

SPI-Simulyzer



Hardware-Version	Ab 2.0
Software-Version	1.16.2
Technisches Datenblatt-Version:	1.5
Artikelnummer	1.0106
Erstellt:	(1.0) 03.05.2013
	(1.1) 04.08.2014 aux sample rates
	(1.2) 20.09.2015 Korrektur supply voltage externer Sensoren
	(1.3) 04.08.2018 HIL Test eingefügt
	(1.4) 18.11.2018 Artikelnummer/Betriebssystem update
	(1.5) 10.10.2021 Unternehmensinformationen bearbeitet
	(1.6) 07.11.2022 CAN-Interface hinzugefügt

Inhaltsverzeichnis:

1. Beschreibung.....	3
2. Hardware-Versionen.....	3
2.1 Single ended.....	3
2.2 LVDS.....	3
3. Betriebsarten.....	3
3.1 ECU-Mode.....	3
3.2 Sensor-Mode.....	3
3.3 Passiv-Mode.....	3
4. SPI-Bus-Parameter.....	4
5. Generierung der ECU-Daten in der Betriebsart ECU-Mode.....	4
6. Messdatenaufzeichnung.....	4
6.1 FPGA-Version.....	4
6.2 Sample Rates der Aux-Eingänge.....	5
7. Darstellung der Messdaten.....	5
7.1 Graphische Darstellung.....	5
7.2 Darstellung in Tabellenform.....	5
8. Stop der Messdaten-Aufzeichnung.....	5
9. Triggerung von Messdaten - Triggerevents.....	5
10. Export der Messdaten.....	6
11. Funktionsumfang.....	6
12. Anschlüsse.....	7
12.1 Interface und Digital Input/Output.....	8
12.2 Pin-Belegung Interface 1 und 2.....	9
12.3 Pin-Belegung Flachbandkabel-Adaption auf 25 pol-Sub-D-Buchse.....	9
12.4 Pin-Belegung 25-pol. Digital/Analog I/O.....	9
12.5 Pin-Belegung LVDS-Version.....	10
13. Verdrahtung.....	11
13.1 Verdrahtung Passiv-Mode.....	11
13.2 Verdrahtung ECU-Mode.....	11
13.3 Pin Verdrahtung Sensor-Mode.....	11
13.4 Verdrahtung für HIL Test.....	12
14. Technische Daten.....	13
15. Elektrische Daten.....	13
16. Umgebungsbedingungen.....	13
17. Software-Voraussetzungen.....	13
18. Weitere Informationsquellen und Tutorials.....	13

1. Beschreibung

Sensoren der Fahrzeugtechnik kommunizieren über einen SPI-Bus mit einem Steuergerät. Mit dem *SPI-Simulyzer* können Daten des SPI-Busses und der Peripheriegeräte schnell und unkompliziert ausgelesen und mittels Simulationsdaten aktiv in das Geschehen eingegriffen werden.

Der *SPI-Simulyzer* ist über ein USB-Kabel mit einer windowsbasierenden Software verbunden.

2. Simulyzer-Versionen

2.1. Single ended:

SPI-Simulyzer mit **2 Interface-Anschlüssen**

(im Lieferumfang eine Flachbandkabel-Adaption auf 25-pol. SUB-D-Stecker)

und einer **25-pol. SUB-D-Buchse** (im Lieferumfang ein leerer SUB-D-Gegenstecker).

2.2. LVDS-Version:

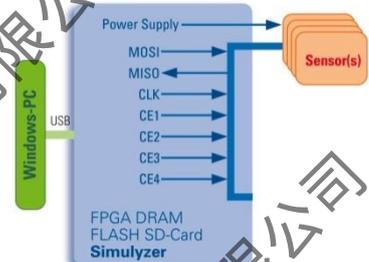
Differential-Signal-Übertragungs-Adaption.

(Adaption mit POD auf 25-pol. SUB-D-Stecker möglich)

3. Betriebsarten

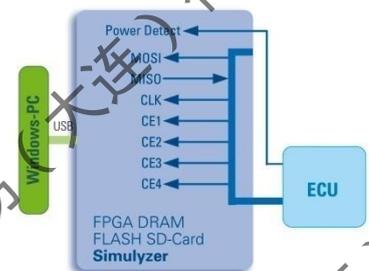
3.1. ECU-Mode

Der *SPI-Simulyzer* simuliert die ECU und versorgt die Sensoren mit Spannung. Die Datenkommunikation zwischen der simulierten ECU (SPI-Master) und den bis zu 4 Sensoren (SPI-Slaves) wird dargestellt und aufgezeichnet.



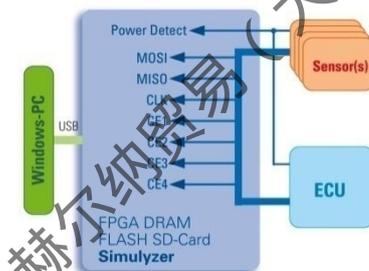
3.2. Sensor-Mode

Der SPI-Simulyzer simuliert bis zu 4 Sensoren durch Generieren entsprechender Daten. Die Datenkommunikation zwischen der ECU (SPI-Master) und den bis zu 4 simulierten Sensoren (SPI-Slaves) wird dargestellt und aufgezeichnet.



3.3. Passiv-Mode:

Die Datenkommunikation zwischen der ECU (SPI-Master) und den bis zu 4 Sensoren (SPI-Slaves) wird dargestellt und aufgezeichnet.



4. SPI-BUS-Parameter

Datenlänge	8 bis 64 Bits pro Datenpaket
SPI-Takt-Rate	10 Hz-20MHz
Baudrate	8 Mbit/s
Fehlererkennung	Parity/CRC
Phasen und Polarität der SPI-Daten/SPI Takt	Alle 4 Varianten: <u>1. Variante:</u> Phase Clock-Idle = 0 SPI-Daten mit 1. Flanke nach SS (High) <u>2. Variante:</u> Phase Clock-Idle = 0 SPI-Daten mit 2. Flanke nach SS (Low) <u>3. Variante:</u> Phase Clock-Idle = 1 SPI-Daten mit 1. Flanke nach SS (High) <u>4. Variante:</u> Phase Clock-Idle = 1 SPI-Daten mit 2. Flanke nach SS (Low)
Protokoll	Inframe communication Outframe communication
Versorgungsspannung der externen Sensoren (ECU-Mode)	2,5 V bis 20 V
High/Low-Spannungsschwelle der Digitalen Eingänge	2V
High/Low-Spannungslevel der Digitalen Ausgänge	CMOS-Ausgänge mit $V_{DD}=2,5\text{ V}$ bis 6 V

5. Generierung der ECU-Daten in der Betriebsart ECU-Mode

Die Datenwerte der ECU (Masterdaten) werden durch die Eingabe von Einzelbefehlen definiert, unterteilt in die einmalig zu durchlaufenden Initialisierungsdaten und die mehrfach abzuarbeitenden Daten des Messzyklusses (Schleifen-Definition). Die Befehle können direkt editiert oder als Text-Datei importiert werden. Die Einspielung von einzelnen Befehlen in den laufenden Messzyklus ist jederzeit möglich.

6. Messdatenaufzeichnung

Die SPI-Daten werden zusammen mit einem Zeitstempel als FPGA dekodierte Daten aufgezeichnet. Die Auflösung beträgt 1 μs . Die Daten werden in einem File mit bis zu 4 Gigabytes gespeichert. Das File ist als Ringpuffer organisiert (First in/Last out). Während der Datenaufzeichnung werden die Messdaten als sich ständig ändernde Werte dargestellt.

6.1. FPGA-Version

Die jeweilige FPGA-Version muss der jeweiligen Software-Version entsprechen.
 Historie der FPGA-Versionen:

FPGA-Version	Software-Version
0x31A	1.15.1
0x31B	1.15.2
0x31C	1.15.3
0x31C	1.15.4
0x31C	1.15.5
0x31C	1.15.6
0x31E	1.16.0
0x31E	1.16.1
0x320	1.16.2

6.2. Sample-Rates der Aux-Eingänge:

LVDS		Single ended	
1 Signal		1 Signal	Jedes Signal mit 100 k samples/s
2 Signale	1. Signal mit 50 k samples/s 2. Signal mit 50 k samples/s	2 Signale	Jedes Signal mit 100 k samples/s
3 Signale	1. Signal mit 50 k samples/s 2. und 3. Kanal mit 25 k samples/s	3 Signale	Jedes Signal mit 100 k samples/s
4 ignale	1. Signal mit 50 k samples/s 2., 3. und 4. Signal mit 16,67 k samples/s	4 Signale	Jedes Signal mit 100 k samples/s

7. Darstellung der Messdaten

7.1. Graphische Darstellung:

Die Daten der ECU wie auch die der Sensoren werden je als Analog-Signal und als Digital-Signal in unterschiedlichen, frei wählbaren Farben über eine Zeitachse dargestellt.

Triggerwerte sowie CRC bzw. Parity-Fehler werden hervorgehoben dargestellt.

Die vertikale Achse ist in LSB eingeteilt, kann jedoch in entsprechende physikalische Einheiten geändert werden. Der Zeitausschnitt und der Zoom der Darstellung kann individuell eingestellt werden.

7.2. Darstellung in Tabellenform

Die Messdaten werden zusätzlich in Tabellenform dargestellt. Dabei stellt jede Reihe einen Datensatz des SPI-Masters bzw. eines SPI-Slaves dar. In den Spalten werden der Zeitstempel, die Anzahl der übertragenen Bits, der Hexadezimalwert des übertragenen Frames von ECU und Sensoren sowie der extrahierte Messdatenwert dargestellt. Bei Auftreten eines Parity/CRC-Fehlers wird die entsprechende Zeile rot hervorgehoben dargestellt.

8. Stop der Messdaten-Aufzeichnung

Nach dem Start der Messdaten-Aufzeichnung kann der Messzyklus durch folgende Varianten gestoppt werden:

- Manuell - durch Klicken auf den Stop-Button der Software
- Nach einer definierten Zeit
- Nach einer definierten Anzahl von Durchläufen

9. Triggerung von Messdaten - Triggerevents

Durch die Definition von Triggerwerten, können gezielte Ereignisse (Triggerevents) dokumentiert und gespeichert werden.

Definition von Triggerwerten:

- Bei Erkennen eines Parity/CRC-Fehlers
- Datenwert-Schwelle: Über- oder Unterschreiten eines Datenwertes
- Externe Triggerspannung:
Erkennen einer Spannung am Trigger-Eingang (min. 2 V- max. 5 V)

Nach der Triggerung wird noch eine frei definierbare Anzahl von Daten aufgezeichnet. Die jeweiligen Triggerevents können in definierten Zeitabschnitten gespeichert werden.

Bei Triggerung wird ein Spannungs-Impuls am Trigger-Ausgang (CMOS-Ausgang mit $V_{dd}=5V$) erzeugt.

10. Export der Messdaten

Die aufgezeichneten Daten werden intern binär gespeichert und können als (*.txt) oder (*.csv)-File in unterschiedlichen Formaten (TCDM, hexadezimal, dezimal und als physikalische Größe) exportiert werden. Dabei ist eine Selektion einzelner Messdaten-Signale möglich.

11. Funktionsumfang

Folgender Funktionsumfang steht zur Verfügung:

- Visualisierung SPI-Bus-Kommunikation
- Aufzeichnung der Messdaten
- Stopp der Aufzeichnung durch definierte Werte
- Simulieren der ECU (Generierung der Masterdaten)
- Export/Import der Masterdaten
- Export/Import der Sensordaten
- Logik-Analyzer der Datenleitungen
- SPI-Bus-Parameter-Editor
- Tabellarische Auflistung der Messdaten
- Einzelbefehls-Editor
- Messdaten-File-Inspektor
- Definition von Trigger-Werten
- Analyse und Speicherung von Trigger-Ereignissen
- Export von Trigger-Ereignissen
- Manipulation der Ein-/Ausgänge
- Einspeisung und Triggerung von Referenzsignalen
- Definition von Projektdaten und deren Verwaltung.

12. Anschlüsse

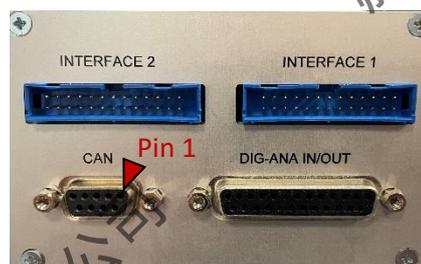


Achtung:
 Versorgungsspannung 12 V über externes Netzteil (1 Ampère)
 Masse der Versorgungsspannung ist mit Gehäuse verbunden!
 Masse der Kabelsignale müssen die gleiche Masse haben!

12.1. Optional CAN Interface

Optional kann bei Neugeräten ein zusätzliches CAN-Interface bereitgestellt werden (nicht nachträglich nachrüstbar).
 Option 2.2001 "Optionales CAN Interface – SPI-Simulyzer eingebaut in Art.Nr. 2.1001".

Pin	Beschreibung
1	nc
2	CAN low
3	GND
4	nc
5	nc
6	GND
7	CAN high
8	nc
9	nc



Hinweis: Abschluss-Widerstand zwischen CAN-high und CAN-low ist **nicht** eingebaut.

12.2. Interface und Digital Input/Output

Beiden gemein ist die Referenzspannungsquelle REF193, die eine Spannung von 3 Volt liefert. Im messenden und ausgebenden Analogzweig werden Widerstände mit einer Genauigkeit von 0,1% verwendet.

Die verwendeten OPs sind hochohmige OPs.

Elkos mit Leckströmen werden im Messkreis nicht verwendet.

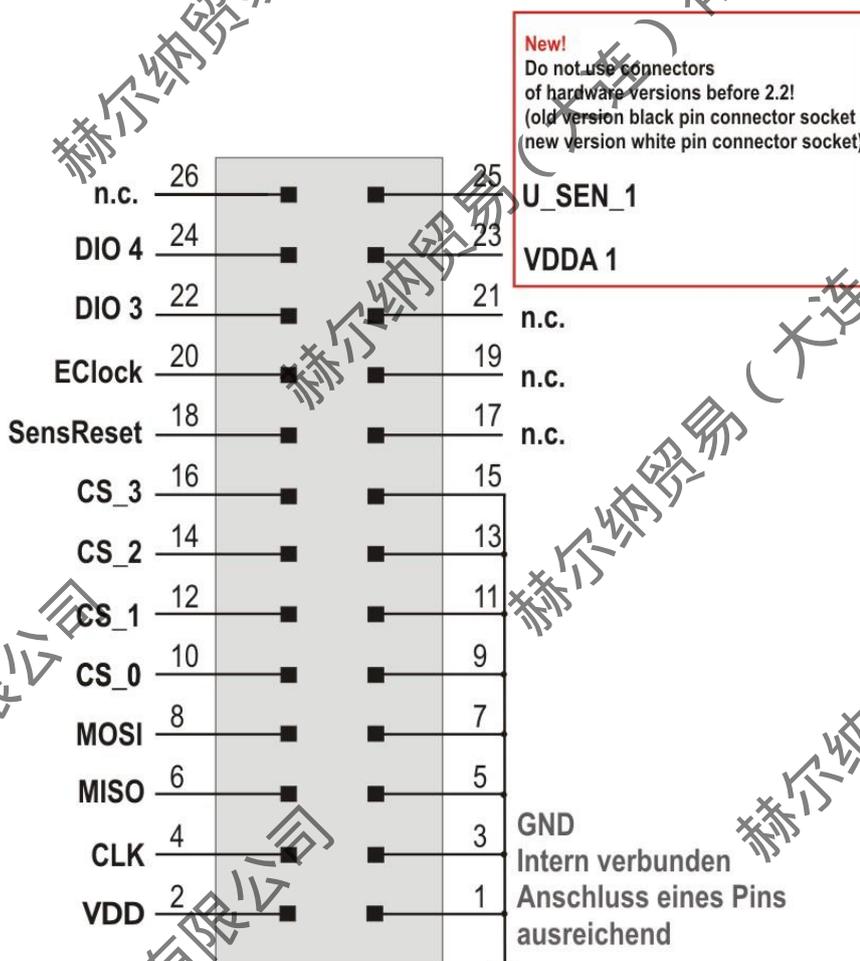
Die AD-Wandler besitzen eine Auflösung von 12 Bit. Die Messschaltung wurde so einfach wie möglich aufgebaut, um keine Addition der Toleranzen zu erhalten.

Die Erfahrung zeigt, dass die angegebene Genauigkeit von <0,5% deutlich unterschritten wird. Eine Temperaturkompensation findet nicht statt, da davon ausgegangen wurde, dass das Gerät unter Laborbedingungen verwendet wird.

Die analogen Ausgänge können zur Ausgabe der analogen Sensordaten verwendet werden. Es handelt sich dabei um logische OP-Ausgänge, die keine große Last treiben können. Um z.B. ein Magnetventil anzusteuern ist ein externer Leistungsverstärker erforderlich.

Über die analogen Ausgänge werden die SPI-Messungen ausgegeben. Sobald ein neuer Wert über SPI erscheint, wird der entsprechende analoge Ausgang aktualisiert.

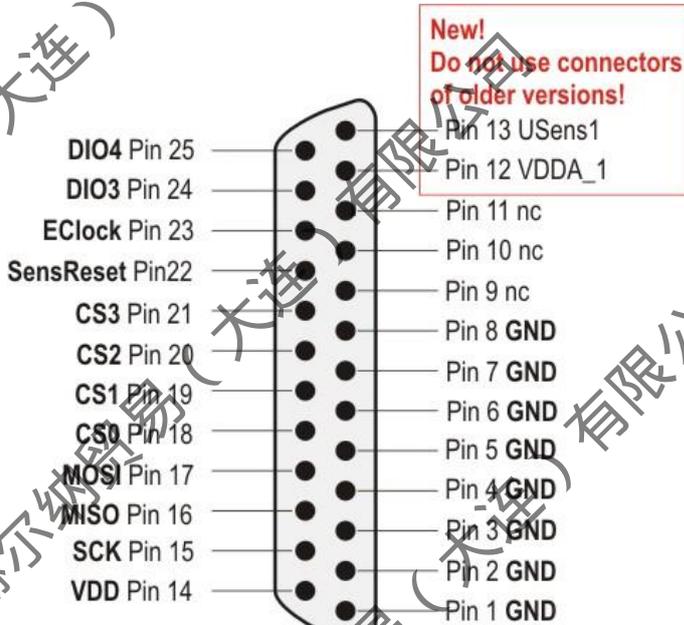
12.3. Pin-Belegung Interface 1 und 2



12.4. Pin-Belegung Flachbandkabel-Adaption auf 25 pol-Sub-D-Buchse - Single ended -

Der SPI-Simulyzer wird mit einem vorkonfektionierte Flachbandkabel mit 25-pol. Sub-D-Buchse ausgeliefert.

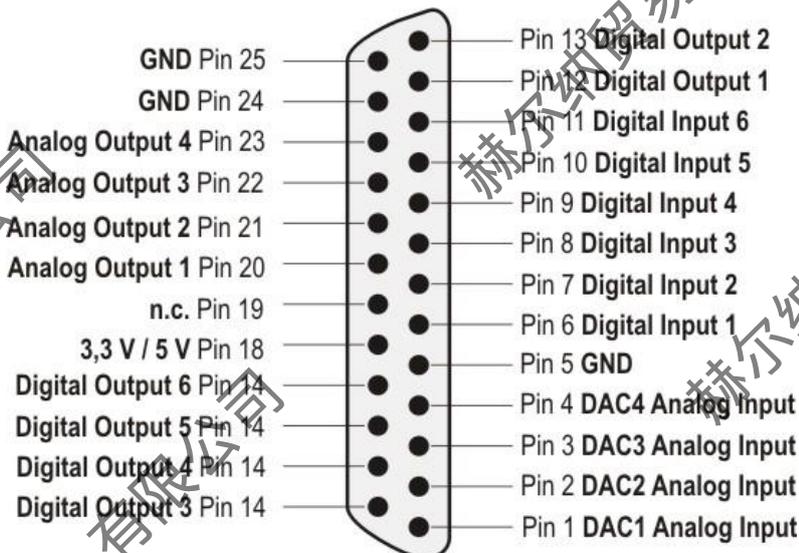
Pin-Belegung 25-pol Sub-D-Buchse:



25-pol. SUB-D-Buchse nach Flachbandkabel-Adaption

GND steckerintern verbunden
1 Anschluss-Pin genügt

12.5. Pin-Belegung 25-pol. Digital/Analog I/O - Buchse -



12.6. Pinbelegung LVDS-Version

Interface-Belegung

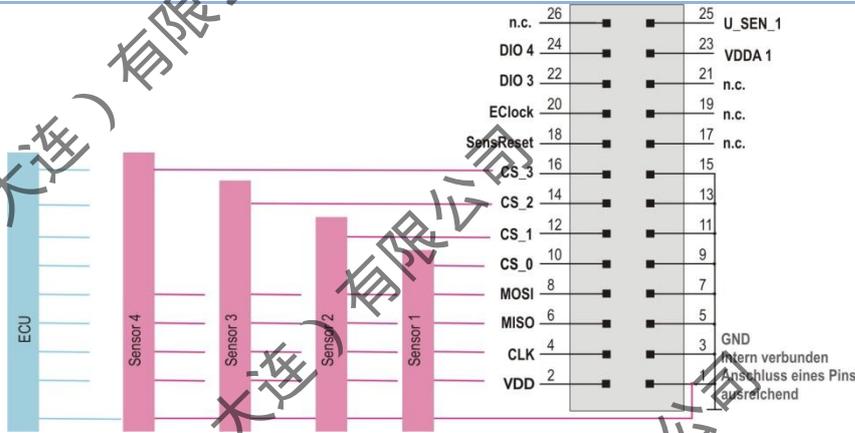


Pin	Signal
1	CS3-
2	CS3+
3	GND
4	CS2-
5	CS2+
6	ISO_MI_EXT
7	!OE_EXT
8	GND
9	CS1-
10	CS1+
11	GND
12	MOSI-
13	MOSI+
14	VDD
15	GND
16	+3,3V
17	CS0-
18	CS0+
19	GND
20	MISO-
21	MISO+
22	GND
23	CLK+
24	CLK-
C1	SensReset
C2	EClock
C3	DAC3
C4	DAC4
C5	GND

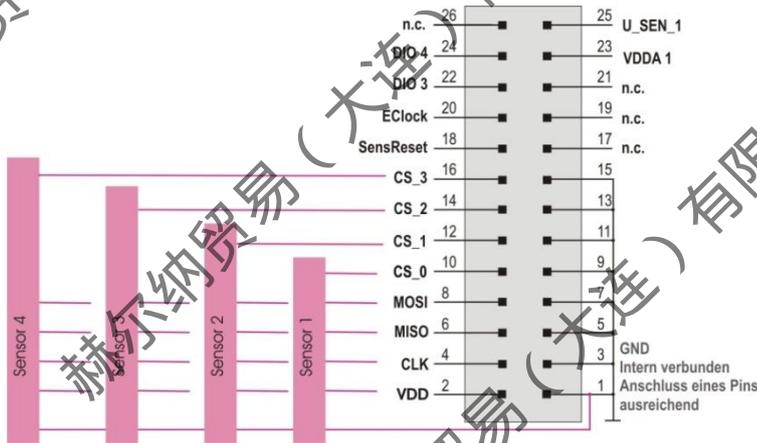
Die PIN-Belegung der 25-pol SUB-D-Buchse nach dem POD entspricht der PIN-Belegung der Single-ended-Version!

13. Verdrahtung

13.1. Verdrahtung Passive Mode



13.2. Verdrahtung ECU Mode



13.3. Verdrahtung Sensor Mode



13.4. Verdrahtung für HIL Test

Im Folgenden wird ein Beispiel zur Verschaltung der SPI Signale eines Steuergerätes mit dem SPI-Simulyzer zur Verwendung von HIL Tests beschrieben.

Hinweis: Ein SPI-Simulyzer kann maximal 4 SPI Sensoren simulieren.

Diese 4 SPI Sensoren können auf einem oder beiden Interfaces verteilt sein.

Es existieren Implementierungen von Sensor-Modellen verschiedener Sensoren.

Die Sensor Modelle müssen individuell mit der SesKion GmbH abgeklärt werden und

eventuell bei SesKion neu erstellt werden. Dazu muss anhand der Sensor-Datenblättern der Funktionsumfang des Sensormodells definieren werden.

Sensormodelle müssen immer lizenziert werden!

Im folgenden Beispiel sind im Steuergerät (ECU) 4 SPI-Sensoren an den Micro-Controller angeschlossen.

Die Sensoren A, B und C über eine gemeinsame SPI Schnittstelle via Chip-Select Leitungen sowie ein weiterer

SPI-Sensoren über eine zweite SPI Schnittstelle via einer Chip-Select Leitungen.

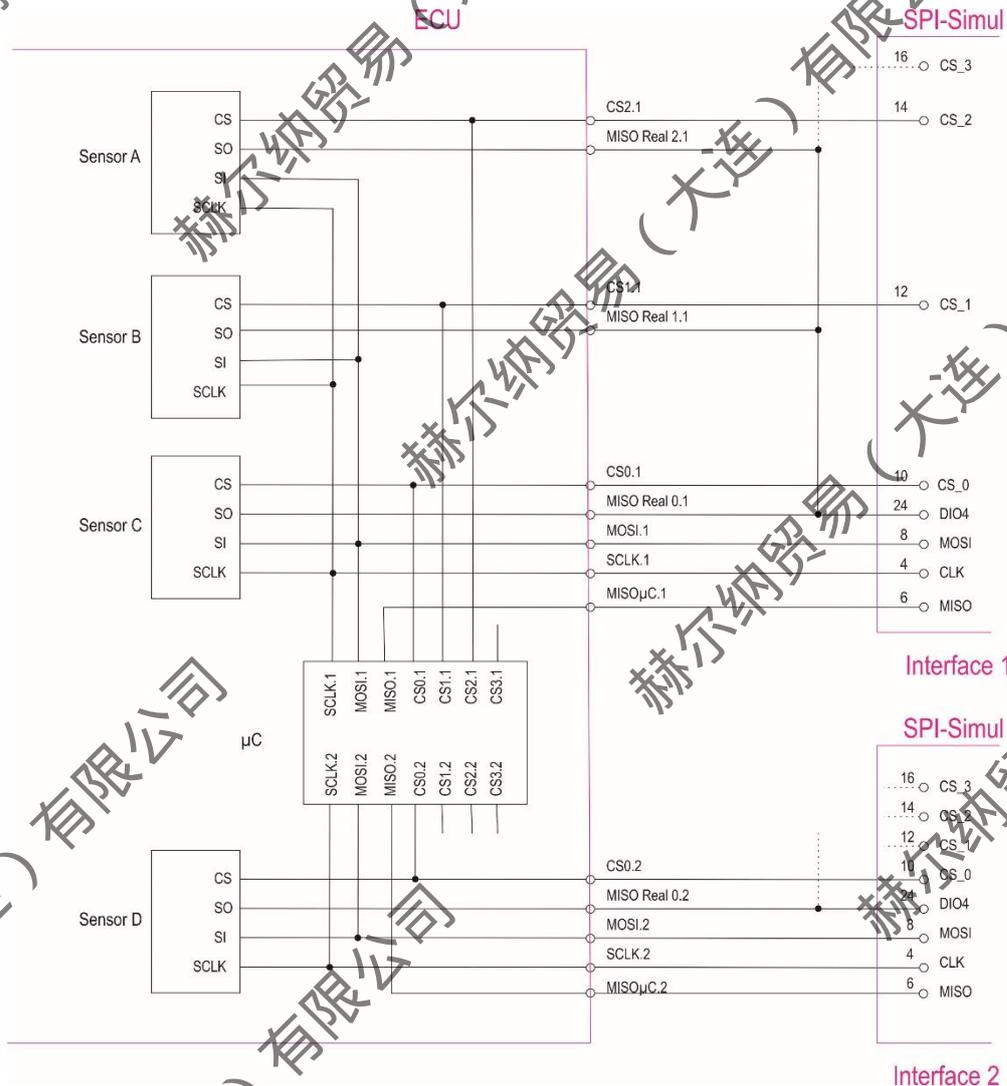
Die Signale $SCLK.i$, $MOSI.i$, $CS0.i$, $CS1.i$ und $CS2.i$ werden parallel zur internen Steuergeräte Verbindung auf einem

Test Stecker herausgeführt. Die ursprüngliche Verbindung zwischen $MISO.i$ und den SO Leitungen der Sensoren

muss dazu aufgetrennt werden und als $MISO\mu C.i$ bzw. $MISOReal.i$ am Test Stecker herausgeführt werden.

Die $MISOReal.i$ werden außerhalb des Steuergerätes verbunden und am $DIO4$ Anschluss des jeweiligen Interface des SPI-Simulyzers angeschlossen.

Sollen einzelne oder alle internen Sensoren der Steuergeräte ohne Simulation betrieben werden, so müssen deren $MISOReal.i$ Signale mit dem $MISO\mu C.i$ Signal direkt verbunden werden!



14. Technische Daten

Abmessungen	165 x 105 x 55 mm
Gehäuse-Material	Alu
Gewicht	1 kg
Betriebstemperatur-Bereich	0°C ... 40°C
Beschleunigung	Max. 3g
Rel. Luftfeuchtigkeit	Max. 85% nicht kondensierend

15. Elektrische Daten

Stromaufnahme über USB	Typ. 500 mA
Versorgungsspannung für die externen Sensoren im ECU-Mode	0 V ... 5,5 V
Intern	
Dynam. RAM	128 MB
FPGA Soft μ C Core	32 bit Soft-Core

ACHTUNG:

Das Anlegen von externen Spannungen über 6 V kann den SPI-Simulyzer zerstören!

16. Umgebungsbedingungen

Der SPI-Simulyzer ist einzig für den Einsatz unter Laborbedingungen geeignet!

17. Software Systemvoraussetzungen

- Operating system: Windows 7 oder 8, XP, 10 (32 oder 64 bit)
- Microsoft .Net 2.0 framework. Wird bei der Installation kostenfrei mitinstalliert.
- Arbeitsspeicher: min. 1GB RAM
- Festplattenspeicher von 100 MB bis 4GB freie Kapazität (Grenzdaten für Daten FIFOs)
- Prozessor: min. 1.5 GHz
- USB 2.0 High-speed Interface
- Bildschirmauflösung: min. 1024x768 Pixel, opt. 1280x720 Pixel

18. Weitere Informationsquellen und Tutorials

Seskion GmbH
Karlsruher Straße 11/1
D-70771 Leinfelden-Echterdingen
Telefon: +49 (711) 990 58 14
Fax: +49 (711) 990 58 27
Email: info@seskion.de
URL: <http://www.seskion.de>