

# CANopen Master Source Code

## Embedded Software für CANopen

Mit dem Master Source Code hat der Anwender die Möglichkeit, die Firmware seiner Produkte um CANopen-Master-Funktionalität zu erweitern. Die Software ist insbesondere für die Integration in Steuerungen geeignet.

### Eigenschaften und Vorteile

Der Master Source Code stellt alle erforderlichen Funktionen für einen CANopen Master zur Verfügung. Hierbei sind die folgenden Standards der Nutzerorganisation CAN in Automation e.V. (CiA) implementiert:

- > CiA DS 301 Version 4.02, Application Layer and Communication Profile
- > CiA DS 302 Version 3.1, Framework for programmable CANopen Devices
- > DSP 405 Version 2.0, Interface and Device Profile for IEC61131-3 Programmable Devices

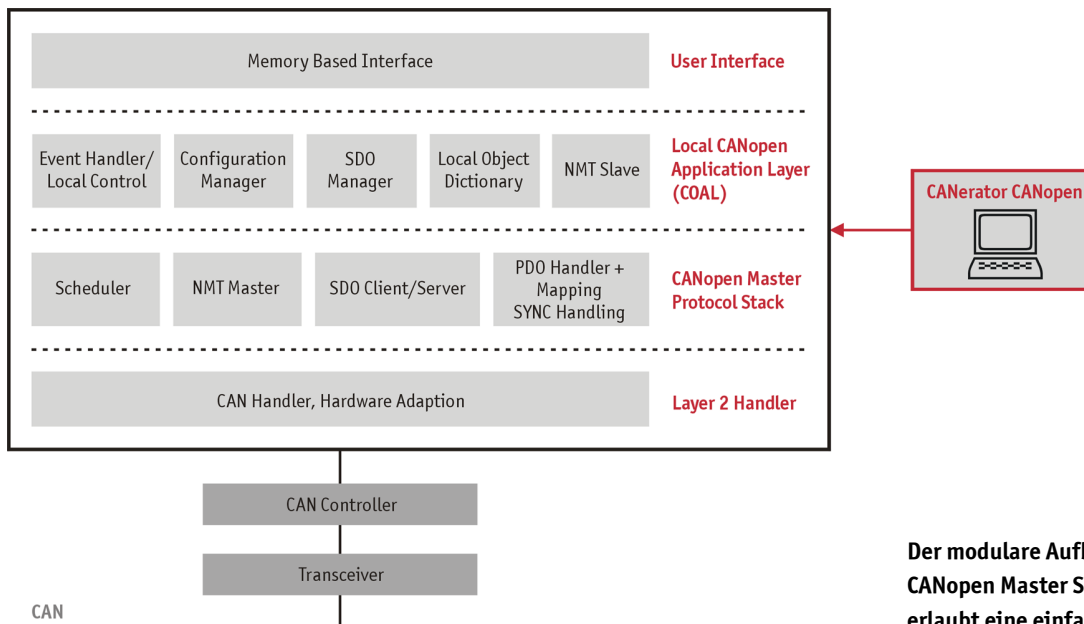
Durch seine generische Struktur können Slaves beliebiger Profile angeschlossen werden. Der Anwender hat die Wahl, die Software in sein eigenes Projekt mit einzubinden, oder die CANopen-Funktionalität als selbständigen Task im System aufzubauen.

Durch die Anwendung des CANopen Source Codes kann bei der Produktentwicklung erheblich Zeit eingespart werden. Der Anwender kann sich auf die Integration seiner eigenen Applikation konzentrieren, die Implementierung des CANopen-Protokolls wird hierbei wesentlich vereinfacht.

### Funktionen

Der Source Code bietet folgende Funktionalitäten an:

- > Netzwerkmanagement (NMT) – Kontrolle der angeschlossenen Knoten über die NMT-Nachricht. Weiterhin werden Guarding oder Heartbeat durchgeführt.
- > PDO-Handling – Über PDOs werden die eigentlichen Prozessdaten übertragen. Die PDOs können über das Objektverzeichnis konfiguriert werden. PDOs werden auch synchron behandelt.
- > Lokales Objektverzeichnis – Der Code stellt über den Bus ein eigenes Objektverzeichnis zur Verfügung, das durch den Anwender erweitert werden kann.
- > SDO-Behandlung – Der Anwender kann über den SDO-Handler eigene SDO-Requests auslösen.
- > Konfigurationsmanager (CMT) – Damit kann in Embedded-Systemen eine zentrale Konfiguration der angeschlossenen CANopen-Knoten erfolgen.



**Der modulare Aufbau des CANopen Master Source Codes erlaubt eine einfache Implementierung.**

V1.3 2005-7

Der CANopen Master Source Code von Vector ist für eine Vielzahl der gängigen Mikrocontroller verfügbar.

Aktuelle Informationen dazu finden Sie auf unserer Homepage unter: [www.vector-informatik.com/sourcecode](http://www.vector-informatik.com/sourcecode)

### Spezielle Funktionen

Die Software unterstützt die Speicherung von Daten in nicht-flüchtigem Speicher. Es stehen Schnittstellen für die Anbindung an ein Dateisystem oder an Flash-Speicher zur Verfügung.

Beim Systemstart wird die „Boot Slave“-Funktionalität voll unterstützt. D.h. das Netzwerkmanagement trägt selbst Sorge für das korrekte Starten und Konfigurieren der angeschlossenen Knoten. Auch die Ausfallkontrolle wird vom CANopen Master übernommen.

### Anwendungsgebiete

Der Code kann überall dort eingesetzt werden, wo der Kunde seine Steuerungen mit CANopen-Funktionalität ausstatten möchte. Anwender sind in erster Linie Firmen, die selbst CANopen-Geräte herstellen, sowie Systemintegratoren, die Gesamtsysteme erstellen müssen.

### Hardware-schnittstellen

Die Software ist in ANSI-C geschrieben und daher auch auf den verschiedensten Plattformen lauffähig. Als Mindestanforderung sollte die Verwendung eines 16-Bit-Mikrocontrollers in Betracht gezogen werden. Für die Ansprache der verschiedenen CAN-Controller-Typen stehen Anpassungen zur Verfügung. Die Schnittstelle des CAN-Treibers ist so ausgelegt, dass die Umstellung auf andere CAN-Controller leicht erfolgen kann.

Für einige Evaluationsumgebungen stehen Board Support Packages (BSP) zur Verfügung. Ein BSP besteht aus einem Makefile, einem

Textfile mit den Einstellungen der #defines und einer Datei mit den notwendigen Systemanpassungen. Auch wenn der Anwender eine andere Zielumgebung hat, liefern diese Dateien wertvolle Hinweise für die Portierung auf die Kundenhardware. Derzeit stehen die folgenden BSP zur Verfügung:

- > Anpassung für Evaluation-Board ec376 mit Motorola 68376 von Würz Elektronik GmbH und GNU-Compiler beziehungsweise Microtec-Research-Compiler
- > Anpassung für kitCON 164CI mit Infineon SAB164CI von Phytec GmbH und Tasking-Compiler
- > Anpassung für Vector CANopen-Treiber unter Windows 95/98/NT und Microsoft-Compiler VC++ 6.0
- > Power PC

### Datenschnittstelle

Die Datenschnittstelle erlaubt die Entkopplung der Applikation vom CANopen Stack als Kommunikations-Task. Damit kann sowohl ein Single-Prozessor-System als auch ein Multi-Prozessor-System realisiert werden. Im letzteren Fall wird die Datenschnittstelle auf ein physikalisches DPRAM gelegt:

- > Kommando-Queue – Die Befehls-Queue wird von der Steuerung genutzt, um Anforderungen an den Application Layer abzusetzen. Hierzu gehören die Übergabe von Initialisierungsdaten oder Anforderungen von Funktionsbausteinen (z.B. zum Absetzen von SDOs).

- > Ereignis-Queue – Die Ereignis-Queue transportiert den Datenstrom vom Application Layer zur Steuerung. Bei diesen Daten handelt es sich im Wesentlichen um asynchrone Bestätigungen (z.B. Antwort auf SDO-Request) und Ereignisse (z.B. Fehlerzustände).
- > Prozessabbild – Die Applikation agiert mit dem CANopen-Netzwerk im Wesentlichen über ein Prozessabbild. Dieses Abbild enthält alle E/A-Daten und sonstigen Prozessgrößen, die vom CANopen Master übertragen werden sollen.

#### **Netzwerkmanagement**

Dieses Modul übernimmt die Verwaltung und die Überwachung der im Netz angeschlossenen Knoten. Hierzu ist es erforderlich, die anzusprechenden und zu überwachenden Knoten anzumelden. Auch die entsprechenden Parameter für die Fehlerüberwachung (Guarding, Heartbeat) müssen übermittelt werden. Diese Informationen werden im lokalen Objektverzeichnis abgelegt und können damit sowohl extern über den Bus als auch intern über das DPRAM abgefragt werden. Die Überwachung und der Start von Knoten läuft weitgehend automatisch.

#### **PDO Handling**

Der PDO Handler hat die Aufgabe, empfangene PDO von Slaves entgegenzunehmen und in das Prozessabbild der Datenschnittstelle weiterzuleiten. Umgekehrt überträgt er Werte der Applikation an

die Slaves. Dies kann zyklisch (per SYNC) oder ereignisgesteuert (bei Änderung) erfolgen.

Die Zuordnung zwischen PDO und Ablage der Daten erfolgt ausschließlich über das lokale Objektverzeichnis. Die Applikation ist damit vollkommen von konkreten Kommunikationsaufgaben entkoppelt.

#### **SDO Handling**

Der SDO Handler ermöglicht der Applikation den Zugriff auf die Objektverzeichnisse der angeschlossenen CANopen-Knoten. Eine Übertragung von Datenblöcken ist in beide Richtungen (Upload/Download) möglich. Für Objekte bis zu einer Größe von 4 Bytes findet das „Expedited Download/Upload Protocol“ Anwendung, für Objekte größer 4 Bytes das „Segmented Download/Upload Protocol“. Der Blocktransfer ist ebenfalls implementiert und kann u.a. auch durch den Anwender angestoßen werden.

Es ist möglich, mehrere SDO-Transfers parallel ablaufen zu lassen (nur für verschiedene SDOs).

#### **Lokales Objektverzeichnis**

Das lokale Objektverzeichnis ermöglicht den Zugriff auf die Master-Funktionen und auf Parameter und Variablen des Masters über das Netzwerk. Somit besteht die Möglichkeit, den Master über das Netz zu konfigurieren, oder aber lokal vorhandene Funktionen (Displays oder Ein- und Ausgänge) zu bedienen. Insbesondere wird hierüber der Konfigurationsmanager bedient.

Die Variablen im Objektverzeichnis können von der Applikation über die entsprechenden SDO Read/Write Requests unter Angabe der eigenen Node-ID angesprochen werden.

### **Konfigurationsmanager (CMT)**

Der Configuration Manager hat die Aufgabe, Parametersätze der CANopen Slaves zu speichern und diese zu einem bestimmten Zeitpunkt im Netzwerk an die angeschlossenen Knoten zu verteilen. Der Zeitpunkt der Verteilung kann folgendermaßen festgelegt werden:

- > Beim Start des Masters wird eine Konfiguration für alle Knoten durchgeführt, die in der Liste der zugeordneten Slaves eingetragen sind und für die eine Konfiguration vorhanden ist.
- > Bei Ausfall von Guarding oder Heartbeat wird die Konfiguration wieder angestoßen.
- > Auslösen der Konfiguration eines bestimmten Knotens zu jeder Zeit durch die Applikation – um zum Beispiel nachträglich angekoppelte Teilnetze zu konfigurieren

Die Konfiguration von CANopen-Geräten bezieht sich auf das Einstellen von Parametern in deren Objektverzeichnis. Hierbei handelt es sich hauptsächlich um PDO-Parameter und Mapping-Information.

### **Unterstützung bei der Integration**

Bei der Integration des CANopen Master Source Codes in Ihre Umgebung können wir Sie natürlich ebenfalls unterstützen. Dabei reichen die Unterstützungsleistungen von zugeschnittenen Schulungsangeboten, über Workshops bis hin zu Projektarbeit.