

PAULUS CANopen Bootloader

Ein modernes Gerätedesigns erfordert enorme Flexibilität in Hard- und Software. Bei Hardware versucht man durch reichliche Ressourcenausstattung spätere Softwareänderungen zu ermöglichen. Die Software muß einen Austausch der Software selbst ermöglichen. Die größten Freiheiten ermöglicht das In-System-Programming, also der Softwaretausch in der Maschine.

Bootloader mit einem Kommunikationsinterface ermöglichen das Einspielen von Softwareerweiterungen über ein entsprechendes Netzwerk. Standardisierte Kommunikationsobjekte und Algorithmen gewährleisten eine hohe Transparenz und Bedienfreundlichkeit.

Software Update

Der PAULUS Bootloader bietet diese Flexibilität für Geräte in CAN/CANopen-Netzwerken, da CANopen mit dem SDO Transfer standardisierte Mechanismen für die Übertragung größerer Datenmengen bereitstellt. Der Bootloader selbst arbeitet unabhängig von der Applikation als minimaler CANopen Slave Knoten nach CiA-301.

Ein Software-Update kann mit einem CANopen Master oder Konfigurationstool über den Anwenderbereich des Code-FLASH-Speichers durchgeführt werden.

PAULUS ist ein auf Codegröße optimierter Bootloader, der weitgehend kompatibel zu CANopen ist.

Um eine minimale Codegröße zu erreichen, wird auf den Einsatz eines flexiblen CANopen Protokoll Stack verzichtet. Die erforderliche CANopen Funktionalität ist fest implementiert. Unabhängig davon wurde das Projekt ist jedoch so flexibel aufgebaut, daß es einen hardware-unabhängigen Kodeteil für die Protokollabwicklung gibt und jeweils einen Prozessor-spezifischen Teil für die Anfangsinitialisierung und speziell an den jeweiligen Mikrokontroller angepassten FLASH Routinen.

Der Code des Bootloaders ist sehr allgemeingültig gehalten und modular geschrieben. Damit lässt er sich sehr leicht auf andere Hardware-Architekturen übertragen. Hinzugefügt werden müssen dann:

- die jeweilige Grundinitialisierung des Prozessors
- das FLASH Handling zur Programmierung
- die Vorschrift zum Linken von Applikationen

Anforderungen

Verschiedene CANopen Dienste sind immer vorhanden, andere können durch eine Konfigurationsdatei aktiviert werden.

- NMT Error
Heart Beat Erzeugung durch den Bootloader ist vorgesehen
- Emergency-Dienste können mit Einschränkungen genutzt werden
- CANopen Layer Setting Services wird unterstützt
- CANopen Bootloader ist SDO Server
- PDO-Dienst wird nicht unterstützt

Durch die Trennung zwischen Protokoll-Schicht und Hardware Application Layer ist prinzipiell der Einsatz auf allen Zielplattformen möglich. Lediglich die FLASH-Routinen sind an die verwendete Hardware anzupassen.

Der Ressourcenverbrauch ist dabei abhängig von dem jeweiligen Mikrokontroller und dem verwendeten Compiler sowie den Kompilereinstellungen.

- Objektverzeichnis

Die folgende Tabelle gibt einen Überblick über die im CANopen Bootloader implementierten Objekte.

Index	Sub Index	Mode	Anmerkung
0x1000	0	co	Device Type
0x1001	0	co	(*)keine Fehler unterstützt
0x1014	0	co	(*)feste Emcy COB-ID
0x1017	0	co	(*)0 oder festgelegte Zeit
0x1018	0-2	co	(*)
0x1018	3-4	co	(*) with LSS
0x1F50	0	co	(*)Number of Elements
0x1F50	1	wo	Domain - neue Firmware
0x1F51	0-1	rw	(*)
0x1F56	0-1	ro	(*)
0x1F57	0-1	ro	(*)

(*) auf diese Objekte sollte zu Gunsten der Codegröße verzichtet werden.

CANopen bietet eine kosteneffiziente Integration von Kommunikation in unterschiedliche Anwendungen. Diesen Kommunikationsmechanismus auch zum Firmware-Update in Feldgeräten zu nutzen liegt nahe. Ein heutzutage nahezu unverzichtbarer Service. Dafür wird ein Bootloader benötigt, der auch CANopen spricht.

Mit dem PAULUS steht eine neue Generation CANopen-Bootloader zur Verfügung, die eine kompakte, modulare und speichereffiziente Bootloader-Plattform für verschiedene Hardwarearchitekturen bereitstellt.

- Speichereffizient
Bis zu 50% weniger Speicherverbrauch gegenüber der vorhergehenden Generation
- Modular
Schnell anpassbar auf weitere Hardwarearchitekturen
- Kompakt
Übersichtlich und verständlicher Code erlaubt schnelle kundenspezifische Anpassungen

Typische Werte für den erforderlichen Speicherbedarf sind:

- 4 bis 8 KB Flash
- 2 KB RAM

Als Client-Gegenstelle kann jede CAN Interface Hardware verwendet werden.

Das können z.B. eine CAN-PCI/PCIe-Baugruppe, ein USB-CAN Interface oder ein Gateway gemäß CiA DSP309-3 (EtherCAN) sein.

Bei der Client-Gegenstelle ist eine CANopen-fähigen Software ausreichend, die ein Domain-Download erlaubt.

0271/01	Benutzerhandbuch CANopen Bootloader
0271/10	PAULUS-BOOTL-SRC-STM32
0271/11	PAULUS-BOOTL-SRC-dsPIC33
0271/12	PAULUS-BOOTL-SRC-ATSAM4E
0271/13	PAULUS-BOOTL-SRC-RL78F13
0271/XX	PAULUS CANopen Bootloader Entwicklung weiterer CANopen Bootloader auf Anfrage Angabe zu verwendeten Mikro-/CAN-Kontroller erforderlich

Lieferumfang

Im Lieferumfang des PAULUS Bootloaders ist neben dem Bootloader Source Code eine Beispielanwendung enthalten, die eine detaillierte Beschreibung für die Installation und Einrichtung enthält.

Das Beispiel umfasst Hinweise zur Vorgehensweise für

- Speichernutzung
- Generierung des Anwender Image
- Checksummen-Berechnung
- Erstellen der Anwendung