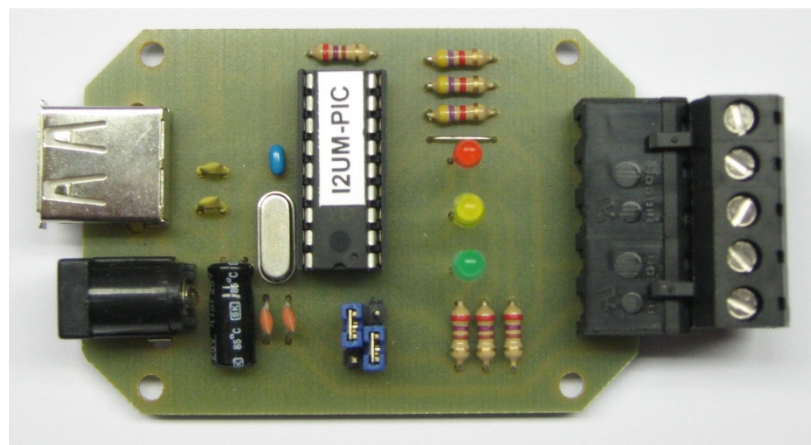


Inhalt

1. Allgemein	2
1.1. Kommunikationsparameter.....	2
1.2. Protokoll.....	2
1.2.1. Befehlsframe.....	2
1.2.2. Antwortframe	3
1.3. Fehlernummern.....	3
2. Befehle.....	4
2.1. Befehle der Gruppe 1 = INFO.....	4
2.1.1. VERSION 11 hex = 17 dez.	4
2.1.2. MODEM-CALL 12 hex =18 dez.....	4
2.2. Befehle der Gruppe 2 = KONFIG.....	5
2.2.1. PULLUP 21 hex = 33 dez.....	5
2.2.1.1. Pullup´s auslesen	5
2.2.1.2. Pullup´s einschalten	5
2.2.1.3. Pullup´s ausschalten	5
2.2.2. I2C-SPEED 22 hex = 34 dez.....	6
2.2.2.1. Speed setzen	7
2.2.2.2. Speed abfragen.....	7
2.3. Befehle der Gruppe 3 = I2C.....	8
2.3.1. I2C-SET 31 hex = 49 dez.....	8
2.3.2. I2C-GET 32 hex = 50 dez.	8
2.3.3. I2C-DATA 33 hex = 51 dez.	9
2.3.3.1. Daten lesen	9
2.3.3.2. Daten schreiben	9



1. Allgemein

1.1. Kommunikationsparameter

Das I²C-USB-Modem empfängt und sendet die Kommandos über einen virtuellen COM-Port mit folgenden Einstellungen:

- Baudrate: 115200
- Parity: none
- Zeichenbits: 8
- Stopbit: 1

Nachdem das Modem ein Kommando vollständig empfangen hat wird es auf den I²C-Bus angewendet. Anschließend generiert es ein Antwort-Byte welches zum PC zurückgeschickt wird.

1.2. Protokoll

Das Modem wird über ein Protokoll gesteuert. Zu einem vollständigen Protokoll (im weiteren Frame genannt) gehören:

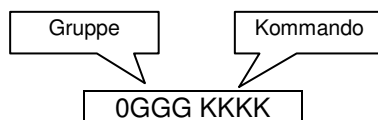
- der Befehl
- die Anzahl der Bytes im Datenblock (maximal 128)
- der Datenblock selbst und
- das Ende-Zeichen (EOT= END OF TRANSMISSION).

1.2.1. Befehlsframe

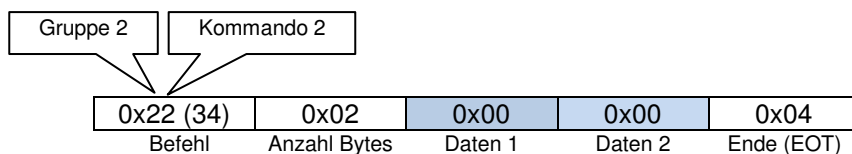
Die Befehle für das I²C-USB-Modem wurden in logische Gruppen zusammengefasst.

- (1) Info in dieser Gruppe sind alle allgemeinen Anfragen an das Modem zusammengefasst.
- (2) Konfig alle Befehle, die sich auf das Modem selbst beziehen
- (3) I²C alle Befehle, die sich auf den I²C-Bus beziehen

Die Gruppennummer steht beim Befehl im oberen Nibbel, das Kommando im unteren Nibbel.



Beispiel für einen Befehls-Frame mit 2 Byte Daten:



Info:

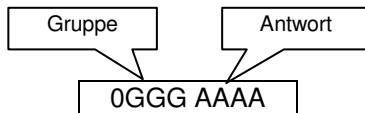
Zum besseren Verständnis wird in dieser Beschreibung der Datenblock immer mit einem hellblauen Hintergrund dargestellt.

Bei den angegebenen Zahlen handelt es sich um hexadezimale Werte im Format 0x_ _ . Bei größeren Hex-Zahlen wurde der Dezimalwert in Klammern dahinter geschrieben.

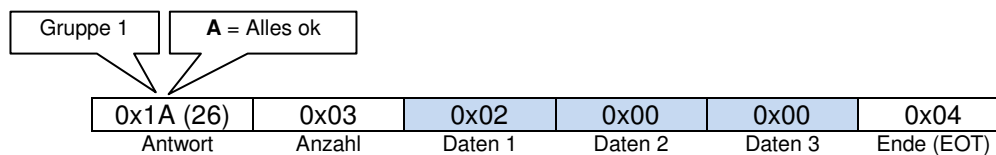
1.2.2. Antwortframe

Wurde ein gesendeter Befehl vom Modem gesendet kommt eine Antwort über den virtuellen COM-Port zum PC zurück.

Im Antwort-Byte ist im oberen Nibbel wieder die Gruppennummer vom übertragenen Befehl angegeben. Hat das Modem den Befehl verstanden und konnte ihn erfolgreich ausführen steht im unteren Nibbel des Antwort-Bytes ein Hexadezimal „A“ für „Alles ok“

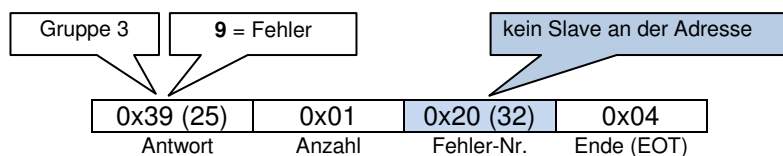


Beispiel für einen Antwort-Frame mit drei Byte Daten:



Im Fehlerfall steht im oberen Nibbel auch die Gruppennummer – im unteren Nibbel jedoch des Antwort-Bytes eine „9“.

Beispiel für einen Antwort-Frame mit FEHLER-Nummer:



1.3. Fehlernummern

In dieser Liste finden Sie eine Aufstellung der möglichen Fehlernummern:

Fehlernummer	Beschreibung
0x01 (1)	alles OK kein Fehler
0x02 (2)	Die Gruppenadresse (oberes Nibbel) ist unbekannt
0x03 (3)	Das Kommando (unteres Nibbel) ist unbekannt
0x04 (4)	Die Datenblocklänge wurde nicht gesendet, oder ist falsch
0x05 (5)	Die Länge des Datenblocks ist zu groß
0x06 (6)	END OF TRANSMISSION (04) fehlt am Ende des Frame
0x07 (7)	Als EOT wurde nicht 04 gesendet
0x08 (8)	Zeitüberschreitung beim Senden des Datenblocks
0x10 (16)	Version Kommando ist falsch aufgebaut. Der Framekopf ist zwar richtig aber Rest stimmt nicht!
0x11 (17)	Zu viele Daten beim Befehl Call-Modem
0x20 (32)	Auf die gesendete Adresse reagiert kein Slave
0x21 (33)	Der Slave hat auf das Acknolw. nicht reagiert
0x22 (34)	Wenn der Slave einen Clock- Stretch auslöst, wartet das Modem maximal 1,5 Sekunden. Danach erfolgt ein Timeout.
0xFF (255)	Unbekanntes Kommando!

2. Befehle

2.1. Befehle der Gruppe 1 = INFO

In dieser Gruppe sind alle Befehle zusammen gefasst die der allgemeinen Information dienen.

2.1.1. VERSION 11 hex = 17 dez.

Mit diesem Befehl lässt sich die aktuelle Firmware-Version ermitteln. Dazu wird ein Frame ohne Daten (Anzahl = 0) das Modem gesendet.

Befehl:

0x11 (17)	0x00	0x04
Befehl	Anzahl	Ende (EOT)

Antwort:

0x1A (26)	0x03	0x02	0x00	0x00	0x04
Antwort	Anzahl	Version	NK1	NK2	Ende (EOT)

Daten mit der Versions-Nummer

Der Antwort-Frame enthält 3 Byte Nutzdaten, welche die Version der Firmware enthalten. Dabei steht das erste Byte im Datenblock für die Versionsnummer und die beiden folgenden für die Unternummer. In unserem Beispiel ist das 2 0 0 und somit die Version 2.00

2.1.2. MODEM-CALL 12 hex =18 dez.

Dieser Befehl hat die Aufgabe das Modem „anzustoßen“. Somit lässt sich ganz einfach feststellen ob das Modem noch mit dem PC verbunden ist oder ob es Probleme mit der USB-Verbindung gibt.

Befehl:

0x0C (12)	0x00	0x04
Befehl	Anzahl	Ende (EOT)

Antwort:

0x1A (26)	0x01	0x23 (35)	0x04
Antwort	Anzahl	#	Ende (EOT)

= Modem ok

Das Modem sendet ein Prompt. In diesem Fall ist es ein „#“.

2.2. Befehle der Gruppe 2 = KONFIG

In der Gruppe sind die Befehle zur Konfiguration des I2C-USB-Mosdems.

2.2.1. PULLUP 21 hex = 33 dez.

Der I2C-Bus braucht an einer Stelle im Netz eine Terminierung nach +5V. Dies wird in der Regel mit Widerständen gemacht. Bei der Kommunikation ziehen die Busteilnehmer diese High-Pegel auf Masse. Es gibt Anwendungen wo nicht am Master terminiert werden soll. Dann müssen die Pull-Up-Widerstände abgeschaltet werden können.

2.2.1.1. Pullup´s auslesen

Mit diesem Befehl kann der aktuelle Status der Pullup-Widerstände ausgelesen werden.

Befehl:

0x21 (33)	0x00	0x04
Befehl	Anzahl	Ende (EOT)

128 = PullUp Widerstände EIN
0 = PullUp Widerstände AUS

Antwort:

0x2A (42)	0x01	0x80 (128)	0x04
Antwort	Anzahl	EIN	Ende (EOT)

Das Modem sendet einen Wert zurück, der den Zustand der Pullup´s repräsentiert. Ist der Wert 0x80 (128) sind die Pullup´s eingeschaltet. Bei 0 sind die Pullup´s stromlos.

2.2.1.2. Pullup´s einschalten

Wird im Datenblock eine „1“ gesendet, schaltet das Modem die Pullup Widerstände EIN.

Befehl:

0x21 (33)	0x01	0x01	0x04
Befehl	Anzahl	EIN	Ende (EOT)

1 = PullUp Widerstände EIN

Antwort:

0x2A (42)	0x01	0x01	0x04
Antwort	Anzahl	OK	Ende (EOT)

2.2.1.3. Pullup´s ausschalten

Wird im Datenblock eine „0“ gesendet, schaltet das Modem seine Pullup Widerstände AUS.

Befehl:

0x21 (33)	0x01	0x00	0x04
Befehl	Anzahl	AUS	Ende (EOT)

0 = PullUp Widerstände AUS

Antwort:

0x2A (42)	0x01	0x01	0x04
Antwort	Anzahl	OK	Ende (EOT)

2.2.2. I2C-SPEED 22 hex = 34 dez.

Mit diesem Befehl kann die Taktgeschwindigkeit am I2C-Bus zwischen 350 kHz und 40 Hz eingestellt werden. Dabei handelt es sich um einen zwei Byte großen Wert, der mit dem LSB voran im Datenblock abzulegen ist.



ACHTUNG:

Bitte beachten Sie bei der Auswahl der Busgeschwindigkeit, dass bei einem Takt von 40 Hz und 128 Byte die zum Slave übertragen oder gelesen werden sollen, die Übertragung ca. 30 Sekunden dauern kann.

Der zu übertragende Wert kann mit folgenden Formeln berechnet werden:

$$\text{Wert} = \frac{1}{\text{Takt [Hz]} * 0,4 * 10^{-6} \text{ s}}$$

$$\text{Takt [Hz]} = \frac{1}{\text{Wert} * 0,4 * 10^{-6} \text{ s}}$$

Die Aufteilung auf die zwei Bytes erfolgt anschließend mit folgender Formel wobei FIX der ganzzahlige Anteil einer Zahl ist.

$$\text{HBy} = \text{FIX}(\text{Wert} : 256)$$

$$\text{LBy} = \text{Wert} - \text{HBy} * 256$$

Beispiel:

Sie möchten einen Bustakt von 2500 Hz einstellen. So ergibt sich aus der Formel ein Wert von 1000

$$1000 / 256 = 3 \text{ Rest } 232$$

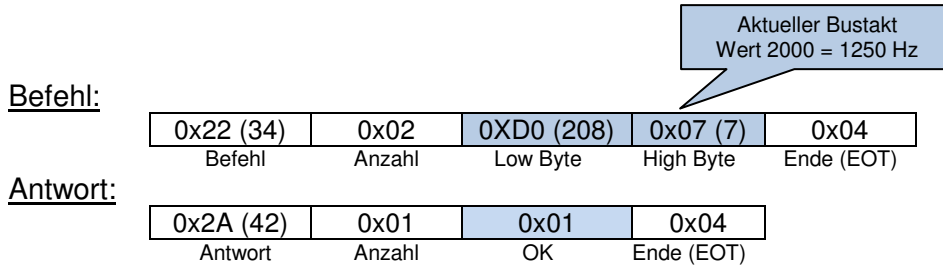
Die 3 muss in das high Byte, die 232 in das low Byte geschrieben werden

In nachfolgender Tabelle finden Sie die berechneten Daten für einige übliche Busgeschwindigkeiten:

Takt [kHz]	Takt [Hz]	Wert	high-Byte	low-Byte
350 kHz	350000 Hz	7	0	7
250 kHz	250000 Hz	10	0	10
125 kHz	125000 Hz	20	0	20
100 kHz	100000 Hz	25	0	25
50 kHz	50000 Hz	50	0	50
25 kHz	25000 Hz	100	0	100
10 kHz	10000 Hz	250	0	250
5 kHz	5000 Hz	500	1	244
2.50 kHz	2500 Hz	1000	3	232
1.25 kHz	1250 Hz	2000	7	208
1 kHz	1000 Hz	2500	9	196
0.5 kHz	500 Hz	5000	19	136
0.1 kHz	100 Hz	25000	97	168

2.2.2.1. Speed setzen

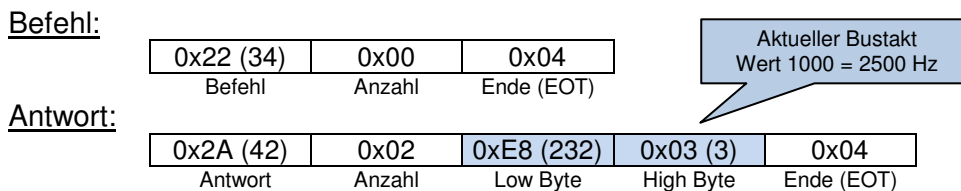
Um die Busgeschwindigkeit einzustellen muss der berechnete 2-Byte-Wert übertragen werden.



Sind keine Fehler aufgetreten, so antwortet das Modem mit einem Ok-Frame

2.2.2.2. Speed abfragen

Wird der Speed-Befehl mit einem leeren Datenblock (Länge 0) aufgerufen, so liefert das Modem den aktuell eingestellten Bus-Takt als 2-Byte-Wert zurück. Mit der angegebenen Formel kann aus den beiden Byte-Wert der aktuelle Bus-Takt errechnet werden.



$$Takt [Hz] = \frac{1}{(HBy * 256 + LBy) * 0,4 * 10^{-6} s}$$

2.3. Befehle der Gruppe 3 = I2C

Diese Gruppe beinhaltet alle Befehle, die man braucht um Daten auf dem I2C-Bus zuzusenden oder zu empfangen.

2.3.1. I2C-SET 31 hex = 49 dez.

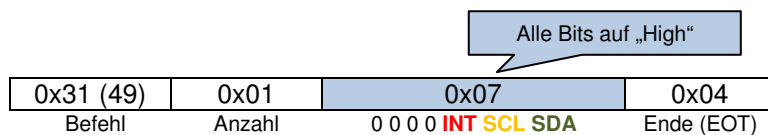
Mit diesem Befehl können zu Testzwecken die Signale **INT**, **SCL** und **SDA** einzeln auf High oder Low gesetzt werden.



ACHTUNG:

Nach dem Test müssen wieder alle Signale auf High gesetzt werden, da sonst der Bus das Start-Signal nicht erkennt.

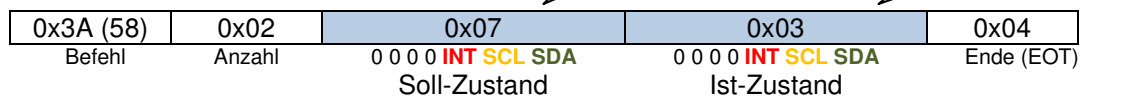
Befehl:



Die Bits 7 – 3 werden ignoriert. Bit zwei wird als **INT**, Bit eins als **SCL** und Bit null als **SDA** interpretiert. Der Zahl im Datenblock muss über die Wertigkeit der Bitstelle berechnet werden. (INT=4, SCL=2, SDA=1)

- 0x01 setzt SDA=1, SCL=0 und INT=0
- 0x02 setzt SDA=0, SCL=1 und INT=0
- 0x04 setzt SDA=0, SCL=0 und INT=1
- 0x07 setzt SDA=1, SCL=1 und INT=1

Antwort:

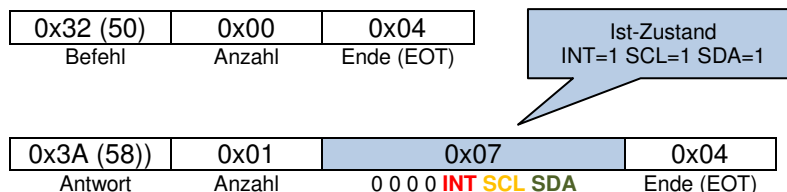


Wurde der Befehl richtig erkannt so antwortet das Modem mit dem Soll-Zustand und dem aktuellen Ist-Zustand der I2C-Signale:

2.3.2. I2C-GET 32 hex = 50 dez.

Mit I2C-Get wird der aktuelle Zustand der Signale **INT**, **SCL** und **SDA** abgefragt.

Befehl:



Info:



Beim I2C-USB-Modem leuchten die Leuchtdiode **SDA** und **SCL** wenn das Bussignal High ist. Die **INT** LED leuchtet wenn das Signal Low ist also wenn an einer Eingabekarte eine Signaländerung festgestellt wurde.

2.3.3. I2C-DATA 33 hex = 51 dez.

Mit diesem Befehl können Sie Daten von einem I2C-Slave empfangen, oder Daten an einen I2C-Slave senden.

Ob Daten gesendet oder empfangen werden sollen erkennt das Modem an dem R/W-Bit in der Adresse. Ist das R/W-Bit (Bit 0) „Low“, werden Daten geschrieben. Ist das R/W-Bit „High“, werden Daten gelesen.

Info:



„An gerade Slave-Adressen werden Daten geschrieben, von ungeraden Adressen werden Daten gelesen..“



ACHTUNG:

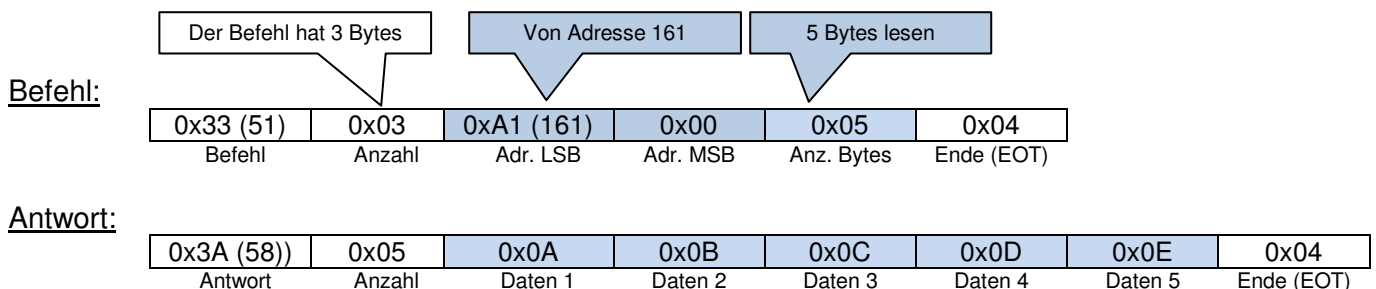
Das I2C-USB-Modem kann mit einer Data-Operation maximal 128 Bytes lesen oder schreiben. Bitte prüfen Sie im Datenblatt des Slaves welche Datenmengen gelesen oder geschrieben werden können. Beim beschreiben von EEPROMs z.B. muss nach acht Byte eine kurze Pause eingehalten werden.

2.3.3.1. Daten lesen

Um Daten von einem I2C-Slave zu lesen wird im Datenblock erst die Adresse abgelegt und dann die gewünschte Anzahl der zu lesenden Bytes angegeben.

Im Normalfall hat die Slave-Adresse eines I2C-Slaves nur 7 (bzw. 8) Bit. Da aber immer mehr IC's mit 10 Bit breiter Adresse auf den Markt kommen wurde die Adresse in Byte aufgeteilt. Ist das MSB null, so interpretiert das Modem die übermittelte Adresse als eine 7 Bit Adresse. Stehen Daten im MSB, so wird ein 10 Bit breiter Adresszugriff generiert.

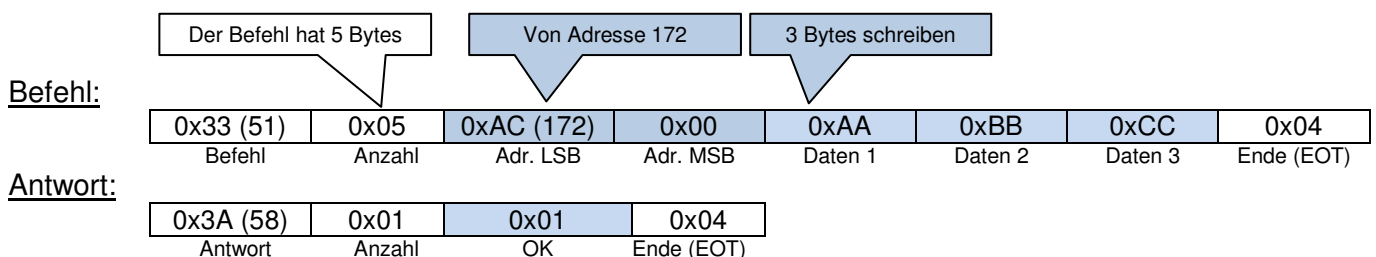
Im nachfolgenden Beispiel sollen 5 Bytes von einem I2C-Slave an Adresse 161 gelesen werden.



2.3.3.2. Daten schreiben

Um I2C-Daten an einen Slave zu senden wird im Datenblock erst die Adresse abgelegt und dann die zu sendenden Daten hinterher geschoben.

Im nachfolgenden Beispiel sollen 3 Bytes an einem I2C-Slave an Adresse 172 geschrieben werden.



Sind keine Fehler aufgetreten, so antwortet das Modem mit einem Ok-Frame