POSITIONIER- UND BAHNSTEUERUNG APCI-8001 und APCI-8008

NCC.DLL

Stand: 19.10.2012, ab Disk V2.53S
Rev. 9/112018

www.addi-data.de

1 Einführung

Der Compiler NCC zur Übersetzung von SAP-Programmen für die Achsensteuerungskarten APCI-8001 und APCI-8008 steht als DLL für Windows-32-Bit-Systeme zur Verfügung. Damit ist es dem Anwender möglich, in eigenen Programmen Quelltext-Files nach rw_SymPas oder G-Code-Files nach DIN 66025 in den Zwischencode (CNC-Binär-Files), der für die zuvor genannten Systeme benötigt wird, zu übersetzen.
Um diese Files ausführen zu können, müssen diese nach den bekannten Methoden an die jeweilige Steuerung übertragen [txbf() / txbf2()] und gestartet [startcnct()] werden.

2 Verwendung von NCC.DLL

Die Library NCC.DLL stellt verschiedene Funktionen bereit, um einen Übersetzungsvorgang zu parametrieren und auszuführen. Für den Übersetzungsvorgang ist die Systemdatei SYSTEM.DAT erforderlich. Standardmäßig wird hier „SYSTEM.DAT“ aus dem Arbeitsverzeichnis verwendet. Ebenso ist das zu übersetzende Quelltext-File erforderlich. Das CNC-Ausgabefile wird normalerweise im gleichen Verzeichnis abgelegt, außer wenn mit der Funktion SetCncFileName() ein anderer Pfad angegeben wird.
Der Übersetzungsvorgang muss stets auf Erfolg abgeprüft werden. Falls ein Fehler zurückgegeben wird, können folgende Fehlerinformationen gelesen werden: Dateiname, Fehlerzeile und Fehlertext. Der Dateiname ist notwendig, da der Fehler auch in einem Include-File aufgetreten sein kann.

3 Lieferumfang

Um die Funktionen von NCC.DLL zu nutzen, werden folgende Softwareteile geliefert:

NCC.DLL	
NCC.H	
NCC.LIB	für diverse C-Compiler
NCC.DLL.pdf	dieses Dokument

4 Funktionen in NCC.DLL

4.1 CncCompile – G-Code-File übersetzen

BESCHREIBUNG:	Mit dieser Funktion wird ein G-Code-File übersetzt. Das erzeugte Binär-File erhält die Extension .CNC und wird im gleichen Verzeichnis wie das Quelltext-File abgelegt.
C:	int CncCompile (char *SrcFileName, int NumberAxis);
PARAMETER:	SrcFileName ist der Name (incl. Laufwerk + Pfad) des Quelltext-Files. In NumberAxis wird die Anzahl der Achsen, für die das Programm übersetzt werden soll, angegeben.
RÜCKGABEWERT:	0 bei Erfolg, Fehler-Nummer, wenn ein Fehler beim Übersetzungsvorgang aufgetreten ist.
ANMERKUNG:	Die Anzahl der tatsächlich vorhandenen Achsen wird bei gebootetem System in der Datenstruktur TOSI bei der Initialisierung mit InitMcuSystem3() zurückgeliefert. In bestimmten Fällen (wenn ein File für unterschiedliche Achskonfigurationen gleich sein soll) kann es notwendig sein, dass ein File für mehr als die vorhandenen Achsen kompiliert werden muss. In diesem Fall kann eine höhere Achsanzahl übergeben werden. Dies entspricht der Option „Full System“ bei den anderen NCC-Versionen. Hierbei sind jedoch zusätzliche Syntaxfehler bezüglich der Achsbenamung möglich. Die Achsnamen werden der Datei SYSTEM.DAT entnommen, und müssen für alle verwendeten Achsen richtig eingetragen sein (mcfg.exe). G-Code-Files werden stets für die TASK 3 übersetzt. Siehe hierzu auch die Funktion DefIpolAxis (Kapitel 4.3)
BEISPIEL:	CompileError = CncCompile („Beispiel.SRC“, 3);

4.2 CncSyntaxCheck – Syntaxüberprüfung bei G-Code-File

BESCHREIBUNG:	Kommando gleich wie CncCompile (), es wird jedoch kein Binär-File erzeugt.
C:	int CncSyntaxCheck (char *SrcFileName, int NumberAxis);
PARAMETER:	SrcFileName ist der Name (incl. Laufwerk und Pfad) des Quelltextfiles. In NumberAxis wird die Anzahl der Achsen, für die das Programm übersetzt werden soll, angegeben.
RÜCKGABEWERT:	0 bei Erfolg, Fehler-Nummer, wenn ein Fehler beim Übersetzungsvorgang aufgetreten ist.

4.3 DefIpolAxis – Default-Interpolationsachsen definieren

BESCHREIBUNG:	Definition der Achsen, die sich im Interpolationszusammenhang befinden; für G-Code Files; kann vor dem erstmaligen Übersetzen einer Datei aufgerufen werden
C:	int DefIpolAxis (unsigned int AxisBits);
PARAMETER:	In Axis Bits sind die Achsen, mit denen interpoliert verfahren werden soll, bitweise codiert.
ANMERKUNG:	Mit diesem Aufruf kann das Übersetzen von G-Code Files parametriert werden [Verwendung des Kommandos CncCompile()]. Wenn dieser Aufruf nicht erfolgt, werden im Online Mode alle vorhandenen Achsen herangezogen. Der Aufruf ist also insbesondere dann notwendig, wenn nicht alle Achsen im System Interpolationsachsen sind. Dadurch ist es möglich, dass z.B. in G01-Kommandos nicht immer alle Achsen angegeben werden müssen. Zur Laufzeit kann dieser Parameter mit G60 umgeschaltet werden.
RÜCKGABEWERT:	immer 0

Beispiel:
Gegeben sei ein kartesisches Koordinatensystem mit X = 1. Achse, Y = 2. Achse und Z = 3. Achse.
Dann muss DefIpolAxis mit dem Wert 7 aufgerufen werden.

4.4 GetErrFileName – Dateiname im Fehlerfall abfragen

BESCHREIBUNG:	Diese Funktion kann im Fehlerfall aufgerufen werden, um den Übersetzungsfehler zu spezifizieren.
C:	void DLLFUNC GetErrFileName (char **Dateiname*); void DLLFUNC GetErrFileNameX (char **Dateiname, int ndx*);
PARAMETER:	Zeiger auf ein char-Array
RÜCKGABEWERT:	Bei GetErrFileName(): keiner Bei GetErrFileNameX():Index des abzufragenden Fehlers
ANMERKUNG:	Die Funktion schreibt den Dateinamen einschließlich Laufwerk und Pfad zum ersten aufgetretenen Fehler in *Dateiname*. *Dateiname* muss auf einen Speicherbereich zeigen, der groß genug ist, den Dateinamen aufzunehmen (max. 260 Zeichen). Bei GetErrFileNameX() kann in ndx der Fehlerindex angegeben werden (siehe Kapitel 4.6).
BEISPIEL:	GetErrFileName (*Dateiname*);

4.5 GetErrLine – Fehlerzeile abfragen

BESCHREIBUNG:	Diese Funktion kann im Fehlerfall aufgerufen werden, um die Zeile zu ermitteln, in der ein Übersetzungsfehler aufgetreten ist.
C:	int GetErrLine (void); int GetErrLineX (int ndx);
PARAMETER:	Bei GetErrLine(): keiner Bei GetErrLineX(): Index des abzufragenden Fehlers
RÜCKGABEWERT:	Zeilennummer, in welcher der Erste (GetErrLine) bzw. der adressierte (GetErrLine X) Übersetzungsfehler erkannt wurde.
ANMERKUNG:	Die Zeilennummer bezieht sich auf die Datei, deren Name mit GetErrFileName() bzw. GetErrFileNameX() zurückgeliefert wird.
BEISPIEL:	Fehlerzeile = GetErrLine ();

4.6 GetErrNum – Anzahl der aufgetretenen Compiler-Fehler abfragen

BESCHREIBUNG:	Diese Funktion kann im Fehlerfall aufgerufen werden, um die Anzahl der aufgetretenen Compiler-Fehler abzufragen. In bestimmten Fällen ist es möglich, dass mehr als ein Fehler auftrat, bevor der Übersetzungsvorgang beendet wurde.
C:	int GetErrNum (void);
PARAMETER:	keiner
RÜCKGABEWERT:	Anzahl der aufgetretenen Übersetzungsfehler des letzten Compiler-Laufs
ANMERKUNG:	Um Fehlerinformationen indiziert auslesen zu können, existieren Funktionen mit einem angehängten „X“. In den meisten Fällen wird der Übersetzungsvorgang nach dem ersten Fehler beendet.
BEISPIEL:	NumErrors = GetErrLine ();

4.7 GetErrText – Fehlertext abfragen

BESCHREIBUNG:	Diese Funktion kann im Fehlerfall aufgerufen werden, um den Übersetzungsfehler zu spezifizieren.
C:	void GetErrText (char * *Fehlertext*); void GetErrTextX (char * *Fehlertext, int ndx*);
PARAMETER:	Zeiger auf ein char-Array
RÜCKGABEWERT:	Bei GetErrText(): keiner Bei GetErrTextX():Index des abzufragenden Fehlers
ANMERKUNG:	Die Funktion schreibt einen Kommentar zum ersten aufgetretenen Fehler in *Fehlertext*. *Fehlertext* muss auf einen Speicherbereich zeigen, der groß genug ist, die Fehlermeldung aufzunehmen (max. 256 Zeichen).
BEISPIEL:	GetErrText (*Fehlertext*);

4.8 GetSymAxisName – Symbolischen Achsnamen ermitteln

BESCHREIBUNG:	Mit dieser Funktion kann der symbolische Achsname einer Achse gelesen werden.
C:	int GetSymAxisName (int an, char *SymAxisName);
PARAMETER:	*an* ist der Index der zu lesenden Achse. In *SymAxisName* wird der aus der Systemdatei SYSTEM.DAT gelesene symbolische Name als nullterminierter String zurückgegeben.
RÜCKGABEWERT:	der angegebene Achs-Index bei Erfolg, -1 bei Misserfolg
ANMERKUNG:	Der Speicherbereich in *SymAxisName* muss groß genug sein, um den Achsnamen einschließlich dem abschließenden Nullzeichen aufnehmen zu können. Die maximale Größe ist 16 Byte.
BEISPIEL:	char AxisName[16]; rvalue = GetSymAxisName (0, AxisName);

4.9 SapCompile – rw_SymPas-File übersetzen

BESCHREIBUNG:	Mit dieser Funktion wird ein rw_SymPas-File übersetzt. Das erzeugte Binär-File erhält die Extension .CNC und wird im gleichen Verzeichnis wie das Quelltext-File abgelegt.
C:	int SapCompile (char *SrcFileName, int NumberAxis);
PARAMETER:	SrcFileName ist der Name (einschließlich Laufwerk und Pfad) des Quelltext-Files. In NumberAxis wird die Anzahl der Achsen, für die das Programm übersetzt werden soll, angegeben.
RÜCKGABEWERT:	0 bei Erfolg, Fehler-Nummer, wenn ein Fehler beim Übersetzungsvorgang aufgetreten ist.
ANMERKUNG:	Die Anzahl der tatsächlich vorhandenen Achsen wird bei gebootetem System in der Datenstruktur TOSI bei der Initialisierung mit InitMcuSystem3() zurückgeliefert. In bestimmten Fällen (wenn ein File für unterschiedliche Achskonfigurationen gleich sein soll) kann es notwendig sein, dass ein File für mehr als die vorhandenen Achsen kompiliert werden muss. In diesem Fall kann eine höhere Achsanzahl übergeben werden. Dies entspricht der Option „Full System“ bei den anderen NCC-Versionen. Hierbei sind jedoch zusätzliche Syntaxfehler bezüglich der Achsbenamung möglich. Die Achsnamen werden der Datei SYSTEM.DAT entnommen und müssen für alle verwendeten Achsen richtig eingetragen sein (mcfg.exe).
BEISPIEL:	CompileError = SapCompile („Beispiel.SRC“, 3);

4.10 SapSyntaxCheck – Syntaxüberprüfung bei rw_SymPas-File

BESCHREIBUNG:	Kommando gleich wie SapCompile(), es wird jedoch kein Binär-File erzeugt.
C:	int SapSyntaxCheck (char *SrcFileName, int NumberAxis);
PARAMETER:	SrcFileName ist der Name (incl. Laufwerk und Pfad) des Quelltextfiles. In NumberAxis wird die Anzahl der Achsen, für die das Programm übersetzt werden soll, angegeben.
RÜCKGABEWERT:	0 bei Erfolg, Fehler-Nummer, wenn ein Fehler beim Übersetzungsvorgang aufgetreten ist.

4.11 SetCaseInsensitive – Unterscheidung Groß-/Kleinschreibung ein/aus

IS ONBESCHREIBUNG:	Mit Hilfe dieser Funktion kann zur Übersetzung von G-Code-Files die Unterscheidung zwischen Groß-/Kleinschreibung deaktiviert werden.
C:	void DLLFUNC SetCaseInsensitive (int CaseInsensitive);
PARAMETER:	0 oder 1 (Default ist 0)
RÜCKGABEWERT:	keiner

4.12 SetCncFileName

BESCHREIBUNG:	Ab Version V2.5.3.58 kann mit Hilfe dieser Funktion ein Dateiname für das CNC-Ausgabefile inklusive Pfad und Laufwerksangabe übergeben werden.
C:	void SetCncFileName (char *str);
PARAMETER:	Dateiname mit optionaler Laufwerks- und Pfadangabe
RÜCKGABEWERT:	0 bei Erfolg, 1 bei Misserfolg
ANMERKUNG:	Wenn diese Funktion mit einem Leerstring aufgerufen wird, kann ein zuvor zugewiesener Dateiname wieder deaktiviert werden. In diesem Fall werden der Dateiname und der Ausgabepfad aus dem Namen der Quelltextdatei ermittelt.
BEISPIEL:	SetCncFileName (“C:\Cnc\Userfile.cnc”);

4.13 SetPostIncFileName

BESCHREIBUNG:	Ab Version V2.5.3.57 kann mit Hilfe dieser Funktion ein Dateiname übergeben werden. Die entsprechende Datei wird dann als Include-Datei am Ende der Quelltext-Datei eingebunden, ohne die Anweisung $I. Diese Include-Datei kann Deklarationen von Unterprogrammen enthalten, die im Programm verwendet werden können, die aber für den Maschinenbediener nicht sichtbar sind. Diese Funktionalität ist nur bei Programmen nach DIN 66025 (G-Code-Programme) möglich. Wenn kein Pfad angegeben ist, wird der Pfad der zu übersetzenden Quelltextdatei verwendet.
C:	void SetPostIncFileName (char *str);
PARAMETER:	Dateiname mit optionaler Laufwerks- und Pfadangabe
RÜCKGABEWERT:	0 bei Erfolg, 1 bei Misserfolg
ANMERKUNG:	Wenn diese Funktion mit einem Leerstring aufgerufen wird, kann eine zuvor zugewiesene Datei wieder deaktiviert werden.
BEISPIEL:	SetPostIncFileName (“C:\Temp\Subroutines.inc”);

4.14 SetPreIncFileName

BESCHREIBUNG:	Ab Version V2.5.3.57 kann mit Hilfe dieser Funktion ein Dateiname übergeben werden. Die entsprechende Datei wird dann als Include-Datei am Anfang der Quelltext-Datei eingebunden, ohne die Anweisung $I. Diese Include-Datei kann Variablen- und Prozedur-Deklarationen enthalten, die im Programm verwendet werden können, die aber für den Maschinenbediener nicht sichtbar sind. Diese Funktionalität ist nur bei Programmen nach DIN 66025 (G-Code-Programme) möglich. Wenn kein Pfad angegeben ist, wird der Pfad der zu übersetzenden Quelltextdatei verwendet.
C:	void SetPreIncFileName (char *str);
PARAMETER:	Dateiname mit optionaler Laufwerks- und Pfadangabe
RÜCKGABEWERT:	0 bei Erfolg, 1 bei Misserfolg
ANMERKUNG:	Wenn diese Funktion mit einem Leerstring aufgerufen wird, kann eine zuvor zugewiesene Datei wieder deaktiviert werden.
BEISPIEL:	SetPreIncFileName (“C:\Temp\Declaration.inc”);

4.15 SetSystemDatName – Systemdatei angeben

BESCHREIBUNG:	Mit dieser Funktion kann die Systemdatei (Laufwerk, Pfad, Dateiname) angegeben werden, mit welcher der Übersetzungsvorgang durchgeführt wird.
C:	void SetSystemDatName (char *str);
PARAMETER:	Dateiname mit optionaler Laufwerks- und Pfadangabe
RÜCKGABEWERT:	keiner
ANMERKUNG:	Wenn diese Funktion nicht aufgerufen wird, ist die Datei „SYSTEM.DAT“ im Arbeitsverzeichnis angewählt.
BEISPIEL:	SetSystemDatName (“C:\Temp\System.dat”);

4.16 SetTaskNum – Task-Nummer angeben

BESCHREIBUNG:	Mit dieser Funktion kann die Task-Nummer, für welche das Quelltext-File übersetzt wird, angegeben werden. Dies ist nur sinnvoll bei der Übersetzung von rw_SymPas-Dateien, da G-Code-Files immer für Task 3 übersetzt werden.
C:	void SetTaskNum (int TaskNr);
PARAMETER:	Task Nummer (0..3), für welche das NC-File erstellt werden soll. Nach dem Laden kann ein entsprechend generiertes File nur in der entsprechenden Task ausgeführt werden.
RÜCKGABEWERT:	keiner
ANMERKUNG:	Falls im Quelltext-File die Task ausgewählt wird mit {$TASK ?}, ist diese Anweisung hinfällig. Default-Wert ist 0.
BEISPIEL:	

4.17 SetTrac – Default-Bahnbeschleunigung setzen

BESCHREIBUNG:	Mit dieser Funktion wird die Default-Bahnbeschleunigung gesetzt.
C:	void SetTRAC (double trac);
PARAMETER:	Beschleunigungswert in den aktuell gesetzten Interpolationseinheiten
RÜCKGABEWERT:	keiner
ANMERKUNG:	siehe auch SetTrvl und SetUnits

4.18 SetTrvl – Default-Bahngeschwindigkeit setzen

BESCHREIBUNG:	Mit dieser Funktion wird die Default-Bahngeschwindigkeit gesetzt.
C:	void SetTRVL (double trvl);
PARAMETER:	Geschwindigkeitswert in den aktuell gesetzten Interpolationseinheiten
RÜCKGABEWERT:	keiner
ANMERKUNG:	siehe auch SetTrac und SetUnits

4.19 SetUnits – Default-Weg- und Zeiteinheit setzen

BESCHREIBUNG:	Mit dieser Funktion können die Default-Interpolationseinheiten definiert werden. Ohne diesen Aufruf sind die Positionseinheit „mm“ und die Zeiteinheit „Sekunde“ angewählt.
C:	void SetUnits (int pu, int tu);
PARAMETER:	Positionseinheit „pu“ und Zeiteinheit „tu“
RÜCKGABEWERT:	keiner
ANMERKUNG:	Siehe hierzu Programmierhandbuch (PHB), Kapitel „ctru, change trajectory units“
BEISPIEL:	SetUnits (0, 1); // Einheiten „mm“ und „min“ auswählen

4.20 SetUserSpecCode – Anwenderspezifischen Code setzen

BESCHREIBUNG:	Mit Hilfe dieser Funktion kann zur Übersetzung ein anwenderspezifischer Code programmiert werden. Mit Hilfe dieses Codes lassen sich anwenderspezifische Funktionen aktivieren.
C:	void DLLFUNC SetUserSpecCode (unsigned UserSpecCode);
PARAMETER:	Projektabhängige Ganzzahlvariable
RÜCKGABEWERT:	keiner

4.21 UseLineNumbers – Zeilennummerierung ein/aus

BESCHREIBUNG:	Mit Hilfe dieser Funktion kann zur Übersetzung von G-Code-Files die Notwendigkeit von Zeilennummern (Nxxx) deaktiviert werden.
C:	void DLLFUNC UseLineNumbers (int UseLineNum);
PARAMETER:	0 oder 1
RÜCKGABEWERT:	keiner