Manual CMM_III

IRS Systementwicklung GmbH
Pfaffenthanner Weg 5
D- 93179 Brennberg
tel: +49 9484 9500 0
fax: +49 (0) 9484 9500 25
home: www.irs-systeme.de
mailto: michael.rost@irs-systeme.de

History

Version	Date	Autor	Remarks
1.0	08.10.2014	Michael Rost	1. draft created
1.1	28.11.2014	Michael Rost	Added features for HW version 1.2: - Higher Voltage @ CMM Off (chapter 2.3.1) - Reverse current characteristics (chapter 2.3.1) - CAN termination selection (chapter 2.4.1) - Reverse current detection (chapter 3.1.2)
1.2	09.12.2014	Michael Rost	Added features for SW version 1.2: - Added On/Off configuration (chapter 3.2.) - Added CAN configuration (chapter 3.7)

Contents

Figures

1 Introduction

The current measurement module CMM_III can be used to measure, evaluate and check current consumption of a dynamic DC load, typically an automotive electronic control unit. It can measure currents from 1µA to $100A_{DC}$ (190A range) with automatic range selection within a few microseconds. The current range from 1µA to 190A is split into 7 Ranges, i.e. seven decades.

1.1 Purpose of this document

Purpose of this document is to describe how to integrate the module in a test system and how to access it from the software point of view. Limits of application are shown in the technical data section.

This document is addressed to system integrators and the users, who are applying the module.

1.2 Definitions and abbreviations

Abbreviation	*Definition*
CMM	**C**urrent **M**easurement **M**odule
CAN	**C**ontroller **A**rea **N**etwork (network for output data)
LVDS	**L**ow **V**oltage **D**ifferential **S**ignalling (used as signal level for SPI)
SPI	**S**erial **P**eripheral **I**nterface (clock synchronous data output)
DUT	**D**evice **U**nder **T**est (Device, whose current should be monitored)

1.3 References

Document	*Date*	*Description*
Datasheet SN65HVD230	Feb 2011	
Datasheet DS90LV049	April 2013	

1.4 Document Overview

This documented contains 3 sections.

- The first section includes an introduction to this manual
- The second section includes description of the hardware
- The third section includes accessing the module from the software point of view.

2 Hardware

The following figure shows a block diagram of CMM_III and its external connections

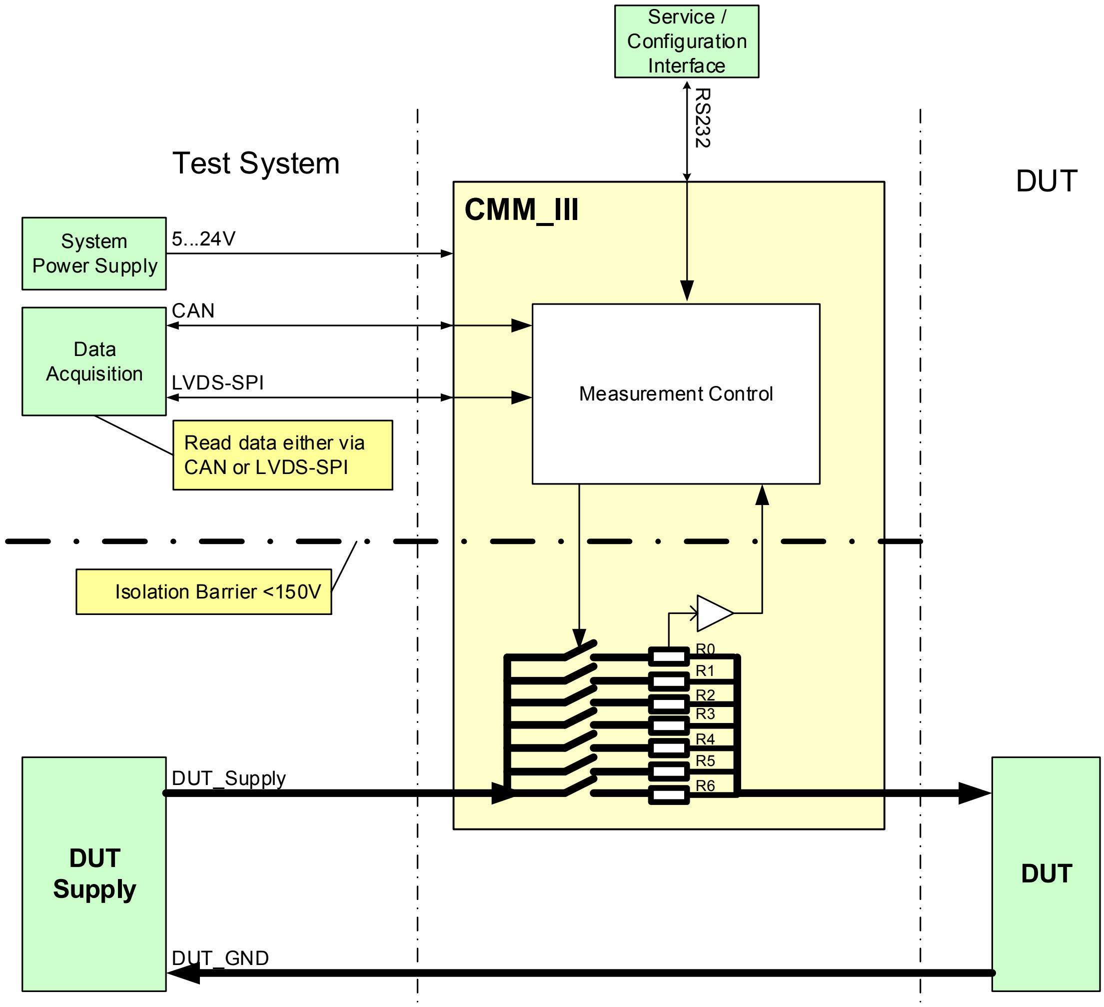

Figure 1: CMM_III HW Overview

General setup:

- CMM_III needs 5...24V power supply on PWR/GND_Ext
- The CMM_III Current Path is integrated in series to the DUT, whose current should be monitored.

The measurement values can be read out by the following:

- Digital output via a 32 bit SPI word at LVDS voltage level.
- CAN (High-Speed-CAN with 250...1000 kBit/s)
- RS232 as ASCII-String via the service interface

Additonal features are:

- The RS232 service interface may be also used to configure the module.
- Permanent parameter storage of the configuration in Flash memory
- Calculation of mean, min and max values.
- Current path is electrically isolated from electronics supply and digital IO (up to 150V)
- Detection of reverse current

Please note, that the module is not hot-pluggable, when LVDS-SPI interface is connected!

2.1 Connector Pinning

2.1.1 System connector

Interface	Signal	Pin	Recommended wire	Remarks
Module Power	**PWR_Ext**	**C13**	0,14 … 0,5mm^2	
	GND_Ext	**A13, A14, A16, A17, A18, A19, A20**	0,14 … 0,5mm^2	Connect at least 1 pin
On/Off	**On/Off**	**A15**	0,14 … 0,5mm^2	
CAN	**CAN_L**	**B14**	0,14 … 0,5mm^2 Twisted pair	Wire with ≈ 120Ω impedance
	CAN_H	**C14**		
LVDS-SPI	**CLK_In+**	**B15**	0,14 … 0,5mm^2 Twisted pair	Wire with ≈ 100Ω impedance
	CLK_In-	**C15**		
	En_In+	**B16**	0,14 … 0,5mm^2 Twisted pair	Wire with ≈ 100Ω impedance
	En_In-	**C16**		
	Rx+	**B17**	0,14 … 0,5mm^2 Twisted pair	Wire with ≈ 100Ω impedance
	Rx-	**C17**		
	Tx+	**B18**	0,14 … 0,5mm^2 Twisted pair	Wire with ≈ 100Ω impedance
	Tx-	**C18**		
	En_In+	**B19**	0,14 … 0,5mm^2 Twisted pair	Wire with ≈ 100Ω impedance
	En_In-	**C19**		
	CLK_Out+	**B20**	0,14 … 0,5mm^2 Twisted pair	Wire with ≈ 100Ω impedance
	CLK_Out-	**C20**		
Current Path	**Curr_In**	**4x High current pin A1-A12 B1-B12 C1-C12**	1…4 x 10mm^2	Depending on applied currents use 1 to 4 wires with appropriate diameter for both input and output, respectively. For currents above 60A use all 4 pins both for input and output.
	Curr_Out	**4x High current pin A21-A32 B21-B32 C21-C32**	1…4 x 10mm^2	

2.1.2 RS232 service connector

Interface	Signal	Pin	Recommended wire	Remarks
RS232	**CMM_Tx**	**2**	Standard 9pol. 1:1 cable	Send data to PC
	CMM_Rx	**3**		Receive data from PC
	GND	**5**		GND connected to GND_Ext

Note, that the RS232 is a service connection. Normal operation uses the system connector with CAN or LVDS-SPI interface.

2.1.3 Mating system connector

The following components may be used for the mating connector on the test system. 1 connector with 8 high current pins is needed. Different versions for crimping or soldering from two manufacturers can be found in the following table.

Item	Quantity	Order Number	Manufacturer	Remarks
DIN41612 Type M 24+8	1	09 03 224 6804	Harting	
		354 116	ERNI	
High current pin	**8**	09 03 000 6115	Harting	Crimp version
		09 03 000 6103	Harting	Solder version
		594 182	ERNI	Crimp version
		594 176	ERNI	Solder version

2.2 Dimensions

CMM_III is designed on a 160mmx 100mm Euro card, including front cover with a width of 25,4mm (5HP) and a standard DIN41612 connector. With this dimensions it fits to standard 19" / 3HU carriers and racks.

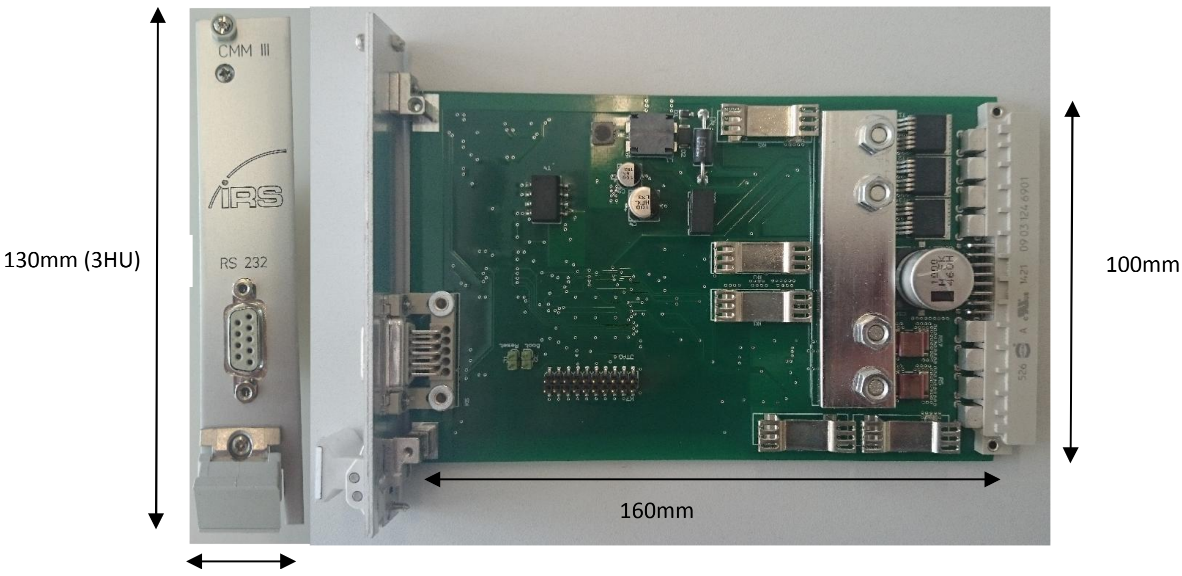

Figure 2: CMM_III Dimensions

2.3 Technical data

2.3.1 Current Path

Depending on the applied currents, the measurement current path has to be wired with a low resistance to maintain low voltage drops. I.e. use short wires and as much high-current-pins as possible – both for current input and current output.

Every High-Current pin may carry up to 40A DC maximum. But more used high current pins with wires of high diameter both yield lower voltage drops and temperature at CMM_III, because of heat conduction. Lower temperature also means lower drift.

Item	Min	Typ	Max	Units	Remarks
DC Current	0		100	A	operation under the following conditions: - 25°C ambient temperature without airflow - 50° Ambient Temperature with airflow ≈2m/s
Worst case DC Current			160	A	- 25° Ambient Temperature with airflow >6m/s on power devices and connector
DC Current **per High-current pin**			40	A	- 50° Ambient Temperature
Measurement Range			190	A	Max. 3 seconds
Single Pulse Current			300	A	Max. 100ms
CMM_III Voltage @ OFF HW-Rev 1.2			36	V	Module disconnects current path when PWR_Ext is below 4,8V, or ON/Off is at low level For HW-Rev 1.1 max. Voltage is 30V!
Leakage current @ OFF	0		20	µA	@ 36V
Voltage difference between GND_Ext and Current Path	-150		150	V	Limit is restricted by distance of wiring on the PCB. Electronic components are specified at least 250V
Reverse Current detection threshold	50		1000	mA	Depending on Temperature and components
Reverse Current continuous	0		30	A	
Reverse Current single pulse			200	A	Max. 1 second.

NOTE: Maximum voltage, applied at the DUT:

When the CMM_III is switched off (i.e. no External power applied or On/Off-Control input is at low level) the DUT supply voltage is applied fully at CMM_III, which can handle up to 36V in OFF-state! Thus, please make sure that CMM_III is always switched ON, when higher voltages than 36V are used for the DUT!

2.3.2 Module Supply

Apply a power source on PWR_Ext/GND_Ext. It is recommended not to use the DUT power supply to make sure that measurement device power is separate from DUT power. Typical applications use 5V, 12V or 24V power supplies. Several other components may be supplied by the same power source.

Nonetheless, because of CMM_III wide input voltage range, it is possible to use DUT-Power for powering CMM_III.

Note, that current path is disconnected, when no power is applied.

Item	Min	Typ	Max	Units	Remarks
Supply Voltage	4,9		26	V	PWR_Ext to GND_Ext
Supply DC Current @ 5V	190	210	250	mA	
Supply DC Current @ 12V	80	92	120	mA	
Supply DC Current @ 24V	40	51	70	mA	
Supply Inrush Current		1,2	1,5	A	Inrush current for approx.5ms @ dU/dt = 1V/ms

2.3.3 DC Accuracy

Accuracy of the module is verified at DC currents. Every module passes a calibration procedure at IRS. The module may be re-calibrated.

Item	Min	Typ	Max	Units	Remarks
Accuracy uncalibrated		0,5	2	% of Range	Max in % of respective Range
Accuracy calibrated		0,2	1	% of Range	Max in % of respective Range
Resolution Range_0			100	nA	Limited by output data step size.
Resolution Range_1			404	nA	
Resolution Range_2			4,04	µA	
Resolution Range_3			40,3	µA	
Resolution Range_4			403	µA	
Resolution Range_5			4,03	mA	
Resolution Range_6			46,8	mA	

2.3.4 DC Voltage Drop

Voltage drop is the voltage between current input and output, measured on the module connector. The drop on the female power connector pins and the wiring has to be added.

Item	Min	Typ	Max	Units	Remarks
Drop @ 100µA		70	80	mV	Range_0
Drop @ 1mA		70	80	mV	Range_1
Drop @ 10mA		70	80	mV	Range_2
Drop @ 100mA		70	80	mV	Range_3
Drop @ 1A		90	100	mV	Range_4
Drop @ 10A		100	120	mV	Range_5
Drop @ 40A		45	60	mV	Range_6
Drop @ 100A		120	150	mV	
Drop @ 160A		200	230	mV	

2.3.5 On/Off Interface

The digital input On/Off is used to switch the current path of the CMM_III on or off.

The current path is **connected**, **when** a **high** level is applied

The current path is open, when a low level is applied.

Item	Min	Typ	Max	Units	Remarks
On/Off Control – Low = OFF	-0,7		1,0	V	
On/Off Control – High = ON	3.0		25	V	
On/Off Control Input Resistance		15		kΩ	0…3,3V
	5		15	kΩ	3,3…25V

NOTE: Maximum voltage, applied at the DUT:

When the CMM_III is switched off (i.e. no External power applied or On/Off-Control input is at low level) the DUT supply voltage is applied fully at CMM_III, which can handle up to 36V in OFF-state!

Thus, please make sure that CMM_III is always switched ON, when higher voltages than 36V are used for the DUT!

2.3.6 CAN Interface

The integrated CAN interface is a common High-Speed CAN interface.

Item	Min	Typ	Max	Units	Remarks
Voltage at CAN_H or CAN_L	-4		16	V	Against GND_Ext
Input range, transient pulse CAN_H and CAN_L	-25		25	V	
Dominant output level	1,2	2	3	V	
Recessive output level	-120	0	12	mV	
Data rate			1000	kbit/s	

For further details, see datasheet SN65HVD230 and chapter 2.4.1

2.3.7 LVDS-SPI Interface

Item	Min	Typ	Max	Units	Remarks
Voltage Range	-0,2		2,7	V	LVDS-SPI inputs
Termination		100		Ω	
Differential input voltage range	100	350	600	mV	
Common mode input voltage range	0.3	1.2	2.2	V	
Differential output voltage range	247	350	600	mV	LVDS output
Common mode output voltage range	1.125	1.25	1.375	V	
LVDS Interface Baudrate (typical no feedback)	500	1250	1500	kBit/s	Without Clock feedback
LVDS Interface Baudrate (maximum with feedback)			15000	MBit/s	With Clock feedback to system

For further details, see datasheet DS90LV049 and chapter 2.4.2

2.4 System Integration

The following section handles the hardware issues about the interfaces for accessing measurement data. For integration of the current path and the power supply, see the remarks in the technical data in section 0.

2.4.1 CAN

The High-speed CAN interface may be used to read measurement data.

Any High-Speed CAN interface from any vendor may be applied as counterpart for data acquisition. Make sure that termination of the entire bus is implemented properly with two times 120Ω at the respective far end of the bus. No other termination resistors are included.

Wires must be twisted pair with an impedance of 120Ω.

Every CMM on the bus should run at the same baud rate and use different CAN IDs. CAN ID and baud rate may be configured via the RS232 service interface and are stored permanently in Flash memory.

For integration as a single module in a test system setup, please add a CAN counterpart with a termination resistor of 120Ω. CMM_III includes 120Ω -termination which can be enabled or disabled by a jumper switch (see figure 3).

If several CMM_III are connected in parallel to one CAN interface. The termination resistor of every single CMM_III has to be disabled. A single termination must be included at the far end of the CAN bus.

See the following figure for disabling of the termination resistor with the **jumper switch J3, which is marked in red**:

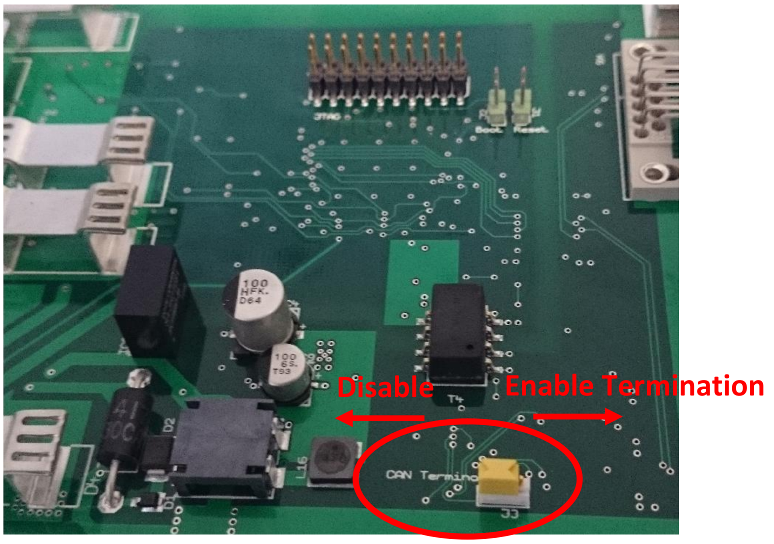

Figure 3: CAN Termination Resistor

See also chapter 0, when using backplane and CAN interface.

2.4.2 LVDS-SPI

2.4.2.1 Implementation Options

LVDS-SPI is integrated to read data from CMM at a higher data rate, than available via CAN. But common SPI may not be used in a harsh test system environment. Thus differential signaling is applied for low noise radiation and high noise immunity.

For a single module, the following setup may be used. Note, that every signal in the diagram is actually a differential pair. With this setup data rates of approximately 1,5Mbit/s can be achieved. With this speed a sample rate of up to 20kS/s can be achieved for a single module.

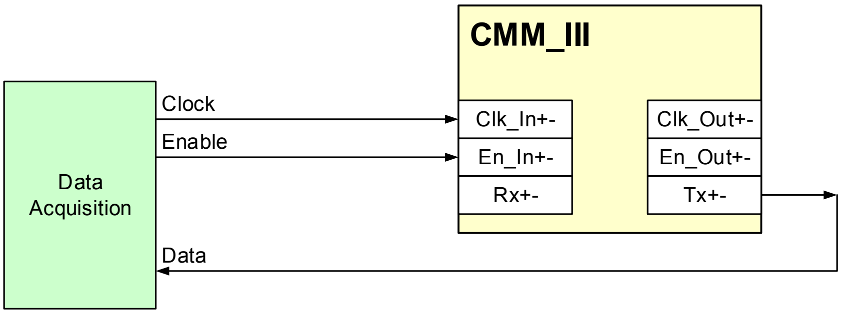

Figure 4: LVDS-SPI for a single module at data rates up to 1,5Mbit/s

When higher data rates than 1,5Mbit/s are applied, the enable and clock line have to be fed back to the data acquisition system. With this setup, Data, clock-feedback and enable-feedback may be evaluated synchronously. Otherwise the delay through the CMM may cause corrupt data.

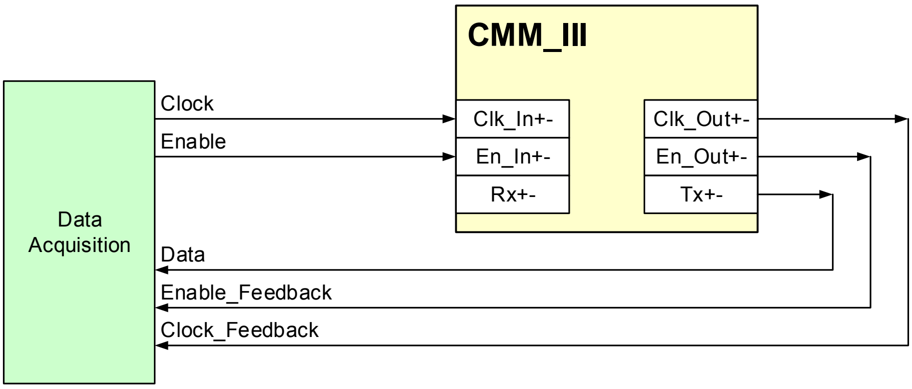

Figure 5: LVDS-SPI for a single module at data rates up to 15Mbit/s

The LVDS-SPI interface is designed in a way to concatenate several CMM’s in a daisy chain type, as Illustrated in the following figure.

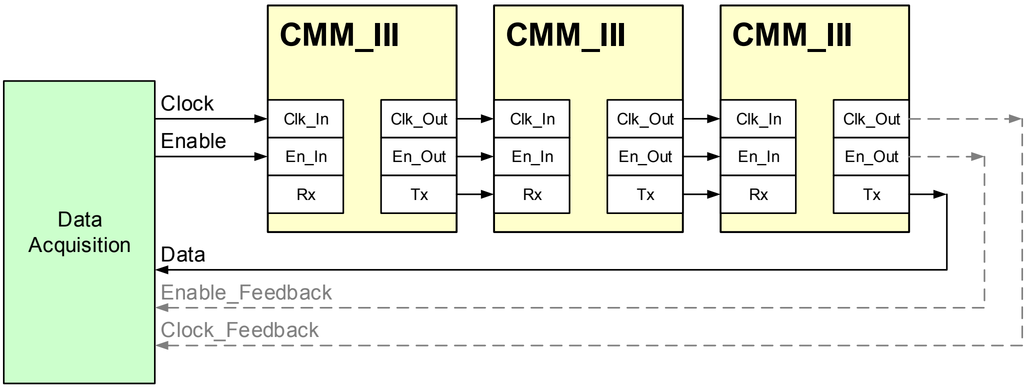

Figure 6: LVDS-SPI "Daisy chain" for several modules

The maximum number of CMMs depends on data rate, sampling rate and performance of the data acquisition system. Typical setups are as follows:

Typical setup 1:

- 1…10 CMMs
- sampling rate of 2,5kS/s.
- clock rate of 1,25Mbit/s
- feedback of clock and enable not necessary

Typical setup 2:

- 1…10 CMMs
- Sampling rate of 20kS/s
- clock rate of 10Mbit/s
- Feedback of clock and enable necessary.

Please note, that in any case, the wires for clock, enable, Rx and Tx have to be at least properly twisted with a maximum length of approx. 1m. When higher distances have to be covered, please use the backplane, described in chapter 0., including shielded CAT5…CAT7 cables. In harsh environments, always use the backplane.

Please note, that the module is not hot-pluggable, when LVDS-SPI interface is connected!

2.4.2.2 Data Acquisition with IRS MesSy

The standard device for data acquisition is the IRS MesSy. MesSy is a generic re-usable multifunction measurement device, especially designed for lifetime test of automotive control units.

All software components are integrated ready to use a single CMM or several CMMs in daisy-chain. Evaluation of minimum, maximum and average is included. Streaming of CMM data is available.

For further information see

http://www.irs-engineering.com/solutions/irs-messy/

http://www.irs-systeme.de/solutions/irs-messy/ (German)

The following table shows the connections of MesSy and CMM:

MesSy signal	MesSy Pin @ X1	CMM_II Signal	CMM_III Pin	wire	Remarks
LVDS_Out_3+	B16	Rx+	B17	Twisted pair	First CMM in chain
LVDS_Out_3-	C16	Rx-	C17		
LVDS_Out_2+	B15	En_In+	B16	Twisted pair	
LVDS_Out_2-	C15	En_In-	C16		
LVDS_Out_1+	B14	Clk_In+	B15	Twisted pair	
LVDS_Out_1-	C14	Clk_In-	C15		
LVDS_In_1+	C12	Tx+	B18	Twisted pair	Tx from the CMM, which is the last in chain
LVDS_In_1-	B12	Tx-	C18		
Trigger 1…16	A7-A12 B7-B11 C7-C11	On/Off	B15	Single wire	Use any of the 16 Trigger lines from MesSy

Following example shows the connection of the communication lines, power and On/Off-control of two CMMs:

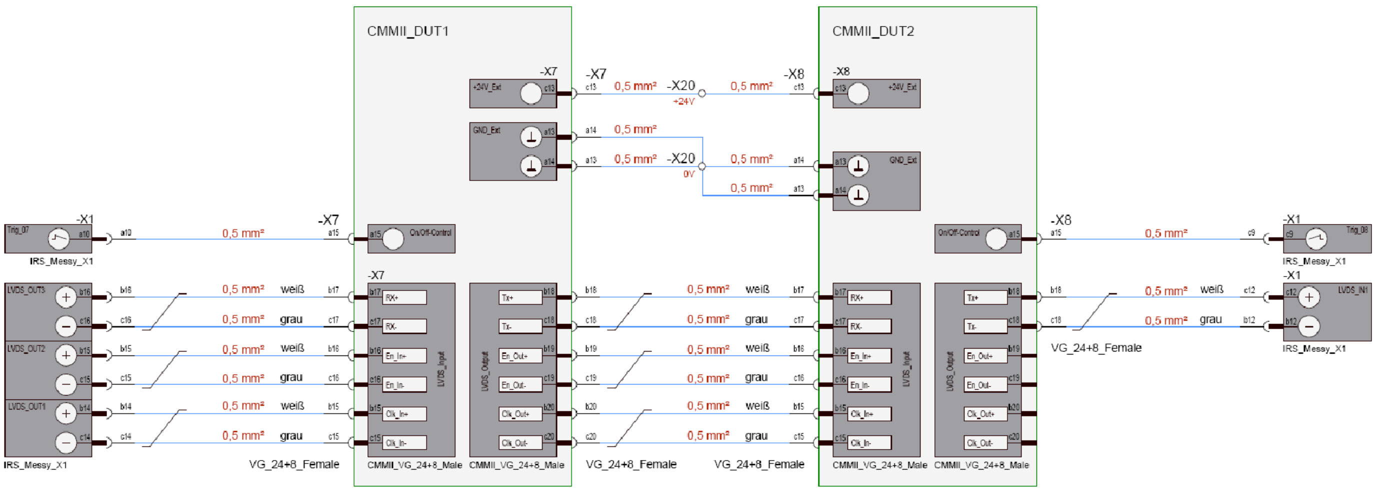

Figure 7: MesSy connections

A Backplane for MesSy, which reduces the wiring efforts by using standard CAT5..7 cables will be available by end 2014.

2.4.2.1 Data Acquisition with NI Compact RIO

For data acquisition IRS provides interface modules for the Compact-RIO platform from National Instruments. The software may be integrated into the user's application. For data acquisition with Compact-RIO please contact IRS.

2.4.2.2 Custom Data Acquisition

Any module with an SPI-interface may be used for data acquisition, like microcontroller devices. The level of each signal must meet the LVDS standard to maintain proper communication over long wires in harsh environments.

See chapter 3.4 for timing and data format of the communication.

2.4.3 Backplane for easy connection

To access the interfaces with an easy wiring, the use of the additional backplane is recommended. Especially when using LVDS-SPI, standard cables may be applied, which reduce the efforts for wiring and provide proper shielding.

Three backplanes are available:

- LVDS-SPI / 5HP width (25,4mm)
- CAN / 5HP width (25,4mm)
- LVDS+CAN / 8HP width (40,64mm)

2.4.3.1 LVDS-SPI Backplane

LVDS-SPI backplane may be used, when no CAN interface is necessary and many CMMs have to be integrated on small room. The backplane enables easy connection of the LVDS-SPI interface with standard CAT5…CAT7 cables, while providing proper shielding

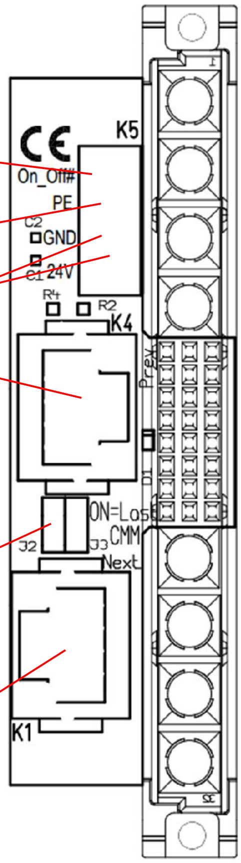

Figure 8: LVDS-SPI backplane

Following figure illustrates the options, how to pass data back to the data acquisition system. First option is used for clock rates up to 1,5Mbit/s, second option needs the clock and enable feedback for higher data rates.

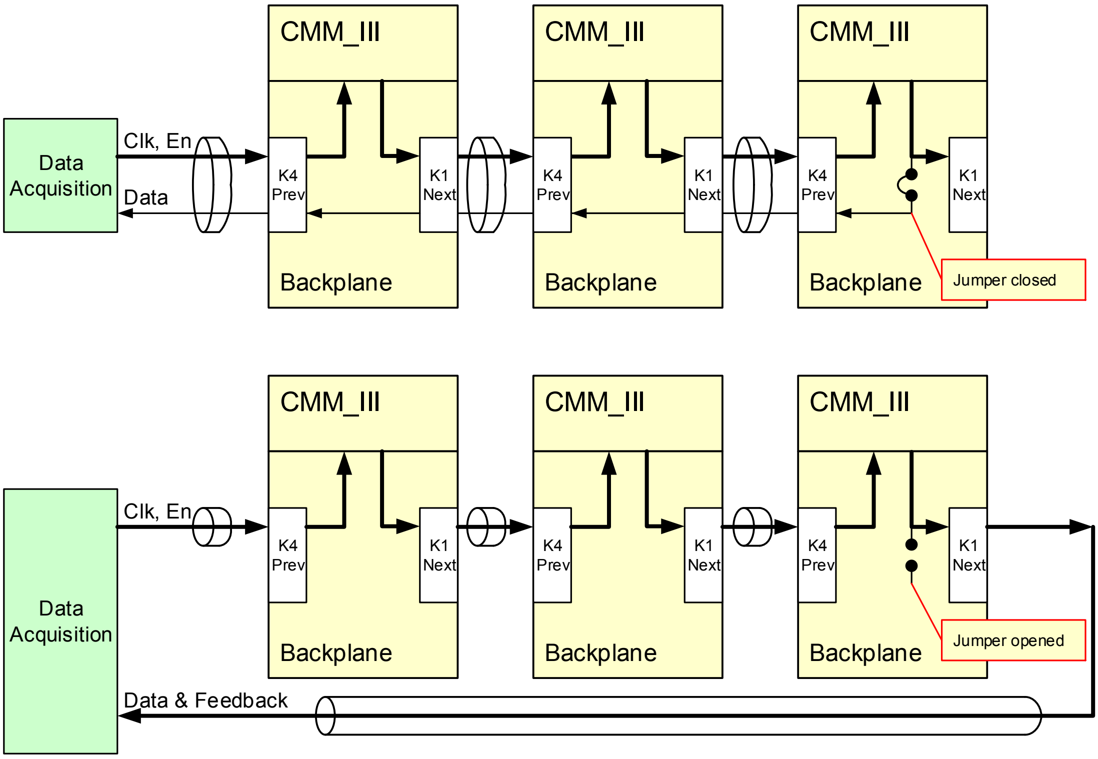

Figure 9: LVDS-SPI backplane options

2.4.3.2 CAN Backplane

CAN Backplane may be used, when no LVDS-SPI interface is necessary and many CMMs have to be integrated on small room.

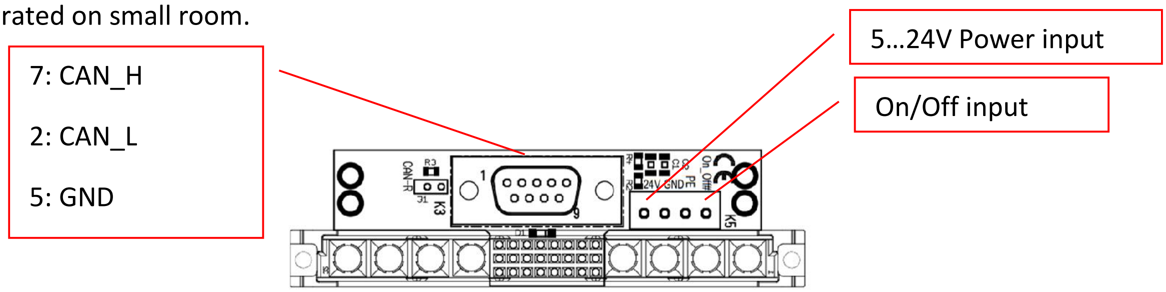

Figure 10: CAN backplane options

2.4.3.3 LVDS+CAN Backplane

LVDS-SPI and CAN may be both accessed, with the drawback that more room has to be reserved for the connectors.

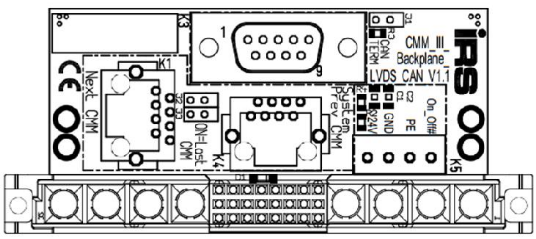

Figure 11: LVDS-SPI + CAN backplane

3 Functional Description

The following section describes the functionality of the internal measurement and every interface.

3.1 Current measurement

3.1.1 Measurement procedure

The current is measured internally at a sampling rate of 320kS/s, where 8 samples are averaged to 1 internal sample. Thus, a "real" internal sampling rate of 40kS/s is achieved.

The current range is selected automatically, when CMM is switched on. Range selection does not influence the output voltage. Except large current changes of several amperes will result in short time voltage drops of few 100 mV. See Chapter 3.1.2 for dynamic behavior.

For every interface (RS232, CAN, LVDS-SPI) an independent averaging is implemented, which starts averaging with every readout. Following figure illustrates the averaging.

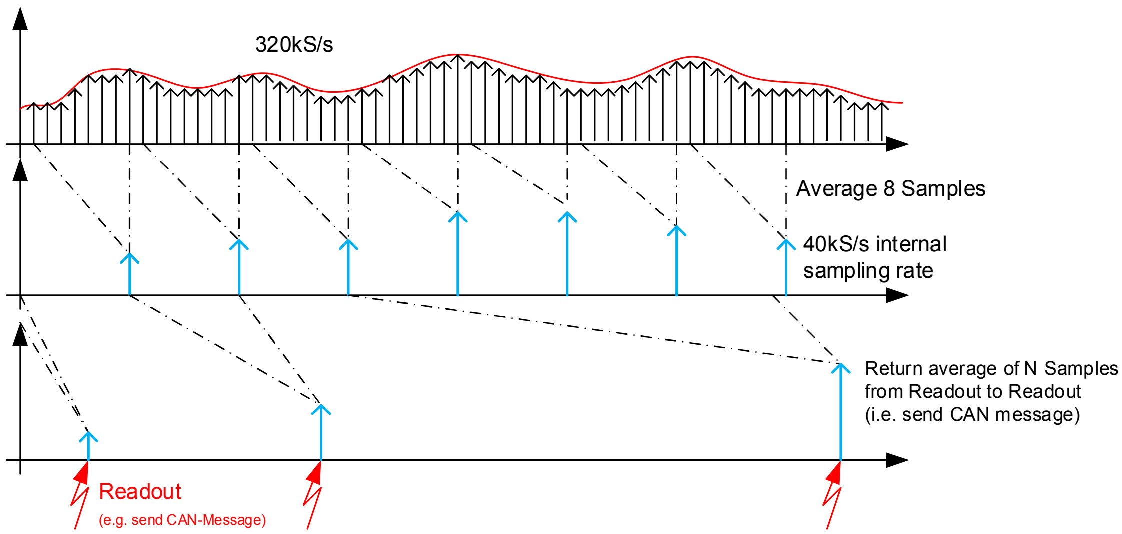

Figure 12: Measurement procedure

I.e. if CAN sends its message every millisecond, 40 internal samples are averaged, which are already an average of 8 samples each.

Averaging from readout to readout does not influence the averaging on other interfaces. I.e. Averaging is independend for

- RS232
- CAN
- LVDS-SPI

An overflow of the averaging may not occur even after a year of readout interval.

3.1.2 Reverse current detection

Reverse currents can't be measured, but a flag is signaled, when a current of several 100mA is applied in reverse direction. The output data on CAN or LVDS-SPI is an invalid number, exceeding the measurement range (0xEEEEEEEE in hexadecimal representation).

3.1.3 Dynamic behavior

Switching between the ranges occurs very fast from low current range to high current ranges within less than 1µs (500ns typical) after the current range has been exceeded.

The following figure shows the typical behavior at a current step from some µA to more than 60 amps.

- yellow trace: current through the module
- Red trace: voltage drop over the module

Figure 13: Dynamic behavior at a current step of 60A

The ring at the voltage drop results from the inductive component of the applied load.

The following figure show the zoom of positive and negative edge of the current step.

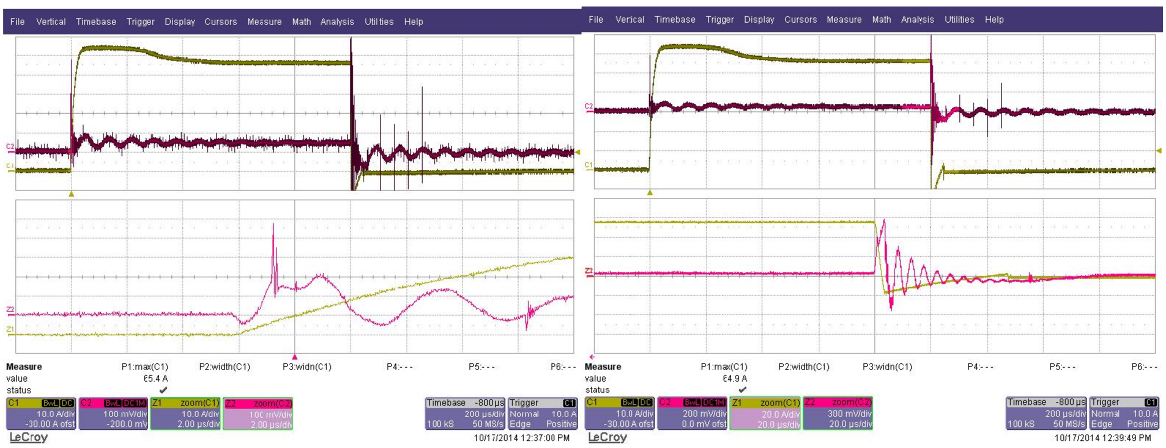

Figure 14: Dynamic behavior at a current step of 60A (Zoom)

Conclusion: The voltage drop while switching occurs is well below 1V for a duration of some microseconds maximum.

3.2 ON/Off control

CMM_III may be switched on or off by a hardware input or by software. The following parameters can be adjusted by configuration:

- Active level (high or low) of the hardware input
- Priority of hardware or software on/off-control

Configuration may be changed by the command “ONMOD=x” via RS232 or CAN using ISO-TP protocol, while x one of the following numbers 0…7:

ID	Name	HW input	SW setting	Description
0	ONOFF_Mode_ExtHighActive	YES	NO	CMM is ON, when high level is applied at HW input, SW setting ignored
1	ONOFF_Mode_ExtLowActive	YES	NO	CMM is ON, when low level is applied at HW input, SW setting ignored
2	ONOFF_Mode_Int	NO	YES	CMM is only controlled by SW
3	ONOFF_Mode_ExtHighActiveAndInt	YES	YES	CMM is ON @ high level is at HW input AND SW setting is ON.
4	ONOFF_Mode_ExtLowActiveAndInt	YES	YES	CMM is ON @ low level is at HW input AND SW setting is ON.
5	ONOFF_Mode_ExtHighActiveOrInt	YES	YES	CMM is ON @ high level is at HW input OR SW setting is ON.
6	ONOFF_Mode_ExtLowActiveOrInt	YES	YES	CMM is ON @ low level is at HW input OR SW setting is ON.
7	ONOFF_Mode_AlwaysOn	NO	NO	CMM is always ON.

3.3 CAN data output

CAN data is sent cyclically on a specified CAN ID. This CAN ID is adjustable by RS232 configuration.

Value	Min	Typical	Max	Unit	Remarks
CAN Type	-	High Speed	-	-	CAN Transceivers are always active. No sleep mode is applied.
CAN Termination	117	120	123	Ω	Termination Resistor is R40 and may be removed if desired
CAN Baud rate	50	1000	2000	kBit/s	Default baud rate is 1MBit/s
CAN ID		0x1C2		Hex	Identifier may be adjusted by RS232 or CAN configuration
Extended ID		No			11-Bit or 29-Bit Identifier may be adjusted by RS232 configuration
CAN transmit interval	1	5	30000	ms	Transmit interval may be adjusted by RS232 configuration
CAN data length		5			4 Bytes for Current and 1 Byte for Range
CAN data resolution		100		nA	One bit of the returned current represents 100nA of real measured current

Content of the transmitted CAN message is as follows:

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4
Current Bit 0..7	Current Bit 15..8	Current Bit 23..16	Current Bit 31..24	Range
0x00000000 … approx. 0x7270 E000				0 … 6

Current is returned as 32 Bit unsigned long value in steps of 100nA split into the first 4 bytes of the CAN message.

When CMM_III is in off state, the returned value is 0xFFFFFFFF, which represents 429.4967295 A, which is no valid measurement value.

When CMM_III is in on state, but reverse current is applied, the returned value is 0xEEEEEEEE, which represents 400.8636142 A, which is no valid measurement value.

3.4 LVDS-SPI data output

The measurement value can be read via an SPI-interface on LVDS level. SPI yields high data rates, while LVDS level yields low noise radiation and higher noise immunity, due to differential signaling.

The protocol for reading data of a single current measurement module is as follows:

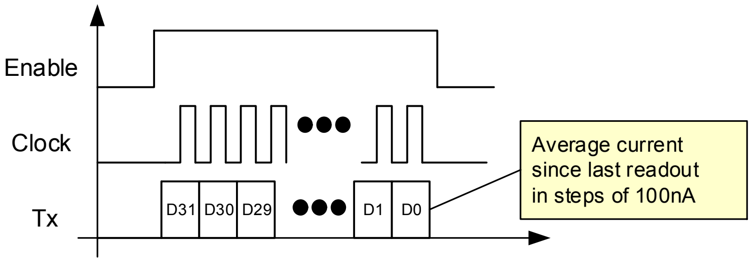

Figure 15: LVDS-SPI for single module

When several modules are connected in daisy chain, the data from the previous module is forwarded to the output as following figure illustrates:

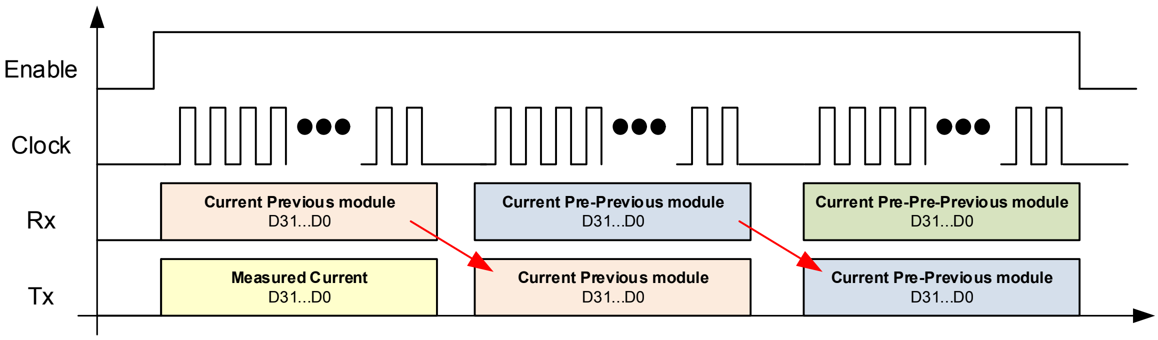

Figure 16: LVDS-SPI "Daisy chain" for several modules

When En_In is asserted, the module sends out its current value during the next 32 clock cycles with MSB first. After 32 clock cycles the received data word is forwarded. En_in must be valid for 32 clock cycles, multiplied by number of CMMs in the chain. The current is returned as 32 Bit unsigned long value in steps of 100nA.

3.5 RS232 data output

The measurement value is sent over RS232 in the interval of 100ms. The interval is adjustable via RS232 configuration. Connect the PC's COM-Port to the RS232 at the front cover with a simple 1:1 Sub-D-Cable (no Null modem).

Use the a standard terminal program with the following settings:

- 9600 Baud
- No parity
- 1 Stop bit
- No Handshake

You will see the module's output as following picture illustrates

```
R=0<HT>      I = 5307.9  uA<HT>  Min=8.2   uA<HT>     Max=14463.8  uA<CR><LF>
R=0<HT>      I = 10.1  uA<HT>    Min=9.2   uA<HT>     Max=10.6  uA<CR><LF>
R=2<HT>      I = 9629.6  uA<HT>  Min=10.5  uA<HT>     Max=14459.8  uA<CR><LF>
R=0<HT>      I = 10.7  uA<HT>    Min=8.1   uA<HT>     Max=822.9  uA<CR><LF>
R=0<HT>      I = 10.8  uA<HT>    Min=10.6  uA<HT>     Max=11.0  uA<CR><LF>
R=0<HT>      I = 10.9  uA<HT>    Min=10.8  uA<HT>     Max=11.1  uA<CR><LF>
R=0<HT>      I = 11.0  uA<HT>    Min=10.9  uA<HT>     Max=11.1  uA<CR><LF>
R=0<HT>      I = 11.0  uA<HT>    Min=10.9  uA<HT>     Max=11.1  uA<CR><LF>
R=0<HT>      I = 10.9  uA<HT>    Min=10.8  uA<HT>     Max=11.1  uA<CR><LF>
R=0<HT>      I = 10.8  uA<HT>    Min=10.7  uA<HT>     Max=11.0  uA<CR><LF>
R=0<HT>      I = 10.6  uA<HT>    Min=10.4  uA<HT>     Max=10.8  uA<CR><LF>
R=0<HT>      I = 10.4  uA<HT>    Min=10.2  uA<HT>     Max=10.6  uA<CR><LF>
R=0<HT>      I = 10.1  uA<HT>    Min=9.9   uA<HT>     Max=10.4  uA<CR><LF>
R=0<HT>      I = 9.5  uA<HT>     Min=9.1   uA<HT>     Max=10.0  uA<CR><LF>
R=0<HT>      I = 8.6  uA<HT>     Min=8.2   uA<HT>     Max=9.2  uA<CR><LF>
R=0<HT>      I = 7.8  uA<HT>     Min=7.5   uA<HT>     Max=8.3  uA<CR><LF>
R=0<HT>      I = 7.1  uA<HT>     Min=6.8   uA<HT>     Max=7.6  uA<CR><LF>
```

Figure 17: RS232 data output

R = Range

I = average current in µA since last readout

Min = minimum current in µA since last readout

Max = maximum current in µA since last readout

Every Line is terminated by "Carriage Return / Line feed" to separate between new data.

3.6 RS232 configuration

Configuration of the module using RS232 gives access to the CAN transmission configuration and the send interval for RS232-interface data output. Connect the PC's COM-Port to the RS232 at the front cover with a simple 1:1 Sub-D-Cable (no Null modem).

Use a standard terminal software with the settings

- 9600 Baud
- No parity
- 1 Stop bit
- No Handshake

You will see the module's output as shown in chapter 3.5.

By sending the commands of the following table the settings may be queried. Every Command has to be finished with a carriage return / line feed (\r\n). I.e. every command string needs to have the following format similar to the following example: "SINTV?\r\n"

- "\r" is ASCII character 0x0D
- "\n" is ASCII character 0x0A

Command	Description	Response	Remarks
SINTV?	Queries the actual serial transmission interval in milliseconds	SINTV=100ms	Default return value is 100ms
CINTV?	Queries the actual CAN transmission interval in milliseconds	CINTV=5ms	Default return value is 5ms
CANBD?	Queries the actual CAN baud rate in kBit/s	CANBD=1000	Default baud rate is 1000 kBit/s
CANID?	Queries the actual CAN ID	CANID=450	Returns the hexadecimal representation of the CAN ID. Default Can ID is 0x1C2
XTEND?	Queries the actual ID's extended or non-extended state	XTEND=0/1	Returns either 0 (default) or 1
SWVER?	reads out software version	SWVER = ssss..	ssss = software version – ASCII string
TEMPR?	Reads out current module temperature in °C	TEMPR = 26	
GLVAL?	Queries the latest values min, max and average	R= #\tI = # uA	Same as shown in chapter 3.5
ONMOD?	Queries the On/Off control configuration	ONMOD=0…7	See chapter 3.2 for description of ONMOD.
CMMON?	Queries the state of the internal ON/Off state	CMMON=0/1 0 => Off / 1 => On	This is only the SW setting. The state of the hardware input is not regarded for this command.
TPLID?	Queries local CAN-ID for ISO-TP	TPLID=451	Local ID is sent by CMM Default is 0x1C3
TPRID?	Queries remote CAN-ID for ISO-TP	TPRID=2047	Remote ID is sent by system Default is 0x7FF
TPLXT?	Queries Xtd-Flag of local CAN ID for ISO-TP	TPLXT=0/1	Default is 0 (short ID)
TPRXT?	Queries Xtd-Flag of remote CAN ID for ISO-TP	TPRXT=0/1	Default is 0 (short ID)

By sending the commands of the following table the settings may be set. Every Command has to be finished with a carriage return / line feed (\n\r). I.e. every command string needs to have the following format similar to the following example: “SINTV=100\n\r”

Command	Description	Response	Valid parameter range
SINTV=nnnnn	sets the actual serial transmission interval in milliseconds	SINTV=100ms	nnnnn = 20 … 12000 default = 100
CINTV=nnnnn	Sets the actual CAN transmission interval in milliseconds	CINTV=5ms	nnnnn = 1 …12000 default = 5
CANBD=nnnn	Sets the actual CAN baud rate in kBit/s	CANBD=1000	nnnn = 50 … 1000 default = 1000
CANID=xxxxxxxx	Sets the actual CAN ID	CANID=0x000001C2	xxxxxxxx = 00000000 … 1FFFFFFF (29bit) = 00000000 … 000007FF (11bit) default = 0x1C2
XTEND=b	Sets the actual ID’s extended or non-extended state	XTEND=0/1	b = 0 / 1, where 0 => 11 Bit CAN ID (default) 1 => 29 Bit CAN ID
ONMOD=x	Sets the On/Off control configuration	ONMOD=0…7	See chapter 3.2 for description of ONMOD values x
CMMON=X	Sets the internal ON/Off state	CMMON=0/1 0 => Off / 1 => On	
TPLID=451	Sets local CAN-ID for ISO-TP	TPLID=451	Local ID is sent by CMM Default is 0x1C3
TPRID=2047	Sets remote CAN-ID for ISO-TP	TPRID=2047	Remote ID is sent by system Default is 0x7FF
TPLXT=b	Sets Xtd-Flag of local CAN ID for ISO-TP	TPLXT=0/1	b = 0 / 1, where 0 => 11 Bit CAN ID (default) 1 => 29 Bit CAN ID
TPRXT=b	Sets Xtd-Flag of remote CAN ID for ISO-TP	TPRXT=0/1	

Furthermore the following additional commands using RS232 is defined. The commands also must be finished with “\r\n”.

Command	Description	Response	Valid parameter range
DEFLT!	all configuration parameters will be set to their default values	DEFAULT SETTINGS RESTORED	not applicable
INITC!	initializes CAN interface	CAN INIT	not applicable
RESET!	Resets the CMM internal microcontroller	<none> Restart with output of SW_Version.	not applicable
BOOTL!	Sets the CMM internal microcontroller to boot mode and resets the controller		not applicable

3.7 CAN configuration

Configuration of CMM may be performed using the CAN interface. For this configuration ISO-TP protocol is applied, according to ISO 15765-2. Normal addressing mode is used.

Configuration is performed by sending commands to CMM. Every command is acknowledged by a response.

3.7.1 ISO TP Header:

3.7.1.1 Single frame commands

Most commands and responses fit into a single CAN message. I.e. single frames according to ISO-15765-2 may be used. The respective CAN message carries the following data:

CAN ID	CAN Data							
		Command Header						
	Data_1	Data_2	Data_3	Data_4	Data_5	Data_6	Data_7	Data_8
TPRID (command) **TPLID** (response)	**SF_N_PCI =** **Length** 0x04…0x07	**Command**	**Action**	**Error-Code**	Reserved	*Command dependent data*		

3.7.1.2 Multiple frame commands

If more than one message is necessary for a command, First frames, flow-control and consecutive frames are used in the following order from top to bottom:

<table>
<tr><td>CAN ID</td><td colspan="8">CAN Data</td></tr>
<tr><td rowspan="4">TPRID
Command
First Frame</td><td>Data_1</td><td>Data_2</td><td>Data_3</td><td>Data_4</td><td>Data_5</td><td>Data_6</td><td rowspan="3">Data_7</td><td rowspan="3">Data_8</td></tr>
<tr><td colspan="2">FF_N_PCI</td><td colspan="4" rowspan="2">Command Header</td></tr>
<tr><td>ID+(Length MSB)</td><td>(Length LSB)</td></tr>
<tr><td>0x10</td><td>Length</td><td>Command</td><td>Action</td><td>Error-Code</td><td>Reserved</td><td colspan="2">Command dependent data</td></tr>
<tr><td rowspan="2">TPLID
Response
Flow control</td><td>FC_N_PCI</td><td>Data_2</td><td>Data_3</td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td>0x30</td><td>BS
Block size = 0</td><td>ST_min
Separation time = 0</td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td rowspan="4">TPRID
Command
Consecutive frame</td><td>CF_N_PCI</td><td>Data_2</td><td>Data_3</td><td>Data_4</td><td>Data_5</td><td>Data_6</td><td>Data_7</td><td>Data_8</td></tr>
<tr><td>0x20</td><td colspan="7">Command dependent data</td></tr>
<tr><td>0x21</td><td colspan="7">Command dependent data</td></tr>
<tr><td>:</td><td colspan="7">:</td></tr>
</table>

If the multiple frame message is a response, exchange the CAN-IDs TPRID and TPLID.

3.7.1.3 Data Byte 1/2 xx_N_PCI:

Data byte 1 contains frame type and data length information according to ISO 15765-2.

For Single frames the values 0x04…0x07 are valid, which represent the number of following data bytes. 4 bytes minimum for Command, Action, Error-code, and Reserved - 7 bytes maximum including 3 command depending data bytes.

For first frames 0x10 in byte 1 is added to the command length, which can be found in byte 2.

For consecutive frames 0x20 is added to a message counter in byte 1.

3.7.2 Command Header

3.7.2.1 Data Byte 2/3 Command:

Byte 2 for single frames, byte 3 for multiple frames

Command ID	Command	Description
0x00	CAN_Cmd_NOOPR	No operation
0x01	CAN_Cmd_RESET	Resets the CMM internal microcontroller
0x02	CAN_Cmd_SWVER	reads out software version
0x03	CAN_Cmd_DEFLT	all configuration parameters will be set to their default values
0x04	CAN_Cmd_ONMOD	Queries or sets the On/Off control configuration
0x05	CAN_Cmd_CMMON	Queries or sets the state of the internal ON/Off state
0x06	CAN_Cmd_GLVAL	Queries the latest values min, max and average
0x07	CAN_Cmd_TEMPR	Reads out current module temperature in °C
0x08	CAN_Cmd_SINTV	Queries or sets the actual serial transmission interval in milliseconds
0x09	CAN_Cmd_CANBD	Queries or sets the actual CAN baud rate in kBit/s
0x0A	CAN_Cmd_CIDIN	Queries or sets both CAN identifier and interval (combines commands CANBD and CANID of serial configuration)
0x0B	CAN_Cmd_TPLID	Queries or sets both CAN identifier and Extended-flag of Identifier (combines commands TPLID and TPLXT of serial configuration)
0x0C	CAN_Cmd_TPRID	Queries or sets both CAN identifier and Extended-flag of Identifier (combines commands TPLID and TPLXT of serial configuration)
0x0D	CAN_Cmd_INITC	initializes CAN interface

3.7.2.2 Data Byte 3/4 Action:

Byte 3 for single frames, byte 4 for multiple frames

Command ID	Command	Valid for Direction	Description
0x00	CAN_Action_Get	Command to CMM	Query operation to read data from CMM
0x01	CAN_Action_Set		Set operation
0x02	CAN_Action_Exe		Execute without data read or write.
0x03	CAN_Action_Ret	Response from CMM	

3.7.2.3 Data Byte 4/5 Error-code:

Byte 4 for single frames, byte 5 for multiple frames

This data byte is only valid for responses from CMM. In commands to CMM this byte should always be 0x00 = No error.

Command ID	Command	Description
0x00	CAN_Error_None	Command has been successful, no error
0x01	CAN_Error_HeaderLength	Header Bytes 1..4 were not complete
0x02	CAN_Error_DataLength	Number of data bytes didn't fit to command
0x03	CAN_Error_UnknownCmd	Unknown command
0x04	CAN_Error_Action	Action not supported for this command
0x05	CAN_Error_VOOR	Value out of range (an invalid parameter has been passed)

3.7.2.4 Data Byte 5/6 Reserved:

Byte 5 for single frames, byte 6 for multiple frames. This byte is reserved for future use.

3.7.3 Command overview:

Following commands are available:

Command	Length	Command Header				*Command dependent data*
		Command	Action	Error-code	Reserved	
CAN_Cmd_NOOPR	4	**0x00**	**0x02** (Exe)	**0x00**	**0x00**	
CAN_Cmd_RESET	4	**0x01**	**0x02** (Exe)	**0x00**	**0x00**	
CAN_Cmd_SWVER	4	**0x02**	**0x00** (Get)	**0x00**	**0x00**	
CAN_Cmd_DEFLT	4	**0x03**	**0x02** (Exe)	**0x00**	**0x00**	
CAN_Cmd_ONMOD	4	**0x04**	**0x00** (Get)	**0x00**	**0x00**	
	5		**0x01** (Set)			**OnMode** = 0…7
CAN_Cmd_CMMON	4	**0x05**	**0x00** (Get)	**0x00**	**0x00**	
	5		**0x01** (Set)			**0 / 1**
CAN_Cmd_GLVAL	4	**0x06**	**0x00** (Get)	**0x00**	**0x00**	
CAN_Cmd_TEMPR	4	**0x07**	**0x00** (Get)	**0x00**	**0x00**	
CAN_Cmd_SINTV	4	**0x08**	**0x00** (Get)	**0x00**	**0x00**	
	8		**0x01** (Set)			**Serial Interval [ms]**: Bit 7…0 | Bit 15..8 | Bit 23…16 | Bit 31…24
CAN_Cmd_CANBD	4	**0x09**	**0x00** (Get)	**0x00**	**0x00**	
	8		**0x01** (Set)			**CAN Baud rate kBit/s**: Bit 7…0 | Bit 15..8
CAN_Cmd_INITC	4	**0x0D**	**0x02** (Exe)	**0x00**	**0x00**	

Please note, that only command header and command dependent data is shown. ISO 15765 header has to be added and messages have to be split into several messages, if necessary.

PLEASE NOTE: Changing CAN baud-rate may cause bus errors. CMM should be restarted afterwards.

Command	Length	Command Header				*Command dependent data*
		Command	Action	Error-code	Reserved	
CAN_Cmd_CIDIN	4	**0x0A**	**0x00** (Get)	**0x00**	**0x00**	
	12		**0x01** (Set)			

Byte_1 … 4					Byte 5…8			
CAN ID				**Xtended** = 0 / 1	**CAN Interval [ms]**			
Bit 7…0	Bit 15..8	Bit 23…16	Bit 30…24	Bit 31	0..	..	..	31

<table>
<tr><th rowspan="2">Command</th><th rowspan="2">Length</th><th colspan="4">Command Header</th><th colspan="5" rowspan="2">Command dependent data</th></tr>
<tr><th>Command</th><th>Action</th><th>Error-code</th><th>Reserved</th></tr>
<tr><td rowspan="4">CAN_Cmd_TPLID</td><td>4</td><td rowspan="4">0x0B</td><td>0x00 (Get)</td><td rowspan="4">0x00</td><td rowspan="4">0x00</td><td colspan="5"></td></tr>
<tr><td rowspan="3">8</td><td rowspan="3">0x01 (Set)</td><td colspan="5">Byte_1 … 4</td></tr>
<tr><td colspan="4">CAN ID</td><td>Xtended
= 0 / 1</td></tr>
<tr><td>Bit 7…0</td><td>15..8</td><td>23…16</td><td>30…24</td><td>Bit 31</td></tr>
<tr><td rowspan="4">CAN_Cmd_TPRID</td><td>4</td><td rowspan="4">0x0C</td><td>0x00 (Get)</td><td rowspan="4">0x00</td><td rowspan="4">0x00</td><td colspan="5"></td></tr>
<tr><td rowspan="3">8</td><td rowspan="3">0x01 (Set)</td><td colspan="5">Byte_1 … 4</td></tr>
<tr><td colspan="4">CAN ID</td><td>Xtended
= 0 / 1</td></tr>
<tr><td>Bit 7…0</td><td>15..8</td><td>23…16</td><td>30…24</td><td>Bit 31</td></tr>
</table>

3.7.4 Response overview:

Every command is acknowledged by a response.

Most commands are executed first and send a response after execution. Exceptions from this rule are as follows:

- RESET:
  - The response is sent immediately.
  - Reset is performed afterwards.
- TPLID, TPRID:
  - The response is sent with the previous CAN ID.
  - Respective CAN ID for ISO-TP is changed after the response has been sent.
- CANBD:
  - Baud rate is changed after response has been sent.
  - Please note, that bus errors may occur after baud rate has changed. CMM should be restarted afterwards.

3.7.4.1 Negative Responses

Negative responses are returned, when a failure occurred. I.e. the command was invalid and has been rejected.

Response To Command	Length	Response Header			
		Command	Action	Error-code	Reserved
Any	4	**0xXX**	**0x03 (Ret)**	**0xYY** (see chapter 3.7.2.3)	**0x00**

3.7.4.2 Positive Responses

Positive responses to the respective commands are as follows. Please note, that only command header and command dependent data is shown. ISO-TP header has to be added and messages have to be split into several messages, if necessary.

Response to Command	Length	Response Header				*Command dependent data*
		Command	Action	Error-code	Reserved	
CAN_Cmd_NOOPR	4	**0x00**	**0x03** (Ret)	**0x00**	**0x00**	
CAN_Cmd_RESET	4	**0x01**	**0x03** (Ret)	**0x00**	**0x00**	
CAN_Cmd_SWVER	18	**0x02**	**0x03** (Ret)	**0x00**	**0x00**	*SW Version ASCII string (14 Bytes)*
CAN_Cmd_DEFLT	4	**0x03**	**0x03** (Ret)	**0x00**	**0x00**	
CAN_Cmd_ONMOD	5	**0x04**	**0x03** (Ret)	**0x00**	**0x00**	**OnMode** = 0…7
CAN_Cmd_CMMON	5	**0x05**	**0x03** (Ret)	**0x00**	**0x00**	0 / 1
CAN_Cmd_TEMPR	4	**0x07**	**0x03** (Ret)	**0x00**	**0x00**	**Temperature [°C]** Bit 0…7 | 8…15
CAN_Cmd_SINTV	8	**0x08**	**0x03** (Ret)	**0x00**	**0x00**	**Serial Interval [ms]** Bit 7…0 | 15..8 | 23…16 | 31…24
CAN_Cmd_CANBD	8	**0x09**	**0x03** (Ret)	**0x00**	**0x00**	**CAN Baud rate kBit/s** Bit 7…0 | 15..8
CAN_Cmd_INITC	4	**0x0D**	**0x03** (Ret)	**0x00**	**0x00**	

Response to Command	Length	Response Header				*Command dependent data*
		Command	Action	Error-code	Reserved	
CAN_Cmd_GLVAL	23	**0x06**	**0x03** (Ret)	**0x00**	**0x00**	

Byte_1	Byte_2	Byte_3	Byte_4…7				Byte 8…11				Byte 12…15				16…19			
CMMON	**Negative**	**Range**	**Average** [100nA steps]				**Min** [100nA steps]				**Max** [100nA steps]				**Nr Samples**			
0/1	0/1	0…6	Bit 7…0	15..8	23…16	31…24	0..	..	..	31	0..	..	..	31	0..	..	..	31

Response to Command	Length	Response Header				*Command dependent data*				
		Command	Action	Error-code	Reserved					
CAN_Cmd_TPLID	8	**0x0B**	**0x03** (Ret)	**0x00**	**0x00**	Byte_1 … 4				
						CAN ID				**Xtended** = 0 / 1
						Bit 7…0	15..8	23…16	30…24	Bit 31
CAN_Cmd_TPRID	8	**0x0C**	**0x03** (Ret)	**0x00**	**0x00**	Byte_1 … 4				
						CAN ID				**Xtended** = 0 / 1
						Bit 7…0	15..8	23…16	30…24	Bit 31

3.7.5 Example CAN traces

The following CAN traces should help to understand the tables above. The command and responses are marked in the respective colors: Command / Response. Please note, that flow control frames are integrated in the respective command or response, but they are sent by the respective counterpart.

CMMON: switch ON

2014.12.11 - 13:52:21.724	0x1C3	0x05 0x05 0x01 0x00 0x00 0x01 0x00 0x00
2014.12.11 - 13:52:21.724	0x7FF	0x04 0x05 0x03 0x00 0x00 0x00 0x00 0x00

CMMON: switch OFF

2014.12.11 - 13:52:29.924	0x1C3	0x05 0x05 0x01 0x00 0x00 0x00 0x00 0x00
2014.12.11 - 13:52:29.924	0x7FF	0x04 0x05 0x03 0x00 0x00 0x00 0x00 0x00

SWVER (Read SW Version)

2014.12.11 - 13:53:59.118	0x1C3	0x05 0x02 0x00 0x00 0x00 0x00 0x00 0x00
2014.12.11 - 13:53:59.118	0x7FF	0x10 0x12 0x02 0x03 0x00 0x00 0x43 0x4D
2014.12.11 - 13:53:59.122	0x1C3	0x30 0x00 0x00 0x00 0x00 0x00 0x00 0x00
2014.12.11 - 13:53:59.122	0x7FF	0x21 0x4D 0x5F 0x49 0x49 0x49 0x5F 0x56
2014.12.11 - 13:53:59.122	0x7FF	0x22 0x5F 0x31 0x5F 0x32 0x00 0x00 0x00

SINTV (set serial interval)

2014.12.11 - 13:56:46.436	0x1C3	0x10 0x08 0x08 0x01 0x00 0x00 0x80 0x00
2014.12.11 - 13:56:46.437	0x7FF	0x30 0x00 0x01 0x00 0x00 0x00 0x00 0x00
2014.12.11 - 13:56:46.440	0x1C3	0x21 0x00 0x00 0x00 0x00 0x00 0x00 0x00
2014.12.11 - 13:56:46.442	0x7FF	0x04 0x08 0x03 0x00 0x00 0x00 0x00 0x00