

## **ENGINEERING PUBLICATION**

*MOTION CONTROL DIVISION*

**PRODUCT: MOTION WORKS +**

**SUBJECT: IMPROVEMENTS TO MW+ VERSION 2.83**

**CATEGORY: UPGRADE NOTE**

**ENGINEER: KEVIN HULL**

**DISTRIBUTION: ALL**

---

### **New Features**

- 1) Windows 2000 compatibility.
- 2) Improved messaging and handling of files that may have the archive bit on, or the read only bit set. These bits are typically modified when a ZIP program is used.
- 3) DeviceNet Master can set the DeviceNet update time. Slave communication type can be set to polled or strobed.
- 4) Table data can be viewed in the Data Watch Window.
- 5) The data selected for viewing in the data monitor window is saved with the project details for quick data monitoring later.
- 6) Larger variable and table memory size allocation. The memory size available to the user is approximately 4 times bigger.
- 7) The text box on the toolbar for documenting programs will all multi line comments by using <CTRL> <ENTER> keystrokes.
- 8) Virtual camming. This is accomplished by setting the new system variable sExternal\_Mode to "1." All System variables from the external encoder are used for the virtual encoder, meaning that speed and acceleration values can be set in the units preferred. If the real external encoder is moving when sExternal\_Mode is set, the virtual encoder takes the speed of the real encoder to provide a smooth changeover if camming was in progress. Defining the external encoder position defines both the real and virtual encoder. All system variables that pertain to the external encoder, such as mPosition\_External will report either the real or virtual

information based on the current configuration. Note: the external latch is only valid for the real encoder.

- 9) System Variable units for main and external encoder are now separate throughout the application.
- 10) New system variables have been added to indicate the following:

System variable	Description
mPosition_Ext_Mod_Latch_Counts	The latch of the external encoder in encoder counts and modularized for the machine cycle, if applicable. This value is useful if the floating-point value does not have enough resolution.
mPosition_External_Shifted	Monitor / use the position of the master as interpreted by the camming function. This value is the true external encoder plus the cam shift value modularized by the machine cycle. Handy for using directly in the CAM block to cause the slave to engage or disengage immediately at the current master position.
mPosition_Main_Mod_Latch_Counts	The latch of the external encoder in encoder counts and modularized for the machine cycle, if applicable. This value is useful if the floating-point value does not have enough resolution.
mState_DIP_Switches	Reflects the status of the DIP switches on the front of the MP940. Useful if the programmer wants to alert the operator / technician if the DIP switches are not set correctly.
mWarningOverload	A bit that indicates that the amplifier has an A71 warning. This bit was masked out of the data that causes the IF FAULT block to pass control to the TRUE port, so the IF FAULT block will not detect this condition.
mWarningRegen	A bit that indicates that the amplifier has an A91 warning. This bit was masked out of the data that causes the IF FAULT block

	to pass control to the TRUE port, so the IF FAULT block will not detect this condition.
sAcceleration_Virtual	Acceleration of the virtual encoder in user units/sec/sec.
sCamShift_Mode	Bit setting to configure the operation of the CAM SHIFT block. By default, it is zero, and the block shifts over a specified time. If set to one, the shift occurs over a specified relative change in the master position.
sCamUpdate	A bit used to make the camming function re-determine the last master and slave position in the cam table. Useful if the machine cycle and / or cam table are updated on the fly. Previously, the camming function would only look for the last master / slave pair when camming was engaged.
sExternal_Mode	Set the external encoder to the real encoder, or a virtual encoder. Great for testing, or for profiles that are time based.
sMachineCycle_External	The machine cycle of the Main axis in user units. It is possible to change the machine cycle on the fly, but it will take up to 20 mSec (The low scan rate to actually update the machine cycle.)
sMachineCycle_Main	The machine cycle of the Main axis in user units. It is possible to change the machine cycle on the fly, but it will take up to 20 mSec (The low scan rate to actually update the machine cycle.)
sPosition_Virtual_Counts	View the current counter position of the external encoder.
sSlaveOffset_Mode	Bit setting to configure the operation of the SLAVE OFFSET block. By default, it is zero, and the block shifts over a specified time. If set to one, the shift occurs over a specified relative change in the master position.
sSpeed_Virtual	Speed of the virtual encoder in user

### Important Note:

System Ladder 283 has been modified to provide different behavior concerning the IF FAULT block. Previously, any alarm other than A99 coming from the amplifier would cause the IF FAULT to use the “True” port. Now, the system ladder filters any A9\* warning. An A9F alarm will no longer cause the IF FAULT to use the “True” port. Program modification is required to achieve the same operation. Typically, users have had to check what type of alarm has occurred after the IF FAULT has gone down the “True” path. Usually, just a couple connection lines changes are required.

### Bug Fixes

- 11) Long directory names causing problems when opening project.
- 12) Extremely long programs no longer get abbreviated when converted to ladder language.
- 13) Increased Gear Ratio capability. Previously, the gear ratio numerator and denominator were calculated by multiplying three integers and storing the result into a long. This limited the values that were possible.

In 2.80, the calculations were changed to allow a higher value for each of the three parameters.

- 14) Improved Cam Engage/Disengage to eliminate the chance of the slave walking after repetitive disengage events.

Previously, the final scan of camming could result in the master being at or just beyond the specified disengage value because of a disengage window that is used. Based on the move profile in the table, the slave position could be commanded to start moving again (the start of the next cycle) and the slave will be left at the “non-zero” location.

Note: The slave position at engagement is taken as the zero point of the slave profile,

this gives the user the flexibility to position the slave before engagement and not cause the slave to jump when the engage occurs.

In 2.80, on the scan when it is determined that the master is in the disengage window, the master position specified as the disengage position is taken directly from the CAM block and run through the cam look up table to set the slave at the final position.

- 15) Import Subroutine bug fixed.
- 16) Variable / Table address overlap bug fixed. Previously, it was possible to corrupt the address space by originally creating a CDD cam table of a given size, then changing that cam table in Cam Tool to have a larger size, and re-import the table. The new, larger size could occupy the same are as previously existing variables.
- 17) Slave Offset Block logic was improved to force the block to cancel the offset if the STOP block is executed, or the servo goes out of position control mode, meaning if the JOG or TORQUE blocks are executed in another task. In addition, the SLAVE OFFSET block will no longer operate as an independent offset without camming or gearing already engaged. In the past, this has caused problems with the slave jumping if a SLAVE OFFSET was in progress during a cam or gear disengage.
- 18) The Cam Shift function will cancel the cam shift if the STOP block is executed while a shift is in progress. The shift will remain at the partial correction. This can be cleared out with sCamShift\_Clear.