User Manual

CANgine

COP

Edition 6 / March 2008
PRELIMINARY

ess
embedded
systems
solutions

Version history

Edition	Date	State
6	Mar 2008	OD entries 2000/2 and 2000/3 exchanged
5	Feb 2008	Modifications in object dictionary
4	Jul 2007	Additions, corrections
3	Jul 2007	first edition in English based on German edition A3

ESS Embedded Systems Solutions GmbH
Industriestr. 15
D-76829 Landau
(49) 6341 3487-0
(49) 6341 3487-29
info@ESSolutions.de
www.ESSolutions.de

www.CANgine.com

Related documents

CiA Draft Standard 4.01

Special characters

[CR] Enter key or ASCII character code carriage return 0x0D
[LF] ASCII character code linefeed 0x0A
[BELL] ASCII character code bell 0x07

Contents

1. Introduction 6
2. Installation 7
3. Configuration 8
3.1 ? Command – Show parameter settings 8
3.2 A Command – Autostart 9
3.3 H Command – Heartbeat Zeit 9
3.4 IT Command – Identifier Transmit 9
3.5 IR Command – Identifier Receive 10
3.6 M Command – operating mode 10
3.7 N Command – Node ID 10
3.8 P Command – Protocol 10
3.9 R Command – Restart after configuration 11
3.10 S Command 11
3.11 T Command – Timeout 11
3.12 U Command – Uart Baudrate 12
3.13 V Command – Version information 12
4. Normal operating mode 12
4.1 CAN Layer 2 mode 12
4.2 CANopen mode 12
4.3 Object Dictionary 13
4.3.1 Device Type (1000) 13
4.3.2 Error Register (1001) 13
4.3.3 Manufacturer Status Register (1002) 14
4.3.4 Predefined Error Field (1003) 14
4.3.5 Manufacturer Device Information (1008 to 100A) 14
4.3.6 Producer Heartbeat Time (1017) 14
4.3.7 Identity Object (1018) 14
4.3.8 SDO 1 Server Parameter (1200) 14
4.3.9 RPDO 1 Communication Parameter (1400) 15
4.3.10 RPDO 1 Mapping Parameter (1600) 15
4.3.11 TPDO 1 Communication Parameter (1800) 15
4.3.12 TPDO 1 Mapping Parameter (1A00) 15
4.3.13 CANopen Parameters (2000) 15
4.3.14 RS232 Parameters (2001) 15
5. Data handling and transmission 16
5.1 Performance 17
5.2 Status byte 17
5.3 Error signaling 18
5.3.1 Error Bits in Error Register OD 1001 18
5.3.2 Error Codes in OD 1003/1 to 1003/4 18
6. Display 18
6.1 Green Run LED 18
6.2 Red Error LED 18
7. Problems and Help 19

ESS Embedded Systems Solutions GmbH
Industriestrasse 15
D-76829 Landau

1. Introduction

Thank you for choosing a product of our CANgine product family. The CANgine family is based on high performance 8 bit microcontrollers with integrated full CAN interface and flash memory. With these controllers we are able to build extremely small but powerful CAN units.

In its standard case, CANgine FMS only measures 53 x 34 x 16 mm³ (2.08 x 1.34 x 0.63 inch³). If this does not fit for some applications CANgine FMS can be delivered in other cases or without case in customer specific variants. Due to the modular concept of CANgine in hardware and firmware this is possible even at lower production volumes. Email or call our sales department if you have special requirements.

The serial link of CANgine FMS supports baud rates from 2.400 up to 115.200 bps. The CAN baud rate is fixed to 250 kbps as defined by the FMS standard. Setting into operation and troubleshooting is made easy due to the two LEDs.

CANgine COP connects RS232 based peripheral devices in CAN or CANopen environments if serial lines are not present or not desired. Due to ist small form factor CANgine COP can be used in nearly all applications.

With a few configuration steps CANgine COP can be adopted to the actual application. CANgine works in one of two operation modes: CAN mode or CANopen (COP) mode.

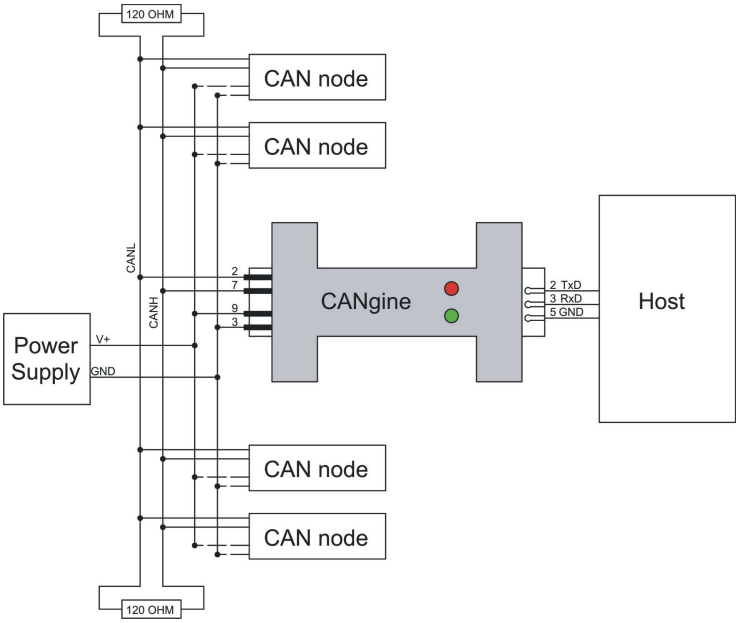
CAN

In CAN mode CANopen features are disabled. The serial byte stream is packed into CAN layer 2 frames and sent on the bus and vice versa. The CAN identifiers used are configurable.

COP

In COP mode CANgine COP behaves like a CANopen slave node with one receive PDO (RPDO) and one transmit PDO (TPDO).

2. Installation



The picture shows how to connect CANgine COP to the power supply, the CAN bus and the host. CANgine is powered via pin 9 and pin 3 of the CAN connector as recommended by CiA[1]. The maximum supply voltage is 30 V. Applying higher voltages will lead to damages. Terminate the CAN bus with a terminating resistor of 120 Ohm on each side.

After switching power on, the red LED flashes one time while CANgine COP initializes. If a severe error is detected while initializing the hardware the red LED stays lit. After successful initialization the green LED flickers for a few seconds. If in this time the string "conf[CR]" is received, CANgine COP switches to configuration mode instead of starting in normal operating mode. Read chapter 3 for details on configuring the device.

Otherwise CANgine COP switches in normal operating mode which is either CAN or COP (CANopen) mode. See chapter 4.1 for details of CAN and chapter 4.2 for details of COP mode.

The factory settings for the serial link are:

- Baud rate 19200
- 8 data bits
- no parity
- 1 stop bit
- no handshake

[1] CANinAutomation (www.can-cia.org)

ESS Embedded Systems Solutions GmbH
Industriestrasse 15
D-76829 Landau

3. Configuration

CANgine COP switches to configuration mode if after reset the string "conf[CR]" is received while the green LED flickers. This seril input is always checked with a baud rate of 19,200 baud independent from the setting of baud rate in operation mode. While CANgien COP is in configuration mode the green LED flickers.

Configuration mode is used to set the operating parameters. The table below shows the different parameters and commands. Depending on the parameterized operation mode not all of the commands are active. All parameters are stored in non volatile memory (EEPROM) and therefore valid after power off.

Befehl	Bedeutung	active in M=	Factory default
?[CR]	Show parameter settings	0, 1	na
An[CR]	Autostart off (n=0) or on (n=1)	0, 1	0
Hn[CR]	Set heartbeat time (ms)	1	1000
ITn[CR]	Set Identifier for transmit frame	0, 1	0x1C0
IRn[CR]	Set Identifier for receive frame	0, 1	0x240
Mn[CR]	Set operating mode to CAN (n=0) or CANopen (n=1)	0, 1	0
Nn[CR]	Set CANopen node ID (n = node ID in hex)	1	0x40
Pn[CR]	Set protocol to Std, 11 Bit ID (n=0) or to ext, 29 Bit ID (n=1)	0	0
R[CR]	Restart after configuration	0, 1	na
Sn[CR]	Set CAN baud rate	0, 1	5 (250 kb/s)
Tn[CR]	Set timeout value for transmission of less than 7 characters (ms)	0, 1	10
Un[CR]	Set RS232 baud rate	0, 1	3 (19200)
V[CR]	Show versions-Information	0, 1	na

If the operating mode is switched or if a new node ID is set some of the parameters are set to their default values. In CANopen mode only the standard protocol with 11 bit identifier is possible.

Switching the operating mode from CAN to CANopen sets the protocol to standard, IT to 0x180 + Node ID and RI to 0x200 + Node ID.

Entering a new Node ID in CANopen mode sets receive and transmit identifier to their standard values IT = 0x180 + Node ID and RI = 0x200 + Node ID. After this the values can now be changed If neccessary.

3.1 ? Command – Show parameter settings

Shows the actual operating parameters.

Format:
?[CR]
Answer:
Parameter display
Error information on syntax errors

The output format in operating mode CAN (M0) is shown below. The values shown are the factory default settings. The default operating mode is CAN (M0).

```
Configuration settings in CAN mode: (M0)
IT (ID Transmit)      0x1C0
IR (ID Receive)       0x240
```

```
P  (Protocol)         0 (Std)
S  (CAN baud rate)    5 (250k)
T  (Timeout)          10 ms
U  (RS232 baud rate) 3 (19.2k)
```

The output format in operating mode COP (M1) is shown below. The values shown are the factory default settings.

```
Configuration settings in CANopen mode: (M1)
A  (Autostart)        0 (off)
H  (Heartbeat)        1000 ms
N  (Node ID)          0x40 (64)
IT (TPDO)             0x1C0
IR (RPDO)             0x240
S  (CAN baud rate)    5 (250k)
T  (Timeout)          10 ms
U  (RS232 baud rate) 3 (19.2k)
```

3.2 A Command – Autostart

This parameter is only for the operating mode COP. In setting A1 (on) the CANopen slave does not wait for a CANopen master to start the node but switches automatically from pre-operational to operational after initialization.

Format:

An[CR]

n = {0 | 1}

Answer:

[CR][LF] if parameter was accepted and stored
Error message on wrong parameter format

3.3 H Command – Heartbeat Zeit

The H command sets the heartbeat time in ms. If the time 0 is selected, the heartbeat function is switched off.

Format:

Hn[CR]

n = {0..65535}

Answer:

[CR][LF] if parameter was accepted and stored
Error message on syntax error or invalid parameter

3.4 IT Command – Identifier Transmit

The IT command sets the identifier used for the CAN transmit frame.

Depending on the active protocol (std or ext) the parameter has a different range.

Format:

ITn[CR]

n = {0..32767} or {0x0..0x7FF} if P=0
n = {0..1073741823} or {0x0..0x3FFFFFFF} if P=1

Answer:

[CR][LF] if parameter was accepted and stored
Error message on syntax error or invalid parameter

ESS Embedded Systems Solutions GmbH
Industriestrasse 15
D-76829 Landau

3.5 IR Command – Identifier Receive

The IR command sets the identifier used for the CAN receive frames.

Depending on the active protocol (std or ext) the parameter has a different range.

Format:

IRn[CR]

n = {0..32767} or {0x0..0x7FF} if P=0

n = {0..1073741823} or {0x0..0x3FFFFFFF} if P=1

Answer:

[CR][LF] if parameter was accepted and stored

Error message on syntax error or invalid parameter

3.6 M Command – operating mode

The M command sets the operating mode to either CAN (n=0) or COP (n=1). If operating mode is switched from CAN to COP the two identifiers are set according to the actual node ID:

PDO transmit = NodeID + 0x180
PDO receive = NodeID + 0x200

In operating mode COP only the standard protocol with 11 bit identifiers is allowed.

Format:

Mn[CR]

n = {0 | 1}

Answer:

[CR][LF] if parameter was accepted and stored

Error message on syntax error or invalid parameter

3.7 N Command – Node ID

The N command sets the node ID. Setting the node ID also sets the two identifiers to their CANopen default values:

PDO transmit = NodeID + 0x180
PDO receive = NodeID + 0x200

Format:

Nn[CR]

n = {1..127}

Answer:

[CR][LF] if parameter was accepted and stored

Error message on syntax error or invalid parameter

3.8 P Command – Protocol

The P command sets the protocol either to standard (11 bit identifier, parameter 's') or to extended (29 bit identifier, parameter 'e'). This command is only used if operating mode CAN is active.

Format:
Pc[CR]
c = {S | s | E | e }
Answer:
[CR][LF] if parameter was accepted and stored
Error message on syntax error or invalid parameter

3.9 R Command – Restart after configuration

With the R command the configuration mode is terminated and the CANgine COP is started with the stored parameters. Keep in mind that the serial baud rate in configuration mode is always 19200 baud while the baud rate after the restart is set according to the stored value (see chapter 3.12, U Command).

Format:
R[CR]
Answer:
if command was executed → no answer, CANgine COP is now in operating mode
Error message on syntax error

3.10 S Command

The S command sets the baud rate of the CAN bus to one of the CiA recommended values:

S0	10 kBit
S1	20 kBit
S2	50 kBit
S3	100 kBit
S4	125 kBit
S5	250 kBit
S6	500 kBit
S7	800 kBit
S8	1 MBit

Format:
Sn[CR]

Answer:
[CR][LF] if parameter was accepted and stored
Error message on syntax error or invalid parameter

3.11 T Command – Timeout

The T command sets the timeout value. The unit is ms (milliseconds). The timeout value is used to start transmission of a CAN frame after receiving less than 7 characters on the serial link.

Format:
Cn[CR]
n = {1..65535}
Answer:
[CR][LF] if parameter was accepted and stored
Error message on syntax error or invalid parameter

ESS Embedded Systems Solutions GmbH
Industriestrasse 15
D-76829 Landau

3.12 U Command – Uart Baudrate

The U command sets the baud rate of the serial link. This baud rate is only used in th enormal operatin modes CAN or COP. In the configuration mode CANgine COP always communicates with 19,200 baud.

U0	115200
U1	57600
U2	38400
U3	19200
U4	9600
U5	4800
U6	2400

Format:
Un[CR]
Answer:
[CR][LF] if parameter was accepted and stored
Error message on syntax error or invalid parameter

3.13 V Command – Version information

The V command shows the actual version aof CANgine COP.

Format:
V[CR]
Answer:
CANgine-COP Vhhff[CR][LF]
with hh = hardware version and ff = firmware version
Error message on syntax error

4. Normal operating mode

CANgine COP reaches normal operating mode either after reset when the 'conf' string is not received in the flickering time or from the configuration mode if the R command is given.

4.1 CAN Layer 2 mode

If operating mode CAN (M0) is stored in the parameter data, CANgine COP switches to normal operating mode after initialization. In this mode all characters received are packed into CAN Layer 2 frames with the identifier stored with IT command and sent to the bus. More details on these functions are found in chapter 5.

4.2 CANopen mode

If operating mode COP (M1) is stored in the parameter data, the behavior after start of normal mode depends from the parameter Autostart (see chapter 3.2). With Autostart off (A0) CANgine COP acts like a standard CANopen slave and stays in mode pre-operational. A CANopen master must bring CANgine COP to operational by a NMT command. With Autostart on (A1) CANgine COP doesn't wait for a CANopen master but switches automatically from pre-operational to operational. The functions used for sending and receiving characters at the serial link is the same in CAN and CANopen mode. Details on these functions are found in chapter 5.

4.3 Object Dictionary

Index	Sub	Name	Type	Acc	Min.	Wert	Max.	M/O
1000	0	Device Type	U32	ro		0x0000.0000		M
1001	0	Error Register	U8	ro		0x00		M
1002	0	Manufacturer status register	U32	ro		0x00		O
1003	0	Number of errors	U8	rw		0x00		O
	1	newest error number	U32	ro		0x0000.0000		O
	2	..	U32	ro		0x0000.0000		O
	3	..	U32	ro		0x0000.0000		O
	4	oldest error number	U32	ro		0x0000.0000		O
1008	0	Manufacturer device name	VIS_STR	const		CANgine COP		O
1009	0	Manufacturer hardware versions	VIS_STR	const		HW version		O
100A	0	Manufacturer software version	VIS_STR	const		FW version		O
1017	0	Producer heartbeat time	U16	rw	0x0000	0x3E8	0xFFFF	O
1018		Identity Object	IDENTITY					M
	0	Highest index supported	U8	ro		0x04		M
	1	Vendor ID	U32	ro		0x0000.01E2		M
	2	Product Code	U32	ro		0x0201.0101		O
	3	Revision Number	U32	ro		0x0000.hhff[1]		O
	4	Serial Number[2]	U32	ro		0xFFFF.FFFF		O
1200		1st SDO server parameter	SDO_PAR					M
	0	Highest index supported	U8	ro		0x02		M
	1	COB-ID client -> server	U32	ro		NID+0x0600		M
	2	COB-ID server -> client	U32	ro		NID+0x0580		M
1400		1st RPDO communication parameter	PDO_COM					M
	0	Highest index supported	U8	ro		0x02		M
	1	COB-ID used by PDO	U32	rw		NID+0x0200		M
	2	Transmission type	U8	ro		0xFF		M
1600		1st RPDO mapping parameter	PDO_MAP					M
	0	Number of mapped objects	U8	ro		0x00		M
1800		1st TPDO communication parameter						M
	0	Highest index supported	U8	ro		0x02		M
	1	COB-ID used by PDO	U32	rw		NID+0x0180		M
	2	Transmission type	U8	ro		0xFF		M
1A00		1st TPDO mapping parameter						M
	0	Number of mapped objects	U8	ro		0x00		M
2000		CANopen Parameters						O
	0	Highest index supported	U8	ro		0x03		O
	1	CAN Baudrate	U8	ro	0x00	0x05	0x08	O
	2	CANopen Node ID	U8	rw	0x01	NID	0x7F	O
	3	CANopen autostart	U8	rw	0x00	0x00	0x01	O
2001		RS232 Parameters						O
	0	Highest index supported	U8	ro		0x01		O
	1	RS232 Baudrate	U8	rw	0x00	0x03	0x06	O
2002	0	Timeout to send (0,1 ms units)	U16	rw	0x01	0x0A	0xFFFF	O

NID: actual Node ID (default = 0x40); **ro:** read only; **rw:** read and write allowed; **const:** constant value

4.3.1 Device Type (1000)

As CANgine COP doesn't follow a standardized device profile, this entry contains 0.

4.3.2 Error Register (1001)

Bit	Meaning
0	generic error
1	0
2	0
3	0
4	communication error (overrun)

[1] hh ist he two-digit hardware version, ff ist he two-digit firmware version
[2] Actually not supported

ESS Embedded Systems Solutions GmbH
Industriestrasse 15
D-76829 Landau

Bit	Meaning
5	0
6	0
7	0

In any error condition Bit 0 is set. If the error disappears bit 0 in the error register is reset.

4.3.3 Manufacturer Status Register (1002)

Bit	Meaning	Bit	Meaning
0	reserved	8	reserved
1	serial transmitter overflow A CAN object with characters for serial output was received but the serial transmit buffer had no more entries	9	CAN receive error detected CAN receiver detected a stuff, crc, form, ack error
2	NMT receive overflow A NMT frame was received before the previous NMT frame was executed	10	reserved
3	SDO receive overflow A SDO frame was received before the previous SDO frame was executed	11	reserved
4	PDO transmit overflow Characters received from the serial link but previous TPDO not yet sended	12	reserved
5	NMT transmit overflow Attempt to transmit a NMT frame while the previous NMT frame not yet sended	13	reserved
6	SDO transmit overflow Attempt to transmit a NMT frame while the previous NMT frame not yet sended	14	reserved
7	CAN error passive state CAN controller has entered the error passive state	15	reserved

All the reserved bits and bit 16 to 31 read as zero.

4.3.4 Predefined Error Field (1003)

This entry shows the last four errors occurred as described in the DS 301. Writing a zero to sub-index 1 clears all four error entries.

4.3.5 Manufacturer Device Information (1008 to 100A)

The device name shows the string "CANgine COP", hardware and software version entries contains a string similar "4.2" in which the digits before the decimal point shows a version number, the digits after the decimal point shows a revision number.

4.3.6 Producer Heartbeat Time (1017)

The cycle time in milliseconds for the generation of heartbeat messages on the CAN bus. If zero no heartbeat messages are generated.

4.3.7 Identity Object (1018)

Contains

- Vendor ID of ESS,
- Product code,
- Version number,
- 0xFFFFFFFF as serial number is not supported.

4.3.8 SDO 1 Server Parameter (1200)

Contains the COB ID used by the client to server SDO and the server to client SDO. Entries are read only.

4.3.9 RPDO 1 Communication Parameter (1400)

The COB-ID is read / write while the transmission type shows 0xFF and is read only.

The COB-ID can only be written in pre-operational state. A write attempt in state operational or stopped leads to an SDO abort 0800.0022 (data cannot be transferred or stored because of the present device state).

After writing a new COB-ID a communication reset has to be executed to transfer the new values to the CAN controller hardware.

4.3.10 RPDO 1 Mapping Parameter (1600)

Because mapping is not supported the number of entries in subindex 1 reads 0.

4.3.11 TPDO 1 Communication Parameter (1800)

The COB-ID is read / write while the transmission type shows 0xFF and is read only.

The COB-ID can only be written in pre-operational state. A write attempt in state operational or stopped leads to an SDO abort 0800.0022 (data cannot be transferred or stored because of the present device state).

After writing a new COB-ID a communication reset has to be executed to transfer the new values to the CAN controller hardware.

4.3.12 TPDO 1 Mapping Parameter (1A00)

Because mapping is not supported the number of entries in subindex 1 reads 0.

4.3.13 CANopen Parameters (2000)

This entry contains 3 fields:

- CAN baud rate (read-only)
  The CAN baud rate can only be changed by the configuration command S (see chapter 3.10).
- CANopen auto start (read / write)
  If auto start is set to 0x00 the device enters the pre-operational state after initializuing and stays in that state until a CANopen master send a NMT start_remote_node message. If auto start is set to 0x01 the device enters automatically the operational state after initializing.
- CANopen node ID (read / write)
  Writing the node ID implies setting the four COB IDs for both SDOs, the RPDO and the TPDO to their default values derived from the node ID.

The writable entries can only be written in pre-operational state. A write attempt in state operational or stopped leads to an SDO abort 0800.0022 (data cannot be transferred or stored because of the present device state).

4.3.14 RS232 Parameters (2001)

This entry contains the RS232 baud rate in sub-index 2. The entry is read / write. The values are the same as described with the configuration command U (see chapter 3.12):

U0	115200
U1	57600
U2	38400

ESS Embedded Systems Solutions GmbH
Industriestrasse 15
D-76829 Landau

U3	19200
U4	9600
U5	4800
U6	2400

5. Data handling and transmission

Data handling and transmission is the same in CAN and COP mode. Characters received on the serial link are packed in a CAN frame which is called transmit process data object (TPDO). The identifier used is fetched from the parameter data (IT command). This CAN frame is sent either when 7 characters are received or when the timeout is expired after the last received character.

Received CAN frames with the identifier parameterized with the IR command are called receive process data object (RPDO). The characters contained in such a frame (1 to 7) are immediately sent to the serial link.

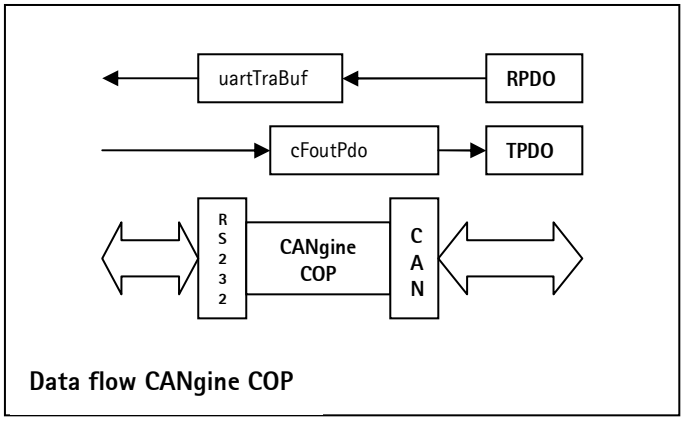
The CAN frames have a fixed data length of 8 data bytes. The identifiers come from the IT and IR command. The first data byte contains status information; the second up to the eighth data byte contain serial characters (1 up to 7).

CAN Identifier	DB 0	DB 1	DB 2	DB 3	DB 4	DB 5	DB 6	DB 7
Transmit/receive identifier	Status Info	SC1	SC2	SC3	SC4	SC5	SC6	SC7

DB Data byte
SC **Serial character**

For each transmission a toggle bit is changed so that the remote station can discover double sending of a frame. A double sended frame is discarded. Status byte is detailed in chapter 5.2.

Data flow scheme is shown in the following picture.



Data flow CANgine COP

There are two buffers:

- cFoutPdo is a CAN frame which holds up to seven the received serial characters,
- uartTraBuf is a 128 byte character buffer which holds the characters to send to the serial link

5.1 Performance

On setting the serial baud rate and the CAN speed make sure that no data overrun can occur due to wrong baud rates. The exact values depend from your application. To assist you in calculating these values find the two tables below which shows transmission times for UART and CAN depending from different bit rates.

Consider that CANgine COP works interrupt controlled. As soon as a RPDO is completely received on the bus the frame is read by an interrupt handler and the characters are put in the UART transmit buffer. As long as characters are in the UART transmit buffer these characters are sent by the UART transmit interrupt handler.

On the other hand if a character is received on the UART this character is packed in the transmit PDO. If seven consecutive characters are received or if the timeout value is reached the transmit PDO is send on the CAN bus.

This is the critical situation because while copying the CAN frame to the CAN hardware registers the UART interrupt must be disabled. Copying needs 220 µs. Therefore in applications where continuous data streams longer than 7 characters must be handled the maximum UART baud rate is 38400 baud.

UART baud rate	Time per char [µs]	Time per 7 chars [µs]
2400	4,167	29,167
4800	2,083	14,583
9600	1,042	7,292
19200	521	3,646
38400	260	1,823
57600	174	1,215
115200	87	608

CAN bit rate [kbit/s]	Standard frame [µs]	Extended frame [µs]
Best case times (no stuff bits)		
10	11,100	12,900
20	5,550	6,450
50	2,220	2,580
100	1,110	1,290
125	888	1,032
250	444	516
500	222	258
800	138.75	161.25
1,000	111	129
Worst case times (max. stuff bits)		
10	13,550	15,800
20	6,775	7,900
50	2,710	3,160
100	1,355	1,580
125	1,084	1,264
250	542	632
500	271	316
800	169.375	197.5
1,000	135.5	158

5.2 Status byte

7	6	5	4	3	2	1	0
RESET	dataByteCnt			usBO	csBO	res1	sTB

ESS Embedded Systems Solutions GmbH
Industriestrasse 15
D-76829 Landau

sTB	Send toggle bit The transmitting CAN node toggles this bit; the receiving CAN node controls this bit. If it doesn't toggle the receiving node discard the last frame as it was send twice.
res1	reserved
csBO	CAN send buffer overflow Data coming from the serial link could not be written into CAN transmit PDO. Data was lost
usBO	UART send buffer overflow Data coming from CAN could not be written into the uart transmit buffer. Data was lost.
dataByteCnt	Number of serial characters transmitted (1 ..7)
RESET	If this bit is set all error information is reset. Also communication is reset.

5.3 Error signaling

5.3.1 Error Bits in Error Register OD 1001

Bit No	Meaning
0	Generic error: CAN frame error
1	0
2	0
3	0
4	CAN communication error (buffer overflow)
5	0
6	0
7	UART transmitter overflow

5.3.2 Error Codes in OD 1003/1 to 1003/4

Code	Meaning
00xx	No Error
1001	CAN Layer 1 Error
1002	UART Transmitter Overflow
8110	CAN Overrun
8120	CAN in Error Passive

6. Display

6.1 Green Run LED

LED green	LED red	CAN Mode	CANopen Mode
Flickering	Off	Configuration mode	Configuration mode
On		Normal operating mode	NMT: operational
Flashing		n.a.	NMT: Preoperational
Flashing once		n.a.	NMT: stopped

6.2 Red Error LED

LED red	LED green	Meaning
On	Off	Severe hardware error, CANgine not operating

On	On	CAN bus off
Flashing four times		Overflow of send or receive buffer
Flashing once		CAN bus error passive
Off	On	Normal operation without errors

7. Problems and Help

Problem	**Cause**	**Help**
After powering up the red LED blinks one time and later four times	There is no other CAN node on the bus and the CANgine is parameterized to work in COP mode with heartbeat. The heartbeat frames leads to an overflow of the CAN send buffer	Connect another CAN node with the appropriate baud rate.

ESS Embedded Systems Solutions GmbH
Industriestrasse 15
D-76829 Landau

Notes

Notes

ESS Embedded Systems Solutions GmbH
Industriestrasse 15
D-76829 Landau

Notes

Notes

ESS Embedded Systems Solutions GmbH
Industriestrasse 15
D-76829 Landau

Disclaimers

Life support

These products are not designed for use in life support appliances, devices or systems where malfunction of these products can reasonably be expected to result in personal injury. ESS Embedded Systems Solutions GmbH customers using or selling these products for use in such applications do so at their own risk and agree to fully indemnify ESS Embedded Systems Solutions GmbH for any damages resulting from such application.

Right to make changes

ESS Embedded Systems Solutions GmbH reserves the right to make changes, without notice, in the products, and/or software, described or contained herein in order to improve design and/or performance. ESS Embedded Systems Solutions GmbH assumes no responsibility or liability for use of any of these products, conveys no license or title under any patent, copyright, or masks work to right to these products, and makes no representations or warranties that these products are free from patent, copyright, or mask work right infringement, unless otherwise specified.

ESS Embedded Systems Solutions GmbH
Industriestrasse 15
D-76829 Landau
Germany
Phone +49 (0) 6341 34870
info@ESSolutions.de