

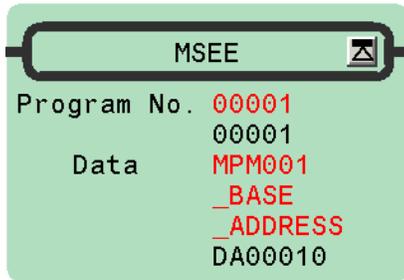
MOTION PROGRAM CALL Instruction (MSEE)

Outline:

MSEE instruction is used to start, stop, control and monitor a Motion Program. This instruction can be used only in a High Speed (DWG.H). It is not possible to use this instruction from DWG A, DWG I or DWG L.

Function:

The MSEE instruction is used to start, stop, control and monitor the state of a MOTION PROGRAM. A Motion program is a textual program that utilized motion language, separate from the ladder programs. A Maximum 256 motion program can be created.



Symbol: MSEE

Full Name: Motion Program Call

Control Category: Motion / Program Flow Control

ICON: 

Parameters:

Parameter Name	Setting
Program No. (Motion Program No.)	<ul style="list-style-type: none"> • Direct specification: Numerical value of 1-256 • Indirect specification: Use of integer type Register
Dest (Work Register)	<ul style="list-style-type: none"> • Register address (except for # and C registers)

The control is done through the bits and words of a specified area of memory called work registers. A minimum of 2 words and a maximum of 4 are used in specified work table. The number of work registers depends on the controller type: MP900 series uses 2 words, MP2000 series uses 4 words. The head (or start) location is defined in the DATA field of the Instruction. The register type used must be read write only.

Work Table Definition

ADR	Type	Symbol	Name	Specification	I/O
0	W	RLY -OUT	Status	Relay Output (contacts in ladder code) ^{*1}	OUT
1	W	RLY-IN	Control (Command)	Relay Inputs (coils in ladder code) ^{*2}	IN
2	W	INTOVR	Interpolation Speed Override	When enabled, percentage of trajectory speed can be adjusted of any interpolated move executing in the Motion Program. The units are in hundreds of a percent (0.01%.) The value is from 0 to 32767.	IN
3	W	WKNUMB	System Work Number	When enabled by address + 1, bit D it is possible for the application programmer to set the location of the work registers. The value is from 1 to 16. If a value of 0 or anything grater than 16 is specified, then "No System Work Error" bit (address + 1, bit E) will turn OFF.	IN

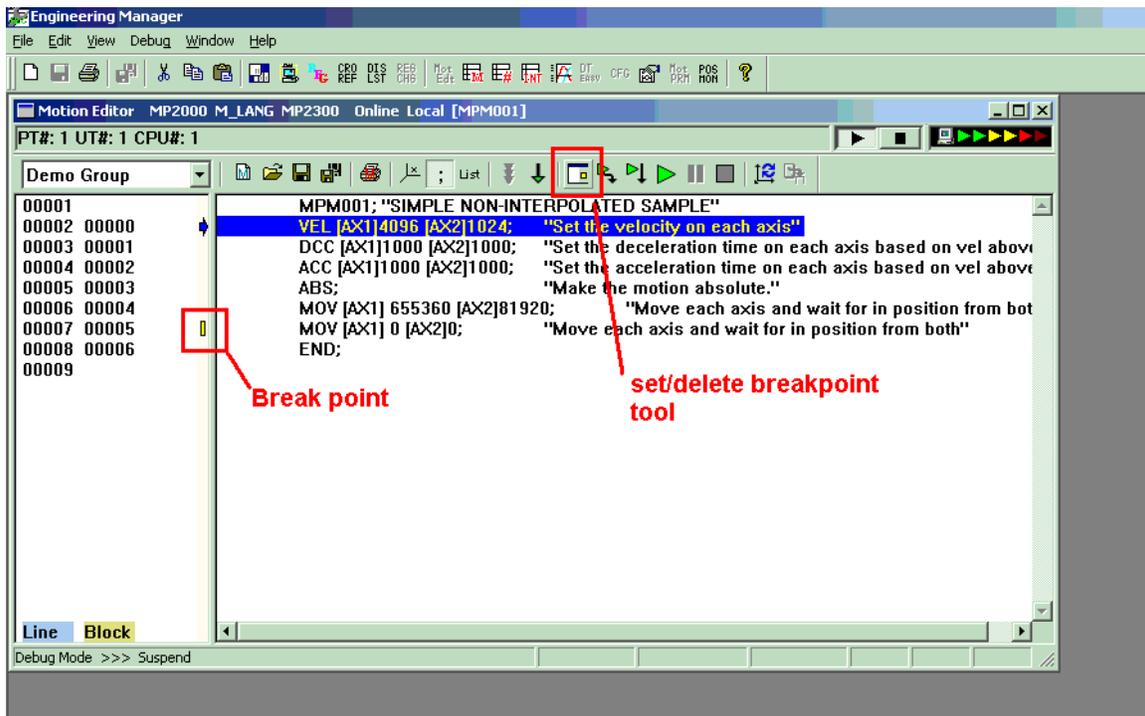
*1: Relay Status Bit Assignment

Bit	Symbol	Name	Specification
0	MPM_RunFB	Program Running	When a motion program is "running" (or executing) this status bit status will turn ON. For the case where a program is running, the speed for any (or all) axis is (are) set to zero so no motion is occurring, this bit will still be ON. There are two types of program execution, "Continuous" and "Single Block Mode." This bit is ON under either execution modes.
1	MPM_PauseFB	Program Paused	When a motion program is "paused" (or held) this status bit will turn ON. The <i>Program Running status bit</i> (Address+0 bit 0) will also be on under a <i>Program Pause</i> condition. A program is paused (held) using the <i>Program Pause Request bit</i> (Address+1, bit 1).
2	MPM_HaltedFB	Program Halted	This bit turns ON when a motion program execution is forced to a permanent stop using the motion <i>Program Stop Request bit</i> (Address+1 bit 2.) This bit will stay on until an <i>Alarm Reset Request bit</i> (Address+1, bit 5) is executed. Stopping a motion program executing in "Continuous" mode will cause an alarm. Stopping a motion program executing in "Single Block Mode" will not cause an alarm.
3	Reserved by the system		
4	MPM_BlkStopFB	Program Single Block Stop	This bit turns ON (flags) when a Motion Program has completed the execution of a single block, when using <i>Single Block Execution Mode</i> . This <i>Program Single Block Stop status bit</i> must be used in conjunction with the <i>Program Single Block Mode Selection bit</i> (Address+1, bit 3) and <i>Program Single Block Start Request bit</i> (address+1, bit 4.) It must be understood by the application programmer that this bit will be OFF during block execution and ON only after the block is executed. In some Yaskawa Manuals and documentation <i>Single Block Execution Mode</i> is named "Debug Mode." "Debug Mode" using the ladder to execute block by block and "Single Block Execution Mode" are synonymous.
5~7	Unused		
8	MPM_AlarmFB	Program in Alarm	This bit turns on when there is a motion program alarm. Many conditions cause a Motion program Alarm. The four most common are: 1. Servo enable loss during a move. 2. Motion command code was changed by ladder during a move or was fighting the ladder for control. (ladder always wins) 3. Some command bit (Address+1) was turned on or off out of sequence. 4. A program stop was issued during a <i>Continuous Program Execution</i> .
9	MPM_BrkPtStopFB	Program Stopped at Breakpoint	This bit turns ON only when the program is in Editor Window Screen (EWS) debug and the program pointer has stopped on that block with a breakpoint marker. It turns OFF when the block with that marker starts execution. In this condition the EWS has control of the program and application programmer is controlling the execution of each block. This control is done using menu commands or toolbar buttons on the EWS. (see note below)
A	Unused		
B	MPM_EWSDebugFB	Program is in EWS debug control	This bit turns ON and stays ON while the Motion program being monitored in the Editor Window Screen (EWS) has control of the program for step by step debug. This is a flag to the ladder code that the Editor Window Screen (EWS) has control of the program.
C	Unused		
D	Unused		

E	MPM_NoWorkErrFB	No System Work error	This bit turns ON if the System Work Block is designated and the block is in use when the Motion Program is executed. With the " Enable System work number setting" bit (address +1, bit D) it is possible for the application programmer to assign the system work block number. Otherwise the operating system assigns the work block number.
F	MPM_PrgNumbErrFB	Invalid Program number	This bit turns ON when attempting to start a motion program where the program number is less than 0 or above 256

Note : Program Stopped at Breakpoint (Address +0, bit 9)

The screen shot below shows the set/delete breakpoint tool and a break point on block 5. This is accomplished by simply placing the cursor anywhere on block 5, then toggling the set/delete breakpoint tool. If there is no breakpoint, one will be added if there is a breakpoint it is removed.



*2: Relay Command Bit Assignment

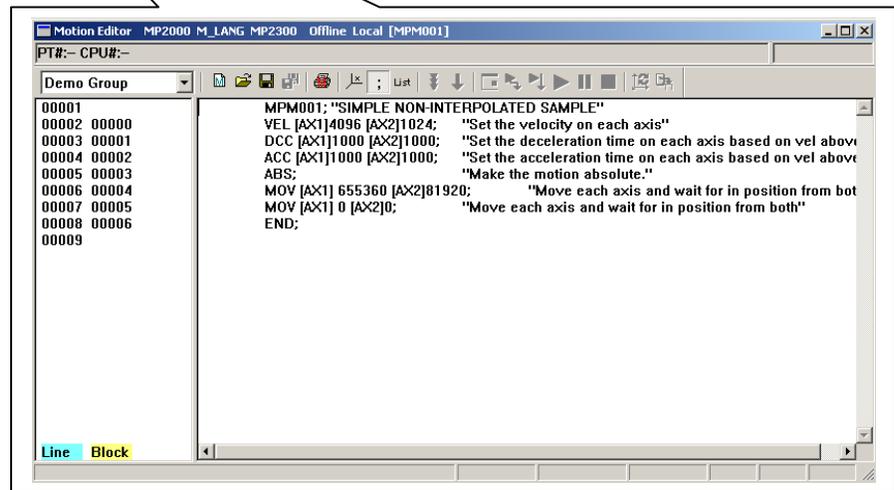
Bit	Symbol	Name	Specification
0	MPM_CStartReq	Program Continuous Start Request	This bit, on a transition from OFF to ON will start a "Continuous" run of the Motion Program. It is not required to remain on for execution to continue. It is recommended (not required) to hold (latch or seal) this bit ON until the <i>Motion Program Running status bit</i> (Address +0 bit 0) transitions ON, then this bit can be turned OFF.
1	MPM_PauseReq	Program Pause Request	This bit will hold program execution (including Motion) while the bit is ON. It is recommended the pause (not required) that the contact or logic to request the pause is interlocked with the <i>Motion Program Running status bit</i> (Address +0 bit 0) if a "Continuous" run of the Motion Program was executed.
2	MPM_StopReq	Program Stop Request	<p>This bit will stop the execution of a Motion Program. In every case the <i>Program Running bit</i> (Address+0, bit 0) will transition OFF and the <i>Program Halted bit</i> (Address+0, bit 2) will transition ON.</p> <p>A <i>Program Alarm</i> (Address+0, bit 8) is generated when this bit stops a Motion Program that was in "Continuous Program Execution." However, when this bit is used to stop the <i>Single Block Mode Execution</i> a <i>Program Alarm</i> (Address+0, bit 8) is NOT generated.</p> <p>If <i>Program Halted bit</i> (Address+0, bit 2) is ON, An Alarm Reset Request (Address +1, bit 5) must be issued to reset the program pointer to the top of the start of the program. This is true if <i>Program Alarm</i> (Address+0, bit 8) exist or not. It is not possible to restart a <i>Single Block Mode Execution</i> from this point. It is required to issue an <i>Alarm Reset Request</i> (Address+1, bit 5) in order to reset the Motion Program execution pointer to the beginning of the program, and allow a new execution of the motion program.</p> <p>It is strongly recommended (and in most situations required) to have the ladder sequence code execute an alarm clear if this bit has stopped a <i>Single Block Mode Execution</i>.</p>

3	MPM_SingleBlkSel	Program Single Block Mode Selection	<p>This bit is used to enable Single block Mode program execution. This bit must remain ON for the <i>Single Block Operation Mode</i>. It is possible to Switch in and out of <i>Single Block Operation Mode</i> by simply enabling or disabling this bit.</p> <p>It is possible to start the execution of a motion Program in <i>Continuous Mode</i> and switch into <i>Single Block Operation Mode</i> on the fly. If the bit turns ON during a "Continuous Operation Mode" program run, the program will hold execution at the end of the block presently being executed (with the exception of the END.) If this bit turns OFF the program mode will switch to "<i>Continuous Mode</i>."</p> <p>In some Yaskawa Manuals and documentation <i>Single Block Execution Mode</i> is named "<i>Debug Mode</i>." "<i>Debug Mode</i>" using the ladder to execute block by block and "<i>Single Block Execution Mode</i>" are synonymous.</p> <p>Because the <i>Single Block Mode</i> is primary for debugging, the <i>Program Start Request bit (address+1, bit 0)</i> should not be allowed to initiate a regular program execution. It is recommended (not required) that interlocks be used to keep the <i>Program Start Request bit (address+1, bit 0)</i> from transitioning ON.</p> <p>This <i>Program Single Block Mode Selection bit</i> is used in conjunction with the <i>Program Single Block Stop status bit (Address+0 bit 4)</i> and <i>Program Single Block Start Request bit (address+1, bit 4)</i>.</p>
---	------------------	--	--

Bit	Symbol	Name	Specification
4	MPM_SingleBlkStartReq	Program Single Block Start Request	This bit is used to request the execution of the next block when using <i>Single Block Execution Mode</i> . With the exception of the Motion Programs first block, the <i>Program Single Block Stop status bit</i> (Address+0 bit 4) will transition ON when the program is ready to receive this request and execute the next block – including the END instruction. This <i>Program Single Block Start Request bit</i> is used in conjunction with the <i>Program Single Block Mode Selection bit</i> (address+1, bit 4.) The end of the block is flagged when the <i>Program Single Block Stop status bit</i> (Address+0 bit 4) turns ON.
5	MPM_AlarmResetReq	Alarm Reset Request	This bit will reset an alarm generated by the motion program. <i>Program Alarm</i> (Address+0, bit 8.) It is also required to reset a <i>Program Halted bit</i> , (Address+0, bit 2.) If an Alarm reset is issued and the <i>Program Alarm</i> (Address+0, bit 8) is cleared, the program pointer is reset to the beginning of the program and it is impossible to restart from the previously stopped point in the program execution. The motion program will now require a <i>Single Block Mode</i> or <i>Continuous Operation Mode</i> Motion Program start. If you want to stop a motion program in <i>Single Block Mode</i> with the ability to restart it, then add sequence logic that switches modes to <i>Continuous Operation Mode</i> , stops the program and back into single after a restart is requested from Inputs or HMI.
6	MPM_ContinueReq	Program Continue From Stop Request	This bit allows a motion program to <u>restart if and only if</u> , the Motion Program was in <i>Continuous Operation Mode</i> and stopped using the <i>Program Stop Request bit</i> (Address+1, bit 2.) Once the Motion Program in <i>Continuous Operation Mode</i> is stopped using <i>Program Stop Request bit</i> (Address+1, bit 2) and the status bits reflect this condition, then the execution can be restarted. It will not be necessary to clear the alarm condition using the <i>Alarm Reset Request bit</i> (Address+1, bit 5) – this restart bit will do it.
7			Unused
8	MPM_Skip1	Skip Signal 1	When the skip signal turns ON during axis movement in a SKP command block, axis movement is decelerated to stop, the remaining travel distance for that block is cancelled, and program execution proceeds to the next block. In this way, the SKP command enables the programming of motion control that can respond to external conditions.
9	MPM_Skip2	Skip Signal 2	
A			Unused
B			Unused
C			Unused
D	MPM_EnaSysWkNo	Enable System work number setting	When this bit is turned ON it is possible for the application programmer to designate the “System Work Number” (address +3) as the work number block.
E	MPM_EnaIntOVR	Enable Interpolation override setting	Enables the use of Interpolation speed override, in Address+2
F			Unused

Program Example:

MSEE	
Program No.	00001
	00001
Data	MPM001
	_BASE
	_ADDRESS
	DA00010



Glossary:

Block

Block is a single line of code in a motion program.

Continuous Program Execution

Continuous Program Execution is started by the OFF to ON transition of the *Start Program Request bit* (Address+1 bit 0.)

Single Block Mode Execution

Single Block Mode Execution is started when the *Program Single Block Mode Selection bit* (Address+1 bit 3) is ON and *Program Single Block Start Request bit* (address+1, bit 4) transitions from OFF to ON.

Editor Window Screen (EWS)

This refers to the motion program editor window in MPE720's Engineering manager.

System Work Number (For Motion Programs)

System work number is the pointer for where information about program execution can be located in the system registers. Since 16 programs can be executed at once, the work numbers are 1 through 16. This information includes the program status, state of the program's control signals, the block number of that program in execution, alarm code, axis position information. If the application programmer desire he/she can assign a work number to a program by tuning ON the "Enable System work number setting" bit (address +1, bit E) and using a value from 1 to 16 in the "System work number" location (address +3). If the "Enable System work number setting" bit (address +1, bit E) is OFF, then the system assigns the work number. *Please refer to section 6.3.3. of MP2300 (siepc88070003) or MP2200 (siepc88070014) user manuals.*