PMC
Precision MicroControl Corp.

TechNOTE

Title: MultiFlex ETH 1000 Series Stand-Alone Operation
Products(s): All MultiFlex ETH 1000 Series Ethernet motion controllers
Keywords: MultiFlex, ETH, MFX, Ethernet, non-volatile, macro, stand-alone, MCCL
ID#: TN1073
Date: Feb. 3, 2009

Summary

Users can configure and program PMC's MultiFlex ETH 1000 series Ethernet motion controllers to operate in true stand-alone mode – eliminating the need for a host PC.

More Information

All MultiFlex family controllers can be configured to store and execute user-defined MCCL macros. However the MultiFlex ETH 1000 Series Ethernet controllers also allow users to save command sequences to non-volatile flash memory so that they can be available to the controller and executed automatically on power-up. After defining an MCCL macro sequence and saving it to non-volatile flash memory as **macro 0**, the controller can be configured via a DIP switch (see Figure 1 below) to execute macro 0 automatically during boot-up.

To define the MCCL command sequences (macros) required for stand-alone operation, users can open a Telnet session with the controller via any terminal emulation program or PMC's WinControl program, or from a DOS or Linux command line. MCCL macros are defined using the following syntax at the command prompt (MD = "Macro Define", CMD = "MCCL command"):

```
MDx, [CMD1], [CMD2], …, [CMDn]   x = macro number
```

The total string length of each macro is limited to 255 characters, including the line-feed character. The controller can store up to 2000 macros, numbered 0 -1999.

By default, all macros defined from the ASCII terminal (Telnet) are stored in volatile controller memory (SDRAM) and therefore will be erased when power is removed or the controller is reset. To ensure that the macros remain available even after a reset or re-boot, users can copy all RAM based macros to non-volatile flash memory by entering the following command:

```
MD-1        stores all RAM macros into flash memory
```

The RM command to removes macros from RAM, flash or both, as follows:

```
RM0         resets all macros
RM1         resets all RAM macros
RM2         resets all flash macros
```

Macro numbering is arbitrary with the exception of macro 0; If a user defines a macro 0 and stores it in non-volatile flash memory, the instructions in that macro will be executed immediately after system initialization following a power-on or hardware reset.

Users should limit the tasks defined by macro 0 to only the most critical user-defined system initialization functions. This is because the embedded system initialization task will not complete – and therefore the binary and ascii command interfaces will not be available – until the macro 0 task has returned from execution.

Additionally, it is possible that a defined macro 0 will contain a bad instruction which could cause a lock-up of the firmware and system failure. For example, a user might code into Macro 0 an illegal instruction, such as a direct memory read of an unmapped location in embedded memory space, as follows:

```
RW0
```

It this occurs and the macro is stored in non-volatile memory, the controller will be placed in an infinite loop sequence such as… <reset controller, execute macro 0, illegal instruction call, reset controller, execute macro 0…etc.>. In this case, the user can break out of the infinite loop by disabling macro 0. Macro 0 can is enabled or disabled via a DIP switch (switch bank **SW2** position **5**, shown in Figure 1 below). The factory-default setting for SW2 position 5 is **macro 0 execution disabled** (off) position.

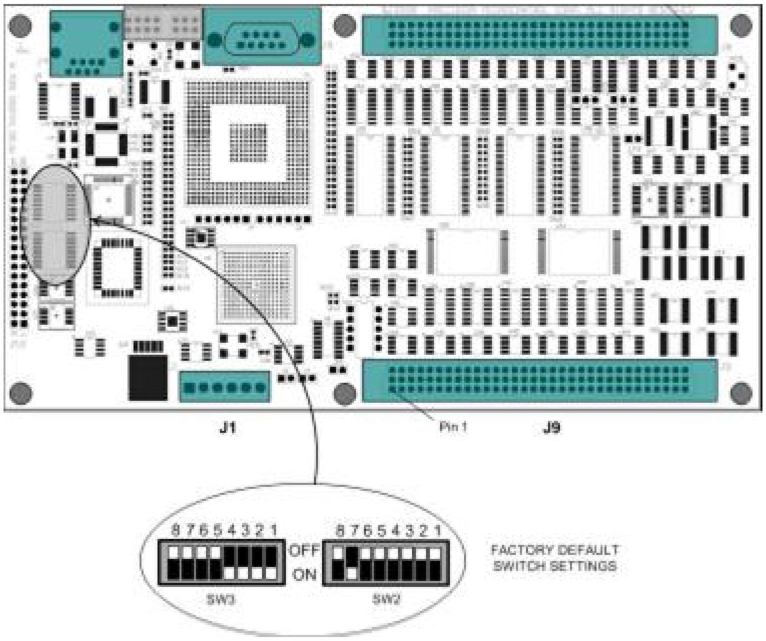

Figure 1. MultiFlex ETH 1000 Series DIP Switches