

CDF600-0300
Field bus module

Preliminary



Module for connecting a SICK bar code scanner
CLV6xx to EtherCAT® networks



Index of Technical Information (TI) – Integrator Commissioning Description

1.	Function of this Document	3
1.1	Integrator Commissioning Description	3
1.2	Target Group	3
1.3	Depth of Information	4
2.	Electrical Installation	5
2.1.	Avoiding equipotential bonding currents in the cable shields	5
2.2.	Electrical connections and cables	7
2.3.	Pin Assignment	7
2.4.	Device Description	9
2.5.	Connecting the sensor to the field bus module	11
2.6.	Wiring switching inputs at the field bus module	12
2.7.	Wiring switching inputs at the field bus module	13
3.	Installation	14
3.1.	Switches	19
3.2.	Mode	19
4.	CDF600-0300 Software	20
4.1.	CDF600-0300 Communication mode	20
4.2.	Digital 'status'/'control' Word	21
4.3.	Confirmed Messaging Protokoll (CM)	21
4.3.1.	Initialization	23
4.3.2.	Resynchronisation	23
4.3.3.	Transmitting unblocked individual telegrams	24
4.3.4.	Transmitting blocked telegrams	24
4.3.5.	4.8.3 The Binary Inputs and Binary Outputs Bytes	26
5.	EC Declaration of Conformity	28
6.	EtherCAT Conformity declaration	29
7.	Further Reading	30

Please read this chapter carefully before you use the CDF600-0300

1. Function of this Document

1.1 Integrator Commissioning Description

This additional technical information is a supplement to the operating instructions of the CDF600-0300 (8013919/2011-08-10) and should be used as an integrator commissioning description.

1.2 Target Group

Target of this document are people for the following activities:

Integrators, who have proven to VdS conformant construction and installations which are admitted by VdS.

1.3 Depth of Information

The content is valid for the CDF600-0300 in combination with CLV6xx bar code scanners:

- Product description CDF600-0300
- Ordering Information
- Mounting
- Electrical Installation
- Commissioning and Configuration

This technical information contains the following information:

- Maintenance
- Fault, error diagnosis and troubleshooting
- Technical information and dimensional drawings
- Ordering information
- Conformity and approval

Notice The CDF600-0300 field bus module meets the requirements for Class A (industrial environment) as defined by the generic specifications 'Emissions'. It may cause radio interference in homes. In this case, the party suffering the interference can demand that the operator takes appropriate radio interference suppression measures.

- Use the CDF600-0300 field bus module only in industrial environments.

Recommendation The latest version of this technical information can be obtained as PDF file at www.mysick.com/en/CDF600-0300

Further information on the CDF600-0300 is available from SICK AG, Division Auto Ident, and in the Internet at www.mysick.com/en/CDF600-0300

Copyright notices Copyright © 2011
SICK AG
Auto Ident, Reute Plant
Nimburger Straße 11
79276 Reute
Germany

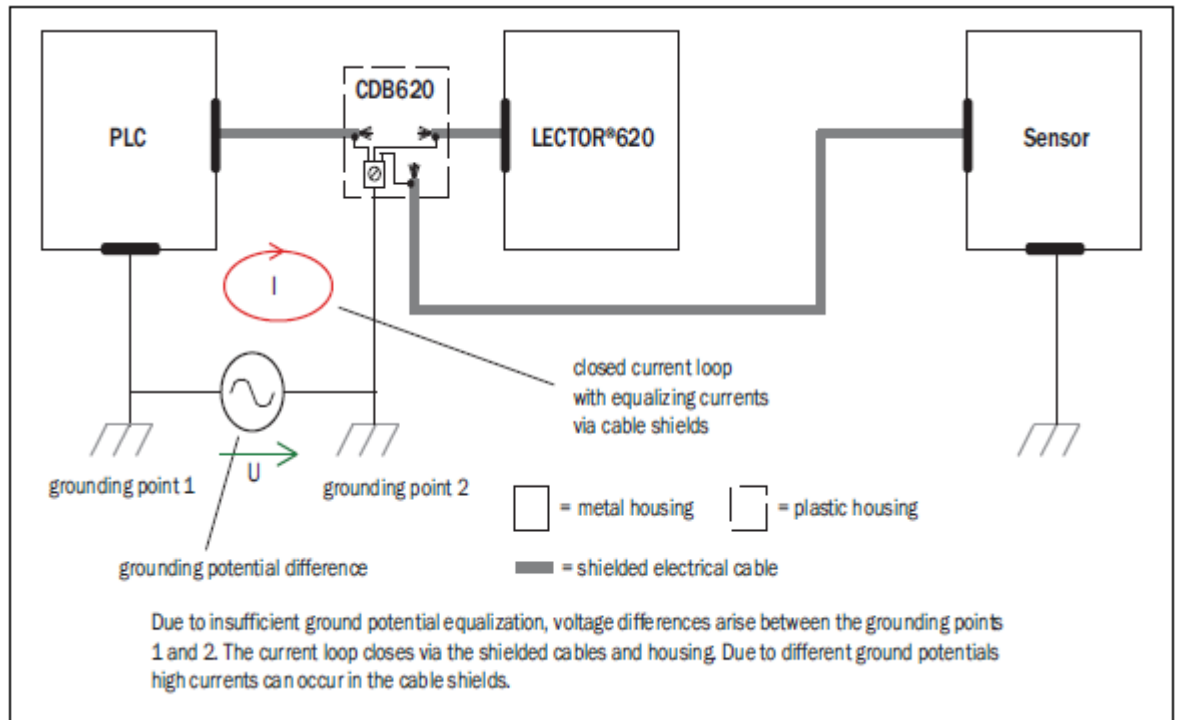
'EtherCAT® is registered trademark and patented technology, licensed by Beckhoff Automation GmbH, Germany.'

Windows 2000™, Windows XP™, Windows Vista™ and Internet Explorer™ are registered trademarks of Microsoft Corporation in the USA and other countries.
Acrobat® Reader™ is a trademark of Adobe Systems Incorporated.

2. Electrical Installation

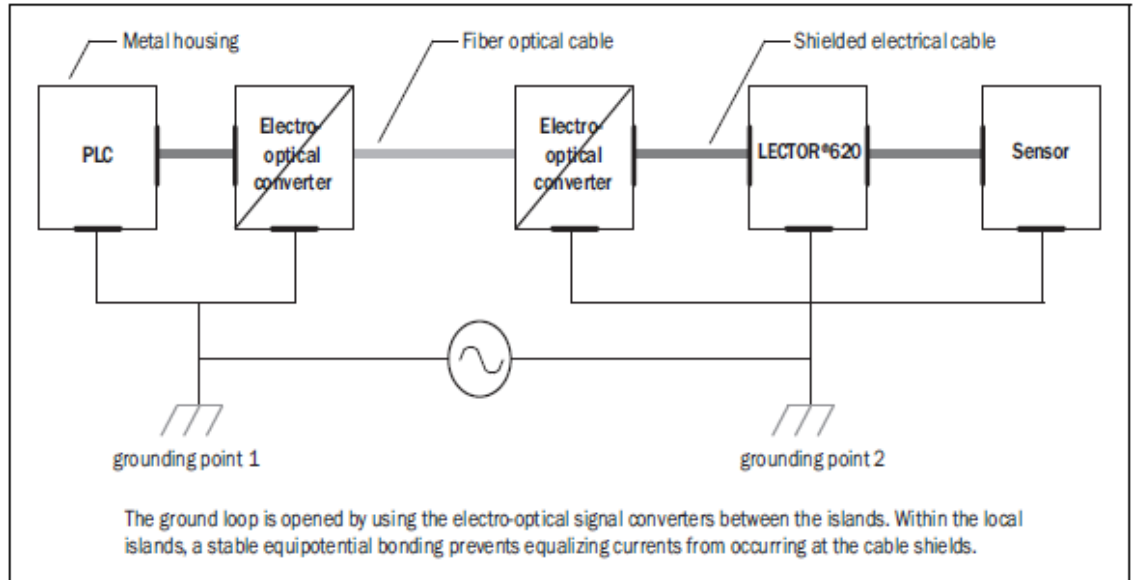
2.1. Avoiding equipotential bonding currents in the cable shields

As a result of the different ground potentials of devices in a distributed system, high currents can occur in the cable shields and damage or destroy them.

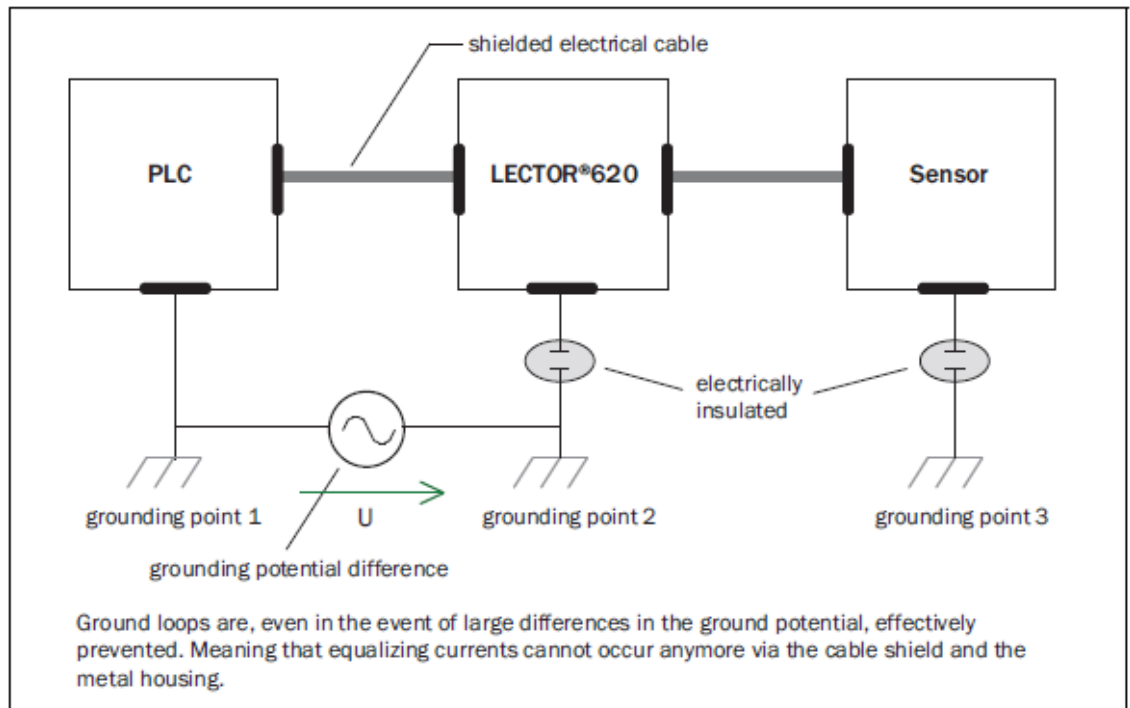


The primary solution for avoiding equipotential bonding currents on the cable shields is to ensure low-impedance equipotential bonding that is able to carry currents. If this is not possible, the two solutions provided below can serve as a suggestion. We expressly advise against the measure suggested in EN 60950 of disconnecting the cable shields. If this measure is carried out, there is no guarantee that the EMC thresholds can be observed or that the devices' data interfaces will operate reliably.

In the case of system installations distributed over wide areas, with the corresponding high differences in potential, we recommend the setup of local islands and the connection of these islands via commercially available optical signal conditioners. This measure results in a high degree of robustness against electromagnetic interference while at the same time meeting all requirements of EEN 60950



or smaller installations with only small potential differences, the isolated installation of the CDF600 and peripherals may be sufficient.

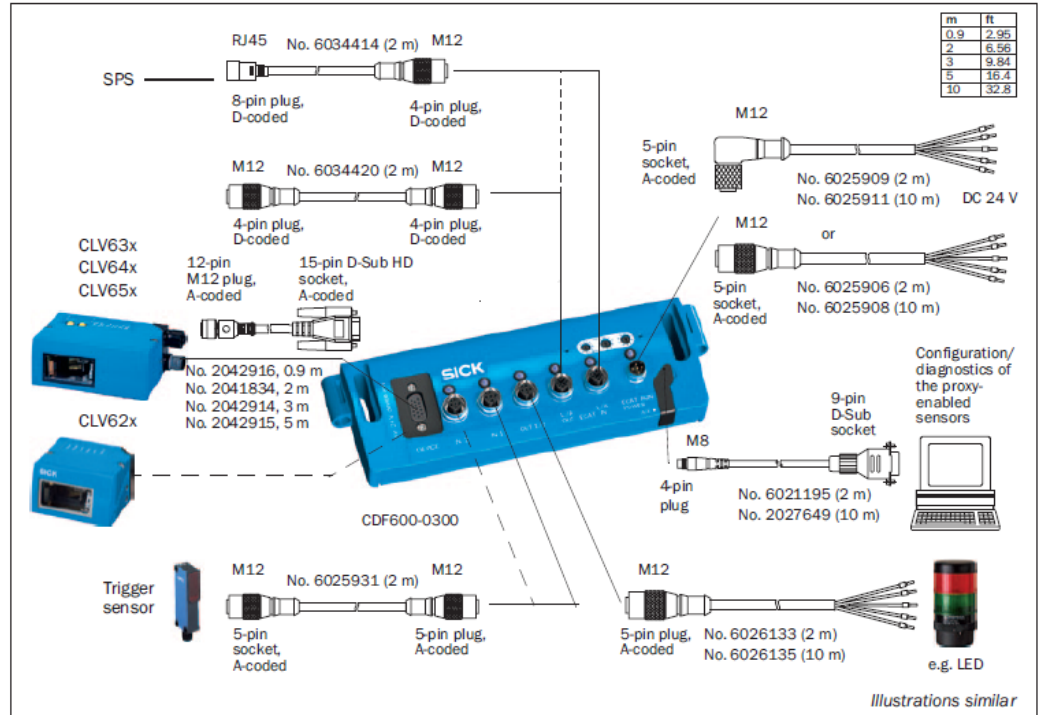


The power supply for the CDF600 and the connected periphery must then also guarantee the required isolation.

In some cases a tangible potential may arise between the isolated metal housings and the local ground.

2.2. Electrical connections and cables


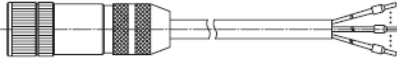
Overview of all interfaces and connection options



Overview: CDF600-0300 connections and cables with M12/M8 plug connectors

2.3. Pin Assignment

Cable article no. 6025906/6025908 (supply voltage)

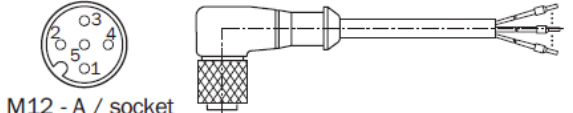
M12 - A / socket

Pin	Signal	Function	Color of lead
1	V _s in	Supply voltage IN	brown
2	CAN L	CAN bus ^{*)}	white
3	0 V in	0 V	blue
4	CAN H	CAN bus ^{*)}	black
5	n.c.	-	gray

^{*)} For CLV6xx bar code scanners

Pin assignment of the 5-pin M12 socket (A-coded), straight/lead colors on the open end

Cable article no. 6025909/6025911 (supply voltage)



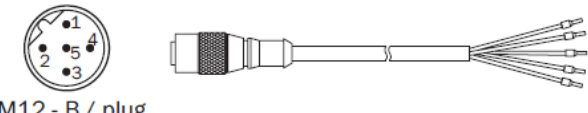
M12 - A / socket

Pin	Signal	Function	Color of lead
1	Vs in	Supply voltage IN	brown
2	CAN L	CAN bus *)	white
3	0 V in	0 V	blue
4	CAN H	CAN bus *)	black
5	n.c.	-	gray

*) For CLV6xx bar code scanners

Pin assignment of the 5-pin M12 socket (A-coded), at a 90° angle/lead colors on the open end

Cable article no. 6026133/6026135 (output signal switching devices)

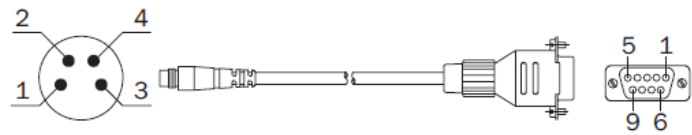


M12 - B / plug

Pin	Signal	Function	Color of lead
1	n.c.	-	Brown
2	Ext OUT2	External output 2	White
3	GND	Ground	Blue
4	Ext OUT1	External output 2	Black
5	n.c.	-	Green/ yellow

Pin assignment of the 5-pin M12 plug (A-coded), straight and lead colors on the open end

Cable article no. 6021195/2027649 (RS-232: CDF600 on PC)



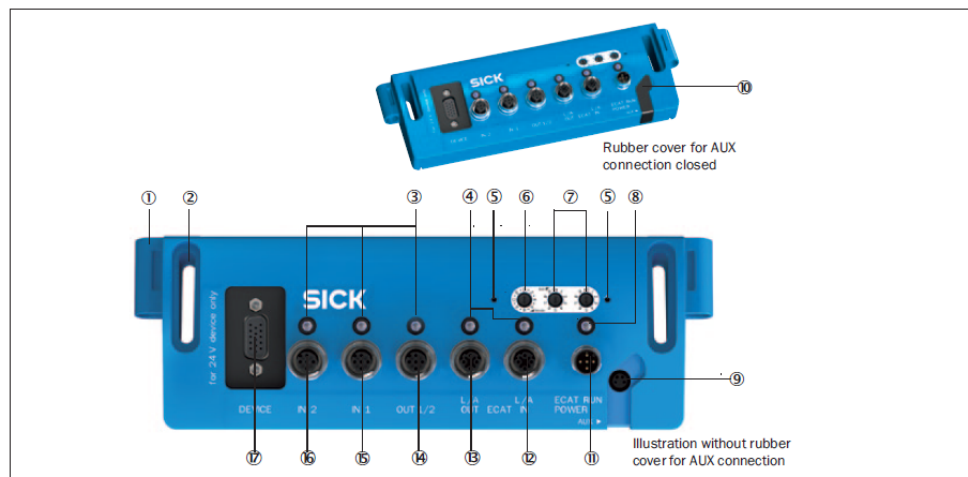
Pin	Signal	Color of lead	Signal	Pin
4	TxD	Black	RxD	2
2	RxD	White	TxD	3
3	GND	Blue	GND	5
1	-	Brown	-	-

Pin assignments of the 4-pin M8 plug and the 9-pin D-Sub socket

"DEVICE" connection

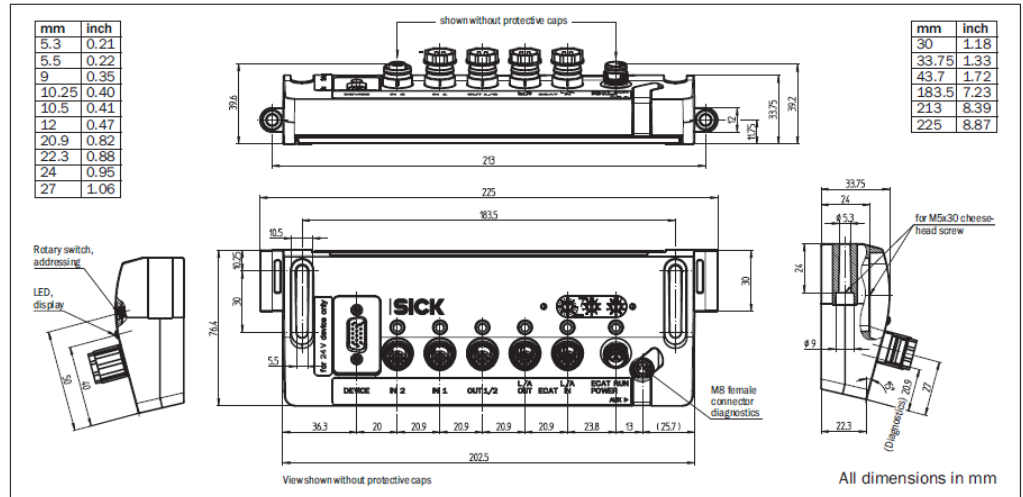
Pin	Signal	Function
1	V _S out	Supply voltage OUT
2	TxD (Aux)	Transmitter
3	RxD (Aux)	Receiver
4	n.c.	-
5	GND	Data ground reference potential
6 to 9	n.c.	-
10	CAN H	CAN bus
11	CAN L	CAN bus
12 to 14	n.c.	-
15	SensGND	Ground

2.4. Device Description



View of the CDF600-0300 fieldbus module, electrical connections shown without yellow protective caps/plugs here.

- ① 1 Dome (2 x), for mounting with M5 screw
- ② Oblong hole (2 x), unpainted, for mounting with M5 screw, alternative for ①
- ③ LED for status display (7 x, OUT1/2 doubly occupied)
- ④ LED for L/A (Link/Activity) for ECAT IN and ECAT OUT
- ⑤ M2.5 threaded bolt (2 x) for mounting the optional cover no. 2052296 for the rotary coding switch
- ⑥ Rotary coding switch for operating mode
- ⑦ Rotary coding switch (2 x) reserved for bus address, no function provided for EtherCAT®.
- ⑧ LED for ECAT RUN
- ⑨ AUX connection (4-pin M8 socket)
- ⑩ Rubber cover for the AUX connection
- ⑪ POWER connection (5-pin M12 plug, A-coded)
- ⑫ ECAT IN connection (4-pin M12 socket, D-coded)
- ⑬ ECAT OUT connection (4-pin M12 socket, D-coded)
- ⑭ OUT 1/2 connection, ECAT status (5-pin M12 socket, A-coded)
- ⑮ IN 1 connection (5-pin M12 socket, A-coded)
- ⑯ IN 2 connection (5-pin M12 socket, A-coded)
- ⑰ DEVICE connection (15-pin D-Sub HD socket)



2.5. Connecting the sensor to the field bus module

The communication between the field bus module and the sensor runs via serial AUX interface (RS-232) between the field bus module and the sensor. All states of the local switching inputs of the CDF are also transferred via this data interface.

Place the 15-pin D-Sub-HD connector of the sensor cable onto the corresponding socket of the device connection on the field bus module and lock it in place with the aid of the two screws.

Connecting the sensor with heating to the field bus module

The incoming power supply cable (DC 24V +/- 10%) for the heating of the sensor must be protected by a fuse of 3 to 4 A in the control cabinet. For each field bus module combined with a bar code scanner a separate fuse is required.

The power supply voltage of the CDF is routed via the temperature-dependant switc of the heating. First only the heating is fed with electricity. After the time of the warmup-phase (approx. 40 min at -35° C (-31° F) and 24 V Dc power supply), the switch releases the supply voltage for the CDF. The CDF noow supplies the electronics of the device.

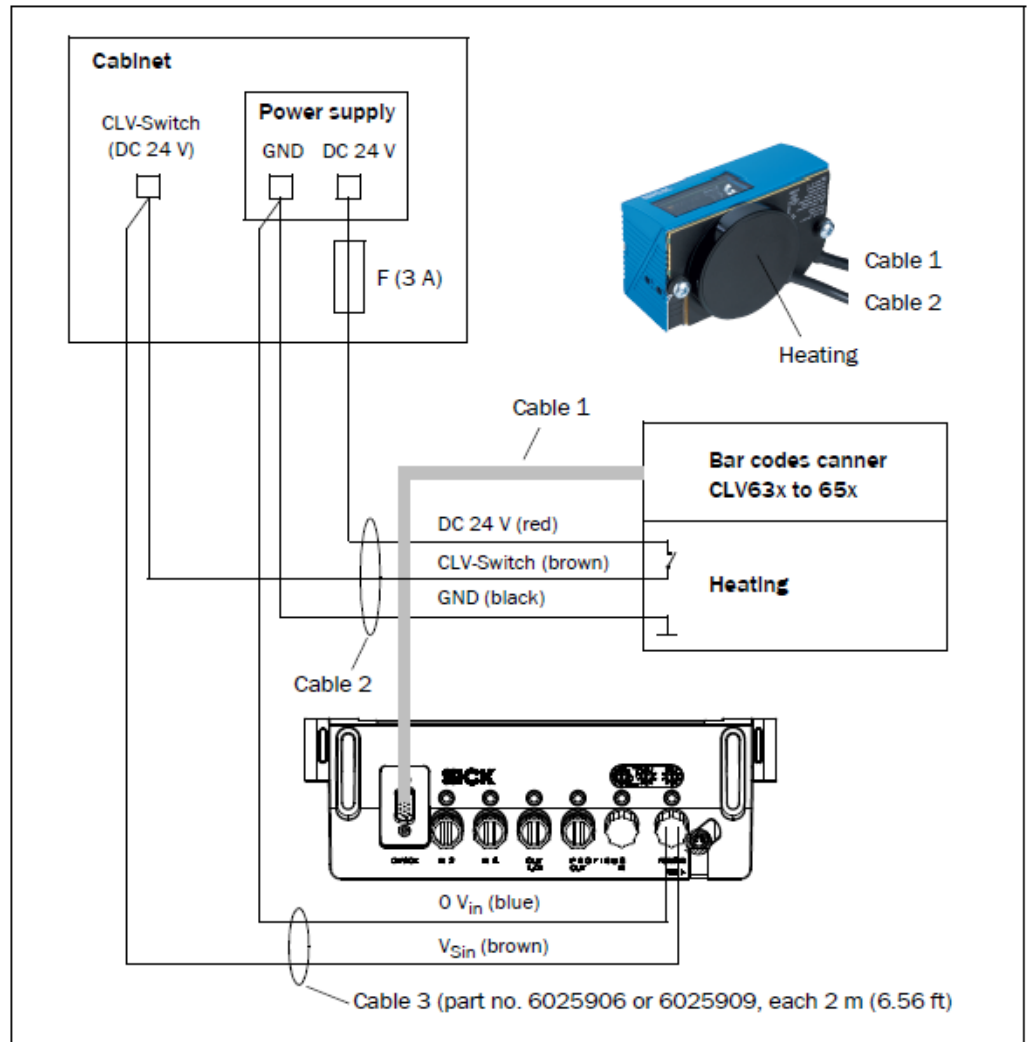


Fig. 6-7: Wiring the power supply voltage for the field bus module combined with a bar code scanner CLV63x to 65x with heating in the control cabinet

2.6. Wiring switching inputs at the field bus module

The digital switching inputs IN 1 and IN 2 of the field bus module can be used to e.g. connect a reading pulse sensor (e.g. light barrier) that triggers the reading process of the bar code scanner. A sensor with a PNP output is best suited for this.

Depending on the type of bar code scanner connected and its parameterization, the inputs also serve as

- Reading pulse generator for reading pulse end
- Trigger for the focus position switching (CLV640)
- Trigger for teaching in a match code (Teach-in)
- Triggering of the oscillating mirror (CLV63x and CLV640)

The function assignment and the applied logic of the switching input is configured in the bar code scanner in SOPAS-ET as EXT. INPUT 1 and EXT. INPUT 2.

For connecting the switching inputs, the assembled cable part no. 6025931 (2 m (6.56 ft) with a 5-pin M12 plug and a 5-pin M12-socket is available in each case. It is used to directly connect e.g. corresponding SICK sensors (e.g. light barriers).

Fig. 6-8 shows as an example the wiring of the input IN 1 Table 6-13 contains the ratings for the input IN 1 and the input IN 2.

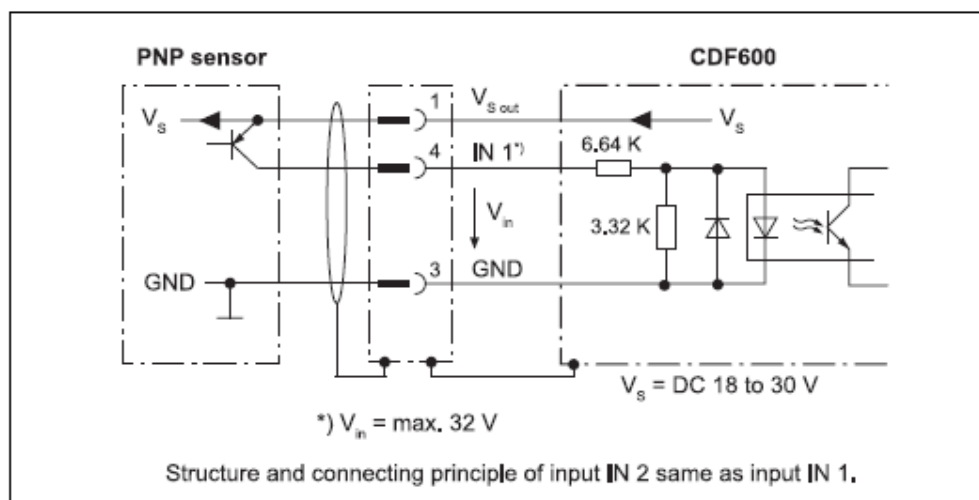


Fig. 6-8: Example wiring of the switching input IN 1 at the 5-pin M12 socket (A-coded)

- On the connection IN 1 or IN 2 of the field bus module, remove the protective plug, place the 5-pin M12 cable plug (A-coded) into position and lock it in place. Connect the opposite side of the cable to the corresponding plug of the sensor (e.g. light barrier).

Switching behaviour	Power fed to the input starts the assigned function in the bar code scanner. (Default setting: level: active high; debouncing: 10 ms)
Features	<ul style="list-style-type: none"> - Opto-decoupled, reverse polarity protected - Can be wired with the PNP output of a sensor - GND of IN 1 connection connected to GND of IN 2 connection
Electrical values	Low: $ V_{in} \leq 2 \text{ V}; I_{in} \leq 0.3 \text{ mA}$ High: $6 \text{ V} \leq V_{in} \leq 32 \text{ V}; 0.7 \text{ mA} \leq I_{in} \leq 5 \text{ mA}$
LED IN 2, LED IN 1 (yellow)	Low: off High: on

2.7. Wiring switching inputs at the field bus module

The two digital switching outputs OUT 1 and OUT 2 are jointly available at the OUT 1/2 connection of the field bus module. They can each be assigned various functions independently of each other. If the assigned result occurs in the reading process, the corresponding switching output at the end of the reading pulse is live for the selected impulse duration (exception: static signalling of Device Ready).

The function assignment and the applied logic of the switching outputs is configured in the bar code scanner in SOPAS-ET as Ext. OUTPUT 1 and Ext. OUTPUT 2 configured.

Important The "Result" LED of the bar code scanner is not connected to any switching output.

The assembled cable is available for the common wiring of the switching outputs part no. 6025931 with a 5-pin M12 plug and a 5-pin M12 socket.

Fig. 6-9 shows as an example the wiring of the output OUT 1, Table 6-14 contains the ratings for the output OUT 1 and the output OUT 2.

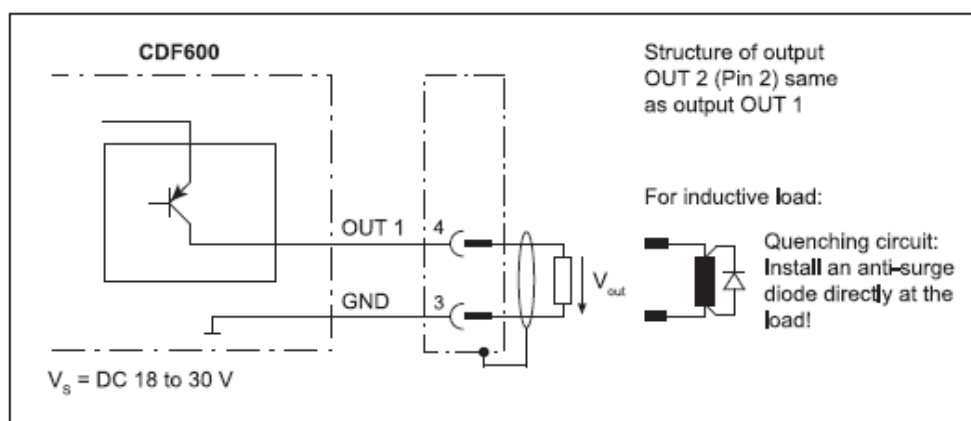


Fig. 6-9: Example wiring of the switching output OUT 1 at the 5-pin M12 socket (A-coded)

Switching behaviour	PNP switching against the supply voltage V_S , high-side switch
Features	<ul style="list-style-type: none"> - Short-circuit proof + temperature-protected - Galvanically not separated from V_S
Electrical values	$0 \leq V_{out} \leq V_S$ Guaranteed: $(V_S - 1.5 V) \leq V_{out} \leq V_S$ at $I_{out} \leq 100 \text{ mA}$
LED OUT 1/2 OUT 1: green OUT 2: yellow	Low: off High: on

Table 6-14: Ratings for the OUT 1 and OUT 2 switching outputs

Important Capacitance loads at the switching output affect the switching behaviour. Threshold is a max. capacitance of 100 nF. Exceeding this value can lead to unwanted pulsing behaviour of the output.

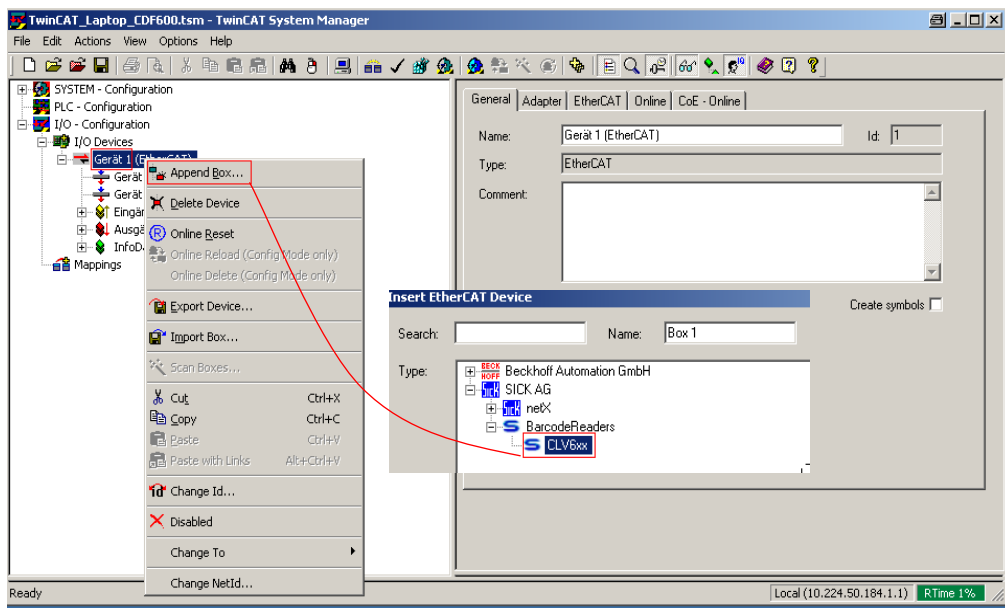
1. Connect switching output depending on application.
To do this, on the connection OUT 1/2 of the field bus module, remove the protective plug, place the 5-pin M12 cable plug (A-coded) into position and lock it in place.
2. Wire the switching output with a load resistance to test the switching functions using a high-resistance digital voltmeter.
Indication of incorrect voltages/switching statuses is avoided this way.

3. Installation

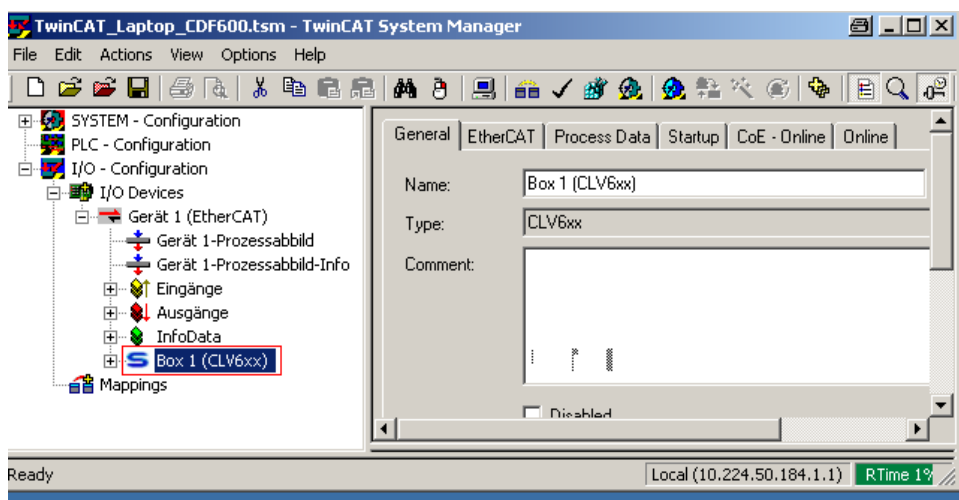
This is a step-by-step installation description how to integrate the CDF600-0300 in the TwinCAT system. The CDF600-0300 will be displayed with the name of the connected identification sensor (e.g. CLV6xx).

Step 1: Configuration in TwinCAT

Start TwinCAT System Manager. Select 'CLV6xx' on the defined EtherCAT device append box.

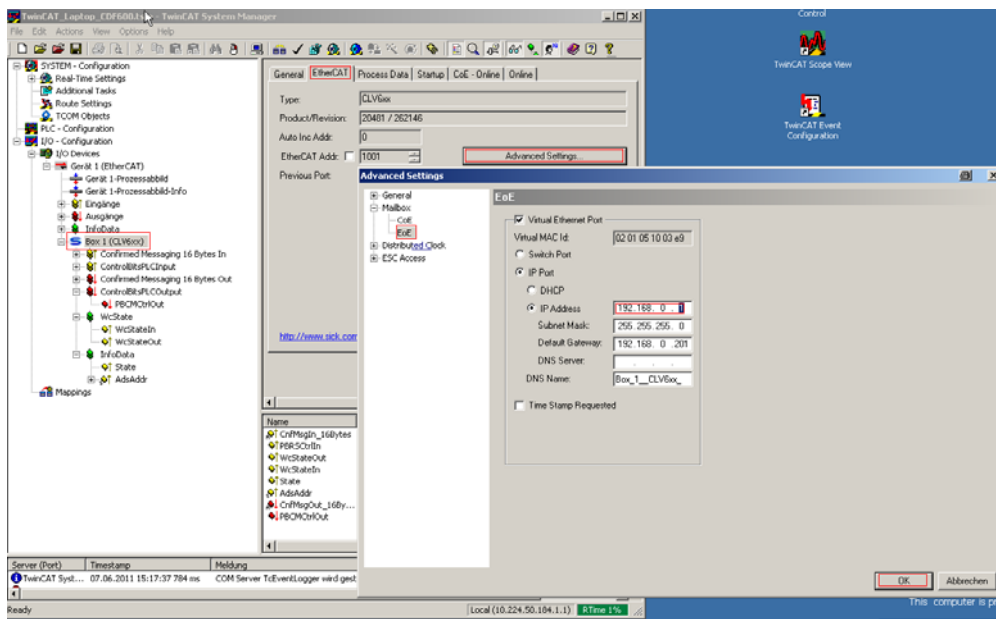


The bar code scanner CLV6xx is integrated in configuration area!



Step 2: EoE configuration in TwinCAT

Notice EoE is used for parameterization of the sensor via PLC to the CDF600-0300. EoE is not used for the operational data exchange.



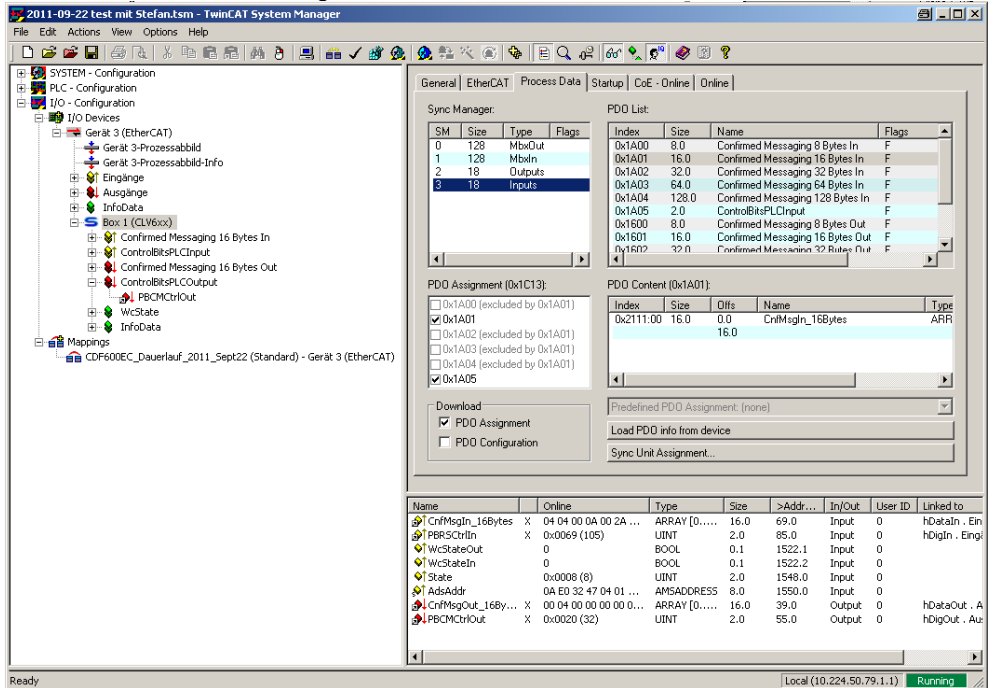
Notice With an error message like 'Unspecified error: 'eoe_init'' you have to change the IP address for the EoE connection. The default usage of the network subnet address (e.g. 192.168.0.0) will not work.

To do: Set valid IP address (e.g. 192.168.0.1)

1. Select 'CLV6xx' in the device tree.
2. Select 'Advanced Settings ..' in the 'EtherCAT' tab.
3. In the 'Mailbox/EoE' selection the 'virtual EthernetPort' contains the 'IP Address'.
4. Set IP address for this sensor (e.g. CLV6xx) different to other EoE devices in the network.

Step 3: PDO Configuration in TwinCAT

The user can select the best suited configuration for the communication situation. Inside the ESI file different module configurations are available.

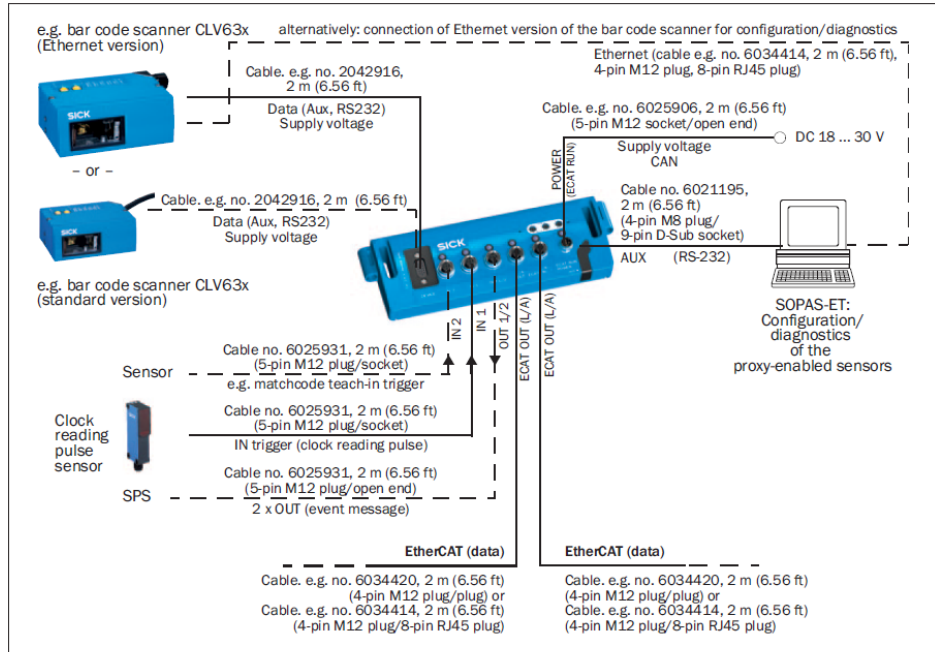


In the selection above a configuration with 16 Byte (5 byte 'header' + 11 byte 'data') data module and the 'status' word for digital input is marked.

Step 4: Connecting Device

Connect SICK identification sensor (e.g. CLV6xx) to 'DEVICE' socket on the CDF600-0300. Connect the CDF600-0300 via its 'ECAT IN' and 'ECAT OUT' sockets in the desired position in the network.

CDF600-0300 Fieldbus Module for EtherCAT®



Block diagram of CDF600-0300 fieldbus module in connection with CLV6xx bar code scanner

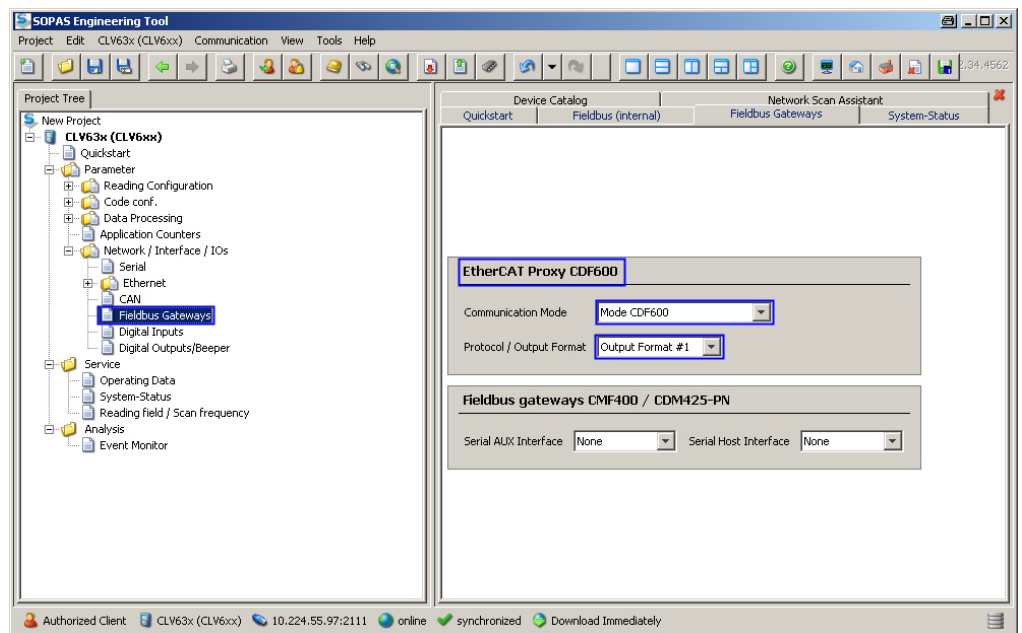
Step 5:

After power-up the CDF600-0300 the 'ECAT RUN' LED begins flickering according the run level progress (see 'ECAT RUN' LED table). In 'operational' state of the device the ECAT RUN LED is permanently on. For CLV6xx a minimum version of V5.10 is required.

Step 6:

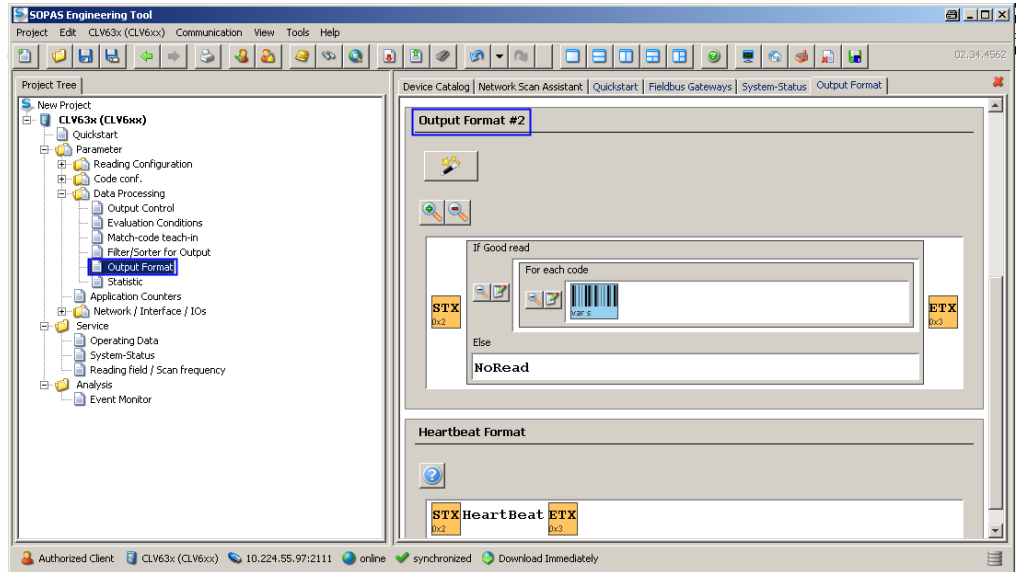
Setup the sensor in SICK SOPAS ET configuration software for the reading situation (e.g. 'Object Trigger').

In the 'communication protocol' the handshake mode can be activated



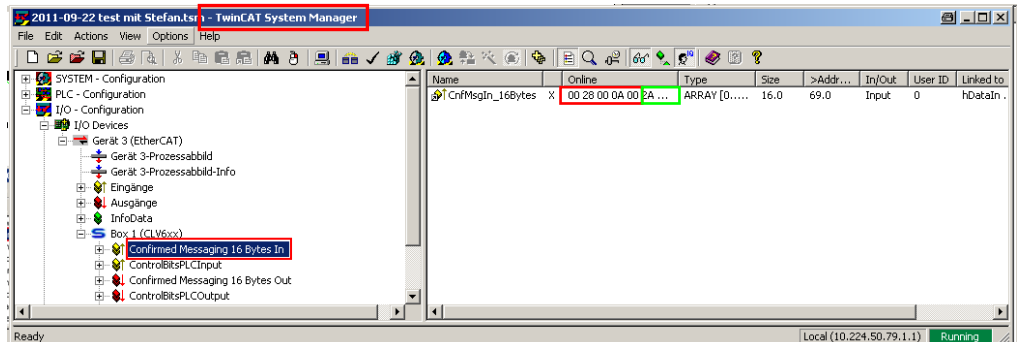
CDF600-0300 Fieldbus Module for EtherCAT®

The selected 'Protocol/Output Format' may be defined in the 'Output Format' section of the CLV6xx configuration in SOPAS ET.



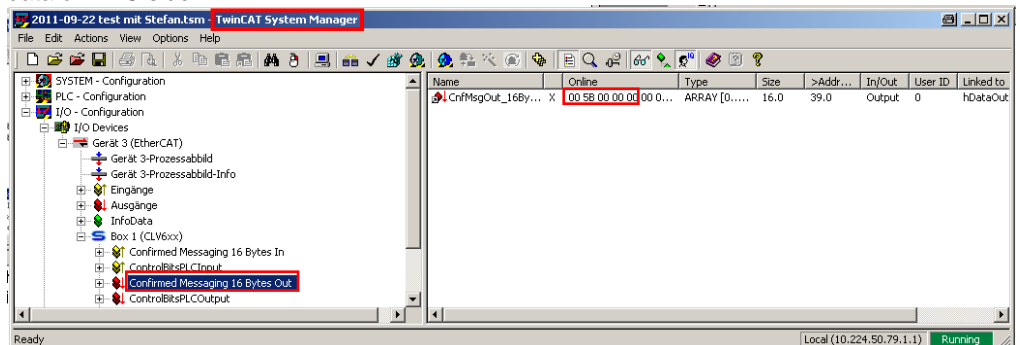
Step 7: Communication from sensor to PLC

After the sensor has got a trigger the read result string is transmitted via the data module to the PLC.



The data module has the header part, red marked, with the counters and the data part with the read result string, green marked.

In the example via TwinCAT the RecCntBack counter is incremented to confirm the received data on PLC side.



3.1. Switches

There are 3 rotary coding switches on the CDF600-0300.

'Mode' (S1) to connect either 'Sopas'/'Non-Sopas' devices

'Adr' (S2/3) is reserved (not used for EtherCAT)



3.2. Mode

The Mode switch is observed by the CDF600-0300. Here the mode of the CDF600-0300 device is set during power-up (no subsequent interpretation after power-up).

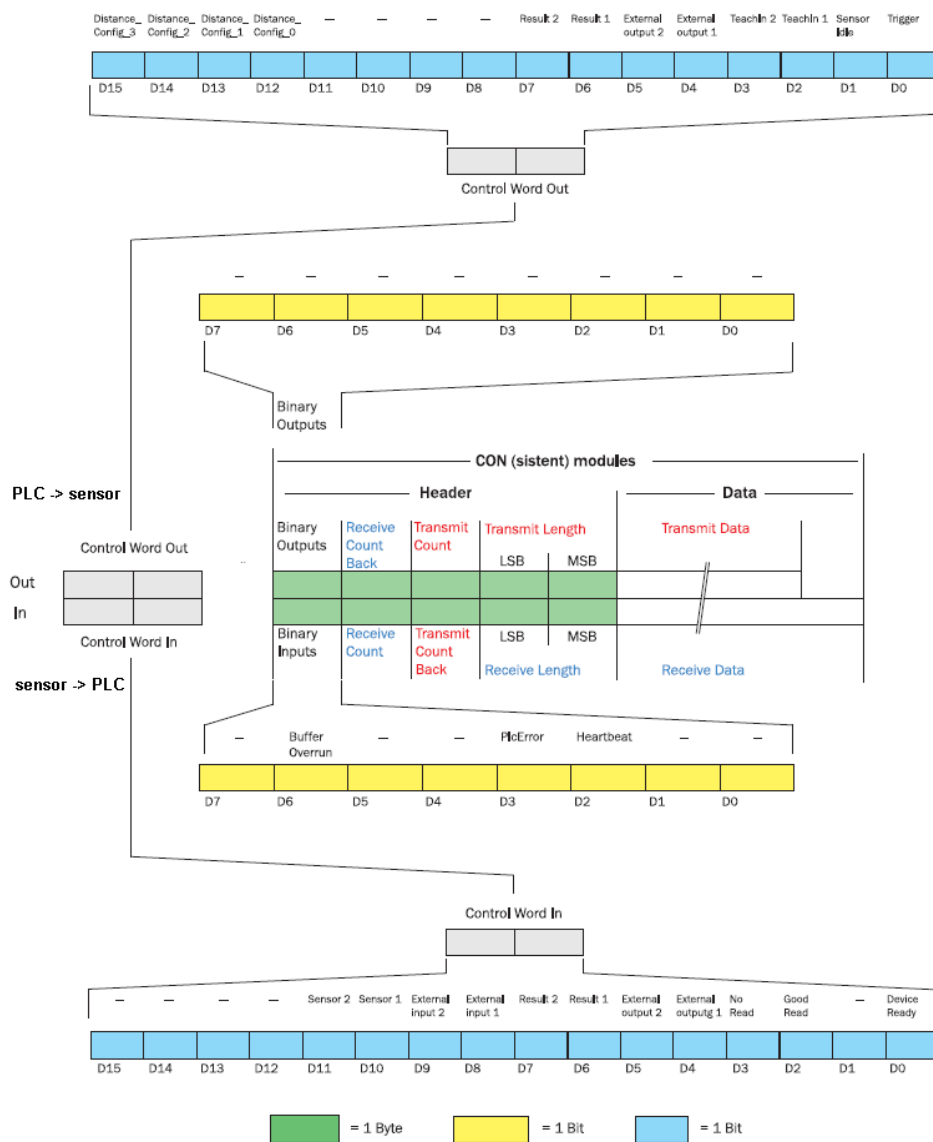
No	Description
0	Sopas Proxy / CLV6xx (parameter cloning)
1	Sopas Proxy / CLV6xx (parameter cloning)
2	RS232 Gateway (57,6 kBd)
3	RS232 Gateway (57,6 kBd)
4	RS232 Gateway (9,6 kBd)
5	RS232 Gateway (9,6 kBd)
6	reserved
7	reserved
8	reserved
9	reserved
A	reserved
B	reserved
C	reserved
D	reserved
E	CDF FW up-date with 115kBd via AUX
F	transparent mode (57kBd, no CDF functionality, use for Sensor FW up-date)

4. CDF600-0300 Software

The CDF600-0300 software and the sensor parameters are stored in the internal flash memory. Software and parameters can be restored multiple times.

4.1. CDF600-0300 Communication mode

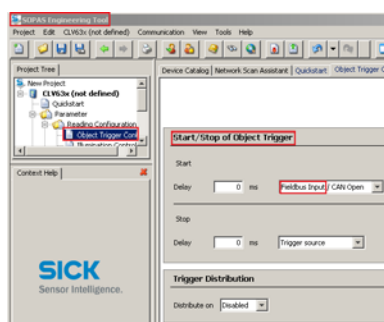
This mode provides data modules for communication (e.g. read result from sensor) in both directions. Additionally a 'status' word for digital bit information from sensor and a 'control' word for the PLC side to control sensor activity.



4.2. Digital 'status'/'control' Word

'status' word provides information from sensor
 'control' word is set by PLC and controls sensor

Note: Additional sensor parameterization required!



4.3. Confirmed Messaging Protokoll (CM)

The CDF600-0300 EtherCAT variant only supports the communication mode 'CDF600-0300' (with/without handshake) and digital I/Os in 'Control Word In' and 'Control Word Out'

Important Differences between the two CDF600-0300 modes:

- The mode with handshake is used for data exchange in normal operation. If a telegram is not acknowledged by the field bus master after a timeout of 10 s, the fieldbus module resets the received input data (ReceiveCount, ReceiveData). The fieldbus module identifies this handshake error by setting the *PlcError-Bit* in *Binary Inputs*.
- The mode without handshake is used for test purposes in data transfer. Here, the following deviations from normal operation apply:
 - No time monitoring for the byte handshake
 - No fragmentation (the required data string length is limited by the set useful data length)

The CDF600-0300 mode requires consistent data transfer of the header and data for the string transfer. Additionally, in CDF600-0300 modus, it is possible to configure modules with digital switching input and output functions.

The communication procedure between the field bus module and the field bus master is defined with 5 control bytes in the header. The transmission method is described below. This transmission method can be easily handled in the field bus master (PLC). Two structures comprising a header and the current user data are created for the cyclical input and output data of the field bus. In the header, a handshake uses counters to control transmission of the user data.

The table below shows the address layout regarded from the field bus master.

Address	Input	Output
0	Binary Inputs	Binary Outputs
1	ReceiveCount (counter)	ReceiveCountBack (counter)
2	TransmitCountBack (counter)	TransmitCount (counter)
3	ReceiveLength Lowbyte	TransmitLength Lowbyte
4	ReceiveLength Highbyte	TransmitLength Highbyte
5	ReceiveData, Byte 1	TransmitData, Byte 1
6	ReceiveData, Byte 2	TransmitData, Byte 2
...
...	ReceiveData, Byte ...	TransmitData, Byte ...

BLUE: Receive telegrams (e.g. reading data from bar code scanner via the field bus module to the field bus master)
 RED: Transmit telegrams (e.g. commands from field bus master to the bar code scanner via the field bus module)

Table 4-8: Overview: Address layout regarded from the field bus master

Input data format of the field bus master

Binary Inputs	Receive Count	Transmit CountBack	ReceiveLength	ReceiveData
8 bits	8 bits	8 bits	16 bits INTEL format: LSB, MSB	$N_{in} = 3$ to 235 bytes of user data

Input data format of the field bus master (send telegram of the field bus module)

Output data format of the field bus master

Binary Outputs	Receive CountBack	Transmit Count	Transmit Length	TransmitData
8 bits	8 bits	8 bits	16 bits INTEL format: LSB, MSB	$N_{out} = 3$ to 235 bytes of user data

Output data format of the field bus master (receive telegram of the field bus module)

The functions of the *Binary Inputs* byte and the *Binary Outputs* byte are described in chapter ??????

The *ReceiveCountBack* and *ReceiveCount* counters control user data transmission from the fieldbus module to the field bus master, whereby the field bus module is the active node (client) that must activate user data transmission by incrementing *ReceiveCount* by 1. Both counters must have the same status beforehand. The field bus master is the passive node (server).

ReceiveLength specifies the data length currently remaining to be transmitted from the fieldbus module to the field bus master (reading result of the bar code scanner). Attention: For compatibility reasons, *ReceiveLength* is in INTEL format!

The *TransmitCount* and *TransmitCountBack* counters control user data transmission from the fieldbus master (e.g. commands from the PLC) to the field bus module, whereby the fieldbus master is the active node (client) that must activate user data transmission by incrementing *TransmitCount* by 1. Both counters must have the same status beforehand. The field bus module is the passive node (server).

TransmitLength specifies the data length currently remaining to be transmitted from the field bus master to the field bus module. Attention: For compatibility reasons *TransmitLength* is in INTEL format!

In *TransmitData*, the field bus master enters the user data (left justified) to be transmitted to the field bus module.

The length of this field is N_{Out} = parameterized output data length minus 5.

In *ReceiveData*, the field bus module enters the user data (left justified) to be transmitted to the field bus master.

The length of this field is N_{In} = parameterized input data length minus 5

Direction of data transfer: Field bus module to Field bus master

Important : The following section describes the procedure for the data transfer direction "field bus module to field bus master". The procedure for the data transfer direction "field bus master to field bus module" is similar.

Transmitting incoming telegrams

Incoming telegrams at the field bus master are telegrams (e.g. reading results) that the bar code scanner formats as set in its configuration and then forwards to the field bus master via the field bus module.

A distinction is drawn here between two different types of telegram:

Telegrams that can be transmitted within a single incoming field bus telegram (user data fits in the *ReceiveData* element).

Telegrams that have to be divided into successive blocks due to their length (user data does not fit in the *ReceiveData* element).

4.3.1. Initialization

During reset or power-up, that is, before the first incoming telegram is transmitted, the field bus module sets the *ReceiveCount*, *TransmitCountBack*, and *ReceiveLength* to zero. Accordingly, the field bus master must set its *ReceiveCountBack*, *TransmitCount*, and *TransmitLength* to zero.

4.3.2. Resynchronisation

If the field bus module registers a field bus error (interruption in communication, incorrect counter statuses), it terminates data transmission in the affected data direction(s). The field bus module then requests the field bus master to resynchronize transmission. To do so, the field bus module sets the corresponding counters (*ReceiveCount* and/or *TransmitCountBack*) to zero. The field bus master then has to set its corresponding counters (*ReceiveCountBack* and/or *TransmitCount*) to zero too. The initialization status (see "Initialization") is now restored and the active node (client) can repeat or start data transmission.

Counter overflow

When incremented to 255, the *ReceiveCount* counter is assigned the value "1" (0 1 2 to 254 255 1 2 to). The value "0" only appears during initialization or if an error occurs (see "Resynchronisation").

4.3.3. Transmitting unblocked individual telegrams

As soon as an incoming telegram that is to be transmitted to the field bus master is available, the field bus module checks whether the previous data transmission has been completed (*ReceiveCountBack* is identical to *ReceiveCount*). The incoming telegram is then entered in the *ReceiveData* field (left justified). The bytes in the *ReceiveData* field that are not required are set to zero. The length of the incoming telegram is then entered in the *ReceiveLength* field. Finally, the *ReceiveCount* counter is incremented as described in “Counter overflow”. The field bus master is always activated when the *ReceiveCount* field is incremented, that is, when the value is different to the *ReceiveCountBack* counter value. The field bus master monitors whether *ReceiveLength* is less than or equal to NIn (parameterized incoming data

length minus 5) and then copies the *ReceiveLength* bytes in the *ReceiveData* field to a different buffer.

The field bus master then sets its *ReceiveCountBack* counter to the current *ReceiveCount* value. This signals to the field bus module that the field bus master is ready to transmit data again.

If the field bus master requires some time to evaluate and process the field bus module data, it can delay acknowledging the *ReceiveCountBack* counter max. 10 s if the byte handshake is enabled.

4.3.4. Transmitting blocked telegrams

If the length (NIn) of the *ReceiveData* field is not sufficient for transmitting an incoming telegram, the incoming telegram is divided into several blocks. These blocks are transmitted in succession. The same fields are used as for the individual telegram.

With the first block, the first part of the incoming telegram is entered in *ReceiveData*. The entire length of the incoming telegram is entered in *ReceiveLength*. The field bus master responds once *ReceiveCount* has been incremented. It recognizes from the telegram length (*ReceiveLength*) entered, which is greater than the *ReceiveData* field, that a blocked data transmission is running here. It transmits the entire *ReceiveData* field to a buffer and then sets *ReceiveCountBack* to the current value at *ReceiveCount*.

The field bus module now recognizes that the field bus master has processed the first part of the incoming telegram. The next part of the incoming telegram is then entered in the

ReceiveData field. *ReceiveLength* is assigned a value reduced by the length of the *ReceiveData* = NIn field. Finally, *ReceiveCount* is incremented again.

The reincremented *ReceiveCount* value reactivates the field bus master, which checks *ReceiveLength* again. If this value is now less than or equal to NIn, the remaining telegram is copied to the buffer and telegram transmission is complete. If the value is greater than NIn, the entire telegram is copied to the buffer and the field bus master waits for the next block.

In both cases field bus master must set the value in *ReceiveCountBack* to the current value in *ReceiveCount*.

An example on the following pages clarifies the described transfer sequence.

Example 1: The field bus master receives data from the field bus module

The incoming telegram (user data) from the field bus module is "123456789".

The input and output data length of the field bus is in each case 10 bytes, which means that NIn and NOut both comprise 5 bytes.

The counters are equalized, that is, the previous data transmissions are complete in both directions.

The data bytes are specified in the order in which they appear on the field bus. The length of the incoming telegram (9 bytes), for example, is specified in the *ReceiveLength* field as 09h 00h (INTEL format: LSB/MSB).

1. Initial situation: No new data
2. Incoming telegram, first block (block not yet processed by field bus master)

Binary Inputs	Receive Count	Transmit CountBack	ReceiveLength	ReceiveData
00h	01h	00h	09h 00h	1 2 3 4 5

Input data to the field bus master (send telegram from the field bus module)

Binary Outputs	Receive CountBack	Transmit Count	Transmit Length	TransmitData
00h	00h	00h	00h 00h	00h 00h 00h 00h 00h

Output data from the field bus master (receive telegram of the field bus module)

ReceiveCount is now 01h and thereby indicates data receipt.

3. Incoming telegram, first block (block processed by field bus master)

Binary Inputs	Receive Count	Transmit CountBack	ReceiveLength	ReceiveData
00h	01h	00h	09h 00h	1 2 3 4 5

Input data to the field bus master (send telegram from the field bus module)

Binary Outputs	Receive CountBack	Transmit Count	Transmit Length	TransmitData
00h	01h	00h	00h 00h	00h 00h 00h 00h 00h

Output data from the field bus master (receive telegram of the field bus module)

ReceiveCount was copied by the field bus master to **ReceiveCountBack** in order to confirm data receipt. This allows the field bus module to deliver subsequent blocks.

4. Incoming telegram, second and final block (block not yet processed by field bus master)

Binary Inputs	Receive Count	Transmit CountBack	ReceiveLength	ReceiveData
00h	02h	00h	04h 00h	6 7 8 9 00h

Input data to the field bus master (send telegram from the field bus module)

Binary Outputs	Receive CountBack	Transmit Count	Transmit Length	TransmitData
00h	01h	00h	00h 00h	00h 00h 00h 00h 00h

Output data from the field bus master (receive telegram of the field bus module)

The field bus module provides the second block and displays this by incrementing the **ReceiveCount** by 1.

4.3.5. 4.8.3 The Binary Inputs and Binary Outputs Bytes

The first input byte (*Binary Inputs*, Table 4-9) in the header contains status bits and a Heartbeat bit.

The first output byte (*Binary Outputs*, Table 4-10) in the header is reserved. It has to be set to zero.

Binary Inputs is assigned as follows:

Bit	Name	Meaning
D7	-	Reserved
D6	<i>BufferOverrun</i>	All the output buffers in the field bus module are full. The field bus module does not accept any transmitted data when this error is present. The field bus master can repeat data transfer later stage and then, if necessary, report an error to the user.
D5	-	Reserved
D4	-	Reserved
D3	<i>PlcError</i>	The field bus module has detected a handling error in the field bus master (PLC). The field bus module does not accept any transmitted data when this error is present and requests resynchronization with the field bus master. The PLC program must be corrected in line with the error.
D2	<i>Heartbeat</i>	The field bus module makes this heartbeat bit available to the field bus master. The heartbeat bit toggles between 0 and 1 at 1-second intervals.
D1	-	Reserved
D0	-	Reserved

Table 4-9: "Binary Inputs" bytes assignment in the CDF600 mode

Important The fieldbus module sets status bits (error bits) D7 to D3 when it acknowledges data transmission from the field bus master with *TransmitCountBack*, that is *TransmitCount* and *TransmitCountBack* are the same again. Status bits D7 to D3 are reset by the field bus module when the field bus master activates the next data transmission with *TransmitCount*. The field bus master, therefore, must always evaluate the status bits before transmitting the next batch of data to the field bus module.

The output buffer in the field bus module does not allow the status bits (D7, D5, and D4) to be assigned to a specific data transmission procedure initiated by the field bus master!

Binary Outputs is assigned as follows:

Bit	Name	Meaning
D7	-	Reserved
D6	-	Reserved
D5	-	Reserved
D4	-	Reserved
D3	-	Reserved
D2	-	Reserved
D1	-	Reserved
D0	-	Reserved

Table 4-10: "Binary Outputs" bytes assignment in the CDF600 mode

4.8.4 The Control Word In and Control Word Out modules

Important The modules *Control Word In* and *Control Word Out* for the digital function bits may only be configured in CDF600-0300 mode.

Control Word In (sensor → PLC).

This module consists of 2 bytes. The value represented by *Control Word In* is transmitted as a 16-bit value in the Motorola format.

Control Word In (Ctrl Bits in) is assigned as follows:

Important The functions with a grey background are planned for future firmware versions of the field

Control Word Out (PLC → sensor)

The module contains control bits from the field bus master to the field bus module. The defined function bits can be parameterized in the bar code scanner as input functions

Control Word Out (Ctrl Bits out) is assigned as follows:

Bit	Assignment	Name	Comment
D0	Fixed	Trigger	Object trigger for the CLV6xx bar code scanner
D1	Fixed	Sensor idle	
D2	Fixed	Teach-in 1	Control of Teach-In 1
D3	Fixed	Teach-in 2	Control of Teach-In 2
D4	Fixed	External output 1	Output on OUT 1 of the field bus module
D5	Fixed	External output 2	Output on OUT 2 of the field bus module
D6	Fixed	Result 1	Output on Result 1 of the bar code scanner
D7	Fixed	Result 2	Output on Result 2 of the bar code scanner
D8	Fixed	PLC_Out_08	
D9	Fixed	PLC_Out_08	
D10	Fixed	PLC_Out_08	
D11	Fixed	PLC_Out_08	
D12	Definable	Distance_Config_0	
D13	Definable	Distance_Config_1	
D14	Definable	Distance_Config_2	
D15	Definable	Distance_Config_3	

Bit	Assignment	Name	Comment
D0	Fixed	Device Ready	State of the CLV6xx bar code scanner
D1	Fixed	System Ready	Not implemented for the CLV6xx bar code scanner
D2	Fixed	Good Read	State of the reading result
D3	Fixed	No Read	State of the reading result
D4	Fixed	External output 1	OUT 1 output level of the field bus module
D5	Fixed	External output 2	OUT 2 output level of the field bus module
D6	Fixed	Result 1	Result 1 output level of the bar code scanner
D7	Fixed	Result 2	Result 2 output level of the bar code scanner
D8	Fixed	External input 1	IN 1 input level of the field bus module
D9	Fixed	External input 2	IN 2 input level of the field bus module
D10	Fixed	Sensor 1	Sensor 1 input level of the bar code scanner
D11	Fixed	Sensor 2	Sensor 2 input level of the bar code scanner
D12	Definable	Defined by the configuration of the bar code scanner	
D13	Definable		
D14	Definable		
D15	Definable		

5. EC Declaration of Conformity

Below the page 1 of the EC Declaration of Conformity (size reduced) is mapped.
The full EC Declaration of Conformity is available on request.



EC Declaration of conformity

en

Ident-No. : 9122730

The undersigned, representing the following manufacturer

SICK AG
Nimburger Straße 11
79276 Reute
Germany

herewith declares that the product

CDF600

is in conformity with the provisions of the following EC directive(s) (including all applicable amendments), and that the standards and/or technical specifications referenced overleaf have been applied.

Reute, *11-04-2008*


.....
ppa. Pierenkemper
(Manager Development Division Auto Ident)


.....
ppa. Walter
(Manager Production Division Auto Ident)

6. EtherCAT Conformity declaration

	<h2>Certificate</h2> <h3>EtherCAT Conformance Test</h3>	
	SICK AG Erwin-Sick-Str. 1, 79183 Waldkirch, Germany	
	EtherCAT Technology Group hereby confirms the above named company that the following family devices are successfully EtherCAT Conformance Tested .	
	Device under Test 1	
	Product Name:	CLV630-0120 with connection device CDF600-0300
	Product Code:	0x5001
	Revision Number:	0x50010
	Device under Test 2	
	Product Name:	CLV620-0120 with connection device CDF600-0300
	Product Code:	0x5001
Revision Number:	0x50010	
Device family is listed on one following page.		
Assigned Vendor ID:	0x1000056	
Test Report Number:	0x1000056_002	
EtherCAT Test Center:	Beckhoff Automation GmbH, Nuremberg, Germany	
The following tests were performed:		
<ul style="list-style-type: none">- EtherCAT Protocol Test (CTT Ver.1.20.0.0)- Indicator Test- Labeling Test- Interoperability Test		
Nuremberg, July 14, 2011	 _____ Martin Rostan, Executive Director EtherCAT Technology Group	

7. Further Reading

Further reading (literature) for CDF600-0300 is on disposal and is mandatory to be used together with this „integrator commissioning description“.

Information Type	Article number
Multipack containing	8013917
CDF600-0300 Operating Instructions Englisch	8013919
CDF600-0300 Operating Instructions German	8013918

Australia
Phone +61 3 9497 4100
1800 33 48 02 - tollfree
E-Mail sales@sick.com.au

Belgium/Luxembourg
Phone +32 (0)2 466 55 66
E-Mail info@sick.be

Brasil
Phone +55 11 3215-4900
E-Mail sac@sick.com.br

Ceská Republika
Phone +420 2 57 91 18 50
E-Mail sick@sick.cz

China
Phone +852-2763 6966
E-Mail ghk@sick.com.hk

Danmark
Phone +45 45 82 64 00
E-Mail sick@sick.dk

Deutschland
Phone +49 211 5301-270
E-Mail info@sick.de

España
Phone +34 93 480 31 00
E-Mail info@sick.es

France
Phone +33 1 64 62 35 00
E-Mail info@sick.fr

Great Britain
Phone +44 (0)1727 831121
E-Mail info@sick.co.uk

India
Phone +91-22-4033 8333
E-Mail info@sick-india.com

Israel
Phone +972-4-999-0590
E-Mail info@sick-sensors.com

Italia
Phone +39 02 27 43 41
E-Mail info@sick.it

Japan
Phone +81 (0)3 3358 1341
E-Mail support@sick.jp

Nederlands
Phone +31 (0)30 229 25 44
E-Mail info@sick.nl

Norge
Phone +47 67 81 50 00
E-Mail austefjord@sick.no

Österreich
Phone +43 (0)22 36 62 28 8-0
E-Mail office@sick.at

Polska
Phone +48 22 837 40 50
E-Mail info@sick.pl

Republic of Korea
Phone +82-2 786 6321/4
E-Mail kang@sickkorea.net

Republika Slovenija
Phone +386 (0)1-47 69 990
E-Mail office@sick.si

România
Phone +40 356 171 120
E-Mail office@sick.ro

Russia
Phone +7 495 775 05 34
E-Mail info@sick-automation.ru

Schweiz
Phone +41 41 619 29 39
E-Mail contact@sick.ch

Singapore
Phone +65 6744 3732
E-Mail admin@sicksgp.com.sg

Suomi
Phone +358-9-25 15 800
E-Mail sick@sick.fi

Sverige
Phone +46 10 110 10 00
E-Mail info@sick.se

Taiwan
Phone +886 2 2375-6288
E-Mail sickgrc@ms6.hinet.net

Türkiye
Phone +90 216 587 74 00
E-Mail info@sick.com.tr

USA/Canada/México
Phone +1(952) 941-6780
1 800-325-7425 - tollfree
E-Mail info@sickusa.com

More representatives and agencies
in all major industrial nations at
www.sick.com