

smartIO 8x24

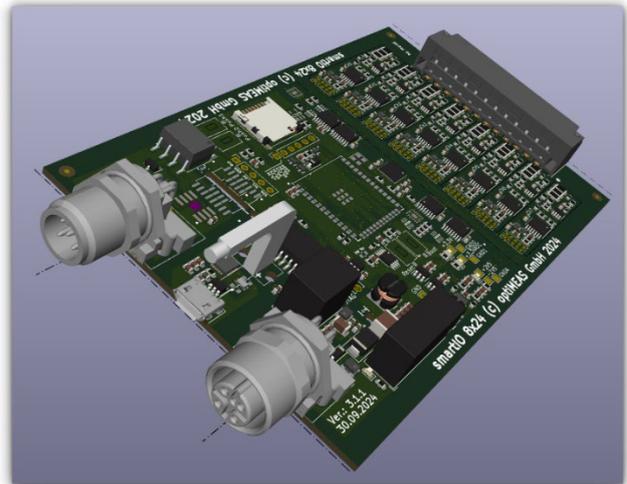
Technische Daten und Beschreibung

Das **smartIO 8x24** bietet Anschlussmöglichkeiten für 8 analoge Signalquellen für Spannungs- oder Strommessung mit folgenden, wählbaren Messbereichen:

- $\pm 160\text{V}$, $\pm 80\text{V}$, $\pm 40\text{V}$, $\pm 20\text{V}$, $\pm 10\text{V}$, $\pm 5\text{V}$
- $\pm 200\text{mA}$, $\pm 100\text{mA}$, $\pm 50\text{mA}$, $\pm 25\text{mA}$

Weitere Eckparameter sind

- Abtastrate bis 1 kHz
- Common-Mode Spannung bis 100V
- Stromeingang mit Überlastungsschutz
- Sensor-Linearisierung über Polynom
- Verschiedene Filtermöglichkeiten
- Isolation zwischen Prozess, Versorgung und digitalen Schnittstellen bis 1700 V DC

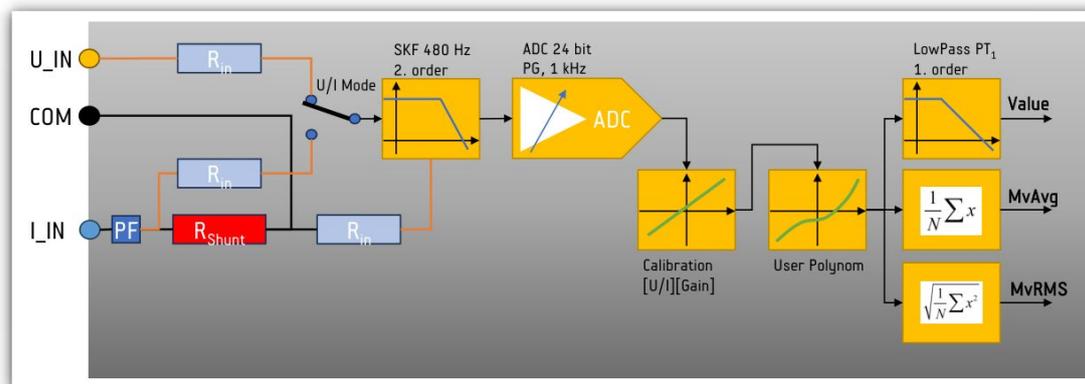


Die Konfiguration des Moduls erfolgt über eine serielle Schnittstelle (USB) und den umfangreichen SCPI-Befehlssatz mit einem beliebigen Terminal-Programm oder bevorzugt *optiCONTROL*. Diese Konfiguration umfasst auch die Adressen, Senderaten und Belegung der CAN-Botschaften. Eine Konfiguration im Feld kann über den CAN-Bus mittels der Konsole-Applikation *smiocan* erfolgen. Diese ist Bestandteil in den aktuellen Versionen im Yocto-Linux der smart-Familie (*smartMINI*, *smartRAIL*, ...).

Jedes Modul ist als RAIL-Ausführung einzeln im Hinblick auf Spannungsfestigkeit und Isolation zwischen Prozess- und USB-Seite geprüft. Mit dem stabilen Aluminiumgehäuse und der Schutzlackierung der Platine nach EN50155 werden auch die Anforderungen für einen Einsatz auf Schienenfahrzeugen erfüllt.

Auch wenn das Modul im Gehäuse des *smartRAIL* integriert ist, erfolgt die Verbindung für CAN und Spannungsversorgung über die beiden M12 Anschlüsse. Diese können mit einem kurzen M12-Anschlusskabel in Serie mit dem Anschluss des *smartRAIL* geschaltet werden.

Messfunktionen



Das **smartIO 8x24** besitzt 8 Eingänge, die jeweils identisch aufgebaut sind:

- Die Anschlüsse des Spannungseingangs (U_IN, COM) sind jeweils mit hoher Impedanz von $R_{in} = 1 \text{ M}\Omega$ und den Betriebsartenwahlschalter und den nachfolgenden Eingangsverstärker mit Sallen-Key-Filter (SKF) angeschlossen.
- Die Anschlüsse des Stromeingangs (I_IN, COM) sind über eine Poly-Fuse (PF) zum Schutz vor Überlast und einen kleinen Shunt-Widerstand $R_{Shunt} = 10 \text{ }\Omega$ miteinander verbunden. Der Spannungsabgriff am Shunt erfolgt mit hoher Impedanz von $R_{in} = 1 \text{ M}\Omega$ und führt über den Betriebsartenwahlschalter zum nachfolgenden Eingangsverstärker mit Sallen-Key-Filter (SKF).
- Die COM-Anschlüsse der 8 Kanäle sind untereinander nur über die Eingangsimpedanzen R_{in} und einen virtuellen Massepunkt miteinander verbunden. Damit sind die Eingänge in jeder Betriebsart für differenzielle Messung ausgelegt. Die Festlegung des COM Anschlusses auf ein Bezugspotential ist somit an jedem Kanal zwingend.
- Die Common-Mode Spannung¹ zwischen den Eingängen darf bis zu 100V DC betragen.
- Das Sallen-Key-Filter begrenzt die Bandbreite des Eingangs auf 480 Hz und wirkt damit als Anti-Aliasing-Filter für den nachfolgenden AD-Wandler.
- Der AD-Wandler nach dem Delta-Sigma-Verfahren ist konfigurierbar bis zu Abtastraten von 1 kHz, synchron für alle Eingangskanäle. Die programmierbare Eingangsverstärkung ermöglicht verschiedene Messbereiche, die Wandlungsbreite beträgt 24 bit.
- Für jede Verstärkung und Betriebsart (U/I) ist ein Kalibrierdatensatz hinterlegt, mit dem die Messspannung oder der Messtrom am Eingang berechnet wird.
- Diese kalibrierte Messgröße kann mit einem Anwenderpolynom² zur Abbildung einer Sensorkennlinie verrechnet werden. Damit kann das Modul direkt die physikalische Messgröße des Sensors über CAN ausgeben.
- Diese physikalische Messgröße wird parallel verschiedenen Filterstufen zugeführt:
 - einem Tiefpassfilter 1. Ordnung
 - einem gleitenden Mittelwert-Filter
 - einem gleitenden RMS-Filter

¹ Ausgehend vom mittleren Spannungspotential zwischen U_IN bzw. I_IN und COM.

² Polynom bis $a_{14}x^{14} + \dots + a_2x^2 + a_1x + a_0$, das Polynom wird nach dem Horner-Schema berechnet

Die Fensterbreite der gleitenden Filter und die Zeitkonstante für die Tiefpassfilter sind gemeinsam für alle Kanäle einstellbar. Mit einer Fensterbreite von 300 Samples können direkt auch True-RMS-Werte für AC-Signale mit den typischen Frequenzen von 50 Hz (15 Perioden), 60 Hz (18 Perioden) und 16 2/3 Hz (5 Perioden) gemessen werden.

Kommunikation

Die Messwerte werden auf dem CAN-Bus in einzelnen Botschaften übertragen. Bis zu 20 CAN-Botschaften können konfiguriert werden mit:

- CAN-ID (11-bit, 29-bit)
- Senderate in ms, nach Geräte-Start oder RTR
- Inhalt

Für den Inhalt steht eine große Auswahl an Datenelementen zur Verfügung, die nicht nur die gefilterten Messwerte, sondern auch Statusinformationen zum Kanal, die interne Temperatur des Moduls, Firmware- und Hardware-Stand, oder die physikalische Einheit umfassen. Bevorzugt werden die Messwerte als 4-byte <float> übertragen, wodurch bei entsprechender Planung und Einrichtung des Moduls eine weitere Skalierung auf der Empfängerseite entfällt. Die Übertragung als <int16> ist ebenfalls möglich. Dann ist die Skalierung der Ausgangsgröße allerdings vom gewählten Messbereich oder dem Anwender-Polynom abhängig, um den eingeschränkten Wertebereich bestmöglich auszunutzen.

Die Konfiguration erfolgt mit Hilfe eines erweiterten SCPI Protokolls über die serielle Schnittstelle, die am USB-Konfigurations-Anschluss verfügbar ist (COM, /dev/tty). Die serielle Schnittstelle³ kann mit einem beliebigen Terminal-Programm⁴ angesprochen werden. Das Modul ergänzt eine editierbare Kommandozeile mit Historie und bietet eine optional farbliche Ausgabe. Für bestimmte Befehle, die auf mehrere Kanäle gleichzeitig wirken, ist die Ausgabe lesbar strukturiert. Im Übrigen wird das verwendete Zeilenende (CR, LF, CRLF) automatisch ermittelt und so der Betrieb an einem (klassischen) Terminal ermöglicht. Mittels des ___? – Befehls kann eine ausführliche, dokumentierte Auflistung aller Befehle direkt vom Gerät abgerufen werden. Kritische Befehle sind durch einen Pass-Code geschützt. Die Konfigurationssoftware *optiCONTROL* stellt für die Konfiguration des Moduls leicht bedienbare Eingabemasken für die wesentlichen Einstellungen zur Verfügung.

```

___?
The help could be called in several ways:
___?  -> Print all commands
___? ? -> Print the list of all sections
'can:___?' -> Print the 'can' section help
___? can? -> Other way to print the 'can' section help
___? sta? -> Print commands starting with 'sta'

The commands can be entered in the short form (shown in capital letters in the help)
or in the long form. Intermediate lengths are not permitted.
The commands can be specified in uppercase or lowercase letters.
Parameters are separated with commas.
Several commands in one line are separated with ';'.
Comments start with '//'.
Not finished commands are shown in Gray.

SCPI commands:
=====
General Commands:
___? Print this help
? Prints an identifying string. The response ___?
will show the following information:
<company name>, <model nr>, <serial nr>, <firmware rev>
'SAVE' <par> Save parameters 'save 0'
'LOAD' <par> Load parameters 'load 0'
      par = 0: Flash
      #: set to defaults
      #: ALL to def, including serial NR...
'REPEAT' <period> Repeat the command line periodic, <adviceval? 0> 'rep 1000'
Command is added to another command with ';'.
It repeats the commands with the given period [ms]
Range 10 - 10000ms
The periodic repetition is ended by pressing any key
[esc] Print all following answers in one '*inl*'rep 1000;adviceval? 0;adviceval? 1
line separated by TAB. This is useful in
combination with the 'repeat' command.
Optional the separator characters could
be given as parameter. Example '*inl \tHallo\n'
Special Strings: '\n', '\t', '\t',
'&' = space, '\c' = '\', '\.' = '.'.

System Commands:
SYSTEM:___? Print System Help system:___?
SYSTEM:ERROR? Output last error system:serial 1800390000
SYSTEM:SERIAL? Set serial number
Structure: YYYYYPPPPNNNN
YY = production Year
PPPP = Product code (0062)
NNNN = serial Number
SYSTEM:SERIAL? Output serial number system:serial?
SYSTEM:PROD? Set production date system:prod?
SYSTEM:PROD? Output production date system:prod?
SYSTEM:COMMENT? Save comment to device system:comment Hello World
SYSTEM:COMMENT? Output comment system:comment?
SYSTEM:VARIANT? Set the variant of the module system:variant +-60V
SYSTEM:VARIANT? Output variant system:variant?
SYSTEM:DEBUG? Set debug level system:debug 0
0 = Off, 1, 2, 3
SYSTEM:DEBUG? Output debug level system:debug?
    
```

³ 38400 baud, 8 Daten, 1 Stopp, No Parity, [x] Local Echo, [x] Implicit CR with LF (Empfang)

⁴ Wie z.B. PuTTY oder TeraTerm – nicht Bestandteil des Lieferumfangs

Über zwei reservierte CAN-Botschaften (0x011, 0x012)⁵ kann zu der neuen smartIO Familie auch im Feld eine Service-Verbindung aufgebaut werden. Über die Botschaften wird ein erweitertes ISO-TP Protokoll gefahren. Neben der eigentlichen Datenverbindung (ASCII/binär) werden zusätzlich noch asynchron Statusinformationen zum Verbindungsmanagement und der Verarbeitung des Befehls auf Geräteseite implementiert. Hiermit wird es möglich, jeweils ein Gerät für die Kommunikation auf denselben CAN-Botschaften auszuwählen und selbst große Datenmengen sicher, bidirektional zu übertragen. Ferner ist ein Zugriffsschutz mittels Seed-Key-Verfahren implementiert.

Über diese Service-Verbindung

- wird der SCPI-Befehlssatz des Gerätes zur Konfiguration bedient und
- kann ein Firmware-Update der Gerätesoftware durchgeführt werden.

In den aktuellen Versionen der YOCTO Umgebung kann auf den Geräten der smart-Familie die Konsole-Anwendung *smiocan* zur Kommunikation über diese Service-Verbindung eingesetzt werden. Die App bietet folgende Möglichkeiten:

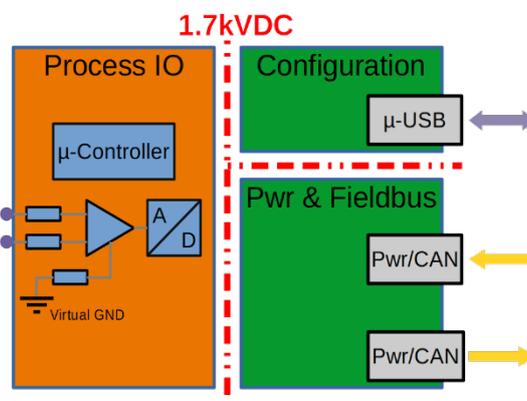
- Gerätesuche ausführen
- Gerätezeit/-datum stellen
- Diverse Logbuch-Ausgaben
- Kommandos von
 - der Linux Kommandozeile
 - einer Batch-Datei oder
 - interaktiv auszuführen
- Firmware-Updates für ein bestimmtes oder alle Geräte auszuführen

```

root@smartio-037:~# smiocan -h
Usage:
-----
smiocan [options]
Options:
Short Long Parameter Description
-----
-h --help          output command help
-p --port          <CAN-port> specify CAN port, mandatory
                    e.g. can0, can1, ...
-B --baud          <baud>   if assigned, setup CAN baudrate, may use 'k' for *1000
-G --gpio         <code>   if assigned, set GPIO to auto[<code><code>
                    auto:   automatic by port
                    0x<code><code>: ci chip, pi pin, li logic
                    example: 0x2b1: c = 2, p = 5, l = 1 (high)
-U --utcTime      <uint>   CAN id for UTC time broadcast message (0x10)
-T --taid         <uint>   CAN id to transmit SCPI stream to device (0x11, iso-tp#)
-B --raid         <uint>   CAN id to receive SCPI stream from device (0x11, iso-tp#)
-d --dump         <mask>   dump CAN frames to DEBUG log, 0x0d bitmask from
                    0x001: transmitted
                    0x002: received (known)
                    0x020: received (unknown)
                    0x040: loop
                    0x080: loop-ext
                    0x400: clock
-E --liblog       <path>   route libraries (Debug/Info/Error) log to specified output file
                    if none specified, out and cerr are used
-e --liblevel     <uint>   log message if >= level
-d --devlog       <path>   0..5, DEBUG, DUMP, SCPI, SYSTEM, INFO, WARNING, ERROR, FATAL
                    echo device output & SCPI commands to specified output file
-Q --query        <path>   Query and list all smartIO ESP32 devices
-k --pkey         <path>   default password for queried devices, implies run all mode
-M --hello        <str#> <str#> <str#> <str#> open device channel for netio using pkey, may be defined
                    multiple times, second device implies run all mode
feeding commands to smiocan and the device in the following order:
-c --command      <scpi>   add command line, multiple adds possible
-s --script       <path>   feed command lines from script
-x --fwimage      <path>   activate extended fw-upload sequence for given image
-i --interactive            enter interactive terminal mode
root@smartio-037:~#
    
```

Isolation der Anschlussgruppen

Die Eingangsgruppe für Versorgungsspannung und CAN-Bus, die USB Konfigurationsschnittstelle und die Anschlüsse zum Prozess sind voneinander isoliert, bahntauglich für Anlagen mit 110V nominaler Betriebsspannung bis zu einer Isolationsspannung von 1,7kV DC.



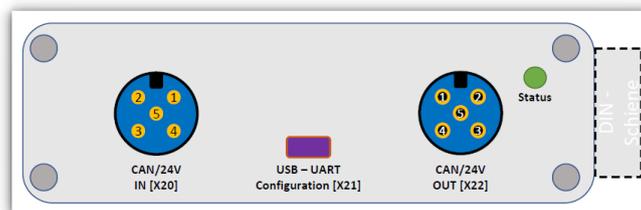
⁵ Die verwendeten Adressen stellen den optiMEAS Standard zur Administration der *smartIO* Familie dar. Sie sind für spezielle Einsatzbereiche änderbar, um eine Kollision mit bestehenden Kommunikationsprotokollen zu vermeiden. Im CANopen Standard sind diese Botschaften frei.

Prozesseingänge untereinander sind, je nach Ausführung des Moduls, hochohmig gegen den internen analogen Referenzpunkt (virtuelles GND) verbunden und somit – in der Regel vernachlässigbar - auch untereinander.

Schnittstellen des Moduls

Spannungsversorgung und CAN-Bus

Versorgungsspannung und CAN-Bus sind auf zwei M12 Steckern [X20] und [X22] entsprechend der Device-Net Belegung verfügbar. Die Spannungsversorgung von 24V wird ebenso wie die CAN-Bus Verbindung vom Stecker zur Buchse durchgeschleift. So können mehrere verschiedene smartIO-Module in Reihe geschaltet werden. Sofern kein weiteres smartIO-Modul folgt, ist der CAN-Bus mit einem Stecker mit integriertem Abschlusswiderstand (120 Ω) oder direkt an der Phoenix-Klemme zu terminieren. CAN-GND ist mit der Versorgungsmasse identisch.



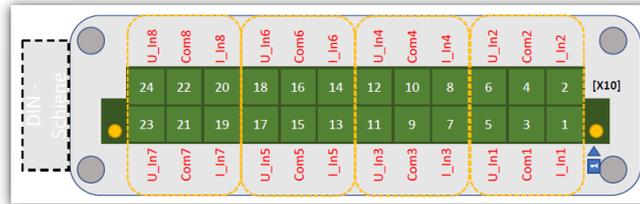
Pin	Signal	Beschreibung
1	CAN-Shield	
2	+24V	Spannungsversorgung +24V (nom., siehe techn. Daten)
3	GND	Bezugsmasse für Spannungsversorgung und CAN-Bus
4	CAN_H	CAN-Bus: CAN-High
5	CAN_L	CAN-Bus: CAN-LOW

Es können handelsübliche Device-Net-Kabel verwendet werden, die von Buchse auf Stecker führen. Bei konfektionierten Kabeln müssen für die Anschlusspaare 2 und 3 sowie 4 und 5 jeweils Twisted Pair Leitungsbündel gewählt werden.

Hinter der USB-UART Konfigurationsschnittstelle [X21] verbirgt sich ein USB-Seriell-Umsetzer (CP2102N), der unter Windows als virtueller COM-Port, unter Linux als `/dev/tty*` und `/dev/serial/by*...` abgebildet wird. Auch dieser Anschluss ist zur Prozessebene elektrisch isoliert. Zur Konfiguration des Moduls wird das ASCII-basierte SCPI-Protokoll verwendet. Eine Beschreibung der Befehle ist auf Anfrage erhältlich. Zur Konfiguration der smartIO-Module wird die *optiCONTROL* Software von optiMEAS empfohlen, die eine grafische Benutzeroberfläche mit entsprechend vorbereiteten Dialogen bereithält.

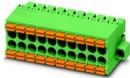
Messeingang

Für die Messeingänge [X10] werden doppelreihige Phoenix-Klemmen im Raster 3,5 mm verwendet.

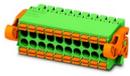


Passende Stecker von Phoenix sind

- Art.-Nr. 1790399, DFMC 1,5/12-STF-3,5

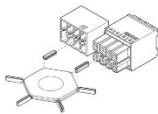


- Art.-Nr. 1790580, DFMC 1,5/12-ST-3,5-LR



Das Grundgehäuse am smartIO und die Stecker können nach Bedarf, falls mehrere smartIO Module zum Einsatz kommen, über das Kodierprofil verwechslungssicher markiert werden. Dadurch wird eine Vormontage der Stecker und ein verwechslungssicherer Anschluss möglich.

- Art.-Nr. 1790647, CP-DMC 1,5 NAT - Kodierprofil



Besondere Zulassungen und Erklärungen

Für die *smartRAIL-Ausführung* werden neben der CE Konformitätserklärung auch die nachfolgenden Zertifizierungen zur EN50155 vorgenommen:

<p>EG-Konformitätserklärung</p>		<p>Das CE-Zeichen zeigt die Übereinstimmung mit der</p> <ul style="list-style-type: none"> • EMV-Richtlinie, • RoHS 2011/65/EU (08.06.2011) und der • Niederspannungsrichtlinie an.
<p>Bahnanwendungen – Elektronische Einrichtungen auf Bahnfahrzeugen. EN 50155:2017</p>	   	<p>Die Systeme erfüllen die Norm für folgende Eigenschaften:</p> <ul style="list-style-type: none"> • <i>Umgebungsbedingungen:</i> <ul style="list-style-type: none"> ○ AX (2000m) EN50125-1 §4.2.1 ○ TX EN50155 §4.1.2 • <i>Klima</i>⁶ <ul style="list-style-type: none"> ○ Kälte EN50155 §13.4.4 ○ Trockene Wärme EN50155 §13.4.5 ○ Feuchte Wärme EN50155 §13.4.7 zyklisch • <i>Schwingen</i> IEC61373 §8 + 9 • <i>Schocken</i> IEC61373 §10 • <i>EMV, Isolation</i> EN50121-3-2 EN 61000-3-2/3 EN 55016-2-1/2 • <i>Brandschutz</i> EN45545-2 <p><i>Stückprüfungen gemäß</i> EN50155 §12.2</p> <ul style="list-style-type: none"> • <i>Sichtprüfung</i> EN50155 §12.2.1 • <i>Isolation</i> EN50155 §12.2.9.1 • <i>Stehspannung</i> EN50155 §12.2.9.2

⁶ Bereits nach EN 50155:2017
Änderungen und Irrtümer vorbehalten.
www.optimeas.com

Technische Daten

Versorgungsspannung / Umgebungsbedingungen

Symbol	Parameter	Bemerkung	Min	Typ	Max	Einheit
V _{CC}	Versorgungsspannung	mit Verpolungsschutz	8		36	V DC
	Überspannungsschutz		nein			
	ESD-Schutz	TVS-Diode			40	V
I _{CC}	Stromaufnahme	@ 24V		50	60	mA
	Steckverbinder	M12 A-Codiert (M + F)	M12 in / M12 out			
T _{operating}	Betriebstemperatur	EN 50155 / Bereich TX	-40		85	°C
	Relative Luftfeuchte	Nano-Beschichtung, 50°C	5		95	%
	Gehäuse		Aluminium			
L	Maße: Länge	ohne Stecker / Füße / Clip		124		mm
B	Breite			85		mm
H	Höhe			35		mm
m	Gewicht			330		g
	Verschraubung	Frontplatten gegen Korpus	M3			
	Montage (RAIL)	Tragschiene (EN 50022)	TS 35			
	Kühlung		passiv			
	Schutzklasse	(ISO 20653 - 2013)	IP54			
	Isolationswiderstand RAIL	@500V	10			GΩ
	Stehspannungsprüfung, 60s RAIL	Zu Messeingang		1.7		kV

CPU

Symbol	Parameter	Bemerkung	Min	Typ	Max	Einheit
	Prozessor		ESP32-S3			
	Familie		Xtensa® 32-bit LX7			
	Clock				240	MHz
	ROM	FLASH		384		kByte
	RAM	SRAM		512		kByte
	Datenbusbreite			32		bit

Analoge Eingänge, Digitalisierung

Symbol	Parameter	Bemerkung	Min	Typ	Max	Einheit
N_{AInU}	Typ / Anzahl	Spannung, differentiell		8		
U_{SENSOR}	Messbereich	Gain = 1	-160		160	V
		Gain = 2	-80		80	V
		Gain = 4	-40		40	V
		Gain = 8	-20		20	V
		Gain = 16	-10		10	V
		Gain = 32	-5		5	V
$r_U = \text{RMS}(\Delta U) / U_{SENSOR}$	Genauigkeit	Gain = 1 .. 8			0,02	%
R_{Ain}	Eingangswiderstand	zum virtuellen Massepunkt		1		M Ω
N_{AInI}	Typ / Anzahl	Strom, differentiell		8		
I_{SENSOR}	Messbereich	Gain = 1	-200		200	mA
		Gain = 2	-100		100	mA
		Gain = 4	-50		50	mA
		Gain = 8	-25		25	mA
$r_I = \text{RMS}(\Delta I) / I_{SENSOR}$	Genauigkeit	Gain = 1 .. 8			0,5	% FS
$c_I = \Delta c / c_{max}$	CM-Unterdrückung	ADC-Counts vor Kalibrierung			0,05	% FS
R_{Shunt}	Mess-Shunt	1.5 W ⁷		10		Ω
	Poly-Fuse	32V, fast			500	mA
R_{Ain}	Eingangswiderstand	zum virtuellen Massepunkt		1		M Ω
		Wandler				
	Wandler	Delta-Sigma		24		bit
f_{Ain}	Abtastrate				1000	Hz
f_{-3dB}	Bandbreite AAF	Sallen-Key-Filter, 2. Ord.		480		Hz
f_{Aout}	Ausgaberate	{CAN-Botschaft}	0		1000	Hz

⁷ Die gemessenen Ströme werden an den Shunts direkt in eine entsprechende Wärmeleistung von bis zu 0.4W / Kanal umgesetzt. Werden alle Kanäle konstant bei 200 mA betrieben, entsteht auf dem Modul eine zusätzliche Wärmeleistung von 3.2W! Diese ist zwar als Dauerlast erprobt und zulässig, führt aber je nach Umgebungsbedingungen zu einer deutlichen Erwärmung des gesamten Moduls. Für einen typischen Messbetrieb mit dynamischen Messsignalen sind keine negativen Auswirkungen zu erwarten.

Schnittstellen

Symbol	Parameter	Bemerkung	Min	Typ	Max	Einheit
	Typ / Anzahl	CAN 2.0B		1		
	Baudrate	parametrierbar		500	1000	kBit/s
	Steckverbinder	M12 A-Codiert	M12 in (M) / M12 out (F)			
	Abschlusswiderstand		nein			
	Isolation RAIL	zu Messeingang und USB	1,7 kV			
	Typ / Anzahl	Seriell / USB		1		
	Baudrate	Fest		38400		Bit/s
	Steckverbinder		Micro-USB			
	Chipsatz	Versorgung durch USB	CP2102N			
	Isolation RAIL	zu CAN und Messeingang	1,7 kV			

Das Modul meldet sich auf die SCPI-Anfrage *idn? mit einer Identifikation nach folgendem Muster, wobei Seriennummer, Versionsstände und Datum/Zeitstempel variieren werden:

```
optiMEAS, smartIO, 8X24, 1, 24006200011, 1.35, 0, Nov 22 2024 15:50:07, 20240610
```

Prozessabbild

Folgende Botschaften sind für die smartIO Familie mit ESP-CPU reserviert:

CAN-ID	CAN-ID (hex)	Datentyp	Richtung	Beschreibung	Einheit
16	0x010	uint48	an alle smartIO	Zeitstempel (Unix)	ms
		uint8		0	
		uint8		Prüfsumme	
17	0x011	ISO-TP+	an alle smartIO	Diagnoseschnittstelle (*)	
18	0x012	ISO-TP+	von aktiviertem smartIO	Diagnoseschnittstelle (*)	

(*) Für die Diagnoseschnittstelle ist das ISO-TP Protokoll um den Type Code 0xE0 für den Connection Management Layer (CML) erweitert. Die Datenpakete haben je nach Inhalt eine Länge von 1 bis 8 Bytes.

Das Prozessabbild für Messdaten und Statusinformationen wird über die SCPI-Schnittstelle frei konfiguriert. Mit den Werkseinstellungen werden folgende Botschaften und Inhalte im Motorola-Format (MSB) mit einer Baudrate von 500kBit gesendet:

```
can:msg 0x0F0, -2, 9000, 9001, 9002
can:msg 0x0F1, 10, 0, 1
can:msg 0x0F2, 10, 2, 3
can:msg 0x0F3, 10, 4, 5
can:msg 0x0F4, 10, 6, 7
can:msg 0x0F5, 100, 200, 201
can:msg 0x0F6, 100, 202, 203
can:msg 0x0F7, 100, 204, 205
can:msg 0x0F8, 100, 206, 207
can:msg 0x0FD, 1000, 8000, 8001, 8002, 8300, 8301, 8110
```

CAN-ID	CAN-ID (hex)	Param.-ID	Datentyp	Ausgabetak	Beschreibung	Einheit
240	0x0F0	9000	uint40	Startup + RTR	Serial Number 240062#####	
		9001	uint8		HW Version	
		9002	uint16		SW Version	· 0.01
241	0x0F1	0	float	10 ms	Input 1, PT ₁ -Filter	V, mA, X
		1	float		Input 2, PT ₁ -Filter	V, mA, X
242	0x0F2	2	float	10 ms	Input 3, PT ₁ -Filter	V, mA, X
		3	float		Input 4, PT ₁ -Filter	V, mA, X
243	0x0F3	4	float	10 ms	Input 5, PT ₁ -Filter	V, mA, X
		5	float		Input 6, PT ₁ -Filter	V, mA, X
244	0x0F4	6	float	10 ms	Input 7, PT ₁ -Filter	V, mA, X
		7	float		Input 8, PT ₁ -Filter	V, mA, X
245	0x0F5	200	float	100 ms	Input 1, RMS[300]	V, mA, X
		201	float		Input 2, RMS[300]	V, mA, X
246	0x0F6	202	float	100 ms	Input 3, RMS[300]	V, mA, X
		203	float		Input 4, RMS[300]	V, mA, X
247	0x0F7	204	float	100 ms	Input 5, RMS[300]	V, mA, X
		205	float		Input 6, RMS[300]	V, mA, X
248	0x0F8	206	float	100 ms	Input 7, RMS[300]	V, mA, X
		207	float		Input 8, RMS[300]	V, mA, X
253	0x0FD	8000	uint8	1000 ms	CPU0 load	%
		8001	uint8		CPU1 load	%
		8002	uint16		PCB Temperature	0.1 °C
		8300	uint8		Status SK-Filter, bool[]	
		8301	uint8		Current-Modes, bool[]	
		8110	uint16		Missing ADC cycles	

Kontakt

optiMEAS GmbH
 Am Houiller Platz 4
 64381 Friedrichsdorf

phone: +49 (6172) 997712-0
 e-mail: info@optimeas.de
 web: www.optimeas.com

