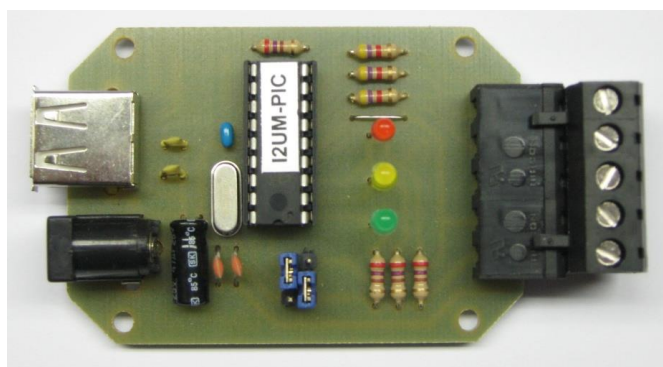


Inhalt

1. Allgemein	2
1.1. Kommunikationsparameter.....	2
1.2. Protokoll.....	2
1.2.1. Befehlsframe.....	2
1.2.2. Antwortframe	3
1.3. Fehlernummern.....	4
2. Befehle.....	5
2.1. Befehle der Gruppe 1 = INFO.....	5
2.1.1. VERSION 11 hex = 17 dez.	5
2.1.2. MODEM-CALL 12 hex =18 dez.....	5
2.2. Befehle der Gruppe 2 = KONFIG.....	6
2.2.1. PULLUP 21 hex = 33 dez.....	6
2.2.1.1. Pullup´s auslesen	6
2.2.1.2. Pullup´s einschalten	6
2.2.1.3. Pullup´s ausschalten	6
2.2.2. I2C-SPEED 22 hex = 34 dez.....	7
2.2.2.1. Speed setzen	8
2.2.2.2. Speed abfragen.....	8
2.3. Befehle der Gruppe 3 = I2C.....	9
2.3.1. I2C-SET 31 hex = 49 dez.....	9
2.3.2. I2C-GET 32 hex = 50 dez.....	9
2.3.3. I2C-DATA 33 hex = 51 dez.....	10
2.3.3.1. Aufbau der Slave-Adresse 8/10-Bit.....	10
2.3.3.2. Daten lesen	11
2.3.3.3. Daten schreiben	11
2.4. Befehle der Gruppe 4 = ANALYSE.....	12
2.4.1. SET FILTER 0x41 hex = 65 dez.....	12
2.4.2. LISTEN 0x42 hex = 66 dez.....	13
2.4.2.1. LISTEN mit zeitlicher Begrenzung.....	13
2.4.2.2. LISTEN ohne zeitliche Begrenzung.....	15
2.4.2.3. LISTEN-ERROR 0x4B hex = 75 dez.....	15
2.4.3. LOAD TABLE 0x43 hex = 67 dez.....	16
2.4.4. CLEAR TABLE 0x44 hex = 68 dez.....	17
2.4.5. CHECK-INT 0x45 hex = 69 dez.....	17



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:

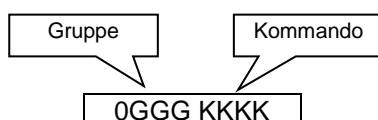
- der Befehl
- die Anzahl der Bytes im Datenblock (maximal 128)
- der Datenblock selbst und
- das Ende-Zeichen (EOT= END OF TRANSMISSION).

1.2.1. Befehlsframe

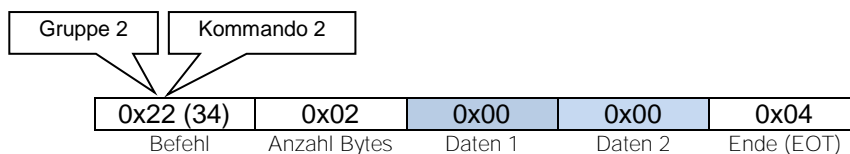
Die Befehle für das I²C-USB-Modem wurden in logische Gruppen zusammengefasst.

- (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
- (4) Analyse die Befehle für die Analyse- und Interrup-Funktionen

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



Beispiel für einen Befehls-Frame mit 2 Byte Daten:



Info:

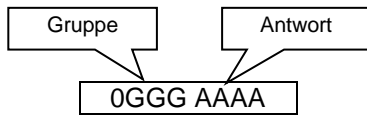
Zum besseren Verständnis wird in dieser Beschreibung der Datenblock immer mit einem hellblauen Hintergrund dargestellt.

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

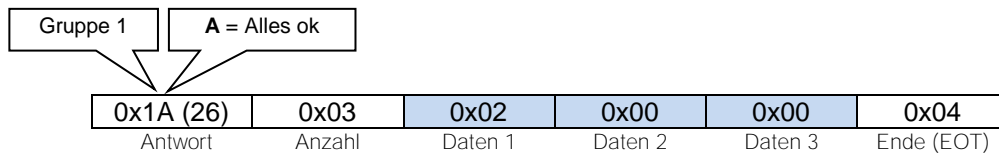
1.2.2. Antwortframe

Wurde ein gesendeter Befehl vom Modem bearbeitet 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 Hexadezimals „A“ für „Alles ok“

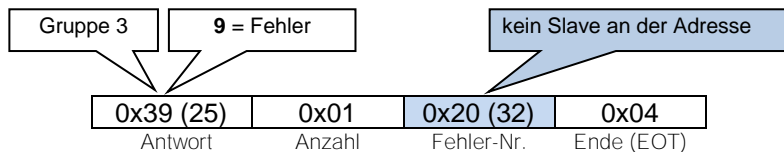


Beispiel für einen Antwort-Frame mit drei Byte Daten:



Im Fehlerfall steht im oberen Nibbel auch die Gruppennummer – im unteren Nibbel des Antwort-Bytes jedoch 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 dez)	alles OK kein Fehler
0x02 (2 dez)	Die Gruppenadresse (oberes Nibble) ist unbekannt
0x03 (3 dez)	Das Kommando (unteres Nibble) ist unbekannt
0x04 (4 dez)	Die Datenblocklänge wurde nicht gesendet, oder ist falsch
0x05 (5 dez)	Die Länge des Datenblocks ist zu groß
0x06 (6 dez)	END OF TRANSMISSION (04) fehlt am Ende des Frame
0x07 (7 dez)	Als EOT wurde nicht 04 gesendet
0x08 (8 dez)	Zeitüberschreitung beim Senden des Datenblocks
0x10 (16 dez)	Version Kommando ist falsch aufgebaut. Der Framekopf ist zwar richtig aber Rest stimmt nicht!
0x11 (17 dez)	Zu viele Daten beim Befehl Call-Modem
0x20 (32 dez)	Auf die gesendete Adresse reagiert kein Slave
0x21 (33 dez)	Der Slave hat auf das Acknow. nicht reagiert
0x22 (34 dez)	Wenn der Slave einen Clock- Stretch auslöst, wartet das Modem maximal 1,5 Sekunden. Danach erfolgt ein Timeout.
0x42 (66 dez)	Der Timeout-Wert für Listen ist ungültig
0x43 (67 dez)	Der Timeout-Wert für Listen wurde überschritten. Die gesuchte Adresse trat während der Überwachung nicht auf.
0x44 (68 dez)	Beim Warten auf den nächsten SCL-Imp. gab es ein Timeout. Ein Slave kann die SCL-Leitung blockieren, wenn er Zeit braucht um Daten aus internen Registern zu lesen (clock stretch). Diese Zeit ist auf max. 1,5 Sek. begrenzt.
0x45 (69 dez)	IS-Tabelle ist voll, kein weiter Eintrag möglich. Obwohl schon 16 Einträge in der Tabelle eingetragen sind, wurde versucht einen weiteren Eintrag vorzunehmen.
0x46 (70 dez)	Beim Senden der IS-Tabellen Daten wurden zu wenig Bytes gesendet. Pro Eintrag sind immer 3 Bytes erforderlich, Adresse-Low-Byte, Adresse-High-Byte und die Anzahl der Leseoperationen.
0x47 (71 dez)	Es können nicht mehr als 4 Leseoperationen in die IS-Tabelle eingetragen werden.
0x48 (72 dez)	Fehler beim Löschen der IS-Tabelle
0x49 (73 dez)	Eine Schreibadresse sollte in die IS-Tabelle eingetragen werden. Die Tabelle akzeptiert aber ausschließlich Leseadressen.
0x4A (74 dez)	Fehler beim Starten der INT-Überwachung
0x4B (75 dez)	INT-Überwachung soll mit einer leeren Tabelle gestartet werden
0x4C (76 dez)	Die Tabelle soll geändert werden obwohl die INT-Überwachung läuft.
0x4D (77 dez)	Alle Adressen in IS-Tabelle wurden ausgelesen und das INT-Signal ist immer noch auf LOW
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	0x30	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 Nachkommastelle. In unserem Beispiel ist das 2 3 0 und somit die Version 2.30

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:

0x12 (18)	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:

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

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

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

$$HBy = FIX (Wert : 256)$$

$$LBy = Wert - HBy * 256$$

Beispiel:

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

1000 / 256 = 3 Rest 232

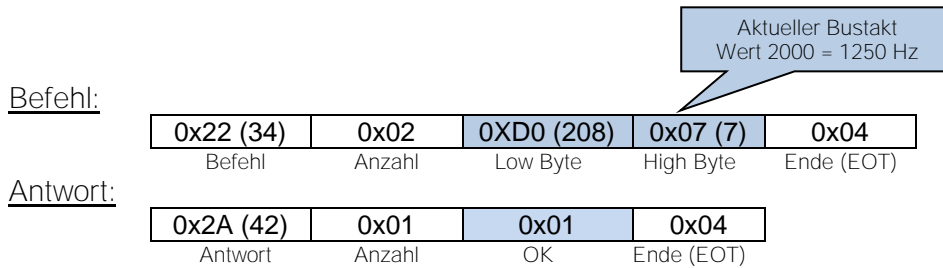
Die 3 muss in das High-Byte, die 232 in das Low-Byte geschrieben werden

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

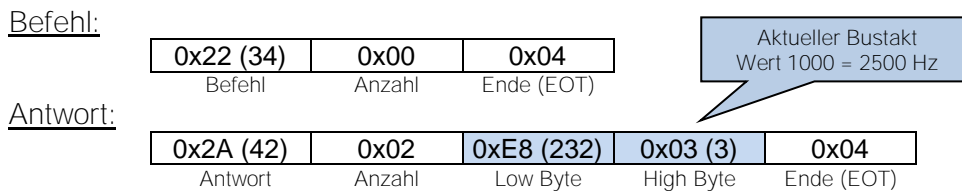
Um die Busgeschwindigkeit einzustellen muss der berechnete 2-Byte-Wert übertragen werden.



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 zusenden oder zu empfangen.

2.3.1. I2C-SET 31 hex = 49 dez.

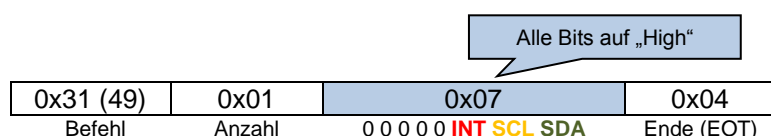
Mit diesem Befehl können zu Testzwecken die Signale **INT**, **SCL** und **SDA** einzeln auf High oder Low gesetzt werden.



ACHTUNG:

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

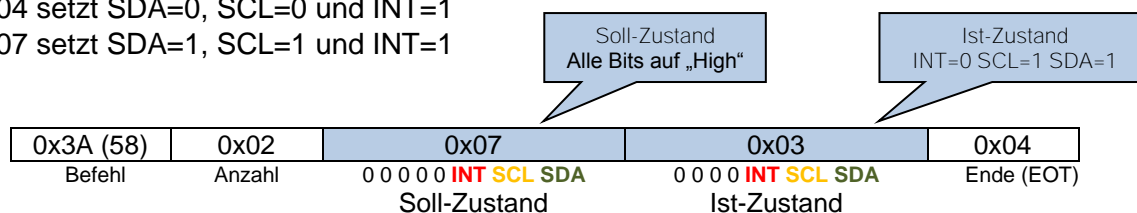
Befehl:



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:

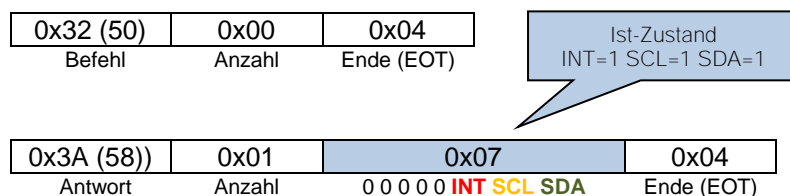


Wurde der Befehl richtig erkannt so antwortet das Modem mit dem Soll-Zustand und dem aktuellen Ist-Zustand der I2C-Signale:

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:

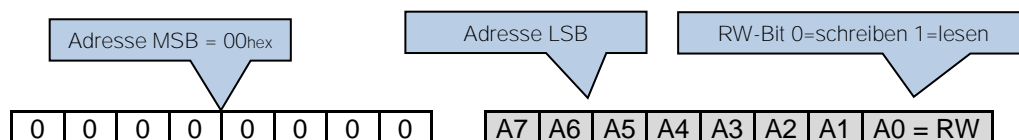
Das I2C-USB-Modem kann mit einer Data-Operation maximal 128 Bytes lesen oder schreiben. Bitte prüfen Sie im Datenblatt des Slaves welche Datenmengen gelesen oder geschrieben werden können. Beim Beschreiben von EEPROMs z.B. muss nach acht Byte eine kurze Pause eingehalten werden.

2.3.3.1. Aufbau der 8-Bit und 10-Bit Slave-Adresse

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.

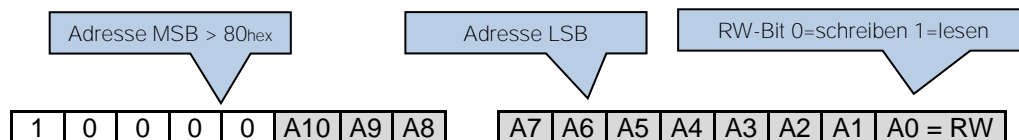
Aufbau 8-Bit Adresse:



Info:

Soll der Slave mit einer 8-Bit Adresse angesprochen werden müssen im MSB-Byte alle Bits auf LOW gesetzt sein.

Aufbau 10Bit-Adresse

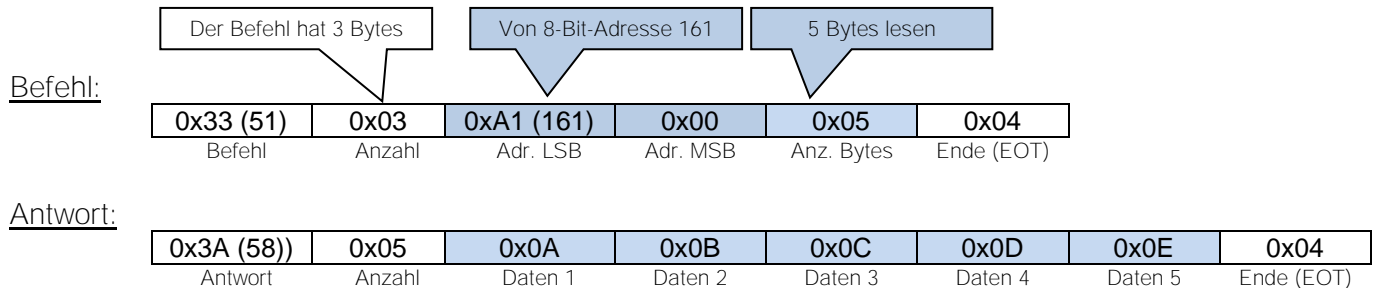


Info:

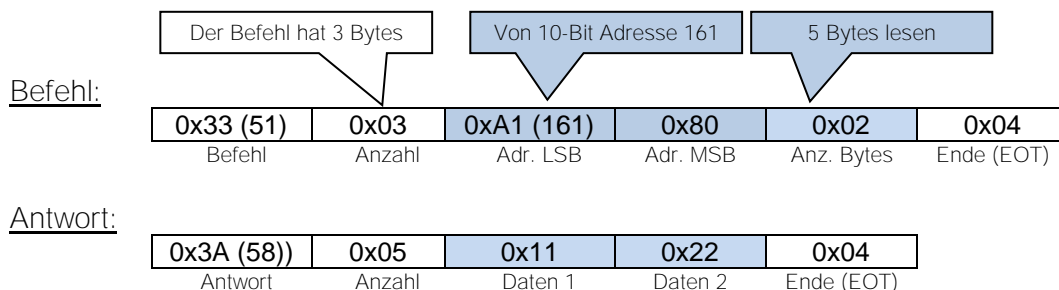
Soll der Slave mit einer 10-Bit Adresse angesprochen werden muss das höchste Bit im MSB-Byte auf HIGH gesetzt sein.

2.3.3.2. Daten lesen

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



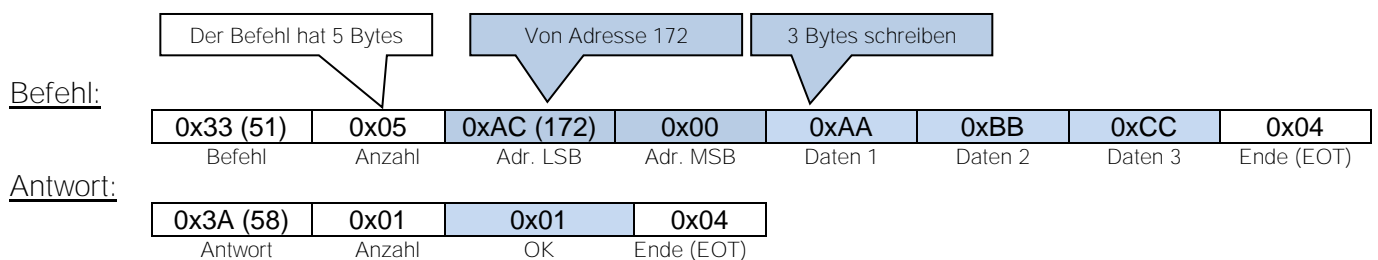
Hier sollen zwei Bytes von einem I2C-Slave mit 10-Bit Adresse gelesen werden.



2.3.3.3. 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.



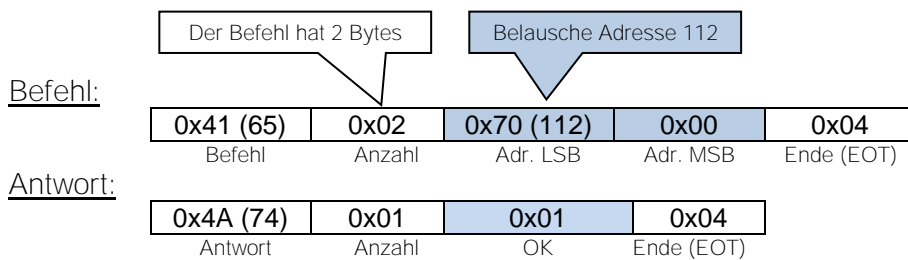
Sind keine Fehler aufgetreten, so antwortet das Modem mit einem Ok-Frame

2.4. Befehle der Gruppe 4 = ANALYSE

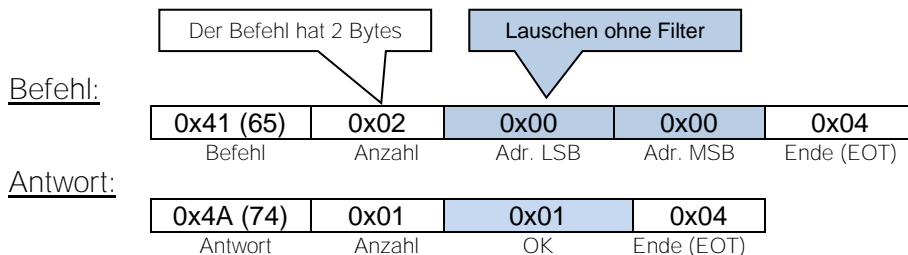
Diese Gruppe beinhaltet Befehle die für eine Analyse des Datenverkehrs auf dem I2C-Bus genutzt werden können um Fehler in der I2C-Kommunikation zu finden.

2.4.1. SET FILTER 0x41 hex = 65 dez.

Da eine generelle Überwachung des I2C-Busses eine zu große Datenmenge liefern würde kann mit diesem Befehl ein Filter auf eine I2C-Slaveadresse gesetzt werden. Wenn das USB-Modem mit dem Befehl LISTEN den Datenverkehr „belauscht“ erfolgt eine Aufzeichnung der Daten erst wenn die im Filter eingestellte Adresse im I2C-Protokoll erkannt wird.



Ist die Adresse und der Frame gültig, so wird die Adresse in den Filter eingetragen und das Modem antwortet mit einem Ok-Frame.



Wird die Adresse 0x0000 in den Filter eingetragen zeichnet das Modem alle Daten auf dem I2C-Bus auf und sendet diese an den PC. (→ Kapitel 2.4.2.2 „LISTEN ohne zeitliche Begrenzung“)

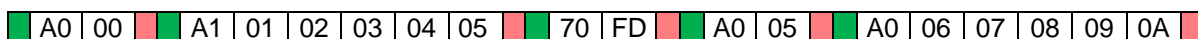
2.4.2. LISTEN 0x42 hex = 66 dez.

Mit diesem Befehl wird die Aufzeichnung des Datenverkehrs am I2C-Bus gestartet. Es stehen vier verschiedenen Modi für die Aufzeichnung zur Verfügung:

- Zeitlich begrenzt mit Adressfilter
- Zeitlich unbegrenzt mit Adressfilter
- Zeitlich begrenzt ohne Adressfilter
- Zeitlich unbegrenzt ohne Adressfilter

Info:
 Die Filter-Adresse muss vorher mit dem Befehl SET FILTER an das Modem übermittelt werden.

Beispiel für einen Datenstrom ■ =I2C-Start ■ =I2C-Stop



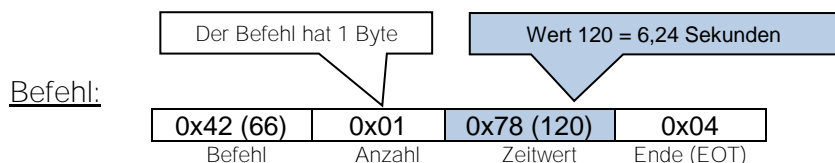
Im ersten Block wird an einem I2C-EEPROM (Adresse 160) der Adresszeiger auf 0 eingestellt.
 Im nächsten Block liest der Master aus dem EEPROM (Adresse 161) 5 Bytes Daten aus.
 Im dritten Block wird an einem Portbaustein (Adresse 112) das Bitmuster „11111101“ ausgegeben.
 Im vierten Block wird am I2C-EEPROM (Adresse 160) der Adresszeiger auf 5 eingestellt.
 Im fünften Block werden 5 Bytes Daten in das EEPROM (Adresse 160) geschrieben

2.4.2.1. LISTEN mit zeitlicher Begrenzung

Die „Lauschzeit“ kann mit dem Befehl zwischen 0,052 und 13 Sekunden eingestellt werden. Die Berechnung erfolgt mit folgender Formel.

$$Lauschzeit [Sek] = Wert * 0,052 Sek$$

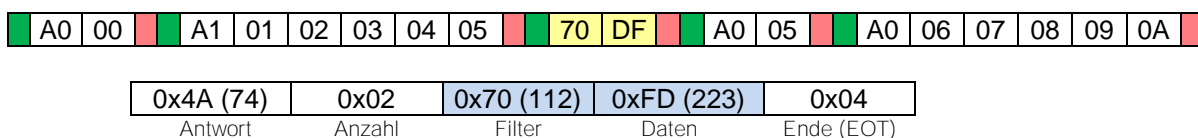
Das folgende Beispiel startet die Listenfunktion für ca. 6 Sekunden mit dem zuvor eingestellten Filter.



Antwort:

Wird die Filteradresse im Datenstrom gefunden sendet das Modem einen Antwortframe.

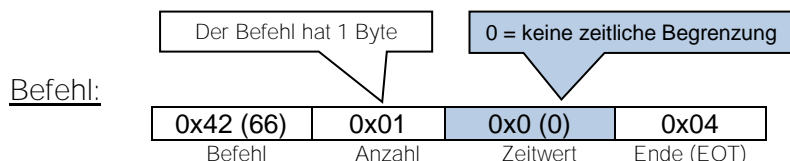
Steht zum Beispiel der Filter auf 0x70 und ein Master schickt 0xDF an die Adresse 0x70 wird nachfolgender Frame von I2C-USB-Modem erzeugt:



Im Datenbereich findet man als erstes die Adresse wieder die belauscht wurde, danach die Daten die auf dem Bus gesehen wurden.

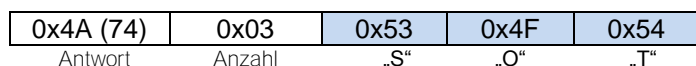
2.4.2.2. LISTEN ohne zeitliche Begrenzung

Wird mit dem LISTEN-Befehl die Zeit 0 übergeben, startet das Modem eine zeitlich unbegrenzte Überwachung. Diese wird durch den Host beendet, wenn dieser EOT 0x04 (4 dez.) an das Modem sendet.



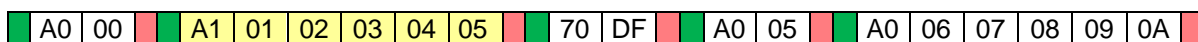
Antwort:

Zum Beginn der Überwachung sendet das Modem „SOT“ (Start Of Transmission) in folgendem Frame:

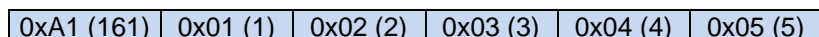


Dabei fällt auf, dass das Frame-Ende 0x04 (4 dez.) fehlt.

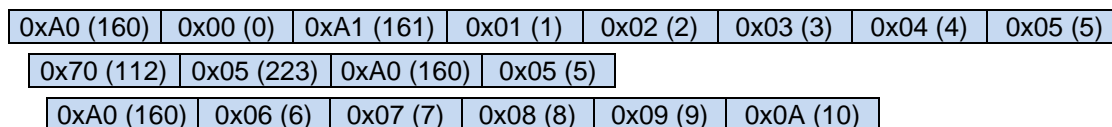
Nun folgen die Daten die das Modem auf dem I2C-Bus vorfindet. Dabei beginnt das Modem mit der Datenübertragung an den PC sobald ein I2C-Start Signal im Datenstrom erkannt wird. Mit dem I2C-Stop hält auch die Datenübertragung an. Es entsteht ein „Open Frame“, der vom PC beendet werden muss.



Bei einem eingestellten Filter von 0xA1 werden vom Modem an den PC folgende Bytes gesendet

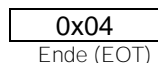


Ist der Filter ausgeschaltet werden an das Modem alle Bytes gesendet die am Bus erkannt werden

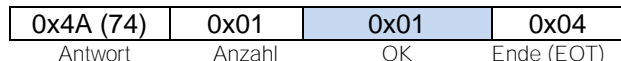


Werden vom Modem keine Daten mehr benötigt muss der PC ein EOT = 0x04 (4 dez.) senden.

Befehl:



Antwort:



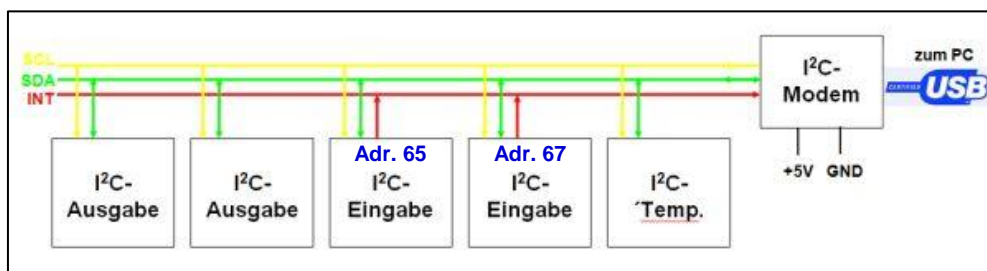
2.4.2.3. LISTEN-ERROR 0x4B hex = 75 dez.

Beim Belauschen des I2C-Busses kann es vorkommen, dass Daten unvollständig gesendet werden. Wird z.B. die Adresse und das erste Datum noch richtig und das zweite Datum unvollständig gesendet, überträgt das I2C-USB-Modem die bis zu diesem Zeitpunkt belauschten Daten an den PC. Um anzuzeigen das bei der Datenübermittlung auf dem I2C-Bus etwas schief gelaufen ist, trägt das Modem eine 0x4B (75 dez) in den Framekopf ein.

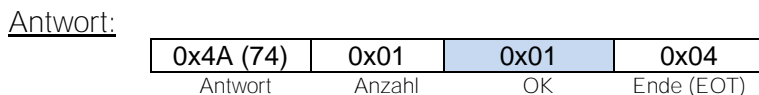
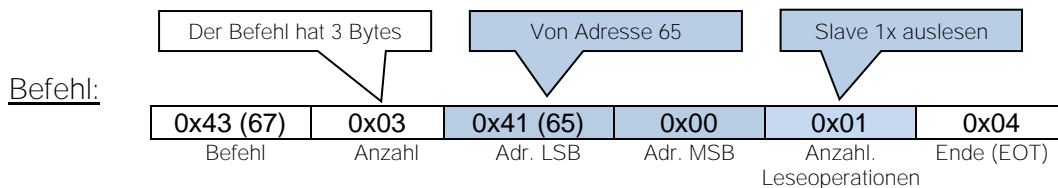
2.4.3. LOAD TABLE 0x43 hex = 67 dez.

Für die Interrupt Bearbeitung am I2C-Bus haben wir ein interessantes Feature in unser I2C-USB-Modem integriert. Am Eingang INT können alle Interrupt fähige Slaves parallel angeschlossen werden. Bei unseren Eingabekarten mit dem Baustein PCF8574 bzw. PCF8574A wird der INT auf „LOW“ gezogen sobald an den Eingängen ein Signalwechsel erkannt wird. Das USB-Modem ist ab der Version 2.2 in der Lage, bei einer fallenden INT-Flanke, selbstständig eine Reihe von Slaves abzufragen und geänderte Eingangssignale an den PC zu melden ohne dass eine Leseoperation ausgeführt werden muss.

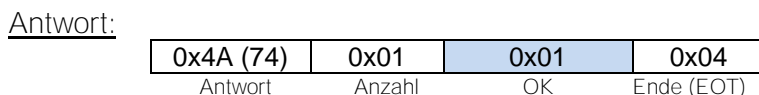
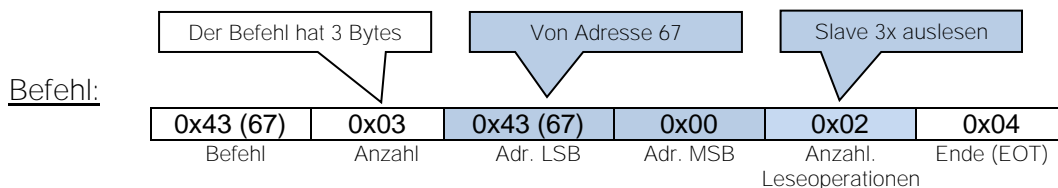
Im Modem wird dazu eine Tabelle angelegt in der die Adressen aller angeschlossenen Eingabekarten und die Anzahl der Leseoperationen hinterlegt wird. Ist die Interrupt Bearbeitung freigegeben (CHECK-INT = 1) wird diese Tabelle bei einem Interrupt von oben nach unten abgearbeitet. Liest das Modem Daten von dem Slave, der den Interrupt ausgelöst hat, verschwindet der Interrupt. Das I2C-USB-Modem erkennt daran, dass dieser Slave den Interrupt ausgelöst hat und teilt dem PC die Slave-Nummer und den Inhalt der ausgelesenen Daten mit.



Um einen Eintrag in der IS-Tabelle des I2C-USB-Modems vorzunehmen ist folgender Frame zu senden



Sind die Daten gültig und erfolgreich in die Tabelle eingetragen worden, so antwortet das Modem mit einem OK. Anschließend kann die nächste Adresse abgelegt werden



Info: Die IS-Tabelle kann maximal 16 Adressen + Anzahl der Leseoperationen aufnehmen. Sie wird im EEPROM des PIC-Prozessors abgelegt und bleibt bei Spannungsausfall erhalten.

ACHTUNG: Wenn die INT-Überwachung aktiv ist, kann die IS-Tabelle nicht bearbeitet werden.

2.4.4. CLEAR TABLE 0x44 hex = 68 dez.

Wird dieser Befehl an das Modem gesendet, so werden alle 16 Einträge der IS-TABELLE gelöscht.

Befehl:

0x44 (68)	0x00	0x04
Befehl	Anzahl	Ende (EOT)

Antwort:

0x4A (74)	0x01	0x01	0x04
Antwort	Anzahl	OK	Ende (EOT)

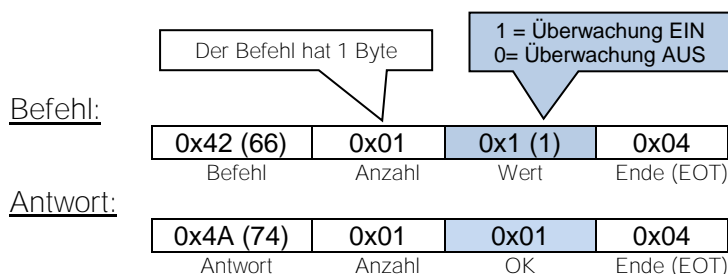


ACHTUNG:

Wenn die INT-Überwachung aktiv ist, kann die Tabelle nicht gelöscht werden.

2.4.5. CHECK-INT 0x45 hex = 69 dez.

Dieser Befehl startet (Wert=1) oder stoppt (Wert=0) die INT-Überwachung.

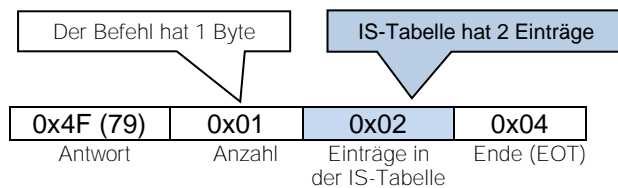


Ist CHECK-INT aktiv, sind lediglich die beiden Befehle zur Manipulation der IS-Tabelle gesperrt. Ansonsten arbeitet das Modem ganz normal weiter und beobachtet im Hintergrund den Status der INT-Leitung.

Tritt ein INT-Signal auf (rote LED leuchtet) passiert folgendes:

- Das Modem sendet einen BUSY-Frame an den PC.
- Das Modem arbeitet nacheinander die IS-Tabelle ab und liest die gespeicherte Anzahl von Bytes aus dem Slave aus. Verschwindet das INT-Signal ist der Slave gefunden worden, der den Interrupt ausgelöst hat.
- Das Modem sendet die gelesenen Daten von diesem Slave an den PC.
- Das Modem geht wieder in den Normalbetrieb über.

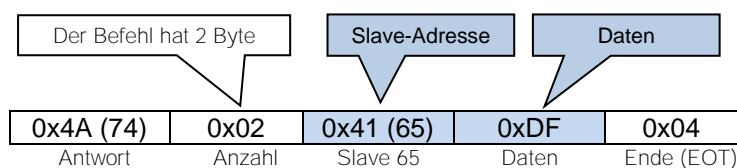
Aufbau BUSY-Frame:



Der BUSY-Frame, sagt aus dass das Modem für eine gewisse Zeit nicht erreichbar ist, da nun die IS-Tabelle abgearbeitet werden muss. Die Zeit wie lange es dauert, wird durch den Umfang der Tabelle (Anzahl der Einträge und der Leseoperationen) bestimmt. Wird bei einer dieser Leseoperationen das INT-Signal gelöscht, so antwortet das Modem mit einem Datenframe:

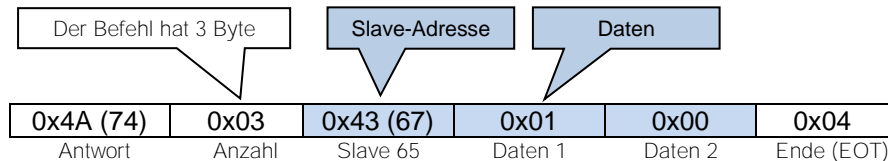
Aufbau DATEN-Frame:

Beispiel 1:



Während der Leseoperation an der Adresse 0x41 wurde das INT-Signal gelöscht. Empfangen hat das Modem das Datum 0xDF

Beispiel 2:



Während der Leseoperation an der Adresse 0x43 wurde das INT-Signal gelöscht. Empfangen hat das Modem das Datum 0x01 und anschließend 0x00. Demzufolge war an der gepufferten I2C-Eingabekarte ein Signalwechsel am Bit 0 von „HIGH“ auf „LIOW“.



ACHTUNG

Das INT-Signal besteht nur aus einer Leitung an der mehrere Bausteine angeschlossen sind. Lösen nun mehrere Bausteine gleichzeitig ein INT-Signal aus, überlagern sich diese. Das Modem ist nicht in der Lage eine Überlagerung zu erkennen!

Beispiel:

In der Tabelle sind die Adressen 0x71, 0x75 und 0x79 als INT-Quellen eingetragen. Nun lösen gleichzeitig 0x71 und 0x79 einen INT aus.

Laut Tabelle liest das Modem die Daten von Adresse 0x71, aber das INT-Signal bleibt stehen, da es vom Slave 0x79 auf „low“ gehalten wird.

Erst wenn das Modem die 0x79 ausliest wird das INT-Signal wieder freigegeben. (INT-Pegel wird high – rote LED geht aus) Die Daten von Adresse 0x79 werden an den PC übertragen. Die Daten des Bausteins an der Adresse 0x71 gehen wegen der Überlagerung verloren