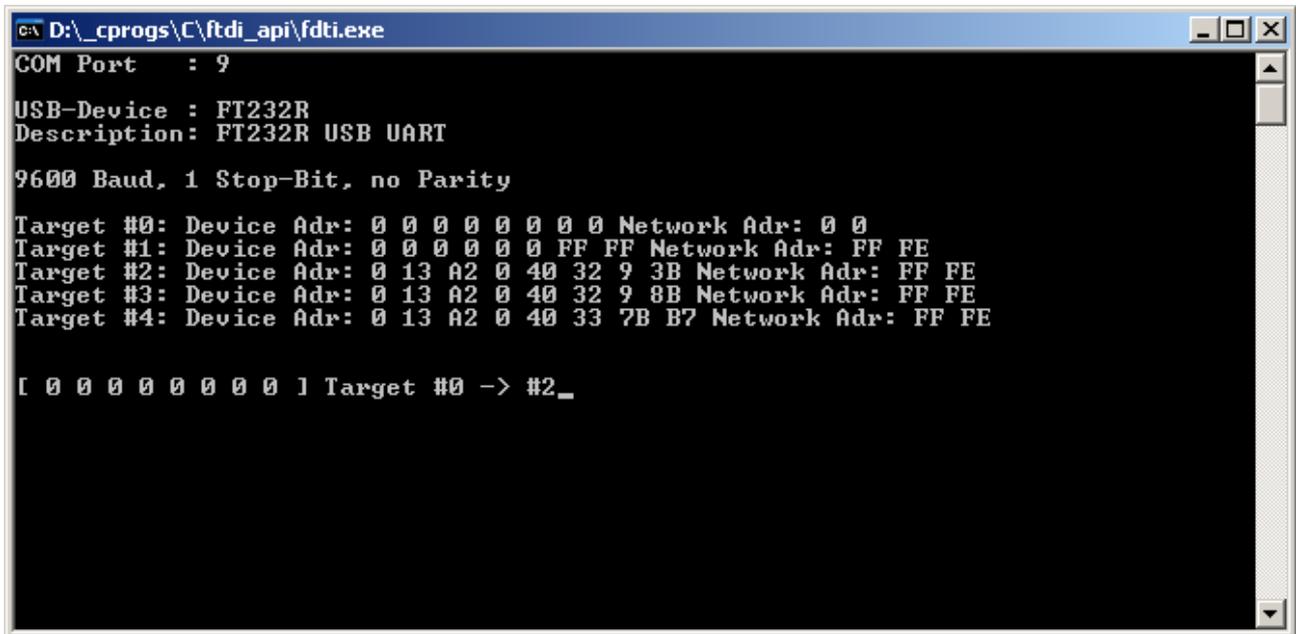


Zur Abschreckung hier einige Screendumps, die vorbeugend darauf hinweisen sollen, welche Provokation das Tool für das verwöhnte Auge des Windows User darstellt !

Beim Programmstart werden einigen Statusmeldung ausgegeben. Möglicherweise auch Fehlermeldungen, wenn erforderliche Hardware oder dll's nicht gefunden werden.

Aktion: Eingabe von '#2' um Target #2 als Zieladresse für die folgenden Remote Eingaben zu machen.



```
C:\_cprogs\C\ftdi_api\fdti.exe
COM Port : 9
USB-Device : FT232R
Description: FT232R USB UART
9600 Baud, 1 Stop-Bit, no Parity
Target #0: Device Adr: 0 0 0 0 0 0 0 0 Network Adr: 0 0
Target #1: Device Adr: 0 0 0 0 0 0 FF FF Network Adr: FF FE
Target #2: Device Adr: 0 13 A2 0 40 32 9 3B Network Adr: FF FE
Target #3: Device Adr: 0 13 A2 0 40 32 9 8B Network Adr: FF FE
Target #4: Device Adr: 0 13 A2 0 40 33 7B B7 Network Adr: FF FE

[ 0 0 0 0 0 0 0 0 ] Target #0 -> #2_
```

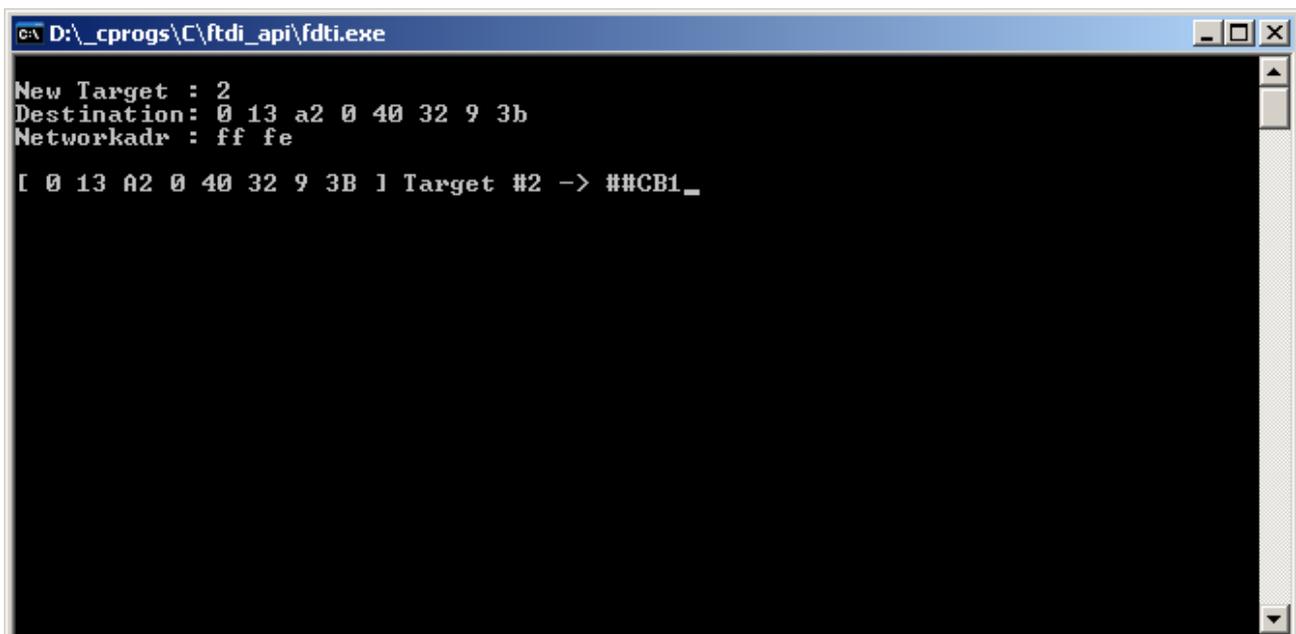
Die Adressdaten zu Target #2 werden ausgegeben.

Aktion: Eingabe des Remote AT-Command 'ATCB1', das auf das Target mit der im Prompt angegebenen Seriennummer versandt wird.

ATCB1 simuliert das Betätigen des Commissioning Pushbutton Pin.20 am remote device.

Anstelle von 'AT' wird hier übrigens '##' eingegeben: damit wird das remote device #2 adressiert.

Wird anstelle von 'AT' ein '++' verwendet, dann erfolgt die Ausgabe auf dem lokalen device (das ist das Teil, das direkt per USB-Adapter am PC angeschlossen ist !).



```
C:\_cprogs\C\ftdi_api\fdti.exe
New Target : 2
Destination: 0 13 a2 0 40 32 9 3b
Networkadr : ff fe

[ 0 13 A2 0 40 32 9 3B ] Target #2 -> ##CB1_
```

Das AT-Command wird ausgeführt, in den oberen Bildschirmzeilen die Bestätigung der Eingabe sowie die Byte-Sequenz, die an das Radio gesandt wird.

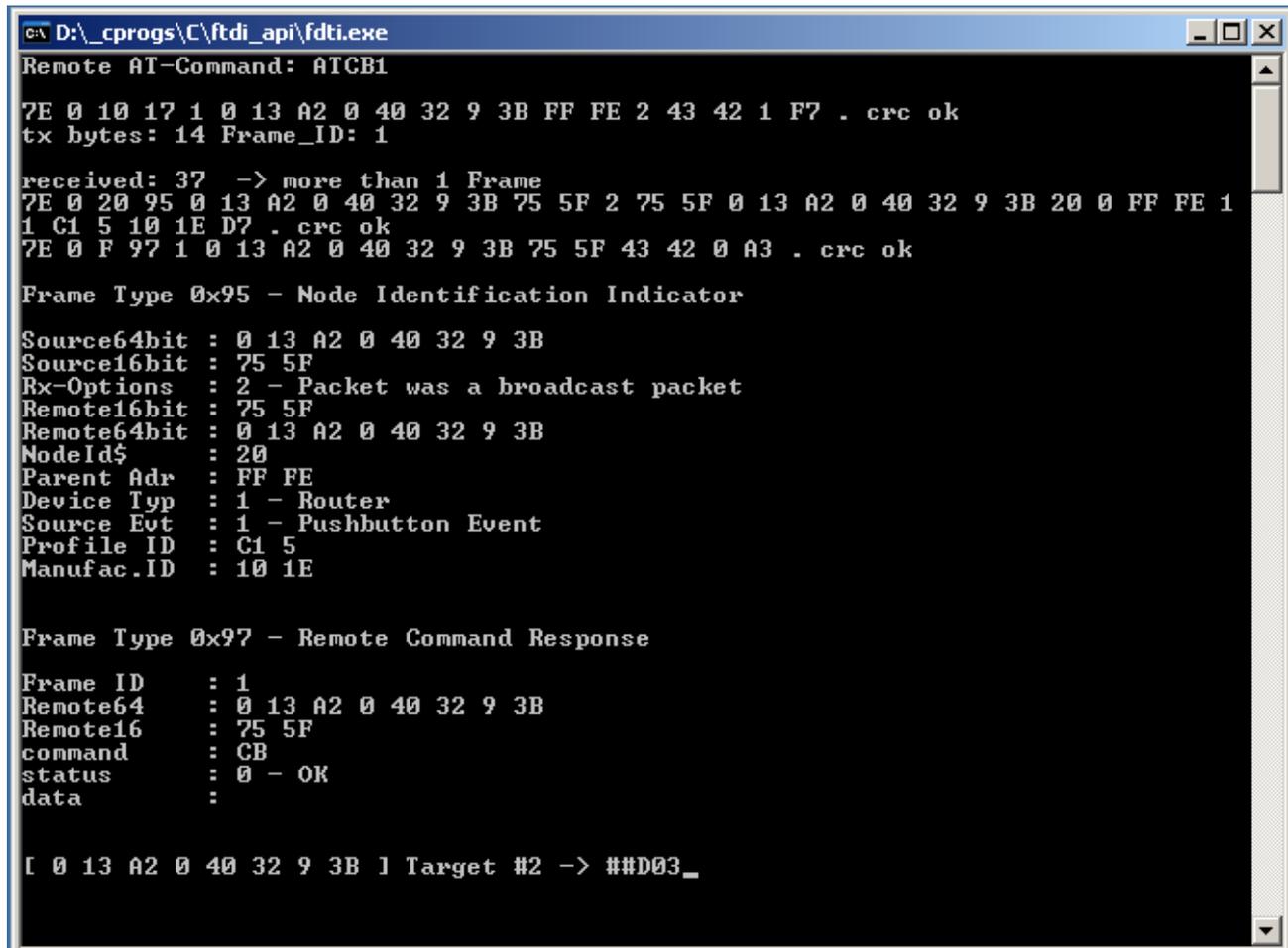
Es folgen die Rückmeldungen des adressierten Devices ebenfalls zunächst als Byte-Folge (in diesem Falle werden kurz nacheinander 2 Frames empfangen).

Anschließend werden die Daten entsprechend der Frame-Struktur formatiert dargestellt.

Frame 0x95 wird vom Remote Device als Broadcast versandt und deswegen hier empfangen.

Frame 0x97 ist die Rückmeldung über die Ausführung des AT-Command ATCB.

Aktion: Eingabe von 'ATD03' - den Pin.0 am remote device als digitalen Input schalten



```
D:\_cprogs\C\ftdi_api\fdti.exe
Remote AT-Command: ATCB1

7E 0 10 17 1 0 13 A2 0 40 32 9 3B FF FE 2 43 42 1 F7 . crc ok
tx bytes: 14 Frame_ID: 1

received: 37 -> more than 1 Frame
7E 0 20 95 0 13 A2 0 40 32 9 3B 75 5F 2 75 5F 0 13 A2 0 40 32 9 3B 20 0 FF FE 1
1 C1 5 10 1E D7 . crc ok
7E 0 F 97 1 0 13 A2 0 40 32 9 3B 75 5F 43 42 0 A3 . crc ok

Frame Type 0x95 - Node Identification Indicator
Source64bit : 0 13 A2 0 40 32 9 3B
Source16bit : 75 5F
Rx-Options : 2 - Packet was a broadcast packet
Remote16bit : 75 5F
Remote64bit : 0 13 A2 0 40 32 9 3B
NodeID$ : 20
Parent Adr : FF FE
Device Typ : 1 - Router
Source Evt : 1 - Pushbutton Event
Profile ID : C1 5
Manufac.ID : 10 1E

Frame Type 0x97 - Remote Command Response
Frame ID : 1
Remote64 : 0 13 A2 0 40 32 9 3B
Remote16 : 75 5F
command : CB
status : 0 - OK
data :

[ 0 13 A2 0 40 32 9 3B 1 Target #2 -> ##D03_
```

Das AT-Command ATD03 wird ausgeführt, die Bestätigung als Frame 0x97 wird empfangen.

Aktion: Eingabe 'ATIS' - den IO-Status der Eingangsports des remote device abholen

```
D:\_cprogs\C\ftdi_api\fdti.exe
Remote AT-Command: ATD03

7E 0 10 17 2 0 13 A2 0 40 32 9 3B FF FE 2 44 30 3 5 . crc ok
tx bytes: 14 Frame_ID: 2

received: 13 -> Frame complete !
7E 0 F 97 2 0 13 A2 0 40 32 9 3B 75 5F 44 30 0 B3 . crc ok

Frame Type 0x97 - Remote Command Response
Frame ID      : 2
Remote64     : 0 13 A2 0 40 32 9 3B
Remote16    : 75 5F
command      : D0
status       : 0 - OK
data         :

[ 0 13 A2 0 40 32 9 3B ] Target #2 -> ##IS
```

Das AT-Command ATIS wird ausgeführt, die Bestätigung und der Portstatus wird via Frame 0x97 zurückgeliefert.

Im Datenbereich ist der Pin-Status codiert.

Aktion: Senden des Textstrings "ABCDEFGG"

```
D:\_cprogs\C\ftdi_api\fdti.exe
Remote AT-Command: ATIS

7E 0 F 17 3 0 13 A2 0 40 32 9 3B FF FE 2 49 53 DF . crc ok
tx bytes: 13 Frame_ID: 3

received: 19 -> Frame complete !
7E 0 15 97 3 0 13 A2 0 40 32 9 3B 75 5F 49 53 0 1 0 1 0 0 1 87 . crc ok

Frame Type 0x97 - Remote Command Response
Frame ID      : 3
Remote64     : 0 13 A2 0 40 32 9 3B
Remote16    : 75 5F
command      : IS
status       : 0 - OK
data         : 1 0 1 0 0 1

[ 0 13 A2 0 40 32 9 3B ] Target #2 -> ABCDEFGG_
```

Der Textstring wird versandt, die Erfolgsmeldung erfolgt via Frame 0x8B

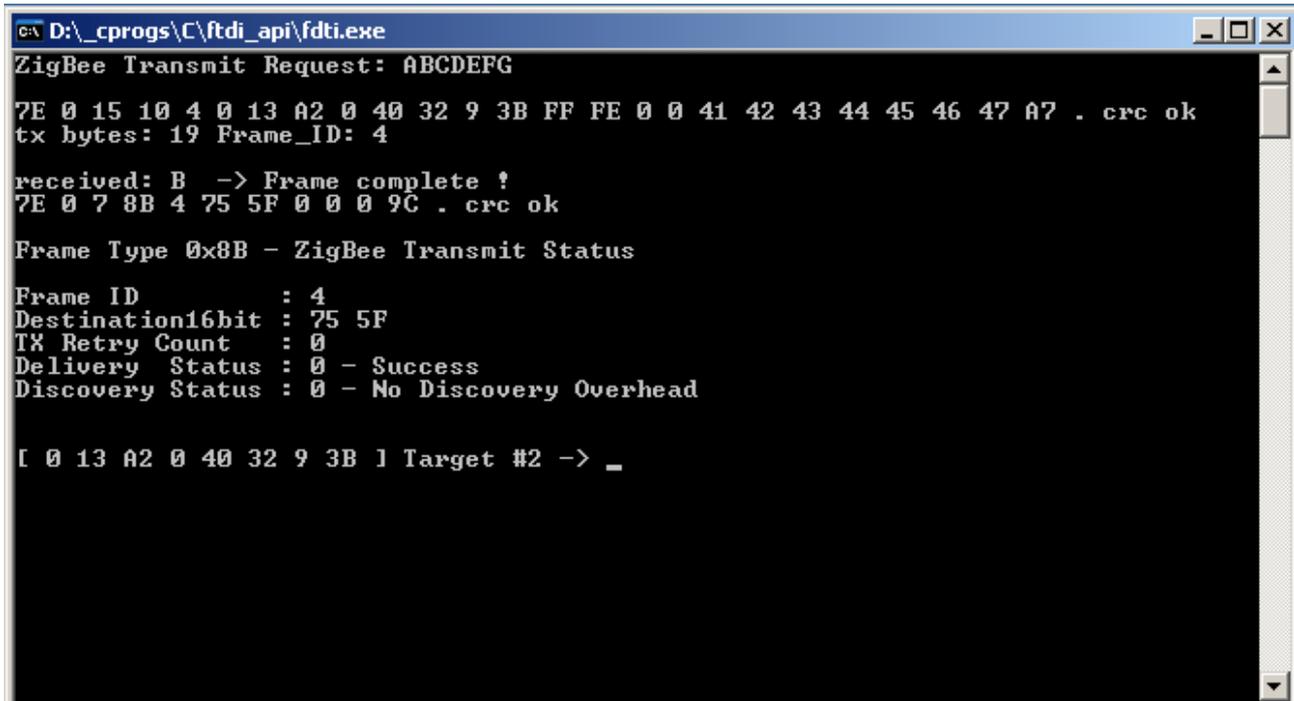
Hinweis:

Die Frame_ID ist identisch mit der Frame_ID des auslösenden Kommandos.

Damit kann die Bestätigung einem vorausgegangenen Kommando zugeordnet werden.

Die Frame_ID wird im Programm automatisch incrementiert.

Eine '0' wird nicht zugewiesen, weil Frame_ID == 0 bedeutet, dass keine Bestätigung erfolgen soll.



```
C:\_cprogs\C\ftdi_api\fdti.exe
ZigBee Transmit Request: ABCDEFG
7E 0 15 10 4 0 13 A2 0 40 32 9 3B FF FE 0 0 41 42 43 44 45 46 47 A7 . crc ok
tx bytes: 19 Frame_ID: 4
received: B -> Frame complete !
7E 0 7 8B 4 75 5F 0 0 0 9C . crc ok
Frame Type 0x8B - ZigBee Transmit Status
Frame ID          : 4
Destination16bit  : 75 5F
TX Retry Count    : 0
Delivery Status   : 0 - Success
Discovery Status  : 0 - No Discovery Overhead

[ 0 13 A2 0 40 32 9 3B ] Target #2 -> _
```

Ende.