

Durch modularen Ansatz zum flexiblen CANopen Service-Tool

In den vergangenen Jahren wurden CANopen-Geräte in Hinsicht auf Prozess I/O-Funktionalität und Konfigurationsmechanismen mehr und mehr erweitert. Traditionell war die Inbetriebnahme von CANopen-Geräten auf die Parametrisierung von I/Os und Kommunikationsfunktionalität über ein CANopen-Netzwerk beschränkt. Diese Konfiguration wurde beispielsweise über ein eigenständiges CANopen-Konfigurations-tool oder einen CANopen-Konfigurationsmanager durchgeführt, welcher in einem NMT-Master (Netzwerk-Management) integriert war.

Neben der konventionellen Konfiguration der Kommunikationseigenschaften können heute auch, wie in CiA 305 spezifiziert, Parameter wie Node-ID und Bitrate mittels Layer Setting Services (LSS) konfiguriert werden. Firmware-Updates können über die Programm-Downloadmechanismen, wie in CiA 302 beschrieben, durchgeführt werden.

Diese beiden zusätzlichen Funktionen für die Geräteinbetriebnahme werden jedoch nicht standardmäßig von CANopen-Konfigurationstools unterstützt. Besonders für den Firmware-Update wird von Geräteherstellern oft eine proprietäre Lösung gewählt, da der Markt an Tools mit Schnittstellen für gerätespezifische Erweiterungen zur Zeit sehr begrenzt ist.

Eine wichtige Anforderung an erweiterbare Konfigurationstools ist dabei die Bereitstellung von stark abstrahierten CANopen-Diensten, damit sich der Anwender nicht mit den Details des CANopen-Protokolls beschäftigen muss.

Anforderungen

Zusätzlich zu den allgemein üblichen Funktionen für das Herunterladen der Geräte-Konfiguration, welche mittels eines Standard-CANopen-Konfigurationstools generiert wurde und als DCF (Standardized Device Configuration File) bereitgestellt wurde, sollte ein generisches Inbetriebnahme- und Service-Tool die LSS Master-Funktionalität und den Programm-Download-Mechanismus implementiert haben.

Ferner sollte das Tool die Durchführung von gerätespezifischen Diagnoseaufgaben erlauben. Hierbei ist jedoch ein umfangreicher CANopen-Analyse nur von begrenztem Wert, da die Service-Mitarbeiter in der Regel nicht das für die Interpretation der CAN und CANopen Trace-Dateien erforderliche Fachwissen haben. Abgesehen von diesen technischen Gesichtspunkten, sollte das Tool mobil und kompakt sein sowie idealerweise auch auf einem PDA lauffähig sein.

Flexibilität, Erweiterbarkeit und eine einfach bedienbare Oberfläche, die auch den Betrieb auf PDAs ermöglicht, waren daher die Hauptgesichtspunkte bei der Entwicklung der nächsten Generation des IXXAT CANopen Device Managers.

Als Entwicklungsplattform wurde Microsoft .Net Framework V2.0 und C# gewählt, da hierdurch auch eine direkte Migration in mobile Geräte ermöglicht wird. Ein weiterer Vorteil von einer .Net-Umgebung ist die komfortable Code- und Oberflächengenerierung durch UML basierende Tools.

Modularer Ansatz

Die vorangegangenen Überlegungen führen zu einem modularen Plug-In-Konzept, bei dem alle CANopen-Kommunikationsmechanismen und -Dienste in einem Basis-Tool bereitgestellt werden.

Das Basis-Modul stellt definierte Schnittstellen zu Verfügung, über welche die dynamisch installierten Plug-Ins auf die CANopen-Dienste zugreifen können. Spezifische Diagnose- und Konfigurationsfunktionen sowie eine entsprechende Oberfläche werden über die individuellen Plug-Ins zur Verfügung gestellt [Bild 1]. Die Plug-In-Module sind als unabhängige .Net Assemblies realisiert, welche eine zukünftige Erweiterung der Funktionalität ohne Änderung der Basis-Komponenten erlaubt.

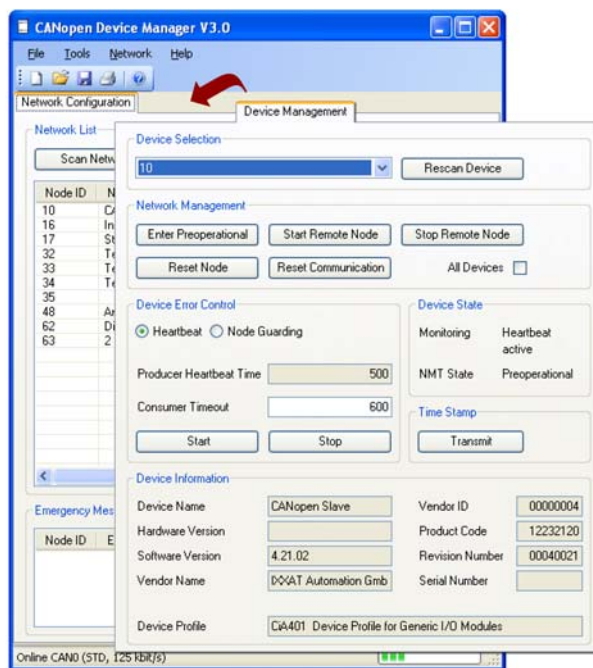


Bild 1: Spezielle Plug-In Module erweitern die Grundfunktionen des CANopen Device Managers

Vorteil dieses Ansatzes ist, dass das Tool die gängigsten CANopen-Dienste, wie Netzwerk-Management, PDO, SDO-Gerätezugriff, LSS und den Download von Konfigurationsdaten beinhaltet und darüber hinaus einfach um zusätzliche Funktionen, welche für spezifische Geräte erforderlich sind, erweitert werden kann.

Anwendungsbereiche

Zur Zeit sind für den CANopen Device Manager Plug-In-Module für das generische Gerätemanagement, den Geräte-Zugriff sowie den Programm- und DCF-Download verfügbar. Ferner ist ein Modul für die LSS Master Funktionalität verfügbar.

Während das Device-Management die traditionellen NMT, SDO (mittels Device Object Browser) und PDO Funktionen unterstützt, ermöglicht das Download-Plug-In den Upload der Firmware auf entsprechend implementierte CANopen-Geräte unter Verwendung der Programm Download Mechanismen, welche in CiA 302 spezifiziert sind (benötigt die Objecte 1F50_h "Program data", und 1F51_h "Program control"). Die integrierte LSS Master Funktionalität entspricht der kürzlich veröffentlichten Version 2.0 der CiA 305 "Layer Setting Services and Protocols" Spezifikation.

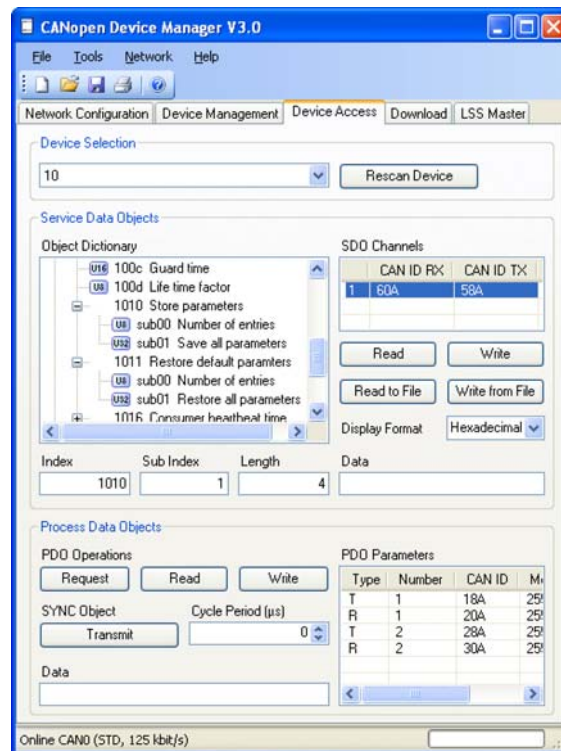


Bild 2: Device Access Plug-In Modul mit SDO und PDO Funktionalität

Der CANopen Device Manager ist ein einfach zu bedienendes Test-Tool und kann sowohl von Entwicklungsingenieuren für den Test von neu entwickelten Geräte verwendet werden als auch als Grundlage für ein vollständiges End-of-Line Testprogramm.

Ein weiterer Anwendungsbereich liegt im Feld-Test von Geräten durch Entwickler und Servicetechniker. Hierfür können spezielle Plug-Ins erstellt werden.

Ausblick

Vorteil der modularen und Plug-In orientierten Softwarestruktur ist, dass der CANopen Device Manager flexibel auf aktuelle und zukünftige Kundenanforderungen für die Inbetriebnahme und den Service von CANopen-Geräten angepasst werden kann.

Zusätzlich zu den heute bereits vorhandenen CANopen Device Manager Plug-Ins wird IXXAT auch spezifische Plug-Ins für die wichtigsten CANopen-Geräteprofile anbieten. Plug-Ins für CiA 401 "Generic I/O devices" und CiA 402 "Drives and motion control" sind bereits in Planung. Die Beschreibung der internen Schnittstelle kann darüber hinaus für OEM-Kunden offengelegt werden, damit diese spezifische Plug-Ins für ihre Produkte erstellen können.