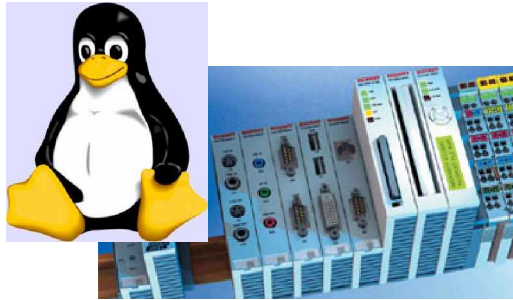


## **Beckhoff CX1000, die offene Plattform: grafisch programmierbares PLC - Laufzeitsystem unter Embedded Linux für den Embedded PC CX1000**

Bisher kennt man die CX1000 mit den Betriebssystemen Windows CE und Embedded XP. Doch nun gesellt sich ein drittes Betriebssystem dazu, Embedded Linux. Einmal mehr beweist die CX1000 damit eine offene Plattform zu sein. Die Firma ProSign hat in Zusammenarbeit mit der Firma Eurotope ein Linux Betriebssystem auf dem Embedded PC portiert und darauf ein grafisch programmierbares PLC - Laufzeitsystem implementiert.



**Abb. 1 Linux auf der CX1000**

### **Linux in der Automatisierung**

Linux ist aus der Automatisierung mittlerweile nicht mehr weg zu denken. Dieses enorme Wachstumspotential ist nicht zuletzt darauf zurückzuführen, dass Linux durch den freien Quellcode offen, technisch durchsichtig und damit schnell an wechselnde Anforderungen im Bereich Automatisierung anpassbar ist. Linux ist gleichzeitig netzwerkfähig. In Zeiten zunehmender Vernetzung von Embedded Systemen nimmt die Netzwerkfähigkeit eine enorm wichtige Position in Embedded Betriebssystemen ein. Oft werden Embedded Systeme in großen Stückzahlen eingesetzt. Selbst geringe Lizenzkosten für das Betriebssystem können so enorme Größen erreichen. Linux ist also wegen seiner offenen Quellen und den günstigen Lizenzbedingungen sehr interessant für den Einsatz in Embedded Systemen.

Die Entscheidung zu Linux fiel den Entwickler - Firmen ProSign und Eurotope aufgrund dieser Bedingungen nicht schwer. Aber wie sieht es mit der Unterstützung bei Entwicklungswerkzeugen aus um Applikationen unter Linux zu entwickeln?

Hier bietet die Linux- Gemeinschaft leistungsstarke, durchgängige und freie Werkzeuge an. An dieser Stelle sei auf den allseits bekannten GNU C/C++ Compiler gcc sowie dem GNU Debugger gdb hingewiesen. Mittlerweile stehen auch viele Firmen bereit, die sich auf Distribution, Entwicklung und Wartung rund um Embedded Linux spezialisiert haben.

### **Echtzeit-Linux**

Bei allen Vorteilen die das freie Betriebssystem bietet, hat Linux allerdings mit den offiziellen Kernelversionen eine Einschränkung im Bereich der Echtzeitfähigkeit. Die Echtzeitfähigkeit eines Embedded Systems ist aber unumgänglich. D.h. es können keine harten Echtzeitbedingungen, wie es oft von Maschinensteuerungen gefordert wird, erfüllt werden. In den meisten Systemen die mit einem Embedded Linux ausgerüstet sind, ist daher eine Echtzeitbetriebssystem-Erweiterung (z.B. RTAI, RT-Linux, KURT,..) implementiert. Mit einer solchen Erweiterung sind dann extrem geringe Latenzzeiten möglich, wie man sie von echtzeitfähigen Betriebssystemen erwartet.

### **CX1000 –das ideale System**

Das Betriebssystem allein macht noch kein Embedded System aus. Für ein gutes System ist auch die Hardware von entscheidender Bedeutung. Die CX1000 ist modulares Steuerungssystem. Modular bedeutet hier, dass die Steuerung je nach Aufgabenstellung durch Module zusammengesteckt werden kann. Dies ermöglicht es immer die kostengünstigste Variante zu wählen. Sogar ein Betrieb ohne Display und Tastatur ist möglich. In diesem Falle erfolgt die Kommunikation über die im CPU - Modul enthaltene RS-232 Schnittstelle oder über die Ethernetschnittstelle. Weitere Kommunikationsschnittstellen können durch optionale Module ergänzt werden. Es existieren Feldbusanschlungen für Profibus, CANopen, DeviceNet, SERCOS und Lightbus. Die CX1000 basiert auf einem x86-kompatiblen Prozessor und unterstützt den IBM-PS2 Standard. Also die besten

Voraussetzungen für ein Linuxsystem. Linux, und das ist allgemein bekannt, hat die größte Verbreitung und damit die beste Unterstützung bei den x86-er Systemen.

Um Feldbusklemmen mit der CX1000 anzusteuern, benötigt man eine entsprechende Applikation für das Linux Betriebssystem. In den meisten Fällen wird hier wahrscheinlich eine mit C oder C++ erstellte Applikation ausreichend sein. Diese „statischen“ Programme bieten aber hinsichtlich Ihrer Wartbarkeit und Pflege kein erschöpfendes Potenzial. Zudem ist es üblich, Applikationen für Embedded Systeme mit so genannten Cross – Developmentsystemen zu erstellen. Bei einem Cross - Developmentsystem wird das Steuerungsprogramm auf einem Desktop-PC erstellt und über eine Kommunikationschnittstelle (meistens Ethernet oder RS232) auf das Embedded System (in diesem Zusammenhang auch Zielsystem genannt) geladen. Der Vorteil den solche Systeme bieten, besteht in zumeist komfortableren Entwicklungsumgebungen und der einheitlichen Sicht auf verschiedenste Zielsysteme.

### Grafisches Programmieren der CX1000

Eine solches komfortables Entwicklungstool ist das grafische Programmiersystem iCon-L. Es unterteilt sich in die grafische Entwicklungsoberfläche auf einem Windows PC und das Laufzeitsystem auf der CX1000 unter dem Betriebssystem Embedded Linux. Das mit der grafischen Oberfläche iCon-L unter Windows erstellte SPS - Programm wird dann wahlweise über Ethernet oder über eine einfache RS232 auf die CX1000 übertragen und dort abgearbeitet.

Mit iCon-L wird konsequent in Funktionsbausteinsprache programmiert. In Bibliotheken organisiert bietet iCon-L einen umfangreichen Vorrat an Bausteinen.

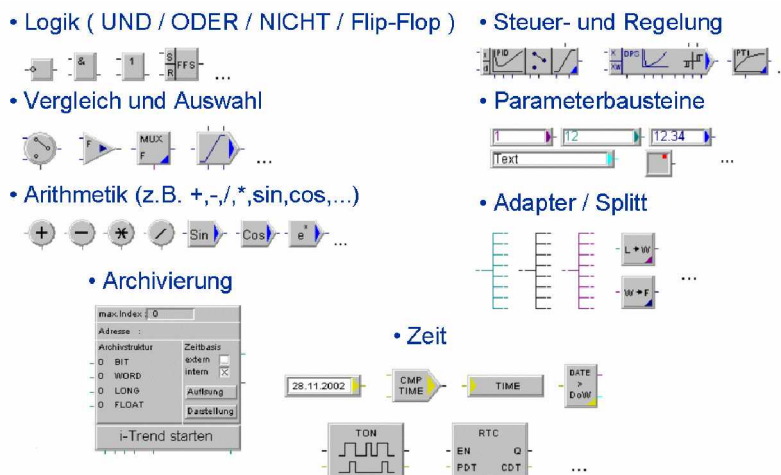


Abb. 2 Einige Standardbausteine von iCon-L®

Für die Programmierung bietet iCon-L zwei grundsätzliche Paradigmen an: einerseits die Ablaufsprache andererseits den Datenfluss. Für die Ablaufsprache bedient sich iCon-L den auf Petrinetzen basierenden Schritten und Transitionen (Abb.3). In iCon-L können jedoch beide Methoden miteinander vermischt werden.

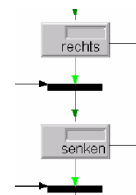


Abb. 3 Schritte / Transitionen für die Ablaufsprache

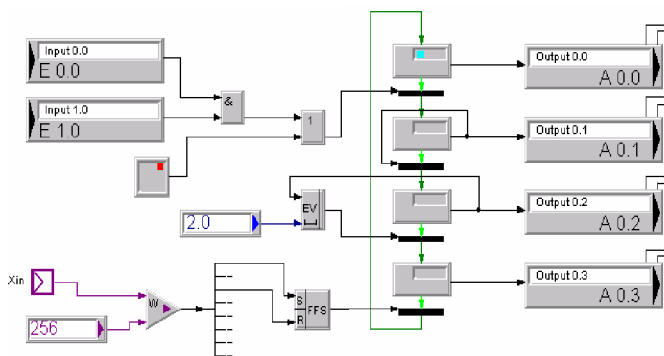


Abb. 4 Vermischung von Datenfluss- und Ablaufsprache

Eine weitere Gruppe neben den Standardbibliotheken, bilden die E/A Bausteinbibliotheken. Diese dienen dem Zugriff auf die Ein- und Ausgänge der Hardware. Bei einem CX1000 System sind das die Feldbusklemmen.

Für maßgeschneiderte Lösungen bietet iCon-L die Möglichkeit, eigene Bibliotheken zu erstellen. Diese Bibliotheken sind dann stark technologieorientiert und bieten nicht selten Datenbankverbindungen oder ähnliche mächtige Schnittstellen auch zu bereits bestehendem an.

## Programme vorher simulieren

iCon-L bietet neben der Programmentwicklung auch eine Simulationsumgebung an. Die Simulation bildet auf dem Windows-PC ein CX1000 Zielsystem nach. D.h. bevor man das SPS-Programm in die Steuerung lädt und dort abarbeiten lässt, kann man es „im Trockenen“ auf seine Funktionalität testen. Ein- und Ausgänge können in der Simulation per Mausklick simuliert werden. Für umfangreichere Tests steht auch ein Testfolgenerator für reproduzierbare Eingangsmuster zu Verfügung.

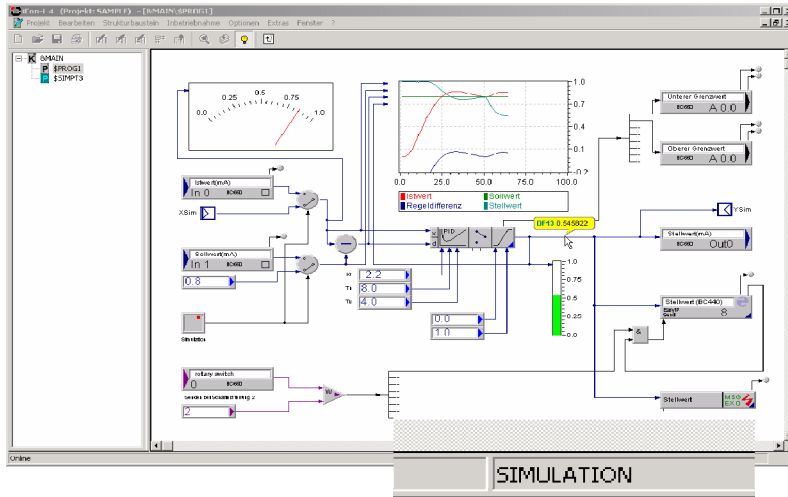


Abb. 5 Test vom Programmen in der in iCon-L integrierten Simulationsumgebung

Während der Simulation stehen hilfreiche Funktionen zur Fehlersuche (Debuggen) des Programms zur Verfügung, wie zum Beispiel das Anzeigen der Werte von Signallinien, Datenlogger sowie Testfolgenerator. Diese Hilfsmittel sind ebenso bei der Inbetriebnahme am realen Zielsystem (CX1000) nutzbar.

## Zusammenfassung

Die CX1000 mit dem Embedded Linux Betriebssystem bietet eine lizenzkostengünstige Alternative zu den Embedded Windows Systemen. Der freie Quellcode von Linux stellt zudem sicher, dass Linux offen, technisch durchsichtig und schnell an wechselnde Anforderungen anpassbar ist. Mit einer Echtzeiterweiterung können auch unter Linux extrem geringe Latenzzeiten erreicht werden. Diese Plattform wird durch das grafisch programmierbares PLC Laufzeitsystem iCon-L sinnvoll ergänzt. Dienste wie FTP und Telnet bieten zudem die Möglichkeit der Fernwartung des Embedded Systems. Für das grafische Programmiersystem kann von der Firma ProSign ([www.pro-sign.de](http://www.pro-sign.de)) eine kostenlose Demoversion bestellt werden. Für weitere Informationen zu Linux auf der CX1000 wenden Sie sich bitte an Thomas Wust, ProSign GmbH, ([prosign@pro-sign.de](mailto:prosign@pro-sign.de)).

## Literatur Quellen:

- [1] [www.pro-sign.de](http://www.pro-sign.de); „iCon-L – ein modernes Engineeringtool – von der ersten bis zur letzten Projektphase grafisch“; Thomas Wust ProSign GmbH
- [2] [www.linux-magazin.de](http://www.linux-magazin.de); „Linux im Vormarsch – Embedded Linux“