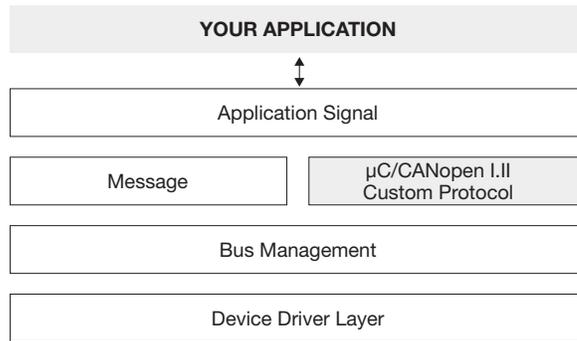# Micriµm

µC/USB CAN™ CAN-Bus Module

DESCRIPTION

µC/CAN is a CAN protocol framework that enables easy and clean implementation of CAN communication paths. µC/CAN is a source code library optimized for speed, flexibility and size and built with high portability and clean documentation in mind.

µC/CAN reduces development efforts to embed CAN. Developers need only an understanding of signals, messages and bus configurations. Different abstraction layers can be used independently.

µC/CAN can communicate via one or more CAN buses simultaneously. The bus management layer organizes the CAN buses and distributes messages to different device drivers. The device driver layer buffers the CAN messages to be sent and received. All hardware dependencies are capsulated in this layer. An embedded target can also have multiple, different CAN controllers.

A validation suite provides all documentation necessary to support the use of µC/CAN in safety-critical systems. in

```
                  YOUR APPLICATION
                          ↕
                   Application Signal

    Message              µC/CANopen I.II
                         Custom Protocol

                   Bus Management

                   Device Driver Layer
```

APPLICATION SIGNAL LAYER

This layer gives access to logical signals, located in one or more CAN messages. Changes in signals can be checked, and predefined actions executed.

MESSAGE LAYER

The message layer organizes all information necessary for sending and receiving CAN messages. Identifier, data length code (DLC) and data (and linked signals) are bundled and dedicated to one or more buses.

APPLICATIONS

- Industrial
- Automotive
- Aerospace

FEATURES

- Hardware independent
- Signal Scalability – Between 1 bit to 4 bytes
- Sufficiently robust to meet rigorous safety-critical system requirements

PROTOCOLS

CANopen Stack

The CANopen stack is a scalable solution for limited-resource embedded systems. The stack is delivered in ANSI-C source code and can be compiled with any ANSI-C compliant compiler. There are three variants of the stack depending on target application, and a windows application for the automatic source code generation of the object directory and EDS files.

CANopen Sensor Slave

CANopen Sensor Slave is designed for CANopen protocol handling on such minimal slave systems as intelligent sensors, digital I/O modules, etc. This variant supports the following features:

- SDO Server support
- SDO Expedited Transfers
- PDO Producer and Consumer with static mapping and communication parameters
- NMT Slave
- NMT Heartbeat Producer
- EMCY Producer
- LSS Slave
- Object Stings with up to 4 characters

Micriµm

µC/USB CAN™ CAN-Bus Module

CANopen Slave

CANopen Slave is full featured for more complex slave applications. This variant includes the features of CANopen Sensor Slave with the following additions:

- SDO Client support
- SDO Normal and block transfers
- Dynamic PDO mapping- and communications parameters
- EMCY History
- Dynamic Object Directory

CANopen Master

CANopen Master is full featured for network management applications. This variant includes the features of CANopen Slave with the following additions:

- NMT Master
- NMT Heartbeat Consumer
- LSS Master
- Object strings with unlimited length
- Object domains supported

CANopen Config

This is a windows application for configuration of the object directory and the EDS-file of all CANopen stack variants. The tool allows the template-based generation of ANSI-C code and EDS files.

µC/CAN

Maximum ROM Footprint (Unscaled)	7.5 kB (on MPC565) to 11.7 kB (on MPC5200B)
RAM usage	Configuration dependent: Per Signal • 29 bytes (unscaled without padding) • 11 bytes (scaled without padding) Per Message • 10 bytes (unscaled without padding) • 3 bytes (scaled without padding)

µC/CANopen Small Slave

ROM Footprint (117 µC/CAN signals; 20 Parameters; 150 Object Directory Entries; no PDOs: LSS disabled)	ROM / RAM: µC/CANopen: 13kB / 6 kB Board Support Package: 700B /12 B µC/CANopen Config: 4.5 kB / 1 kB µC/CAN Config: 2 kB µC/CAN: 7.5 kB

DRIVERS

Device drivers are available to support different CAN controllers. Drivers added on a regular basis. Please visit the web site for the latest driver list.

Overview

Component	Sensor Slave	Slave	Master
SDO Server	Up to 127	Up to 127	Up to 127
SDO Client	No	No	Up to 127
SDO Exp. Transfer	Yes	Yes	Yes
SDO Normal Transfer	No	Yes	Yes
SDO Block Transfer	No	Yes	Yes
PDO Producer	Up to 512	Up to 512	Up to 512
PDO Consumer	Up to 512	Up to 512	Up to 512
PDO Mapping	Static	Dynamic	Dynamic
PDO Com. Parameter	Static	Dynamic	Dynamic
PDO Remote Transfer	No	No	No
NMT Slave	Yes	Yes	Yes
NMT Master	No	No	Yes
NMT Heartbeat Producer	Yes	Yes	Yes
NMT Heartbeat Consumer	No	No	Yes
NMT Node Guarding	No	No	No
EMCY Producer	Yes	Yes	Yes
EMCY Consumer	No	No	Yes
EMCY History Size	1	Up to 250	Up to 250
LSS Slave	Yes	Yes	Yes
LSS Master	No	No	Yes
OD Size	Unlimited		Unlimited
OD Entries	Static	Static	Dynamic
OD String Length	Up to 4	Up to 4	Unlimited
OD Domains	No	Yes	Yes

For pricing, delivery, and ordering information, please contact Micrium at +1 954-217-2037, or visit Micrium's website at: **www.micrium.com**.

Micriµm

For the way Engineers work

+1 954 217 2036 | email: sales@micrium.com | www.micrium.com