

EMS Analyser Software User Guide

Version 3.2



EMS Computers Pty Ltd
Unit 9 / 171 Power St
Glendenning NSW 2761
Australia

Ph: +612 9675 1414
Email: support@fuel-injection.com
Web: www.emsefi.com

Overview	5
EDL-7 Dash Manager	6
Initialising an SD Card.....	6
Saving changes to the SD card.....	6
Clearing Log Files.....	7
Updating EDL7 Race Dash Firmware.....	7
Backing up the Race Dash settings.....	7
Setting Channel Translations and Names.....	8
Changing the channel name.....	8
Define a channel without a translation method.....	9
Defining a channel that uses a direct calculation.....	9
Filtering channel data.....	9
Creating Translation Tables.....	10
Linear translation.....	11
Step translation.....	11
Loading and Saving Translation Tables.....	12
Translating EMS Motorsport and Stinger ECU data.....	13
Translating MoTec M800 ECU Data.....	15
Special Log fields.....	16
Lap_Number:.....	16
Lap_Counter:.....	16
Track_Time:.....	16
Lap_Distance:.....	16
Selecting channels to log.....	17
EMS Maths Definition Language.....	18
Downloading log files from the SD card.....	18
Types and Expressions.....	19
Operators.....	19
Constants.....	19
Predefined functions.....	20
Numeric functions:.....	20
Statistical functions.....	20
Logical functions:.....	20
Specialised Functions.....	21
Variables.....	22
Global Variables.....	22
Local Variables.....	22
Programming syntax.....	23
Using predefined functions.....	23
Return(value):.....	23
Block statements.....	24
If else statements.....	25
Saving MDL programs.....	27
Testing your code.....	27
MDL Editor.....	28
Operators.....	28
Math Functions.....	28
Log Channels.....	28
Maths Channels.....	28
Add to Repository.....	28

New Channel.....	29
Global Variables.....	29
Code Repository.....	29
Dash Emulator.....	30
Setting Gauge Properties.....	30
Repositioning and resizing Gauges.....	30
Removing Gauges.....	30
Adding Gauges to the Main Dash Screen.....	31
Adding gauges to the second screen.....	31
Dial gauge.....	32
Data Channel.....	32
Minimum Value and Maximum Value.....	32
Number formatting.....	32
Start and End Danger Zone.....	33
Divider.....	33
Number of Labels.....	33
Number of Ticks.....	33
Number of digits.....	33
Decimal places.....	33
Gauge Title.....	33
Layer Order.....	33
Danger Colour.....	34
Gauge Face Colour.....	34
Background Colour.....	34
Text Colour.....	34
Show Danger Zone.....	34
Text Gauge.....	35
Data Channel.....	35
Gauge Title.....	35
Layer Order.....	35
Danger Limit.....	35
Number formatting.....	35
Danger Text Colour.....	36
Normal Text Colour.....	36
Background Colour.....	36
Show gauge tile.....	36
Segment Gauge.....	37
Data Channel.....	37
Number of digits.....	37
Number Decimals.....	37
Gauge Title.....	37
Layer Order.....	37
Danger Limit.....	37
Danger Colour.....	38
Normal Colour.....	38
Background Colour.....	38
Text Colour.....	38
Show gauge tile.....	38
Indicator gauge.....	39
Data Channel.....	39

Minimum Value and Maximum Value.....	39
Number formatting.....	39
End of colour zone 1.....	39
End of colour zone 2.....	40
Number of Leds.....	40
Gauge Title.....	40
Layer Order.....	40
Background colour.....	40
Text Colour.....	40
Show Value as text.....	40
Show Gauge name.....	40
Vertical indicator.....	40
LED Contrast.....	40
Data Analyser.....	41
Opening a Log File.....	41
Channel Data.....	42
Selecting a channel to display.....	42
Remove a data channel from the chart.....	42
Sorting the data.....	42
Un-sorting the data.....	42
Set Display Options.....	42
Lap Selection.....	43
Selecting a particular lap number.....	43
Viewing lap times.....	43
Viewing your race line.....	43
Session Information.....	44
Selecting different Chart options.....	44
Separated charts.....	44
Overlaying charts.....	44
Line charts.....	44
Scatter charts.....	44
Zooming into the chart.....	45
Dragging a chart.....	45
Creating a track map.....	46
Overlaying a race line over a track map.....	46
Recording lap sectors and times using the GPS.....	47
Installing the sector file onto the Race Dash.....	47
Recording Sector Messages using the GPS.....	48
Installing the Message file onto the Race Dash.....	48

Overview

The EDL Analyser software has 2 parts.

1. The Analyser provides the tools you need to analyse data from the EDL Race Dash
2. The EDL Manager provides the tools you need to set up translation tables initialise SD Cards and import data files from the SD card into the analyser software.

The EDL Race dash comes pre-configured to read data from standard EMS ECU data channels, however you may want to connect your own sensors to any of the data input channels of the ECU or the data input channels of the Race Dash. These channels read either voltage or pulses this data must be translated to a meaningful unit of measure such as Degrees, Pressure, Speed, etc. This is done using one of the translation methods, either a direct calculation or a translation table for each data channel.

The Race Dash configuration and channel translations are set up and saved onto the SD card using the EDL Manager. You must specify the Dash model and ECU model. This will configure the SD Card with the correct data channels and Dash features for your ECU and Race Dash.

EDL-7 Dash Manager

The Dash Manager software is disabled until you select an SD Card from the SD Card selector. Once an SD Card drive letter is selected the software will automatically read the configuration, if one exists and enable the screens. If the SD card is initialised for the Street Dash model then only the SD Card screen will be active. If the SD card is not initialised then you must first initialise it before you can define any configuration and translation tables. If you make any changes then you must remember to save the changes to the SD card.

Initialising an SD Card

Initialising an SD card will place the correct configuration onto the card.

To initialise an SD card select the SD Card drive letter, the ECU model and the Dash model. Then select menu options **SD Card -> Initialise New SD Card** or simply press the “**Save to SD Card**” button. After a few seconds the SD card will be initialised and ready to insert into the Race Dash. After initialising you can use the Dash Emulator to customise the dash screen to your preferences. If you are using any of the user definable data channels such as the analogue inputs or pulse inputs then you must also define the translation tables or direct calculations for each channel.

Saving changes to the SD card

If you made changes to the configuration or translations then you must save these changes. To save changes press the “**Save to SD Card**” button. If the card has not been initialised then you will be prompted to initialise it. The card will be initialised according to the selected ECU and Race Dash model. When initialised you can make changes and save them.

Clearing Log Files

The Race Dash allows you to save multiple logs onto an SD card. Each log file is given a name plus a log number appended to the end of the name. This number is incremented for each new log. Eventually you will need to clear the logs and reset the log number counter.

To clear the logs select menu option **SD Card -> Clear Logs from card** or press the **“Clear All Logs”** button.

Updating EDL7 Race Dash Firmware

The Analyser software comes with the latest version of the EDL7 Firmware. However, there are times when new firmware is released without new Analyser software. You can download this from the EMS website as an EDL7 Firmware update. This will install a new firmware version into the existing Analyser software so you can then use it to update the EDL7 Race Dash.

To update the EDL7 firmware requires 2 steps.

Step 1:

Insert the SD card for the EDL7 Race Dash and select the drive letter assigned to it. Select menu option **SD Card -> Load new firmware to card** or press the **“Load New EDL 7 Firmware”** button. You should see a selection of firmware versions. Select the version you want to install and press the Open button. This will copy the file to the SD card.

Step 2:

Insert the SD card into the EDL7 Race Dash and turn the power on. The Race dash will recognise that there is a version upgrade and automatically install it into its memory.

Backing up the Race Dash settings

You can backup the Race Dash settings and restore them again if needed. The settings will be saved according to the SD Card ID. This way you can backup settings for many different Race Dashes. When restoring previously backed up settings the system will automatically restore from the backup that exactly matches the SD Card ID. If the SD Card is newly initialised then you must change the Card ID to match a known backup and press the **“Save to SD Card”** button first before restoring. You must do this so the system knows which backup from which to restore.

Setting Channel Translations and Names

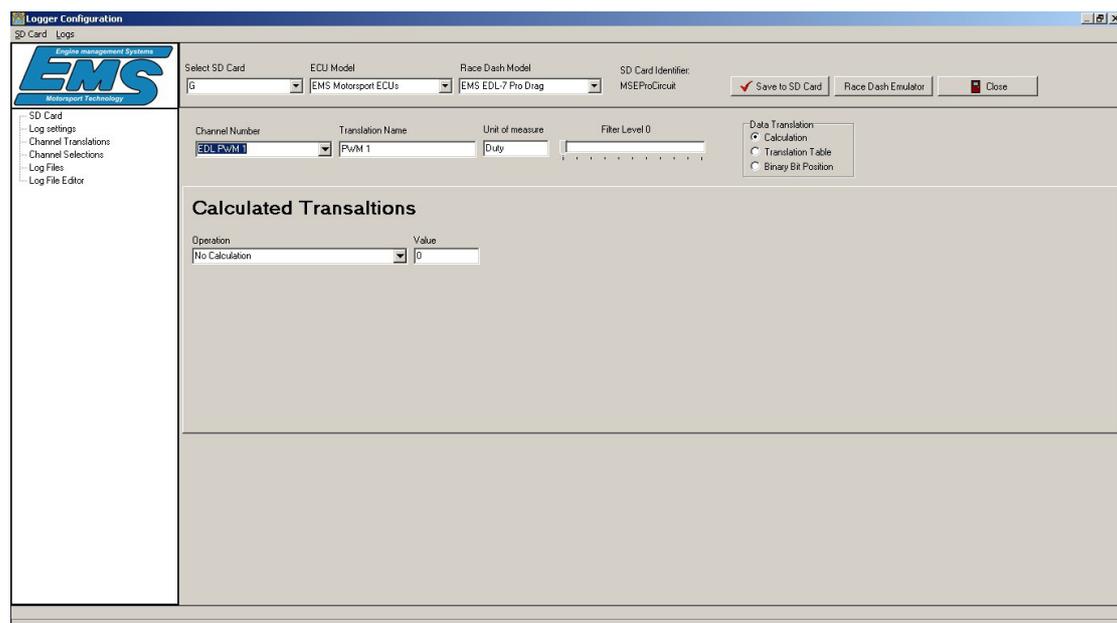
Every channel has a default name, however you can change the display name and unit of measure via the translation configurations.

Changing the channel name

A channel named Analogue 1 is not meaningful because it does not describe the information it is recording. You can give it a new name by typing it into the translation name field. You can also type a unit of measure into the unit of measure field. These names will be shown on the Race Dash gauges and the data log files.

EG: If you have a pressure gauge connected to Analogue 1 that is used to measure oil pressure then a meaningful name could be “Oil Press.” and unit of measure could be “psi”.

Data from some channels may need a translation to convert the raw data into a value that is easy to understand. Other channels may not need any translation because it has already been translated by the ECU. Either case must be defined for each channel.



The screenshot shows the 'Logger Configuration' window for EMS. The interface includes a sidebar with navigation options: 'SD Card', 'Log settings', 'Channel Translations', 'Channel Selections', 'Log Files', and 'Log File Editor'. The main area is divided into several sections:

- Top Section:** Contains dropdown menus for 'Select SD Card' (G), 'ECU Model' (EMS Motorsport ECUs), 'Race Dash Model' (EMS EDL-7 Pro Drag), and 'SD Card Identifier' (MSEProCircuit). It also has buttons for 'Save to SD Card', 'Race Dash Emulator', and 'Close'.
- Channel Configuration Section:** Features dropdowns for 'Channel Number' (EDL-PWM 1), 'Translation Name' (PWM 1), 'Unit of measure' (Duty), and 'Filter Level 0'.
- Data Translation Section:** Includes radio buttons for 'Calculation' (selected), 'Translation Table', and 'Binary Bit Position'.
- Calculated Translations Section:** Contains a table with columns for 'Operation' and 'Value'. The 'Operation' dropdown is set to 'No Calculation' and the 'Value' field contains '0'.

Define a channel without a translation method

Select the channel you want to define using the drop-down selection. Type a meaningful name and unit of measure. Click "Calculation" in the Data Translation box. This will show the "Calculated Translations" definition screen. Select "No Calculation" from the operation drop-down field. You can leave the value field because it will not be used.

Defining a channel that uses a direct calculation

Select the channel you want to define using the drop-down selection. Type a meaningful name and unit of measure. Click "Calculation" in the Data Translation box. This will show the "Calculated Translations" definition screen.

There are four calculation formulas you can use to translate raw data into a meaningful value.

1. Channel plus value
2. Channel minus value
3. Channel multiply by value
4. Channel divided by value

Each one of these will use the raw data and apply one of the formulas with the value entered into the value field.

IE: If you want to translate a raw 0 - 5000 mV analogue to digital number into millivolts then you would select "Channel divide by value" and then type "0.2046" into the value field. (If the AtoD channel is a 10 bit channel).

You can measure engine RPM by connecting one of the pulse inputs to one of the coil igniter trigger wires.

To convert frequency from one of the pulse inputs into engine RPM.

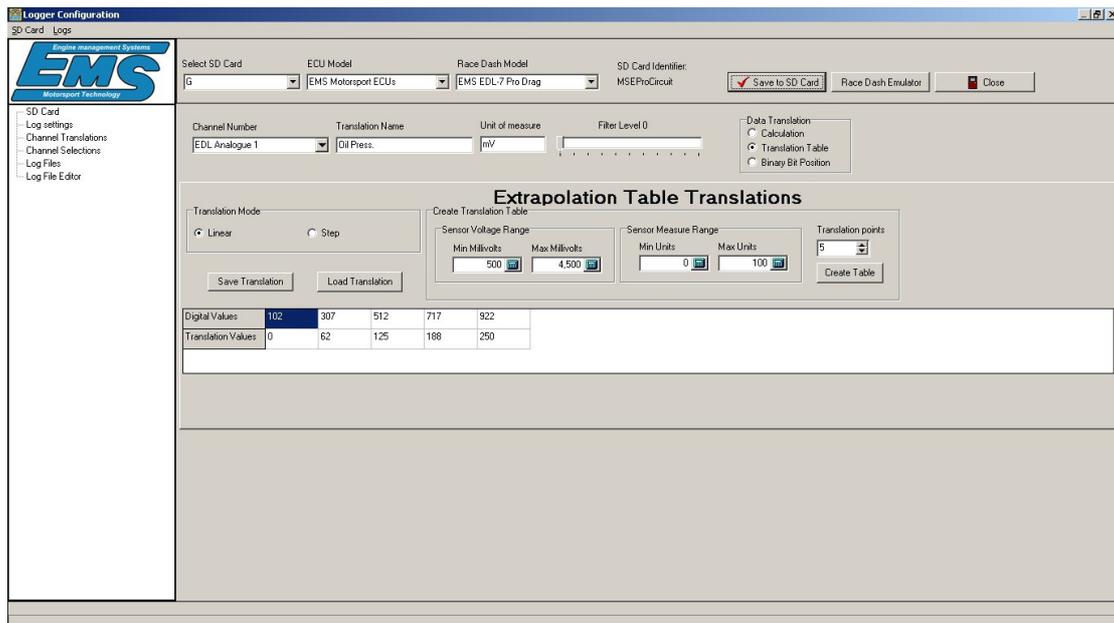
For 4 stroke engines then the formula is $120 / \text{number of pulses per } 720 \text{ degrees} * \text{channel frequency}$. IE: if you have a 4 cylinder with four igniters then the formula is $120 / 1 * \text{channel frequency}$. In this case the value you should use in the calculation is $120 / 1 = \mathbf{120}$. If you have a 4 cylinder with distributor and one igniter then the formula is $120 / 4 * \text{channel frequency}$. In this case the value you should use in the calculation is $120 / 4 = \mathbf{30}$.

Filtering channel data

There are times when you need to smooth out the incoming data from a sensor. This would be useful when the incoming data is too erratic such as the signals from a fuel level sensor.

Use the filter level slider to apply filtering to the incoming data. There are 11 levels of filtering. 0 = no filtering and 10 = most filtering. Move the slider to the filter level you want to achieve.

Creating Translation Tables



Use this method of translation for channels that need a complicated formula or have a non-linear coefficient.

Select the channel you want to define using the drop-down selection. Type a meaningful name and unit of measure, these will appear on the gauge when the gauge channel. Click **“Translation Table”** in the **“Data Translation”** box to show the **“Extrapolation Translation Tables”** definition screen.

There are 2 translation modes use can select from, Extrapolation or Step.

Extrapolation mode will allow you to define a table for a sensor that does not have a linear coefficient. You can define up to 70 translation points and enter a translation value for each point. The Race Dash will perform a linear extrapolation between each point to calculate an exact data value.

Step mode will allow you to define a table with up to 70 translation points. But instead of performing an extrapolation between each point to calculate an exact value it will use the entered value directly. This is useful if you are using a voltage to determine the selected gear.

To create a translation table you need 3 parameters, the number of translation points, the sensor voltage range and the sensor unit of measure range. The number of translation points should be kept to a minimum. If the sensor is linear then you only need 3 – 5 translation points. If the sensor is non-linear then you can use up to 70 translation points.

To create a translation table for a sensor that is rated to produce a voltage from 500 mV (0.5V) to 4500 mV (4.5V) and gives a PSI unit of measure from 0 to 150 PSI. First, enter 500 in the **“Min Millivolts”** field, enter 4500 in the **“Max Millivolts”** field, enter 0 (zero) in the **“Min Units”**, enter 150 in the

“Max Units”, Enter 5 in the **“Translation points”** field and press the **“Create Table”** button. For a non-linear sensor you can use this method to create an initial table to start with then adjust the **“Translation Value”** row of values according to the sensor’s ratings.

The EDL-7 Race Dash will interpolate between translation points to produce a smooth gauge reading and data log value.

Linear translation

This method of translation interpolates between translation points to produce a linear value.

Step translation

This method of translation does not interpolate between translation points. It looks for the first digital point that is less than the data channel value and produces the value set for the particular point. This is useful for displaying a gear number if you are using a variable voltage to indicate the currently selected gear.

Loading and Saving Translation Tables

Once you have defined a translation table of a particular sensor you can save it and reload it for another channel.

To save the table press the “Save Translation” button. This will display the Save screen where you can leave the default folder location or select a different one. Enter a file name for the table and press the “Save” button to save the new file.

To load a saved translation table press the “Load Translation” button. This will show the open screen where you can select a table from the default folder location or select a different location. Select the table you need and press the “Open” button.

Translating EMS Motorsport and Stinger ECU data

The data streaming from an ECU may need some translating for 2 reasons.

1. The data may be untranslated A to D numbers.
2. The data may need to be converted into a different unit of measure such as from miles per hour to kilometres per hour or pounds per square inch to kilo-pascals.

ECU RPM	Does not require translation
ECU Load.	Use translation table provided by EMS. The actual formula is (Channel Value - 200) / 2. This will give you KPA.
ECU TPS	To read actual TPS percentage divide by 10.
ECU Air/Fuel Ratio for sensor 1	To record actual A/F ratio divide by 10. If you want to record lambda then divide the value by 147.
ECU Air/Fuel Ratio for sensor 2. (not available in Stinger ECUs)	To record actual A/F ratio divide by 10. If you want to record lambda then divide the value by 147.
ECU Air temperature.	To record actual air temp subtract 20. This will give temperature in degrees Celsius.
ECU Engine temperature.	To record actual water temp subtract 20. This will give temperature in degrees Celsius
ECU Ignition Timing.	To record actual degrees multiply by 0.2. This will give degrees BTDC.
ECU Injection milliseconds.	To record actual milliseconds multiply by 0.002
ECU Ethanol percentage in fuel (not available in Stinger ECUs)	If flex fuel sensor is connected. This value is actual percentage and there are no needed calculations.
ECU Fuel Temperature.	This value is actual degrees Celsius.
ECU Battery voltage.	To record actual voltage divide the value by 10.
ECU end of injection angle. (not available in Stinger ECUs)	To record degrees BTDC multiply this value by 2.823.

<p>ECU Analogues 1 – 9 (not available in Stinger ECUs)</p>	<p>These channels provide an A to D number from 0 to 1024. This represents a voltage from 0 to 5 volts (0 - 5000 millivolts). These translations are completely user definable.</p>
<p>ECU Aux Outputs 1 – 4</p>	<p>These channels provide an A to D number from 0 to 255. This represents a pulse width duty from 0 to 100 percent in 255 increments. To record actual percentage multiply by 0.392.</p>
<p>ECU Aux Outputs 5 – 8 (not available in Stinger ECUs)</p>	<p>These channels provide an A to D number from 0 to 255. These outputs are ether on or off. On being 255 and off being 0.</p>

Translating MoTec M800 ECU Data

ECU RPM	No Translation Required
ECU TPS	Actual TPS percentage in 0.1% increments
ECU Man. Press.	Actual manifold pressure in KPA in 0.1Kpa increments.
ECU Air Temp.	Actual temperature in degree Celsius in 0.1 degree increments
ECU Eng. Temp.	Actual temperature in degree Celsius in 0.1 degree increments
ECU Lambda 1	Actual Lambda in 0.001La increments
ECU Lambda 2	Actual Lambda in 0.001La increments
ECU Exh. Man. Press	Actual exhaust manifold pressure in KPA in 0.1Kpa increments.
ECU Fuel Temp.	Actual fuel temp in degrees Celsius in 0.1 degree increments
ECU Fuel Press.	Actual fuel pressure in KPA in 0.1Kpa increments.
ECU Oil Temp.	Actual oil temp in degrees Celsius in 0.1 degree increments
ECU Oil Press.	Actual oil pressure in KPA in 0.1Kpa increments.
ECU Knock Volts	Actual voltage in 0.01V increments
ECU Gear Sft Force	Actual force in 0.1Kg increments
ECU User Channel 1 - 4	User defined in increments of 0.1
ECU Battery Volts	Actual voltage in increments of 0.01V
ECU Digital Speed 1 - 4	Actual speed in 0.1km/h increments
ECU Drive Speed	Actual speed in 0.1km/h increments
ECU Ground Speed	Actual speed in 0.1km/h increments
ECU Slip	Actual slip in 0.1km/h increments
ECU Ign. Advance	Actual timing in 0.1 Deg BTDC
ECU Load Point	Actual load point in 0.1 increments
ECU Efficiency Point	Actual load point in 0.1 increments
ECU Fuel Used	User definable in increments of 1.0
ECU Inj. Act. Pulse Width	Actual pulse width in 0.5us increments
ECU Inj. Duty Cycle	Actual inj. Duty in 0.1% increments
ECU Gear	Divide by 10 = gear
ECU Diag. Err. Group 6	1 = Low Battery error 2 = Over Boost error 4 = No Sync signal error 8 = Sync error 16 = No Ref signal error 32 = Ref error 64 = Over rev error 128 = inj max duty error

Special Log fields

There are certain log fields in each log record that are used internally by the analyser software to determine end of lap and end of sector records. These can also be used in the maths definition program scripting.

Lap_Marker:

This field normally has a value of 0 (zero) but a value of 1 (one) in this marks the end of one lap and start of a new lap. This record also contains the lap and sector times for the lap and sector just completed.

Sector_Marker:

This field normally has a value of 0 (zero) but a value of 1 (one) in this field marks the end of one sector and the start of a new sector. This record also contains the sector time for the sector that was just completed.

Lap_Difference:

This field normally has a value of 0 (zero) but whenever the lap_marker has a value of 1 then the Lap_Difference field contains the difference between the actual lap time and the targeted lap time.

Sector_Difference:

This field normally has a value of 0 (zero) but whenever the sector_marker has a value of 1 then the Sector_Difference field contains the difference between the actual sector time and the targeted sector time.

Lap_Number:

This field holds the lap number of the recorded log. This value is recorded on every log record and is incremented each time a new lap is detected.

Lap_Counter:

This is the same as the Lap_Number but it can be re-set during the race session. This is useful for keeping track of the number of laps since the last fuel refill.

Track_Time:

This holds the value of the running timer. The time starts when logging starts and is not used for lap time recording.

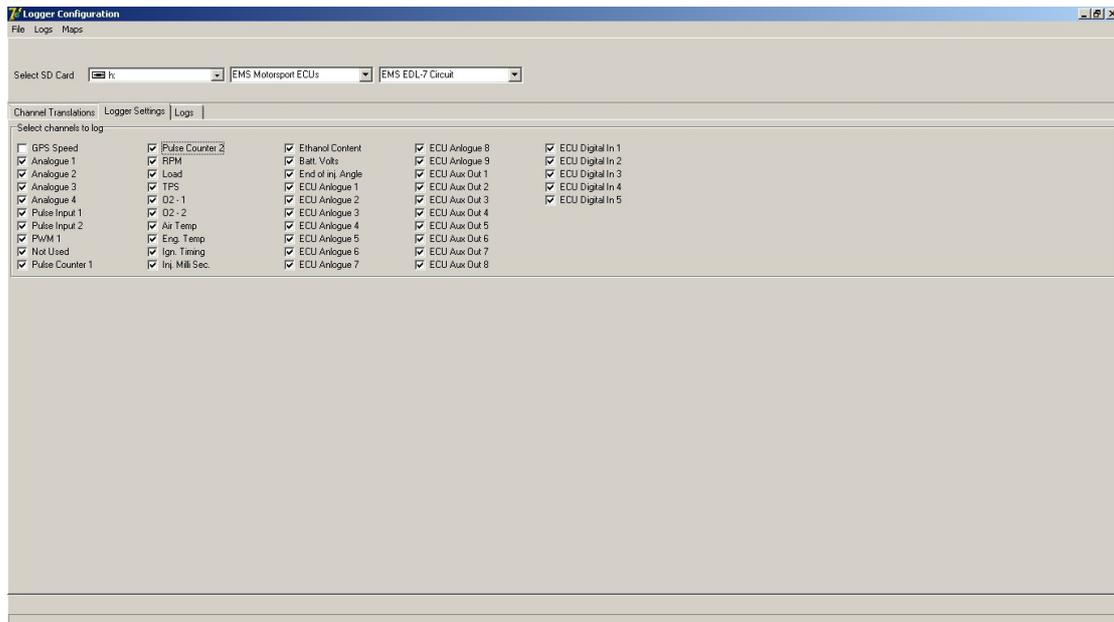
Lap_Distance:

This records the distance travelled from the start line. Each time the start / finish line is passed the distance is reset to zero.

Selecting channels to log

The Race Dash can log up to 48 channels but you can select to log only the channels you need. Select the channel you want to log by clicking each channel name to place a tick in the tick box.

The channel names that appear here are names defined in the translation screen. These names will remain in the data log files and imported into the analysis software.



EMS Maths Definition Language

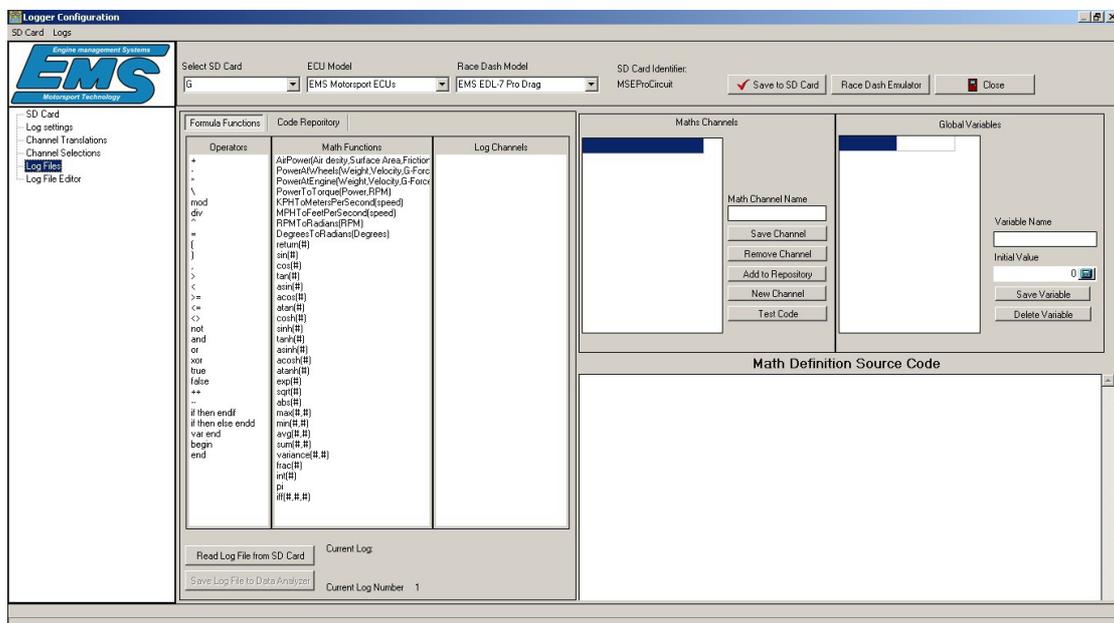
Downloading log files from the SD card

In order for the analyser to read log files they must be extracted from the SD Card. At this point you can apply any maths channels that were created.

To import a log file from the SD Card go to the **Maths definition Editor** screen and press the **“Read log file from SD Card”** button. This will display all saved log files in the selected SD Card. Choose a log file and press the **Open** button. This will load all of the logged channels so you can then use them in the EMS Maths Definition Language. After reading the log file you must save it into the Data Analysis file system. When you select to save the imported log file, any defined maths channels will be added to the file. The file will be saved to the correct folder location so you can open it via the Data Analysis software. The folder location will always match the SD Card Identification.

To save an imported log file go to press the **“Save log file to Data Analyser”** button. This will display the save dialogue screen where you can leave the default file name or change it then press the **Save** button.

Please note that each SD card will have its own folder where all logs will be saved. The folder will be assigned automatically based on the SD Card identifier. This will help to organise the files from many Race Dashes and SD cards.



EMS MDL Syntax

The EMS Data analysis software gives you a sophisticated Maths Definition Language to describe complicated maths channel formulas. The system allows you to describe up to 128 maths channels, 128 global variables and also local variables.

The Maths channel programs can be used to modify an existing log data channel or create new data channels. When naming a maths channel, if the name matches exactly with an existing log channel then that channel will be modified. If no exact match exists then the maths channel will be added as a new data channel.

The Global variables can be used to hold and accumulate values that can be shared and changed by all maths channel programs. The global variables can also be used to for holding constant values such as gear ratios.

The local variables are used to hold values used internally within each maths channel program.

Types and Expressions

Numeric literals (integer and real) are be represented in common decimal notation.

Operators

Numeric operators:

- + Add
- Subtract
- * Multiply
- / Divide
- % Modulus
- \ Integer divide
- ^ raise to a power

Comparison operators:

- = is equal to
- <> is not equal to
- > is greater than
- < is less than
- >= is greater then or equal to
- <= is less than or equal to

Constants

Constants can be defined as global variables with an initial value.

Predefined functions

EMS MDL has a wide number of predefined functions. These can be used in simple single line formulas and also in complicated programmed formulas.

Numeric functions:

Trigonometric, hyperbolic and invert functions:

$\sin(x)$, $\cos(x)$, $\tan(x)$, $\text{asin}(x)$, $\text{acos}(x)$, $\text{atan}(x)$, $\text{cosh}(x)$, $\text{sinh}(x)$, $\text{tanh}(x)$,
 $\text{asinh}(x)$, $\text{acosh}(x)$, $\text{atanh}(x)$.

Exponential and logarithmic functions:

$\log(x)$, gives the log base 10 of a number
 $\text{sqrt}(x)$, give the square root of a number

Integer and factional parts of a number:

$\text{Int}(x)$, gives the value to the left of the decimal point.
 $\text{frac}(x)$, gives the value to the right of the decimal point

Statistical functions

All statistical functions take variable a argument list

$\text{max}(x1,x2,...)$ - maximum of passed values,

$\text{min}(x1,x2,...)$ - minimum of passed values,

$\text{avg}(x1,x2,...)$ - average value,

$\text{stddev}(x1,x2,...)$ - standard deviation,

$\text{sum}(x1,x2,...)$ - sum of passed values,

$\text{count}(x1,x2,...)$ - count of passed values,

$\text{variance}(x1,x2,...)$ – variance.

Logical functions:

$\text{iff}(e,x,y)$ - checks the expression passed as e and returns x if it evaluates to true, or y if it evaluates to false.

Specialised Functions

Although you can use the EMS Maths Definition Language to write these formulas. EMS has included them as special functions. As new product versions are being developed EMS will add more special functions.

AirPower – is used to calculate the amount of power generated to push a car through the air.

PowerAtWheels – is use to calculate the power developing at the wheels.

PowerAtEngine – is used to calculate the power being developed at the flywheel.

PowerToTorqure – is used to convert power into torque.

KPHToMetersPerSecond – is used to convert kilometres per hour to meters per second.

MPHToFeetPerSecond – is used to convert mile per hour to feet per second.

RPMToRadians – is used to convert reves per minute to radians per second.

DegreesToRadians – is use convert degrees to radians.

pi – is the value of the Pythagorean PI (3.1415926535)

Variables

The EMS Math Description Language has two kinds of variables, Global and Local.

Global Variables

Global variables are defined in the variable list and can be assigned an initial value. The value can be altered by a maths channel formula and then used in another maths channel formula. You reference global variables by name with a \$ sign added to the front of the name. IE: a global variable named HP will be referenced in the EMS Maths Definition Language as \$HP. The Global variables are automatically saved onto the SD card and become available whenever the SD card is used.

Local Variables

Local variables are accessible only within the “begin/end” block statements. They are defined as follows:

```
begin  
  var "name1", "name2", "name3" end;  
  program code to perform calculations  
end
```

Local variables are used to temporarily hold calculated values so the value can be used elsewhere in the program code.

To get the value from variable “name1” and copy it into global variable HP use the following syntax:

\$HP := \$name1. This will copy the value of name1 to HP.
The value of HP can now be used in another maths channel program.

To get the value of variable “name1” and use it in a maths function such as sin(#) and the save the resulting value to global variable HP use the following syntax.

\$HP := Sin(\$name1); This will save the resulting value to HP
If the value of name1 was 30 then the value of HP will be 0.5.

To set the value of a variable use the following syntax:

\$name := 1.5; This will set the value of name1 to 1.5.

You can also increase or decrease the value of a variable by one (1) using the following syntax:

Increase the current value by 1
\$name++

Decrease the current value by 1
\$name--

Programming syntax

Using predefined functions

If you need a to apply a single line formula to a log channel then you can simply write the formula as normal. There is no need to use the “begin/end” block statements IE:

Max(wheelspeed1,wheelspeed2,wheelspeed3,wheelspeed4)
OR
Avg(wheelspeed1,wheelspeed2,wheelspeed3,wheelspeed4) + 20.0
OR
15 + (20 * 100)

The first example will return the maximum of each of the wheel speed log channels. The second example will return the average of each wheel speed channel then add 20. The third example will return the result of the formula. Notice, that there is no need to end each statement with the “;” symbol.

Return(value):

This is a very important statement when writing program code blocks, if you wrote a single line formula then it is not needed. It is used to return the calculated value to the maths channel. It stops program execution and gives the maths channel a value at the same time. If not used then the maths channel will have unpredictable values.

Return(1.5); will stop program execution and return the value 1.5 to the maths channel.

Return(\$name1); will stop program execution and return the value of variable “name1” to the maths channel.

Block statements

Begin and End statements are used to define a program block. When writing a complicated formula that requires many lines of code then you must start the code with a “**begin**” statement and end it with an “**end**” statement. You would use a program block if you need to perform a complicated calculation. Each statement in the block is performed in the sequence it is written. E1:

Begin

```
Var "V1", "V2" end;  
$V1 := 2.5;  
$V2 := $V1 * 3.7;  
Return($V2);
```

End

The example above defines 2 variables (V1 and V2).

Sets the value of V1 to 2.5.

Sets the value of V2 to be $2.5 * 3.7$

Then returns the resultant value of V2 to the maths channel.

There are three things to notice here:

1. The “;” symbol at the end of each statement but not at the end of the Begin and End. This defines the end of each statement.
2. When assigning a value to a variable the “=” sign is always preceded by “:” symbol. This is to distinguish it from the comparison form of “Is equal to”.
3. The word “Var” denotes that you are defining a list of local variables and the “end” statement denotes completion of the list.

If else statements

The “if (*comparison*) then (*do this*) else (*do that*) programming statement allows you to perform a different block of statements depending on a value from a previous calculation or value. The (*comparison*) value must evaluate to either true or false. IE: (1<2) evaluates to true and (1>2) evaluates to false.

The syntax is as follows:

```
begin  
  if ($V1 < 100.00) then return($V1);  
  else return($V2); endd;  
end
```

OR

```
begin  
  if ($V1 < 100.00) then return($V1); endif;  
end
```

OR

```
begin  
  if ($GV1 <= 20) then  
    $V1 := $GV1 * $V2;  
  else  
    $V1 := $GV1 * ($V2 + 20);  
  endd;  
  return($V1);  
end
```

Notice that if the “if” statement also includes the “else” statement then it must end with “endd;”. If it does not include an “else” statement then it must end with “endif”. This rule is important and will cause a syntax error if not observed correctly.

The “if” statement examples above are described as follows:

1. if the local variable V1 is less than 100 then return its value to the maths channel else return the value of local variable V2.
2. if the local variable V1 is less than 100 then return its value to the maths channel. If the block of statements inside the “if/endif” block is not executed then a zero result will be returned to the maths channel.
3. if global variable GV1 is less than or equal to 20 then assign local variable V1 with the value of global variable GV1 multiplied by the value of local variable V2 else assign the local variable V1 with the value in global variable GV1 multiplied by the sum of the value in local variable V2 and 20. Then return the resultant value of V1 to the maths channel.

You can join multiple comparison statements in an “**if then else**” statement by using the “**and**” or “**or**” words.

The syntax is as follows:

```
begin  
  if ($V1 < 100.00) and ($V2 > 110.00) then return($V1);  
  else return($V2); endd;  
end
```

The Code above will execute the **return(\$V1)** statement only if both comparison statements evaluate to true else the return(\$V2) statement will be executed.

OR

```
begin  
  if ($V1 < 100.00) or ($V2 > 110.00) then return($V1);  
  else return($V2); endd;  
end
```

The Code above will execute the **return(\$V1)** statement if either comparison statements evaluates to true else the return(\$V2) statement will be executed.

You are not restricted to one-line statements between the if/else and else/ endd or if/endif. You can place multiple lines of code in the same way as placing multiple lines of code between the begin/end statements. Don't forget the “;” symbol at the end of each line.

Saving MDL programs

MDL programs are automatically save onto the SD Card whenever you press the “**Save Channel**” button. However, they can also be saved to the Code Repository and reused on other SD Cards.

Once you have created the maths channel program you can save it by pressing the “**Add to Repository**” button. This will save the program you’ve created in to the Code Repository. It will be given the same name as the “**Maths Channel Name**”. If the name already exists in the repository then it will be overwritten. Global variables are not saved in the repository, they are saved onto the SD Card. Remember this when re-using code from the repository for different SD Cards.

Testing your code

If you have defined any maths channels, then before you save the log file we recommend that you test your program. If there is any error in your code then the new maths channel will have value of zero. If you have errors then you must either delete the offending maths channel or fix it. If you need to fix the error at a later time then you can save it to the repository first then delete it from the SD Card.

Press the “**Test Code**” button to test your program. If you are using any global variables or log channels in the program then you will be prompted to enter values for these fields. This will help ensure that your formulas are correct. If there are no errors then you will be shown the result. If there are errors then a message will be displayed describing the error and place the cursor near the invalid code.

Some errors may be a little vague. In these case the error is normally caused by incorrect syntax. You should look for mistakes where you have not ended a statement with the ‘;’ symbol or placed this symbol on a line that should not have it.

For Example:

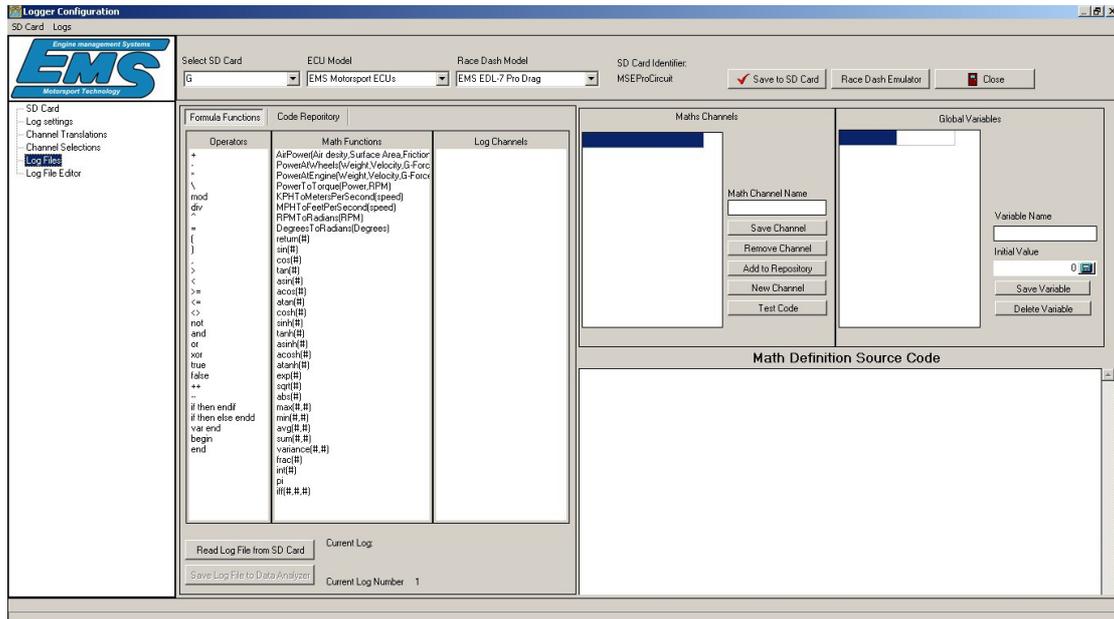
Placing this symbol at the end of a **Begin** or **End** statement will cause the “**syntax error**” message to show. The exception to this is if the **end** statement is ending a variable definition block of code. In this case you must place the “;” after the end statement.

Placing this symbol at the end of an else statement will cause the “**) expected**” message to show.

Placing this symbol at the end of a **then** statement will cause the “**Not Enough Parameters**” message to show.

NOT putting the “;” symbol at the end of an **endif** or **endd** statement will cause the “**) Expected**” message to show.

MDL Editor



Operators

This list contains all of the available language syntax. Double click on one of the entries to insert it into the Maths Definition Source Code area.

Math Functions

This list contains all of the available maths functions. Double click on one of the entries to insert it into the Maths Definition Source Code area. The “#” means that you must place a numeric value, a global or local variable or a log channel between the parenthesis “()”.

Log Channels

This list contains all of the available data log channels after you have imported a log from the SD Card. Double click on one of the entries to insert it into the Maths Definition Source Code area. If a “#” is found in the source code area then it will be replaced by the selected log channel.

Maths Channels

This list contains all of the maths channels you have defined. To define a Maths channel you first enter a maths channel name and then write the programme code in the maths definition language source code section. To add newly created maths channel to the log file you must press the “**Save Channel**” button. To delete a maths channel press the “**Remove Channel**” buttons. This will remove the channel from the SD card but it will still remain in the repository if it was previously saved there.

Add to Repository

After you have created a maths channel program you can save it to the **Code Repository** for later use.

New Channel

Press this button to clear the Maths Definition Source Code section and Maths Channel Name.

Global Variables

This is a list of global variables you have previously defined. Double click on one of the entries to insert it into the Maths Definition Source Code area.

To define a global variable you simply enter a name in the **Variable Name** field, give it an initial value if you need else leave it as zero and then press the **Save Variable** button.

Code Repository

This is a list of previously created maths channel programmes that you added to the repository. To use the code simply select the program you want to use and then press the **Use Selected Code** button. To delete the selected code from the repository press the **Delete Selected Code** button

Dash Emulator

The Dash Emulator is a working version of the Race Dash Firmware on your laptop. It behaves just like the actual race dash. This feature becomes active whenever a valid SD Card is detected.

Use this feature to modify the Race Dash settings directly from your laptop. You can use it to change gauge settings, logger settings, PWM settings, Etc. Any changes made with Dash Emulator are saved to the SD Card and will become active once the SD card is inserted into the Race Dash.



Setting Gauge Properties

The EDL Dash Emulator can set the gauge titles and other properties. Place the mouse pointer on the gauge you want to modify and press the right mouse button. Select the Gauge Properties option to display the property screen for the selected gauge

Repositioning and resizing Gauges

You can reposition and resize any gauge using the EDL Dash Emulator. Place the mouse pointer on the gauge you want to modify and press the right mouse button and select the Move/Resize/Lock option. This will show a window border around the gauge. Use this top border to drag and side borders to resize the gauge. To lock the gauge position and size press the right mouse button again and select the Move/Resize/Lock option.

Removing Gauges

To remove a gauge place the mouse pointer on the gauge you want to remove and press the right mouse button and select the Remove Gauge option.

Adding Gauges to the Main Dash Screen

To add new gauges place the mouse pointer on an area of the screen that is clear of any gauge and press the right mouse button. A menu will be shown allowing for the selection of various gauges. Select the desired gauge to add to the screen. You can now resize and position it and also set its properties. The gauge selection menu is split into 2 groups. The first group is used to add gauges that read the input data channels. The second group is used to add gauges that read data from the lap timer channels.

Adding gauges to the second screen

Adding gauges to the second screen is the same as adding to the main screen. From the main screen touch the “**next**” icon located on the bottom right of the main screen. This will show the second screen where you can add a customise gauges in the same way as the main screen. When finished, press the “**back**” icon.

Dial gauge

Dial Gauge Properties

Channel: Boost

Close

Appearance

Gauge Title	Boost
Minimum Value	-15
Maximum Value	20
Number Formatting	###0.00
Start Danger Zone	15
End Danger Zone	20
Divider	1
Number of Labels	7
Number of Ticks	70
Layer Order	2

Colours

Danger	cRed
Gauge Face	cBlack
Background	cBlack
Text	cWhite

Show Danger Zone

Data Channel

Select the data channel you want displayed on this gauge.

Minimum Value and Maximum Value

These values set the minimum and maximum number span for the gauge. IE if this gauge is used to show Exhaust gas temperature then you will set the max value to 1200, min value to 0, max danger zone to 1200 and min danger zone to 750.

Number formatting

This describes the numerical formatting for text values.

= Do not show leading zeros.

0 = Show leading zeros

. = Decimal place

This format “###0.00” will show numbers with 3 significant digits and decimal digits. The “#” symbols will only show numbers that are greater than zero.

Start and End Danger Zone

These numbers determine where to start and stop the danger zone for the gauge.

Divider

This allows you to show the full value in the text but a divided value on the gauge. This is useful when displaying RPM. If you want to show RPM in thousands (1 = 1000, 5 = 5000) then you set the divider to 1000, the Max Value to a value greater than your engine maximum RPM and the Min Value to 0.

IE: If you want the gauge to show a maximum of 8 for 8000 RPM then you would set the gauge parameters Max Value to 8000 and the divider to 1000. This will draw a gauge with numbers from 0 to 8 that represent 0 to 8000 RPM.

Number of Labels

This number of major marks visible on the gauge. Set this value to a number that divides evenly into the gauge span. This will provide an even range of numbers around the gauge.

To calculate the gauge span then subtract the gauge minimum from the gauge maximum. IE: if the gauge is used to show manifold pressure in KPA from -100 to +100 KPA then the gauge span is $+100 - -100 = 200$.

Number of Ticks

Set this number to a value that the Number of Marks can divide evenly into. This will give you an even number of small ticks between each mark.

Number of digits

This refers to the text value that is shown on the bottom of the gauge. Set this to a value that can accommodate the maximum value of the channel plus a decimal point and the decimal precision.

Decimal places

This refers to the precision of the value being shown. If you only want to show whole numbers then set it to zero. If you want an accuracy of 100ths then set this value to 2.

Gauge Title

The text displayed on each gauge depends on the channel selected. This is set when you define the channel translation table. The gauge will display the Translation name and Unit of Measure (UOM) defined using the EDL7 Dash Manager option found in the Analysis Software.

Layer Order

This controls whether a gauge is shown on top of another. A gauge with a higher number is shown on top of a gauge with a lower number. This allows you to place gauges in positions that partly obscure another gauge.

Danger Colour

This allows you to choose the colour you want to display in the danger zone.

Gauge Face Colour

This is the colour for the round dial background colour.

Background Colour

This is the colour for the area of gauge outside of the round dial face.

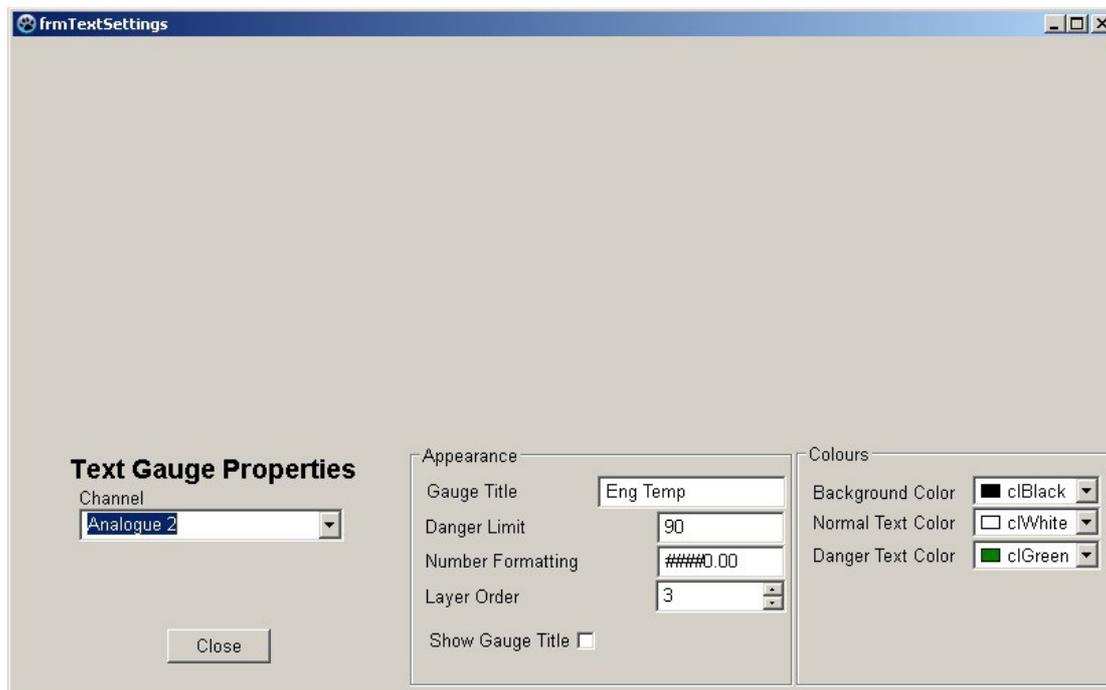
Text Colour

This is the colour of the text and ticks of the gauge.

Show Danger Zone

Tick this field if you want to show the danger zone.

Text Gauge



Data Channel

Select the data channel you want displayed on this gauge.

Gauge Title

The title text displayed on each gauge. If this is left blank then the channel name will be displayed

Layer Order

This controls whether a gauge is shown on top of another. A gauge with a higher number is show on top of a gauge with a lower number. This allows you place gauges in positions that partly obscure another gauge.

Danger Limit

This allows you to display a different colour for values below this value. See normal colour and danger colour properties.

Number formatting

This describes the numerical formatting for text values.

= Do not show leading zeros.

0 = Show leading zeros

. = Decimal place

This format "**###0.00**" will show numbers with 3 significant digits and decimal digits. The "**#**" symbols will only show numbers that are greater than zero

Danger Text Colour

The gauge will change to the selected colour when its value is above the danger limit.

Normal Text Colour

The gauge will displayed in this colour when its value is below the danger limit.

Background Colour

This is the colour for the area behind the displayed value.

Show gauge tile

Tick this field if you want to show the gauge title.

Segment Gauge

The screenshot shows a dialog box titled "frmNumberSettings" with a "Segment Gauge Properties" section. It contains three main panels: "Appearance", "Colours", and a "Channel" dropdown. The "Appearance" panel has fields for "Gauge" (Eng Temp), "Number of Digits" (5), "Number of Decimals" (2), "Danger Limit" (90.00), and "Layer Order" (3). The "Colours" panel has dropdowns for "Background Color" (cBlack), "Text Color" (cWhite), "Normal Color" (cLime), and "Danger Color" (cGreen), along with a "Show Gauge Title" checkbox.

Property	Value
Channel	Analogue 1
Gauge	Eng Temp
Number of Digits	5
Number of Decimals	2
Danger Limit	90.00
Layer Order	3
Background Color	cBlack
Text Color	cWhite
Normal Color	cLime
Danger Color	cGreen
Show Gauge Title	<input type="checkbox"/>

Data Channel

Select the data channel you want displayed on this gauge.

Number of digits

This refers to the text value that is show on the bottom of the gauge. Set this to a value that can accommodate the maximum value of the channel plus a decimal point and the decimal precision.

Number Decimals

This refers to the precision of the value being shown. If you only want to show whole numbers then set it to zero. If you want an accuracy of 100ths then set this value to 2.

Gauge Title

The title text displayed on each gauge. If this is left blank then the channel name will be displayed

Layer Order

This controls whether a gauge is shown on top of another. A gauge with a higher number is show on top of a gauge with a lower number. This allows you place gauges in positions that partly obscure another gauge.

Danger Limit

This allows you to display a different colour for values below this value. See normal colour and danger colour properties.

Danger Colour

The gauge will change to the selected colour when its value is above the danger limit.

Normal Colour

The gauge will be displayed in this colour when its value is below the danger limit.

Background Colour

This is the colour for the area behind the displayed value.

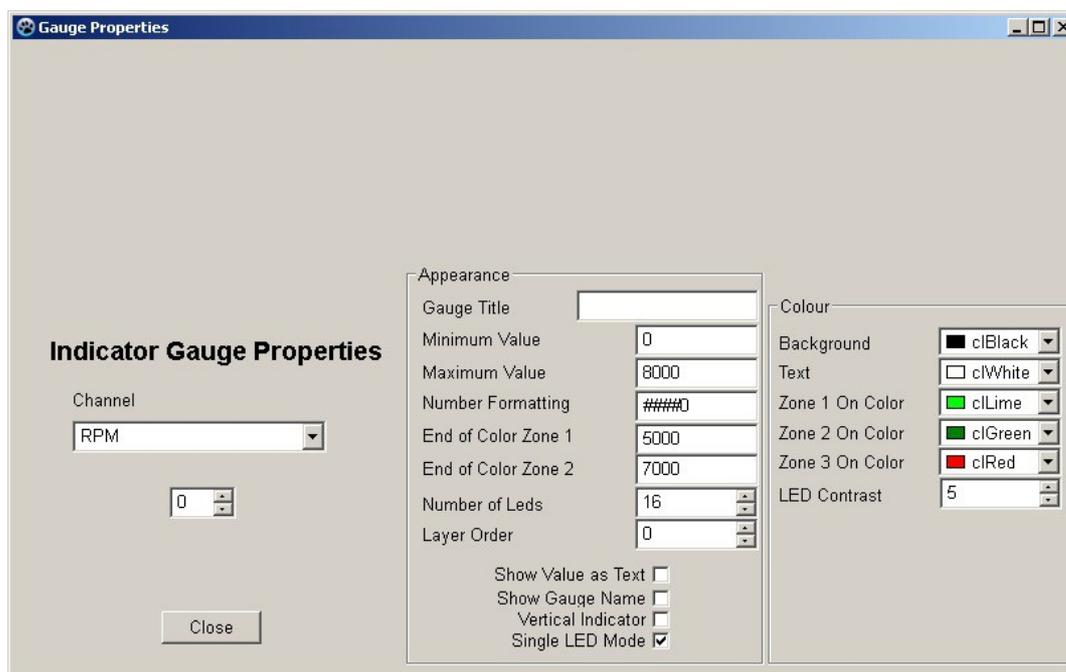
Text Colour

This is the colour of the gauge title text.

Show gauge tile

Tick this field if you want to show the gauge title.

Indicator gauge



Data Channel

Select the data channel you want displayed on this gauge.

Minimum Value and Maximum Value

These values set the minimum and maximum number span for the gauge. IE if this gauge is used to show Exhaust gas temperature then you will set the max value to 1200, min value to 0.

Number formatting

This describes the numerical formatting for text values.

= Do not show leading zeros.

0 = Show leading zeros

. = Decimal place

This format “##0.00” will show numbers with 3 significant digits and decimal digits. The “#” symbols will only show numbers that are greater than zero.

End of colour zone 1

The indicator gauge can display 3 colour zones. Set the value for the end of zone one. Gauge values below this value will be shown using the “**Zone 1 On Colour**”

End of colour zone 2

Set the value for the end of zone two. Gauge values below this value and above zone one value will be shown using the “**Zone 2 On Colour**”. Any values above the “End of colour zone 2” will be shown using the “**Zone 3 On Colour**”.

Number of Leds

This gauge shows a series of LEDs that span from left to right or bottom to top. Select the number of LEDs to use in span. More LEDs give a finer gauge. If you select too many compared to the length of the gauge then they will not be displayed

Gauge Title

The text displayed on each gauge depends on the channel selected. This is set when you define the channel translation table. The gauge will display the Translation name and Unit of Measure (UOM) defined using the EDL7 Dash Manager option found in the Analysis Software.

Layer Order

This controls whether a gauge is shown on top of another. A gauge with a higher number is show on top of a gauge with a lower number. This allows you place gauges in positions that partly obscure another gauge.

Background colour

This is the colour for the area of gauge outside of the round dial face.

Text Colour

This is the colour of the text and ticks of the gauge.

Show Value as text

Tick this field if you want to show the gauge value as text just above the gauge.

Show Gauge name

Tick this field if you want to show the gauge title just above the gauge.

Vertical indicator

This gauge can span from left to right or bottom to top. To make the gauge span from bottom to top then tick this property.

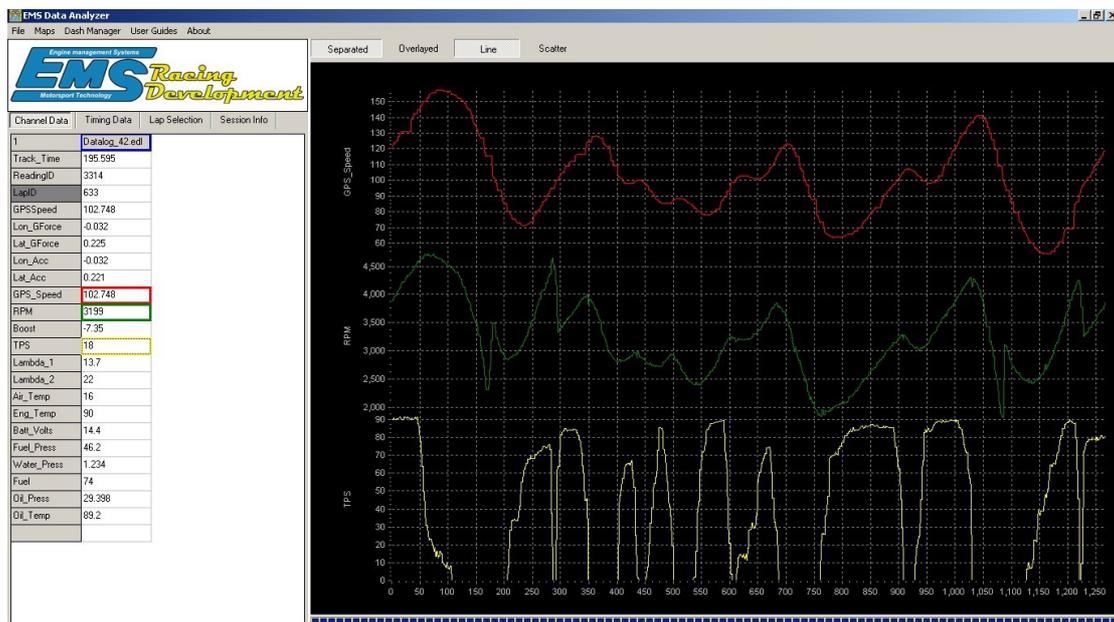
LED Contrast

This is a number between 5 and 10 that controls the LED brightness difference between ON and OFF. A value of 5 will show the OFF LED brightness at half the level of the ON LED brightness. A value of 9 will show the LED at 9 times less bright and a value of 10 will completely turn the LED OFF.

Data Analyser

The Data Analyser allows you to view log data in a graphical form. You can view multiple log files together so you can compare two or more sessions. You can also open a session file as separate laps so you can compare different laps from the same session.

When you select channels to display, the software will automatically display channels with the same name together in the same graph. This allows you to overlay similar data for easier comparison. The channel data is also shown in text form on the left side of the screen. As you move the mouse pointer over chart the text data changes to show the numeric value for each channel in the log pointed to by the mouse cursor.



Opening a Log File

You can open a log file in two ways. You can open it as a whole session or as separate laps.

To open an imported log file as a session, go to menu option File -> Open Log As Session. Then select the log you want to view and press the Open button. This will create a column of channel values. You can open a second log file in the same way. The system will create a second column for the newly opened log.

To open an imported log as separate laps, go to menu option File -> Open Log as Separate Laps. This will open the log with separate data columns for each lap. You can then choose to compare data from different laps on the same chart.

The amount of time to open a log file will depend on the amount of data recorded in the file.

Channel Data

The Channel Data section displays data in a textual way and also provides a way to select individual channels for displaying and sorting.

Selecting a channel to display.

You can select as many channels as can reasonably fit onto the chart screen. If you are analysing multiple logs or multiple laps, the system will overlay data channels with the same name. If you select channels with different names then the software will create a new chart for each channel.

To display data for a channel first click on the desired channel then press the right mouse button. This will show a pop-up menu where you must select the Add to Chart option. Do this for each channel you want to view.

Remove a data channel from the chart

To remove a data channel from the chart, first click on the channel you want to remove then press the right mouse button to display the pop-up screen, now select the Remove From Chart option.

Sorting the data

The logged data is sorted in the same sequence that it was recorded, however there are times when you may want to sort the information in ascending order of a particular channel.

To sort the data according to a particular channel first click on the desired channel and press the right mouse button to display the pop-up menu. Select the Sort on Selected Channel option and wait for the software to re-sort the data. Once sorted the channel name of the sorted channel will be shown as a darker grey colour.

Un-sorting the data

If you sorted the data but now want it sorted back into the default recorded sequence first click on any data channel, then press the right mouse button and select the Un-sort Data option. You must wait until the software re-sorts the data back to the default sequence.

Set Display Options

This allows you to set the graph span value (Min / Max) for the selected channel, the plot color, grid color, text color, background color, grid increments (spacing) and grid lines. These settings can be saved permanently and will be used as the default settings each time you open another log file.

Lap Selection

This section allows you to extract and sort your data according to a particular lap.

When you open a log file the software will extract each lap so you can then select to view only selected laps from each log file. This section allows you to select which laps you want to view. Each opened log file will have its file name displayed along the left of this section and a selection tick box will appear for each lap recorded in the log data file.

Selecting a particular lap number

Each lap has a selection tick box that you can click. When you click a lap number the system will then extract the data only for the selected lap.

Viewing lap times.

Lap times must be recorded either with an IR beacon or GPS coordinates.

Press the Timing Data button. If there are recorded laps in the opened log file these will be shown as separate line entries on this section if each lap has sectors then you can expand the lap to show sector times.

Viewing your race line.

Your race line is recorded via the GPS up 20 times per second. This data can be displayed on the track map screen. You can overlay race line data from various log files onto this screen.

After you have loaded a log file select the Show Track Map option from the Maps menu. To display the race line first click on any data channel then press the right mouse button. This will show a pop-up menu where you must select the Add Race Line Data option. This may take several seconds depending on the amount of data in the log file

Session Information

This section allows you to record information for the session such as event information, weather information, vehicle information, driver information, date and time. Most data fields are memo fields that can record up to 255 characters each.

Selecting different Chart options

You can display information in different ways.

You can display data channels overlaid on the same chart or view them in a different chart format such as scattered or as bars. You can do this by pressing the appropriate button just above the chart screen.

Separated charts

This is the default format. This mode is shown when the Separate button is a light grey colour. In this mode each data channel with a different name will be shown in a separate chart.

Overlaying charts

If you want to overlay every channel onto the same chart then press the overlaid button before selecting the channels you want to view.

Line charts

This is the default data display. It shows recorded information as a line. This is the most common and easiest to understand chart form. This chart is selected when the Line button is shown as light grey.

Scatter charts

This is useful when you want to see where a particular occurrence happened the most often. To use this effectively you must also sort the data in the correct sequence. The sequence will depend on the information you want to view. A scatter chart will show the information as a collection of points (dots). You can then easily see where the majority of points are grouped together. This chart is selected when the Scatter button is shown as light grey.

Zooming into the chart

To zoom into an area of the chart, start the mouse pointer at the top left corner of the area, press the right mouse button and drag a square that covers the area you want to zoom into then release the button. To un-zoom the chart simply perform the reverse action starting from bottom right. This same action can also be done on the track maps.

Dragging a chart

You can drag the chart to show hidden data. Place the mouse pointer anywhere on the chart, press the right mouse button and drag to move the entire chart left, right or down.

Creating a track map

A track map must show an outside perimeter and an inside perimeter. To draw a track map you must have an IR beacon or GPS Module so the system can detect a full lap.

1. Record a log driving slowly around the outside perimeter of the track for one lap and then driving around the inside perimeter for another lap.
2. Import the log into the data analyser software as per a normal data log.
3. Open the log as a session.
4. Open the track editor window.
5. Click the Lap selection button and select the lap you want to use for the inside perimeter.
6. Click the channel data button to show the data fields again.
7. Click on any data field of the log and then right-click and select import as inside track map.
8. Click on Lap selection button again and select the lap to use for the outside perimeter.
9. Click the channel data button again.
10. Click on any data field and then right-click and select import as outside track map.
11. On the track editor window select file->Save track map to save the imported track.

Overlaying a race line over a track map.

1. Go to File->Maps and select Show Track Map.
2. In the track map window select File-Open Track File and select a track file.
3. Open a data log and select the lap you want to view. If you do not select a lap then the system will draw all of the laps.
4. Click on any data field then right-click and select add race line data.

As you move the cursor over the chart a dot will be drawn on the track map showing your position on the track.

Recording lap sectors and times using the GPS.

The EDL-7 race dash can record lap times and sector times. It uses the GPS coordinates to measure track times

To define the start/finish line and any track sectors you must use the track editor window. You can use this window to define timing sectors, target lap and sector times and also messages that are displayed as you enter certain sectors.

Sectors are defined by drawing lines across the track. These lines should be perpendicular to the race line for best results. They should also be at least 2 to 3 times wider than the track.

1. Open the track editor window.
2. Open a track file you want to define the start/finish line and any sectors.
3. Select the Track Sectors button.
4. Press the Add sectors button. The first sector line will be the Start/Finish line.
5. Press the Add sectors button to add more sectors.

Each entry has a square next to it. You must tick the square of the sector you want to define on the track map.

1. Tick the square of the Start/Finish line and then move the mouse pointer onto the track map.
2. Press the left mouse button to select the starting point of the sector. As you move your mouse a line will be drawn from the first anchor point to the mouse pointer. Move the pointer to the location you want the sector line to end and press the left mouse button again.
3. Add a target time (in seconds) for each sector (optional). This will be used by the race dash to measure timing performance.
4. Do the above step for each sector line.

When you have finished you must save the sector definitions as a sector file so they can be installed into the Race dash. The Race Dash will use this information to record sector times and lap times. Press the Save Sector File button, give the file a name and press Save.

Installing the sector file onto the Race Dash.

1. Go to the Dash Manager -> EDL-7 option.
2. Select the SD card drive letter.
3. Select the Log Settings tab.
4. Press the Install Sector file to SD card button.
5. Select the sector file you want to use and press the Open button.

Recording Sector Messages using the GPS.

The EDL-7 race dash can display different messages as you drive around the track. It used the GPS coordinates to determine which message to display.

To define a set of messages you must use the track editor window. You can use this window to define message sectors and a message for each sector. These messages are displayed as you enter each sector.

Sectors are defined by drawing lines across the track. These lines should be perpendicular to the race line for best results. The length of the line determines the start of the sector from the line and the end of the sector from the line. This is an imaginary rectangle with its center of gravity being the line drawn.

6. Open the track editor window.
7. Open a track file you want to use to define the message sectors.
8. Select the Track Messages button.
9. Press the Add Message button.
10. Press the Add Message button to add more message sectors.

Each entry has a square next to it. You must tick the square of the sector you want to define on the track map.

5. Tick the square of the first message line and then move the mouse pointer onto the track map.
6. Press the left mouse button to select the starting point of the message sector. As you move your mouse a line will be drawn from the first anchor point to the mouse pointer. Move the pointer to the location you want the message sector line to end and press the left mouse button again.
7. Add a message text for each message sector. This will be displayed whenever the car enters the defined sector and is then cleared when the sector is exited.
8. Do the above step for each message sector line.

When you have finished you must save the message sector definitions as a message file so they can be installed into the Race dash. The Race Dash will use this information to display the messages at the correct time. Press the Save Message File button, give the file a name and press Save.

Installing the Message file onto the Race Dash.

1. Go to the Dash Manager -> EDL-7 option.
2. Select the SD Card drive letter.
3. Select the Log Settings tab.
4. Press the Install Message file to SD card button.
5. Select the Message file you want to use and press the Open button.