



Title: Using MCDecodeStatus() to Decode Controller Status Words
Products(s): DC2-PC, DC2-STN, DCX-PC100, DCX-AT100, DCX-AT200
Keywords: MCAPI, status, MCDecodeStatus
ID#: TN1004
Date: June 30, 1998

Summary

All PMC motion controllers maintain status information for each controlled motor in the form of a status word - a collection of yes/no flags. Because the feature set of each controller model is unique to that model, the layout of the status word is also unique to each model. In order to isolate the application program from the controller dependent status word, and to make it easier to isolate individual flags in the status word, the MCDecodeStatus() function was added to the Motion Control API.

More Information

MCDecodeStatus() takes the output of the MCGetStatus() function and a symbolic constant as arguments, and returns the state of the status word flag as TRUE (set) or FALSE (clear). The following code waits for axis 1 to complete its trajectory (the Trajectory Complete flag is set):

```
DWORD dwStatus = MCGetStatus( hCtrlr, 1 );  
while (!MCDecodeStatus( hCtrlr, dwStatus, MC_STAT_TRAJ )) {  
    // do something while trajectory is not complete  
    dwStatus = MCGetStatus( hCtrlr, 1 ); // get a new status word  
}
```

The advantage of using MCDecodeStatus() is that the code snippet above will work for any PMC controller, even though the Trajectory Complete status bit is not always located in the same location within the status word. If you test the status word bit explicitly, say by AND'ing with a numeric constant, you would need to modify the code to work with different controller models.

The MCAPI online help (version 2.0 and later) includes two tables to simplify the selection of symbolic constants for use with MCDecodeStatus(). The Status Word Lookup Table allows you find the proper symbolic constant for a given bit position for any controller (an aid in migrating existing code to MCDecodeStatus()). The Status Word Cross Reference Table shows what status bits are supported by each controller model.