



# OBD2-SIMULATOR DX-M

## TECHNIK, BEDIENUNGSANLEITUNG

© 2020-2021 DIAMEX GMBH

Erwin Reuß und Folker Stange

# OBD2-Simulator DX-M

Technik, Bedienung

---

## Inhaltsverzeichnis

Bedienelemente und Display an der Vorderseite .....	2
Anschlüsse und Schalter an der Rückseite .....	2
Stromversorgung .....	3
Verbindung D-Sub9-Anschluss mit der OBD-Buchse.....	4
Was ist OBD2? .....	5
Geschichte der OBD2-Simulatoren .....	5
Fähigkeiten des OBD2-Simulator DX-M .....	7
Hardware .....	10
Software .....	10
Bedienungsanleitung .....	11
USB-Schnittstelle, Treiber.....	16
USB-Schnittstelle, AT-Befehle .....	16
Fehlermeldungen .....	17
AT-Befehlstabelle Übersicht .....	18
Beschreibung der AT-Befehle .....	19
Auf Werkseinstellungen zurücksetzen .....	26
Fehler, Bugs .....	26
Spezialversionen und Updates .....	26
Versionen .....	27
v1.0 - v1.3 .....	27
V1.4.....	27
Hinweise .....	28
Links .....	28
Literaturnachweise .....	28
Kontakt .....	29

# OBD2-Simulator DX-M

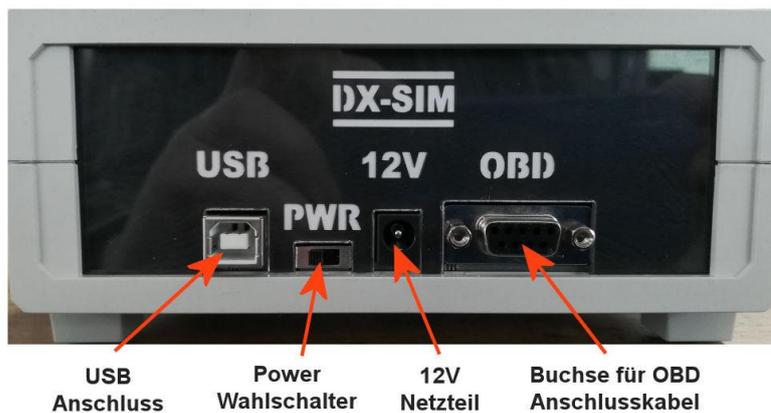
Technik, Bedienung

## Bedienelemente und Display an der Vorderseite



- RGB-LC-Display, 1,8", 160 x 128 Pixel
- System-RGB-LED, flackert blau bei OBD2-Datenübertragungen.
- Funktionstasten rechts neben dem Display, Hinweis auf die jeweilige Funktion durch Symbole auf dem Display.
- Drehregler mit Tastfunktion, Funktion abhängig vom Menüpunkt
- Alle Funktionstasten und die Taste am Drehregler können je nach Menüpunkt unterschiedliche Funktionen auslösen, wenn sie **kurz** oder **lange** gedrückt werden. Lesen Sie hierzu bitte die Anleitung zu den Menüfunktionen.

## Anschlüsse und Schalter an der Rückseite



- USB-A Buchse, zur Stromversorgung und zur Verbindung mit dem PC.
- Stromversorgungsbuchse, für Steckernetzteil 12V.
- Schiebeschalter, zur Auswahl der Stromversorgung für die OBD2-Anschlussbuchse.
- 9-poliger D-Sub-Anschluss für OBD-Adapterkabel.

*Hinweise: Es befindet sich kein Ein/Aus-Schalter am Gerät. Der Simulator wird durch Einstecken des USB- oder 12V-Steckers in Betrieb genommen und bleibt so lange aktiv, bis beide Stecker entfernt wurden. Der Power-Wahlschalter beeinflusst nur die Stromquelle für die OBD2-Anschlussbuchse, die Stromversorgung des Simulators erfolgt automatisch über USB- oder 12V-Anschluss.*

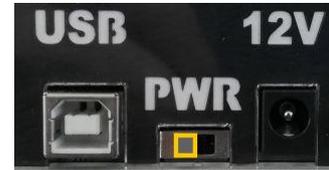
# OBD2-Simulator DX-M

Technik, Bedienung

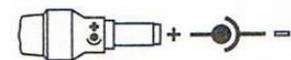
## Stromversorgung

Die Stromversorgung des OBD2-Simulator DX-M kann über zwei verschiedene Varianten geschehen, diese werden mit dem Schiebeschalter an der Rückseite des Gerätes ausgewählt:

1. Ein Highlight des OBD2-Simulators DX-M ist die Versorgung nur über die USB-Buchse (Schalterstellung nach LINKS, Richtung USB-Buchse). Einer der Einsatzorte dieser Variante ist z.B. der Messestand, an dem keine Steckdose für ein Steckernetzteil in der Nähe ist. Hier kann der Simulator über eine handelsübliche 5V Powerbank über das USB-Kabel mit Strom versorgt werden. Ein an die OBD-Buchse angeschlossene Diagnoseadapter erhält über einen im Simulator integrierten Spannungswandler die erforderliche 12 Volt Stromversorgung. Bitte beachten Sie dabei, dass die Stromaufnahme des Diagnoseadapters die USB-Stromversorgung nicht überlastet. Dies ist z.B. daran zu erkennen, dass das Display des Simulators während des Betriebes dunkler wird oder flackert. Sollte das Display nur beim Einstecken des Diagnoseadapters in den Simulator einmal kurz flackern, ist dies normal und kann ignoriert werden. Bricht die Spannung jedoch ständig zusammen, empfehlen wir die Verwendung von Variante 2.



2. Hierfür benötigen Sie ist ein 12 Volt Netzteil (Schalterstellung nach RECHTS, Richtung 12-Volt Buchse), das zur Stromversorgung des Simulators und einem angeschlossenen Diagnoseinterface dient. Das Netzteil sollte bei 12 Volt genügend Strom für den Simulator und das angeschlossene Diagnoseinterface liefern können und einen 5,5 mm Hohlstecker besitzen. Achten Sie bitte auf die Polarität, Plus muss sich in der Mitte befinden. Ein im Simulator eingebauter Verpolungsschutz verhindert Beschädigungen bei falscher Polarität. Die Spannung des Steckernetzteils steht direkt an PIN 16 der OBD-Buchse des Simulators zur Verfügung.



Die Stromaufnahme des Simulators ohne angeschlossenen Diagnoseadapter beträgt abhängig von der eingestellten Helligkeit des Displays zwischen 100 und 180 mA. Im Idealfall sollte das 12V Netzteil mindestens 500mA Strom liefern können.

**ACHTUNG!** Sollte das Display nach Anstecken des Diagnoseinterface an den Simulator erlöschen, liegt wahrscheinlich ein Problem mit dem Diagnoseinterface vor. Eine integrierte Sicherung (Polyswitch) schützt die Schaltung in diesem Fall. Ziehen Sie bitte sofort den OBD-Stecker und/oder die Stromversorgung (Netzteil, USB) vom Simulator ab. Es kann mehrere Sekunden dauern, bis sich die Sicherung selbstständig zurückgesetzt hat und der Simulator wieder einsatzbereit ist. Ein Auswechseln der Sicherung ist nicht erforderlich.

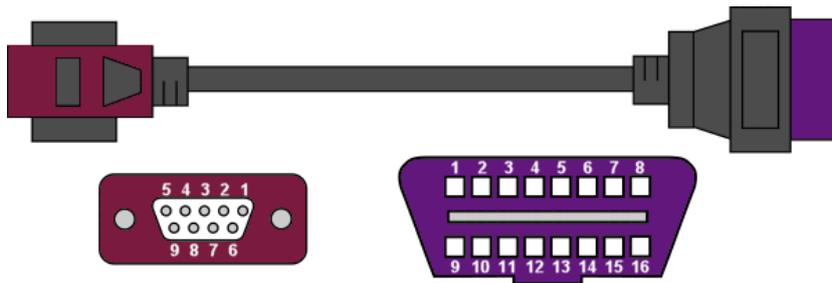
# OBD2-Simulator DX-M

Technik, Bedienung

## Verbindung D-Sub9-Anschluss mit der OBD-Buchse

Stecken Sie den OBD2-Diagnoseadapter an die OBD-Buchse des mitgelieferten Verbindungskabels an und verbinden den D-Sub-Stecker mit der OBD2-Buchse an der Rückseite des Simulators. Dies kann auch gefahrlos während des Betriebs des Simulators geschehen.

Hinweis: Sollten Sie die OBD-Buchse am Verbindungskabel häufig stark beanspruchen, kann es zu Kontaktproblemen oder Kabelbruch kommen. Denken Sie bitte rechtzeitig daran, immer ein Ersatzkabel in Reserve zu haben (Bezugsquelle: Diamex-Shop).



DSUB9 male		OBD female
PIN1	Signalmasse (S-GND)	PIN5
PIN2	Fahrzeugmasse (C-GND)	PIN4
PIN3	CAN High	PIN6
PIN4	K-Line	PIN7
PIN5	CAN Low	PIN14
PIN6	PWM Minus	PIN10
PIN7	PWM Plus, VPWM	PIN2
PIN8	L-Line (nicht benutzt)	PIN15
PIN9	+12 Volt	PIN16

# OBD2-Simulator DX-M

Technik, Bedienung

---

## Was ist OBD2?

Zur Beantwortung dieser Frage verweisen wir gerne auf die vielen im Internet vorhandenen Artikel und Beschreibungen.

Hier nur einige Antworten auf Fragen, die und immer wieder gestellt wurden:

OBD bezeichnet die generelle Onboard-Diagnose bei Fahrzeugen. Diese kann je nach Hersteller unterschiedlich sein und die verschiedensten Funktionen umfassen. OBD2 ist genormt und für alle Fahrzeuge vorgeschrieben. Während die Grundfunktionen wie Service-IDs und das Hardware-Protokoll vorgeschrieben sind, können die Hersteller dennoch die unterstützten PIDs der Service-IDs festlegen sowie eigene PIDs hinzufügen.

Mit OBD2 ist es nicht möglich, auf sicherheitsrelevante Daten eines Fahrzeuges zuzugreifen oder diese zu verändern. Auch ist kein Motortuning oder Tachomanipulation darüber möglich. Es gibt derzeit insgesamt 9 verschiedene OBD2-Protokolle, die an der OBD2-Diagnosebuchse von verschiedenen Herstellern benutzt werden können. Dabei wurde PWM vor allem bei der Firma Ford und VPWM bei den meisten amerikanischen Fahrzeugen eingesetzt. ISO und KWP2000 benutzen die meisten europäischen Fahrzeughersteller. Inzwischen setzen so ziemlich alle Hersteller bei Neufahrzeugen das CAN-Protokoll in einer seiner 4 Varianten ein. Und... es gibt kein OBD1 und kein OBD3.

Alle Diamex-Simulatoren unterstützen nur die genormten OBD2-Protokolle. Fahrzeugspezifische Onboard-Diagnose ist derart komplex und die Protokolle und Befehle sind von den meisten Herstellern nicht öffentlich verfügbar oder nur gegen enorme Summen zu erhalten.

## Geschichte der OBD2-Simulatoren

Angefangen hat alles im Jahr 2005, als wir begonnen haben, die ersten AGV-OBD2-Diagnosegeräte zu entwickeln. Weil wir uns keine realen Fahrzeuge an den Entwickler-PC stellen konnten, haben wir uns mehrere Simulatoren eines Herstellers besorgt, der damals führend auf dem europäischen Markt war. Es waren jedoch keine Multi-Protokoll-Simulatoren, für jedes OBD2-Protokoll musste ein eigener Simulator her, was bei der Entwicklung eines Multi-Protokoll-Diagnoseinterfaces sehr umständlich war. Außerdem hatten diese Simulatoren einen sehr begrenzten Funktionsumfang, waren besonders im CAN-Bereich sehr fehlerhaft und zudem sehr anfällig für elektrostatische Ladungen. So passierte es uns mehrfach, dass der teure Microcontroller auf den Simulatoren ausgetauscht werden musste, was uns der Hersteller selbstverständlich gegen einen nicht geringen Betrag angeboten hatte, aber die Ausfallzeiten wogen schlimmer, als die für den Austausch erforderlichen Gelder.

So begannen wir, unseren ersten eigenen OBD2-Simulator zu entwickeln. Es entstand nach vielen Monaten im Jahr 2008 der Diamex-OBD2-Simulator, der noch heute im Diamex-Shop verkauft wird und quasi ein Standard bei den OBD2-Simulatoren in Europa und darüber hinaus geworden ist. Herz dieses Simulators ist ein 8-Bit AVR-Microcontroller mit 16 MHz Taktfrequenz, 32 Kilobyte Flash und 2 Kilobyte RAM. Zum Zeitpunkt der Entwicklung einer der besten verfügbaren CAN-fähigen Microcontroller.

⇒ Zum Vergleich: Der neue OBD2-Simulator DX-M besitzt einen 32-Bit ARM-Microcontroller mit 168 MHz Taktfrequenz, 128 Kilobyte Flash und 32 Kilobyte RAM.

Da unsere Kunden eine Erweiterung des Funktions- und Konfigurationsumfanges des Diamex-Simulators wünschten, haben wir uns schon vor einigen Jahren nach neuer Hardware für einen größeren Simulator umgeschaut. Da die Microcontroller-Technik so schnell voranschreitet, dauerte es wieder mehrere Jahre, bis wir die richtige Hardware gefunden haben. In der Zwischenzeit entstanden etliche



# OBD2-Simulator DX-M

## Technik, Bedienung

---

Prototypen des Nachfolgers für den DiameX-Simulator, die aber jedes Mal aus unterschiedlichen Gründen verworfen wurden. Erst Ende 2019 war der Startschuss für den neuen Simulator gefallen, nachdem eine neue Generation von sehr leistungsfähigen 32-Bit ARM-Cortex Microcontrollern des Herstellers STMicroelectronics auf den Markt gekommen war. Auch bei diesem Simulator waren mehrere Anläufe erforderlich, bis die endgültige Hardware zusammengestellt und der erste funktionstüchtige Prototyp fertig war.

Auch wenn der OBD2-Simulator DX-M schon viele Kundenwünsche erfüllt, planen wir für die Zukunft. Bedienung per Touchscreen oder per Smartphone sowie reale Simulationen von gespeicherten Fahrdaten auf SD-Karte sind zum Beispiel Wünsche der Entwickler, die vielleicht demnächst in eine kommende Generation der Simulatoren Einzug finden.

# OBD2-Simulator DX-M

Technik, Bedienung

## Fähigkeiten des OBD2-Simulator DX-M

Ein OBD2-Simulator, so modern und umfangreich er auch sein mag, wird nie die Steuergeräte im Fahrzeug vollständig simulieren können. Es fehlen ihm einfach die Werte von den etlichen im Fahrzeug verbauten Sensoren. So sind viele Sensorwerte nur statisch vorhanden, einige können über den Einstellregler des Simulators verändert werden. Alle diese Werte werden nie den realen Werten entsprechen, da viele Werte auch im normalen Fahrbetrieb untereinander abhängig sind sich gegenseitig beeinflussen können. Da ein OBD2-Simulator meist dafür eingesetzt wird, Diagnoseadapter auf ihre Funktion zu testen, sind realistische Werte normalerweise auch nicht erforderlich.

**Der OBD2-Simulator DX-M unterstützt alle 9 verschiedenen OBD2-Protokolle:**

P1	J1850 PWM
P2	J1850 VPWM
P3	ISO 9141-2
P4	ISO 14230-4 (KWP2000, slow init)
P5	ISO 14230-4 (KWP2000, fast init)
P6	ISO 15765-4 CAN 11/500
P7	ISO 15765-4 CAN 29/500
P8	ISO 15765-4 CAN 11/250
P9	ISO 15765-4 CAN 29/250

Die Spannungspegel und das Timing aller Protokolle entsprechen den mittleren empfohlenen Werten in den SAE-Spezifikationen.

**Folgende Service-IDs werden unterstützt:**

\$01	Abfrage von Diagnosedaten
\$02	Abfrage der Freeze-Frame-Daten
\$03	Gespeicherte Fehlercodes (DTC)
\$04	Löschen der Fehlercodes
\$07	Sporadische Diagnose Fehlercodes
\$09	Abfrage von Fahrzeuginformationen (nur ECU1)
\$0A	Permanente Fehlercodes (nur CAN-Protokolle)

Laut SAE J1979 wird Service-ID \$0A nur von Steuergeräten mit CAN-Protokoll unterstützt.

Nicht aufgelistete Service-IDs werden vom Simulator nicht beantwortet.

Bei allen Protokollen sind Antworten von einem oder zwei Steuergeräten möglich.

**Service-ID \$01 PID-Liste:**

PID	Beschreibung	ECU1	ECU2	Var.
\$00	PID used to determine PID support for PIDs 01-20	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
\$01	Monitor status since DTCs cleared	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
\$03	Fuel system 1 status	<input checked="" type="checkbox"/>		
\$04	Calculated LOAD Value	<input checked="" type="checkbox"/>		<input checked="" type="checkbox"/>
\$05	Engine Coolant Temperature	<input checked="" type="checkbox"/>		<input checked="" type="checkbox"/>
\$0A	Fuel Rail Pressure (gauge)	<input checked="" type="checkbox"/>		
\$0B	Intake Manifold Absolute Pressure	<input checked="" type="checkbox"/>		
\$0C	Engine RPM	<input checked="" type="checkbox"/>		<input checked="" type="checkbox"/>
\$0D	Vehicle Speed Sensor	<input checked="" type="checkbox"/>		<input checked="" type="checkbox"/>
\$0F	Intake Air Temperature	<input checked="" type="checkbox"/>		<input checked="" type="checkbox"/>
\$10	Air Flow Rate from Mass Air Flow Sensor	<input checked="" type="checkbox"/>		
\$11	Absolute Throttle Position		<input checked="" type="checkbox"/>	
\$13	Location of Oxygen Sensors	<input checked="" type="checkbox"/>		

# OBD2-Simulator DX-M

## Technik, Bedienung

\$14	Bank 1 – Sensor 1	<input checked="" type="checkbox"/>		
\$15	Bank 1 – Sensor 2	<input checked="" type="checkbox"/>		
\$16	Bank 1 – Sensor 3	<input checked="" type="checkbox"/>		
\$17	Bank 1 – Sensor 4	<input checked="" type="checkbox"/>		
\$1C	OBD requirements to which vehicle is designed	<input checked="" type="checkbox"/>		
\$1F	Time Since Engine Start		<input checked="" type="checkbox"/>	
\$20	PID used to determine PID support for PIDs 21-40		<input checked="" type="checkbox"/>	
\$21	Distance Travelled While MIL is Activated		<input checked="" type="checkbox"/>	
\$2F	Fuel Level Input		<input checked="" type="checkbox"/>	
\$30	Number of warm-ups since DTC cleared		<input checked="" type="checkbox"/>	
\$31	Distance since DTC cleared		<input checked="" type="checkbox"/>	

**ECU1:** PID wird von ECU1 unterstützt.

**ECU2:** PID wird von ECU2 unterstützt.

**Var.:** Variabel, Wert am Simulator einstellbar

PIDs können in der aktuellen Firmware-Version nicht gelöscht oder hinzugefügt werden.

Bei den CAN-Protokollen (6-9) ist es möglich, bis zu 6 PIDs mit einem Befehl abzufragen. Hierbei dürfen jedoch nicht die Support-PIDs \$00, \$20 usw. mit den Daten-PIDs gemischt werden.

Beispiele:

01 00 20 40 (gültig)

01 04 0C 0D 1F (gültig)

01 00 0C 20 (ungültig)

### Service-ID \$02 Freeze-Frame PID-Liste:

PID	Beschreibung	ECU1	ECU2
\$04	Calculated LOAD Value	<input checked="" type="checkbox"/>	
\$05	Engine Coolant Temperature	<input checked="" type="checkbox"/>	
\$0C	Engine RPM	<input checked="" type="checkbox"/>	
\$0D	Vehicle Speed Sensor	<input checked="" type="checkbox"/>	
\$0F	Intake Air Temperature	<input checked="" type="checkbox"/>	
\$11	Absolute Throttle Position		<input checked="" type="checkbox"/>
\$1F	Time Since Engine Start		<input checked="" type="checkbox"/>

**ECU1:** Freeze-Frame Daten für ECU1 werden gespeichert.

**ECU2:** Freeze-Frame Daten für ECU2 werden gespeichert.

### Service-ID \$09 Fahrzeug-Informationen Infotype-Liste:

Infotype	Beschreibung	PVIK	CAN
\$00	Used to determine support for Infotype 01-20	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
\$01	MessageCount VIN	<input checked="" type="checkbox"/>	
\$02	VIN	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
\$03	MessageCount Calibration ID	<input checked="" type="checkbox"/>	
\$04	Calibration ID	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
\$05	MessageCount Calibration Verification Number	<input checked="" type="checkbox"/>	
\$06	Calibration Verification Number	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
\$07	MessageCount In-use Performance Tracking	<input checked="" type="checkbox"/>	
\$08	In-use Performance Tracking	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

Service-ID \$09 wird nur von ECU1 unterstützt

# OBD2-Simulator DX-M

## Technik, Bedienung

---

PVIK = Protokolle: PWM-P1, VPWM-P2, ISO9141-P3, KWP2000-P4/P5

CAN = Protokolle: P6/P7/P8/P9

### Steuergeräte IDs:

		ECU1	ECU2
P1	J1850 PWM	\$10	\$18
P2	J1850 VPWM	\$10	\$18
P3	ISO 9141-2	\$10	\$18
P4	ISO 14230-4 (KWP2000, slow init)	\$10	\$18
P5	ISO 14230-4 (KWP2000, fast init)	\$10	\$18
P6	ISO 15765-4 CAN 11/500	\$7E0	\$7E1
P7	ISO 15765-4 CAN 29/500	\$10	\$18
P8	ISO 15765-4 CAN 11/250	\$7E0	\$7E1
P9	ISO 15765-4 CAN 29/250	\$10	\$18

Über die Tasten und den Drehregler können einige Service-ID 01 PID-Werte im laufenden Betrieb verändert und maximal 8 Fehlercodes pro Steuergerät aktiviert werden. Die FIN/VIN (Fahrzeug-Identifizierungsnummer, Fahrgestellnummer) ist direkt am Gerät mit Hilfe der Bedienelemente veränderbar.

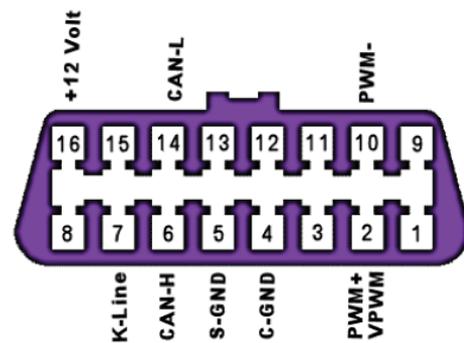
Weitaus mehr Möglichkeiten zur Modifikation der OBD2-Antworten des Simulators sind über den USB-Anschluss möglich. Mit Hilfe eines Terminal-Programmes auf dem PC können viele Werte mittels AT-Befehlen abgefragt und verändert werden. So ist es z.B. möglich, die Fehlercodes (DTC) und die Service-ID 09 Antworten zu verändern. Auch ist die Modifikation vieler Service-ID 01 Parameter möglich, die nicht über die Bedienelemente verändert werden können. Eine genaue Beschreibung aller AT-Befehle findet sich weiter unten in dieser Anleitung.

# OBD2-Simulator DX-M

Technik, Bedienung

## Hardware

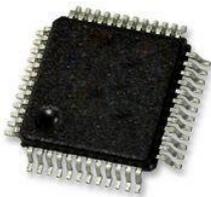
Gesteuert wird der OBD2-Simulator DX-M von einem ARM-Cortex M4 Microcontroller mit 168 MHz Taktfrequenz. 128 Kilobyte Flash und 32 Kilobyte RAM ermöglichen diesem Controller neben der Low-Level Ansteuerung der OBD2-Schnittstelle auch die Abfrage der Tasten und des Drehreglers, des USB-Ports, der Ansteuerung des LC-Displays und die Ausgabe von Signaltönen über den integrierten Mini-Lautsprecher. In einem EEPROM mit 8 Kilobyte Speichergröße werden wichtige Parameter permanent gespeichert. Drei Taster sowie ein Drehimpulsgeber mit Tastenfunktion dienen der Bedienung am Simulator. Ein Farb-LC-Display mit 160 x 128 Pixeln dient zur Visualisierung der Parameter und der Einstellenmenüs. Für die verschiedenen OBD2-Protokolle werden unterschiedliche Interfaces eingesetzt. Der CAN-Treiberchip verbindet den Microcontroller mit den Pins 6 (CAN-HI) und 14 (CAN-LOW) der OBD2-Buchse. Ein K-Line Treiberbaustein wird für die ISO- und KWP2000-Protokolle (Pin 7) eingesetzt. Eine Treiberschaltung mit mehreren Transistoren und Operationsverstärkern ist für die Signale PWM+/VPWM (Pin 2) und PWM- (Pin 10) zuständig. Die L-Leitung (Pin 15) wird beim OBD2-Simulator DX-M nicht abgefragt, diese wird nur bei sehr wenigen Fahrzeugen tatsächlich benutzt und dient bei diesen zur Reizung des Steuergerätes oder zur Übertragung von nicht OBD2-kompatiblen Daten. Pin 4 (Chassis-Ground) und Pin 5 (Signal-Ground) der OBD-Buchse sind zusammengeschaltet und liegen gemeinsam auf Massepotential.



Für die Stromversorgung der Simulator-Hardware werden mehrere Spannungsregler verwendet, die für die ausreichende Versorgung des Microcontrollers und normgerechten Spannungspegeln auf der OBD2-Schnittstellen zuständig sind.

## Software

Da kein bekannter handelsüblicher Microcontroller alle OBD2-Protokolle von sich aus beherrscht, ist sehr viel Software für die Erkennung und Erzeugung der Protokoll-Signale erforderlich. In einigen Teilen unterstützt die Hardware dies zwar, wie z.B. durch den integrierten CAN-Bus und die UART-Schnittstelle für die ISO- und KWP2000 Protokolle, aber der eigentliche OBD2-Rahmen muss per Software erzeugt werden. Vor allem die beiden PWM-Protokolle P1-PWM und P2-VPWM können nur durch Software generiert und auf die Anfragen vom Diagnoseadapter ausgewertet werden. Da der Microcontroller mit einer recht hohen Frequenz getaktet wird, bleibt auch noch genug Zeit um nebenbei die Bedienelemente abzufragen und das Display anzusteuern. Die Prioritäten in der Software wurden hierbei jedoch immer auf das Behandeln der OBD2-Signale gelegt.



Die Firmware des OBD2-Simulator DX-M ist „Linear“ aufgebaut, ein Realtime-Betriebssystem kann hier aufgrund des engen Timings bei der Erzeugung der Protokolle nicht verwendet werden.

Ein Update der Firmware ist über die USB-Schnittstelle möglich. Der integrierte Bootloader kann nicht überschrieben werden und übernimmt den Startvorgang des Simulators. Sollte es dennoch vorkommen, dass sich der Simulator nicht mehr starten lässt, trennen Sie den Simulator von der Stromversorgung (12V Netzteil und USB-Anschluss) halten Taste 1 gedrückt und stecken den USB-Stecker ein. Das Display bleibt dunkel und die System-LED blinkt rot. Nun kann das Firmware-Update erneut aufgespielt werden.

# OBD2-Simulator DX-M

Technik, Bedienung

## Bedienungsanleitung

Über die Tasten, dem Drehregler mit Tastfunktion sowie der visuellen Darstellung auf dem Farbdisplay können mehrere Funktionen direkt am OBD2-Simulator DX-M eingestellt werden. Die Menütexte sind für den internationalen Markt auf Englisch gehalten und sollten trotzdem auch für jeden verständlich sein, der nicht über die englische Sprache verfügt. Natürlich werden alle Texte hier in dieser Anleitung auf Deutsch erklärt.

Das Display hat keine Touch-Funktion, Symbole am rechten Rand erklären die Funktionen der rechts daneben befindlichen Tasten. Der Drehregler stellt verschiedene Werte ein oder dient zur Auswahl von Menüpunkten. Die Taste des Drehreglers hat verschiedene Funktionen, die hier jeweils erklärt werden.

### PROGRAMMSTART

Nach Anlegen der Stromversorgung wird kurz der Startbildschirm mit der aktuellen Version der Firmware angezeigt. Die hier im Beispiel angezeigte Versionsnummer kann von Ihrem Gerät abweichen.



### HAUPTANZEIGE

Die Hauptanzeige besteht aus drei Teilen:

1. Kopfzeile mit Anzeige des Protokolls, Verbindungsstatus, Nummern der Steuergeräte und MIL-Status.
2. Anzeige der einstellbaren PID-Werte. Der große grüne Wert in der Mitte der Liste kann über den Drehregler geändert werden.
3. Funktionstasten-Symbole am rechten Rand.



### DREHREGLER:

Der grün angezeigte Wert wird über den Drehregler verändert. Die gelbe Markierung unter der Anzeige gibt die Schrittweite an, die bei jedem Impuls des Drehreglers zum Wert addiert, bzw. subtrahiert wird. Standard ist hier 1. Soll die Einstellung in größeren Schritten vorgenommen werden, ändert ein kurzer Druck auf den Drehregler die Markierung und damit die Schrittweite in 10, 100 oder 1000. Nicht bei allen PID-Einstellungen sind alle Schrittweiten verfügbar, so ist z.B. bei der Einstellung der Geschwindigkeit (VSS) nur eine Schrittweite von 1 und 10 möglich. Bei der Einstellung der Drehzahl (RPM) sind jedoch Schrittweiten von 1, 10, 100 und 1000 möglich. Die Einstellung wird immer auf den minimalen und maximalen Wert der jeweiligen PID begrenzt und kann auch bei großer Schrittweite nicht über- oder unterschritten werden. Ein langer Druck auf den Drehregler speichert den aktuellen Wert permanent ab (langer Signalton), so dass dieser auch nach Neustart sofort wieder zur Verfügung steht.

### FUNKTIONSTASTEN:

Taste 1 = Kurzer Druck öffnet das Fehlercode-Menü (DTC), langer Druck öffnet das Konfigurationsmenü (CONFIG).

Taste 2 = Liste der einstellbaren PIDs nach oben scrollen.

Taste 3 = Liste der einstellbaren PIDs nach unten scrollen.

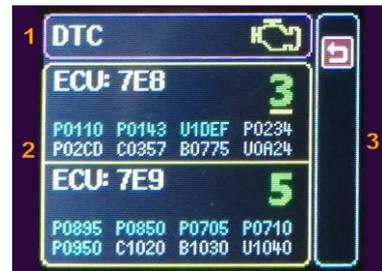
# OBD2-Simulator DX-M

## Technik, Bedienung

### FEHLERCODE-MENÜ (DTC)

Das Fehlercode-Menü besteht aus folgenden Teilen:

1. Kopfzeile mit Anzeige des MIL-Status.
2. Anzeige und Einstellmöglichkeit der Fehlercodes für Steuergerät 1 und 2 (nur wenn 2 Steuergeräte im Protokoll gewählt sind).
3. Funktionstasten-Symbole am rechten Rand.



### DREHREGLER:

Die Markierung unter der grünen Ziffer gibt an, ob der Einstellregler für Steuergerät 1 oder Steuergerät 2 aktiv ist. Kurzer Druck auf die Taste des Einstellreglers wechselt zwischen Steuergerät 1 und 2. Steuergerät 2 ist nur verfügbar, wenn ein Protokoll mit 2 Steuergeräten ausgewählt ist. Der in der Kopfzeile angezeigte MIL-Status ist automatisch aktiv, wenn einer der beiden Werte ungleich Null ist. Die aktiven Fehlercodes sind in der jeweiligen Liste blau markiert. Die Nummern der Fehlercodes können über den AT-Befehlssatz bzw. ein externes Einstellprogramm verändert werden. Ein langer Druck auf den Einstellregler bewirkt den Rücksprung zur Hauptanzeige.

Die Anzahl der Fehlercodes kann nicht permanent abgespeichert werden. Bei Neustart des Simulators sind immer alle Fehlercodes beider Steuergeräte abgeschaltet.

### FUNKTIONSTASTEN:

Taste 1 = Fehlercode-Menü verlassen und zur Hauptanzeige zurückkehren (wie langer Druck auf den Drehregler).

Taste 2 = Ohne Funktion.

Taste 3 = Ohne Funktion.

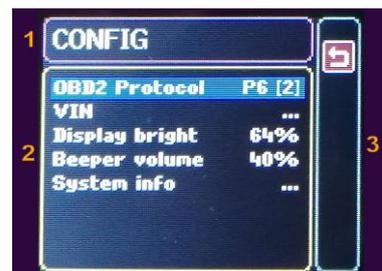
Die eingestellten Fehlercodes beider Steuergeräte werden vom angeschlossenen Diagnoseadapter mit Service-ID \$04 auf null gesetzt.

Ein Rücksprung zur Hauptanzeige erfolgt automatisch nach 30 Sekunden, wenn keine Eingabe vorgenommen wird.

### EINSTELL-MENÜ (CONFIG)

Das Einstell-Menü besteht aus folgenden Teilen:

1. Kopfzeile.
2. Anzeige der Menüpunkte und aktuellen Werten.
3. Funktionstasten-Symbole am rechten Rand.



### DREHREGLER:

Änderung des blau unterlegten Menüpunktes durch Drehen des Reglers und Auswahl durch kurzen Druck auf die Taste des Reglers. Ein langer Druck auf den Einstellregler bewirkt den Rücksprung zur Hauptanzeige.

### FUNKTIONSTASTEN:

Taste 1 = Einstell-Menü verlassen und zur Hauptanzeige zurückkehren (wie langer Druck auf den Drehregler).

Taste 2 = Ohne Funktion.

Taste 3 = Ohne Funktion.

Ein Rücksprung zur Hauptanzeige erfolgt automatisch nach 30 Sekunden, wenn keine Eingabe vorgenommen wird.

# OBD2-Simulator DX-M

## Technik, Bedienung

### CONFIG: OBD2-PROTOCOL

Die Einstellung des Protokolls besteht aus folgenden Teilen:

1. Kopfzeile.
2. Liste der verfügbaren Protokolle mit blau unterlegten aktuell gewählten Protokoll.
3. Funktionstasten-Symbole am rechten Rand.



### DREHREGLER:

Änderung des blau unterlegten Protokolls durch Drehen des Reglers. Durch langen Druck auf die Taste des Einstellreglers wird das gewählte Protokoll abgespeichert und aktiviert, das Protokoll-Menü verlassen und zum Einstell-Menü zurückgekehrt.

### FUNKTIONSTASTEN:

Taste 1 = Gewähltes Protokoll abspeichern und aktivieren, Protokoll-Menü verlassen und zum Einstell-Menü zurückkehren (wie langer Druck auf die Taste des Einstellreglers).

Taste 2 = Ohne Funktion.

Taste 3 = Wechsel zwischen einem oder zwei Steuergeräten.

### CONFIG: EDIT VIN

Die Einstellung der Fahrzeug-Identifizierungsnummer besteht aus folgenden Teilen:

1. Kopfzeile.
2. Anzeige der aktuellen Fahrzeug-Identifizierungsnummer (17-stellig)
3. Funktionstasten-Symbole am rechten Rand.



### DREHREGLER:

Änderung der gelb markierten Position durch Drehen des Reglers. Laut ISO-Norm 3779 dürfen nur die folgenden Zeichen in der Fahrzeug-Identifizierungsnummer vorkommen:

1234567890ABCDEFGHIJKLMN P RSTUVWXYZ

Die Buchstaben I, O und Q dürfen wegen der Verwechslungsgefahr mit 0 und 1 nicht verwendet werden. Diese können mit dem Einstellregler auch nicht angewählt werden.

Ein kurzer Druck auf die Taste des Einstellreglers verschiebt die Markierung der Eingabeposition um eine Stelle nach rechts. Ein langer Druck übernimmt die aktuell angezeigte Fahrzeug-Identifizierungsnummer, speichert diesen permanent ab und kehrt zum Einstell-Menü zurück.

### FUNKTIONSTASTEN:

Taste 1 = Aktuell angezeigte Fahrzeug-Identifizierungsnummer übernehmen, permanent abspeichern und zum Einstell-Menü zurückkehren (wie langer Druck auf den Drehregler).

Taste 2 = Abbruch der Einstellung, alle Änderungen werden verworfen.

Taste 3 = Markierung der Eingabeposition um eine Stelle nach Links verschieben.

# OBD2-Simulator DX-M

## Technik, Bedienung

### CONFIG: DISPLAY-BRIGHT

Die Einstellung der Display-Helligkeit besteht aus folgenden Teilen:

1. Kopfzeile.
2. Anzeige der aktuellen Helligkeit in Prozent.
3. Funktionstasten-Symbole am rechten Rand.



### DREHREGLER:

Änderung der Display-Helligkeit (5..100%) durch Drehen des Reglers. Beachten Sie bitte, dass die minimale Helligkeit 5% beträgt und das Gerät damit weiterhin zu bedienen ist. Ein langer Druck auf die Taste des Einstellreglers speichert den aktuellen Wert permanent ab und kehrt zum Einstell-Menü zurück.

### FUNKTIONSTASTEN:

Taste 1 = Aktuell angezeigten Wert permanent abspeichern und zum Einstell-Menü zurückkehren (wie langer Druck auf den Drehregler).

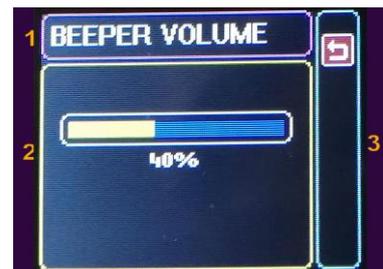
Taste 2 = Keine Funktion.

Taste 3 = Keine Funktion.

### CONFIG: BEEPER VOLUME

Die Einstellung der Lautstärke des im Gerät eingebauten Signalgebers besteht aus folgenden Teilen:

1. Kopfzeile.
2. Anzeige der aktuellen Lautstärke in Prozent.
3. Funktionstasten-Symbole am rechten Rand.



### DREHREGLER:

Änderung der Lautstärke (0..100%) durch Drehen des Reglers. Ein langer Druck auf die Taste des Einstellreglers speichert den aktuellen Wert permanent ab und kehrt zum Einstell-Menü zurück.

### FUNKTIONSTASTEN:

Taste 1 = Aktuell angezeigten Wert permanent abspeichern und zum Einstell-Menü zurückkehren (wie langer Druck auf den Drehregler).

Taste 2 = Keine Funktion.

Taste 3 = Keine Funktion.

# OBD2-Simulator DX-M

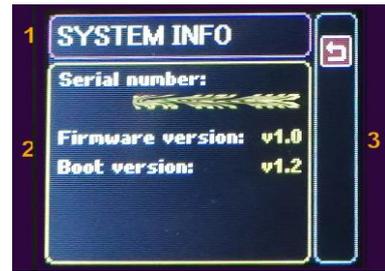
## Technik, Bedienung

---

### CONFIG: SYSTEM INFO

Die Anzeige der System-Informationen besteht aus folgenden Teilen:

1. Kopfzeile.
2. Anzeige der Seriennummer, Firmware- und Bootloader-Version.
3. Funktionstasten-Symbole am rechten Rand.



### DREHREGLER:

Ein kurzer Druck auf die Taste kehrt zum Einstell-Menü zurück.

### FUNKTIONSTASTEN:

Taste 1 = Zum Einstell-Menü zurückkehren (wie kurzer Druck auf den Drehregler).

Taste 2 = Keine Funktion.

Taste 3 = Keine Funktion.

Dieser Menüpunkt dient nur zur Anzeige der Seriennummer sowie der Firmware- und Bootloader-Version. Obiges Beispiel kann von Ihrem Gerät abweichen.

Es sind keine Einstellmöglichkeiten vorhanden. Bitte geben Sie bei Problemen oder Fragen zu Ihrem Gerät immer diese Informationen an.

# OBD2-Simulator DX-M

Technik, Bedienung

---

## USB-Schnittstelle, Treiber

Wir empfehlen die Benutzung des OBD2-Simulators DX-M unter Windows 10. Hier ist keine Installation eines Treiber erforderlich. Der OBD2-Simulator DX-M meldet sich automatisch als „Seriellles USB-Gerät“ an und es wird ein COM-Port zugewiesen. Welcher dies ist, können sie am besten herausfinden indem sie sich die Liste der COM-Ports im Terminalprogramm vor und nach Anstecken des Simulators anschauen.

Unter Windows 7 und Windows 8.x muss die Treiberdatei „diamex-cdc.inf“ über den Gerätemanager geladen werden. Wie das funktioniert, finden sie in den Links am Ende dieser Anleitung.

*Hinweis: Bei Windows 8.x muss eventuell noch die Treibersignierung abgeschaltet werden. Hinweise hierzu finden sie, wenn Sie im Internet nach „Treibersignatur Windows 8“ suchen.*

## USB-Schnittstelle, AT-Befehle

Vom PC aus über die USB-Schnittstelle sind viele Funktionen und Parameter des OBD2-Simulator DX-M einstellbar. So sind auch Einstellungen möglich, die nicht über die Bedienelemente und das Display verändert werden können.

Am USB-Anschluss steht ein virtueller COM-Port zur Verfügung, der mit jedem Terminalprogramm (wie z.B. Hyper-Terminal unter Windows) angesprochen werden kann. Stellen sie bitte im Terminalprogramm den passenden COM-Port ein. Die Baudrate und die Protokollparameter werden aufgrund der USB-Verbindung ignoriert, hier kann jeder beliebige Wert eingestellt werden, es werden immer die maximale USB-Geschwindigkeit sowie die Standardparameter 8N1 benutzt.

Alle Eingaben erfolgen nach dem Promptzeichen >. Nach Ausführung eines Befehles muss der Anwender auf dieses Zeichen warten, um einen erneuten Befehl einzugeben. Ein eingegebener 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.

Alle Befehle beginnen mit den Zeichen „AT“, dieses wurde von den Steuerbefehlen bei Modems übernommen und bedeutet **AT**tention. Viele OBD2-Diagnoseadapter, wie z.B. die Diamex-Adapter AGV4000 arbeiten ebenfalls mit einem ähnlichen Befehlssatz.

In den folgenden Beispielen muss jede Eingabe mit dem Zeilenende-Zeichen abgeschlossen werden, es wird nicht extra angegeben.

### **Beispiele:**

**at i**

Wird intern nach ATI gewandelt

**A T Z**

Wird intern nach ATZ gewandelt

Falls das Zeilenende-Zeichen CR ausbleibt, wird der Befehl automatisch nach 10 Sekunden abgebrochen und ein „?“ wird als Fehlermeldung ausgegeben.

Befehle, die der Simulator nicht versteht oder Falscheingaben bei Werten, werden mit einer Fehlermeldung im Klartext quittiert

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.

# OBD2-Simulator DX-M

## Technik, Bedienung

---

Eine Liste aller AT-Befehle wird durch Eingabe von

>?

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

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**)

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

## Fehlermeldungen

Beschreibung
<b>ERROR: unknown command</b> Der eingegebene AT-Befehl ist unbekannt.
<b>ERROR: syntax</b> Es wurde falsche Zeichen eingegeben oder ein Parameter fehlt.
<b>ERROR: wrong value</b> Der eingegebene Wert über- oder unterschreitet die gültigen Werte.
<b>ERROR: no sid/pid</b> Die angesprochene SID (Service-ID) und/oder PID (Parameter-ID) existiert nicht.

# OBD2-Simulator DX-M

Technik, Bedienung

---

## AT-Befehlstabelle Übersicht

Hinweise zur Befehlstabelle:

- AT-Befehl des OBD2-Simulator DX-M.
- Die Spalte „Memory“ gibt an, welche Parameter abgespeichert werden können.
- Standardwerte sind **Fett** dargestellt

Befehl	Memory	Funktionsbezeichnung
ATI		Gibt die Interface-ID und Version aus
ATE	<input checked="" type="checkbox"/>	Terminal-Echo [ <b>ein</b> /aus]
ATL	<input checked="" type="checkbox"/>	Linefeed (0x0A, ASCII 10) nach Carriage Return [ <b>ein</b> /aus]
ATZ		Warmstart Reset, Standardwerte wiederherstellen
ATSB	<input checked="" type="checkbox"/>	Helligkeit des Displays einstellen [ <b>20</b> ]
ATSV	<input checked="" type="checkbox"/>	Lautstärke des Signalgebers einstellen [ <b>2</b> ] (0 = aus)
ATP	<input checked="" type="checkbox"/>	OBD2 Protokoll und Anzahl der Steuergeräte einstellen [ <b>6:1</b> ]
ATAE		Steuergeräte (ECU) Adressen ausgeben
AT3C		Anzahl DTC für Service-ID \$03 setzen [ <b>0:0</b> ]
AT3D	<input checked="" type="checkbox"/>	DTC für Service ID \$03 setzen
AT7C		Anzahl DTC für Service-ID \$07 setzen [ <b>2:1</b> ]
AT7D	<input checked="" type="checkbox"/>	DTC für Service ID \$07 setzen
ATAC		Anzahl DTC für Service-ID \$0A setzen (nur CAN) [ <b>1:0</b> ]
ATAD	<input checked="" type="checkbox"/>	DTC für Service ID \$0A setzen
ATSP	<input checked="" type="checkbox"/>	Setze Service-ID \$01 PID-Wert
ATGL		Lese PID-Listen für Service-ID \$01, \$02, \$09
ATUD		Werte auf dem Display aktualisieren
ATKL		Eingabetasten und Drehregler sperren (Key-Lock) [ <b>ein/aus</b> ]
ATVIN	<input checked="" type="checkbox"/>	VIN (Fahrzeug-Identifizierungsnummer) ändern
AT!		Sonderfunktionen, Anzeigen interner Daten

# OBD2-Simulator DX-M

Technik, Bedienung

---

## 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.
- Wenn ein Befehl ohne Parameter eingegeben wird, antwortet der Simulator mit dem aktuell eingestellten Wert.

---

Befehl: **ATI**

Bezeichnung: Zeigt die Interface-ID an

Beispiel:

```
>ATI
```

```
DIAMEX OBD2-Simulator DX-M v1.0
```

Der Typ und die Firmware-Version des Simulators wird ausgegeben.

---

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
```

```
> .ATE OFF
```

```
>ATE
```

```
E1
```

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. Bei der Auswertung der Antworten mit Microcontrollern ist es sinnvoller, ohne Echo zu arbeiten.

---

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

Bezeichnung: Schaltet die Ausgaben von Linefeed ein/aus.

Standardwert: 1 ON

Beispiele:

```
>ATL1
```

```
> .ATL OFF
```

```
>ATL
```

```
L1
```

# OBD2-Simulator DX-M

## Technik, Bedienung

---

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.

---

Befehl: **ATZ**

Bezeichnung: Warmstart Reset

Beispiel:

```
>ATZ
```

```
RESET
```

Dieser Befehl bricht alle Funktionen ab und der Simulator 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.

---

Befehl: **(.)ATSB [1..31]**

Bezeichnung: Helligkeit des Displays (**S**et **B**acklight)

Standardwert: 20

Beispiele:

```
>ATSB15
```

```
>.ATSB24
```

```
>ATSB
```

```
SB20
```

Die Helligkeit des Displays wird mit dieser Funktion eingestellt. 1 = dunkel, 31 = hell. Aus Sicherheitsgründen kann das Display nicht komplett abgeschaltet werden, damit noch eine Einstellung am Gerät vorgenommen werden kann, wenn kein PC angeschlossen ist.

---

Befehl: **(.)ATSV [0..5]**

Bezeichnung: Lautstärke des Signalgebers, Beeper (**S**et **V**olume)

Standardwert: 2

Beispiele:

```
>ATSV3
```

```
>.ATSV1
```

```
>ATSV
```

```
SV2
```

Die Lautstärke des Signalgebers wird mit dieser Funktion eingestellt. 0 = aus, 5 = Maximum.

---

# OBD2-Simulator DX-M

## Technik, Bedienung

---

Befehl: **(.)ATP [1..9] (: [1..2])**

Bezeichnung: OBD2-Protokoll und Anzahl der Steuergeräte einstellen

Standardwert: 6:1

Beispiele:

```
>ATP3
```

```
>.ATP1:2
```

Der erste Parameter gibt das gewünschte Protokoll an (Siehe Liste im Kapitel „Fähigkeiten des OBD2-Simulators DX-M“). Wenn kein zweiter Parameter angegeben wird, ist automatisch nur ein Steuergerät eingestellt. Mit Eingabe des zweiten Parameters können zwei Steuergeräte aktiviert werden.

Beachten Sie, dass bei diesem Befehl immer das Protokoll initialisiert und eine eventuelle Verbindung getrennt wird. Es ist vom angeschlossenen Diagnoseadapter danach bei den Protokollen 3, 4 und 5 unbedingt eine Init-Sequenz erforderlich.

```
>ATP
```

```
P1:2:U
```

Bei der Abfrage des aktuellen Protokolls werden als weitere Parameter die Anzahl der eingestellten Steuergeräte und der Verbindungsstatus zurückgegeben (U = nicht verbunden, unconnected, C = verbunden, connected). In diesem Fall wird das Protokoll nicht initialisiert und die Verbindung wird nicht getrennt.

---

Befehl: **ATAE**

Bezeichnung: Adressen der Steuergeräte (ECU) ausgeben

Beispiel:

```
>ATAE
```

```
AE:10:18
```

Die Adressen aller Steuergeräte werden ausgegeben. Auch wenn nur ein Steuergerät aktiv ist, werden immer die Adressen beider Steuergeräte ausgegeben. Hier sollte vorher mit ATP die Anzahl der aktiven Steuergeräte ausgelesen werden.

---

Befehl: **AT3C [1..2] : [0..8]**

Bezeichnung: Anzahl der Diagnostic Trouble Codes (DTC) für SID \$03 setzen

Standardwert: 3C1:0, 3C2:0

Beispiele:

```
>AT3C2 (gibt die Anzahl der Fehlercodes für ECU 2 aus)
```

```
>AT3C1:4
```

Der erste Parameter gibt die Nummer des Steuergerätes an, für das die Fehler gesetzt werden sollen. Der zweite Parameter gibt die Anzahl der Fehlercodes an.

Diese Werte beeinflussen auch den Status der MIL (Motorkontrollleuchte). Ist in einem Steuergerät ein Fehlercode vorhanden, wird die MIL eingeschaltet. Nur wenn in beiden Steuergeräten keine Fehlercodes vorhanden sind, ist die MIL aus.

# OBD2-Simulator DX-M

## Technik, Bedienung

---

Befehl: **(.)AT3D [1..2] : [0..7] : [DTC]**

Bezeichnung: Diagnostic Trouble Code (DTC) für SID \$03 setzen

Beispiele:

>AT3D1:0:P1234

>AT3D1 (gibt die Liste aller Codes für ECU1 aus)

Der erste Parameter gibt die Nummer des Steuergerätes an, für das der Fehlercode gesetzt werden soll. Der zweite Parameter gibt die Nummer des Fehlercodes an. Der dritte Wert (DTC) muss einen gültigen Fehlercode darstellen, das erste Zeichen = P,C,B,U, danach ein 4-stelliger Hexcode 0000..3FFF.

---

Befehl: **AT7C [1..2] : [0..8]**

Bezeichnung: Anzahl der Diagnostic Trouble Codes (DTC) für SID \$07 setzen

Standardwert: 7C1:2, 7C2:1

Beispiele:

>AT7C1 (gibt die Anzahl der Fehlercodes für ECU 1 aus)

>AT7C1:2

Der erste Parameter gibt die Nummer des Steuergerätes an, für das die Fehler gesetzt werden sollen. Der zweite Parameter gibt die Anzahl der Fehlercodes an.

---

Befehl: **(.)AT7D [1..2] : [0..7] : [DTC]**

Bezeichnung: Diagnostic Trouble Code (DTC) für SID \$07 setzen

Beispiele:

>AT7D1:0:P1234

>AT7D2 (gibt die Liste aller Codes für ECU2 aus)

Der erste Parameter gibt die Nummer des Steuergerätes an, für das der Fehlercode gesetzt werden soll. Der zweite Parameter gibt die Nummer des Fehlercodes an. Der dritte Wert (DTC) muss einen gültigen Fehlercode darstellen, das erste Zeichen = P,C,B,U, danach ein 4-stelliger Hexcode 0000..3FFF.

---

Befehl: **ATAC [1..2] : [0..8]**

Bezeichnung: Anzahl der Diagnostic Trouble Codes (DTC) für SID \$0A setzen

Standardwert: AC1:1, AC2:0

Beispiele:

>ATAC1 (gibt die Anzahl der Fehlercodes für ECU 1 aus)

>ATAC1:4

Der erste Parameter gibt die Nummer des Steuergerätes an, für das die Fehler gesetzt werden sollen. Der zweite Parameter gibt die Anzahl der Fehlercodes an.

Beachten Sie bitte, dass Service-ID \$0A nur von den CAN-Protokollen unterstützt wird.

---

# OBD2-Simulator DX-M

## Technik, Bedienung

---

Befehl: **(.)ATAD [1..2] : [0..7] : [DTC]**

Bezeichnung: Diagnostic Trouble Code (DTC) für SID \$0A setzen

Beispiele:

```
>ATAD1:0:P1234
```

```
>ATAD1 (gibt die Liste aller Codes für ECU1 aus)
```

Der erste Parameter gibt die Nummer des Steuergerätes an, für das der Fehlercode gesetzt werden soll. Der zweite Parameter gibt die Nummer des Fehlercodes an. Der dritte Wert (DTC) muss einen gültigen Fehlercode darstellen, das erste Zeichen = P,C,B,U, danach ein 4-stelliger Hexcode 0000..3FFF.

---

Befehl: **ATUD**

Bezeichnung: Aktualisiert die angezeigten Display-Inhalte (**Update Display**)

Beispiel:

```
>ATUD
```

Werden Parameter über die USB-Verbindung verändert, kann es sein, dass die angezeigten Werte auf dem Display nicht mit den tatsächlichen Werten übereinstimmen. In diesem Fall sollte der Display-Inhalt mit diesem Befehl aktualisiert werden.

---

Befehl: **ATKL**

Bezeichnung: Sperren der Bedienungselemente am Gerät (**Key Lock**)

Beispiel:

```
>ATKL ON
```

```
>ATKL 0
```

Wird der Simulator mittels AT-Befehlen bedient, würden Eingaben über die Tasten oder dem Drehregler am Gerät Werte verfälschen oder unstimmig machen. Hier sollten sofort nach Start des Terminalprogrammes die Bedienelemente gesperrt und vor Beenden wieder freigegeben werden. Sollte die Freigabe ausbleiben, kann das Gerät erst wieder bedient werden, wenn die Stromversorgung kurz unterbrochen wurde oder der Freigabebefehl über das Terminalprogramm erfolgt. Diese Einstellung kann nicht permanent abgespeichert werden, Standardwert ist 0/OFF.

---

Befehl: **(.)ATSP [xx] : [value]**

Bezeichnung: PID (xx) - Wert für Service-ID \$01 ändern (**Set Pid**)

Beispiel:

```
>.ATSP 0C:1200
```

```
>ATSP 0C
```

```
SPOC:1200:4B0 (Ausgabe des aktuellen Wertes in dezimal und hex)
```

Der erste Parameter gibt die PID-Nummer als zweistellige Hexnummer an (z.B. 0C = RPM). Der zweite Parameter gibt den Wert an, der dieser PID-Nummer zugewiesen wird. Dieser Wert kann als Dezimalwert (z.B. 123) oder als Hexwert (z.B. 0x70) eingegeben werden.

---

# OBD2-Simulator DX-M

## Technik, Bedienung

---

Wenn ein Wert an eine nicht vom System unterstützte PID zugewiesen werden soll, erscheint die Fehlermeldung „Error: no sid/pid“.

Die PIDs \$00, \$20, \$40 usw. können nicht verändert werden.

Bei PID \$01 können nur die unteren 24 Bit geändert werden, da die oberen 8 Bit vom System mit dem MIL-Status und der Anzahl der Fehlercodes aufgefüllt werden.

Der Maximalwert für „value“ ist von der PID abhängig. Hier können nur die reinen positiven Ganzzahlwerte eingegeben werden, z.B. wird eine Temperaturangabe von -40 bis +215° über die Dezimalwerte 0 = -40° bis 255 = +215° eingegeben. Eine genaue Kenntnis der Werte ist hierfür erforderlich. Diese sind im Datenblatt SAE J1979 detailliert aufgeführt.

---

**Befehl: ATGL [1..2] : [1|2|9]**

Bezeichnung: PID-Liste für Service-ID \$01, \$02 oder \$09 lesen

Beispiel:

```
>ATGL 1:1 (Pid-Liste SID $02 für ECU1 lesen)
```

```
GL1:01:B87B3E10
```

```
>ATGL 1:2 (Pid-Liste SID $01 für ECU2 lesen)
```

```
GL1:02:80008003:80038000
```

```
>ATGL 1:9 (Pid-Liste SID $09 für ECU1 lesen)
```

```
GL1:09:55000000
```

```
>ATGL 2:2 (Pid-Liste SID $02 für ECU2 lesen)
```

```
GL2:02:40008002
```

Der erste Parameter gibt die Nummer des Steuergerätes an, dessen PID-Liste ausgelesen werden soll. Der zweite Wert entspricht dem auszulesenden Service-ID. Gültig sind hier nur die Werte 1, 2 oder 9. Als Antwort wird eine Liste mit 32-Bit Hex-Werten ausgegeben, bei denen jedes gesetzte Bit für eine vom Simulator unterstützte PID steht. Hier gilt Bit32 = PID \$01, Bit31 = PID \$02 usw. Ist Bit0 gesetzt (PID \$20), wird ein weiterer Wert ausgegeben.

---

**Befehl: (.) ATVIN**

Bezeichnung: VIN (Fahrzeug-Identifizierungsnummer) ändern

Beispiel:

```
>.ATVIN ABC123DEF456GHJ78
```

```
>ATVIN
```

```
VINABC123DEF456GHJ78
```

Anzeigen/ändern der 17-stelligen Fahrzeug-Identifizierungsnummer. Es müssen immer genau 17 Zeichen eingegeben werden und es dürfen nur die folgenden Zeichen benutzt werden:

```
1234567890ABCDEFGHIJKLMNPRSTUVWXYZ
```

Bei falscher Länge oder ungültigen Zeichen wird eine Fehlermeldung ausgegeben und der neue Wert nicht übernommen.

---

# OBD2-Simulator DX-M

## Technik, Bedienung

---

Befehl: **AT!** [**xx**]

Bezeichnung: Sonderfunktionen

Mit diesem Befehl werden interne Daten des OBD2-Simulator DX-M ausgelesen.

Seriennummer anzeigen:

```
>AT!00
```

```
1234-5678-9012
```

Geräte-Identifikation anzeigen:

```
>AT!01
```

```
SIMDXM-14
```

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

# OBD2-Simulator DX-M

Technik, Bedienung

---

## Auf Werkseinstellungen zurücksetzen

So setzen Sie den OBD2-Simulator DX-M auf die Werkseinstellungen zurück:

- Stromversorgung entfernen (USB- und 12V-Stecker ziehen).
- Tasten 2 und 3 drücken und festhalten.
- USB- oder 12V-Stecker einstecken, System-LED leuchtet weiß.
- Tasten gedrückt halten, bis die System-LED grün blinkt.
- Tasten loslassen, Simulator startet mit Standardwerten neu.



## Fehler, Bugs

Keine Software ist fehlerfrei. Obwohl für die Entwicklung der Firmware des Diamex OBD2-Simulator DX-M viele Arbeitsstunden aufgewendet und unzählige Funktionstests durchgeführt wurden, sind Fehler in den Funktionen oder der Bedienung nicht auszuschließen. Wenn Sie Fehler finden, kontaktieren Sie uns am besten per Mail an die am Ende dieser Anleitung angegebene E-Mail-Adresse. Beschreiben Sie den Fehler bitte konkret und vergessen Sie bitte nicht, den Typ des verwendeten Diagnoseadapters anzugeben. Auch sollten Sie immer die Seriennummer und die Firmware-Version Ihres OBD2-Simulator DX-M bei allen Mitteilungen an uns angeben.

## Spezialversionen und Updates

Auch wenn der Diamex OBD2-Simulator DX-M schon über viele Funktionen verfügt, die zum Testen von Diagnoseadaptern ausreichen sollten, besteht bei einigen Anwendern eventuell der Wunsch, zusätzliche Funktionen zu implementieren. Sprechen Sie uns bitte an, wenn Sie eine bestimmte Funktion benötigen. Wir werden dies hier intern beraten und entscheiden, ob diese Funktion möglich ist, ob sie in einer öffentlichen oder in einer für den Kunden entwickelten Spezialversion integriert werden sollte und in welchem Zeitraum diese realisiert werden kann.

Der Diamex OBD2-Simulator DX-M verfügt über einen Bootloader, über den Firmware-Updates möglich sind. Ein Update-Tool wird bei Bedarf verfügbar sein, die Bedienungsanleitung und die Änderungen werden dem Tool als Dokument beigelegt sein.

# OBD2-Simulator DX-M

Technik, Bedienung

---

## Versionen

### **v1.0 - v1.3**

Interne Testversionen und Versionen für externe Tester.

### **v1.4**

Erste öffentliche Version.

Fehler im CAN-Protokoll beseitigt, Consecutive-Frame Zähler korrigiert.

Fehler bei Service \$09, Infotype \$08 im CAN-Protokoll beseitigt.

Neue AT-Befehle: ATAE, ATKL

# OBD2-Simulator DX-M

Technik, Bedienung

---

## 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.

## Links

Diamex-Shop:

<http://www.diamex.de>

Zum Thema OBD2:

[https://www.blafusel.de/obd/obd2\\_start.html](https://www.blafusel.de/obd/obd2_start.html)

<https://de.wikipedia.org/wiki/On-Board-Diagnose>

Treiber installieren in Windows 7 und 8.x:

<https://www.giga.de/downloads/windows-10/tipps/treiber-installieren-in-windows-10-7-8-und-co.-so-gehts/>

Treibersignatur ausschalten unter Windows 8.x:

<https://www.pc-magazin.de/ratgeber/windows-8-1-treibersignierung-ausschalten-abschalten-deaktivieren-windows-8-tipp-1939952.html>

## Literaturnachweise

Für die Erstellung der Hard- und Software des OBD2-Simulator DX-M wurden mehrere SAE-Datenblätter von [www.sae.org](http://www.sae.org) benutzt.

- SAE J1979\_2006 edition
- SAE Standard J1850, Class B Data Communication Network Interface (PWM, VPWM)
- ISO 14230-2 (KWP2000)
- ISO 9141-2 (ISO)
- ISO 15765-2 (CAN)

# OBD2-Simulator DX-M

Technik, Bedienung

---

## Kontakt

Wenn Sie Kontakt mit uns aufnehmen wollen, schreiben Sie uns am besten eine E-Mail an die unten angegebene Adresse. Wir leiten diese an den zuständigen Sachbearbeiter weiter. Geben Sie bei allen Anfragen unbedingt auch die **Seriennummer** und die **Firmware-Version** Ihres OBD2-Simulator DX-M an. Diese Angaben finden Sie im CONFIG-Menü unter „System info“.

### VERTRIEB



#### DIAMEX Produktion und Handel GmbH

Innovationspark Wuhlheide  
Köpenicker Straße 325, Haus 41  
12555 Berlin

Telefon: 030-65762631

E-Mail: [info@diamex.de](mailto:info@diamex.de)

Homepage: <http://www.diamex.de>

### HERSTELLUNG



[www.tremex.de](http://www.tremex.de)

Köpenicker Str. 325 12555 Berlin  
Tel. 030-65762631

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