

<b>SICK</b>		<b>CAN Open Example using PC-Adapter</b>	<b>CLV6xx RFH6xx ICR6xx</b>
			<b>CANopen</b>

## Content

1	Example CLV65x connected to USB CAN-Adapter.....	1
1.1	Used devices .....	1
1.2	Settings for CLV at Sopas.....	2
1.3	Settings at proCANtool .....	3
1.4	Receiving result data via PDO with up to 49 bytes.....	5
1.5	Triggering CLV using Fieldbus trigger bit via CAN .....	7

## 1 Example CLV65x connected to USB CAN-Adapter

This example demonstrates how data are received via CAN-Open and Triggering is done, using an easy CAN-Adapter connected to a PC, simulating CAN-Open telegrams.

### 1.1 Used devices

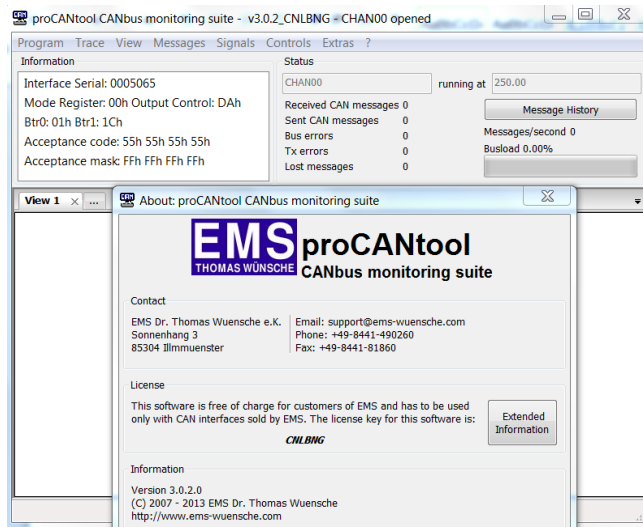
- CLV65x with V5.51 connected to CDM420 with CAN-termination active

Device Information			
Manufacturer	<input type="text" value="SICK AG"/>	Software Version	<input type="text" value="V5.51"/>
Device Type	<input type="text" value="CLV650-6120"/>	Hardware Version	<input type="text" value="HWID 1F6AD1000 REV 1"/>
SOPAS Interface Version	<input type="text" value="V5.51"/>	Serial Number	<input type="text" value="12090051"/>
Order Number	<input type="text" value="1042125"/>		

- CAN interface: USB/CAN Interface CPC-USB/ARM7  
<http://www.ems-wuensche.de/product/datasheet/pdf/can-usb-adapter-converter-interface-cpcusb.pdf>

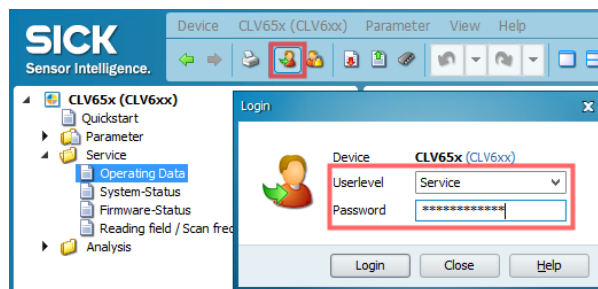


- CAN-Software: proCANtool (delivered with the adapter)  
for monitoring data and send trigger via CAN:



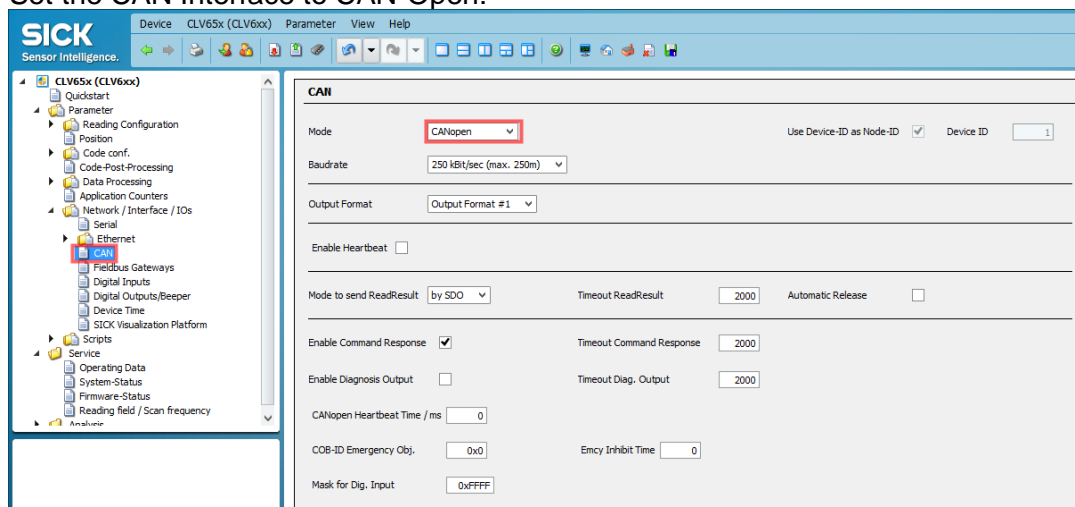
## 1.2 Settings for CLV at Sopas

The CLV65x is connected to Sopas and set at first to **factory defaults**. Then a login as **“Service”** is done with the password **“servicelevel”**:



With this login all needed parameters for CAN-Open are shown in Sopas.

Set the CAN Interface to CAN-Open:



At “Network/Interfaces/IO” the CAN-ID can be chosen. Here 1 is used:

<b>SICK</b>		<b>CAN Open Example using PC-Adapter</b>	<b>CLV6xx RFH6xx ICR6xx</b>
			<b>CANopen</b>

**Network Options**

Device ID  Device Name

Enabling Heartbeat Time (here set to 3 sec) to see that the CLV is alive at CAN-Open:

**CAN**

Mode:  Use Device-ID as Node-ID:  Device ID:

Baudrate:

Output Format:

Enable Heartbeat:

Mode to send ReadResult:  Timeout ReadResult:  Automatic Release:

Enable Command Response:  Timeout Command Response:

Enable Diagnosis Output:  Timeout Diag. Output:

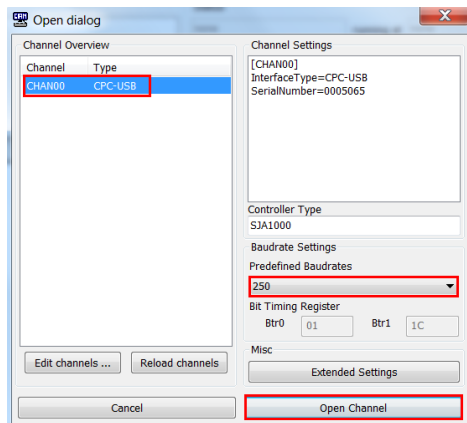
**CANopen Heartbeat Time / ms:**

COB-ID Emergency Obj.:  Emcy Inhibit Time:

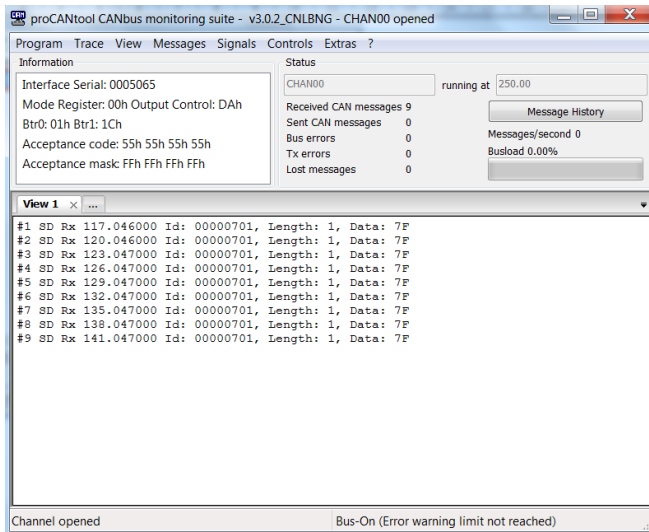
Mask for Dig. Input:

### 1.3 Settings at proCANtool

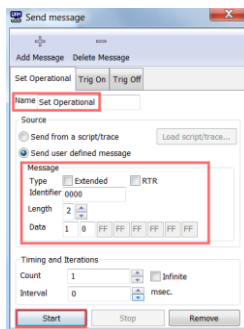
Start then the proCANtool suite and open a channel at “Programm/Open Channel” and choose the interface and the used CAN-baudrate (here 250 kBit/sec):



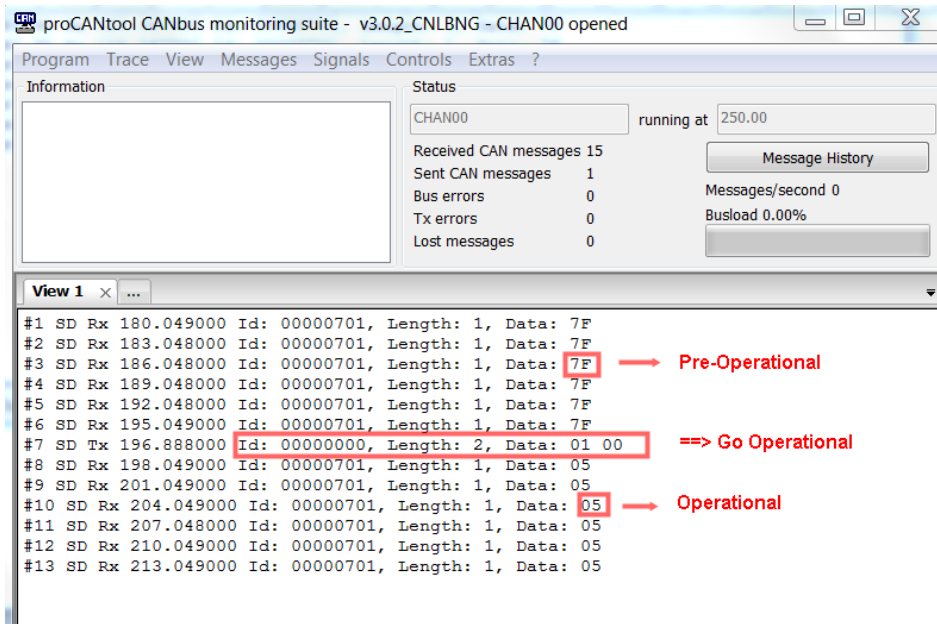
- ⇒ You can monitor then the heart beat telegram (Object ID 700 + node ID) coming every 3 sec send from the CLV, which indicates that the CAN-Bus is running. The data “7F” indicates here that the CLV is the CAN-Open state “Preoperational”:



The CLV can be set in “Operational Mode” by sending at Object ID 0 two bytes with the content 01 00 hex. This is at a CAN-Open system normally automatically done. Here it can be done using “Message/Send Message”. It must be done again after each change in Sopas at the CAN-Open parameters, because it sets the CLV back in Preoperational.



After sending this command the CLV is “Operational” which is indicated by the data “05” in the heartbeat message.

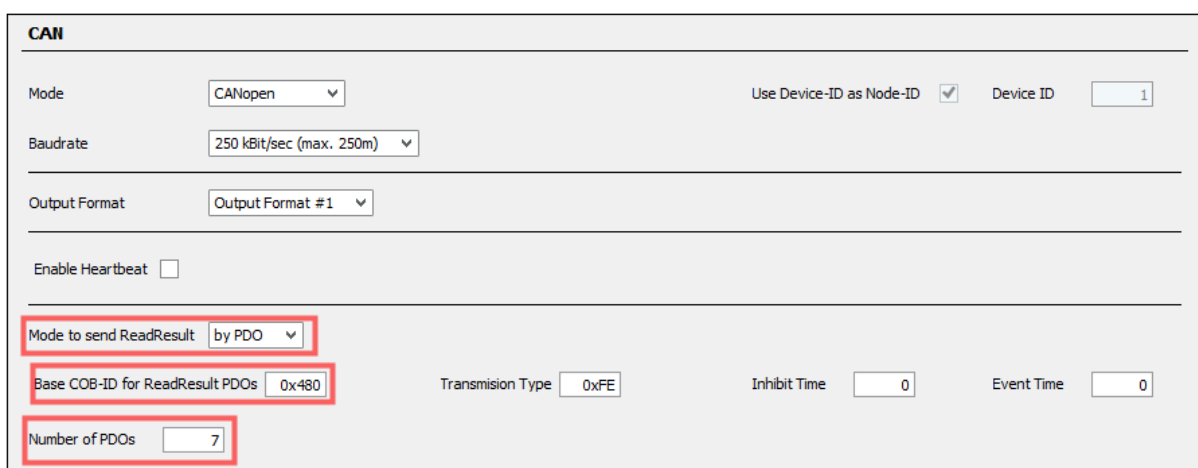


## 1.4 Receiving result data via PDO with up to 49 bytes

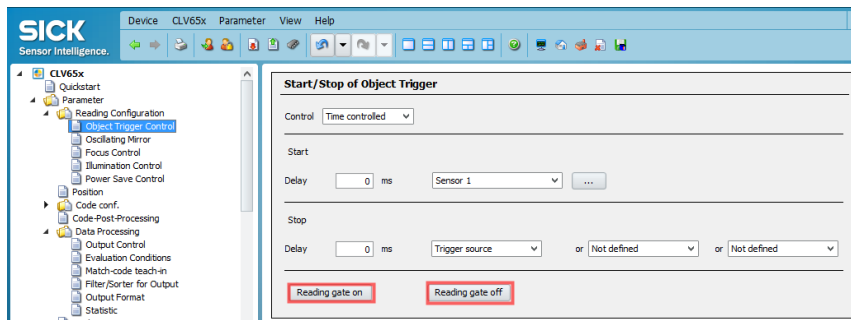
When the CLV is sending out only short data length of up to 49 bytes the data can be transferred using up to 7 PDO's which may contain each up to 7 bytes. If the data are longer the first 49 bytes are sent and the further data are lost.

→ So this can be used when the bytes the output format 1 creates is not longer than 49 bytes.

Set the Mode to PDO and choose the number of PDO's which should be sent out. The base COB-ID can be set for the first PDO, the following PDOs will use the following number.

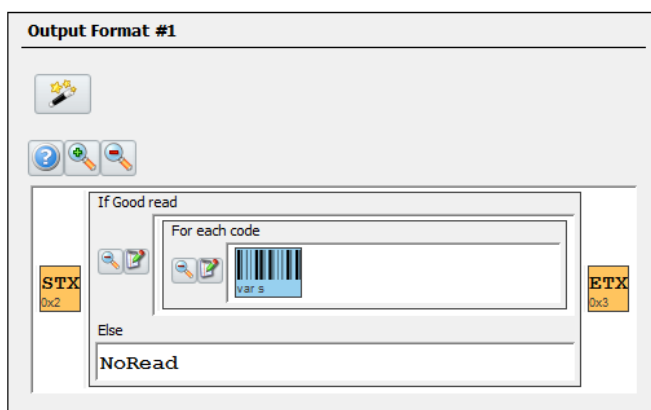


You can trigger the CLV by pressing the buttons "Reading Gate On / OFF" at Object trigger or trigger the CLV in any other way.



The output format is here at default.

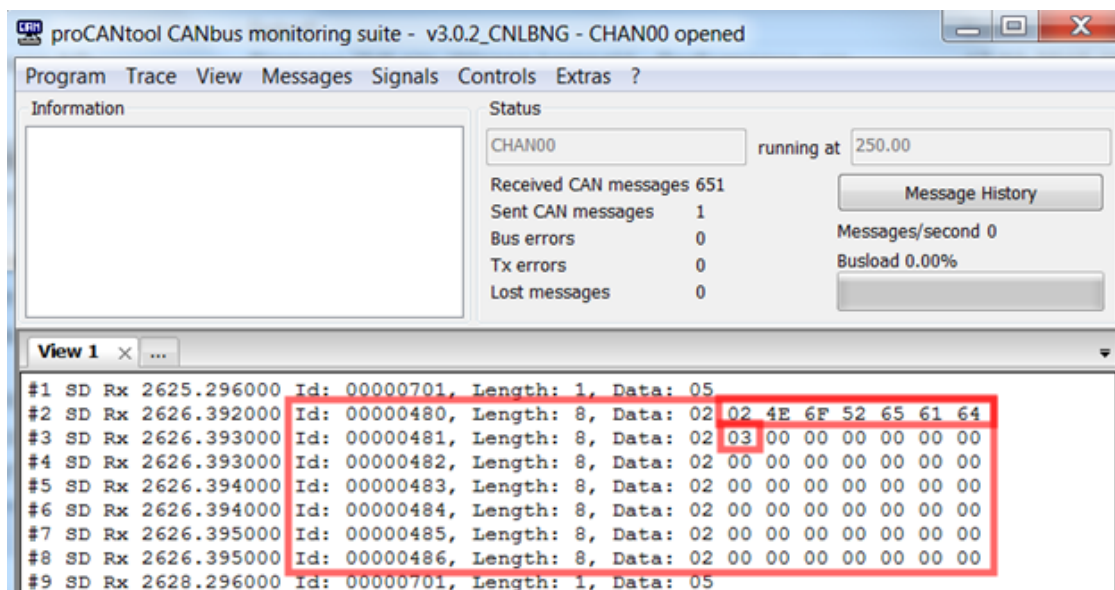
It can be free programmed at CAN-Open, as no STX ETX framing is needed.



Then 7 PDO's are send out at an trigger with COB-ID 480, 481, ... 486.

The data field do contain here 8 bytes "STX NoRead ETX" ( STX = 02 hex, ETX = 03) because here no code was read.

The 8 byte message needs two PDO's and the not uses data are filled up with 00.



## 1.5 Triggering CLV using Fieldbus trigger bit via CAN

When the CLV should be triggered via CAN you can use the **Fieldbus trigger bit** at the CANopen outputs of object 6200, subindex 1, Bit 0 (see chapt. 4.2)

Bit	Object	assignment	name	comment
Byte 0, Bit 0	0x6200/1	fixed	Trigger	
Byte 0, Bit 1	0x6200/1	fixed	Sensor-Idle	
Byte 0, Bit 2	0x6200/1	fixed	TeachIn1	
Byte 0, Bit 3	0x6200/1	fixed	TeachIn2	
Byte 0, Bit 4	0x6200/1	fixed	External Output_1	Physical Output 1 of CDF600
Byte 0, Bit 5	0x6200/1	fixed	External Output_2	Physical Output 2 of CDF600
Byte 0, Bit 6	0x6200/1	fixed	Digital Output_1 (Result_1)	
Byte 0, Bit 7	0x6200/1	fixed	Digital Output_2 (Result_2)	
Byte 1, Bit 0	0x6200/2		PLC_Out_08	
Byte 1, Bit 1		soft	PLC_Out_09	
Byte 1, Bit 2	0x6200/2	soft	PLC_Out_10	
Byte 1, Bit 3	0x6200/2	soft	PLC_Out_11	
Byte 1, Bit 4	0x6200/2	fixed	Distance_Config_0	LSB
Byte 1, Bit 5	0x6200/2	fixed	Distance_Config_1	
Byte 1, Bit 6	0x6200/2	fixed	Distance_Config_2	
Byte 1, Bit 7	0x6200/2	fixed	Distance_Config_3	MSB

To enable it please set the trigger source to “Fieldbus / CAN-Open” in Sopas:

**Start/Stop of Object Trigger**

Control: Time controlled ▼

---

Start

Delay:  ms    Fieldbus Input / CANopen ▼

---

Stop

Delay:  ms    Trigger source ▼    or    Not defined ▼    or    Not defined ▼

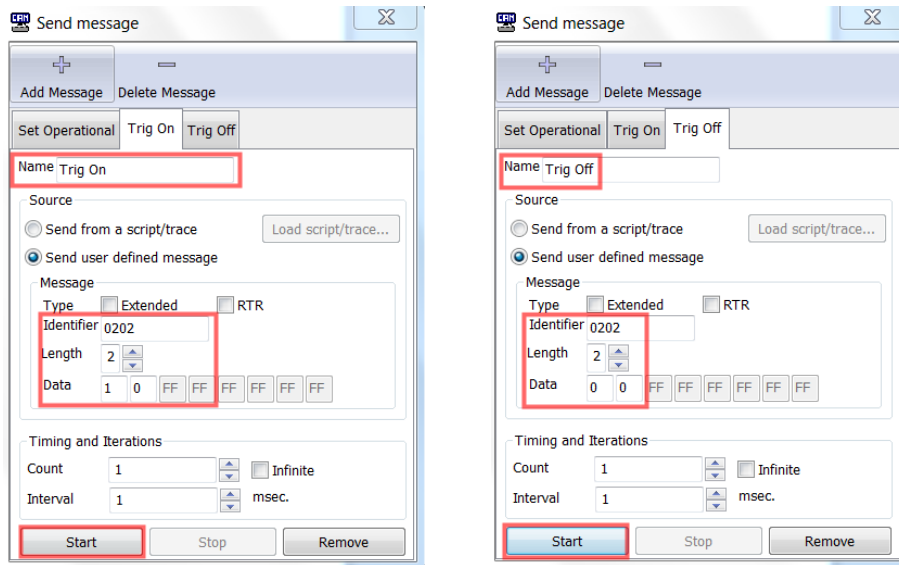
To allow the CLV to accept this the Receive PDO, set PDO 1:

Set the Predifined Connection to “no”, and choose the COB-ID, here 202 hex, set the number of mapped objects to 1 and set the first Object to 62000108 (meaning: Object 6200, subindex 01, 08 bits long). If needed save the parameter set and power cycle the CLV.

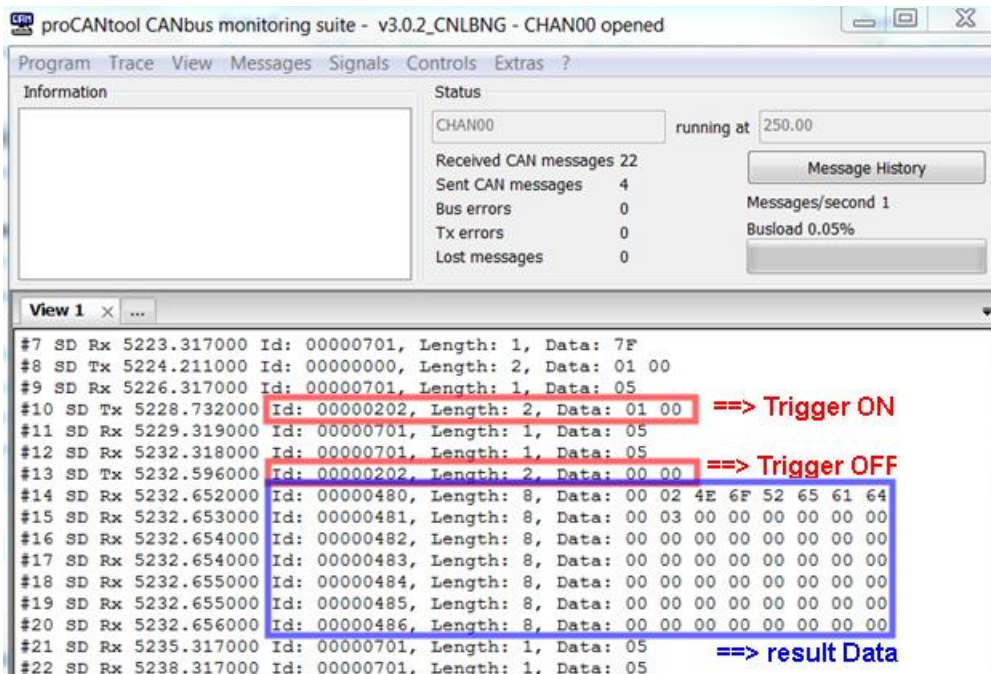
CANopen Receive PDOs 1 .. 4				
	RPDO1	RPDO2	RPDO3	RPDO4
Predef. conn.:	No	Yes	Yes	Yes
COB-ID	0x00000202	0x80000000	0x80000000	0x80000000
Transm. Type	0xFE	0xFE	0xFE	0xFE
Num of map. o	1	0	0	0
Map Obj. 1	0x62000108	0x00000000	0x00000000	0x00000000
Map Obj. 2	0x00000000	0x00000000	0x00000000	0x00000000
Map Obj. 3	0x00000000	0x00000000	0x00000000	0x00000000
Map Obj. 4	0x00000000	0x00000000	0x00000000	0x00000000
Map Obj. 5	0x00000000	0x00000000	0x00000000	0x00000000
Map Obj. 6	0x00000000	0x00000000	0x00000000	0x00000000
Map Obj. 7	0x00000000	0x00000000	0x00000000	0x00000000
Map Obj. 8	0x00000000	0x00000000	0x00000000	0x00000000

The message to start and stop the trigger can be send using “Message/Send Message”. You may add a Message and name them “Trig On” and “Trig Off”.

Please send also after any parameterization change at the CLV also the set operational first.



Sending COB-ID 202 with 2 bytes 01 00 → trigger start  
 Sending COB-ID 202 with 2 bytes 00 00 → trigger stop (needs to be all times send\*)  
 ⇨ Data are send out using COB-ID 480 – 486.



Please note that the **trigger stop (00 00)** needs to be send all times, also when trigger stop at GoodRead or timer is used. The next trigger start (01 00) can be only recognized when a real stop (00 00) was send before.