

## PC-Kommunikation mit dem DX35

Die Kommunikation des PC mit dem DX35 geschieht über die serielle Schnittstelle. Der USB-Anschluß des DX35 wird als virtuelle serielle Schnittstelle auf dem PC eingerichtet, so dass Standardsoftware auf einen COM-Port zugreifen kann. Aufgrund der virtuellen Schnittstelle, muss keine feste Baudrate im Steuerprogramm auf dem PC eingestellt werden. Es werden alle Baudraten von 300 bis 500.000 Baud akzeptiert, die Übertragungsrate erfolgt immer über die USB-Baudrate von 12 Megabit.

Zur Kommunikation mit dem PC muss das DX35-Interface nicht am OBD2-Stecker des Fahrzeugs angeschlossen sein. Der Microcontroller im DX35 bekommt seine Stromversorgung über die USB-Leitung. In diesem Modus kann zum Beispiel der Treiber installiert und die Kommunikation mit dem Interface auf dem PC getestet werden. Es kann jedoch kein OBD2-Verbindungsaufbau gestartet werden, hierbei erscheint die Fehlermeldung „NO POWER“.

Wenn Sie ein Terminalprogramm benutzen, drücken Sie nach dem ersten Anschluß des DX35 die „Enter“-Taste bis das Zeichen „>“ erscheint. Das Zeichen „>“ bedeutet, daß das Interface bereit ist, Befehle zu empfangen. Das Interface unterscheidet nun zwei verschiedene Kommandogruppen:

1. Interne Befehle zur Konfiguration und Initialisierung des Interface-Controllers. Alle diese Befehle beginnen mit den Zeichen „AT“, dieses wurde von den Steuerbefehlen bei Modems übernommen und bedeutet „Attention, Achtung“.
2. Daten, die an den OBD2-Bus für das Steuergerät des Fahrzeugs weitergeleitet werden. Alle diese Befehle werden als hexadezimale Zahlen übermittelt, es dürfen deshalb nur ASCII-Zeichen 0-9 und A-F paarweise eingegeben werden.

Alle eingegebenen Befehle müssen mit einem Zeilenende-Zeichen (Carriage Return, Dez. 13, Hex \$0D) abgeschlossen werden. Leerzeichen oder Tabulatoren werden automatisch herausgefiltert, Klein- und Großschreibung wird nicht unterschieden. In den folgenden Beispielen muss jede Eingabe mit dem Zeilenende-Zeichen abgeschlossen werden, es wird nicht extra angegeben.

### *Beispiele:*

```
at dp
    Wird intern nach ATDP gewandelt
A T Z
    Wird intern nach ATZ gewandelt
01 1c
    Wird intern nach 011C gewandelt
```

Falls das Zeilenende-Zeichen CR ausbleibt, wird der Befehl automatisch nach 5 Sekunden abgebrochen und ein „?“ wird als Fehlermeldung ausgegeben.

Befehle, die der Controller nicht versteht oder Falsch-eingaben bei Hex-Werten, werden ebenfalls mit einem „?“ als Fehlermeldung quittiert.

OBD2-Befehle müssen immer mit geraden Anzahl von Hex-Zeichen eingegeben werden:

### *Beispiel:*

```
0100
oder
01 00
```

Eine ungerade Anzahl von Zeichen erzeugt eine Fehlermeldung.

Einige AT-Befehle benötigen als zusätzliche Parameter einen oder mehrere Hex-Zeichen. Die genaue Anzahl entnehmen Sie bitte der Befehlsliste. Eine falsche Anzahl der Parameter wird ebenfalls mit einer Fehlermeldung quittiert.

Obwohl der DX35 weitgehend denselben Befehlssatz wie die AGV- oder ELM-Controller benutzt (damit ist gewährleistet, daß Standardprogramme funktionieren) gibt es doch einige Unterschiede. Die von den Standardprogrammen benutzten Befehle wurden zu 100% emuliert, andere Befehle wurden teilweise einfacher oder sinnvoller belegt. Zusätzlich sind einige Befehle zur Konfiguration und für die besonderen Features des DX35 vorhanden, wie die Änderung des Identifikationstextes und das Auslesen und Setzen spezieller Parameter aus dem bzw. in den Controller.

**Der AT-Befehlssatz des DX35****>ATZ**

Dieser Befehl bewirkt einen sofortigen Abbruch aller laufenden Funktionen und einen Warmstart des Controllers. Alle Parameter werden in den Grundzustand gesetzt und der Selbsttest (durch blinken der LEDs angezeigt) wird durchgeführt. Zum Schluß wird der Ident-Text ausgegeben.

**Ausgabe:**

DIAMEX DX35 v1.0

&gt;

Hinweis: Der Text ist abhängig von der Bios-Version.

**>ATWS**

Dieser Befehl bewirkt einen sofortigen Abbruch aller laufenden Funktionen und einen Warmstart des Controllers. Alle Parameter werden in den Grundzustand gesetzt. Zum Schluß wird der Ident-Text ausgegeben. Dieser Befehl hat dieselbe Wirkung wie ATZ, wird jedoch wesentlich schneller ausgeführt, da der Selbsttest übersprungen wird.

**Ausgabe:**

DIAMEX DX35 v1.0

&gt;

Hinweis: Der Text ist abhängig von der Bios-Version.

**>ATI**

Hiermit wird nur der Identifikationstext ausgegeben, ohne einen Warmstart durchzuführen.

**Ausgabe:**

DIAMEX DX35 v1.0

&gt;

Hinweis: Der Text ist abhängig von der Bios-Version.

**>ATD**

Alle Parameter werden in den Grundzustand wie nach einem Kalt- oder Warmstart versetzt. Sehr sinnvoll, wenn man mal viele Parameter verstellt hat und keine Daten mehr empfangen werden können.

**Ausgabe:**

OK

&gt;

**>ATE0****>ATE1**

Dieser Befehl schaltet das serielle Echo ein (1) oder aus (0). Alle Daten, die über die serielle Schnittstelle empfangen werden, werden bei eingeschaltetem Echo wieder zum PC geschickt.

**Ausgabe:**

OK

&gt;

Im Auslieferungszustand ist E1 voreingestellt.

**>ATL0****>ATL1**

Über diesen Befehl wird die Übermittlung des Linefeed-Zeichens am Ende der Zeile ein- (1) oder ausgeschaltet (0). Alle zum PC gesendeten Antworten sind in der Regel nur mit einem Carriage-Return abgeschlossen (13 Dez., \$0D Hex). Mit eingeschaltetem Linefeed wird hinter jedem Carriage-Return noch das Linefeed-Zeichen (10 Dez, \$0A Hex) ausgegeben. Besonders wenn man mit einem Terminalprogramm die Befehl von Hand übermittelt, ist es recht sinnvoll, den Linefeed eingeschaltet zu lassen, da sonst alle Antworten in einer Zeile dargestellt werden und neue Antworten die Alten überschreiben.

**Ausgabe:**

OK

&gt;

Im Auslieferungszustand ist L1 voreingestellt.

**>ATM0****>ATM1**

Schaltet die Memory-Funktion des letzten Protokolls ein bzw. aus.

Wenn häufig dasselbe OBD2-Protokoll verwendet wird, kann es sinnvoll sein, dieses als Voreinstellung im internen EEPROM des DX35 abzuspeichern. Hierzu schalten Sie die Memory-Funktion mit ATM1 ein, dann stellen Sie die Verbindung mit dem Fahrzeugsteuergerät her, so daß ein Protokoll aktiviert wird, zum Schluß schalten Sie die Memory-Funktion mit ATM0 wieder aus. Ab sofort wird das aktuelle Protokoll als Standard benutzt.

Nähere Informationen zu den verschiedenen Protokollen finden Sie im Abschnitt „OBD2-Protokolle“.

Im Auslieferungszustand ist M0 voreingestellt.

**>ATH0****>ATH1**

Mit diesem Befehl kann eingestellt werden, ob bei OBD2-Antworten der Header und das Checksummenbyte mit ausgegeben werden soll.

**Ausgabe:**

Mit ATH0:

```
>0100
41 00 E8 19 30 12
```

Mit ATH1:

```
>0100
48 6B 10 41 00 E8 19 30 12 47
```

48, 68, 10 sind die 3 Headerbytes  
47 ist das Checksummenbyte

**Ausgabe:**

OK

>

Da beim CAN-Protokoll keine Header- und Checksummenbytes existieren, wird stattdessen die CAN-ID übermittelt. Nähere Informationen hierzu finden Sie in der Beschreibung zum CAN-Protokoll später in diesem Dokument.

Im Auslieferungszustand ist H1 voreingestellt.

**>ATBD**

Die Abkürzung von „**B**uffer **D**ump“ bewirkt, den internen OBD2-Empfangsspeicher auszugeben. Gültige Daten stehen hier jedoch nur, wenn zuvor ein OBD2-Befehl ausgeführt wurde.

**Beispiel:**

```
>0100
41 00 E8 19 30 12
```

**>ATBD**

```
0A 48 6B 10 41 00 E8 19 30 12 47 00 00
```

Das erste Byte gibt die Anzahl der gültigen Zeichen im Speicher an. In diesem Fall handelt es sich um 10 Zeichen (Hex 0A). Das letzte Byte ist in diesem Fall ungültig und kann einen beliebigen Wert beinhalten.

**>ATB**

Formatierte Anzeige des OBD2-Empfangsspeichers. Im Gegensatz zum vorherigen Befehl ATBD, dessen Ausgabelänge auf 12 Zeichen begrenzt ist, kann mit ATB der gesamte genutzte Speicher angezeigt werden.

**Beispiel:**

```
>0100
41 00 E8 19 30 12
```

**>ATB**

```
0A 01
48 6B 10 41 00 E8 19 30 12 47
```

In der ersten Ausgabezeile wird die gesamte Anzahl der im OBD2-Buffer stehenden Daten angezeigt (0A = 10 Bytes). Der 2. Wert gibt die Anzahl der empfangenen Frames an. In diesem Fall wurde nur 1 Frame empfangen.

**Bios Version 1.0**

---

**>ATSR xx**

Eingabe der RX-Adresse für OBD2-Antworten.

Über diesen Befehl wird die Adresse für die empfangenen OBD2-Pakete festgelegt. Dieser Wert ist abhängig vom Protokoll und wird in der Regel automatisch eingestellt. Bei KWP2000 sollte der Wert identisch zur Tester-Adresse sein (F1) damit die passenden Antwortdaten empfangen werden können. Bei ISO-9141, PWM und VPWM muss der Wert gleich mit dem 2. Byte des Requestheaders + 1 sein.

**Beispiel (ISO9141 Functional Request):**

Request Header    68 6A F1  
Antwort            48 6B 10

Werden hier falsche Werte eingegeben, können entweder falsche oder gar keine Pakete empfangen werden.

Die Filteradresse kann durch Eingabe von ATZ, ATWS, ATD oder ATSR00 ausgeschaltet werden.

Dieser Parameter hat keine Funktion beim CAN-Protokoll.

**>ATN**

Anzeige des aktuell benutzten Protokolls als Hexwert. F0..F9

**Beispiel:**

>ATN  
F2

>

00:    Kein Protokoll aktiv  
01:    PWM-Protokoll  
02:    VPWM-Protokoll  
03:    ISO9141-Protokoll  
04:    KWP2000-Protokoll (5 Baud Init)  
05:    KWP2000-Protokoll (Fast Init)  
06:    CAN-Protokoll 11Bit-ID, 500kBaud  
07:    CAN-Protokoll 29Bit-ID, 500kBaud  
08:    CAN-Protokoll 11Bit-ID, 250kBaud  
09:    CAN-Protokoll 29Bit-ID, 250kBaud

Eine Ausgabe von F0 bedeutet, daß zur Zeit kein Protokoll benutzt wird (zum Beispiel nach ATZ).

**>ATDP**

Anzeige des aktuell benutzten Protokolls im Klartext.

**Beispiel:**

>ATDP  
ISO 9141-2

>

Hier die Liste aller möglichen Ausgaben:

NOT CONNECTED  
SAE J1850 / PWM  
SAE J1850 / VPWM  
ISO 9141-2  
ISO 14230-4, KWP2000 (5 Baud Init)  
ISO 14230-4, KWP2000 (Fast Init)  
ISO 15765-4, CAN (11/500)  
ISO 15765-4, CAN (29/500)  
ISO 15765-4, CAN (11/250)  
ISO 15765-4, CAN (29/250)

**Bios Version 1.0****>ATP [A] x**

Manuelle Einstellung des aktuellen Protokolls oder der automatischen Protokoll-Suchfunktion.

Hiermit kann ein Protokoll fest voreingestellt werden. Es wird kein anderes Protokoll gesucht, sondern mit einer genauen Fehlermeldung abgebrochen, wenn das Steuergerät nicht antwortet.

ATP1 PWM  
 ATP2 VPWM  
 ATP3 ISO9141-2  
 ATP4 KWP2000 5 Baud Init  
 ATP5 KWP2000 Fast Init  
 ATP6 CAN 11/500  
 ATP7 CAN 29/500  
 ATP8 CAN 11/250  
 ATP9 CAN 29/250

Wenn statt dessen **ATPA<sub>x</sub>** (x = Protokollnummer) eingegeben wird, durchsucht der Controller automatisch alle anderen Protokolle, wenn das Aktuell eingestellte Protokoll nicht gefunden wird.

Mit **ATP0** oder **ATPA0** (identisch) wird die automatische Suchfunktion aktiviert. Es ist kein Protokoll voreingestellt und es werden nach einem Neustart alle Protokolle durchsucht, bis ein passendes gefunden ist.

**Vorsicht!**

Wenn ein festes Protokoll ohne Auto-Suchfunktion voreingestellt ist, wird auch kein anderes Protokoll gesucht, wenn das Steuergerät im Fahrzeug nicht auf das Eingestellte antwortet. In diesem Fall bitte die Suchfunktion mit **ATP0** oder **ATPA<sub>x</sub>** (x = aktuelles Protokoll) aktivieren.

Ist die Memory-Funktion mit **ATM1** eingeschaltet, wird die Änderung der Einstellung sofort ins EEPROM abgespeichert und bei Neustart wieder verwendet. Durch Eingabe von **ATP** ohne zusätzliche Parameter wird der eingestellte Modus angezeigt. In diesem Fall wird keine Änderung durchgeführt.

Im Auslieferungszustand ist **ATP0** voreingestellt. Es werden alle Protokolle durchsucht.

**>ATSH xx yy zz**

Manuelles Setzen der Headerbytes für ISO9141, KWP2000, PWM und VPWM Protokolle.

Es werden 3 Hexwerte erwartet:

xx = Priority/Type-Byte  
 yy = Target-Address  
 zz = Source-Address

Den genauen Aufbau der 3 Bytes entnehmen Sie bitte den SAE-Protokoll-Spezifikationen.

Bei KWP2000 wird die Längenangabe im 1. Byte (xx) automatisch angepasst. Beim schwedischen KWP2000-Format wird im 4. Headerbyte die Länge eingesetzt.

Wenn das Protokoll mit ATP<sub>x</sub> festgelegt ist, können die Headerbytes bereits vor dem 1. Verbindungsaufbau geändert werden. Ist die Auto-Suchfunktion aktiviert, werden für den Verbindungsaufbau immer die Standardwerte benutzt, hier können die Headerbytes nur nach erfolgtem Connect geändert werden.

**Standardwerte:**

ISO9141-2: 68 6A F1  
 KWP2000: C<sub>x</sub> 33 F1 (x = Längenangabe)  
 PWM: 61 6A F1  
 VPWM: 68 6A F1

Die Headerbytes können durch Eingabe von **ATZ**, **ATWS** oder **ATD** auf die Standardwerte zurückgesetzt werden.

Dieser Parameter hat keine Funktion beim CAN-Protokoll.

**>ATK**

Anzeige der Keyword-Bytes bei ISO-9141 und KWP2000.

**Beispiel:**

```
>ATK
8F E9
```

>

Dient zur Analyse den unterstützten Keyword-Protokolls. Die Bedeutung der Keyword-Bytes entnehmen Sie bitte den SAE-Datenblättern zu ISO9141-2 bzw. KWP2000.

**Bios Version 1.0**

---

**>ATKW0****>ATKW1**

Bei der Protokollsuche werden die Protokolle ISO9141-2 und KWP2000 anhand der gelieferten Keyword-Bytes unterschieden. Für Experimentierzwecke kann diese Funktion deaktiviert werden und das mit ATPx fest eingestellte Protokoll erzwungen werden.

*Beispiel:*

```
>ATKW1
OK
```

>

Im Auslieferungszustand ist KW1 voreingestellt.

**>ATWM xx yy zz aa [bb] [cc]**

Manuelles Setzen der Wakeup-Message-Bytes für ISO9141 und KWP2000.

Es werden 4-6 Hexwerte erwartet:

xx = Priority/Type-Byte (mit Längenangabe bei KWP)

yy = Target-Address

zz = Source-Address

aa,bb,cc = 1-3 Befehlsbytes

Bei KWP2000 muss das 1. Byte die richtige Längenangabe enthalten.

*Beispiel (ISO9141-2):*

```
>ATWM 68 6A F1 03
OK
```

*Beispiel (KWP2000):*

```
>ATWM C2 33 F1 01 04
OK
```

**Standardwerte:**

ISO9141-2: 68 6A F1 01 00

KWP2000: C1 33 F1 3E

Die Wakeup-Message-Bytes können durch Eingabe von ATZ, ATWS oder ATD auf die Standardwerte zurückgesetzt werden.

Dieser Parameter hat keine Funktion beim PWM, VPWM und CAN-Protokoll.

**>ATSW xx**

Mit diesem Befehl kann die Zeitdauer zwischen den automatischen Wakeup-Befehlen bei einer bestehenden ISO9141 oder KWP2000 Verbindung eingestellt werden.

ISO9141 und KWP2000 Fahrzeugcontroller erwarten einen regelmäßigen Datenverkehr zum angeschlossenen OBD2-Interface. Sollte dieser längere Zeit ausbleiben (laut SAE-Norm 5000ms), wird die Verbindung getrennt und muss durch einen erneuten Init wieder hergestellt werden.

Dieser Befehl erwartet die Zeitdauer als 2-stelligen HEX-Code. Die Zeitdauer zwischen den Wakeup-Befehlen ergibt sich aus dem Hex-Wert \* 50 Millisekunden.

*Beispiel:*

```
>ATSW32
OK
```

>

Hier wird eine Wakeup-Zeit von 0x3C (60) \* 50ms = 3 Sekunden eingestellt.

Der Standardwert wird mit ATZ, ATWS oder ATD auf 0x32 (50) eingestellt, was 2,5 Sekunden entspricht. Wird als Wert 0 eingetragen, ist der automatische Wakeup abgeschaltet und die Verbindung wird nach 5 Sekunden getrennt, wenn keine OBD2-Kommandos übertragen werden.

**Bios Version 1.0****>ATCA0****>ATCA1**

Mit diesem Befehl kann die automatische Formatierung der gesendeten und empfangenen CAN-Datenpakete ein- bzw. ausgeschaltet werden.

Ein Ausschalten der automatischen Formatierung ist nur sinnvoll, wenn DX35 in CAN-Bussen betrieben werden soll, die nicht OBD2-kompatibel sind.

Ist die Formatierung ausgeschaltet, müssen immer alle gewünschten Bytes (max. 8) des CAN-Datenpaketes eingegeben werden, bei eingeschalteter Formatierung wird automatisch das Längenbyte hinzugefügt und es dürfen max. 7 Bytes zum Senden eingegeben werden.

*Beispiel:*

Mit Auto-Format an:

```
>0100
wird umgewandelt nach:
0201000000000000
```

Im OBD2-Modus werden immer Datenpakete mit 8 Bytes versendet. Folgender Befehl ist gleichbedeutend mit obigem Beispiel, jedoch bei ausgeschaltetem Auto-Format:

```
>0201000000000000
```

Wird statt dessen nur

```
>020100
eingegeben, wird nur ein Datenpaket mit 3 Bytes gesendet, das nicht OBD2-konform ist.
```

Die Einstellung der Auto-Formatierung beeinflusst ebenfalls die Ausgabe der empfangenen Antworten auf dem CAN-Bus.

*Beispiel:*

Mit Auto-Format aus werden alle 8 Bytes des CAN-Frame ausgegeben. Es wird keine Auswertung der empfangenen Daten vorgenommen:

```
06 41 00 B8 7B B0 10 00
```

Mit Auto-Format an wird das empfangende OBD2-Datenpaket ausgewertet. Das 1. Byte zeigt in diesem Fall an, daß 6 gültige Bytes folgen. Damit wird folgendes ausgegeben:

```
41 00 B8 7B B0 10
```

In Multiframe-Antworten ist die Ausgabe der Datenpakete mit eingeschalteter Formatierung an das Format der ELM-Controller angepasst um die Softwarekompatibilität zu gewährleisten.

*Beispiel, mit Auto-Format an.**Abfrage der Fehlercodes mit 4 Fehlern:*

```
>03
00A
0: 43 04 01 15 02 30
1: 03 50 04 60 00 00 00
```

Die 1. Zeile zeigt an, daß die Antwort aus 00A (hex) = 10 gültigen Bytes besteht. Jede weitere Zeile beginnt mit einem Zähler mit anschließendem Doppelpunkt, der die Reihenfolge der empfangenen Datenpakete angibt. Es wird von 0 bis F (hex) gezählt und beginnt dann wieder bei 0, wenn noch mehr Pakete übertragen werden müssen.

Mit ausgeschalteter Formatierung sieht die Antwort folgendermaßen aus:

```
10 0A 43 04 01 15 02 30
21 03 50 04 60 00 00 00
```

Hier müssen PCI- und Längenbyte von der PC-Software ausgewertet werden.

Mehr Infos zum OBD2-Datenformat finden Sie in den SAE J1979 – Spezifikationen.

Standardeinstellung: Ein (ATCA1).

**>ATCC0****>ATCC1**

In OBD2-Systemen müssen bei Multiframe-Antworten so genannte Flow-Control Nachrichten vom Tester gesendet werden, die dem Steuergerät anzeigen, daß nachfolgende Pakete akzeptiert werden. Dieses wird durch den DX35 automatisch vorgenommen.

Soll der Controller in CAN-Systemen eingesetzt werden, die nicht OBD2-konform sind, kann es sinnvoll sein, diese automatischen Flow-Control Nachrichten auszuschalten.

*Beispiel:*

```
>ATCC0
OK
```

```
>
```

Standardeinstellung: Ein (ATCC1).

### >ATCD xx

Flow-Control Datenpakete, die bei Multiframe-Antworten gesendet werden müssen, enthalten neben einem Statusbyte (FS), einem Blockgrößenbyte (BS) auch ein Byte für die Zeitdauer (ST), die zwischen den nachfolgenden Antwortpaketen eingefügt werden soll, damit der CAN-Controller Zeit hat, diese Daten zu verarbeiten. Mit diesem Befehl kann die Zeitdauer verändert werden.

*Beispiel:*

```
>ATCD 7F
OK
```

>

Stellt die Zeitdauer auf den maximalen Wert von 127ms ein.

Dieser Befehl ist nur zu Experimentierzwecken vorhanden und kann zum Testen von Simulatoren verwendet werden. In der Regel muss die Einstellung nicht verändert werden.

Standardwert: 0A (10 ms)

### >ATCI xxx

### >ATCI xxxxxxxxx

Die zu sendende CAN-ID wird mit diesem Befehl gesetzt. Dies kann sinnvoll sein, wenn der DX35 in nicht OBD2-konformen CAN-Systemen eingesetzt wird.

Standardwerte:

11 Bit ID	7DF
29 Bit ID	18 DB 33 F1

*Beispiele:*

```
>ATCI 7E0
OK
```

>

```
>ATCI 18 DA 10 F1
OK
```

>

Bitte darauf achten, daß durch die Anzahl der eingegebenen Zeichen bestimmt wird, ob die 11Bit oder die 29Bit ID verändert werden soll. Dies wird durch die vorherige Wahl des Protokolls vorgegeben. Für die 11Bit ID müssen immer 3 Hex-Zeichen eingegeben für die 29Bit ID müssen immer 8 Hex-Zeichen eingegeben werden.

Durch Eingabe von ATWS, ATZ oder ATD werden die ID auf die Standardwerte zurückgesetzt.

## Bios Version 1.0

>ATCF xxx

>ATCF xxxxxxxx

CAN-RX-Filter setzen. Wenn zu viele Daten auf dem CAN-Bus übertragen werden, kann es passieren, daß der Controller-interner Puffer überläuft wenn die Daten nicht rechtzeitig zum PC übertragen werden können. In diesem Fall sollten aus den empfangenen Daten die gewünschten ausgefiltert werden. Der Filter wird zusammen mit der RX-Maske verwendet, die mit dem Befehl ATCM verändert werden kann.

Standardwerte:

11 Bit Filter	7E8
29 Bit Filter	18 DA F1 00

*Beispiele:*

>ATCF 7E0  
OK

>  
>ATCI 18 DA F1 10  
OK

>

Bitte darauf achten, daß durch die Anzahl der eingegebenen Zeichen bestimmt wird, ob der 11Bit oder der 29Bit Filter verändert werden soll. Dies wird durch die vorherige Wahl des Protokolls vorgegeben. Für den 11Bit Filter müssen immer 3 Hex-Zeichen eingegeben für den 29Bit Filter müssen immer 8 Hex-Zeichen eingegeben werden.

Durch Eingabe von ATWS, ATZ oder ATD werden die Filter auf die Standardwerte zurückgesetzt.

>ATCM xxx

>ATCM xxxxxxxx

CAN-RX-Maske setzen. In Verbindung mit dem RX-Filter (ATCF) kann die Maske dazu benutzt werden, einzelne oder eine Gruppe von Daten auszufiltern.

Standardwerte:

11 Bit Maske	7F8
29 Bit Maske	1F FF FF 00

*Beispiele:*

>ATCM 7F0  
OK

>

>ATCM 1F FF 00 00  
OK

>

In Verbindung mit dem Filter gibt ein 1-Bit in der Maske an, ob die ankommende Nachricht mit dem Filter verglichen werden soll. Wenn das Maskenbit 0 ist, wird dieses Bit als „ok“ angenommen.

Bitte darauf achten, daß durch die Anzahl der eingegebenen Zeichen bestimmt wird, ob die 11Bit oder die 29Bit Maske verändert werden soll. Dies wird durch die vorherige Wahl des Protokolls vorgegeben. Für die 11Bit Maske müssen immer 3 Hex-Zeichen eingegeben für die 29Bit Maske müssen immer 8 Hex-Zeichen eingegeben werden.

Durch Eingabe von ATWS, ATZ oder ATD werden die Masken auf die Standardwerte zurückgesetzt

## Bios Version 1.0

### >AT!00

Ausgabe der Seriennummer des DX35. Alle DX35 besitzen eine einzigartige nicht veränderbare Seriennummer. Sie kann mit diesem Befehl abgefragt werden.

*Beispiel:*

```
>AT!00  
A123456789
```

>

### >AT!01

Ausgabe des Controllertyps und der Bios-Versionsnummer. Da der Identifikations-String veränderbar ist, ist keine eindeutige Identifizierung des Controllertyps und der Bios-Version über den ATI Befehl möglich. Aus diesem Grund wurde dieser Befehl eingefügt, dessen Ausgabe unveränderlich ist und immer den richtigen Controller-Typ anzeigt.

*Beispiel:*

```
>AT!01  
DX35-10
```

>

Obiges Beispiel zeigt die Ausgabe eines DX35 mit Bios-Version 1.0

Weitere Befehle, speziell für die Änderung des Identifikationstextes und zum internen Test des DX35 bei der Herstellung werden hier nicht aufgeführt. Interessenten können diese Informationen unter Angabe des Verwendungszwecks beim Entwickler der DX35 per E-Mail anfordern.

## Direkteingabe eines OBD2-Befehls

Zur Eingabe eines OBD2-Befehls werden nur die für diesen Befehl notwendigen Daten im Hexcode übergeben. Sollte zuvor noch keine Verbindung mit dem OBD2-Bus im Fahrzeug aufgebaut worden sein, wird dies bei Eingabe des ersten Befehls einmalig vorgenommen. Die verschiedenen Protokolle werden abhängig von der Protokoll Einstellung abgefragt (siehe Abschnitt „OBD2-Protokolle“)

Sobald der Fahrzeugcontroller eine Antwort bei Anfrage mit einem der Protokolle liefert, wird der weitere Test abgebrochen und das erkannte Protokoll auch für alle weiteren Anfragen benutzt. Die Dauer bis zur Ausführung des ersten Befehls dauert somit maximal 2,5 Sekunden, bedingt durch die lange Antwortzeit beim Slow-Init. VPWM, PWM und CAN wird innerhalb 1 Sekunde erkannt.

*Beispiel:*

Ausgabe bei ISO9141 oder KWP2000 Connect (Header aus, ATH0)

```
>0100  
SEARCHING...  
41 00 E8 19 30 12
```

>

```
>0100  
41 00 E8 19 30 12
```

*Beispiel:*

Ausgabe bei VPWM, PWM oder CAN Connect.

```
>0100  
41 00 E8 19 30 12
```

>

**Übersicht der AT-Befehle des DX35-Controllers****Allgemeine Befehle:**

<b>ATD</b>	Alle Werte zurücksetzen
<b>ATE0</b>	Echo aus
<b>ATE1</b>	Echo ein (standard)
<b>ATI</b>	Identifizierungstext ausgeben
<b>ATL0</b>	Linefeed aus (standard)
<b>ATL1</b>	Linefeed ein
<b>ATM0</b>	Protokoll-Speicher aus (standard)
<b>ATM1</b>	Protokoll-Speicher ein
<b>ATWS</b>	Warmstart
<b>ATZ</b>	Reset
<b>AT!00</b>	Ausgabe der Seriennummer
<b>AT!01</b>	Ausgabe der Chip-Kennung

**OBD2 Befehle:**

<b>ATBD</b>	Buffer Dump
<b>ATB</b>	Formatierte Ausgabe des Buffers
<b>ATDP</b>	Aktuelles Protokoll im Klartext ausgeben
<b>ATH0</b>	Header aus (standard)
<b>ATH1</b>	Header ein
<b>ATN</b>	Aktuelles Protokoll als Nummer ausgeben
<b>ATP</b>	Aktuelle Protokollkonfiguration anzeigen
<b>ATP x</b>	Protokoll x voreinstellen
<b>ATP Ax</b>	Protokoll x voreinstellen, mit Auto-Suchfunktion
<b>ATP0</b>	Immer Protokoll Auto-Suchfunktion benutzen
<b>ATSH xx yy zz</b>	Header definieren
<b>ATSR xx</b>	RX-Empfangsadresse einstellen

**ISO/KWP2000-spezifische Befehle:**

<b>ATSW xx</b>	Wakeup Intervall-Zeit einstellen
<b>ATWM xx yy zz aa [bb] [cc]</b>	Wakup Message setzen
<b>ATK</b>	Anzeige des der Keyword-Bytes
<b>ATKW0</b>	Keine Überprüfung des ISO/KWP Keywordes beim Verbindungsaufbau
<b>ATKW1</b>	Überprüfung des ISO/KWP Keywordes beim Verbindungsaufbau (standard)

**CAN-spezifische Befehle:**

<b>ATCA0</b>	CAN Autoformat aus
<b>ATCA1</b>	CAN Autoformat ein (standard)
<b>ATCC0</b>	CAN Flow Control aus
<b>ATCC1</b>	CAN Flow Control ein (standard)
<b>ATCD xx</b>	CAN Flow Delay setzen
<b>ATCF xxx</b>	CAN RX Message Filter 11 Bit
<b>ATCF xxxxxxxx</b>	CAN RX Message Filter 29 Bit
<b>ATCI xxx</b>	CAN TX Message Identifier 11 Bit
<b>ATCI xxxxxxxx</b>	CAN TX Message Identifier 29 Bit
<b>ATCM xxx</b>	CAN RX Message Mask 11 Bit
<b>ATCM xxxxxxxx</b>	CAN RX Message Mask 29 Bit

**OBD2-Protokolle**

Der DX35 Controller unterstützt alle derzeit aktuellen OBD2-Protokolle. Alle 9 verschiedenen Protokolle können damit eindeutig identifiziert und selektiert werden.

Modus	Protokoll
1	SAE J1850 PWM
2	SAE J1850 VPWM
3	ISO 9141-2
4	ISO 14230-4 KWP2000 5 Baud Init
5	ISO 14230-4 KWP2000 Fast Init
6	ISO 15765-4 CAN 11Bit-ID, 500 kBaud
7	ISO 15765-4 CAN 29Bit-ID, 500 kBaud
8	ISO 15765-4 CAN 11Bit-ID, 250 kBaud
9	ISO 15765-4 CAN 29Bit-ID, 250 kBaud

Im Auslieferungszustand ist der DX35 so konfiguriert, daß automatisch alle Protokolle durchsucht werden, bis ein passendes gefunden ist (ATP0). Sollte kein passendes Protokoll gefunden werden, wird folgender Text ausgegeben:

```
UNABLE TO CONNECT
```

Die Protokolle werden in folgender Reihenfolge getestet:

1	1: PWM
2	2: VPWM
3	5: KWP2000 Fast Init
4	3+4: ISO+KWP 5-Baud Init
5	6: CAN 11/500
6	7: CAN 29/500
7	8: CAN 11/250
8	9: CAN 29/250

Sobald ein passendes Protokoll gefunden wird, wird der Suchvorgang abgebrochen und das gefundene Protokoll wird ab sofort bei allen weiteren Befehlen benutzt. Das aktuelle Protokoll kann mit dem Befehl ATPD angezeigt werden.

```
>ATDP
SAE J1850 VPWM
```

```
>
```

Gleichzeitig wird das gefundene Protokoll als Suchstart benutzt, wenn die Verbindung zuvor mit ATWS oder ATZ geschlossen wurde. Dies kann mit dem Befehl ATP angezeigt werden.

```
>ATP
AUTO 2 = SAE J1850 VPWM
```

```
>
```

Ist die Protokoll Memory Funktion aktiviert (ATM1), wird das aktuelle Protokoll im EEPROM des Controllers abgespeichert und bleibt auch nach Trennung der USB-Stromversorgung erhalten.

Nach erneutem Anlegen der Stromversorgung wird der Inhalt des EEPROM ausgelesen und das gespeicherte Protokoll als erstes für den erneuten Suchvorgang benutzt. Wurde das gespeicherte Protokoll nicht gefunden, werden die Protokolle in der zuvor angegebenen Reihenfolge getestet, dabei wird jedoch das bereits als negativ getestete Protokoll übersprungen.

Soll der Suchvorgang gezielt mit einem gewünschten Protokoll beginnen, kann dies auch manuell eingegeben werden:

```
>ATPA5
AUTO 5 = ISO 14230-4 KWP2000 Fast Init
```

```
>
```

Hier wird der Suchvorgang mit dem Protokoll 5 begonnen. Wird das Protokoll nicht gefunden, werden die Protokolle in der Reihenfolge der Suchliste getestet.

Soll der DX35-Controller immer nur ein bestimmtes Protokoll benutzen, kann dies durch folgenden Befehl festgelegt werden:

```
>ATP5
5 = ISO 14230-4 KWP2000 Fast Init
```

```
>
```

Hier wird nur noch das Protokoll 5 benutzt. Wird das Protokoll nicht gefunden, wird die Suche abgebrochen und UNABLE TO CONNECT oder eine andere Fehlermeldung ausgegeben. Ist die Protokoll Memory Funktion eingeschaltet, wird dieses Protokoll im EEPROM abgespeichert und auch nach Trennung und erneutem Anlegen der Stromversorgung wieder benutzt. Deshalb Achtung! Wenn keine Verbindung aufgebaut werden kann, zunächst mit ATP schauen, welches Protokoll voreingestellt ist.

## **Fehlermeldungen**

Viele der im Folgenden aufgeführten Fehlermeldungen treten nur während des ersten Verbindungsaufbaus auf und erscheinen nur, wenn ein festes Protokoll mit `ATPx`

voreingestellt ist. Im automatischen Protokoll-Suchmodus werden die Fehlermeldungen unterdrückt um mit dem Test des nächsten Protokolls fortzufahren.

### **BUS BUSY**

Tritt bei PWM und VPWM auf, wenn ein Befehl innerhalb der vorgeschriebenen Zeitspanne nicht gesendet werden konnte.

### **NO DATA**

Wird ausgegeben, wenn ein OBD2-Befehl nicht innerhalb der vorgeschriebenen Zeitspanne beantwortet wurde, zum Beispiel weil er im Steuergerät nicht unterstützt wird.

### **FEEDBACK ERROR**

Bei der Aktivierung des ISO/KWP-, PWM- oder VPWM-Busses wird sofort getestet, ob eine Rückmeldung am Eingang des Controllers erscheint. Diese Fehlermeldung zeigt an, daß keine Rückmeldung erfolgt ist.

### **UNABLE TO CONNECT**

Erscheint, wenn keine Verbindung zu einem Steuergerät aufgebaut werden konnte.

### **DATA ERROR**

Zeigt einen Checksummenfehler beim ISO, KWP2000, PWM oder VPWM Protokoll an.

### **ERROR (nn)**

Verschiedene Fehlermeldungen beim Verbindungsaufbau mit einem festen Protokoll.

### **CAN ERROR**

Fehler im CAN-Protokoll. Da das CAN-Protokoll relativ sicher gegen Übertragungsfehler auf dem Bus ist, wird diese Meldung nur ausgegeben, wenn die Verbindung zum CAN-Bus getrennt wird.

### **?**

Syntax-Fehler. Wird ausgegeben, wenn der AT-Befehl nicht existiert oder wenn zum Befehl gehörende Parameter fehlen oder falsch eingegeben wurden.

### Hinweise

© Erwin Reuß; Folker Stange. Nutzung und Weitergabe dieser Informationen auch Auszugsweise nur mit Erlaubnis der Copyright-Inhaber.

Änderungen am Bios und Befehlssatz des Controllers vorbehalten.

DIAMEX® ist eingetragenes Warenzeichen.

Alle Markennamen, Warenzeichen und eingetragenen Warenzeichen sind Eigentum Ihrer rechtmäßigen Eigentümer und dienen hier nur der Beschreibung.

### Haftungshinweis

Der Hersteller übernimmt keine Haftung für Schäden die durch Anwendung der DIAMEX Interfaces und der Diagnose-Software entstehen könnten.

### Kontakt, Forum, Software, Updates

<http://www.diamex.de>

<http://www.obd-diag.de>