

PAULUS CANopen Bootloader

Modern device designs need enormous flexibility in hard- and software. Today hardware is equipped with plenty of resources in order to allow software changes later on. Software itself has to provide means to enable an update/upgrade of the current firmware. At this point in-system-programming offers the greatest possibilities, and thus a change of the software too.

Bootloader with a communication interface allow to do firmware updates by using an appropriate network. Standardized communication objects and algorithms guarantee a high level of transparency and user friendliness.

Software Update

PAULUS bootloader offers this flexibility for devices in CAN/CANopen networks, because CANopen provides with the SDO Transfer standardized mechanisms for the transmission of big data volume. The bootloader itself works independently from the application as a minimal CANopen slave node according to CiA-301.

A software update can be executed with a CANopen master or configuration tool in the user flash code memory.

PAULUS is optimized to code size and widely compatible to CANopen.

In order to achieve a minimal code size there is no flexible CANopen stack used. Instead the necessary CANopen functions are implemented. Nonetheless the project is designed flexible. It consists partly of hardware independent code for protocol handling and partly of processor specific code for the initialization and for especially adopted flash routines for the microcontroller.

The bootloader code is written rather universal and modular. It is easily applicable to different hardware architectures. To do so, it is necessary to add:

- basis initialization of the processor
- flash handling for programming
- rules to link applications

Requirements

Some CANopen services are always available, other can be activated by a configuration file.

- NMT Error heartbeat creation by bootloader is available

- emergency services can be used with limitations
- CANopen Layer Setting Services are supported
- CANopen bootloader is SDO server
- PDO service is not supported

To divide protocol layer and hardware application layer principally allows the usage on all targets. Only the flash routines need to be adopted to the used hardware.

How much resource is needed depends on the microcontroller and the used compiler and the compiler settings.

- object dictionary

The following table gives an overview of the implemented objects in PAULUS

Index	Sub Index	Mode	Note
0x1000	0	co	Device Type
0x1001	0	co	(*no errors supported
0x1014	0	co	(*fixed Emcy COB-ID
0x1017	0	co	(*0 or fixed time
0x1018	0-2	co	(*
0x1018	3-4	co	(* with LSS
0x1F50	0	co	(*Number of Elements
0x1F50	1	wo	domain - new firmware
0x1F51	0-1	rw	(*
0x1F56	0-1	ro	(*
0x1F57	0-1	ro	(*

(*) it is recommended to abandon these objects for a small code size

Features

With PAULUS a new generation of CANopen bootloaders is available. Aim of the development was the creation of a compact, modular and memory saving bootloader platform for different hardware architectures.

- memory saving
up to 50% less memory space is needed in contrast to the old bootloader generation
- modular
applicable to further hardware architectures in short time
- compact
clear and easily understandable code allows fast customer specific adaptations

typical values for needed memory are:

- 4 up to 8 KB flash
- 2 KB RAM

Any CAN interface can be used for the client.

Possible are for example CAN-PCI/PCIe cards, USB-CAN interfaces or gateways according to CiA DSP309-3 (EtherCAN).

For the client a CANopen capable software that allows a domain download is sufficient.

Scope of Delivery

The scope of delivery includes the bootloader source code and in addition an example application. This example contains a detailed description for installation and configuration.

Also included in the example are hints for

- memory
- generating of user image
- calculation of check-sums
- creation of an application

Ordering Information

0271/02	User Manual 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 Development of another CANopen Boot- loader on request information about used micro-/CAN con- troller required