

CANopen Library Driver Packages

Overview

The *CANopen Driver Packages* (DP) provide access to the hardware of the *CANopen Library* of **port**. The Driver Packages are the keys for the usage of the target system independent of CANopen Library. They are designed for target systems with and without operation systems.

Description

The *CANopen Driver Packages* provide all necessary functions for

- initialization of the hardware
- controlling the CAN Controller
- message filtering
- message buffering and
- provision of a timer period

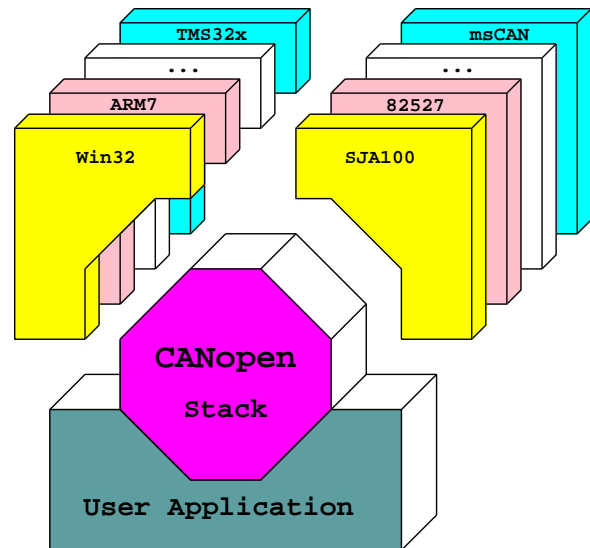
by means of a well defined interface. A *CANopen Driver Package* is comprised from a CPU- (*DP/CPU*), a CAN- (*DP/CAN*) and where applicable a driver for an operating system (*DP/OS*). These drivers can be used in any combination.

The following table shows hardware combinations of CPU and CAN drivers that are often used by our customers.

	microcontroller family																	
	Microcontroller ANSI-C driver	ANSI-C driver	8051 Family	Infineon C505	Atmel T89C51	Atmel AVR	Infineon C16xCy	Infineon XC16x	Intel 80x86	Microchip dsPIC	PowerPC	Fujitsu 16Lx	Fujitsu 16FX	TMS320Cxx	Renesas M16C	Freescale 56F80XX	ARM 7/9	F.HC08/CS12
CAN Controller																		
Integrated CAN																		
SJA1000																		
Intel 82527/CC770																		
TouCAN/FlexCan																		
CANary																		
msCAN																		
Bosch C_CAN																		
Infineon TwinCAN/MultiCAN																		
Microchip ECAN																		
Microchip MCP2515																		

All drivers are available as documented ANSI-C source code. This allows easy and quick adoption to system specific hardware and trouble-free compilation with all ANSI-C compliant ANSI-C compilers.

The comprehensive documentation ensures that the driver can be easily ported to another hardware platform by the user.



Configuration possibilities

The driver provides access to the CAN controller by several means:

- Memory Mapped Mode (addressing the CAN controller via the address space of the CPU)
- I/O Mapped Mode (addressing the CAN controller via the I/O address space of the CPU)
- Latched Mode (addressing the CAN controller via address latches)

The used access method for accessing the CAN controller register depends heavily on the used hardware and can be adopted by means of access macros.

The driver for the FullCAN controller types provide the usage of

- FullCAN-Mode filtering CAN messages in hardware for all channels available (1 channel for sending or receiving)
- FullCAN mode with 1 transmit channel one transmit channel for all transmit messages all other channels are configured as receiving channels using the hardware filter
- BasicCAN mode 1 transmit channel for all CAN messages 1 receiving channel for all CAN messages

The last two modes provide the possibility to use devices with FullCAN controller, that support more CAN objects as are provided from the hardware of the CAN controller.

Many drivers do contain code for

- step by step initial operation of the driver
- quick error detection
- extensions that have been added due to customer requests
- 29 Bit extended CAN Frames, besides the 11 Bit Base-Frame format

that can be enabled via appropriate compiler defines.

Besides the drivers for processors with multiple CAN controller there are several other drivers that support configuration for multi-line usage with the *CANopen Library*. Thus there is a wide range of drivers that can be used for multi-line devices.

Please note that there are possible hardware constraints especially with older CAN controller.

Configuration of the driver is done with the *CANopen Design Tool*, which is enclosed to the *CANopen Library* as light version.

Resources

All drivers normally transmit and receive interrupt driven and therefore require the integration into the interrupt system of the device. It also can be used in polling mode for special-purpose applications.

The time base for the *CANopen Library* can either be provided by the *CANopen Library* itself by using a timer or the timer interrupt function is called cyclically by another system timer.

Order Information

The *CANopen Driver Packages* are available as:

- reviewed hardware/system driver (DP 0565/xx) for typical hardware configurations
- freely selectable hardware driver CAN controller combinations (DP/CPU 0566/xx, DP/CAN 0567/xx)

We recommend the purchase order of pre-configured driver packages that in addition to the CPU and CAN driver contain ready to use examples. These have been prepared with project files for the compilers and configurations we use.

For application of the *CANopen Library* with operating systems like Windows™ or LINUX™ *port* provides drivers for active and passive PC cards, parallel dongles, PCMCIA cards, USB interface and external Ethernet to CAN converter (*EtherCAN*).

Predefined Driver Packages

Order Code	Name of System
0565/06	DP C505C microMODUL-505C/plain (<i>Phytec</i>) (on Request/8051 + 82527) ¹
0565/05	DP C515C miniMODUL-515C/plain (<i>Phytec</i>) (on Request/8051 + 82527) ¹
0565/03	DP C167CS for miniMODUL-167C/plain (<i>Phytec</i>) (C166 + 82527) ¹
0565/34	DP Infineon XC164CS (XC166 + TwinCAN)
0565/56	DP Infineon XMC4500 (XMC4500 + MultiCAN)
0565/69	DP Infineon XE164 (XE166 + MultiCAN)
0565/10	DP Atmel T89C51CC01/02/03 (8051 + CANary)
0565/35	DP Atmel AT90CAN32/128 (AVR + CANary)
0565/47	DP Atmel AT91SAM7A3/SAM7X(C) (ARM7 with internal CAN)
0565/83	DP Atmel AT91SAM9263 (32-bit AVR with internal CAN)
0565/94	DP Atmel AT32UC3CO512 (32-bit AVR with internal CAN)
0565/112	DP Atmel ATSAM4E (32-bit ARM Cortex-M4 with internal CAN)
0565/115	DP Atmel ATSAM4E-μC/OS (32-bit ARM Cortex-M4 with internal CAN) under μC/OS
0565/25	DP STMicroelectronics STR730 (with C_CAN)
0565/79	DP STMicroelectronics STR911 (with C_CAN)
0565/117	DP STMicroelectronics STM32F0x (with bxCAN)
0565/72	DP STMicroelectronics STM32F10x (with bxCAN)
0565/107	DP STMicroelectronics STM32F107 (with bxCAN)
0565/98	DP STMicroelectronics STM32F2xx (with bxCAN)
0565/110	DP STMicroelectronics STM32F3xx (with bxCAN)
0565/106	DP STMicroelectronics STM32F4xx (with bxCAN)
0565/33	DP SiLabs C8051F040 (8051 + C_CAN)
0565/80	DP SiLabs C8051F50x (8051 + C_CAN)
0565/07	DP Freescale HCS12(X) for MC9S12(X) family (HCS12(X) + msCAN)

Order Code	Name of System
0565/67	DP Freescale MC56F8037 (56800E + msCAN)
0565/08	DP Freescale DSP56F807 (DSP56800 + msCAN)
0565/38	DP Freescale MPC565/566 (MPC5xx + TouCAN) ¹
0565/36	DP Coldfire 5282/5223 (with internal FlexCAN) ¹
0565/48	DP Freescale MC56F8323 (56800E + FlexCAN) ¹
0565/118	DP Freescale MC56F827xx (MSCAN) ¹
0565/30	DP Spansion MB90F543 (with internal Spansion CAN)
0565/66	DP Spansion MB96F348 (with C_CAN)
0565/78	DP Spansion MB91F467 (with C_CAN)
0565/109	DP Spansion MB9BF524K (with C_CAN)
0565/31	DP TI TMS320LF2407 (TMS320 + internal CAN)
0565/32	DP TI TMS320F2808/2812 (TMS320 + eCAN)
0565/57	DP TI TMS320F28335 (TMS320 + eCAN)
0565/26	DP TI TMS470R1B1M (ARM7 with HECC)
0565/101	DP TI Sitara AM335x (ARM Cortex A8 with DCAN)
0565/103	DP TI Concerto FM28M35X (ARM Cortex M3 with DCAN)
0565/19	DP Microchip dsPIC33F256 (dsPIC33F with ECAN)
0565/28	DP Microchip PIC24H (PIC24H with ECAN)
0565/37	DP Microchip dsPIC30F6011/12/13/14 (dsPIC30F60xx with CAN)
0565/99	DP Microchip PIC32MX (PIC32MX795 with ECAN)
0565/43	DP Microchip PIC18F2680 (PIC18F2x8x with ECAN TM)
0565/44	DP Microchip MCP2515 (with SPI-Interface, example for XC164)
0565/95	DP NXP LPC1768 (with internal CAN)
0565/39	DP NXP LPC2129 (with internal CAN)
0565/85	DP NXP LPC2368 (with internal CAN)
0565/74	DP NXP LPC2468 (with internal CAN)
0565/46	DP Beck IPC@Chip 1x3 (with internal CAN)
0565/49	DP RENESAS PD70F3231 (V850/ES-FE2 with aFCAN) ¹

Order Code	Name of System
0565/55	DP RENESAS PD70F3577 (V850/E2-FG4L with aFCAN) ¹
0565/65	DP RENESAS PD70F3377 (V850/ES-Fx3 with aFCAN) ¹
0565/93	DP RENESAS PD70F3476 (V850/E-SJ3 with aFCAN) ¹
0565/113	DP RENESAS PD70F4022 (V850E2/ML4 with FCAN) ¹
0565/21	DP RENESAS M16C29 (with internal CAN)
0565/22	DP RENESAS M32C/87 (with internal CAN)
0565/59	DP RENESAS RL78F13 (with internal CAN)
0565/96	DP RENESAS RX63N (with internal CAN)
0565/108	DP RENESAS RX62N (with internal CAN)
0565/114	DP RENESAS R-IN32M3 (with FCN-CAN)
0565/116	DP RENESAS RX64M (with internal CAN)
0565/119	DP RENESAS RL78F14 (with internal CAN)
0565/73	DP Luminary LM3S2965 (Cortex-M3 with C_CAN)
0565/92	DP Luminary LM3S5B91 (Cortex-M3 with C_CAN)
0565/24	DP netX (with internal CAN)
0565/13	DP CPC-WindowsTM CPC CANopen driver for Windows TM (<i>EMS Wunsche</i>) e.g. useable for passive CPC-PCI card, CPC-USB Interface
0565/15	DP Peak-light WindowsTM CANopen driver for PCAN-light (<i>PEAK-Service GmbH</i>), (on request)
0565/29	DP Arcom Boards WindowsTM-CE with AIM104-CAN-Module
0565/50	DP can4linux CANopen driver for can4linux compati- ble hardware under LINUX TM
0565/14	DP CPC-Linux CPC CANopen driver for LINUX TM , (<i>EMS Wunsche</i>) useable for passive CPC-PCI card, CPC-USB Interface
0565/51	DP CAN232 Linux Lawicel CAN232 driver for LINUX TM
0565/52	DP CAN232 WindowsTM Lawicel CAN232 driver for Windows TM
0565/53	DP SocketCAN SocketCAN driver for LINUX TM
0565/63	DP Kvaser CANopen driver for Windows TM useable for cards with Kvaser Interface

Order Code	Name of System
0565/75	DP Kvaser/Kontron Kvaser CANlib on Kontron Board for Windows™-CE
0565/70	DP horch LINUX™ CANopen driver communicate with the standard horch server by using TCP/IP
0565/84	DP horch Windows™ CANopen driver communicate with the standard horch server by using TCP/IP
0565/87	DP SCIOPTA with FlexCAN
0565/88	DP CAN-IP von Xilinx XPS CAN - Xilinx Part Number EF-DI-CAN-XC-SITE and Xilinx Microblaze v7.20.c operating system Micrium µC/OS-II
0565/90	DP generic hardware independent CANopen driver for own driver adaptations

Adaptable Drivers CAN driver

Order Code	CAN controller type
0566/01	DP/CAN Philips SJA1000
0566/03	DP/CAN Intel 82527¹
0566/04	DP/CAN Siemens SAE 80C90/91 (on request)
0566/06	DP/CAN Freescale TouCAN¹
0566/07	DP/CAN Freescale msCAN msCAN8, msCAN12, msCAN12/2
0566/08	DP/CAN Spansion CAN (16LX serie)
0566/09	DP/CAN Atmel CANary
0566/11	DP/CAN Mitsubishi M16C/6N
0566/13	DP/CAN TI eCAN (e.g. TMS320F28xx serie)
0566/14	DP/CAN Bosch C_CAN (e.g. Silabs C8051F040, Spansion MB91xxx, ST STR730)
0566/15	DP/CAN Infineon TwinCAN
0566/16	DP/CAN Freescale FlexCAN¹
0566/19	DP/CAN Philips LPC21xx
0566/20	DP/CAN Microchip MCP2515
0566/22	DP/CAN Atmel AT91SAM7A3/SAM7X(C)
0566/23	DP/CAN NEC aFCAN¹
0566/24	DP/CAN TI HECC (e.g. TMS470 serie)

CPU driver

Order Code	CPU type
0567/01	DP/CPU Freescale MC683xx (on request)
0567/02	DP/CPU Freescale MC680x0 (on request)
0567/05	DP/CPU Infineon C16x
0567/06	DP/CPU TI TMS320F2000
0567/07	DP/CPU Spansion MB90F54x,49x
0567/08	DP/CPU Freescale MPC823e
0567/09	DP/CPU Atmel T89C51CC0x
0567/10	DP/CPU Dallas 80C390/400 (on request)
0567/11	DP/CPU Mitsubishi M16C/6N
0567/12	DP/CPU Freescale DSP 56F800
0567/13	DP/CPU Freescale HCS12(X)(X)
0567/14	DP/CPU SiLabs 8051
0567/16	DP/CPU Infineon XC166
0567/17	DP/CPU MicroChip dsPIC30F60xx
0567/18	DP/CPU Atmel AT91SAM7A2
0567/19	DP/CPU Philips LPC21xx
0567/20	DP/CPU Freescale Coldfire
0567/21	DP/CPU Atmel AVR AT90CAN128
0567/22	DP/CPU Microchip PIC18F2x8x
0567/23	DP/CPU Atmel AT91SAM7A3
0567/24	DP/CPU Freescale 56800E
0567/25	DP/CPU NEC PD70F3231
0567/26	DP/CPU TI TMS470B1M
0567/27	DP/CPU Microchip dsPIC33F
0567/28	DP/CPU Microchip PIC24H

The supported number of drivers is constantly extended. If it happens to happen that your hardware is not yet listed we would be happy to provide support for developing your own driver for the *CANopen Library*.

¹) hardware dependent deviations from standard functionality



hkaco.com



关注我们

需要详细信息? 请通过sales@hkaco.com联系我们 | 电话: 400-999-3848
办事处: 广州 | 北京 | 上海 | 深圳 | 西安 | 武汉 | 成都 | 沈阳 | 香港 | 台湾 | 美国