

Vital Engineering Ltd

Car-Pal commands for Vehicle Data Access

The Car-Pal command set has been designed to be simple yet powerful. The Car-Pal OBD interface unit takes care of all the OBDII protocol level details so you do not have to. Even for CAN.

The Car-Pal OBD interface unit is available in two different units, one with a serial (RS232) port connection and one with Bluetooth.

Connection to the Car-Pal OBD Interface Unit

Serial: To access the Car-Pal OBD interface unit using a serial (RS232) set your communication to 8 data, no parity, 1 stop bit and no handshaking.

Bluetooth: To access the Car-Pal OBD interface unit using Bluetooth, first discover the Car-Pal OBD interface unit which has a Bluetooth device name of “Car-Pal OBD XXXXXX”, “XXXXXX” is a six digits number which is unique to each Car-Pal OBD interface unit with Bluetooth. Note that this is NOT Bluetooth Address. Every Car-Pal OBD interface unit with Bluetooth has been set up with a passkey (pass code) PIN number of “0000” (zero, zero, zero, zero). Your Bluetooth software will need to handle a pairing request and submit this passkey PIN number when requested by the Car-Pal OBD interface unit with Bluetooth.

Perform a service discovery on the Car-Pal OBD interface unit with Bluetooth and you will discover one Bluetooth service supported. This is a Bluetooth serial port profile compliant service which is named “OBD Serial”. Connect to this using your Bluetooth software.

Once connected via (RS232) or Bluetooth (via the serial port service profile) the communications to the Car-Pal interface unit will be the same. The underlying communications system has been abstracted.

Car-Pal OBD Commands

There are two different types of commands for communication with the Car-Pal OBD interface unit.

High Level Commands:

When sending to the Car-Pal OBD interface unit the Service Number and the PID a response of 7 data bytes (as specified in ISO 1531-5) is received.

For example:

<STX> <03> Service number 3 and no PID to get the stored error codes.

<STX> <02> <frame> gets the freeze frame stored in ECU.

<STX><04> clears the DTC stored

<STX><01><00> gets the PID supported and

<STX><01><04> gets the coolant temperature in 1 Byte in the response Bytes.

Low Level Commands:

In this mode the adaptor can speak to the ECU directly. First we need to request the protocol found in the vehicle. For J1850 for example, to talk to the ECU you need to send the following command to the Car-Pal OBD interface unit:

< length> <0x68> <0x6a> <0xf1> <service number> <PID>

As a response we get the following:

< length><0x48><0x6b><ECU address ><0x40 + service ><PID> (for service 1)

At power up the host must wait until Car-Pal discovers the OBD protocol the vehicle is using. This should take no more than 6 sec. The found protocol can be requested with command 0x05. If no protocol is found this number will be zero. The host software can also search for the OBD protocol used in the vehicle. The software must set the timeout of 6 seconds for the response.

Function numbers are all decimals, but 0x30 means hexadecimal. Car-Pal has a timeout period of 1000ms for an incoming message. Unused commands will return <Command number> <00>

First let Car-Pal search for a valid OBD protocol using default parameters. Use set header command to set different ECU address. Generic commands can only be used when Car-Pal finds a valid protocol at power up.

Command	00
Function	Reserved
Request	-
Positive response	N/A
Negative response	N/A

Command	02
Function	OBDII / EOBD generic command
Request	02
Positive response	<06>
THEN SEND	
MODE 1	<01><PID>
MODE 2	<02><PID>
MODE 3	<03>
MODE 4	<04>
MODE 5	<05> <PID> <O2SNO>
MODE 6	<06><PID>
MODE 7	<07>
MODE 8	<08><PID><DataA><DataB><DataC><DataD><DataE>
MODE 9	<09><INFO_Type>
Negative response	N/A

Command	03
Function	Read Car-Pal serial number
Request	03
Positive response	03 HB LB HB = high byte of the serial number LB = Lower byte of the serial number
Negative response	N/A

Command	04
Function	Read version number of the Car-Pal interface unit
Request	04
Positive response	04 HB LB HB = high byte of the serial number LB = Lower byte of the serial number
Negative response	N/A

Command	05																												
Function	Read found protocol																												
Request	05																												
Positive response	<div>05 HB LB</div> <div> <table> <tr> <td>0x0000</td><td>No protocol found</td></tr> <tr> <td>0x0001</td><td>ISO9141-2 keywords 0808</td></tr> <tr> <td>0x0002</td><td>ISO9141-2 keywords 9494</td></tr> <tr> <td>0x0004</td><td>KWP2000 slow init</td></tr> <tr> <td>0x0008</td><td>KWP2000 fast init</td></tr> <tr> <td>0x0010</td><td>J1850 PWM</td></tr> <tr> <td>0x0020</td><td>J1850 VPWM</td></tr> <tr> <td>0x0040</td><td>CAN 11 bit ID, 250KB Speed</td></tr> <tr> <td>0x0080</td><td>CAN 11 bit ID, 500KB Speed</td></tr> <tr> <td>0x0100</td><td>CAN 29 bit ID, 250KB Speed</td></tr> <tr> <td>0x0200</td><td>CAN 29 bit ID, 500KB Speed</td></tr> <tr> <td>0x0400</td><td>Reserved for SAE J1939</td></tr> <tr> <td>0x0800</td><td>Reserved for KW1281 /KW71</td></tr> <tr> <td>0x1000</td><td>Reserved for KW82</td></tr> </table> </div>	0x0000	No protocol found	0x0001	ISO9141-2 keywords 0808	0x0002	ISO9141-2 keywords 9494	0x0004	KWP2000 slow init	0x0008	KWP2000 fast init	0x0010	J1850 PWM	0x0020	J1850 VPWM	0x0040	CAN 11 bit ID, 250KB Speed	0x0080	CAN 11 bit ID, 500KB Speed	0x0100	CAN 29 bit ID, 250KB Speed	0x0200	CAN 29 bit ID, 500KB Speed	0x0400	Reserved for SAE J1939	0x0800	Reserved for KW1281 /KW71	0x1000	Reserved for KW82
0x0000	No protocol found																												
0x0001	ISO9141-2 keywords 0808																												
0x0002	ISO9141-2 keywords 9494																												
0x0004	KWP2000 slow init																												
0x0008	KWP2000 fast init																												
0x0010	J1850 PWM																												
0x0020	J1850 VPWM																												
0x0040	CAN 11 bit ID, 250KB Speed																												
0x0080	CAN 11 bit ID, 500KB Speed																												
0x0100	CAN 29 bit ID, 250KB Speed																												
0x0200	CAN 29 bit ID, 500KB Speed																												
0x0400	Reserved for SAE J1939																												
0x0800	Reserved for KW1281 /KW71																												
0x1000	Reserved for KW82																												
Negative response	N/A																												

Command	06
Function	Read chip ident from the Car-Pal OBD interface unit
Request	06
Positive response	06 HB LB HB = High byte of the Car-Pal ident LB = Lower byte of the Car-Pal ident TBD = Car-Pal
Negative response	N/A

Command	07
Function	Connect to the ECU
Request	07
Positive response	07 HB LB HB = High byte of the protocol found LB = Lower byte of the protocol found
Negative response	N/A

Command	08
Function	Disconnect
Request	08
Positive response	08 HB LB HB = High byte of the disconnect protocol LB = Lower byte of the disconnect protocol Note: ISO and KWP2000 needs 5 sec. to disconnect
Negative response	N/A

Command	09
Function	Get found keywords
Request	09
Positive response	09 KW1 KW2 This command returns the ISO / KWP read keywords
Negative response	N/A

Command	10
Function	K-Line monitoring
Request	10
Positive response	Car-Pal sends continuous data This command is inactive after RESET
Negative response	N/A

Command	12
Function	Send data direct to the ECU ISO9141-2
Request	12
Positive response	<ACK> Send <length><ISO message without checksum> Receive <12><length><response data>
Negative response	N/A

Command	13
Function	Send data direct to the ECU KWP2000
Request	13
Positive response	<ACK> Send <length><KWP message without checksum> Receive <13><length><response data>
Negative response	N/A

Command	18
Function	KWP2000 fast user init
Request	18
Positive response	<ACK> Send <length><KWP message without checksum> Receive <18><length><response data>
Negative response	N/A

Command	19
Function	Stop auto keep alive
Request	19
Positive response	<ACK> Send <keepISO> , <keepKWP> keepISO = 0 stop ISO keep alive keepISO = 1 start ISO keep alive keepKWP = 0 stop KWP keep alive keepKWP = 1 start KWP keep alive Receive <19><ACK>
Negative response	N/A

Command	24
Function	Send data direct to the ECU J1850 PWM
Request	24
Positive response	<ACK> Send <length><PWM message without checksum> Receive <24><length><response data>
Negative response	N/A

Command	25
Function	Send data direct to the ECU J1850 VPWM
Request	25
Positive response	<ACK> Send <length><VPWM message without checksum> Receive <25><length><response data>
Negative response	N/A

Command	33
Function	Send data direct to the ECU CAN 11/250
Request	33
Positive response	<ACK> Send <10><CAN message without crc> Receive <33><10><response data>
Negative response	N/A

Command	34
Function	Send data direct to the ECU CAN 11/500
Request	34
Positive response	<ACK> Send <10><CAN message without crc> Receive <34><10><response data>
Negative response	N/A

Command	35
Function	Send data direct to the ECU CAN 29/250
Request	35
Positive response	<ACK> Send <12><CAN message without crc> Receive <35><12><response data>
Negative response	N/A

Command	36
Function	Send data direct to the ECU CAN 29/500
Request	36
Positive response	<ACK> Send <12><CAN message without crc> Receive <36><12><response data>
Negative response	N/A