

### ACHTUNG!

Bei diesem Dokument handelt es sich um eine Vorabversion der Beschreibung des AT-Befehlssatzes des Diamex-Wireless-Interfaces DX60/61.

Ergänzungen sowie die Korrektur eventueller Fehler werden bis zur Serienfertigung des Produktes vorgenommen.



**Befehlssatz/Kommunikation mit dem PC**

Die Kommunikation des PC mit den DX61-USB-Stick geschieht über eine virtuelle serielle Schnittstelle, die beim installieren des FTDI-Treibers automatisch eingerichtet wird. Welcher COM-Port eingerichtet wurde, stellen Sie am besten mit einem Terminalprogramm oder im Gerätemanager in der Systemsteuerung von Windows fest. Es kann mit jedem Standard-Terminalprogramm oder speziellen OBD2-Programmen auf das Wireless-Interface zugegriffen werden. Die Datenrate ist aufgrund des virtuellen COM-Ports ohne Bedeutung, es wird immer mit der höchst möglichen USB-Rate 12MBit übertragen.

Wenn Sie ein Terminalprogramm auf dem PC starten, und den passenden COM-Port wählen, sollte nach Eingabe des Befehls

```
ATZ
```

auf dem PC folgender Text erscheinen:

```
DIAMEX DX61 v1.0
```

```
>
```

Hinweis: Diese Meldung variiert je nach Version und kann zusätzlich mit dem Tool „AGV-Internals“ geändert werden.

Welche Meldung auch erscheint, sie zeigt an, daß die Kommunikation des Interface mit dem PC funktioniert. 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 si
    Wird intern nach ATSI gewandelt
A T Z
    Wird intern nach ATZ gewandelt
01 1c
    Wird intern nach 011C gewandelt
```

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

Befehle, die der Controller nicht versteht, werden ebenfalls mit einem „?“ als Fehlermeldung quittiert.

Einige interne AT-Befehle sowie die Dateneingabe für den OBD2-Bus werden mit Werten übermittelt, die aus 2-stelligen HEX-Zeichen zusammengesetzt sind. Auch wenn der zu übermittelnde Wert nur einstellig ist muß er immer 2-stellig mit führender Null eingegeben werden.

*Beispiele:*

```
ATST 80
    Richtig! Der Wert $80 wird übermittelt
ATSW8
    Falsch! Es muß immer eine grade Anzahl an
    Hexzeichen eingegeben werden.
```

**ACHTUNG!**

Einige AT-Befehle werden direkt im Stick umgewandelt oder verarbeitet, andere werden an das DX60-OBD2 Interface weitergeschickt. Falls keine Funkverbindung zum OBD2-Interface besteht (erkennbar am wechselseitigen blinken der Leuchtdioden auf dem Stick) wird als Antwort auf die Befehle nur

```
NO ACK
```

ausgegeben.

**Der AT Befehlssatz des DX60/61-Interfaces:****>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 DX61 v1.0

&gt;

**>ATWS**

Dieser Befehl hat dieselbe Funktion wie ATZ und wurde nur aufgrund der Kompatibilität übernommen.

**>ATI**

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

Laufende Funktionen, wie das automatische Wakeup bei ISO oder KWP2000, werden nicht abgebrochen.

*Ausgabe:*

DIAMEX DX61 v1.0

&gt;

**>ATD**

Alle Parameter werden in den Grundzustand wie nach einem Kalt- oder Warmstart versetzt.

*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;

Grundeinstellung nach ATZ: E0

**>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;

Voreinstellung nach ATZ: L0

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

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

*Ausgabe:*

Mit ATH0:

&gt;0100

41 00 E8 19 30 12

Mit ATH1:

&gt;0100

48 6B 10 41 00 E8 19 30 12 47

48, 68, 10 sind die 3 Headerbytes

47 ist das Checksummenbyte

*Ausgabe:*

OK

&gt;

Voreinstellung nach ATZ: H0

**OBD2 Wireless Interface, AT-Befelssatz****>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.

**>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 SLOW- oder FAST-Init wieder hergestellt werden.

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

*Beispiel:*

```
>ATSW40
OK
```

```
>
```

Hier wird eine Wakeup-Zeit von \$40 (64) \* 50ms = 3,2 Sekunden eingestellt.

Der Standardwert wird mit ATZ und ATD auf \$32 (50) eingestellt, was genau 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.

**>ATSR xx**

Eingabe der RX-ECU-Filteradresse für OBD2-Antworten.

Falls bei einem „Functional Request“ mehrere ECU antworten, kann durch setzen des RX-Filters das gewünschte ausgefiltert werden.

*Beispiel (KWP2000 mit eingeschalteten Header):*

```
>0100
C6 F1 10 41 00 B8 7B B0 10 FB
C6 F1 18 41 00 08 28 00 00 40
```

Hier antworten 2 verschiedene Steuergeräte. Sollen nur die Antworten des 2. Steuergerätes (18) ausgefiltert werden, kann dies durch Eingabe des Befehls ATSR18 erzwungen werden.

Bei KWP2000 wird immer das 3. Byte (Target-Address) als Filteradresse benutzt. Bei ISO9141, PWM und VPW wird abhängig vom Requesttyp das 2. oder 3. Byte benutzt. Beim „Physical Request“ wird das 3. Byte, beim „Functional Request“ wird das 2. Byte + 1 benutzt.

*Beispiel (ISO9141 Functional Request):*

```
Request Header 68 6A F1
Antwort         48 6B 10
```

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

Dieser Parameter hat keine Funktion beim CAN-Protokoll.

**OBD2 Wireless Interface, AT-Befelssatz**

---

**>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.

Wenn das Protokoll mit ATPx 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: Cx 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.

**>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.

**OB2 Wireless Interface, AT-Befelssatz****>ATN**

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

*Beispiel:*

```
>ATN
F2
```

```
>
```

```
F0:   Kein Protokoll aktiv
F1:   PWM-Protokoll
F2:   VPWM-Protokoll
F3:   ISO9141-Protokoll
F4:   KWP2000-Protokoll (5 Baud Init)
F5:   KWP2000-Protokoll (Fast Init)
F6:   CAN-Protokoll 11Bit-ID, 500kBaud
F7:   CAN-Protokoll 29Bit-ID, 500kBaud
F8:   CAN-Protokoll 11Bit-ID, 250kBaud
F9:   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 alle 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)
```

**>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  
UNABLE TO CONNECT  
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  $ATP_{Ax}$  ( $x$  = Protokollnummer) eingegeben wird, durchsucht der Controller automatisch alle anderen Protokolle, wenn das Aktuelle nicht gefunden wird.

Mit  $ATP_0$  oder  $ATPA_0$  (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  $ATP_0$  oder  $ATPA_x$  ( $x$  = aktuelles Protokoll) aktivieren.

Ist die Memory-Funktion mit  $ATM_1$  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.

Mehr Informationen hierzu im Abschnitt „OB2-Protokolle“.

Im Auslieferungszustand ist  $ATP_0$  voreingestellt. Es werden alle Protokolle durchsucht.

### >ATV

Mit diesem Befehl wird eine Übersicht aller veränderbaren Parameter des DX60-OB2-Interfaces angezeigt.  
Diese Funktion ist nur für die Softwareentwicklung erforderlich. Die genaue Bedeutung der einzelnen Parameter ergeben sich aus der AT-Befehlstabelle.

### >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 der DX60 Controller in CAN-Bussen betrieben werden soll, die nicht OB2-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 OB2-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 OB2-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 OB2-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.

*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 OB2-Datenformat finden Sie in den SAE J1979 – Spezifikationen.

Voreinstellung: Ein (ATCA1).

### >ATCC0

### >ATCC1

In OB2-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 DX60 Controller automatisch vorgenommen.

Soll der Controller in CAN-Systemen eingesetzt werden, die nicht OB2-konform sind, kann es sinnvoll sein, diese automatischen Flow-Control Nachrichten auszuschalten.

#### Beispiel:

```
>ATCC0
OK
>
```

Voreinstellung: Ein (ATCC1).

## OB2 Wireless Interface, AT-Befelssatz

### >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.

Voreinstellung: 0A (10 ms)

### >ATCI **xxx**

### >ATCI **xxxxxxxx**

Die zu sendende CAN-ID wird mit diesem Befehl gesetzt. Dies kann sinnvoll sein, wenn der AGV4000 Controller 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. 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.

### >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. 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.

## OBD2 Wireless Interface, AT-Befelssatz

>ATCM xxx

OK

>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.

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.

Standardwerte:

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

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

**Beispiele:**

>ATCM 7F0

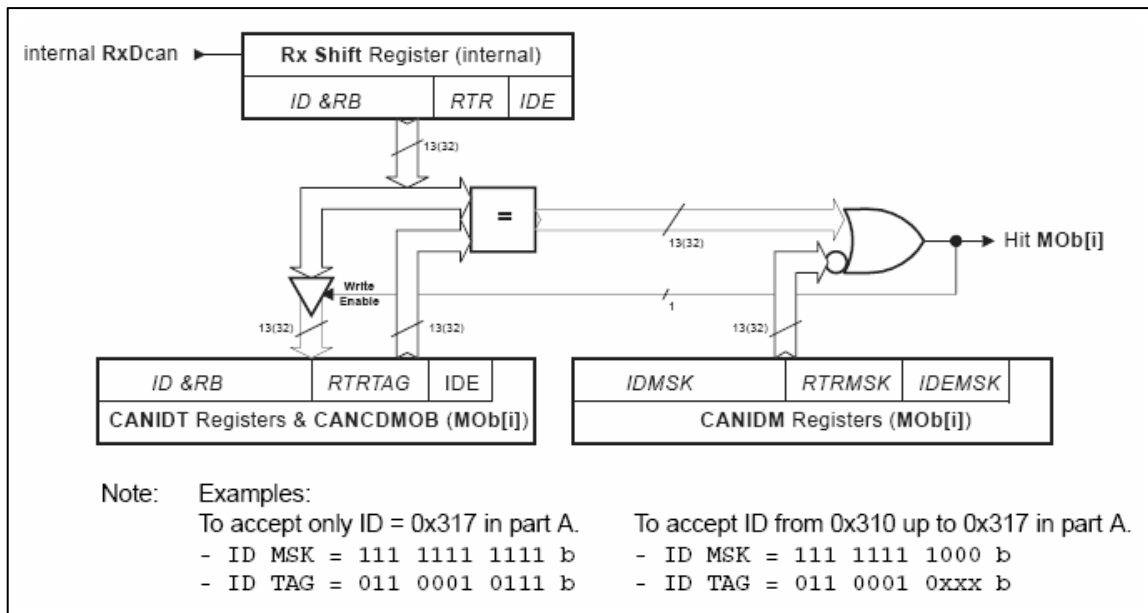
OK

>

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

>ATCM 1F FF 00 00

AT90CAN128 Filter- und Maskenregister (Auszug aus dem ATMEL® Datenblatt)



## OB2 Wireless Interface, AT-Befelssatz

---

### >AT!00

Ausgabe der Seriennummer des DX61-Controllers im USB-Stick. Alle Diamex-Controller besitzen eine einzigartige nicht veränderbare Seriennummer. Diese kann mit diesem Befehl abgefragt werden.

*Beispiel:*

```
>AT!00  
A123456789
```

>

### >AT!01

Ausgabe des Controllertyps und der Bios-Versionsnummer des DX61-Controllers im USB-Stick. Da der Identifikations-String veränderbar ist, ist keine eindeutige Identifizierung des Controllertyps und der Bios-Version über den AT! Befehl möglich. Aus diesem Grund wurde dieser Befehl eingefügt, dessen Ausgabe unveränderlich ist und immer den richtigen Controller-Typ anzeigt.

*Beispiel:*

```
>AT!01  
DX61-10
```

>

Obiges Beispiel zeigt die Ausgabe eines DX61 Controllers mit Bios-Version 1.0

### >AT%00

Ausgabe der Seriennummer des DX60-Controllers im OB2-Interface. Kann nur abgefragt werden, wenn eine Funkverbindung besteht.

*Beispiel:*

```
>AT%00  
B987654321
```

>

### >AT%01

Ausgabe des Controllertyps und der Bios-Versionsnummer des DX60-Controllers im OB2-Interface. Kann nur abgefragt werden, wenn eine Funkverbindung besteht.

*Beispiel:*

```
>AT%01  
DX60-10
```

>

Obiges Beispiel zeigt die Ausgabe eines DX60 Controllers mit Bios-Version 1.0

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

**Direkteingabe eines OB2-Befehls**

Zur Eingabe eines OB2-Befehls werden nur die für diesen Befehl notwendigen Daten im Hexcode übergeben. Sollte zuvor noch keine Verbindung mit dem OB2-Bus im Fahrzeug aufgebaut worden sein, wird dies bei Eingabe des ersten Befehls einmalig vorgenommen. Die verschiedenen Protokolle werden in folgender Reihenfolge durchgetestet:

1. J1850-VPWM
2. J1850-PWM
3. ISO9141/KWP2000 Slow-Init
4. KWP2000 Fast-Init
5. CAN 250 kBaud (wenn erkannt dann weiter mit 5)
6. CAN 500 kBaud
7. CAN 11 Bit
8. CAN 29 Bit

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.

*Beispiel:*

Ausgabe bei ISO9141 oder KWP2000 Connect.

```
>0100
BUS INIT: ...OK
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
```

>

### Die OBD2-Anschlußbuchse

Alle neu hergestellten Fahrzeuge müssen über solch eine 16-polige Buchse nach OBD2-Norm verfügen. Die Buchse muß sich in 1 Meter Umkreis vom Fahrersitz befinden. Obwohl die Bauform genormt ist, benutzen die verschiedenen Fahrzeughersteller verschiedene Übertragungsprotokolle.

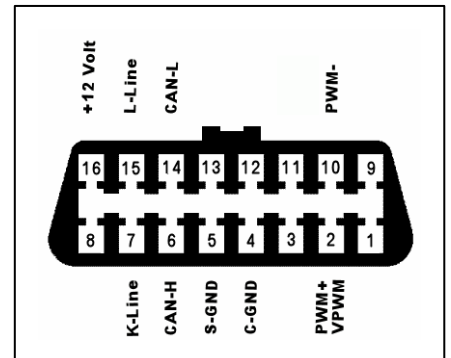
Genormt ist jedoch, welches Protokoll an welchem Pin der Buchse anliegt. In den meisten Fahrzeugen sind nur die Kontakte der 16-poligen Buchse vorhanden, die auch Signale führen. Oft sind zusätzliche (nicht genormte) Kontakte an Pins vorhanden und auch mit dem Steuergerät verbunden, die zur internen Diagnose der Fahrzeughersteller dienen. Da es für diese Kontakte keine offiziellen Dokumentationen gibt, sollte man diese möglichst auch immer unbelegt lassen.

Folgende Kontakte sind immer vorhanden:

PIN 4	Fahrzeug-Chassis-Masse	(Pin 4 + 5 sind meist verbunden)
PIN 5	Signal-Masse	
PIN 16	Stromversorgung + 12 Volt	

Folgende Kontakte sind je nach verwendetem Protokoll vorhanden und belegt:

PIN 7	ISO9141 / ISO14230 K-Line	(Ein- und Ausgang)
PIN 15	ISO9141 / ISO14230 L-Line	(Eingang, Ein- und Ausgang bei VAG)
PIN 2	J1850, PWM+	(Differentialsignal Bi-Direktional)
PIN 10	J1850, PWM-	(Differentialsignal Bi-Direktional)
PIN 2	J1850, VPWM	(Bi- Direktional)
PIN 6	ISO15765, CAN-H	(Differentialsignal Bi-Direktional)
PIN 14	ISO15765, CAN-L	(Differentialsignal Bi-Direktion)



### 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.

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 Interface und der Diagnose-Software entstehen könnten.

### Kontakt, Forum, Software, Updates

<http://www.diamex.de>