

# Bedienungsanleitung



**RS232-ADC16/24**

**RS232 ADC 16/24: Version 1.03d**

Copyright © 2021 taskit GmbH

# **taskit GmbH**

**Groß-Berliner Damm 37**

**D 12487 Berlin**

**Telefon +49 (0) 30 611 295 – 0**

**Fax +49 (0) 30 611 295 – 10**

**[www.taskit.de](http://www.taskit.de)**

Alle Rechte an dieser Dokumentation und an dem hierin beschriebenen Produkt verbleiben bei

**taskit GmbH.**

Bei der Erstellung der Dokumentation wurde mit Sorgfalt vorgegangen. Selbstverständlich können Fehler trotzdem nicht vollständig ausgeschlossen werden, so dass weder die o.a. Firma noch der Vertreiber für fehlerhafte Angaben, daraus resultierende Fehlfunktion oder deren Folgen eine juristische Verantwortung oder irgendeine Haftung übernehmen. Waren-, Marken- und Firmennamen werden ohne Gewährleistung der freien Verwendbarkeit benutzt.

Kein Teil davon darf ohne ihre schriftliche Genehmigung in irgendeiner Form reproduziert werden.

## Inhaltsverzeichnis

1	Eigenschaften.....	4
2	Einleitung.....	4
3	Inbetriebnahme.....	5
4	Applikationsbeispiel.....	6
4.1	Sicherheitshinweis.....	6
5	Befehlsformat (frame layout).....	7
5.1	LRC Kalkulation.....	8
6	Kommandos.....	9
6.1	Holding Register lesen.....	9
6.2	Input Register lesen.....	9
6.3	Single Register schreiben.....	10
6.4	Multiple Register schreiben.....	10
6.5	Neukalibrierung.....	11
6.6	Kalibrierung speichern.....	11
6.7	Fehlermeldungen.....	12
7	Beschreibung der Register.....	13
7.1	Holding Register.....	13
7.2	Input Register (analoge Eingänge).....	17
8	Technische Details.....	18
8.1	RS232 Interface.....	18
8.2	DSUB-25 Signalstecker.....	18
8.3	Elektrische Daten.....	19
8.4	Reaktionszeit.....	20
8.5	Stromaufnahme.....	21
8.6	Baudratenabweichung.....	22
8.7	Analog/Digital-Wandler.....	22
9	Konformitätserklärung.....	24
9.1	Konformitätserklärung für Europa.....	24
9.2	Konformität nach EMV-Richtlinie 2014/30/EU.....	24

# RS232-ADC 16/24

## 1 Eigenschaften

- Microcontroller C8051F350/C8051F352, Taktfrequenz 3 – 49 MHz
- Einfache Kommunikation mittels RS232-Interface
- Acht analoge Eingänge mit 16bit oder 24bit Auflösung, max. 2.5 Volt
- Acht digitale 3.3V-I/O (5V tolerant)
- Separate Masse-Anschlüsse für Analog und Digital
- Betriebsspannungsbereich 4.5V – 15V (max. 20V)
- Herausgeführte 3.3V Versorgung (z.B. für Sensoren)
- Erweiterter Temperaturbereich (-40°C – 85°C)
- Kleiner Befehlssatz (Modbus)
- Jederzeit rekompilierbar
- C-Library verfügbar

## 2 Einleitung

Das RS232-ADC16/24 ermöglicht den einfachen Zugriff auf externe Sensoren und Peripheriegeräte über eine Standard-RS232-Verbindung. Bei Verwendung von Windows-, DOS- oder Linux-Systemen sind keine zusätzlichen Treiber erforderlich. Mikrocontroller-Systeme benötigen nur eine einfache Implementierung ihrer UART-Schnittstelle.

An das Gerät gesendete Anweisungen haben eine für Menschen lesbare Form, die eine manuelle Prüfung des Datenverkehrs für Debugging ermöglicht. Um das RS232-ADC16/24 in Ihre Anwendung zu integrieren, wird eine C-Bibliothek bereitgestellt, welche die Arbeit der Protokollanalyse und Paketkonstruktion übernimmt.

Das Gerät ist in zwei Versionen erhältlich, eine mit 16-Bit- und eine mit 24-Bit-ADC-Kanälen. Diese Versionen werden jeweils nach ihrer ADC-Kanalaufösung RS232-ADC16 und RS232-ADC24 benannt.

Dieses Dokument enthält Informationen zur Hardware, dem Softwareprotokoll und den daraus resultierenden Registerzuordnungen. Es enthält zusätzliche Informationen für die neuen Firmware-Versionen 1.11 und 1.12.

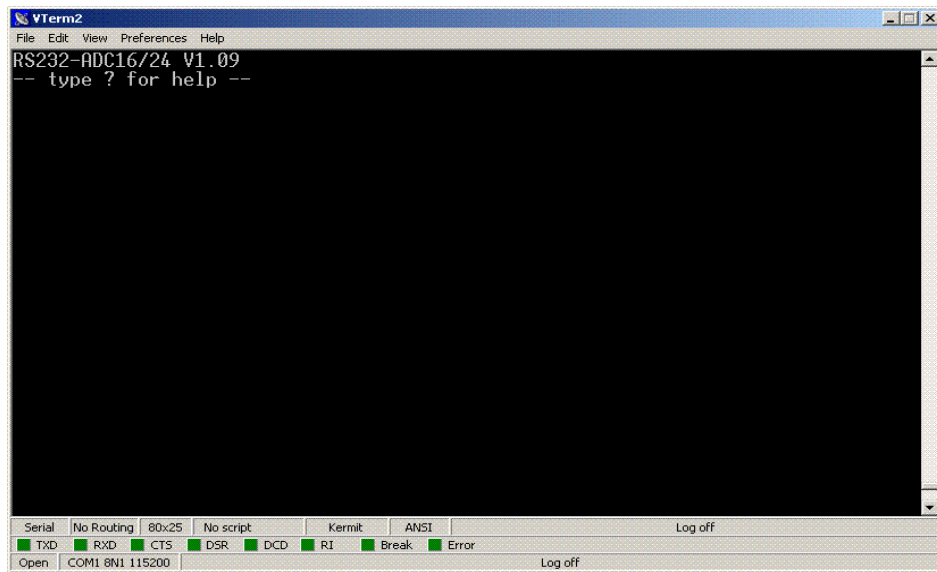
# RS232-ADC 16/24

## 3 Inbetriebnahme

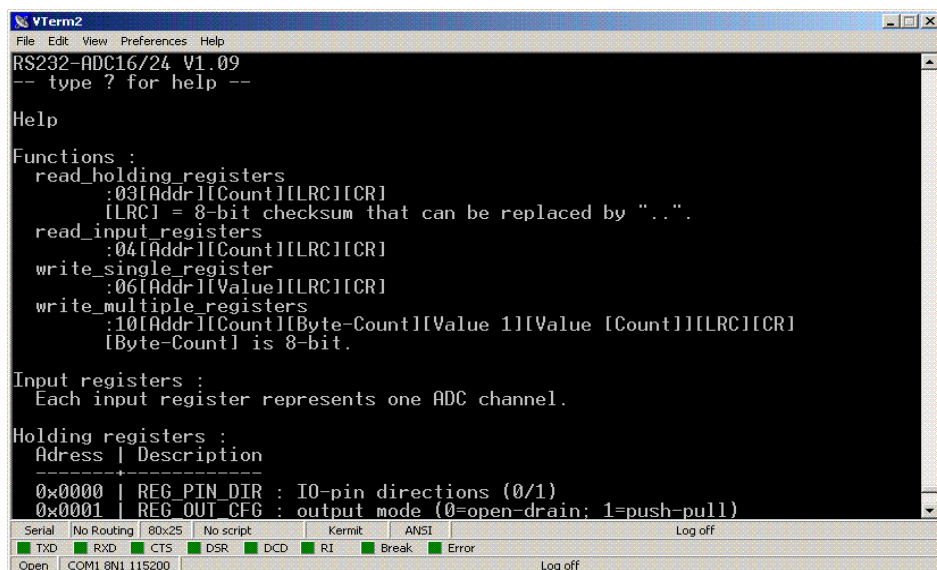
Die folgenden Schritte veranschaulichen die korrekte Einrichtung des Geräts.

1. Die DSUB-9 Buchse mit einer RS232-Schnittstelle verbinden
2. Den Serialport wie folgt konfigurieren und ein Terminalprogramm starten: 115200 Baud, 8N1 (8 bit data, no parity, 1 stop bit).
3. Stromversorgung über den DSUB-25 Stecker herstellen: Minus/Masse an Pin 18 (GND) und Plus an Pin 24 (VIN, max.15V)

Darauf hin sollte folgende Bootmeldung des RS232-ADC zu sehen sein.



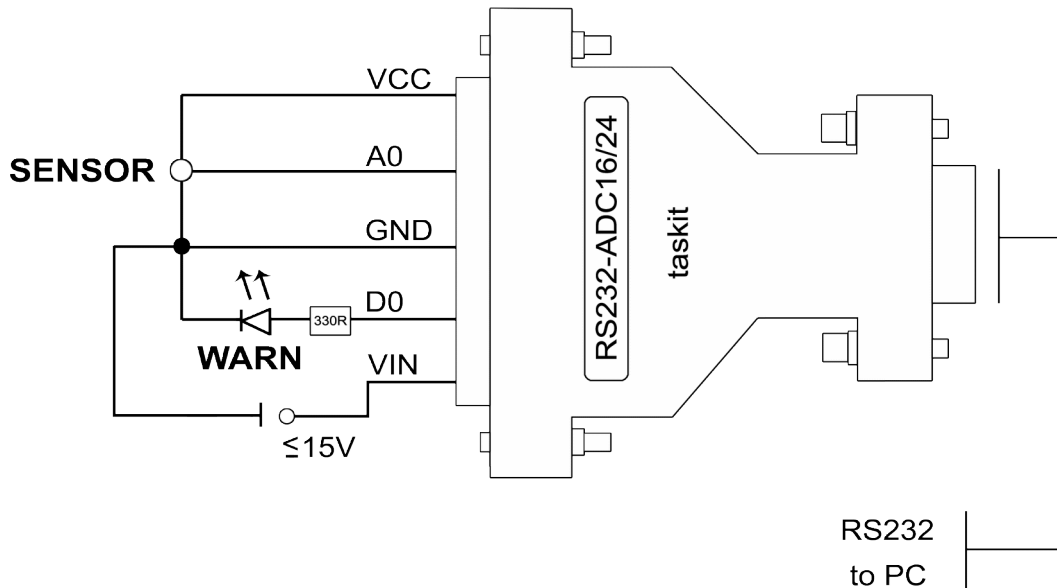
Wird ein Fragezeichen eingegeben bzw. gesendet, wird ein Hilfetext ausgegeben



## RS232-ADC 16/24

### 4 Applikationsbeispiel

In der folgenden Skizze wird die Verwendung von RS232-ADC16/24, insbesondere in Bezug auf VCC und VIN dargestellt.



Die Aufgabe besteht darin, die Sensordaten abzurufen und an eine Überwachungssoftware zu übertragen. Bei kritischen oder ungewöhnlichen Bedingungen könnte eine LED in Sichtweite den Status anzeigen. Wie man sieht, können bis zu acht Paare von Sensor und LED mit einem RS232-ADC16/24 gesteuert werden.

Wie bereits beschrieben, wird das Gerät durch eine Spannungsversorgung mit maximal 15 V betrieben. Der interne Spannungsregler des Geräts liefert 3,3 V an VCC, um externe Sensoren mit Strom zu versorgen. Demnach sollte VCC **niemals** zur Stromversorgung des RS232-ADC16/24 verwendet werden. Weiterhin ist der Bereich des Analogeingangs (0 - 2,5V) zu berücksichtigen.

#### 4.1 Sicherheitshinweis

Um Beschädigungen am RS232-ADC16 bzw. RS232-ADC24 und/oder den angeschlossenen Komponenten zu vermeiden, dürfen die Betriebsspannungen aller miteinander verbundenen Komponenten erst angeschlossen/eingeschaltet werden, wenn alle sonstigen Verbindungen am RS232-ADCxx hergestellt wurden (kein "hot-plugging").

## 5 Befehlsformat (frame layout)

Dieses Kapitel gibt den verwendeten Formatrahmen (im folgenden als Frame bezeichnet) an. Ein Frame enthält nicht mehr als einen Befehl. Jeder Befehl wird zusammen mit allen seinen Parametern gesendet.

Parameter, die größer als acht Bit sind, werden im Big-Endian-Format erwartet, was bedeutet, dass das höchstwertige Byte (MSB) vor dem niedrigstwertigen Byte (LSB) gesendet wird.

Auf der seriellen Leitung wird jedes Byte mit acht Bits, einem Startbit und einem Stoppbit gesendet. In der Kommunikation sind keine Paritätsbits enthalten. Die Standard-Baudrate beträgt 115200 Baud. Im Moment wird kein Handshake verwendet, da der Host auf eine Antwort warten soll, bevor er eine neue Anfrage sendet.

Ein Frame wird in einem für Menschen lesbaren ASCII-Format codiert. Daher wird jedes Byte der Frame-Daten in zwei HEX-Ziffern konvertiert. Der konvertierte Frame besteht dann aus einem Startzeichen, dem Befehl und den Parametern in HEX-Ziffern, einer Paket-LRC - ebenfalls in HEX-Ziffern - und einem Endzeichen.

Byte <sub>1</sub>	Byte <sub>2</sub>	Byte <sub>3</sub>	Byte <sub>4</sub>	Byte <sub>5</sub>	...	Byte <sub>n-2</sub>	Byte <sub>n-1</sub>	Byte <sub>n</sub>
:	Command		Parameter <sub>1</sub>			LRC		<CR>

Das Endzeichen ist ein "Carriage Return" (<CR>, '\r'), der zum Hex-Code 0x0D führt. Jedes zusätzliche "New Line" (<NL>, '\n', 0x0A) wird vom Gerät ignoriert und zu jeder Antwort hinzugefügt.

**Beispiel:** Lese zwei ADC-Kanäle, beginnend bei 0x0001

Abfrage :

:0400010002..  
<CR><NL>

Antwort :

:0404XXXXYYYY..  
<CR><NL>

**Hinweis:** Der erste Parameter ist die Byte-Anzahl der gelesenen Daten, XXXX der Wert von Kanal1, JJJJ der Wert von Kanal2. LRC-Checksummen können in Abfragen vermieden werden, indem LRC auf ".." gesetzt wird. Antworten haben **immer** eine LRC, obwohl sie in diesem Beispiel auf ".." gesetzt ist.

## RS232-ADC 16/24

### 5.1 LRC Kalkulation

Die Paket-LRC ist das Zweierkomplement der Summe über die Parameter- und dem Kommando-Byte (command) des Pakets.

$LRC = -(\sum data_n) + command$ ; mit  $data_n$  als **binärem** Byte-Wert der Packet-parameterliste und dem Kommando als **Binärwert**.

Der korrekte LRC für die obige Beispielabfrage wäre 0xF8 gewesen.

Die Einbeziehung der LRC in die Berechnung eines empfangenen Pakets führt zu einer Summe gleich 0, andernfalls ist ein Übertragungsfehler aufgetreten.

**Hinweis:** Jedes Binärdatenbyte wird als zwei HEX-Ziffern übertragen, die **vor** dem Hinzufügen in eine Binärform umgewandelt werden müssen.



## 6 Kommandos

Das Gerät verwendet einen MODBUS-ähnlichen Befehlssatz, bei dem alle Parameter als 16-Bit-Werte angenommen werden.

Die folgenden Befehle werden unterstützt.

### 6.1 Holding Register lesen

Ein zusammenhängender Block von 16-Bit "Holding Register" kann sofort gelesen werden. Die Adressierung beginnt bei 0, die Anzahl ist größer als 0 und kleiner oder gleich 125.

Kommando : Holding Register lesen		
Funktionscode	1 Byte	0x03
Startadresse	2 Bytes	0x0000 bis 0xFFFF
Registeranzahl	2 Bytes	0x0000 bis 0x007D

Wenn die Funktion erfolgreich ist, befindet sich die Antwort in den Leseregistern.

Antwort : Holding Registers lesen		
Funktionscode	1 Byte	0x03
Byte-Anzahl	1 Byte	(Registeranzahl) * 2
Registerdaten	(Registeranzahl) * 2 Bytes	

Andernfalls wird eine Fehlerantwort gesendet.

### 6.2 Input Register lesen

Diese Funktion entspricht dem Lesen der "Holding Register", bezieht sich aber auf die Eingangsregister.

Kommando : Eingangsregister lesen		
Funktionscode	1 Byte	0x04
Startadresse	2 Bytes	0x0000 bis 0xFFFF
Registeranzahl	2 Bytes	0x0000 bis 0x007D

Wenn die Funktion erfolgreich ist, befindet sich die Antwort in den Leseregistern.

## RS232-ADC 16/24

Antwort : Eingangsregister lesen		
Funktionscode	1 Byte	0x04
Byte-Anzahl	1 Byte	(Registeranzahl) * 2
Registerwert	(Registeranzahl) * 2 Bytes	

Andernfalls wird eine Fehlerantwort gesendet.

### 6.3 Single Register schreiben

Mit dieser Funktion können einzelne „Holding Register“ geschrieben werden. Dem Funktionscode folgen eine 16-Bit-Registeradresse und ein 16-Bit-Registerwert.

Kommando/Antwort : Single Register schreiben		
Funktionscode	1 Byte	0x06
Registeradresse	2 Bytes	0x0000 bis 0xFFFF
Registerwert	2 Bytes	

Wenn die Funktion erfolgreich ist, entspricht die Antwort dem empfangenen Befehl, andernfalls wird eine Fehlerantwort gesendet.

### 6.4 Multiple Register schreiben

Wenn mehr als ein „Holding Register“ geschrieben werden muss, kann diese Funktion verwendet werden. Ein Block von Registerwerten folgt dem Funktionscode, der (Start-) Adresse und der Registeranzahl. Bis zu 123 Register können gleichzeitig geschrieben werden.

Kommando : Multiple Register schreiben		
Funktionscode	1 Byte	0x10
Registeradresse	2 Bytes	0x0000 bis 0xFFFF
Registeranzahl	2 Bytes	0x0001 bis 0x007B
Byte-Anzahl	1 Byte	(Registeranzahl) * 2
Registerwerte	(Registeranzahl) * 2 Bytes	

Wenn die Funktion erfolgreich ist, wird die entsprechende Antwort gesendet.

## RS232-ADC 16/24

Antwort : Multiple Register schreiben		
Funktionscode	1 Byte	0x10
Registeradresse	2 Bytes	
Registeranzahl	2 Bytes	

Andernfalls wird eine Fehlerantwort gesendet.

### 6.5 Neukalibrierung

Dieser Befehl kalibriert das Gerät unter Zuhilfenahme einer externen Spannung. Die analogen Kanäle 0 und 1 müssen während der Kalibrierung auf die folgenden Werte eingestellt werden.

Kanal 0 muss mit analoger Masse verbunden sein.

Kanal 1 muss mit dem höchsten erwarteten gültigen Pegel verbunden sein.

**Hinweis:** Das Gerät wird kalibriert ausgeliefert

Kommando/Antwort : Analogteil kalibrieren		
Funktionscode	1Byte	0x64

Wenn die Funktion erfolgreich ist, entspricht die Antwort dem empfangenen Befehl, andernfalls wird eine Fehlerantwort gesendet.

### 6.6 Kalibrierung speichern

Die Kalibrierungsdaten werden beim Zurücksetzen des Systems geladen. Wenn standardmäßig andere Kalibrierungsdaten benötigt werden, verwenden Sie diesen Befehl nach der Gerätekalibrierung (siehe oben). Zum Entsperren dieses Vorgangs wird ein Schlüssel benötigt.

Seit Firmware Version 1.09 speichert diese Funktion auch die Einstellung der Baudrate, um das Booten mit einer benutzerdefinierten Baudrate zu ermöglichen.

**Hinweis:** Durch das Speichern der Kalibrierungsdaten werden die Werkseinstellungen überschrieben! Verwenden Sie diese Funktion nicht, falls Sie *versehentlich* „Gerät kalibrieren“ aufgerufen haben.

Kommando/Antwort : Kalibrierungsdaten speichern		
Funktionscode	1 Byte	0x65
Schlüsselwort	2 Bytes	0xABCD

Wenn die Funktion erfolgreich ist, entspricht die Antwort dem empfangenen Befehl, andernfalls wird eine Fehlerantwort gesendet.

## RS232-ADC 16/24

### 6.7 Fehlermeldungen

Jede Befehlssequenz, die zu einem Fehlerzustand führt, antwortet mit einer Zwei-Byte-Antwort, die den angegebenen Funktionscode und einen zusätzlichen Fehlercode enthält. Die Anzeige erfolgt durch Setzen des höchsten Bits des Funktionscodes.

Fehlermeldung		
Indikator	1 Byte	0x80   (Funktionscode)
Fehlercode	1 Byte	

Fehlercode	Erklärung
1	Illegaler Funktionsaufruf
2	Adresse außerhalb des zulässigen Bereichs
3	Daten inkonsistent

# RS232-ADC 16/24

## 7 Beschreibung der Register

Die Konfiguration des Geräts erfolgt über lesbare und beschreibbare "Holding Register". Auch die PIO-Daten sind in diesen Registern verfügbar. ADC-Kanäle werden als schreibgeschützte Eingangsregister angezeigt.

### 7.1 Holding Register

Registerübersicht : Holding Register		
Adresse	Name	Beschreibung
0x0000	REG_PIN_DIR	Datenrichtung für die I/O-Pins konfigurieren
0x0001	REG_OUT_CFG	Output-Modus für die I/O-Pins konfigurieren
0x0002	REG_OUT_VAL	Pegel der Output-Pins setzen
0x0003	REG_IN_VAL	Pegel der Input-Pins auslesen
0x0004	REG_VERSION	Firmware version <sup>1</sup>
0x000D	REG_ADC_DEC	ADC-Takt (für alle Kanäle) festlegen
0x000E	REG_BAUD	Einstellung der Baudrate
0x000F	REG_SYSCLK	Systemtakt einstellen

<sup>1</sup> verfügbar ab Version 1.11.

#### REG\_PIN\_DIR 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
-	-	-	-	-	-	-	-	Dir <sub>7</sub>	Dir <sub>6</sub>	Dir <sub>5</sub>	Dir <sub>4</sub>	Dir <sub>3</sub>	Dir <sub>2</sub>	Dir <sub>1</sub>	Dir <sub>0</sub>

Dir<sub>n</sub> wählt zwischen Pin-Eingang und Pin-Ausgang. Ein gesetztes Bit konfiguriert den entsprechenden Pin D<sub>n</sub> als Ausgang. Die Eingangspins sind intern auf ‚HIGH‘ gezogen. Der Standardwert dieses Registers ist 0x0000.

#### REG\_OUT\_CFG 0x0001

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
-	-	-	-	-	-	-	-	OM <sub>7</sub>	OM <sub>6</sub>	OM <sub>5</sub>	OM <sub>4</sub>	OM <sub>3</sub>	OM <sub>2</sub>	OM <sub>1</sub>	OM <sub>0</sub>

OM<sub>n</sub> programmiert seinen entsprechenden Pin D<sub>n</sub> entweder in den Push-Pull- oder den Open-Drain-Modus, wenn D<sub>n</sub> als Ausgangs-Pin konfiguriert ist. Eine logische Null wählt den Open-Drain-Modus. Änderungen an den Eingangspins werden ignoriert. Der Standardwert dieses Registers ist 0x0000.

## RS232-ADC 16/24

### REG\_OUT\_VAL 0x0002

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
-	-	-	-	-	-	-	-	Out <sub>7</sub>	Out <sub>6</sub>	Out <sub>5</sub>	Out <sub>4</sub>	Out <sub>3</sub>	Out <sub>2</sub>	Out <sub>1</sub>	Out <sub>0</sub>

Out<sub>n</sub> legt den Pin-Pegel des entsprechenden Pin D<sub>n</sub> fest. Änderungen an den Eingangspins werden ignoriert. Der Standardwert dieses Registers ist 0x00FF.

### REG\_IN\_VAL 0x0003

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
-	-	-	-	-	-	-	-	In <sub>7</sub>	In <sub>6</sub>	In <sub>5</sub>	In <sub>4</sub>	In <sub>3</sub>	In <sub>2</sub>	In <sub>1</sub>	In <sub>0</sub>

Eingangspegel an Pin D<sub>n</sub> erscheinen im Bit In<sub>n</sub> wenn D<sub>n</sub> als Eingang programmiert ist. Andernfalls kann der tatsächliche Ausgangspegel von D<sub>n</sub> zurück gelesen werden. Alle Schreibvorgänge in dieses Register werden ignoriert.

### REG\_VERSION 0x0004

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Major Version								Minor Version							

Seit der Firmware-Version 1.11 ist dieses Register schreibgeschützt und enthält die Versionsnummer des Geräts. Frühere Versionen enthalten dort eine ,0', solange dieses Register nicht überschrieben wird.

### REG\_ADC\_DEC 0x000D

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADC-Zyklen (ADC-Cycles)															

Wählt die Anzahl der ADC-Zyklen aus, mit denen der Pegel eines ADC-Eingangs gemessen wird. Das Erhöhen dieses Werts führt zu genaueren Messwerten, verringert jedoch die maximale Ausgangsfrequenz des ADC. Akzeptierte Werte liegen zwischen 5 und 15.<sup>1</sup> Jede Registereinstellung, die diese Grenzen überschreitet, führt zu 11, was die Standard- und Grundeinstellung dieses Registers ist.<sup>2</sup>

Die Ausgangsfrequenz wird folgendermaßen berechnet:

$$f_{\text{output}} = \text{clock}_{\text{SYS}} / 3 / 128 / 2^{\text{ADC cycles}} \cdot 3$$

## RS232-ADC 16/24

**Hinweis:** Die Ausgangsfrequenz wird natürlich auch durch die ausgewählte Baudrate begrenzt. Bei der maximal möglichen Systemtaktfrequenz ( $\text{clock}_{\text{SYS}}$ ) von 49 MHz beträgt der vom ADC verwendete Takt 24,5 MHz.

<sup>1</sup> Firmware-Version 1.12 erweitert den Bereich der akzeptierten Werte auf 5 bis 15, für frühere Versionen beträgt dieser Bereich 5 bis 11.

<sup>2</sup> Gilt für Geräte mit mindestens Firmware-Version 1.11; bei älteren Versionen ist der Standardwert 5.

<sup>3</sup> Bei Firmware-Versionen älter als 1.11 wird folgende Formel verwendet:

$$f_{\text{output}} = \text{clock}_{\text{ADC}} / 3 / 2^{\text{ADC cycles}}, \text{ with } \text{clock}_{\text{ADC}} = 19200 \text{ Hz.}$$

### REG\_BAUD 0x000E

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Baudrate wählen															

Wählt die Baudrate aus, die für die serielle Übertragung verwendet wird. Die Baudrate kann in fünf Schritten zwischen 9600 und 115200 Baud gewählt werden. Die Standard-Baudrate ist 115200. Jede Registereinstellung über 4 wird als 4 interpretiert. Nach erfolgreicher Befehlsausführung wird die Startnachricht mit der neuen Baudrate gesendet. Die Antwort auf den Befehl wird jedoch mit den alten Einstellungen gesendet.

REG_BAUD	Baudrate
0	9600
1	19200
2	38400
3	57600
4	115200

## RS232-ADC 16/24

### REG\_SYSCLK 0x000F

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Systemtaktfrequenz (clock <sub>sys</sub> )															

Der Systemtakt kann in fünf Schritten von 3,0625 MHz bis 49 MHz gewählt werden. Jede Registereinstellung außerhalb des Bereichs führt zu einem Standardtakt mit 12,25 MHz. Das Ändern der Systemtaktes führt zu einer kurzen Verzögerung vor dem Senden der Startnachricht. Alle Pin- und Registereinstellungen bleiben vom Taktwechsel unberührt. Wenn Sie Takt und Baudrate zusammen ändern (indem Sie mehrere Register schreiben), wird nur eine Startnachricht gesendet, nicht zwei. Es wird empfohlen, auf den Neustart des Geräts zu warten, indem Sie versuchen, ein ‚Holding Register‘ zu lesen.

Die Verwendung von 12,25 MHz als Systemtakt wird empfohlen, da dies den besten Kompromiss zwischen ADC-Ausgaberate und Genauigkeit herstellt. Das Erhöhen von clock<sub>sys</sub> bietet eine bessere Reaktionszeit und höhere Ausgaberraten auf Kosten der Genauigkeit. Durch Verringern von clock<sub>sys</sub> wird hauptsächlich der Stromverbrauch gesenkt.

REG_SYSCLK	Systemtakt in MHz
0	3.0625
1	6.125
2	12.25
3	24.5
4	49

**Hinweis:** Wenn Sie REG\_SYSCLK auf 0 setzen, wird die maximale Baudrate auf 57600 reduziert.



## RS232-ADC 16/24

### 7.2 Input Register (analoge Eingänge)

Registerübersicht : Input Register		
Adresse	Name	Beschreibung
0x0000	REG_ADC0_VAL	16-bit Wert gemessen an A <sub>0</sub>
0x0001	REG_ADC1_VAL	16-bit Wert gemessen an A <sub>1</sub>
0x0002	REG_ADC2_VAL	16-bit Wert gemessen an A <sub>2</sub>
0x0003	REG_ADC3_VAL	16-bit Wert gemessen an A <sub>3</sub>
0x0004	REG_ADC4_VAL	16-bit Wert gemessen an A <sub>4</sub>
0x0005	REG_ADC5_VAL	16-bit Wert gemessen an A <sub>5</sub>
0x0006	REG_ADC6_VAL	16-bit Wert gemessen an A <sub>6</sub>
0x0007	REG_ADC7_VAL	16-bit Wert gemessen an A <sub>7</sub>

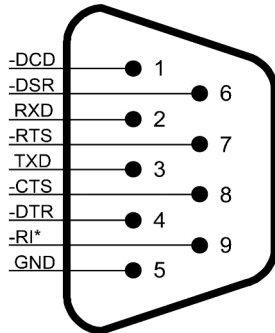
Eine Messung bei A<sub>n</sub> wird jedes Mal gestartet, wenn "Eingangsregister lesen" mit der Adresse n aufgerufen wird. Wenn mehrere Register gelesen werden, wird jeder betroffene ADC-Kanal nacheinander behandelt. Die Funktion blockiert, bis alle Messungen abgeschlossen sind, bevor das Ergebnis zurückgesendet wird.

Der Zugriff auf die unteren acht Bits einer 24-Bit-Konvertierung ist durch Lesen der Eingangsregister 0x0008 bis 0x000F (mit den Namen REG\_ADC0\_LOW bis REG\_ADC7\_LOW) möglich, die jeweils das niedrige Byte des zuletzt bei A<sub>0</sub> bis A<sub>7</sub> gemessenen 24-Bit-Werts enthalten. Bei RS232-ADC16-Modulen werden diese Register immer mit 0 gelesen.

$$\text{adc\_value}_{24} = (\text{REG\_ADC}_x\_\text{VAL} \text{ shl } 8) + \text{REG\_ADC}_x\_\text{LOW}$$

## 8 Technische Details

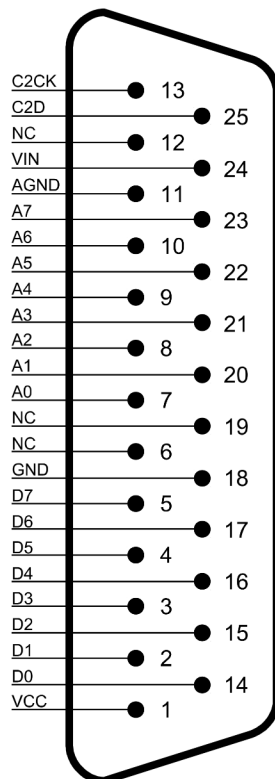
### 8.1 RS232 Interface



Die Verbindung zwischen Host und RS232-ADC16/24-Gerät wird über eine 9-polige D-SUB-Buchse hergestellt. Das Pin-Layout stimmt mit der Standard-RS232 Pin-Zuordnung überein, mit Ausnahme von -RI\*, das entweder zur Erkennung des Geräts oder als alternativer Stromversorgungs-Pin des Geräts verwendet werden kann. Daher ist -RI\* intern mit VIN (Spannungsversorgung) verbunden. Wenn der -RI\*-Pin mit dem Host verbunden ist, ist die Eingangsspannung der Stromversorgung des Geräts aufgrund der RS232-Spezifikation auf maximal 15 V begrenzt.

**Hinweis:** Handshake wird derzeit nicht verwendet.

### 8.2 DSUB-25 Signalstecker



Analoge Sensoren und digitale Peripheriegeräte sind am 25-poligen D-SUB-Stecker anzuschließen. Zusätzlich zu den acht digitalen I/Os D0 bis D7 und den acht analogen Eingängen A0 bis A7 bietet dieser Anschluss eine stabilisierte 3,3-V-Stromversorgung für angeschlossene Sensoren und Peripheriegeräte, sodass der Spannungsregler des RS232-ADC16/24-Geräts zur Stromversorgung dieser externen Komponenten verwendet werden kann und keine zusätzlichen Spannungsregler benötigt werden.

Die Pins 6, 12 und 19 werden derzeit nicht verwendet, sind jedoch für mögliche Erweiterungen reserviert und sollten nicht angeschlossen werden.

Die Anschlüsse C2D und C2CK ermöglichen die Programmierung des verwendeten C8051F35x-Mikrocontrollers.

AGND ist die Bezugsmasse für A0 bis A7.

**Hinweis:** Die Stromversorgung über VIN darf 15 V nicht überschreiten, wenn der -RI\* Pin des RS232-Steckers mit einem RS232-Gerät verbunden ist.

## RS232-ADC 16/24

### 8.3 Elektrische Daten

Digital I/O		
	Min	Max
Ausgangspegel „HIGH“	2.5 V	3.3 V
Ausgangspegel „LOW“	-	1.0 V
Eingangspegel „High“	2.0 V	5.0 V
Eingangspegel „LOW“	-	0.8 V
Maximaler Ausgangsstrom <sup>1</sup>	-	75 mA
Analogeingänge		
	Min	Max
Eingangsspannung <sup>2</sup>	0 V	2.5 V (analog max.) 3.6 V (absolut max.)
Auflösung <sup>3</sup>	-	16/24 Bit
Betriebsdaten		
	Min	Max
Betriebsspannung (VIN, -RI*)	4.5 V	15 V (absolut max. 20V)
Ausgangsspannung (VCC)	-	3.3 V
Baudrate	9600	115200
Interner Systemtakt	~3 MHz	49 MHz

<sup>1</sup> Der dauerhafte Stromverbrauch (einschließlich des Geräts selbst) darf 100 mA nicht überschreiten.

<sup>2</sup> Obwohl die digitalen I/O 5V-tolerant sind, darf die Spannung an den analogen Eingängen 3,6V nicht überschreiten - Spannungen oberhalb 2,5 V sind nicht zu unterscheiden.

<sup>3</sup> Die Auflösung hängt vom Gerätetyp ab (RS232-ADC16 oder RS232-ADC24).

## RS232-ADC 16/24

### 8.4 Reaktionszeit

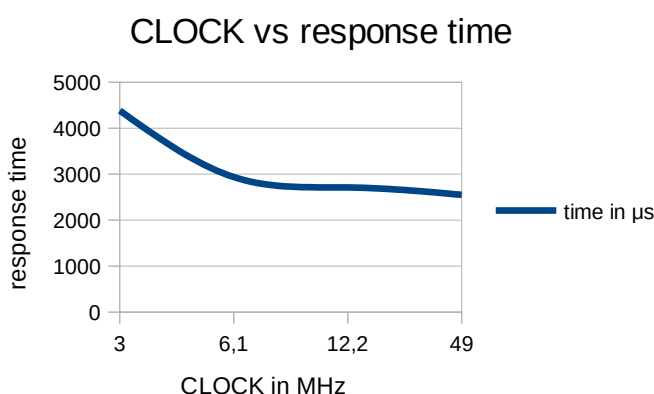
Die Reaktionszeit hängt von zwei Dingen ab - der Baudrate und natürlich dem Systemtakt. Jede Anfrage muss empfangen und dekodiert werden, bevor eine Aktion ausgeführt werden kann. Die Gesamtzeit setzt sich aus der Übertragung und der Decodierung zusammen. Je höher der Systemtakt ist, desto entscheidender ist die Übertragungsrate. Je niedriger die Übertragungsrate ist, desto unbedeutender ist der Systemtakt.

Beachten Sie, dass der Übertragungs- und Dekodierungsaufwand bei genaueren ADC-Messungen nicht der entscheidende Zeitfaktor ist, sondern die ADC-Zyklen.

Die folgenden Daten betrachten ein Szenario, in dem die Zeit zwischen dem Senden eines Befehls (z. B. Umschalten des Pegels eines Pins) und dem Ende der Ausführung (Pin hat sich auf diesen Pegel geändert) gerade erreicht wird.

Reaktionszeit	
$f_{\text{SYSCLK}}$ in MHz	Zeit in $\mu\text{s}$
6.125	4383
12.25	2937
24.5	2711
49	2551

Diese Daten werden bei 115200 Baud ermittelt, ohne die Zeit für die Antwort (auf die Abfrage) zu berücksichtigen.



Die Aufgabe bestand darin, den Pegel von  $D_0$  auf LOW zu stellen. Die Zeitmessung begann beim Senden der Anforderung und wurde gestoppt, als  $D_0$  durch Interrupt ausgelöst wurde. Die Länge der Anfrage betrug 14 Bytes. Demnach würde die für die reine Übertragung benötigte Zeit 1215  $\mu\text{s}$  bei 115200 Baud betragen.

# RS232-ADC 16/24

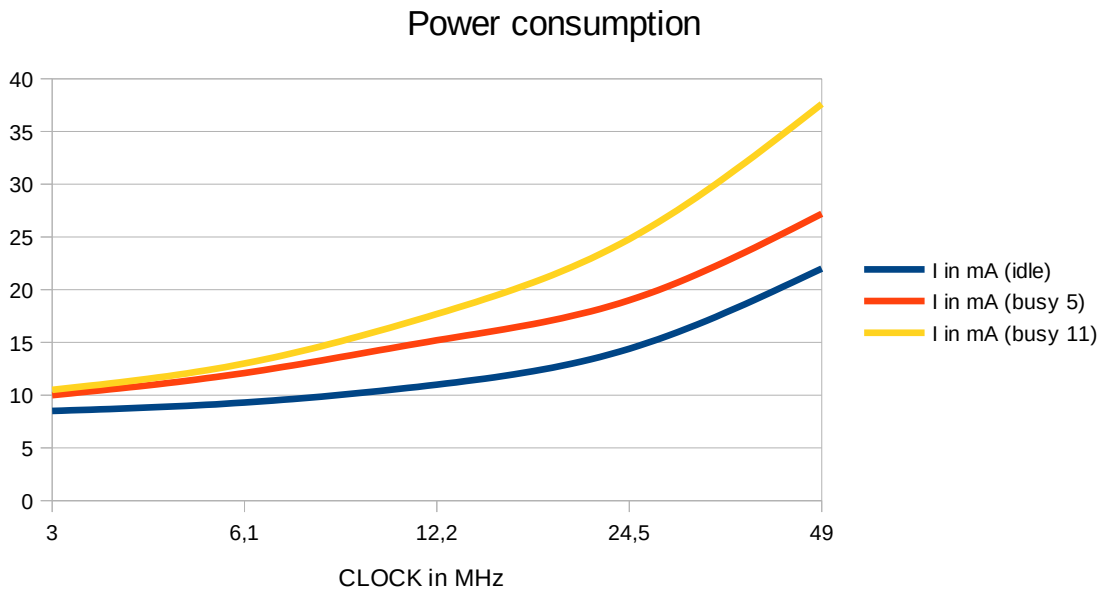
## 8.5 Stromaufnahme

Der Stromverbrauch hängt von den SYSCLK-Einstellungen, der ADC-Dezimierungsrate (ADC-Cycles) und den Aktivitätszyklen (busy cycle) des Geräts ab. Wenn keine Befehle verarbeitet werden müssen, wird der Leerlaufmodus (Idle) aufgerufen, was zu einem geringeren Stromverbrauch führt.

Das Unterbrechen der RS232-Verbindung führt zu niedrigeren Werten.

Stromaufnahme			
$f_{SYSCLK}$ MHz	Idle mA	Busy (REG_ADC_DEC = 5) mA	Busy (REG_ADC_DEC = 11) mA
3.0625	8.5	10.0	10.5
6.125	9.3	12.1	13.0
12.25	11.0	15.2	17.7
24.5	14.4	19.0	24.8
49	22.0	27.2	37.6

Die Daten wurden mit  $V_{in} = 5,0V$  an einem RS232-ADC16-Gerät erfasst.



## RS232-ADC 16/24

### 8.6 Baudratenabweichung

Da SYSCLK nicht durch Standard-Baudraten teilbar ist, treten beim Einstellen der Übertragungsgeschwindigkeit Fehler auf. Die Tabelle zeigt die Abweichung für den Standardwert gemäß SYSCLK.

Baud rate errors			
$f_{\text{SYSCLK}}$ MHz	Gewählte Baudrate	Erzielte Baudrate	Abweichung in %
3.0625	57600	56179	2.47
6.125	115200	114285	0.79
12.25	115200	114942	0.22
24.5	115200	117647	2.12
49	115200	113636	1.36

### 8.7 Analog/Digital-Wandler

Die Daten der folgenden Tabellen wurden mit einem RS232-ADC16-Gerät mit Firmware-Version 1.11 bei verschiedenen Takt- und Dezimierungsraten erfasst.

Präzision und Ausgaberate (in Hz)							
REG_ADC_DEC	I in mA	$f_{\text{output}}$ Hz	Min mV	Max mV	Noise mV	Noise %	Bit
5	19.0	261	-4.73	5.53	10.26	0.4104	7.93
6	19.8	231	-1.03	1.11	2.14	0.0856	10.19
7	20.1	188	-0.21	0.23	0.44	0.0176	12.47
8	22.1	136	-0.15	0.11	0.26	0.0140	13.23
9	23.3	88	-0.12	0.08	0.20	0.0080	13.61
10	24.2	52	-0.08	0.08	0.16	0.0064	13.93
11	24.8	29	-0.08	0.04	0.12	0.0048	14.35

SYSCLK = 24.5 Mhz

## RS232-ADC 16/24

Da die Firmware ab Version V1.11 ADC-Ausgabegeräten bietet, die mit SYSCLK skaliert werden können, kann der Einfluss von SYSCLK auf die Ausgaberate und -genauigkeit mitberücksichtigt werden.

REG_ADC_DEC = 11							
SYSCLK MHz	I in mA	f <sub>output</sub> Hz	Min mV	Max mV	Noise mV	Noise %	Bit
3	10.5	4	-0.04	0.03	0.07	0.0028	15.12
6.1	13.0	7.5	-0.04	0.07	0.11	0.0044	14.47
12.2	17.7	15	-0.04	0.04	0.08	0.0032	14.93
24.5	24.8	29	-0.08	0.04	0.12	0.0048	14.35
49	37.6	29	-0.07	0.08	0.15	0.0060	14.02

REG_ADC_DEC = 5							
SYSCLK MHz	I in mA	f <sub>output</sub> Hz	Min mV	Max mV	Noise mV	Noise %	Bit
3	10.0	76	-4.85	4.84	9.69	0.3876	8.01
6.1	12.1	148	-5.23	4.84	10.07	0.4028	7.96
12.2	15.2	218	-4.81	4.65	9.46	0.3784	8.05
24.5	19.0	231	-4.69	5.38	10.07	0.4028	7.96
49	27.2	261	-4.70	5.53	10.23	0.4092	7.93

## RS232-ADC 16/24

### 9 Konformitätserklärung

#### 9.1 Konformitätserklärung für Europa

Für folgende Produkte:

**Allgemeiner Produktname: RS232-ADC 16/24**

Art.-Nr.: 542869: RS232-ADC 16 im erw. Temperaturbereich

Art.-Nr.: 542880: RS232-ADC 24 im erw. Temperaturbereich

Wir erklären, dass sie den folgenden Richtlinien der Europäischen Gemeinschaft entsprechen:

**Konformität nach EMV-Richtlinie 2014/30/EU**

**European Conformity According to EMC-Directive 2014/30/EU**

#### 9.2 Konformität nach EMV-Richtlinie 2014/30/EU

Die Produkte, auf die sich diese Erklärung bezieht, entsprechen den folgenden Normen oder normativen Dokumenten der elektromagnetischen Verträglichkeit.

Sicherheit /  
Gesundheit:

EN 62368-1:2016-05  
EN 62479:2011-09

EMV:

EN 301 489-1 V2.1.1:2017-02  
EN 301 489-17 V3.1.1:2017-02

RoHS:

EU-Richtlinie 2011/65/EU

Aufgrund des in der Richtlinie 2014/30/EU beschriebenen Konformitätsbewertungsverfahrens sollte die Endkundenausrüstung wie folgt gekennzeichnet werden:



**Die Endkundenausrüstung muss den tatsächlichen Sicherheits- / Gesundheitsanforderungen nach den gen. Normen entsprechen.**