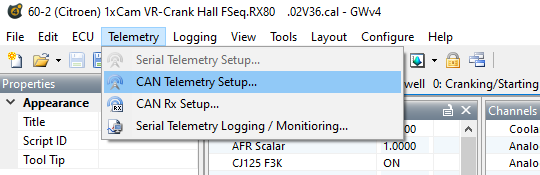
**Setting up a CAN bus for communication with G.E.M.S. CAN based displays and loggers**

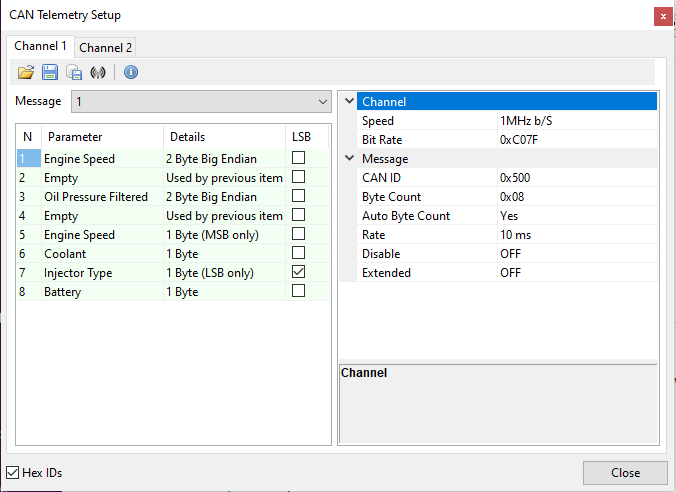
The first thing to check, is that the ECU that you are using is capably of transmitting data (telemetry) over a CAN network. To do this, open your calibration in GWv4, using the appropriate .GIN file and open the ‘Telemetry’ dropdown list. If the ‘CAN Telemetry Setup’ option is greyed out, then the ECU is incapable of transmitting CAN telemetry to a display.



In the example above, it shows that this particular ECU (an RX80) has the capability of transmitting CAN telemetry but Serial Telemetry is not available, as it is ‘greyed out’.

Once that the ECU’s telemetry capability is established, we can set about defining the parameters that you will want to appear on the ‘bus’. It is suggested that you make a list of the parameters that you want to display or log, so as not forget anything important.

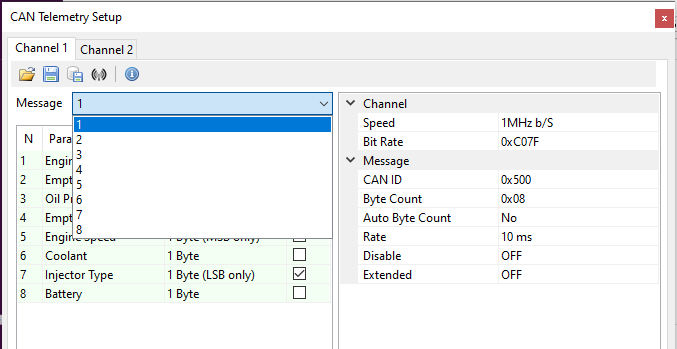
Once you have a good idea on what is going to be sent from the ECU, left click on the ‘CAN Telemetry Setup’ option to reveal the setup panel, as shown below.



In this example, the ECU is capable of transmitting telemetry on two individual CASN busses, as can be seen to the top/left of the panel, shown as ‘Channel 1’ and ‘Channel 2’. Ensure that you are putting the parameters onto the correct bus, i.e. the one that is connected to the display or logger that you want to use.

We will concentrate on Channel 1 as that is the usual one that is used for display purposes, as Channel 2 historically is used for loggers. It is worth noting at this point, that if an ECU is capable of transmitting on two busses, that they are independent to each other and the same parameter may be transmitted on both busses at different speed, if needs be.

In the left-hand panel, it you click on the ‘down’ in the ‘Messages’ dropdown box, you will see how many messages are available in the chosen bus.



In this example, there are 8 individual messages, each containing 8 bytes of data. This can be used as ‘8 bytes’ or 4 ‘words’ or a mixture of the two, as shown on the previous page. So, this ECU has the capability of 64 bytes of data. If a parameter requires 2 bytes of data (as Engine Speed) but you only require Most Significant Bit (as in Byte #5) just add the next required parameter in byte #6 and you will receive a message showing (MSB only). If you require the Least Significant Bit, then tick the LSB box, as shown above. For a parameter that is ‘word’ in length, this will change the ‘Endian-ness’ of the data. Continue to fill in the parameter that you require until message #1 is complete.

Next, we will look at the right-hand panel that will give the information for Message #1. From the top, the channel speed is selected, either choosing the ‘Speed’ in b/sec or by specifying a ‘Bit Rate’ in hexadecimal, if one of the ‘Speed’ options is not suitable.

Then we must choose the arbitration (or CAN) code that will be carrying the 8 bytes of data that we have selected. This must be unique but is up to you, the user what it will be. In the above example a ‘non-extended’ code of 0x500 as been selected for this particular message.

If you set ‘Auto Byte Count’ to ON, then the ‘Byte Count’ will automatically update if the parameter list has a parameter removed.

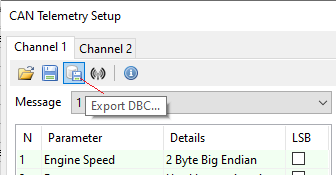
The ‘Rate’ specifies the departure intervals of the data on the bus, so every 10 milliseconds in the case above.

To switch off a particular message, set ‘Disable’ to ON and if you require extended arbitration codes, set the ‘Extended’ feature to on also.

Once Message #1 is complete, select Message #2 and continue to add more parameters if required, not forgetting the change the ‘CAN ID’ field, as all messages have to have a unique code - remember...?! Other options in the right-hand panel should remain the same.

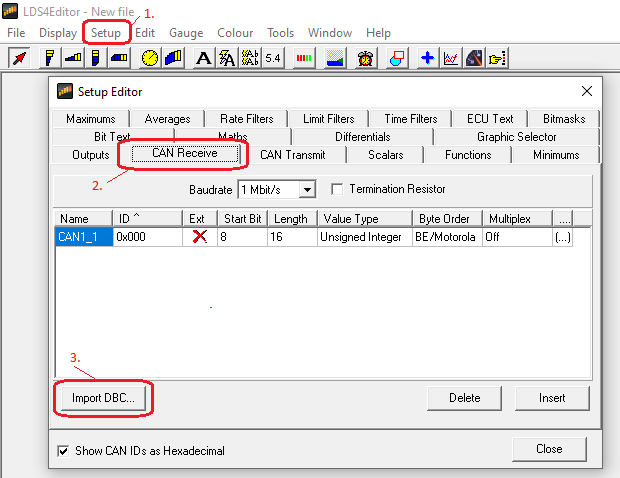
Once you have finished compiling your CAN bus in your ECU (and saving the calibration), the simplest way of setting up a G.E.M.S. display with the correct CAN information, is to Export the setup from the calibration, using the ‘Export DBC’ option in the CAN Telemetry Setup.

A DBC file, is an ASCII based translation file used to apply identifying names, scaling, offsets, and defining information, to data transmitted within a CAN frame. For any given CAN ID, a DBC file can identify some or all of the data within the CAN frame/message.

Simply, choose a file name and make it accessible to the display editor, when you come to design your screens.

Most of the latest G.E.M.S. display hardware, has the ability to read a DCB file, which will set the unit with the same scalers and offset etc. as are set for displaying the parameter in the ECU using GWV4 software. For the purpose of this document, we will use the LDS4 display editor.

When you either open an existing or start a new setup in the LDS4Editor on the tabs is marked ‘Setup’ and this is where most of the work is done. Download the appropriate manual from the GEMS website for full details on how the editing software works.



Now you can design your pages using representative CAN telemetry data, exactly as the ECU is generating.