POSITIONIER- UND BAHNSTEUERUNG APCI-8001 UND APCI-8008

PROGRAMMIER- UND REFERENZ-HANDBUCH FÜR LINUX

PHB Linux

Stand: 09.06.2017, Revision V2.53JK
Rev. 6/112018

www.addi-data.de

1 Einführung

Unter Linux wird heute ein freies Betriebssystem für Computer verstanden. Es ist unter den Bedingungen der GNU General Public License für die Allgemeinheit freigegeben. Jeder darf es verwenden, kopieren, weitergeben und verändern. Die Quelltexte sind frei verfügbar (siehe auch Open Source). Linux stellt unter anderem eine Alternative zum proprietären Microsoft Windows und den kommerziellen UNIX-Systemen dar.
Seit geraumer Zeit wird der Einsatz des Linux-Betriebssystems im industriellen Umfeld immer häufiger gewünscht und gefordert. Diesem Trend wird jetzt auch durch Linux-Software-Unterstützung für die APCI-8001 und APCI-8008 (= APCI-800x) Rechnung getragen. Über die vielen Vorteile dieses sehr stabilen, frei erhältlichen und ausgereiften Betriebssystems gibt es unzählige Berichte und Erfolgsgeschichten, auf die hier jedoch nicht näher eingegangen wird.
Das vorliegende Dokument PHB-Linux (Programmierhandbuch für Linux) wurde für Programmentwickler geschrieben. Es beinhaltet die notwendigen Schritte zur Inbetriebnahme und Konfiguration der TOOLSET-Software xPCI-800x für Linux und komplettiert die Programmierung mit entsprechenden Beispielen.
Der interessierte Anwender oder Programmierer sollte grundlegende Kenntnisse über das Linux-Betriebssystem besitzen und über Kenntnisse in der Programmiersprache C verfügen. Voraussetzung für die Inbetriebnahme sind ebenfalls Administrationsrechte zur Konfiguration des Linux-Kernels und zur Installation von notwendigen Kernel-Quellen und Kernel-Modultreibern.

2 Dokumentation

Die zur Steuerungsfamilie APCI-800x gehörenden Dokumente BHB (Bedienungshandbuch), IHB Inbetriebnahme-Handbuch) und PHB (Programmierhandbuch) behalten nach wie vor ihre Gültigkeit und sollten für die Inbetriebnahme und Programmentwicklung herangezogen werden.
Der Linux-API-Befehlssatz (Application Programming Interface) ist bis auf die Ausnahme eines zusätzlich bei allen Funktionen vorangestellten Handle-Parameters [Kapitel 7.1] identisch zu dem xPCI-800x-Windows-API-Befehlssatz. Somit behält das Programmierhandbuch (PHB) auch Gültigkeit für die Linux-Programmierumgebung. Diese Tatsache ist insbesondere für Anwender interessant, welche schon Erfahrung mit den xPCI-800x-Controllern unter dem Betriebssystem Windows gemacht haben.

3 Lieferumfang der xPCI-800x-TOOLSET-Software für Linux

Der Lieferumfang der xPCI-800x-TOOLSET-Software beinhaltet mehrere Archive, welche wiederum in verschiedene Teile gegliedert sind. Dabei handelt es sich um Dateien mit den Namen:

- mcug3-linux-i686-2012-08-08.tar.gz (Beschreibung in Kapitel 4 und 5)
- mcug3-linux-mipsel-2012-08-08.tar.gz (Beschreibung in Kapitel 4 und 5)
- mcug3-windows-2012-08-08.tar.gz (Beschreibung in Kapitel 6)

Das Datum in den Dateinamen repräsentiert den entsprechenden Entwicklungsstand. Grundsätzlich besteht das Archiv mcug3-linux-{i686, mipsel} aus den Treiberquellen, Bibliotheken, gsoap-basierter Webserver und Beispielprogrammen. Das Archiv mcug3-windows beinhaltet das Inbetriebnahme- und Konfigurationsprogramm mcfg.exe einschließlich Treiber-DLL für Webservices. Zunächst sollten die Archive auf das Zielsystem kopiert und in einem beliebigen Arbeitsverzeichnis ausgepackt werden:

- *tar -xzf mcug3-linux-i686-2012-08-08.tar.gz (Zielsystem PC)*
- *tar -xzf mcug3-linux-mipsel-2012-08-08.tar.gz (Zielsystem CANTastic, APCI-6000, MSX-Box, APCI-8008)*
- *tar -xzf mcug3-windows-2012-08-08.tar.gz*

Die verschiedenen Bestandteile dieser soeben ausgepackten Archive werden im weiteren Verlauf dieses Dokuments erläutert.

4 Kernel-Modul mcug3.[o,ko]

Ein Kernel-Modul (auch LKM für engl. Loadable Kernel Module) ist ein spezieller Programmcode, der im laufenden Betrieb in den Kernel eines Betriebssystems, hier Linux, geladen oder aus diesem wieder entfernt werden kann. Häufig finden Kernel-Module für Gerätetreiber Verwendung, da eine große Auswahl an Kernel-Modulen für die unterschiedlichsten Hardware-Komponenten mit dem Betriebssystem mitgeliefert werden können, aber nur die wirklich benötigten Treiber in den Speicher geladen werden müssen.
Das Verfahren des dynamischen Hinzufügens von Kernel-Modulen wird beim Linux-Kernel dazu verwendet, um einen Standard-Kernel an die Hardware, auf der er betrieben wird, dynamisch anzupassen. Beispielsweise kann der Treiber einer vorgefundenen Motion-Control-Karte geladen werden, während die vorliegenden Treiber für nicht vorhandene Hardware ignoriert werden können und somit auch keinen Hauptspeicher belegen. Ein weiterer Vorteil liegt darin, dass Erweiterungen für den Kernel integriert werden können, ohne dass das Betriebssystem neu gestartet werden muss.

Der unter GPL (General Public License) stehende LKM-Treiber *mcug3.[o, ko]* arbeitet mit allen APCI-800x-Geräten (APCI-8001 und APCI-8008) und umfasst die Funktionen für die Initialisierung, Parametrierung, Datenaustausch, Kommandovorgaben und Übertragung von Statusinformationen. Die Treiberquellen befinden sich im Unterverzeichnis mcug3lkm des oben genannten Archivs.

4.1 Erstellen des Kernel-Moduls mcug3.[o, ko]

Die verfügbaren Kernel-Modulquellen wurden für das Linux-Betriebssystem mit Kernel-Version 2.4, 2.6 und 3.x erstellt. Da es keine generelle verfügbare Variante des Linux-Betriebssystems gibt, d.h. jeder Anwender kann eine andere Version des Kernels in seiner Linux-Umgebung benutzen, muss der Kernel-Modultreiber mcug3.[o, ko] für das jeweilige Zielsystem erstellt werden. Voraussetzung hiefür wiederum ist, dass die jeweiligen Kernel-Quellen für den Übersetzungsvorgang ebenfalls vorhanden sein müssen. Wie und von wo die Kernel-Quellen installiert werden können, sollte aus der Dokumentation der jeweiligen Linux-Distribution entnommen werden.
Bevor der Make-Prozess zur Erstellung des mcug3.[o, ko] Treibers gestartet werden kann, sind evtl. noch wenige Anpassungen in der Make-Datei des Verzeichnisses mcug3lkm vorzunehmen. Hierzu könnten bei Kernel-Version 2.4 evtl. die Macros KERNELDIR und INSTALLDIR gehören. Bei den Kernel-Versionen 2.6 hingegen kann gegebenenfalls das Macro KDIR angepasst werden. Sofern alle Anpassungen vorgenommen wurden, kann jetzt der Make-Prozess durch Aufruf des Make-Kommandos im Unterverzeichnis mcug3lkm gestartet werden. Dies wird vorzugsweise in einer Konsole des Entwicklungsrechners durchgeführt. Bei fehlerfreier Ausführung des Make-Prozesses sollte am Ende der Treiber mcug3.o (Linux-2.4) bzw. mcug3.ko (Linux-2.6, Linux-3.x) im Unterverzeichnis mcug3lkm vorhanden sein.

4.2 Installieren des Kernel-Moduls mcug3.[o, ko]

Sofern der Kernel-Modultreiber wie oben beschrieben erfolgreich erstellt wurde, kann dieser nun in das Archiv der Modultreiber mitaufgenommen werden. Dies erfolgt durch Ausführen des Befehls *make install*. Für diesen Vorgang benötigen Sie jedoch Administrator-Rechte. Zur Überprüfung des Modul-Treibers kann dieser jetzt geladen werden. Für den Funktionstest muss mindestens ein xPCI-800x-Controller im Zielsystem installiert sein. Der Modul-Ladevorgang wird mit dem Befehl *modprobe mcug3* ausgelöst. Zur ersten Überprüfung des Ladevorgangs kann der Befehl *cat /proc/mcug3* ausgeführt werden. Dieser sollte in etwa folgende Bildschirmmeldung ausgeben:

```
ADDI-DATA GmbH  mcug3 LKM driver, version: 1.0.1.
ADDI-DATA APCI-8008 High Performance Motion Controller at
0xf75a6880 (dev 16)
PCI Registers:
0x00: 0x0180102f 0x02b00017 0x05800020 0x00004008
0x10: 0x00000000 0x00000000 0xfc000008 0x00000000
0x20: 0xfea00000 0x0000d801 0x00000000 0x04015257
0x30: 0x00000000 0x000000dc 0x00000000 0x0a020103
```

4.3 Laden und Entladen des Kerneltreibers mcug3.[o,ko]

Das Laden des oben erstellten Kernel-Moduls kann entweder mit dem Befehl insmod ./mcug3.[o,ko] oder modprobe mcug3 jederzeit manuell durchgeführt werden. Das Entladen hingegen wird mit der Anweisung rmmod mcug3 veranlasst.

4.3.1 Besonderheiten Linux Kernel 2.4

Zum Laden des Kernel-Moduls ist noch ein zweites Programm mit dem Namen *mcug3.sh* im Unterverzeichnis „scripts“ der Linux TOOLSET-Software enthalten. Dieses Shellscript muss vermutlich noch an Ihre Systemumgebung angepasst werden.
Das Skript dient dazu, das Kernel-Modul mcug3.o zu laden oder zu entladen. Das *mcug3.sh*-Shellscript unterstützt auch den Start des Webservice unter Zuhilfenahme des Init-V-Prozesses. Je nach Distribution gibt es verschiedene Verzeichnisse, und zwar eines für jeden Runlevel. Diese Verzeichnisse stehen entweder direkt in /etc/ (Debian) oder unter /etc/rc.d/ (Mandrake, RedHat) oder unter /sbin/init.d (SuSE) und heißen dem Runlevel entsprechend rc0.d, rc1.d, rc2.d... Das Script kann in das Scriptverzeichnis der entsprechenden Distribution (z.B. /etc/init.d) kopiert werden und es können entsprechende Verknüpfungen in den gewünschten Runlevels erzeugt werden. Dies hat den Vorteil, dass das Laden des Kerneltreibers *mcug3.o* automatisch beim Systemstart des Linux-Betriebssystems gestartet wird.

4.3.2 Besonderheiten Linux Kernel 2.6 und 3.x

Das Laden des Moduls bei Kernel-Version 2.6 und 3.x erfolgt normalerweise automatisch.
Wenn das automatische Laden nicht funktionieren sollte, kann gegebenenfalls das in Kapitel 4.3.1 beschriebene mcug3.sh-Script verwendet werden.

Für technisch Interessierte hier eine Kurzbeschreibung für das automatische Laden des Kernel-Treibers mcug3.ko:
Als Modul kompilierte Gerätetreiber können Aliasse eingebaut haben, was bei dem mcug3-Treiber der Fall ist. Diese kann man sich mit dem Kommando modinfo ansehen. Sie hängen üblicherweise mit den bus-spezifischen Kennmarken eines vom Treiber unterstützten Geräts zusammen. Beispielsweise unterstützt der Treiber mcug3 unter anderem PCI-Geräte mit der Hersteller-ID 0x102F und Geräte-ID 0x0180.
Der zugehörige Alias lautet „pci:v0000102Fd00000180sv00005257sd00000401bc*sc*i*“. Für die meisten Geräte exportiert der Bus-Treiber den Alias des notwendigen Treibers nach sysfs. So würde beispielsweise die Datei /sys/bus/pci/devices/0000:02:02.0/modalias den Wert „pci:v0000102Fd00000180sv00005257sd00000401bc05sc80i00“ enthalten. Die bei den gängigen Linux-Distributionen installierten Udev-Regeln sorgen dafür, dass udevd /sbin/modprobe mit dem Inhalt der uevent-Umgebungsvariable MODALIAS aufruft (sie sollte das gleiche enthalten wie die Datei modalias in sysfs). Dadurch werden alle Module aufgerufen, deren Aliasse dem Wert entsprechen.

4.4 Erstellen der Device-Nodes für mcug3.[o, ko]

Bei dem Gerätekonzept unter Linux werden die Geräte oder Gerätegruppen genauso behandelt wie Dateien oder Dateiordner. Ein Gerät (z.B. ein Diskettenlaufwerk) oder eine Gerätegruppe (z.B. xPCI-800x-Controller) erscheint z.B. als Dateiordner, in dem sich verschiedene Geräte als Dateien befinden können. Ein Drucker, ein Bildschirm, eine Maus etc. erscheinen genauso nur als Datei oder Dateiordner. Im Unterschied zu anderen („normalen“) Dateien können Gerätedateien nicht einfach gelöscht oder neu angelegt werden. Geräte werden als normale Dateien angesprochen und befinden sich nach dem Wurzelverzeichnis im Verzeichnis /dev (vom engl. „Device“). Der Name wird als Abkürzung der englischen Bezeichnung des Geräts dargestellt. Gerätedateien werden unter Linux als „special files“ bezeichnet.

4.4.1 Besonderheiten Linux Kernel 2.4

Für die bereits installierten xPCI-800x-Controller müssen entsprechende Gerätedateien erzeugt werden. Dies kann sehr einfach durch Ausführen des Skripts *mcug3.sh* erledigt werden. Das Skript befindet sich im Unterverzeichnis „scripts“. Folgende Anweisung ist am Anfang des Scripts einzufügen: mcug3_cdn=y
Nach Ausführung des Skripts mit der Anweisung ./mcug3.sh start sollten in etwa folgende Dateien im Device-Verzeichnis angelegt worden sein:

```
tux@knoppix:~$ ls /dev/mcu* -l
crw-rw----  1 root staff 253,  0 2005-06-10 13:27 /dev/mcug3c0
crw-rw----  1 root staff 253,  1 2005-06-10 13:27 /dev/mcug3c1
crw-rw----  1 root staff 253,  2 2005-06-10 13:27 /dev/mcug3c2
crw-rw----  1 root staff 253,  3 2005-06-10 13:27 /dev/mcug3c3
crw-rw----  1 root staff 253, 16 2005-06-10 13:27 /dev/mcug3i0
crw-rw----  1 root staff 253, 17 2005-06-10 13:27 /dev/mcug3i1
crw-rw----  1 root staff 253, 18 2005-06-10 13:27 /dev/mcug3i2
crw-rw----  1 root staff 253, 19 2005-06-10 13:27 /dev/mcug3i3
```

Für jeden xPCI-800x-Controller (hier bis zu 4 Karten) werden zwei verschiedene Gerätedateien erstellt, wobei jene vom Typ mcug3cx [x = 0 .. 3] für den Normalbetrieb und die Gerätedateien vom Typ mcug3ix für den Interrupt-Betrieb benötigt werden.

4.4.2 Besonderheiten Linux Kernel 2.6 und 3.x

Das Erstellen der Device-Nodes unter Linux 2.6 und 3.0 erfolgt weitgehend automatisch beim Laden des Kerneltreibers. Zusätzlich sollte noch die udev-Rules-Datei 92-mcug3.rules in das Verzeichnis /etc/udev/rules.d/ kopiert werden. Diese erzeugt zusätzlich zu den bereits bestehenden Device-Nodes symbolische Links.

Folgende Einträge sollten dann im Device-Verzeichnis zu sehen sein:

```
ls /dev/mcug3* -l
      lrwxrwxrwx 1 root root  9 Aug  8 15:07 /dev/mcug3c0 -> mcug3/c/0
      lrwxrwxrwx 1 root root  9 Aug  8 15:07 /dev/mcug3i0 -> mcug3/i/0
ls /dev/mcug3/c/0 -l
      crw-rw---T 1 root plugdev 252, 0 Aug  8 15:07 /dev/mcug3/c/0
ls /dev/mcug3/i/0 -l
      crw-rw---T 1 root plugdev 252, 16 Aug  8 15:07 /dev/mcug3/i/0
```

5 SOAP – Simple Object Access Protocol und mehr

SOAP ist ein Protokoll, mit dessen Hilfe Daten zwischen Systemen ausgetauscht und Remote Procedure Calls durchgeführt werden können. SOAP stützt sich auf die Dienste anderer Standards: XML zur Repräsentation der Daten sowie Internet-Protokolle der Transport- und Anwendungsschicht (vgl. TCP/IP-Referenzmodell) zur Übertragung der Nachrichten. Die gängigste Kombination ist SOAP über HTTP und TCP. Ursprünglich war SOAP die Abkürzung für Simple Object Access Protocol (Einfaches Objekt-Zugriffs-Protokoll), seit Version 1.2 ist SOAP jedoch offiziell keine Abkürzung mehr, da es nicht (nur) dem Zugriff auf Objekte dient.

5.1 Webservice mcug3Xserver – basiert auf gsoap

Für die xPCI-800x-Familie ist ein auf gsoap (http://gsoap2.sourceforge.net) basierter Webservice *mcug3Xserver* ebenfalls im Lieferumfang der TOOLSET-Software enthalten. Dieser Webservice wurde hauptsächlich für die Projektierung und Inbetriebnahme der Steuerung entwickelt und beinhaltet nahezu alle Funktionsschnittstellen zum API der xPCI-800x. Er dient als Schnittstelle für Client-Anwendungen wie z.B. das *mcfg.exe*-Programm [Kapitel 6]. Diese Anwendung wurde für das Windows-Betriebssystem erstellt und hat in der Zwischenzeit einen so großen Funktionsumfang erreicht, der eine Portierung für das Linux-Betriebssystem aus heutiger Sicht ausschließt. Mit dem soeben beschriebenen *mcug3Xserver* kann jedoch die jeweils aktuelle mcfg.exe-Anwendung auch für das Linux-Betriebssystem zum Einsatz kommen. Hierzu gibt es derzeit zwei Varianten.

Variante 1: Sofern ein PC mit i386-kompatibler Systemarchitektur und grafischer Benutzeroberfläche (Graphical User Interface, GUI) zum Einsatz kommt, kann mit Hilfe eines Emulators wie *WINE* [Kapitel 6.1] die für Windows erstellte 32-Bit-Anwendung mcfg.exe direkt auf das Linux-Zielsystem geladen und dort zur Ausführung gebracht werden. Bei dieser Methode kommuniziert die Anwendung mcfg.exe unter Zuhilfenahme einer speziellen DLL-Bibliothek *mcug3X.dll* [Kapitel 6.2] mit dem Webservice mcug3Xserver [Kapitel 5.2], welcher wiederum die Kommunikation mit der xPCI-800x-Steuerung übernimmt. Der Webservice läuft als sogenannter Dämon als Hintergrundprozess im Linux-Betriebssystem und wartet im Ruhezustand auf auszuführende Aktionen der Client-Anwendungen.

Variante 2 ist identisch mit Variante 1, jedoch wird die *mcfg.exe*-Anwendung nicht auf dem Zielsystem, sondern auf einem zweiten PC mit Windows- oder Linux-Betriebssystem zur Ausführung gebracht. Dies wäre z.B. dann der Fall, wenn auf dem Zielsystem kein GUI zur Verfügung steht oder keine auf i386-CPU basierte Architektur vorhanden ist.

5.2 mcug3Xserver – Installation und Konfiguration

Der Webservice *mcug3Xserver* befindet sich im Unterverzeichnis *gsoap* der Linux TOOLSET-Software. Dieses Programm kann in ein beliebiges Verzeichnis des Zielsystems kopiert werden. Zum Start des Webservice ist noch ein zweites Programm mit dem Namen *mcug3Xserver.sh*, im Unterverzeichnis scripts der Linux TOOLSET-Software enthalten. Dieses Shellscript muss vermutlich noch an Ihre Systemumgebung angepasst werden. Insbesondere die Variablen DAEMON_PATH und LOGFILE sollten überprüft werden! Die im Script enthaltene PORT-Variable mit dem Standard-Wert 10001 wird später auch noch für die mcfg.exe-Projektierung [Kapitel 6.3] benötigt.
Das Skript dient dazu, den Webservice zu starten oder zu beenden. Das *mcug3Xserver*-Shellscript unterstützt auch den Start des Webservice unter Zuhilfenahme des Init-V-Prozesses. Je nach Distribution gibt es verschiedene Verzeichnisse, und zwar eines für jeden Runlevel. Diese Verzeichnisse stehen entweder direkt in /etc/ (Debian) oder unter /etc/rc.d/ (Mandrake, RedHat) oder /sbin/init.d (SuSE) und heißen rc0.d, rc1.d, rc2.d ... entsprechend dem Runlevel. Das Skript kann in das Skript-Verzeichnis der entsprechenden Distribution (z.B. /etc/init.d) kopiert werden und es können entsprechende Verknüpfungen in den gewünschten Runlevels erzeugt werden. Dies hat den Vorteil, dass der Webservice *mcug3Xserver* automatisch beim Systemstart des Linux-OS gestartet wird.

6 mcfg.exe – Konfiguration und Inbetriebnahme

Das Programm mcfg.exe wird ausführlich im BHB-Handbuch der xPCI-800x beschrieben. Die Anwendung *mcfg.exe* befindet sich im Unterverzeichnis *windows* der Linux TOOLSET-Software und kann in ein beliebiges Arbeitsverzeichnis eines Windows- oder Linux-Betriebssystems kopiert werden. Alternativ kann auch ein mcfg MSI-Installationspaket wie in Kapitel 6.2 beschrieben verwendet werden.

6.1 WINE – Emulator für die Windows-Anwendung mcfg.exe

WINE, ein rekursives Akronym für »WINE Is Not an Emulator«, ist ein Computerprogramm für Linux- und Unix-Rechner. Mit *WINE* ist es möglich, Programme, die für das Betriebssystem Windows geschrieben wurden, auf dem X Windows-System ablaufen zu lassen. *WINE* beinhaltet den Quellcode und die Dokumentation mit Beispielen und darf frei eingesetzt werden. Portierungen existieren unter anderem für Linux, Solaris und für die verschiedenen BSD-Varianten. *WINE* kann ohne vorhandene Windows-Installation verwendet werden. Da jedoch einige Bibliotheken noch unvollständig implementiert sind, bietet *WINE* die Möglichkeit, DLLs und die Registry einer vorhandenen Windows-Version zu verwenden, um so die Kompatibilität zur Windows-Applikationen zu verbessern.

Sofern das *mcfg.exe*-Programm unter Linux zum Einsatz kommen soll, ist die Installation des *WINE*-Pakets notwendig. Aktuelle WINE-Emulatoren sollten die mcfg.exe-Anwendung ohne Anpassungen direkt ausführen können.

Für ältere WINE-Versionen sind zwei zusätzliche Anmerkungen hinsichtlich der Konfiguration des *WINE*-Pakets zu machen, wobei die hier angegebenen Pfadangaben gegebenenfalls anzupassen sind:

- Das *WINE*-Paket wird nur mit wenigen Schriftarten ausgeliefert, da diese nicht beliebig weitergegeben werden dürfen. Eine mögliche Lösung hierfür wäre das Kopieren der TTF-Schriftarten-Dateien von einem Windows-Rechner (im Windows-Unterverzeichnis „Fonts“) in das *WINE*-Zielverzeichnis z.B. ~/.wine/fake_windows/Windows/Fonts/ .
  Ohne diese zusätzlichen Schriftarten ist es sehr mühsam, mit mcfg.exe zu arbeiten.
- Wenn der integrierte Texteditor der *mcfg.exe*-Anwendung benutzt werden soll, sollten die nativen DLLs der Rich-Edit-Komponenten *riched20.dll* und *riched32.dll* von einem Windows-Rechner aus dem Windows-System-Verzeichnis in das Verzeichnis .wine/fake_windows/Windows/System kopiert werden, da die in *WINE* eingebetteten Rich-Edit-Komponenten nicht korrekt mit der mcfg.exe-Anwendung zusammenarbeiten. Damit die nativen Rich-Edit-Komponenten vom *WINE*-Paket auch benutzt werden, muss die .Datei wine/config angepasst werden. Hierzu sind die beiden Einträge in der Sektion *DllOverrides* notwendig:
  [DllOverrides]
  ...
  "riched32" = "native"
  "riched20" = "native"

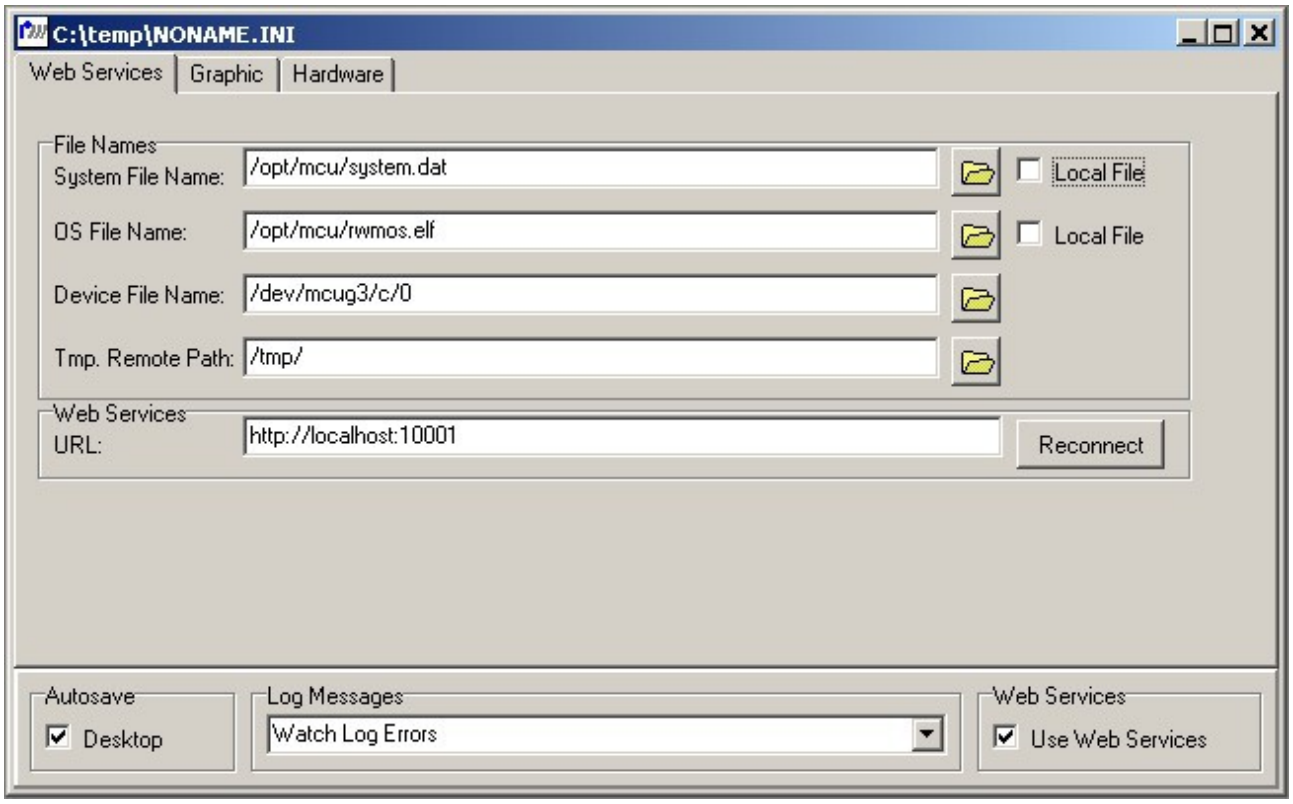
6.2 mcug3X.dll – Schnittstelle zum mcug3Xserver

Die DLL-Bibliothek mcug3X.dll wurde für die Unterstützung der Webservice-Funktionalität für die xPCI-800x-Controller erstellt und arbeitet u.a. auch mit der *mcfg.exe*-Anwendung zusammen. Sofern der Zugriff auf xPCI-800x-Geräte über Webservice-Methoden ausgeführt werden sollen, sollte diese DLL in Abhängigkeit davon, wo die *mcfg.exe*-Anwendung zur Ausführung kommt, in das Systemverzeichnis des Windows-Rechners oder in das Systemverzeichnis .wine/drive_c/windows/system32 des Linux-Systems kopiert werden. Die Treiber-DLL mcug3X.dll befindet sich im Unterverzeichnis *windows* der Linux-TOOLSET-Software.

Das Komplettpaket mcfg.exe + mcug3X.dll kann auch mit Hilfe eines MSI-Setup-Pakets unter wine installiert werden. In diesem Fall werden die Programme und Dateien automatisch an die richtigen Stellen in der WINE-Umgebung kopiert. Das aktuelle mcfg-Paket (setup.exe) kann jederzeit unter www.addi-data.de/downloads heruntergeladen und mit dem WINE-Emulator ausgeführt werden. Die Installation erfolgt dann wie bei den normalen Windows-Programminstallationen.

6.3 Projektparameter – Webservice-Schnittstelle aktivieren

Um die mcfg.exe-Anwendung für Remote-Zugriffe über Webservice-Funktionalität einzurichten, sind nur wenige Konfigurationsschritte erforderlich. Die Konfiguration wird anhand folgender Grafik erläutert:



Die Konfiguration der Webservice-Schnittstelle erfolgt im Dialog [File][Project Parameter]. Dort ist zunächst das Kästchen „Use Web Services“ zu aktivieren. Bei der Angabe des URL können sowohl feste IP-Adressen als auch symbolische Netzwerknamen verwendet werden. Einfach formuliert ist der URL der Name eines Webservers im Internet, über den dieser angesprochen werden kann. Wichtig bei der Angabe des URL ist die Portauswahl (hier 10001), welche mit dem des Webservice *mcug3Xserver* übereinstimmen muss!

Beispiele:

- http://localhost:10001
  Der Webserver (mcug3Xserver) läuft auf dem lokalen Gerät, in welchem die Steuerung eingebaut ist.
- http://192.168.178.98:10001
  Der Webserver läuft auf dem Rechner mit der IP-Adresse 192.168.178.98.
- http://mcu:10001
  Der Webserver läuft auf einem Gerät im lokalen Netzwerk.
- http://company.org:10001
  Der Webserver läuft auf einem Gerät im externen Netzwerk. Dort muss ein IP-Forward zu einem entsprechenden Gerät innerhalb des Netzwerk-Routers eingerichtet werden.

Die Dateien system.dat, rwmos.elf und die Geräte-Datei sollten nun ab diesem Zeitpunkt mit Hilfe des integrierten Dateidialogs bequem auszuwählen sein. Ebenso sollte nun die Steuerung gebootet werden können [TOOLS][SYSTEM BOOT]. Im Feld „Tmp.Remote Path“ muss ein temporärer Pfad auf dem Remote-System angegeben werden, in welchem Zwischendateien abgelegt werden können. Wenn lokale Dateien für RWMOS.ELF oder System.DAT verwendet werden sollen, kann der entsprechende Merker gesetzt werden. Im Bedarfsfall werden diese dann auf das Remote-System in den Tmp.Remote Path kopiert.

7 Programmierung der xPCI-800x-Geräte unter Linux

Wie bereits in der Einführung erwähnt, ist die Programmierung der xPCI-800x-Geräte unter Linux identisch mit der Programmierung unter Windows, mit der Ausnahme, dass bei jeder Funktion ein sogenanntes Handle zusätzlich als erster Parameter übergeben werden muss.

7.1 Treiber-Bibliothek libmcug3.a, libmcug3.so, Header-Datei mcug3.h

Zur Erstellung von Anwenderprogrammen für Linux wird eine zusätzliche Treiber-Bibliothek *libmcug3.a* bzw. libmcug3.so benötigt. Diese befindet sich im Unterverzeichnis *mcug3lib* der TOOLSET-Software für Linux. Ebenfalls in diesem Verzeichnis befindet sich die Header-Datei *mcug3.h*, welche die in *libmcug3.a [a, so]* enthaltenen Funktionen auflistet. Die Treiber-Bibliothek ist zur Erstellung von User-Mode-Programmen notwendig und kommuniziert über das Memory-Mapped-I/O-Zugriffsverfahren unter Zuhilfenahme des LKM-Treibers mcug3.[o, ko] mit dem xPCI-800x-Controller. Die Bibliotheken beinhalten diverse Funktionen und Schnittstellen, die nicht unter der GPL-Lizenz weitergegeben werden können. Deshalb kann der Quelltext zu *mcug3lib.a [a, so]* nicht an Dritte weitergegeben werden.
Die Bibliothek libmcug3.a (archive) wird für statisches Binden mit der Benutzeranwendung benötigt. Die Bibliothek libmcug3.so (shared object) hingegen wird für dynamisches Binden benötigt und hat den Vorteil, dass sich verschiedene Anwendungen den Speicher teilen und damit möglichst effizient die Ressourcen (Flash / Ram) nutzen. Es ist darauf zu achten, dass die Bibliothek mcug3lib.so und deren symbolische Verknüpfungen (mcug3lib.so*) im Zielsystem im Verzeichnis /usr/lib installiert sein müssen. Die im Lieferumfang enthaltenen Beispielprogramme (siehe Kapitel 7.3) sind seit Revision V2.53S für dynamisches Binden ausgelegt.

7.2 Linux-Gerätetreiber öffnen

Im folgenden Beispiel wird gezeigt, wie der Kernel-Modul-Treiber mcug3.[o, ko] geöffnet wird, um das entsprechende Geräte-Handle für die weiteren Zugriffe auf den Gerätetreiber zu erhalten.

```
/* Open and initialise MCUG3 device */
int hndl;

if (McuG3Open("dev/mcug3c0", &hndl) == 0)
{
  /* driver successfully opened. */
} else {
  /* driver open error /
  exit 1;
}
…
/* Insert your application code */
…
McuG3Close(hndl); /* close the driver */
```

7.3 Beispiele

Im Lieferumfang der Linux-TOOLSET-Software sind zwei Programmbeispiele für die xPCI-800x-Geräte enthalten, welche sich in den Unterverzeichnissen *sample01* und *sample02* befinden. Diese Beispiele sind zum Teil an die Windows-Beispielprogramme angelehnt und zeigen die einfache Portierungs- und Nutzungsmöglichkeit für das Linux-Betriebssystem auf.

Achtung: Zur Ausführung der Beispielprogramme werden noch folgende Dateien und Programme benötigt:

- rwmos.elf (Betriebssystem-Datei für die xPCI-800x-Geräte, evtl. kundenspezifisch!)
- system.dat (System-Datei mit Konfigurationsdaten für Antriebe und Maschine).

Diese Dateien sind nicht Bestandteil der Linux-TOOLSET-Software. Sie sind aber auf der Standard-CD der TOOLSET-Software enthalten bzw. können im Internet heruntergeladen werden.

8 Schlussbemerkung

Das Linux-Betriebssystem eröffnet neue und vielfältige Möglichkeiten zur Nutzung von Hard- und Software. Die Konfiguration und Administration erfordern jedoch zum Teil genaue Kenntnisse des jeweiligen Systems. Dieses Dokument wurde mit der Absicht verfasst, so weit wie möglich allgemein zu bleiben und die Eigenschaften der jeweiligen Distributionen außer Betracht zu lassen.
Sofern Sie als Anwender ein Problem oder lückenhafte oder sogar fehlerhafte Informationen entdecken, würden wir uns über ein entsprechendes Feedback von Ihnen freuen, damit wir die Dokumente und Software gegebenenfalls an die entsprechenden Erfordernisse anpassen können. Für diese Unterstützung möchten wir uns jetzt schon recht herzlich bei Ihnen bedanken.
Bitte richten Sie Ihre Support-Anfragen an: info@addi-data.com