is ON. A "." will be displayed when the input is OFF. For example, if inputs 1, 4 and 8 are ON, "1..4…8" will be displayed.

The state of Drive Inputs with MMC Application Input checked is available via READ_SV variable 71.

Drive Model (Digital)

Category:	Basic/System
Method of Selection:	Dropdown List
Units:	n/a
Defaults:	n/a
Range:	Extensive, see dropdown list for available choices

Description:

Displays the drive model. The drive model is selected when the drive is being inserted into the drive list. Changes are made from the File column. Select the Drive Model parameter and pick the desired model from the dropdown list. The Drive Model parameter value in the actual column is read from the drive and must match the selected drive model to Download File Data to the drive. When the drive model is selected, all drive specific parameters are set to factory default values.

Drive Name (Digital)

Category:	Basic/System
Method of Selection:	Read-Only
Units:	n/a
Defaults:	n/a
Range:	n/a

Description:

Displays the Drive Name. The Drive Name is defined when a drive is inserted in the Drive List. To change the Drive Name, highlight the drive in the Drive List and select Edit Drive Name and Address. The Drive Name is a descriptive string and the File and Actual Drive Names are not required to match. All Digital Link communications use the drive Address Switch to identify the drive.

Drive Outputs (Digital)

Category:	Basic/System
Method of Selection:	Read-Only
Units:	n/a
Defaults:	n/a
Range:	n/a

Description:

Monitor the state of the drive outputs. An output number will be displayed when the output is ON. A "." will be displayed when the output is OFF. For example, if inputs 1, 3 and 5 are ON, "1.3.5" will be displayed.

Drive Outputs, with MMC Application Output checked, can be monitored and controlled using READ_SV/WRITE_SV variable 72.

Drive Over Current Fault Limit (Digital)

Category:	Limits & Indicators
Method of Selection:	Read-Only
Units:	Amps
Defaults:	n/a
Range:	n/a

Description:

If the Current – Feedback exceeds this limit, the drive will fault. This protects the drive's power section from damage due to excessive current output.

Drive Status (Digital)

Category:	Basic/System
Method of Selection:	Read-Only
Units:	n/a
Defaults:	n/a
Range:	n/a

Description:

Monitor the Drive Status indicating the state of the Drive Ready, Drive Enabled, Fault and Warning indicators.

Read_SV variable 67 AND (16#00000100) = Drive Ready, AND (16#00000080) = Drive Enabled, AND (16#00000400) = Drive Fault, AND (16#00000800) = Drive Warning.

Specific drive fault status is available using READ_SV variable 68. Specific drive warning status is available using READ_SV variable 69.

English/Metric Units (Digital)

Category:	Motor
Method of Selection:	Dropdown List
Units:	n/a
Defaults:	Metric
Range:	Metric, English

Description:

Select the display units for Motor – Kt, Inertia and Max Torque.

External Brake Release (Digital)

Category:	Motor
Method of Selection:	Read-Only
Units:	n/a
Defaults:	n/a
Range:	n/a

Description:

Monitor the status of the External Brake Release function. When the External Brake Release function is assigned to a drive input, the Release Brake output will follow the state of the External Brake Release input and the Brake Apply Time and the Brake Release time will be ignored. This allows an external control to coordinate operation of the brake.

External Brake Release Control (Digital)

Category:	Motor
Method of Selection:	Read-Only
Units:	n/a
Defaults:	n/a
Range:	n/a

Description:

ON when the External Brake Release is assigned to an input indicating that the Release Brake output will follow the External Brake Release input. When the External Brake Release function is assigned to an input, the Brake Apply Time and the Brake Release Time are ignored.

F1 Counter (Digital)

Category:	Feedback and Scaling/Motor
Method of Selection:	Read-Only
Units:	Counts
Defaults:	n/a
Range:	n/a

Description:

F1 feedback device’s position in counts (not scaled).

F1 Error (Digital)

Category:	Feedback and Scaling
Method of Selection:	Read-Only
Units:	n/a
Defaults:	n/a
Range:	n/a

Description:

Monitor the state of the F1 Error indicator. ON if there is an error with F1 feedback.

F1 Feedback - A (Digital)

Category:	Feedback and Scaling
Method of Selection:	Read-Only
Units:	n/a
Defaults:	n/a
Range:	n/a

Description:

ON when the A signal of F1 quadrature feedback is ON. This parameter only applies to firmware version 4.01 and higher.

F1 Feedback - B (Digital)

Category:	Feedback and Scaling
Method of Selection:	Read-Only
Units:	n/a
Defaults:	n/a
Range:	n/a

Description:

ON when the B signal of F1 quadrature feedback is ON. This parameter only applies to firmware version 4.01 and higher.

F1 Feedback - I (Digital)

Category:	Feedback and Scaling
Method of Selection:	Read-Only
Units:	n/a
Defaults:	n/a
Range:	n/a

Description:

ON when the Z signal of F1 quadrature feedback is ON. This parameter only applies to firmware version 4.01 and higher.

F1 Feedback Type (Digital)

Category:	Feedback and Scaling
Method of Selection:	Feedback Assignment Dialog
Units:	n/a
Defaults:	Motor Feedback Type
Range:	Resolver Incremental Encoder Stegmann Singleturn Encoder Stegmann Multiturn Encoder ENDAT Singleturn Encoder ENDAT Multiturn Encoder BiSS Singleturn Encoder BiSS Multiturn Encoder

Description:

Identifies the feedback type of F1.

F1 Latched Count (Digital)

Category:	Feedback and Scaling
Method of Selection:	Read-Only
Units:	n/a
Defaults:	n/a
Range:	n/a

Description:

F1 feedback device's position as latched by Capture F1 Position in counts (not scaled). To enable latching of F1, the Capture F1 Position checkbox for Drive Input 7 must be checked.

F1 Lines Per Rev (Digital)

Category:	Feedback and Scaling
Method of Selection:	Feedback Assignment Dialog
Units:	Lines
Defaults:	Motor lines per rev
Range:	n/a

Description:

Identifies the number of feedback device lines per motor revolution of F1.

F1 Position Polarity (Digital)

Category:	Feedback and Scaling
Method of Selection:	Dropdown List
Units:	n/a
Defaults:	Forward = CW
Range:	Forward = CW, Forward = CCW

Description:

Selecting Forward=CW will result in the F1 Counting up when F1 rotates in a clockwise direction looking at F1's shaft. Selecting Forward=CCW will result in the F1 Counting up when F1 rotates in a counterclockwise direction looking at F1's shaft.

F2 Counter (Digital)

Category:	Feedback and Scaling
Method of Selection:	Read-Only
Units:	Counts
Defaults:	n/a
Range:	n/a

Description:

F2 feedback device's position in counts (not scaled).

F2 Enable Pulse Counter (Digital)

Category:	Feedback and Scaling
Method of Selection:	Dropdown List
Units:	n/a
Defaults:	Off
Range:	Off, On

Description:

Tells the drive to treat F2 as a pulse counter. This is only valid if feedback configuration is 1. (In feedback configuration 1, the feedback assignment for motor commutation, motor velocity, and motor position is F1 and the digitized feedback assignment is F2.)

If *F2 Enable Pulse Counter* is On

- F2 Feedback - A represents pulse count.
- F2 Feedback - B represents direction where zero means count up, non-zero means count down.

F2 Error (Digital)

Category:	Feedback and Scaling
Method of Selection:	Read-Only
Units:	n/a
Defaults:	n/a
Range:	n/a

Description:

Monitor the state of the F2 Error indicator. ON if there is an error with F2 feedback.

F2 Feedback - A (Digital)

Category:	Feedback and Scaling
Method of Selection:	Read-Only
Units:	n/a
Defaults:	n/a
Range:	n/a

Description:

ON when the A signal of F2 quadrature feedback is ON. This parameter only applies to firmware version 4.01 and higher.

F2 Feedback - B (Digital)

Category:	Feedback and Scaling
Method of Selection:	Read-Only
Units:	n/a
Defaults:	n/a
Range:	n/a

Description:

ON when the B signal of F2 quadrature feedback is ON. This parameter only applies to firmware version 4.01 and higher.

F2 Feedback - I (Digital)

Category:	Feedback and Scaling
Method of Selection:	Read-Only
Units:	n/a
Defaults:	n/a
Range:	n/a

Description:

ON when the Z signal of F2 quadrature feedback is ON. This parameter only applies to firmware version 4.01 and higher.

F2 Latched Count (Digital)

Category:	Feedback and Scaling
Method of Selection:	Read-Only
Units:	n/a
Defaults:	n/a
Range:	n/a

Description:

F2 feedback device's position as latched by Capture F2 Position in counts (not scaled). To enable latching of F2, the Capture F2 Position checkbox for Drive Input 8 must be checked.

F2 Lines Per Rev (Digital)

Category:	Feedback and Scaling
Method of Selection:	Feedback Assignment Dialog
Units:	Lines
Defaults:	Motor lines per rev
Range:	n/a

Description:

Identifies the number of feedback device lines per motor revolution of F2.

F2 Position Polarity (Digital)

Category:	Feedback and Scaling
Method of Selection:	Dropdown List
Units:	n/a
Defaults:	Forward = CW
Range:	Forward = CW, Forward = CCW

Description:

Selecting Forward=CW will result in the F2 Counting up when F2 rotates in a clockwise direction looking at F2's shaft. Selecting Forward=CCW will result in the F2 Counting up when F2 rotates in a counterclockwise direction looking at F2's shaft.

Fault Reset (Digital)

Category:	System
Method of Selection:	Read-Only
Units:	n/a
Defaults:	n/a
Range:	n/a

Description:

Monitor the status of the Fault Reset function. Reset Drive Faults by calling the DRSETFLT function for the servo axis associated with this drive by pulsing the Fault Reset drive input ON or by selecting the PiCPro Reset Faults command to reset faults in the connected drive.

Feedback - S1 (Digital)

Category:	Feedback and Scaling
Method of Selection:	Read-Only
Units:	n/a
Defaults:	n/a
Range:	n/a

Description:

ON when the Hall Effect alignment signal S1 is ON.

Feedback - S2 (Digital)

Category:	Feedback and Scaling
Method of Selection:	Read-Only
Units:	n/a
Defaults:	n/a
Range:	n/a

Description:

ON when the Hall Effect alignment signal S2 is ON.

Feedback - S3 (Digital)

Category:	Feedback and Scaling
Method of Selection:	Read-Only
Units:	n/a
Defaults:	n/a
Range:	n/a

Description:

ON when the Hall Effect alignment signal S3 is ON.

Feedback Type (Digital)

Category:	Motor
Method of Selection:	Read-Only
Units:	n/a
Defaults:	n/a
Range:	Resolver Incremental Encoder Stegmann Singleturn Encoder Stegmann Multiturn Encoder ENDAT Singleturn Encoder ENDAT Multiturn Encoder BiSS Singleturn Encoder BiSS Multiturn Encoder

Description:

Presents the type of feedback device provided by the selected motor.

Firmware Date/Time Stamp (Digital)

Category:	Basic/System
Method of Selection:	Read-Only
Units:	n/a
Defaults:	Date/Time Stamp specified in personality file
Range:	n/a

Description:

Provides the date and time of the firmware build that constitutes the firmware revision selected in parameter **Firmware Revision**.

Format: mm/dd/yyyy hh:mm AM/PM

Firmware Revision (Digital)

Category:	Basic/System
Method of Selection:	Dropdown List
Units:	Major.Minor
Defaults:	Revision specified in personality file
Range:	1.00 to 99.99

Description:

Select a new version of firmware when the drive firmware is being changed. PiCPro will default to the latest version of firmware when inserting new drives.

Hall Offset (Digital)

Category:	Motor
Method of Selection:	Read-Only
Units:	Degrees
Defaults:	n/a
Range:	n/a

Description:

The angular offset between the Hall Effect sensors and the motor windings.

Hardware Revision (Digital)

Category:	System
Method of Selection:	Read-Only
Units:	Max-min
Defaults:	n/a
Range:	01.00 to 99.99

Description:

Display drive hardware revision level.

Index Offset (Digital)

Category:	Motor
Method of Selection:	Read-Only
Units:	Degrees
Defaults:	n/a
Range:	n/a

Description:

The angular offset between the index mark and the motor windings.

In Speed Window (Digital)

Category:	Limits & Indicators
Method of Selection:	Read-Only
Units:	n/a
Defaults:	n/a
Range:	n/a

Description:

Monitor the In Speed Window indicator. When the absolute value of the Motor Velocity is greater than or equal to the Speed Window – Lower Limit and less than or equal to the Speed Window – Upper Limit, the In Speed Window indicator will be ON. The indicator can be assigned to a drive output. In Speed Window can be monitored using READ_SV variable 67 AND (16#4).

Inductance (Digital)

Category:	Motor
Method of Selection:	Read-Only
Units:	milliHenry (mH)
Defaults:	n/a
Range:	n/a

Description:

Phase-to-phase motor inductance.

Inertia (Digital)

Category:	Motor
Method of Selection:	Read-Only
Units:	Metric: kg-cm^2, English: in-lbs^2
Defaults:	n/a
Range:	n/a

Description:

The inertia of the motor rotor.

Inertia Ratio (Digital)

Category:	Tuning
Method of Selection:	Edit
Units:	Rate of Application Inertia to Motor Inertia
Defaults:	1
Range:	0.00 to 500.0

Description:

Enter the ratio of the application inertia to the motor inertia. Auto-tune can be used to calculate the inertia ratio. Setting the inertia ratio to zero turns off Acceleration Feedforward in the Velocity Loop. This can be helpful in difficult to tune applications.

Input (Digital)

Category:	I/O
Method of Selection:	Read-Only
Units:	n/a
Defaults:	n/a
Range:	n/a

Description:

This parameter displays the resultant state of the inputs assigned to it.

Input Assignment (Digital)

Category:	I/O
Method of Selection:	Dropdown Check List
Units:	n/a
Defaults:	Input 1: Hardware Enable Input Input 2: Fault Reset
Range:	See List Below

Description:

Use Input Assignments to assign drive functions to drive inputs. When a drive input is ON, all functions assigned to it will be ON. Multiple functions can be assigned to a single input and a function can be assigned to multiple inputs. If a function is assigned to multiple inputs and any of the inputs are ON, the function will be ON.

Input Function Assignments	Monitors in these Categories
Hardware Enable Input	I/O
Fault Reset	System
Current – Plus Enable	Tuning
Current – Minus Enable	Tuning
External Brake Release	Motor
Velocity Loop Integrator Inhibit	Tuning
Velocity Loop Integrator Hold	Tuning
Drive Cold Restart	System
Overtravel Plus Fault	I/O
Overtravel Minus Fault	I/O
MMC Application Input	I/O

Hardware Enable Input - ON indicates that the drive power output be enabled when CLOSLOOP is executed. Drive power will be enabled while Drive Enable is ON and CLOSLOOP has been executed unless an overriding condition such as Fault is ON. Drive Enable OFF disables the drive power section. Drive Enable OFF will immediately disable drive power unless the Brake Apply Time is set to a negative value to delay power section shutdown. Drive Enable is always ON if it is not assigned to an input. The Hardware Enable Input state can be monitored using READ_SV variable 67 AND (16#0000 0080).

Fault Reset - Pulse ON to reset the Fault indication. Fault Reset should be OFF during normal operation. DRSETFLT can be used to achieve the equivalent function over the Digital Link instead of using hardware.

Current – Plus Enable - ON allows the drive to output positive current. OFF prevents the drive from outputting positive current. Current – Plus Enable defaults to ON if it is not assigned to an input. Current -Plus Enable can be monitored and controlled using READ_SV/WRITE_SV variable 84.

Current – Minus Enable - ON allows the drive to output negative current. OFF prevents the drive from outputting negative current. Current – Minus Enable defaults to ON if it is not assigned to an input. Current - Minus Enable can be monitored and controlled using READ_SV/WRITE_SV variable 85.

External Brake Release - Assign External Brake Release to an input to use an external system to control the state of the Release Brake output. When the External Brake Release input is ON, the Release Brake output is turned on to close the relay and release the brake. When the External Brake Release input is OFF, the Release Brake output is turned off to open the relay and engage the brake. The Brake Apply Time and Brake Release Time are ignored when External Brake Release is assigned to an input.

Velocity Loop Integrator Inhibit - When ON, I Gain is zero and the integrator is zeroed. When OFF, I Gain is the velocity loop I Gain and the integrator functions normally. Velocity Loop Integrator Inhibit can be monitored and controlled by READ_SV/WRITE_SV variable 82.

Velocity Loop Integrator Hold - When ON, the integrator value is held at its present value, When OFF, the integrator functions normally. Velocity Loop Integrator Hold can be monitored and controlled by READ_SV/WRITE_SV variable 83.

Drive Cold Restart - An OFF-to-ON transition of Drive Cold Restart initiates the equivalent of a drive control power-on sequence when the control application program is stopped.

Overtravel Plus Fault - Used in linear motion applications to interface the plus hardware overtravel input. When the input is off and Prevent Drive Overtravel Plus Fault is off an Overtravel Plus Fault condition exists. To move off the plus hardware overtravel limit switch set Prevent Drive Overtravel Plus Fault using WRITE_SV variable 86 to temporarily override this fault. While the Overtravel Plus Fault input is set, motion is always prevented in the plus direction, regardless of the state of Prevent Drive Overtravel Plus Fault.

Overtravel Minus Fault - Used in linear motion applications to interface the minus hardware overtravel input. When the input is off and Prevent Drive Overtravel Minus Fault is off an Overtravel Minus Fault condition exists. To move off the minus hardware overtravel limit switch set Prevent Drive Overtravel Minus Fault using WRITE_SV variable 86 to temporarily override this fault. While the Overtravel Minus Fault input is set, motion is always prevented in the minus direction, regardless of the state of Prevent Drive Overtravel Minus Fault.

MMC Application Input - Assigning this function to a Drive Input allows it to be monitored using READ_SV variable 71.

Input Offset (Digital)

Category:	Command
Method of Selection:	Edit
Units:	Millivolts (mV)
Defaults:	0
Range:	-10000 to 10000

Description:

Enter an offset to be subtracted from the Analog Command. This is used to cancel the offset between the external control and the drive so a zero output from the control is seen as an Analog Command of zero. Select the PiCPro Remove Input Offset command to automatically calculate and enter the Input Offset for an ONLINE drive.

Input Polarity (Digital)

Category:	Command
Method of Selection:	Dropdown List
Units:	n/a
Defaults:	Forward
Range:	Forward, Reverse

Description:

Select the direction to move the motor in response to a positive input command. The motor will move in the direction that causes increasing feedback position when Forward is selected. The motor will move in the direction that causes decreasing feedback position when Reverse is selected.

Ke (Digital)

Category:	Motor
Method of Selection:	Read-Only
Units:	Volts / 1000 RPM
Defaults:	n/a
Range:	n/a

Description:

Motor Back EMF voltage parameter.

Kt (Digital)

Category:	Motor
Method of Selection:	Read-Only
Units:	Metric: n-m / Amp English: in-lb / Amp
Defaults:	n/a
Range:	n/a

Description:

Motor torque parameter.

Last/Previous Fault Message (Digital)

Category: Feedback and Scaling

Method of Selection: Read-Only

Units: n/a

Defaults: n/a

Range: n/a

Description:

These parameters are a record of the faults as they occur with the most recent on top.

Lines Per Rev (Digital)

Category: Motor

Method of Selection: Read-Only

Units: Lines

Defaults: n/a

Range: n/a

Description:

The number of feedback device lines per motor revolution.

When modifying a custom motor, this field can be edited with a range of 1 to 10000 lines.
For a feedback type of BiSS, use one of the following choices for Lines per Rev:

Bit Resolution	Lines per Rev
19	512
22	4096

Low Pass Filter Bandwidth (Digital)

Category: Tuning

Method of Selection: Edit

Units: Hz

Defaults: 150

Range: 1 to 32767

Description:

Enter the bandwidth of the Low Pass Filter. The lower the bandwidth, the greater the filter. This is used to eliminate noise due to mechanical resonance or feedback jitter.

Low Pass Filter Enable (Digital)

Category:	Tuning
Method of Selection:	Dropdown List
Units:	n/a
Defaults:	Active
Range:	Active, Not Active

Description:

Select Active to enable the Low Pass Filter.

Magnetic Offset (Digital)

Category:	Motor
Method of Selection:	Read-Only
Units:	Degrees
Defaults:	n/a
Range:	n/a

Description:

When using resolvers, it is the signed value in feedback counts of the distance between the resolver null and Magnetic zero.

Max Speed (Digital)

Category:	Motor
Method of Selection:	Read-Only
Units:	RPM
Defaults:	n/a
Range:	n/a

Description:

The maximum motor operating speed.

Maximum Current (Digital)

Category:	Motor
Method of Selection:	Read-Only
Units:	Amps
Defaults:	n/a
Range:	n/a

Description:

The peak current the motor can accept.

Maximum Motor Temp (Digital)

Category:	Motor
Method of Selection:	Read-Only
Units:	Degrees C
Defaults:	n/a
Range:	n/a

Description:

The maximum allowable motor winding temperature.

Minus Current Limit (Digital)

Category:	Basic/Tuning
Method of Selection:	Edit
Units:	Amps
Defaults:	Value of Maximum Current (lesser of Motor Max Current and Drive Max Current)
Range:	0.01 to Maximum Current (lesser of Motor Max Current and Drive Max Current)

Description:

Enter a negative output current limit to protect the application. If the drive and motor system can deliver more torque than the application can accept, use this limit to protect it. Minus Current Limit can be monitored and controlled by READ_SV/WRITE_SV variable 76.

MMC Close Loop Request (Digital)

Category: System

Method of Selection: Read-Only

Units: n/a

Defaults: n/a

Range: n/a

Description:

MMC Close Loop Request indicates when the application program is requesting to close or open the drive control loop. This indicator is ON when the control has requested to close the loop by executing the CLOSLOOP function or by executing the E_RESET function when the loops had been closed prior to the E-stop condition. This indicator is OFF when the control has requested to open the loop by executing the OPENLOOP function or when the control detects an E-stop condition.

Drive Status will indicate Drive Enabled when Drive Enable is ON, MMC Close Loop Request is ON, and Fault is OFF.

Drive Enabled indicates that the drive power section is on and the drive control loops are closed.

MMC Rollover Position (Digital)

Category: Feedback and Scaling

Method of Selection: Read-Only

Units: FU

Defaults: n/a

Range: n/a

Description:

This is the value the user was prompted to enter when performing an absolute reference. This entered value should be the same as the rollover position currently used in the MMC for the axis associated with F1. The drive used this value to calculate the absolute feedback's power-on offset" as the machine rotates through multiple rollover distances.

Motor Calculated Temp. (Digital)

Category: Motor

Method of Selection: Read-Only

Units: Degrees C

Defaults: n/a

Range: n/a

Description:

Monitor the calculated motor temperature. The Motor Calculated Temperature is the anticipated motor temperature based on the amount of current being delivered over time. When the selected motor does not provide a thermistor or a thermal switch, the calculated motor temperature can be monitored using READ_SV variable 77.

Motor Calculated Temp. Fault Limit (Digital)

Category:	Limits & Indicators
Method of Selection:	Read-Only
Units:	Degrees C
Defaults:	n/a
Range:	n/a

Description:

If the Motor Calculated Temperature exceeds this limit, the drive will fault. This protects the motor from damage due to excessive heat. The Motor Calculated Temperature is the anticipated motor temperature based on the amount of current being delivered over time.

Motor Calculated Temp. Warning Limit (Digital)

Category:	Limits & Indicators
Method of Selection:	Edit
Units:	Degrees C
Defaults:	110
Range:	0 to Maximum Motor Temp

Description:

If the Motor Calculated Temperature exceeds this limit, the warning indicator will be set. The Motor Calculated Temperature is the anticipated motor temperature based on the amount of current being delivered over time. The warning indicator can be assigned to a drive output to provide an early warning before a fault limit is exceeded.

Motor Commutation Feedback Assignment (Digital)

Category:	Feedback and Scaling
Method of Selection:	Feedback Assignment Dialog
Units:	n/a
Defaults:	F1
Range:	F1, F2

Description:

Identifies the feedback to be used for motor commutation.

Refer to: *Feedback Assignment Dialog*

Motor Data Status (Digital)

Category:	System/Motor
Method of Selection:	Read-Only
Units:	n/a
Defaults:	n/a
Range:	n/a

Description:

This parameter indicates the current status of all of the motor data. There are three states that the motor data can be:

- **Same** – The current motor data is the same as the information in the motor database file on the workstation.
- **Different** – The current motor data is different than the information in the motor database file on the workstation.
- **Not Found** – The current motor data was not found in the motor database file on the workstation.

Motor Feedback - A (Digital)

Category:	Motor
Method of Selection:	Read-Only
Units:	n/a
Defaults:	n/a
Range:	n/a

Description:

ON when the A signal of quadrature feedback is ON.

Motor Feedback - B (Digital)

Category:	Motor
Method of Selection:	Read-Only
Units:	n/a
Defaults:	n/a
Range:	n/a

Description:

ON when the B signal of quadrature feedback is ON.

Motor Feedback - I (Digital)

Category:	Motor
Method of Selection:	Read-Only
Units:	n/a
Defaults:	n/a
Range:	n/a

Description:

ON when the Z signal of quadrature feedback is ON.

Motor Feedback - S1 (Digital)

Category:	Motor
Method of Selection:	Read-Only
Units:	n/a
Defaults:	n/a
Range:	n/a

Description:

ON when the Hall Effect alignment signal S1 is ON.

Motor Feedback - S2 (Digital)

Category:	Motor
Method of Selection:	Read-Only
Units:	n/a
Defaults:	n/a
Range:	n/a

Description:

ON when the Hall Effect alignment signal S2 is ON.

Motor Feedback - S3 (Digital)

Category:	Motor
Method of Selection:	Read-Only
Units:	n/a
Defaults:	n/a
Range:	n/a

Description:

ON when the Hall Effect alignment signal S3 is ON.

Motor Model (Digital)

Category:	Basic/System/Motor
Method of Selection:	Dropdown List
Units:	n/a
Defaults:	n/a
Range:	Extensive, see dropdown list for available choices

Description:

Select the Model of the motor being used. The motor model is selected when the drive is being inserted into the drive list. When working ONLINE, Drive Enable must be OFF and the application ladder scan must be stopped to select a motor. When a motor model is selected, all motor specific parameters are set to factory default values.

Motor Position (Digital)

Category:	Basic/System/Motor
Method of Selection:	Read-Only
Units:	Counts
Defaults:	n/a
Range:	-2,147,483,648 to 2,147,483,647

Description:

Monitor the accumulated motor feedback.

Motor Position Feedback Assignment (Digital)

Category:	Feedback and Scaling
Method of Selection:	Feedback Assignment Dialog
Units:	n/a
Defaults:	F1
Range:	F1, F2

Description:

Identifies the feedback to be used for position loop.

Refer to: *Feedback Assignment Dialog*

Motor Temp (Digital)

Category:	System
Method of Selection:	Read-Only
Units:	See below
Defaults:	n/a
Range:	See below

Description:

Monitor the Motor Temperature. The Motor Temperature is available via READ_SV variable 77. READ_SV variable 77 will return the actual temperature if the motor has a thermistor, 0 or 1 if the motor has a thermal switch, or the calculated temperature if the motor has neither.

Motor – Thermal Device	Motor Temp. display	Range
none	Motor Calculated Temp.	0 – 255 Degrees C
Thermal Switch	Closed – OK, Open – motor is overheated	
Thermistor	Measured Motor Temperature	0 – 255 Degrees C

Motor Temp. Fault Limit (Digital)

Category:	Limits & Indicators
Method of Selection:	Read-Only
Units:	Degrees C
Defaults:	n/a
Range:	n/a

Description:

If the Motor Thermistor Temperature exceeds this limit, the drive will fault. This protects the motor from damage due to excessive heat. This typically occurs when the motor is undersized for the application and excessive output current causes the motor temperature to rise. This may also occur when the motor heatsink is inadequate or the ambient temperature around the motor is too high. This only applies when the Motor – Thermal Device is a thermistor.

Motor Temp. Warning Limit (Digital)

Category:	Limits & Indicators
Method of Selection:	Edit
Units:	Degrees C
Defaults:	Maximum Motor Temp
Range:	0 to Maximum Motor Temp

Description:

If the Motor Thermistor Temperature exceeds this limit, the warning indicator will be set. The warning indicator can be assigned to a drive output to provide an early warning before a fault limit is exceeded. This only applies when the Motor – Thermal Device is a thermistor.

Motor Thermal Switch (Digital)

Category: Motor

Method of Selection: Read-Only

Units: n/a

Defaults: n/a

Range: Open, Closed

Description:

Monitor the state of the motor thermal switch when the Motor – Thermal Device is set to Thermal Switch. Open when the motor has overheated, closed when the motor is not overheated.

Motor Thermistor Temp. (Digital)

Category: Motor

Method of Selection: Read-Only

Units: Degrees C

Defaults: n/a

Range: n/a

Description:

Monitor the temperature of the motor thermistor when the Motor – Thermal Device is set to Thermistor. Motor Temperature can be monitored using READ_SV variable 77.

Motor Velocity (Digital)

Category: Basic/System/Tuning/Motor

Method of Selection: Read-Only

Units: RPM

Defaults: n/a

Range: n/a

Description:

Monitor the actual motor velocity.

Motor Velocity Feedback Assignment (Digital)

Category:	Feedback and Scaling
Method of Selection:	Feedback Assignment Dialog
Units:	n/a
Defaults:	F1
Range:	F1, F2

Description:

Identifies the feedback to be used for velocity loop.

Refer to: *Feedback Assignment Dialog*

Notch Filter Bandwidth (Digital)

Category:	Advanced Tuning
Method of Selection:	Edit
Units:	Hz
Defaults:	20
Range:	2 to 20

Description:

Enter the bandwidth of the Notch Filter. The bandwidth of the Notch Filter is the approximate 3db width of the Notch filter in hertz. This is used in conjunction with the Notch Filter Center Frequency to eliminate mechanical resonance. Note, the center frequency minus ½ the bandwidth may not be below 6hz.

Notch Filter Center Frequency (Digital)

Category:	Advanced Tuning
Method of Selection:	Edit
Units:	Hz
Defaults:	600
Range:	7 to 600

Description:

Enter the Center Frequency of the Notch Filter. The center frequency of the Notch Filter is the frequency in hertz that the 3db bandwidth is centered about. This is used in conjunction with the Notch Filter Bandwidth to eliminate mechanical resonance. Note, the center frequency minus ½ the bandwidth may not be below 6hz.

Notch Filter Enable (Digital)

Category:	Advanced Tuning
Method of Selection:	Dropdown List
Units:	n/a
Defaults:	Not Active
Range:	Active, Not Active

Description:

Select Active to enable the Notch Filter. The Notch Filter and the 2-Pole filter may not be enabled at the same time.

Observer Bandwidth (Digital)

Category:	Advanced Tuning
Method of Selection:	Edit
Units:	Hz
Defaults:	50
Range:	50 to 350

Description:

Enter the cut off Frequency response of the Observer. The frequency selected will determine the observer PID gains. A higher frequency corresponds to a more responsive observer PID.

Observer Enable (Digital)

Category:	Advanced Tuning
Method of Selection:	Dropdown List
Units:	n/a
Defaults:	Not Active
Range:	Active, Not Active

Description:

Select Active to enable the Observer. Enabling the observer will cause the velocity loop IPD to use the observer output rather than the actual velocity derived from the feedback position.

Observer Feedforward (Digital)

Category:	Advanced Tuning
Method of Selection:	Edit
Units:	Ratio
Defaults:	1.0
Range:	0.0 to 5.0 (by 0.1)

Description:

Enter the acceleration feedforward for the Observer PID. The Observer Feedforward term is multiplied by the actual current and summed with the PID acceleration. A higher Observer Feedforward will cause the observer velocity to be more responsive to actual current.

Operation Time Elapsed (Digital)

Category:	System/Fault History
Method of Selection:	Read-Only
Units:	d(days) h(hours) m(minutes) s(seconds)
Defaults:	n/a
Range:	0 to 49710d 6h 28m 16s

Description:

Monitor the time that the drive has been powered.

Output 1-4 (Digital)

Category:	I/O
Method of Selection:	Read-Only
Units:	n/a
Defaults:	n/a
Range:	n/a

Description:

This parameter displays the resultant state of the outputs assigned to it.

Output Assignment (Digital)

Category:	I/O
Method of Selection:	Dropdown Check List
Units:	n/a
Defaults:	Output 1-3: MMC Application Output Output 4: Drive Ready Output 5: Brake Release
Range:	n/a

Description:

Use Output Assignments to assign drive status indicators to drive outputs. When a status indicator is ON, the output it is assigned to will be ON. A status indicator can be assigned to multiple outputs and multiple status indicators can be assigned to a single output. When multiple status indicators are assigned to a single output, the output will be ON, when any of the status indicators are ON.

Output Status Assignments	Monitors in these Categories
Drive Ready	System (Drive Status)
Drive Enabled	System (Drive Status)
At Plus Current Limit	Tuning
At Minus Current Limit	Tuning
Fault	System (Drive Status)
Warning	System (Drive Status)
Drive Bus Charged	System
Drive Ready and Bus Charged	----------
Release Brake	Motor
At Zero Speed	Limits & Indicators
In Speed Window	Limits & Indicators
Up to Speed	Limits & Indicators
Startup Commutation Complete	Motor
220V Shunt on 440V Drive	Limits & Indicators
MMC Application Output	I/O

Drive Ready – ON indicates the drive power section will be enabled and loops will be closed when Drive Enable is ON and CLOSLOOP is executed. Drive Ready is ON when Fault is OFF and Cold Restart Required is OFF. Drive Ready is OFF when Fault is ON or Cold Restart Required is ON. Drive Ready can be monitored using READ_SV variable 67 AND (16#100).

Drive Enabled – ON when Drive Enable is ON and the drive power section is enabled and loops closed. OFF when the drive power section is disabled and loops are open. See parameter Drive Status for a detailed description. Drive Enabled can be monitored using READ_SV variable 67 AND (16#80).

At Plus Current Limit – At Plus Current Limit – ON when the Current Command equals the Plus Current Limit. OFF when the Current command is less than the Plus Current Limit. At Plus Current Limit can be monitored using READ_SV variable 67 AND (16#10).

At Minus Current Limit – ON when the Current Command equals the Minus Current Limit. OFF when the Current command is greater than the Minus Current Limit. At Minus Current Limit can be monitored using READ_SV variable 67 AND (16#20).

Fault – ON when a fault condition has been detected. OFF when no faults are present. Select the PiCPro Reset Faults command or pulse the Reset Faults input or execute DRSETFLT to reset the Fault indicator. Fault can be monitored using READ_SV variable 67 AND (16#400).

Warning – ON when any warning limit is being exceeded. OFF when no warning limits are being exceeded. Warning can be monitored using READ_SV variable 67 AND (16#800).

Drive Bus Charged – ON when the Drive Bus Voltage is above the minimum level required to safely operate a motor. OFF when the Drive Bus Voltage is below the minimum level require to safely operate a motor. Drive Bus Charged can be monitored using READ_SV variable 67 AND (16#40).

Drive Ready and Bus Charged – ON when the both Drive Ready and Drive Bus Charged are ON. OFF when either Drive Ready or Drive Bus Charged is OFF. Drive Ready and Bus Charged can be monitored using READ_SV variable 67 AND (16#2000).

Release Brake – ON to close the brake relay and release the brake. OFF to open the brake relay and engage the brake. See Motor – Brake Apply Time and Motor – Brake Release Time parameters for a detailed description of Release Brake control. Release Brake can be monitored using READ_SV variable 67 AND (16#200).

At Zero Speed – ON when the absolute value of the Motor Velocity is less than or equal to the Zero Speed Rate. At Zero Speed can be monitored using READ_SV variable 67 AND (16#2).

In Speed Window – ON when the absolute value of the Motor Velocity is greater than or equal to the Speed Window – Lower Limit and less than or equal to the Speed Window – Upper Limit. In Speed Window can be monitored using READ_SV variable 67 AND (16#4).

Up to Speed – ON when the absolute value of the Motor Velocity is greater than or equal to the Up to Speed Rate. Up to Speed can be monitored using READ_SV variable 67 AND (16#8).

Startup Commutation Complete – ON when the drive has completed startup and the motor is being controlled normally. Startup Commutation Complete can be monitored using READ_SV variable 67 AND (16#1).

220V Shunt on 440V Drive – ON when bus voltage exceeds 415V while using a 220V motor. OFF when the bus voltage then goes below 400V. Note: This indicator can only be assigned to Output 3. If assigned to Output 3, ‘220V Shunt on 440V Drive’ must be the only indicator assigned. If other items are assigned, a message will result and the other items will be cleared. This output cannot be overridden if this indicator is assigned. If this indicator is assigned on a 460 volt motor or a 230 volt MMC Smart Drive, it will be ignored. Note also: The drive will disable PWM with an over voltage fault, when the motor is a 220 volt motor and if the bus voltage exceeds 440 volts (regardless of whether this indicator is enabled). 220V Shunt on 440V Drive can be monitored using READ_SV variable 67 AND (16#1000).

MMC Application Output – Assigning this to a Drive Output allows the output to be controlled by the ladder.

Over Current Fault Limit (Digital)

Category:	Limits & Indicators
Method of Selection:	Edit
Units:	Amps
Defaults:	Maximum Current (lesser of Motor Max Current and Drive Max Current)
Range:	0.01 to Maximum Current (lesser of Motor Max Current and Drive Max Current)

Description:

If the Current – Average exceeds this limit, the drive will fault. This is used to protect the application from excessive torque. If the drive and motor can deliver more torque than the application can accept, use this limit to protect it.

Over Speed Fault Limit (Digital)

Category:	Limits & Indicators
Method of Selection:	Edit
Units:	RPM
Defaults:	120% of Motor Maximum Speed
Range:	1 to 120% of Motor Maximum Speed

Description:

If the Motor Velocity exceeds this limit, the drive will fault. This typically occurs when an unexpected event causes system instability resulting in improper motor control. If the Motor – Maximum Speed is greater than that required by the application, this limit can be set to create a fault at a lower speed.

Override Mode (Digital)

Category:	System
Method of Selection:	Dropdown List
Units:	n/a
Defaults:	Velocity
Range:	Velocity, Torque

Description: Select the mode that will be the Active Operating Mode when Override Mode Select is ON.

Override Mode Select (Digital)

Category:	System
Method of Selection:	Read-Only
Units:	n/a
Defaults:	n/a
Range:	n/a

Description:

Monitor the Override Mode Select function. The Active Operating Mode equals Default Mode when Override Mode Select is OFF. The Active Operating Mode equals Override Mode when Override Mode Select is ON. Override Mode Select can be assigned to a drive input.

Plus Current Limit (Digital)

Category:	Basic/Tuning
Method of Selection:	Edit
Units:	Amps
Defaults:	Value of Maximum Current (lesser of Motor Max Current and Drive Max Current)
Range:	0.01 to Maximum Current (lesser of Motor Max Current and Drive Max Current)

Description:

Enter a positive output current limit to protect the application. If the drive and motor system can deliver more torque than the application can accept, use this limit to protect it. Plus Current Limit can be monitored and controlled by READ_SV/WRITE_SV variable 75.

Poles Per Rev (Digital)

Category:	Motor
Method of Selection:	Read-only
Units:	Poles
Defaults:	n/a
Range:	n/a

Description: Number of poles per motor revolution.

Position Actual (Digital)

Category:	Basic/System/Tuning
Method of Selection:	Read-only
Units:	FU (See Feedback and Scaling)
Defaults:	n/a
Range:	n/a

Description:

Monitor the actual position of the motor scaled in FU. Position Actual can be monitored using READ_SV variable 1.

Position Command (Digital)

Category:	Basic/System/Tuning
Method of Selection:	Read-Only
Units:	FU (See Feedback and Scaling)
Defaults:	n/a
Range:	n/a

Description:

Monitor the Position Command being received via the Digital Link. When the drive is operating in Position Control mode and enabled, the motor will be commanded to this position.

Position Command can be monitored using READ_SV variable 3.

Position Error (Digital)

Category:	Tuning
Method of Selection:	Read-Only
Units:	Drive Units
Defaults:	n/a
Range:	n/a

Description:

Monitor the difference between the Position Command and the Position Actual. The Position Error is the following error of the motor. Position Error can be monitored using READ_SV variable 4. If the Position Error exceeds the Servo Setup Following Error Limit then application program E Stop error (16#8002), Excess Error, will be set.

Position Feedforward Percent (Digital)

Category:	Tuning
Method of Selection:	Edit
Units:	%
Defaults:	0
Range:	0 to 100

Description:

Enter the Position Loop Feedforward Percent. Applications using linear and circular interpolation or master/slave position control, use Position Loop Feedforward to run at all velocities with zero following error. Position Loop Feedforward Percent can be monitored and controlled using READ_SV/WRITE_SV variable 79.

Position Latched (Digital)

Category:	Tuning
Method of Selection:	Read-Only
Units:	FU
Defaults:	n/a
Range:	n/a

Description:

Monitor the latched position of the motor. Unless otherwise specified in Feedback and Scaling, the position of the motor is the F1 feedback in FU. To enable latching of F1, the Capture F1 Position checkbox for Drive Input 7 must be checked. Position Latched is available via READ_SV variable 22.

Position Loop Feedforward Filter (Digital)

Category:	Tuning
Method of Selection:	Dropdown List
Units:	msec
Defaults:	0
Range:	0, 1, 10, 100

Description:

A position loop feedforward filter may be needed if the user chooses coarse position scaling values causing the axis to run rough when Position Loop Feedforward Percent is set to 100. The value only applies if Position Loop Feedforward Percent has a non zero value.

Position Loop FU/Load Rev (Digital)

Category:	Feedback and Scaling
Method of Selection:	Edit
Units:	n/a
Defaults:	Number of counts per motor rev
Range:	n/a

Description:

Enter the number of desired feedback units for each rev of the load. Double-click to edit in dialog box.

Position Loop D Gain (Digital)

Category:	Tuning
Method of Selection:	Edit
Units:	(FU / Min) / (1000 * FUFE / Min)
Defaults:	0
Range:	0 to 32767

Description:

Derivative gain determines the corrective action proportional to the magnitude of change for the following error. Enter the value that represents the number of drive units per minute for each axis unit of following error per minute * 1000. Typical Value = 0.

Position Loop I Gain (Digital)

Category:	Tuning
Method of Selection:	Edit
Units:	((FU / Min) * 1000) / (FUFE * Min)
Defaults:	0
Range:	0 to 32767

Description:

Enter the Position Loop Integral Gain. The Position Loop integrator is always active. The Position Loop integrator is used in point-to-point motion applications to drive a motor into position when a constant force offset needs to be overcome.

Position Loop I Minus Limit (Digital)

Category:	Tuning
Method of Selection:	Edit
Units:	FUFE * 1 msec
Defaults:	-10000
Range:	-32767 to 0

Description:

Enter the value of any additional integral error that the position loop should ignore when the axis is moving in a negative direction. Typical Value = 0.

Position Loop I Plus Limit (Digital)

Category:	Tuning
Method of Selection:	Edit
Units:	FUFE * 1 msec
Defaults:	10000
Range:	0 to 32767

Description:

Enter the value of any additional integral error that the position loop should ignore when the axis is moving in a positive direction. Typical Value = 0.

Position Loop Load Rev (Digital)

Category:	Feedback and Scaling
Method of Selection:	Edit
Units:	n/a
Defaults:	1
Range:	n/a

Description:

Used in conjunction with Position Loop Motor Rev. Enter the number of revs the load turns for a given machine distance. In Position Loop Motor Rev, enter the number of revs the Motor turns for the same machine distance. Double-click to edit in dialog box.

Position Loop Motor Rev (Digital)

Category:	Feedback and Scaling
Method of Selection:	Edit
Units:	n/a
Defaults:	1
Range:	n/a

Description:

Used in conjunction with Position Loop Load Rev. Enter the number of revs the motor turns for a given machine distance. In Position Loop Load Rev, enter the number of revs the Load turns for the same machine distance. Double click to edit in dialog box.

Position Loop P Gain (Digital)

Category:	Basic/Tuning
Method of Selection:	Edit
Units:	FU / Min / FUFE
Defaults:	2000
Range:	0 to 32767

Description:

Enter the Position Loop Proportional Gain. Increasing the P gain improves response time. Excessive P gain can cause instability. An extremely low P gain can cause poor servo response. Position Loop P Gain can be monitored and controlled using READ_SV/WRITE_SV variable 78.

Ramp Velocity to Zero on Controllable Fault (Digital)

Category:	Tuning
Method of Selection:	Dropdown List
Units:	n/a
Defaults:	On
Range:	On, Off

Description:

Set this parameter to On if the value of Decel Limit should be used to specify the rate-of-change of the Velocity Command when it is being ramped to zero.

Rated Voltage (Digital)

Category:	Motor
Method of Selection:	Read-Only
Units:	Volts
Defaults:	n/a
Range:	n/a

Description:

Voltage rating of the motor.

Release Brake (Digital)

Category:	Motor
Method of Selection:	Read-Only
Units:	n/a
Defaults:	n/a
Range:	n/a

Description:

Monitor the status of the Release Brake indicator. By default, Release Brake is assigned to drive output 5, a relay output. When Release Brake is ON, the relay is closed. When Release Brake is OFF, the relay is open. Release Brake can be monitored using READ_SV variable 67 AND (16#200).

Resistance (Digital)

Category:	Motor
Method of Selection:	Read-Only
Units:	Ohms
Defaults:	n/a
Range:	n/a

Description:

Phase-to-phase motor resistance.

Speed Window - Lower Limit (Digital)

Category:	Limits & Indicators
Method of Selection:	Edit
Units:	RPM
Defaults:	20
Range:	0 to 32767

Description:

Enter the lower limit for the In Speed Window indicator. When the absolute value of the Motor Velocity is greater than or equal to the Speed Window – Lower Limit and less than or equal to the Speed Window – Upper Limit, the In Speed Window indicator will be ON.

Speed Window - Upper Limit (Digital)

Category:	Limits & Indicators
Method of Selection:	Edit
Units:	RPM
Defaults:	200
Range:	0 to 32767

Description:

Enter the upper limit for the In Speed Window indicator. When the absolute value of the Motor Velocity is greater than or equal to the Speed Window – Lower Limit and less than or equal to the Speed Window – Upper Limit, the In Speed Window indicator will be ON.

Startup Commutation Complete (Digital)

Category:	Motor
Method of Selection:	Read-Only
Units:	n/a
Defaults:	n/a
Range:	n/a

Description:

ON after the drive has completed the motor startup commutation process. Startup Commutation Complete can be monitored using READ_SV variable 67 AND (16#1).

Thermal Capacitance (Digital)

Category:	Motor
Method of Selection:	Read-Only
Units:	Watt-sec / deg C
Defaults:	n/a
Range:	n/a

Description:

Thermal capacitance from the winding to ambient.

Thermal Device (Digital)

Category:	Motor
Method of Selection:	Read-Only
Units:	n/a
Defaults:	n/a
Range:	No Device, Thermal Switch, Thermistor

Description:

The type of thermal sensor embedded in the motor windings.

Thermal Resistance (Digital)

Category:	Motor
Method of Selection:	Read-Only
Units:	deg C / Watt
Defaults:	n/a
Range:	n/a

Description:

Describes the head dissipation properties of the motor.

Torque Mode Command Scale (Digital)

Category:	Command
Method of Selection:	Read-Only
Units:	mA / Volt
Defaults:	1
Range:	1 to 10000

Description:

Enter the scale factor used to convert incoming Analog Command voltage to a Current Command when the Active Operating Mode is Torque. Enter the Current Command in milliamps for an Analog Command of 1 volt.

Up to Speed (Digital)

Category:	Limits & Indicators
Method of Selection:	Read-Only
Units:	n/a
Defaults:	n/a
Range:	n/a

Description:

Monitor the Up to Speed indicator. The indicator is ON when the absolute value of the Motor Velocity is greater than or equal to the Up to Speed Rate. The indicator can be assigned to a drive output. Up to Speed can be monitored using READ_SV variable 67 AND (16#8).

Up to Speed Rate (Digital)

Category:	Limits & Indicators
Method of Selection:	Edit
Units:	RPM
Defaults:	300
Range:	0 to 32767

Description:

Enter the Up to Speed indicator rate. When the absolute value of the Motor Velocity is greater than or equal to the Up to Speed Rate, the Up to Speed indicator will be ON.

User Defined Digitizing Scaling (Digital)

Category:	Feedback and Scaling
Method of Selection:	Dropdown List
Units:	n/a
Defaults:	Off
Range:	Off, On

Description:

Select ON to enable scaling of the digitizing feedback device.

User Defined Position Scaling (Digital)

Category:	Feedback and Scaling
Method of Selection:	Dropdown List
Units:	n/a
Defaults:	Off
Range:	Off, On

Description:

Select ON to enable scaling of the position loop feedback device.

Velocity Command (Digital)

Category:	Basic/System/Tuning
Method of Selection:	Read-Only
Units:	RPM
Defaults:	n/a
Range:	-Max Speed of motor to +Max Speed of motor

Description:

Monitor the command to the velocity loop. When the Active Operating Mode is Position Control, this is the output of the drive Position Loop. When using the Control Panel Velocity or Auto Tune control modes, this is the user specified velocity.

Velocity Error (Digital)

Category:	Tuning
Method of Selection:	Read-Only
Units:	RPM
Defaults:	n/a
Range:	n/a

Description:

Monitor the difference between Velocity Command and Motor Velocity.

Velocity Loop D Gain (Digital)

Category:	Tuning
Method of Selection:	Edit
Units:	n/a
Defaults:	0
Range:	0.000 to 32.000

Description:

Enter the Velocity Loop Derivative Gain. This can be used to compensate for mechanical resonance. Typically, this is set to zero.

Velocity Loop I Gain (Digital)

Category:	Basic/Tuning
Method of Selection:	Edit
Units:	n/a
Defaults:	2000
Range:	0 to 32767

Description:

Enter the Velocity Loop Integral Gain. Increasing the I gain, increases the stiffness of the system. Excessive I gain can cause instability. Velocity Loop I Gain can be monitored and controlled using READ_SV/WRITE_SV variable 81.

Velocity Loop Integrator Hold (Digital)

Category:	Tuning
Method of Selection:	Read-Only
Units:	n/a
Defaults:	n/a
Range:	n/a

Description:

Monitor the Velocity Loop Integrator Hold function. When ON, the integrator value is held at its present value. When OFF, the integrator functions normally. Velocity Loop Integrator Hold can be assigned to a drive input. Velocity Loop Integrator Hold can be monitored and controlled using READ_SV/WRITE_SV variable 83.

Velocity Loop Integrator Inhibit (Digital)

Category:	Tuning
Method of Selection:	Read-Only
Units:	n/a
Defaults:	n/a
Range:	n/a

Description:

Monitor the Velocity Loop Integrator Inhibit function. When ON, I gain is zero and the integrator is zeroed. When OFF, the I gain is the Velocity Loop I Gain and the integrator functions normally. Velocity Loop Integrator Inhibit can be assigned to a drive input. Velocity Loop Integrator Inhibit can be monitored and controlled using READ_SV/WRITE_SV variable 82.

Velocity Loop P Gain (Digital)

Category:	Basic/Tuning
Method of Selection:	Edit
Units:	n/a
Defaults:	50
Range:	0 to 3276.7

Description:

Enter the Velocity Loop Proportional Gain. Increasing the P gain improves response time. Excessive P gain can cause instability. An extremely low P gain can cause poor servo response. Velocity Loop P Gain can be monitored and controlled using READ_SV/WRITE_SV variable 80.

Velocity Mode Command Scale (Digital)

Category:	Command
Method of Selection:	Edit
Units:	RPM / Volt
Defaults:	0.1
Range:	0.1 to 6499.9

Description:

Enter the scale factor used to convert the Analog Command voltage to a Velocity Command when the Active Operating Mode is Velocity. Enter a Velocity Command in RPM for an Analog Command of 1 volt.

Zero Speed Rate (Digital)

Category:	Limits & Indicators
Method of Selection:	Edit
Units:	RPM
Defaults:	10
Range:	0 to 32767

Description:

Enter the At Zero Speed indicator rate. When the absolute value of the Motor Velocity is less than or equal to the Zero Speed Rate, the At Zero Speed indicator will be ON.

APPENDIX S - Digital Drive Faults and Warnings

(01) Drive Heatsink Temp. Warning (Digital)

Category: Faults & Warnings

Method of Selection: Read-Only

Description:

This warning is set if the Drive Heatsink Temperature exceeds the Drive Heatsink Temperature Warning limit. The warning indicator can be assigned to a drive output to provide an early warning before a fault limit is exceeded. Drive Heatsink Temp. Warning status can be monitored using READ_SV variable 69 AND (16#1).

(02) Drive Ambient Temp. Warning (Digital)

Category: Faults & Warnings

Method of Selection: Read-Only

Description:

This warning is set if the Drive Ambient Temperature exceeds the Drive Ambient Temperature Warning limit. This warning only applies to MMC-SD-x-460 systems. The warning indicator can be assigned to a drive output to provide an early warning before a fault limit is exceeded. Drive Ambient Temp. Warning status can be monitored using READ_SV variable 69 AND (16#2).

(03) Motor Temp. Warning (Digital)

Category: Faults & Warnings

Method of Selection: Read-Only

Description:

This warning is set if the Motor – Thermal Device is a thermistor and Motor Thermistor Temperature exceeds the Motor Temperature Warning limit. The warning indicator can be assigned to a drive output to provide an early warning before a fault limit is exceeded. Motor Temp. Warning status can be monitored using READ_SV variable 69 AND (16#4).

(04) Motor Calculated Temp. Warning (Digital)

Category: Faults & Warnings

Method of Selection: Read-Only

Description:

This warning is set if the Motor Calculated Temperature exceeds the Motor Calculated Temperature Warning limit. The Motor Calculated Temperature is the anticipated motor temperature based on the amount of current being delivered over time. The warning indicator can be assigned to a drive output to provide an early warning before a fault limit is exceeded. Motor Calculated Temp. Warning status can be monitored using READ_SV variable 69 AND (16#8).

(05) Overtravel Plus Warning (Digital)

Category:	Faults & Warnings
Method of Selection:	Read-Only

Description:

This warning is set when the Overtravel Plus Fault input is off. This warning is not affected by the state of READ_SV/WRITE_SV variable 86 - Prevent Drive Overtravel Plus Fault.

(06) Overtravel Minus Warning (Digital)

Category:	Faults & Warnings
Method of Selection:	Read-Only

Description:

This warning is set when the Overtravel Minus Fault input is off. This warning is not affected by the state of READ_SV/WRITE_SV variable 87 - Prevent Drive Overtravel Minus Fault.

(11) Drive Memory Fault (Digital)

Category:	Faults & Warnings
Method of Selection:	Read-Only

Description:

This fault is set if the drive's non-volatile memory is not functioning properly. If this fault occurs, contact Danaher Motion Tech Support. Drive Memory Fault status can be monitored using READ_SV variable 68 AND (16#1). Use DRSETFLT to reset fault indications.

(12) Drive Bus Over Voltage Fault (Digital)

Category:	Faults & Warnings
Method of Selection:	Read-Only

Description:

This fault is set if the Drive Bus Voltage exceeds the Drive Bus Over Voltage Fault limit. This protects the drive's power section from voltage levels that would damage it. If a motor is rated for 220/230 volts and is used with a 440/460 volt drive, this limit will protect the motor from voltage levels that would damage it. Typical causes for this fault include an incoming line voltage that is too high or back EMF generated by the motor during deceleration requiring an external shunt system to dissipate energy. Drive Bus Over Voltage Fault status can be monitored using READ_SV variable 68 AND (16#2). Use DRSETFLT to reset fault indications.

(13) Drive PM1 Over Current Fault (Digital)

Category:	Faults & Warnings
Method of Selection:	Read-Only

Description:

This fault is set if the Current – Feedback exceeds the Drive Over Current Fault limit. This protects the drive's power section from damage due to excessive current output. Drive PM1 Over Current Fault status can be monitored using READ_SV variable 68 AND (16#4). Use DRSETFLT to reset fault.

(14) Drive Bus Under Voltage Fault (Digital)

Category:	Faults & Warnings
Method of Selection:	Read-Only

Description:

This fault is set if the Drive Bus Voltage is below the Drive Bus Under Voltage Fault limit. This prevents the drive from operating at low bus voltages. The typical cause for this fault is a load greater than the drive capability. Drive Bus Under Voltage Fault status can be monitored using READ_SV variable 68 AND (16#8). Use DRSETFLT to reset fault indications.

The Under Voltage Fault limit is 60 volts DC.

(15) Motor Temp. Fault (Digital)

Category:	Faults & Warnings
Method of Selection:	Read-Only

Description:

This fault is set if the motor overheats. This protects the motor from damage due to excessive heat. This typically occurs when the motor is undersized for the application and excessive output current causes the motor temperature to rise. This may also occur when the motor heatsink is inadequate or the ambient temperature around the motor is too high

Motor – Thermal Device	Motor Temp. Fault
None	-
Thermal Switch	Set if Motor – Thermal Switch is open
Thermistor	Set if Motor Temp. > Motor Temp. Fault Limit

Motor Temp. Fault status can be monitored using READ_SV variable 68 AND (16#10). Use DRSETFLT to reset fault indications.

If this fault condition occurs while the motor is in motion, the commanded velocity will be ramped to zero at a rate defined by the **Decel Limit** parameter, before the fault indication is set.

(16) Continuous Current Fault (Digital)

Category: Faults & Warnings

Method of Selection: Read-Only

Description:

This fault indicates the current has exceeded the continuous motor current rating for an extended period of time. Continuous Current Fault status can be monitored using READ_SV variable 68 AND (16#20). Use DRSETFLT to reset fault indications.

If this fault condition occurs while the motor is in motion, the commanded velocity will be ramped to zero at a rate defined by the **Decel Limit** parameter, before the fault indication is set.

(17) Drive Heatsink Temp. Fault (Digital)

Category: Faults & Warnings

Method of Selection: Read-Only

Description:

This fault is set if the Drive Heatsink Temperature exceeds the Drive Heatsink Temperature Fault limit. This protects the drive's power section from damage due to overheating. This occurs when the heatsink temperature rises due to excessive power output, excessive ambient temperature or inadequate airflow. Drive Heatsink Temp. Fault status can be monitored using READ_SV variable 68 AND (16#40). Use DRSETFLT to reset fault indications.

If this fault condition occurs while the motor is in motion, the commanded velocity will be ramped to zero at a rate defined by the **Decel Limit** parameter, before the fault indication is set.

(21) Drive F2 Feedback Fault (Digital)

Category: Faults & Warnings

Method of Selection: Read-Only

Description:

Selection of a feedback configuration that uses F2 feedback for motor control will result in the detection of a loss of feedback condition on F2 setting this fault and shutting down motor control. Typical causes for this fault include a faulty feedback cable, electrical noise on the feedback signal due to improper feedback cable routing or failure of the feedback device in the motor. Drive F2 Feedback Fault status can be monitored using READ_SV variable 68 AND (16#80). Use DRSETFLT to reset fault indications.

(22) Drive F1 Feedback Fault (Digital)

Category: Faults & Warnings

Method of Selection: Read-Only

Description:

Selection of a feedback configuration that uses F1 feedback for motor control will result in the detection of a loss of feedback condition on F1 setting this fault and shutting down motor control. Typical causes for this fault include a faulty feedback cable, electrical noise on the feedback signal due to improper feedback cable routing or failure of the feedback device in the motor. Drive F1 Feedback Fault status can be monitored using READ_SV variable 68 AND (16#100). Use DRSETFLT to reset fault indications.

(23) Drive Ambient Temp. Fault (Digital)

Category: Faults & Warnings

Method of Selection: Read-Only

Description:

This fault is set if the Drive Ambient Temperature exceeds the Drive Ambient Temperature Fault limit. This protects from erratic operation of the drive due to excessive heat. This fault only applies to MMC-SD-x-460 systems. Typically, this occurs if the drive enclosure temperature rises due to excessive power dissipation. Drive Ambient Temp. Fault status can be monitored using READ_SV variable 68 AND (16#200). Use DRSETFLT to reset fault indications.

If this fault condition occurs while the motor is in motion, the commanded velocity will be ramped to zero at a rate defined by the **Decel Limit** parameter, before the fault indication is set.

(24) Motor Calculated Temp. Fault (Digital)

Category: Faults & Warnings

Method of Selection: Read-Only

Description:

This fault is set if the Motor Calculated Temperature exceeds the Motor Calculated Temperature Fault limit. This protects the motor from damage due to excessive heat. The Motor Calculated Temperature is the anticipated motor temperature based on the amount of current being delivered over time. Motor Calculated Temp. Fault status can be monitored using READ_SV variable 68 AND (16#400). Use DRSETFLT to reset fault indications.

If this fault condition occurs while the motor is in motion, the commanded velocity will be ramped to zero at a rate defined by the **Decel Limit** parameter, before the fault indication is set.

(25) Drive Timing Fault (Digital)

Category: Faults & Warnings

Method of Selection: Read-Only

Description:

This fault is set if a timing error is detected in the execution of the control algorithms performed by the drive's digital signal processor. If this fault occurs, contact Danaher Motion Tech Support. Drive Timing Fault status can be monitored using READ_SV variable 68 AND (16#800). Use DRSETFLT to reset fault indications.

(26) Drive Interface Fault (Digital)

Category: Faults & Warnings

Method of Selection: Read-Only

Description:

This fault is set if a communication error is detected in the transmission of information between the drive's digital signal processor and the drive's power section. If this fault occurs, contact Danaher Motion Tech Support. Drive Interface Fault status can be monitored using READ_SV variable 68 AND (16#1000). Use DRSETFLT to reset fault indications.

(27) User Set Fault (Digital)

Category: Faults & Warnings

Method of Selection: Read-Only

Description:

This fault is set by selecting the PiCPro Set User Fault command or when activating the Control Panel mode while the drive is enabled. User Set Fault status can be monitored using READ_SV variable 68 AND (16#2000). Use DRSETFLT to reset fault indications.

If this fault condition occurs while the motor is in motion, the commanded velocity will be ramped to zero at a rate defined by the **Decel Limit** parameter, before the fault indication is set.

(31) Drive F1 Communication Fault (Digital)

Category:	Faults & Warnings
Method of Selection:	Read-Only

Description:

This fault is set if a communication error is detected in the transmission of information between the drive and a high resolution or multi-turn absolute feedback device. Typical causes for this fault include a faulty feedback cable, electrical noise on the feedback signal due to improper feedback cable routing or failure of the feedback device in the motor. Drive F1 Communication Fault status can be monitored using READ_SV variable 68 AND (16#4000). Use DRSETFLT to reset fault indications.

(32) Over Speed Fault (Digital)

Category:	Faults & Warnings
Method of Selection:	Read-Only

Description:

This fault is set if the Motor Velocity exceeds the Over Speed Fault limit. This typically occurs when an unexpected event causes system instability resulting in improper motor control. Over Speed Fault status can be monitored using READ_SV variable 68 AND (16#8000). Use DRSETFLT to reset fault indications.

(33) Over Current Fault (Digital)

Category:	Faults & Warnings
Method of Selection:	Read-Only

Description:

This fault is set if the Current – Average exceeds the Over Current Fault limit. This is used to protect the application from excessive torque. If the drive and motor can deliver more torque than the application can accept, use this limit to protect it. Over Current Fault status can be monitored using READ_SV variable 68 AND (16#10000). Use DRSETFLT to reset fault indications.

(34) Control Panel Disconnect Fault (Digital)

Category: Faults & Warnings

Method of Selection: Read-Only

Description:

This fault is set if a communication error occurs while drive control is being performed using the PiCPro Control Panel tools. A typical cause for this fault is disconnecting the cable between the drive and the workstation while in Control Panel mode. Control Panel Disconnect Fault status can be monitored using READ_SV variable 68 AND (16#20000). Use DRSETFLT to reset fault indications.

(35) Drive Power Module Fault (Digital)

Category: Faults & Warnings

Method of Selection: Read-Only

Description:

This fault is set by the drive’s power section if it detects a fault condition. This fault is typically set when the power section detects a demand for power that would cause damage to the power section. Drive Power Module Fault status can be monitored using READ_SV variable 68 AND (16#40000). Use DRSETFLT to reset fault indications.

(36) Feedback Type Mismatch Fault (Digital)

Category: Faults & Warnings

Method of Selection: Read-Only

Description:

This fault occurs if:

- Resolver feedback is selected, but the resolver feedback board is not installed.
- BiSS feedback is selected and the drive hardware does not support BiSS feedback (HDW revision less than 6.0)
- Stegmann multi-turn or single-turn selected, but the opposite is installed.

Feedback device selection must match appropriate hardware.

This fault cannot be reset.

(37) ENDAT / BiSS Fault (Digital)

Category: Faults & Warnings

Method of Selection: Read-Only

Description:

This fault is set if one of the following conditions exists:

- The Endat / BiSS encoder did not respond to a serial message.
- The Endat / BiSS Encoder responded to a serial message but there was a CRC error with the message
- The Endat Encoder Counts Per Revolution does not match the entered value.
- Hardware does not match the selected device (see fault 36).

Use DRSETFLT to reset fault indications.

Make sure all cable connections are secure and proper grounding is applied.

(41) Drive Relay Fault (Digital)

Category:	Faults & Warnings
Method of Selection:	Read-Only

Description:

This fault is set if the drive's power section detects an error during power up. If this fault occurs, contact Danaher Motion Tech Support. Drive Relay Fault status can be monitored using READ_SV variable 68 AND (16#200000). Use DRSETFLT to reset fault indications.

(42) Drive PM2 Over Current Fault (Digital)

Category:	Faults & Warnings
Method of Selection:	Read-Only

Description:

This fault is set if the Current – Feedback exceeds the Drive Over Current Fault limit. This protects the drive's power section from damage due to excessive current output. This fault applies to MMC-SD-x-460 drives only. Drive PM2 Over Current Fault status can be monitored using READ_SV variable 68 AND (16#400000). Use DRSETFLT to reset fault indications.

(43) Drive PM Temp Fault (Digital)

Category:	Faults & Warnings
Method of Selection:	Read-Only

Description:

This fault is set if the drive power module temperature exceeds the Drive Power Module Temperature Fault limit. This protects the drive's power section from damage due to overheating. This occurs when the power module temperature rises due to excessive power output, excessive ambient temperature or inadequate airflow. Drive PM Over Temp Fault status can be monitored using READ_SV variable 68 AND (16#800000). Use DRSETFLT to reset fault indications.

If this fault condition occurs while the motor is in motion, the commanded velocity will be ramped to zero at a rate defined by the **Decel Limit** parameter, before the fault indication is set.

(44) Motor Ground Fault (Digital)

Category:	Faults & Warnings
Method of Selection:	Read-Only

Description:

This fault indicates a ground fault has occurred. Specifically, the current in the motor ground is excessive. Note: This fault only applies to MMC-SD-x-460 systems. Motor Ground Fault status can be monitored using READ_SV variable 68 AND (16#1000000). Use DRSETFLT to reset fault indications.

(45) Drive AC Input Over Voltage Fault (Digital)

Category:	Faults & Warnings
Method of Selection:	Read-Only

Description:

This fault indicates the incoming AC voltage is too high. Note: This fault only applies to MMC-SD-x-460 systems. Drive AC Input Over Voltage Fault status can be monitored using READ_SV variable 68 AND (16#2000000). Use DRSETFLT to reset fault indications.

(46) Overtravel Plus Fault (Digital)

Category:	Faults & Warnings
Method of Selection:	Read-Only

Description:

This fault is set when the Overtravel Plus Fault input is off and Prevent Drive Overtravel Plus Fault is off. Overtravel Plus Fault status can be monitored using READ_SV variable 68 AND (16#400 0000). Use DRSETFLT to reset fault indications. To prevent the Overtravel Plus Fault input from setting a fault, write a 1 to WRITE_SV variable 86, Prevent Drive Overtravel Plus Fault. While Prevent Drive Overtravel Plus Fault is set to a 1, the Overtravel Plus Fault input will not cause a fault, but will still prevent plus motion. To reactivate the fault checking of the Overtravel Plus Fault input write a 0 to WRITE_SV variable 86.

(47) Overtravel Minus Fault (Digital)

Category: Faults & Warnings

Method of Selection: Read-Only

Description:

This fault is set when the Overtravel Minus Fault input is off and Prevent Drive Overtravel Minus Fault is off. Overtravel Minus Fault status can be monitored using READ_SV variable 68 AND (16#800 0000). Use DRSETFLT to reset fault indications. To prevent the Overtravel Minus Fault input from setting a fault, write a 1 to WRITE_SV variable 87, Prevent Drive Overtravel Minus Fault. While Prevent Drive Overtravel Minus Fault is set to a 1, the Overtravel Minus Fault input will not cause a fault, but will still prevent minus motion. To reactivate the fault checking of the Overtravel Minus Fault input write a 0 to WRITE_SV variable 87.

(51) Digital Link Communication Error (Digital)

Category: Faults & Warnings

Method of Selection: Read-Only

Description:

This fault is set when two consecutive corrupt Digital Link messages are detected or no Digital Link messages are received within 250 microseconds. Digital Link Communication Error status can be monitored using READ_SV variable 68 AND (16#10000000). This fault requires that the user servo setup function and DSTRTSRV be executed prior to executing DRSETFLT to reset the fault indication.

If this fault condition occurs while the motor is in motion, the commanded velocity will be ramped to zero at a rate defined by the **Decel Limit** parameter, before the fault indication is set.

(52) Invalid Switch Setting Fault (Digital)

Category: Faults & Warnings

Method of Selection: Read-Only

Description:

This fault is set when the drive address switch setting is set to 0 or greater than 64 or its setting is changed while the Digital Link is operating in cyclic communications mode. Invalid Switch Setting Fault status can be monitored using READ_SV variable 68 AND (16#20000000). Use DRSETFLT to reset fault indications.

Note:

Digital Link initialization must be performed before this fault can be reset.

(53) Hardware Failure Fault (Digital)

Category:	Faults & Warnings
Method of Selection:	Read-Only

Description:

This fault indicates an unrecoverable hardware error.

Last/Previous Fault Message (Digital)

Category:	Feedback and Scaling
Method of Selection:	Read-Only
Units:	n/a
Defaults:	n/a
Range:	n/a

Description:

These parameters are a record of the faults as they occur with the most recent on top.

Index

1

2

4

5

H

I

K

L

M