

Dynamische Achspositionierung mit MCStep

Eine allgemeingültige Anleitung zur Nutzung von MCStep, StarStep, ProfiStep und kompatiblen Systemen zur PC-gestützten Achspositionierung

| | |
|---------|-----------------|
| Autor: | Michael W. Folz |
| Stand: | 02.09.2005 |
| Status: | Public Draft |

MCStep und verwandte Systeme sind prinzipiell nicht dafür konzipiert, vollständig über den PC ferngesteuert zu werden. Dennoch hat es in der Vergangenheit immer wieder erfolgreiche Lösungen in diesem Zusammenhang gegeben. Die hier vorgestellte Lösung ist die bislang flexibelste, denn Sie erlaubt das PC-gesteuerte Setzen von Zielposition, Geschwindigkeit und Beschleunigung vor jeder Bewegung.

Systemvoraussetzungen

- MCStep oder kompatibles Steuerungssystem.
- VMC Workbench X2 (oder UniED oder andere kompatible Software).
- C/C++ oder Delphi Programmiersystem (andere Programmiersysteme sind möglich, jedoch nicht direkt vom Hersteller unterstützt).
- Steuerungsdatei „DynAxis.MSS“ als Basis für die Implementation.

Vorbereitungen

- VMC Workbench X2 installieren, einwandfreie Kommunikation zwischen PC und Steuerung muss gewährleistet sein (zumindest die Programme MCSpy und MCStep müssen problemlos online arbeiten).
- Die API-Datei MCStep.H, zu finden unterhalb des Installationsverzeichnis in „.\Win32 API\MCStep“) muss in das C/C++ Projekt eingebunden werden.
- Die API-Datei VMC_DLL.H (zu finden unterhalb des Installationsverzeichnis in „.\Win32 API\VMC Compatibility Layer“) muss in das C/C++ Projekt eingebunden werden.
- In der Datei VMC_DLL.H findet sich ein Verweis auf die Datei MCSave.H. Sollten Sie keine MCSave Lizenz erworben haben, kommentieren Sie diesen Verweis bitte aus.
- Die Datei „DynAxis.MSS“ muss im gleichen Verzeichnis vorhanden sein.

Grundlagen im C/C++ Programm

Um auf die API-Funktionen zuzugreifen, müssen einige Funktionen sowohl aus der MCStep API als auch aus der VMC API geladen werden. Folgende Funktionen werden benötigt:

- MCStep_Online
- MCStep_IsAutomatic
- MCStep_ProgramActiveGet
- MCStep_ProgramWrite
- VMC_SetVariable
- VMC_GetVariable
- VMC_SetMarker
- VMC_GetMarker

Gehen Sie hierzu wie folgt vor:

```
HANDLE hVMC = 0;
HANDLE hMCStep = 0;
hVMC = LoadLibrary("VMC_DLL.DLL");
hMCStep = LoadLibrary("MCStep.DLL");

(PVOID) MCStep_Online = GetProcAddress(hMCStep, "MCStep_Online");
(PVOID) MCStep_IsAutomatic = GetProcAddress(hMCStep, "MCStep_IsAutomatic");
(PVOID) MCStep_ProgramActiveGet = GetProcAddress(hMCStep, "MCStep_ProgramActiveGet");
(PVOID) MCStep_ProgramWrite = GetProcAddress(hMCStep, "MCStep_ProgramWrite");
(PVOID) VMC_SetVariable = GetProcAddress(hVMC, "VMC_SetVariable");
(PVOID) VMC_GetVariable = GetProcAddress(hVMC, "VMC_GetVariable");
(PVOID) VMC_GetMarker = GetProcAddress(hVMC, "VMC_GetMarker");
(PVOID) VMC_SetMarker = GetProcAddress(hVMC, "VMC_SetMarker");
```

Die Fehlerprüfungen haben wir in obigem Beispiel ausgelassen, natürlich sollten Sie in der Praxis das erfolgreiche Laden der DLLs und Funktionen prüfen.

Online-Verbindung und Statustest

Vor jeder Positionierung wird dringend empfohlen, den Status der Verbindung und der Steuerung zu überprüfen. Auch prüfen wir, ob die Steuerung den zuletzt erteilten Fahrauftrag bereits durchgeführt hat. Dies erfolgt mit den Funktionen

- MCStep_Online,
- MCStep_ProgramActiveGet,
- MCStep_IsAutomatic und
- VMC_GetMarker

Im Code sieht das ungefähr so aus:

```
if (MCStep_Online() != 0) {
    // Error: not online, no PLC controller found
    return FALSE;
}
if (MCStep_ProgramActiveGet() != 1) {
    // Error: active (selected) program must be program 1
    return FALSE;
}
if (MCStep_IsAutomatic() != 1) {
    // Error: PLC controller is not in automatic mode
    return FALSE;
}
ULONG dwBitValue = 0;
if (VMC_GetMarker(3101, &dwBitValue, -1) != 0) {
    // Error: problem in communication with PLC controller
    return FALSE;
}
if (dwBitValue) {
    // Error: previous job not finished yet
    return FALSE;
}
```

Wenn Sie bis zu diesem Punkt erfolgreich durchlaufen, können wir nun den Fahrauftrag übertragen.

Fahrauftrag übertragen

Um die größtmögliche Flexibilität zu erhalten, kombinieren wir die Grundfunktionalität des MCStep Systems mit einigen internen Ressourcen. Funktionell gehen wir wie folgt vor:

- Das MCStep Ablaufprogramm DynAxis.MSS wird geladen, die gewünschte Geschwindigkeit und die gewünschte Zielposition werden in die entsprechend vorbereiteten Felder eingetragen.
- Das so modifizierte Ablaufprogramm wird aus dem Speicher des PCs an die Steuerung übertragen. Hierzu wird die Funktion MCStep_ProgramWrite verwendet.
- Die Rampe kann nur über eine interne Funktion verändert werden und wird mit der Funktion VMC_SetVariable in das interne Register 364 geschrieben.
- Mit der Funktion VMC_SetMarker wird das Bitregister 3101 aktiviert. Dieses Bitregister entspricht dem Ausgang 501 innerhalb des MCStep Systems und löst die Positionierung aus.
- Während wir auf das Ende der Positionierung warten (Bitregister 3101 ausgeschaltet), können wir mit VMC_GetVariable aus dem internen Register 171 die aktuelle Position der Achse auslesen.

Im Code sieht dies, vereinfacht und ohne jede Fehlerkontrollen, dann so aus:

```
HANDLE hFile = CreateFile("DynAxis.MSS", GENERIC_READ, FILE_SHARE_READ, NULL,
                        OPEN_EXISTING, FILE_ATTRIBUTE_NORMAL, NULL);
ULONG dwSize = GetFileSize(hFile, NULL);
CHAR szBuffer = new CHAR[dwSize + 8];
ZeroMemory(szBuffer, dwSize + 8);
ULONG dwReadSize = 0;
ReadFile(hFile, szBuffer, dwSize, &dwReadSize, NULL);
CloseFile(hFile);

CHAR szData = new CHAR[dwSize + 8];
ZeroMemory(szData, dwSize + 8);
sprintf(szData, szBuffer, dwMyDesiredSpeed, dwMyDesiredTargetPosition);
MCStep_ProgramWrite(szData);
delete szData;
delete szBuffer;

VMC_SetVariable(364, dwMyDesiredAcceleration, -1);
VMC_SetMarker(3101, TRUE, -1);

ULONG dwStatus = 0;
LONG liCurrentPosition = 0;
do {
    VMC_GetVariable(171, &liCurrentPosition, -1);
    VMC_GetMarker(3101, &dwStatus, -1);
} while (dwStatus);
```

In der Praxis müssen Sie natürlich zwingend alle Funktionen – insbesondere die Funktionen für Datei E/A und Online-Zugriff – auf einen korrekten Rückgabewert überprüfen.