# CAN-CBM-AI4

## Expansion Module With 4 Analog Inputs

## Manual Extension

to product C.2831.xx

| | |
|---|---|
| **Document file:** | I:\texte\Doku\MANUALS\CAN\CBM\AI4\AI4-12.en9 |
| **Date of print:** | 20.06.2005 |

**Changes in the chapters**

The changes in the user's manual listed below affect changes in the *hardware* as well as changes in the *description* of the facts only.

| Chapter | Changes versus previous version |
|---|---|
| 1.2 | Chapter 'Position of Connectors and DIP Switches' inserted. |
| 1.6 | Order Notes inserted |
| 2.2.1 | Critical frequencies inserted |

Technical details are subject to change without notice.

N O T E

The information in this document has been carefully checked and is believed to be entirely reliable. **esd** makes no warranty of any kind with regard to the material in this document, and assumes no responsibility for any errors that may appear in this document. **esd** reserves the right to make changes without notice to this, or any of its products, to improve reliability, performance or design.

**esd** assumes no responsibility for the use of any circuitry other than circuitry which is part of a product of **esd** gmbh.

**esd** does not convey to the purchaser of the product described herein any license under the patent rights of **esd** gmbh nor the rights of others.


**esd electronic system design gmbh**
Vahrenwalder Str. 207
30165 Hannover
Germany

Phone: +49-511-372 98-0
Fax: +49-511-372 98-68
E-mail: info@esd-electronics.com
Internet: www.esd-electronics.com

**USA / Canada:**
**esd electronics Inc.**
12 Elm Street
Hatfield, MA 01038-0048
USA

Phone: +1-800-732-8006
Fax: +1-800-732-8093
E-mail: us-sales@esd-electronics.com
Internet: www.esd-electronics.us

# Contents

Side

This page is intentionally left blank.

# 1. Overview

## 1.1 Preface

This supplement to the system manual 'CAN-CBM-AI4' contains **additional** information about the analog inputs of the analog module CAN-CBM-AI4. General information about the modules (also about the analog inputs and outputs) can be taken from the system manual 'CAN-CBM-DIO8'.

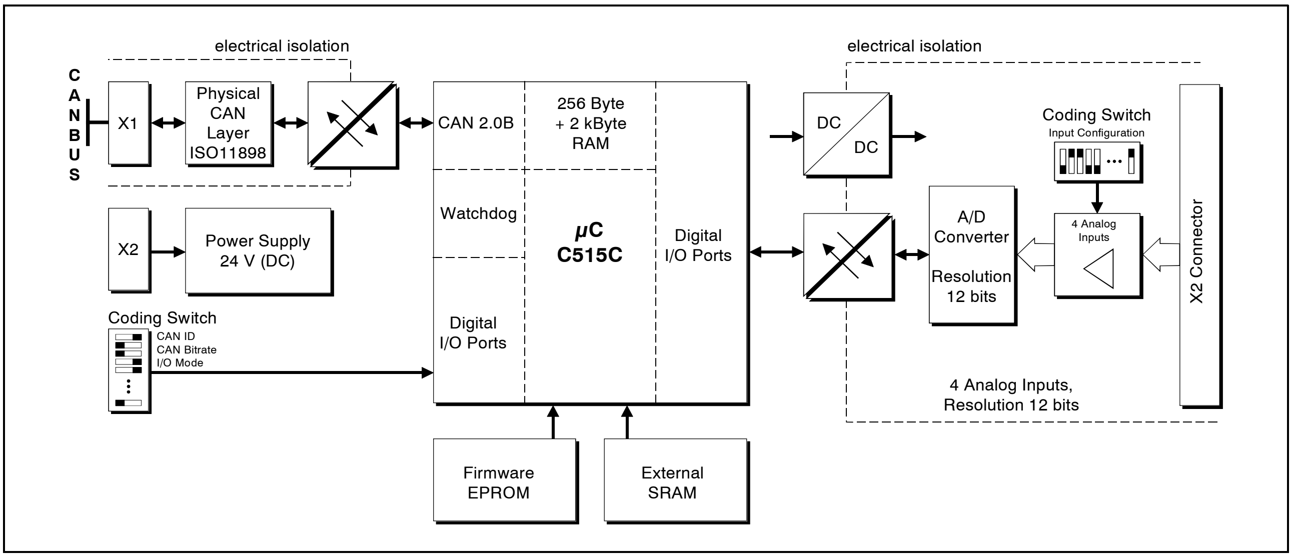



**Fig. 1:** Block diagram of the CAN-CBM-AI4 module

## 1.2 Position of the Connectors and DIP switches

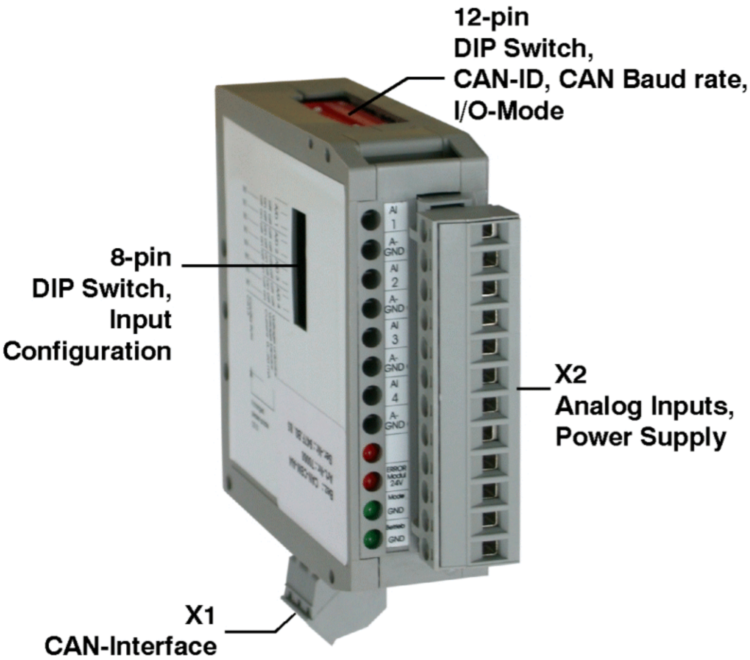
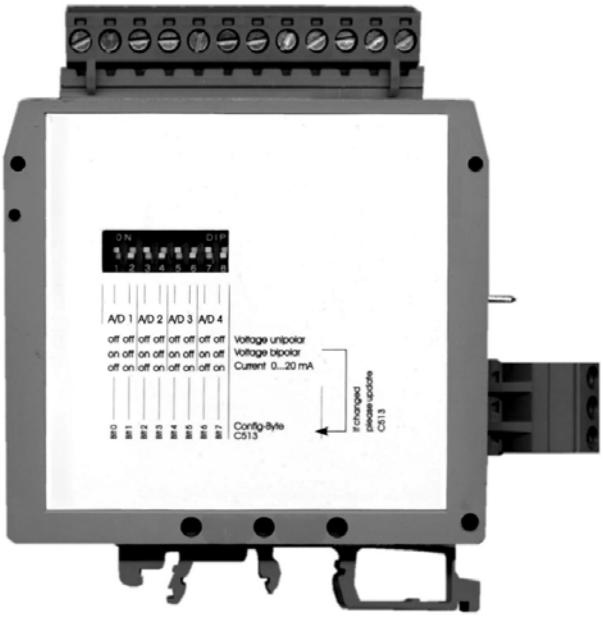



**Fig. 2:** CAN-CBM-AI4 module with position of the connectors and DIP switches

### 1.2.1 Assignment of the 8-pin DIP Switch for Input Configuration

The analog inputs can be configured via the 8-pin DIP switch (see page 11).
The following figure shows the 8-pin DIP switch with the default setting at delivery.



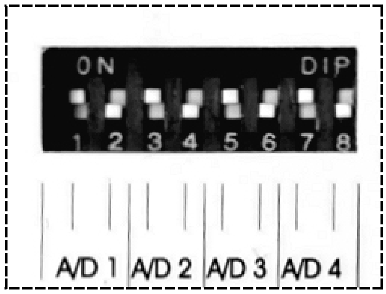



**Fig. 3:** Default setting of the 8-pin DIP switch at delivery

### 1.2.2 Assignment of the 12-pin DIP Switch

The CAN-ID, the CAN baud rate and the I/O-mode can be set via the 12-pin DIP-switch (see page 13).

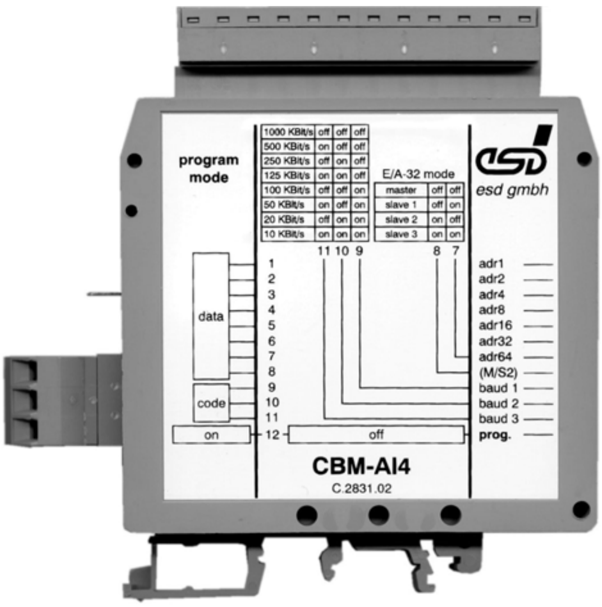



**Fig. 4:** Default setting of the 12-pin DIP switch at delivery

## 1.3 Connector Assignment X2, Analog Inputs, Power Supply

The connector is placed in the front panel of the housing. The module is equipped with a 12-pin Combicon connector (male) MSTB 2,5 of Phoenix Contact.

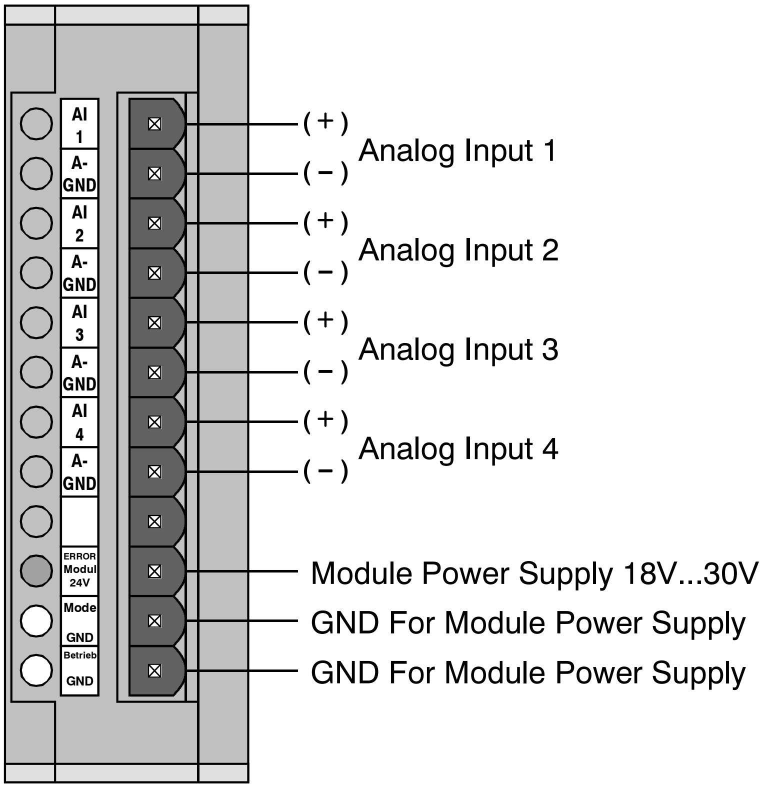
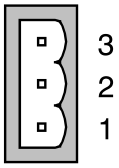



**Fig. 5:** Connector assignment of analog inputs and power supply

## 1.4 Connector Assignment X1, CAN

The CAN connector is located underside of the housing. The module is equipped with a 3-pin Combicon connector (male) MSTB2,5 of Phoenix Contact.

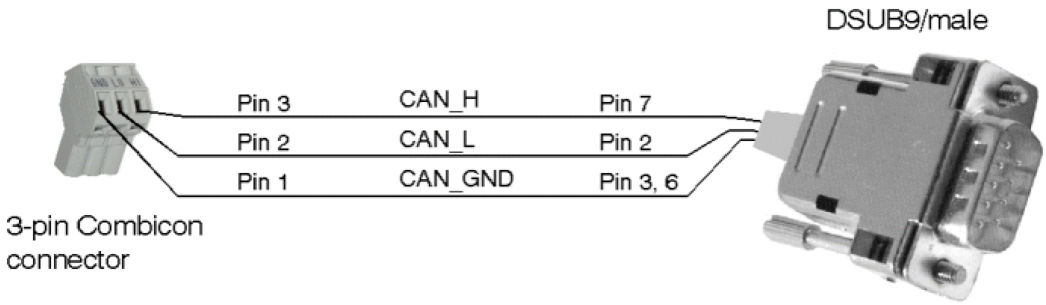



| Pin | Signal |
|---|---|
| 3 | CAN_H |
| 2 | CAN_L |
| 1 | CAN_GND |

**Signal description:**

CAN_L, CAN_H ... CAN signals
CAN_GND... reference voltage of the local CAN physical layers

**Recommendation of a CAN-CBM-cable, 3-pin Combicon to 9-pin DSUB:**




The Assignment of the 9-pin DSUB connector complies with CiA DS 102.

**Fig. 6:** Adapter cable 3-pin Combicon to 9-pin DSUB

## 1.5 Technical Data

| | |
|---|---|
| Number of analog inputs | 4 |
| Resolution | 12 bit or 11 bit + sign |
| Selection of the input range | via DIP-switch |
| Input ranges,<br>input resistors,<br>critical frequencies (low-pass) | • 0 ... +10 V $R_i \geq 300\ k\Omega$ $f_g = 26\ Hz$,<br>• -10 ... +10 V $R_i \geq 300\ k\Omega$ $f_g = 57\ Hz$,<br>• 0 ... +20 mA $R_i = 260\ \Omega$ $f_g = 61\ Hz$ |
| Electrical isolation | via optocoupler and DC/DC-converter |
| Supply voltage | $U_{VCC}$ = 12 V (order no.: C.2831.03)<br>$U_{VCC}$ = 24 V (order no.: C.2831.02) |
| Ambient temperature | 0...50 ˚C |

## 1.6 Order Notes

| Type | Properties | Order No. |
|---|---|---|
| CAN-CBM-AI4 | CAN-CBM-AI4 module with four analog inputs,<br>24 V power supply,<br>CANopen | C.2831.02 |
| CAN-CBM-AI4 12V | CAN-CBM-AI4 module with four analog inputs,<br>12 V power supply,<br>CANopen | C.2831.03 |
| CAN-CBM-Cable | CAN cable for CAN-CBM modules, length 0,3 m,<br>one end DSUB9, one end wire end sleeves | C.1323.03 |
| CAN-CBM-AI4-ME | User manual in English [1*)] | C.2831.20 |

1*)... If module and manual are ordered together, the manual is free of charge.

# 2. Analog Inputs of the CAN-CBM-AI4 Module

## 2.1 Internal Sampling Rate

The internal sampling rate is 200 samples per second, i.e. every 5 milliseconds four analog values are digitalized by the A/D converter.
In order to avoid aliasing effects a first order low-pass filter with a critical frequency of about 50 Hz is connected in series to the A/D converter. For special applications where the phase shift caused by the filter would be disturbing, a special design with a modified hardware low-pass filter is available.

## 2.2 Filtering the Input Signals

### 2.2.1 Digital Filtering of the Analog Input Signal

In order to suppress low-frequency disturbances of the analog input signal (such as mains noise, variations of DMS signals caused by vibrations, etc.) the signal of the A/D converter is handled in a two-stage digital signal processing, if desired. These two stages are:

1. Smoothing by means of an averaging of a maximum of 8 measured values.

2. Filtering by means of a first order digital low-pass filter with critical frequency which can be set.

When filtering a square-wave signal, averaging generally provides ‘ramps’ instead of steep edges. This requires less calculating than filtering by means of a digital low-pass filter. All channels are averaged every 5 ms.

When filtering by means of a digital low-pass filter, square-wave signals provide ‘e-functions’ rather than steep edges. The digital filter is called every 10 ms and currently needs about 5 ms per calculation. The calculation of the digital filter requires calculation time of the local processor. This could lead to loss of performance for other duties of the processor.

### 2.2.2 Configuration of the Filter Settings

The sum of the input filters of the analog signal results as described in the figure below.

| | | | | |
|---|---|---|---|---|
| • voltage input | unipolar | 0 ... +10 V | $R_i \geq 300$ kΩ | $f_g = 26$ Hz |
| • voltage input | bipolar | -10 ... +10 V | $R_i \geq 300$ kΩ | $f_g = 57$ Hz |
| • current input | unipolar | 0 ... +20 mA | $R_i = 260$ Ω | $f_g = 61$ Hz |

The critical frequencies of the analog low-pass depend on the input circuit selected.

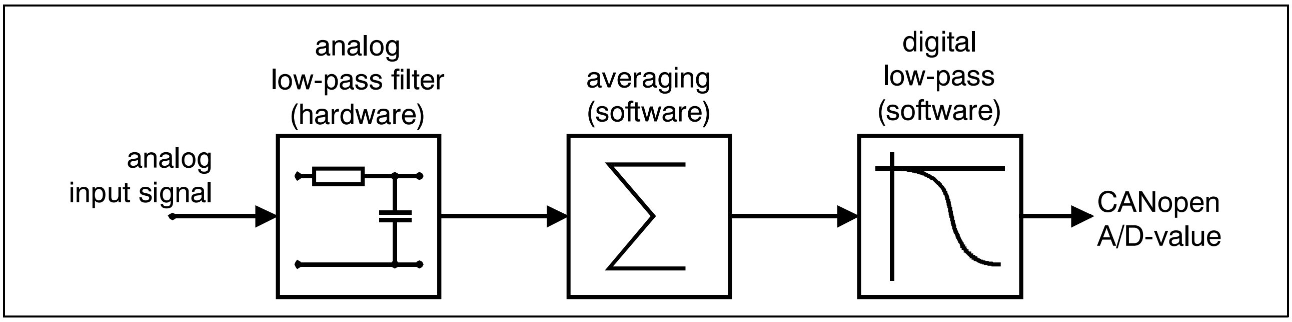



**Fig. 7:** Coupling of the filters of the analog input signals

The entries in code position 13 are used to configure the filtering. They can be accessed via service-data object in the ‘manufacturer-specific’ part of the CANopen dictionary.
The exact structure of the according object had not yet been determined at the time the manual went into print (according parameters have unfortunately not been designed for CiA DSP 401 Version 1.4).

**Code position 13, subindex 1, low byte:** Number of addends for averaging

permissible range:
0: no averaging
2: averaging the last two measured values
...
8: averaging the last eight measured values

**Code position 13, subindex 1, high byte:** time constant for digital low-pass (IIR)

permissible range:
0: no digital filtering
1: low-pass with $f_g$ = 6,7 Hz (highest critical frequency)
2: low-pass with $f_g$ = 6,0 Hz
3: low-pass with $f_g$ = 5,3 Hz
4: low-pass with $f_g$ = 4,5 Hz
...
8: low-pass with $f_g$ = 3,5 Hz
...
16: low-pass with $f_g$ = 1,0 Hz
20: low-pass with $f_g$ = 0,6 Hz
...
30: low-pass with $f_g$ = 0,4 Hz
31: lowest critical frequency ca. 0,3 Hz

scan rates:
5 msec. without digital filter
20 msec. with digital filter

Integration factor and filter-time constant are combined to a 16-bit parameter.
The value **0x0000** means ‘no averaging and no filtering’,
the value **0xFFFF** means ‘averaging with maximum addend number and filtering with smallest possible critical frequency’.

### 2.2.3 Displaying Over Modulation of Analog Inputs

Especially when filtering by means of software, probable over modulations of the A/D converter caused by superimposed alternating signals cannot be recognized by means of the measured value provided by the filter. Therefore, an additional monitoring of values provided by the A/D converter before the filtering has been implemented. It works as follows:

If the value provided by the converter is at modulation limit, an internal error counter will be risen (this is ‘4095’ for the 12-bit A/D converter).
If the value provided by the converter, however, is not at the limit, the internal error counter will be lowered.

If the internal error counter crosses a certain limit, the red module-error LED will be flashing in ‘short-long’ cycle. The monitoring does not have any further effects.

# 3. Setting the Analog Input Type

Each of the four analog inputs can be configured individually.
The following input types are available as a standard for the CAN-CBM-AI4:

- voltage input unipolar 0 ... +10 V $R_i \geq 300\ k\Omega$ $f_g = 26$ Hz
- voltage input bipolar -10 ... +10 V $R_i \geq 300\ k\Omega$ $f_g = 57$ Hz
- current input unipolar 0 ... +20 mA $R_i = 260\ \Omega$ $f_g = 61$ Hz

The inputs are configured via an eight-pin coding switch on the PCB of the extension module **and** via code position (see appendix). **Both** positions must have the same contents to make sure that the CPU knows the current position of the DIP switch on the analog PCB.
Therefore, the positions of all eight switches of the analog board are set in C513 (this is also optionally possible with the 12-pin DIP switch on the basic board).

The eight-pin DIP switch on the analog extension board has the following meaning:

| 8-pin DIP switch No. | Addend for C513 | Meaning |
|---|---|---|
| 1 | 1 ($01_h$) | channel 1 "bipolar" |
| 2 | 2 ($02_h$) | channel 1 "current input" |
| 3 | 4 ($04_h$) | channel 2 "bipolar" |
| 4 | 8 ($08_h$) | channel 2 "current input" |
| 5 | 16 ($10_h$) | channel 3 "bipolar" |
| 6 | 32 ($20_h$) | channel 3 "current input" |
| 7 | 64 ($40_h$) | channel 4 "bipolar" |
| 8 | 128 ($80_h$) | channel 4 "current input" |

**Example:** All four analog inputs are to run as voltage inputs in the range of -10 V to +10 V. For this, the ‘bipolar’ switch has to be closed for each channel (S1, 3, 5, 7 closed), the according ‘current’ switch has to remain open (S2, 4, 6, 8 open).
The according configuration byte for C513 is this case is 1+4+16+64 = $55_h$.

If there is no possibility to configure the module, C513 can also be programmed by means of the 12-pin DIP switch on the basic board (see page 13).

# 4. Calibrating the Analog Inputs

## 4.1 Calling the Automatic Calibration Routine

**The modules have already been calibrated at the manufacturer.**
**The user should not change this calibration!**

Principal course of the 'automatic' calibration (e.g. for channel 1):

1. Ensure that the contents of C513 correspond to the position of the eight-pin DIP switch on the analog extension board.

2. Enable access to calibration data by special write access onto C512.10.

3. Apply minimum voltage or current from reference to input, e.g. -10 Volt.

4. Calling 'AutoCalibrate' by writing '0000' in C512.11 (see appendix): the calibration routine 'memorizes' the first value pair.

5. Apply maximum voltage or current from reference to input, e.g. +10 Volt.

6. Calling 'AutoCalibrate' by writing '4095' in C512.12 (see appendix): The calibration routine 'memorizes' the second value pair and determines the offset and the scaling. (These variables are stored in C520.x and C521.x).

7. Check the newly determined calibration values by reading out C520 and C521.

8. In 'operational' status the analog values in PDO (CANopen format) have to be watched In a short-circuited input (0 Volt) the value in the PDO has to at about '0'. In bipolar measurings the minimum is '-32768' according to DS401, in unipolar measurings the minimum is '0' according to DS401. The maximum in both cases is '32767'.

# 5. Programming via the Coding Switch

## 5.1 Operation

In order to be able to change the most important parameters of the I/O unit without SDO access, as well, a procedure to reprogram some code positions by means of DIP switches (basic board) has been implemented. The course of procedure for the programming mode has been slightly modified, therefore, and **does not correspond to the description in the CAN-CBM-DIO8 manual (V0.xx) anymore** for the CAN-CBM-AI4 with 4 analog inputs.

As usually, the **programming mode** is activated by closing **S12** briefly (see page 4).
In the further course of the programming mode, S12 has about the function of an 'Enter' key with which the data set via DIP switch are written into a code position.

**Note:** in this hardware 'ON'=below and 'OFF'=above !!

By means of switches 9,10,11 the code position which is to be programmed is selected. Via switches 1-8 the (maximum) eight data bits are set which are to be programmed into the code position.
Eight different combinations can be set by means of switches 9, 10, 11, and therefore a maximum of eight different code positions can be programmed in this way.
The code set via S9,10,11 corresponds to a code number according to the following table:

| Switch 11 | Switch 10 | Switch 9 | Code position | Meaning |
|---|---|---|---|---|
| 0 | 0 | 0 | C10 | port directions |
| 0 | 0 | 1 | C19 | option flags |
| 0 | 1 | 0 | C25 | abbreviated Sync configuration |
| 0 | 1 | 1 | C26 | communication cycle |
| 1 | 0 | 0 | C513 | position of the 8-pin analog DIP switch |
| 1 | 0 | 1 | C13 | abbreviated analog filter configuration |
| 1 | 1 | 0 | C2048 | default settings (only for manufacturer) |
| 1 | 1 | 1 | reserved | |

The number of the code position is shown in programming mode via the mode LED (green, hundred's), module error LED (red, ten's) and output error LED (red, unit). In addition, each LED flashes according to the decimal place. For the digit '0' the according LED does not flash, for digit '9' it flashes accordingly nine times. The entire 'flash cycle' is repeated every seven seconds so that the activated code position can be recognized by counting the flash impulses.

**Programming via the Coding Switch**

In order to store the set data the programming switch (S12) has to be switched to ‘ON’ briefly and than back to ‘OFF’. When switching from ‘ON’ to ‘OFF’ all green and red LED’s switch on for about 0.5 seconds at the same time to indicate a successful programming.

You can only leave the programming mode by switching off the mains supply. Before you switch the voltage on again, the right module ID and bit rate is to be set.

## 5.2 Unusual Features of Some Code Positions when Programmed via DIP Switch

### 5.2.1 Shortened Sync Configuration (C25)

The contents of this code position are identical to the MS byte from object $1005_h$ according to CiA Draft Standard 301. Object $1005_h$ is called (slightly misleading) a ‘COB-ID Sync Message’, there. Actually, however, the most significant byte of this CANopen object defines the handling or generating of the SYNC telegram. The bit 31 (‘1: Device consumes SYNC message’) which is described there corresponds to S8 of the DIP switch when C25 is programmed.
Bit 30 from DS301 (‘1: Device generates SYNC message’) corresponds to S7 of the DID switch.
Writing onto this code position affects also the contents of object $1005_h$!

### 5.2.2 Communication Cycle (C26)

The contents of this code position corresponds to the object $1006_h$ (Communication Cycle Period), described in the CiA Draft Standard 301, unless the time unit. The parameter are set however in units of **milli seconds** and not in µs as defined in the specification! A maximum of 255 milliseconds can be programmed via DIP switch, a maximum of 32767 via SDO.
If a module has been configured as SYNC transmitter (see C25), this time is the interval between two SYNC transmissions.
Writing onto this code positions also affects the contents of object $1006_h$!

### 5.2.3 Shortened Input Filter Configuration (C 13)

When programming C13 via DIP switch (in programming mode), identical filter constants are written onto all analog inputs, because here (in contrast to programming via SDO) no subindex can be specified.
By means of switches 1..8 filter constant AND averaging are activated:
0 = neither averaging nor digital filter
1...8 = only averaging with (1...8) addends
9..255 = averaging with eight addends AND digital filtering (fg from DIP code)

A different filter setting for more than one analog input is only possible by means of SDO.

## 5.3 Examples for Manipulating Code Positions via DIP Switch

### 5.3.1 Setting the PDO Transmission Type to 'SYNC Reception'

1. Change into the programming mode by closing S12.
   All red and green LED's flash 'short-long-long-short' to indicate the programming mode.

2. Open S12 again.
   The red and green LED's now show the code position selected via S9...S11.

3. Set S9,10,11 for the 'shortened SYNC configuration' via C25:
   S9 off, S10 on, S11 off

4. Set the value 0x80 (hex) via S1..S8:
   S1..S7 off,   S8 on

5. Program:
   S12 briefly 'on' and 'off' again   =>   red and green LED's switch on briefly.

**Note:** In this setting the 'PDO Transmission Type' (CANopen object $1800_h$) is automatically switched to 'SYNC operation', as well, if this had not been the case until then. When programming by means of the SDO, this does not occur automatically.

### 5.3.2 Setting the PDO Transmission Type to 'SYNC Transmission'

For special applications the module can also work as a SYNC telegram transmitter. In this case its own transmit PDO is 'synchronized' as well.

1. Change into programming mode by closing S12.
   All red and green LED's flash 'short-long-long-short' to indicate the programming mode.

2. Open S12 again.
   The red and green LED's now show the code positions selected via S9..S11.

3. Set S9, 10, 11 for the shortened SYNC configuration via C25:
   S9 off, S10 on, S11 off

4. Set the value 0x40 (hex) via Sa...S8:
   S1..S6 off,   S7 on,  S8 off.

5. Program:
   S12 briefly 'on' and 'off' again   =>   red and green LED's switch on briefly.

**Note:** In this setting the 'PDO Transmission Type' (CANopen object $1800_h$) is automatically switched to 'SYNC operation', as well, if this had not been the case until then. In addition, the 'Communication Cycle' (CANopen object $1006_h$) is set to a valid value, if this had not been the case before (at least 5 milliseconds). When programming by means of the SDO, this does not occur automatically.

# 6. Example Application: Monitor Program CAN-Scope

The analog input values of the CAN-CBM-AI4 are displayed in the CAN-Scope as follows:

| Byte 1 | Byte 2 | Byte 3 | Byte 4 | Byte 5 | Byte 6 | Byte 7 | Byte 8 |
|---|---|---|---|---|---|---|---|
| Channel 1_L | Channel 1_H | Channel 2_L | Channel 2_H | Channel 3_L | Channel 3_H | Channel 4_L | Channel 4_H |

If the bipolar input measuring area of -10 V...+10 V is selected, the positive or negative results are calculated as follows:

## 6.1 Calculation of Positive Results

The value **+2,8 V** for example is displayed in the CAN-Scope window as **HEX: 70 24**.

1. Due to the various data organisation in the MOTOROLA and INTEL format the high byte and low byte must be swapped:
   **HEX: 70 24 => HEX: 24 70**

2. Than convert the hexadecimal number into the according decimal number:
   **HEX: 24 70 => decimal: 9328**

3. Calculation of the **LSB** (**l**east **s**ignificant **b**it):
   - voltage measuring area: **10 V**
   - **MSB** (**m**ost **s**ignificant **b**it): 0 => positive in sign / 1 => negative in sign
   - data format: 15 bit (= 32768)+1 sign bit (MSB)

   That is: 10 V/32768 = **0.000305 V** as **LSB**

   From that the result is: 9328 x 0.000305 V = **2.8 V**

## 6.2 Calculation of Negative Results

The value **-2,6 V** for example is displayed in the CAN-Scope window as **HEX: E0 DD**.

1. Due to the various data organisation in the MOTOROLA and INTEL format the high byte and low byte must be swapped:
   **HEX:E0 DD => HEX: DD E0**

2. Convert the hexadecimal number values into binary numbers:

| **HEX** | D | D | E | 0 |
|---|---|---|---|---|
| **binary** | 1101 | 1101 | 1110 | 0 |

3. Form the complement to two:

| **1101** | **1101** | **1110** | **0** | |
|---|---|---|---|---|
| 1101 | 1101 | 1101 | 1111 | **-1** |
| 10 | 10 | 10 | 0 | **negation** |

4. Convert the binary numbers in the according hexadecimal number values:

| **binary** | 10 | 10 | 10 | 0 |
|---|---|---|---|---|
| **HEX** | 2 | 2 | 2 | 0 |

5. Convert the hexadecimal number into the according decimal number:
   **HEX: 22 20 => decimal: 8736**

6. Calculation of the **LSB** (**l**east **s**ignificant **b**it):
   - voltage measuring area: **10 V**
   - **MSB** (**m**ost **s**ignificant **b**it): 0 => positive in sign / 1 => negative in sign
   - data format: 15 bit (= 32768)+1 sign bit (MSB)

   That is: 10 V/32768 = **0.000305 V** as **LSB**

   From that the result is: 8736 x 0.000305 = **2.6 V** => **-2.6 V** because MSB = 1 (negative in sign)

# 7. Appendix: Code Positions

| General Code Positions for CAN-CBM module with analog I/Os |
|---|

| Code No. [dec] | Contents | Data type, [unit] | Access r = read w= write | Set / Display possibilities | Default settings | Notes, references |
|---|---|---|---|---|---|---|
| 13.1 ... 13.4 | filter constants of the analog inputs | 16 bit | r, w | see page 9 | 0 = no filtering | subindex = channel number ( 1 to 4 ) |
| 10 | port directions | | r, w | | | see manual CAN-CBM-DIO8 |
| 25 | shortened SYNC configuration | 8 bit | r, w | 0x80 = Sync received<br>0x40 = transmit Sync | | see page 15 |
| 26 | communication cycle | 8..16 bit | r, w | e.g. Sync transmit interval in milliseconds | | see page 15 |

| Code positions for CAN-CBM-AI4 (4 analog inputs) |
|---|

| Code No. [dec] | Contents | Data type [unit] | Access r = read w= write | Set / Display possibilities | Default settings | Notes, references |
|---|---|---|---|---|---|---|
| 512 | number 'AD4' register | 8 bit | r | 0 (not installed)<br>... <number register> | | |
| 512,1 | status register | 8 bit | r | bit combination | | |
| 512,2 | debugging register | 8 bit | r | bit combination | | |
| 512,3 | setting the 'default' calibration values | 8 bit | r | bit combination | | |
| 512,4 | | 8 bit | r | | | |
| 512,5 | | 8 bit | r | | | |
| 512,6 | | 8 bit | r | | | |
| 512,7 | | 8 bit | r | | | |
| 512,8 | | 8 bit | r | | | |
| 512,9 | | 8 bit | r | | | |
| 512,1 | requesting calibration options | 8 bit | r | | | only for manufacturer |
| 512,11 | auto calibration channel 1, min. | 16 bit | r, w (manuf.) | Requesting a calibration routine (see page 12 ) | | only for manufacturer |
| 512,12 | auto calibration channel 1, max. | 16 bit | r, w (manuf.) | Requesting a calibration routine | | only for manufacturer |
| 512,13 | auto calibration channel 2, min. | 16 bit | r, w (manuf.) | Requesting a calibration routine | | only for manufacturer |
| 512,14 | auto calibration channel 2, max. | 16 bit | r, w (manuf.) | Requesting a calibration routine | | only for manufacturer |
| 512,15 | auto calibration channel 3, min. | 16 bit | r, w (manuf.) | Requesting a calibration routine | | only for manufacturer |

| Code No. [dec] | Contents | Data type [unit] | Access r = read w= write | Set / Display possibilities | Default settings | Notes, references |
|---|---|---|---|---|---|---|
| 512,16 | auto calibration channel 3, max. | 16 bit | r, w (manuf.) | Requesting a calibration routine | | only for manufacturer |
| 512,17 | auto calibration channel 4, min. | 16 bit | r, w (manuf.) | Requesting a calibration routine | | only for manufacturer |
| 512,18 | auto calibration channel 4, max. | 16 bit | r, w (manuf.) | Requesting a calibration routine | | only for manufacturer |
| 513 | DIP switch of four-channel analog extension | 8 bit | r, w | bit combination as DIP switch:<br>Bit 0 = C.1 bipolar<br>Bit 1 = C.1 current<br>....<br>Bit 6 = C.4 bipolar<br>Bit 7 = C.4 current | “ | has to be updated when changing the DIP switch setting ! |
| 514.1 ... 514.8 | Uncorrected analog values, ‘directly’ from the converter | 16 bit (array) | r | value range 0 ... 4095 | “ | only for diagnostical matters |
| 516 | customized factor for analog value conversion | 16 bit | r,w | 0 = passive, otherwise -32767 .... + 32767 | 0x00 | not yet documented |
| 517 | customized divisor for analog value conversion | 16 bit | r,w | 0 = passive, otherwise -32767 .... + 32767 | 0x00 | not yet documented |
| 518 | customized offset for analog value conversion | 16 bit | r,w | -32767 .... + 32767 | 0x00 | not yet documented |
| 520.1 .... 520.16 | analog calibration values: offsets | 16 bit (array) | r, w (manuf.) | -32767 .... + 32767 | depending on hardware | only for manufacturer |
| 521.1 .... 521.16 | analog calibration values: scaling | 16 bit (array) | r, w (manuf.) | -32767 .... + 32767 | depending on hardware | only for manufacturer |