

Thomas Beck

Pi-OBD Handheld GUI – HHGui

OBD2 Software für das DIAMEX® Pi-OBD Modul

Bedienungsanleitung

Version vom 04.01.2017 / 15:50:39 für HHGui V2.81

Hinweise

Copyright © 2016, 2017 Thomas Beck

Alle Rechte vorbehalten.

Vervielfältigung und Verbreitung auch auszugsweise nur mit schriftlicher Erlaubnis des Copyright Inhabers.

Der Autor übernimmt keine Haftung für Schäden, die durch die Anwendung der in diesem Dokument beschriebenen HHGui Software entstehen können.

Der Autor übernimmt keine Haftung für die Vollständigkeit und Korrektheit der in diesem Dokument gemachten Angaben.

Die in diesem Dokument verwendeten Bezeichnungen können auch dann Markennamen und eingetragene Warenzeichen sein, wenn nicht explizit darauf hingewiesen wird. Sie dienen nur der Beschreibung.

Der Autor ist für Fragen, Korrekturhinweise etc. unter [tbeck At-Zeichen freenet Punkt de](mailto:tbeck@freenet.de) erreichbar.

Inhaltsverzeichnis

1 Einleitung.....	1
2 Installation.....	1
3 Betriebsarten und Bedienung.....	2
3.1 Betriebsart OBD2 Software.....	2
3.2 Bedienung der OBD2 Software.....	3
3.3 Betriebsart OBD2 Simulator.....	4
4 Menüführung.....	8
4.1 Main Menu (Hauptmenü).....	8
4.2 Start Diagnostics (Starte Diagnose).....	8
4.3 System Information.....	9
4.4 System Settings (System Einstellungen).....	9
4.5 Diagnostics Menu (Diagnose Menü).....	9
4.5.1 Trouble Codes Menu (Fehlercodes Menü).....	10
4.5.1.1 Sicherheitsabfrage Fehler Löschen.....	11
4.5.1.2 Fehlerumgebungsdaten.....	11
4.5.2 I/M Readiness (TÜV Prüfbereitschaft).....	12
4.5.2.1 Monitor Settings (Monitor Anzeige Einstellungen).....	12
4.5.3 Vehicle/ECU Info (Fahrzeug/Steuergeräte Information).....	13
4.5.4 Protocol Information.....	13
5 Unterstützte OBD2 Dienste und Unterfunktionen.....	14
5.1 Auflistung der Unterfunktionen (PIDs) für Services 0x01 und 0x02.....	14
5.2 Auflistung der Unterfunktionen (InfoTypes) für Service 0x09.....	17
6 Einschränkungen.....	18

1 Einleitung

Die Pi-OBd Handheld GUI Software HHGui ist eine für den Raspberry Pi und das DIAMEX Pi-OBd Modul angepasste Variante des Elektor OBd2-Analyser NG / DIAMEX Handheld Open Emulators HHEmu. Dieser verwendet intern die Firmware des OBd2-Analyser NGs. Deswegen kann die Anleitung in weiten Teilen auch für die HHEmu Software und für den OBd2-Analyser NG benutzt werden.

Die Beschreibung des HHGui/Pi-OBd Modul Projekts findet sich hier:

<http://www.elektormagazine.com/labs/obd2-for-raspberry-pi>

Die Beschreibung des HHEmu/OBd2-Analyser NG Firmware Update Projekts findet sich hier:

<http://www.elektormagazine.com/labs/firmware-update-and-emulator-for-obd2-analyser-ng-wireless-obd2>

Da beide Projektbeschreibungen und die Beschreibungen der Projekt-Updates inzwischen ziemlich umfangreich und in englischer Sprache verfasst sind, wird hier eine deutsche Zusammenfassung präsentiert.

2 Installation

Die Pi-OBd Handheld GUI Software HHGui verwendet das GTK3+ Toolkit zur Realisierung seines GUIs. Auf dem Raspberry Pi/Raspbian mit LXDE Desktop ist GTK3+ bereits installiert, so dass die benötigten GTK3+ shared Libraries bereits vorhanden sind.

Die Programmdatei *hhgui* aus dem *HHGui_HHEmu_OBd2_Software.zip* Archiv kann in jeden beliebigen Ordner kopiert werden. Falls die Datei nach dem Entpacken nicht ausführbar ist, in der Shell folgendes Kommando eingeben:

```
sudo chmod +x ./hhgui
```

Damit ist die Basisinstallation bereits abgeschlossen.

Falls der Raspberry Pi mit einem 7" Touchscreen oder einem ähnlich großen Display betrieben wird, kann zusätzlich das Hintergrundbild *HHGuiBackgroundPic.png* in denselben Ordner wie die *hhgui* Programmdatei kopiert werden, so dass das Hintergrundbild beim Programmstart gefunden wird. Das hat Auswirkungen auf die Darstellung und die Bedienung des Programms, was im nächsten Kapitel beschrieben wird.

Da der Raspberry Pi in der Anwendung als OBd2 Tester mit einem Touchscreen ohne Maus und Tastatur betrieben werden soll, empfiehlt sich das Erzeugen eines Menüeintrags zum Starten des Programms über das Hauptmenü des Raspberry Pi LXDE Desktops. Alternativ kann das im nächsten Kapitel beschriebene Shellskript *initobd.sh* auf dem Desktop abgelegt werden, so dass es von dort einfach per Doppelklick gestartet werden kann.

3 Betriebsarten und Bedienung

3.1 Betriebsart OBD2 Software

Im Zusammenspiel mit dem Pi-OBD Modul muss HHGui wie folgt gestartet werden.

```
./hhgui /dev/ttyAMA0 115200 init=AGV
```

Das Device kann abweichend auch `/dev/ttyS0` oder `/dev/serial0` sein. Die nötigen Informationen zum Einrichten der seriellen Schnittstelle in Abhängigkeit von der Pi Version sind der Anleitung des Pi-OBD Moduls zu entnehmen.

Mit weiteren Kommandozeilenoptionen kann das Aussehen von HHGui konfiguriert werden (siehe dazu auch die Ausgabe von `./hhgui help`).

Die Option `nocursor` schaltet die Anzeige des Mauszeigers ab.

Die Option `maximize` bringt das HHGui Fenster mit Titelzeile in der Breite auf Displaybreite. Die Fensterhöhe wird so verändert, dass das Seitenverhältnis beibehalten wird.

Die Option `fullscreen` bringt das HHGui Fenster ohne Titelzeile auf Displaygröße. Falls HHGui ohne Hintergrundbild verwendet wird, bleibt dabei das Seitenverhältnis erhalten und die Darstellung von HHGui ist vertikal zentriert. Mit Hintergrundbild wird das Seitenverhältnis verändert und die Anzeige erscheint beim 7" Touchscreen leicht gestreckt.

Da die Option `fullscreen` keine Titelzeile und somit keinen Close-Button anzeigt, wird generell unabhängig von der Option ein 30 x 30 Pixel großer Bereich oben rechts im HHGui Fenster reserviert, der als unsichtbarer Button fungiert. Über diesen Button kann das HHGui Fenster von Fullscreen auf eine kleinere Fenstergröße mit Titel und Close-Button umgeschaltet werden und umgekehrt.

Es empfiehlt sich die Aufnahme der gewünschten Kommandozeile in ein Initialisierungsskript für das Pi-OBD Modul. Dafür kann der `nano` Editor verwendet werden.

```
nano ./startobd.sh
```

Im Editor folgende Zeilen eingeben und abspeichern:

```
#!/bin/sh
sudo gpio export 18 out
sudo gpio export 23 out
sudo gpio -g write 18 0
sudo gpio -g write 23 0
sleep 1
sudo gpio -g write 18 1
sleep 1
cd <Installationspfad von hhgui> (z.B. cd /home/pi/Desktop)
./hhgui /dev/ttyAMA0 115200 init=AGV
```

Die Datei muss noch ausführbar gemacht werden:

```
sudo chmod +x ./startobd.sh
```

3.2 Bedienung der OBD2 Software

Die Bedienung des gestarteten Programms erfolgt bevorzugt über einen Touchscreen. Die Bedienung mit der Maus oder der Tastatur ist jedoch ebenfalls möglich.

Grundsätzlich gibt es 4 mögliche Tastenfunktionen, die teilweise mit Kurzdruck oder Langdruck variiert werden können:

Up – Kurzdruck – bewegt den Listenauswahl-Cursor nach oben oder
inkrementiert ein Zahlenfeld +1

Up – Langdruck – bewegt den Listenauswahl-Cursor eine Menü-/Listenseite nach oben oder
inkrementiert ein Zahlenfeld +10

Down – Kurzdruck – bewegt den Listenauswahl-Cursor nach unten oder
dekrementiert ein Zahlenfeld -1

Down – Langdruck – bewegt den Listenauswahl-Cursor eine Menü-/Listenseite nach unten oder
dekrementiert ein Zahlenfeld -10

OK – führt den mit dem Listenauswahl-Cursor angewählten Menüpunkt aus oder
springt zur nächsten Menüseite innerhalb einer Menürolle oder
springt aus der tiefsten Menüebene eine Menüebene höher (*) oder
bestätigt eine Aktion

ESC – Kurzdruck – springt aus der aktuellen Menüebene eine Menüebene höher oder
springt zur vorhergehenden Menüseite innerhalb einer Menürolle oder
bricht eine Aktion ab

ESC – Langdruck – Reset der Software und Rücksprung ins Hauptmenü

(*) Diese Sonderfunktion ermöglicht zum Beispiel das schnelle Wechseln zwischen Listenanzeige und Einzelanzeige und umgekehrt bei den im Kapitel 4 beschriebenen Menüs mit nur einer Taste.

HHGui Bedienung mit Hintergrundbild

Wenn HHGui beim Programmstart das Hintergrundbild findet (siehe Kapitel 2), erfolgt die Bedienung über die im Hintergrundbild dargestellten 4 Tasterflächen.

Der Taster links oben fungiert als Up-Taste.

Der Taster links unten fungiert als Down-Taste.

Der Taster rechts oben fungiert als ESC-Taste.

Der Taster rechts unten fungiert als OK-Taste.

HHGui Bedienungsanleitung

HHGui Bedienung ohne Hintergrundbild

Wenn HHGui beim Programmstart das Hintergrundbild nicht findet (siehe Kapitel 2), erfolgt die Bedienung über unsichtbare Tasterflächen. Um auch kleine Displays mit 320 x 240 Pixeln am Pi zu unterstützen, ist die Touchscreen Eingabe wie folgt realisiert. Das Programmfenster ist in 4 gleich große unsichtbare Bereiche geteilt, wobei der Bereich rechts oben für die Sonderfunktion Fullscreen Umschaltung noch einen kleinen fünften Bereich abgetrennt hat.

Der Bereich links oben fungiert als Up-Taste.

Der Bereich links unten fungiert als Down-Taste.

Der Bereich rechts oben fungiert als ESC-Taste.

Der Bereich rechts unten fungiert als OK-Taste.

Ein 30 x 30 Pixel großer Bereich ganz rechts oben übernimmt eine Umschaltfunktion zwischen Fullscreen Fenster ohne Titelzeile und ohne Close Button und Fenster mit Titelzeile und mit Close Button. So kann das Programm auch bei Touchscreen Bedienung und *fullscreen* Kommandozeilenoption (siehe Kapitel 3.1) beendet werden.

HHGui Bedienung mit Maus

Die Bedienung mit der Maus entspricht der Touchscreen Bedienung.

HHGui Bedienung mit Tastatur

Für die Bedienung mit der Tastatur können folgende Tasten verwendet werden:

Up – Cursor hoch, k-Taste

Down – Cursor runter, j-Taste

ESC – Cursor links, ESC-Taste, h-Taste

OK – Cursor rechts, Return-Taste, l-Taste

Die Cursor-Tasten auf dem Ziffernblock (4, 6, 8, 2) funktionieren ebenfalls als Cursor-Tasten.

3.3 Betriebsart OBD2 Simulator

Wenn das Programm HHGui ohne die Angabe einer seriellen Schnittstelle gestartet wird, arbeitet das Programm als OBD2 Simulator. Dafür sind keinerlei Konfigurationseinstellungen auf dem Raspberry Pi nötig, da nicht auf die serielle Schnittstelle zugegriffen wird.

```
./hhgui
```

Im Simulatorbetrieb können alle Menüs und OBD2 Funktionen getestet werden. Das ist mehr als in einem Fahrzeug möglich wäre, da diese schon aufgrund der Unterschiede zwischen Dieselmotor und Ottomotor nur eine Untermenge aller Funktionen anbieten können.

Neben den im Kapitel 3.1 beschriebenen Kommandozeilenoptionen *nocursor*, *maximize* und *fullscreen* ist die Angabe einer weiteren Option *dtc=Xxxxx* möglich, die am Ende dieses Kapitels erklärt wird.

HHGui Bedienungsanleitung

Die Bedienung erfolgt wie unter Kapitel 3.2 beschrieben. Allerdings wird für die Eingabe von Kommandos eine Tastatur (eventuell genügt eine virtuelle Tastatur) benötigt. Ein Kommando besteht entweder aus einer maximal 2-stelligen Zahl, die in einem Zahlenspeicher gespeichert wird, gefolgt von einem Buchstaben, oder nur einem Buchstaben. Falls im Zahlenspeicher bereits die gewünschte Zahl für das nächste Kommando steht, genügt die Eingabe des Buchstabens.

Folgende Kommandos stehen zur Verfügung:

a – setzt die Zahl der permanenten DTCs (Diagnostic Trouble Codes (Fehlercodes)) für das aktive Steuergerät (nur für CAN Protokolle)

Zahlenspeicher 0 - 16

b – setzt die Zahl der Zylinderbänke für PID (Parameter Identifier (Unterfunktion)) 0x13 (Verbauorte der Lambdasonden) für alle Steuergeräte

Zahlenspeicher 1 – 2 (*)

c – setzt die Zahl der Zylinderbänke für PID 0x1D (Verbauorte der Lambdasonden) für alle Steuergeräte

Zahlenspeicher 1 – 4 (*)

(*) Der Standard ISO 15031-5 schreibt vor, dass alle Steuergeräte eines Fahrzeugs, die die Unterfunktion Verbauorte der Lambdasonden unterstützen, entweder die PID 0x13 oder die PID 0x1D unterstützen müssen, aber auf keinen Fall beide unterstützen dürfen. Der Simulator löscht deswegen bei einem Kommando für PID 0x13 automatisch die PID 0x1D aus der Liste der unterstützten PIDs und markiert die PID 0x13 als unterstützt für alle Steuergeräte und umgekehrt.

d – setzt die Zahl der bestätigten DTCs für das aktive Steuergerät

Zahlenspeicher 0 - 16

e – setzt die Zahl der antwortenden OBD2 Steuergeräte

Zahlenspeicher 1 – 8

Die Eingabe dieses Kommandos muss vor der Protokollsuche erfolgen. D.h., vor der Auswahl des Menüpunkts "Main Menu → Start Diagnostics → Select Protocol" oder vor der Auswahl des Menüpunkts "Diagnostics Menu → Reload OBD Data"

m – schaltet die Motorkontrollleuchte (MIL) an und setzt einen bestätigten Fehler für das aktive Steuergerät, falls für dieses keine bestätigten Fehler vorhanden sind

n – schaltet die Anzeige des Zahlenspeichers an/aus

p – setzt die Zahl der schwebenden Fehler für das aktive Steuergerät

Zahlenspeicher 0 - 16

t – setzt das Protokoll, mit dem das oder die Steuergeräte antworten

Zahlenspeicher 0 – 9

HHGui Bedienungsanleitung

0 = kein Protokoll gesetzt (jedes bei der Protokollsuche ausgewählte Protokoll funktioniert, bei automatischer Protokollsuche antworten das oder die Steuergeräte mit PWM),

1 = PWM, 2 = VPWM, 3 = ISO 9141-2, 4 = KWP2000 5 Baud Init, 5 = KWP2000 Fast Init,
6 = CAN 11/500, 7 = CAN 29/500, 8 = CAN 11/250, 9 = CAN 29/250

Die Eingabe dieses Kommandos muss vor der Protokollsuche erfolgen. D.h., vor der Auswahl des Menüpunkts "Main Menu → Start Diagnostics → Select Protocol"

v – schaltet variable OBD Daten ein/aus. Variable Daten werden unter den Menüpunkten "Diagnostics Menu → Current Data" oder "Diagnostics Menu → Vehicle/ECU Info → Performance Tracking 08/0B" angezeigt.

x – setzt einen negativen Antwort Code für die OBD2 Dienste Fehler löschen (Diagnostics Menu → Trouble Codes → Clear DTC Data → YES) und CVN Abfrage (Diagnostics Menu → Vehicle/ECU Info → Calibration Verification).
Zahlenspeicher: erlaubte Codes sind 00, 10, 11, 12, 21, 22, 78
00 oder letzten Code erneut eingeben (= Buchstabe x ohne Änderung des Zahlenspeichers eingeben) schaltet die negative Antwort aus.

y – setzt NO DATA Antwort für die nächste Antwort vom simulierten Pi-OBd Modul

z – setzt CAN ERROR Antwort für die nächste Antwort vom simulierten Pi-OBd Modul (nur für CAN Protokolle)

q – beendet HHGui

Beispiele:

Die Eingabe von 3d setzt 3 bestätigte Fehlercodes für das aktuell ausgewählte Steuergerät.

Die Eingabe von 1e setzt die Zahl der antwortenden Steuergeräte auf 1.

Die meisten Kommandos erfordern die Auswahl des Menüpunkts "Diagnostics Menu → Reload OBD Data", damit die Auswirkungen des Kommandos sichtbar werden und HHGui einen definierten Zustand annimmt.

Beim Programmstart nimmt der OBD2 Simulator folgenden Ausgangszustand an:

Protokollsuche mit Auto: Antwort erfolgt mit PWM

Protokollsuche mit Protokoll x: Antwort erfolgt mit Protokoll x

Anzahl der antwortenden Steuergeräte: 2

Anzahl der bestätigten Fehler: 1 pro antwortendem Steuergerät

Anzahl der schwebenden Fehler: 1 pro antwortendem Steuergerät

HHGui Bedienungsanleitung

Anzahl der permanenten Fehler (nur CAN Protokolle): 1 pro antwortendem Steuergerät
MIL aktiviert

Die Anzahl der bestätigten Fehler und der/die Fehlercode/s können beim OBD2 Simulator über eine weitere Kommandozeilenoption *dtc=Xxxxx* beim Programmstart konfiguriert werden. Durch die mehrfache Angabe dieser Option können bis zu 16 bestätigte Fehler für das erste simulierte Steuergerät konfiguriert werden.

Beispiel:

```
./hhgui dtc=P014A dtc=P0235
```

Die Eingabe dieser Befehlszeile verändert den Ausgangszustand des OBD2 Simulators so, dass das erste simulierte Steuergerät die bestätigten Fehler P014A und P0235 liefert. Diese können dann im DTC Summary Menü 4.5.1 mit zugehörigem Fehlertext angezeigt werden, sofern der Text in HHGui vorhanden ist. Im Kapitel 5 wird beschrieben, zu welchen Fehlercodes Fehlertexte angezeigt werden können.

4 Menüführung

Beim Programmstart wird kurz die Versionsnummer der von HHGui intern verwendeten OBD2-Analyser NG Firmware und die Up/Down/ESC/OK-Tastenzuordnung angezeigt. Dann erfolgt automatisch ein Sprung ins Hauptmenü.

4.1 Main Menu (Hauptmenü)

Das Hauptmenü zeigt die folgenden Menüpunkte:

- Start Diagnostics – Starte OBD2 Diagnose → 4.2
- System Information – Pi-OBD Modul Information → 4.3
- System Settings – System Einstellungen → 4.4

4.2 Start Diagnostics (Starte Diagnose)

- Select Protocol – Wähle Protokoll
- Automatic Protocol Scan – Automatische Protokollsuche (bei unbekanntem Protokoll)
- CAN 11bit Id, 500 kpbs
- CAN 29bit Id, 500 kpbs
- CAN 11bit Id, 250 kpbs
- CAN 29bit Id, 250 kpbs
- ISO 9141-2
- KWP2000 5Baud init
- KWP2000 fast init
- SAE J1850 VPWM
- SAE J1850 PWM

Für Fahrzeuge ab Modelljahr 2008 sind nur noch CAN Protokolle erlaubt (Ausnahmen sind Fahrzeuge/Motoren mit Facelift), deswegen sind diese in der Auswahlliste weiter oben angeordnet.

Die Reihenfolge der Menüpunkte ist dynamisch, wenn im System Settings Menü die Option "Save last Protocol" ausgewählt ist. Das vom Anwender jeweils zuletzt ausgewählte Protokoll wird dann beim nächsten Aufruf des Menüs in der Menüliste ganz oben angezeigt.

Nach der Auswahl eines Protokolls wechselt die Anzeige in die Connect Anzeige (Verbinde mit Fahrzeug). Bei Erfolg der Protokollsuche wechselt die Connect Anzeige in die Reading Anzeige (Lese Daten/Konfiguration). Dort werden die vom Fahrzeug unterstützten OBD2 Dienste und Unterfunktionen abgefragt, siehe Kapitel 5. Danach wechselt die Anzeige in das Diagnostics Menu (Diagnose Menü), siehe Kapitel 4.5.

Falls die Connect Anzeige in die Fehlermeldung "No protocol found" (kein Protokoll gefunden) wechselt, ist entweder die vom Fahrzeug für die Diagnose nötige Zündschlüsselposition (Zündung ein mit oder ohne Motorlauf) nicht korrekt, oder das gewählte Protokoll ist nicht das tatsächlich

vom Fahrzeug unterstützte Protokoll. In seltenen Fällen gab es einen Verbindungsfehler und die Protokollsuche kann einfach wiederholt werden.

4.3 System Information

Das System Information Menü zeigt die folgenden 3 Informationen an:

Seriennummer vom Pi-OBd Modul/AGV Interface oder Erstelldatum HHGui (OBd2 Simulator)

Firmware Version vom Pi-OBd Modul/AGV Interface oder Version HHGui (OBd2 Simulator)

aktuell gemessene Spannung vom Pi-OBd Modul/AGV Interface (wird jede Sekunde aktualisiert)

4.4 System Settings (System Einstellungen)

Orientation 180° – dreht LCD Anzeige und Tastenfunktion um 180° (ein eventuell sichtbarer Zahlenspeicher des OBd2 Simulators wird nicht gedreht)

Save last Protocol – speichert das zuletzt bei der Protokollsuche ausgewählte Protokoll (auch Automatic Protocol Scan), speichert also nicht das gefundene Protokoll!

LCD Contrast → Kontrasteinstellung 0 .. 100%

LCD Backlight Red → Farbeinstellung Rotanteil 0 .. 100%

LCD Backlight Green → Farbeinstellung Grünanteil 0 .. 100%

LCD Backlight Blue → Farbeinstellung Blauanteil 0 .. 100%

Einstellungen bleiben über einen Software Reset erhalten, werden aber bisher nicht auf dem Raspberry Pi abgespeichert (im echten OBd2-Analyser NG sind das Werte, die in einem EEPROM gespeichert werden).

4.5 Diagnostics Menu (Diagnose Menü)

Current Data – OBd2 Service 0x01 Momentanwerte → Listenanzeige → Einzelanzeige

Trouble Codes – Fehlercodes → 4.5.1

I/M Readiness – TÜV Prüfbereitschaft → 4.5.2 (Menüpunkt ist nur vorhanden, wenn das aktive Steuergerät die PID 0x01 unterstützt)

Vehicle/ECU Info – Fahrzeug/Steuergeräte Information → 4.5.3 (Menüpunkt ist nur vorhanden, wenn das aktive Steuergerät mindestens eine der vom Menü benötigten Unterfunktionen unterstützt)

Protocol Information – Protokoll Information → 4.5.4

Change ECU – Wechsle Steuergerät → springt in ein Menü zum Wechseln des aktiven Steuergeräts (Menüpunkt ist nur vorhanden, wenn mehr als ein Steuergerät geantwortet hat). Beachte den Hinweis am Ende des Kapitels 6!

Reload OBd Data – liest alle OBd2 Daten neu ein

HHGui Bedienungsanleitung

Im Diagnose Menü wird in der Titelzeile oben rechts das aktuelle Protokoll angezeigt. Die Nummer vor dem Protokoll entspricht der vom Pi-OBd Modul benutzten Protokollnummer.

1 – PWM

2 – VPWM

3 – ISO 9141-2

4 – KWP2000 5Baud init

5 – KWP2000 Fast init

6 – CAN 11/500

7 – CAN 29/500

8 – CAN 11/250

9 – CAN 29/250

Im abgetrennten Bereich rechts neben der Diagnose Menüliste oder Trouble Codes Menüliste werden folgende Daten angezeigt:

"---" oder MIL Fehlersymbol (Motorkontrollleuchte)

DTC: x, x = Anzahl der bestätigten Fehler aller Steuergeräte + Anzahl der schwebenden Fehler aller Steuergeräte + Anzahl der permanenten Fehler aller Steuergeräte

E x/y:zz = Steuergerät x von y Steuergeräten aktiv, zz = Steuergeräteadresse (hexadezimal)

Die im Diagnose Menü und Trouble Codes Menü angezeigten Daten MIL und Anzahl der Fehler und die im Trouble Codes → DTC Summary Menü angezeigten Fehlercodes sind statische Daten, die vor dem ersten Einsprung ins Diagnose Menü gelesen wurden. Um neue aktuelle Daten einzulesen, muss bei Bedarf der Menüpunkt "Reload OBD Data" ausgeführt werden.

4.5.1 Trouble Codes Menu (Fehlercodes Menü)

DTC Summary – Übersicht der Fehler aller Steuergeräte → Listenanzeige → Einzelanzeige

Die Zusätze hinter einem Fehler haben folgende Bedeutung:

confirmed – bestätigter Fehler

pending – schwebender Fehler

permanent – permanenter Fehler (nur CAN Protokolle) (*)

Clear DTC Data – bestätigte und schwebende Fehler und weitere Daten löschen → 4.5.1.1

Freeze Frames – Fehlerumgebungsdaten → 4.5.1.2

(*) Permanente Fehler werden vom Steuergerät selbst zurückgenommen, wenn der Fehler nach dem ein- oder mehrmaligem Durchlaufen des entsprechenden Prüfprogramms nicht mehr erkannt wird. Permanente Fehler können deshalb nicht durch das Abklemmen der Batterie oder mit Hilfe von OBD2 Software oder OBD2 Testern gelöscht werden.

4.5.1.1 Sicherheitsabfrage Fehler Löschen

Bevor bestätigte und schwebende Fehler, MIL, OBD2 Monitordaten (Readiness Menü), Daten der PIDs 0x21, 0x30, 0x31, 0x4D und 0x4E, sowie weitere Daten gelöscht werden, muss die Sicherheitsabfrage mit YES beantwortet werden. Die Auswahl von NO bricht das Fehler Löschen ab.

Wenn die Sicherheitsabfrage mit YES bestätigt wurde, können mehrere Fälle auftreten:

1. Alle Steuergeräte antworten, dass das Fehler Löschen ausgeführt wurde
2. Nur ein Teil der Steuergeräte antwortet, dass das Fehler Löschen ausgeführt wurde
3. Alle Steuergeräte antworten, dass das Fehler Löschen nicht ausgeführt wurde
4. Kein Steuergerät sendet eine Antwort

Im Fall 1 wechselt die Anzeige in die Anzeige "Diagnostic Data cleared ..." (Diagnose Daten gelöscht), dann werden die OBD2 Daten neu eingelesen und es erfolgt ein Rücksprung ins Diagnose Menü.

In den Fällen 2-4 wechselt die Anzeige in eine Übersicht "Check ECU Error Responses" (Prüfe Steuergeräte Fehlermeldungen). Dort werden für maximal 7 Steuergeräte die Antworten (Fehlertyp oder "no response" (keine Antwort) angezeigt. Als Fehlertyp darf hier nur der Fehler 0x22 CNCORSE (Conditions not correct or request sequence error (Bedingungen nicht erfüllt oder korrekte Reihenfolge der Requests nicht eingehalten)) auftreten. Dieser Fehler wird nur von CAN oder KWP2000 Protokollen unterstützt. Dieser Fehler kann auftreten, wenn das Fehler Löschen bei laufendem Motor ausgeführt wurde. Viele Steuergeräte lassen das Fehler Löschen nur bei Zündung ein ohne Motorlauf zu. Falls ein Steuergerät nicht antwortet, hat es aktuell keine OBD2 Daten, die zu löschen sind.

Im Fall 2 wechselt die Anzeige nach der Bestätigung der "Check ECU Error Responses" Übersicht in die Anzeige "Erase incomplete ..." (Löschen unvollständig), dann werden die OBD2 Daten neu eingelesen und es erfolgt ein Rücksprung ins Diagnose Menü.

Im Fall 3 und 4 wechselt die Anzeige nach der Bestätigung der "Check ECU Error Responses" Übersicht in das Trouble Codes Menü und die OBD2 Daten werden nicht neu eingelesen.

4.5.1.2 Fehlerumgebungsdaten

Falls hier "No freeze frames stored" angezeigt wird, sind keine Fehlerumgebungsdaten im aktiven Steuergerät abgespeichert worden. Ansonsten wird hier das "Select Freeze Frame" (Wähle Fehlerdaten Datensatz) Menü angezeigt. Für OBD2 spezifiziert war ursprünglich nur der Datensatz 0, der im Auswahlfeld voreingestellt ist. Herstellerabhängig können aber auch mehrere Datensätze für einen Fehler abgespeichert sein. Falls diese vom Aufbau nicht dem standardisierten Datensatz 0 entsprechen, können sie nicht korrekt angezeigt werden.

Rechts neben dem Auswahlfeld steht die Fehlernummer von dem Fehler, der das Speichern der Fehlerumgebungsdaten ausgelöst hat. Wenn dort stattdessen "No data" (keine Daten) steht, ist der aktuell im Auswahlfeld ausgewählte Datensatz nicht vorhanden.

Nach der Bestätigung eines existierenden Datensatzes wechselt die Anzeige in die Freeze Frame Listenanzeige. Dort werden die zum Fehler gehörenden statischen Daten, die beim Auftreten des Fehlers abgespeichert wurden, angezeigt. Einzelwerte können in der Einzelanzeige mit Beschreibung angezeigt werden.

4.5.2 I/M Readiness (TÜV Prüfbereitschaft)

I/M = Inspection/Maintenance (Inspektion/Wartung)

Supported Monitors – unterstützte OBD2 Monitore → Listenanzeige → Einzelanzeige

Monitor Status – OBD2 Monitor Status → Listenanzeige → Einzelanzeige
Falls hinter dem Menüpunkt (DTC) oder (Cycle) steht, ist ein im Monitor Settings Menü beschriebener Anzeigefilter aktiv

Monitor Settings – Einstellungen für die Anzeige der I/M Readiness Untermenüs → 4.5.2.1

Ein OBD2 Monitor ist ein Prüfprogramm, das entweder mindestens einmal pro Fahrzyklus oder kontinuierlich ausgeführt wird.

Ein Fahrzeug ist aus OBD2 Monitorsicht für die TÜV Prüfung bereit, wenn im Monitor Status Menü für alle Monitore der Status "xxx_RDY: YES" angezeigt wird (für nicht unterstützte Monitore wird immer YES angezeigt). Dann sind alle Prüfprogramme seit dem letzten Fehler Löschen mindestens einmal durchlaufen worden. Das ist gleichbedeutend mit dem sogenannten Readiness-Code = 0. Der Readiness-Code ist eine 12bit Binärzahl, bei der jedes nicht gesetzte Bit für einen durchlaufenen oder für einen nicht unterstützten Monitor steht.

Im I/M Readiness Menü und seinen Untermenüs wird zyklisch jede Sekunde die PID 0x01 (MIL Status, Anzahl bestätigter Fehler, Monitor Status seit letztem Fehler Löschen) für das aktive Steuergerät eingelesen. Falls die PID 0x41 unterstützt wird, geschieht das im Wechsel mit der PID 0x41 (Monitor Status des aktuellen Fahrzyklus). D.h., im I/M Readiness Menü und seinen Untermenüs werden aktuelle Daten angezeigt. Falls sich in diesen Menüs der Fehlerzustand ändert, kann nach dem Rücksprung ins Diagnose Menü und Trouble Codes Menü und seinen Untermenüs ein inkonsistenter Fehlerzustand angezeigt werden. Um neue aktuelle Daten einzulesen, muss bei Bedarf der Menüpunkt "Reload OBD Data" ausgeführt werden.

4.5.2.1 Monitor Settings (Monitor Anzeige Einstellungen)

Show Status [all/since DTCs cleared/this Cycle] – setzt Anzeigefilter für das Monitor Status Menü

- all – kein Anzeigefilter aktiv
- since DTCs cleared – zeigt nur Status seit letztem Fehler Löschen
- this Cycle – zeigt nur Status des aktuellen Fahrzyklus

Dieser Menüpunkt ist nur verfügbar, wenn das Steuergerät die PID 0x41 (Monitor Status des aktuellen Fahrzyklus) unterstützt.

Show unsupported Monitors – zeigt auch nicht vom Fahrzeug unterstützte Monitore in den Menüs Supported Monitors und Monitor Status an

Show A/C System Monitor – zeigt veralteten Klimaanlage Monitor in den Menüs Supported Monitors und Monitor Status an. Das ist ein Prüfprogramm für Lecks in der Klimaanlage für alte Fahrzeuge mit dem inzwischen verbotenen R-12 Kühlmittel.

4.5.3 Vehicle/ECU Info (Fahrzeug/Steuergeräte Information)

VIN	– Vehicle Identification Number (Fahrzeugidentifikationsnummer)
Calibration Identifications	– Kalibrierungsnummern (z.B. Software Stand, Datensatz Stand)
Calibration Verification	– Verifikation für Software Kalibrierungsnummern (Checksummen)
Performance Tracking 08	– Leistungskontrolle, Statistik einzelner Teile/Systeme der Abgasregelung für Dieselfahrzeuge vor dem Modelljahr 2010 und für Benziner
ECU Name	– Steuergerätename
Performance Tracking 0B	– Leistungskontrolle, Statistik einzelner Teile/Systeme der Abgasregelung für Dieselfahrzeuge ab Modelljahr 2010
O2 Sensor Locations	– Anzahl und Verbauorte der Lambdasonden (der Menüpunkt ist verfügbar, wenn das aktive Steuergerät die PID 0x13 oder 0x1D unterstützt)

Das Vehicle/ECU Info Menü ist nur verfügbar, wenn das aktive Steuergerät mindestens eine Unterfunktion unterstützt, die für seine Menüpunkte benötigt wird.

Zwischen den Menüpunkten Calibration Identifications und Calibration Verification kann mit den Up/Down Tasten schnell gewechselt werden, da diese Daten in der Regel zusammengehörende Paare bilden.

4.5.4 Protocol Information

Das Protocol Information Menü besteht aus 4 Seiten. Auf Seite 1 wird das aktuelle Protokoll angezeigt. Der Text wird vom Pi-OBd Modul geliefert. Bei ISO 9141-2 und KWP2000 Protokollen werden zusätzliche Daten für Keyword Protokolle aus dem Protokoll-Header angezeigt.

Korrekt müsste das Protocol Information Menü wie das Vehicle/ECU Info Menü als Protocol/ECU Info Menü bezeichnet werden, da die nächsten 3 Seiten Steuergeräte abhängige Daten, die vom Steuergerät unterstützen OBD2 Unterfunktionen, enthalten.

Seite 2 zeigt die vom aktuellen Steuergerät unterstützten PIDs (Unterfunktionen) für OBD2 Service 0x01 (Momentanwerte) an. Jedes gesetzte Bit in den dargestellten Hexadezimalwerten steht für eine PID. Gestartet wird mit PID 0x01 beim höchstwertigsten Bit (0x80000000).

Seite 3 zeigt die vom aktuellen Steuergerät unterstützten PIDs für OBD2 Service 0x02 (Fehlerumgebungsdaten) an. Beachte: Es muss zu Beginn und nach jedem Steuergerätewechsel (Change ECU Menü) das Freeze Frame Menü einmalig geöffnet werden, damit die tatsächlichen Werte ungleich 0 angezeigt werden. Bei Steuergeräten, die vom aktuellen Fehler abhängige Fehlerumgebungsdaten unterstützen, muss das Freeze Frame Menü bei jedem Fehler, der neue Fehlerumgebungsdaten abspeichert, einmalig geöffnet werden.

Seite 4 zeigt die vom aktuellen Steuergerät unterstützten InfoTypes (Unterfunktionen) für OBD2 Service 0x09 (Vehicle/ECU Info) an.

5 Unterstützte OBD2 Dienste und Unterfunktionen

Die Pi-OBDD Handheld GUI Software HHGui unterstützt die folgenden OBD2 Dienste:

- Service 0x01 Current Data – aktuelle Daten (Momentanwerte)
- Service 0x01 Inspection/Maintenance Readiness Monitors – Monitor Daten
- Service 0x01 Location of Oxygen Sensors – Anzahl und Verbauorte der Lambdasonden
- Service 0x02 Freeze Frame Data – Fehlerumgebungsdaten
- Service 0x03 Read Confirmed DTCs – Lese bestätigte Fehlercodes
- Service 0x04 Clear/Erase Fault Data – Löschen der Fehler- und Monitor Daten
- Service 0x07 Read Pending DTCs – Lese schwebende Fehlercodes
- Service 0x09 Read Vehicle Information (InfoTypes) – Fahrzeug-/Steuergeräteinformationen, Statistik
- Service 0x0A Read Permanent DTCs – Lese permanente Fehlercodes (nur CAN Protokolle)

Für die Services 0x01 und 0x02 wird die Abfrage folgender Unterfunktionen (Parameter Identifier = PID) unterstützt:

0x00 – 0x60, 0x70, 0x80, 0x8D, 0xA0, 0xC0, 0xE0

Für den Service 0x09 wird die Abfrage folgender Unterfunktionen (InfoTypes) unterstützt:

0x00, 0x02, 0x04, 0x06, 0x08, 0x0A, 0x0B

Die tatsächlich unterstützten Dienste und Unterfunktionen sind abhängig vom jeweiligen Fahrzeug und seinen OBD2 Steuergeräten (maximal 8). In den meisten Fällen antwortet nur das Motorsteuergerät, bei Fahrzeugen mit Automatikgetriebe oft auch das Getriebesteuergerät.

Eine genaue Beschreibung der Dienste und Unterfunktionen findet sich im Standard ISO 15031-5 oder dem inhaltlich identischen Standard SAE J1979. Beide Dokumente sind nur in englischer Sprache erhältlich. In den folgenden 2 Unterkapiteln findet sich eine einfache Auflistung.

Für die Services 0x03, 0x07 und 0x0A werden zu Fehlercodes, die im Standard SAE J2012:2007 definiert sind, Fehlertexte angezeigt. Das sind 4277 Fehlercodes mit zugehörigen Fehlertexten.

5.1 Auflistung der Unterfunktionen (PIDs) für Services 0x01 und 0x02

Für OBD2 Service 0x01 erfolgt die Anzeige der meisten Unterfunktionen im Current Data Menü. Für OBD2 Service 0x02 erfolgt die Anzeige der meisten Unterfunktionen im Freeze Frame Menü. Falls die Anzeige in anderen Menüs erfolgt, ist das Menü bei der Unterfunktion angegeben.

0x00, 0x20, 0x40 ... 0xC0, 0xE0 – unterstützte PIDs → Protocol Info Menü 4.5.4 (beachte dort den Hinweis zur den unterstützten PIDs für Service 0x02!)

0x01 – MIL Status, Anzahl bestätigter Fehler und Monitor Status seit letztem Fehler Löschen
→ I/M Readiness Menü 4.5.2

0x02 – Fehler, der zum Abspeichern der Fehlerumgebungsdaten geführt hat
→ Freeze Frame Menü 4.5.1.2

HHGui Bedienungsanleitung

0x03 – Status der Kraftstoffregelung, mögliche Anzeigewerte sind:

OL (Open Loop) – Regelkreis offen,

Bedingungen für einen geschlossenen Regelkreis noch nicht erfüllt

CL (Closed Loop) – Regelkreis geschlossen,

Daten der Lambdasonden werden zur Berechnung der Einspritzmenge verwendet

OL-Drive – Regelkreis offen wegen aktueller Fahrsituation (Vollgas, Schubbetrieb)

OL-Fault – Regelkreis offen wegen Systemfehler

CL-Fault – Regelkreis geschlossen, aber mindestens eine Lambdasonde liefert falsche Daten

0x04 – berechneter Motorlastwert

0x05 – Kühlwassertemperatur (selten auch Öltemperatur, falls PID 0x5C nicht unterstützt wird)

0x06, 0x08, 0x55, 0x57 – Kurzzeit-Einspritzkorrektur für jede Zylinderbank

0x07, 0x09, 0x56, 0x58 – Langzeit-Einspritzkorrektur für jede Zylinderbank

0x0A, 0x22, 0x23, 0x59 – Kraftstoffdruck

0x0B – Absolutdruck im Saugrohr

0x0C – Motordrehzahl

0x0D – Fahrzeuggeschwindigkeit (Maximum ist 255 km/h)

0x0E – Zündzeitpunkt (Zündwinkel)

0x0F – Lufteinlassstemperatur

0x10 – Luftdurchsatz

0x11, 0x45, 0x47, 0x48, 0x8D – Drosselklappenstellung (diverse Sensoren)

0x12 – Sollzustand Sekundärluftzufuhr, mögliche Anzeigewerte sind:

UPS (upstream), DNS (downstream), OFF, DIAG

UPS – Sekundärluft wird (luft-)stromaufwärts vor dem ersten Katalysator eingeblasen

DNS – Sekundärluft wird (luft-)stromabwärts nach dem ersten Katalysator Einlass eingeblasen

OFF – Sekundärluft aus (Sekundärluftpumpe aus, Sekundärluftventil geschlossen)

DIAG – System befindet sich im Diagnosemodus und wird extern angesteuert

0x13, 0x1D – Verbauorte und Anzahl der Lambdasonden → Vehicle/ECU Info Menü 4.5.3

0x14-0x1A, 0x24-0x2B, 0x34-0x3B – Daten der Lambdasonden (Spannung oder Strom,

Kurzzeit-Einspritzkorrektur oder Lambda) für jede Zylinderbank und jede Lambdasonde

0x1C, 0x5F – OBD/Emissionsvorschriften, die dieser Motor/dieses Fahrzeug einhält

0x1E – Status des Nebenabtriebs (Nebenantrieb, PTO (Power Take Off))

0x1F – Motorlaufzeit seit Motorstart

0x21 – mit aktivierter Motorkontrollleuchte zurückgelegte Wegstrecke

HHGui Bedienungsanleitung

- 0x2C – Sollwert der Abgasrückführung (EGR = Exhaust Gas Recirculation)
 - 0x2D – Abweichung vom Sollwert der Abgasrückführung
 - 0x2E – Sollwert der Tankentlüftung (Ventil des Tankentlüftungssystems)
 - 0x2F – Füllstand des Kraftstofftanks
 - 0x30 – Anzahl der Fahrzyklen seit letztem Fehlerlöschen
 - 0x31 – zurückgelegte Wegstrecke seit letztem Fehlerlöschen
 - 0x32, 0x53, 0x54 – Dampfdruck im Tankentlüftungssystem
 - 0x33 – Umgebungsluftdruck
 - 0x3C-0x3F Katalysatortemperatur
 - 0x41 – Monitor Status des aktuellen Fahrzyklus → I/M Readiness Menü 4.5.2
 - 0x42 – Versorgungsspannung des Steuergeräts
 - 0x43 – absoluter Motorlastwert
 - 0x44 – Sollwert Kraftstoff/Luft Verhältnis (Lambda)
 - 0x46 – Umgebungstemperatur
 - 0x49, 0x4A, 0x4B, 0x5A – Gaspedalstellung (diverse Sensoren)
 - 0x4C – Sollwert Drosselklappenstellung
 - 0x4D – Motorlaufzeit mit aktivierter Motorkontrollleuchte
 - 0x4E – Motorlaufzeit seit letztem Fehlerlöschen
 - 0x4F – Tester Konfiguration 1 (kann Schrittweite und Maximalwerte der PIDs 0x0B, 0x24-0x2B, 0x34-0x3B und 0x44 verändern) (*)
 - 0x50 – Tester Konfiguration 2 (kann Schrittweite und Maximalwert der PID 0x10 verändern) (*)
- (*) Mögliche Werte ungleich 0 werden von HHGui automatisch für die Anzeige der betroffenen PIDs verwendet. Die Werte selbst werden bisher gemäß Standard ISO 15031-5 nicht angezeigt.
- 0x51 – aktuell verwendeter Kraftstoff, Beispiele:
 - GAS – Fahrzeug verwendet Benzin (GAS bedeutet Gasoline = Benzin)
 - DSL – Fahrzeug verwendet Diesel
 - BI_LPG – Bi-Fuel Fahrzeug aktuell im LPG (Liquefied Petroleum Gas) Betrieb
 - HYB_ELEC – Hybrid Fahrzeug aktuell im rein elektrischen Betrieb
 - 0x52 – Prozentsatz des Alkohols (Methanol oder Ethanol) im Kraftstoffgemisch
 - 0x5B – verbleibende Ladung / Ladezustand (Elektroantrieb)
 - 0x5C – Öltemperatur
 - 0x5D – Einspritzwinkel
 - 0x5E – Kraftstoffverbrauch (Momentanverbrauch in Liter/h berechnet über die letzte Sekunde)
 - 0x70 – Ladedruck Kontrolle (Sollwert, Istwert, Status für maximal 2 Turbolader/Kompressoren)

5.2 Auflistung der Unterfunktionen (InfoTypes) für Service 0x09

Für den OBD2 Service 0x09 erfolgt die Anzeige der meisten Unterfunktionen im Vehicle/ECU Info Menü. Falls die Anzeige in anderen Menüs erfolgt, ist das Menü bei der Unterfunktion angegeben.

0x00 – unterstützte InfoTypes → Protocol Info Menü 4.5.4

0x02 – VIN (Vehicle Identification Number) – FIN (Fahrzeugidentifikationsnummer)

0x04 – CALIDs (Calibration Identifier)

– Steuergeräte Software Kalibrierungsnummern (z.B. Software Stand, Datensatz Stand)

0x06 – CVNs (Calibration Verification Number)

– Verifikation der Steuergeräte Software Kalibrierung (Checksummen)

0x08 – IPT (In-use Performance Tracking)

– Leistungskontrolle, Statistik einzelner Teile/Systeme der Abgasregelung für Dieselfahrzeuge vor dem Modelljahr 2010 und für Benziner

0x0A – ECU Name (Electronic Control Unit) – Steuergerätename

0x0B – IPT (In-use Performance Tracking)

– Leistungskontrolle, Statistik einzelner Teile/Systeme der Abgasregelung für Dieselfahrzeuge ab Modelljahr 2010

6 Einschränkungen

Da der Kern des HHGui Programms die bis auf die Display-Auflösung und bis auf die Anzahl von Fehlertexten unveränderte AVR AT90CAN128 Mikrocontroller Firmware des OBD2-Analyser NGs ist, leidet HHGui unter den Hardware Eigenschaften dieses Mikrocontrollers. Vor allem die RAM Beschränkung von nur 4096 Bytes ist ein Problem. Wenn mehr als 2 OBD2 Steuergeräte in einem Fahrzeug antworten, kann es in seltenen Fällen zum Überlaufen des 256 Bytes großen Empfangspuffers (Ringpuffer) kommen, so dass Daten verloren gehen. Das ist teilweise abhängig von der Rechenleistung des verwendeten Raspberry Pis und weiteren gleichzeitig auf dem Pi laufenden Programmen. Beides hat Einfluss darauf, wie schnell Daten aus dem Empfangspuffer für die Weiterverarbeitung umkopiert werden können.

Maximale Anzahl bestätigter Fehler pro Steuergerät: 16

Maximale Anzahl bestätigter Fehler für alle Steuergeräte: > 16 (*)

Maximale Anzahl schwebender Fehler pro Steuergerät: 16

Maximale Anzahl schwebender Fehler für alle Steuergeräte: > 16 (*)

Maximale Anzahl permanenter Fehler pro Steuergerät: 16

Maximale Anzahl permanenter Fehler für alle Steuergeräte: > 16 (*)

Maximale Anzahl CALIDs: 3 (CAN Protokolle), 2-3 (non-CAN Protokolle, abhängig vom Pi, s.o.)

Maximale Anzahl CVNs: 3

(*) Die maximale Anzahl von Fehlern hängt vom Protokoll, von der Anzahl der antwortenden Steuergeräte, von der Reihenfolge der von den Steuergeräten empfangenen Daten und von der Rechenleistung des verwendeten Raspberry Pis ab. Bei 2 mit CAN Protokoll antwortenden Steuergeräten wird die für 2 Steuergeräte maximale Zahl von 32 (2 x 16 Fehler pro Steuergerät) erreicht.

Wenn mehr als 2 Steuergeräte antworten, die alle den Service 0x09 InfoType 0x0A (Steuergerätename) unterstützen, werden im Change ECU Menü zuerst nur die Namen von 0 - 2 Steuergeräten angezeigt. Erst nachdem der Steuergerätename für die anderen Steuergeräte im Vehicle/ECU Info Menü gelesen wurde (in diesem Menü ist im Pi-OBd Modul ein Filter für das aktive Steuergerät gesetzt, so dass die Daten immer in den Empfangspuffer passen) können auch weitere Steuergerätenamen im Change ECU Menü angezeigt werden.