MicroCANopen

Part No.:
MPESA-Embeded-Communi-basic

Small footprint, flexible CANopen stack. Designed to run on 8-bit microcontrollers with limited resources. Easy to adapt to specific applications. Royalty free. Written in C.

MicroCANopen is a small footprint, commercial grade CANopen implementation, with some advanced features. Ideal for situations requiring minimal configurability but great performance on 8-bit microcontrollers, MicroCANopen provides the perfect solution.

Designed and implemented by the authors of the book "Embedded Networking with CAN and CANopen" and members of the CAN in Automation User's Group.

MicroCANopen is written in 100% standard C code. As a consequence, clients can decide to port the software to processors not yet supported, including microprocessors, microcontrollers and DSPs. Designers familiar with their target processor can easily perform the port themselves.

MicroCANopen offers a low, one-time fee and no royalties on deployed products. And you get the entire source code for with every purchase!

Adapt the basics

MicroCANopen does not aim at 100% CANopen conformance. While it can be made 100% conformant, the primary idea behind MicroCANopen is to simply adapt the minimal set of features that is required to cover standard communication methods and that allow making use of exiting CANopen tools like configurators and analyzers. If the system growths in the future, an upgrade path is already laid out - upgrading to "true" CANopen does not require any changes in the communication channels.

Minimal resource requirements

Full-size CANopen implementations require more resources than some of the latest 8-bit microcontrollers with CAN interface have. On 8051 derivatives, MicroCANopen can fit in as little as 4k of code and about 170 bytes of RAM. Full-featured CANopen implementations are more in the range of 40k-60k of code and 600 or more bytes of RAM.

Minimal learning curve

MicroCANopen reduces the learning curve for newcomers to CANopen. As advanced features are not implemented, engineers do not need to learn about them immediately. For example, MicroCANopen allows the transmitting and receiving of CAN messages with process data, without needing the process of "mapping variables from an Object Dictionary into PDOs - Process Data Objects".

Features

The following is a list of features in MicroCANopen. The list is not exhaustive by any means, but does give a good overview.

- NMT State Machine
- Heartbeat
- Object Dictionary (OD)
- Expedited SDO
- Static PDO
- PDO with event time
- PDO with inhibit time
- Economical One Time Fee
- Full Source Code Provided
- No Royalties on Deployed Products

Suggested Application Usage

MicroCANopen is best suited for minimal CANopen slaves that are pre-configured and do not need to be re-configured during operation. CAN baud rate, the node ID and all PDO parameters are known at implementation and hard-coded into the module.

MicroCANopen Plus is best suited for CANopen slaves requiring minimal configurability. CAN baud rate, the node ID and PDO communication parameters are configurable. This allows using MicroCANopen Plus for nodes that typically require some setup during installation of the node.

MicroCANopen Classic is best suited for full-featured, highly flexible CANopen slave nodes and for minimal CANopen NMT (Network Management) Master applications. Using a setup file, CMX-CANopen can be completely re-configured. Without re-compilation Object Dictionary entries can be modified, added or removed. This allows one CMX-CANopen implementation to be used for a wide variety of devices.

Comparison Between CANopen Implementations

Feature	MicroCANopen	MicroCANopen Plus	MicroCANopen Classic
All CANopen baud rates supported	✔	✔	✔
Network Management state machine with autostart option	✔	✔	✔
Heartbeat producer, [1017H]	✔	✔	✔
Heartbeat consumer, [1016H]		✔	✔
Node Guarding responses		✔	✔
Setup via hard-coding in program	✔	✔	✔
Setup via CANopen Task Setup File (read/write to [1F50H])			✔
Object Dictionary support for data types of up to 4 bytes (expedited SDO transfer)	✔	✔	✔

Object Dictionary support for any data type		Segmented	Segmented and Block
PDO default configuration can be hard-coded	✔	✔	✔
Dynamic PDO Communication Parameters (write to [14xxH] and [18xxH] allowed)		✔	✔
Dynamic PDO Mapping Parameters (write to [16xxH] and [1AxxH] allowed)			✔
Store PDO parameters in non-volatile memory ([1010H], [1011H] and [1020H])		✔	✔
TPDO Trigger by Event Time	✔	✔	✔
TPDO Trigger by COS with Inhibit Time	✔	✔	✔
TPDO Trigger by SYNC		✔	✔
TPDO Trigger by RTR			✔
Emergency Producer, [1014H] and [1015H]		✔	✔
Emergency Consumer, [1028H]			✔
Layer-Setting Services		Regular and FastScan	Regular
SDO Client		With optional Manager Add-On	✔
Implements NMT Master		With optional Manager Add-On	✔

Common CAN driver interface		✔	✔
Flexible integration into RTOS		✔	✔
Maximum number of PDOs	8	1024	1024
Maximum size of process image storing all data that can be mapped to PDOs (in bytes)	254	65,534	65,534
Smallest timer tick resolution available	1ms	1ms	100us
Minimal SDO Manager			✔
DS447 Car Add On Devices Support		With optional DS447 Add-On	
Error and Emergency History, [1003H]		✔	

Microport®
標高電子股份有限公司
Microport Computer Electronics Inc.
Tel/06-2753783 Fax/06-2375031
http://www.microporttaiwan.com