MP2000 Series Function Block Details

Please click a Function Block to jump to the details.

Motion Blocks

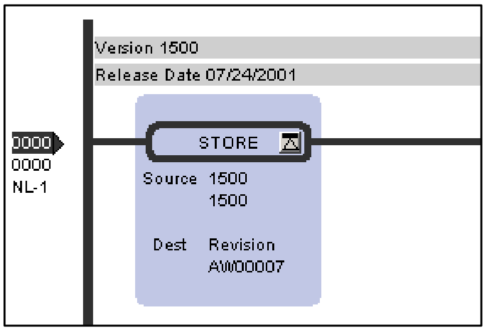
Administrative Blocks

I/O Blocks

Revision Level

As part of best practices, it is advised to keep track of the revision level of each function block for the ease of technical support and maintenance. Each block has a revision number that is defined in the first rung of function code. This value must be altered if the block is modified. The number is broken into two parts as follows.

Revision Number format: **XXYY**
XX = Revision by Yaskawa engineering and registered
YY - Revision by Non Yaskawa personnel and unregistered changes



Version 1500
Release Date 07/24/2001

Source 1500
1500

Dest Revision
AW00007

The revision number uses an 'A' register to allow the value to be monitored. The revision level can be determined by looking at the last word from the base address including the base address, the revision can be seen.

If it is necessary to replace a function block with a newer revision, simply import the newer function block over the existing version. Doing this may clear all of the I/O fields. If this occurs, simply re-enter the input and output data into the block, then save and verify the project.

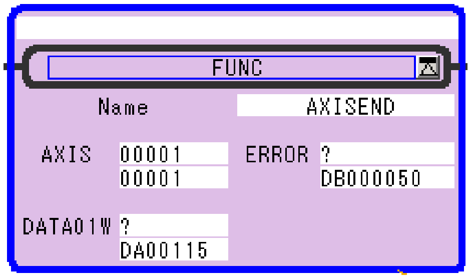
AXIS END function

Function block for MP2000 series

<AXISEND> Function Block Summary

This function block writes necessary values to the RDA at the end of the motion scan cycle. This block is essential if any of the Blocks in the set are to be used.

Function Block Diagram



FUNC

Name AXISEND

AXIS 00001
00001

ERROR ?
DB000050

DATA01W ?
DA00115

<AXISEND> Function Block Operation Note

- NOTE: SVON and AXISEND blocks must be used together, all motion type blocks for a specific axis must be placed between the SVON and AXISEND blocks. This block must be used in the High Speed Scan drawings, and should be the last block executed after all other motion function blocks within the scan.
- This block writes values from RDA to specific axis control registers required for controlling motion. This block was created to reduce the overall volume of code in all other function blocks for multi axis. It handles circuit number and axis number within the circuit, and relates it to the logical axis number selected by the programmer from the RDAINIT blocks.
- Example: for axis #1, this block writes to OW8000 RUN mode, OW8003, the unit selection, OW8004 latch demand detection signal selection, OW8005 home position return, OW8008 motion command code, OW8009 motion command control flag, OL8010 speed reference, OL801C positioning command, OL802A latch zone lower limit of the interval setting, OL802C latch zone upper limit setting, OL8036 acceleration time, OL8038 deceleration time, (see RDA matrix for details)
- This block also writes to ERRORID1 and ERRORID2
- One word is used as working registers for this function, starting at the address in *Data01W*.

<AXISEND> Input and Output Register Map

Output Registers

The following registers are used from the function block as an output. To check the execution of the function, they can be monitored by the MPE720 program.

Output	Type	Description
ERROR	bit	If an error occurs during block execution, this output bit turns ON, but will reset if error condition goes away.

Input Registers

The following registers are used to block the function as an input. It is necessary to select the option, define the parameter, and make these function if necessary by the user.

Input	Type	Description	Range and state
AXIS	Word	Axis number related to the block.	1～16

<AXISEND> Block Fault Condition:

The following table outlines several situations that may cause an error. The block error can be cleared by the *ENABLE* bit going low.

Error bit of internal	Cause	Attention
axisinerr ZB000020	The axis number entered on the input is not an acceptable value	Function blocks can only control 1 to 16 axes. Any value greater or smaller then this will cause an error. This does not set the RDA Error ID.

<AXISEND> Working Registers

This table outlines the data in the registers used by the function block. There is not usually any need for the user to access any of these bits directly.

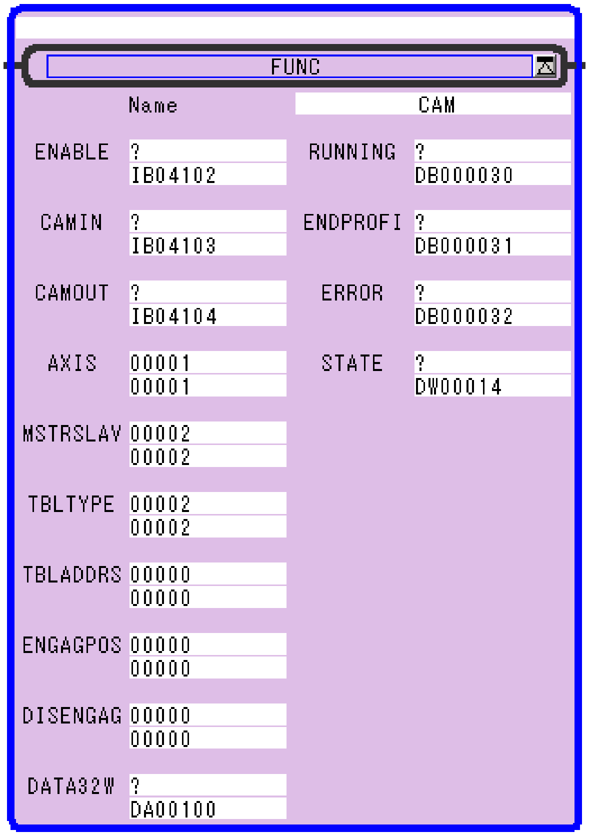
Register No	TYPE	Name	Content
AW00000	Working	Revision	Revision Level of the function block.

CAM function
Function block for MP2000 series

<CAM> Function Block Summary

The Cam block is a state block, different from the PLCopen spec in order to take advantage of the power of the MP products. Master/Slave pair is a numeric input indicating the master/slave source data from the Modulus Engine, and for the Cam Offset, Slave Offset, and Cam Scale. This block will position the slave based on the position of the master and the cam table. To operate, this block must have a Modulus Engine function block associated with the master/slave pair defined, to provide modulated master data. Slave offset, Cam offset, and Cam Scale function blocks can be used to make this block more flexible.

Function Block Diagram



FUNC

	Name	CAM	
ENABLE	? IB04102	RUNNING	? DB000030
CAMIN	? IB04103	ENDPROFI	? DB000031
CAMOUT	? IB04104	ERROR	? DB000032
AXIS	00001 00001	STATE	? DW00014
MSTRSLAV	00002 00002		
TBLTYPE	00002 00002		
TBLADDRS	00000 00000		
ENGAGPOS	00000 00000		
DISENGAG	00000 00000		
DATA32W	? DA00100		

<CAM> Function Block Operation Notes

- **IMPORTANT NOTE: If the CamScale value in the RDA (ML56**8) is zero the slave will not move.**
- Slave Offset, Cam Scale, Cam Offset, Mod Engine, and Change Dynamics function blocks may be useful when using this block.
- *ENABLE*: Rising edge of the *ENABLE* input initiates block operation, and several block input values will be read once. These inputs are: *AXIS, MSTRSLAV, TBLTYPE, TBLADDRS.* This event will also verify & set the accel & decel values in the servopack to zero, to guarantee tight response. Note that this only means that the MP controller will control the accel/decel rates as defined in the CAM Table Profile (and the servopack will not limit the accel/decel).
- *ENABLE*: If the *ENABLE* input goes off during operation, the block will stop operating and the axis will *immediately* hold on its last commanded position from the Cam table. Warning: This can cause a rapid deceleration, which may damage machine mechanisms!
- *CAMIN*: While *ENABLE* input is high, Rising edge of *CAMIN* input initiates camming operation, and *ENGAGPOS* input value is read. Slave will then wait to engage camming until the master reaches the engage position. *STATE* output will indicate the camming operation state.
- *CAMOUT:* While *ENABLE* input is high, Rising edge of *CAMOUT* input initiates cam stop operation, and *DISENGAG* input value is read. The slave axis will continue to cam until the disengage (*DISENGAG*) input position is reached by the master. Then the slave axis will hold on the position defined by the cam table referenced by the *master disengage position*.
- Speed: The maximum speed the slave axis can go is the Max speed, set in the RDA. If the master commands the slave to go faster then this speed, axis will travel at the max. speed, and will output and latch the *ERROR* output. Note that the slave will continue to cam.
- Master Data: Modulated master data (according to desired machine cycle), must come from the Mod Engine function block if CAMSCALE, CAMOFFST, or SLAVEOFF function blocks are used. See MOD_ENG function block.
- Scale: If the Cam Scale (ML56**8) value is 0 the slave will not move because this is a multiplier to the commanded position (1=0.01%). See CAMSCALE function block.
- Master Offset: The Cam Offset from the RDA is used to shift the value of the master as it's position relates to the slave. See CAMOFFST function block.
- Slave Offset: The slave offset from the RDA is used to shift the slave as its position as it relates to the Cam table. See SLAVEOFF function block.

- CAM Table: Cam table is expected to be in 32-bit long integer words. Hence, each CAM table master/slave pair entry will take up 4 16-bit regular words (two words for master, two words for slave). The first long word of every table contains the table size in points. Example of a cam table with 10 master/slave pairs consumes MW04000 thru MW04041 where ML04000=10, ML04002=first master position, ML04004=first slave position, ML04006=second master position, ML04038=tenth master position, ML04040=tenth slave position. The M register address range is 0000-29999.
  Range of register:
  MW00000 to MW29999 (M-register limit is 00000 – 29999 due to RDA)
  CW00000 to CW16382 (C-register limit is 0-16382)
  DW00000 to DW16382 (D-register limit is 0-16382)
- One-way vs Cyclic camming comment: If the first and last slave positions are identical then it is a cyclic cam, otherwise it is treated as a one-way cam, and the slave will be offset by the difference each end of profile is reached (this is to prevent it from jumping).
- Note that the master/slave pairs are separated by 50 words in the RDA (up to 10 pairs). Example: Master/Slave pair #1 starts at MW56000, Master/Slave pair #2 starts at MW56050, Master/Slave pair #3 starts at MW56100, etc
- Internal Operation1: This block uses Motion Command Code. When the master reaches the engage position, the output of the function generator (FGN) function is assumed to be the current position of the slave. Any error in these positions is placed into an offset to avoid motor jumping. The slave command position (position control mode) is updated each scan from the function generator (FGN) output multiplied by cam scale and shifted by the slave offset. The result is placed in OL**1C position command register for the slave axis
- Internal Operation2: Under normal operation, the camming block operates in 'Interpolation mode' (OW**08=4). If a latch block is used to capture a latched position, the mode will be switched to 'Interpolation mode with latch' (OW**08=6). After the latch occurs, the block is returned to standard 'Interpolation mode'. This switching feature happens in the background without affecting intended slave camming motion
- Thirty-two words are used as working registers for this function, starting at the address in *Data32W*.

<CAM> Input and Output Register Map

Output Registers

The following registers are used as outputs from the function block. They can be monitored by the LadderWorks program to check the execution of the function.

Output	TYPE	Content
RUNNING	Bit	Bit goes on when slave is camming, and the slave is being controlled by the master (STATE = 2)
ENDPROFI	Bit	Puts out a single scan high pulse every time it reaches the end of the cam profile in either direction.
ERROR	Bit	This bit is latched ON when the EXECUTE input is high and an error occurred in the block.
STATUS	Word	The State of the block 0 = Disengaged 1 = Waiting to engage 2 = Camming 4 = Disengaging

Input Registers

The following registers are used as inputs to the function block. They select the options and define the parameters that the user needs to make the Home function work as necessary.

Input	Type	Description	Range and state
ENABLE	Bit	Block enable – Block cannot execute unless this is TRUE.	Rising Edge – see operation notes TRUE – Block enable FALSE – Block disable
CAMIN	Bit	Sets block to search for ENGAGE position in master. When reached axis begins camming	RISING EDGE- initiates search for engage position and sets up for camming
CAMOUT	Bit	Sets block to search for DISENGAG position in master. When reached axis stops and holds.	RISING EDGE- initiates search for disengage position and stops camming and holds when position is reached by master.
AXIS	Word	Axis number related to the block	1 to 16
M-S-PAIR	Word	Master Slave Pair as defined by the RDA	1 to 10 integer. This value is needed to match the value of the MOD-ENG function block.
TBLTYPE	Word	Defines what register type the cam table will be in. CAM table consists of 32-bit long integer words	1 = M registers 2 = C registers 3 = D registers
TBLADDRS	Word	Starting address number of the cam table. In case of D register table, this value is relative to the assigned data address at 'DATA31W'. TBLADDRS+DATA31W = Actual Starting Address	0～3276 If the end of the table is greater than 32767, the an error will be set. (The limit of M-reg is MW00000 to MW29999)
ENGAGPOS	Long	Master position in counts to start the slave camming	0 to the machine cycle value defined in the MOD-ENG block
DISENGAG	Long	Master position in counts to stop the slave camming	0 to the machine cycle value defined in the MOD-ENG block
DATA32W	Address	Address of the first working register.	Thirty-two words of register space are used by this function.

<CAM> Block Fault Condition

The following table outlines several situations that may cause an error, and will turn on the blocks "Error" output bit. The block "Error" output will cleared if the *EXECUTE* bit goes low, but the Error ID (MW3**81 and MW3**82) will remain in the RDA. To reset the Error ID, use the Alarm Reset Function Block.

Note that each axis has its own Error ID stored in its RDA axis section, offset by 200 for each axis. Example: Axis#1 stores to MW30181, Axis #20 stores to MW30381, etc (note that Master/Slave pair is irrelevant).

Internal Fault bit	Cause	Note
msinrng AB000010	Master/Slave pair input out of range	Value on input must be from 1 to 10. If this bit goes low while execute is high an error will occur. This will also generate an error in the RDA at MB3**813
ctlnrng AB000011	Table type input is out of range.	Value on input must be from 1 to 3. If this bit goes low while execute is high an error will occur. This will also generate an error in the RDA at MB3**813.
addrInRng AB000012	Final address of table is beyond the range of the register.	Checks that final address of the table (Start address + size) does not exceed max register size (29999 for M, 16482 for D and C type registers). If this bit is low on the rising edge of execute an error will occur. This will also generate an error in the RDA at MB3**813.
mstErr AB00001F	Last value in Cam table is greater then machine cycle.	If the last master value in the cam table is greater then the machine cycle an error will occur. This will also generate an error in the RDA at MB3**81C.
Direr AB00001F	Direction not allowed by SVON	The direction commanded was not enabled by SVON function. This will also generate an error in the RDA at MB3**814.
CmndError AB00001E	Another block took control of axis	If another axis controlling block while this block is running. It will take control from this function. This does not set the RDA Error ID.
AxisInErr AB000050	The axis number entered on the input is not an acceptable value	The function blocks can only control 1 to 16 axis. Any value greater or smaller then this will cause an error. This does not set the RDA Error ID.

<CAM> Working Register

This table outlines the data in the thirty-three registers used by the function block. There is not usually any need for the user to access any of these bits directly.

Register No.	Type	Name	Description
AW00000			
Bit 0	IN	EXECUTE	*ENABLE* input (XB000000)
Bit 1	IN	CamIn	*CAMIN* input (XB000001)
Bit 2	IN	CamOut	*CAMOUT* input (XB000002)
Bit 3	Working	mRegister	Indicates table is in M register.
Bit 4	Working	cRegister	Indicates table is in C register.
Bit 5	Working	dRegister	Indicates table is in D register.
Bit 6	Working	engaging	On when waiting for master to reach engage position.
Bit 7	OUT	running	Directly controls YB000000 (*RUNNING* Output)
Bit 8	OUT	endOfTrvl	Directly controls YB000001 (*ENDPROFI* Output)
Bit 9	OUT	Mistake	Directly controls YB000002 (ERROR Output)
Bit A	Working	oneshotA	Reserved
Bit B	Working	firstPass	One shot coil for initializing axis when CAMIN input goes high.
Bit C	Working	oneshotC	Reserved
Bit D	IN	runInit	One shot coil for initializing axis when engage position is reached.
Bit E	IN	oneshotE	Reserved
Bit F	Working	oneshotF	Reserved
AW00001			
Bit 0	Working	msinrng	Verifies Master/Slave input is in range.
Bit 1	Working	ctInrng	Verifies Table Type input is in range.
Bit 2	Working	addrInRng	Verifies that the entire Cam Table is in the register and did not go beyond the register limits.
Bit 3	Working	disengaging	On when camming and CAMOUT input goes high.
Bit 4	Working	above_point	Reserved
Bit 5	Working	d_abovepos	Reserved
Bit 6	Working	oneWay	On when first and last value of cam table are not equal.
Bit 7	Working	exeInit	One shot coil on for first pass of block being executed.
Bit 8	Working	disengaged	One shot coil for when disengage position is reached and called for.
Bit A	Working	mstrErr	On if last master value in cam table is greater then master machine cycle.
Bit B	Working	posout	Reserved
Bit C	Working	backwords。	Reserved
Bit D	Working	negOut	Reserved

Bit E	Working	cmndError	One shot coil for when another block takes control from this block.
Bit F	Working	direrr	On if a direction is commanded that is not enabled by the SVON block.
AW00002	Working	mstr_slave	MstrSlave [value of (XW00001) – 1]
AW00003	Working	filter1	Temporary values used to filter bits.
AW00004	OUT	camstate	Directly controls YW00001 (*STATE* Output).
AW00005			
Bit 0	Working	axisInErr	Goes high for one scan if Axis input is out of range.
AL00006	Working	yFirst	First position of slave in Cam table.
AL00008	Working	yLast	Last position of slave in Cam table.
AL00010	Working	xLast	Last position of Master in Cam table.
AL00012	Working	lastAddress	Final register address of Cam table.
AL00014	Working	Window	Tolerance value for engage and disengage position. Divide XLAST by 120. IE – this window is 0.8% of the machine cycle.
AL00016	Working	masterpos	Value of Master after CAM shift.
AL00018	Working	engage_difference	Distance from engage or disengage point.
AL00020	Working	base_slave_pos	Reserved
AL00022	Working	fgnOut	Value returned from Cam table function generator
AL00024	Working	slave_Pos_Cmnd	Reserved
AL00026	Working	deltaY	Difference between first and last slave position in table, used if it's a one-way cam to offset the slave
AL00028	Working	o pos_o	Reserved
AW00030	Working	rdaMult	Value for address offset to locate proper RDA
AW00031	Working	Revision	Revision Level of Block.

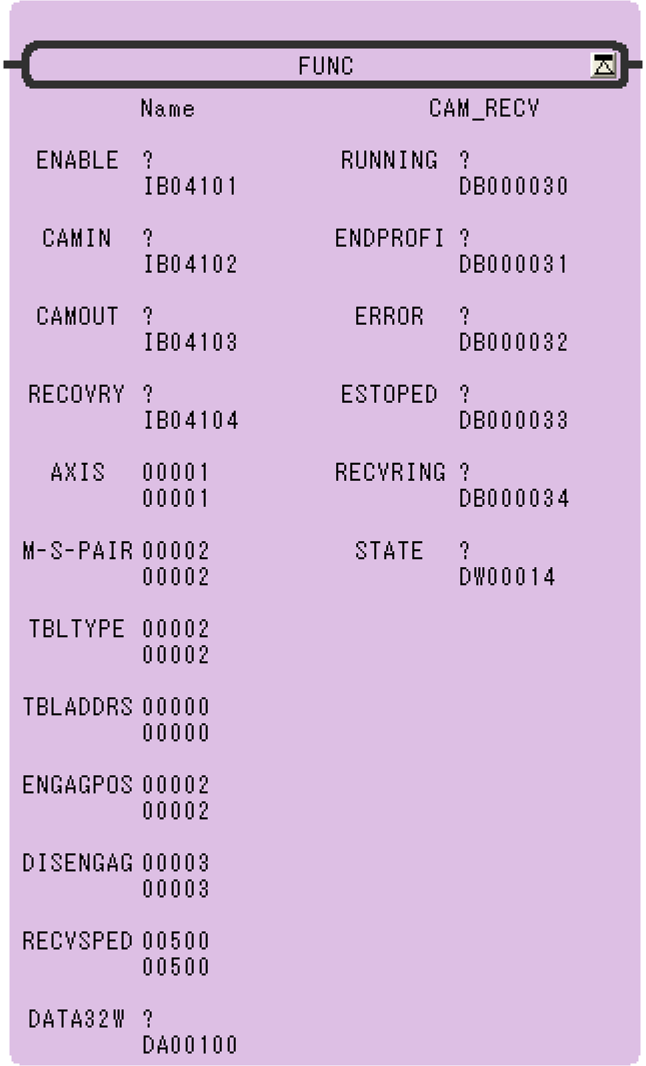
CAM RECOVERY function

Function block for MP2000 series

<CAM_RECV> Function Block Summary

The Cam block is a state block, different from the PLCopen spec in order to take advantage of the power of the MP products. Master/Slave pair is a numeric input indicating the master/slave source data from the Modulus Engine, and for the Cam Offset, Slave Offset, and Cam Scale. This block will position the slave based on the position of the master and the cam table. To operate, this block must have a Modulus Engine function block associated with the master/slave pair defined, to provide modulated master data. Slave offset, Cam offset, and Cam Scale function blocks can be used to make this block more flexible. CAM_RECV function block has E-stop recovery capability in addition to the CAM Function Block.

Function Block Diagram



```
                        FUNC

          Name                    CAM_RECV

 ENABLE   ?                 RUNNING  ?
          IB04101                    DB000030

  CAMIN   ?                ENDPROFI  ?
          IB04102                    DB000031

 CAMOUT   ?                  ERROR   ?
          IB04103                    DB000032

RECOVRY   ?                 ESTOPED  ?
          IB04104                    DB000033

   AXIS   00001            RECVRING  ?
          00001                      DB000034

M-S-PAIR  00002               STATE  ?
          00002                      DW00014

TBLTYPE   00002
          00002

TBLADDRS  00000
          00000

ENGAGPOS  00002
          00002

DISENGAG  00003
          00003

RECVSPED  00500
          00500

DATA32W   ?
          DA00100
```

<CAM_RECV> Function Block Operation Notes

- **IMPORTANT NOTE: If the CamScale value in the RDA (ML56**8) is zero the slave will not move.**
- Slave Offset, Cam Scale, Cam Offset, Mod Engine, and Change Dynamics function blocks may be useful when using this block.
- *ENABLE*: Rising edge of the *ENABLE* input initiates block operation, and several block input values will be read once. These inputs are: *AXIS, MSTRSLAV, TBLTYPE, TBLADDRS.* This event will also verify & set the accel & decel values in the servopack to zero, to guarantee tight response. Note that this only means that the MP controller will control the accel/decel rates as defined in the CAM Table Profile (and the servopack will not limit the accel/decel).
- *ENABLE*: If the *ENABLE* input goes off during operation, the block will stop operating and the axis will *immediately* hold on its last commanded position from the Cam table. Warning: This can cause a rapid deceleration, which may damage machine mechanisms!
- *CAMIN*: While *ENABLE* input is high, Rising edge of *CAMIN* input initiates camming operation, and *ENGAGPOS* input value is read. Slave will then wait to engage camming until the master reaches the engage position. *STATE* output will indicate the camming operation state.
- *CAMOUT:* While *ENABLE* input is high, Rising edge of *CAMOUT* input initiates cam stop operation, and *DISENGAG* input value is read. The slave axis will continue to cam until the disengage (*DISENGAG*) input position is reached by the master. Then the slave axis will hold on the position defined by the cam table referenced by the *master disengage position*.
- Speed: The maximum speed the slave axis can go is the Max speed, set in the RDA. If the master commands the slave to go faster then this speed, axis will travel at the max. speed, and will output and latch the *ERROR* output. Note that the slave will continue to cam.
- Master Data: Modulated master data (according to desired machine cycle), must come from the Mod Engine function block if CAMSCALE, CAMOFFST, or SLAVEOFF function blocks are used. See MOD_ENG function block.
- Scale: If the Cam Scale (ML56**8) value is 0 the slave will not move because this is a multiplier to the commanded position (1=0.01%). See CAMSCALE function block.
- Master Offset: The Cam Offset from the RDA is used to shift the value of the master as it's position relates to the slave. See CAMOFFST function block.
- Slave Offset: The slave offset from the RDA is used to shift the slave as its position as it relates to the Cam table. See SLAVEOFF function block.

- CAM Table: Cam table is expected to be in 32-bit long integer words. Hence, each CAM table master/slave pair entry will take up 4 16-bit regular words (two words for master, two words for slave). The first long word of every table contains the table size in points. Example of a cam table with 10 master/slave pairs consumes MW04000 thru MW04041 where ML04000=10, ML04002=first master position, ML04004=first slave position, ML04006=second master position, ML04038=tenth master position, ML04040=tenth slave position. The M register address range is 0000-29999.
  Range of register:
  MW00000 to MW29999 (M-register limit is 00000 – 29999 due to RDA)
  CW00000 to CW16382 (C-register limit is 0-16382)
  DW00000 to DW16382 (D-register limit is 0-16382)
- One-way vs Cyclic camming comment: If the first and last slave positions are identical then it is a cyclic cam, otherwise it is treated as a one-way cam, and the slave will be offset by the difference each end of profile is reached (this is to prevent it from jumping).
- Note that the master/slave pairs are separated by 50 words in the RDA (up to 10 pairs). Example: Master/Slave pair #1 starts at MW56000, Master/Slave pair #2 starts at MW56050, Master/Slave pair #3 starts at MW56100, etc
- Internal Operation1: This block uses Motion Command Code. When the master reaches the engage position, the output of the function generator (FGN) function is assumed to be the current position of the slave. Any error in these positions is placed into an offset to avoid motor jumping. The slave command position (position control mode) is updated each scan from the function generator (FGN) output multiplied by cam scale and shifted by the slave offset. The result is placed in OL**1C position command register for the slave axis
- Internal Operation2: Under normal operation, the camming block operates in 'Interpolation mode' (OW**08=4). If a latch block is used to capture a latched position, the mode will be switched to 'Interpolation mode with latch' (OW**08=6). After the latch occurs, the block is returned to standard 'Interpolation mode'. This switching feature happens in the background without affecting intended slave camming motion
- Thirty-two words are used as working registers for this function, starting at the address in *Data32W*.
- E-Stop Condition: Control Power of both the Servo pack and the Controller are ON and Main power of the Servo Pack is OFF while CAM is engaged.
- E-Stop Recovery motion: The Slave axis moves to the designated position based on the current Master position at the speed set in the RECVSPD [counts/sec].

<CAM_RECV> Input and Output Register Map

Output Registers

The following registers are used as outputs from the function block. They can be monitored by the LadderWorks program to check the execution of the function.

Output	TYPE	Content
RUNNING	Bit	Bit goes on when slave is camming, and the slave is being controlled by the master (STATE = 2)
ENDPROFI	Bit	Puts out a single scan high pulse every time it reaches the end of the cam profile in either direction.
ERROR	Bit	This bit is latched ON when the EXECUTE input is high and an error occurred in the block.
ESTOPED	Bit	This bit is latched ON when E-Stop condition occurs while CAM is engaged. This bit will be OFF when ENABLE input is OFF or recovery motion is done.
RECVRING	Bit	This bit is ON when the axis is in the recovery motion.
STATUS	Word	The State of the block 0 = Disengaged 1 = Waiting to engage 2 = Camming 4 = Disengaging

Input Registers

The following registers are used as inputs to the function block. They select the options and define the parameters that the user needs to make the Home function work as necessary.

Input	Type	Description	Range and state
ENABLE	Bit	Block enable – Block cannot execute unless this is TRUE.	Rising Edge – see operation notes TRUE – Block enable FALSE – Block disable
CAMIN	Bit	Sets block to search for ENGAGE position in master. When reached axis begins camming	RISING EDGE- initiates search for engage position and sets up for camming
CAMOUT	Bit	Sets block to search for DISENGAG position in master. When reached axis stops and holds.	RISING EDGE- initiates search for disengage position and stops camming and holds when position is reached by master.
RECOVRY	Bit	Start Recovery Motion when E-stop occurred.	RISING EDGE- initiates the recovery motion.
AXIS	Word	Axis number related to the block	1 to 16
M-S-PAIR	Word	Master Slave Pair as defined by the RDA	1 to 10 integer. This value is needed to match the value of the MOD-ENG function block.
TBLTYPE	Word	Defines what register type the cam table will be in. CAM table consists of 32-bit long integer words	1 = M registers 2 = C registers 3 = D registers
TBLADDRS	Word	Starting address number of the cam table. In case of D register table, this value is relative to the assigned data address at 'DATA31W'. TBLADDRS+DATA31W = Actual Starting Address	0～3276 If the end of the table is greater than 32767, the an error will be set. (The limit of M-reg is MW00000 to MW29999)
ENGAGPOS	Long	Master position in counts to start the slave camming	0 to the machine cycle value defined in the MOD-ENG block
DISENGAG	Long	Master position in counts to stop the slave camming	0 to the machine cycle value defined in the MOD-ENG block
DATA32W	Address	Address of the first working register.	Thirty-two words of register space are used by this function.

<CAM_RECV> Block Fault Condition

The following table outlines several situations that may cause an error, and will turn on the blocks "Error" output bit. The block "Error" output will cleared if the *EXECUTE* bit goes low, but the Error ID (MW3**81 and MW3**82) will remain in the RDA. To reset the Error ID, use the Alarm Reset Function Block.

Note that each axis has its own Error ID stored in its RDA axis section, offset by 200 for each axis. Example: Axis#1 stores to MW30181, Axis #20 stores to MW30381, etc (note that Master/Slave pair is irrelevant).

Internal Fault bit	Cause	Note
msinrng AB000010	Master/Slave pair input out of range	Value on input must be from 1 to 10. If this bit goes low while execute is high an error will occur. This will also generate an error in the RDA at MB3**813
ctInrng AB000011	Table type input is out of range.	Value on input must be from 1 to 3. If this bit goes low while execute is high an error will occur. This will also generate an error in the RDA at MB3**813.
addrInRng AB000012	Final address of table is beyond the range of the register.	Checks that final address of the table (Start address + size) does not exceed max register size (29999 for M, 16482 for D and C type registers). If this bit is low on the rising edge of execute an error will occur. This will also generate an error in the RDA at MB3**813.
mstErr AB00001A	Last value in Cam table is greater then machine cycle.	If the last master value in the cam table is greater then the machine cycle an error will occur. This will also generate an error in the RDA at MB3**81C.
Direr AB00001F	Direction not allowed by SVON	The direction commanded was not enabled by SVON function. This will also generate an error in the RDA at MB3**814.
CmndError AB00001E	Another block took control of axis	If another axis controlling block while this block is running. It will take control from this function. This does not set the RDA Error ID.
AxisInErr AB000050	The axis number entered on the input is not an acceptable value	The function blocks can only control 1 to 16 axis. Any value greater or smaller then this will cause an error. This does not set the RDA Error ID.

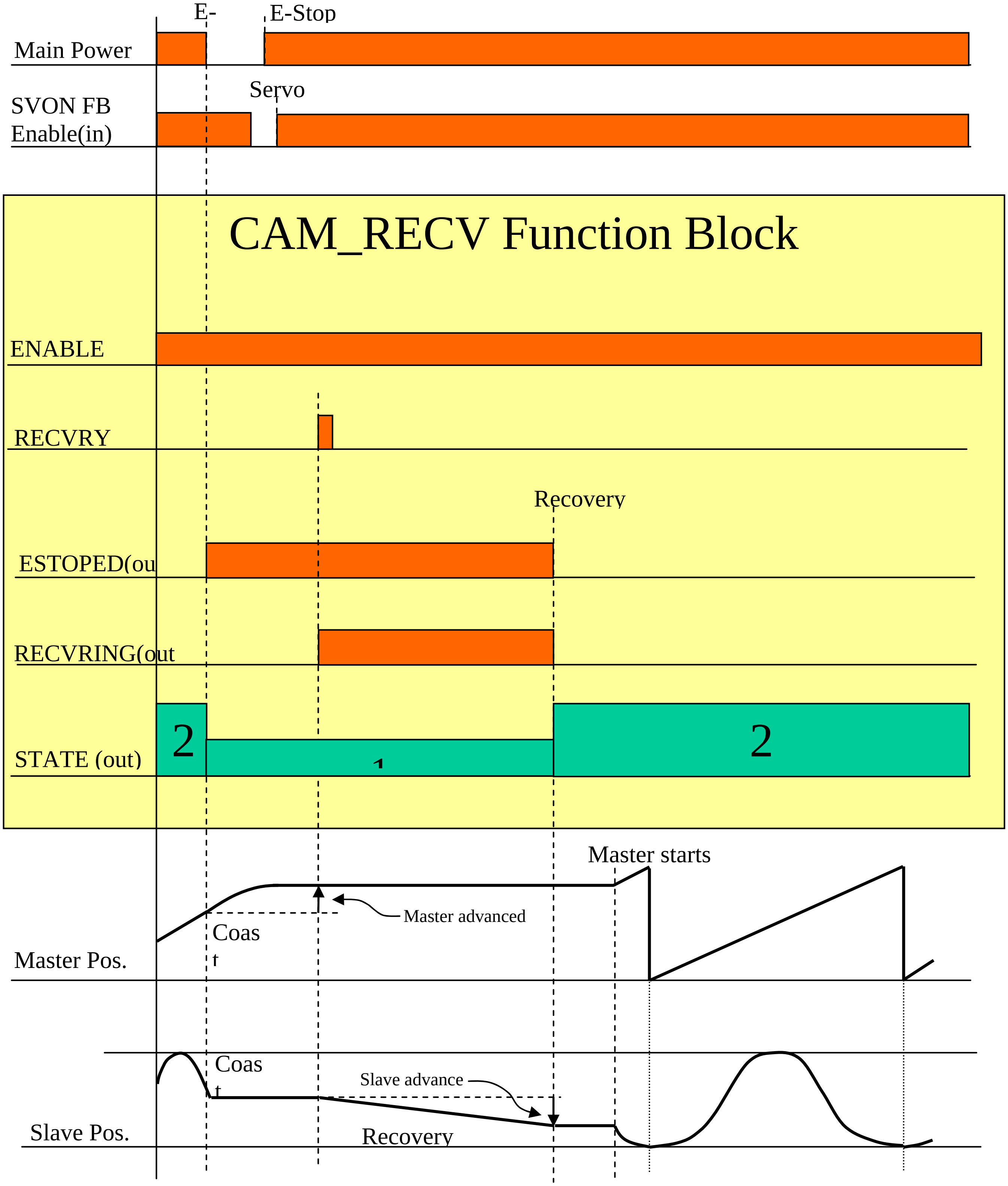
<CAM_RECV> Working Register

This table outlines the data in the thirty-three registers used by the function block. There is not usually any need for the user to access any of these bits directly.

Register No.	Type	Name	Description
AW00000			
Bit 0	IN	EXECUTE	*ENABLE* input (XB000000)
Bit 1	IN	CamIn	*CAMIN* input (XB000001)
Bit 2	IN	CamOut	*CAMOUT* input (XB000002)
Bit 3	Working	mRegister	Indicates table is in M register.
Bit 4	Working	cRegister	Indicates table is in C register.
Bit 5	Working	dRegister	Indicates table is in D register.
Bit 6	Working	engaging	On when waiting for master to reach engage position.
Bit 7	OUT	running	Directly controls YB000000 (*RUNNING* Output)
Bit 8	OUT	endOfTrvl	Directly controls YB000001 (*ENDPROFI* Output)
Bit 9	OUT	Mistake	Directly controls YB000002 (ERROR Output)
Bit A	Working	oneshotA	Reserved
Bit B	Working	firstPass	One shot coil for initializing axis when CAMIN input goes high.
Bit C	Working	oneshotC	Reserved
Bit D	IN	runInit	One shot coil for initializing axis when engage position is reached.
Bit E	IN	oneshotE	Reserved
Bit F	Working	oneshotF	Reserved
AW00001			
Bit 0	Working	msinrng	Verifies Master/Slave input is in range.
Bit 1	Working	ctlnrng	Verifies Table Type input is in range.
Bit 2	Working	addrInRng	Verifies that the entire Cam Table is in the register and did not go beyond the register limits.
Bit 3	Working	disengaging	On when camming and CAMOUT input goes high.
Bit 4	Working	above_point	Reserved
Bit 5	Working	d_abovepos	Reserved
Bit 6	Working	oneWay	On when first and last value of cam table are not equal.
Bit 7	Working	exeInit	One shot coil on for first pass of block being executed.
Bit 8	Working	disengaged	One shot coil for when disengage position is reached and called for.
Bit A	Working	mstrErr	On if last master value in cam table is greater then master machine cycle.
Bit B	Working	posout	Reserved
Bit C	Working	backwords。	Reserved
Bit D	Working	negOut	Reserved

Bit E	Working	cmndError	One shot coil for when another block takes control from this block.
Bit F	Working	direrr	On if a direction is commanded that is not enabled by the SVON block.
AW00002	Working	mstr_slave	MstrSlave [value of (XW00001) – 1]
AW00003	Working	filter1	Temporary values used to filter bits.
AW00004	OUT	camstate	Directly controls YW00001 (*STATE* Output).
AW00005			
Bit 0	Working	axisInErr	Goes high for one scan if Axis input is out of range.
AL00006	Working	yFirst	First position of slave in Cam table.
AL00008	Working	yLast	Last position of slave in Cam table.
AL00010	Working	xLast	Last position of Master in Cam table.
AL00012	Working	lastAddress	Final register address of Cam table.
AL00014	Working	Window	Tolerance value for engage and disengage position. Divide XLAST by 120. IE – this window is 0.8% of the machine cycle.
AL00016	Working	masterpos	Value of Master after CAM shift.
AL00018	Working	engage_difference	Distance from engage or disengage point.
AL00020	Working	base_slave_pos	Reserved
AL00022	Working	fgnOut	Value returned from Cam table function generator
AL00024	Working	slave_Pos_Cmnd	Reserved
AL00026	Working	deltaY	Difference between first and last slave position in table, used if it's a one-way cam to offset the slave
AW00028			
Bit 0	Working	ESTOP_Latch	Latch E-stop Condition
Bit 1	Working	RECV_MODE	In Recovery Motion
Bit 2	Working	RECV_COMP	Recovery Motion Complete
Bit 3	Working	Oneshot283	Reserved
Bit 4	Working	Oneshot284	Reserved
Bit 5	Working	RECV_SHOT	Reset Error after Recovery motion is done
AW00029	Working	Timer	Positioning Completion Timer for E-Stop recovery
AW00030	Working	rdaMult	Value for address offset to locate proper RDA
AW00031	Working	Revision	Revision Level of Block.

CAM_RECV Time Chart



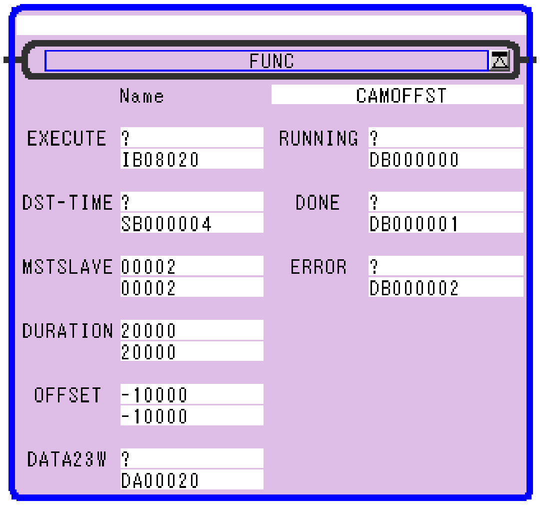
CAM OFFSET function Block

Function block for MP2000 series

<CAMOFFST> Function Block Summary

The Cam Offset (CAMOFFST) block is used to shift the value of the master (for synchronizing or dwell) while camming. The rising edge of the *EXECUTE* input bit will cause the *OFFSET* value to be added to the RDA offset at the rate determined by the *DURATION* input. *DST-TIME* input bit determines whether the shift duration is base upon distance the master travels or based on time in milliseconds.

Function Block Diagram



Name CAMOFFST

EXECUTE ? IB08020

DST-TIME ? SB000004

MSTSLAVE 00002 00002

DURATION 20000 20000

OFFSET -10000 -10000

DATA23W ? DA00020

RUNNING ? DB000000

DONE ? DB000001

ERROR ? DB000002

<CAMOFFST> Function Block Operation Notes

- Rising edge of EXECUTE input initiates block operation, and all block input values are read once.
- To use the function block, the *EXECUTE* bit must be held ON. If the *EXECUTE* bit goes off during operation, the block will stop executing (will not complete the change), the offset up to that point will remain, and all outputs will be set to zero.
- The *RUNNING* output bit will be held high until the Offset is completed, *EXECUTE* goes low, or an *Error* occurs.
- The *DST-TIME* bit must be set to TRUE to use the master position change or FALSE for the time duration.
- Distance mode note: Be sure to set the duration in the same direction as the master movement. Example: If the block is in Distance mode and the master is moving in the negative direction, and the DURATION is set positive, it will subtract the relative offset and hence the resulting offset will not complete. As a result, the DONE output will not go on because the full move is not complete in the correct direction.
- The 'Block Running' value will appear in the Master/Slave pair section of the RDA RDA(MW56**6).
- Issuing a STOP block on the slave axis will have no affect on the OFFSET block execution.
- Note that the master/slave pairs are separated by 50 words in the RDA (up to 10 pairs). Example: Master/Slave pair #1 starts at MW56000, Master/Slave pair #2 starts at MW56050, Master/Slave pair #3 starts at MW56100, etc.
- Twenty-three words are used as working registers for this function, starting at the address in *Data23W*.

<CAMOFFST> Input and Output Register Map

Output Registers

The following registers are used as outputs from the function block. They can be monitored by the LadderWorks program to check the execution of the function.

Output	Type	Description
RUNNING	Bit	Goes high while offset value is being updated in RDA and there are no errors
DONE	Bit	Goes high when offset change is complete in RDA
ERROR	Bit	Latches high if any block errors occur

Input Registers
The following registers are used as inputs to the function block. They select the options and define the parameters that the user needs to make the function work as necessary.

Input	Type	Description	Range and state
EXECUTE	Bit	Block enable – Block cannot execute unless this is TRUE.	Rising edge initiates block TRUE – Block enable FALSE – Block disable, set all outputs to zero and off
DST-TIME	Bit	Selects master or time for offset duration	TRUE – Master position is used for duration FALSE – Time is used for duration.
M-S-PAIR	Word	Defines which Master-Slave pair is updated in RDA	1 to 10 value. Any other value will cause the error output to go on
DURATION	Long	Depends on DST-TIME setting. Either time in msec or distance master must travel to complete change in offset.	-2147483648 to 2147483647. If in time mode no negative numbers are allowed.
OFFSET	Long	Relative shift to be applied to the master. RDA will hold absolute offset value level (The RDA offset is ML56**4)	-2147483648 to 2147483647
DATA23W	Address	Address of the first working register.	Twenty-three words of register space are used by this function.

<CAMOFFST> Block Fault Condition

The following table outlines several situations that may cause an error, and will turn on the blocks "Error" output bit. The block "Error" output will cleared if the *EXECUTE* bit goes low, but the Error ID (MW3**81and MW3**82) will remain in the RDA. To reset the Error ID, use the Alarm Reset Function Block.
The following table outlines several situations that may cause an error.

Internal Fault bit	Cause	Note
Inrng1 AB00000E	MSTRSLV value out of range	Inrng1 must be high if execute is on or an error will occur. MSTRSLV must have a value from one to ten for this bit to be on. Sets RDA Error ID (MW30181) bit 3.

<CAMOFFST> Working Register

This table outlines the data in the eight registers used by the function block. There is not usually any need for the user to access any of these bits directly.

Register No	TYPE	Name	Description
AW00000			
Bit 0	IN	Execute	EXECUTE input (XB000000)
Bit 1	IN	Distmode	DST-TIME input (XB000001)
Bit 3	Working	Zerodiv	If duration is set to zero for one scan change, this turns on for one scan (one-shot)
Bit 6	Working	distShft_init	High for one scan to initialize for master position shifting if first pass is high and DISTMODE is high. To initialize for master position control of offset.
Bit 7	OUT	Rining	Directly controls YB000000 (*RUNNING* Output)
Bit 8	OUT	Done	Directly controls YB000001 (*DONE* Output)
Bit 9	OUT	Eeror	Directly controls YB000002 (*ERROR* Output)
Bit A	OUT	OneshotA	Tied to rising edge of EXECUTE input (AB000000)
Bit B	Working	FirstPass	High for first scan of EXECUTE being high
Bit C	Working	OneshotC	One shot coil for resetting outputs. Controlled by (AB00000A)
Bit D	Working	timeshft_init	High for one scan to initialize for time based shifting if first pass is high and DISTMODE is low.
Bit E	Working	Inrng1	On if EXECUTE is on and MSTSLAVE (XW000001) value is in range
AW00001	Working	scans_per_shift	Number of scannings necessary to complete shift of base at time
AW00002	Working	iStore	Storage of I
AW00003	Working	mstrSlv	MSTSLAVE input (XW000001) -1
AL00004	Working	maxShift_scan	Maximum counts allowed to shift per scan on a time based scan.
AL00006	Working	posLAUlim	Positive definite of maxShift_scan
AL00008	Working	negLAUlim	Negative value of maxShift_scan
AL00010	Working	total_Shifted	Amount CAM Offset has changed during the execution of the block.
AL00012	Working	shiftvalue	Value to change the RDA (ML56**4) CAM offset for that scan.
AL00014	Working	startmaster	Value of Master counter at start of block for position based move.
AL00016	Working	masterDiff	How much the master has moved since the execution of the block.
AL00018	Working	desired_shift	Total amount needed to be shifted for the master position mode.
AL00020	Working	timemod	Reserved.
AW00022	Working	Revision	Revision Level of the function block.

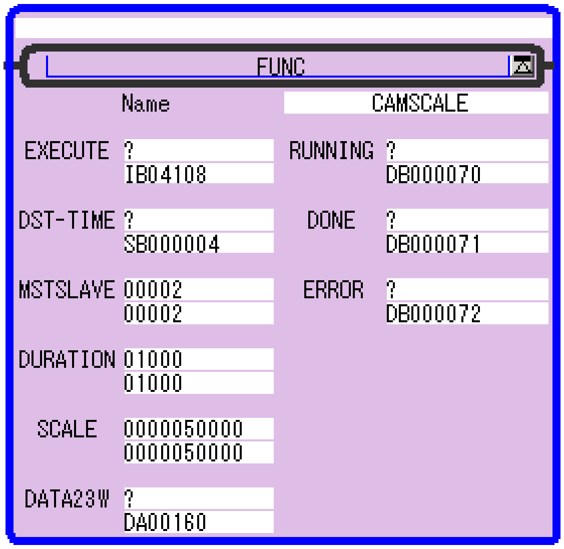
CAM SCALE function

Function block for MP2000 series

<CAMSCALE> Function Block Summary

The Cam Scale “CAMSCALE” block is used to scale the value of the Cam table output during camming. The rising edge of the *EXECUTE* input bit will cause the *OFFSET* value to be added to the RDA offset at the rate determined by the *DURATION* input. *DST-TIME* input bit determines whether the shift duration is base upon distance the master travels or based on time in milliseconds.1 = 0.01% for scaling.

Function Block Diagram



FUNC

Name CAMSCALE

EXECUTE	? IB04108	RUNNING	? DB000070
DST-TIME	? SB000004	DONE	? DB000071
MSTSLAVE	00002 00002	ERROR	? DB000072
DURATION	01000 01000		
SCALE	0000050000 0000050000		
DATA23W	? DA00160		

<CAMSCALE> Function Block Operation Notes

- **IMPORTANT NOTE: If the CamScale value in the RDA (ML56**8) is zero the slave will not move.**
- Rising edge of EXECUTE input initiates block operation, and all block input values are read once.
- To use the function block, the *EXECUTE* bit must be held ON. If the *EXECUTE* bit goes off during operation, the block will stop executing (will not complete the change), the offset (or relative scale change) up to that point will remain, and all outputs will be set to zero.
- The *RUNNING* output bit will be held high until the Offset (or relative scale change) is completed, *EXECUTE* goes low, or an *Error* occurs.
- The *DST-TIME* bit must be set to TRUE to use the master position change or FALSE for the time duration.

- Distance mode note: Be sure to set the duration in the same direction as the master movement. Example: If the block is in Distance mode and the master is moving in the negative direction, and the DURATION is set positive, it will subtract the relative offset and hence the resulting offset (or relative scale change) will not complete. Hence, because the master moves in the wrong direction, the cam scale change will not complete and the DONE output will not turn on.
- The SCALE input is an absolute value (RDA holds the absolute scale ML56**8). Note: 10000 is equal to a multiplier of 1 in the Cam block.
- The Block Running value will appear in the Master/Slave pair section of the RDA (MW56**6).
- Issuing a STOP block on the slave axis will have no affect on the OFFSET block execution.
- Note that the master/slave pairs are separated by 50 words in the RDA (up to 10 pairs). Example: Master/Slave pair #1 starts at MW56000, Master/Slave pair #2 starts at MW56050, Master/Slave pair #3 starts at MW56100, etc.
- Twenty-three words are used as working registers for this function, starting at the address in *Data23W*.

<CAMSCALE> Input and Output Register Map

Output Registers

The following registers are used as outputs from the function block. They can be monitored by the LadderWorks program to check the execution of the function.

Output	Type	Description
RUNNING	Bit	Goes high while offset scale value is being updated in RDA and there are no errors
DONE	Bit	Goes high when offset scale change is complete in RDA
ERROR	Bit	Goes high if any block errors occur

Input Registers
The following registers are used as inputs to the function block. They select the options and define the parameters that the user needs to make the function work as necessary.

INPUT	TYPE	Content	Range and state
EXECUTE	Bit	Block enable – Block cannot execute unless this is TRUE.	Rising edge initiates block TRUE – Block enable FALSE – Block disable, set all outputs to zero and off
DST-TIME	Bit	Selects master or time for offset duration	TRUE – Master position is used for duration FALSE – Time is used for duration
M-S-PAIR	Word	Defines which Master-Slave pair is updated in RDA	1 to 10 value. Any other value will cause the error output to go on
DURATION	Long	Depends on DST-TIME setting. Either time in msec or distance master must travel in counts to complete change in offset.	-2147483648 to 2147483647 If in time mode no negative numbers are allowed.
SCALE	Long	Absolute scale to be applied to the slave. RDA will hold current scale level. (RDA offset is ML56**8)	2147483648～2147483647
DATA23W	Address	Address of the first working register.	Twenty-three words of register space are used by this function.

<CAMSCALE> Block Fault Condition:

The following tables might cause the error, and outline some situations. If the EXECUTE input bit is turned off, the value is cleared.

Internal Fault Bit	Cause	Note
inrng1 AB00000E	MSTRSLV value out of range	Inrng1 must be high if execute is on or an error will occur. MSTRSLV must have a value from one to ten for this bit to be on. Sets RDA Error ID (MW3**81) bit 3.

<CAMSCALE> Working Registers

This table outlines the data in the eight registers used by the function block. There is not usually any need for the user to access any of these bits directly.

Register No.	Type	Name	Description
AW00000			
Bit 0	IN	execute	EXECUTE input (XB000000)
Bit 1	IN	distmode	DST-TIME input (XB000001)
Bit 3	Working	zerodiv	If duration is set to zero for one scan change, this turns on for one scan (one-shot)
Bit 6	Working	distShft_init	High for one scan to initialize for master position shifting if first pass is high and DISTMODE is high. To initialize for master position control of offset.
Bit 7	OUT	running	Directly controls YB000000 (*RUNNING* Output)
Bit 8	OUT	complete	Directly controls YB000001 (*DONE* Output)
Bit 9	OUT	error	Directly controls YB000002 (*ERROR* Output)
Bit A	Working	oneshotA	Reserved
Bit B	Working	firstPass	High for first scan of EXECUTE being high
Bit D	Working	timeshft_init	High for one scan to initialize for time based shifting if first pass is high and DISTMODE is low.
Bit E	Working	inrng1	On if Execute is on and MSTSLAVE (XW000001) value is in range
AW00001	Working	scans_per_shift	Number of scans needed to complete a time based shift.
AW00002	Working	iStore	Pass through value for i.
AW00003	Working	mstrSlv	MSTSLAVE input (XW000001) – 1.
AL00004	Working	maxShift_scan	Maximum counts allowed to scale per scan on a time based scan.
AL00006	Working	posLAUlim	Positive value of maxShift_scan
AL00008	Working	negLAUlim	Negative value of maxShift_scan
AL00010	Working	total_Shifted	Amount Scale Offset has changed during the execution of the block.
AL00012	Working	shiftvalue	Relative value to change the RDA (ML03**8) cam scale for that scan.
AL00014	Working	startmaster	Value of Master counter at start of block for position based move.
AL00016	Working	RelativeChange	Difference between set scale value and current scale value in RDA.
AL00018	Working	desired_shift	Total amount needed to be scaled for the master position mode.
AL00020	Working	timemod	Reserved.
AW00022	Working	Revision	Revision Level of the function block.

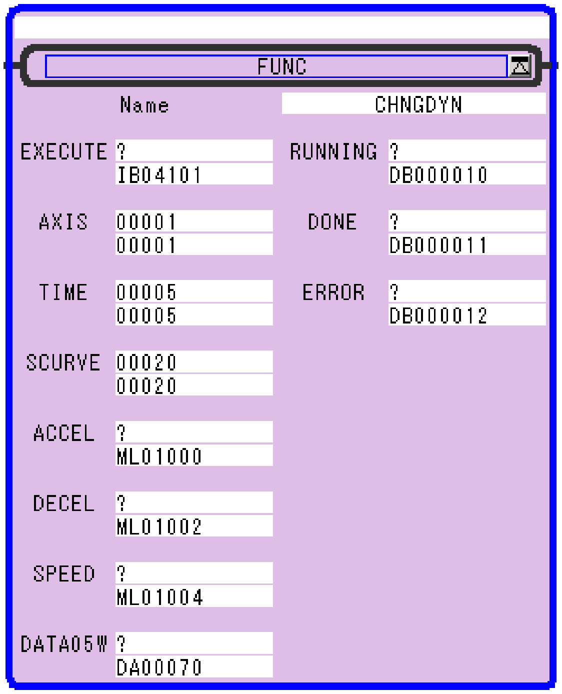
CHANGE DYNAMICS function

Function block for MP2000 series

<CHNGDYN> Function Block Summary

This function block sets acceleration, deceleration, S-Curve and the speed. Each parameter is set in the first execution scan.

Function Block Diagram



FUNC

Name CHNGDYN

EXECUTE	? IB04101	RUNNING	? DB000010
AXIS	00001 00001	DONE	? DB000011
TIME	00005 00005	ERROR	? DB000012
SCURVE	00020 00020		
ACCEL	? ML01000		
DECEL	? ML01002		
SPEED	? ML01004		
DATA05W	? DA00070		

<CHNGDYN> Function Block Operation Notes

- Rising edge of EXECUTE input initiates block operation, and all block input values are read once.
- To use the function block, the *EXECUTE* bit must be held ON. If the *EXECUTE* bit goes off during operation, the block will finish move but all outputs will be set to zero.
- The *RUNNING* output bit will be held high parameter change is completed, *EXECUTE* goes low or an *Error* occurs.
- If any of the inputs (absolute value) are greater then the Maximum value in the RDA an Error will occur.
- When parameter setting is completed, the DONE output is turned on.
- Each value (SPEED, ACCEL, DECEL, and SCURVE) is also set to the corresponding RDA.
- S-Curve value is for the running average filter. S-curve outputs literally “S" reference on acceleration and the deceleration.

- Five words are used as working registers for this function, starting at the address in *Data05W*.

<CHNGDYN> Input and Output Register Map

Output Registers

The following registers are used as outputs from the function block. They can be monitored by the LadderWorks program to check the execution of the function.

OUTPUT	TYPE	DESCRIPTION
RUNNING	BIT	THE FUNCTION BLOCK EXECUTES (*O) IN.
DONE	BIT	THERE ARE NEITHER EXECUTION COMPLETION (ALL THE VALUES ARE SET) NOR ERROR GENERATION.
ERROR	BIT	ERROR GENERATION (THE INPUT VALUE IS OUTSIDE A SET RANGE)

Input Registers

Input	Type	Description	Range and state
EXECUTE	Bit	Block enable – see block operation notes.	Rising edge initiates block TRUE – continue executing FALSE – see block operation notes
AXIS	Word	Axis number related to the block.	1～16
TIME	Word	Timer setting	1～32767[sec]
SCURVE	Long	S-Curve value to store in RDA and controller	Zero to Max value is limited to max value in RDA (ML3**34). Value is stored in milliseconds
ACCEL	Long	Acceleration value to store in RDA and controller.	Zero to Max value is limited to max value in RDA (ML3**22). Value is stored in counts/sec^2
DECEL	Long	Deceleration value to store in RDA and controller.	Zero to Max value is limited to max value in RDA (ML3**28). Value is stored in counts/sec^2
SPEED	Long	Velocity value to store in RDA and controller.	Max absolute value is limited to max value in RDA (ML3**28). Value is stored in counts/sec.
DATA05W	Address	Address of the first working register.	Five words of register space are used by this function.

<CHNGDYN> Block Fault Condition

The following table outlines several situations that may cause an error, and will turn on the blocks “Error” output bit. The block “Error” output will cleared if the *EXECUTE* bit goes low, but the Error ID (MW3**81 and MW3**82) will remain in the RDA. To reset the Error ID, use the Alarm Reset Function Block.
Note that each axis has its own Error ID stored in its RDA axis section, offset by 200 for each axis. Example: Axis#1 stores to MW30181, Axis #2 stores to MW30381, etc.

Internal Fault Bit	Cause	Note
JerkOut AB000001	S-curve value is not within acceptable range.	Bit must be on or error will occur. Sets RDA Error ID (MW3**81) bit 3 on if error state exists.
AccelOut AB000002	Acceleration value is not within acceptable range.	Bit must be on or error will occur. Sets RDA Error ID (MW3**81) bit 3 on if error state exists.
DecelOut AB000003	Deceleration value is not within acceptable range	Bit must be on or error will occur. Sets RDA Error ID (MW3**81) bit 3 on if error state exists.
VelOut AB000004	Speed value is not within acceptable range.	Bit must be on or error will occur. Sets RDA Error ID (MW3**81) bit 3 on if error state exists.
axisInErr AB000005	The axis number entered on the input is not an acceptable value	The function block can only control 1 to 16 axis. Any value greater or smaller then this will cause an error. This does not set the RDA Error ID.

<CHNGDYN> Working Register

This table outlines the data of six registers used in the function block. The user is, and there is not a necessity and either is usually no what directly accessed any of these bits.

Register No	TYPE	Name	Content
AW00000			
Bit 0	IN	execute	*Execute* input (XB000000)
Bit 1	Working	jerkOut	Verifies that the *SCURVE* input (XW00001) is in range.
Bit 2	Working	accelOut	Verifies that the *ACCEL* input (XW00002) is in range.
Bit 3	Working	decelOut	Verifies that the *DECEL* input (XW00003) is in range.
Bit 4	Working	velOut	Verifies that the *SPEED* input (XW00004) is in range.
Bit 5	Working	axisInErr	Goes high for one scan if Axis input is out of range.
Bit 7	OUT	Running	Directly controls YB000000 (*Running* Output)
Bit 8	OUT	done	Directly controls YB000001 (*Complete* Output)
Bit 9	OUT	Mistake	Directly controls YB000002 (*BlkFault* Output)
Bit A	Working	osExecute	Reserved
Bit B	Working	firstPass	One shot coil for initializing block on the rising edge of the Execute bit.
AW00001			

Bit 0	Working	holdCntrl	Goes High and remains high until block is complete or timer times out.
Bit 1	Working	Reserved	Reserved
Bit 2	Working	Reserved	Reserved
Bit 3	Working	Reserved	Reserved
Bit 4	Working	Reserved	Reserved
Bit 5	Working	Reserved	Reserved
Bit 6	Working	Reserved	Reserved
Bit 7	Working	Time-out	Goes high when timer (TIME) runs out.
Bit 8	Working	clrComnd	Reserved
Bit F	Working	oneshotF	Reserved
AW00002	Working	Counter	Counter register for time out timer.
AW00003	Working	rdaMult	Value for address offset to locate proper RDA
AW00004	Working	Revision	Revision level of block.

GEAR function

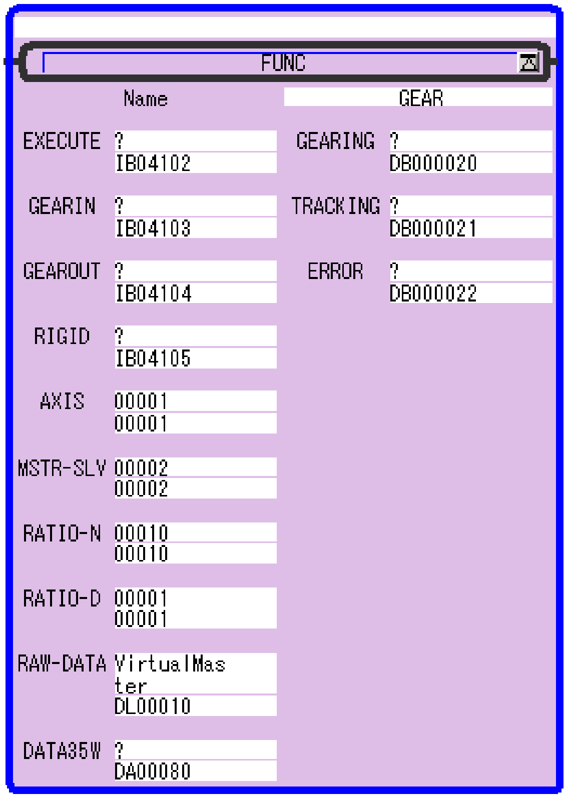
Function block for MP2000 series

<GEAR> Function Block Summary

The “Gear” block is used to slave the given axis to any defined real or virtual master pulse data. There are two gearing modes of operation, Rigid Gearing and Non-Rigid Gearing. In either mode, when the slave is successfully following the master position at the geared ratio (speed match event), the tracking output will turn on. Other features of ‘slave stop’ or ‘slave continue at current speed’, are also available when disengaging gearing. Refer to the attached timing chart for details.

> **Rigid** gearing will phase lock the slave, following position to a specified ratio of the master position data, at the exact moment of the gear engage signal. If the master pulse is already moving, the slave will accelerate at a rate set by the RDA, and over-speed to catch up to the master position phase. The over-speed is limited to the maximum velocity set in the RDA.
>
> **Non-Rigid** gearing will cause the slave to follow the speed of the master. If the master pulse is already moving, the slave will accelerate at a rate set by RDA, but will lag in position equal to the area under the acceleration curve (it will remain phase locked *after* speed is matched and in tracking mode).

Function Block Diagram



FUNC

Name	GEAR		
EXECUTE	? IB04102	GEARING	? DB000020
GEARIN	? IB04103	TRACKING	? DB000021
GEAROUT	? IB04104	ERROR	? DB000022
RIGID	? IB04105		
AXIS	00001 00001		
MSTR-SLV	00002 00002		
RATIO-N	00010 00010		
RATIO-D	00001 00001		
RAW-DATA	VirtualMas ter DL00010		
DATA35W	? DA00080		

<GEAR> Function Block Operation Notes

- Slave Offset and Change Dynamics function blocks may be useful when using this block.
- Rising edge of EXECUTE input initiates block operation, and several block input values are read once. These inputs are: RIGID, AXIS, MSTR-SLV. This event will also verify & set the ACCEL & DECEL values in the servopack to zero. Note that this only means that the MP controller will control the ACCEL/DECEL rates as set in the RDA (acceleration time (ML3**46) and deceleration time (ML3**48)) and the servopack will not limit the accel/decel.
- **IMPORTANT NOTE:** Gearing sets Acceleration and Deceleration in the servopack to 0 ms. (acceleration time (ML3**46) →(OL**36), →(Pn80B), and deceleration time (ML3**48) →(OL**38) →(Pn80E)). In Gearing the Accel/Decel rates are controlled inside the Gearing block. After the Gearing Block is used and its motion has stopped (motion command IW**08=0), before initiating any other motion blocks, a Change Dynamics block must be used to restore the Accel and Decel values in the servopack. This will restore the expected operation of the other motion blocks.
- The following inputs are always read when EXECUTE bit is high: GEARIN, GEAROUT, RATIO-N, RATIO-D, RAW-DATA.
- To use the function block, the *EXECUTE* bit must be held ON. If the *EXECUTE* bit goes OFF while axis is gearing, the block will set the axis in speed mode (Motion command 7(constant speed sending)) at the speed it was going at that moment, but all outputs will be set to zero.
- If the GEAROUT bit goes high while the execute bit is high, and the block is gearing, then the axis will decelerate to a stop, at the set deceleration in the RDA.
- The Slave offset value in the RDA is set to zero when the gearing block is engaged.
- If the RIGID input is set high, the slave axis will be set in a phase lock mode at GEARIN timing. The slave will over-speed (limited to max speed of slave) to catch up in position if necessary, when the slave axis matches the (master phase position)*(gear ratio), and (master speed)*(gear ratio), the tracking bit will turn on. See graph below.
- If the RIGID input is set low, the slave axis will only speed match. The slave axis will ramp up to speed, when the slave axis matches speed of the master, the tracking bit will turn on.
- The *GEARING* output bit will be held high as long as the EXECUTE input is held high and the GEAROUT bit stays low, and no other block takes control.
- The maximum speed the slave axis can operate at is limited by the RDA Maximum speed setting for that axis. If the master pulse input commands the slave to go faster then this speed, the axis will travel at the max. speed, set the ERROR output to high, and set the TRACKING output to low. The error bit will return to off state when commanded speed becomes less than max velocity setting. An error bit will be registered in the RDA. Note that this condition does not occur during acceleration over-speed for rigid gearing.

- Thirty-five words are used as working registers for this function, starting at the address in *Data35W*.

<GEAR> Input and Output Register Map

Output Registers

The following registers are used as outputs from the function block. They can be monitored by the LadderWorks program to check the execution of the function.

Output	Type	Description
GEARING	Bit	This bit is ON when gearing is active in the slave axis
TRACKING	Bit	This bit is ON when the slave is following the commanded phase and speed (multiplied by gear ratio) of the Master
ERROR	Bit	Latches high if any error occurs in block (see table below) or on the servo axis (see IL**04). This is Reset if EXECUTE bit goes low.

Input Registers

The following registers are used as inputs to the function block. They select the options and define the parameters that the user needs to make the Gear function work as necessary.

· Gear ratio = (RATIO-N/RATIO-D)

· Example: for slave to follow at half the speed of the master, set RATIO-N=1 and RATIO-D=2.

Input	Type	Description	Range and State
EXECUTE	Bit	Block enable – Block cannot execute unless this is TRUE. FALSE clears all outputs to zero or FALSE.	Rising edge initiates block TRUE – Block enable FALSE – Block disable, see block operation note for additional info
GEARIN	Bit	Sets the axis in gearing mode with the specified gear ratio.	Always read when EXECUTE=high RISING EDGE- If the execute bit is high this will set the axis to be in gearing mode.
GEAROUT	Bit	Takes the axis out of gearing mode and stops the axis with the set deceleration value from the RDA	Always read when EXECUTE=high RISING EDGE- If the execute bit is high this will set the axis to be in gearing mode.
RIGID	Bit	Determines weather the block will gear with phase lock or strictly speed matching.	Read only on EXECUTE rise edge TRUE – Rigid enable FALSE – Non-Rigid
AXIS	Word	Axis number related to the block.	Read only on EXECUTE rise edge 1 to 16
M-S-PAIR	Word	Master Slave Pair as defined by the RDA	Read only on EXECUTE rise edge 1 to 10 integer. This value is needed to match the value of the Slave offset function block.
RATIO-N	Word	Numerator value of gear ratio *	Always read when EXECUTE=high -32768 to 32767 integer.
RATIO-D	Word	Denominator value of gear ratio *	Always read when EXECUTE=high -32768 to 32767 excluding zero. A zero will cause an error
RAW-DATA	Long	Pulse counter of master	Always read when EXECUTE=high The (change in this value * RATIO-N) / RATIO-D will be the commanded change in the Slave.
DATA35W	**Address**	Address of the first working register.	Thirty-five words of register space are used by this function.

<GEAR> Block Error Conditions:

The following table outlines several situations that may cause an error, and will turn on the blocks "Error" output bit. The block "Error" output will cleared if the *EXECUTE* bit goes low.

Internal error bit	Cause	Attention
notTracking AB000003	IF the commanded change is faster then the axis is allowed to go	The max. speed set by the RDA can not be exceeded. This is the only error that will allow the axis to continue moving in this blocks control. This will also generate an error in the RDA at MB3**819.
inrng1 AB000004	Master/Slave pair input out of range	Value on input must be between 1 and 10. If this bit goes low while execute is high an error will occur. This will also generate an error in the RDA at MB3**813.
inrng2 AB000005	Ratio-D input is set to zero	Cannot have a zero in the denominator If this bit goes low while execute is high an error will occur. This will also generate an error in the RDA at MB3**813.
dirError AB00000C	Direction not allowed by SVON	The direction commanded was not enabled by SVON function. This will also generate an error in the RDA at MB3**814.
cmndErr AB00000E	Another block took control of axis.	If another axis controlling block is executed it will take control from this function. This does not set the RDA Error ID.
axisInErr AB000304	The axis number entered on the input is not an acceptable value	The function blocks can only control 1 to 16 axes. Any value greater or smaller then this will cause an error. This does not set the RDA Error ID.

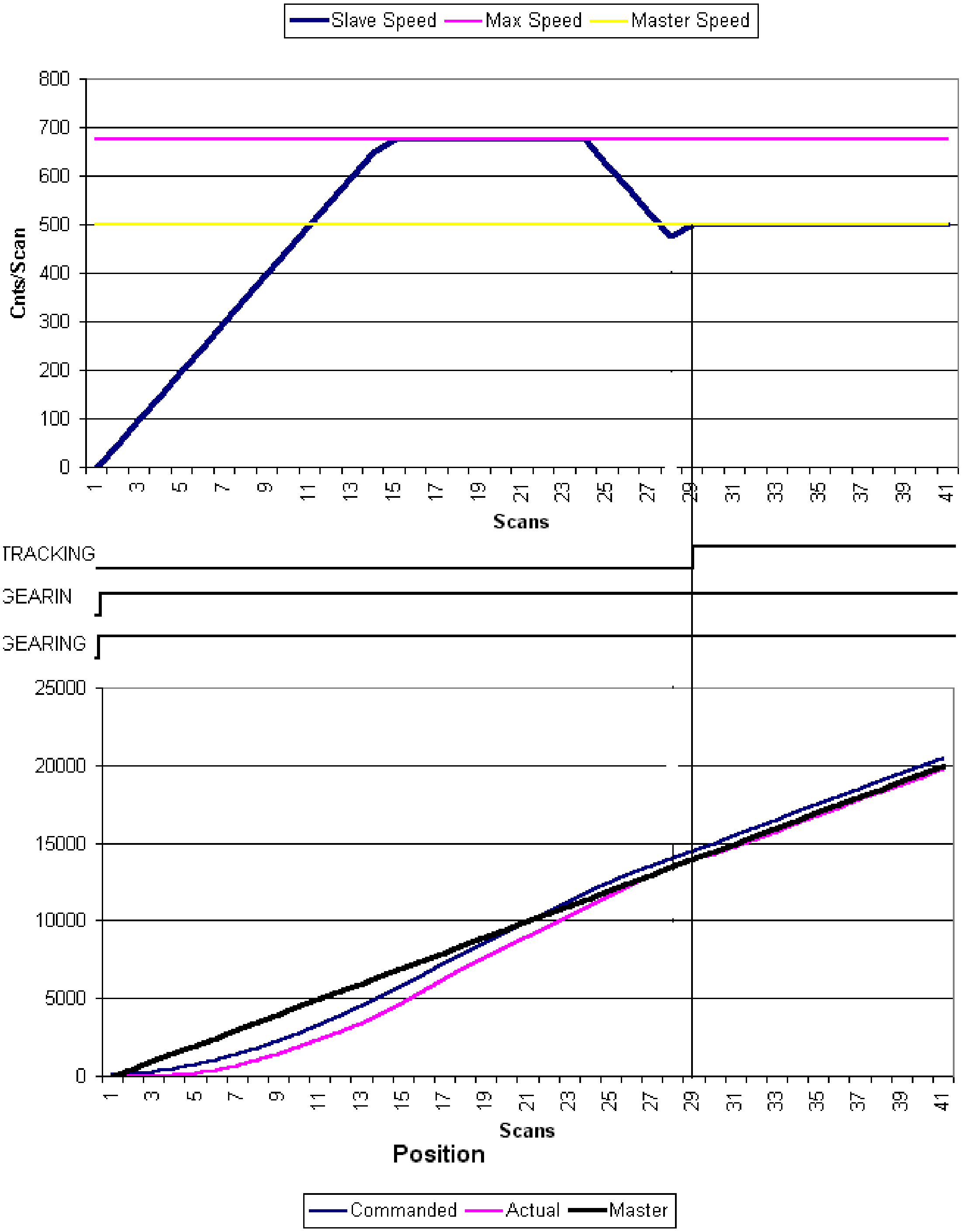
<GEAR> Working Registers

This table outlines the data in the thirty-four registers used by the Gear function block. There is not usually any need for the user to access any of these registers directly.

Register No	TYPE	Name	Content
AW00000			
Bit 0	IN	execute	*EXECUTE* input (XB000000)
Bit 1	IN	gearin	*GEARIN* input (XB000001)
Bit 2	IN	gearout	*GEAROUT* input (XB000002)
Bit 3	Working	notTracking	*notTracking* bit goes high when commanded speed exceeds max. allowed
Bit 4	Working	inrng1	*Goes high while EXECUTE input* is high and master/slave MSTR-SLV value is in range
Bit 5	Working	inrng2	*Goes high while EXECUTE input* is high and RATIO-D value is in range
Bit 6	Working	match	High when previous scanned command counts equal counts moved by slave.
Bit 7	OUT	gearing	Directly controls YB000000 (*RUNNING* Output)
Bit 8	Working	oneshot8	Reserved.
Bit 9	OUT	error	Directly controls YB000002 (*ERROR* Output)
Bit A	Working	oneshotA	Rising edge of GEARIN (AB000001)
Bit B	Working	firstPass	On for single scan to initiate gearing if all conditions are met. Sets OW**20=4
Bit C	Working	dirError	Indicates BSVON has not enabled axis to run in that direction
Bit E	Working	cmndErr	One shot Goes high when another block takes control away
Bit F	Working	inrtrack	On if commanded move is within speed limits
AL00004	Working	Master_Difer	Change in Master counts over one scan.
AL00006	Working	ModVal	Reserved.
AL00010	Working	oldoffset	Value of offset from RDA at last scan
AL00012	Working	delta_Offset	Change in offset (AL0010) since last scan
AL00014	Working	LastScan	Value of RAW-DATA (XL00004) at last scan
AL00016	Working	Prev_Move	Move commanded last scan
AL00022	Working	slv_oldPos	Store Target position (IL**10)
AL00024	Working	Max_Chnge	Maximum count change per scan based on Max speed limit in RDA
AL00026	Working	Max_limit	Positive max. count per scan allowed.
AL00028	Working	Min_Limit	Negative max. count per scan allowed.
AW00030			
Bit 0	Working	stopping	Goes high while in stopping mode (OW**08 = 7)
Bit 1	Working	stopped	Goes high for one scan once axis has stopped moving (set OW**08 = 0)
Bit 2	Working	speedset	Goes high for falling edge of EXECUTE (set OW**08 = 7) and (set OL**10 = current speed).
Bit 3	Working	onePass	Goes High on rising edge of Execute for one scan
Bit 4	Working	axisInErr	Axis number out of range

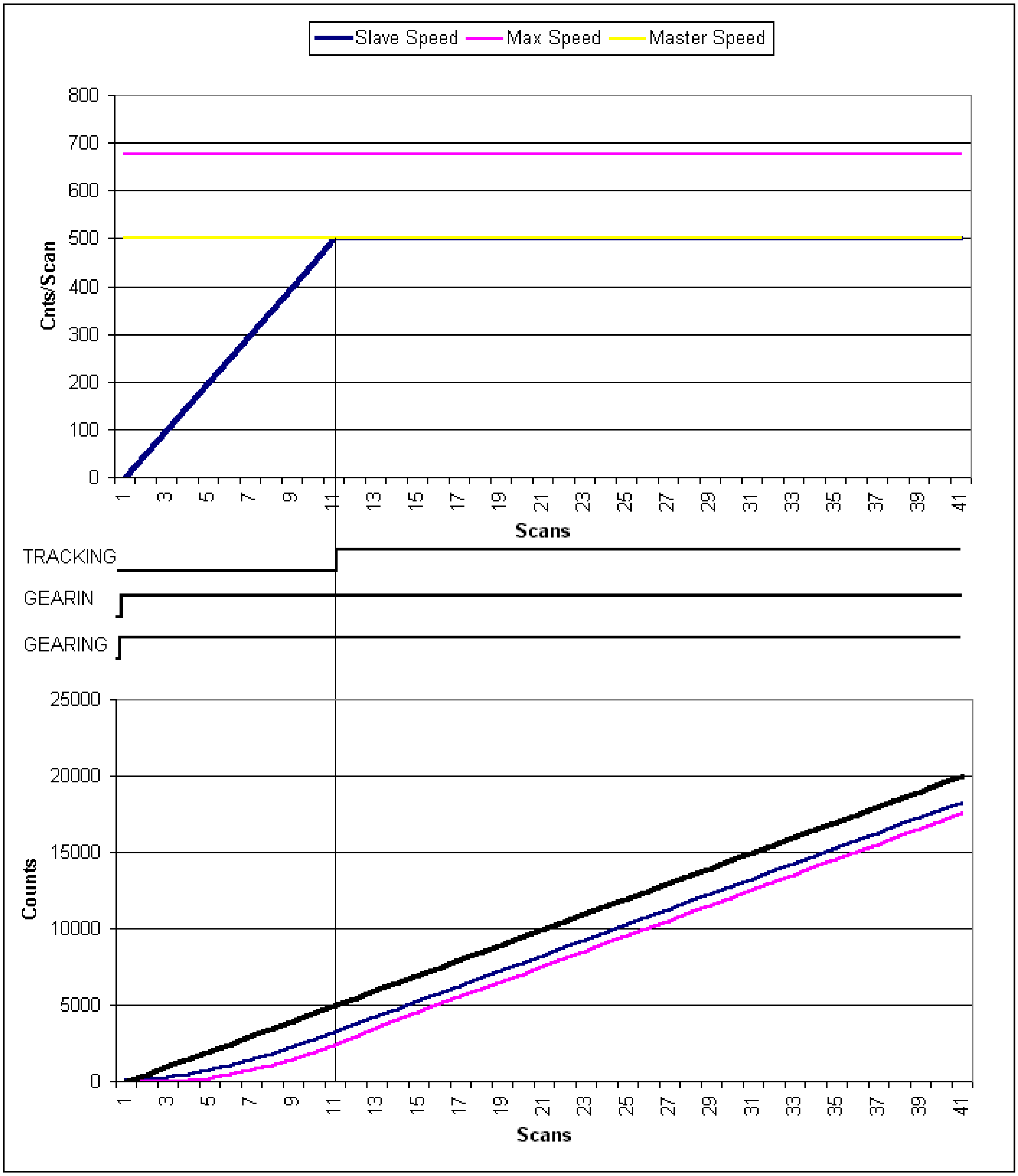
Bit 5	Working	RDA_Error	Reserved
Bit 6	IN	atStop	*RIGID* input (XB000003)
Bit 8	Working	speed_ok	High when speed is matched to master
Bit A	Working	oneshot30A	Reserved
Bit B	Working	oneshot30b	Reserved
Bit C	Working	oneshot30c	Reserved
Bit D	Working	oneswhot30d	Reserved
Bit F	Working	oneshot30f	Reserved
AW00033	Working	rDAmult	Value for address offset to locate proper RDA
AW00034	Working	Revision	Revision level of block.

<GEAR> Timing chart RIGID=ON



* Note that the commanded leads the master for PLC scan compensation. This is to insure that the slave actual position is precisely synchronized to master pulse reference.

<GEAR> Timing chart RIGID=OFF



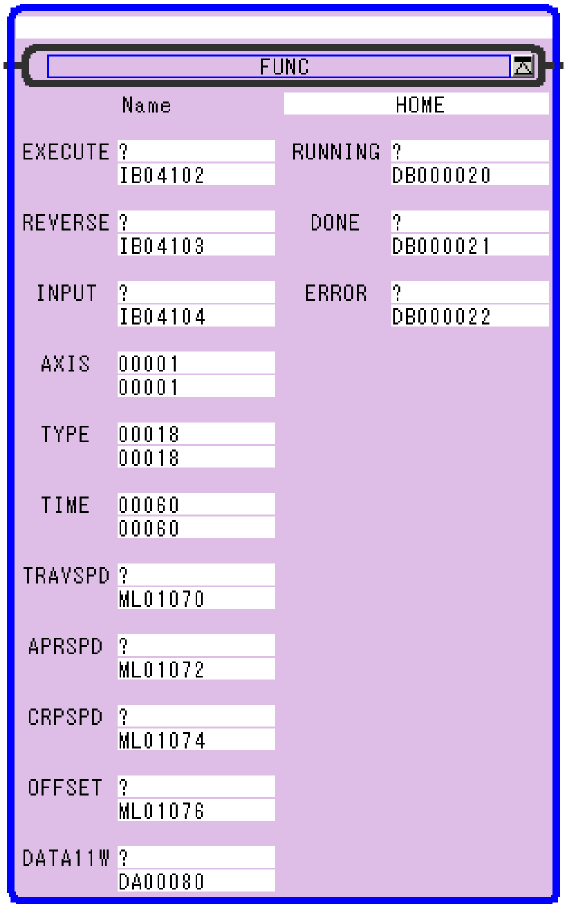
HOME function

Function block for MP2000 series

<HOME> Function Block Summary

The HOME function is used to home the axis. Home is a position marked by a home switch input(DEC), over travel (POT, NOT) input signals and C phase pulses of the motor or ZERO signal. There are 13 kinds of methods of the home position return (Refer to the chapter of "Home position return method" for details). Only the incremental encoder is supported to this block.

Function Block Diagram



FUNC

Name			HOME
EXECUTE	? IB04102	RUNNING	? DB000020
REVERSE	? IB04103	DONE	? DB000021
INPUT	? IB04104	ERROR	? DB000022
AXIS	00001 00001		
TYPE	00018 00018		
TIME	00060 00060		
TRAVSPD	? ML01070		
APRSPD	? ML01072		
CRPSPD	? ML01074		
OFFSET	? ML01076		
DATA11W	? DA00080		

<HOME> Function Block Operation Notes

- This is *not* valid for Absolute Encoder application, only for Incremental encoder.
- Rising edge of EXECUTE input initiates block operation, and all block input values are read once.
- To use the function block, the *EXECUTE* bit must be held ON. If the *Execute* bit goes off during operation, all outputs will be set to zero and axis will remain in whatever state it was in when the *Execute* bit went low.
- The *RUNNING* output bit will be held high until homing is completed or an *Error* occurs.
- The motor will move at TRVSPD (speed of the rapid feed) to find the falling edge of the home switch input (DEC signal, the input to CN1 connector/DEC) when the TYPE input (home position return method) is “0" (DEC+C phase pulse). The rotation direction is selected by the REVERSE input. The searching motion decelerates to APRSPD (approach speed) when the falling edge of the DEC signal is detected, and decelerates to CRPSPD (creep speed) with the rising edge of the C phase pulse. OFFSET distance (final distance) is advanced from C phase pulse and the position is set to the OFFSET value.
- When the home process is completed, the *DONE* bit will turn ON and the *RUNNING* bit will turn OFF. If the time limit defined by *TIMELIM* expires before the homing process is completed, a *block Error* is generated.
- Eleven words are used as working registers for this function, starting at the address in *Data11W*.

<HOME> Input and Output Register Map

Output Registers

The following registers are used as outputs from the function block. They can be monitored by the LadderWorks program to check the execution of the function.

Output	Type	Description
RUNNING	Bit	High while block has control and move is in progress.
DONE	Bit	Turns on when Homing process is completed
ERROR	Bit	Latches high if any error occurs in block (see table below) or on the servo axis (see IL**04).

Input Registers

The following registers are used as inputs to the function block. They select the options and define the parameters that the user needs to make the Home function work as necessary.

Input	Type	Description	Range and State
EXECUTE	Bit	Block enable – Block cannot execute unless this is TRUE.	Rising edge initiates block TRUE – Block enable FALSE – Block disable, set all outputs to zero and off
REVERSE	Bit	Determines direction of homing.	TRUE – Seek in reverse FALSE – Seek in forward
INPUT	Bit	“INPUT” signal for Homing type that uses it.	Only homing type 18 and 19 use the INPUT signal.
AXIS	Word	Axis number related to the block	1～16
TIME	Word	Maximum time to complete the homing, if timeout then ERROR.	Time in seconds. Range: 0 to 32767 integers
TRAVSPD	Long	Traverse speed of servo while searching for the home switch. In counts per second.	0 to maximum speed (ML3**12)
APRSPD	Long	Approach speed reference unit/sec	0 to maximum speed (ML3**12)
CRPSPD	Long	Creep rate reference unit/sec	0 to maximum speed (ML3**12)
Offset	Long	Final moving distance. And the position will be this value.	Position in encoder counts. Range -2^{31} to 2^{30} integers
DATA11W	Address	Address of the first working register.	Eleven words of register space are used by this function.

<HOME> Block Fault Conditions:

The following table outlines several situations that may cause an error, and will turn on the blocks “Error” output bit. The block “Error” output will cleared if the *EXECUTE* bit goes low, but the Error ID (MW3**81) will remain in the RDA. To reset the Error ID, use the Alarm Reset Function Block.

Note that each axis has its own Error ID stored in its RDA axis section, offset by 200 for each axis. Example: Axis#1 stores to MW30181, Axis #2 stores to MW30281, etc.

Internal error bit	Cause	Attention
axisInErr AB000006	The axis number entered on the input is not an acceptable value	The function blocks can only control 1 to 16 axes. Any value greater or smaller then this will cause an error. This does not set the RDA Error ID.
cmndErr AB000007	Another block took control of the axis.	If the block looses control of the axis while running an error will occur. This does not set the RDA Error ID.
errStop AB000008	Motion commanded and servo is disabled	If block is running and the servo is disabled the error bit will be set. Sets RDA Error ID (MW3**81) bit B on if error state exists.
OTAlarm AB00000C	Over travel Alarm	An over travel bit went low and *REVONOT* was FALSE. No home switch was found. Sets RDA Error ID (MW3**81) bit 0 on if error state exists.
VEL_ERROR AB00000D	Either the speed of rapid feed, approach speed or creep speed is not acceptable value.	Turns on when the set value is not acceptable. Sets RDA Error ID (MW3**81) bit 3 on if error state exist.
TimeOver AB00000F	Time Limit reached	The total time defined by the *TIMELIM* input word was exceeded. Sets RDA Error ID (MW3**81) bit 2 on if error state exists.

<HOME> Homing Type

(1) Homing Type Option

The following table shows thirteen Homing Type. Please select the best one for the machine.

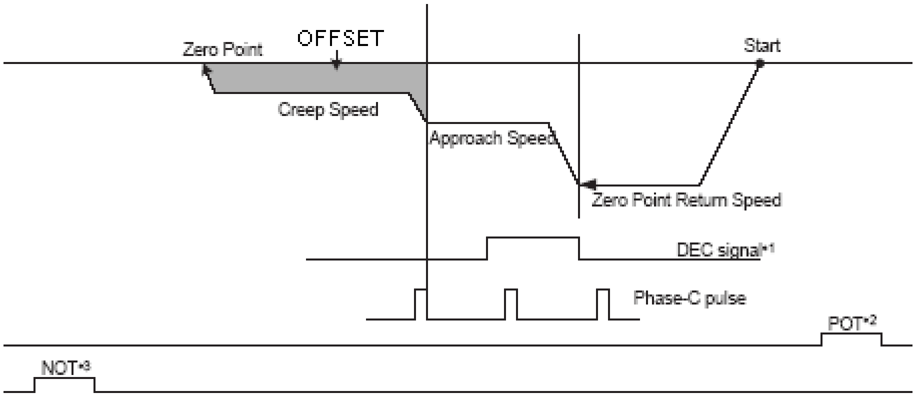
TYPE	Name	Method	Note
0	DEC1+C phase pulsed system	Three step deceleration method with deceleration limit switch and C phase pulse.	DEC1 signal: DEC signal of the servo amplifier.
1	ZERO signaling system	Homing method with ZERO signal.	ZERO signal: EXT1 signal of the servo amplifier.
2	DEC 1+ZERO signaling system	Three step deceleration method with deceleration limit switch and ZERO signal	DEC1 signal: DEC signal of the servo amplifier. ZERO signal: EXT1 signal of the servo amplifier.
3	C phase pulsed system	Homing method with C phase pulse	
11	C phase pulse	Method only with C phase pulse	
12	POT & C phase pulse	Method with positive OT signal and C phase pulse	POT: POT signal of the servo amplifier.
13	POT	Method only with positive OT signal	POT: POT signal of the servo amplifier. This method is not applicable if repeatability is required.
14	HOME LS&C phase pulse	Method with HOME signal and C phase pulse	HOME: EXT1 signal of the servo amplifier.
15	HOME LS	Method only with HOME signal	HOME: EXT1 signal of the servo amplifier.
16	NOT&C phase pulse	Method with reverse OT signal and C phase pulse	NOT: NOT signal of the servo amplifier.
17	NOT	Method only with reverse OT signal	NOT: NOT signal of the servo amplifier. This method is not applicable if repeatability is required.
18	INPUT & C phase pulse	Method with INPUT signal and C phase pulse	INPUT: INPUT signal
19	INPUT	Method only with INPUT signal	INPUT: INPUT signal This method is not applicable if repeatability is required.

(2) Homing operation and parameters

The motion after the function is executed and parameters to be set at the execution of the function are shown below.

(a) DEC1+C phase pulse system

The movement starts at the speed of the homing, speed of the rapid feed, in the direction specified by the REVERSE input. When the rising edge of the DEC1 signal is detected, the speed is decelerated to the approach speed. The speed is decelerated to the creep speed when the first C phase pulse is detected after the DEC1 signal is passed and positioning is performed. The machine coordinate system is established with the position to be the HOME. Distance from C phase pulse is set in the OFSET. If OT signal is detected while in Homing process, it becomes OT alarm.

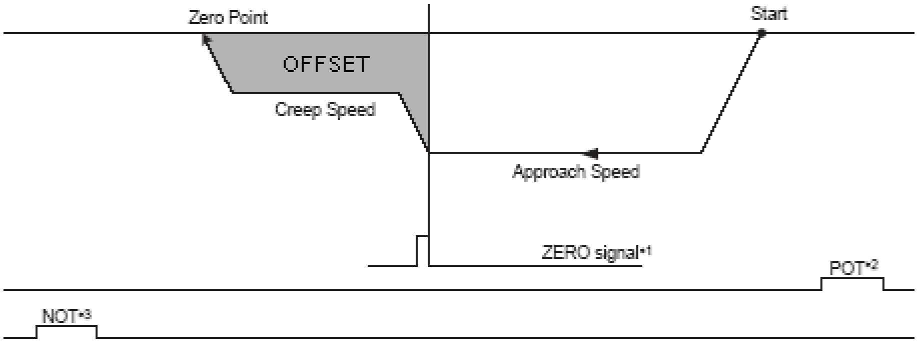


* 1. The SERVOPACK DEC signal.
* 2. The SERVOPACK P-OT signal.
* 3. The SERVOPACK N-OT signal.

Input	Name	Set Content
EXECUTE	Execution	Block enable
REVERSE	Direction selection	The direction of the homing. 0: Seek in forward 1: Seek in reverse
INPUT	INPUT signal	Not used
AXIS	Axis setting	Axis number related to the block. 1～16
TYPE	Home position return method	0: "DEC+C phase pulse" The method is selected.
TIME	Timer limit setting	Maximum time to complete the Homing, if timeout then Error
TRAVSPD	Speed of rapid feed	Traverse speed in count/second. 0-Max speed (ML3**12)
APRSPD	Approach speed	Approach speed in count/second. 0-Max speed (ML3**12)
CRPSPD	Creep rate	Creep speed in count/second. 0-Max speed (ML3**12)
OFFSET	Offset value	Final moving distance. And the position will be this value. If the value is positive, the final moving is in the same direction of homing. If it is negative the final moving will be in the opposite of homing

(b) ZERO signal method

The movement starts at the approach speed in the direction specified by the parameter. Speed is decelerated to the creep speed at the rising edge of the ZERO signal and finishes positioning. The machine coordinate system is established with the position to be the HOME.
The amount of the movement from the ZERO signal is set in the final travel distance of homing.
When the OT signal is detected while in homing process, it becomes OT alarm.

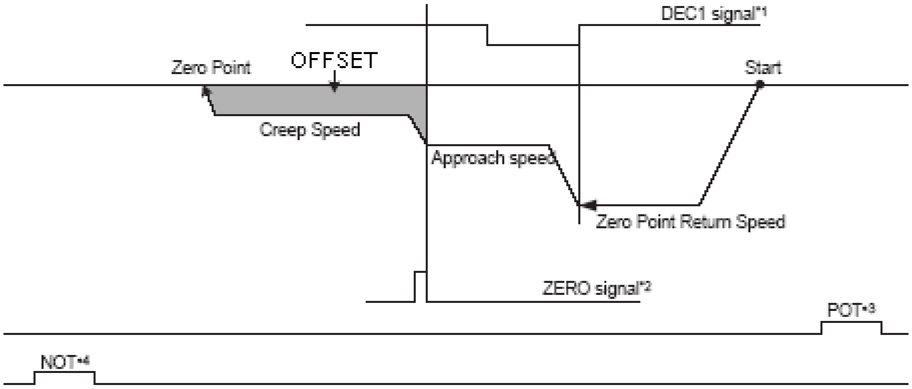


* 1. The SERVOPACK EXT1 signal.
* 2. The SERVOPACK P-OT signal.
* 3. The SERVOPACK N-OT signal.

Input	Name	Set content
EXECUTE	Execution	Block enable
REVERSE	Direction selection	The direction of the homing. 0: Seek in forward 1: Seek in reverse
INPUT	INPUT input	Not used
AXIS	Axis setting	Axis number related to the block. 1～16
TYPE	Home position return method	1: "ZERO signal" method is selected.
TIME	Timer limit setting	Maximum time to complete the Homing, if timeout then Error
TRAVSPD	Speed of rapid feed	Not used
APRSPD	Approach speed	Approach speed in count/second. 0-Max speed (ML3**12)
CRPSPD	Creep rate	Creep speed in count/second. 0-Max speed (ML3**12)
OFFSET	Offset value	Final moving distance. And the position will be this value. If the value is positive, the final moving is in the same direction of homing. If it is negative the final moving will be in the opposite of homing

(c) DEC 1+ZERO signal method

The movement starts at the speed of the homing in the direction specified by the parameter. When the rising edge of the DEC1 signal is detected, speed is decelerated to the approach speed. Speed is decelerated to the creep speed at the rising edge of the ZERO signal after DEC1 is passed at the approach speed, and final positioning is performed. The machine coordinate system is established with the position to be the HOME .
The amount of the movement from the ZERO signal is set in the final travel distance of homing.
When the OT signal is detected while in homing process, it becomes OT alarm.

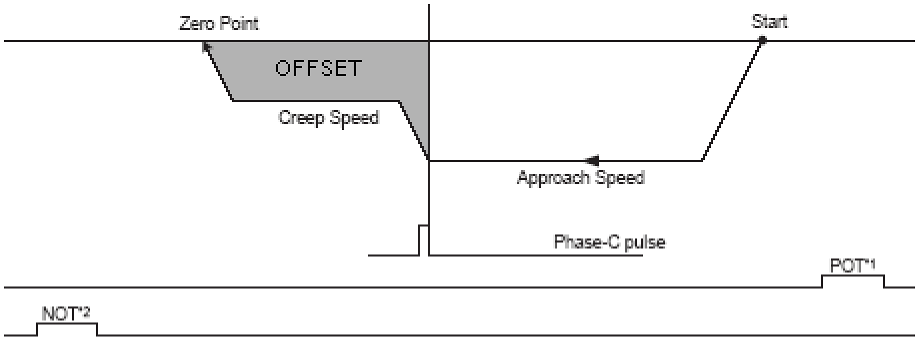


* 1. The SERVOPACK DEC signal.
* 2. The SERVOPACK EXT1 signal.
* 3. The SERVOPACK P-OT signal.
* 4. The SERVOPACK N-OT signal.

Input	Name	Set content
EXECUTE	Execution	Block enable
REVERSE	Direction selection	The direction of the homing. 0: Seek in forward 1: Seek in reverse
INPUT	INPUT input	Not used
AXIS	Axis setting	Axis number related to the block. 1～16
TYPE	Home position return method	2: "DEC+ZERO signal" method is selected.
TIME	Timer limit setting	Maximum time to complete the Homing, if timeout then Error
TRAVSPD	Speed of rapid feed	Traverse speed in count/second. 0-Max speed (ML3**12)
APRSPD	Approach speed	Approach speed in count/second. 0-Max speed (ML3**12)
CRPSPD	Creep rate	Creep speed in count/second. 0-Max speed (ML3**12)
OFFSET	Offset value	Final moving distance. And the position will be this value. If the value is positive, the final moving is in the same direction of homing. If it is negative the final moving will be in the opposite of homing

(d) C phase pulse method

The movement starts at the approach speed in the direction specified by the parameter. Speed is decelerated to the creep speed at the rising edge of C phase pulse, and final positioning is performed. The machine coordinate system is established with the position to be the HOME. The amount of the movement from C phase pulse is set in the final travel distance of homing.
When the OT signal is detected while in homing process, it becomes OT alarm.

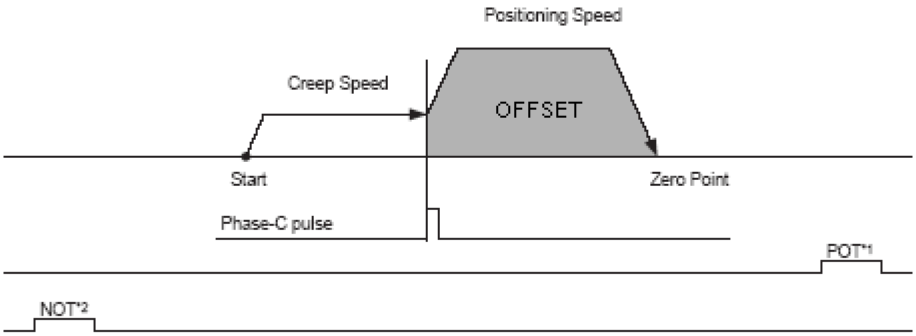


* 1. The SERVOPACK P-OT signal.
* 2. The SERVOPACK N-OT signal.

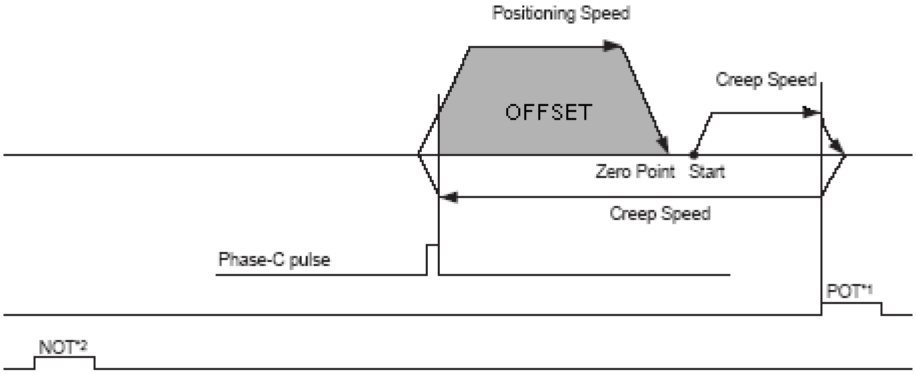
Input	Name	Set content
EXECUTE	Execution	Block enable
REVERSE	Direction selection	The direction of the homing. 0: Seek in forward 1: Seek in reverse
INPUT	INPUT input	Not used
AXIS	Axis setting	Axis number related to the block. 1～16
TYPE	Home position return method	3: "C phase pulsed system" is selected.
TIME	Timer limit setting	Maximum time to complete the Homing, if timeout then Error
TRAVSPD	Speed of rapid feed	Not used
APRSPD	Approach speed	Approach speed in count/second. 0-Max speed (ML3**12)
CRPSPD	Creep rate	Creep speed in count/second. 0-Max speed (ML3**12)
OFFSET	Offset value	Final moving distance. And the position will be this value. If the value is positive, the final moving is in the same direction of homing. If it is negative the final moving will be in the opposite of homing

(e) C phase pulse method

The movement starts at the creep speed in the direction specified by the parameter, and final positioning is performed at the rising edge of C phase pulse with the positioning speed. The machine coordinate system is established with the position to be the HOME. The amount of the movement from C phase pulse is set in the final travel distance of homing. The positioning speed is set in the rapid feed speed. OT signal does not cause an alarm when the is detected while moving at the creep rate, but reverses the direction. And homing process looks for C phase pulse. If OT signal is detected at the positioning speed, it becomes OT alarm.



OT Signal Detected during Creep Speed Operation



* 1. The SERVOPACK P-OT signal.
* 2. The SERVOPACK N-OT signal.

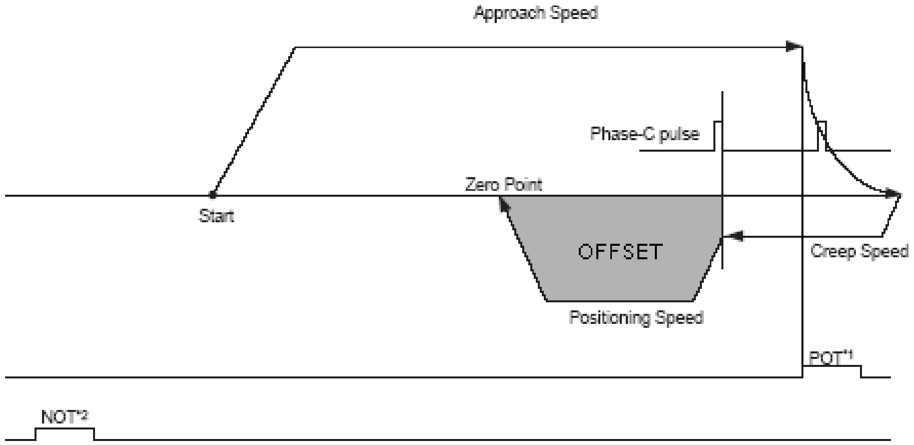
Note: The stopping method when the OT signal is detected depends on the setting of SERVOPACK parameters.

Input	Name	Set content
EXECUTE	Execution	Block enable
REVERSE	Direction selection	The direction of the homing. 0: Seek in forward 1: Seek in reverse
INPUT	INPUT input	Not used
AXIS	Axis setting	Axis number related to the block. 1～16
TYPE	Home position return method	11: "C phase pulse" is selected.
TIME	Timer limit setting	Maximum time to complete the Homing, if timeout then Error
TRAVSPD	Speed of rapid feed	Traverse speed in count/second. 0-Max speed (ML3**12)
APRSPD	Approach speed	Not used
CRPSPD	Creep rate	Creep speed in count/second. 0-Max speed (ML3**12)
OFFSET	Offset value	Final moving distance. And the position will be this value. If the value is positive, the final moving is in the same direction of homing. If it is negative the final moving will be in the opposite of homing

(f) POT & C phase pulse method

The movement starts at the approach speed, and moves to the stroke limit in a positive direction. The movement is reversed when POT signal is detected at the creep speed.
POT signal is passed in reversed direction and C pulse is detected, the final positioning is performed. The machine coordinate system is established with the position to be the HOME.
The amount of the movement from C phase pulse is set in the final travel distance of homing.
Positioning speed is set in the rapid feed speed.
When the OT signal is detected while moving at the positioning speed, it becomes OT alarm.

Note: The stop method when the OT signal is detected follows the user constant setting of the servo amplifier.

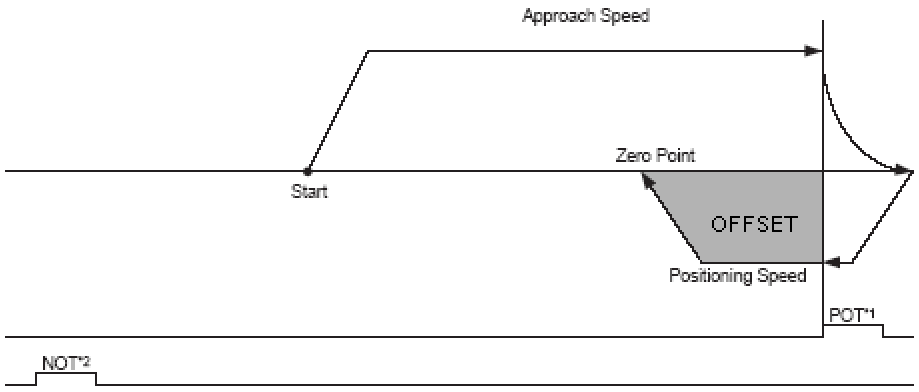


* 1. The SERVOPACK P-OT signal.
* 2. The SERVOPACK N-OT signal.
Note: The stopping method when the OT signal is detected depends on the setting of SERVOPACK parameters.

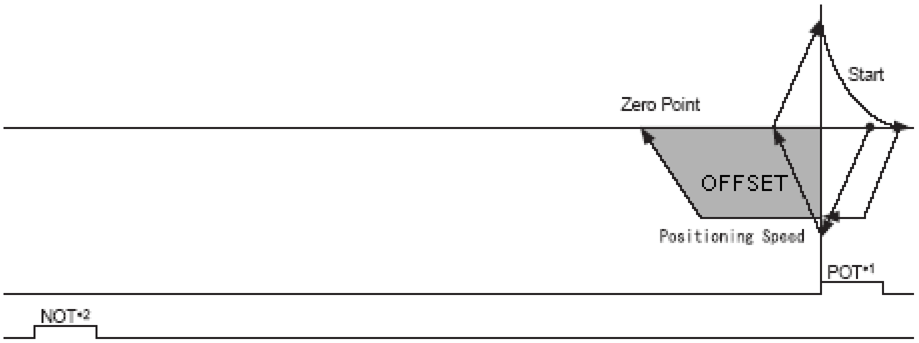
Input	Name	Set content
EXECUTE	Execution	Block enable
REVERSE	Direction selection	The direction of the homing. 0: Seek in forward 1: Seek in reverse
INPUT	INPUT input	Not used
AXIS	Axis setting	Axis number related to the block. 1～16
TYPE	Home position return method	12: "POT&C phase pulse" is selected
TIME	Timer limit setting	Maximum time to complete the Homing, if timeout then Error
TRAVSPD	Speed of rapid feed	Traverse speed in count/second. 0-Max speed (ML3**12)
APRSPD	Approach speed	Approach speed in count/second. 0-Max speed (ML3**12)
CRPSPD	Creep rate	Creep speed in count/second. 0-Max speed (ML3**12)
OFFSET	Offset value	Final moving distance. And the position will be this value. If the value is positive, the final moving is in the same direction of homing. If it is negative the final moving will be in the opposite of homing

(g) POT

The movement starts at the approach speed, and moves to the stroke limit in a positive direction. The movement reverses when the POT signal is detected. While in reverse motion at positioning speed and POT is cleared the final positioning is performed. The machine coordinate system is established with the position to be the HOME.
The amount of the movement from the change detection point of the POT signal status is set in the final travel distance of homing and the positioning speed is set in the rapid feed speed.
When the OT signal is detected while moving at the positioning speed, it becomes OT alarm.
Detection of the OT signal status change is done by the software. Therefore, positioning accuracy can not be guaranteed because of high-speed scan time and positioning speed.
Please do not use this method if repeatability is required for the home position.



Starting on the Positive Overtravel Limit (POT)

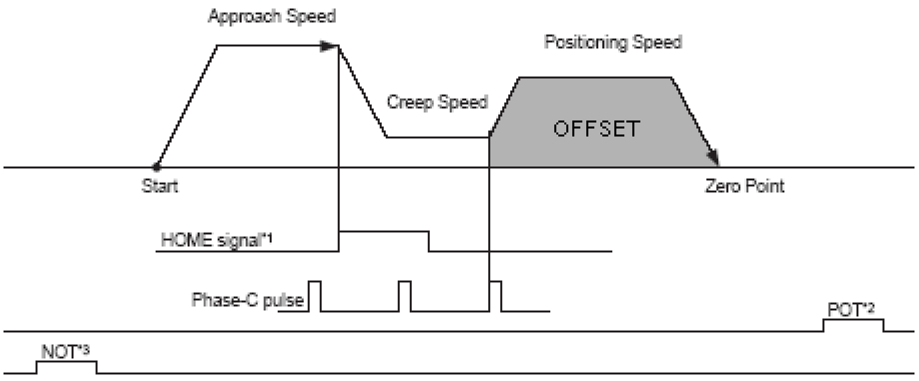


* 1. The SERVOPACK P-OT signal.
* 2. The SERVOPACK N-OT signal.
Note: The stopping method when the OT signal is detected depends on the setting of SERVOPACK parameters.

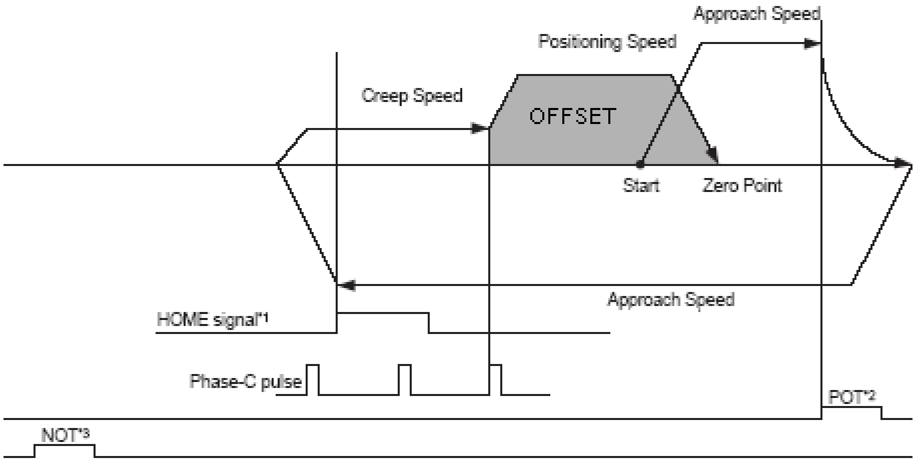
Input	Name	Set content
EXECUTE	Execution	Block enable
REVERSE	Direction selection	The direction of the homing. 0: Seek in forward 1: Seek in reverse
INPUT	INPUT input	Not used
AXIS	Axis setting	Axis number related to the block. 1～16
TYPE	Home position return method	13: “POT" is set.
TIME	Timer limit setting	Maximum time to complete the Homing, if timeout then Error.
TRAVSPD	Speed of rapid feed	Traverse speed in count/second. 0-Max speed (ML3**12)
APRSPD	Approach speed	Approach speed in count/second. 0-Max speed (ML3**12)
CRPSPD	Creep rate	Not used.
OFFSET	Offset value	Final moving distance. And the position will be this value. If the value is positive, the final moving is in the same direction of homing. If it is negative the final moving will be in the opposite of homing

(h) HOME LS & C phase pulse method

The movement starts at the approach speed in the direction specified by the parameter. Speed is changed to the creep speed at the rising edge of the HOME signal. The final positioning is performed at the positioning speed after the first C phase pulse is detected after the falling edge of the HOME signal. The machine coordinate system is established with the position to be the HOME.
The amount of the movement from C phase pulse is set in the final travel distance of homing. Positioning speed is set in the rapid feed speed. OT detection at the approach speed does not become an alarm, but the movement reverses, and looks for the HOME signal. If the OT signal is detected while moving at the positioning speed, it becomes OT alarm.



Detecting the OT Signal during Approach Speed Movement



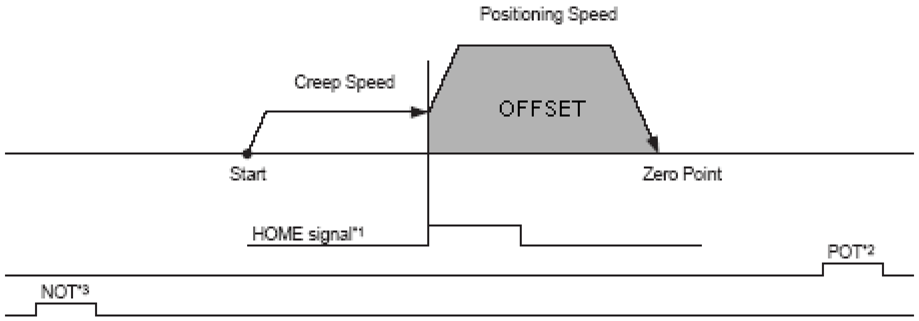
* 1. The SERVOPACK EXT1 signal.
* 2. The SERVOPACK P-OT signal.
* 3. The SERVOPACK N-OT signal.

Note: The stopping method when the OT signal is detected depends on the setting of SERVOPACK parameters.

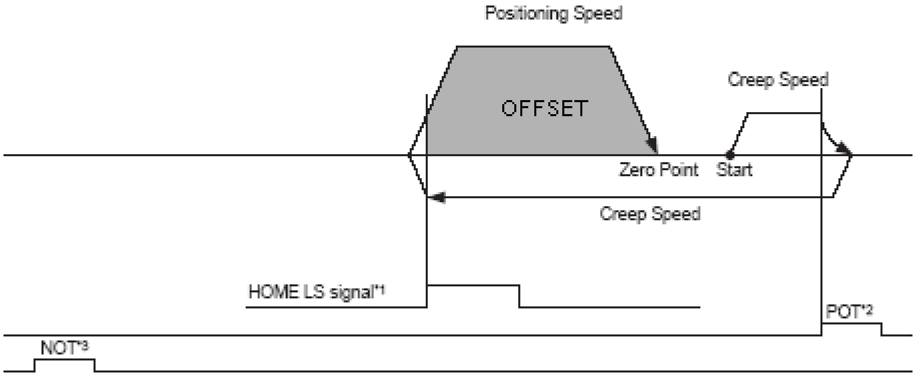
Input	Name	Set content
EXECUTE	Execution	Block enable
REVERSE	Direction selection	The direction of the homing. 0: Seek in forward 1: Seek in reverse
INPUT	INPUT input	Not used
AXIS	Axis setting	Axis number related to the block. 1～16
TYPE	Home position return method	14: "HOME LS&C phase pulse" is selected.
TIME	Timer limit setting	Maximum time to complete the Homing, if timeout then Error
TRAVSPD	Speed of rapid feed	Traverse speed in count/second. 0-Max speed (ML3**12) Moving direction follow the sign of the OFFSET.
APRSPD	Approach speed	Approach speed in count/second. 0-Max speed (ML3**12)
CRPSPD	Creep rate	Creep speed in count/second. 0-Max speed (ML3**12)
OFFSET	Offset value	Final moving distance. And the position will be this value. If the value is positive, the final moving is in the same direction of homing. If it is negative the final moving will be in the opposite of homing

(i) HOME LS

The movement starts at the creep speed in the direction specified by the parameter. At the rising edge of the HOME signal, the final positioning is performed at the positioning speed. The machine coordinate system is established with the position to be the HOME. The amount of the movement from the rising edge of the HOME signal is set in the final travel distance for homing and the positioning speed is set in setting the rapid feed speed.
OT signal at the creep speed does not cause alarm but the movement is reversed and looks for the HOME signal. If OT signal is detected while moving at the positioning speed, it becomes OT alarm.



Detecting the OT Signal during Creep Speed Movement



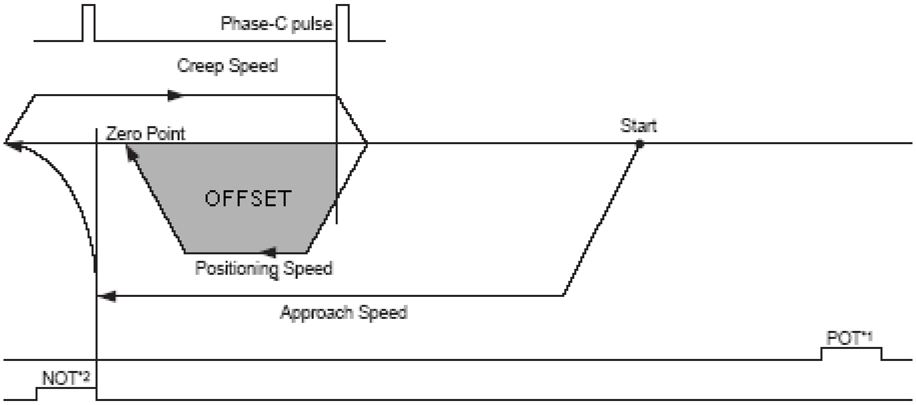
* 1. The SERVOPACK EXT1 signal.
* 2. The SERVOPACK P-OT signal.
* 3. The SERVOPACK N-OT signal.

Note: The stopping method when the OT signal is detected depends on the setting of SERVOPACK parameters.

Input	Name	Set content
EXECUTE	Execution	Block enable
REVERSE	Direction selection	The direction of the homing. 0: Seek in forward 1: Seek in reverse
INPUT	INPUT input	Not used
AXIS	Axis setting	Axis number related to the block. 1～16
TYPE	Home position return method	15: “HOME LS" is selected.
TIME	Timer limit setting	Maximum time to complete the Homing, if timeout then Error
TRAVSPD	Speed of rapid feed	Traverse speed in count/second. 0-Max speed (ML3**12) Moving direction follow the sign of the OFFSET.
APRSPD	Approach speed	Not used
CRPSPD	Creep rate	Creep speed in count/second. 0-Max speed (ML3**12)
OFFSET	Offset value	Final moving distance. And the position will be this value. If the value is positive, the final moving is in the same direction of homing. If it is negative the final moving will be in the opposite of homing

(j) NOT & C phase pulse method

The movement starts at the approach speed, and moves to the stroke limit in a negative direction. When the NOT signal is detected, the movement is reversed and the speed is changed to the creep speed. When C phase pulse is detected in reversing after the NOT signal is passed the final positioning is performed. The machine coordinate system is established with the position to be the HOME. The amount of the movement from C phase pulse is set in the final travel distance for homing and the positioning speed is set in the rapid feed speed.
When the OT signal is detected while moving at the positioning speed, it becomes OT alarm.

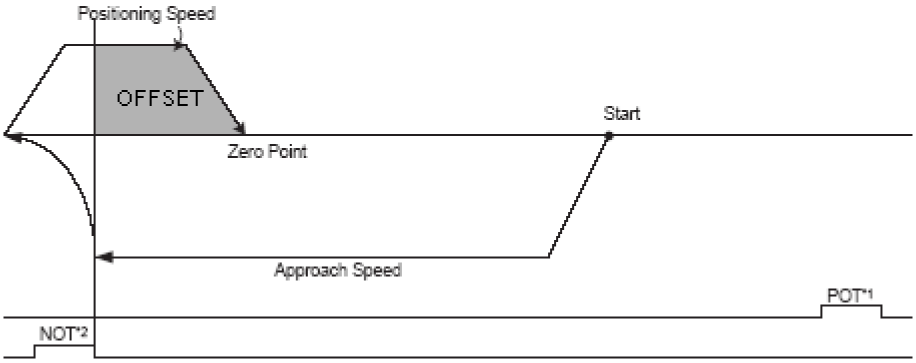


* 1. The SERVOPACK P-OT signal.
* 2. The SERVOPACK N-OT signal.
Note: The stopping method when the OT signal is detected depends on the setting of SERVOPACK parameters.

Input	Name	Set content
EXECUTE	Execution	Block enable
REVERSE	Direction selection	The direction of the homing. 0: Seek in forward 1: Seek in reverse
INPUT	INPUT input	Not used
AXIS	Axis setting	Axis number related to the block. 1～16
TYPE	Home position return method	16: "NOT&C phase pulse" is selected.
TIME	Timer limit setting	Maximum time to complete the Homing, if timeout then Error
TRAVSPD	Speed of rapid feed	Traverse speed in count/second. 0-Max speed (ML3**12) Moving direction follow the sign of the OFFSET.
APRSPD	Approach speed	Not used
CRPSPD	Creep rate	Creep speed in count/second. 0-Max speed (ML3**12) The movement direction is in positive.
OFFSET	Offset value	Final moving distance. And the position will be this value. If the value is positive, the final moving is in the same direction of homing. If it is negative the final moving will be in the opposite of homing

(k) NOT

The movement starts at the approach speed, and moves to the stroke limit in a negative direction. When the NOT signal is detected the movement reverses, at the positioning speed. The final positioning is performed when NOT signal is cleared in reversing. The machine coordinate system is established with the position to be the HOME.
The amount of the movement from the change detection point of the NOT signal status is set in the final travel distance of homing and the positioning speed is set in the rapid feed speed.
When the OT signal is detected while moving at the positioning speed, it becomes OT alarm.
It was changeable in the OT signal status by the software processing. Therefore, (*S) sets (*O) a high-speed scanning.
Detection of the OT signal status change is done by the software. Therefore, positioning accuracy can not be guaranteed because of high-speed scan time and positioning speed.
Please do not use this method if repeatability is required for the home position.



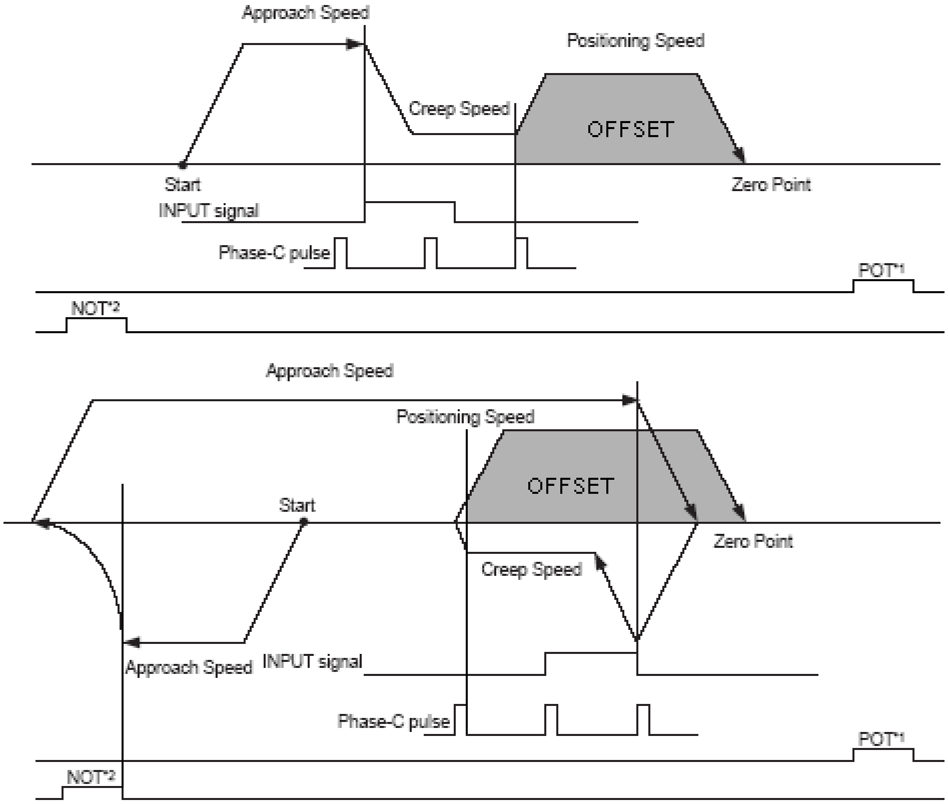
* 1. The SERVOPACK P-OT signal.
* 2. The SERVOPACK N-OT signal.

Note: The stopping method when the OT signal is detected depends on the setting of SERVOPACK parameters.

Input	Name	Set content
EXECUTE	Execution	Block enable
REVERSE	Direction selection	The direction of the homing. 0: Seek in forward 1: Seek in reverse
INPUT	INPUT input	Not used
AXIS	Axis setting	Axis number related to the block. 1～16
TYPE	Home position return method	17: “NOT" is selected.
TIME	Timer limit setting	Maximum time to complete the Homing, if timeout then Error
TRAVSPD	Speed of rapid feed	Traverse speed in count/second. 0-Max speed (ML3**12) Moving direction follow the sign of the OFFSET.
APRSPD	Approach speed	Approach speed in count/second. 0-Max speed (ML3**12)
CRPSPD	Creep rate	Not used
OFFSET	Offset value	Final moving distance. And the position will be this value. If the value is positive, the final moving is in the same direction of homing. If it is negative the final moving will be in the opposite of homing

(I) INPUT & C phase pulse

The movement starts at the approach speed in the direction specified by the parameter. Speed changes into the creep speed if the rising edge of the INPUT signal is detected. When the first C phase pulse after the INPUT signal falling edge is detected, the final positioning is performed at the positioning speed. The machine coordinate system is established with the position to be the HOME.
The amount of the movement from C phase pulse detection point is set in the final travel distance for homing and the positioning speed is set in the rapid feed speed.
If OT signal is detected while moving at the approach speed, it does not become an alarm, but the movement is reversed, and looks for the INPUT signal. When the OT signal is detected while moving at the positioning speed, it becomes OT alarm.
It is necessary to turn on the INPUT signal by the ladder program.

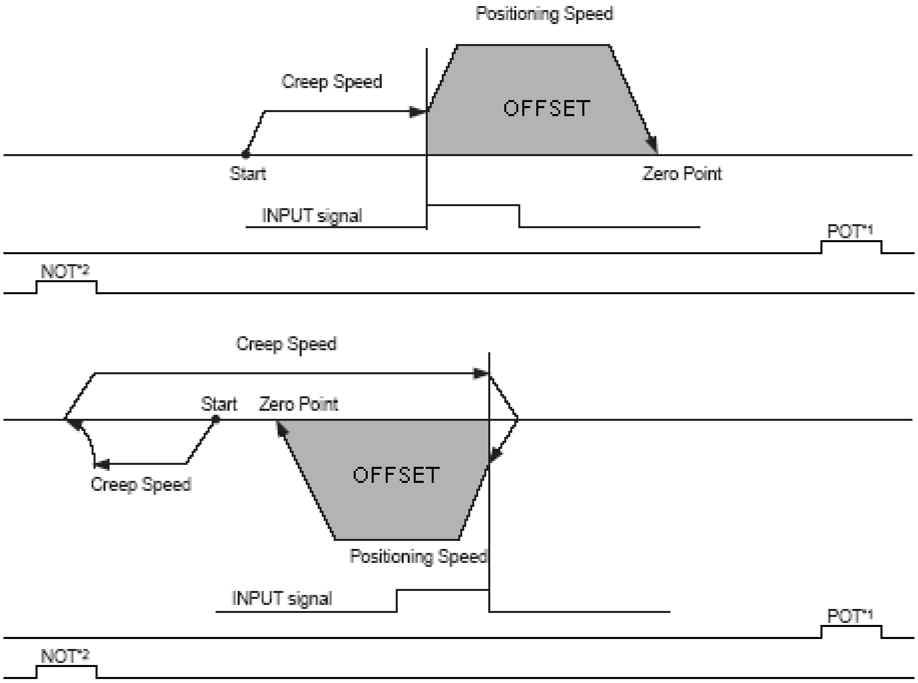


* 1. The SERVOPACK P-OT signal.
* 2. The SERVOPACK N-OT signal.
Note: The stopping method when the OT signal is detected depends on the setting of SERVOPACK parameters.

Input	Name	Set content
EXECUTE	Execution	Block enable
REVERSE	Direction selection	The direction of the homing. 0: Seek in forward 1: Seek in reverse
INPUT	INPUT signal	Allocation of INPUT signal Function block sets to OBxx05B.
AXIS	Axis setting	Axis number related to the block. 1～16
TYPE	Home position return method	18: "INPUT&C phase pulse" is selected.
TIME	Timer limit setting	Maximum time to complete the Homing, if timeout then Error
TRAVSPD	Speed of rapid feed	Traverse speed in count/second. 0-Max speed (ML3**12) Moving direction follow the sign of the OFFSET.
APRSPD	Approach speed	Approach speed in count/second. 0-Max speed (ML3**12)
CRPSPD	Creep rate	Creep speed in count/second. 0-Max speed (ML3**12)
OFFSET	Offset value	Final moving distance. And the position will be this value. If the value is positive, the final moving is in the same direction of homing. If it is negative the final moving will be in the opposite of homing

(m) INPUT

The movement starts at the creep speed in the direction of the sign of the creep speed. The final positioning is performed at the rising edge of the INPUT signal at the positioning speed. The machine coordinate system is established with the position to be the HOME.
The amount of the movement from the rising edge of the INPUT signal is set in the final travel distance of homing and the positioning speed is set in the rapid feed speed.
OT signal while moving at the creep speed does not cause alarm but the movement is reversed, and looks for the INPUT signal.
If OT signal is detected while moving at the positioning speed, it becomes OT alarm.
The INPUT signal is set to OBxx05B. A temporary home position can be set for the trial run since no actual sensor wiring is necessary. Detection of the rising of the INPUT signal is processed by the software. Therefore, home position can be different by the high-speed scanning setting and the positioning speed. Please do not use this method if repeatability is necessary for the home position.
INPUT signal must be controlled in the ladder program.



* 1. The SERVOPACK P-OT signal.
* 2. The SERVOPACK N-OT signal.
Note: The stopping method when the OT signal is detected depends on the setting of SERVOPACK parameters.

Input	Name	Set content
EXECUTE	Execution	Block enable
REVERSE	Direction selection	The direction of the homing. 0: Seek in forward 1: Seek in reverse
INPUT	INPUT input	Allocation of INPUT signal Function block sets to OBxx05B.
AXIS	Axis setting	Axis number related to the block. 1～16
TYPE	Home position return method	19: “INPUT" is selected.
TIME	Timer limit setting	Maximum time to complete the Homing, if timeout then Error
TRAVSPD	Speed of rapid feed	Traverse speed in count/second. 0-Max speed (ML3**12) Moving direction follow the sign of the OFFSET.
APRSPD	Approach speed	Not used
CRPSPD	Creep rate	Creep speed in count/second. 0-Max speed (ML3**12)
OFFSET	Offset value	Final moving distance. And the position will be this value. If the value is positive, the final moving is in the same direction of homing. If it is negative the final moving will be in the opposite of homing

<HOME> Working Registers

This table outlines the data in the ten registers used by the Home function block. There is not usually any need for the user to access any of these bits directly.

Register No	TYPE	Name	Content
AW00000			
Bit 0	IN	ECECUTE。	EXECUTE input (XB000000)
Bit 1	IN	REVERSE	REVERSE input (XB000001)
Bit 2	IN	INPUT	INPUT signal (XB000002)
Bit 3	IN	Reservation	Reserved
Bit 4	IN	firstPass	Single shot input of ENABLE input to initialize axis and block
Bit 5	Working	inrngAxis	Goes high for one scan if Axis input is in range.
Bit 6	Working	inrngAxis	Goes high If the RDA register is not set appropriately.
Bit 7	Working	cmndErr	Another block controlled the axis.
Bit 8	Working	ErrSTOP	Servo not enabled when motion is commanded
Bit 9	OUT	Blkfault	Directly controls YB000002 (*ERROR* Output)
Bit C	Working	OTalarm	Overtravel alarm
Bit D	Working	Vel_Err	Either Rapid feed speed, Approach speed or Creep speed is out of range.
Bit F	Working	TimeOver	Time expired will set *ERROR output*
AW00001			
Bit 1	OUT	RUNNING	Directly controls YB000000 (*RUNNING* Output)
Bit 2	OUT	COMPLETE	Directly controls YB000001 (*DONE* Output)
AW00002	Working	Reserved	Reserved
AW00003	Working	Reserved	Reserved
AW00004	Working	Timer	Timer for limiting home time before *ERROR*. Tied to *TIMELIM* input (XW00003).
AL00005	Working	Reserved	Reserved
AW00007	Working	Reserved	Reserved
AW00008	Working	Reserved	Reserved
AW00009	Working	rdaMult	Value for address offset to locate proper RDA
AW00010	Working	Revision	Revision Level of the function block.

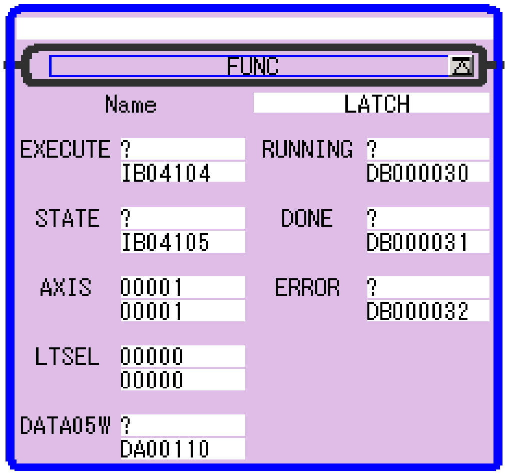
LATCH function

Function block for MP2000 series

<LATCH> Function Block Summary

The LATCH function block enables the high speed latch for a motor axis encoder. This block should be used any time the high-speed latch is required except when the Latch Target (Index Move) block is used.

Function Block Diagram



FUNC

Name	LATCH		
EXECUTE	? IB04104	RUNNING	? DB000030
STATE	? IB04105	DONE	? DB000031
AXIS	00001 00001	ERROR	? DB000032
LTSEL	00000 00000		
DATA05W	? DA00110		

<LATCH> Function Block Operation notes

- Rising edge of EXECUTE input initiates block operation, and the following block input values are read once: AXIS.
- To use the function block, the *EXECUTE* bit must be held ON. During this, the STATE bit is constantly monitored (STATE indicates latch request from user).
- The *RUNNING* output bit will be held high until a latch is received, *EXECUTE* goes low, or an *Error* occurs. The block will continue to monitor the axis for errors as long as the EXECUTE input stays True.
- If it is EXECUTE=High, and STATE=High, the controller arms latch detection, and wait for the signal specified in the LTSEL be ON. DONE output bit becomes High when latch detection is completed. Latched data of the machine coordinate system is stored to IL**18.
- Latch speed is 30microsec or less.
- Latch detection signal is “/EXT1" if LTSEL=1 and SGDS-***12A (with M□) servo amplifier parameter is default. Input of latch sensor connected with SGDS 1CN connector pin10 can be monitored at IB**2E6
- The latch detection signal is selected by setting LTSEL. 0= C phase pulse, 1=/ EXT1, 2=/ EXT2 and 3=/ EXT3
- Five words are used as working registers for this function, starting at the address in *Data05W*.

<LATCH> Input and Output Register Map

Output Registers

The following registers are used as outputs from the function block. They can be monitored by the LadderWorks program to check the execution of the function.

Output	Type	Description
RUNNING	Bit	This bit is ON when the function block is executing
DONE	Bit	This bit is latched ON when the Latch has been detected (IB**0C2=ON).
ERROR	Bit	Goes high when the error occurs in execution. Goes low when the error condition is released.

Input Registers

The following registers are used as inputs to the function block. They select the options and define the parameters that the user needs to make the function work as necessary.

INPUT	TYPE	Content	Range of setting and state
EXECUTE	Bit	Block enable – Block cannot execute unless this is TRUE	TRUE – Block enable FALSE – Block disable
STATE	Bit	Latch Enable or disable request from user	TRUE – Latch enabled FALSE – Latch disabled
AXIS	Word	Axis number related to the block	1～16
LTSEL	Word	Latch detection signal selection	0: C phase pulse signal 1: /EXT1 (servo amplifier DI) 2: /EXT2 (servo amplifier DI) 3: /EXT3 (servo amplifier DI)
DATA05W	Address	Address of the first working register.	Five words of register space are used by this function.

<LATCH> Block Fault Conditions:

The following table outlines several situations that may cause an error, and will turn on the blocks "Error" output bit. The block "Error" output will cleared if the *EXECUTE* bit goes low.

Internal error bit	Cause	Note
input_OK AB00000E	LTSEL input is not an acceptable value	LTSEL input range is 0 through 3 integer. This error turns ON RDA Error ID(MW3**81) bit3.
outRngAxis AB00000F	The axis number entered on the input is not an acceptable value	The function blocks can only control 1 to 16 axes. Any value greater or smaller then this will cause an error. This does not set the RDA Error ID

<LATCH> Working Registers

This table outlines the data in the four registers used by the function block. There is not usually any need for the user to access any of these bits directly.

Register No	TYPE	Name	Content
AW00000			
Bit 0	IN	execute	*EXECUTE* input (XB000000)
Bit 1	IN	state	*STATE* input (XB000001)
Bit 4	Working	inrngAxis	Goes high for one scan if Axis input is in range.
Bit 6	Working	closePass	Function internal processing for execution next time.
Bit 7	OUT	running	Directly controls YB000000 (*RUNNING* Output)
Bit 8	OUT	done	Directly controls YB000001 (*DONE* Output)
Bit 9	OUT	error	Directly controls YB000002 (*ERROR* Output)
Bit A	Working	oneshotA	Reserved.
Bit B	Working	firstPass	One shot coil for initiating Latch and block
Bit C	Working	oneshotC	Reserved.
Bit E	Working	input_OK	Goes high for one scan if LTSEL input is in range.
Bit F	Working	outRngAxis	Goes high for one scan if Axis input is out of range.
AW00003	Working	rDAmult	Value for address offset to locate proper RDA
AW00004	Working	Revision	Revision Level of the function block.

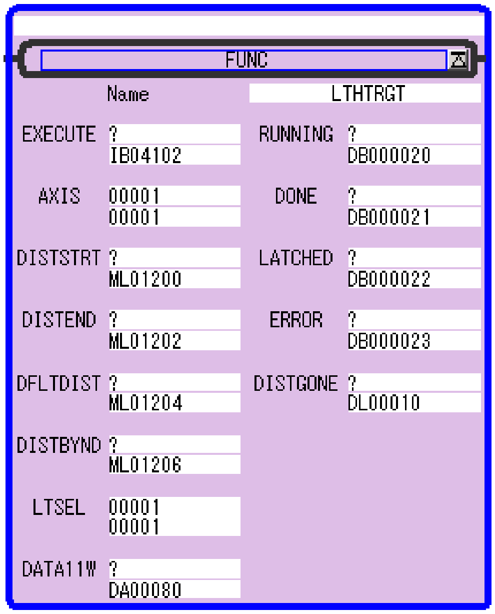
LATCH TO TARGET function

Function block for MP2000 series

<LTHTRGT> Function Block Summary

The Latch Target (LTHTRGT) block is used to detect a latch on an index move. The axis will be instructed to move a default distance (DFLTDIST), if a latch is received within the window during the index (window is defined between DISTSTRT and DISTEND), the final target position is changed to latch position plus a distance beyond (final target position = Latched position + DISTBYND), and the axis is commanded to move to this new position without interruption. See external timing diagram below for a graphical representation.

Function Block Diagram



FUNC

Name	LTHTRGT		
EXECUTE	? IB04102	RUNNING	? DB000020
AXIS	00001 00001	DONE	? DB000021
DISTSTRT	? ML01200	LATCHED	? DB000022
DISTEND	? ML01202	ERROR	? DB000023
DFLTDIST	? ML01204	DISTGONE	? DL00010
DISTBYND	? ML01206		
LTSEL	00001 00001		
DATA11W	? DA00080		

<LTHTRGT> Function Block Operation Notes

- Rising edge of EXECUTE input initiates block operation, and the following block input values are read once: AXIS, DISTSTRT, DISTEND, DFLTDIST, and DISTBYND.
- To use the function block, the *EXECUTE* bit must be held ON. If the *EXECUTE* bit goes off during operation, the block will finish move but all outputs will be set to zero.
- DISTBYND is only read at the Latch Event.
- The *RUNNING* output bit will be held high until: move is completed, *EXECUTE* goes low, or an *Error* occurs.
- LATCHED output will turn on if a latch was received within the window during the move (IB**0C2=ON).
- DONE bit turns on when the move is completed. (IW**08=0 running completion of operation).
- If the *EXECUTE* input bit goes off during operation, all outputs will go to zero and the axis will travel to the default distance unless the latch was detected prior to the bit going low. In that case the axis will go the distance beyond the Latch position.
- Latch is only active between the latch start and latch end window, defined by *DISTSTRT* and *DISTEND* inputs. If the latch occurs inside the window the axis will travel the *DISTBYND* input value beyond the latched position. If a latch occurs outside the window, it will be ignored because the latch is disabled (the motion command changes back). In this case, the axis will go the default distance DFLTDIST.
- If the Latch occurs during the deceleration part of the default distance curve (and it is within the defined window), the axis will accelerate at the defined rate to set speed to achieve the final distance move.
- All distance inputs are relative to the start position except the *DISTBYND* input. It is relative to the latch position.
- The Distance beyond value should be sufficient a distance for the axis to come to a stop with the set deceleration value or the axis will have to back up to the point desired.
- The latch detection signal is selected by LTSEL input. 0= C phase pulse, 1=/ EXT1, 2=/ EXT2 and 3=/ EXT3
- Eleven words are used as working registers for this function, starting at the address in *Data11W*.

<LTHTRGT> Input and Output Register Map

Output Registers

The following registers are used as outputs from the function block. They can be monitored by the LadderWorks program to check the execution of the function

Output	Type	Description
RUNNING	Bit	High while block has control and move is in progress.
DONE	Bit	Goes high when move is completed and remains high until execute is turned off or another block takes control. (IW**08=0 running completion of operation).
LATCHED	Bit	Goes high if the latch was detected within the specified window (IB**0C2=ON).
ERROR	Bit	Goes high if any error occurs in block or on the axis. Goes low if the error condition is released.
DISTGONE	Long	Current Distance in counts axis has traveled in this move.

Input Registers
The following registers are used as inputs to the function block. They select the options and define the parameters that the user needs to make the function work as necessary.

INPUT	TYPE	Content	Range and state
EXECUTE	Bit	Block enable – Block cannot execute unless this is TRUE	Rising edge initiates block TRUE – Block enable FALSE – Block disable, set all outputs to zero and off
AXIS	Word	Axis number related to the block	1～16
DISTSTRT	Long	Sets the beginning of the latch activation window. Defined as distance in counts *from* the start position of the axis. Latch is armed between DISTSTRT and DISTEND.	2147483648～2147483647
DISTEND	Long	Sets the end of the latch activation window. Defined as distance in counts *from* the start position of the axis. Latch is armed between DISTSTRT and DISTEND.	2147483648～2147483647
DFLTDIST	Long	Default distance to travel in counts *from* the start position of the axis. This is the relative distance the axis will travel if a latch is not detected	2147483648～2147483647
DISTBYND	Long	Distance to travel in counts from the Latch position. This is the additional relative distance the axis will travel (from the latch detection position) if a latch is detected.	2147483648～2147483647
LTSEL	Word	Latch detection signal selection	0: C phase pulse signal 1: /EXT1 (servo amplifier DI) 2: /EXT2 (servo amplifier DI) 3: /EXT3 (servo amplifier DI)
DATA11W	**Address**	Address of the first working register.	Eleven words of register space are used by this function.

<LTHTRGT> Block error condition:

The following table outlines several situations that may cause an error, and will turn on the blocks “Error” output bit. The block “Error” output will cleared if the *EXECUTE* bit goes low.

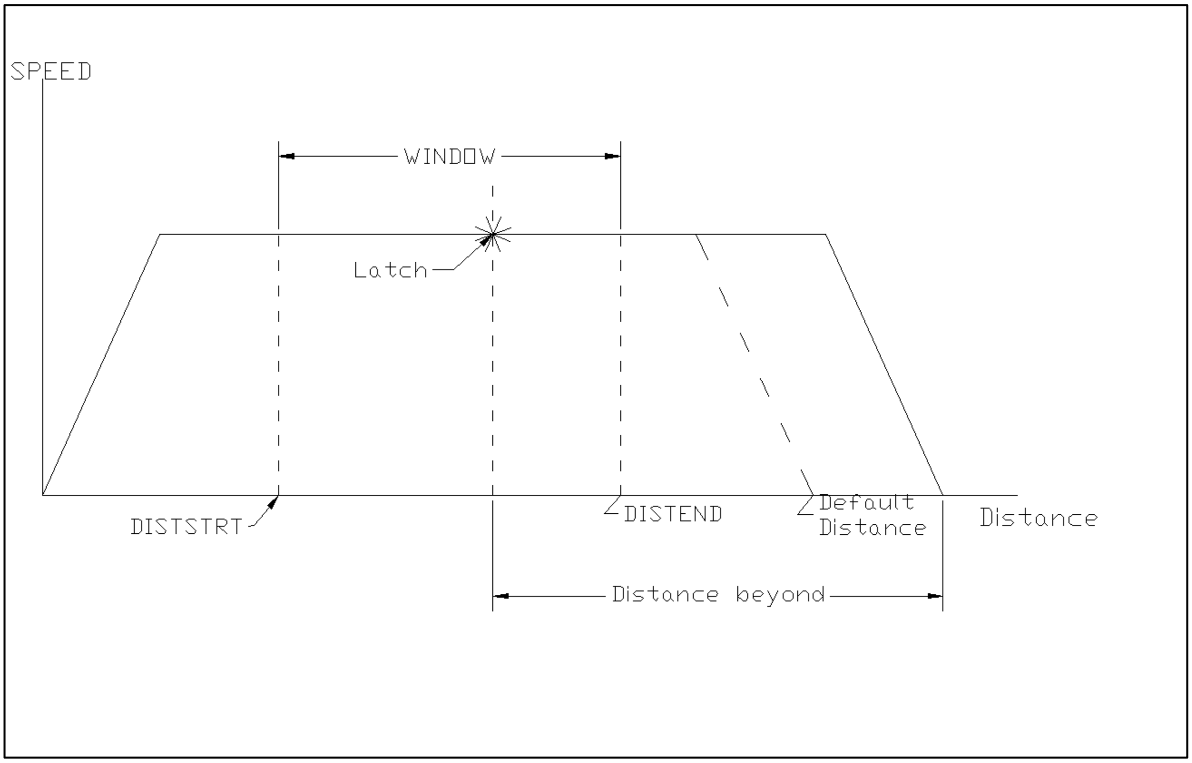
Internal error bit	Cause	Attention
errStop AB000005	Motion commanded and servo is disabled.	If block is running and the servo is disabled the error bit will be set. Sets RDA Error ID (MW3**81) bit B on if error state exists.
commanderr AB00000E	Another block took control of the axis.	If the block looses control of the axis while running an error will occur. This does not set the RDA Error ID.
direrror AB00000F	The direction commanded to travel is disabled from the SVON block.	The SVON block must be used and have the direction commanded enabled. Sets RDA Error ID (MW3**81) bit 4 on if error state exists.
axisInErr AB000083	The axis number entered on the input is not an acceptable value	The SVB multiaxis blocks can only control 1 to 16 axes. Any value greater or smaller then this will cause an error. This does not set the RDA Error ID..

<LTHTRGT> Working Registers

This table outlines the data in the twelve registers used by the function block. There is not usually any need for the user to access any of these bits directly.

Register No	TYPE	Name	Content
AW00000			
Bit 0	IN	execute	*EXECUTE* input (XB000000)
Bit 3	Working	latched	Goes high for one scan, if a latch occurs inside the latch activation window (one-shot)
Bit 4	Working	oneshot	Reserved.
Bit 5	Working	errStop	Goes high if servo faults out while running.
Bit 6	OUT	Latched	Directly controls YB000002 (*LATCHED* Output)
Bit 7	OUT	running	Directly controls YB000000 (*RUNNING* Output)
Bit 8	OUT	complete	Directly controls YB000001 (*DONE* Output)
Bit 9	OUT	error	Directly controls YB000003 (*ERROR* Output)
Bit A	Working	oneshotA	Reserved
Bit B	Working	firstPass	One shot coil for initiating axis and block.
Bit C	Working	oneshotC	Reserved.
Bit E	Working	commanderr	Rising Edge indicates another block took control.
Bit F	Working	direrror	Indicates direction commanded is not enabled.
AL00002	Working	startpos	Value of axis position at start of move.
AL00004	Working	distGone	Directly controls YL000001 (*DISTGONE* Output)
AW00009	Working	rdaMult	Value for address offset to locate proper RDA
AW00010	Working	Revision	Revision Level of the function block.

<LTHTRGT> Timing chart



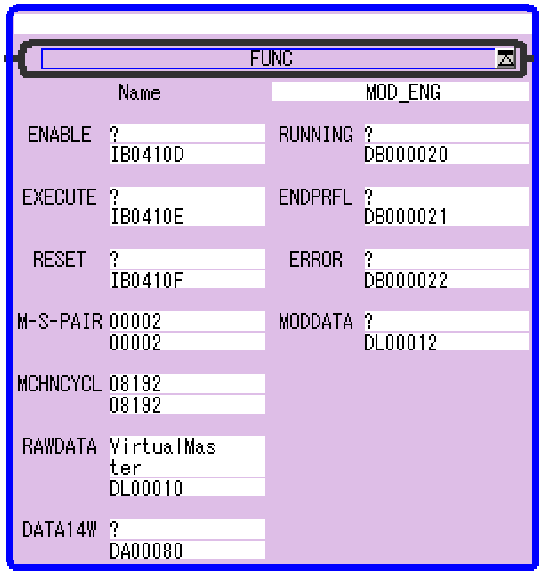
MODULUS ENGINE function

Function block for MP2000 series

<MOD_ENG> Function Block Summary

The “MOD-ENG” block is used to modulate any counter (true encoder or virtual counter) into a saw tooth shape pattern with a cycle equivalent to the machine cycle desired (see diagram below). The machine cycle is the highest value the output can go before it returns to zero. This block stores its data in the RDA based on its Master/Slave value. This block is required for the Cam function block to work.

Function Block Diagram



FUNC

Name		MOD_ENG	
ENABLE	? IB0410D	RUNNING	? DB000020
EXECUTE	? IB0410E	ENDPRFL	? DB000021
RESET	? IB0410F	ERROR	? DB000022
M-S-PAIR	00002 00002	MODDATA	? DL00012
MCHNCYCL	08192 08192		
RAWDATA	VirtualMaster DL00010		
DATA14W	? DA00080		

<MOD_ENG> Function Block Operation Notes

- This block should be used with CAM function block.
- To use the function block, the *ENABLE* bit must be held ON. The block will begin to execute (output the modulated RAWDATA input) on a rising edge from the *EXECUTE* bit.
- If the *ENABLE* bit goes off during operation, the Raw data as well as the modulated data will stop being sent to the RDA and all outputs will go to zero.
- The *RESET* bit sets the modulated data to zero.
- Note that the master/slave pairs are separated by 50 words in the RDA(up to 10 pairs). Example: Master/Slave pair #1 starts at MW56000, Master/Slave pair #2 starts at MW56050, Master/Slave pair #3 starts at MW56100, etc
- Fourteen words are used as working registers for this function, starting at the address in *Data14W*.

<MOD_ENG> Input and Output Register Map

Output Registers

The following registers are used as outputs from the function block. They can be monitored by the LadderWorks program to check the execution of the function.

Output	Type	Description
RUNNING	Bit	This bit is ON when the function block is executing.
ENDPRFL	Bit	End of cycle marker pulse. When the full length of the machine cycle has been reached from either direction this bit flickers goes on (one-shot).
ERROR	Bit	If an error occurs during block execution, this output latches ON.
MODDATA	Long	This is the modulated master output. The value of the modulated data is also stored in the RDA.

Input register

The following registers are used as inputs to the function block. They select the options and define the parameters that the user needs to make the function work as necessary.

Input	Type	Description	Range of state
ENABLE	Bit	Block enable – Block cannot execute unless this is TRUE	Rising edge initiates block TRUE – Block enable FALSE – Block disable, set all outputs to zero and off
EXECUTE	Bit	Starts the Modulus engine, modulates the master counter input.	Rising edge starts block modulation TRUE – modulation active FALSE – holds the modulated data
RESET	Bit	Sets the Modulated data to zero, also active when EXECUTE is on.	TRUE – Mod Data = 0 FALSE – Modulus engine free to run
M-S-PAIR	Word	Defines which Master-Slave pair is updated in RDA	1 to 10 value. Any other value will cause the error output to go on.
MCHNCYCL	Long	Highest value in modulated data	1～2147483647
RAWDATA	Long	Input to be modulated (feed virtual counter or encoder here)	2147483648～2147483647
DATA14W	**Address**	Address of the first working register.	Fourteen words of register space are used by this function.

<MOD_ENG> Block error condition:

The following table outlines several situations that may cause an error, and will turn on the blocks “Error” output bit. The block “Error” output will cleared if the *EXECUTE* bit goes low.

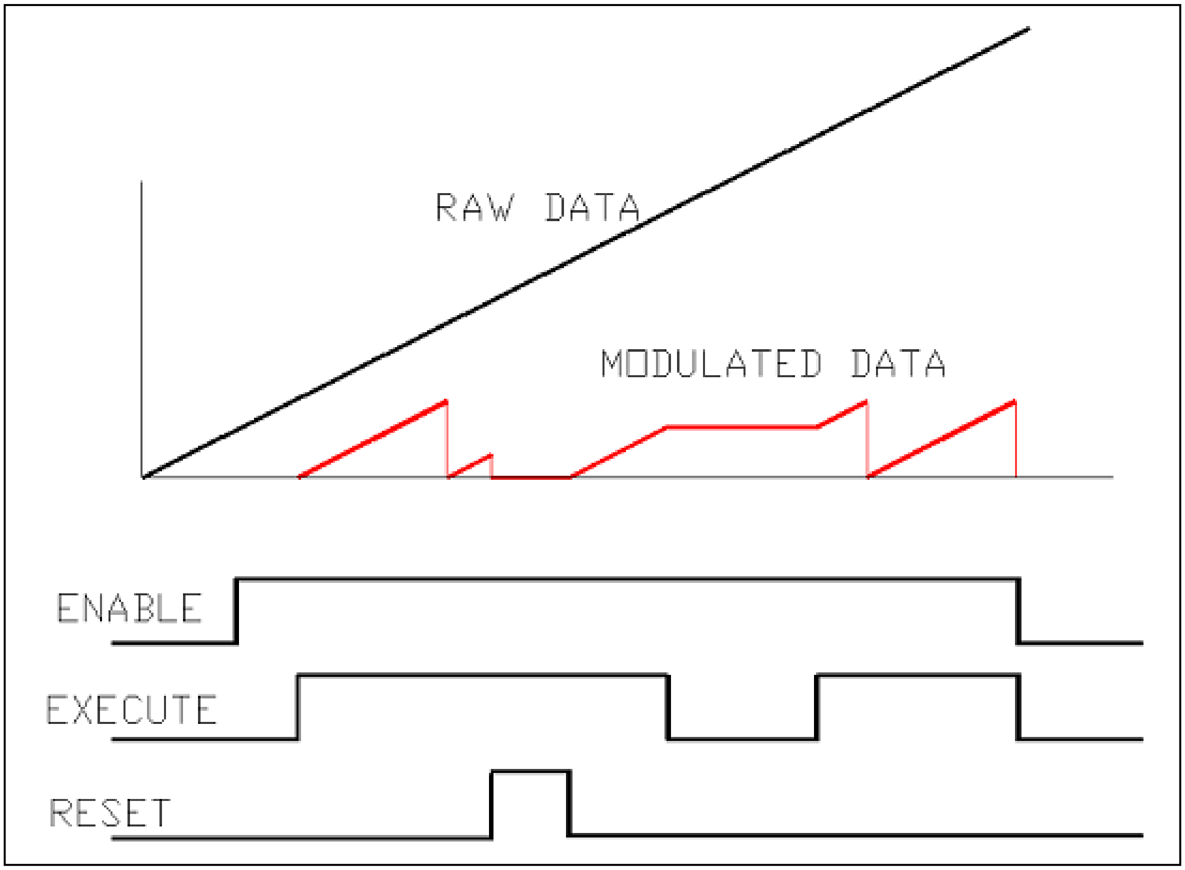
Fault bit	Cause	Note
inOK AB00000D	Machine Cycle input is out of range	inOK must be high if enable is on or an error will occur. *MCHNCYCL* must be in range for this bit to be on. Sets RDA Error ID (MW30181) bit 3.
msSelect AB00000E	*M-S-PAIR* input value out of range	msSelect must be high if enable is on or an error will occur. *M-S-PAIR* must have a value from one to ten for this bit to be on. Sets RDA Error ID (MW30181) bit 3.

<MOD_ENG> Working Registers

This table outlines the data in the eight registers used by the function block. There is not usually any need for the user to access any of these bits directly.

Register No	TYPE	Name	Content
AW00000			
Bit 0	IN	enable	*ENABEL* input (XB000000)
Bit 1	IN	reset	*RESET* input (XB000002)
Bit 2	IN	posOut	Reserved
Bit 3	IN	negOut	Reserved
Bit 4	OUT	endofTravl	Directly controls YB000001 (*ENDPRFL* Output)
Bit 7	OUT	running	Directly controls YB000000 (*RUNNING* Output)
Bit 9	OUT	error	Directly controls YB000002 (*ERROR* Output)
Bit A	Working	oneshotA	Reserved
Bit B	Working	firstPass	Rising edge of *EXECUTE* (XB000001)
Bit D	Working	inOK	Verifies that the *MCHNCYCL* input (XW00002) is in range.
Bit E	Working	msSelect	Verifies that the *M-S-PAIR* input (XW00001) is in range.
AW00001	Working	store_i	Reserved
AW00002	Working	pairvalue	*M-S-PAIR* input (XW00001) - 1.
AL00003	Working	o modpos_o	Reserved
AL00005	Working	pos	Modulated data being stored
AL00007	Working	lscratch	Reserved.
AL00009	Working	offset	Reserved
AL00011	Working	modData	Modulated data as read back from the RDA.
AW00013	Working	Revision	Revision Level of the function block.

<MOD_ENG> Timing chart



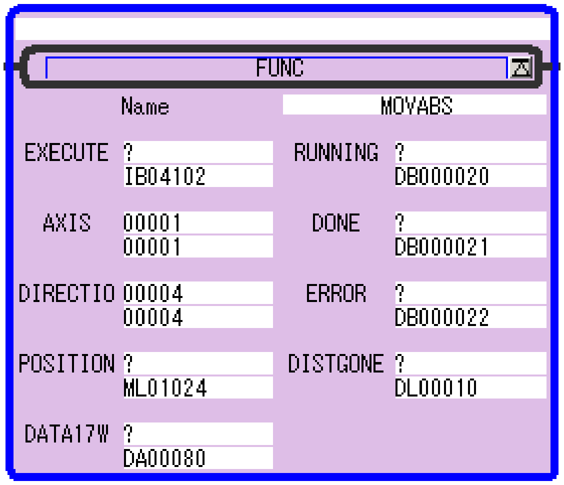
MOVE ABSOLUTE function

Function block for MP2000 series

<MOVEABS> Function Block Summary

This function block commands a controlled motion at a specified absolute position. The parameters are used at the time the motion is started. To modify any parameter it is necessary to put the correct SET of values in and to trigger again the motion.

Function Block Diagram



FUNC

Name		MOVABS	
EXECUTE	? IB04102	RUNNING	? DB000020
AXIS	00001 00001	DONE	? DB000021
DIRECTIO	00004 00004	ERROR	? DB000022
POSITION	? ML01024	DISTGONE	? DL00010
DATA17W	? DA00080		

<MOVEABS> Function Block Operation Notes

- MOVABS must be in H drawing, and is executed after SVON and CHANGE DYNAMICS function.
- Rising edge of EXECUTE input initiates block operation, and all block input values are read once.
- The RUNNING output bit maintains the operation in execution like High. The output bit becomes Low by the positioning completed, EXECUTE=Low or Error.
- The *RUNNING* output bit will be held high until move is completed (IB**0C1=ON), *EXECUTE* goes low, or an *Error* occurs.
- If another block takes control of the axis while the block is running a command error will occur. The *ERROR* output will go on, but the RDA will not indicate an error.
- The S-curve, Acceleration, Deceleration and the Speed will be set based on the values in the RDA, set by Change Dynamics Block.
- Seventeen words are used as working registers for this function, starting at the address in *Data17W*.

<MOVEABS> Input and Output Register Map

Output Registers

The following registers are used as outputs from the function block. They can be monitored by the LadderWorks program to check the execution of the function.

OUTPUT	TYPE	Content
RUNNING	Bit	High while block has control and move is in progress.
DONE	Bit	Indicates Move Complete(IB**0C1=ON). When the servo has reached target position within "In-position" window(target position ± positioning completed width), this output bit turns ON.
ERROR	Bit	Latches high if any error occurs in block (see table below) or on the servo axis (see IL**04). This is Reset if EXECUTE bit goes low.
DISTGONE	Long	Distance in counts axis has traveled in this move, while this block has control.

Input Registers

The following registers are used as inputs to the function block. They select the options and define the parameters that the user needs to make the function work as necessary

Input	Type	Description	Range and state
EXECUTE	Bit	Block enable – Block cannot execute unless this is TRUE	Rising edge initiates block TRUE – Block enable FALSE – Block disable, set all outputs to zero and off
AXIS	Word	Axis number related to the block	1～16
DIRECTIO	Word	Direction to travel	1 = Positive direction 2 = Negative direction *3 = Continue in current direction *4 = shortest path *: Only can be used if axis is in rotary mode.
POSITION	Long	Absolute position in counts for axis to go to.	-2147483648 to 2147483647 [0 to 2147483647 for rotary axis]
DATA17W	**Address**	Address of the first working register.	Seventeen words of register space are used by this function.

<MOVEABS> Block error condition:

The following table outlines several situations that may cause an error, and will turn on the blocks “Error” output bit. The block “Error” output will cleared if the *EXECUTE* bit goes low.

Internal error bit	Cause	Attention
inRng AB000005	Direction input set out of range	The internal register is reverse logic, it must be ON to indicate the direction input is within valid range. Sets RDA Error ID (MW3**81) bit 3 on if error state exists.
errStop AB000143	Servo Error while in motion	If the error occurs in the servo while the function is operating, this relay becomes on. Bit B of RDA Error ID(MW3**81) is turned on.
axisInErr AB00014A	The axis number entered on the input is not an acceptable value	The function blocks can only control 1 to 16 axis. Any value greater or smaller then this will cause an error. This does not set the RDA Error ID.
commanderr AB00000E	Another block took control of the axis.	If the block looses control of the axis while running an error will occur. This does not set the RDA Error ID.
direrror AB00000F	The direction commanded to travel is disabled from the SVON.	The SVON block must be used and have the direction commanded enabled. Sets RDA Error ID (MW3**81) bit 4 on if error state exists.

<MOVEABS> Working Register

This table outlines the data in the eighteen registers used by the function block. There is not usually any need for the user to access any of these bits directly.

Register No	TYPE	Name	Content
AW00000			
Bit 0	IN	execute	*Enable* input (XB000000)
Bit 1	IN	shortest	Shortest direction mode is selected
Bit 2	Working	posDir	Positive direction mode is selected
Bit 3	Working	movback	If shortest path is selected and reverse direction is shortest
Bit 4	Working	movfor	If shortest path is selected and positive direction is shortest
Bit 5	Working	inRng	The input of DIRECTIO and POSITION is outside ranges.
Bit 6	Working	negDir	Negative direction mode is selected
Bit 7	OUT	running	Directly controls YB000000 (*Running* Output)
Bit 8	OUT	complete	Directly controls YB000001 (*Done* Output)
Bit 9	OUT	error	Directly controls YB000002 (*Error* Output)
Bit A	Working	oneshotA	Reserved.
Bit B	Working	firstPass	One shot coil for initiating axis
Bit C	Working	correntDir	Current direction mode is selected
Bit D	Working	movingneg	If in current direction mode and axis is currently going in negative direction.
Bit E	Working	commanderr	Rising Edge indicates another block took control.
Bit F	Working	direrror	Indicates direction commanded is not enabled.

AW00001			
<bit> 0	Working	oneshot10	Reserved
AL00002	Working	startPos	Commanded position of axis at execution of block.
AL00004	Working	distGone	Distance traveled by axis under the control of this block
AL00006	Working	movPos	Commanded Position
AL00008	Working	encoder_start	Fed back position at start.
AL00010	Working	mcHalf	½ machine cycle
AL00012	Working	negMChalf	Negative ½ machine cycle
AW00014			
Bit 0	Working	posjump	Reserved
Bit 1	Working	negJump	Reserved
Bit 2	Working	oneshot2	Reserved
Bit 3	Working	errStop	Goes high if servo faults out while running.
Bit 4	Working	oneshot4	Reserved
Bit 5	Working	comp_oneshot	Rising pulse of Positioning complete
Bit 6	Working	poswrap	Reserved
Bit 7	Working	negWrap	Reserved
Bit 8	Working	oneshot148	Reserved
Bit 9	Working	inrngAxis	Goes high for one scan if Axis input is in range.
Bit A	Working	axisInErr	Latches high if Axis input is out of range.
Bit B	Working	onePass	One shot coil for initializing axis data
Bit C	Working	oneshot14c	Reserved
Bit D	Working	run	Run Command.
Bit E	Working	pos_comp	Confirmation of positioning completed.
Bit F	Working	closePass	Internal use for the next time execution.
AW00015	Working	rDAmult	Value for address offset to locate proper RDA
AW00016	Working	Revision	Revision Level of the function block.

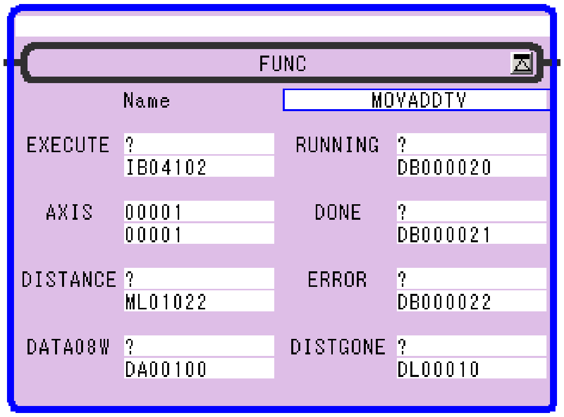
MOVE ADDITIVE function

Function block for MP2000 series

<MOVADDTV> Function Block Summary

This function block adds the set distance to the currently designated target position on the fly of discontinuous motion (MOVABS, MOVINC etc.).

Function Block Diagram



FUNC

Name	MOVADDTV		
EXECUTE	? IB04102	RUNNING	? DB000020
AXIS	00001 00001	DONE	? DB000021
DISTANCE	? ML01022	ERROR	? DB000022
DATA08W	? DA00100	DISTGONE	? DL00010

<MOVADDTV> Function Block operation Notes

- This block must be in H drawing, and adds the set distance to the currently designated target position on the fly of discontinuous motion (MOVABS, MOVINC etc.).
- Rising edge of EXECUTE input initiates block operation and input data of AXIS and DISTANCE are read. The EXECUTE input must be held ON while in operation.
- The RUNNING output bit will be held high until: the positioning completed, EXECUTE=Low or Error.
- The DONE bit turns on when positioning is completed (IB**0C1=ON). When distributing target position is done and the current position is within the range of the positioning completed (Original target position + DISTANCE – Position Feedback < ± Positioning complete width), the positioning completed bit becomes High.
- If another block takes control of the axis while the block is running, a command error will occur. The ERROR output will go on, but the RDA will not indicate an error.
- The S-curve, Acceleration, Deceleration, and the speed will be set based on the values in RDA, set by Change Dynamics Block.
- Eight words are used as working registers for this function, starting at the address in *Data08W*.

<MOVADDTV> Input and Output Register Map

Output Registers
The following registers are used as outputs from the function block. They can be monitored by the LadderWorks program to check the execution of the function.

Output	Type	Description
RUNNING	Bit	High while block has control and move is in progress.
DONE	Bit	Indicates Move Complete. When the servo has reached target position within "In-position" window, this output bit turns ON. (IB**0C1=ON).
ERROR	Bit	Latches high if any error occurs in block (see table below) or on the servo axis (see IL**04).
DISTGONE	Long	Distance in counts block has traveled in this move, while this block has control.

Input Registers
The following registers are used as inputs to the function block. They select the options and define the parameters that the user needs to make the Home function work as necessary.

Input	Type	Description	Range of state
EXECUTE	Bit	Block enable – Block cannot execute unless this is TRUE	Rising edge initiates TRUE – Block enable FALSE – Block disable, set all outputs to zero and off
AXIS	Word	Axis number related to the block	1～16
DISTANCE	Long	Additional distance to current target position.	- 2147483648～2147483647
DATA08W	Address	Address of the first working register.	Eight words of register space are used by this function.

<MOVADDTV> Block Fault Conditions:

The following table outlines several situations that may cause an error, and will turn on the blocks “Error” output bit. The block “Error” output will cleared if the *EXECUTE* bit goes low.

Internal error bit	Cause	Attention
errStop AB00000D	Servo Error	If block is running and the servo faults the error bit will be set. Sets RDA Error ID (MW3**81) bit B on if error state exists the
axisInErr AB000002	The axis number entered on the input is not an acceptable value	The function blocks can only control 1 to 16 axis. Any value greater or smaller then this will cause an error. This does not set the RDA Error ID.
Commanderr AB00000E	Another block took control of the axis..	If the block looses control of the axis while running an error will occur. This does not set the RDA Error ID.
Direrror AB00000F	The direction commanded to travel is disabled from the SVON block.	The SVON block must be used and have the direction commanded enabled. Sets RDA Error ID (MW3**81) bit 4 on if error state exists.

<MOVADDTV> Working Registers

This table outlines the data in the nine registers used by the function block. There is not usually any need for the user to access any of these bits directly.

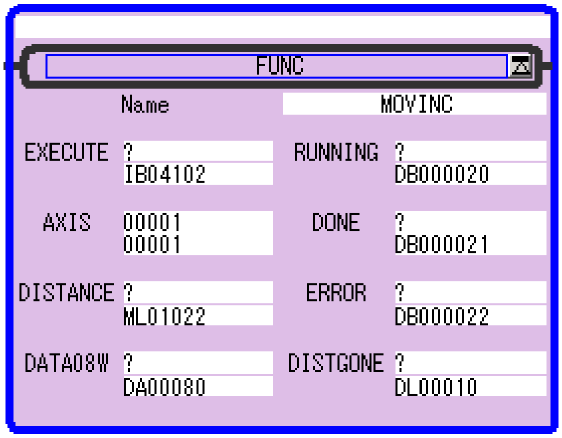
Register No	TYPE	Name	Content
AW00000			
Bit 0	IN	execute	*EXECUTE* input (XB000000)
Bit 1	IN	onepass	One shot on rising edge of EXECUTE input to set axis data.
Bit 2	Working	axisInErr	Latches high if Axis input is out of range.
Bit 3	Working	oneshot3	Reserved
Bit 4	Working	inrngaxis	Goes high for one scan if Axis input is in range.
Bit 5	Working	oneshot5	Reserved.
Bit 6	Working	completeSet	Rising pulse of Positioning complete.
Bit 7	OUT	running	Directly controls YB000000 (*RUNNING* Output)
Bit 8	OUT	complete	Directly controls YB000001 (*DONE* Output)
Bit 9	OUT	error	Directly controls YB000002 (*EEROR* Output)
Bit A	Working	oneshotA	Reserved
Bit B	Working	firstPass	One shot on rising edge of EXECUTE input to initialize axis.
Bit C	Working	oneshotC	Reserved
Bit D	Working	errStop	Goes ON when servo error occurs while running.
Bit E	Working	commanderr	Rising Edge indicates another block took control
Bit F	Working	direrror	Indicates direction commanded is not enabled..
AW00001			
Bit 0	Working	oneshot10	Reserved
Bit 1	Working	closepath	Function internal processing for execution next time.
Bit 2	Working	second_Pass	Reserved
Bit 6	Working	position_Mode	Confirmation of positioning mode.
Bit 7	Working	move_Done	Confirmation of positioning completed.
Bit D	Working	oneshot1D	Reserved
AL00002	Working	startpos	Position of axis at start of move.
AL00004	Working	distGone	Directly controls YL000001 (*DISTGONE* Output)
AW00006	Working	rdaMulti	Value for address offset to locate proper RDA
AW00007	Working	revision	Revision Level of the function block.

MOVE INCREMENTAL function
Function block for MP2000 series

<MOVINC> Function Block Summary

This function block commands a controlled motion of a specified distance relative to the actual position at the time of the execution.

Function Block Diagram



FUNC

Name			MOVINC
EXECUTE	? IB04102	RUNNING	? DB000020
AXIS	00001 00001	DONE	? DB000021
DISTANCE	? ML01022	ERROR	? DB000022
DATA08W	? DA00080	DISTGONE	? DL00010

<MOVINC> Function Block Operation Notes

- MOVINC must be in H drawing, and to be executed after execution of SVON and CHANGE DYNAMICS function.
- Rising edge of EXECUTE input initiates block operation, and all block input values are read.
- The *RUNNING* output bit will be held high until: move is completed, *EXECUTE* goes low, or an *Error* occurs
- DONE bit turns on when positioning is completed (IB**0C1=ON). When distributing target position and the current position is within the range of the positioning complete (target position ± positioning completed width), the positioning completed becomes High.
- If another block takes control of the axis while the block is running a command error will occur. The *ERROR* output will go on, but the RDA will not indicate an error.
- The S-curve, Acceleration, Deceleration and the Speed will be set based on the values in the RDA, set by Change Dynamics Block.
- Eight words are used as working registers for this function, starting at the address in *Data08W*.

<MOVINC> Input and Output Register Map

Output Registers

The following registers are used as outputs from the function block. They can be monitored by the LadderWorks program to check the execution of the function.

Output	Type	Description
RUNNING	Bit	High while block has control and move is in progress..
DONE	Bit	Indicates Move Complete. When the servo has reached target position within "In-position" window ON (IB**0C1=ON), this output bit turns.
ERROR	Bit	Latches high if any error occurs in block (see table below) or on the servo axis (see IL**04).
DISTGONE	Long	Distance in counts block has traveled in this move, while this block has control.

Input Registers

The following registers are used as inputs to the function block. They select the options and define the parameters that the user needs to make the Home function work as necessary.

Input	Type	Description	Range and state
EXECUTE	Bit	Block enable – Block cannot execute unless this is TRUE.	Rising edge initiates TRUE – Block enable FALSE – Block disable, set all outputs to zero and off
AXIS	Word	Axis number related to the block.	1～16
DISTANCE	Long	Relative distance to travel in counts to complete move.	-2147483648～2147483647
DATA08W	Address	Address of the first working register.	Eight words of register space are used by this function.

<MOVINC> Block error condition:

The following tables might cause the error, and outline some situations. If the EXECUTE input bit is turned off, the value is cleared.

Internal error bit	Cause	Attention
errStop AB00000D	Motion commanded and servo is faulted	If block is running and the servo is faulted the error bit will be set. Sets RDA Error ID (MW3**81) bit B on if error state exists.
axisInErr AB000002	The axis number entered on the input is not an acceptable value	The function blocks can only control 1 to 16 axes. Any value greater or smaller then this will cause an error. This does not set the RDA Error ID.
Commanderr AB00000E	Another block took control of the axis	If the block looses control of the axis while running an error will occur. This does not set the RDA Error ID.
Direrror AB00000F	The direction commanded to travel is disabled from the SVON block.	The SVON block must be used and have the direction commanded enabled. Sets RDA Error ID (MW3**81) bit 4 on if error state exists.

<MOVINC> Working Register

This table outlines the data of the register of eight that the function block use is done. The user is, and there is not a necessity and either is usually no what directly accessed any of these bits.

Register No	TYPE	Name	Content
AW00000			
Bit 0	IN	execute	*EXECUTE* input (XB000000)
Bit 1	IN	onepass	One shot on rising edge of EXECUTE input to set axis data.
Bit 2	Working	axisInErr	Latches high if Axis input is out of range.
Bit 3	Working	oneshot3	Reserved
Bit 4	Working	inrngaxis	Goes high for one scan if Axis input is in range.
Bit 5	Working	oneshot5	Reserved
Bit 6	Working	completeSet	Reserved
Bit 7	OUT	running	Directly controls YB000000 (*RUNNING* Output)
Bit 8	OUT	complete	Directly controls YB000001 (*DONE* Output)
Bit 9	OUT	error	Directly controls YB000002 (*EEROR* Output)
Bit A	Working	oneshotA	Reserved
Bit B	Working	firstPass	One shot on rising edge of EXECUTE input to initialize axis.
Bit C	Working	oneshotC	Reserved
Bit D	Working	errStop	Indicates axis faulted and is commanded to move.
Bit E	Working	commanderr	Rising Edge indicates another block took control.
Bit F	Working	direrror	Indicates direction commanded is not enabled.
AW00001			
Bit 0	Working	oneshot10	Reserved
Bit 1	Working	closepath	Internal use for the next execution.
Bit 2	Working	second_Pass	Reserved
Bit 6	Working	position_Mode	Confirmation of positioning mode.
Bit 7	Working	move_Done	Confirmation of positioning completed.
Bit D	Working	oneshot1D	Reserved
AL00002	Working	startpos	Position of axis at start of move.
AL00004	Working	distGone	Directly controls YL000001 (*DISTGONE* Output)
AW00006	Working	rdaMulti	Value for address offset to locate proper RDA
AW00007	Working	revision	Revision Level of the function block.

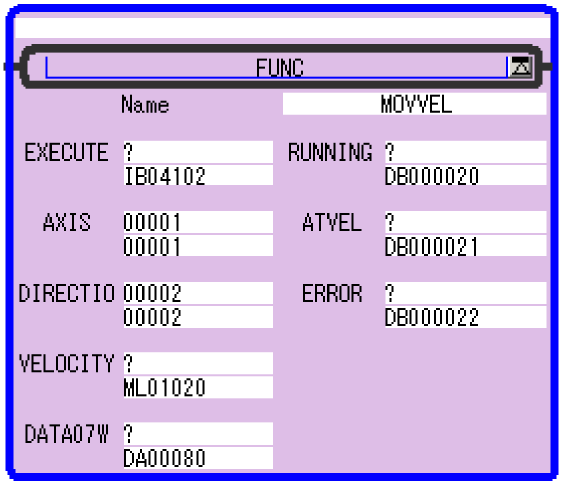
MOVE VELOCITY “JOG” function

Function block for MP2000 series

<MOVVEL> Function Block Summary

This function block commands a never ending controlled motion at a specified velocity.

Function Block Diagram



FUNC			
Name		MOVVEL	
EXECUTE	? IB04102	RUNNING	? DB000020
AXIS	00001 00001	ATVEL	? DB000021
DIRECTIO	00002 00002	ERROR	? DB000022
VELOCITY	? ML01020		
DATA07W	? DA00080		

<MOVVEL> Function Block Operation Notes

- MOVVEL must be in H drawing and to be executed after execution of SVON and CHANGE DYNAMICS function.
- Rising edge of EXECUTE input initiates block operation, and all block input values are read once
- If the *EXECUTE* bit goes low the outputs will be set to zero. The block will loose control of the axis and the axis will continue to move at the set speed originally specified by the block.
- Motion is stopped by executing STOP block.
- Seven words are used as working registers for this function, starting at the address in *Data07W*.

<MOVVEL> Input and Output Register Map

Output Registers

The following registers are used as outputs from the function block. They can be monitored by the LadderWorks program to check the execution of the function.

Output	Type	Description
RUNNING	Bit	High while block has control and move is in progress.
ATVEL	Bit	This bit is ON when the Axis feedback speed is at +/- 1% of the set speed.
ERROR	Bit	Latches high if any error occurs in block (see table below) or on the servo axis (see IL**04).

Input Registers

The following registers are used as inputs to the function block. They select the options and define the parameters that the user needs to make the function work as necessary.

Input	Type	Description	Range and state
EXECUTE	Bit	Block enable – Block cannot execute unless this is TRUE	Rising edge initiates block TRUE – Block enable FALSE – Block disable, set all outputs to zero and off
AXIS	Word	Axis number related to the block	1～16
DIRECTIO	Word	Direction to travel	1 = Positive direction 2 = Negative direction 3 = Continue in current direction.
VELOCITY	Long	Speed in counts per second to travel.	0 to Max. velocity (MW3**12) in RDA
DATA07W	**Address**	Address of the first working register.	Seven words of register space are used by this function.

<MOVVEL> Block Fault Condition:

The following table outlines several situations that may cause an error, and will turn on the blocks “Error” output bit. The block “Error” output will cleared if the *EXECUTE* bit goes low

Internal error bit	Cause	Attention
inerr AB000004	DIRECTIO and VELOCITY input are out of range.	Value must be in range or an error will occur. Sets RDA Error ID (MW3**81) bit 3 on if error state exists.
inspeed AB00000C	VELOCITY input is out of range.	Value must be in range or an error will occur.
mover AB00000E	Direction input is set to current direction and axis is not moving.	The axis has to be moving in a direction for this block to work in ‘Continue in current direction’ mode. This does not set the RDA Error ID.
errStop AB000042	Motion commanded and servo is faulted.	If block is running and the servo is faulted the error bit will be set. Sets RDA Error ID (MW3**81) bit B on if error state exists.
axisInErr AB000045	The axis number entered on the input is not an acceptable value	The function blocks can only control 1 to 16 axes. Any value greater or smaller then this will cause an error. This does not set the RDA Error ID.
commanderr AB000041	Another block took control of the axis	If the block looses control of the axis while running an error will occur. This does not set the RDA Error ID.
direrror AB00000F	The direction commanded to travel is disabled from the SVON block.	The SVON block must be used and have the direction commanded enabled. Sets RDA Error ID (MW3**81) bit 4 on if error state exists.

<MOVVEL> Working Registers

This table outlines the data in the eight registers used by the function block. There is not usually any need for the user to access any of these bits directly.

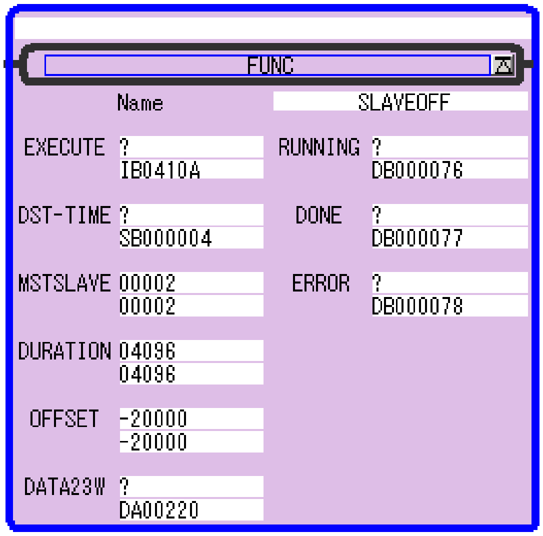
Register No.	Type	Name	Description
AW00000			
Bit 0	IN	execute	*EXECUTE* input (XB000000)
Bit 1	Working	positive	Positive direction mode is selected
Bit 2	Working	negative	Negative direction mode is selected
Bit 3	Working	inrng2	Verifies *DIRECTIO* input is in range.
Bit 4	Working	inerr	Goes High if inrng2 does not go high when block is executed.
Bit 5	Working	currnet	Current direction mode is selected
Bit 6	Working	movepos	If current direction mode is selected and axis is moving in a positive direction.
Bit 7	OUT	running	Directly controls YB000000 (*RUNNING* Output)
Bit 8			Reserve.
Bit 9	OUT	error	Directly controls YB000002 (*ERROR* Output)
Bit A	Working	oneshotA	Reserved
Bit B	Working	firstPass	One shot coil for initializing axis. Goes High with rising edge of EXECUTE input.
Bit C	Working	inspeed	Velocity selected is out of range.
Bit D			Reserved
Bit E	Working	moverr	Latches on when current direction is called for and axis is not moving
Bit F	Working	direrror	SVON block did not allow this direction for the move
AL00002	Working		Speed set to RDA and controller.
AW00004			
Bit 0	Working	oneshot10	Reserved
Bit 1	Working	commanderr	Rising Edge indicates another block took control.
Bit 2	Working	errStop	Goes high if servo faults out while running
Bit 3	Working	onepass	One shot coil for initiating axis data
Bit 4	Working	inRngAxis	Goes high for one scan if Axis input is in range.
Bit 5	Working	axisInErr	Goes high for one scan if Axis input is out of range.
Bit 6	Working	oneshot46	Reserved.
Bit 7	Working	closepass	Internal use for the next execution.
Bit 8	Working	atSpd	Directly controls YB000001(ATVEL Output)
Bit A	Working	oneshot4A	Reserved.
Bit D	Working	oneshot4D	Reserved.
AW00005	Working	rdaMulti	Value for address offset to locate proper RDA
AW00006	Working	revision	Revision Level of the function block.

SLAVE OFFSET function
Function block for MP2000 series

<SLAVEOFF> Function Block Summary

The Slave Offset (SLAVEOFF) block is used to shift the slave axis while gearing or camming. The rising edge of the *EXECUTE* input bit will cause the *OFFSET* value to be added to the RDA offset at the rate determined by the *DURATION* input. *DST-TIME* input bit determines whether the shift duration is base upon distance the master travels or based on time in milliseconds.

Function Block Diagram



Name			SLAVEOFF
EXECUTE	? IB0410A	RUNNING	? DB000076
DST-TIME	? SB000004	DONE	? DB000077
MSTSLAVE	00002 00002	ERROR	? DB000078
DURATION	04096 04096		
OFFSET	-20000 -20000		
DATA23W	? DA00220		

<SLAVEOFF> Function Block Operation Notes

- Rising edge of EXECUTE input initiates block operation, and all block input values are read once.
- To use the function block, the *EXECUTE* bit must be held ON. If the *EXECUTE* bit goes off during operation, the block will stop executing (will not complete the change), the offset up to that point will remain, and all outputs will be set to zero.
- The *RUNNING* output bit will be held high until the Offset move is completed, *EXECUTE* goes low, or an *Error* occurs.
- The *DST-TIME* bit must be set to TRUE to use the master position change or FALSE for the time duration.
- Distance mode note: Be sure to set the duration in the same direction as the master movement. Example: If the block is in Distance mode and the master is moving in the negative direction, and the DURATION is set positive, it will subtract the relative offset and hence the resulting offset will not complete. As a result, the DONE output will not go on because the full move is not complete in the correct direction.
- The value <'of block in execution (Block Running)> is indicated in master/slave pair part RDA(MW56**6).
- Issuing a STOP block on the slave axis will have no affect on the OFFSET block execution.
- Note that the master/slave pairs are separated by 50 words in the RDA (up to 10 pairs). Example: Master/Slave pair #1 starts at MW56000, Master/Slave pair #2 starts at MW56050, Master/Slave pair #3 starts at MW56100, etc.
- Twenty-three words are used as working registers for this function, starting at the address in *Data23W*.

<SLAVEOFF> Input and Output Register Map

Output Registers

The following registers are used as outputs from the function block. They can be monitored by the LadderWorks program to check the execution of the function.

Output	Type	Description
RUNNING	Bit	Goes high while offset value is being updated in RDA and there are no errors
DONE	Bit	Goes high when offset change is complete in RDA
ERROR	Bit	Goes high if any block errors occur

Input Registers

The following registers are used as inputs to the function block. They select the options and define the parameters that the user needs to make the Home function work as necessary.

INPUT	TYPE	Content	Range of setting and state
EXECUTE	Bit	Block enable – Block cannot execute unless this is TRUE.	Rising edge initiates block TRUE – Block enable FALSE – Block disable, set all outputs to zero and off
DST-TIME	Bit	Selects master position [count] or time [ms] for offset duration.	TRUE – Master position is used for duration FALSE – Time is used for duration
M-S-PAIR	Word	Defines which Master-Slave pair is updated in RDA	1 to 10 value. Any other value will cause the error output to go on
DURATION	Long	Depends on DST-TIME setting. Either time or distance master must travel to complete change in offset.	Distance: 2147483648～2147483647 [count] Time: 0-2147483647 [ms]
OFFSET	Long	Total value offset in RDA will change.	-2147483648～2147483647
DATA23W	**Address**	Address of the first working register.	Twenty-three words of register space are used by this function.

<SLAVEOFF> Block Fault Condition:

The following table outlines several situations that may cause an error, and will turn on the blocks “Error” output bit. The block “Error” output will cleared if the *EXECUTE* bit goes low.

Internal Fault Bit	Cause	Note
inrng1 AB00000E	MSTRSLV value out of range	Inrng1 must be high if execute is on or an error will occur. MSTRSLV must have a value from one to ten for this bit to be on. Sets RDA Error ID (MW30181) bit 3.

<SLAVEOFF> Working Register

This table outlines the data in the eight registers used by the Home function block. There is not usually any need for the user to access any of these bits directly.

Register No	TYPE	Name	Content
AW00000			
Bit 0	IN	execute	*EXECUTE* input (XB000000)
Bit 1	IN	distmode	*DST-TIME* input (XB000001)
Bit 3	Working	zerodiv	If duration is set to zero for one scan change, this turns on for one scan (one-shot)
Bit 6	Working	distShft_init	High for one scan to initialize for master position shifting if first pass is high and DISTMODE is high. To initialize for master position control of offset.
Bit 7	OUT	running	Directly controls YB000000 (*RUNNING* Output)
Bit 8	OUT	complete	Directly controls YB000001 (*DONE* Output)
Bit 9	OUT	error	Directly controls YB000002 (*ERROR* Output)
Bit A	Working	oneshotA	Resered
Bit B	Working	firstPass	High for first scan of execute being high
Bit D	Working	timeshft_init	One shot coil for resetting outputs. Controlled by (AB00000A)
Bit E	Working	inrng1	High for one scan to initialize for time based shifting if first pass is high and DISTMODE is low.
AW00001	Working	scans_per_shift	Number of scans needed to complete a time based shift.
AW00002	Working	iStore	Pass through value for i.
AW00003	Working	mstrSlv	MSTSLAVE input (XW000001) – 1.
AL00004	Working	maxShift_scan	Maximum counts allowed to shift per scan on a time based scan.
AL00006	Working	posLAUlim	Positive value of maxShift_scan
AL00008	Working	negLAUlim	Negative value of maxShift_scan
AL00010	Working	total_Shifted	Amount Slave Offset has changed during the execution of the block.
AL00012	Working	shiftvalue	Value to change the RDA (ML56**6) Slave offset for that scan.
AL00014	Working	startmaster	Value of Master counter at start of block (AB0006=ON) for position based move.
AL00018	Working	desired_shift	Total amount needed to be shifted for the master position mode.
AL00020	Working	timemod	Reserved.
AW00022	Working	Revision	Revision Level of the function block.

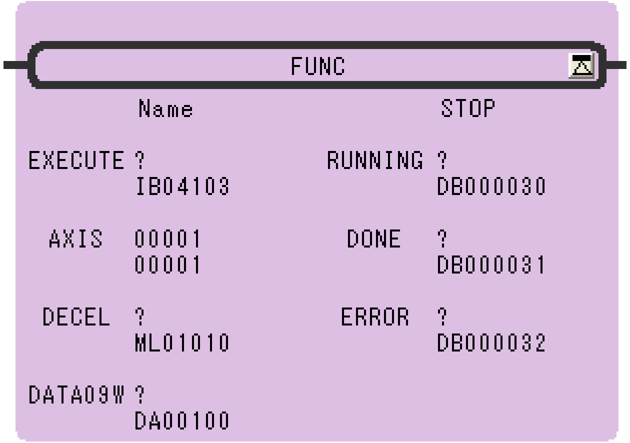
STOP function

Function block for MP2000 series

<STOP> Function Block Summary

This function block commands a controlled motion stop and transfers the axis to the state “Stopping”. It aborts any ongoing function block execution. With the DONE output set, the state is transferred to the Stand Still. While the axis is in state Stopping, no other FB can perform any motion on the same axis.

Function Block Diagram



```
                        FUNC
        Name                     STOP

EXECUTE ?              RUNNING ?
        IB04103                  DB000030

  AXIS  00001             DONE   ?
        00001                    DB000031

 DECEL  ?                ERROR   ?
        ML01010                  DB000032

DATA09W ?
        DA00100
```

<STOP> Function Block Operation Notes

- Rising edge of EXECUTE input initiates block operation, and all block input values are read once.
- To use the function block, the *EXECUTE* bit must be held ON. If the *EXECUTE* bit goes off during operation, all outputs will go low but the block will still stop the axis and prevent any other motion block from executing until axis has stopped.
- The stop block will force the axis operation mode to Speed mode (OW**08=7) and decelerate to zero velocity at a rate defined by the RDA Set Deceleration value (ML3**24)
- No other motion blocks can take control while this block is under execution or EXECUTE input remains high.
- Nine words are used as working registers for this function, starting at the address in *Data09W*.

<STOP> Input and Output Register Map

Output Registers

Output	Type	Description
RUNNING	Bit	High while block has control and move is in progress.
DONE	Bit	Goes high when the axis comes to a full stop. IE. actual speed feedback in RDA is zero (ML3**16=0), and motion command mode feedback indicates zero (IW3**16=0) When the axis stops completely,
ERROR	Bit	Latches high if any error occurs in block (see table below) or on the servo axis (see IL**02 and IL**04). This is Reset if EXECUTE bit goes low.

Input Registers

The following registers are used as inputs to the function block. They select the options and define the parameters that the user needs to make the function work as necessary.

Input	Type	Description	Range and state
Execute	Bit	Block enable	Rising edge executes block TRUE – nothing else can take control of the axis FALSE & Block Done - Block disabled, all outputs set to zero
AXIS	Word	Axis number related to the block.	1～16
DECEL	Long	Deceleration during stop in counts per second squared.	1 to Max. Deceleration in RDA, (ML3**28).
DATA09W	Address	Address of the first working register.	Nine words of register space are used by this function.

<STOP> Block Fault Condition

The following table outlines several situations that may cause an error, and will turn on the blocks "Error" output bit. The block "Error" output will cleared if the *EXECUTE* bit goes low, but the Error ID (MW3**81) will remain in the RDA. To reset the Error ID, use the Alarm Reset Function Block.

Note that each axis has its own Error ID stored in its RDA axis section, offset by 200 for each axis. Example: Axis#1 stores to MW30181, Axis #2 stores to MW30281, etc.

Internal error bit	Cause	Attention
axisinerr AB00000F	The axis number entered on the input is not an acceptable value	The function blocks can only control 1 to 16 axis. Any value greater or smaller then this will cause an error. This does not set the RDA Error ID.

<STOP> Working Registers

This table outlines the data in the seven registers used by the function block. There is not usually any need for the user to access any of these bits directly.

Register No	TYPE	Name	Content
AW00000			
Bit0	IN	execute	*EXECUTE* input (XB000000)
Bit1	Working	InRangeAxis	The set Axis number is within 1 through 16.
Bit2	OUT	Running	Directly controls YB000000 (*RUNNING* Output)
Bit3	OUT	done.	Directly controls YB000001 (*DONE* Output)
Bit4	IN	Interruption	If aborting from Discrete motion (RDA axis state MW3**49 was 4), this bit goes on
Bit5	Working	inrng	The set deceleration rate is in normal range.
Bit6	Working	Oneshot06	Reserved
Bit7	Working	closePass	One scan ON at Execute input transition On to Off.
Bit8	Working	Execute2	On while Execution is demanded.
BitA	Working	oneshotA	Reserved
BitB	Working	firstPass	One shot coil to initialize axis and block.
BitC	Working	oneshot	Reserved
BitF	Working	axisInErr	Goes high if Axis input is out of range.
AW00001	Working	Filter	Reserved
Bit3	Out	Complete	On when Stoping is completed
AF00002	Working	speedMEMO	Speed Memory at Stopping starts[count/sec]
AW00004	Working	rdaMult	Value for address offset to locate proper RDA
AL00006	IN	DECEL	Set Deceleration (XL0002)
AW0008	Work	revision	

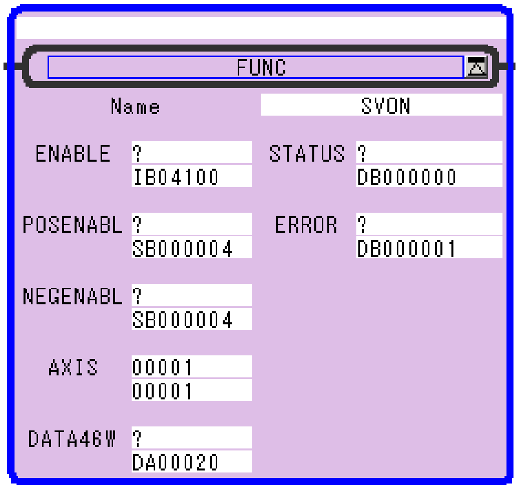
SERVO ENABLE POWER function

Function block for MP2000 series

<SVON> Function Block Summary

This function block controls the power stage (on or off) of the axis as well as monitoring and updating the RDA with Actual Position, Actual Velocity, Actual acceleration, Actual deceleration and Actual S-curve. This block is essential if any of the Blocks that control motion to be used. This block also enables the user to limit the direction the axis can go..

Function Block Diagram



FUNC

Name		SVON	
ENABLE	? IB04100	STATUS	? DB000000
POSENABL	? SB000004	ERROR	? DB000001
NEGENABL	? SB000004		
AXIS	00001 00001		
DATA46W	? DA00020		

<SVON> Function Block Operation Notes

- This block must be used in the High Speed Scan drawings, and should be the first block executed before any other motion function blocks.
- Interlock parameters used inside SVON block, the servo will not enable until all of the following conditions are clear:
  - SVCRDY (IB**000):Motion Controller SVB/SVA Module Run Ready. This is high when the run preparation for motion module have been completed. This is low when major fault occurred, axis set to not used, motion fixed parameter setting error or has changed, in asynchronous communication state and MPE720 are accessing the servo amplifier parameter and the motion parameter screen is opened by MPE720.
  - ALM (IB**2C0):Servo Axis alarm bit (high state indicates alarm occurred)
  - PON (IB**2C4):Servo amplifier main power ON. This is high when the main circuit power of the servo amplifier is on.
  - HS-1SEC (SB000018): H-Scan delay-on service register (off for one second after first H-Scan executes when control power is applied. After one second it is always on).
- Upon Rising edge of ENABLE, the following two events will occur. ENABLE input must remain high to maintain servo enable.

  - Servo will enable only if axis is normal (SVCRDY=high, ALM=low, PON=high), and HS-1SEC=high.
  - The S-Curve filter type will be set to moving average.
- If the *ENABLE* input goes off during operation, the servo will be disabled but its position in the RDA will continue to update. Note that all outputs will be off or zero.
- If a servo fault occurs and the servo is enabled the ERROR output will go high.
- Positive and Negative direction operation enable inputs (POSENABL and NEGENABL) inputs are constantly monitored when the ENABLE input is ON, and sets the RDA (MB003**770 servo on, MB03**771 pos enable, MB3**772 neg enable). Each motion block will then monitor the RDA status bits to insure proper direction operation.
- All of the motion function blocks rely on data computed in this block (Actual Position, Actual Velocity, Actual acceleration, Actual deceleration, Actual S-curve, updates latched position, handles rotary mode positioning, monitors over-travel status). Data computed in this block is stored to the RDA.
- Input feedback from specific axis registers are read into the RDA from this function block. Example – for axis #1 the following is read and written to RDA:
  - IW8000 run status (motion controller operation ready, servo enabling, system busy, servo ready)
  - IW800C position management status (position compete, latch complete, home complete).
  - IW802C network servo status (servo alarm, servo enabled, main power on, pos complete, latch complete, speed lim, etc)
  - IL8010 target position (CPOS)
  - IL8018 latch position (LPOS)
  - IL8016 motor actual feedback position (APOS)
  - IW8009 motion command status (command executing, holding, error, reset abs enc done).
  - IW8008 motion command code response
  - IL8004 alarm (servo err, POT, NOT, soft POT, soft NOT, excessive spd or following error or timeout, homing error, servo comm. err, servo miss-match)
  - IL8002 Warnings (parameter setting warning, following error warning, comm. warning, POT/NOT warnings, servo enable warning)
  - IW802D servo alarm code
  - IW802E servo DI monitor (CN1 input signal status level)
  - IL8020 speed reference
  - IL8040 feedback speed

For more information on these registers, see MP2300/2200 Motion Module Users Manual SIEPC88070016A, section 4.3.3 (Motion Monitoring Parameter Details.

See below for a table of RDA values:

Table: Axis #1 RDA values from SVON Axis Mapping

Run Status-1	MW30200	IW8000 – Run Status
Run Status-2	MW30201	IW800C – Position Status
Servo Status	MW30202	IW802C – Servo Status
	MW30203	–
Target Position	ML30204	IL8010 – Target Position
Latched Position	ML30206	IL8018 – Latched Position
Actual Position	ML30208	IL8016 – Acutal Position
Command Status	MW30210	IW8009 – Motion Command Status
Command Response	MW30211	IW8008 – Motion Command Response
Alarms	ML30212	IL8004 – Alarm
Warning	ML30214	IL8002 – Warning
Servo Alarm Code	MW30216	IW802D – Servo Alarm Code
Servo DI Monitor	MW30217	IW802E – Servo DI Monitor
Speed Ref. Monitor	ML30218	IL8020 – Speed Reference Monitor
Feedback Speed	ML30220	IL8040 – Feedback Speed Monitor

- If the servo is disabled while motion is in progress (OW**08<>0), the servo axis will generate an error (IB**2C0), and an error bit will be set in the RDA (MB3**81D). A SVON block fault will not occur until the next enable attempt. To correct this, clear the alarm using the alarm reset function block, BALRMRST.
- The AXISEND block MUST be used in conjunction with this function block.
- Forty-six words are used as working registers for this function, starting at the address in *Data46W*.

<SVON> Input and Output Register Map

Output Registers

The following registers are used as outputs from the function block. They can be monitored by the LadderWorks program to check the execution of the function.

OUTPUT	TYPE	Content
STATUS	Bit	This bit is ON when the servo is enabled (Axis normal, Servo is enabled, S-curve filter is set)
ERROR	Bit	If an error occurs during block execution (see table below) or on the servo axis (see IL**04), this output bit turns ON, but will reset if error condition goes away.

Input Registers

The following registers are used as inputs to the function block. They select the options and define the parameters that the user needs to make the function work as necessary.

INPUT	TYPE	Content	Range of setting and state
ENABLE$_{\circ}$	Bit	Servo enable – This will control the servo enable	Rising Edge – enables servo (see operation notes for conditions) TRUE – Hold Servo enable FALSE – Servo disable
POSENABL	Bit	Enable Axis in the positive direction. This input is always active.	TRUE – Axis can be commanded in the positive direction FALSE – Axis can *not* be commanded in the positive direction
NEGENABL	Bit	Enable Axis in the negative direction. This input is always active.	TRUE – Axis can be commanded in the negative direction FALSE – Axis can *not* be commanded in the negative direction
AXIS	**Word**	Axis number related to the block	1to 16

<SVON> Block Fault Conditions:

The following table outlines several situations that may cause an error. The block error can be cleared by the *ENABLE* bit going low.

Internal error bit	Cause	Attention
rdaError AB000004	The RDA is not properly set up.	If Motor Rated Speed (MW3**73) or Encoder resolution (ML3**74) is not appropriately set up in the RDA, an error will occur. Sets bit A of RDA Error ID (MW3**81) if error state exists.
axisinerr AB000028	The axis number entered on the input is not an acceptable value	The function blocks can only control 1 to 16 axis. Any value greater or smaller then this will cause an error. This does not set the RDA Error ID.
RDA_Error AB000026	The module or axis number in the RDA is not acceptable	RDA module number MW3**83 refers to the address of Module on the rack, and can be from 1 to 16. The axis number MW3**84 refers to the axis location on the module and can be from 1-16. This typically would occur if an RDAINIT function block is not used for that axis.
Servo_Alarm/POWER_OFF IB**2C0/IB**2C4	Servo Alarm	Servo amplifier fault occurred or main power lost. Sets RDA Error ID (MW3**81) bit 6 on if no motor power, and bit 7 if there is a servo alarm.

<SVON> Working Register

This table outlines the data in the forty-five registers used by the function block. There is not usually any need for the user to access any of these bits directly.

Register No	TYPE	Name	Content
AW00000			
Bit 0	IN。	Enable	*ENABLE* input (XB000000)
Bit 1	IN。	posEnable	*POSENABL* input (XB000001)
Bit 2	IN。	negEnable	*NEGENABL* input (XB000002)
Bit 3	IN。	Normal	Servo has main power (IB**2C4), no servo alarm (IB**2C0), and motion controller is ready (IB**000)
Bit 4	IN。	rdaError	Goes high if the RDA registers are not set properly
Bit 5		Moving	High when motion is commanded
Bit 6	OUT。	rotary	Goes high if RDA calls for Rotary mode
Bit 7	OUT。	State	Directly controls YB000000 (*STATUS* Output). Must be normal (AB000003), Servo on must be true (IB**002), and filter set must be on (AB00002C).
Bit 9	OUT。	Mistake	Directly controls YB000001 (*ERROR* Output)
Bit A	Working	osEnable	Reserved
Bit B	Working	firstPass	One shot coil for initializing Axis and block on for the rising edge of enable input
Bit C	Working	osDisable	Reserved
AW00001	Working	Filter	Rreserved.
AW00002			

Bit 0	Working	latchedfirst	High when Latch is detected
Bit 1	Working	rdaSpdErr	RDA error. Motor rated speed is set to zero
Bit 2	Working	rdaResErr	RDA error. Encoder resolution is set to zero
Bit 3	Working	inrngAxis	Goes high for one scan if Axis input is in range.
Bit 4	Working	rdaMODok	Goes high for one scan if RDA Module value is in range.
Bit 5	Working	rdaAXISok	Goes high for one scan if RDA Axis value is in range.
Bit 6	Working	rdaAxisErr	Goes High if RDA module or axis number is not in range.
Bit 7	Working	delaypulse	Goes high for one scan one second after first high scan.
Bit 8	Working	axisinerr	Goes high for one scan if Axis input is out of range.
Bit 9	Working	offSwtch	Goes high for one scan after execute bit goes low or the servo faults out.
Bit A	Working	oneshot38a	Reserved
Bit B	Working	Oneshot2b	Reserved
AL00003	Working	posLast	Axis position on last scan.
AL00005	Working	cntPPulse	Counts moved each scan.
AL00007	Working	scancomp	Adjusted value due to scan time delay.
AL00009	Working	lScratch	Reserved
AL00011	Working	oldTarget	Commanded position Last scan (ILC**10)
AL00013	Working	targetChange	Change in target position between scans
AL00015	Working	oldChange	Change in target position last scan
AL00017	Working	Speed	Actual velocity stored to RDA ML3**16 in Counts per sec.
AW00019	Working	Reservation.	Reserved
AF00020	Working	temp1	Reserved
AF00022	Working	scanTime	Scan time in milliseconds
AL00024	Working	o mpos_o	Reserved
AL00026	Working	oldCommand	Commanded position previous scan
AL00028	Working	deltaCommand	Change from previous scan
AL00030	Working	o mpos_Commanded	Reserved
AL00032	Working	mPosCommanded	Actual Set position stored in RDA ML3**08
AL00034	Working	latchpos	Latched position value converted for rotary mode and stored in RDA ML3**46
AL00036	Working	velChange	Change in velocity from the previous scan
AL00038	Working	oldVelocity	Velocity at previous scan
AL00040	Working	accel	Actual acceleration or deceleration stored to RDA
AW00042	Working	rdaMult	Value for address offset to locate proper RDA
AW00043	Working	axisMult	Value for address offset to locate proper Axis

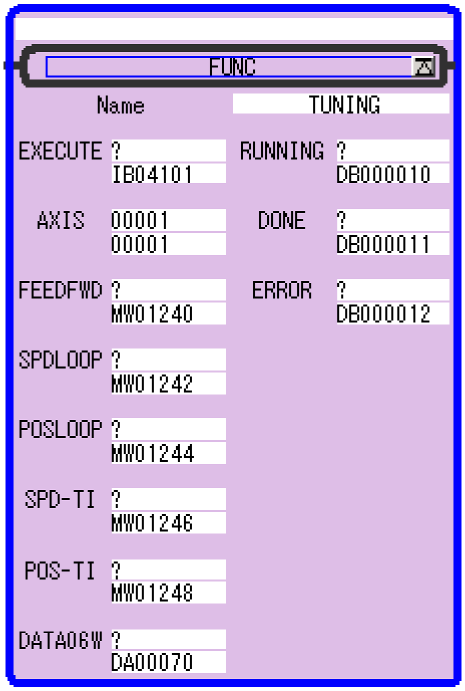
AW00044	Working	Timer	Timer for error detection of servo on
AW00045	Working	Revision	Revision Level of the function block.

TUNING function
Function block for MP2000 series

<TUNING> Function Block Summary

This function block sets the speed feed forward, speed loop gain, position loop gain, speed loop integral time constant and the position loop integral time constant in the motion setup parameter and the servo amplifier user constant.

Function Block Diagram



```
                FUNC
Name              TUNING
EXECUTE ?              RUNNING ?
        IB04101                DB000010
AXIS    00001          DONE    ?
        00001                  DB000011
FEEDFWD ?              ERROR   ?
        MW01240                DB000012
SPDLOOP ?
        MW01242
POSLOOP ?
        MW01244
SPD-TI  ?
        MW01246
POS-TI  ?
        MW01248
DATA06W ?
        DA00070
```

<TUNING> Function Block Operation Notes

- This function can be executed while in motion by another function block. But it is possible that the motor occasionally does unexpected operation by changing the parameter. So it is recommended to execute this block while the motor is stopped.
- To use the function block, the *EXECUTE* bit must be high. If the *EXECUTE* bit goes low the outputs will be set to zero.
- If any of the input values are out of range, an error will occur.
- When the EXECUTE input becomes High, the speed feed forward, speed loop gain, position loop gain, speed loop integral time constant, and position loop integral time constant can be set in the motion set parameter and the servo amplifier user constant.
- RDA will be set after each value (FEEDFWD, SPDLOOP, POSLOOP) is set in the servo amplifier.

- Six words are used as working registers for this function, starting at the address in *Data06W*.

<TUNING>Input and Output Register Map

Output Registers

The following registers are used as outputs from the function block. They can be monitored by the LadderWorks program to check the execution of the function.

Output	Type	Description
RUNNING	Bit	This bit is ON when the function block is executing.
DONE	Bit	This bit is ON when the function block is complete and there are no errors
ERROR	Bit	This bit is ON when one of the values is out of range. However, if the error condition is cleared, this output is reset to OFF.

Input Registers

The following registers are used as inputs to the function block. They select the options and define the parameters that the user needs to make the function work as necessary.

INPUT	TYPE	Content	Range of setting and state
EXECUTE	Bit	Block enable – Block cannot execute unless this is TRUE.	Rising edge initiates block TRUE – Block enable FALSE – Block disable, set all outputs to zero and off
AXIS	Word	Axis number related to the block	1～16
FEEDFWD	Word	Speed Feed forward	0～100[%] Set parameter OWxx30
SPDLOOP	Word	Speed loop gain	1～2000[Hz] Set parameter OWxx2F
POSLOOP	Word	Position loop gain	10～20000[0.1/s] Set parameter OWxx2E
SPD-TI	Word	Speed loop integral time constant	15～32767[0.01ms] Set parameter OWxx34
POS-TI	Word	Position loop integral time constant	0～5000[ms] Set parameter OWxx32
DATA06W	Address	Address of the first working register.	Six words of register space are used by this function.

<TUNING> Block Fault Condition:

The following table outlines several situations that may cause an error, and will turn on the blocks “Error” output bit. The block “Error” output will cleared if the *EXECUTE* bit goes low.

Internal Fault Bit	Cause	Note
feedfwdOut AB000001	Feed forward value is not within acceptable range.	Goes high if the set value is in range. Sets bit3 of RDA Error ID(MW3**81) if error occurs.
spdLoopOut AB000002	Speed loop gain value is not within acceptable range.	Goes high if the set value is in range. Sets bit3 of RDA Error ID(MW3**81) if error occurs
posLoopOut AB000003	Position loop gain value is not within acceptable range.	Goes high if the set value is in range. Sets bit3 of RDA Error ID(MW3**81) if error occurs.
SPDTiout AB000019	Speed loop integration time value is not within acceptable range.	Goes high if the set value is in range. Sets bit3 of RDA Error ID(MW3**81) if error occurs
PosTiOut AB00001A	Position loop integration time value is not within acceptable range.	Goes high if the set value is in range. Sets bit3 of RDA Error ID(MW3**81) if error occurs
axisInErr AB00014A	The axis number entered on the input is not an acceptable value	The function blocks can only control 1 to 16 axis. Any value greater or smaller then this will cause an error. his does not set the RDA Error ID.
RDA_Error AB000006	The module or axis number in the RDA is not acceptable.	Goes high if the Motor Rated Speed (MW3**73) or Encoder resolution (ML3**74) is not appropriately set up in RDA. Sets bit A of RDA Error ID(MW3**81) if error occurs..

<TUNING> Working Registers

This table outlines the data in the six registers used by the function block. There is not usually any need for the user to access any of these bits directly.

Register No	TYPE	Name	Content
AW00000			
Bit 0	IN	execute	EXECUTE input (XB000000)
Bit 1	Working	feedfwdOut	Goes high for one scan if FEEDFWD input is out of range.
Bit 2	Working	spdLoopOut	Goes high for one scan if SPDLOOP input is out of range.
Bit 3	Working	posLoopOut	Goes high for one scan if POSLOOP input is out of range.
Bit 5	Working	axisInErr	Goes high for one scan if Axis input is out of range.
Bit 6	Working	RDA_Error	Goes High if RDA module or axis number is not in range.
Bit 7	OUT	running	Directly controls YB000000 (*RUNNING* Output)
Bit 9	OUT	error	Directly controls YB000002 (*ERROR* Output)
Bit A	Working	osExecute	Reserved
Bit B	Working	firstPass	One shot coil for initializing block on the rising edge of the EXECUTE input.
AW00001			
Bit 9	Working	SPDTiout	Goes high for one scan if SPD-TI input is out of range.
Bit A	Working	PosTiOut	Goes high for one scan if POS-TI input is out of range.
Bit C	Working	oneshot1c	Reserved
Bit D	Working	Done_latch	Directly controls YB000001 (*DONE* Output)
AW00003	Working	rdaMult	Value for address offset to locate proper RDA
AW00004	Working	axisMult	Value for address offset to locate proper Axis
AW00005	Working	revision	Revision level of block.

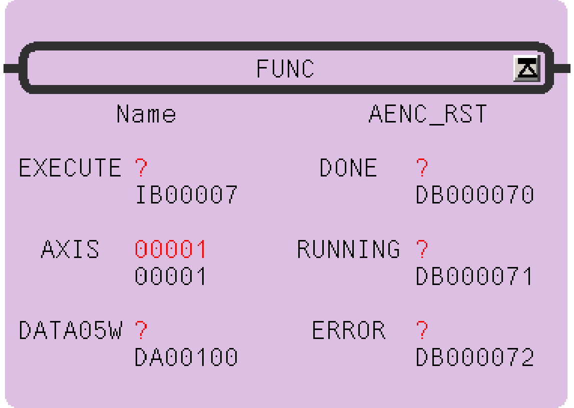
ABSOLUTE ENCODER RESET Function Block

Function block for MP2000 Series SVB Module

<AENC_RST> Function Block Summary

This function block commands a clear of an Absolute encoder's muti-turn counter and (if required) will clear a A.81 (Absolute backup loss) alarm.

Function Block Diagram



```
                  FUNC
        Name                AENC_RST

EXECUTE ?              DONE    ?
        IB00007                DB000070

  AXIS  00001          RUNNING ?
        00001                  DB000071

DATA05W ?              ERROR   ?
        DA00100                DB000072
```

<AENC_RST> Function Block Operation Notes

- This is valid for MP2x00 SVB module
- Rising edge of EXECUTE input initiates block operation, and all block input values are read.
- To use the function block, the *EXECUTE input* must be held ON at least until the *DONE* output bit turns ON or the *ERROR* output bit turns ON.
- The *RUNNING* output bit will be held high until: the entire operation is complete. This means the RUNNING output bit will turn OFF when *DONE* is ON, *EXECUTE* becomes low, or an *ERROR* occurs.
- *DONE* bit turns ON when the Absolute encoder position reset is complete. This happens when the MP2x00 controller confirms the ABS_RST (Motion Command Code #22) followed by an alarm reset.
  - For a SIGMA 2 Servopack (SGDH-□ □ □ E) with NS100 or NS115 option card, Servopack control power and MP2x00 power must be cycled.
  - For a Sigma 3 Mechatrolink II Servopack (SGDS-□ □ □ 1 A) no power cycle, on any device, is required.
- The function block operation can only happen with the Servo Off and after all motion has stopped. If another block has control of the axis or a move is not finished, and the *EXECUTE* was turned ON, the *ERROR* output will go ON. The when the *ERROR* output is on, the RDA will not indicate an error.

- If EXECUTE has been turned ON, But the Absolute encoder rest operation does not complete within 5 seconds, ERROR output bit will turn ON. For the time between the EXECUTE ON and the ERROR output bit turns ON, the RUNNING output bit will be ON
- Five words are used as working registers for this function, starting at the address in *Data05W*

<AENC_RST> Input and Output Register Map

Output Registers

The following registers are used as outputs from the function block. They can be monitored by the LadderWorks program to check the execution of the function.

Output	**Type**	**Description**
RUNNING	Bit	High while block has control and Absolute Encoder Reset Procedure is in process
DONE	Bit	Indicates the Absolute Encoder Reset Procedure has completed successfully
ERROR	Bit	Latches high if any error occurs in block.

Input Registers

The following registers are used as inputs to the function block. They select the options and

Input	**Type**	**Description**	**Range and state**
EXECUTE	Bit	Block enable – Block cannot execute unless this is TRUE.	Rising edge initiates TRUE – Block enable FALSE – Block disable, set all outputs to zero and off
AXIS	Word	Axis number related to the block.	1 to 16
DATA05W	Addres s	Address of the first working register.	Five (5) words of register space are used by this function.

<AENC_RST> Block Fault Conditions:

The following table outlines several situations that may cause an error, and will turn on the blocks "Error" output bit. The block "Error" output will cleared if the *EXECUTE* bit goes low, but the Error ID (MW***81) will remain in the RDA. To reset the Error ID, use the Alarm Reset Function Block.
Note that each axis has its own Error ID stored in its RDA axis section, offset by 300 for each axis. Example: Axis#1 stores to MW30181, Axis #2 stores to MW30481, etc.

Internal Fault Bit	Cause	Note
TimeOut_Error AB000010	Absolute encoder rest operation does not complete within 5 seconds	It is possible there is no absolute encoder. It is a Sigma I (SGD-xxxN or SGDB-xxxN) Mechatrolink I system, and the operation can not be done. The operation already occurred and can not happen a second time, with out a power cycle (Sigma II) or some motion (SigmaIII)
RDA_Range_Error AB000012	Axis number is out of range.	
CmdCode_Error AB000014	Motion Command Code (MCC) was other than Zero (NOP) when EXECUTE transitioned from OFF -> ON.	This is determined by monitoring the Motion Command Code response register (IWxx08) for Zero.
Svon_Error AB000015	If was ON when EXECUTE transitioned from OFF -> ON.	This side effect of this operation is loss of Mechatrolink II synchronization. Synchronization is restablished by the function block after it commands an Alarm Reset. If Servo was ON, it would be turned OFF as due to the loss of synchronization. It is better that the Servo is off before the operation begins.

<AENC_RST> Working Registers

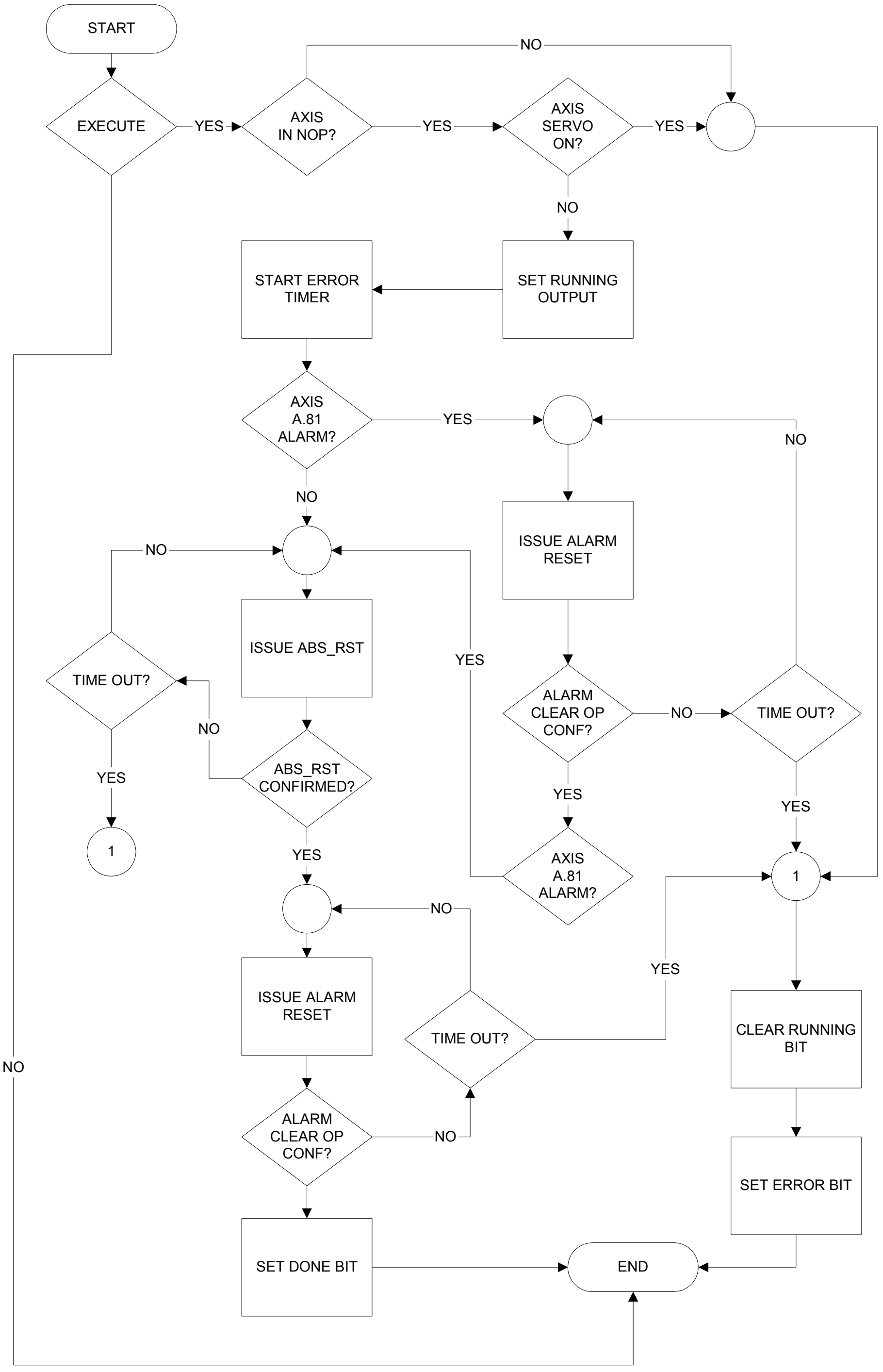
This table outlines the data in the five registers used by the function block. There is not usually any need for the user to access any of these bits directly.

Register No.	Type	Name	Description
AW00000		WORKING_REG0	
Bit 0	IN	EXECUTE_TRUE	*EXECUTE* input (XB000000)
Bit 1	IN	EXECUTE_1stPASS	One shot on rising edge of Execute input to set axis data.
Bit 2	Working	Operate	Start operation
Bit 3	Working	EXECUTE_TRANS_OFF	One shot on falling edge of Execute input to set axis data.
Bit 4	Working	InRDA_Range	Goes high for one scan if "AXIS" input is in range.
Bit 5	Working	MCC_RESP_ABS_RST	Goes High when Motion Command Code (MCC) Response Register (IWxx08) is equal to 22, "ABS_RST" operation has happened
Bit 6	Working	Send_ABS_RST_MCC	Goes High when it is time to send MCC 22 "ABS_RST"
Bit 7	Working	MCC_RESPONCE_NOP	Goes High when Motion Command Code (MCC) Response Register (IWxx08) is equal to 0, "NOP" (No Operation) is true

Bit 8	Working	START_STEP2	Goes High at the start of Step 2. Step1 and Step2 operation are different if A81 alarm exist or not.
Bit 9	Working	OneShotAB09	Reserved
Bit A	Working	Running	Directly controls YB000000 (*RUNNING* Output)
Bit B	Working	Step2_Complete	Goes High at the end of Step 2. Step1 and Step2 operation are different if A81 alarm exist or not.
Bit C	Working	OneShotAB0C	Reserved
Bit D	Working	OneShotAB0D	Reserved
Bit E	Working	NWORK_SVPACK_ALM	Goes ON if any Network Servopack alarms exist.
Bit F	Working	ABS_ENCODER_A81	Goes ON if Network Servo alarm is an A.81 – Absolute Encoder Backup Alarm.
AW00001		WORKING_REG1	
Bit 0	Working	TimeOut_Error	Goes high for second scan of Block only.
Bit 1	Working	OneShotAB11	Reserved
Bit 2	Working	RDA_Range_Error	Goes ON If RDA is out of range
Bit 3	Working	ALARM_CLR_STEP3	Issue of step 3: and Alarm clear
Bit 4	Working	CmdCode_Error	On first pass, MCC was other than Zero
Bit 5	Working	Svon_Error	On first pass, Servo was ON (enabled)
Bit 6	-	Unused	
Bit 7	_	Unused	
Bit 8	OUT	Op_Error	Directly controls YB000002 (*ERROR* Output)
Bit 9		OneShotAB19	Reserved
Bit A	OUT	Op_Done	Directly controls YB000001 (*DONE* Output)
Bit B		OneShotAB1B	Reserved
Bit C	Working	Op_DoneOneShot	Operation Done Flag
Bit D	_	Unused	
Bit E	Working	Step1	Step one has started and in operation
Bit F	Working	Step1_Complete	Step one as completed
AW00002	Working	OFFSET_STORAGE	Value for address offset to locate proper RDA
AW00003	Working	DRIVE_STAUS_LOCAL	State of Network Servo Axis (bit by bit) from RDA[1]
Bit 1	Working	Svon_FB	Axis Servo ON (enabled)
AW00004	Working	Revision	Revision Level of the function block.

[1] This register (AW00003)holds the entire bit by bit value of motion Monitor Parameter IWxx01. Only one bit is used in this function block: Servo ON (Enabled), bit 1.

<AENC_RST> Flow Chart


START
EXECUTE
YES
AXIS IN NOP?
YES
AXIS SERVO ON?
YES
NO
NO
SET RUNNING OUTPUT
START ERROR TIMER
AXIS A.81 ALARM?
YES
NO
NO
ISSUE ALARM RESET
NO
ISSUE ABS_RST
YES
TIME OUT?
NO
ABS_RST CONFIRMED?
YES
1
ALARM CLEAR OP CONF?
NO
TIME OUT?
YES
YES
AXIS A.81 ALARM?
1
YES
NO
ISSUE ALARM RESET
TIME OUT?
YES
CLEAR RUNNING BIT
NO
ALARM CLEAR OP CONF?
NO
SET ERROR BIT
SET DONE BIT
END

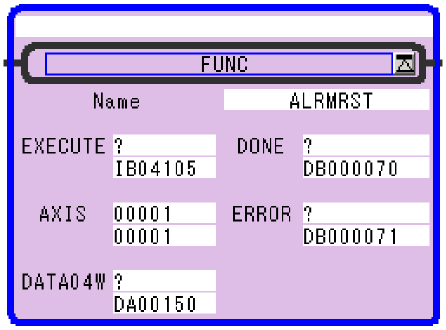
ALARM RESET Function Block

Function block for MP2000 series

<ALRMRST> Function Block Summary

The “ALARMRST” block is used to reset any servo alarm as well as RDA errors in MW3**81 and MW3**82. Not all errors can be cleared from the reset clock. Some errors will require cycling the power.

Function block diagram



FUNC

Name			ALRMRST
EXECUTE	? IB04105	DONE	? DB000070
AXIS	00001 00001	ERROR	? DB000071
DATA04W	? DA00150		

<ALRMRST> Function block Operation Notes

- To use the function block, the *EXECUTE* input must see a rising edge. The reset bit (ACR) will go on as long as the EXECUTE input is on.
- Related Function Blocks: Errors that can be reset with this block can be detected by Read Error Function Block (RDERROR).
- Alarm Indication and Reset Registers Used:
    - ACR (OBxx00F): Alarm Clear, used to clear alarms in system
    - Warning (ILxx02): Warning information on the axis displayed in the bit.
    - Alarm (ILxx04): Alarm information on the axis displayed in the bit.
    - Servo driver alarm (IBxx2C0): Servo driver alarm
    - Servo alarm code (IWxx2D): Refer to the manual of the servo amplifier for details of the servo amplifier alarm code.
    - CPU error (SW00041): CPU Error Status System Register. This is not used for the detection purpose, but can be used to help the user determine more information about the alarm. This register indicates bitwise: Serious failure, Program memory error, User operation error, I/O error, and Transmission errors. Refer to Chapter 10 of the user's manual (Trouble shoot) for details.
- The servo alarm: This block will set ACR to initiate alarm clear. The ACR bit is reset request of the alarm to the controller. If the alarm is not cleared within 500msec, the block will error out and the ERROR output will turn on. If the alarm is cleared, the DONE output will be latched on.
- RDA alarms: : This block will set MW3xx81(ERRID1) and MW3xx82(ERRID2) both to zero.

- This block will also set SW00041=0 (CPU error status system register). Note that this register is not used to trigger any alarms. It is only reset.
- Four words are used as working registers for this function, starting at the address in *Data04W*.

<ALRMRST> Input and Output Register Map

Output Registers

The following registers are used from the function block as an output. To check the execution of the function, they can be monitored by the MPE720 program.

Output	Type	Description
DONE	Bit	This bit is ON when the function block is Complete in resetting axis.
ERROR	Bit	Latches high if any block errors occur

Input Registers

The registers are used as inputs to the function block.

Input	Type	Description	Range and state
EXECUTE ○	Bit	Block executes reset when rising edge occurs. As long as execute input is on, then OB**00F is on.	RISING EDGE – Block executes TRUE – OB**00F is on FALLING EDGE – Reset input (OB**00F) gets set back to zero.
AXIS	Word	Axis number related to the block	1～31
DATA04W	Addres s	Address of the first working register.	Four words of register space are used by this function.

<ALRMRST> Block Error Condition:

The following table outlines several situations that may cause an error, and will turn on the blocks "Error" output bit.

Internal Fault Bit	Cause	Note
axisout AB00000C	Axis specified by user is out of range	Valid range is 1 to 16. Will not write to RDA
Time-out AB00000D	Alarm condition did not clear within default timeout (500msec).	Alarm condition may still exist. Will not write to RDA. May need to cycle power on controller and servopacks to clear the error, set-up parameter may be out of range, or hardware failure may exist.

<ALRMRST> Working Registers

This table outlines the data in the four registers used by the Alarm Reset function block. There is not usually any need for the user to access any of these bits directly.

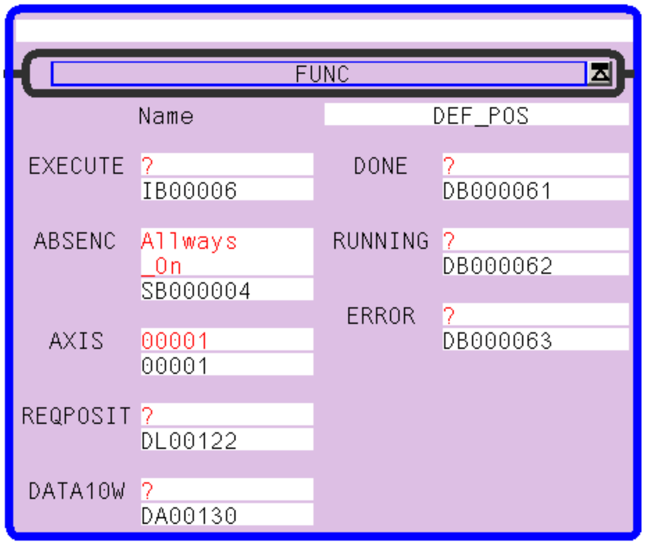
Register No	TYPE	Name	Description
AW00000			
Bit 0	IN	EXECUTE	EXECUTE input (XB000000)
Bit 8	OUT	DONE	DONE output (YB000000)
Bit 9	OUT	ERROR	ERROR output (YB000001)
Bit A	Working	oneshotA	Rising edge
Bit B	Working	firstPass	Initialize block
Bit C	Working	axisout	Indicates axis input is out of range
Bit D	Working	Time-out	Indicates that the alarm did not reset within a specified (500msec default) timeout period.
AW00001	Working	Counter	Timer for pausing post reset
AW00002	Working	rdaMult	Value for address offset to locate proper RDA
AW00003	Working	Revision	Revision level for block

DEFINE POSITION Function Block
Function Block for MP2000 Series

<DEF_POS> Function Block Summary

This function block adjusts the absolute position offset of the defined axis. It will also handle the position of a rotary load.

Function Block Diagram



FUNC

Name DEF_POS

EXECUTE ? IB00006
ABSENC Allways_On SB000004
AXIS 00001 00001
REQPOSIT ? DL00122
DATA10W ? DA00130

DONE ? DB000061
RUNNING ? DB000062
ERROR ? DB000063

<DEF_POS> Function Block Operation Notes

- This is valid for MP2x00 family of products.
- Rising edge of EXECUTE input initiates block operation, and all block input values are read.
- If the ABSENC input is FALSE The Offset register ill be cleared every power up, even if EXECUTE input is FALSE.
- To use the function block, the *EXECUTE input* must be held ON at least until the *DONE* output bit turns ON or the *ERROR* output bit turns ON.
- The *RUNNING* output bit will be held high until: the entire operation is complete. This means the RUNNING output bit will turn OFF when *DONE* is ON, *EXECUTE* becomes low, or an *ERROR* occurs.
- *DONE* bit turns ON when the new position is set.
- If another block has control of the axis or a move is not finished, and the *EXECUTE* was turned ON, the *ERROR* output will go ON.   The when the *ERROR* output is on, the RDA will not indicate an error.
- If EXECUTE has been turned ON, The Axis is defined in the RDA as rotary, and the REQPOSIT is negative or larger than the RDA defined PERIOD, ERROR output bit will turn ON.
- If an Absolute encoder is present then ABSENC input must be ON.

- Ten words are used as working registers for this function, starting at the address in *Data10W*

<DEF_POS> Input and Output Register Map

Output Registers

The following registers are used as outputs from the function block. They can be monitored by the LadderWorks program to check the execution of the function.

Output	**Type**	**Description**
DONE	Bit	Indicates the Define Position Procedure has completed successfully
RUNNING	Bit	Indicates the Define Position Procedure is in process.
ERROR	Bit	Latches high if any error occurs in block.

Input Registers

The following registers are used as inputs to the function block. They select the options and set necessary parameters internally.

Input	Type	Description	Range and State
EXECUTE	Bit	Block enable – Block cannot execute unless this is TRUE.	Rising edge initiates TRUE – Block enable FALSE – Block disable, set all outputs to zero and off
ABENC	Bit	For power up operational purposes the existence of Absolute Encoder on this axis needs to be known.	TRUE – An absolute encoder is mounted on the servomotor for this axis. FALSE – An incremental encoder is mounted on the servomotor for this axis.
AXIS	Word	Axis number related to the block.	1 to 16
REQPOSIT	Long	The Desired Position the Axis is to be set at.	For linear axis, the value can be $–2^{31}$ to $(2^{31}–1)$ For a Rotary Axis the value must be 0 to the PERIOD (ML***06) defined in the RDA
DATA10W	Address	Address of the first working register.	Ten (10) words of register space are used by this function.

<DEF_POS> Block Fault Conditions:

The following table outlines several situations that may cause an error, and will turn on the blocks "Error" output bit. The block "Error" output will cleared if the *EXECUTE* bit goes low, but the Error ID (MW***81) will remain in the RDA. To reset the Error ID, use the Alarm Reset Function Block.

Note that each axis has its own Error ID stored in its RDA axis section, offset by 300 for each axis. Example: Axis#1 stores to MW30181, Axis #2 stores to MW30481, etc.

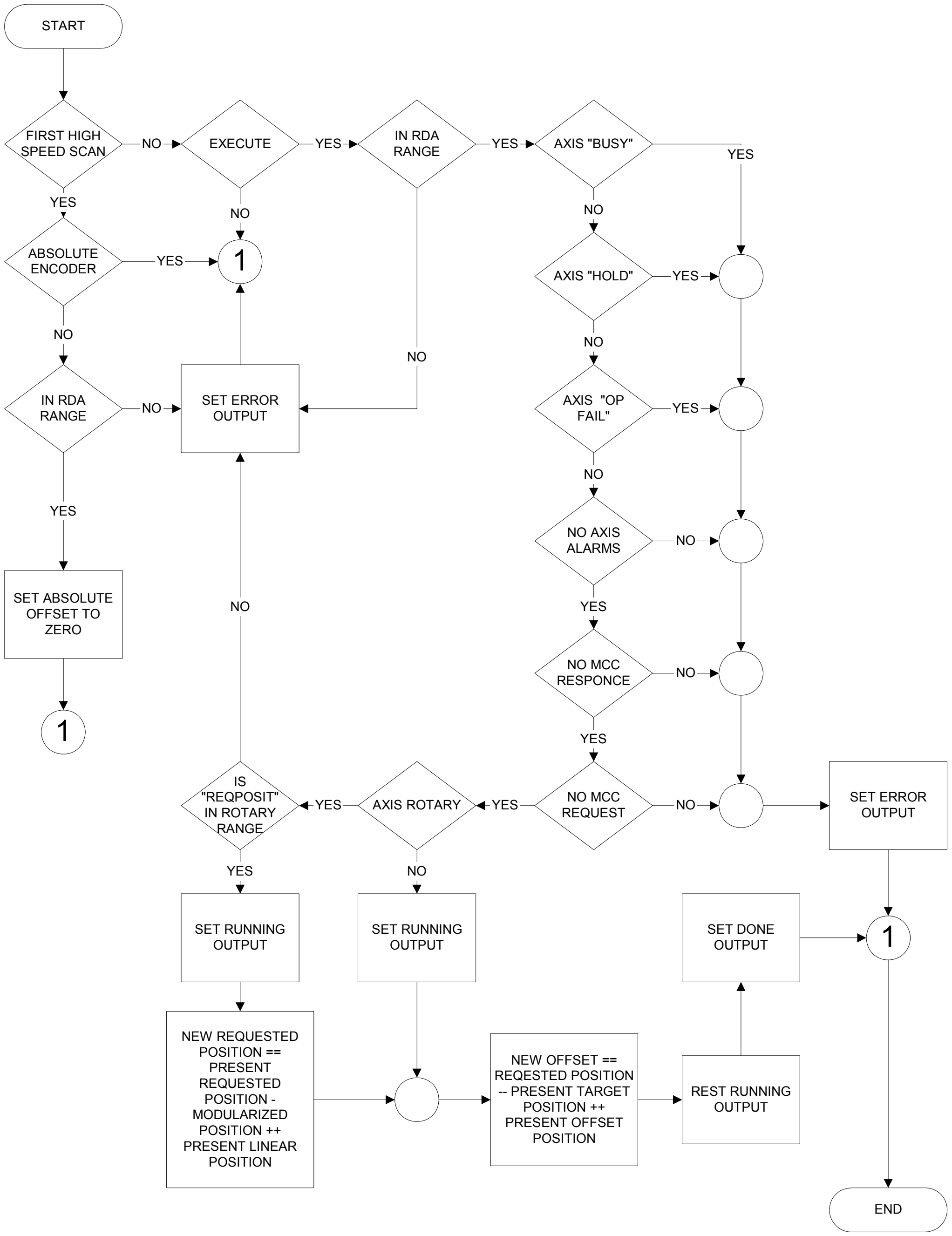
Internal Fault Bit	**Cause**	**Note**
Rotary_ReqPosOK AB000001	IF this bit is OFF, when EXECUTE transitioned from OFF -> ON., an error will occur. REQPOSIT is outside range.	See note above for the "REQPOSIT input.)
IN_RDA_RANGE AB000007	IF this bit is OFF, when EXECUTE transitioned from OFF -> ON., an error will occur. Axis number is outside range.	See above note for the "AXIS" input.
No_Alarms AB000008	IF this bit is OFF, when EXECUTE transitioned from OFF -> ON, an error will occur. The axis was in an error or had an Alarm.	This bit Monitors RDA MLxxx12 (ILxx04)
No_MCC_Requeste d AB000009	IF this bit is OFF, when EXECUTE transitioned from OFF -> ON, an error will occur. Motion Command Code (MCC) in execution was other than Zero (NOP).	This is determined by monitoring the Motion Command Code response register (IWxx08) for Zero.
No_MCC_InQue AB00000A	If was OFF when EXECUTE transitioned from OFF -> ON,an error will occur. Motion Command Code (MCC) requested was other than Zero (NOP).	This is determined by monitoring the Motion Command Code request register (OWxx08) for Zero.
No_OpInQue AB00000B	If was OFF when EXECUTE transitioned from OFF -> ON,an error will occur. The Axis is BUSY, in HOLD, or aprevious motion command FAILED.	This is determined by monitoring specific BITS in the Motion Command status register (IWxx09). The Bits are :0, 1, and 3 BUSY, HOLD, and FAILED respectively.

<DEF_POS> Working Registers

This table outlines the data in the ten registers used by the function block. There is not usually any need for the user to access any of these bits directly.

Register No.	Type	Name	Description
AW00000		Working_Register0	
Bit 0	IN	Execute	*EXECUTE* input (XB000000)
Bit 1	Working	Rotary_ReqPosOK	
Bit 2	Working	FirstFB_Scan	One shot on rising edge of Execute input to set axis data.
Bit 3	Working	LastFBScan	One shot on falling edge of Execute input to set axis data.
Bit 4	Working		Reserved
Bit 5	Working		Reserved
Bit 6	IN	AbsEnc_Present	*ABSENC* input (XB000001)
Bit 7	Working	IN_RDA_RANGE	Goes High when Motion Command Code (MCC) Response Register (IWxx08) is equal to 0, “NOP” (No Operation) is true
Bit 8	Working	No_Alarms	Goes High when there are no Alarms
Bit 9	Working	No_MCC_Requested	Goes High when Motion Command Code (MCC) Response Register (IWxx08) is equal to 0, “NOP” (No Operation) is true
Bit A	Working	No_MCC_InQue	Goes High when Motion Command Code (MCC) Request Register (OWxx08) is equal to 0, “NOP” (No Operation) is true
Bit B	Working	No_OpInQue	Goes High when the axis is NOT BUSY, NOT IN HOLD And NOT FAIED
Bit C	Working	Allow_DefPosExe	This bit is true if the conditions are correct to allow a define position operation.
Bit D	Working	Exec_DefPos	Goes High when the actual Define position operation happens.
Bit E	Working	AxisIsRotary	This is TRUE if the Axis is defined as Rotary in the RDA
Bit F	Working	Set_OffsetToZero	This bit is only true on power up If the “ABSENC” input is FALSE.
AW00001	Working	Working_Register1	
Bit 0	OUT	Op_Done	DONE Output (YB000000)
Bit 1	OUT	OpError	ERROR Output (YB000002)
Bit 2	Working	RotaryReqPos_InRange	Result of Range Check on “REPOSIT” Input value. Only used if axis is rotary.
Bit 3	OUT	OpRunning	RUNNING Output (YB000001)
AW00002	Working	rdaMult	Value for address offset to locate proper RDA
AW00003	Working	Working_Register3	Unused (For expansion)
AL00004	IN	Requested_Position	The “REQPOSIT” input that the axis position will be set to.
AW00006	Working	Work_Register6	Unused (For expansion)
AW00007	Working	Working_Regester7	Unused (For expansion)
AW00008	Working	Working_Regester8	Unused (For expansion)
AW00009	Working	Revision	Revision Level of the function block.

<DEF_POS> Flow Chart


START
FIRST HIGH SPEED SCAN
NO
EXECUTE
YES
IN RDA RANGE
YES
AXIS "BUSY"
YES
YES
NO
NO
ABSOLUTE ENCODER
YES
1
AXIS "HOLD"
YES
NO
NO
NO
IN RDA RANGE
NO
SET ERROR OUTPUT
AXIS "OP FAIL"
YES
NO
YES
NO AXIS ALARMS
NO
SET ABSOLUTE OFFSET TO ZERO
NO
YES
NO MCC RESPONCE
NO
1
YES
IS "REQPOSIT" IN ROTARY RANGE
YES
AXIS ROTARY
YES
NO MCC REQUEST
NO
SET ERROR OUTPUT
YES
NO
SET RUNNING OUTPUT
SET RUNNING OUTPUT
SET DONE OUTPUT
1
NEW REQUESTED POSITION == PRESENT REQUESTED POSITION - MODULARIZED POSITION ++ PRESENT LINEAR POSITION
NEW OFFSET == REQESTED POSITION -- PRESENT TARGET POSITION ++ PRESENT OFFSET POSITION
REST RUNNING OUTPUT
END

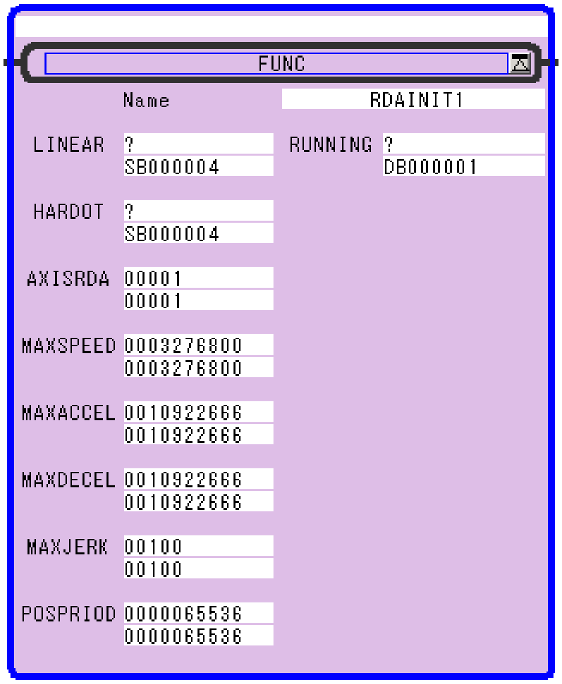
RDA INITIALIZATION #1 Function

Function block for MP2000 series

<RDAINIT1> Function Block Summary

The Reserved Data Area (RDA) Initialization function block (RDAINIT1) block is used to initialize the values in the RDA for a specific axis. Without the proper values in the RDA the some motion blocks may not work properly. The RDAINIT2 block must also be used to complete the initialization of all mandatory RDA values.

Function Block Diagram



Name RDAINIT1

LINEAR	? SB000004	RUNNING	? DB000001
HARDOT	? SB000004		
AXISRDA	00001 00001		
MAXSPEED	0003276800 0003276800		
MAXACCEL	0010922666 0010922666		
MAXDECEL	0010922666 0010922666		
MAXJERK	00100 00100		
POSPRIOD	0000065536 0000065536		

<RDAINIT1> Function Block Operation Notes

- To use the function block, it must be placed in an ‘A’ drawing.
- Both RDA init 1 and 2 blocks should be used together for each axis. RDA Init 1 block sets Linear or Rotary positioning mode, Maximum Speed/Accel/Decel/Jerk values, and the rotary position period.
- This drawing uses no drawing registers to run.

<RDAINIT1> Input and Output Register Map

Output Registers

The following registers are used as outputs from the function block. They can be monitored by the LadderWorks program to check the execution of the function.

Output	Type	Description
RUNNING	Bit	Goes high while block is updating RDA. Updating completes in one scan.

Input Registers

The following registers are used as inputs to the function block. They select the options and define the parameters that the user needs to make the function work as necessary.

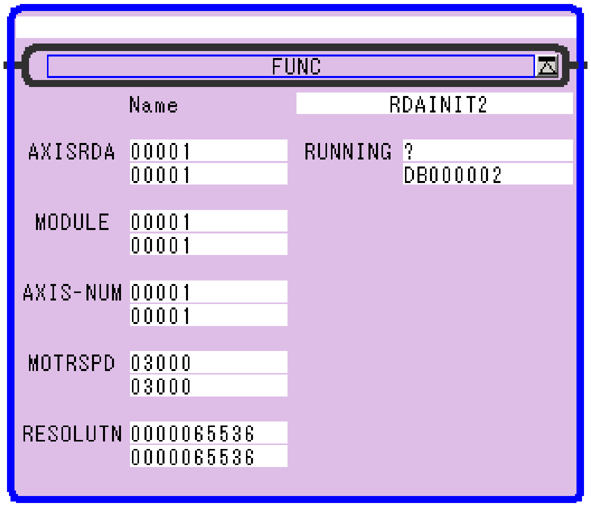
Input	Type	Description	Range and state
LINEAR	Bit	Defines if axis is in linear mode or rotary mode.	TRUE – axis is linear FALSE – axis is rotary
HARDOT	Bit	This determines if hard over travels should be alarmed and indicated in the RDA. This will result in SVON block writing to ErrorID1 (MW3**81)	TRUE – write OT alarm status to ErrorID1. FALSE – don't write OT alarm status to ErrorID1.
AXISRDA	Word	Axis number related to the block.	1～16
MAXSPEED	Long	Maximum speed the axis is allowed to travel (counts per second)	1 to 2147483647. The physical limitations of the motor and encoder resolution will make this number smaller.
MAXACCEL	Long	Maximum acceleration the axis is allowed to travel (in counts per second$^{2)}$	1 to 2147483647. The physical limitations of the motor and encoder resolution will make this number smaller.
MAXDECEL	Long	Maximum deceleration the axis is allowed to travel (in counts per second$^{2)}$	1 to 2147483647. The physical limitations of the motor and encoder resolution will make this number smaller.
MAXJERK	Long	More accurately defined as the S-Curve time in milliseconds. The greater the number the slower to ramp to the desired acceleration or deceleration.	1 to 2147483647. The number may go down based on scan rate. The smoothest S-Curve allowed is 255 scans if the scan time is 2 millisec. Then the max time will be MAXJERK = 510 msec
POSPRIOD	Long	For Rotary mode only. This is the largest value the axis position can reach before rolling over to zero.	1 to 2147483647.

RDA INITIALIZATION #2 function
Function block for MP2000 series

<RDAINIT2> Function Block Summary

The Reserved Data Area (RDA) Initialization function block (RDAINIT2) block is used to initialize the values in the RDA for a specific axis. Without the proper values in the RDA the some motion blocks may not work properly. The RDAINIT1 block must also be used to complete the initialization of all mandatory RDA values.

Function Block Diagram



FUNC

Name		RDAINIT2	
AXISRDA	00001 00001	RUNNING	? DB000002
MODULE	00001 00001		
AXIS-NUM	00001 00001		
MOTRSPD	03000 03000		
RESOLUTN	0000065536 0000065536		

<RDAINIT2> Function Block Operation Notes

- To use the function block, it must be placed in an ‘A’ drawing.
- Both RDA init 1 and 2 blocks should be used together for each axis. RDAINIT2 block sets the hardware module number, the axis number within the module, rated speed, and post-quad axis encoder resolution.
- This function uses no drawing registers to run.

<RDAINIT2> Input and Output Register Map

Output Registers

The following registers are used as outputs from the function block. They can be monitored by the LadderWorks program to check the execution of the function.

Output	TYPE	Content
RUNNING	Bit	Goes high while block is updating RDA, Updating completes in one scan.

Input Registers

The following registers are used as inputs to the function block. They select the options and define the parameters that the user needs to make the function work as necessary.

Input	Type	Description	Range and state
AXISRDA	Word	Logical axis number related to the block	1 to 16
MODULE	Word	Module address given to SVA-01 or SVB. This is same as ‘Circuit Number’ defined in module configuration. This links to the AXISRDA which is used in all motion blocks.	1 to 16
AXIS-NUM	Word	Axis number axis is plugged into on SVA-01 or SVB module. This is same as “Axis Number” within the defined “Circuit Number’. This links to the AXISRDA which is used in all motion blocks.	1to 2 (SVA-01 module) 1 to 16 (SVB-01 module)
MTRSPD	Word	Rated speed of motor on axis on RPM. This is necessary for calculating accels and decals.	1 to 32767
RESOLUTN	Long	Resolution of motors encoder (post-quad). This is necessary to convert units for counts and revs.	1 to 2147483647.

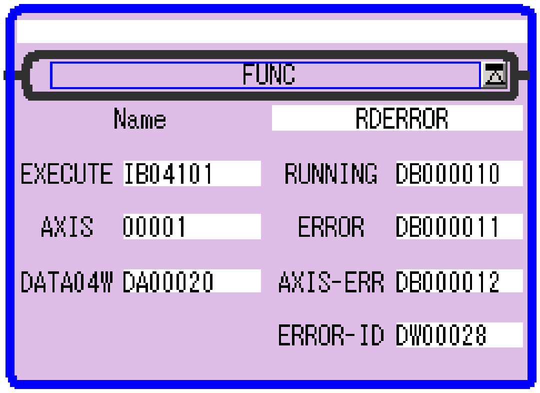
READ SERVO ERROR function

Function block for MP2000 series

<RDERROR> Function Block Summary

This function block displays servo alarm BIT and the servo alarm code in this function block regardless to other function blocks. BALRMRST function is used to reset the alarm. Some alarm need cycle power to clear it. Please refer to Chapter 10 of MP2200/MP2300 motion module user's manual SIJPC88070016 for the content of the alarm.

Function block diagram



Name RDERROR

EXECUTE IB04101 RUNNING DB000010

AXIS 00001 ERROR DB000011

DATA04W DA00020 AXIS-ERR DB000012

ERROR-ID DW00028

<RDERROR> Function Block Operation Notes

- Related function blocks are: SVON and ALRMRST.
- Rising edge of EXECUTE input initiates block operation, and all block input values are read once.
- The block will continue to monitor the axis for errors as long as the EXECUTE input stays True. If the *EXECUTE* bit goes off during operation, all outputs will be set to zero.
- The AXIS-ERR output will go on if servo alarm (IB**2C0=ON) occurs.
- The *ERROR-ID* output data can be referenced by the servo alarm code, IW**2D. Further information regarding alarm can be viewed by examining bits in IL**04. Please see appendix for details
- This block will not report any alarms until 5 seconds after the first execution of the low scan. This gives the system time to stabilize.
- Four words are used as working registers for this function, starting at the address in *Data04W*

<RDERROR> Input and Output Register Map

Output Registers

The following registers are used as outputs from the function block. They can be monitored by the LadderWorks program to check the execution of the function.

Output	Type	Description
RUNNING	Bit	This bit is ON when the function block is executing.
ERROR	Bit	If axis number is incorrect the ERROR output will go high. Also RDA Error ID (MW3**81) bit 3 is turned on.
AXIS-ERR	Bit	If the servo alarm (IB**2C0=ON) error occurs while the block is in execution, this output is on. However, if the error condition is released, this output is also reset.
ERROR-ID	Word	If an error occurs on the axis, this output will read the exact integer value in IW**2D error register. See appendix for alarm code meaning.

Input Registers

The following registers are used as inputs to the function block. They select the options and define the parameters that the user needs to make the function work as necessary.

Input	Type	Description	Range and state
EXECUTE	Bit	Block enable – Block cannot execute unless this is TRUE.	TRUE – Block enable FALSE – Block disable
AXIS	Word	Axis number related to the block.	1to16
DATA04W	Address	Address of the first working register.	Four words of register space are used by this function.

<RDERROR> Axis Fault Condition:

The following table outlines several situations that may cause an error. The block error can be cleared by the *EXECUTE* bit going low.

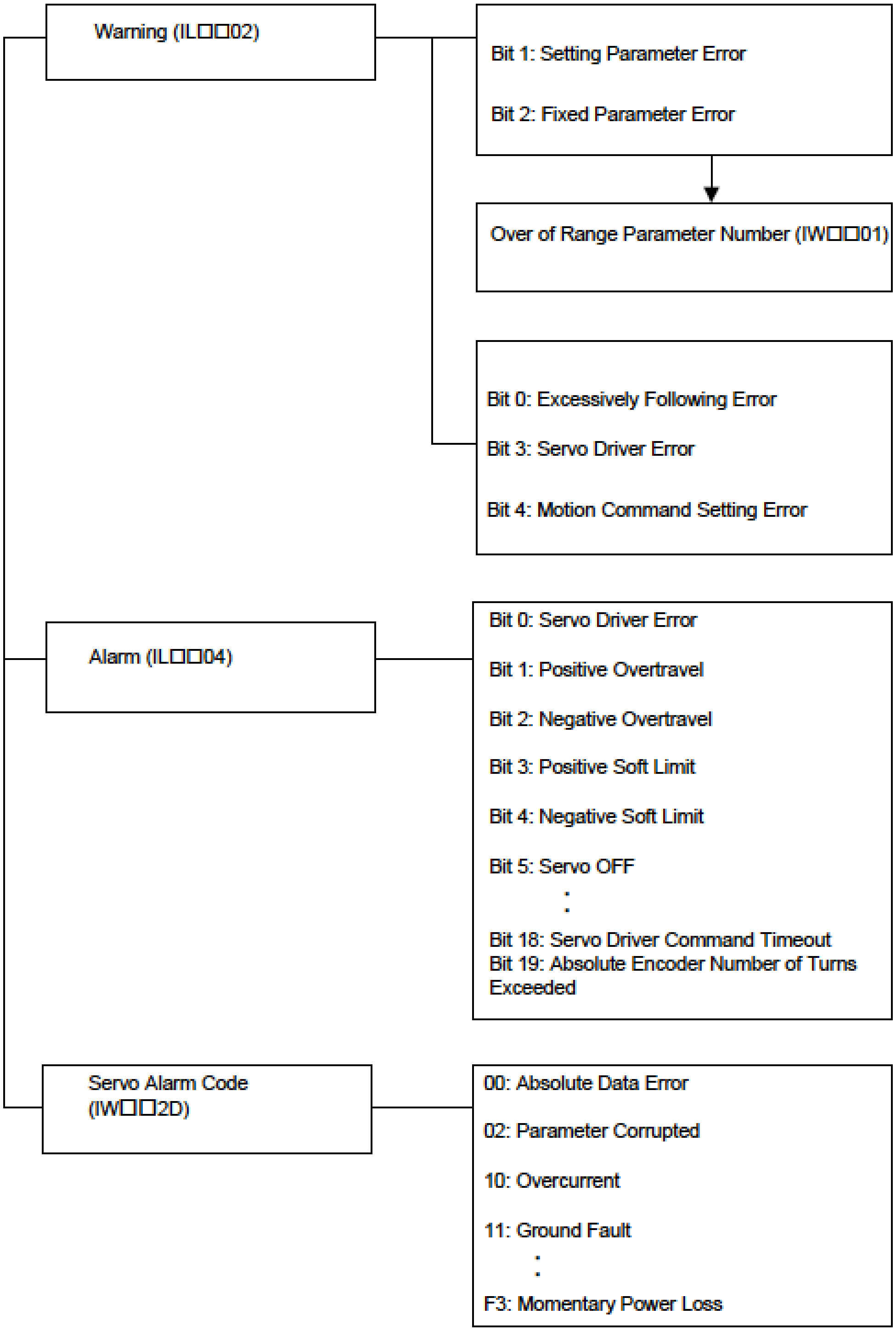
Internal error bit	Cause	Attention
IDerror AB000003	RDA ERROR	Will go high if Error ID <> 0 (MW3**81)
fault AB00012	Servo Alarm	If Servo alarm condition exists. Sets RDA Error ID (MW3**81) bit 7 on if error state exists.
axisInErr AB00000F	The axis number entered on the input is not an acceptable value	The functions blocks can only control 1 to 16 axis. Any value greater or smaller then this will cause an error. This does not set the RDA Error ID.

<RDERROR> Working Registers

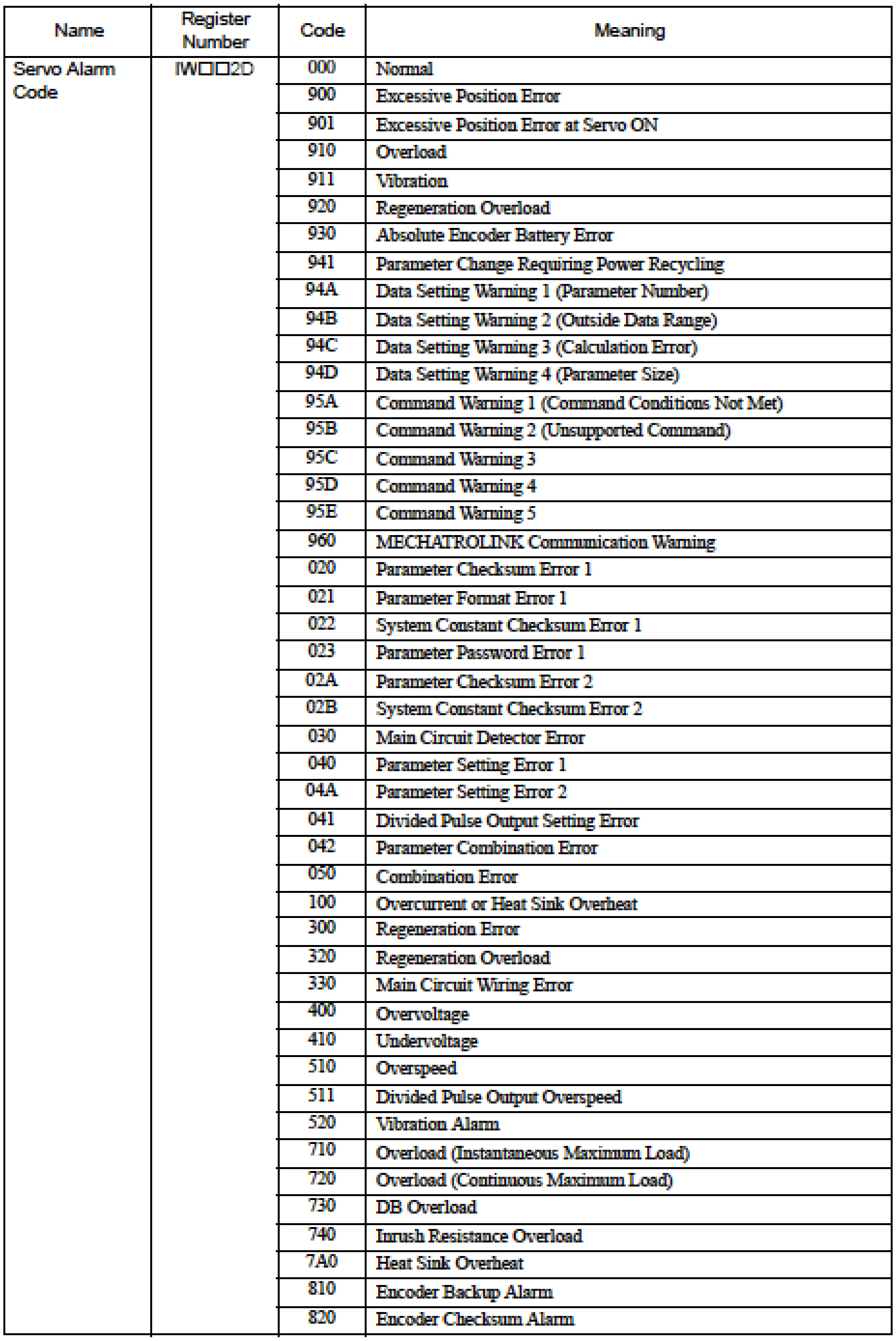
This table outlines the data in the five registers used by the function block. There is not usually any need for the user to access any of these bits directly.

Register No.	Type	Name	Description
AW00000			
Bit 0	IN	execute	*EXECUTE* input (XB000000)
Bit 1	Working	inrngAxis	Goes high for one scan if Axis input is in range.
Bit 3	Working	IDerror	Goes high if RDA Error ID (MW3**81) <> 0.
Bit 7	OUT	running	Directly controls YB000000 (*RUNNING* Output)
Bit A	Working	oneshotA	Reserved
Bit B	Working	firstPass	One shot coil to initialize RDA of blocks execution
Bit F	Working	axisInErr	Goes high for one scan if Axis input is out of range.
AW00001			
Bit 2	OUT	fault	Directly controls YB000002(AXIS-ERR Output).
AW00002	Working	rdaMult	Value for address offset to locate proper RDA
AW00003	Working	Revision	Revision Level of the function block.

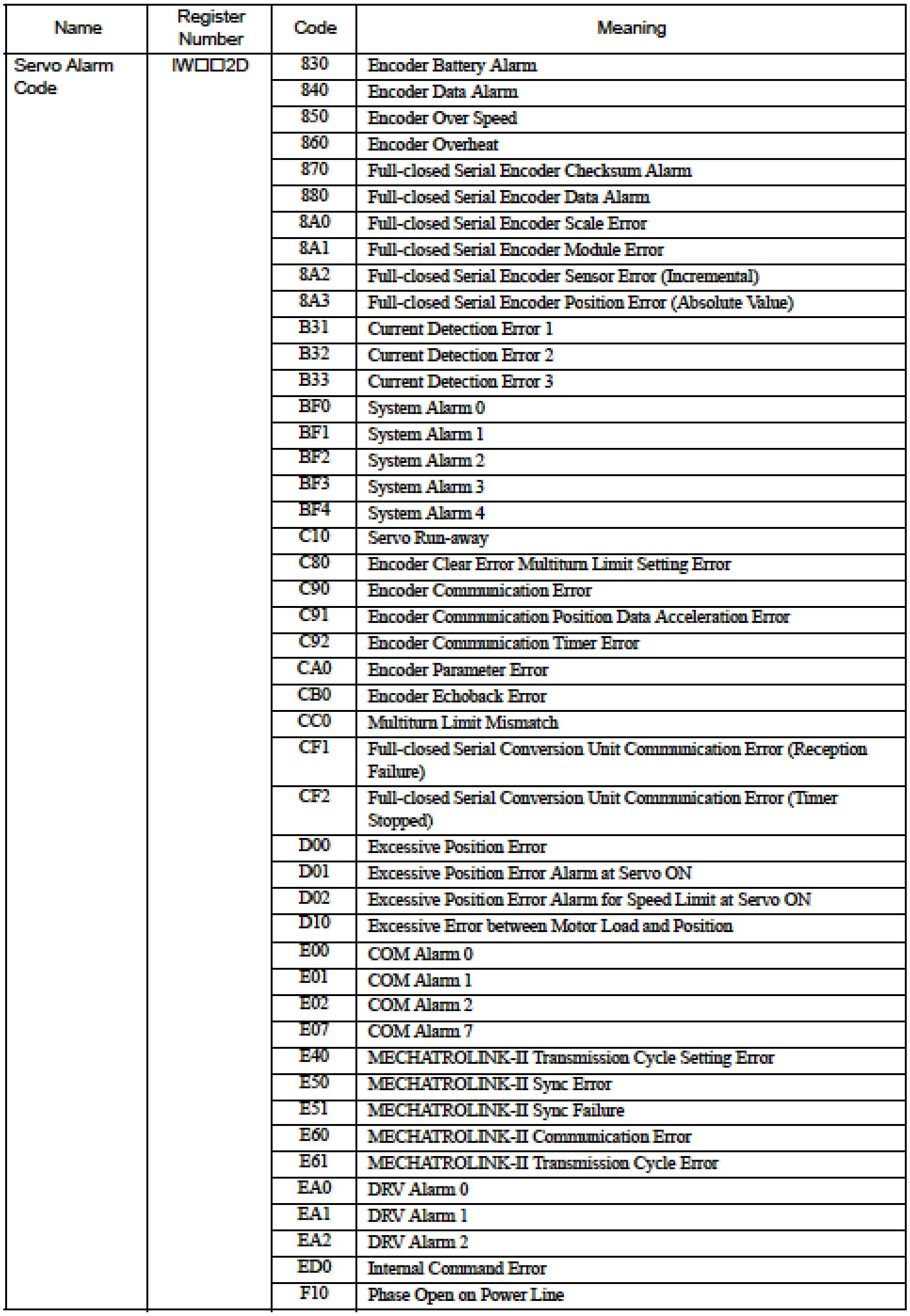
<RDERROR> Appendix:


Warning (IL□□02)
Bit 1: Setting Parameter Error
Bit 2: Fixed Parameter Error
Over of Range Parameter Number (IW□□01)
Bit 0: Excessively Following Error
Bit 3: Servo Driver Error
Bit 4: Motion Command Setting Error
Alarm (IL□□04)
Bit 0: Servo Driver Error
Bit 1: Positive Overtravel
Bit 2: Negative Overtravel
Bit 3: Positive Soft Limit
Bit 4: Negative Soft Limit
Bit 5: Servo OFF
:
Bit 18: Servo Driver Command Timeout
Bit 19: Absolute Encoder Number of Turns Exceeded
Servo Alarm Code (IW□□2D)
00: Absolute Data Error
02: Parameter Corrupted
10: Overcurrent
11: Ground Fault
:
F3: Momentary Power Loss

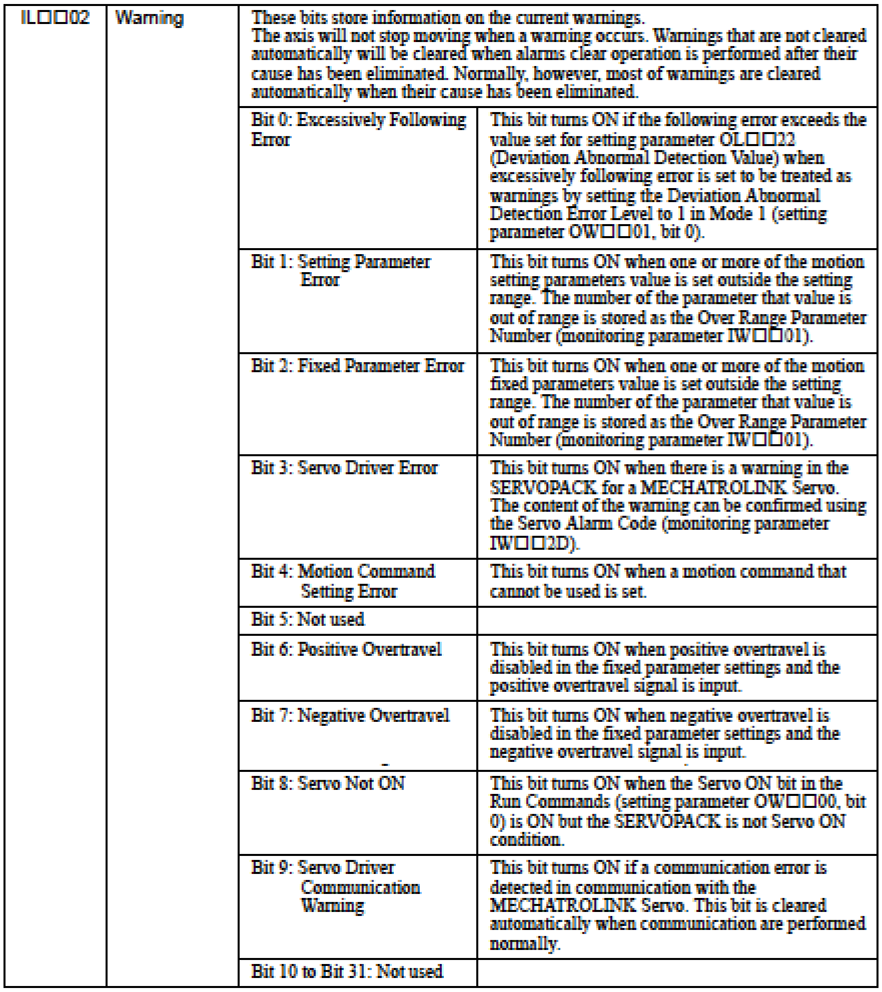
Σ-III Series



Name	Register Number	Code	Meaning
Servo Alarm Code	IW□□2D	000	Normal
		900	Excessive Position Error
		901	Excessive Position Error at Servo ON
		910	Overload
		911	Vibration
		920	Regeneration Overload
		930	Absolute Encoder Battery Error
		941	Parameter Change Requiring Power Recycling
		94A	Data Setting Warning 1 (Parameter Number)
		94B	Data Setting Warning 2 (Outside Data Range)
		94C	Data Setting Warning 3 (Calculation Error)
		94D	Data Setting Warning 4 (Parameter Size)
		95A	Command Warning 1 (Command Conditions Not Met)
		95B	Command Warning 2 (Unsupported Command)
		95C	Command Warning 3
		95D	Command Warning 4
		95E	Command Warning 5
		960	MECHATROLINK Communication Warning
		020	Parameter Checksum Error 1
		021	Parameter Format Error 1
		022	System Constant Checksum Error 1
		023	Parameter Password Error 1
		02A	Parameter Checksum Error 2
		02B	System Constant Checksum Error 2
		030	Main Circuit Detector Error
		040	Parameter Setting Error 1
		04A	Parameter Setting Error 2
		041	Divided Pulse Output Setting Error
		042	Parameter Combination Error
		050	Combination Error
		100	Overcurrent or Heat Sink Overheat
		300	Regeneration Error
		320	Regeneration Overload
		330	Main Circuit Wiring Error
		400	Overvoltage
		410	Undervoltage
		510	Overspeed
		511	Divided Pulse Output Overspeed
		520	Vibration Alarm
		710	Overload (Instantaneous Maximum Load)
		720	Overload (Continuous Maximum Load)
		730	DB Overload
		740	Inrush Resistance Overload
		7A0	Heat Sink Overheat
		810	Encoder Backup Alarm
		820	Encoder Checksum Alarm



Name	Register Number	Code	Meaning
Servo Alarm Code	IW□□2D	830	Encoder Battery Alarm
		840	Encoder Data Alarm
		850	Encoder Over Speed
		860	Encoder Overheat
		870	Full-closed Serial Encoder Checksum Alarm
		880	Full-closed Serial Encoder Data Alarm
		8A0	Full-closed Serial Encoder Scale Error
		8A1	Full-closed Serial Encoder Module Error
		8A2	Full-closed Serial Encoder Sensor Error (Incremental)
		8A3	Full-closed Serial Encoder Position Error (Absolute Value)
		B31	Current Detection Error 1
		B32	Current Detection Error 2
		B33	Current Detection Error 3
		BF0	System Alarm 0
		BF1	System Alarm 1
		BF2	System Alarm 2
		BF3	System Alarm 3
		BF4	System Alarm 4
		C10	Servo Run-away
		C80	Encoder Clear Error Multiturn Limit Setting Error
		C90	Encoder Communication Error
		C91	Encoder Communication Position Data Acceleration Error
		C92	Encoder Communication Timer Error
		CA0	Encoder Parameter Error
		CB0	Encoder Echoback Error
		CC0	Multiturn Limit Mismatch
		CF1	Full-closed Serial Conversion Unit Communication Error (Reception Failure)
		CF2	Full-closed Serial Conversion Unit Communication Error (Timer Stopped)
		D00	Excessive Position Error
		D01	Excessive Position Error Alarm at Servo ON
		D02	Excessive Position Error Alarm for Speed Limit at Servo ON
		D10	Excessive Error between Motor Load and Position
		E00	COM Alarm 0
		E01	COM Alarm 1
		E02	COM Alarm 2
		E07	COM Alarm 7
		E40	MECHATROLINK-II Transmission Cycle Setting Error
		E50	MECHATROLINK-II Sync Error
		E51	MECHATROLINK-II Sync Failure
		E60	MECHATROLINK-II Communication Error
		E61	MECHATROLINK-II Transmission Cycle Error
		EA0	DRV Alarm 0
		EA1	DRV Alarm 1
		EA2	DRV Alarm 2
		ED0	Internal Command Error
		F10	Phase Open on Power Line



IL□□02	Warning	These bits store information on the current warnings. The axis will not stop moving when a warning occurs. Warnings that are not cleared automatically will be cleared when alarms clear operation is performed after their cause has been eliminated. Normally, however, most of warnings are cleared automatically when their cause has been eliminated.	
		Bit 0: Excessively Following Error	This bit turns ON if the following error exceeds the value set for setting parameter OL□□22 (Deviation Abnormal Detection Value) when excessively following error is set to be treated as warnings by setting the Deviation Abnormal Detection Error Level to 1 in Mode 1 (setting parameter OW□□01, bit 0).
		Bit 1: Setting Parameter Error	This bit turns ON when one or more of the motion setting parameters value is set outside the setting range. The number of the parameter that value is out of range is stored as the Over Range Parameter Number (monitoring parameter IW□□01).
		Bit 2: Fixed Parameter Error	This bit turns ON when one or more of the motion fixed parameters value is set outside the setting range. The number of the parameter that value is out of range is stored as the Over Range Parameter Number (monitoring parameter IW□□01).
		Bit 3: Servo Driver Error	This bit turns ON when there is a warning in the SERVOPACK for a MECHATROLINK Servo. The content of the warning can be confirmed using the Servo Alarm Code (monitoring parameter IW□□2D).
		Bit 4: Motion Command Setting Error	This bit turns ON when a motion command that cannot be used is set.
		Bit 5: Not used	
		Bit 6: Positive Overtravel	This bit turns ON when positive overtravel is disabled in the fixed parameter settings and the positive overtravel signal is input.
		Bit 7: Negative Overtravel	This bit turns ON when negative overtravel is disabled in the fixed parameter settings and the negative overtravel signal is input.
		Bit 8: Servo Not ON	This bit turns ON when the Servo ON bit in the Run Commands (setting parameter OW□□00, bit 0) is ON but the SERVOPACK is not Servo ON condition.
		Bit 9: Servo Driver Communication Warning	This bit turns ON if a communication error is detected in communication with the MECHATROLINK Servo. This bit is cleared automatically when communication are performed normally.
		Bit 10 to Bit 31: Not used	

IL□□04	Alarm	These bits store information on the current alarms. The axis will stop moving when an alarm occurs. Alarms are cleared when the alarm clear operation is performed after their cause has been eliminated.	
		Bit 0: Servo Driver Error	This bit turns ON when there is an alarm in the SERVOPACK for a MECHATROLINK Servo. The content of the alarm can be confirmed using the Servo Alarm Code (monitoring parameter IW□□2D).
		Bit 1: Positive Overtravel	This bit turns ON when the positive overtravel signal is input and a move command is executed in the positive direction.
		Bit 2: Negative Overtravel	This bit turns ON when the negative overtravel signal is input and a move command is executed in the negative direction.
		Bit 3: Positive Soft Limit	This bit turns ON if a move command that exceeds the positive soft limit is executed with the followinging condtions: ・Zero point return has been complited ・The positive soft limit function is enabled ・An infinite length axis is selected.
		Bit 4: Negative Soft Limit	This bit turns ON if a move command that exceeds the negative soft limit is executed with the followinging condtions: ・Zero point return has been complited ・The negative soft limit function is enabled ・An infinite length axis is selected.
		Bit 5: Servo OFF	This bit turns ON when a move command is executed with the Servo OFF status.
		Bit 6: Positioning Time Over	This bit turns ON when positioning is not completed within the specified time after the pulse distribution end. The time is set for the Position Complete Timeout (setting parameter OW□□26).
		Bit 7: Excessive Positioning Moving Amount	This bit turns ON when a moving amount is specified that exceeds the setting range for the positioning moving amount.
		Bit 8: Excessive Speed	This bit turns ON when a speed is set that exceeds the setting range for the speed reference.
		Bit 9: Excessively Following Error	This bit turns ON if the following error exceeds the value set for the Deviation Abnormal Detection Value (setting parameter OL□□22) when Excessively Following Error is set to be treated as alarm by setting the Deviation Abnormal Detection Error Level to 0 in Mode 1 (setting parameter OW□□01, bit 0).
		Bit 10: Filter Type Change Error	This bit turns ON if the filter type is changed when the pulses are still distributing.
		Bit 11: Filter Time Constant Change Error	This bit turns ON if the filter time constant is changed when the pulses are still distributing.
		Bit 12: Not used	
		Bit 13: Zero Point Not Set	This bit turns ON if a move command (except for JOG or STEP) is performed when an infinite length axis is set and the zero point has not been set.
		Bit 14: Zero Point Set during Travel	This bit turns ON if the zero point is set during axis moving.
		Bit 15: Servo Driver Parameter Setting Error	This bit turns ON if a failure occurs while changing MECHATROLINK Servo parameter settings.
		Bit 16: Servo Driver Synchronization Communication Error	This bit turns ON if a synchronization communication error is detected with the MECHATROLINK Servo.
		Bit 17: Servo Driver Communication Error	This bit turns ON if two communication errors are detected consecutively in communication with the MECHATROLINK Servo.
		Bit 18: Servo Driver Command Timeout Error	This bit turns ON if a command sent to the MECHATROLINK Servo is not completed within a specific amount of time.
		Bit 19: ABS Encoder Count Exceeded	This bit turns ON if the number of turns from the absolute encoder exceeds the range that the MP2300 can handle. This parameter is valid when using an absolute encoder and an infinite length axis.
		Bit 20 to Bit 31: Not used	

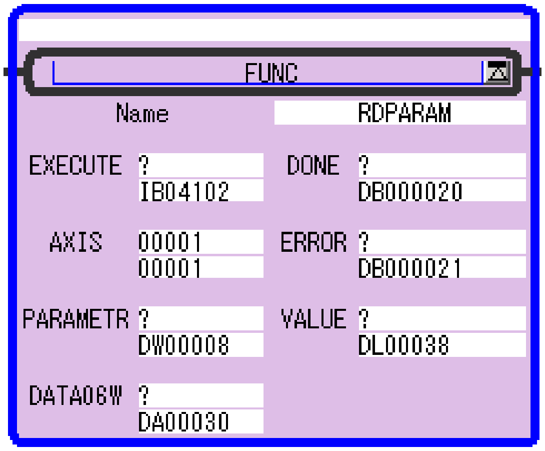
READ RDA PARAMETER function

Function block for MP2000 series

<RDPARAM> Function Block Summary

This function block reads the parameters of the RDA. The output is always a long even if the parameter is a word or a bit. For bits the value will be a 1 for true and a 0 for false. The parameter input value is the RDA number from the PLC Open Specification.

Function Block Diagram



FUNC

Name RDPARAM

EXECUTE ? IB04102

DONE ? DB000020

AXIS 00001 00001

ERROR ? DB000021

PARAMETR ? DW00008

VALUE ? DL00038

DATA06W ? DA00030

<RDPARAM> Function Block Operation Notes

- To use the function block, the *EXECUTE* bit must be True, and AXIS and PARAMETER will be constantly read.
- The block will continue to monitor the RDA for Values as long as the EXECUTE input stays True. If the *EXECUTE* bit goes low the outputs will be set to zero.
- Even though the VALUE output is a long it will also represent the bits and words.
- Six words are used as working registers for this function, starting at the address in *Data06W*.

<RDPARAM> Input and Output Register Map

Output Registers

The following registers are used as outputs from the function block. They can be monitored by the LadderWorks program to check the execution of the function.

Output	Type	Description
DONE	Bit	This bit is ON when the function block has a valid value from the RDA.
ERROR	Bit	If an error occurs during block execution, this output bit Latches ON If the error condition is cleared, this output is reset.
VALUE	Long	The value from the RDA. For bit conditions 1 = TRUE 0 = FALSE

Input Registers

The following registers are used as inputs to the function block. They select the options and define the parameters that the user needs to make the function work as necessary.

Input	Type	Description	Range and state
EXECUTE	Bit	Block enable – Block cannot execute unless this is TRUE.	Rising edge initiates block TRUE – Block enable FALSE – Block disable, set all outputs to zero and off
AXIS	Word	Axis number related to the block.	1～16
PARAMETER	Word	RDA parameter number.	The number is specified by the PLCOpen and Yaskawa specification.
DATA06W	**Address**	Address of the first working register.	Six words of register space are used by this function.

<RDPARAM> Block Fault Condition:

The following table outlines several situations that may cause an error, and will turn on the blocks “Error” output bit. The block “Error” output will cleared if the *EXECUTE* bit goes low.

Fault Bit	Cause	Note
axisInErr AB00014A	The axis number entered on the input is not an acceptable value	The function blocks can only control 1 to 16 axes. Any value greater or smaller then this will cause an error. This does not set the RDA Error ID.
error AB000009	Parameter specified is not a valid RDA number	The number specified must be either the Yaskawa specified RDA area or the PLCOpen specified area. This block can not monitor Master/Slave data. Sets RDA Error ID (MW3**81) bit 3 on if error state exists.

<RDPARAM> Working Register

This table outlines the data in the six registers used by the function block. There is not usually any need for the user to access any of these bits directly.

Register No.	Type	Name	Description
AW00000	Working	Reserved	Reserved
AW00001	Working	Reserved	Reserved
AW00002	Working	iStore	Reserved
AW00003	Working	ivalue	Reserved
AW00004	Working	rdaMult	Value for address offset to locate proper RDA
AW00005	Working	Revision	Revision Level of the function block.

<RDPARAM> Parameter List

The following tables show the accessible parameters by RDPARAM. # 001-#999 are PLC Open specification and above # 1000 are Yasukawa specification.

RDA#	Data name	Details
005	Movement Type	0: Rotary 1: Liner
006	Position Period	Length of Period for rotational systems. [count]
007	Set Position	Commanded position. [count]
008	Act Position	Actual position. [count]
009	Max Velocity	Maximum velocity. [count/sec]
010	Set Velocity	Commanded velocity. [count/sec]
011	Act Velocity	Actual Velocity of axes. [count/sec]
012	Set Acceleration	Commanded acceleration. [count/sec^2]
013	Act Acceleration	Actual acceleration. [count/sec^2]
014	Max Acceleration	Maximum acceleration. [count/sec^2]
015	Set Deceleration	Commanded deceleration. [count/sec^2]
016	Act Deceleration	Actual deceleration. [count/sec^2]
017	Max Deceleration	Maximum deceleration. [count/sec^2]
018	Set S-Curve Filter	Commanded S-Curve Filter [ms] (S curve time)
019	Act S-Curve Filter	Actual S-Curve Filter [ms] (S curve time)
020	Max S-Curve Filter	Maximum S-Curve filter [ms] (S curve time)
021	Act Torque	Actual Torque [0.01% of rated torque]
027	HW Limit Enable	Enable / disable hardware end switch (to be used after
028	Capt Position	Capture position [count]
029	Capture Occurred	Capture signal occurred (reset with writing)
030	Ramp Shape	Shape of Acc/Dec profile. 0 = Trapezoid; 1= S-Shape; rest supplier dependent
031	Axis State	State of the Axis 0: reserved for power off situation 1:ErrorStopped Motion 2: Stopped Motion 3: Standstill 4: Discrete motion 5: Continuous motion

1000	Clear Pending Action	
1001	Abort	Stop axis discontinuous motion (RDA#031)
1002	Command Bit	One Scan Pulse of Motion Block Execution
1003	Accelerating	The motion is in acceleration.
1004	Decelerating	The motion is in deceleration.
1005	Steady	?
1006	Ext Pos. Captured	Registration occurred
1033	Motor Rated Speed	Rated speed of servo motor [rpm]
1034	Encoder Resolution	Servo motor PG post-quad resolution value [pulse/rev] resolution
1035	Factor FF	Feed Forward gain (1000=100%)
1036	Enable Type	bit0: Servo ON, bit1: Positive Enabled, bit2: Negative Enabled
1037	Move State	The last motion commanded; 0:Stop, 1:MOVVEL, 2:MOVINC, 3:MOVABS, 4:MOVADDTV, 5:HOME, 6:LTCHTGT, 7:GEAR, 8:CAM
1038	Block Running	The last executed Function block Number.
1039	STATUS	Bit0:Running, Bit1:Error, bit2:Interruption
1040	Error ID 1	bit0: No Home Switch, bit1: Over travel, bit2: Time-out, bit3: Value to Great, bit4: Direction Not Allowed, bit5: Position error, bit6: No Motor Power, bit7: Servo alarm bit8: Spare, bit9: Track Fail, bitA: RDA error, bitB: Error stop, bitC: Table error, bitD: Servo off
1041	Error ID 2	Spare
1043	Ext Pos. Captured	Registered position [count]
1044	Module number	Module No
1045	Axis number	Axis No

1060	CAM type	OFF: Return, ON: One way
1061	Positive Cycle End	On when CAM Master position, includes CAM-Shift, rolled over in positive direction.
1062	Negative Cycle End	On when CAM Master position, includes CAM-Shift, rolled over in Negative direction.
1092	CAM state	State of CAMing. bit0: Disengaged, bit1:Waiting to Engage, bit2:CAMing is locked, bit3:Waiting to disengage
1093	TABLE size	Cam Table Size
1094	CAM shift	Absolute CAM Shift amount [count]
1095	SLAVE offset	Absolute Offset amount [count]
1096	CAM scale	Absolute Acale amount [0.01%]
1097	Machine Cycle	Machine Cycle [count]
1098	Data Counter	Raw master data for master-slave pair [count]
1099	Mod Data	Modulated master data for master/slave pair. [count]
1105	GEAR state	State of Gearing. bit0:Disengaged, bit1:Accelerating, bit2:Gearing is locked and synched, bit3:Decelerating

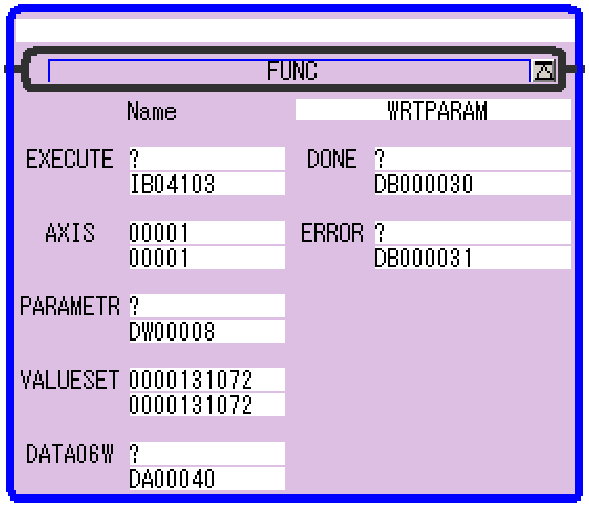
WRITE RDA PARAMETER function

Function block for MP2000 series

<WRTPARAM> Function Block Summary

This function block writes values to the RDA parameter. The value set is always a long even if the parameter calls to a bit. In case of bit information, ON=1 and OFF=0. The input value of PARAMETR is RDA number of the PLC Open specification.

Function Block Diagram



FUNC

Name			WRTPARAM
EXECUTE	? IB04103	DONE	? DB000030
AXIS	00001 00001	ERROR	? DB000031
PARAMETR	? DW00008		
VALUESET	0000131072 0000131072		
DATA06W	? DA00040		

<WRTPARAM> Function Block operation Notes

- To use the function block, the *EXECUTE* bit must be True.
- The block will write the value to the parameter in the RDA only on the rising edge of the Execute input and if there is no errors reported in the block.
- If the *EXECUTE* bit goes low the outputs will be set to zero.
- To set a bit in the RDA, enter the value of a one in the VALUESET input and execute block. A zero in the VALUESET input will reset the bit.
- Even though this block may write to a parameter care must be taken that the parameter not be one that is controlled by another block. (ie. Actual position is written to by the SVON block. Any changes to this value will cause the value to be over written by the SVON block the very next scan.
- Six words are used as working registers for this function, starting at the address in *Data06W*.

<WRTPARAM> Input and Output Register Map

Output Registers

The following registers are used as outputs from the function block. They can be monitored by the LadderWorks program to check the execution of the function.

Output	Type	Description
DONE	Bit	When the value has been successfully stored and the execute input is high, this bit will go high
ERROR	Bit	If an error occurs during block execution, this output bit Latches ON.

Input Registers

The following registers are used as inputs to the function block. They select the options and define the parameters that the user needs to make the function work as necessary.

Input	Type	Description	Range and state
EXECUTE	Bit	Block enable – Block cannot execute unless this is TRUE.	Rising edge initiates block TRUE – Block enable FALSE – Block disable, set all outputs to zero and off
AXIS	Word	Axis number related to the block	1～16
PARAMETR	Word	RDA parameter number	It must be RDA number of the PLC Open or Yasukawa specification
VALUESET	Long	Value to be stored in the RDA	Since a long is used for all RDA parameters an Error will occur if the value is to great for the word or bit that is being set. For bit, ON=1 and OFF=0.
DATA06W	**Address**	Address of the first working register.	Six words of register space are used by this function.

<WRTPARAM> Block Fault Condition:

The following table outlines several situations that may cause an error. The block error can be cleared by the *EXECUTE* bit going low

Fault Bit	Cause	Note
axisInErr AB00000F	The axis number entered on the input is not an acceptable value	The function blocks can only control 1 to 16axis. Any value greater or smaller then this will cause an error. This does not set the RDA Error ID.
Notspec AB00000E	Parameter specified is not a valid RDA number	The number specified must be either the Yaskawa specified RDA area or the PLCOpen specified area. This block can not monitor Master/Slave data. Sets RDA Error ID (MW3**81) bit 3 on if error state exists.
valueOut AB00000D	Value out of range for the RDA	Value must be within range of value being stored or error will occur. Sets RDA Error ID (MW3**81) bit 3 on if error state exists.

<WRTPARAM> Working Registers

This table outlines the data in the six registers used by the function block. There is not usually any need for the user to access any of these bits directly.

Register No	TYPE	Name	Content
AW00000	Working	Reserved	Reserved
AW00001	Working	Reserved	Reserved
AW00002	Working	iStore	Reserved
AW00003	Working	ivalue	Reserved
AW00004	Working	rdaMult	Value for address offset to locate proper RDA
AW00005	Working	Revision	Revision Level of the function block.

<WRTPARAM> Parameter list

The following table shows the accessible parameters by RDPARAM. #001 through #999 are PLCOpen specification and above #1000 are supplier specific..

RDA#	Data name	Details
005	Movement Type	0: Rotor 1: Liner
006	Position Period	Length of Period for rotational systems. [count]
007	Set Position	Commanded position. [count]
008	Act Position	Actual position. [count]
009	Max Velocity	Maximum velocity. [count/sec]
010	Set Velocity	Commanded velocity. [count/sec]
011	Act Velocity	Actual Velocity of axes. [count/sec]
012	Set Acceleration	Commanded acceleration. [count/sec^2]
013	Act Acceleration	Actual acceleration. [count/sec^2]
014	Max Acceleration	Maximum acceleration. [count/sec^2]
015	Set Deceleration	Commanded deceleration. [count/sec^2]
016	Act Deceleration	Actual deceleration. [count/sec^2]
017	Max Deceleration	Maximum deceleration. [count/sec^2]
018	Set S-Curve Filter	Commanded S-Curve Filter [ms] (S curve time)
019	Act S-Curve Filter	Actual S-Curve Filter [ms] (S curve time)
020	Max S-Curve Filter	Maximum S-Curve filter [ms] (S curve time)
021	Act Torque	Actual Torque [0.01% of rated torque]
027	HW Limit Enable	Enable / disable hardware end switch (to be used after
028	Capt Position	Capture position [count]
029	Capture Occurred	Capture signal occurred (reset with writing)
030	Ramp Shape	Shape of Acc/Dec profile. 0 = Trapezoid; 1= S-Shape; rest supplier dependent
031	Axis State	State of the Axis 0: reserved for power off situation 1:ErrorStopped Motion 2: Stopped Motion 3: Standstill 4: Discrete motion 5: Continuous motion

1000	Clear Pending Action	
1001	Abort	Stop axis discontinuous motion (RDA#031)
1002	Command Bit	One Scan Pulse of Motion Block Execution
1003	Accelerating	The motion is in acceleration.
1004	Decelerating	The motion is in deceleration.
1005	Steady	?
1006	Ext Pos. Captured	Registration occurred
1033	Motor Rated Speed	Rated speed of servo motor [rpm]
1034	Encoder Resolution	Servo motor PG post-quad resolution value [pulse/rev] resolution
1035	Factor FF	Feed Forward gain (1000=100%)
1036	Enable Type	bit0: Servo ON, bit1: Positive Enabled, bit2: Negative Enabled
1037	Move State	The last motion commanded; 0:Stop, 1:MOVVEL, 2:MOVINC, 3:MOVABS, 4:MOVADDTV, 5:HOME, 6:LTCHTGT, 7:GEAR, 8:CAM
1038	Block Running	The last executed Function block Number.
1039	STATUS	Bit0:Running, Bit1:Error, bit2:Interruption
1040	Error ID 1	bit0: No Home Switch, bit1: Over travel, bit2: Time-out, bit3: Value to Great, bit4: Direction Not Allowed, bit5: Position error, bit6: No Motor Power, bit7: Servo alarm bit8: Spare, bit9: Track Fail, bitA: RDA error, bitB: Error stop, bitC: Table error, bitD: Servo off
1041	Error ID 2	Spare
1043	Ext Pos. Captured	Registered position [count]
1044	Module number	Module No
1045	Axis number	Axis No

1060	CAM type	OFF: Return, ON: One way
1061	Positive Cycle End	On when CAM Master position, includes CAM-Shift, rolled over in positive direction.
1062	Negative Cycle End	On when CAM Master position, includes CAM-Shift, rolled over in Negative direction.
1092	CAM state	State of CAMing. bit0: Disengaged, bit1:Waiting to Engage, bit2:CAMing is locked, bit3:Waiting to disengage
1093	TABLE size	Cam Table Size
1094	CAM shift	Absolute CAM Shift amount [count]
1095	SLAVE offset	Absolute Offset amount [count]
1096	CAM scale	Absolute Acale amount [0.01%]
1097	Machine Cycle	Machine Cycle [count]
1098	Data Counter	Raw master data for master-slave pair [count]
1099	Mod Data	Modulated master data for master/slave pair. [count]
1105	GEAR state	State of Gearing. bit0:Disengaged, bit1:Accelerating, bit2:Gearing is locked and synched, bit3:Decelerating

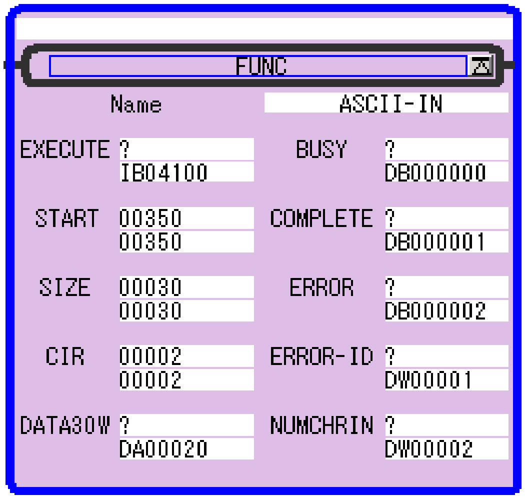
ASCII CHARACTER INPUT Function Block

Function block for MP2000 series

<ASCIIIN> Function Block Summary

This function block receives a sequence of 7-bit or 8-bit ASCII characters of a specified size through the controllers serial port, RS232 or RS422/485 from a transmitter device. This sequence of characters is then stored in table format at an offset global MW register location.

Function Block Diagram



Name		ASCII-IN	
EXECUTE	? IB04100	BUSY	? DB000000
START	00350 00350	COMPLETE	? DB000001
SIZE	00030 00030	ERROR	? DB000002
CIR	00002 00002	ERROR-ID	? DW00001
DATA30W	? DA00020	NUMCHRIN	? DW00002

<ASCIIIN> Function Block Operation Notes

- To use the function block, the *EXECUTE* bit must be held ON.
- If the *EXECUTE* bit goes off during operation, the block will stop receiving data.
- On the rising edge of the EXECUTE input, the following two are executed. The EXECUTE input should be maintained ON while in operation.
  - The set value of START and SIZE are read.
  - The storage table is initialized by data H2020(Null).
- ERROR–ID output has been added to better troubleshoot communication problems. A zero in the value means no errors or block disabled.
- Protocol settings must be configured in controller module setup and transmitter setup, and must match (baud rate, parity, stop bit, # data bits).
- This ASCIIin function block uses a MSG-RCV an embedded function. Since the transmission protocol is set to none (internal setting =3(no procedure 2)(The data of each byte is transferred according to a no procedure)), the Master/Slave setting in module configuration is not necessary.

- This function expects the characters to be received from the source in a continuous stream (Note: if testing is performed by typing in one character at a time from a keyboard, the data reception MSG-RCV will time-out, and only the first character will be received, and the DONE signal will come on).
- Thirty words are used as working registers for this function, starting at the address in *Data30W*

<ASCIIN> Input and Output Register Map

Output Registers

The following registers are used as outputs from the function block. They can be monitored by the LadderWorks program to check the execution of the function.

OUTPUT	TYPE	Content
BUSY	Bit	High while block is receiving data and there are no errors.
DONE	Bit	Goes high when receiving is completed and remains high until execute is turned off.
ERROR	Bit	Goes high if any error occurs in block or on the axis
ERROR-ID	Word	Displays the value of the error in block.
NUMCHRIN	Word	Displays the number of characters received.

Input Registers

The following registers are used as inputs to the function block. They select the options and
define the parameters that the user needs to make the function work as necessary.

INPUT	TYPE	Content	Range of State
EXECUTE		Block enable – Block cannot execute unless this is TRUE.	TRUE – Block enable FALSE – Block disable
START	Word	Starting address to store ASCII set. This is fed directly into Param12 of MSG-RCV function. This is the address for an M-register type	0-29999(unused M register area) and 30000 and higher have been reserved by the RDA parameter.
SIZE	Word	Maximum number of ASCII characters to be received. Start + Size is fed directly into Param13 of MSG-RCV function	0～28767
CIR	Word	Which port is to transmitted from	For 217IF-01 1=Port 1 (RS-232C) 2=Port 2 (RS-422/485)
DATA30W	**Address**	Address of the first working register.	Thirty words of register space are used by this function.

<ASCIIIN> Block Fault Conditions:

The following table outlines several situations that may cause an error. The block error can be cleared by the *EXECUTE* bit going low.

Internal Fault Bit	Cause	Note
rcverror AB00001F	MSG-RCV has an error.	The MSG-RCV internal function has an error. Refer to ERROR-ID output for value. 81= Function code error 82= Address set error 83= Data size error 84= Line number setting error 85= Channel number error 86= Station Address error 88= Transmission unit error 89= Device selection error

<ASCIIIN> Working Registers

This table outlines the data in the Thirty registers used by the function block. There is not usually any need for the user to access any of these bits directly. The MSG-RCV type registers can be further understood by referencing the MSG-RCV function block in the on line help.

Register No.	Type	Name	Description
AW00000			
Bit 0	IN	execute	*EXECUTE* input (XB000000)
Bit 7	OUT	running	Directly controls YB000000 (*RUNNING* Output)
Bit 8	OUT	done	Directly controls YB000001 (*COMPLETE* Output)
Bit 9	OUT	error	Directly controls YB000002 (*ERROR* Output)
Bit A	Working	oneshotA	Reserved
Bit B	Working	firstPass	One shot on rising edge of EXECUTE input to initialize axis.
AW00001	MSG-RCV	Reserved	Reserved
AW00002	MSG-RCV	Reserved	Reserved
AW00003	MSG-RCV	slave	Called station # (Param2)
AW00004	MSG-RCV	Reserved	Reserved
AW00005	MSG-RCV	Reserved	Reserved
AW00006	MSG-RCV	startAdd	Data address (Param05)
AW00007	MSG-RCV	size	Data size (Param06)
AW00008	MSG-RCV	called_CPU	Called CPU# (Param7)
AW00009	MSG-RCV	Reserved	Reserved

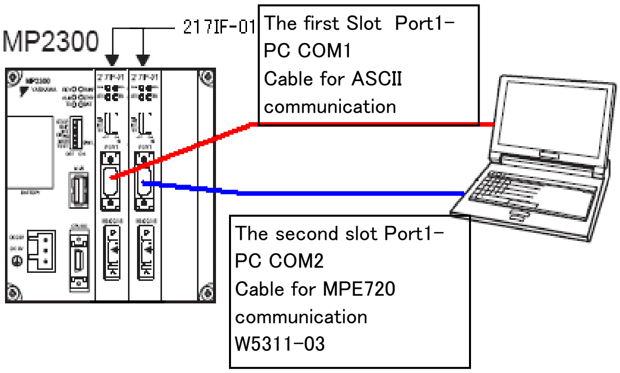
AW00010	MSG-RCV	Reserved	Reserved
AW00011	MSG-RCV	Reserved	Reserved
AW00012	MSG-RCV	holding_offset	Value to shift data in register (Param11, this is meaningless for non-protocol mode)
AW00013	MSG-RCV	starAddress	First address of M register to store data (This sets Param12)
AW00014	MSG-RCV	endAdd	End address of M register to store data (this sets param13)
AW00015	MSG-RCV	system	Reserved
AW00016	MSG-RCV	Reserved	Reserved
AW00017	MSG-RCV	Reserved	Reserved
AW00018	Working	Reserved	Reserved
AW00019	Working	Reserved	Reserved
AW00020	OUT	errorID	Directly controls *ERROR-ID* output (YW00001)
AW00021	Working	iStore	Reserved
AW00022	Working	step	Reserved
AW00023	Working	endAddress	Final M register to store data
AW00024	Working	Reserved	Reserved
AW00025	Working	Reserved	Reserved
AW00026	Working	filterLow	Reserved
AW00027	Working	filterHigh	Reserved
AW00028	OUT	chrCount	Number of Characters received Directly controls *ERROR-ID* output (YW00001).
AW00029	Working	Revision	Revision Level of the function block.

<ASCIIN> Example of application:

The following examples illustrates how to receive ASCII characters from PC COM1 to Port#1 of the MP2300 217IF-01 module (In the First Slot) via RS-232C. MotionWorks is used to monitor the received Character data in registers MW04032 through MW04039 (16 characters). Hyper-terminal^{tm} is used to send the character string in a Text file.

Hardware setup

Yaskawa W5211-03 cable connects PC COM2 and MP2300 217IF-01 module (the second slot) Port#1 for MotionWorks.
PC COM1 and MP2300 217IF-01 module (The first slot) Port#1 is connected for ASCII communication. Refer to the user's manual for the pin assignment.

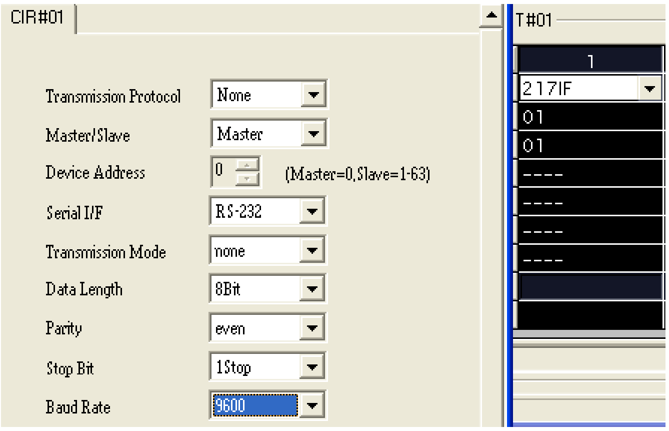


The first Slot Port1-
PC COM1
Cable for ASCII
communication

The second slot Port1-
PC COM2
Cable for MPE720
communication
W5311-03

MP2300 MotionWorks Software Module Setup

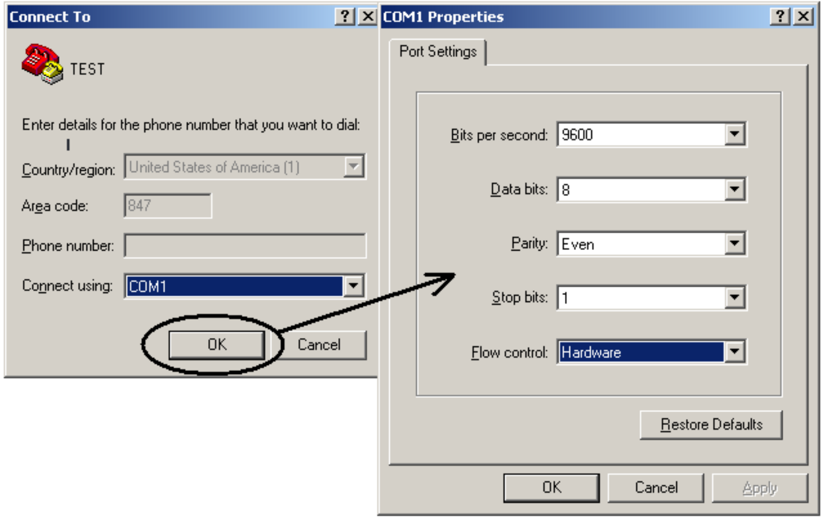
Set the MP2300 217IF-01 module Port#1 (The first slot) serial interface for ASCII communication. Note that Transmission protocol is set to None. Master/Slave and Serial I/F settings are not used so they will be ignored . Protocol settings are as follows:
Transmission protocol: No procedure
Transmission mode: None
Data length: 8bit
Parity bit: Even
Stop bit: 1 Stop
Baud rate: 9600bps



Transmission Protocol None
Master/Slave Master
Device Address 0 (Master=0,Slave=1-63)
Serial I/F RS-232
Transmission Mode none
Data Length 8Bit
Parity even
Stop Bit 1Stop
Baud Rate 9600

Hyper-Terminal Serial Port (PC COM1) Setup

Setting the serial interface on the PC COM1 for ASCII communication is as follows. (Hyper–Terminal is used)
Baud rate:9600bps
Data length:8bit
Parity bit:Even number
Stop bit:1 Stop
Flow control:Hardware



Connect To

TEST

Enter details for the phone number that you want to dial:

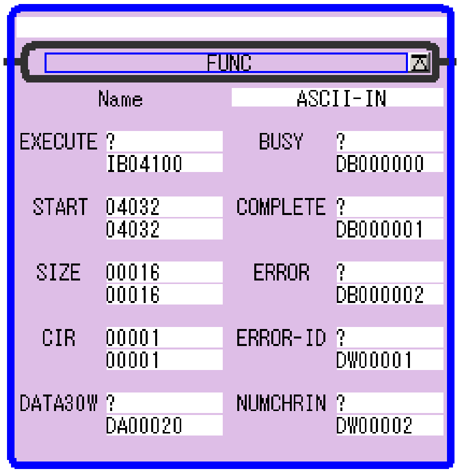
Country/region: United States of America (1)
Area code: 847
Phone number:
Connect using: COM1

COM1 Properties

Bits per second: 9600
Data bits: 8
Parity: Even
Stop bits: 1
Flow control: Hardware

Setup of ASCII-IN function

The ASCII–IN function block is inserted in the Low Speed drawing L01, called from L using the LadderWorks editor (see block on right). It is important to also note that the function block uses 30 words of local registers, care must be taken not to overwrite those registers. In this example, local I/O is used to control the function block execution.
Note that the "START" input is set to 4032. With the "SIZE" input set to 16, the Character table source is then setup from MW04032 through MW04039 (8 16–bit words holds 16 ASCII characters), The CIR input is set to 1 to indicate Port#1 will be used to receive ASCII characters from the PC Htperterminal. Also MW30000 and higher are reserved for function block operation (RDA). Avoid using RDA area for ASCII character table.



FUNC

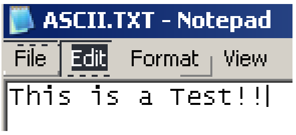
Name ASCII-IN

EXECUTE	? IB04100	BUSY	? DB000000
START	04032 04032	COMPLETE	? DB000001
SIZE	00016 00016	ERROR	? DB000002
CIR	00001 00001	ERROR-ID	? DW00001
DATA30W	? DA00020	NUMCHRIN	? DW00002

Making of ASCII file (TEXT file)

The character string can be created in a Text file like Notepad.

ASCII.TXT

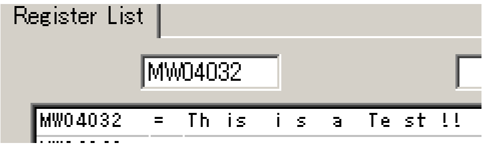


ASCII.TXT - Notepad

```
This is a Test!!
```

Test of program operation

Execution of the function block is started by switching 'on' input bit IB04100. Then send the Text file with the HyperTerminal. The received characters are seen in the Register List.



Register List

MW04032

```
MW04032  =  Th is  i s  a  Te st !!
```

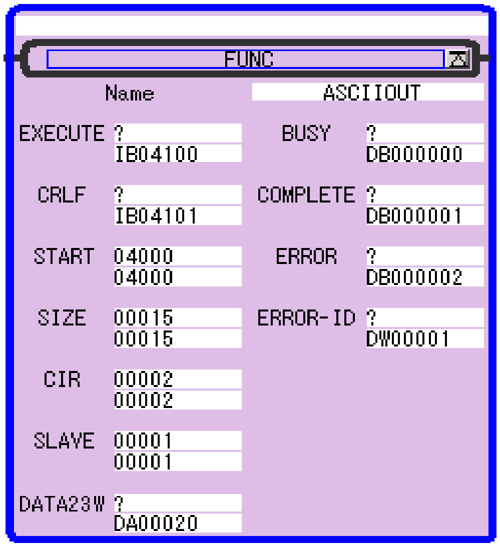
ASCII CHARACTER OUTPUT function

Function block for MP2000 series

<ASCIIOUT> Function Block Summary

This function block sends a sequence of 7-bit or 8-bit ASCII characters of a specified size through the controllers serial port, RS232 or RS422/485 to a receiver device. This sequence of characters is stored in table format at an offset global MW register location. A carriage return (CR) and line feed (LF) can be added to the end of the table if needed.

Function Block Diagram



FUNC

Name			ASCIIOUT
EXECUTE	? IB04100	BUSY	? DB000000
CRLF	? IB04101	COMPLETE	? DB000001
START	04000 04000	ERROR	? DB000002
SIZE	00015 00015	ERROR-ID	? DW00001
CIR	00002 00002		
SLAVE	00001 00001		
DATA23W	? DA00020		

<ASCIIOUT> Function Block Operation Notes

- Character transmission starts at positive edge of *EXECUTE input* .
- To use the function block, the *EXECUTE input* must be held ON.
- If a carriage return (CR) and line feed (LF) are needed on the end of the table the *CRLF* bit must be set high. Note that this will insert the Carriage Return and Line Feed (which takes up 8 bits each) at the end of the table. The last 16 bit word data-point in the table will be over written with this before it is sent out the serial port. In this case SIZE indicates the total size of the data to be *transmitted*, including CrLf.
- ERROR–ID output has been added to better troubleshoot communication problems. A zero in the value means no errors or block disabled.
- Protocol settings must be configured in controller module setup and receiver setup, and must match (baud rate, parity, stop bit, # data bits).
- This ASCIIout function block uses a MSG-SEND an embedded function. Since the transmission protocol is set to none (version 3, unit is by byte), the Master/Slave setting in module configuration is not necessary.
- Twenty-three words are used as working registers for this function, starting at the address in *Data23W*.

<ASCIIOUT> Input and Output Register Map

Output Registers

The following registers are used as outputs from the function block. They can be monitored by the LadderWorks program to check the execution of the function.

Output	Type	Description
BUSY	Bit	High while block is transmitting data and there are no errors.
DONE	Bit	Goes high when transmission is completed and remains high until execute is turned off.
ERROR	Bit	Goes high if any error occurs in block or on the axis
ERROR-ID	Word	Displays the value of the error in block.

Input Registers

The following registers are used as inputs to the function block. They select the options and
define the parameters that the user needs to make the function work as necessary.

Input	Type	Description	Range of State
EXECUTE	Bit	Block enable – Block cannot execute unless this is TRUE.	Pos Edge – Block execute TRUE – Block enable FALSE – Block disable
CRLF	Bit	*Inserts* a Carriage return and line feed to the end of the table (takes up one 16-bit word, table size remains the same (note this will overwrite the last word of the table with the CrLf).	TRUE – CR and LF are added FALSE – no addition
START	Word	Starting address of ASCII set. This is directly written to Param5 of MSG-SND function. It will represent the address of an M-register type.	0-29999(unused M register area). 30000 and higher have been reserved by the RDA parameter.
SIZE	Word	Number of ASCII characters to be sent (number of bites), also indicates table size This is directly written to Param6 of MSG-SND function	0～28767
CIR	Word	Indicates the circuit number or port number to send the characters	For 217IF-01 1=Port 1 (RS-232C) 2=Port 2 (RS-422/485)
SLAVE	Word	Destination Station Number This value is set in PARAM02 of the MSG-SND function.	1～256
DATA23W	**Address**	Address of the first working register.	Twenty-three words of register space are used by this function.

<ASCIIOUT> Block Fault Conditions:

The following table outlines several situations that may cause an error, and will turn on the blocks "Error" output bit. The block "Error" output will OFF if the *EXECUTE* bit goes low.

Internal Fault Bit	Cause	Note
inRng1 AB000003	Improper cir number. Out of range.	On the initial pass of the block the bit must be high or an error will occur. Sets RDA Error ID (MW3**81) bit 3 on if error state exists.
inRng2 AB000004	Improper slave address	On the initial pass of the block the bit must be high or an error will occur. Sets RDA Error ID (MW3**81) bit 3 on if error state exists.
snderr AB00001F	MSG-SND has an error.	The MSG-SND internal function has an error. Refer to ERROR-ID output for value. 81= function code error 82= address set error 83= data size error 84= line number set error 85= channel number set error 86= station address error 88= transmission part error 89= device selection error

<ASCIIOUT> Working Register

This table outlines the data in the Twenty-three registers used by the function block. There is not usually any need for the user to access any of these bits directly. The MSG-SND type registers can be further understood by referencing the MSG-SND function block in the on line help.

Register No	Type	Name	Description
AW00000			
Bit 0	IN	execute	EXECUTE input (XB000000)
Bit 1	Working	crlf	If True, a carriage return and Line feed are inserted at the end of the table.
Bit 2	Working	oddnum	Reserved
Bit 3	OUT	inRng1	Verifies that *CIR-NUM* input (XW00003) is in range
Bit 4	OUT	inRng2	Verifies that *SLAVE* input (XW00004) is in range
Bit 7	OUT	running	Directly controls YB000000 (*RUNNING* Output)
Bit 8	OUT	done	Directly controls YB000001 (*COMPLETE* Output)
Bit 9	OUT	error	Directly controls YB000002 (*ERROR* Output)
Bit A	Working	oneshotA	Reserved
Bit B	Working	firstPass	One shot on rising edge of EXECUTE input to initialize RDA & sets parameters in the message send function

AW00001	MSG-SND	Reserved	Reserved
AW00002	MSG-SND	Reserved	Reserved
AW00003	MSG-SND	slave	Called station # (Param2)
AW00004	MSG-SND	Reserved	Reserved
AW00005	MSG-SND	Reserved	Reserved
AW00006	MSG-SND	startAdd	Data address, starting address of data to be sent out (Param5)
AW00007	MSG-SND	size	Data size, number of characters to be sent out (Param6)
AW00008	MSG-SND	called_CPU	Called CPU# (Param7)
AW00009	MSG-SND	Reserved	Reserved
AW00010	MSG-SND	Reserved	Reserved
AW00011	MSG-SND	Reserved	Reserved
AW00012	MSG-SND	holding_offset	Holding register offset (Param11)
AW00013	MSG-SND	sys_send	For system (Param12)
AW00014	MSG-SND	Reserved	Reserved
AW00015	MSG-SND	Reserved	Reserved
AW00016	MSG-SND	Reserved	Reserved
AW00017	MSG-SND	Reserved	Reserved
AW00018	Working	Reserved	Reserved
AW00019	Working	Reserved	Reserved
AW00020	OUT	errorID	Directly controls *ERROR-ID* output (YW00001)
AW00021	Working	mod	Reserved
AW00022	Working	Revision	Revision Level of the function block.

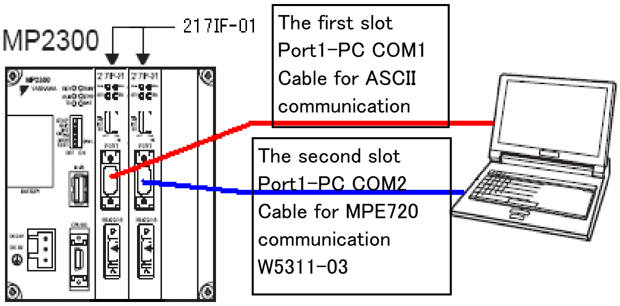
<ASCIIOUT> Application Example:

The following example illustrates how to send characters from Port #1 of an MP2300 217IF-01 module (in the first slot) to COM1 of a PC via RS232 . MotionWorks is used to set the Character table data in registers MW04000 through MW04007 (16 characters). Hyper-terminal^{tm} is used to receive and display the character string.

Hardware setup

Yaskawa W5311-03 cable is connected from Port#1 of the MP2300 217IF-01 module (the second slot) to PC COM2 for MotionWorks interface software.

Port#1 of the MP2300 217IF-01 module (the first slot) and PC COM1 are connected for ASCII communication (refer to the user's manual for wiring).



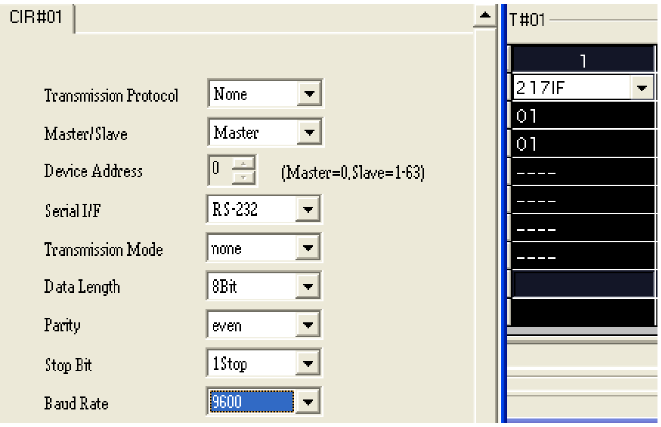
The first slot
Port1-PC COM1
Cable for ASCII
communication

The second slot
Port1-PC COM2
Cable for MPE720
communication
W5311-03

MP2300 MotionWorks Software Module setup

Setting the serial interface of MP2300 217IF-01 module Port#1 (the first slot) for ASCII communication is as follows. Note that Transmission protocol is set to None. Master/Slave and Serial I/F settings are not used so they will be ignored . Protocol settings are as follows:

Transmission protocol: No procedure
Transmission mode: None
Data length: 8bit
Parity bit: Even
Stop bit: 1 Stop
Baud rate: 9600bps



Transmission Protocol: None
Master/Slave: Master
Device Address: 0 (Master=0,Slave=1-63)
Serial I/F: RS-232
Transmission Mode: none
Data Length: 8Bit
Parity: even
Stop Bit: 1Stop
Baud Rate: 9600

Hyper-Terminal Serial Port Setup

A terminal session was opened up called "test". The serial port configuration for Com1 is set up identical to the MP2300 217IF-01 (the first slot) Port#1 settings

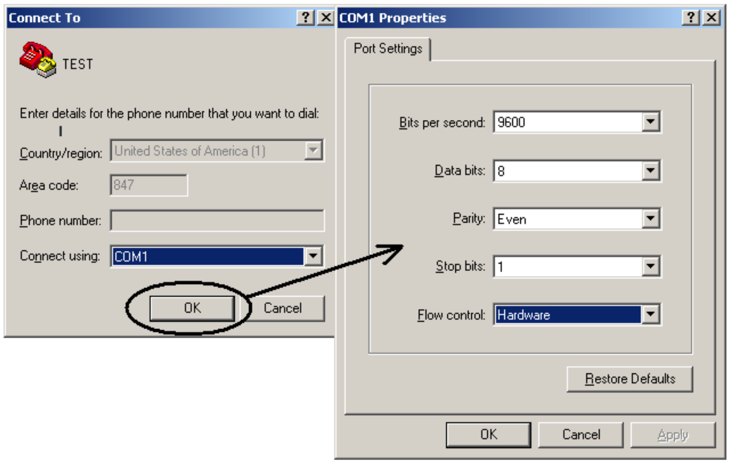
Baud rate:9600bps

Data length:8bit

Parity bit:Even number

Stop bit:1 Stop

Flow control:Hardware


Connect To

TEST

Enter details for the phone number that you want to dial:

Country/region: United States of America (1)

Area code: 847

Phone number:

Connect using: COM1

COM1 Properties

Port Settings

Bits per second: 9600

Data bits: 8

Parity: Even

Stop bits: 1

Flow control: Hardware

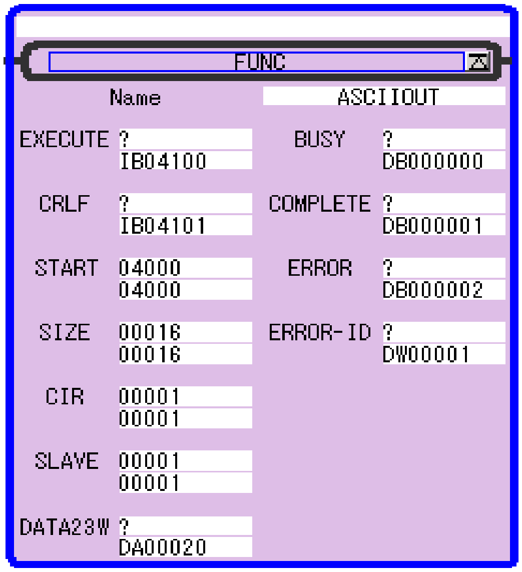
MotionWorks Application Software Setup

The ASCIIOUT function block is inserted in the LOW Speed drawing L01, called from L using the LadderWorks editor (see block on right). It is important to also note that the function block uses 23 words of local registers, care must be taken not to overwrite those registers. In this example, local I/O is used to control the function block execution, CR/LF select.

With setting "START" as 4000 and "SIZE" as 16, the character table source is then setup from MW04000 to MW04007 (ASCII character 16 characters are stored in the register of eight words) If CR/LF is used, then two more characters will be inserted at the end of the table, decreasing the user character table size to MW4000 to MW4006 (7-16-bit words holds 14 ASCII characters).

The "CIR" input is set to 1 to indicate Port#1 of the module will be used to send ASCII characters to the PC Hyper-Terminal.

MW30000 and higher are RDA parameter. Avoid using RDA for ASCII character table.

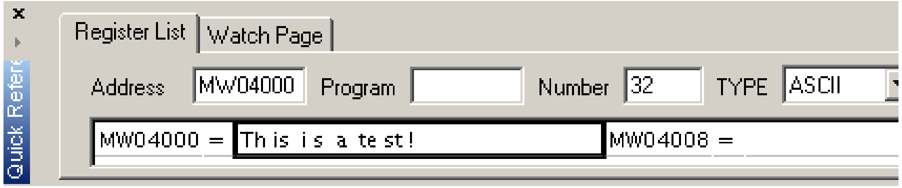

FUNC

Name ASCIIOUT

EXECUTE	? IB04100	BUSY	? DB000000
CRLF	? IB04101	COMPLETE	? DB000001
START	04000 04000	ERROR	? DB000002
SIZE	00016 00016	ERROR-ID	? DW00001
CIR	00001 00001		
SLAVE	00001 00001		
DATA23W	? DA00020		

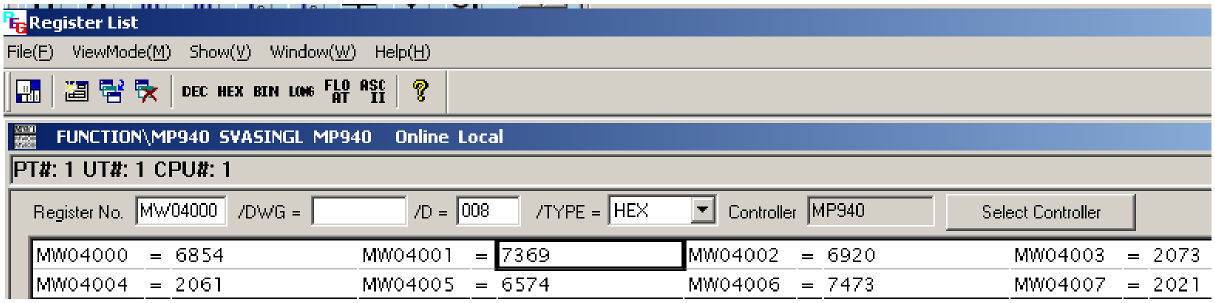
Character Table Setup

The character table is can be set up easily through the ‘quick reference’ window. In this case 16 ASCII characters are typed in at MW04000 as “This is a test! “. The HEX representation can be viewed by ‘register list’ tool, as shown. Note that each character takes up 8 bits which is two HEX digits. Each 16 bit word can hold two characters.



Address MW04000 Program Number 32 TYPE ASCII

MW04000 = This is a test! MW04008 =


Register List
File(F) ViewMode(M) Show(V) Window(W) Help(H)
DEC HEX BIN LONG FLOAT ASCII
Select Controller

FUNCTION\MP940 SVASINGL MP940 Online Local

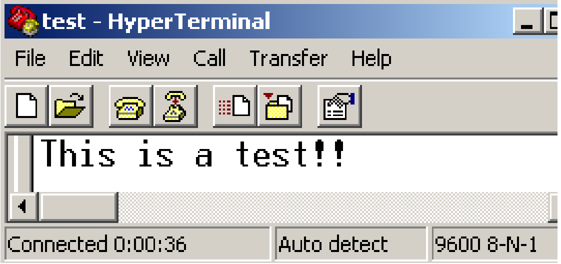
PT#: 1 UT#: 1 CPU#: 1

Register No. MW04000 /DWG = /D = 008 /TYPE = HEX Controller MP940

```
MW04000  = 6854      MW04001  = 7369      MW04002  = 6920      MW04003  = 2073
MW04004  = 2061      MW04005  = 6574      MW04006  = 7473      MW04007  = 2021
```

Application Test Run:

Execution of the function block is started by switching ‘on’ input bit IB00001. The HyperTerminal window shows the text output. Note that the data that is sent to the PC and displayed is exactly what is stored in the ASCII character table.



```
This is a test!!
```

Connected 0:00:36 Auto detect 9600 8-N-1

General use note:

To communicate to any device using ASCII characters, a slave ID and protocol wrapper may need to be developed. This will need to be loaded in the ASCII character register table.

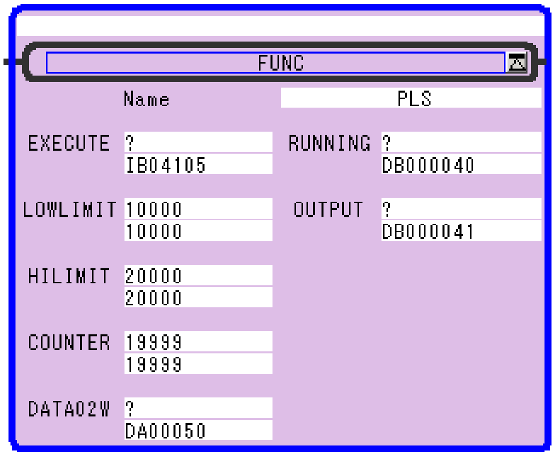
PLS – PROGRAMMABLE LIMIT SWITCH function

Function block for MP2000 series

<PLS> Function Block Summary

The “PLS” block is used to turn a bit on and off based on a range the axis is in.

Functional Block Diagram



FUNC

Name		PLS	
EXECUTE	? IB04105	RUNNING	? DB000040
LOWLIMIT	10000 10000	OUTPUT	? DB000041
HILIMIT	20000 20000		
COUNTER	19999 19999		
DATA02W	? DA00050		

<PLS> Function Block Operation Notes

- To use the function block, the *EXECUTE Input* must be held ON.
- If the *EXECUTE input* goes off during operation, the outputs will go to zero or false.
- The *COUNTER* input can be any long register designated to control the state of the *OUTPUT*.
- If the value of the *COUNTER* input is within the *LOWLIMIT* and the *HILIMIT* then the *OUTPUT* output bit will be True.
- Two words are used as working registers for this function, starting at the address in *Data02W*.

<PLS> Input and Output Register Map

Output Registers

The following registers are used as outputs from the function block. They can be monitored by the LadderWorks program to check the execution of the function.

Output	Type	Description
RUNNING	Bit	Goes true if block is enabled.
OUTPUT	Bit	Goes true when *Counter* input value is within the *LOWLIMIT* and *HILIMIT* inputs and the *EXECUTE* input is True.

Input Registers

The following registers are used as inputs to the function block. They select the options and define the parameters that the user needs to make the Home function work as necessary.

Input	Type	Description	Range and state
EXECUTE	Bit	Block enable – Block cannot execute unless this is TRUE.	True-effective False-invalidity
LOWLIMIT	Long	Defines the output enable window. Minimum value of counter for *OUTPUT* output to go TRUE	-2147483648～2147483647
HILIMIT	Long	Defines the output enable window. Maximum value of counter for *OUTPUT* output to go TRUE	-2147483648～2147483647
COUNTER	Long	The controlling counter for the bit	A long register containing the counter value.
DATA02W	Address	Address of the first working register.	Two words of register space are used by this function.

<PLS> Working Register

This table outlines the data in the two registers used by the function block. There is not usually any need for the user to access any of these bits directly.

Register No	TYPE	Name	Description
AW00000			
Bit 0	IN	execute	*EXECUTE* input (XB000000)
Bit A	IN	oneshotA	Reserved
Bit B	IN	firstPass	One shot coil from rising edge of *EXECUTE* input to initiate block.
AW00002	Working	Revision	Revision Level of the function block.