

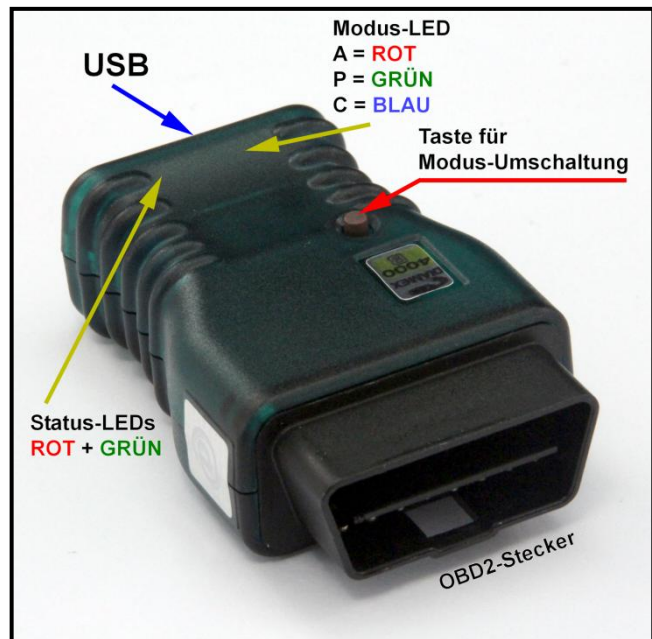
AGV4000C

OB2-Diagnose-Interface

AGV4000 Historie

Die Entwicklung des ersten AGV4000 basiert auf das Jahr 2005. Da es in der Elektronik wie in der Computertechnik rasend mit der Entwicklung neuer Komponenten voran geht, hatten wir uns im Jahr 2010 zu einem „Upgrade“ auf eine neue, moderne Technik entschlossen. Statt eines 8-Bit Controllers in Verbindung mit einem USB/Seriell-Wandlers kam nun ein moderner 32-Bit-ARM-Controller zum Einsatz, in dem alle erforderlichen Schnittstellen integriert sind. Vorteil ist die wesentlich höhere Arbeitsgeschwindigkeit und der größere Speicher, durch den einige Funktionen hinzugefügt werden konnten, die bei der alten Version wegen Speicherknappheit fehlten. Im Jahr 2016 war nun wieder einmal ein Hardware-Update fällig. Hier musste der Hersteller des ARM-Controllers gewechselt werden, weil es immer wieder Beschaffungsschwierigkeiten des Controllers in der Version B gegeben hatte.

Dies bedeutet jedoch, dass die Versionen B und C weitgehend Software-kompatibel sind. Alle vorhandenen OB2-Programme, die zuvor mit der Version B funktioniert hatte (z.B. MoDiag, CarPort), funktionieren auch weiterhin mit der Version C.



Änderungen zwischen Version AGV4000B und AGV4000C:

- Der im AGV4000B vorhandene Service-Mode zum Bios-Update ist nicht mehr vorhanden. Ein Bios-Update wird nun einfach durch starten des Update-Tools durchgeführt.
- Die Auswahl des Betriebsmodus über die Taste am AGV4000C wurde vereinfacht.
- Die 3 Betriebsmodi werden nun mit A, P, C bezeichnet. Modus P (Paththru) entspricht dem Modus B des AGV4000B. Modus C ist nur in der AGV4000C-Expert-Version verfügbar.
- Kleinere Änderungen betreffen die Filter-Funktionen des Lowlevel-CAN-Modus. Durch den bxCAN-Kern des verwendeten Microcontrollers ist die Einstellung von Filter-ID und Filter-Maske unterschiedlich gegenüber der des AGV4000B.

Treiberinstallation

Der Treiber für den AGV4000C ist identisch zu dem des AGV4000B. Sollte sich bereits ein Treiber für den AGV4000B auf dem Rechner befinden, wird dieser weiter benutzt. Im Windows-Gerätemanager wird in diesem Fall trotz angeschlossenem AGV4000C noch die alte Kennung des AGV4000B angezeigt, dies stört den Betrieb jedoch nicht.

Die Treiberdatei „AGV4000B.inf“ oder „AGV4000BC.inf“ kann über den Gerätemanager von Windows 7 und 8.x geladen werden. Bei Windows 8.x muss eventuell die Treibersignierung abgeschaltet werden. Hinweise hierzu finden Sie in den Links am Ende dieses Dokumentes.




Bei Windows 10 ist kein Treiber erforderlich. Hier wird der AGV4000C nach erstmaligem Anstecken an den PC als „Seriell-USB-Gerät (COMxx)“ im Gerätemanager angezeigt und kann sofort benutzt werden. Hinweise hierzu finden Sie ebenfalls in den Links am Ende dieses Dokumentes.

AGV4000C

OBD2-Diagnose-Interface

Modus-Auswahl

AGV4000C besitzt 3 verschiedene Betriebsmodi, die je nach Anwendungszweck aktiviert werden können. Modus A und P sind immer aktiviert, Modus C ist nur in der AGV4000C-Expert-Version verfügbar.

	MODUS A	OBD-Modus	Immer freigeschaltet
	MODUS P	TRANSIT- (PathThru-) Modus	Immer freigeschaltet
	MODUS C	CAN-Modus	In der Expert-Version freigeschaltet

Modus A (AT-Modus) wird für die fahrzeugunabhängige OBD2-Diagnose benötigt. Dieser Modus kann mit OBD2-Software wie MoDiag oder CarPort benutzt werden, aber auch mit einem Terminalprogramm auf dem PC können Diagnosefunktionen benutzt werden. Hierbei werden alle Kommandos wie bei Modems mit AT-Befehlen gestartet und die Antworten werden mit lesbaren ASCII-Zeichen zurückgegeben.

Modus P (PathThru-Modus) ist für die Tiefendiagnose bei Fahrzeugen des VAG-Konzerns vorgesehen. Das Programm CarPort benutzt diesen Modus um Daten der Steuergeräte auszulesen, die nicht über den OBD2-Modus verfügbar sind. Dieser Modus eignet sich nicht für die Bedienung mit einem Terminalprogramm.

Modus C (CAN-LowLevel-Modus) kann direkt Daten auf den CAN-Bus senden und Datenübertragungen loggen. Vielfältige Einstellungen erfordern die genaue Kenntnis des CAN-Bus Protokolls und ist deshalb nur in der AGV4000C-Expert-Version freigeschaltet. Programme wie CarPort greifen über diesen Modus direkt auf die Steuergeräte neuerer Fahrzeuge zu.

Die Umschaltung der Betriebsmodi geschieht mit Hilfe des Tasters, der ausgewählte Modus ist an der Farbe der Modus-LED zu erkennen.

1. Drücken Sie die Taste für ca. 2 Sekunden bis die Modus-LED blinkt.
2. Lassen Sie den Taster nun los, die LED blinkt weiter.
3. Nun drücken sie den Taster so oft kurz, bis der gewünschte Modus erreicht ist. Die Farbe der blinkenden LED zeigt den zukünftigen Modus an (siehe oben).
4. Drücken Sie die Taste für ca. 2 Sekunden, die LED leuchtet konstant und der gewünschte Modus wird aktiviert.
5. Wird die Taste 10 Sekunden lang nicht betätigt, wird automatisch der blinkende Modus aktiviert.
6. Der zuletzt eingestellte Modus bleibt gespeichert und steht somit beim nächsten Anstecken an den USB-Anschluss sofort wieder zur Verfügung.

Hinweis: Die LEDs befinden sich im Inneren des AGV4000C. Durch das halbtransparente Gehäuse können die Farben jedoch gut unterschieden werden.

AGV4000C

OBD2-Diagnose-Interface

A: AT (OBD2) - MODUS

Im OBD-Modus steht ein virtueller COM-Port zur Verfügung, der mit jedem Terminalprogramm (wie z.B. HyperTerminal unter Windows) angesprochen werden kann. Stellen Sie bitte im Terminalprogramm den passenden COM-Port ein, als Protokollparameter stellen Sie 8N1 ein (8-Bit, No Parity, 1 Stoppbit). Die Baudrate wird aufgrund der USB-Verbindung ignoriert, hier kann jeder beliebige Wert eingestellt werden, es wird immer die maximale USB-Geschwindigkeit benutzt. Für die komfortable Benutzung des AGV4000C empfehlen wir die Programme MoDiag oder CarPort.

Die Ausgabe der Ergebnisdaten kann in Grenzen geändert werden um zum Beispiel die Auswertung der Daten durch Microcontroller zu erleichtern.

Viele per AT-Befehl veränderbaren Parameter können sofort permanent im EEPROM gespeichert werden. Welche dies sind, finden Sie in der Befehlstabelle. Hierzu ist vor dem Befehl einfach ein Punkt zu setzen:

.ATE0 Ändert den Befehl nur temporär bis zum nächsten Reset (**ATZ**, **ATWS**, **ATD**)

.ATE0 Speichert den Befehl permanent ab, ist also auch nach einem Reset gültig.

Protokollauswahl und Verbindungsaufbau

Der AGV4000C Controller unterstützt alle derzeit aktuellen OBD2-Protokolle. Alle Protokolle können damit eindeutig identifiziert und selektiert werden.

Im Auslieferungszustand ist AGV4000C 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
```

Reihenfolge für den OBD2-Verbindungsaufbau	
1	PWM
2	VPWM
5	KWP-Fast-Init
3 + 4	ISO/KWP-5-Baud-Init
6	CAN11/500
7	CAN29/500
8	CAN11/250
9	CAN29/250

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

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

AGV4000C

OBD2-Diagnose-Interface

```
>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 AGV4000C 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.

Achtung! Wenn keine Verbindung aufgebaut werden kann, zunächst mit ATP schauen, welches Protokoll voreingestellt ist.

Eingabe von Funktionen und OBD2-Befehlen

Alle Eingaben erfolgen nach dem Promptzeichen `>`. Nach Ausführung einer Funktion oder eines OBD2-Befehles muss der Anwender auf dieses Zeichen warten, um einen erneuten Befehl einzugeben. Ein Befehl muss durch das Carriage-Return-Zeichen (ASCII 13d, 0x0D) abgeschlossen werden. In der Grundeinstellung werden alle Zeichen unter ASCII 33d, 0x21 ignoriert und alle Kleinbuchstaben in Großbuchstaben umgewandelt. Eingegebene Leerzeichen dienen nur zur besseren Lesbarkeit, werden jedoch komplett ignoriert.

Das Interface unterscheidet zwei verschiedene Kommandogruppen:

- 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 **AT**-tention.
- 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.

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 10 Sekunden abgebrochen und ein „?“ wird als Fehlermeldung ausgegeben. Diese Zeitdauer kann mit dem Befehl ATST geändert werden.

Befehle, die der Controller nicht versteht oder Falscheingaben bei Hex-Werten, werden ebenfalls mit einem „?“ als Fehlermeldung quittiert. Im Auslieferungszustand werden alle Fehlermeldungen zusätzlich als Klartext ausgegeben, dies kann zur Kompatibilität mit älteren Interfaces abgeschaltet werden.

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

Beispiel:

```
0100
```

AGV4000C

OBD2-Diagnose-Interface

oder
01 00

Eine ungerade Anzahl von Zeichen erzeugt eine Fehlermeldung.

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

Eine Liste aller AT-Befehle wird durch Eingabe von

>?

angezeigt. Hier werden alle Befehle ausgegeben. Die Reihenfolge der Befehle in der Liste ist nicht sortiert, sondern wurde während der Programmierung der Software festgelegt.

AGV4000C

OBD2-Diagnose-Interface

Fehlermeldungen

Code	Beschreibung
01	INPUT TIMEOUT Die Zeitdauer zwischen zwei eingegebenen Zeichen ist zu lang.
02	NO INPUT CHAR Es wurde kein gültiges Zeichen für die Ermittlung eines Befehles eingegeben.
03	UNKNOWN COMMAND Die eingegebene Anzahl von Zeichen konnte aufgrund eines unbekanntes Befehles nicht ermittelt werden.
04	WRONG HEXCHAR COUNT Bei der Eingabe eines OBD2-Befehles wurde eine ungültige Anzahl von Zeichen eingegeben.
05	ILLEGAL COMMAND Der eingegebene AT-Befehl existiert nicht.
06	SYNTAX ERROR Bei der Befehlseingabe wurden ungültige Zeichen eingegeben.
07	WRONG VALUE/RANGE Bei der Eingabe von Hexwerten wurden falsche oder ungültige Werte eingegeben.
08	UNABLE TO CONNECT Im automatischen Protokoll-Suchmodus wurde kein gültiges Protokoll gefunden. Versuchen Sie es noch einmal, oder stellen Sie ein Protokoll fest ein. Einige Steuergeräte reagieren empfindlich auf den automatischen Suchmodus.
09	BUS BUSY Der OBD-Bus wird zu lange von einem anderen Steuergerät belegt, es kann keine Anfrage gesendet werden. Nur ISO/KWP, PWM, VPWM
0A	NO FEEDBACK Aufgrund eines Hardwarefehlers können auf dem OBD-Bus gesendete Daten nicht wieder gelesen werden. Nur ISO/KWP, PWM, VPWM
0B	NO SYNCBYTE ISO/KWP Slow-Init lieferte kein Sync-Byte (0x55) zurück.
0C	NO KEYBYTE ISO/KWP Slow-Init lieferte kein Keybyte zurück.
0D	NO ADDRESSBYTE ISO/KWP Slow-Init lieferte kein Addressbyte zurück.
0E	WRONG PROTOCOL Bei der festen Voreinstellung von Protokoll 3 (ISO) wird Protokoll 4 (KWP) erkannt, oder bei der festen Voreinstellung von Protokoll 4 (KWP) wird Protokoll 3 (ISO) erkannt.
0F	NO DATA Kein Steuergerät antwortete auf eine OBD2Anfrage.
10	DATA ERROR Ein Fehler bei der Datenübertragung ISO/KWP, PWM, VPWM ist aufgetreten.
11	CHECKSUM ERROR Das empfangene OBD2-Paket bei ISO/KWP, PWM, VPWM hat einen Checksummenfehler.
12	NO ANSWER Bei KWP-Fast-Init (Protokoll 5) antwortet kein Steuergerät.

AGV4000C

OB2-Diagnose-Interface

13	COLLISION DETECT Diese Fehlermeldung kann bei den Protokollen PWM und VPWM auftreten, wenn zwei verschiedene Steuergeräte und OB2-Testgeräte gleichzeitig versuchen auf dem Bus zu senden.
14	CAN NO ANSWER Ein gesendetes CAN-Paket wird nicht beantwortet. Eventuell ist der Empfangsfilter falsch eingestellt.
15	CAN NO ACK Beim Senden eines CAN-Paketes ist kein anderer CAN-Node mehr verfügbar, der das Paket positiv beantwortet.
16	PROTOCOL 8 OR 9 REQUIRED Beim Befehl ATCB muss zuvor ein CAN-Protokoll 8 oder 9 eingestellt werden.
17	CAN ERROR Verschiedene Fehler bei der CAN-Übertragung

AGV4000C

OBD2-Diagnose-Interface

Befehlstabelle Übersicht

Hinweise zur Befehlstabelle:

- Befehl des AGV4000C.
- Die Spalte „ATZ“ gibt an, welche Parameter nach ATZ, ATD oder ATWS auf den gespeicherten Standardwert zurückgestellt werden.
- Die Spalte „Speicher“ gibt an, welche Parameter abgespeichert werden können.
- Standardwerte sind **Fett** dargestellt

Befehl	ATZ	Speicher	Funktionsbezeichnung
ATI			Zeigt die Interface-ID an
ATE	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Schaltet Terminal-Echo [ein/aus]
ATL	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Linefeed (0x0A, ASCII 10) nach Carriage Return [ein/aus]
ATZ			Warmstart Reset, Standardwerte wiederherstellen
ATD			Auf Standardwerte zurückstellen (ohne Reset)
ATWS			Schneller Warmstart, gleiche Funktion wie ATZ
ATOEN	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Fehler als Nummern statt Klartext ausgeben [ein/aus]
ATOSY	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Syntax-Error als Klartext anzeigen [ein/aus]
ATOMY	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Eigene Interface-ID anzeigen [ein/aus]
ATOID		<input checked="" type="checkbox"/>	Eigene Interface-ID eingeben
ATOF	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Microcontroller Ausgabeformat ein/aus
ATOHS	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Leerzeichen zwischen HEX-Bytes [ein/aus]
ATOUC	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Zeichen in Großbuchstaben umwandeln [ein/aus]
AT!			Sonderfunktionen, Anzeigen interner Daten
ATV			Liste der veränderbaren Parameter anzeigen
ATH	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	OBD2-Header ausgeben [ein/aus]
ATP		<input checked="" type="checkbox"/>	Protokoll voreinstellen [AUTO]
ATONI	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Verbindungsaufbau ohne Initialisierungssequenz [ein/aus]
ATONR	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Nach Initialisierung keine Anfrage senden [ein/aus]
ATN			Aktuelles Protokoll als Nummer ausgeben
ATDP			Aktuelles Protokoll im Klartext ausgeben
ATK			ISO/KWP Keybytes anzeigen
ATB			Bufferinhalt komplett anzeigen
ATBD			Bufferinhalt anzeigen (kompatibel, kurz)
ATSF	<input checked="" type="checkbox"/>		ECU-Filter setzen [0x00 = aus]
ATSW	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Wakeup-Zeit setzen [0x19]
ATST	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Eingabe-Timeout setzen [0x64]
ATSH			Header für ISO, KWP, PWM, VPWM setzen
ATWM			Wakeup-Message für ISO, KWP setzen

AGV4000C

OBD2-Diagnose-Interface

ATRST			Kaltstart Reset, alle EEPROM Werte auf Standard
ATCB			CAN Bitrate für die Protokolle 8 und 9 einstellen
ATCA	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	CAN Autoformat [ein /aus]
ATCC	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	CAN Flow-Control [ein /aus]
ATCD			CAN Flow-Delay [0x0A]
ATCT			CAN TX-Adresse einstellen [7DF, 18DB33F1]
ATCR			CAN RX-Adresse einstellen [7E8, 18DAF100]
ATCM			CAN RX-Maske einstellen [7F8, 1FFFFFF00]

Beschreibung der AT-Befehle

Hinweise zur Befehlstabelle:

- (x) Optionale Parameter in runden Klammern.
- [x|y|z] Erforderliche Parameter in eckigen Klammern mit der Liste der erlaubten Werte.
- Ein Punkt vor dem Befehl bedeutet, dass die Änderung direkt im EEPROM abgespeichert wird und damit nach einem Reset oder Trennung der Versorgungsspannung gespeichert bleibt.

Befehl: **ATI**

Bezeichnung: Zeigt die Interface-ID an

Beispiel:

```
>ATI  
AGV4000C v1.0
```

Die Interface-ID des AGV4000C wird ausgegeben. Diese kann jedoch für Kompatibilitätsanpassungen geändert werden, wenn einem Anwenderprogramm das neue Interface noch nicht bekannt ist und ihm die Kennung für ein bekanntes Interface übermittelt werden soll. Mit den weiter unten beschriebenen Optionsbefehlen ATOMY und ATOID kann die Kennung temporär oder permanent geändert werden.

Befehl: **(.)ATE [1|0|ON|OFF]**

Bezeichnung: Schaltet das Terminal Echo an oder aus (ASCII 10) nach Carriage Return ein/aus.

Standardwert: 1 ON

Beispiele:

```
>ATE1  
OK
```

```
>ATE OFF  
OK
```

Schaltet das Terminal-Echo ein oder aus. Wird mit einem Terminalprogramm gearbeitet, ist es in den meisten Fällen recht sinnvoll, das Echo eingeschaltet zu lassen, da man ein eingegebenes Zeichen sofort wieder zurückgesendet bekommt und damit kontrollieren kann, ob die Eingabe in Ordnung ist.

AGV4000C

OBD2-Diagnose-Interface

Befehl: **(.)ATL [1|0|ON|OFF]**

Bezeichnung: Schaltet die Ausgaben von Linefeed

Standardwert: 1 ON

Beispiele:

```
>ATL1
```

```
OK
```

```
> .ATL OFF
```

```
OK
```

Mit diesem Befehl wird die Ausgabe des zusätzlichen Linefeed-Zeichens (ASCII 10, HEX 0x0A) nach einem Carriage-Return gesteuert. Viele Terminalprogramme fügen ein eigenes Linefeed-Zeichen hinzu, wenn sie ein Carriage-Return erkennen, dies führt dazu, dass hinter jeder ausgegeben Zeile eine zusätzliche Leerzeile eingefügt und die Ausgabe unnötig verlängert wird. Auch ist es bei der Auswertung der Antworten mit Microcontrollern nicht notwendig, ein zusätzliches Linefeed-Zeichen zu übertragen.

Nach ATZ, ATWS oder ATD wird der Standardwert eingestellt. Mit einem Punkt vor dem Befehl wird der eingestellte Parameter fest im EEPROM des Interface abgespeichert und dient somit als Standardwert nach ATZ, ATWS oder ATD.

Befehl: **ATZ**

Bezeichnung: Warmstart Reset

Beispiel:

```
>ATZ
```

```
AGV4000C v3.0
```

Dieser Befehl bricht alle Funktionen ab und das Interface führt einen Warmstart-Reset aus. Die aktuelle Verbindung wird getrennt und verschiedene Parameter werden auf die Standardwerte zurückgestellt, wenn sie nicht permanent im EEPROM abgespeichert wurden (siehe Funktionstabelle).

Befehl: **ATD**

Bezeichnung: Standardwerte wiederherstellen

Beispiel:

```
>ATD
```

```
OK
```

Verschiedene Parameter werden auf die Standardwerte zurückgestellt, wenn sie nicht permanent im EEPROM abgespeichert wurden (siehe Funktionstabelle). Im Gegensatz zum Befehl ATZ wird eine eventuell vorhandene Verbindung nicht getrennt.

Befehl: **ATWS**

Bezeichnung: Schneller Warmstart Reset

Beispiel:

```
>ATWS
```

```
AGV4000C v3.0
```

```
>
```

Dieser Befehl bricht alle Funktionen ab und das Interface führt einen Warmstart-Reset aus. Die aktuelle Verbindung wird getrennt und verschiedene Parameter werden auf die Standardwerte zurückgestellt, wenn sie nicht permanent im EEPROM abgespeichert wurden (siehe Funktionstabelle). Der Unterschied zum Befehl ATZ besteht darin, dass dieser Befehl schneller ausgeführt wird.

AGV4000C

OBD2-Diagnose-Interface

Befehl: (.)ATOEN [1|0|ON|OFF]

Bezeichnung: Fehler als Nummer statt als Klartext ausgeben (enable numbers)

Standardwert: 0 OFF

Beispiele:

```
>ATOEN1
```

```
OK
```

```
> .ATOEN OFF
```

```
OK
```

In der Standardeinstellung werden Fehler als Klartext ausgegeben. Um die Auswertung mit einem Microcontroller zu vereinfachen, kann hiermit der Fehlercode als Hexwert ausgegeben werden.

Beispiel mit ATOEN0:

```
>AT
```

```
? UNKNOWN COMMAND
```

Beispiel mit ATOEN1:

```
>AT
```

```
? Error #03
```

Nach ATZ, ATWS oder ATD wird der Standardwert eingestellt. Mit einem Punkt vor dem Befehl wird der eingestellte Parameter fest im EEPROM des Interface abgespeichert und dient somit als Standardwert nach ATZ, ATWS oder ATD.

Befehl: (.)ATOSY [1|0|ON|OFF]

Bezeichnung: Syntax-Error als Klartext anzeigen

Standardwert: 1 ON

Beispiele:

```
>ATOSY1
```

```
OK
```

```
> .ATOSY OFF
```

```
OK
```

In der Standardeinstellung ATOSY1 wird ein Eingabefehler bei einem AT-Befehl mit

```
? SYNTAX ERROR
```

ausgegeben.

Zur Kompatibilität mit vorhandener Software kann durch ATOSY0 die bei älteren Interfaces benutzte kurze Antwort

```
?
```

eingestellt werden.

Nach ATZ, ATWS oder ATD wird der Standardwert eingestellt. Mit einem Punkt vor dem Befehl wird der eingestellte Parameter fest im EEPROM des Interface abgespeichert und dient somit als Standardwert nach ATZ, ATWS oder ATD.

AGV4000C

OBD2-Diagnose-Interface

Befehl: (.)ATOMY [1|0|ON|OFF]

Bezeichnung: Eigene Interface-ID anzeigen (show my id)

Standardwert: 0 OFF

Beispiele:

```
>ATOMY1
```

```
OK
```

```
>.ATOMY OFF
```

```
OK
```

Zur Personalisierung des AGV4000C kann die Meldung, die bei den Befehlen ATZ oder ATI erscheint, geändert werden. Diese muss zunächst mit dem folgenden Befehl ATOID vorbereitet werden. Mit ATOMY1 wird der eigene Text nun aktiviert.

Beispiel mit ATOMY0:

```
>ATI
```

```
OBD-Diag AGV4000C v3.0
```

Beispiel mit ATOMY1:

```
>ATI
```

```
Mein tolles OBD-Interface
```

Bitte beachten Sie, dass das Interface durch die Veränderung des Textes von verschiedenen Programmen eventuell nicht mehr erkannt werden kann.

Nach ATZ, ATWS oder ATD wird der Standardwert eingestellt. Mit einem Punkt vor dem Befehl wird der eingestellte Parameter fest im EEPROM des Interface abgespeichert und dient somit als Standardwert nach ATZ, ATWS oder ATD.

Befehl: (.)ATOID "EIGENER TEXT

Bezeichnung: Interface-ID eingeben

Standardwert: diamex.de

Beispiele:

```
>ATOID OHNELEERZEICHEN
```

```
OK
```

```
>.ATOID "Mit Leerzeichen
```

```
OK
```

Zur Personalisierung des AGV4000C kann die Meldung, die bei den Befehlen ATZ oder ATI erscheint, geändert werden. Die Länge des Textes kann maximal 32 Zeichen betragen, werden mehr Zeichen eingegeben, wird der Text auf 32 Zeichen gekürzt. Sollen im ID-Text Kleinbuchstaben oder Leerzeichen erscheinen, muss der Text mit einem " – Zeichen beginnen. Erst mit dem Befehl ATOMY1 wird der eingegebene Text aktiviert.

Mit einem Punkt vor dem Befehl wird der eingegebene Text fest im EEPROM des Interface abgespeichert und steht dadurch auch nach Trennung der Stromversorgung wieder zur Verfügung.

AGV4000C

OBD2-Diagnose-Interface

Befehl: (.)ATOF [1|0|ON|OFF]

Bezeichnung: Microcontroller Ausgabeformat umschalten (output format)

Standardwert: 0 OFF

Beispiele:

>ATOF1

OK

>.ATOF OFF

OK

Für die Auswertung der ausgegebenen Daten mit einem angeschlossenen Microcontroller oder einer PC-Software kann das Ausgabeformat umgeschaltet werden. In Verbindung mit dem nachfolgenden Befehl `ATOHS` kann hierdurch einiges an unnötigen Zeichen bei der Übertragung eingespart werden.

Im Auslieferungszustand ist das Ausgabeformat kompatibel mit älteren OBD2-Interfaces.

Nach `ATZ`, `ATWS` oder `ATD` wird der Standardwert eingestellt. Mit einem Punkt vor dem Befehl wird der eingestellte Parameter fest im EEPROM des Interface abgespeichert und dient somit als Standardwert nach `ATZ`, `ATWS` oder `ATD`.

Befehl: (.)ATOHS [1|0|ON|OFF]

Bezeichnung: Leerzeichen zwischen Hexbytes anzeigen (hex space)

Standardwert: 1 ON

Beispiele:

>ATOHS1

OK

>.ATOHS OFF

OK

Zur Reduzierung der übertragenen Zeichen und zur einfacheren Auswertung der Ergebnisdaten können alle störenden Leerzeichen zwischen den Hex-Ausgaben unterdrückt werden.

Beispiel mit `ATOHS1`:

11 22 33 44 55 66

Beispiel mit `ATOHS0`:

112233445566

Im Auslieferungszustand ist das Ausgabeformat kompatibel mit älteren OBD2-Interfaces.

Nach `ATZ`, `ATWS` oder `ATD` wird der Standardwert eingestellt. Mit einem Punkt vor dem Befehl wird der eingestellte Parameter fest im EEPROM des Interface abgespeichert und dient somit als Standardwert nach `ATZ`, `ATWS` oder `ATD`.

AGV4000C

OBD2-Diagnose-Interface

Befehl: (.)ATOUC [1|0|ON|OFF]

Bezeichnung: Zeichen in Großbuchstaben umwandeln (upper case)

Standardwert: 1 ON

Beispiele:

>ATOUC1

OK

>.ATOUC OFF

OK

Alle AT-Befehle im Auslieferungszustand müssen in Großbuchstaben eingegeben werden. Deshalb ist eine automatische Umwandlung von Klein- in Großbuchstaben aktiviert. Durch Änderung der Befehlsliste können auch Befehle in Kleinbuchstaben erstellt werden, in diesem Fall sollte auch die Umwandlung in Großbuchstaben deaktiviert werden.

Nach ATZ, ATWS oder ATD wird der Standardwert eingestellt. Mit einem Punkt vor dem Befehl wird der eingestellte Parameter fest im EEPROM des Interface abgespeichert und dient somit als Standardwert nach ATZ, ATWS oder ATD.

Befehl: AT! [xx]

Bezeichnung: Sonderfunktionen

Mit diesem Befehl können interne Daten des AGV4000C ausgelesen, jedoch nicht verändert werden.

Seriennummer anzeigen:

>AT!00

12345-67890-123

Geräte-Identifikation anzeigen:

>AT!01

AGV4000C-30

Boardspannung anzeigen:

>AT!10

13.5V

Weitere Befehle dienen nur dem Systemtest bei der Herstellung des Gerätes und sind deshalb hier nicht aufgeführt.

AGV4000C

OBD2-Diagnose-Interface

Befehl: **ATV**

Bezeichnung: Liste der veränderbaren Parameter anzeigen

Die folgende Ausgabe sollte im Auslieferungszustand erscheinen (Protokoll 6 aktiv):

```
>ATV
  IO-CFG: E1    L1    OHS1 OUC1
  OUT-CFG: H0    OF0   OEN0 OSY1 OMY0
  OBD-CFG: ONI0 ONR0
  VALUES: SF00 SW19 ST64
           OID: [diamex.de]
CAN-CFG: CA1   CC1
BPS (CB): BRP=2 TSEG1=15 TSEG2=8 SJW=4 (500000 bps)
TX  (CT): 7DF   18DB33F1
RX  (CR): 7F8   1FFFFFF00
FLOW(CD): 30 00 (0A)
```

Befehl: **(.)ATH [1|0|ON|OFF]**

Bezeichnung: OBD2-Header und Checksummenbytes ausgeben

Standardwert: 0 OFF

Beispiele:

```
>ATH1
```

```
OK
```

```
> .ATH OFF
```

```
OK
```

Hier kann gewählt werden, ob der OBD2-Header und die Checksumme bei PWM, VPWM, ISO, KWP-Ausgaben oder die CAN-Adresse und PCI-Bytes bei CAN-Ausgaben eingeschaltet werden sollen. Den genauen Aufbau der Ausgabedaten finden Sie in einem separaten Kapitel.

Sollten im Fahrzeug mehrere Steuergeräte eingebaut sein, ist es immer besser, die Header eingeschaltet zu lassen, damit sofort erkannt werden kann, welches Steuergerät die Daten liefert. Wenn davon auszugehen ist, dass es nur eine Antwort gibt (zum Beispiel bei ständiger Abfrage der Fahrzeuggeschwindigkeit), kann der Header auch abgeschaltet werden um unnütze Zeichen bei der Übertragung und Auswertung zu unterdrücken.

Wenn das alternative Ausgabeformat `ATOF1` gewählt ist, hat die Einstellung keine Funktion, in diesem Fall werden die Headerdaten immer ausgegeben.

Nach `ATZ`, `ATWS` oder `ATD` wird der Standardwert eingestellt. Mit einem Punkt vor dem Befehl wird der eingestellte Parameter fest im EEPROM des Interface abgespeichert und dient somit als Standardwert nach `ATZ`, `ATWS` oder `ATD`.

AGV4000C

OBD2-Diagnose-Interface

Befehl: (.)ATP ([A][0..9])

Bezeichnung: Protokoll einstellen oder anzeigen

Standardwert: 0 AUTO

Beispiele:

>ATP

2 = SAE J1850 / VPWM

>.ATP 6

OK

Die genaue Bedeutung der hier eingegebenen Werte finden Sie im Kapitel „Protokollauswahl und Verbindungsaufbau“.

Mit einem Punkt vor dem Befehl wird der eingestellte Parameter fest im EEPROM des Interface abgespeichert und steht somit als Standardwert nach Trennung der Stromversorgung wieder zur Verfügung.

Befehl: (.)ATONI [1|0|ON|OFF]

Bezeichnung: Verbindungsaufbau ohne Initialisierungssequenz (no Init)

Standardwert: 0 OFF

Beispiele:

>ATONI1

OK

>.ATONI OFF

OK

Bei den Protokollen PWM (1), VPWM (2) und CAN (6..9) ist keine spezielle Initialisierungssequenz erforderlich, um eine Verbindung des OBD2-Interfaces mit dem Steuergerät im Fahrzeug herzustellen. Zur sicheren Erkennung des Protokolls sendet AGV4000C jedoch zum ersten Verbindungsaufbau immer die Sequenz 01 00, die von jedem OBD2-fähigen Steuergerät beantwortet werden muss. Dieser erste Verbindungsaufbau kann durch `ATONI1` unterdrückt werden. In diesem Fall wird die Verbindung aufgebaut, wenn das Steuergerät eine gültige Antwort auf eine OBD2-Anfrage liefert.

☞ **HINWEIS:** Das Ausschalten der Initialisierungssequenz ist nur bei fest angewähltem Protokoll `ATPx` möglich.

Nach `ATZ`, `ATWS` oder `ATD` wird der Standardwert eingestellt. Mit einem Punkt vor dem Befehl wird der eingestellte Parameter fest im EEPROM des Interface abgespeichert und dient somit als Standardwert nach `ATZ`, `ATWS` oder `ATD`.

AGV4000C

OBD2-Diagnose-Interface

Befehl: (.)**ATONR** [1|0|ON|OFF]

Bezeichnung: Nach Initialisierung keine Anfrage senden (no request)

Standardwert: 0 OFF

Beispiele:

```
>ATONR1
```

```
OK
```

```
> .ATONR OFF
```

```
OK
```

Wenn zum Verbindungsaufbau der automatische Protokoll-Suchmodus benutzt wird (**ATP0**), wird sofort nach erkanntem Protokoll der eingegebene OBD2-Befehl gesendet. Dies kann mit **ATONR1** unterdrückt werden. In diesem Fall wird bei erkanntem Protokoll nicht die Antwort des OBD2-Befehls sondern 3 Bytes zurückgeliefert:

```
05 8F E9 Hier wurde Protokoll 5 mit den Keybytes 8FE9 gefunden.
```

```
06 00 00 Hier wurde Protokoll 6 gefunden, für dieses Protokoll gibt es keine Keybytes
```

Nach **ATZ**, **ATWS** oder **ATD** wird der Standardwert eingestellt. Mit einem Punkt vor dem Befehl wird der eingestellte Parameter fest im EEPROM des Interface abgespeichert und dient somit als Standardwert nach **ATZ**, **ATWS** oder **ATD**.

Befehl: **ATN**

Bezeichnung: Aktuelles Protokoll als Nummer ausgeben

Beispiel:

```
>ATN
```

```
F5
```

Das aktuell aktive Protokoll wird als Hexwert **F1** - **F9** ausgegeben.

Wenn keine Verbindung besteht, wird **F0** ausgegeben.

Befehl: **ATDP**

Bezeichnung: Aktuelles Protokoll im Klartext ausgeben

Beispiel:

```
>ATDP
```

```
ISO 14230-4, KWP2000 (Fast Init)
```

Das aktuell aktive Protokoll wird im Klartext ausgegeben.

Wenn keine Verbindung besteht, wird **NO CONNECTED** ausgegeben.

Befehl: **ATK**

Bezeichnung: ISO/KWP Keybytes anzeigen

Beispiel:

```
>ATK
```

```
8F E9
```

Die Keybytes der aktiven ISO/KWP-Verbindung (Protokolle 3,4,5) werden ausgegeben.

Wenn das aktive Protokoll keine Keybytes unterstützt, wird **00 00** ausgegeben.

AGV4000C

OB2-Diagnose-Interface

Befehl: **ATB**

Bezeichnung: Buffer-Inhalt anzeigen

Beispiel:

```
>ATB
```

```
86 F1 10 41 00 B8 7B 30 10 3B
```

Die kompletten empfangenen Daten des letzten OB2-Befehles inklusive Header- und Checksummenbytes werden ausgegeben.

Befehl: **ATBD**

Bezeichnung: Bufferinhalt anzeigen, kompatibel (buffer dump)

Beispiel:

```
>ATBD
```

```
0A 86 F1 10 41 00 B8 7B 30 10 3B 10 00
```

Die ersten 12 Bytes des letzten OB2-Befehles werden ausgegeben. Das erste Byte gibt die Anzahl der gültigen angezeigten Bytes an. 0A = 10 gültige Bytes.

☞ HINWEIS: Dieser Befehl ist nur zur Kompatibilität mit älteren OB2-Interfaces vorhanden und kann in zukünftigen Versionen nicht mehr vorhanden sein. Bitte verwenden Sie stattdessen den Befehl **ATB**.

Befehl: **ATSF [xx]**

Bezeichnung: ECU-Filter setzen (set filter)

Beispiel:

```
>ATSF10
```

```
OK
```

Bei Automatikfahrzeugen antworten in der Regel die Steuergeräte mit den Kennungen 10 (Motor) und 17 (Getriebe). Hier können die Ausgaben durch Einsetzen des Filters auf das gewünschte Steuergerät reduziert werden um die Auswertung durch die Software zu erleichtern. Der Filter lässt sich durch Eingabe von **ATSF00** abschalten.

☞ HINWEISE: Wird eine ungültige Kennung als Filter eingegeben, können alle Ausgaben unterdrückt werden und es wird nur **NO DATA** ausgegeben. Der eingestellte Filterwert kann nicht abgespeichert werden, nach **ATZ** ist der Filter automatisch deaktiviert.

Befehl: **(.)ATSW [xx]**

Bezeichnung: Wakeup-Zeit setzen (set wakeup)

Standardwert: 19 (2.5 Sekunden)

Beispiele:

```
>ATSW28
```

```
OK
```

```
> .ATSW 19
```

```
OK
```

Bei den Protokollen 3,4 und 5 (ISO/KWP) muss zum halten der Verbindung regelmäßig ein gültiger OB2-Befehl zum Steuergerät gesendet werden. Geschieht dies nicht, wird die Verbindung automatisch nach 5 Sekunden getrennt. Das AGB4000C sendet aus diesem Grund selbständig regelmäßig sogenannte Wakeup-Pakete, wenn keine OB2-Befehle vom PC angefordert werden. Die Zeitdauer zwischen den Paketen kann hier eingestellt werden. Der eingegebene Wert ist die Zeitdauer * 100 ms als Hexwert. 0x19 = Dezimal 25 = 2,5 Sekunden.

☞ HINWEISE: Bitte keine Werte größer als 0x32 (50 dez.) einstellen, da der Wakeup zu spät kommt und das Steuergerät bereits die Verbindung getrennt hat. Mit dem Wert 00 kann der automatische Wakeup abgeschaltet

AGV4000C

OB2-Diagnose-Interface

werden. Hier muß sich die PC-Software selber darum kümmern, dass die Zeit zwischen den Abfragen nicht grösser als 5 Sekunden wird.

Nach `ATZ`, `ATWS` oder `ATD` wird der Standardwert eingestellt. Mit einem Punkt vor dem Befehl wird der eingestellte Parameter fest im EEPROM des Interface abgespeichert und dient somit als Standardwert nach `ATZ`, `ATWS` oder `ATD`.

Befehl: `(.)ATST [xx]`

Bezeichnung: Eingabe-Timeout setzen (set timeout)

Standardwert: 64 (10 Sekunden)

Beispiele:

```
>ATST32
```

```
OK
```

```
> .ATST 64
```

```
OK
```

Wird die hier eingestellte Zeitdauer bei der Eingabe eines Befehles überschritten, erscheint die Meldung `INPUT TIMEOUT` und die Eingabe wird abgebrochen.

Der eingegebene Wert ist die Zeitdauer * 100 ms als Hexwert. `0x64` = Dezimal 100 = 10 Sekunden.

☞ HINWEIS: Wenn der Wert zu kurz eingestellt wird, ist eventuell keine Eingabe mehr möglich. In diesem Fall muss die Grundeinstellung wiederhergestellt werden.

Nach `ATZ`, `ATWS` oder `ATD` wird der Standardwert eingestellt. Mit einem Punkt vor dem Befehl wird der eingestellte Parameter fest im EEPROM des Interface abgespeichert und dient somit als Standardwert nach `ATZ`, `ATWS` oder `ATD`.

Befehl: `ATSH [xx xx xx]`

Bezeichnung: Header für ISO, KWP, PWM, VPWM setzen (set header)

Beispiel:

```
>ATSH 11 22 33
```

```
OK
```

Dieser Befehl ist für Experten vorgesehen, die Steuergeräte außerhalb der OBD2-Norm ansprechen wollen. Die 3 übergebenen Werte werden als Header in den darauffolgenden OBD2-Befehl eingefügt. Das Längenbyte wird bei KWP2000 automatisch angepasst.

☞ HINWEISE: Die Werte können nur nach einem erfolgten Verbindungsaufbau geändert werden. Die eingestellten Werte können nicht abgespeichert werden, nach `ATZ` werden die Standardwerte wiederhergestellt. Bei CAN-Protokollen ist diese Funktion ohne Bedeutung.

AGV4000C

OBD2-Diagnose-Interface

Befehl: **ATWM** [**xx** . . .]

Bezeichnung: Wakeup-Message für ISO, KWP setzen

Beispiel:

```
>ATWM 68 6A F1 01 00
```

OK

Die 4 bis 6 übergebenen Werte werden als Wakeup-Message gesendet. Hier müssen auch die Headerbytes und das KWP2000 Längenbyte mit eingegeben werden. Das Checksummenbyte wird automatisch berechnet und darf nicht eingegeben werden.

☞ HINWEISE: Die Werte können nur nach einem erfolgten Verbindungsaufbau geändert werden. Die eingestellten Werte können nicht abgespeichert werden, nach **ATZ** werden die Standardwerte wiederhergestellt. Bei CAN-Protokollen ist diese Funktion ohne Bedeutung.

Befehl: **ATRST**

Bezeichnung: Kaltstart Reset

Beispiel:

```
>ATRST
```

```
AGV4000C v3.0
```

Dieser Befehl bricht alle Funktionen ab und das Interface führt einen Kaltstart-Reset wie nach dem Trennen und erneuten Anschließen der Stromversorgung aus. Die aktuelle Verbindung wird getrennt und alle Parameter werden auf die Standardwerte zurückgestellt, wenn sie nicht permanent im EEPROM abgespeichert wurden (siehe Funktionstabelle).

Befehl: **ATCB** [**d**]

Bezeichnung: CAN Bitrate für die Protokolle 8 und 9 einstellen

Standardwert: 250.000

Beispiele:

```
>ATCB 100k
```

OK

```
>ATCB 250000
```

OK

Die CAN-Bitrate für die Protokolle 8 und 9 kann hiermit verändert werden. Hierzu muss eines der Protokolle 8 oder 9 voreingestellt sein: **ATP8** oder **ATP9**.

Die Eingabe des Bitratenwertes geschieht im Dezimalwert. Folgende Eingaben sind möglich:

100000 Direkteingabe des Wertes

100k Eingabe als Kilobit, wobei 1k = 1000 bps ist.

1M Eingabe als Megabit, wobei 1 Megabit = 1000000 gleichzeitig der größte mögliche Wert ist.

☞ HINWEIS: Wird eine unübliche Bitrate eingegeben, kann es passieren, dass diese nicht genau vom CAN-Controller des AGV4000C erzeugt werden kann. Die genaue eingestellte Bitrate können Sie mit dem Befehl **ATV** kontrollieren.

Die gewählte Bitrate wird ab dem nächsten Verbindungsaufbau benutzt. Die Bitrate bleibt auch nach **ATZ**, **ATD** oder **ATWS** erhalten, wird jedoch nach **ATRST** auf den Standardwert 250.000 zurückgestellt.

AGV4000C

OB2-Diagnose-Interface

Befehl: (.)ATCA [1|0|ON|OFF]

Bezeichnung: CAN Autoformat ein/aus

Standardwert: 1 ON

Beispiele:

```
>ATOCA1
```

```
OK
```

```
>.ATOCA OFF
```

```
OK
```

Das OB2-spezifische CAN-Übertragungsformat kann hiermit ein- oder ausgeschaltet werden. Im OB2-Format wird automatisch das PCI-Byte dem CAN-Paket hinzugefügt, es sind maximal 7 Datenbytes möglich und es wird immer ein 8-Byte Datenpaket gesendet. Ist das Autoformat ausgeschaltet, wird immer die Anzahl der eingegebenen Bytes übertragen.

Beispiel mit Autoformat:

```
>01 00
```

Gesendet wird:

```
02 01 00 00 00 00 00 00
```

Beispiel ohne Autoformat:

```
>01 00
```

Gesendet wird:

```
01 00
```

Um ein gültiges OB2-Paket ohne Autoformat zu versenden, muss folgendes eingegeben werden:

```
>02 01 00 00 00 00 00 00
```

Nach ATZ, ATWS oder ATD wird der Standardwert eingestellt. Mit einem Punkt vor dem Befehl wird der eingestellte Parameter fest im EEPROM des Interface abgespeichert und dient somit als Standardwert nach ATZ, ATWS oder ATD.

Befehl: (.)ATCC [1|0|ON|OFF]

Bezeichnung: CAN Flow-Control ein/aus

Standardwert: 1 ON

Beispiele:

```
>ATOCF1
```

```
OK
```

```
>.ATOCF OFF
```

```
OK
```

Besteht eine OB2-Antwort aus mehreren CAN-Paketen (zum Beispiel bei Abfrage der Fahrgestellnummer), wird diese auf mehrere CAN-Pakete aufgeteilt. Nach Senden des ersten Paketes (First Frame) muss der OB2-Diagnoseadapter ein Flow-Control-Paket zurücksenden, um dem Steuergerät zu signalisieren, in welchem zeitlichen Abstand die nachfolgenden Pakete gesendet werden dürfen. Die Zeitdauer zwischen den Paketen wird mit dem nachfolgenden Befehl ATCD eingestellt.

Zu Testzwecken und zum Gebrauch des AGV4000C an CAN-Bussen, die nicht der OB2-Spezifikation entsprechen, kann das Flow-Control-Paket abgeschaltet werden.

Nach ATZ, ATWS oder ATD wird der Standardwert eingestellt. Mit einem Punkt vor dem Befehl wird der eingestellte Parameter fest im EEPROM des Interface abgespeichert und dient somit als Standardwert nach ATZ, ATWS oder ATD.

AGV4000C

OBD2-Diagnose-Interface

Befehl: (.)ATCD [xx]

Bezeichnung: CAN Flow-Delay einstellen

Standardwert: 0A

Beispiel:

```
>ATCD 16
```

OK

Der Flow-Delay-Wert gibt an, in welchem Zeitraum die nachfolgenden Antworten von Multiframe-Paketen gesendet werden. Die Angabe ist in Millisekunden, Standardwert ist 0A = 10ms.

Nach ATZ, ATWS oder ATD wird der Standardwert eingestellt. Mit einem Punkt vor dem Befehl wird der eingestellte Parameter fest im EEPROM des Interface abgespeichert und dient somit als Standardwert nach ATZ, ATWS oder ATD.

Befehl: (.)ATCT [xxx | xxxxxxxxx]

Bezeichnung: CAN TX-Adresse einstellen

Standardwert: 7DF, 18DB33F1

Beispiele:

```
>ATCT 750
```

OK

```
>ATCT 18DB22E8
```

OK

Die Standardeinstellungen der CAN-Adressen sind kompatibel zu den OBD2-Vorgaben. Wer dennoch eigene Experimente – selbstverständlich auf eigene Gefahr – machen will, kann die CAN-Sendeadressen nach eigenem Belieben verändern.

Die Anzahl der eingegebenen Zeichen für die CAN-Adresse gibt an, ob diese dem 11-Bit oder dem 29-Bit CAN-Protokoll zugeordnet wird. Eine 11-Bit Adresse ist gültig im Wertebereich 000-7FF und gilt für die Protokolle 6 und 8. Eine 29-Bit Adresse ist gültig im Wertebereich 00000000-1FFFFFFF und gilt für die Protokolle 7 und 9. Ungültige Werte oder eine falsche Anzahl der eingegebenen Zeichen werden mit Fehlermeldungen quittiert.

Mit einem Punkt vor dem Befehl wird der eingestellte Parameter fest im EEPROM des Interface abgespeichert und steht somit als Standardwert nach Trennung der Stromversorgung wieder zur Verfügung.

Befehl: (.)ATCR [xxx | xxxxxxxxx]

Bezeichnung: CAN RX-Filter-Adresse einstellen

Standardwert: 7E8, 18DAF100

Beispiele:

```
>ATCR 760
```

OK

```
>ATCR 18DA22E8
```

OK

Der Empfangsfilter-Adresse für CAN-Pakete wird hiermit eingestellt. Die Anzahl der eingegebenen Zeichen für die CAN-Adresse gibt an, ob diese dem 11-Bit oder dem 29-Bit CAN-Protokoll zugeordnet wird. Eine 11-Bit Adresse ist gültig im Wertebereich 000-7FF und gilt für die Protokolle 6 und 8. Eine 29-Bit Adresse ist gültig im Wertebereich 00000000-1FFFFFFF und gilt für die Protokolle 7 und 9. Ungültige Werte oder eine falsche Anzahl der eingegebenen Zeichen werden mit Fehlermeldungen quittiert.

Mit einem Punkt vor dem Befehl wird der eingestellte Parameter fest im EEPROM des Interface abgespeichert und steht somit als Standardwert nach Trennung der Stromversorgung wieder zur Verfügung.

AGV4000C

OB2-Diagnose-Interface

Befehl: (.) **ATCM** [**xxx** | **xxxxxxxx**]

Bezeichnung: CAN RX-Filter-Maske einstellen

Standardwert: 7F8, 1FFFFFF0

Beispiele:

>ATCR 7F0

OK

>ATCR 1FFFFFF80

OK

Der Empfangsfilter-Maske für CAN-Pakete wird hiermit eingestellt. Die Anzahl der eingegebenen Zeichen für die CAN-Adresse gibt an, ob diese dem 11-Bit oder dem 29-Bit CAN-Protokoll zugeordnet wird. Eine 11-Bit Adresse ist gültig im Wertebereich 000-7FF und gilt für die Protokolle 6 und 8. Eine 29-Bit Adresse ist gültig im Wertebereich 00000000-1FFFFFFF und gilt für die Protokolle 7 und 9. Ungültige Werte oder eine falsche Anzahl der eingegebenen Zeichen werden mit Fehlermeldungen quittiert.

Mit einem Punkt vor dem Befehl wird der eingestellte Parameter fest im EEPROM des Interface abgespeichert und steht somit als Standardwert nach Trennung der Stromversorgung wieder zur Verfügung.

AGV4000C

OBD2-Diagnose-Interface

B: TRANSIT - MODUS

Im Transit-Modus (K/L, K/KL-Modus) wird das AGV4000C Interface in einen Transparent-Modus geschaltet, das heißt, alle seriellen Daten, die über die USB-Schnittstelle übertragen werden, erscheinen ungefiltert und unverändert sofort an der K-Line des OBD-Stecker. Das Interface befindet sich somit in einem passiven Modus, in dem alle Daten vom PC geliefert werden müssen. Programme wie CarPort unterstützen diesen Modus.

C: CAN-INTERFACE-MODUS (EXPERT)

Der AGV4000C Controller besitzt einen CAN-Interface-Modus. Über eine nur wenige Befehle umfassende Schnittstelle können verschiedene Aktionen auf CAN-Bus-System ausgeführt werden:

- Senden von Datenpaketen zum CAN-Bus.
- Empfangen von Datenpaketen auf dem CAN-Bus.
- Protokollieren des Datenverkehrs auf dem CAN-Bus.

Hierzu können verschiedene Parameter eingestellt werden:

- Feste Baudraten zwischen 10kBit und 1 MBit.
- Variable Baudrate durch direkte Eingabe des gewünschten Wertes (10kBit – 1MBit)
- 11 oder 29 Bit CAN-ID.
- 2 verschiedene Empfangsfilter konfigurierbar, auch bei aktivem CAN-Bus veränderbar
- Senden von RTR-Datenanforderungen.
- Zeitstempel beim Empfang der Datenpakete.

Der Befehlssatz des CAN-Interface-Modus ist weitgehend zu anderen existierenden Interfaces kompatibel. Durch den abweichenden Controller-Typ erwarten einige Befehle jedoch ein etwas unterschiedliches Datenformat.

Serielle Datenverbindung

Die Baudrate für die virtuelle serielle Schnittstelle wird vom AGV4000C ignoriert, es wird immer die maximale USB-Geschwindigkeit benutzt. Es wird kein Echo der ankommenden Daten zurückgeschickt. Alle Eingaben und Antworten erfolgen im Klartext als ASCII-Zeichen mit abschließendem CR (ASCII 13). Fehler bei der Eingabe werden mit BELL (ASCII 7) quittiert.

Die Software sollte als erste Zeichen mehrfach ein CR senden, damit eventuell vorhandene Befehlsreste gelöscht und nachfolgende Befehle sicher erkannt werden.

Aufgrund des schnellen USB-Interface zum PC können pro Sekunde bis zu 1000 CAN-Pakete zum PC übertragen werden.

AGV4000C

OBD2-Diagnose-Interface

Befehlssatz

Befehl: ?

Bezeichnung: Liste aller Befehle ausgeben

Beispiel: ? [CR]

Rückgabe: ?TtRrMmHhOLABCNniFGSsVz[CR]

Eine Liste aller möglichen Befehle ausgeben.

Befehl: **S [1..8]**

Bezeichnung: CAN-Bitrate (fest) einstellen, abfragen

Standardwert: S6 (500 kBit)

Beispiel: S5 [CR]

Stellt die Baudrate auf 250kBit ein

Rückgabe: [CR] für OK, [BELL] für Fehler

Beispiel: S [CR]

Zeigt die aktuell eingestellte Bitrate an

Rückgabe: S6: BTR=007FC003 BRP=4 TSEG1=16 TSEG2=8 SJW=4 (250000 bps)

Befehl	Bitrate
S0	10 kBit
S1	20 kBit
S2	50 kBit
S3	100 kBit
S4	125 kBit
S5	250 kBit
S6	500 kBit (standard)
S7	800 kBit
S8	1 MBit

Dieser Befehl kann nicht benutzt werden, wenn der CAN-Bus mit einem der Befehle O, L, A, B aktiviert ist. Unbedingt vorher mit dem Befehl C schließen.

Befehl: **s [n]**

Bezeichnung: CAN-Bitrate (variabel) einstellen, abfragen

Beispiel: s300k [CR]

Stellt die Baudrate auf 300kBit ein

Beispiel: s200000 [CR]

Stellt die Baudrate auf 200kBit ein

Rückgabe: [CR] für OK, [BELL] für Fehler

Beispiel: s [CR]

Zeigt die aktuell eingestellte Bitrate an

Rückgabe: S*: BTR=005CC003 BRP=4 TSEG1=13 TSEG2=6 SJW=4 (312500 bps)

Aufgrund der verwendeten Taktfrequenz des CAN-Controllers können nicht alle Bitraten genau erzeugt werden. Ob eine gewünschte Bitrate genau erzeugt werden kann, sollte mit dem Befehl s geprüft werden. Es sind Bitraten zwischen 10kBit und 1MBit möglich.

Dieser Befehl kann nicht benutzt werden, wenn der CAN-Bus mit einem der Befehle O, L, A, B aktiviert ist. Unbedingt vorher mit dem Befehl C schließen.

AGV4000C

OBD2-Diagnose-Interface

Befehl: **O**

Bezeichnung: CAN-Bus öffnen (Open, Bus aktiv)

Beispiel: O [CR]

Rückgabe: [CR] für OK, [BELL] für Fehler

Die CAN-Kommunikation wird gestartet, Senden und Empfangen von Datenpaketen ist nun möglich.

Dieser Befehl ist nur möglich, wenn der CAN-Bus inaktiv ist.

Befehl: **C**

Bezeichnung: CAN-Bus schließen (Close, Bus inaktiv)

Beispiel: C [CR]

Rückgabe: [CR] für OK, [BELL] für Fehler

Die CAN-Kommunikation wird beendet.

Dieser Befehl kann nur benutzt werden, wenn der CAN-Bus mit einem der Befehle **O**, **L**, **A**, **B** aktiviert ist.

Befehl: **L**

Bezeichnung: CAN-Bus zum Loggen öffnen (Listen only, Bus passiv)

Beispiel: L [CR]

Rückgabe: [CR] für OK, [BELL] für Fehler

Die CAN-Kommunikation wird nur zum loggen der Daten auf dem Bus geöffnet. Der AGV4000C Controller verhält sich absolut passiv. Es können keine CAN-Pakete gesendet sondern nur empfangen werden.

Dieser Befehl kann nur benutzt werden, wenn der CAN-Bus inaktiv ist.

Befehl: **A**

Bezeichnung: CAN-Bus öffnen (Open, no Ack, Bus aktiv)

Beispiel: A [CR]

Rückgabe: [CR] für OK, [BELL] für Fehler

Die CAN-Kommunikation wird gestartet, Senden und Empfangen von Datenpaketen ist nun möglich.

Dieses ist ein Testmodus, der vor allem zur Überprüfung der gesendeten Daten auf dem Bus benötigt wird. Es ist kein 2. CAN-Gerät auf dem Bus erforderlich, das ein gesendetes CAN-Paket bestätigt (acknowledge).

Dieser Befehl ist nur möglich, wenn der CAN-Bus inaktiv ist.

Befehl: **B**

Bezeichnung: CAN-Bus öffnen (Open, Self Receive, Bus aktiv)

Beispiel: B [CR]

Rückgabe: [CR] für OK, [BELL] für Fehler

Die CAN-Kommunikation wird gestartet, Senden und Empfangen von Datenpaketen ist nun möglich.

Dieses ist ein Testmodus, bei dem die gesendeten Datenpakete selber wieder empfangen werden. Dier Modus ist besonders dafür geeignet, um die Empfangsfilter zu testen.

Dieser Befehl ist nur möglich, wenn der CAN-Bus inaktiv ist.

Befehl: **V**

Bezeichnung: Versionsnummer des Bootloaders und Bios ausgeben

Beispiel: V [CR]

Rückgabe: Vxxyy [CR]

Die Versionsnummern des Bootloaders und Bios im AGV4000C werden ausgegeben.

xx = Bootloader-Version (10 = Version 1.0)

yy = Bios-Version (12 = Version 1.2)

AGV4000C

OBD2-Diagnose-Interface

Befehl: **N**

Bezeichnung: Die Seriennummer des AGV4000C wird ausgegeben (kompatibel, kurz)

Beispiel: N[CR]

Rückgabe: Nxxxx[CR]

Zur Kompatibilität mit anderen CAN-Controllern wird die Seriennummer des Gerätes 4-stellig ausgegeben.

Befehl: **n**

Bezeichnung: Die Seriennummer des AGV4000C wird ausgegeben (lang)

Beispiel: N[CR]

Rückgabe: N12345-67890-123[CR]

Die Seriennummer des AGV4000C wird komplett ausgegeben.

Befehl: **i**

Bezeichnung: Identifikation ausgeben

Beispiel: i[CR]

Rückgabe: iAGV4000C * CAN-Mode[CR]

Identifikationstext des verwendeten CAN-Interface ausgeben.

Befehl: **F**

Bezeichnung: CAN-Status abfragen

Beispiel: F[CR]

Rückgabe: Fttrr[CR]

Nach einem Fehler bei der CAN-Übertragung kann hier der Status des CAN-Controllers abgefragt werden.

tt = Anzahl der Fehler bei TX

rr = Anzahl der Fehler bei RX

Befehl: **Z [0 | 1]**

Bezeichnung: Zeitstempel ein-/ausschalten, Status abfragen

Beispiel: Z1[CR]

Rückgabe: [CR] für OK, [BELL] für Fehler

Beispiel: Z[CR]

Rückgabe: z0[CR]

Mit Z1 wird an jede CAN-Nachricht vom Interface zu PC ein 16-Bit Zeitstempel angehängt. Der 16-Bit Wert wird im Millisekunden Bereich erhöht und endet bei 59999d = EA5Fh, ein Durchlauf dauert genau 1 Minute.

Befehl: **M [xxxxxxxx]**

Bezeichnung: Software Filter-Id einstellen (11 oder 29 bit) (Standardwert 00000000)

Beispiel: M7E0[CR]

Rückgabe: [CR] für OK, [BELL] für Fehler

Beispiel: M[CR]

Rückgabe: M000007E0[CR]

29-Bit Filter-Id: Maximalwert 1FFFFFFF.

11-Bit-Filter-Id: Maximalwert 7FF

Es existiert nur ein Filter, der gemeinsam für 11-Bit und 29-Bit-Filterung benutzt wird.
Der Software-Filter kann jederzeit geändert werden.

AGV4000C

OBD2-Diagnose-Interface

Befehl: **m [xxxxxxxx]**

Bezeichnung: Software Filter-Maske einstellen (11 oder 29 bit) (Standardwert 00000000)

Beispiel: m7F0 [CR]

Rückgabe: [CR] für OK, [BELL] für Fehler

Beispiel: m [CR]

Rückgabe: m000007F0 [CR]

29-Bit Filter-Maske: Maximalwert 1FFFFFFF.

11-Bit-Filter-Maske: Maximalwert 7FF

Die Filter-Maske gibt an, welche Bits der Filter-Id zur Paketfilterung benutzt werden sollen. Nur 1-Bits werden zur Filterung benutzt, 0-Bits werden ignoriert.

Beispiel:

```
M7E0      11111100000  ID
m7F0      11111110000  MASKE
-----
          1111110xxxx  FILTER
```

Alle Adressen zwischen 7E0 und 7EF werden vom Filter erkannt und zum PC übertragen.

Es existiert nur eine Maske, die gemeinsam für 11-Bit und 29-Bit-Filterung benutzt wird.

Die Software-Filtermaske kann jederzeit geändert werden.

Befehl: **t [iiindd...]**

Bezeichnung: Sende 11-Bit Datenpaket

Beispiel: t78981122334455667788 [CR]

Rückgabe: [CR] für OK, [BELL] für Fehler

Beispiel: t6001AB [CR]

Rückgabe: [CR] für OK, [BELL] für Fehler

Es wird ein 11-Bit Paket mit der ID *iii* gesendet. *n* gibt die Anzahl der zu sendenden Bytes an. Die Anzahl der Datenbytes *dd* muss exakt mit dem Wert von *n* übereinstimmen.

Die Sendefunktion ist nur aktiv, wenn der CAN-Bus mit einem der Befehle O, A, B zuvor geöffnet wurde.

Befehl: **T [iiiiiiiindd...]**

Bezeichnung: Sende 29-Bit Datenpaket

Beispiel: t1890ABCD81122334455667788 [CR]

Rückgabe: [CR] für OK, [BELL] für Fehler

Beispiel: t1FFF10001AB [CR]

Rückgabe: [CR] für OK, [BELL] für Fehler

Es wird ein 29-Bit Paket mit der ID *iiiiiii* gesendet. *n* gibt die Anzahl der zu sendenden Bytes an. Die Anzahl der Datenbytes *dd* muss exakt mit dem Wert von *n* übereinstimmen.

Die Sendefunktion ist nur aktiv, wenn der CAN-Bus mit einem der Befehle O, A, B zuvor geöffnet wurde.

AGV4000C

OB2-Diagnose-Interface

Befehl: **r[*iiin*]**

Bezeichnung: Sende 11-Bit RTR-Datenanforderung (Remote Transmission Request)

Beispiel: r7890[CR]

Rückgabe: [CR] für OK, [BELL] für Fehler

Beispiel: t6008[CR]

Rückgabe: [CR] für OK, [BELL] für Fehler

Es wird eine 11-Bit RTR-Datenanforderung mit der ID *iii* gesendet. *n* gibt die Anzahl der anzufordernden Datenbytes an. Es werden keine Datenbytes gesendet.

Die Sendefunktion ist nur aktiv, wenn der CAN-Bus mit einem der Befehle O, A, B zuvor geöffnet wurde.

Befehl: **R[*iiiiiiin*]**

Bezeichnung: Sende 29-Bit RTR-Datenanforderung (Remote Transmission Request)

Beispiel: R1890ABCD0[CR]

Rückgabe: [CR] für OK, [BELL] für Fehler

Beispiel: R1FFF10008[CR]

Rückgabe: [CR] für OK, [BELL] für Fehler

Es wird eine 29-Bit RTR-Datenanforderung mit der ID *iiiiiii* gesendet. *n* gibt die Anzahl der anzufordernden Datenbytes an. Es werden keine Datenbytes gesendet.

Die Sendefunktion ist nur aktiv, wenn der CAN-Bus mit einem der Befehle O, A, B zuvor geöffnet wurde.

AGV4000C

OBD2-Diagnose-Interface

Datenpakete

Zum Senden und Empfangen der CAN-Datenpakete wurde folgendes Format festgelegt. Alle Daten bestehen aus lesbaren ASCII-Zeichen, alle Werte werden im Hexadezimal-Format übertragen. Sende- und Empfangspakete sind nahezu identisch, bei den empfangenen Daten kann optional ein Zeitstempel angehängt werden (siehe Befehl Z).

11 Bit-CAN-Datenpaket senden

`tiiiindd...`

`t` = Kennung für 11-Bit Paket mit Daten

`iii` = 11-Bit Sende-Id

`n` = Anzahl der nachfolgenden Bytes

`dd` = Zu sendende Datenbytes (2 Hexzeichen pro Byte)

29 Bit-CAN-Datenpaket senden

`Tiiiiiiiindd...`

`T` = Kennung für 29-Bit Paket mit Daten

`iiiiiii` = 29-Bit Sende-Id

`n` = Anzahl der nachfolgenden Bytes

`dd` = Zu sendende Datenbytes (2 Hexzeichen pro Byte)

11 Bit-CAN-RTR-Paket anfordern

`riiin`

`r` = Kennung für 11-Bit RTR-Anforderung (Remote Transmission Request)

`iii` = 11-Bit Sende-Id

`n` = Anzahl der angeforderten Bytes

Bei einer RTR-Anforderung werden keine Daten gesendet. `n` gibt in diesem Fall die Anzahl der Bytes an, die die Gegenstelle auf die Anforderung senden soll.

29 Bit-CAN-RTR-Paket anfordern

`Riiiiiiiin`

`R` = Kennung für 29-Bit RTR-Anforderung (Remote Transmission Request)

`iiiiiii` = 29-Bit Sende-Id

Bei einer RTR-Anforderung werden keine Daten gesendet. `n` gibt in diesem Fall die Anzahl der Bytes an, die die Gegenstelle auf die Anforderung senden soll.

11 Bit-CAN-Datenpaket empfangen

`tiiiindd...zzzz`

`t` = Kennung für 11-Bit Paket mit Daten

`iii` = 11-Bit Empfangs-Id

`n` = Anzahl der nachfolgenden Bytes

`dd` = Die empfangenen Datenbytes (2 Hexzeichen pro Byte)

`zzzz` = Optionaler Zeitstempel in Millisekunden, wenn mit Befehl Z1 aktiviert.

AGV4000C

OBD2-Diagnose-Interface

29 Bit-CAN-Datenpaket empfangen

Tiiiiiiiindd...zzzz

T = Kennung für 29-Bit Paket mit Daten

iiiiiii = 29-Bit Empfangs-Id

n = Anzahl der nachfolgenden Bytes

dd = Die empfangenen Datenbytes (2 Hexzeichen pro Byte)

zzzz = Optionaler Zeitstempel in Millisekunden, wenn mit Befehl Z1 aktiviert.

11 Bit-RTR-Anforderung empfangen

riiinzzzz

r = Kennung für 11-Bit-Anforderung (Remote Transmission Request)

iii = 11-Bit Empfangs-Id

n = Anzahl der angeforderten Bytes

zzzz = Optionaler Zeitstempel in Millisekunden, wenn mit Befehl Z1 aktiviert.

29 Bit-RTR-Anforderung empfangen

Riiiiiiiinzzzz

R = Kennung für 29-Bit-Anforderung (Remote Transmission Request)

iiiiiii = 29-Bit Empfangs-Id

n = Anzahl der angeforderten Bytes

zzzz = Optionaler Zeitstempel in Millisekunden, wenn mit Befehl Z1 aktiviert.

AGV4000C

OBd2-Diagnose-Interface

Rechtliche Hinweise

© Erwin Reuß; Folker Stange. Nutzung und Weitergabe dieser Informationen auch auszugsweise nur mit Erlaubnis der Copyright-Inhaber. 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 des AGV4000C entstehen könnten.

Links

Informationen zur Treiberinstallation unter Windows 8.x und 10:

<http://forum.diamex.de/showthread.php?1099-Treiberinstallation-unter-Windows-8-x-und-10>

Programm MoDiag:

<http://www.sotima-solutions.de>

Programm CarPort:

<https://carport-diagnose.de>

Vertrieb



DIAMEX Produktion und Handel GmbH

Innovationspark Wuhlheide
Köpenicker Straße 325, Haus 41
12555 Berlin

Telefon: 030-65762630
E-Mail: info@diamex.de
Homepage: <http://www.diamex.de>

Herstellung



www.tremex.de

Köpenicker Str. 325 12555 Berlin
Tel. 030-65762631

Hersteller: Tremex GmbH
DIAMEX × OBd-DIAG × TREMEX
WEE-Reg.Nr. DE 51673403