

OBD Gauge

Automotive Diagnostic Software

for Palm and Pocket PC



Table of Contents

Introduction	3
Description	3
OBD-II	3
License.....	3
Web Site.....	3
OBD-II Interface Hardware	4
ScanTool	4
Multiplex Engineering.....	4
Software Installation	5
Palm	5
Compatible Devices	5
Incompatible Devices	5
Other Devices.....	5
Installation	5
Pocket PC	6
Compatible Devices	6
Prerequisites.....	6
Installation	6
Windows PC.....	6
Starting OBD Gauge.....	7
Displaying Sensor Data	8
Switching Pages	8
Customizing Sensor Display	8
Diagnostic Trouble Codes	9
Acceleration Timer (Palm Only).....	10
Recording Sensor Data	11
Displaying Recorded Sensor Data	11
Preferences	13
Language	13
Display (Palm only)	13
Units	13
Scan	13
Graph	13
Interface	14
Setup.....	14
ELM	14
Multiplex	14
Port (Pocket PC only)	14
OBD-II Sensor Description	15
Throttle Position	15
Engine RPM.....	15
Vehicle Speed.....	15
Calculated Load Value.....	15
Ignition Timing Advance.....	15
Intake Air Pressure	15
Intake Air Flow Rate.....	16
Short Term Fuel Trim.....	16
Long Term Fuel Trim	16
Air Temperature	16
Coolant Temperature	16
Coolant Temperature	16
Fuel Pressure.....	17
Oxygen Sensors	17

Introduction

Description

This document describes the installation and use of *OBD Gauge* and *OBD Graph* software.

OBD Gauge for Palm or Pocket PC provides the following features:

- displays sensor data
- displays Diagnostic Trouble Codes (DTCs)
- clears the Maintenance Indicator Light (MIL) (aka "check engine" light)
- acceleration timer (Palm only)
- records sensor data

The *OBD Graph* software for Microsoft Windows allows data recorded by *OBD Gauge* to be displayed and analyzed. This PC software provides these additional features:

- displays graphs of sensors data
- displays tabular view of sensor data
- exports sensor data to CSV file for Excel or other applications

OBD-II

OBD-II is an acronym for On Board Diagnostics. It is a standardized interface that is present on 1996 and newer vehicles which allows diagnostic equipment to access sensors data and DTCs. *OBD Gauge* requires a hardware interface to adapt the OBD-II interface to a serial interface which is compatible with the Palm or Pocket PC. See the "OBD-II Interface Hardware" section for more information.

License

This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

Web Site

For more information, please visit <http://www.qcontinuum.org/obdgauge/>.

If you find this software useful and wish to give a token of thanks, please consider a donation of \$5 Canadian by clicking the *PayPal* button on this web site. Please do not feel obligated to donate; *OBD Gauge* is free software.

OBD-II Interface Hardware

ScanTool

ScanTool (<http://scantool.net/>) offers these OBD-II interfaces which are compatible with *OBD Gauge*:

- ElmScan ISO
- ElmScan VPW
- ElmScan PWM

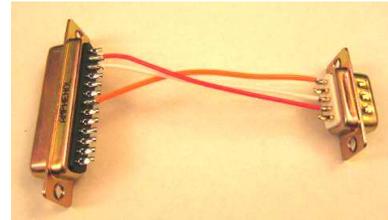


Purchase the version appropriate for your car, as well as an OBD-II cable. The ElmScan 5 is not yet supported by *OBD Gauge*. Support for the ElmScan 5 will be added in a future version.

ElmScan interfaces use a DB-25 serial connector that is intended for connection to a PC. To connect an ElmScan interface to a Palm or Pocket PC, an adapter is needed.

One option is to build your own adapter. You will need a male DB-25 and a male DB-9. Use three wires to connect the following pins:

DB-25	DB-9
Pin 2	Pin 2
Pin 3	Pin 3
Pin 7	Pin 5



If you do not wish to build your own serial adapter, you can use off-the-shelf components to achieve the same result.

Pictured here are:

- Male DB-25 to female DB-9 cable
- Null modem
- Gender changer

Use any other combination of cables and adapters, provided one of them acts as a null modem (swapping Tx and Rx).



Multiplex Engineering

Multiplex Engineering (<http://www.multiplex-engineering.com/>) offers these OBD-II interfaces which are compatible with *OBD Gauge*:

- T16-003 (integrated OBD-II connector, shown here)
- T16-006 (requires OBD-II cable)



The OBD-II protocol is software selectable. The hardware supports ISO, VPW, PWM, and KWP. These interfaces plug directly into a serial PDA cable.

The T16-009 also supports the CAN protocol, but is not yet supported by *OBD Gauge*. Support for the T16-009 will be added in a future version.

Software Installation

Palm

Compatible Devices

OBD Gauge will work on any model of Palm that has serial capability, including:

- PalmPilot series
- Palm III series
- Palm V series
- Palm VII series
- m100 series
- m500 series
- Tungsten (T, T2, T3, C)
- i705
- Zire 71



Newer Palms generally come with a USB cradle only, but also have serial capability. It is necessary to purchase the appropriate serial HotSync cable for your model of Palm.

Older Palms come with a serial cradle which may be used with *OBD Gauge*, although you may wish to purchase a cable for greater convenience.

The Tungsten T and T3 owners may need to purchase shareware software called SerialFix which corrects a serious defect in the PalmOS serial port driver. You can find it at Palm Gear (search <http://www.palmgear.com/>).

Incompatible Devices

There are a few models of Palm will not work because they are USB only, and lack the necessary serial capability. The models which will **NOT** work include:

- Zire, Zire 21, Zire 31, Zire 72
- Tungsten E

It is not possible to use a USB to serial adapter, because these adapters are designed to be used with a PC only.

Other Devices

OBD Gauge may work with other devices that run PalmOS. Because of the variety of devices available, no attempt has been made to catalogue them. Most PalmOS devices which have a serial port will work with *OBD Gauge*.

Installation

On a PC with *Palm Desktop* software installed, use the *Palm Quick Install* utility to load the following files from the "Palm" directory of this distribution:

- OBD Gauge.prc
- DTC-Generic.pdb

The DTC-Generic file is optional. If you do not load it, *OBD Gauge* will display only the DTC codes without the associated descriptions.

Pocket PC

Compatible Devices

OBD Gauge works on any Pocket PC 2002 or Pocket PC 2003 with a serial port. You will likely need to purchase a serial cable as most Pocket PCs come with only a USB cradle.

Prerequisites

OBD Gauge requires the *Microsoft .NET Compact Framework*. Pocket PC 2003 devices already have this software built-in. Pocket PC 2002 devices must have this software installed. Go to <http://msdn.microsoft.com/netframework/downloads/updates/> to download.

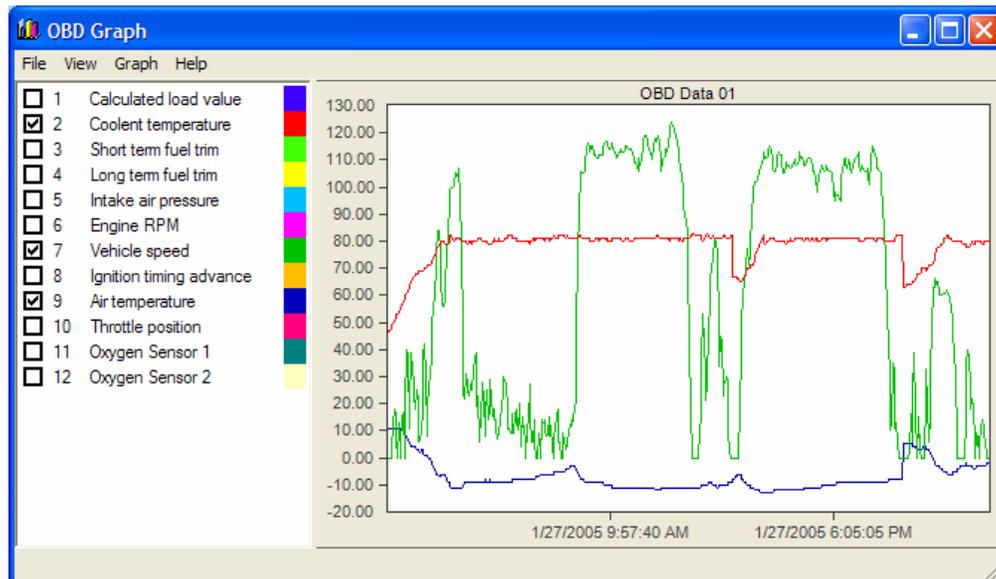
Installation

On a PC with *Microsoft ActiveSync* software installed, connect your Pocket PC device and run **Setup.exe** from the "Pocket PC" directory of this distribution.



Windows PC

OBD Graph is a utility for the PC to view sensor data recorded by *OBD Gauge*. No installation is necessary; run **OBDGraph.exe** from the "PC" directory of this distribution. If an error dialog is produced, the most likely reason is that you do not yet have the *Microsoft .NET Framework* installed. Go to <http://msdn.microsoft.com/netframework/downloads/updates/> to download.



Starting OBD Gauge

After acquiring the necessary hardware and installing the software on your Palm or Pocket PC, you are ready to start *OBD Gauge*.

- Connect a serial cable to the Palm or Pocket PC
- Connect the serial cable to the OBD-II interface, using an adapter if necessary
- Connect the OBD-II interface to the vehicle's OBD-II connector
- Start the vehicle's engine
- Tap the  icon to start *OBD Gauge*

If you are using a Multiplex Engineering OBD-II interface, these additional steps are necessary:

- Choose the following menu item:

Palm	Pocket PC
Options → Preferences	Menu → Preferences

- Under "Interface", select "Multiplex"
- Tap the "Setup" button
- Choose the "Baud" and "Address" setting as appropriate for your interface; typical settings are 19200 baud, address 25
- Choose the "Protocol" setting as appropriate for your vehicle

The title bar will initially display the message "Initializing...".

If successful, the title bar will display "Scan" followed by the average scan rate. The scan rate is the number of sensors reads per second.

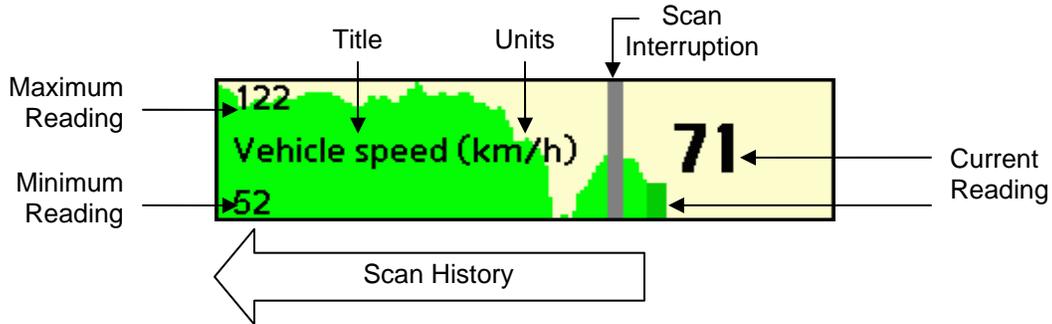
If the title bar continues to display "Initializing...", then check the following:

- Ensure the vehicle engine is started, or at least that the ignition is "on"
- Check the cabling
- If using the Multiplex Engineering interface, check the Preference settings

Displaying Sensor Data

Once *OBD Gauge* passes the “Initializing...” phase and starts scanning, the available sensors will be updated in sequence.

Each sensor is displayed in the following format:



Switching Pages

There are typically too many sensors to fit on the screen at one time. The sensors are divided evenly between 4 pages. Select the active page by choosing “1”, “2”, “3”, or “4” at the bottom of the screen.

On the Palm, the active page may also be selected using the application keys or menu items:

Sensor Page	Palm Application Key	Palm Menu Item
Page 1	Calendar	Page → View Page 1
Page 2	Address Book	Page → View Page 2
Page 3	To Do List	Page → View Page 3
Page 4	Notes	Page → View Page 4

Customizing Sensor Display

To customize the sensor display, select a sensor by tapping it. A highlight rectangle appears around the selected sensor. The selected sensor can then be moved up, down, to another page, or removed from the display using the following menu items:

Palm	Pocket PC
Gauge → Move Up	Menu → Selected Gauge → Move Up
Gauge → Move Down	Menu → Selected Gauge → Move Down
Gauge → Move To Page 1	Menu → Selected Gauge → Move To Page 1
Gauge → Move To Page 2	Menu → Selected Gauge → Move To Page 2
Gauge → Move To Page 3	Menu → Selected Gauge → Move To Page 3
Gauge → Move To Page 4	Menu → Selected Gauge → Move To Page 4
Gauge → Remove	Menu → Selected Gauge → Remove

To restore all sensors which have been removed, and to restore the layout to the default configuration, use the following menu item:

Palm	Pocket PC
Page → Default Layout	Menu → Default Layout

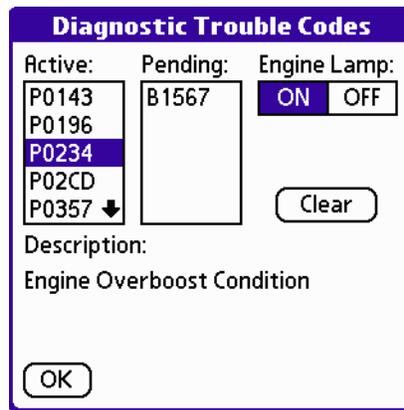
Removing unwanted sensors will increase the rate at which the desired sensors are updated.

Diagnostic Trouble Codes

If the vehicle's "Check Engine" light is illuminated, this is an indication that one or more diagnostic trouble codes have been logged by the on-board computer (ECU). To display these DTCs, select the following menu item:

Palm	Pocket PC
Options → Diagnostic Trouble Codes	Menu → Diagnostic Trouble Codes

The DTCs are displayed along with a description, if a description is available. If no DTCs are present, the lists will be blank.



"Active" DTCs are problems which currently exist or have existed in the past.

"Pending" DTCs are potential problems which may become "Active" if they persist.

The DTCs will remain stored inside your vehicle's ECU even after you fix the problem. To reset the ECU and turn off the MIL:

- Tap the "Clear" button
- Turn the ignition key to "off" to stop the engine
- Return the key to the "on" position
- Tap the "OK" button on the confirmation dialog

If you clear the DTCs without first fixing the underlying problem, the DTCs will likely return.

After you reset the vehicle's ECU, it may need to relearn its optimum performance parameters. Any immediate symptoms such as rough idling should go away as soon as the full vehicle on-board test cycle is complete.

You can find the definitions for the DTCs from many sources on the internet as well as from vehicle manufacturers. A standard named SAE-J2012 on diagnostic trouble codes defined all the trouble codes preceded by the alphanumeric designator 'P0'. However, if the alphanumeric designator is 'P1' then the DTC is manufacturer defined and you will need a shop manual for the vehicle to decode that DTC. For example, if a trouble code of P0150 was displayed, then it is SAE controlled and would indicate a problem in the O2 sensor. If the DTC was P1298 then you would have to consult the shop manual for an explanation.

Acceleration Timer (Palm Only)

Choose the menu item Options → Acceleration Timer.



Step 1

Select a straight road clear of traffic on which you can safely accelerate rapidly. Bring the vehicle to a complete stop.



Step 2

When it is safe to do so, start accelerating from stop. The timer will start running automatically. The target speed is 100 km/h if Metric units are selected in Preferences. Otherwise, the target is 60 MPH.



Step 3

Keep your eyes on the road! The current speed is displayed and the elapsed time since the vehicle was stopped.



Step 4

The elapsed time is displayed, along with an estimate of the measurement error. The faster the scan rate, the smaller the error.

Recording Sensor Data

OBD Gauge has the ability to save the sensor data to a data log. The data log can be viewed using the *OBD Graph* utility on the PC.

To start, stop, or clear the recording of sensor data, select the following menu items:

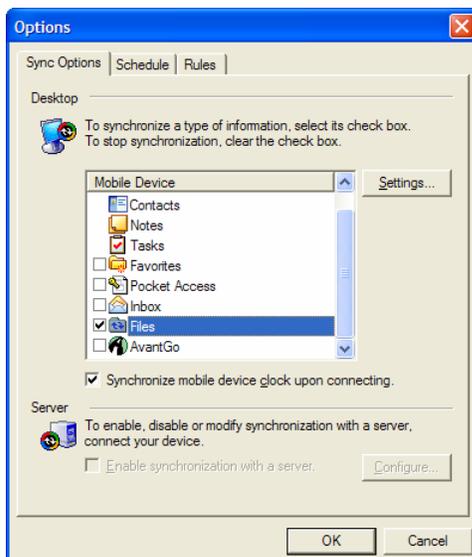
Palm	Pocket PC
Options → Start Recording	Menu → Recording → Start
Options → Stop Recording	Menu → Recording → Stop
Options → Clear Recording	Menu → Recording → Clear

While recording is in progress, the title bar displays “Record” instead of “Scan”.

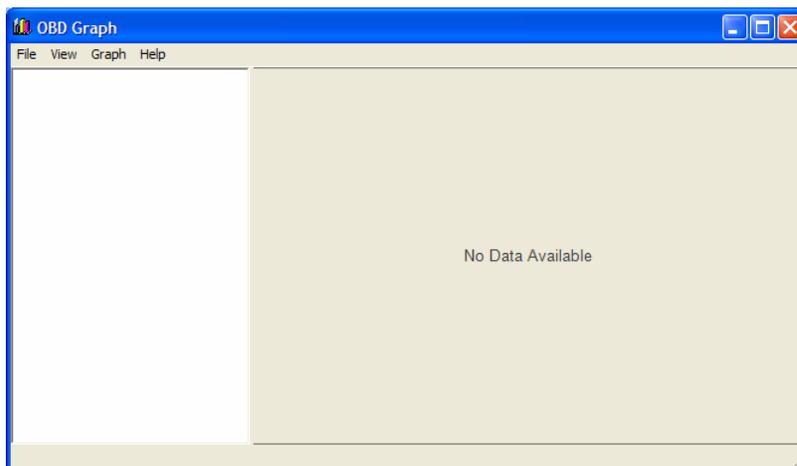
You may wish to verify that the Palm or Pocket PC time is set correctly so that sensor data is given the correct timestamp.

Displaying Recorded Sensor Data

Before sensor data can be displayed on the PC using *OBD Graph*, you must first perform a synchronization to copy the recorded data to the PC.



For the Pocket PC, it is necessary to enable “File” synchronization in the “Options” dialog of *Microsoft ActiveSync* before performing the synchronization.

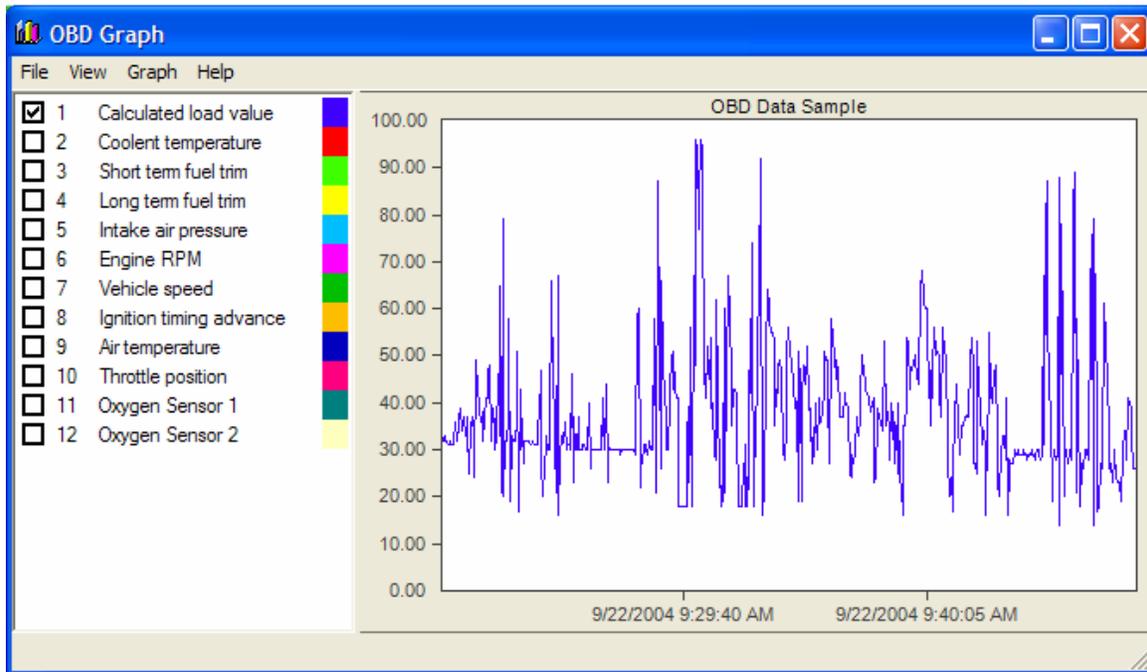


Start **OBDGraph.exe** from the “PC” directory of the distribution.

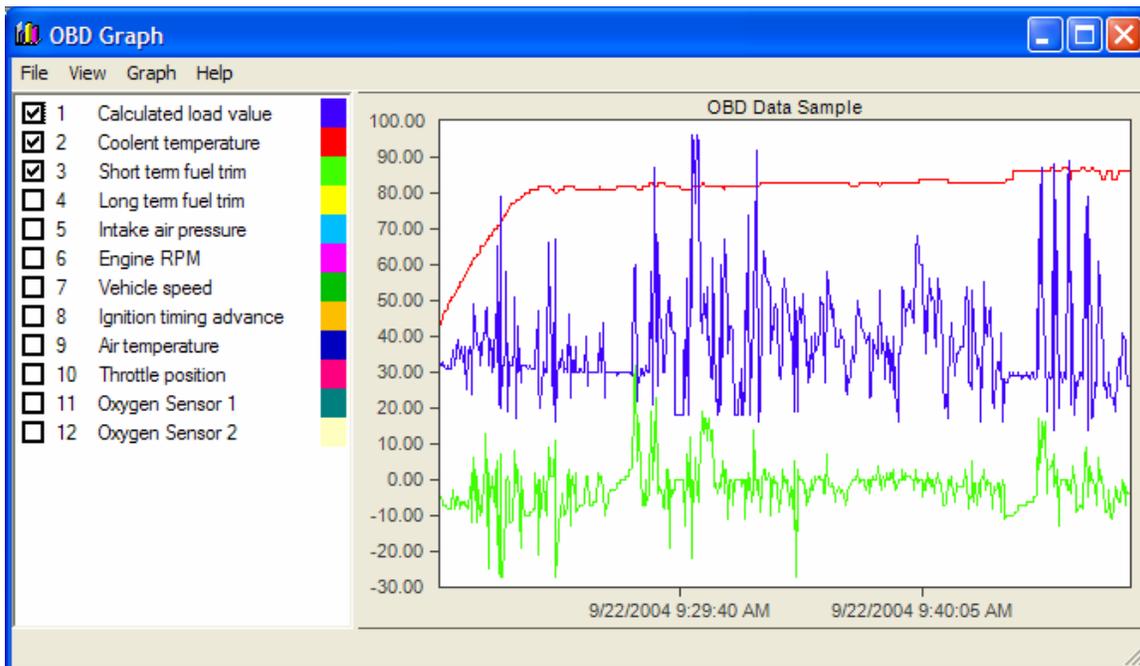
Select the following menu item:

Palm	Pocket PC
File → Open OBD Data from Palm	File → Open OBD Data from Pocket PC

The recorded data will be displayed:



Select the sensors to be displayed using the checkboxes:



Preferences

To adjust *OBD Gauge* preferences, choose the following menu item:

Palm	Pocket PC
Options → Preferences	Menu → Preferences

Language

This option chooses the language used for sensor titles. The rest of the user interface is always English and is unaffected by this selection. The available languages are:

- English (default)
- French
- German
- Dutch
- Spanish

Display (Palm only)

This option allows user interface elements to be hidden to increase the screen real estate available for displaying sensor information. The available selections are:

- Show both (default)
- Hide menu bar
- Hide page numbers
- Hide both

Units

This option selects the type of units used to display sensor information:

- Metric (default)
- U.S. (British Imperial units)
- U.K. (Metric except speed in MPH)

Scan

This option selects which sensors to scan:

- All sensors (default)
- Displayed sensors (only sensors on the active page)

Choosing “displayed sensors” results in a faster update rate. No information will be collected from sensors that are not currently displayed.

Graph

This option selects the format of the sensor graphs. Choose the format you find most appealing and/or readable:

- Line (default)
- Bar

Interface

This option selects the type of OBD-II interface hardware you have connected:

- ELM (an ElmScan interface from ScanTool)
- Multiplex (a Multiplex Engineering interface)

The “Setup” button allows interface-specific options to be changed.

Setup

ELM

The “Timeout” parameter controls how long the interface waits for a response after initiating a sensor query. The default is 200ms. Reducing this time will increase the rate which sensors can be queried. Reducing the timeout too much will cause the scan to fail and stop completely. A setting of 50ms is a typical minimum.

Multiplex

Baud: 9600 or 19200

Address: hexadecimal 00 to FF

Protocol: Disabled, ISO, VPW, PWM, or KWP

The baud and address settings must match that used by the Multiplex interface. Typical settings for a Multiplex interface purchased for use with non-commercial software such as *OBD Gauge*:

Baud: 19200

Address: 25

The protocol selection setting depends on the type of vehicle being scanned.

Port (Pocket PC only)

This option selects which serial port to use. Typically, Pocket PCs have only one serial port, COM1. This can be useful for selecting another port when using Bluetooth to establish a wireless serial connection. Vital Engineering (<http://www.vitalengineering.co.uk/>) makes a Bluetooth capable OBD-II interface that is compatible with *OBD Gauge*.

OBD-II Sensor Description

This is a complete list of the sensors that *OBD Gauge* supports. Only those sensors that are also supported by the connected vehicle are displayed.

Throttle Position

This sensor generates a voltage proportional to instantaneous throttle position (0-90°). It is measured in degrees. Usually a closed throttle will read about 10° and a fully open throttle about 79.3° to 90°. This sensor is used to determine the position of the throttle plate. It informs the engine management computer if the throttle plate is closed, wide open or partially open. The sensor is basically a potentiometer providing an output voltage. The closed throttle position may be used to determine when the accelerator pedal is not being pressed.

Engine RPM

This sensor generates a pulse for every revolution of the engine. It is measured in revolutions per minute. Usually the sensor is a pick up coil looking for a unique keyed tooth on an engine timing rotor. Values range from 0 upwards.

Vehicle Speed

The vehicle speed sensor tells the vehicle's engine management computer how fast the vehicle is moving. Speed is measured in kilometers per hour (km/h) or miles per hour (MPH). The vehicle's engine management computer adjusts the injector pulse times to maintain the proper fuel mixture at any speed and load. It also sends a signal to the speedometer and, in most cases, to the cruise control computer.

Calculated Load Value

The vehicle's engine management computer generates a value of between 0 to 100% to represent current loading.

Ignition Timing Advance

This system is used on spark ignition and combustion ignition engines to start the ignition event earlier by controlling the ignition or fuel system. As engine speed increases, start of ignition or start of injection must occur earlier in order to result in the most effective downward thrust on the piston during the power stroke. The vehicle's ECU generates this value, measured in degrees (-64° to 63.5°).

Intake Air Pressure

This sensor creates a signal that is proportional to the average pressure in the intake manifold. The pressure should be low and fairly steady at idle and light loads and will go higher as the load increases, up to atmospheric pressure.

Intake Air Flow Rate

This sensor provides a voltage signal, corresponding to the air mass entering the engine. The electrical signal is transmitted to the engine control module and is used to determine the required fuelling for petrol engines and for exhaust gas recirculation (EGR) control on diesel engines. It is typically mounted inside the clean air duct (usually at the exit of the air cleaner). Modern vehicles use a 'hot wire anemometer' sensor for this measurement.

Short Term Fuel Trim

(STFT) – The neutral value for the short term fuel trim is 0%. Any deviation from 0% indicates the short term fuel trim is changing the injector pulse width. The amount of pulse width change depends on how far the short term fuel trim value is from 0%. The short term fuel trim is rich at -99% and lean at +99%.

The short term fuel trim changes the pulse width by varying the Closed Loop factor of the base pulse width equation. As the vehicle's engine management computer monitors the oxygen sensors input, it is constantly varying the short term fuel trim value. The value is updated very quickly. The short term fuel trim only corrects for short term mixture trends. The correction of long term mixture trends is the function of long term fuel trim.

Long Term Fuel Trim

(LTFT) – As the engine operating conditions change, the vehicle's engine management computer will determine what long term fuel trim factor to use in the base pulse width equation.

If the short term fuel trim is far enough from 0%, the vehicle's engine management computer will change the long term fuel trim value. Once the LTFT value is changed, it should force the STFT back toward 0%. If the mixture is still not correct, the STFT will continue to have a large deviation from the ideal 0%. In this case, the LTFT will continue to change until the STFT becomes balanced. Both the STFT and LTFT have limits which vary by calibration. If the mixture is off enough so that LTFT reaches the limit of its control and still cannot correct the condition, the STFT would also go to its limit of control in the same direction. If the mixture is still not corrected by both STFT and LTFT at their extreme values, a Fuel Trim DTC will likely result.

Air Temperature

The intake air temperature sensor tells the computer the temperature of the incoming air. Since cold air is denser than warm air, it needs more fuel to achieve the ideal air to fuel ratio. To do this, the computer will open the injectors for a longer period of time.

Coolant Temperature

This sensor measures the temperature of the engine. When the engine is cold, it needs more fuel to operate correctly. In very cold conditions, a large quantity of fuel is needed to start the car. This is called "Cold Start Enrichment" and replaces the function of a choke.

Coolant Temperature

This sensor measures the temperature of the engines coolant. Normally about 82 to 104°C (180 to 220°F) on most cars once the engine has warmed up .

Fuel Pressure

This sensor measures the regulated pressure on the fuel rail. This is a very important value; the fuel/air ratio is determined by length of time the fuel injector is pulsed on. If the fuel pressure is not constant, then the ECU cannot maintain the correct fuel/air ratio, causing emissions or performance problems.

Most regulators have a static pressure of between 38 and 44 PSI.

Oxygen Sensors

All modern vehicles have multiple oxygen sensors. The number of sensors is always even. The sensors are part of the emissions control system and feed data to the engine management computer. The goal of the sensor is to help the engine run as efficiently as possible and also to produce as few emissions as possible.

A gasoline engine burns fuel in the presence of oxygen. An oxygen/fuel ratio of approximately 14.7:1 is ideal (stoichiometry). The ratio depends on the amount of hydrogen and carbon found in a given amount of fuel. If there is less air than this ideal ratio, then there will be fuel left over after combustion. This is called a rich mixture. Rich mixtures are bad because the unburned fuel creates pollution.

If there is more air than this perfect ratio, then there is excess oxygen. This is called a lean mixture. A lean mixture tends to produce more nitrogen-oxide pollutants, and in some cases, it can cause poor performance and even engine damage.

An oxygen sensor is positioned in the exhaust pipe and can detect rich and lean mixtures. The mechanism in most sensors involves a chemical reaction that generates a voltage. The engine's computer looks at the voltage to determine if the mixture is rich or lean, and adjusts the amount of fuel entering the engine accordingly.

When the oxygen sensor fails, the computer can no longer sense the air/fuel ratio. This sensor is much more complicated than the other sensors and deserves a more detailed explanation. Basically, the oxygen sensor indicates the presence or absence of oxygen in the exhaust stream. The ECU, in order to keep emissions low, adjusts the fuel/air ratio at stoichiometry, the exact mixture which will completely burn all fuel with the oxygen present in the cylinder, leaving no fuel or oxygen in the exhaust.

As the fuel/air ratio deviates from stoichiometry, then either oxygen or hydrocarbons (unburned fuel) will be present in the exhaust. The oxygen sensor produces a low voltage ($< 0.1V$) when oxygen is present in the exhaust, and produces a high voltage ($> 0.8V$) when there is no oxygen in the exhaust. The ECU uses the voltage output from the oxygen sensor to make slight adjustments to the amount of fuel injected to keep the fuel/air ratio near stoichiometry. It does this by increasing the fuel to get a high reading, meaning that there is no oxygen in the exhaust, the slightly decreasing the amount of fuel until the oxygen sensor reading drops low. This process goes on continuously to keep the average fuel/air ratio as close to stoichiometry as possible.