

# CEAD20

31-mar-2011.  
Embedded software version 1.

## 1. Main features

This device is designed for voltage measurements in control systems of accelerators as embedded part (temperature monitors and etc.). The device may be used as a general purpose analog-to-digital converter.

The device includes:

- 20/40-channels precise ADC
- 4-channel output register with galvanically isolated outputs;
- 4-channel input register with galvanically isolated inputs;
- CANBUS interface for interaction with control computer;
- On-board micro-controller.

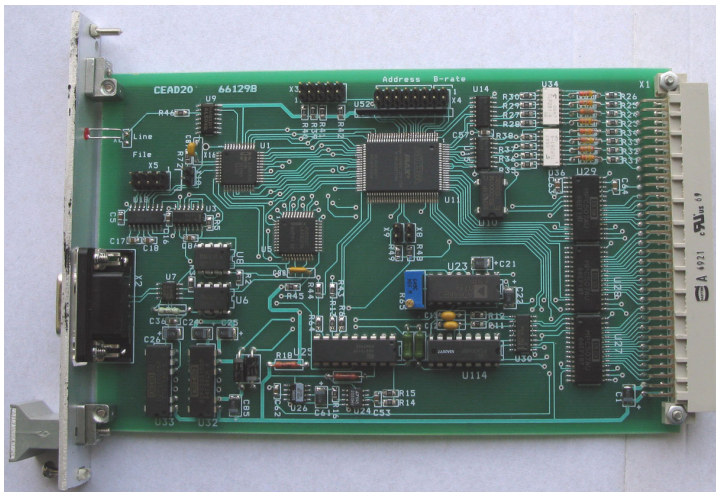


Photo of. CEAD20

ADC can work in different modes. The base mode is a multi-channel mode. In this mode the device scans predefined channels, measures them, stores measured values in internal memory and sends data in CANbus (if it was required by mode). For investigation of powers supply behavior here may be used one-channel mode (digital oscilloscope mode). In this mode the device measures the only channel with defined time of measurements and sends these values to network. In order to record and analyze occasional spices of output current there may be used mode of continual recording. In this mode the device measures chosen channel and stores measured values in internal ring buffer with capacity 128 values. A control computer can break the process any time, request a pointer of ring buffer, read date which was written before break and then analyze behavior of chosen channel. Actually, both latest modes are the one mode. The only difference is a value of flag in mode specification. This flag defines behavior of device- the information should be sent to network or it should be stored in ring buffer.

All ADCs in CANbus network or predefined group of ADC might be started in multi-channel mode simultaneously by broadcast message. This property is implemented by using labels. A user can assign label to device when it is started in multi-channel mode. Later, if device receives broadcast start message, it compares label in broadcast message with label assigned to multi-channel mode. If both labels are identical the broadcast start accepts like address start command. There are stop command only address type or global type, group stops are not defined in protocol.

Hardware implementation of converter consist of delta-sigma ADC chip, and 24-channel bi-wire multiplexer. The device is intended to be embedded in power supply racks. The device requires for proper operation the only power supply with voltage +5V ( $\pm 5\%$ ).

## 2. Specifications (main parameters):

1. ADC resolution - 24 bits.
2. Effective resolution – from 15 bits (time of measurements is 1 mS) to 20 bits (time of measurements is 20 mS and more).
3. ADC offset error - 50  $\mu$ V.
4. ADC accuracy (for integration time 20 mS and more) - 0.003%
5. ADC accuracy (for integration time 1 mS) - 1.0%
6. ADC input ranges-  $\pm 10$ V (main range).
7. ADC input current 1 nA.
8. ADC common-mode input range- 10.5V.
9. ADC common-mode rejection- 75 dB.
10. ADC time of measurements- from 1 mS to 160 mS.
11. Channels of output register – 4.
12. Maximal voltage for output register – 50 V.
13. Maximal current for output register - 5÷10 mA.
14. Channels of input register – 4.
15. Voltage for input register- 2.5-6.0V.
16. Input resistance for input register- 510 Ohm.
17. Input-output isolation voltage for registers - 1500 B.
18. Temperature sensitivity of on-board sensor (typical) – 1,9 mV/ $^{\circ}$ C.
19. Output voltage of temperature sensor at +25  $^{\circ}$ C – 0.56 B  $\pm 10\%$ ..
20. CANbus transceiver is galvanically isolated from network and it is in compliance with ISO 11898-24V (chip PCA82C251).
21. Voltage between transmission line and device- 1000V.
22. Baud rates- 1000, 500, 250, 125 Kbaud (may be chosen by jumpers).
23. Voltage of power supply- +5V,  $\pm 5\%$ .
24. Power supply current- <0,7 A (typical value- 0.4A). Power supply must provide starting current > 1,2 A.
25. Size of device - 3U \* 160 mm.

### 3. External connections

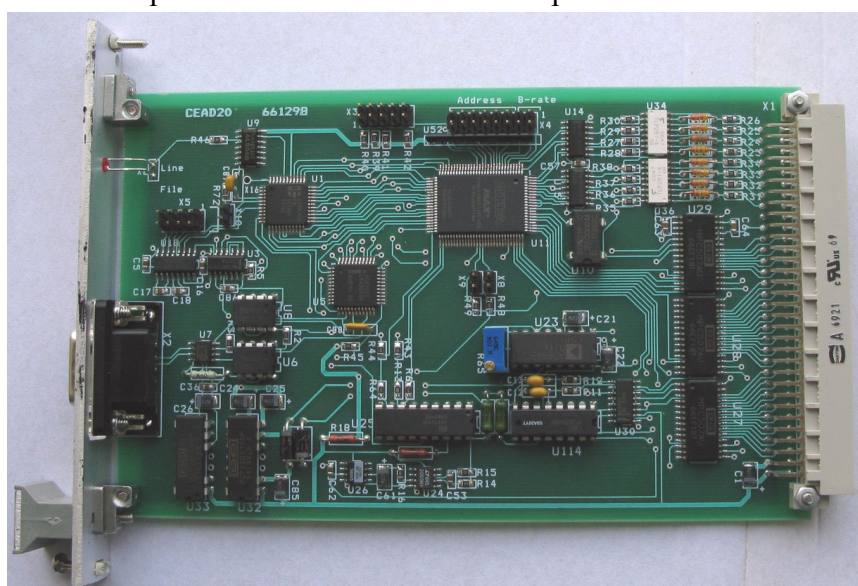
The device CEAD20 is implemented as eurocard 3U\*160. A front panel of the device contains a network connector (DB-9M) and a LEDs. The LED is blinking during processing CANbus messages. Connection with external channels of control and measurements carry out by rear connector (DIN 41612).

#### 3.1. Jumpers

The device CEAD20 has a set of jumpers- X4 and two individual jumpers (X8 and X9).

X4 includes 8 jumpers. 6 jumpers define number (address) of device in network (this number is used to compose identifier of messages) and 2 jumpers define baud rate.

Jumper location is shown on board photo.



Destination of jumpers in X4 group.

Designation	Location	Destination
X4-7	Upper	N5- included in device number (most significant bit)
X4-6	...	N4- included in device number
X4-5	...	N3- included in device number
X4-4	...	N2- included in device number
X4-3	...	N1- included in device number
X4-2	...	N0- included in device number (least significant bit)
X4-1	...	BR1- defines baud rate
X4-0	Lower	BR0- defines baud rate

Jumpers N5...N0 defines logical number (address) of device which is used to compose message identifier for CANbus network (for more detail see PROTOCOL part of this description). An installed jumper should be interpreted as logical 0 and absence of jumper should be interpreted as logical 1.

**Don't use addresses 34, 3C, 3D, 3E и 3F (hexadecimal).**

### Baude rate coding.

BR1	BR0	Baud rate
Connected	Connected	1000 Kbit/sec
Connected	Disconnected	500 Kbit/sec
Disconnected	Connected	250 Kbit/sec
Disconnected	Disconnected	125 Kbit/sec

X9 defines mode of input circuitry of ADC. If there is not jumper then device is configured for differential measurements (20 differential inputs). Presence of jumper configures ADC for single-wire measurements (40 inputs).

X8 is not used while.

State of X8 and X9 and input circuitry mode can be detected by requesting for device attribute. Byte «Hardware Revision» contains state of these jumpers. Bit b1 is equal X9 state, bit b2 is equal X8 state.

State of these jumpers is registered by processor only on Power-On (or Reset).

### NOTES:

1. CANbus is bus with multiple accesses and incorrect baud rate setting may affects on transfer messages of other devices up to impossibility of access to the device.
2. In network may exist concurrently devices with identical numbers (addresses). Formally it is permissibly, but actually it does cause a lot of problem. Connecting to network devices with identical numbers is strictly not recommended.

### 3.2 Front panel



A front panel includes:

**Line LED**

**CANbus connector**

**Line LED** is blinking during processing CANbus messages by onboard processor.

After power-on the device blinks by LED a few times. The LED is blinking during “bus-off recovery” procedure also.

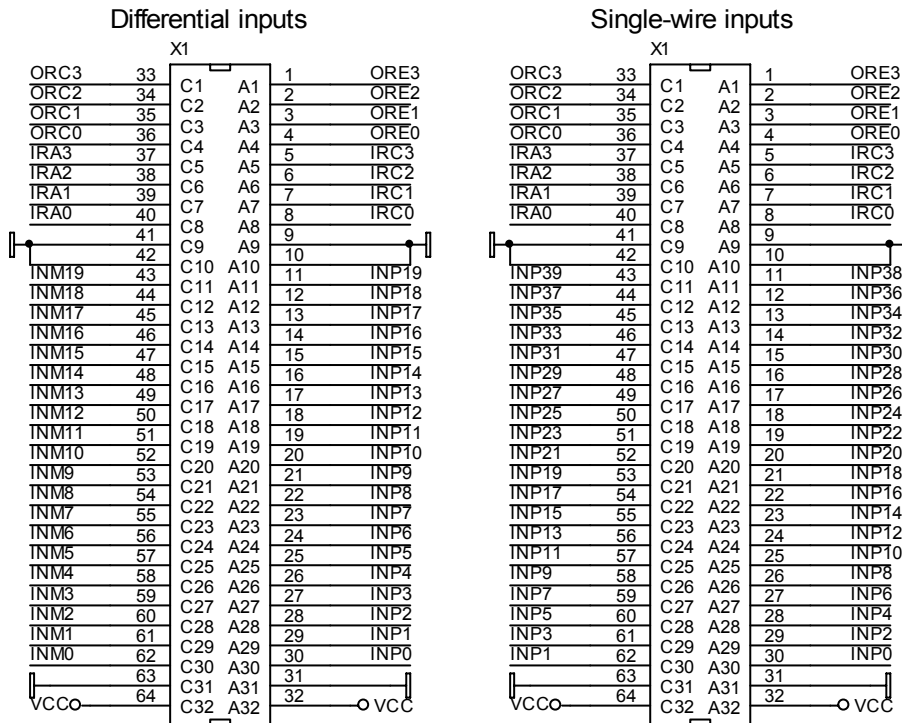
**CANbus** connector (DB-9M) is intended for connection to media. Pin designations follow below in table.

2	CAN-L	One wire in pair
3	GND	Shield of cable
7	CAN-H	One wire in pair

Shielded twisted pair is used as media. According to the ISO 11898-2 it should has nominal impedance 120 Ohm. Line termination has to be provided by termination resistors of 120 Ohm located at both ends of the line.

### 3.3 Main connector

There is DIN 41612 connector for connecting with input and output signals. Both analog and digital signals use the same connector.

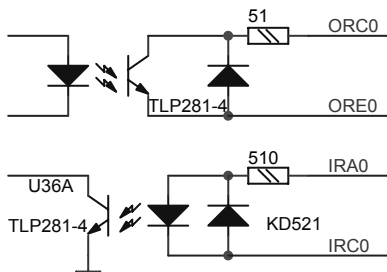


The main connector provides for user 20 differential pair (or 40 single-wire) inputs of ADC, input and output registers and pins for powering device. Connections of ADC inputs with a signal source should be implemented by twisted pairs.

A mnemonics of designation is the following: INPx means “input positive x-number”; INMx means “input negative x-number”.

The device uses the only external power supply +5B (5%). For powering device one should use pins 31, 32, 63 and 64. Other “ground” pins are connected with analog “ground” and should not be used for powering device.

Below is shown a fragment of circuitry input/output registers. Both registers are designed with galvanic isolation that is provided by optocouplers. Both registers use TLP281 optocouplers.



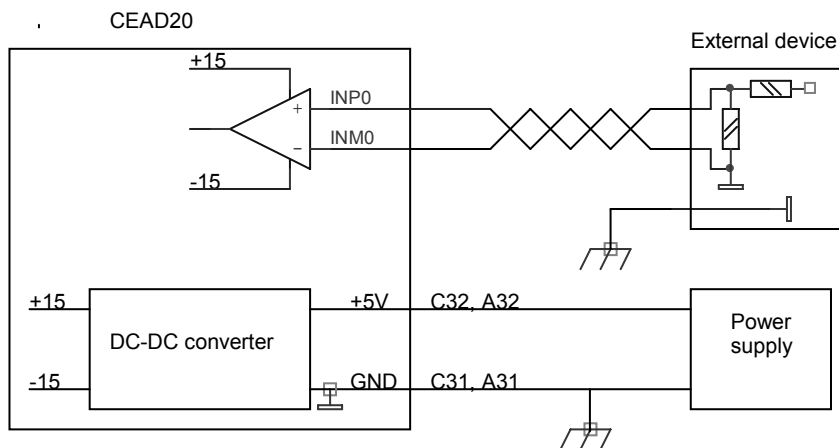
Input register is intended for detection external voltage or current. Input voltage range is from 3V to 12V. Input current range is from 4mA to 20mA. Unconnected input (no current in LED) is interpreted as logical 0.

Output register is able to provide output current up to 10 mA. Output voltage may be up to 30V.

## 4. Basics of operations for CEAD20

The device includes a multi-channel ADC, input register, output register and a micro-controller. The micro-controller integrates all parts together and provides a connection with a control computer by CANbus. Logically input and output registers are not connected with ADC and are controlled by specific messages. After power-up the micro-controller writes into output register zero and sends to host a power-up message. Moreover, microcontroller initiates multichannel measurement mode of ADC with 20 mS integration time, scanning channels from 0 to 23 and without sending data into line. It allows avoid ADC initialization in simple applications.

There are described main features of analog-to-digital converter of device. The picture below may help to connect ADC with signal source more correctly (for differential measurements).



Analog-to-digital converter consists of an ADC chip, a reference source, an analog multiplexer and a calibration reference source. There is used a delta-sigma ADC chip. A delta-sigma converter technology has some specific properties, which affects on operations of device. It is useful to observe these properties for good understanding of device operations.

Delta-sigma converters provide very high resolution with low noise level, but they have low stability. There is used a calibration procedure in order to compensate this instability. An on-board micro-controller performs the calibration procedure in hidden way from a user, but this procedure consumes an extra time and leads to delays in measurements.

Delta-sigma converters use very complicated digital signal processing and as a rule they cannot process any step change of input signal. If voltage is changing on significant value (or on unknown value) as it is in multi-channel measurements, first measured codes may not be authentic. To avoid errors due this effect the micro-controller discards unauthentic (or perhaps unauthentic) codes. These discarded codes are first 4 measured values in multi-channel mode.

The described peculiarities lead to two consequences. At first, if time of measurement is defined 20 ms, the micro-controller will send data into network (or write data into internal memory) with the same rate 20 ms in case of single-channel mode. In case of multi-channel measurements the micro-controller discards first four measured values after changing channel number. It means that actually data rate will be 100 ms. The second consequence is a result of calibration procedure. In digital oscilloscope mode processor performs the calibration procedure once, and then it provides fast measurements for chosen channel. In multi-channel mode the calibration procedure is being performed each start of scanning (before processing first channel). This procedure leads to delay of measured data at 11-12 cycles (approximately 240ms for 20ms range).

A user should keep in mind that ADC effectively rejects an interference with period equal to time of measurements or less in integer times. It is good practice to use ranges 20ms or more. High frequency interference is rejected by ADC and passive circuitry.

#### 4.1. Main modes ADC of CEAD20

A converter of device can work in a few modes as it was mentioned above. The main mode is a multi-channel measurement mode. In this mode the device is involved by CANbus message with code 1. Bit fields of message specify this mode in detail. They define first and last channels in a scanning frame. There is a flag which points if it is required a single cycle of scanning or converter should scan channels up to STOP message. Another flag defines if measured information should be sent to network or it should be stored in on-board memory. One byte contains a label of this mode to allow using a group start message.

In multi-channel mode the device performs a calibration procedure before any measurements. Then micro-controller connects an ADC with input channel defined as first, performs a few measurements, discards possible invalid values, stores correct value in on-board memory and, if it defined, sends it into network. A latest measured data can be requested from on-board memory. Each input channel has a personal location in the memory and on external request the micro-controller sends into network a content of requested location. If requested channel was not measured at all the contents would have an arbitrary value. After processing first channel in frame the micro-processor connects ADC to next input channel and process it. When the last channel in frame was processed micro-controller or goes to idle state (for single frame case) or begins all actions from calibration procedure and so on. If multi-channel mode request contained label (not equal zero) a broadcast message can start measurements all devices with the same value of label simultaneously.

Don't forget that in multi-channel mode an output data rate is five times slower than it should be with defined time of measurements.

For investigation of powers supply behavior here may be used one-channel mode (digital oscilloscope mode). In this mode the device is involved by CANbus message with code 2. Bit fields of message specify this mode in detail. They chose a measured channel, a time of measurement. There is a flag that points if it is required a single measurement or converter should work up to STOP message. Another flag defines if a measured information should be sent in network or it should be stored in ring-buffer of on-board memory. After receiving an one-channel mode command the micro-controller performs a calibration procedure and then begin measure a chosen channel. A data rate corresponds to chosen time of measurements in this mode. Measurements may be stopped by message with code 0. If data are sending into network they aren't storing in on-board memory.

If "send to line" flag is zero then flag "continuous" is considered as 1. So, the device performs continuous measurements and stores data in on-board memory. These values can be requested by message with code 4. A current value of ring-buffer pointer can be requested by message "Status request". The pointer points at measurement number but not at byte.

Actually, ADC of device has 24(48) measurement channels. A 20 differential channels (In0÷In19) from them are intended for an external connection and are connected with connector's pins. Four inputs have on-board connections. 20<sup>th</sup> channel is connected with output temperature sensor, 21<sup>st</sup> channel is connected with power supply, 23<sup>rd</sup> channel is connected with measurements "Ground" and 22<sup>nd</sup> channel is connected with an output of calibration reference source (it voltage is +10V). 22<sup>nd</sup> and 23<sup>rd</sup> channels are used by calibration procedure. Logically all 24 channels are equal. A user can measure any combination of channels. On-board temperature sensor is not intended for accurate measurements. For relative temperature measurements user should read its voltage after power-up and use this value as reference.

On-board temperature sensor is not intended for precise temperature measurements and it is not calibrated. For relative temperature measurements a user should read voltage from this channel after powering device and then use this value as reference.

Hidden (on-board) inputs.

Name	For differential configuration	For single-wire configuration
Temperature	20	40, 44
Power supply voltage	21	41, 45
Reference voltage (+10 V)	22	42, 46
“Ground” voltage	23	43, 47

**Note:**

An ADC data is coded as 24-bit signed integer value. A correspondence between codes and voltages is seen in a table below. A user should take in consideration that some devices allow some overvoltage without loosing accuracy. A user should keep it in mind to reach compatibility of software.

Code (hexadecimal)	Voltage
3FFFFFF	+10 V
000000	+0.0 V
FFFFFFF	-0.0 V
C00000	-10 V

Measurement time

Code decimal	Measurement Time
0	1 ms
1	2 ms
2	5 ms
3	10 ms
4	20 ms
5	40 ms
6	80 ms
7	160 ms



## 5. Protocol (interaction with CANbus) of CEAD20

### Destination of identifier bits

Identifier bits	ID10...ID08	ID07...ID02	ID01...ID00
Bit field	Field 1	Field 2	Field 3
Destination	Priority	Address	Reserve

#### Comments to bit distribution:

Field 1 – priority field (type field):

Code 5 – a broadcast message (field 2 is ignored).

Code 6 – ordinary (address) message.

Code 7 – response (reply for type 6 message).

Code 0 is forbidden, other combination is not used (they are reserved for future extensions).

Field 2 – a physical address field. It defined address device (this address is defined by jumpers on-board). Don't use addresses 34, 3C, 3D, 3E and 3F (hexadecimal).

Field 3: User should set zero in this field. The device can send messages with different values in this field.

Any device on receiving address message interprets an information by its content. If received message requires a reply, the device sends required information by message with code 6 (response type message). A broadcast messages should be received by all devices simultaneously and required actions should be done in all devices. The device doesn't check size of message except for F4 command (sequential write to file).

Size of transmitted messages is not defined and may be different in different versions of device.

#### Data interpretation.

Receiving CANbus message device interprets data the following way: the first byte (byte 0) is interpreted as descriptor of message (command) and the following bytes are parameters.

### 5.1. List of commands (hexadecimal codes).

- 00 - break a measurements procedure
- 01 - defines and starts multi-channel measurements
- 02 - defines and starts one-channel measurements
- 03 - request of a multi-channel value from on-board memory (measured before request)
- 04 - request for value from a ring-buffer
- F8 - request for data from input and output registers
- F9 - write to output register
- FE - device status request
- FF - device attributes request

## 5.2. Detail description of messages (commands) (all codes are hexadecimal)

**Message 00** – break a measurements procedure. There is not additional information. Addressed devices should not reply on this message.

**Message 01** – configuration and start multi-channel measurements procedure. The message looks as:

<b>01</b>	ChBeg	ChEnd	Time	Mode	Label
-----------	-------	-------	------	------	-------

ChBeg- first channel of multi-channel frame.

ChEnd- end channel of multi-channel frame. Channel numbers are from 0 to 47.

Time- time of measurements code. Valid codes are from 0 to 7.

Mode- bit flags to detail procedure.

Label- label for group start command. If label is 0 it means “no label”.

Mode includes the following flags:

Bit 4: 0 means single scanning cycle; 1 means continuous measurements (up to STOP message or message 1 or 2).

Bit 5: 0 means that measured values should be stored in on-board memory and should be sent into network. If this bit is 0 then measured values should not be sent into network.

If a multi-channel measurements was started, a device sends measured data in following message (if bit 5 in Mode is set):

<b>01</b>	Attribute	Low byte	Middle byte	High byte
-----------	-----------	----------	-------------	-----------

Byte Attribute consist of a measured channel number (least 6 bits) which was used during measurement. Other 3 bytes contain measured value.

**Message 02** – configuration and start single-channel measurements procedure. The message looks as:

<b>02</b>	Channel	Time	Mode
-----------	---------	------	------

Channel- contains channel number (least 6 bits) to be measured. Channel numbers are from 0 to 47.

Time- time of measurements code. Valid codes are from 0 to 7.

Mode- bit flags to detail procedure.

Mode includes the following flags:

Bit 4: 0 means single measurement; 1 means continuous measurements (up to STOP message or message 1 or 2).

Bit 5: 1 means that measured values should be stored in on-board memory and should be sent into network. If this bit is 0 then measured values should not be sent into network and device stores them in ring-buffer.

**Comment:** if bit 5 in mode byte is 0 then bit 4 will be ignored (No sense to store a single measurement).

If single-channel measurements were started, a device sends measured data in following message (if bit 5 in Mode is set):

<b>02</b>	Attribute	Low byte	Middle byte	High byte
-----------	-----------	----------	-------------	-----------

Byte Attribute contains a measured channel number (least 6 bits). Other 3 bytes contain measured value. Data in internal buffer are stored in the same format (4 bytes of data and attributes).

**Message 03** – request data from multi-channel buffer (request for previous measured and stored data). The message looks as:

<b>03</b>	Channel
-----------	---------

Channel- a channel number. A request refers to previous measured data for channel specified in this command.

As a response a device sends measured data in following message:

<b>03</b>	Attribute	Low byte	Middle byte	High byte
-----------	-----------	----------	-------------	-----------

Byte Attribute consist of a measured channel number (least 6 bits) which was set at measurement moment. Other 3 bytes contain measured value.

**Message 04** – request data from ring-buffer. The message looks as:

<b>04</b>	Low byte	Middle byte
-----------	----------	-------------

Here low and high bytes compose pointer at current measurement in a ring-buffer. The ring-buffer can hold 128 measurement values. A micro-controller begins to store measured data with pointer value 0. After write al last address (127) a micro-controller continues storing data from first address (0) again. For correct interpretation data (oldest and youngest) user should read current value of ring-buffer pointer. It may be done by “request status” message (code FE).

As a response a device sends measured data in following message:

<b>04</b>	Attribute	Low byte	Middle byte	High byte
-----------	-----------	----------	-------------	-----------

Byte Attribute consists of a measured channel number (least 6 bits) and a gain code (most 2 bits) which was set at measurement moment. Other 3 bytes contain measured value.

**Message F8** - request date from registers. This message has not additional information. In reply a device sends a message with output register byte and input register byte.

<b>F8</b>	Output Register Data	Input Register Data
-----------	----------------------	---------------------

**Message F9** - write data to output register.

Byte 1 contains information to be written into output register. The device don't respond on this message.

<b>F9</b>	Output Register Data
-----------	----------------------

**Message FE** - request device status. It hasn't any parameters. In reply a device sends the following message::

FE	Dev. Mode	Label	Low pADC	High pADC
----	-----------	-------	----------	-----------

Here:

Device Mode- bit field. There are flags:

Bit 4 – SCAN- if this flag is set it means that a device process multi-channel measurements procedure.

Bit 3 – RUN- if this flag is set it means that a device process measurements procedure (multi-channel or one-channel).

Bit 2,1,0 – reserved.

Label- label value for ADC.

Low and high bytes of pADC compose a ring buffer pointer. It points at location that would be updated by next measurement. If ring-buffer was overwritten (after long time) the pointer points at oldest data.

**Message FF** - device attribute request. There is not additional parameters. In reply a device sends the following message:

FF	Device Code	HW version	SW version	Reason
----	-------------	------------	------------	--------

Device Code- device type (for CEAD20 it is equal 23).

HW version- hardware version of device. Bit b0 always is 0, bits b1 and b2 equal states of X8 and X9 jumpers.

SW version- software version of device.

Reason- reason of sending this message:

- 0 - After power-up.
- 1 - After reset by button on front panel.
- 2 - On request by address message with code FF.
- 3 - On request by broadcast message (who is here?).
- 4 - On restart by Watchdog timer.
- 5 - On busoff recovery.

### 5.3. GLOBAL messages (broadcasts)

For broadcast messages all devices analyze only field 1 in CANbus identifier. Valid combination is 5. A first byte of data presents a broadcast command. CEAC124 uses the following broadcast commands:

3 - STOP- to stop ADC measurements procedure.

4 - group START of ADC measurements, group code (label) is placed in second data byte.

FF- request "Who is here". On this broadcast request all devices on-line must send into network message with their attributes (and identifier).

FF	Device Code	HW version	SW version	Reason
----	-------------	------------	------------	--------

Device Code- device type (for CEAD20 it is equal 23).

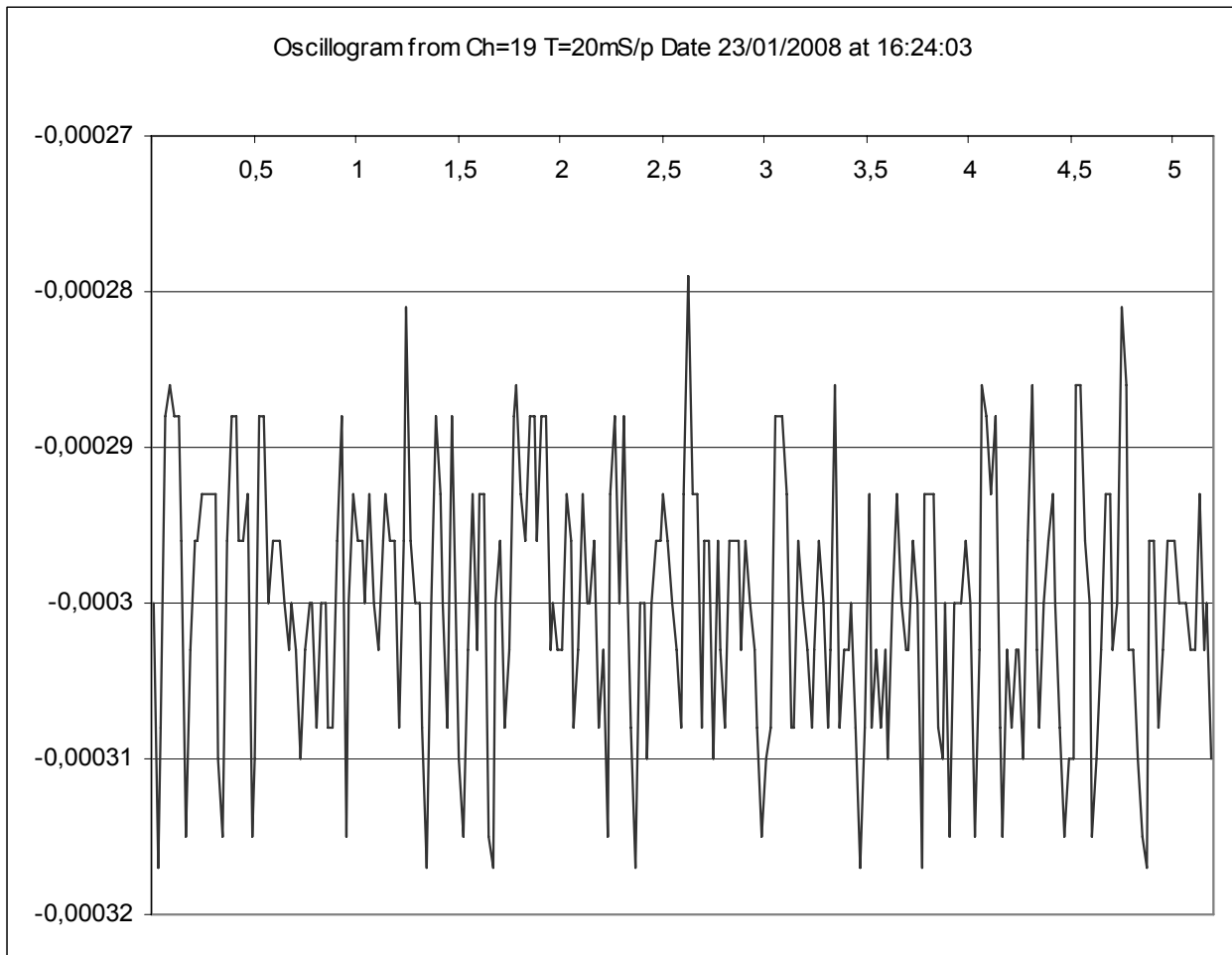
HW version- hardware version of device. Bit b0 always is 0, bits b1 and b2 equal states of X8 and X9 jumpers.

SW version- software version of device.

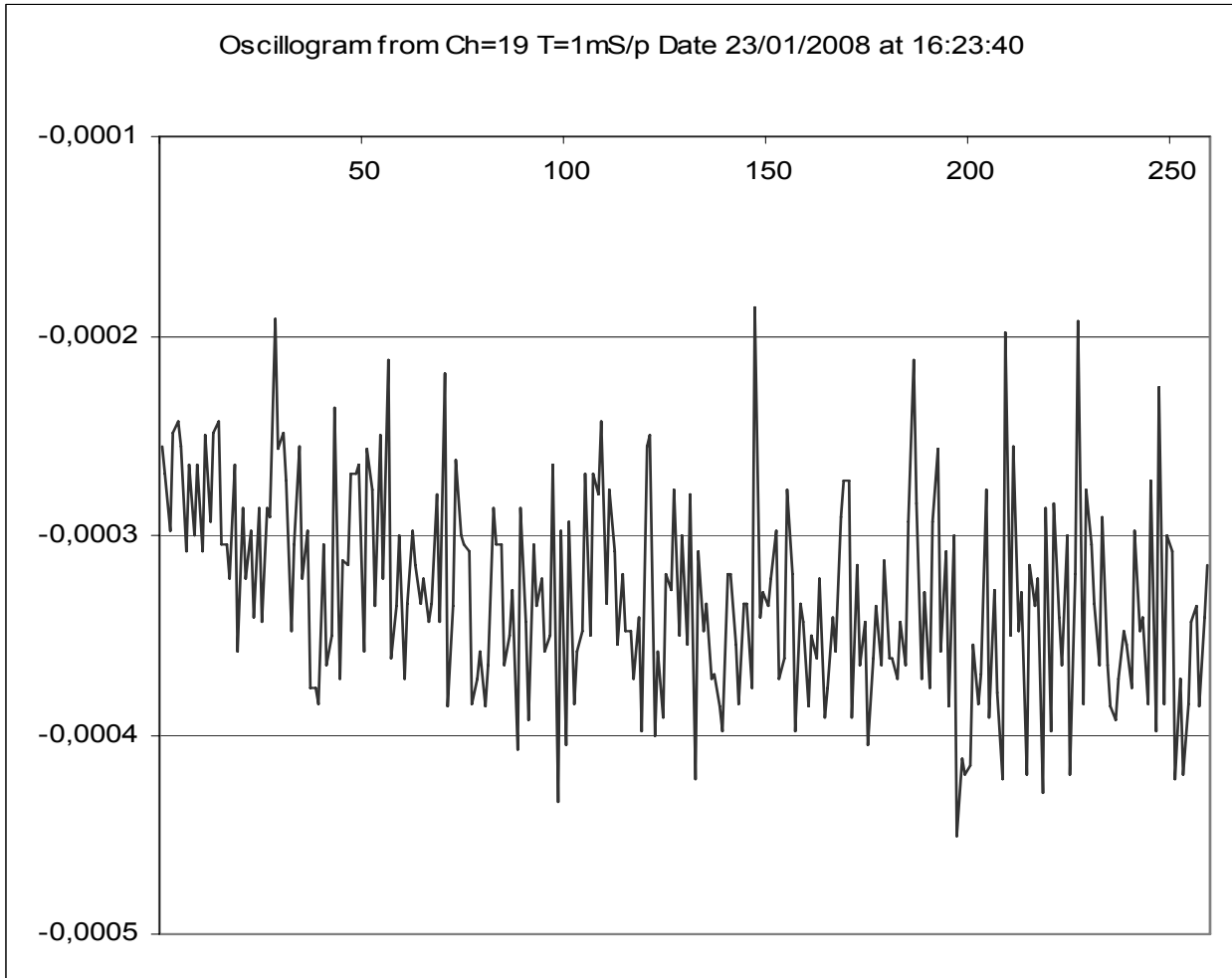
Reason- reason of sending this message:

- 0 - After power-up.
- 1 - After reset by button on front panel.
- 2 - On request by address message with code FF.
- 3 - On request by broadcast message (who is here?).
- 4 - On restart by Watchdog timer.
- 5 - On busoff recovery.

## 6. Some typical pictures for CEAD20



A typical noise of ADC. A data were collected in single-channel mode, data rate was 20 ms/measurement. Axis are voltage (V, 1 division is 10 uV) and time (Sec).



A typical noise of ADC. A data were collected in single-channel mode, data rate was 1 ms/measurement. Axis are voltage (V, 1 division is 100 uV) and time (mS).

## 7. Software versions for CEAD20

### **Version 2.**