Espritmon V6.3 manual

July 11, 2021 – V1.0

Table of Contents

List of Figures

List of Tables

Introduction

Espritmon can read various datastreams from the ALDL connector in the Lotus Esprit models that have an OBD I compliant ECU, parse relevant data and display the results.

Furthermore, Espritmon can read and parse data from the Esprit ABS computer of the Delco Moraine Powermaster system and the SIR computer found in some Esprits.

Espritmon will store data received from the ECU, ABS computer and SIR computer in files with date/time coded filenames preceded with ‘esprit’, ‘abs’ or ‘sir’ and followed by the extension .log. The recorded data is in a comma separated value format and can be imported in other visualization tools or replayed by Espritmon.

Since the first version (Espritmon V0) from 2009, Espritmon has been updated a few times with new features. Also, most of the reported issues were resolved..

Espritmon was designed and is maintained by ErikS4. It is only to be distributed free of any charge and neither the program nor the manual may be sold.

This guide explains the main features of Espritmon V6.2.

System requirements

Espritmon is written to run on the Window OS. Version 6.2 has been successfully tested on Windows XP, Windows 7 and Windows 10.

Espritmon requires a COM port for the communication with an interface that translated the bi-directional TTL-level data from the OBD I interface to the uni-directional +-12V RXD and TXD lines of the RS232 interface. If the computer does not have a RS232 COM port, a USB to serial interface cable can be used.

The required interface between the ALDL port and the RS232 port is described in Appendix B.

Using Espritmon

Configuration

The downloadable version of Espritmon is configured to use serial port COM1. The port can be changed in the file lotus.cfg that is in the subdirectory CFG of Espritmon.

Operation

To start Espritmon, either click on the icon or open a console window, go to the Espritmon directory and type Espritmon.

After start, a text window and a graphics window will open. In the text window, the following messages should appear:

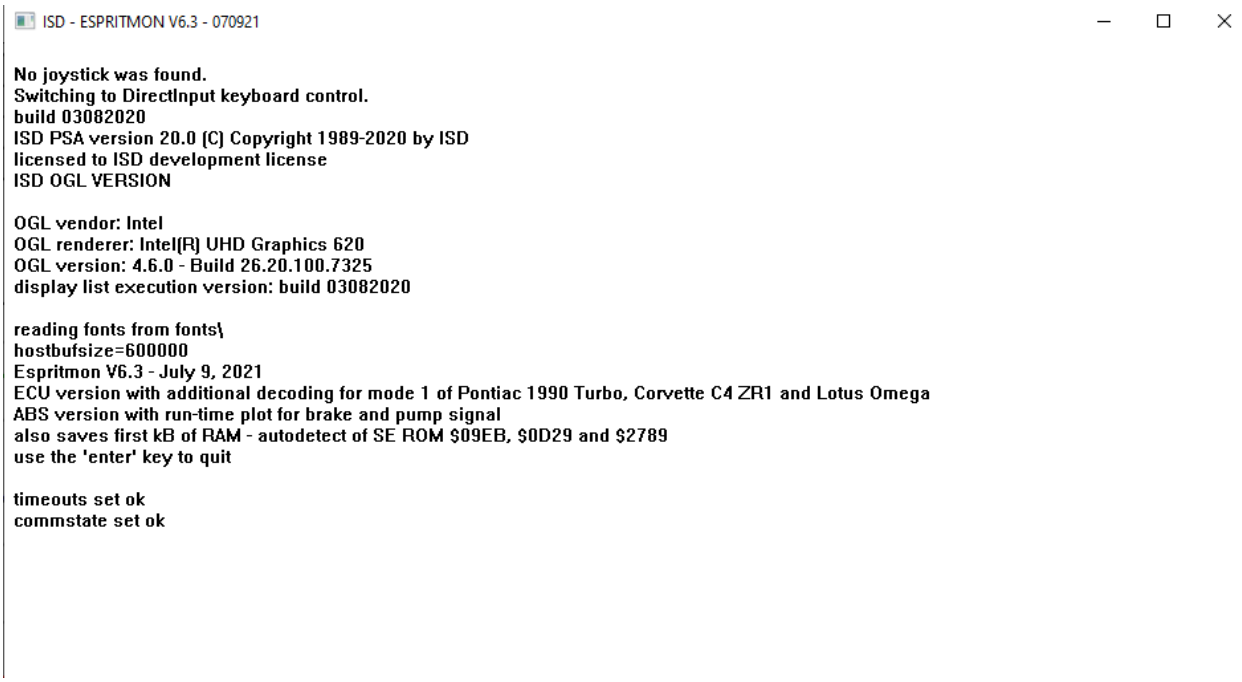


Figure 1. Text startup screen

The graphics window should look like the one in Figure 2 (if the Esprit is still powered off).

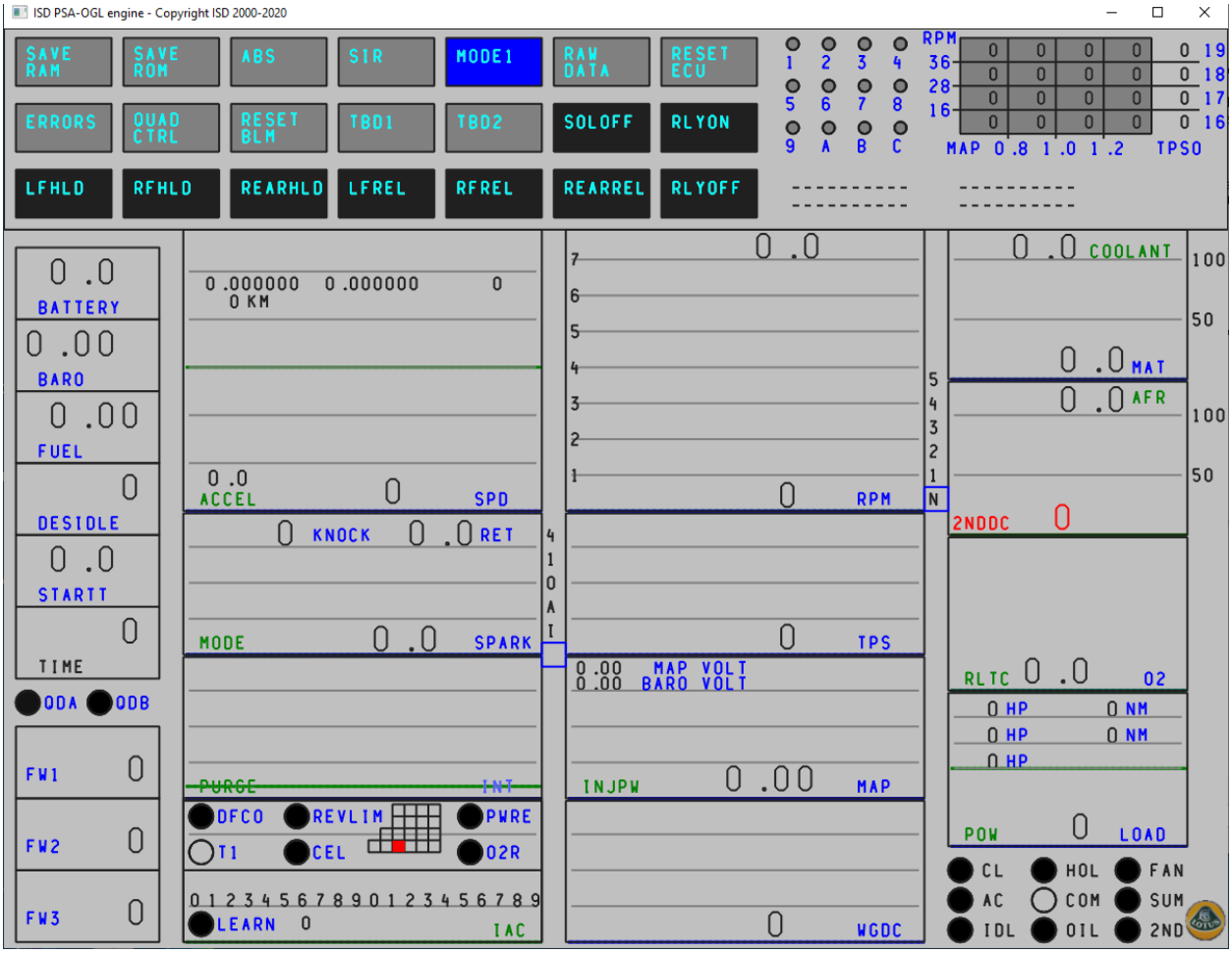


Figure 2. Main screen with ECU powered off

Once the Esprit is started, the incoming data will change the various readouts and graphs. Figure 3 presents an example for the main screen:

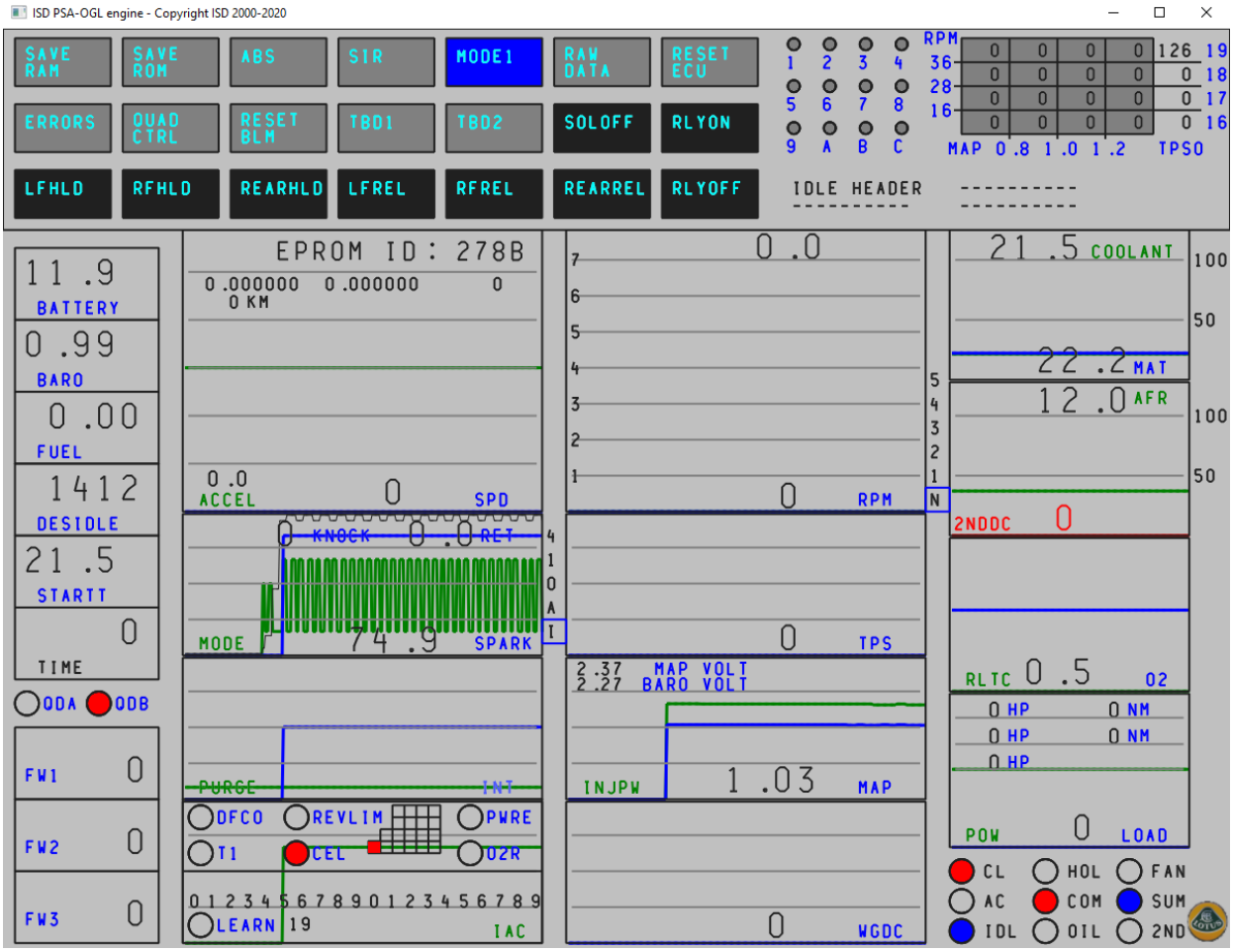
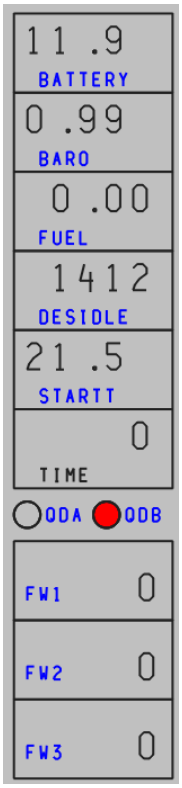


Figure 3. Main screen after ignition on

The following subsections describe the various panels on the main screen

Panel 1



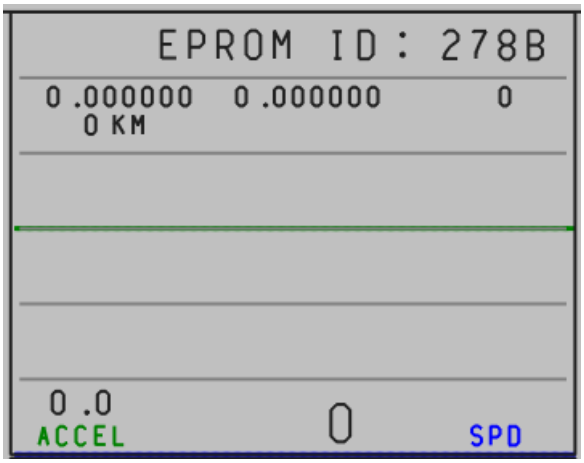
The values for battery voltage, baro pressure, fuel, desired idle, start temperature and time are obtained from the ALDL data. Time represents engine running time since the last engine start.

QDA and QDB indicate the quad driver error state as decoded from the status bits. When the engine is not running, QDB always is set to 1.

FW1 to FW3 display the contents of the 3 fault-word bytes in the mode 1 stream.

When not in mode 1 operation, the occasional ECU chatter of mode 0 and mode A provides only RPM, battery voltage and coolant temperature.

Panel 2



The EPROM id is decoded from bytes 0 and 1 in de ALDL mode 1 data

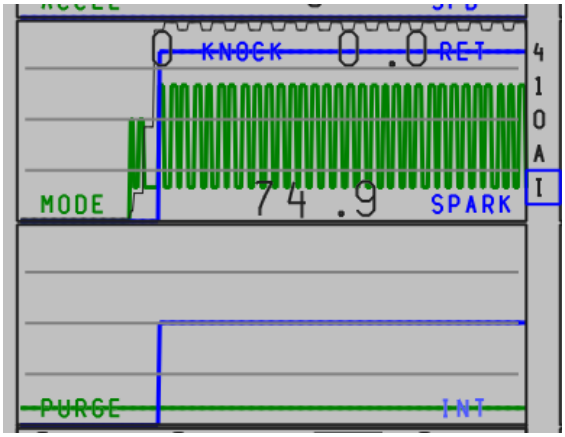
The speed is decoded from byte 13 and converted to km/h. A toggle to mph can be done by pressing the ‘f’ key on the keyboard.

Acceleration is estimated by differentiating speed.

Distance traveled is estimated by integrating speed

..

Panel 3



Knock, spark advance and retard are decoded from bytes 42, 38+39 and 43 respectively (see Appendix A).

The blue line in the upper plot shows a time-history of the spark advance.

The green line in the upper plot is in reference to the received mode indicated in the header and plotted against the scale on the right (I, A, 0, 1, 4)

The blue line in the lower plot shows the time-history of the integrator. The green line shows the time-history of the Canister Purge solenoid.

Panel 4

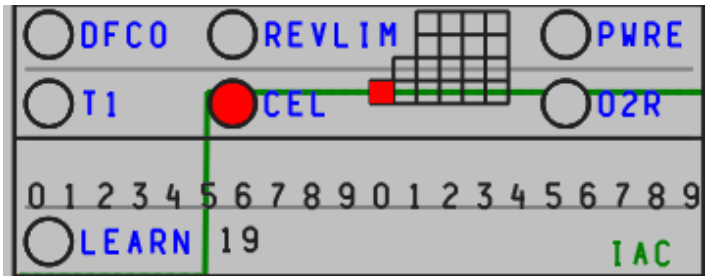
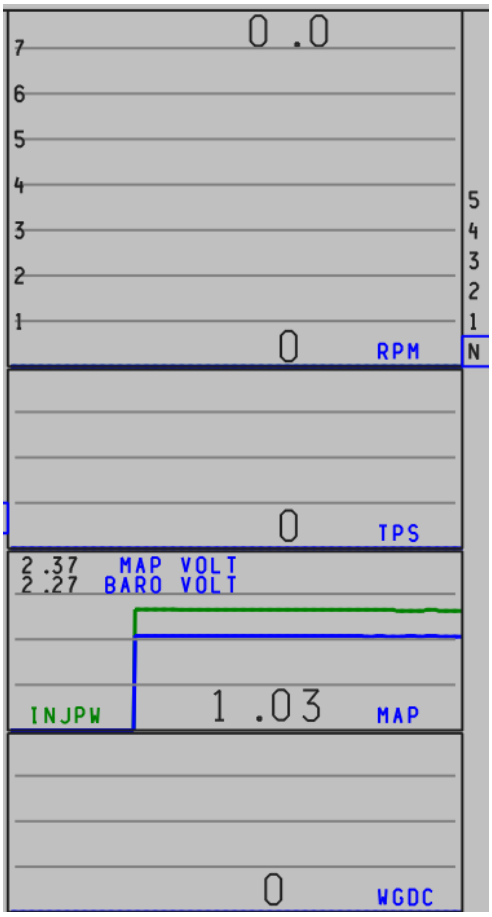


Table 1. Description of the lights in Panel 4

Light	Meaning	Information
DFCO	Deceleration Fuel Cut Off bit	You should see this turn red in case you are going rather fast and then go to a TPS value of 0
REVLIM		Turns red when the rev limiter becomes active
PWRE	Power Enrich	Will light when the command Air/Fuel ratio goes below 14.6
T1		A test bit which I used for further reverse engineering the protocol
CEL	Check Engine Light	
O2R	O2 rich	Will go red when the O2 sensor senses a rich condition, so it should blink periodically. It correlates with the O2 graph on the right going above the center axis
LEARN		Lights when contents of a BLM cell is updated

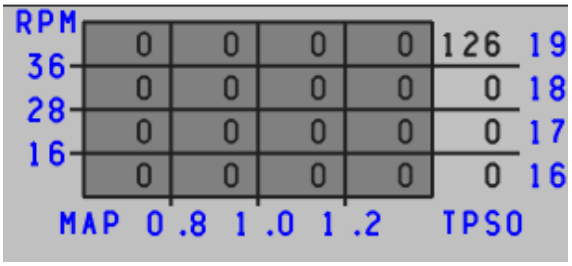
Panel 5



Panel 5 shows the instantaneous data and time history of RPM, Throttle Position Sensor (TPS), Injector Pulsewidth (INJPW), Manifold Air Pressure (MAP) and Wastegate Duty Cycle (WGDC). Also, the raw data (i.e. the voltage reported by the ADC for the MAP and BARO) is presented.

To the right of the RPM graph the computed position of the gear is depicted. This is estimated by using both reported speed and RPM.

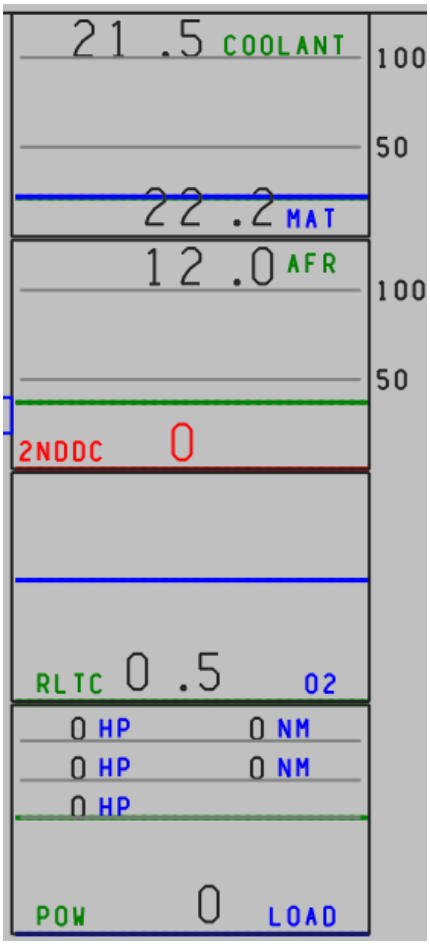
Panel 6



The contents of the BLM cells is depicted in matrix form. For the first 16 cells the horizontal axis depicts the MAP thresholds that determine the column while the vertical axis shows the RPM thresholds that determine the row.

When driving, the matrix is updated during the 'learn' cycle. The matrix is also populated when pressing the 'SAVE RAM' button.

Panel 7



The rightmost panel shows the instantaneous data and time history of coolant temperature, Manifold Air Temperature (MAT), Air Fuel Ratio (AFR), the secondary injector duty-cycle (2NDDC), the Rich-Lean Transition Counter (RLTC), the value reported by the Oxygen Sensor (O2) and the reported Engine Load (LOAD).

Based on an estimated acceleration, assumed total mass (vehicle + occupants) instantaneous power and torque are estimated and displayed. The highest values are maintained and depicted one line below the instantaneous values.

Panel 8

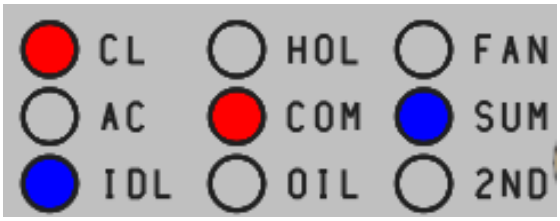


Table 2. Description of the lights in Panel 8

Light	Meaning	Information
CL	Closed loop status	Red when open, blue when closed
HOL	Hot Open Loop	In some GM ECU's Hot Open Loop, still need to figure out what the meaning in the Lotus ECU is, bit may be used for a different purpose, don't know
FAN		Blue when fan is on
AC		Blue when AC is on
COM		Alternate between red and off to indicate communication is going on with ECU
SUM		Is blue when checksum of ALDL datastream is ok
IDL		
OIL		Indicates when the relay is powered that enables the oil pressure light to be activated in case of low oil pressure (to prevent low oil pressure warning at low RPM)
2ND		secondary injectors active (correlates with graph on right, 2nd from top)

On the top of the display a total of 21 virtual buttons are depicted. These can be controlled using the mouse. By default, Espritmon is in MODE1. In this mode it asks the Esprit ECU for a datastream which contains the data needed to populate the inputs to the various numerical readouts and graphs.

Figure 4. Control panel

The control panel also allows selection of two other screens, for the ABS computer and for the SIR computer.

ABS screen

When pressing the ABS button on the control panel, the ABS screen is depicted and Espritmon starts to send requests for mode 1 data to the ABS computer.

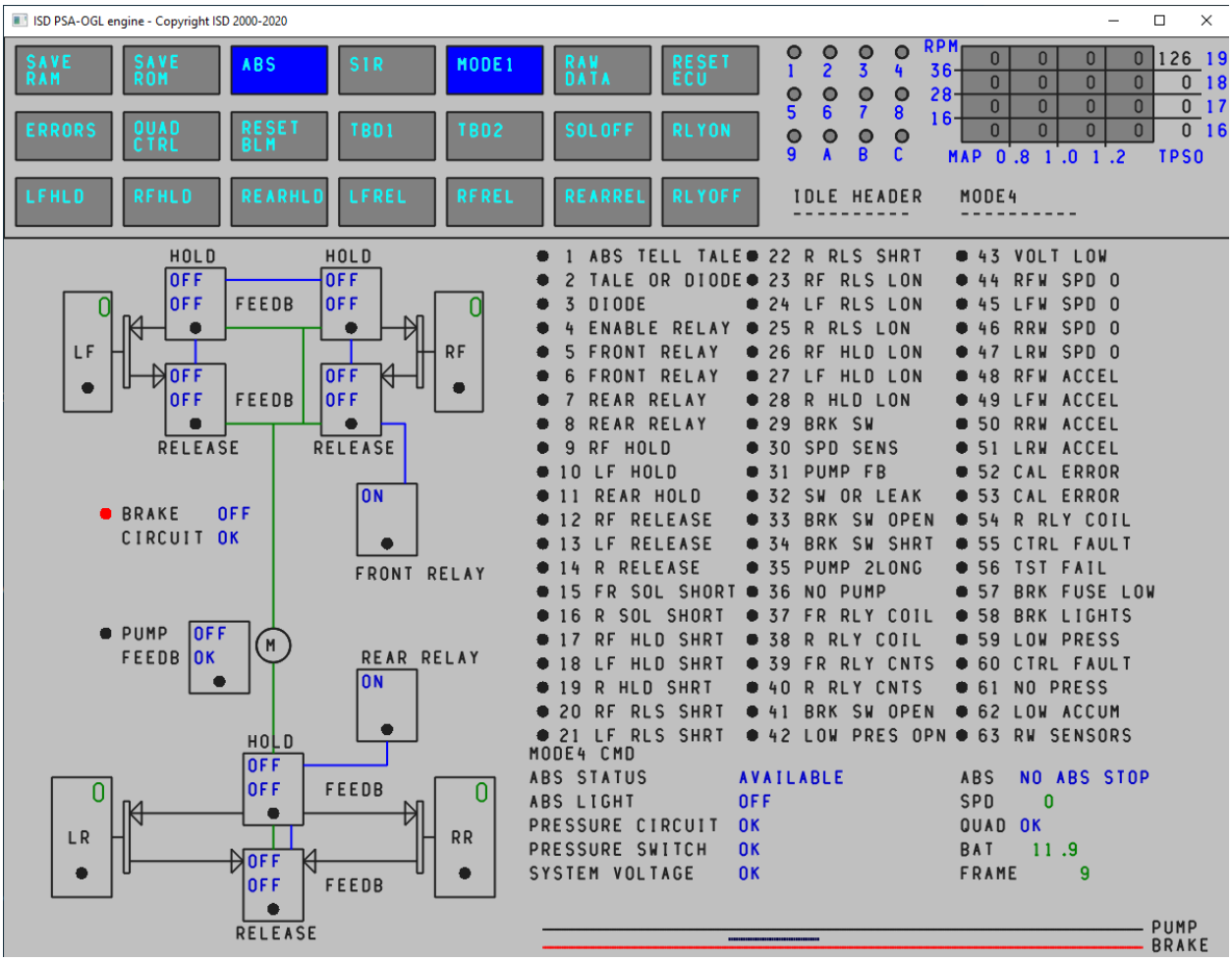


Figure 5. ABS screen

When the ABS screen is active, the buttons for control of the ABS relays and solenoids are enabled. Pressing these buttons will cause Espritmon to send a mode 2 command to the ABS computer to enable the particular solenoid and or enable/disable the relays.

Table 4 lists the functions of the buttons:

Table 3. Overview of the buttons to control ABS solenoids and relays

Button	Function
SOLOFF	Deactivate all solenoids
RLYON	Switch front and rear relays on
RLYOFF	Switch front and rear relays off
LFHLD	Activate left front hold solenoid, de-activate all other
RFHLD	Activate right front hold solenoid, de-activate all other
REARHLD	Activate rear hold solenoid, de-activate all other
LFREL	Activate left front release solenoid, de-activate all other
RFREL	Activate left front release solenoid, de-activate all other
REARREL	Activate rear release solenoid, de-activate all other

The information of the four wheel-speed sensors is plotted in the four wheels. Unless you make an ABS stop, the only other things you will see changing when driving are the brake (on or off) and the pump. When pressing the ABS button another time, the software goes back to the ECU screen. The ECU should commence communication in mode 1 within a few seconds.

At the right bottom of the display the pump and the brake signal are plotted. The leftmost values are those depicted in the schematic on the left (brake on/off and pump on/off).

SIR screen

When pressing the SIR button on the control panel, the SIR screen is depicted and Espritmon starts to send requests for mode 1 data to the SIR computer.

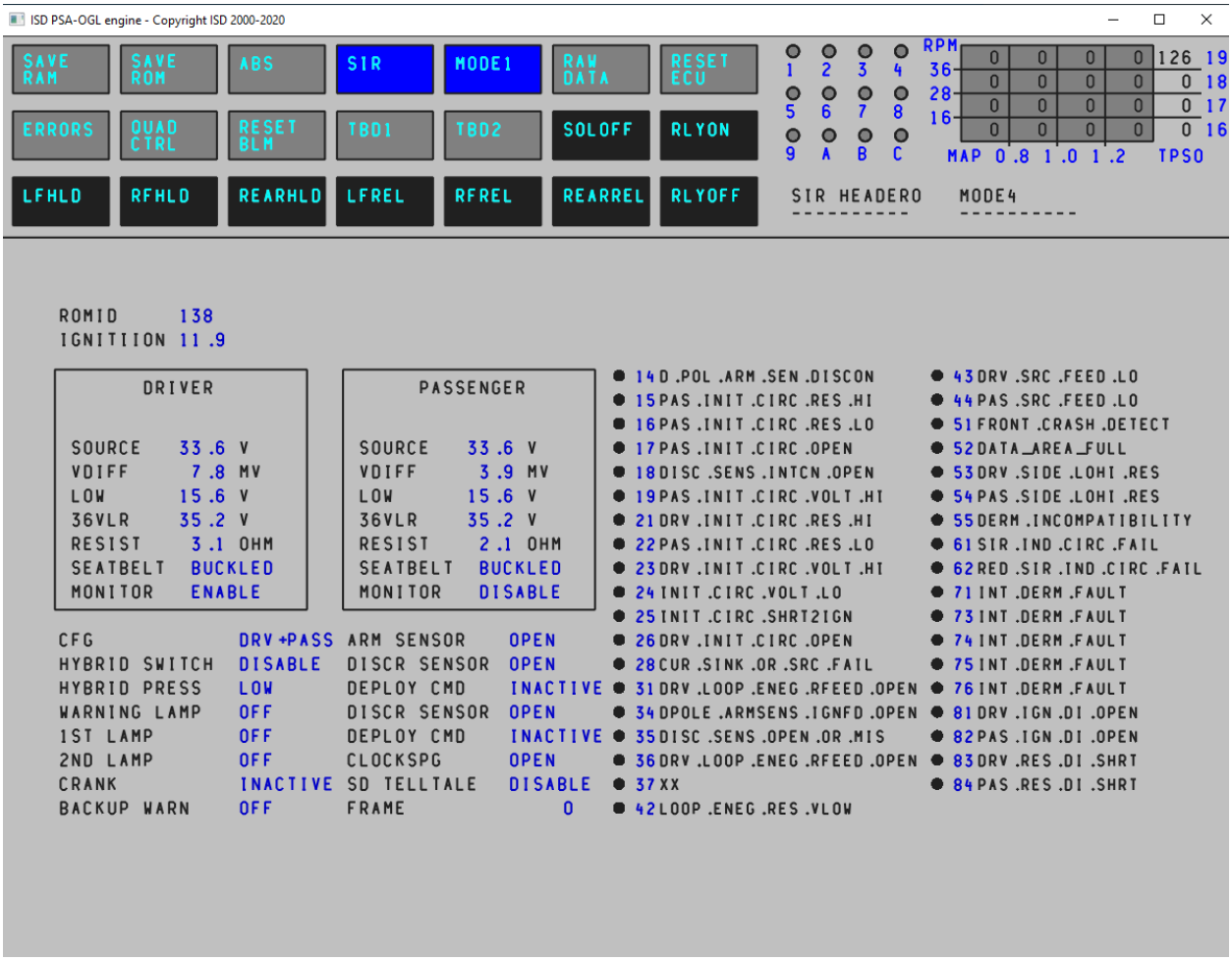


Figure 6. SIR screen

No other commands can be transmitted to the SIR computer by the current version of Espritmon.

Features in the main screen

Inspecting BLM values

When the SAVE RAM button is pressed, the contents of the first kB of RAM is saved as ‘espritram.bin’. Furthermore, the BLM values stored in RAM are depicted in the matrix at the top right of the display.

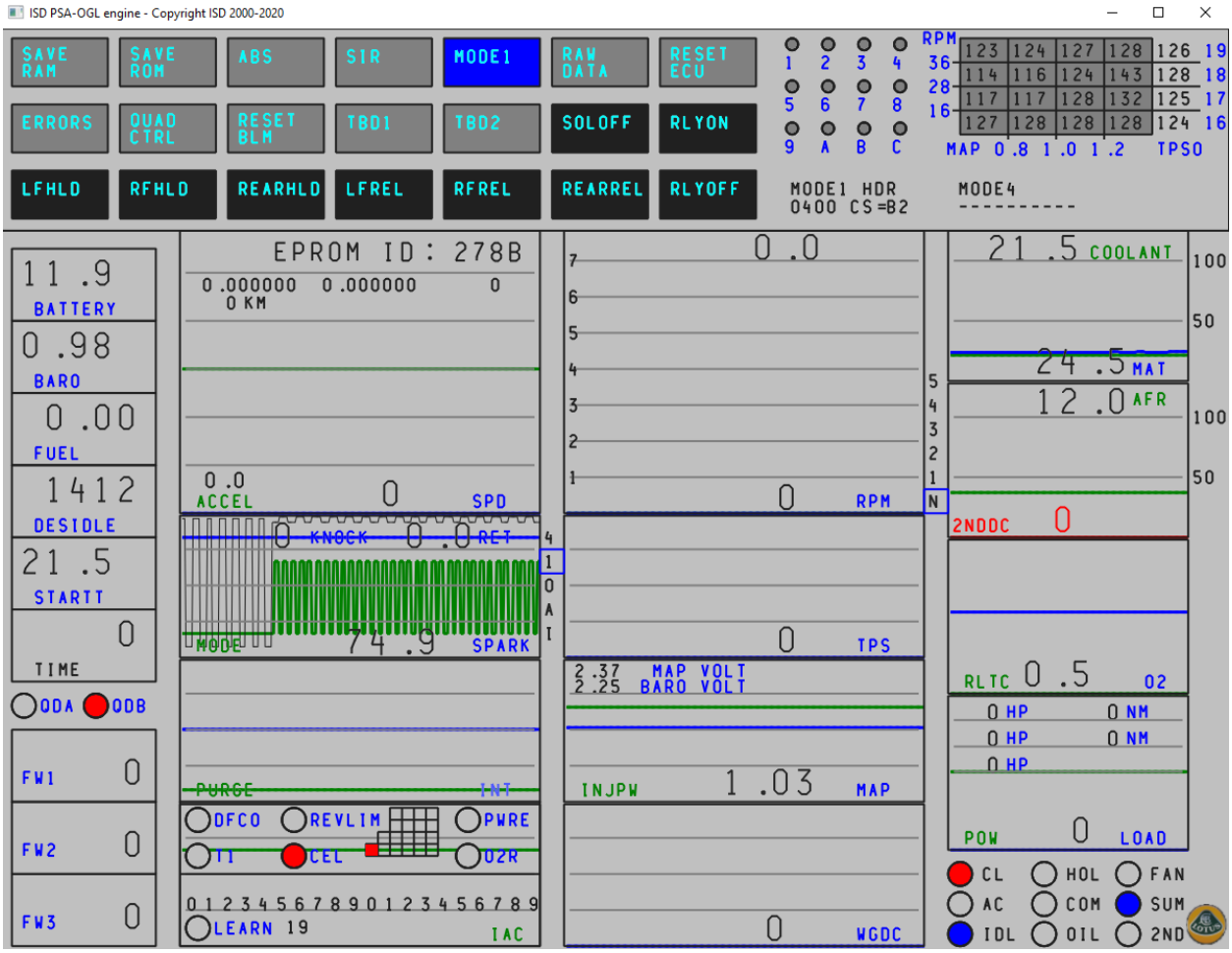


Figure 7. Main screen showing the BLM matrix populated using the stored values in RAM.

Inspecting the raw ALDL data

The raw ALDL data can be viewed by pressing the RAW DATA button. This will show the contents of the bytes in the ALDL datastream in a window at the bottom of the screen, e.g.:

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33
F0	96	01	27	8B	00	00	00	60	80	97	42	20	6B	04	F6	01	02	00	64	00	80	7B	13	00	E8	73	58	B7	00	61	DA	00	00
F0	96	01	27	8B	00	00	00	60	C0	97	42	24	6B	06	09	01	01	03	64	00	80	74	05	00	EB	73	64	D5	00	41	DA	00	00
34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64	65	66	67
23	28	88	42	19	00	C7	00	62	43	42	00	00	00	CD	67	1B	36	6C	09	00	11	18	7F	9D	41	0C	00	01	C6	10	10	20	00
23	28	88	38	1D	00	C7	00	62	43	42	00	00	00	ED	6B	1B	31	EB	09	00	10	18	7F	BD	41	0C	00	01	46	10	10	A0	F6

Saving the contents of the ROM to file

The contents of the ECU ROM can be saved to file by pressing the SAVE ROM button. The filename will be espritom.bin. The process will take close to 6 minutes, filesize should be 32768 bytes.

Error page

The fault word panel on the main page only shows the value of the three fault words contained in the mode 1 data.

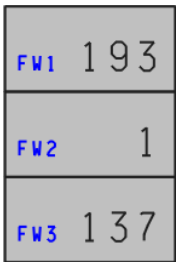


Figure 8. Indication of contents of fault words (technically fault bytes)

Espritmon can decode the various bits in these fault-words and provide more detailed information. The error page can be selected by pressing the ERRORS button:

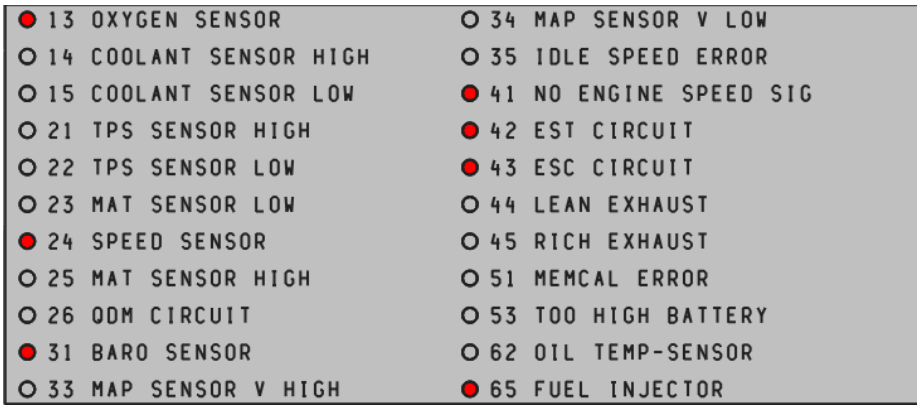


Figure 9. Error panel

The cause for Error 26 (QDM circuit) can be further tracked down using the information provided by the QDA and QDB lights on the main screen. Table 5 provides information on the relays and solenoids that are associated with QDM A and QDM B errors.

Table 4. QDM A and B error sources (source Lotus manual)

QDM A	QDM B
Radiator fan relay	RPM relay
Wastegate solenoid	AC control relay
Canister purge solenoid	Engine overheat relay
Check engine light	Throttle jack solenoid
	Secondary injectors

Quad Control panel

The Quad control panel will be displayed when the QUAD CTRL button is pushed.

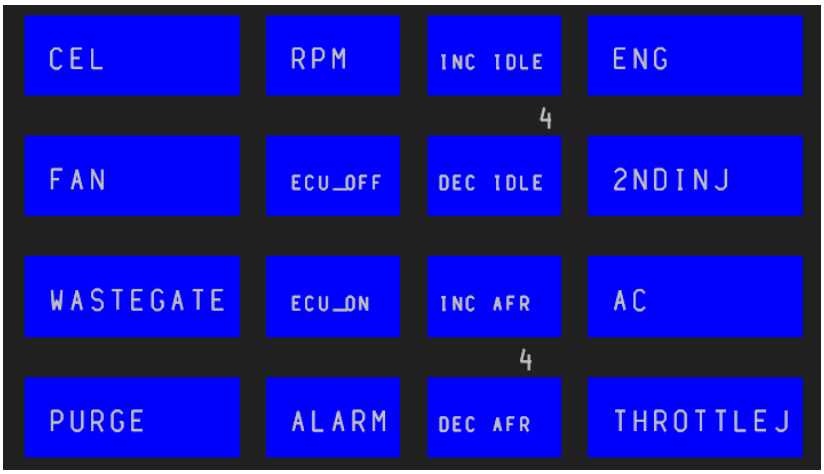


Figure 10. Quad Control Panel

In the Quad Control panel, the mouse can be used to click any of the dark blue buttons (which will turn red). The buttons listed in Table 6 can be of help in tracking down a code 26 error since they provide control over the quad drivers, enabling each solenoid/relay to be activated separately. The engine must be running to control the quad drivers (see also the Lotus manual EMH.4, page 55).

Table 5. Control of the QDM outputs

Button	**Function**
CEL	Switch on check engine light
FAN	Switch on cooling fan relay
WASTEGATE	Provide signal to wastegate solenoid
PURGE	Provide signal to canister purge solenoid
RPM	Switch on RPM relay
ENG	Switch on engine overheat relay (on SE and older S4)
2NDINJ	**!!! disconnect connectors and use test light !!!** engage secondary injectors
AC	Switch on airco relay
THROTTLEJ	Switch on throttle jack relay

The INC IDLE and DEC IDLE keys can be used to change the desired idle (this can also be done with the engine off). Likewise, the INC AFR and DEC AFR can be used to change the commanded air-fuel ratio. When a button is pressed to change it, the readout between the buttons shows the value sent to the ECU. By pressing the QUAD CTRL another time, the panel disappears and after a couple of seconds the software will resume mode 1 communications with the ECU. At that time, the commanded change in either AFR or desired idle should show up in the main screen. Before that time, the main screen will continue to show the previous values because it is not getting mode 1 updates.

Appendix A: Data used from the mode 1 message

After the mode 1 header (3 bytes long), a total of 65 bytes contain data. Espritmon 6.3 does not decode all data (partly due to a lack of information that describes the content of particular bytes). The overview below shows which bytes in the datastream currently are being decoded.

0	1	2	3	4	5	6	7	8	9
ID (hi)	ID (low)	FW1	FW2	FW3	Status1	Status2		Start T	

10	11	12	13	14	15	16	17	18	19
Desired idle	RPM hi	RPM low	speed			O2 sensor cnt	Rich-Lean tcount	Integr. count	Current BLM content

20	21	22	23	24	25	26	27	28	29
Current BLM cell		IAC	baro	MAP		TPS %			CP duty cycle

30	31	32	33	34	35	36	37	38	39
WG duty cycle			Battery V		Engine load	2^{nd} inj duty cycle		Spark adv. hi	Spark adv. low

40	41	42	43	44	45	46	47	48	49
T coolant	MAT	Knock counts	Knock retard	Injector PW hi	Injector PW low	AFR		Fuel hi	Fuel low

50	51	52	53	54	55	56	57	58	59
Status bits	Time hi	Time low	Status bits					Status bits	

60	61	62	63	64
	Status bits	Status bits	Status bits	check

Appendix B: DIY ALDL interface

The interface between the OBD-I side and the RS-232 can be constructed using only a few parts. Various schematics can be found on the internet. I found that the one designed by Carsten Meyer (depicted in Figure 1) works very well. Total cost of the components is estimated below 10 Euro's. Note that on different versions of the MAX232 IC also different values of the capacitors are required.

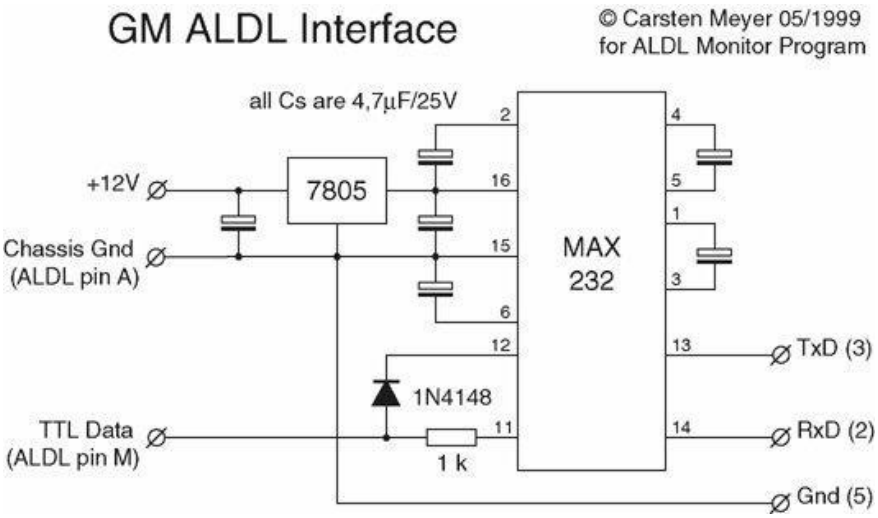


Figure 11. Schematic of interface to convert bi-directional TTL-level data to RS232 interface.

Note that the ALDL line in this schematic refers to pin M, but on the Esprit the 10 pin Opel connector is used, and the TTL data is on pin G.

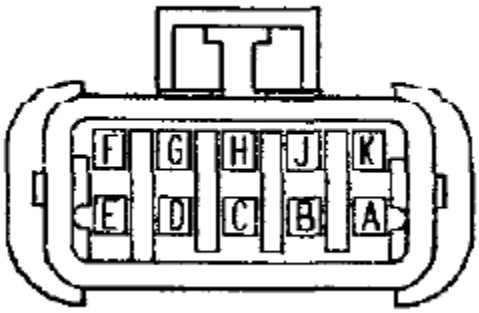


Figure 12. ALDL connector in the Esprit

When using Espritmon to communicate with the ABS or SIR computer the front ALDL connector needs to be used. Dataline G of the rear connector is only connected to the ECU.

Table 1 provides an overview of the parts and typical cost (e.g. at Conrad). The only part that is not on the list is the male ALDL connector. With some skills the required three pins can be taken from a male D-connector and positioned at the right location, for inspiration see e.g. the pictures in Appendix B.

Table 6. Part-list of GM ALDL interface

Component	amount	Typical total cost (Euro)
7805	1	0.52
MAX232	1	2.42
capacitor	6	0.60
Diode 1N4148	1	0.06
Resistor 1K	1	0.11
Small circuit board	1	2.90
9 pin D connector, female	1	0.26
Total		**6.87**

In case a USB to serial cable is needed, there are various manufacturers to choose from. One issue that makes it difficult to use some of these cables is that under Windows 10 support for various of the Prolific PL2303 chipsets that are often present in such a cable is discontinued. Previous versions of Windows still seem to support cables using this chipset. For Windows 10 USB to serial cables using the FTDI chipset seem to work well.

Another issue with some chipsets is that the associated drivers do not allow the non-standard data-rate of 8192 bits/second to be set. For some of the Prolific drivers a change to some registry entries associated with the timer constants can be used to resolve this, but the better alternative is to use a cable with an FTDI chipset, they seem to work out of the box.

With some creativity you can construct your own ALDL connector. Below is an example.

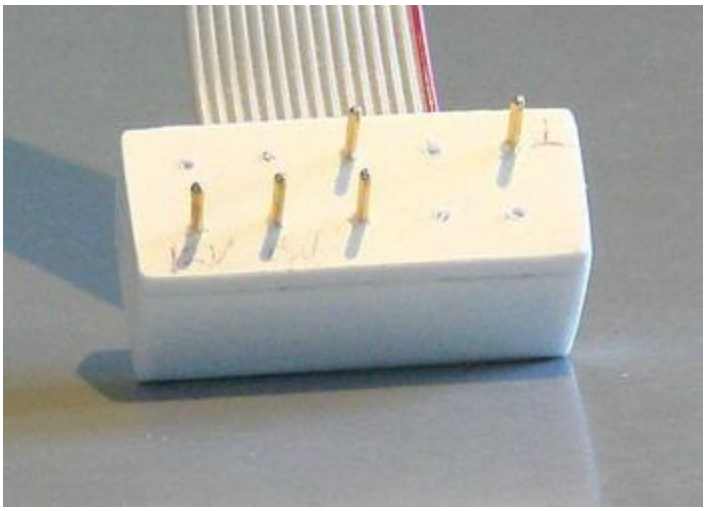


Figure 13. DIY ALDL connector using pins from a D-connector