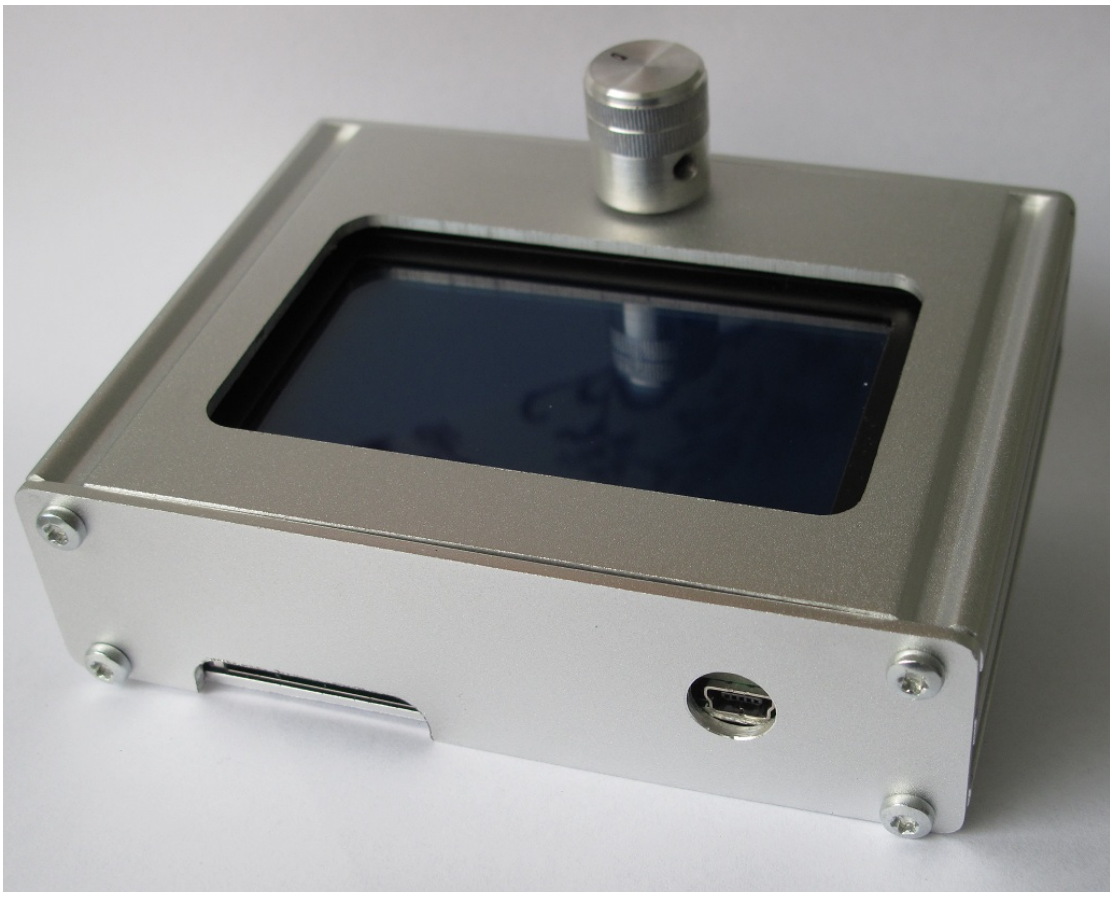
ALDL Logger V2



V1.13.2

Contents

V1.13.2

Overview

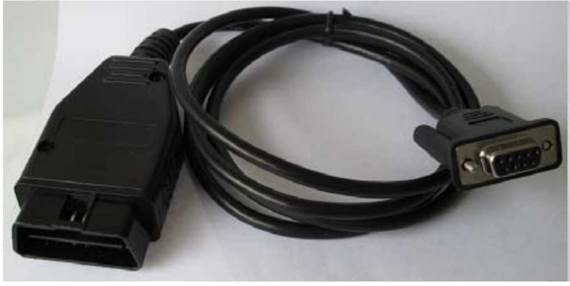
The ALDL logger provides data logging for any 8192 baud ALDL equipped vehicle. The ‘808 VN/VP ECU can be modified to also allow logging by installing a RS232 communications board.

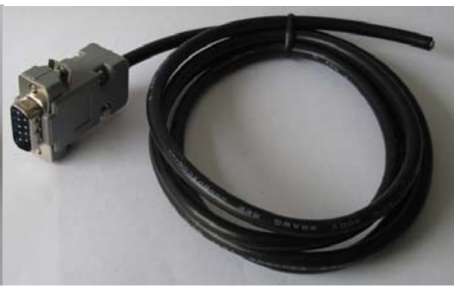
The logger is configured using TunerPro V5 compatible ADX files and logs directly to XDL files for viewing in TunerPro. This gives the greatest ease of configuration and options while keeping all files in a known format for analysis.

Features:

- ALDL and RS232 data interface
- Partial and full pass-through modes to allow the logger to be used as a USB ALDL PC interface
- Send commands to ALDL devices
- Fully configurable using TunerPro ADX files
- Listen logging for VT-VY dash data recording
- SD memory card for recording ALDL data directly to TunerPro XDL files
- Graphic LCD to show live data and flags
- 4 analog and 2 digital inputs recorded to the ALDL stream
- Read and write cal files to NVRAM enabled ECUs
- Read calibration of VX/VY Flash PCMs
- Real-time clock for time stamped files
- User upgradable firmware

Supplied accessories:





V1.13.2

Connectors and Wiring

The logger is provided with three cables

- USB for connecting to a PC. It does not allow connection to a USB comms equipped ECU. This cable can be used to program the real-time clock, log the analog inputs of the logger to TP, upgrade the firmware and provide pass through logging of the vehicle.

- DB9 to ALDL cable. This cable is used to connect the logger to 8192 baud ALDL vehicles. It does not convert the ALDL signal to RS232 so cannot be used for connecting a laptop serial port direct to the vehicle. It must only be used with the logger.
  The DB9 male connector on the logger provides power and data. A standard DB9-DB9 serial cable (not supplied) is used for ‘808 connection.

- DB9 analog and digital connection cable. This cable allows for 4 analog and 2 digital inputs. This data can inserted in an ALDL data stream and recorded to a XDL file. The cable also provides +5V that can be used to power sensors, a maximum of 100mA can be drawn from this output when the logger is powered from +12V (VR and later). For VN/VP ‘808’s instead use the 5V sensor wire that feeds the TPS and MAP sensor but beware this also can only provide less than 100mA.

Logger Connector Pinouts

DB9 Male Pinout

1 - No Connection
2 - RS232 Rx
3 - RS232 Tx
4 - ALDL data

5 - Ground
6 - +12V (Operating voltage 8-16V)
7 - Ground
8 - +5V
9 - +5V

DB9 Female Pinout

1 - Analog Input 2 (Brown)
2 - Analog Input 3 (Blue)
3 - Analog Input 4 (Green)
4 - Digital Input 2 (Purple or White)
5 - Digital Input 1 (Yellow)
6 - Analog Input 1 (Orange)
7 - +5V (Red)
8 - Ground (Black)
9 - No Connection

DB9 Female Pinout (2014 Cable Colours)

1 - Analog Input 2 (Red)
2 – Analog Input 3 (Orange)
3 – Analog Input 4 (Green)
4 – Digital Input 2 (Yellow)
5 – Digital Input 1 (Black)
6 – Analog Input 1 (Purple)
7 – +5V (White)
8 – Ground (Grey)
9 – Fault Output (Blue)

Initial Setup Configuration

The ALDL logger uses TunerPro ADX files for configuration. While these are compatible with TP (and must be used to view the logged XDL file) there are some important differences. There are example ADX files to get you started in the logger support thread at www.pcmhacking.net however the key points are

- Command names – these must be identical to the example ADX files and are case sensitive.
- The logger can only log up to 5 messages at a time.
- Listen logging is supported for logging the data going to the dash in VT-VY without stopping any chatter. The logger aux inputs can also be recorded.
- There are 4 bitmask items. They configure the logger data interface, how it manages the auxiliary inputs, the calibration start/end address and an experimental passive logging mode (for VT and later it is possible to use the

silence on the bus to log at ~3Hz while attempting to retain all vehicle functions such as the dash).

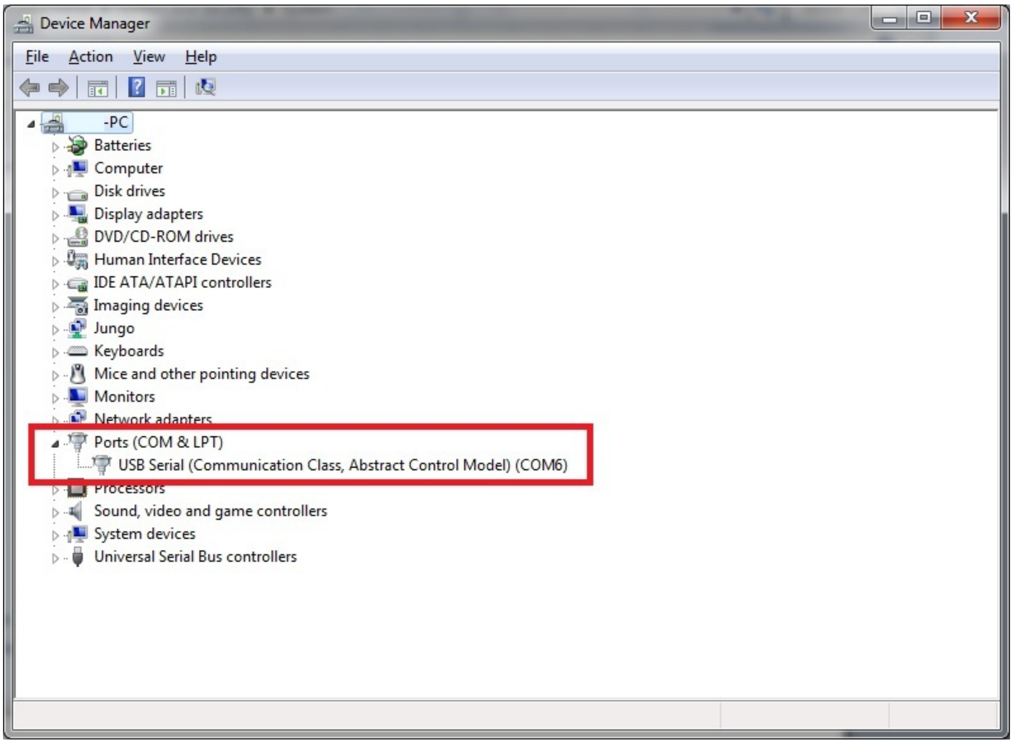
- The logger can display 16 data items and 8 flag items on the LCD. They can be any item in the data stream and are selected by entering the numbers 1-16 and 1-8 in the description field of the ADX items and flag items.
- All equations for LCD displayed items must use X for the data item, only basic math functions (addition, subtraction, multiplication, divide) are supported and a space between each term is required. There is no BIMDAS logic so how the equation is written is how it is processed (no parenthesis). **Always** simplify the equation as far as possible, the logger processor is not capable of processing complex equations quickly, simplifying leaves it free to process ALDL frames faster. Linked items are not supported. This only applies to LCD displayed items, if you do not want to display the item on the LCD then the equation can be written in any format that TP supports. Bear in mind that even if the equation is simplified and you lose some accuracy, the underlying logged data is unchanged i.e. you can still display it in TP on the PC in full resolution and accuracy or add linked items.

The logger supports most SD and SDHC memory cards. You do not require a large capacity card, even a 128MB card will log nearly 50 hours in most applications. The SD card must be formatted to FAT32 prior to use in the logger, once this is done load your ADX files to the card. Multiple ADX files can be loaded to the card and selected using the logger user interface.

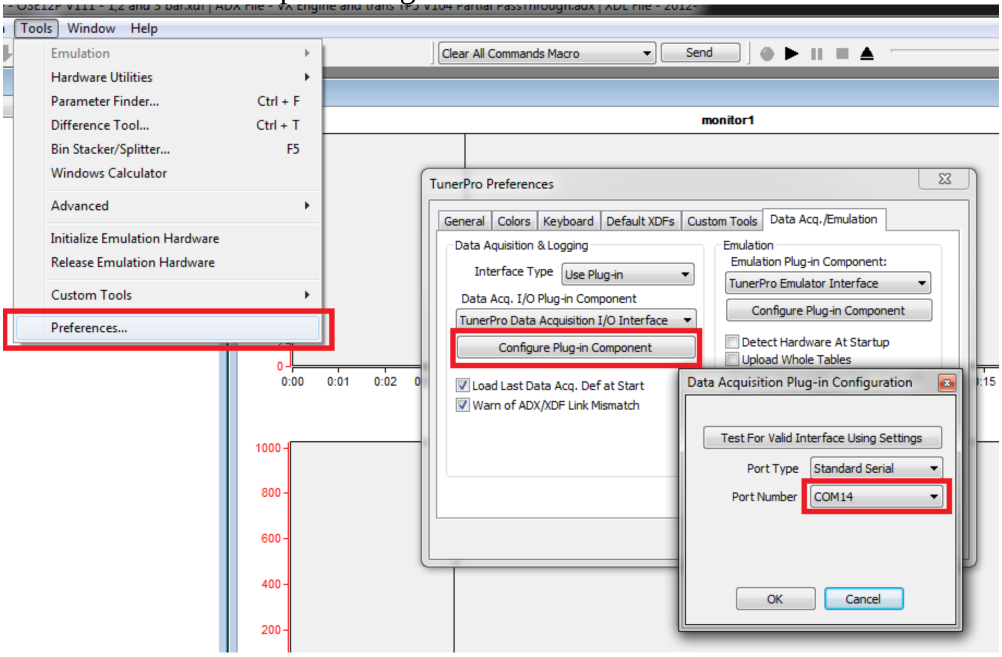
The logger does not support folders, all files must be placed in the root directory. Recorded XDL and cal files will also be placed in the root directory.

USB Virtual Com Port Setup

The USB port on the logger is not by default detected as a virtual serial port by windows. To allow windows to use the USB port as a serial port the included program serial_install.exe needs to be run. After running this program the Windows device manager will show the virtual comm. port assigned to the logger.



For any communications requiring the logger to be used by the PC as a virtual serial port (logging the logger, setting the clock etc) use this assignment. In this case TunerPro would need the port changed from com14 to com6.


Tools Window Help
Emulation
Hardware Utilities
Parameter Finder... Ctrl + F
Difference Tool... Ctrl + T
Bin Stacker/Splitter... F5
Windows Calculator
Advanced
Initialize Emulation Hardware
Release Emulation Hardware
Custom Tools
Preferences...
Clear All Commands Macro
Send
monitor1
TunerPro Preferences
General Colors Keyboard Default XDFs Custom Tools Data Acq./Emulation
Configure Plug-in Component
Configure Plug-in Component
Data Acquisition Plug-in Configuration
Test For Valid Interface Using Settings
OK Cancel
0:00 0:01 0:02
1000
800
600
400
200

Data Aquisition & Logging
Interface Type: Use Plug-in
Data Acq. I/O Plug-in Component: TunerPro Data Acquisition I/O Interface
☑ Load Last Data Acq. Def at Start
☑ Warn of ADX/XDF Link Mismatch

Emulation
Emulation Plug-in Component: TunerPro Emulator Interface
□ Detect Hardware At Startup
□ Upload Whole Tables

Port Type: Standard Serial
Port Number: COM14

Logging the Logger

Once the virtual com port is setup a PC can request data from the ALDL logger. The supplied ADX (LoggerV2.ADX) uses Mode 1 Message 0 to get the aux input data for use as a standalone data logger. Partial pass-through mode can request data to allow logging the logger and a vehicle at the same. Both are described in more detail in the

Aux input section for logging a wideband. Other modes described below are generally only for specialised applications.

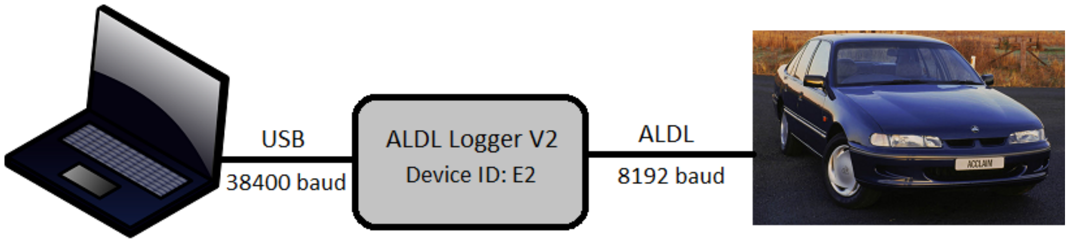
Supported ALDL modes of the logger (device ID 0xE2):
Mode 1 – Request aux data from the logger.
Message 0 – All 6 aux channels (26 byte payload)
Message 1 – Analog 1 and Analog 2 (4 byte payload)
Message 2 – Analog 3 and Analog 4 (4 byte payload)
Message 3 – 4 analog channels (8 byte payload)
Message 4 – Digital 1 (9 byte payload)
Message 5 – Digital 2 (9 byte payload)
Message 6 – Digital 1 and Digital 2 (18 byte payload)

Mode 2 – Request logger firmware version.
Address 0xFF80

Mode 22 – Request ALDL modes
Byte 3
Function 0 – Return to normal
Function 1 – Reserved
Function 2 – Full passthrough (cannot return to normal over ALDL)
Function 3 – Partial Pass-Through
Byte 4
Function 0 – PassThrough Bus Type ‘808
Function 1 – PassThrough Bus Type ALDL

Pass-Through ALDL Modes

The logger can be used as a USB to ALDL interface using a PC and the supplied cables. There is no echo like normally found on USB interfaces, and there is a baud rate conversion from 38400 to 8192. The PC software must be setup to 38400 baud and no echo.



There are two modes of operation, full or partial pass-through. The operation of each is as follows:

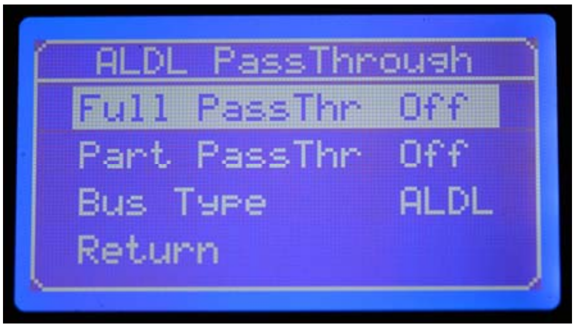
Partial Pass-Through Mode

This mode analyses all data entering the USB side of the logger. If any data frame is directed at the logger (ALDL device ID E2) it processes the frame and returns the requested data. All data incoming on the USB side is still output to the ALDL bus, this ensures the vehicle bus remains under control of the PC once silenced and also allows bus devices to respond if the request is not destined for the logger. As no device on the bus should have the ID E2, only the logger will respond to the PC's request. Any data on the ALDL side is output to the USB with no processing. This mode is used to data log a vehicle while at the same time requesting the aux input data of the logger.

Full Pass-Through Mode

All data on the USB side is sent through to the ALDL side and all data on the ALDL side is sent to the USB side. The logger will not respond to frames with device ID E2. This mode may be used to log a device on the bus with ID E2 or to ensure the logger does not process anything during critical operations such as when writing a calibration or bin.

The pass-through modes can be accessed from the ALDL PassThrough menu on the LCD…



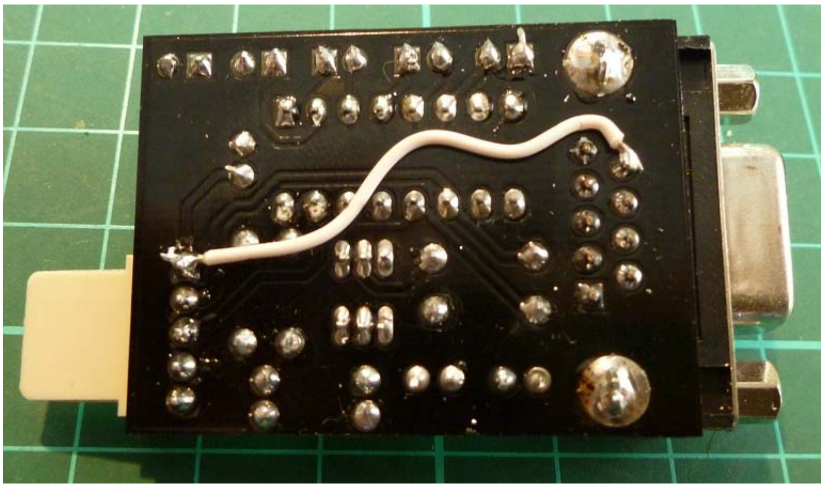
ALDL PassThrough
Full PassThr Off
Part PassThr Off
Bus Type ALDL
Return

Passthrough can also be selected via ALDL commands sent over the USB interface. If full passthrough is selected using the USB interface it is only possible to return to normal using the LCD interface. Detailed information on setting up an ADX for partial pass-through logging is described in the section "Using Pass-Through Mode".

ECU and PCM Connection

Connection to the vehicle differs between models. For VR and later vehicles with a data connector under the steering column use the DB9 to OBD cable supplied. It provides permanent power to the logger along with the data connection.

For VN and VP a RS232 comms interface is the simplest method of connection. If using a www.delcohacking.net '808 comms board a small modification needs to be made to power the logger…

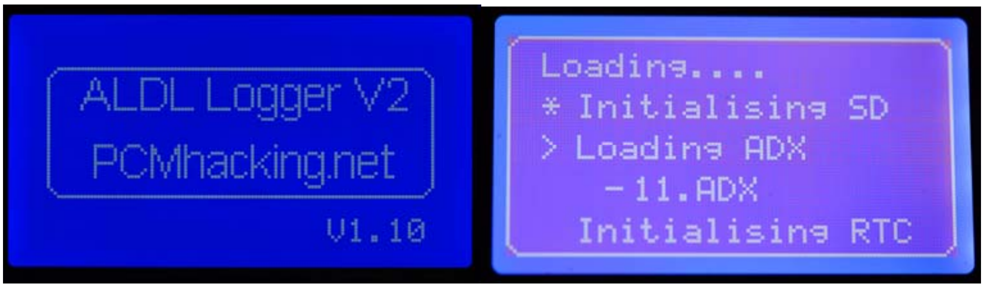


When connecting the logger to the ‘808 **ALWAYS** ensure the key is in the off position. Once the serial cable and logger are connected the key can be turned on and the engine started like normal.

The ADX flag item “Bus Type” specifies which type of connection is being used. For the ‘808 you should select 0 and for VR and later 1 for a one wire connection.

Power Up

When first powering up the logger you will be presented with screens similar to these while the logger boots…

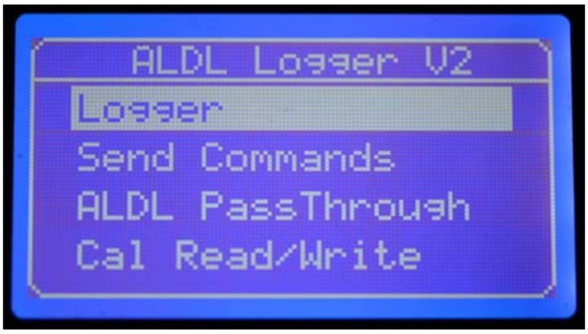


ALDL Logger V2
PCMhacking.net
V1.10

Loading....
* Initialising SD
> Loading ADX
  - 11.ADX
  Initialising RTC

Boot Screens

The logger uses a single rotary encoder and integrated button to operate the user interface. From the main screen you can choose the logger function, Command Sending, PassThrough Modes, ADX load/save, calibration read/write and setup specific to the logger itself. The display button moves through the screens of the user interface while the select button executes the selected item.

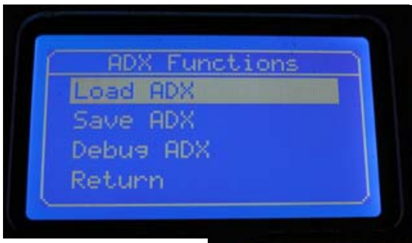


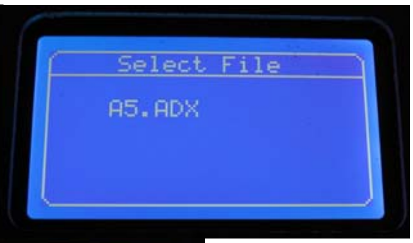
Main Screen

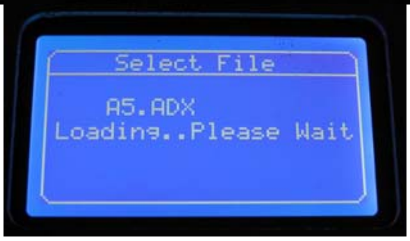
ADX Load and Save

The logger has internal EEPROM memory that is used to keep a copy of the ADX contents and filename. This is done so the logger can be used as a scan tool without a SD card. At boot the internal memory is loaded to get the filename of the last saved ADX, if the filename is found on the SD card it loads the file otherwise it will use the items from the internal memory.

To load an ADX, firstly select ADX Load/Save from the main screen followed by Load ADX. Using the rotary selection scroll through all the ADX files on the SD card until the desired file is found, pressing the encoder selects the file. To exit without loading an ADX hold down the button until the logger returns to the previous screen.

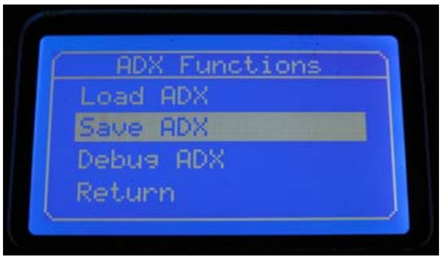






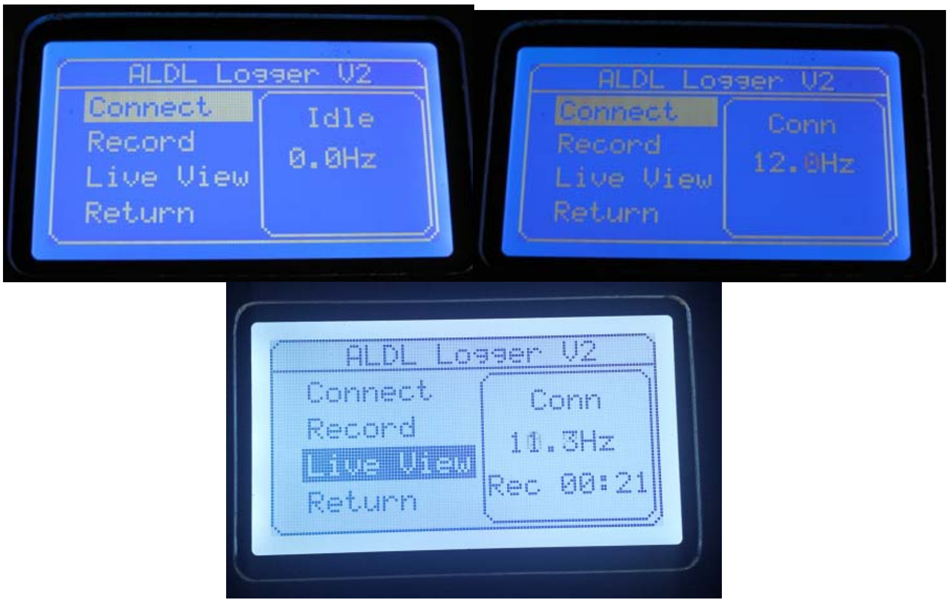
A5.ADX
Loading..Please Wait

Selecting Save ADX stores a copy of the currently loaded ADX to the internal EEPROM. This is required if you would like the current ADX automatically loaded from the SD card at next power on.



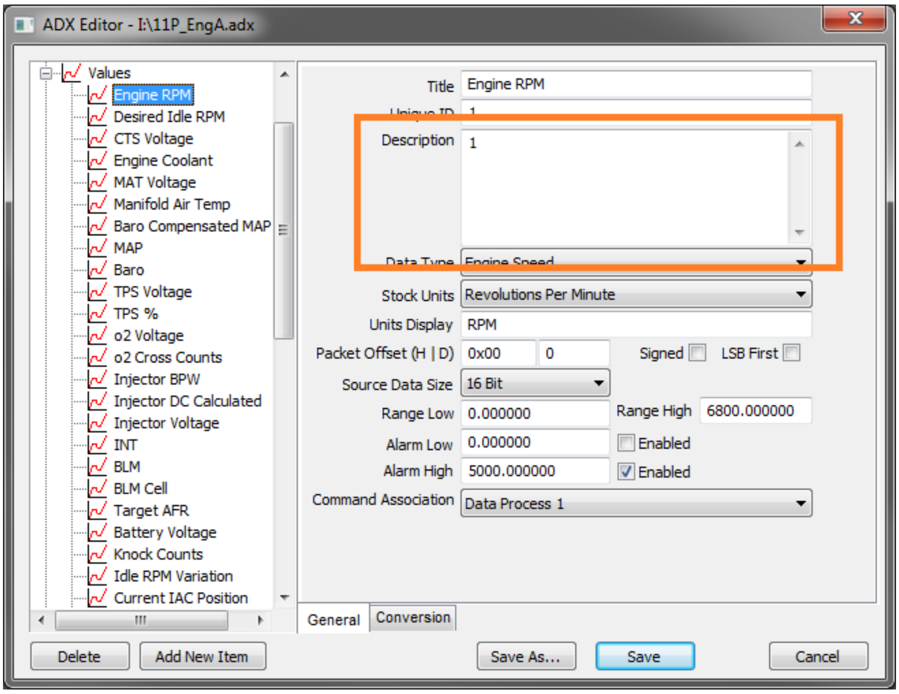
Logging

From the logger menu selecting ‘Connect’ begins logging. The status box shows the current logging conditions. The logging speed is shown in Hz along with if the logger is idle or connected and if recording to the SD card. Clicking connect again disconnects.



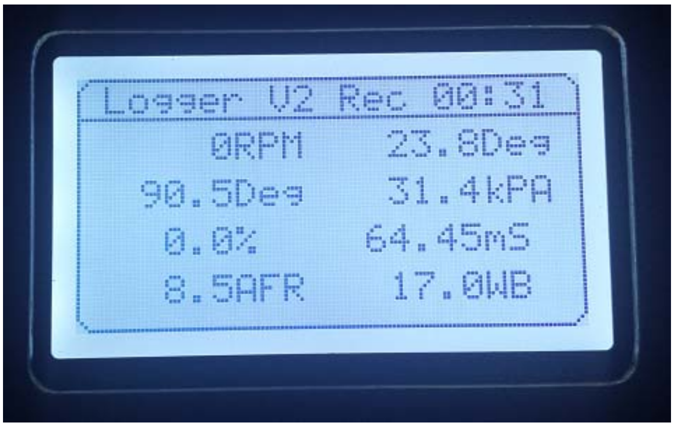
Live View

The item display is split between two screens of 8 items each. These items are specified in the ADX with 1-16 placed in the description field. The same item can be displayed on both screens by defining the same item twice in the ADX and assigning say 1 and 9 to each. The Units display defines the 3 characters displayed in the live view screen. Please note the Initial Setup Configuration section regarding simplifying the equation/conversion.


ADX Editor - I:\11P_EngA.adx

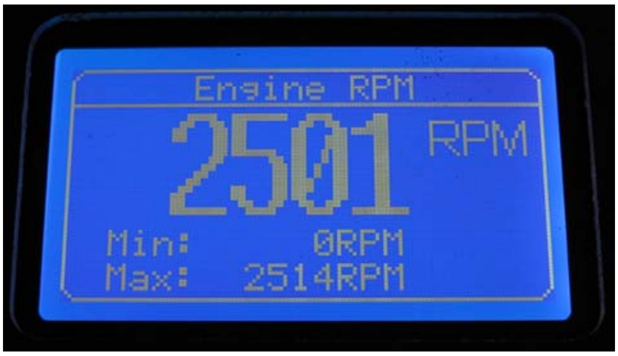
Title: Engine RPM
Unique ID: 1
Description: 1
Data Type: Engine Speed
Stock Units: Revolutions Per Minute
Units Display: RPM
Packet Offset (H | D): 0x00 | 0
Signed □ LSB First □
Source Data Size: 16 Bit
Range Low: 0.000000 Range High: 6800.000000
Alarm Low: 0.000000 □ Enabled
Alarm High: 5000.000000 ☑ Enabled
Command Association: Data Process 1

The following display example shows 8 items defined. If a position is not used the item will show ‘–‘. The units field (such as RPM below) in the ADX can support 3 characters.



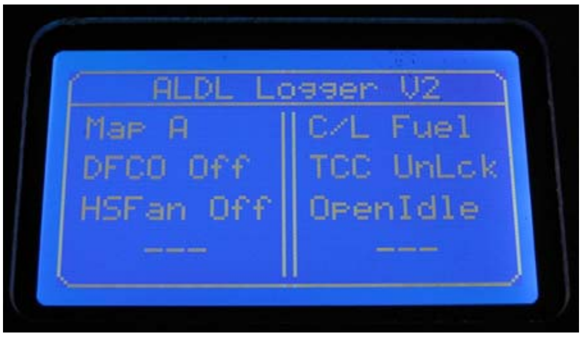
```
 Logger V2 Rec 00:31
      0RPM     23.8Deg
   90.5Deg     31.4kPA
     0.0%     64.45mS
    8.5AFR     17.0WB
```

The big digits display shows the same 16 selected items but only one item at a time. The max and min values are not saved and are cleared when disconnecting. From either of the 2 live item display screens rotate the control dial to scroll between the 8 items from the current display page. The page header (Engine RPM below) can support 19 characters.



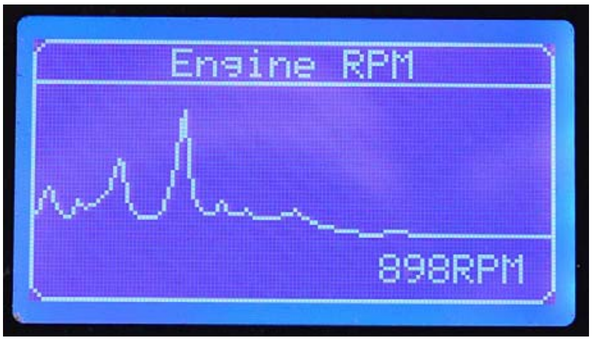
```
     Engine RPM
   2501 RPM
Min:       0RPM
Max:    2514RPM
```

The flag display screen can show 8 flags, the same setup applies except using only 1-8 in the description field of the flag item. The following screen shot shows 6 defined flags and two undefined resulting in ‘---‘. The flag item is able to have 9 characters when configuring the true/false fields in the ADX.



```
    ALDL Logger V2
Map A     || C/L Fuel
DFCO Off  || TCC UnLck
HSFan Off || OpenIdle
   ---    ||    ---
```

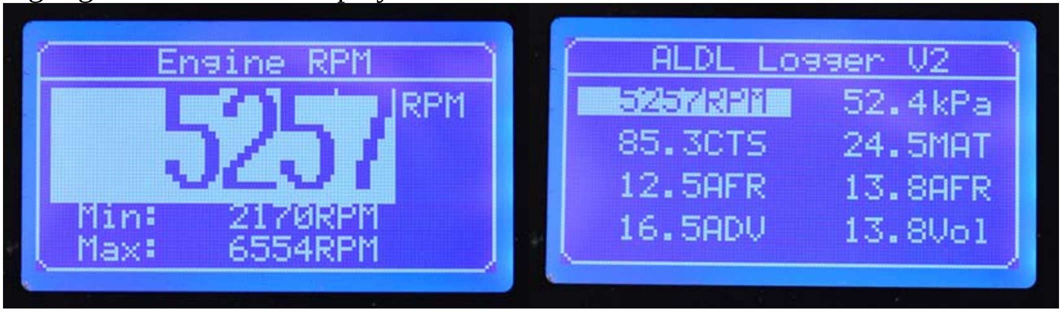
The chart display can show all 16 live view items, by rotating the control dial each item can be viewed.



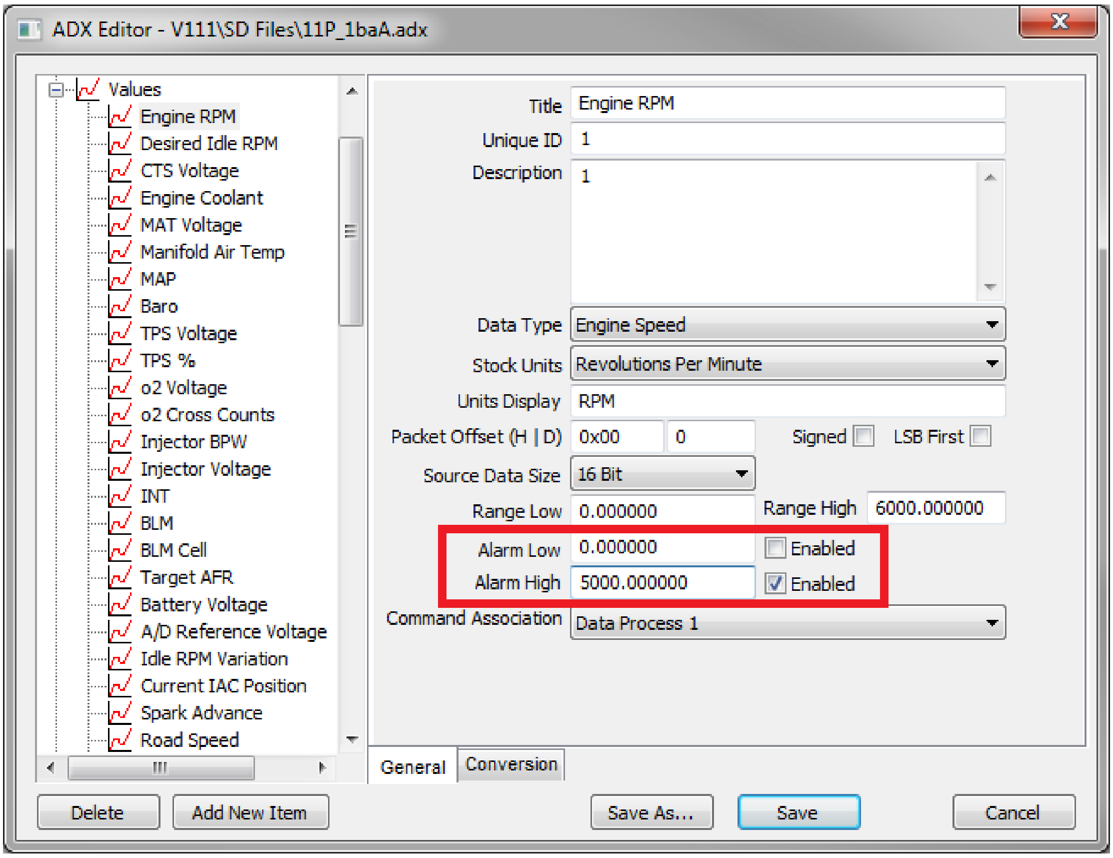
Alarms and Fault Output

While logging the logger can provide alarms for the 16 values being displayed by the logger. On V2.0 hardware there is also a fault output, active low when an alarm is present. The Buzzer option can also be used to sound an audible alarm.

When an alarm occurs the display will jump to the big digits item, it will also be highlighted on the live displays…



To enable the alarm open TunerPro and edit the ADX. The alarm low and alarm high thresholds are set to the value along with the enabled tick box if you want the alarm used. The following example would have no low RPM alarm and 5000RPM for the high alarm.



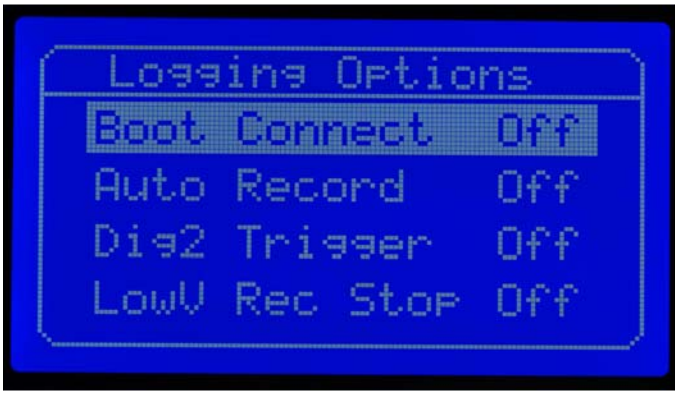
ADX Editor - V111\SD Files\11P_1baA.adx

- Values
  - Engine RPM
  - Desired Idle RPM
  - CTS Voltage
  - Engine Coolant
  - MAT Voltage
  - Manifold Air Temp
  - MAP
  - Baro
  - TPS Voltage
  - TPS %
  - o2 Voltage
  - o2 Cross Counts
  - Injector BPW
  - Injector Voltage
  - INT
  - BLM
  - BLM Cell
  - Target AFR
  - Battery Voltage
  - A/D Reference Voltage
  - Idle RPM Variation
  - Current IAC Position
  - Spark Advance
  - Road Speed

Title: Engine RPM
Unique ID: 1
Description: 1
Data Type: Engine Speed
Stock Units: Revolutions Per Minute
Units Display: RPM
Packet Offset (H | D): 0x00 | 0 Signed □ LSB First □
Source Data Size: 16 Bit
Range Low: 0.000000 Range High: 6000.000000
Alarm Low: 0.000000 □ Enabled
Alarm High: 5000.000000 ☑ Enabled
Command Association: Data Process 1

Logging Options

From the main menu select “Logging Options” to display the following screen…



Logging Options
Boot Connect Off
Auto Record Off
Dig2 Trigger Off
LowV Rec Stop Off

Auto Logging and Recording

The logger can be set to automatically connect and begin recording when it is powered up.

Setting Boot Connect the logger will attempt to connect as it is powered. Setting Auto Record it will also attempt to start recording.

The logger will display the logging screen instead of the main display once connected.

Setting Auto Record the logger will begin recording whenever connected to the ECU, this provides a quick way to begin recording by just selecting Connect from the logger menu.

External Recording Trigger

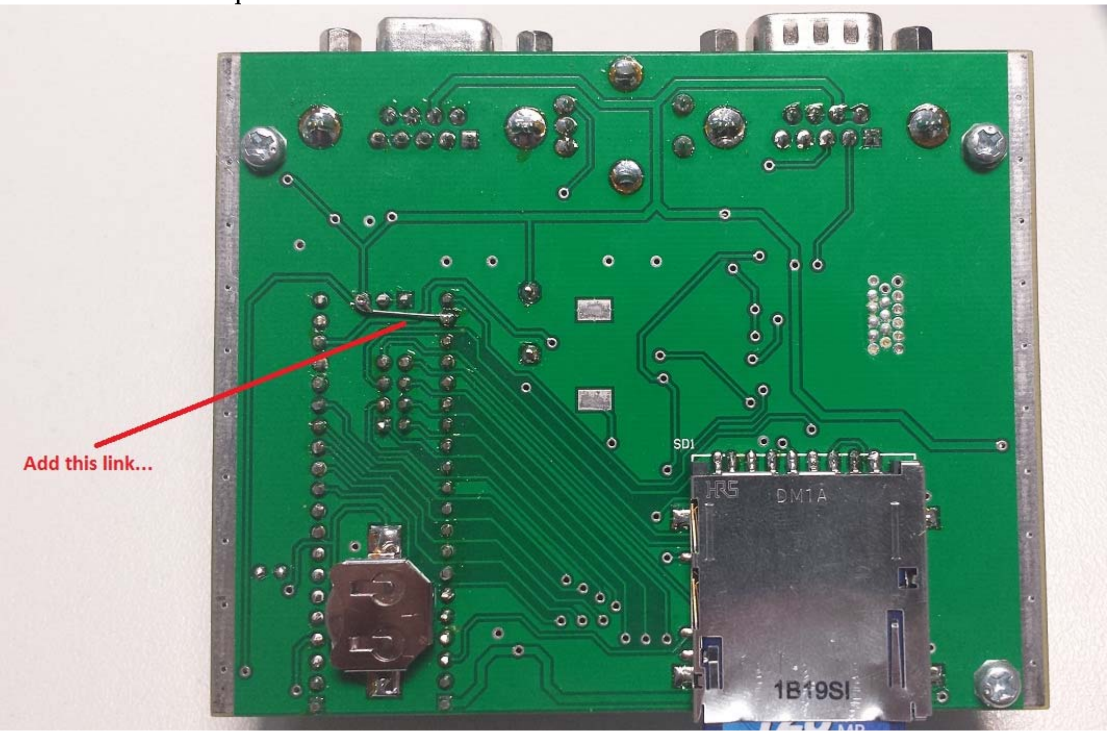
The option Dig2 Trigger enables digital 2 input as a recording trigger instead of digital input. Grounding the input will automatically connect and begin recording. Releasing the input will stop recording.

Low Voltage Recording Stop (‘808 only)

The option LowV Rec Stop attempts to stop recording and save the XDL file when the system voltage falls below 4.50 volts. On ‘808 connections when the ign key is turned off the logger loses power after approx 2 seconds, if recording is active the data log XDL file will either be corrupt or contain no data. If the logger can react fast enough to the system voltage falling the recording will be finalised so TunerPro can read the file. There is a small speed reduction in logging with this option enabled, rather than buffering ALDL data and writing large amounts of data to the SD card in one hit the logger will write every second data frame to ensure when power is removed it can finalise the file as fast as possible.

Note: This must only be used on ‘808 serial connections, not 12V ALDL connections.

On hardware rev 1 the logger is unable to measure the internal battery voltage without a small modification to the circuit board. Rev 2 (shipped with FW 1.10 or later) no modification is required.

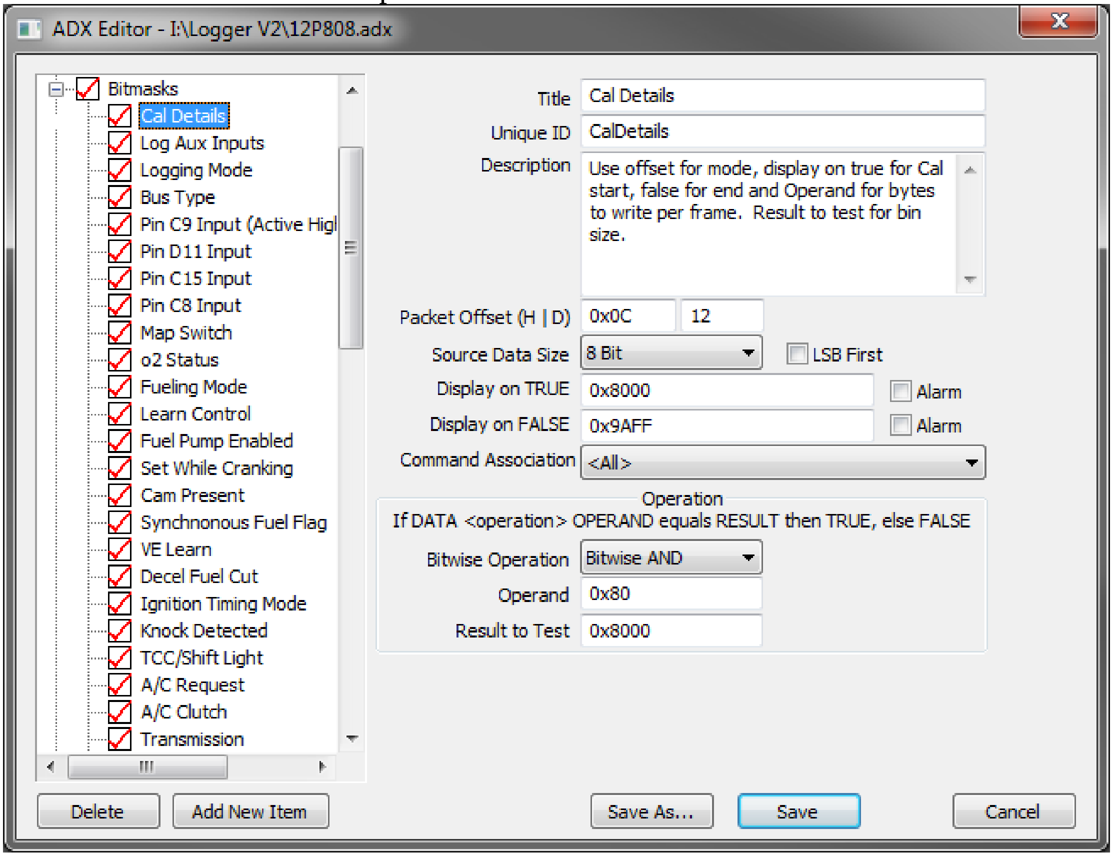


Quick Record

From any screen other than a file selection a long press of the control dial will cause the logger to attempt to connect. If Auto Record is enabled it will also start recording.

Calibration Load and Save

Currently only NVRAM equipped ECU/PCMs are supported for writing to the calibration. The logger can read the calibration from all factory Commodore ECU/PCMs including flash based versions. There is an ADX flag item called “Cal Details” to define the cal/bin paramters…



Title: Cal Details

Unique ID: CalDetails

Description: Use offset for mode, display on true for Cal start, false for end and Operand for bytes to write per frame. Result to test for bin size.

Packet Offset (H | D): 0x0C | 12

Source Data Size: 8 Bit □ LSB First

Display on TRUE: 0x8000 □ Alarm

Display on FALSE: 0x9AFF □ Alarm

Command Association: <All>

Operation

If DATA <operation> OPERAND equals RESULT then TRUE, else FALSE

Bitwise Operation: Bitwise AND

Operand: 0x80

Result to Test: 0x8000

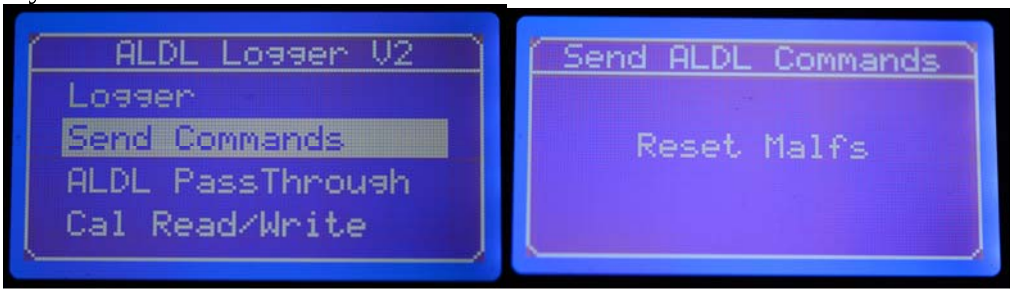
Packet Offset: ALDL Write mode. Depending on the PCMhacking.net bin your using the NVRAM write mode will vary. The default value in the supplied ADX is the recommended mode.
Display On True: Defines the calibration starting address. This is the actual memory address the ECU is using.
Display On False: Defines the calibration end address.
Operand: The number of bytes to write per frame in hex. The default value in the supplied ADX is recommended.
Result To Test: The total size of the bin in hex. This is used for padding the calibration so the saved file is equal to the bin size. When editing the cal file in

TunerPro this is the preferred method. If only the calibration is required then set this to 0x0

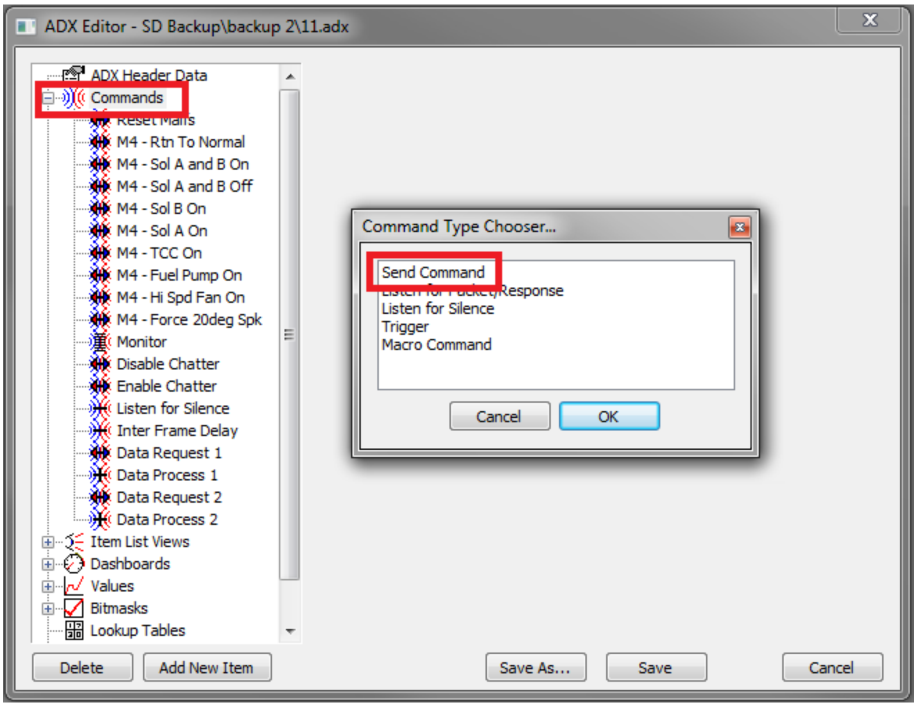
From the main display select Cal Read/Write to choose which function, read or write.

Sending Commands

The logger can be used to send commands to other ALDL devices. Select Send Commands from the main menu and any command in the current ADX will be displayed…



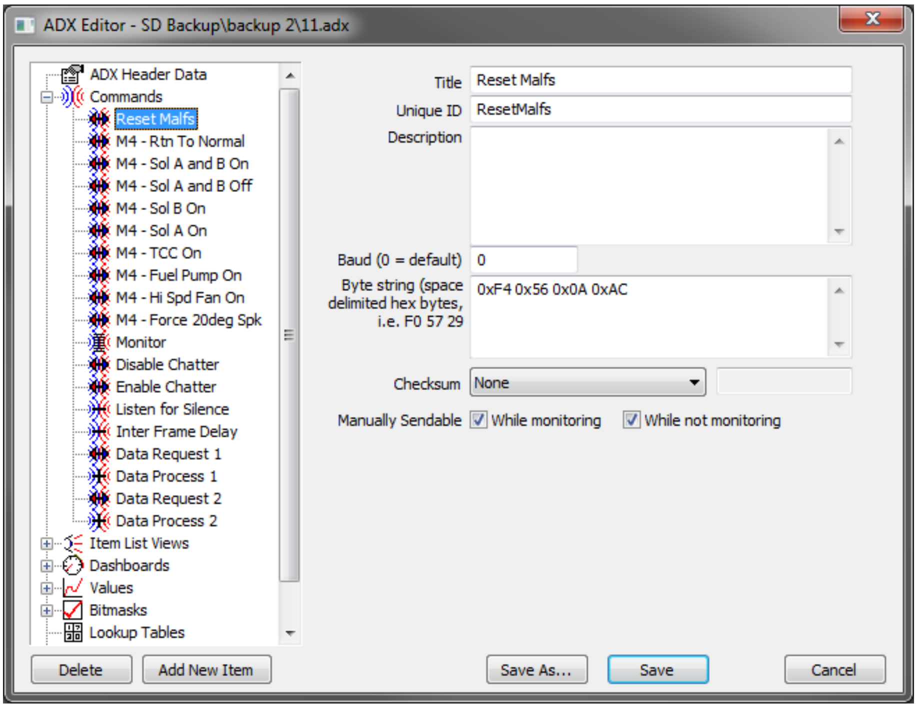
This is commonly used for mode 4 control or to reset malfunctions. Up to 10 commands can be loaded from the ADX at any one time.
To add a send command, open the ADX file in TunerPro and edit it. Select commands and then Add New Item. Select Send Command and click ok…



Command Type Chooser...

Send Command
Listen for Packet/Response
Listen for Silence
Trigger
Macro Command

This will create a new item which can then be setup to send a command.
Title - This should be something meaningful and will be what is displayed on the logger LCD. A maximum of 18 characters can be used.
UniqueID – can be anything that is a unique name in the ADX. TunerPro uses this to ID commands.
Byte String – This is the ALDL command. It will vary depending on the model of ECU and command being sent.
Checksum – Must always be none
Manually Sendable – Optional, only enables the command in TunerPro.



Title: Reset Malfs
Unique ID: ResetMalfs
Description:
Baud (0 = default): 0
Byte string (space delimited hex bytes, i.e. F0 57 29: 0xF4 0x56 0x0A 0xAC
Checksum: None
Manually Sendable ☑ While monitoring ☑ While not monitoring

There is no recurring command option, ie the command is only sent once when selected and not repeated at any time interval. GM have the option to enable mode 4 control commands for only a fixed amount of time. Most calibrations have not used this but if the commands time out and revert to normal control the calibration in the bin will need to be modified.

AUX Inputs

The logger has 4 analog inputs and 2 digital inputs. These 6 items can be inserted in the ALDL data stream to be viewed in TunerPro and as one (or more) of the 16 live view items, logged using partial pass-through mode or logged standalone to a PC. How the aux data is used and interpreted in TP is up to the user; however a setup guide to using analog 1 for wideband logging using the 3 options is included in the section Recording the Aux Data to an SD Card.

The analog inputs are high impedance 0-5V with 0-1023 counts (10-bits). The input can piggyback existing sensors without interfering.

The digital inputs can be used as either an On/Off flag or digital counter (to measure frequency or pulse width). The inputs are pulled high internally to the logger.

Example: Innovate Wideband Setup
The default analog output from innovate products is 0-5V representing 7.35 to 22.39 AFR.

In TunerPro the equation to display the AFR would be –
((22.39 – 7.35) * (X/1024)) + 7.35

This can be simplified for Logger Live View LCD display to: X / 68.085 + 7.35

With a 14point7 wideband the calculation is ((20-10) * (X/1024)) + 10. The simplified calc is X / 102.4 + 10

Example: To measure and display 0-5V the calculation would be X / 1024 * 5

The digital frequency inputs of the logger use a 2MHz counter with each count representing 0.5us. The input is able to withstand up to 25V with a logic high anything over 2V and a logic low anything under 0.8V. The logger records the high counts and low counts on the digital inputs and can be used to determine such things as duty cycle, pulse width, RPM, vehicle speed etc. The maximum count time (either high or low) is 5 seconds, both high and low counts are set to 0 if this occurs. The recommended operating frequency range is 0.5Hz to 6KHz.

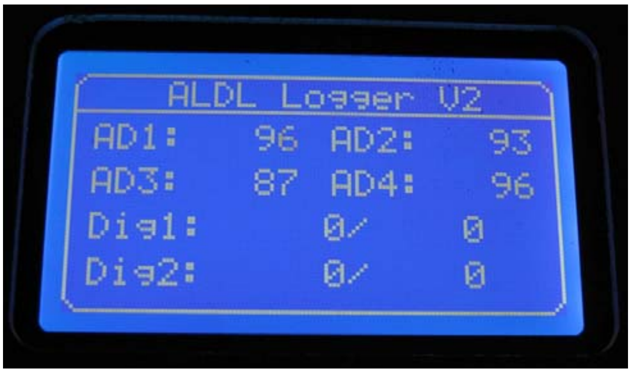
Example: Digital Pulse Width

Pulse width in seconds: X * (1 / 2000000)
Pulse width in milliseconds: X * (1 / 2000)

To display ms on the LCD, simplified: X / 2000

Monitoring the Aux Inputs

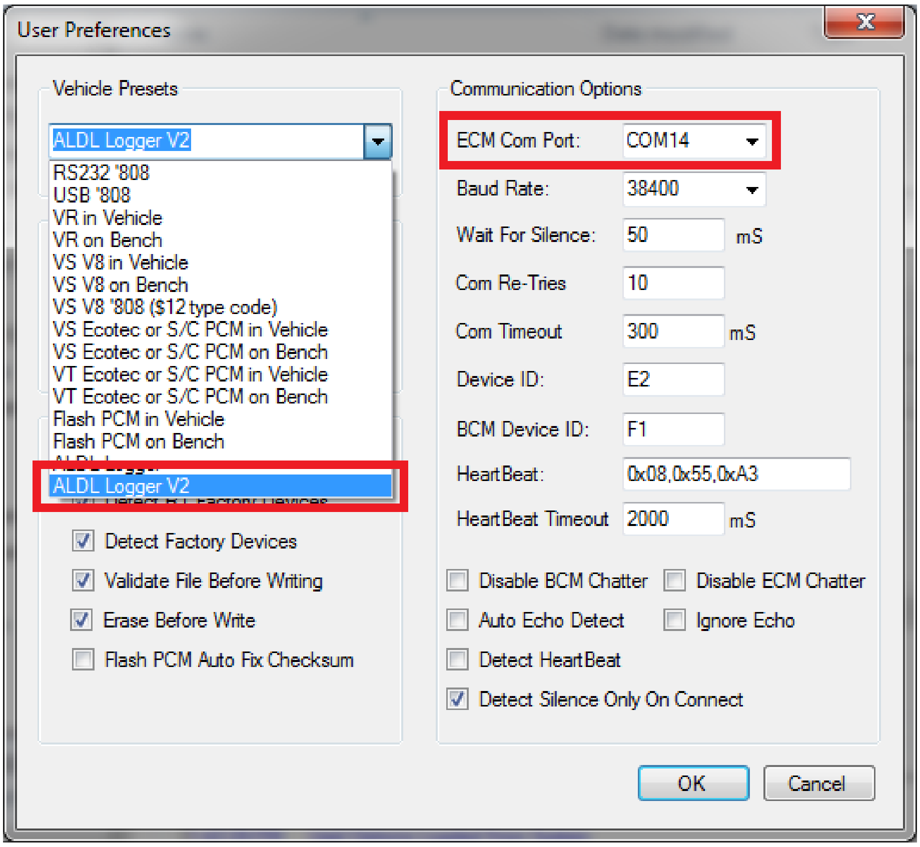
The aux inputs can be monitored from the Live View screen. Clicking through the two item displays and bit mask screen will then show the aux input screen. The values are the raw logged analog and the digital counts.



```
ALDL Logger V2
AD1:    96  AD2:   93
AD3:    87  AD4:   96
Dig1:       0/     0
Dig2:       0/     0
```

Setting the Real-time Clock

The logger is fitted with a real-time clock for time stamped files. This needs to be set to your local time zone, the Flashtool (V1.36 or later) available from www.pcmhacking.net is used to do this. The logger should be connected to the PC using the USB cable. In The Flashtool use the user preferences to select “ALDL logger V2” from vehicle presets menu and the virtual com port of the logger (see the ‘USB Virtual Com Port Setup’ section to setup the port).



User Preferences

Vehicle Presets

ALDL Logger V2

- RS232 '808
- USB '808
- VR in Vehicle
- VR on Bench
- VS V8 in Vehicle
- VS V8 on Bench
- VS V8 '808 ($12 type code)
- VS Ecotec or S/C PCM in Vehicle
- VS Ecotec or S/C PCM on Bench
- VT Ecotec or S/C PCM in Vehicle
- VT Ecotec or S/C PCM on Bench
- Flash PCM in Vehicle
- Flash PCM on Bench
- ALDL Logger V2

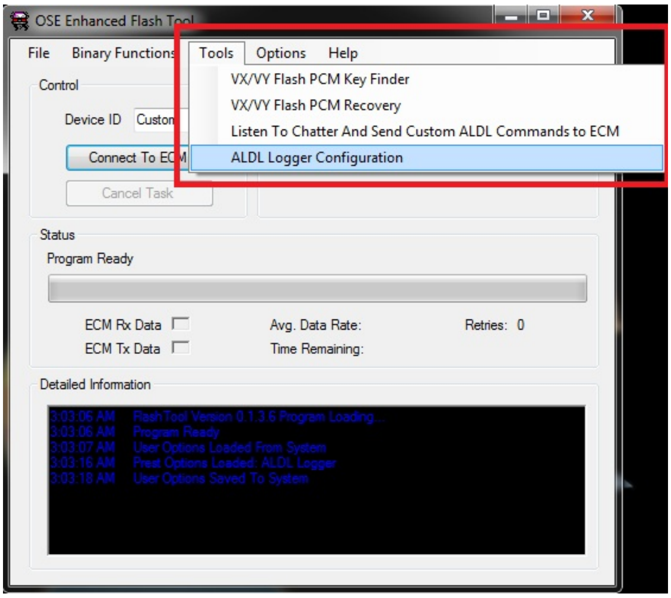
☑ Detect Factory Devices
☑ Validate File Before Writing
☑ Erase Before Write
☐ Flash PCM Auto Fix Checksum

Communication Options

ECM Com Port:	COM14	
Baud Rate:	38400	
Wait For Silence:	50	mS
Com Re-Tries	10	
Com Timeout	300	mS
Device ID:	E2	
BCM Device ID:	F1	
HeartBeat:	0x08,0x55,0xA3	
HeartBeat Timeout	2000	mS

☐ Disable BCM Chatter ☐ Disable ECM Chatter
☐ Auto Echo Detect ☐ Ignore Echo
☐ Detect HeartBeat
☑ Detect Silence Only On Connect

From the main Flashtool screen you can try connecting to the logger using the ‘Connect to ECM’ button, if successful you have configured The Flashtool correctly. Select ALDL Logger Configuration from the tools menu

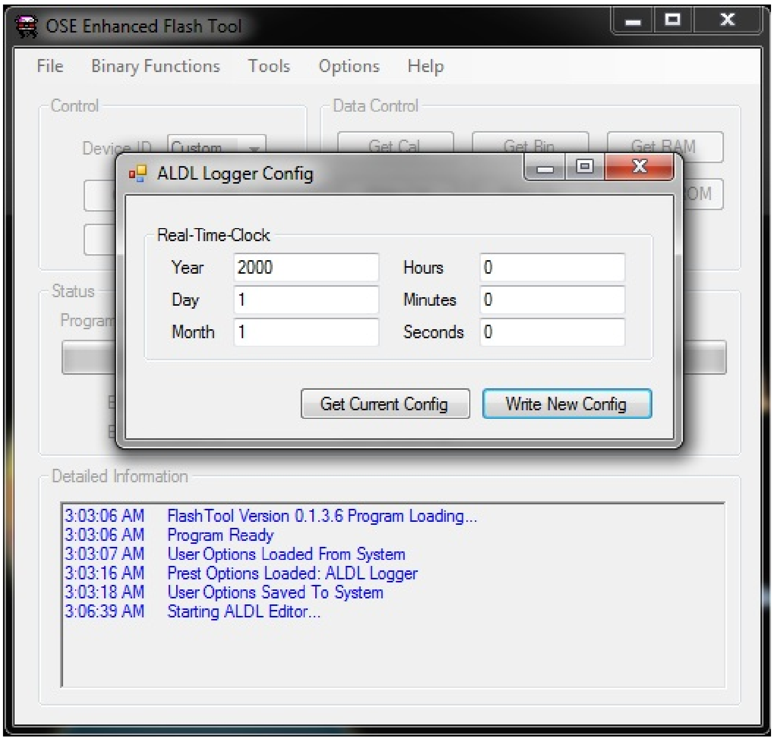


Program Ready

Retries: 0

```
3:03:06 AM    FlashTool Version 0.1.3.6 Program Loading...
3:03:06 AM    Program Ready
3:03:07 AM    User Options Loaded From System
3:03:16 AM    Prest Options Loaded: ALDL Logger
3:03:18 AM    User Options Saved To System
```

You can now either choose to get the current config from the logger and edit that or enter the current date and time in manually. If the RTC has never been set ie after replacing the battery, the screen will show the default values when choosing to get the current config. Set the data to the current date and time, click Write New Config to save them to the logger.



Year 2000
Day 1
Month 1
Hours 0
Minutes 0
Seconds 0

```
3:03:06 AM    FlashTool Version 0.1.3.6 Program Loading...
3:03:06 AM    Program Ready
3:03:07 AM    User Options Loaded From System
3:03:16 AM    Prest Options Loaded: ALDL Logger
3:03:18 AM    User Options Saved To System
3:06:39 AM    Starting ALDL Editor...
```

Setup Options

From the main screen select ‘Setup’. There are options here to configure the logger unit.

Backlight

The backlight is adjustable from 0-100%. Even at 0% it is partially on to be able to faintly view the display.

Dig2 Dimmer

This option enables dimming of the backlight by grounding the digital 2 input. The display is dimmed to 25% of the normal backlight setting above (ie if 80% backlight is set, this option would make the display 20% when digital 2 is grounded).

Buzzer

The Btn Beep option turns the beep on/off when the encoder button is pressed.

Sleep Option

This option can be enabled on setups where the logger is permanently powered, such as VR-VY vehicles. After 5 minutes of no activity, button press or ALDL transmit, the logger will enter a low power mode. Power consumption drops to less than 20mA. To wake the logger press the main control dial.

Firmware Updates

When new firmware is released the logger can be easily user upgraded:

1 - From the main logger screen select ‘Setup’.

2 – Select Firmware upgrade. Selecting this option halts the logger and prepares it for firmware download from a PC. It does not erase the current firmware. If you select this option by mistake just power cycle the logger.

3 – The firmware files are provided as hex files and are loaded using the USB interface and supplied cable. Open the Firmware Loader program included.

4 – From the file menu of the loader select Open HEX File and navigate to the logger firmware to load. Be sure to only load logger hex files, loading the incorrect firmware will require a recovery procedure.

5 – From the operation menu select program. The loader will first erase the logger and then program the new firmware.

6 – Once this is complete the logger can be disconnected from the PC. When the logger is next powered up confirm during boot the splash screen it shows the new version number.

ADX Configuration Debug

When initially setting up an ADX it often can be difficult to diagnose issues, the ALDL logger provides two debug modes to display loaded data.

Ensure a writable SD card is installed and during power on hold down the main control button. Once the logger has finished loading the ADX it will beep until the button is released and continue loading normally. The SD card will have a file called Debug_XX.txt written which contains all the items loaded from the ADX.

The logger can also display the configuration in a terminal program. An example of a free terminal program is Termite which can be downloaded from www.compuphase.com/software_termite.htm . The ADX data extracted from the file is printed to the PC screen.

The terminal program should be configured with 38400, 8, N, 1 and the virtual com port as shown in the device manager.

With the logger connected to the PC and the terminal program configured, select the 'Debug ADX' option in the ADX Load/Save menu. It will then display in the terminal the extracted ADX information.

Support

For further technical support please use the www.pcmhacking.net forums.

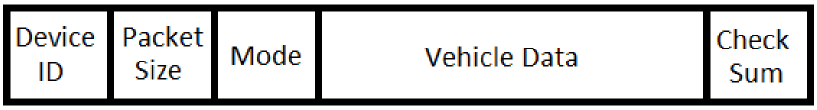
Using the Aux Inputs of the Logger V2

The ALDL logger V2 provides 4 analog inputs and 2 digital frequency inputs. The 4 analog inputs are 0-5V with 10bit (4.88mV) resolution. The 2 digital inputs use a 2MHz 32bit counter for 0.5us resolution. Both types of aux inputs offer 1% or better accuracy.

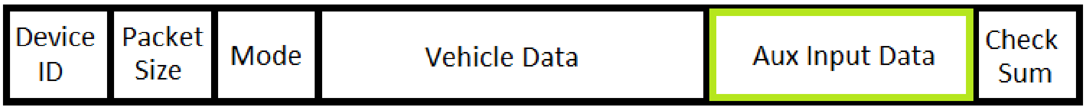
There are 3 ways of using these inputs:

1 – While logging a vehicle using the ALDL logger it can insert the aux input values in the data packet. This (now larger) data frame is recorded to the SD card in XDL format for viewing later in TunerPro. At the time of recording the data to the SD card the packet size and checksum is modified to reflect the changes.

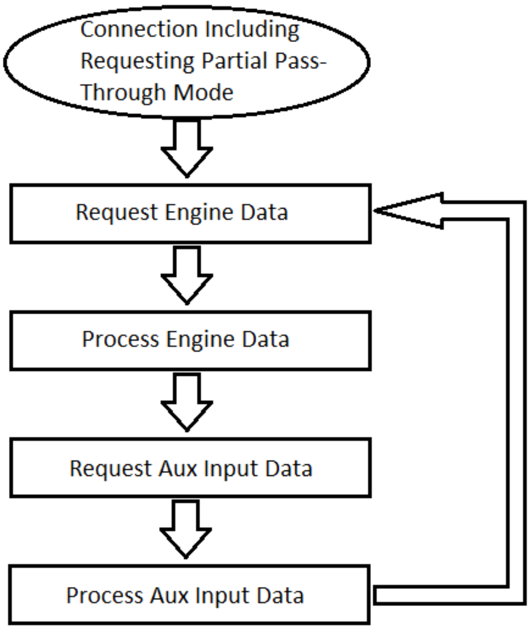
Original Logged Data Frame



With Aux Input Data Inserted



2 – Using TunerPro to log a vehicle through the ALDL Logger V2 in partial pass-through mode. Between normal vehicle data requests a request can be made to the logger for the aux input data.

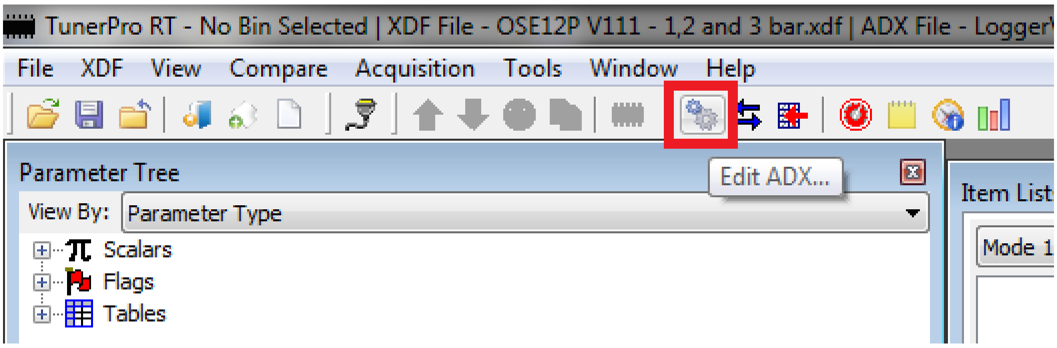


3 – Using TunerPro and the supplied ADX to log the logger. This allows the logger to be used as a 6 channel data logger.

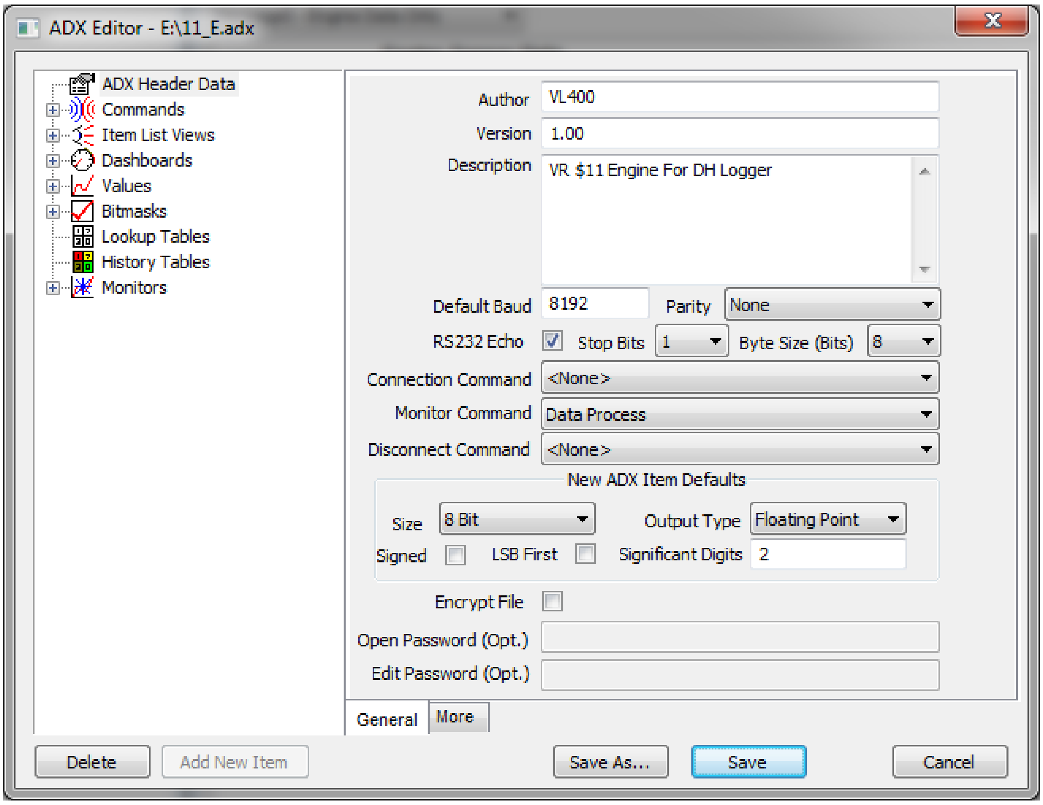
Recording the Aux Data to an SD Card

In this example analog input 1 will be used to log the wideband AFR from a 0-5v output of an Innovate Motorsports wideband controller. The ADX used by the logger needs to be modified to direct the logger to record the extra data and to specify the larger data frame size once the aux data is inserted. A $11 ADX is used in this example.

1) Open TunerPro and load the ADX to modify. It is a good idea to use a copy of a working ADX with a name that describes it recording the aux inputs eg: 11_E_Aux.ADX (ie $11, E for engine and Aux for Auxiliary). Click the Edit ADX button to begin.



A screen similar to below will be shown. On the left are the Commands, Bitmasks and Values main headings, if they are expanded (using the small + symbol next to the heading) all sub items will be shown.



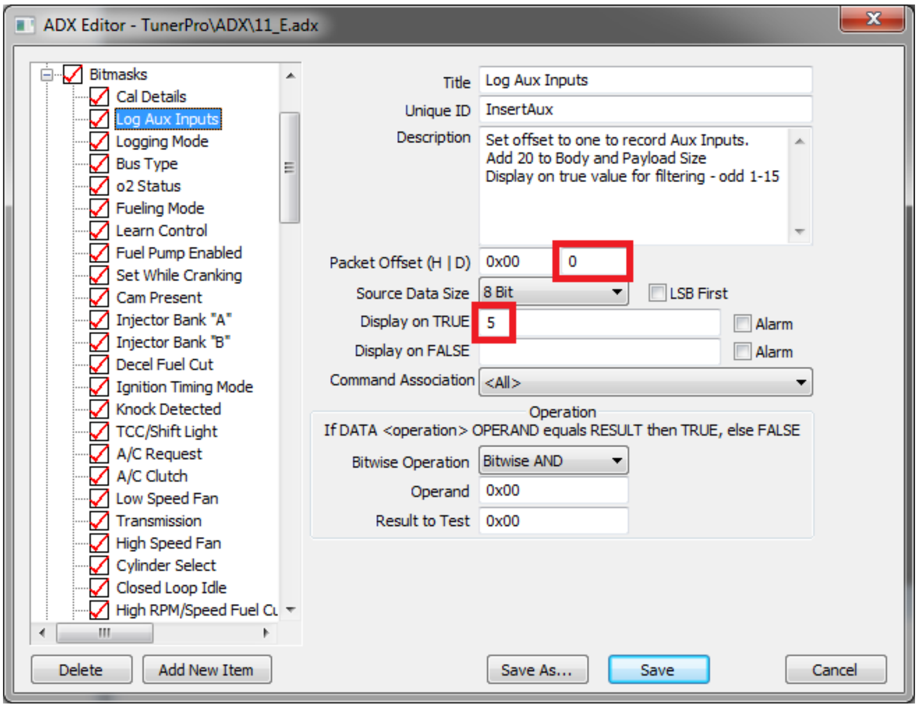
Author: VL400
Version: 1.00
Description: VR $11 Engine For DH Logger
Default Baud: 8192
Parity: None
RS232 Echo: ☑
Stop Bits: 1
Byte Size (Bits): 8
Connection Command: <None>
Monitor Command: Data Process
Disconnect Command: <None>

New ADX Item Defaults

Size: 8 Bit
Output Type: Floating Point
Signed: □
LSB First: □
Significant Digits: 2

Encrypt File: □
Open Password (Opt.):
Edit Password (Opt.):

2) The bitmask item ‘Log Aux Inputs’ needs to have the Packet Offset changed from 0 to 1 to enable the aux inputs to be inserted in the data stream while logging. The ‘Display on TRUE’ item defines the filtering of the input, the number can be any odd number from 1-15 with 1 being no filtering and 15 being maximum.

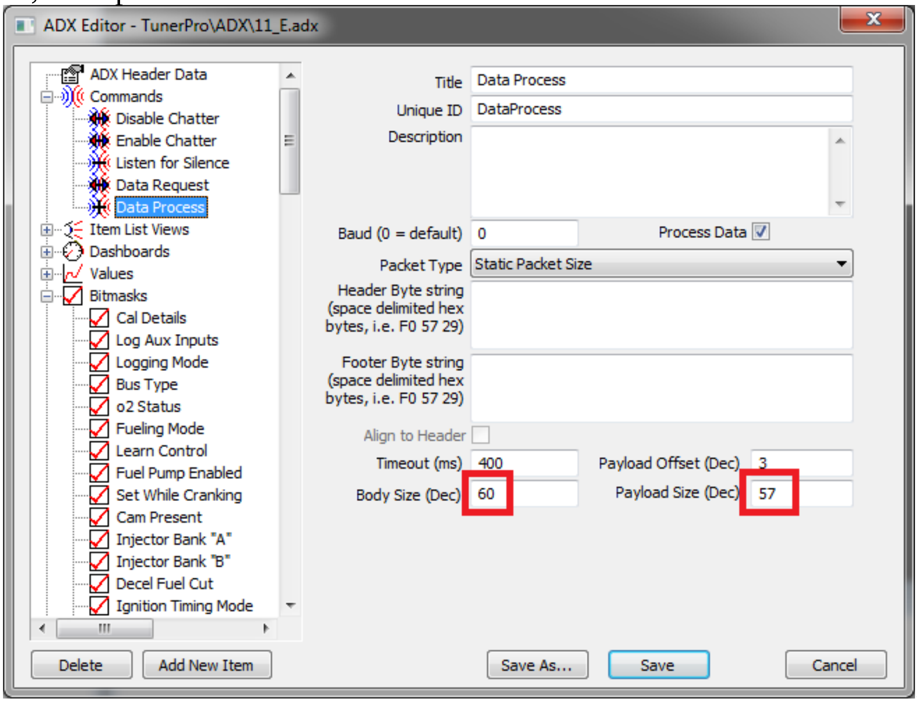


Title: Log Aux Inputs
Unique ID: InsertAux
Description: Set offset to one to record Aux Inputs.
Add 20 to Body and Payload Size
Display on true value for filtering - odd 1-15

Packet Offset (H | D): 0x00 | 0
Source Data Size: 8 Bit   □ LSB First
Display on TRUE: 5   □ Alarm
Display on FALSE: □ Alarm
Command Association: <All>

Operation
If DATA <operation> OPERAND equals RESULT then TRUE, else FALSE
Bitwise Operation: Bitwise AND
Operand: 0x00
Result to Test: 0x00

3) The packet size will now be much larger due to the ALDL logger adding extra bytes to the data frame. To let TunerPro know how large the frame is the packet and body size must be increased by 26 bytes. In this $11 ADX the values would be changed to 86 (60 + 26) and 83 (57 + 26). This must be done on all data process commands, for an ADX configured to use multiple messages there will be a Data Process 1, Data process 2 … etc



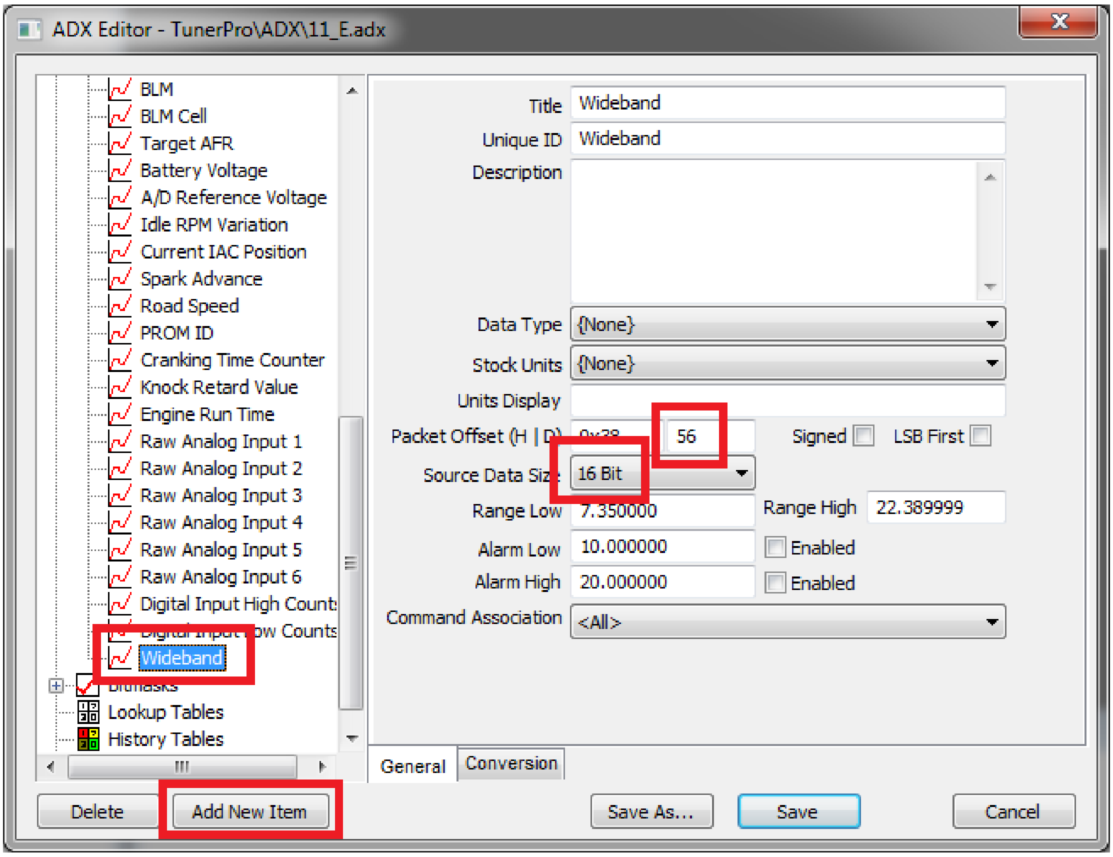
Title: Data Process
Unique ID: DataProcess
Description:
Baud (0 = default): 0   Process Data ☑
Packet Type: Static Packet Size
Header Byte string (space delimited hex bytes, i.e. F0 57 29):
Footer Byte string (space delimited hex bytes, i.e. F0 57 29):
Align to Header □
Timeout (ms): 400   Payload Offset (Dec): 3
Body Size (Dec): 60   Payload Size (Dec): 57

4) The final step is to define where in the packet the data should be taken from and how TunerPro should display the wideband AFR item. A new item can be created by clicking ‘Add New Item’.
- Assign a Title and Unique ID.
- Change the command association to ‘Data Process 1’.
- Set the Source Data Size to 16bits

- Set the Packet Offset to the bytes containing analog 1 data (they are part of the 26 extra bytes added to the end of the stream). To calculate the packet offset for the different aux inputs use the following:

Analog 1 Offset = Body Size - 30
Analog 2 Offset = Body Size - 28
Analog 3 Offset = Body Size - 26
Analog 4 Offset = Body Size - 24
Digital 1 High Offset = Body Size - 22
Digital 1 Low Offset = Body Size - 18
Digital 2 High Offset = Body Size - 14
Digital 2 Low Offset = Body Size – 10
Digital 1 On/Off **Bitmask** = Body Size – 6
Digital 2 On/Off **Bitmask** = Body Size – 5

Example:
Original Body Size = 60
New body size with aux inputs included = 60 + 26 = 86
Analog 1 Offset = 86 – 30 = 56



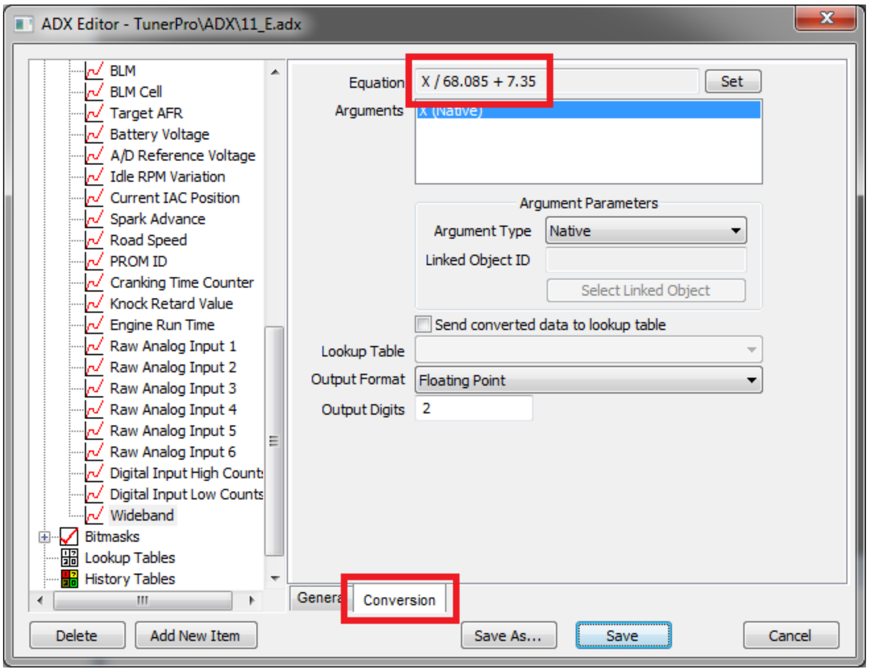
ADX Editor - TunerPro\ADX\11_E.adx

Title: Wideband
Unique ID: Wideband
Description:
Data Type: {None}
Stock Units: {None}
Units Display:
Packet Offset (H | D): 0x38 | 56
Signed ☐ LSB First ☐
Source Data Size: 16 Bit
Range Low: 7.350000
Range High: 22.389999
Alarm Low: 10.000000 ☐ Enabled
Alarm High: 20.000000 ☐ Enabled
Command Association: <All>

The conversion equation will be wideband specific. For the standard Innovate motorsports products the 0-5V analog output relates to 7.35 to 22.39 AFR. The logger represents 0-5V over a 0-1023 count range.

So in TunerPro the equation to display the AFR would be –
((22.39 – 7.35) * (X/1024)) + 7.35

This must be simplified if this item is to be displayed on the Live View LCD display -
X / 68.085 + 7.35



ADX Editor - TunerPro\ADX\11_E.adx

Equation: X / 68.085 + 7.35

Arguments: X (Native)

Argument Parameters

Argument Type: Native

Linked Object ID:

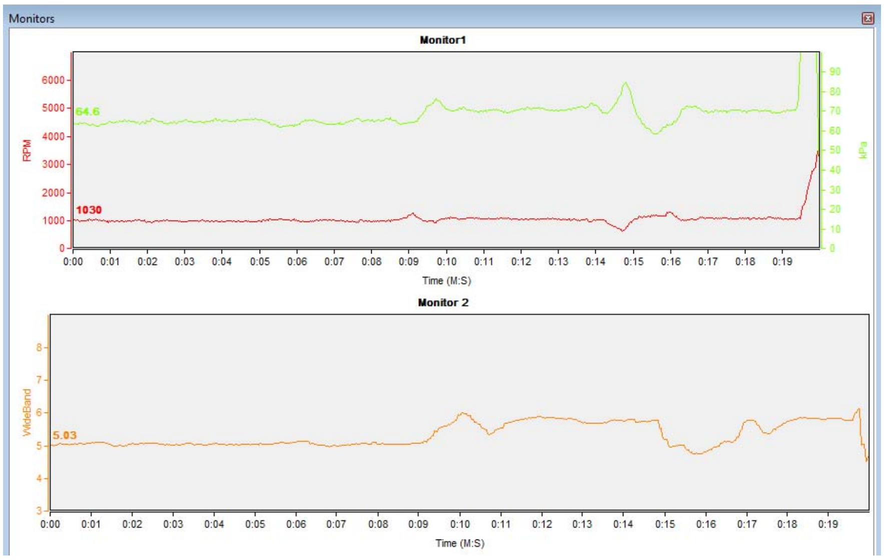
□ Send converted data to lookup table

Lookup Table:

Output Format: Floating Point

Output Digits: 2

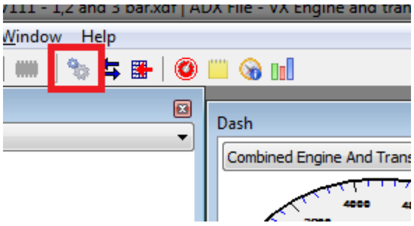
5) That completes the setup for creating a Wideband item to be logged to the SD memory card. Recording and then playing back a XDL log in TunerPro will show the wideband AFR. To display the wideband item in the monitor display, right click one of the monitors and select ‘Wideband’. Shown below is the recorded wideband item on monitor 2.



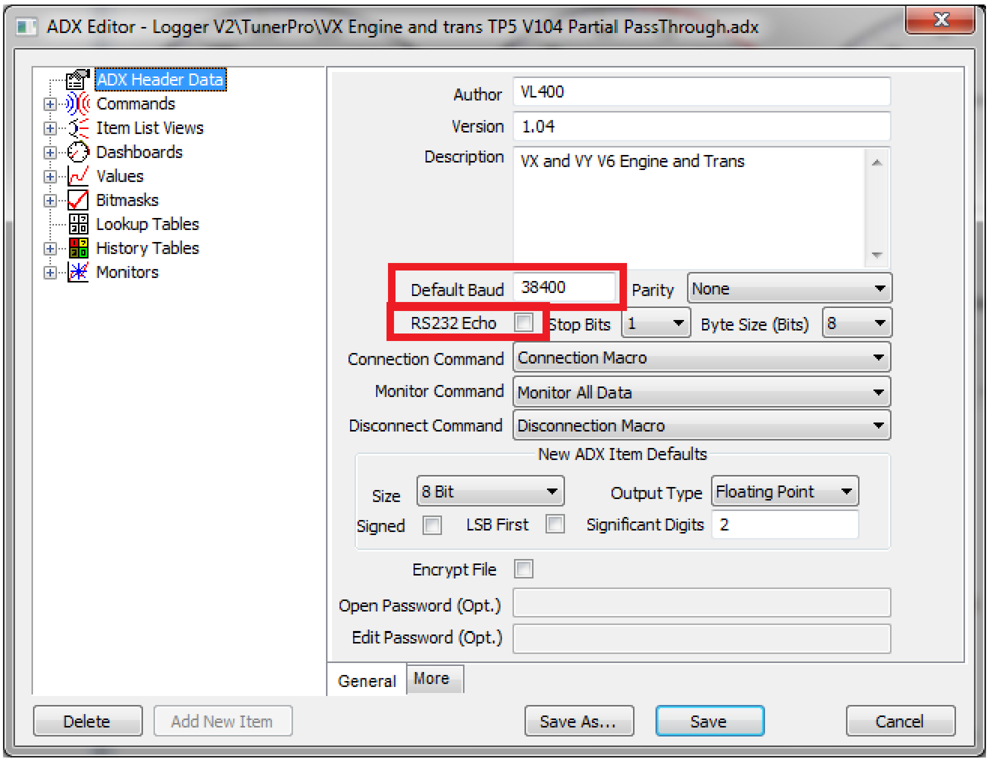
Using Passthrough Mode

To allow full or partial passthrough logging the ADX needs to be modified. To simply use passthrough only two items in TunerPro need to be edited. Change the default baud to 38400 from 8192 and turn off RS232 Echo.

Edit the ADX:


111 - 1,2 and 3 bar.xdf | ADX File - VX Engine and tran
Window
Help
Dash
Combined Engine And Trans

And then change the default baud to 38400 from 8192 and ensure RS232 Echo is unchecked:


ADX Editor - Logger V2\TunerPro\VX Engine and trans TP5 V104 Partial PassThrough.adx

Author: VL400
Version: 1.04
Description: VX and VY V6 Engine and Trans
Default Baud: 38400
Parity: None
RS232 Echo: □
Stop Bits: 1
Byte Size (Bits): 8
Connection Command: Connection Macro
Monitor Command: Monitor All Data
Disconnect Command: Disconnection Macro

New ADX Item Defaults
Size: 8 Bit
Output Type: Floating Point
Signed: □
LSB First: □
Significant Digits: 2

Encrypt File: □
Open Password (Opt.):
Edit Password (Opt.):

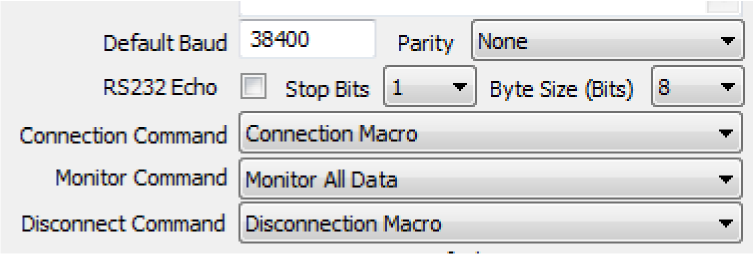
On the logger select the ALDL Modes menu item, set the bus type to match the vehicle type and finally turn Partial PassThr to on. Set TunerPro to use the com port of the logger and connect to the vehicle.

Advanced Partial Pass-through Configuration

Partial passthrough mode can be used for more than just having the logger as a USB to ALDL interface. It is possible to automatically select passthrough mode and request the aux inputs while logging the vehicle. This setup is more involved, for this reason an example ADX is included (VX Engine and trans TP5 V104 Partial PassThrough.ADX) to log the engine and trans along with the logger analog 1 and analog 2 inputs.

Steps to add partial passthrough with wideband logging to an existing ADX are as follows:

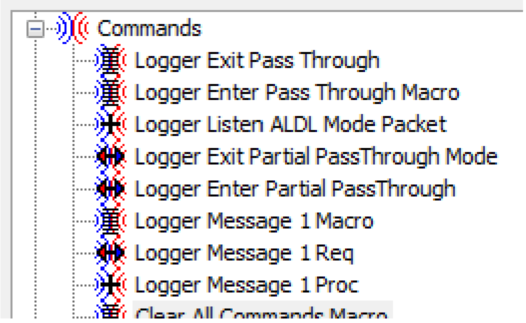
1. Create a copy of a working ADX
2. Open and edit the ADX. Change the default baud rate to 38400 and uncheck RS232 Echo. Take note of the connect, monitor and disconnect macro names as these will need to be edited.



Default Baud 38400 Parity None
RS232 Echo □ Stop Bits 1 Byte Size (Bits) 8
Connection Command Connection Macro
Monitor Command Monitor All Data
Disconnect Command Disconnection Macro

3. Expand the commands heading in the left hand column and create the logger command items by copying the example ADX. Create:

- Logger Exit Pass Through [MACRO]
- Logger Enter Pass Through [MACRO]
- Logger Listen ALDL Mode Packet [LISTEN FOR PACKET/RESPONSE]
- Logger Exit Partial PassThrough Mode[SEND COMMAND]
- Logger Enter Partial PassThrough [SEND COMMAND]
- Logger Message 1 Macro[MACRO]
- Logger Message 1 Req[SEND COMMAND]
- Logger Message 1 Proc[LISTEN FOR PACKET/RESPONSE]

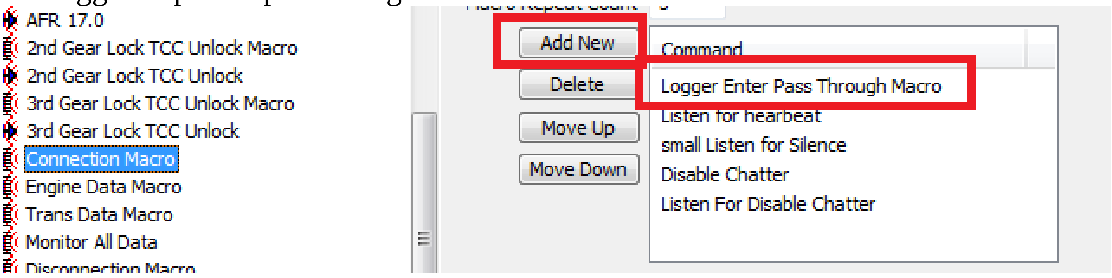


Commands
Logger Exit Pass Through
Logger Enter Pass Through Macro
Logger Listen ALDL Mode Packet
Logger Exit Partial PassThrough Mode
Logger Enter Partial PassThrough
Logger Message 1 Macro
Logger Message 1 Req
Logger Message 1 Proc
Clear All Commands Macro

The 'Logger Enter Partial PassThrough' send command needs to be configured to suit the bus type, either one wire ALDL (VR or later) or '808 using an RS232 board.
'808 mode = 0xE2 0x58 0x22 0x03 0x00
ALDL mode = 0xE2 0x58 0x22 0x03 0x01

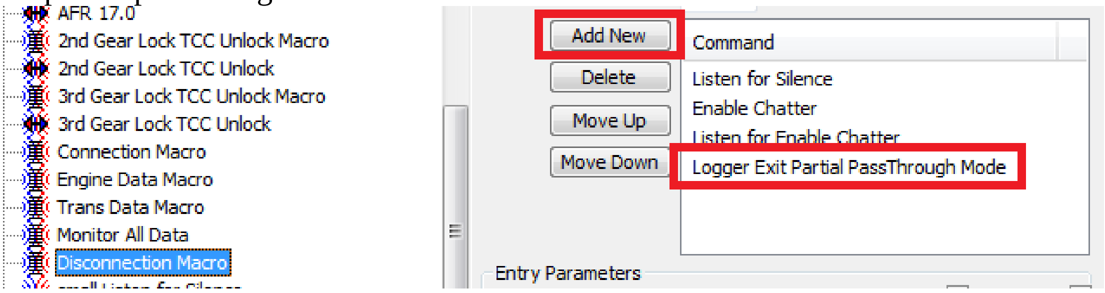
The final number specifies the bus type for pass-through modes. 0x00 for an '808 and 0x01 for one wire ALDL (VR and later)

4. Change the existing connection macro so the first command is to put the logger in partial passthrough mode



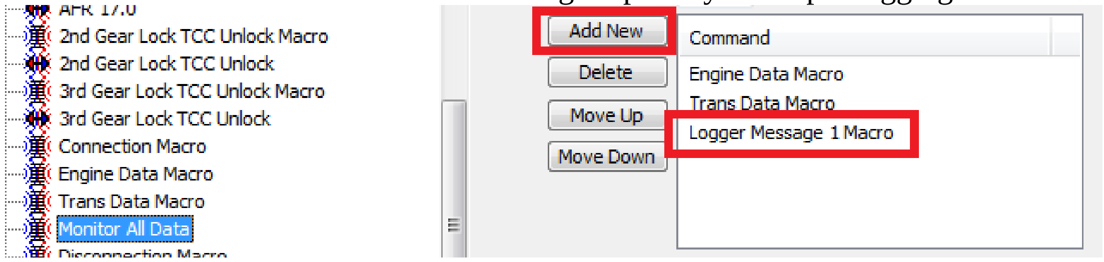
Logger Enter Pass Through Macro
Listen for hearbeat
small Listen for Silence
Disable Chatter
Listen For Disable Chatter

5. Change the existing disconnect macro so the last command is to exit from partial passthrough



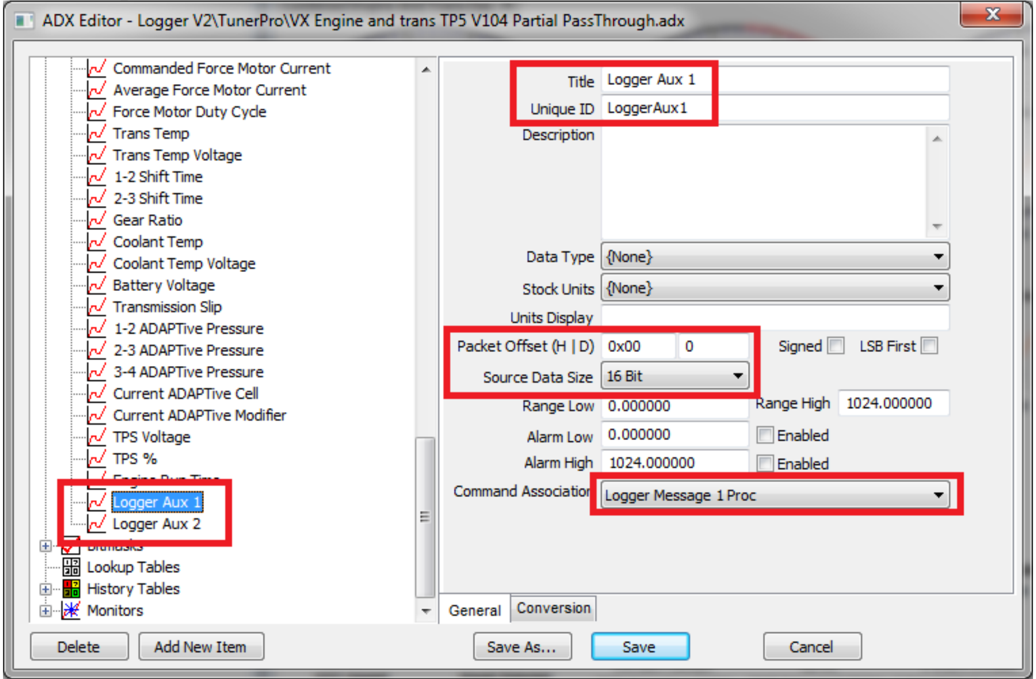
Listen for Silence
Enable Chatter
Listen for Enable Chatter
Logger Exit Partial PassThrough Mode

6. Change the existing monitor macro to include the logger message 1 macro. This can be added more than once for higher priority aux input logging.

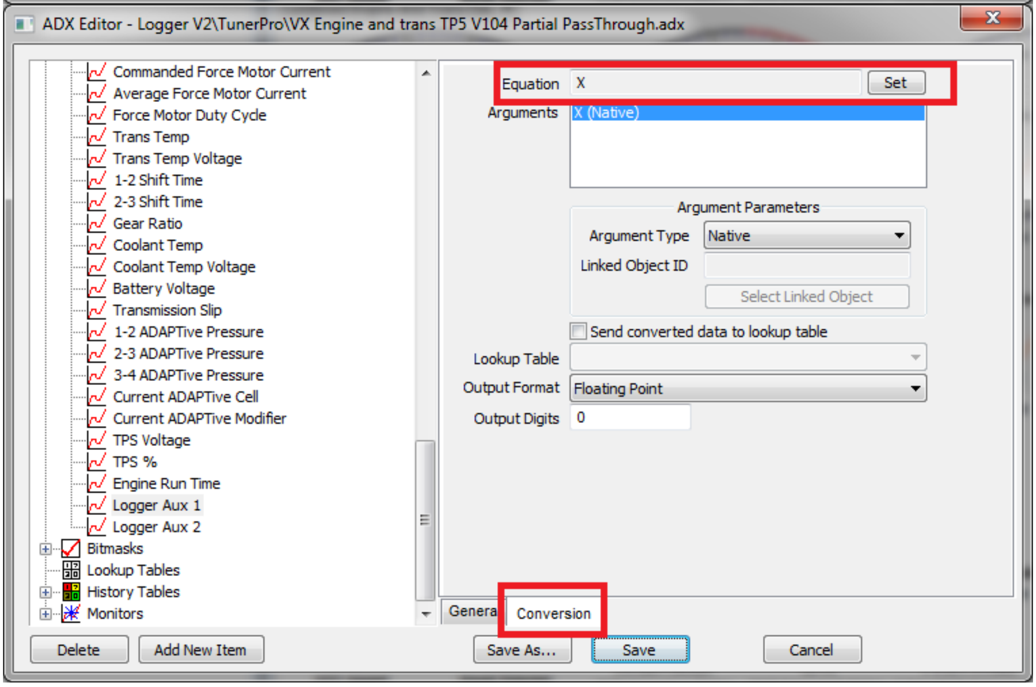


Engine Data Macro
Trans Data Macro
Logger Message 1 Macro

7. Create 2 new items that use the logger aux input data. Expand the ‘Values’ heading in the left column and click ‘Add New Item’. The two items can be called anything that describes the function they will be used for. The packet offset is 0 for Analog 1 and 2 for Analog 2 while the source data size is 16 bits. The item must be associated only with the logger command so change this to ‘Logger Message 1 Proc’. If this item is going to be used for an Innovate Wideband the range high/low and the equation can be setup as per the first example above.


ADX Editor - Logger V2\TunerPro\VX Engine and trans TP5 V104 Partial PassThrough.adx

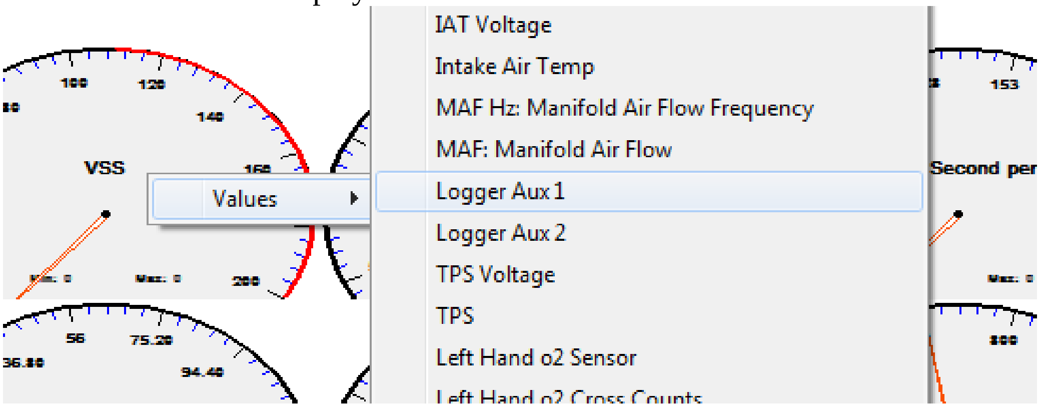
Title: Logger Aux 1
Unique ID: LoggerAux1
Description:
Data Type: {None}
Stock Units: {None}
Units Display:
Packet Offset (H | D): 0x00 | 0
Signed □ LSB First □
Source Data Size: 16 Bit
Range Low: 0.000000
Range High: 1024.000000
Alarm Low: 0.000000 □ Enabled
Alarm High: 1024.000000 □ Enabled
Command Association: Logger Message 1 Proc

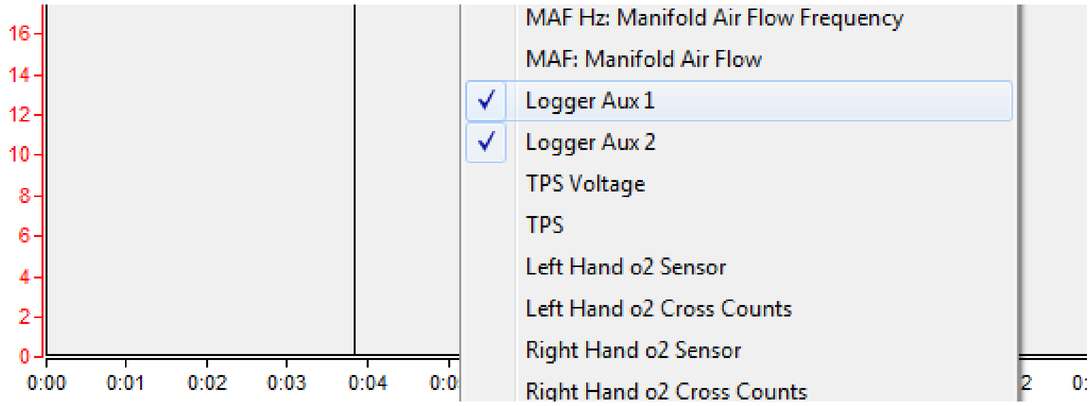

ADX Editor - Logger V2\TunerPro\VX Engine and trans TP5 V104 Partial PassThrough.adx

Equation: X
Arguments: X (Native)
Argument Parameters
Argument Type: Native
Linked Object ID:
□ Send converted data to lookup table
Lookup Table:
Output Format: Floating Point
Output Digits: 0

8. The ADX is now configured to automatically set and release the logger to partial passthrough mode. It will connect to the vehicle, data log the engine while in between frames request the aux input data for analog 1 and 2. When disconnecting the logger is put back to normal mode.

9. To view the analog input data while logging right click either on a dash gauge or in the monitor display





Using the Logger V2 as a Standalone Data Logger in TunerPro

Using the supplied LoggerV2.ADX file the logger has the ability to display the aux inputs in TunerPro for testing or to be used as a standalone data logger. The logger should be connected to a PC using the supplied USB cable. TunerPro must be configured to use the virtual com port of the logger (see the ‘USB Virtual Com Port Setup’ section to setup the virtual port).

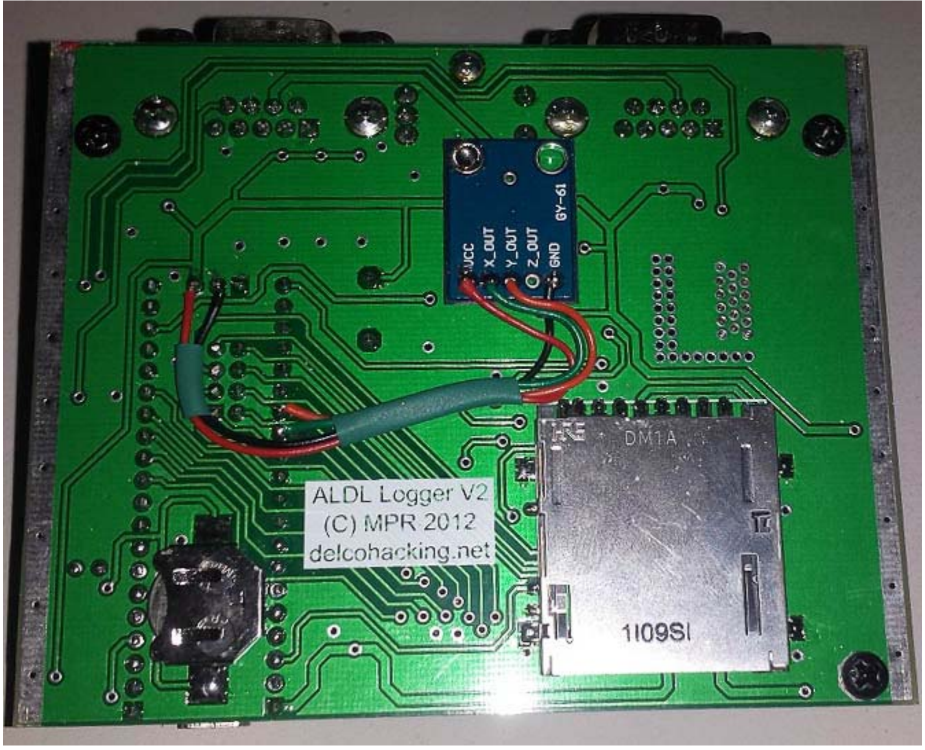
You can now start logging just like you would on a vehicle ECU, TunerPro will show all the auxiliary inputs on your PC screen.

Using Firmware V1.13+ with a 2 channel Accelerometer

There are two firmware files supplied with the logger, the regular version and a + version, with the only difference that the + version supports a 2 channel (X and Y) accelerometer internally to the logger. The requirement is the accelerometer outputs an analog voltage proportional to the G force. At this time digital accelerometers (I2C or SPI) are not supported. A supported accelerometer, and what will be used in these instructions, is one from hobbytronics .. http://www.hobbytronics.co.uk/adxl335-breakout. Any board using the ADXL335 chip outputting the analog 0-3.3volt signal is supported.

The accelerometer is installed internally using 2 spare analog inputs, if you do not wish to install the board internally then 2 of the 4 analog aux inputs on the DB9 connector can be used with standard logger firmware. The + version makes use of two spare analog inputs internal to the logger, leaving the 4 aux inputs free for other logging (effectively allowing 6 analog inputs logged).

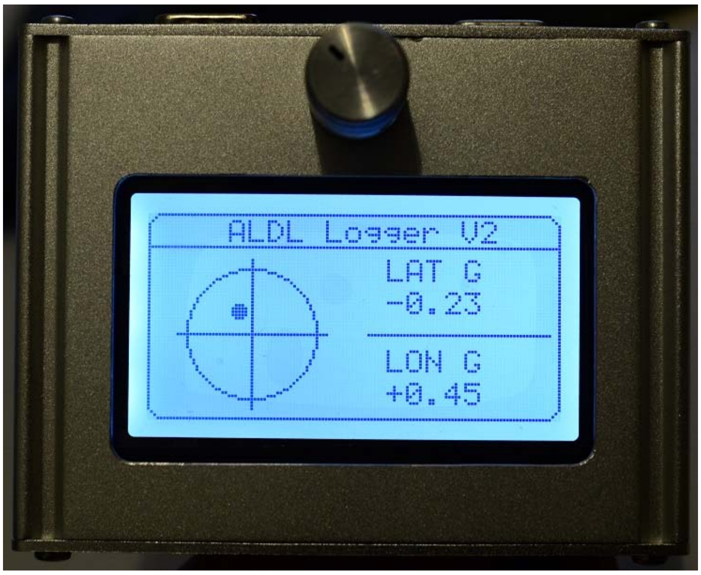
An example of the installation is shown below…



The wiring is as follows…
Red – 5V
Black – Ground
Green – X axis
Orange – Y axis

The board is held in place by being soldered to the ground pad of the push button control dial. If this method is used ensure the solder pad of the ADXL335 board is ground or has no connection, otherwise a short circuit could be introduced.

Once installed the accelerometer can be tested. Load the '+' firmware using the method outlined in 'Firmware Updates'. With the USB cable still connected navigate through the live view screens (the ECU does not need to be connected), there is now a new display shown below…



Physically moving the logger will move the dot and display the lat and long G forces.

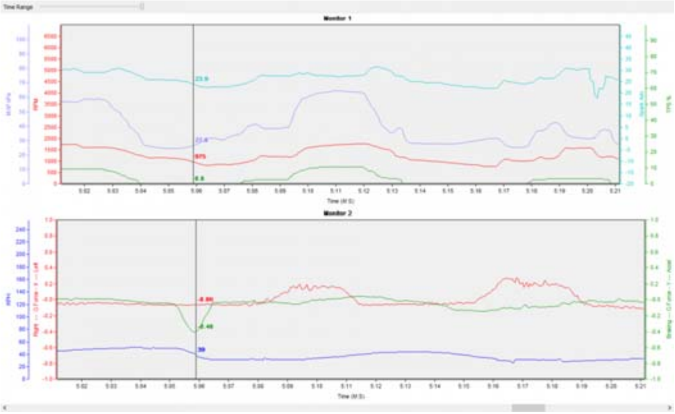
The analog outputs from the ADXL335 board are 0.66 – 2.64 volts covering -3G to +3G, with 1.65V representing 0G. The calculation for TunerPro to display -3G to +3G is … (X / 67.584) – 5. With the loggers 10bit A/D, resolution is approximately 0.01G over the 6G range.

The included LoggerV2+.adx file can be used to log the logger using TunerPro, it also contains the X and Y axis items that can be used as an example to add the two new items to your ADX.

To modify existing ADX files

- The ADX must be previously setup to log the aux inputs
- Increase the body size and payload size by 4 bytes in all data process replies.
- Define the X and Y items. The offset is as follows..
    - X Axis - Body Size – 8
    - Y Axis - Body Size – 6

An example test log showing the accelerometer channels and vehicle speed in the bottom monitor…



Required Changes when Updating From V1.10

When upgrading to V1.12 from V1.10 or eralier and using existing ADX files there are two changes required. Open and edit the ADX in TunerPro.
1 – The ‘Listen For Silence’ item needs to have the timeout longer than the silence length. The recommended value is silence length of 50ms (15ms on VT and later) and timeout of 400ms.
2 – If the aux inputs of the logger are being recorded the Data Process X item needs to be changed the Body size and Payload size increase by 2 bytes. This allows for the digital input 1 and 2 state flags to be recorded. See “Using the Aux Inputs of the Logger V2” on setting up the new bitmasks.

Firmware Changes For V1.11

- ADX: Made all ADXs log the aux inputs by default
- Bug Fix: The ADX parsing of disable and enable chatter commands was incorrect.
- Bug Fix: Improved all file time stamps as some were missed.
- Improved the connection on busy ALDL bus connections. Syncs faster to the bus.
- Added option for get cal and write cal with only the cal file without padding
- Added alarm high and alarm low items, fault output and buzzer.
- Improved low voltage cut out tolerance and SD file writing
- Auto record option now also allows just pressing connect from logger menu.
- Added power disconnect for RS232 '808s when USB is connected to stop the logger powering the ECU with key off (V2.1 hardware only)
- Added digital on/off state for digital inputs
- Added Listen Only Logging
- Added a chart display

Firmware Changes For V1.12

- Recording time added to display
- Display dimming via digital input 2
- Bug Fix: Fixed issue when some units showed ‘ERR’
- Bug Fix: Fixed issue when alarm item 8 jumped to page 2 instead of 1

Firmware Changes For V1.13

- Improved the calculation code for live view items
- Added sleep mode option.

Firmware Changes For V1.131+

- Bug Fix: Incorrect calculation for G display readings
- Changed Accel Display to show +-1.18G