ECM ENGINE CONTROL AND MONITORING

LambdaCAN* Module

(LambdaCANp, LambdaCANd, LambdaCAN)

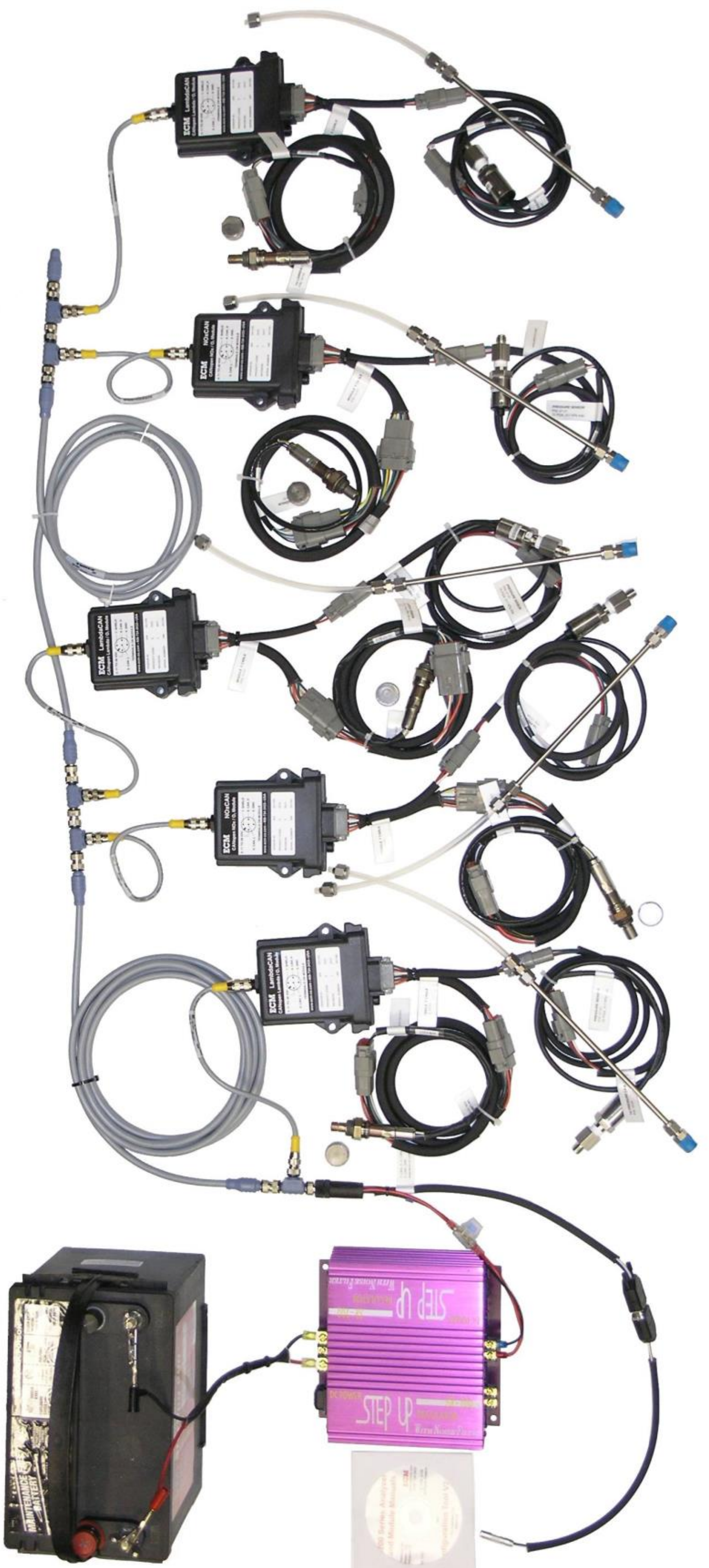
Instruction Manual

Important Note

Due to the nature of CAN instrumentation, you cannot just physically connect a measurement module to a bus and expect data from that module to be available. The LambdaCAN* measurement module has to be set up to send the data required and the receiving device (ex. data acquisition software) has to know what is being sent. The setting up of modules and the production of a .dbc file used by the receiving device to interpret the data sent, is performed using the supplied Configuration Tool (software) which runs on a PC. 1-26-2014

Printed in the United States of America.

Table of Contents



Complex measurement systems can be easily built with ECM LambdaCAN*, NOxCAN*, and other modules. Here is a five-channel lambda, O_2, and NOx pressure-compensated in-vehicle system.

Introduction

ECM offers three Lambda module kits: LambdaCANp, LambdaCANd, and LambdaCAN. LambdaCANp is the latest Lambda module kit and can optionally be supplied with a P-comp (pressure compensation) kit. The P-comp kit for LambdaCANp uses a pressure sensor with an eight terminal connector. This eight terminal connector contains a memory chip with the calibration of the pressure sensor (i.e. "plug-and-play"). LambdaCAN is the original kit released and can also optionally be supplied with a P-comp kit. A LambdaCANd kit is simply a LambdaCAN kit with the P-comp kit supplied as standard equipment. Both the LambdaCAN and LambdaCANd kits use a pressure sensor with a four terminal connector. The calibration numbers for this pressure sensor are written on a label attached to the sensor harness and must be entered into the LambdaCAN or LambdaCANd before use (this is done by ECM when a P-comp kit is delivered with the LambdaCAN or LambdaCANd). See Appendix A for kit contents. The specifications for the kits are as follows:

<u>LambdaCANp Kit:</u>

Ranges: **λ (Lambda):** 0.4 to 25
AFR: 6 to 364
F/A: 0.003 to 0.167
Φ: 0.04 to 2.5
%O2: 0 to 25% (wet)
P (pressure, optional): 0 to 517 kPa (0 to 75 psia)
Accuracies: **λ, AFR, F/A, Φ:** ± 0.6% (at stoichiometric), ± 0.9 (average, elsewhere)
%O2: ± 0.1% (absolute)
P: ± 0.7 kPa, (± 0.10 psia)
Response Times: **λ, AFR, F/A, Φ, O2:** Less than 150 ms
Sensor Mounting: 18mm x 1.5mm thread

<u>LambdaCAN and LambdaCANd Kits:</u>

Ranges: **λ:** 0.4 to 25
AFR: 6 to 364
F/A: 0.003 to 0.167
Φ: 0.04 to 2.5
%O2: 0 to 25% (wet)
P (pressure, optional for LambdaCAN): 0 to 517 kPa (0 to 75 psia)
Accuracies: **λ, AFR, F/A, Φ:** ± 0.6% (at stoichiometric), ± 0.9 (average, elsewhere)
%O2: ± 0.1% (absolute)
P: ± 5.2 kPa, (± 0.75 psia)
Response Times: **λ, AFR, F/A, Φ, O2:** Less than 150 ms
Sensor Mounting: 18mm x 1.5mm thread

Setup and calibration of LambdaCAN* kits is performed using the supplied PC software program called "The Configuration Tool". The Configuration Tool uses a CAN communication device to communicate with one or more ECM *CAN modules (ex. LambdaCAN*, NOxCAN*, baroCAN, appsCAN) or analyzers (ex. NOx 5210, Lambda 5220, EGR 5230). While the tool is being used with modules, just ECM modules set to stand-alone mode (see Appendix B) should be connected to the CAN bus. While the tool is being used with analyzers, just analyzers should be connected to the CAN bus. With analyzers, the Configuration Tool is just used to produce .dbc files. However with modules, the Configuration Tool replaces the analyzer's display head as the user interface so it must do much more. This document focuses on using the Configuration Tool with ECM's LambdaCAN* modules.

The Configuration Tool supports four CAN communication devices: Kvaser, ETAS, Peak USB-to-CAN adapters, and the VectorCAN CAN adapter card. Driver software for one of these adapters must be installed prior to using the Configuration Tool. This software will be supplied with the adapter or be available on-line. ECM's Configuration Tool is delivered on a CD and the latest version is on www.ecm-co.com.

Once the adapter's driver and the Configuration Tool software are installed, and with the module(s) powered and connected to the CAN adapter, start the Configuration Tool software. Click on the "Modules" tab, select the CAN adapter, and click on the "START" button.

The software will identify the modules on the bus and display them in the "Module" field. If this does not happen, make sure that the CAN bus is properly terminated (i.e. resistors). Open the Module field to see all the modules on the bus. If a module is not listed, one reason could be that its Node ID is the same as another module. To resolve this, remove all modules except the "missing" one from the CAN bus, STOP then START the software, and change that module's Node ID. Another reason that a module is not listed could be that the module is in EIB mode instead of stand-alone mode. All modules must be in stand-alone mode.

To configure one of the modules (ex. change its Node ID) or to look at that module's data, you have to select that module in the "Module" field.

There are three things you can do with modules using the Configuration Tool:

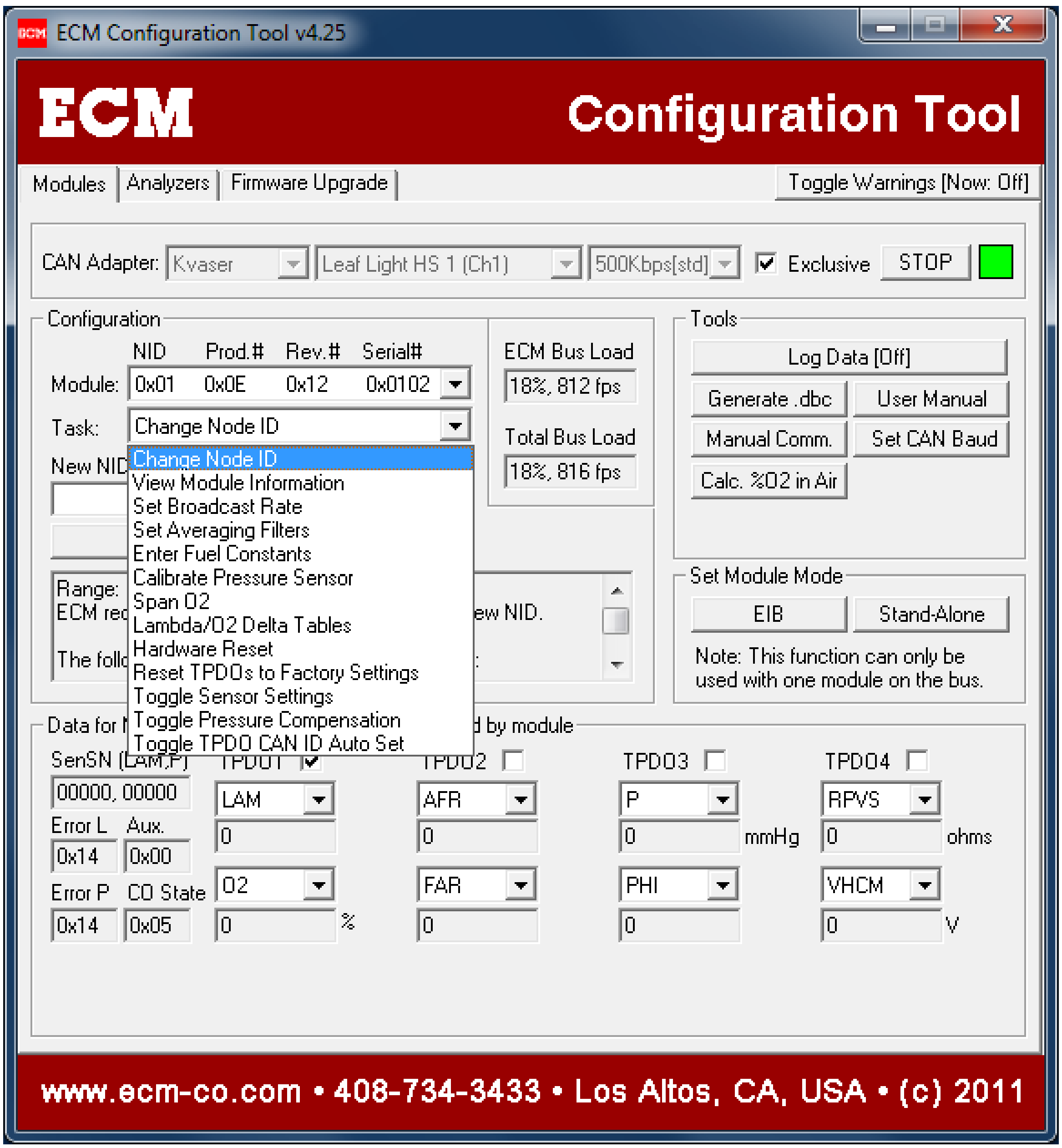
1. Configure a module. This includes calibrating a sensor attached to the module.
2. Look at data coming from that module in real-time and optionally log it.
3. Produce a .dbc file to be used by a device receiving module data.

Alternatively, 1. and 2. (above) can be performed by direct CAN communication with the LambdaCAN* module using user-written software. For information on how to do this and for detailed information about the modules, refer to the LambdaCANp, LambdaCANd, or LambdaCAN Instruction Manual (these manuals are not the same as this "LambdaCAN* Instruction Manual").

Configuring a Module

Normally, you will be configuring a module to be used in stand-alone mode. Stand-alone mode is used when the modules are connected to a CAN bus that goes directly to a data acquisition system. In stand-alone mode, a module's configuration is performed by selecting one of the tasks in the "Task" field. When a module is connected to a NOx 5210, Lambda 5220, or EGR 5230 display head, it must be in EIB mode. In EIB mode, most of these tasks are performed using the display head. Appendix B describes how to configure a module to operate in one mode or the other.

The following assumes that the module is to be used in stand-alone mode. The tasks available for the LambdaCAN* modules are listed in Table 1.



Change Node ID: Allowable range 0x01 to 0x7F (hex). When you assign a Node ID (NID), the following CAN IDs **cannot be used** by any other devices on the bus: 0x00, 0x80 + NID, 0x180 + NID, 0x280 + NID, 0x380 + NID, 0x480 + NID, 0x580 + NID, 0x600 + NID, 0x700 + NID, 0x7E4, 0x7E5.

View Module Information: Manufacturer's Name, Hardware Version, Software Version

Set Broadcast Rate: All activated TPDOs are transmitted every "n" milliseconds. "n" can be programmed. 5 ms is the minimum. Default: 5 ms.

Set Averaging Filters: Before the data is transmitted by the module (at the broadcast rate), it can be averaged. There are two averaging filters (alphas): one for Lambda (i.e. O2R, IP1, IP1X, IP1R, LAMR, AFR, PHI, FAR, LAM, O2) and one for Pressure (i.e. PR16, PR10, PCF, P, PVLT, PKPA, P, PVLT, PKPA, PBAR, PPSI). Alphas can range from 0.001 to 1 and are used in the recursive averaging filter:

$$AvgData_t = alpha \times Data_t + (1-alpha) \times AvgData_{t-1}.$$

Where: $AvgData_t$ is the transmitted data at time "t".

$AvgData_{t-1}$ is the previously transmitted data.

$Data_t$ is the raw data at time "t".

This formula is executed every 5 ms regardless of broadcast rate. Note that if alpha = 1, there is no averaging and the data taken at time "t" becomes the average value at time "t" and hence the broadcast value. Default Alpha values: 0.375.

Enter Fuel Constants: H:C, O:C, and N:C ratios or if the fuel is H2. Defaults: H:C = 1.85, O:C = 0, N:C = 0

Calibrate Pressure Sensor (LambdaCANp only): Gives access to the voltage versus pressure curve of the pressure sensor. Curves can be edited and downloaded to the pressure sensor.

Enter Pressure Sensor Constants (LambdaCANd and LambdaCAN only): N and C are the gain and offset of the pressure sensor. N and C are written on a label attached to the pressure sensor. P (psia) = N x (V - C) where V = voltage from sensor.

Span O2: User enters displayed %O2 (TPDO data) and the actual %O2 of the span gas. This is how you calibrate the Lambda sensor. See Appendix C.

Open Lambda/O2 Delta Tables: These tables are used to modify the calculated O2 and Lambda (and hence AFR, FAR, and PHI). There are two tables. The first, the Delta O2 Table, allows modification of the calculated %O2 via a user-entered look-up table. Delta O2 Table entries will not influence Lambda, AFR, FAR or PHI values. The second, the Delta Lambda Table, allows modification of the calculated Lambda (and hence AFR, FAR, and PHI) via a user-entered look-up table. The Delta Lambda table will not influence O2 values.

Hardware Reset: Equivalent to powering the module down then up again.

Reset TPDOs to Factory Settings: These are the parameters displayed and transmitted. TPDO stands for Transmit Process Data Object.

Toggle Sensor Settings: Allows power to lambda sensor to be turned on and off (Default: On) and enables rapid sensor warm-up scheduling (Default: On (fast)).

Toggle Pressure Compensation: Enable and disable pressure compensation of O2 (Lambda, AFR, FAR, PHI) data. Default: On.

Toggle TPDO CAN ID Auto Set: Enable and disable TPDO CAN ID Auto Set.

Table 1: Task List for LambdaCAN* Modules

Calibration of the Lambda Sensor

Lambda sensors operate on a diffusion mechanism. Molecules leaving the combustion chamber diffuse through a passage into the sensor where oxidation, oxygen liberating, and oxygen pumping occurs. The diffusion passage is like a filter and like a filter it can get clogged. In fact, it is impossible for the passage not to get clogged when the sensor is being used. This clogging of the diffusion passage is the main mechanism by which a sensor "ages" resulting in a calibration shift.

All Lambda sensors are calibrated (i.e. spanned) before leaving the factory. It is recommended that they be calibrated periodically during use. How often can only be determined by your experimentation. Alternatively, the sensors can be sent to ECM for recalibration.

Calibration information (both factory calibration and user calibration) for the Lambda sensor is stored in a memory chip in the sensor's connector. Therefore, the sensor does not have to be used with the module it was calibrated with. Once calibrated, the sensor can be sent to another site to be used. After use, the sensor can be returned to the calibration center for recalibration thus compensating for any sensor aging. Centralization of calibration in this way improves measurement consistency and extends useful sensor life.

Calibration of the Lambda Sensor and the Delta Tables

The "Span O2" task is used to calibrate the Lambda sensor. This task calibrates both the %O2 and Lambda (AFR, FAR, PHI) measurements from the sensor. To perform an O2 span:

1. The sensor should on for at least 20 minutes. Most of the time, you only need to wait 2 minutes but some sensors, due to condensed material on them, take longer to stabilize.
2. Put the sensor and the pressure sensor (if equipped) in ambient, stationary air. The pressure during calibration is required if the sensor's %O2 and Lambda (AFR, FAR, PHI) measurements are to be pressure compensated.
3. Calculate the %O2 in the air. The %O2 in air with no humidity is 20.945. This percentage decreases with increasing humidity. To calculate the %O2 in non-zero humidity air, use the "Calculate %O2 in Air" tool in the Configuration Software or Appendix C. 20.7 is a common number.
4. Select O2 as a TPDO parameter.
5. Select the Task "Span O2". Enter the displayed (as the TPDO) O2 and the actual %O2 (as calculated in 3 above), then click on "Span".

◆ Delta O2 Table and Delta Lambda Table

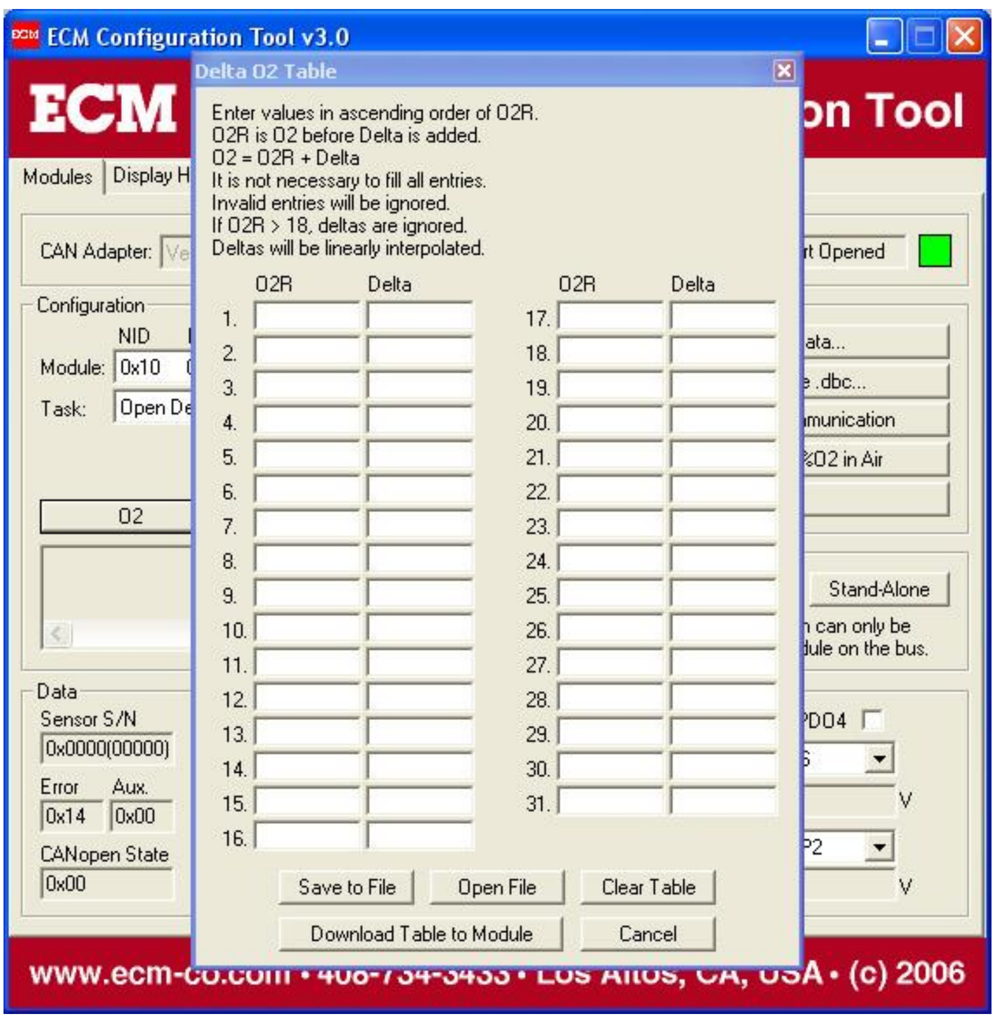
The %O2 calculated by LambdaCAN* modules is denoted as "O2R" and is a wet (i.e. water taken into consideration in %O2 calculation), at chemical-equilibrium (i.e. not frozen-equilibrium) %O2 measurement (see Appendix D). The Lambda (AFR, FAR, PHI) calculated by LambdaCAN* modules is denoted as "LAMR" and matches that calculated by mass flowrates of air and fuel entering the engine.

Sometimes users would like dry %O2 readings, or frozen-equilibrium %O2 readings, or dry frozen-equilibrium readings, or readings that match another instrument. For this purpose, the Delta O2 Table is used. The Delta O2 Table (found in the task "Lambda/O2 Delta Tables") allows the user to add a number (a "delta") to the "O2R" calculated by the LambdaCAN* module giving the "O2" parameter:

O2 = O2R + Delta O2 Table value (interpolated from table)*

* except when O2R > 18 for which O2 = O2R, regardless of table data.

A Delta O2 Table can be created, edited, and saved as a text file (.txt) on your PC. The file must be downloaded into the LambdaCAN* module for it to be used. Unlike the sensor calibration information (i.e. span) which is stored in a memory chip in the sensor's connector, the Delta O2 Table is stored in the module.



Similarly, sometimes users would like Lambda (AFR, FAR, PHI) readings to match other instruments such as gas-bench analyzer calculated values (ex. via Spindt or Brettschneider) or company "heritage" Lambda measurement techniques. The Delta Lambda Table (found in the task "Lambda/O2 Delta Tables") allows to user to add a number (a "delta") to "LAMR" calculated by the LambdaCAN* module giving the "LAM" parameter:

LAM = LAMR + Delta Lambda Table value (interpolated from table)

AFR, FAR, and PHI are calculated from "LAM". A Delta Lambda Table can be created, edited, and saved as a text file (.txt) on your PC. The file must be downloaded into the LambdaCAN* module where it is used. File LA42.TXT (for LSU4.2 sensor) or file LA49.TXT (for LSU4.9 sensor) can be downloaded to make LambdaCAN* modules match default ETAS LA4 data (see Appendix D). Unlike the sensor calibration information (i.e. span) which is stored in a memory chip in the sensor's connector, the Delta Lambda Table is stored in the module.

Selecting What Data is to be Sent (TPDOs)

Data send from LambdaCAN* modules is packaged as TPDOs (Transmit Process Data Object). Each TPDO contains two pieces of data and each module can send up to four TPDOs. All selected TPDOs will be sent at the broadcast rate. For example, if the broadcast rate is 5 ms and four TPDOs were selected to be sent, then eight pieces of data would be transmitted every 5 ms. To avoid slowing down the effective data rate on the CAN bus, select the number of TPDOs to be sent and the broadcast rate sparingly. For the case of multiple modules sending multiple TPDOs on the same CAN bus, the minimum (i.e. fastest) broadcast rate is given by:

Minimum Broadcast rate (ms) = The total number of TPDOs for all modules x 0.3125

For example, if there are eight modules, each sending two TPDOs, the minimum broadcast rate is 5 ms.

The data transmitted is selected in the “Data” area of the Configuration Tool. Activate the number of TPDOs to be used by clicking in its box to put in a check mark. Select the data contained in each TPDO using the pull-down windows. The list of available parameters for the LambdaCAN* module is given in Table 2.

Note that in the parameter list, there are two Lambdas: LAMR (Lambda Real) and LAM (Lambda).

LAMR is the Lambda value calculated by the module.

LAM is the Lambda value calculated by the module and modified by the Delta Lambda Table according to the relationship:

LAM = LAMR + Delta Lambda Table value (interpolated from table)

The modified LAM (Lambda) will also influence AFR, FAR, and PHI.

Similarly for O2R and O2:

O2 = O2R + Delta O2 Table value (interpolated from table)*

* except when O2R > 18 for which O2 = O2R, regardless of table data.

Parameter Name Displayed	Full Parameter Name	Parameter Description
O2R	%O2real (%)	%O2 before addition of Delta O2 Table
IP1	Ip1 (A)	Pressure-compensated Lambda sensor pumping current
RPVS	RPVS (Ohms)	Lambda sensor internal VS cell resistance
VHCM	VH Commanded (V)	Desired heater voltage commanded by the module
VS	VS (V)	Lambda sensor internal VS cell voltage
VP1P	VP+ (V)	Lambda sensor pumping voltage
VSW	Vsw (V)	Supply voltage measured at the module
VH	VH Measured (V)	Actual heater voltage at the module
TEMP	Circuit Board Temp (°C)	Temperature of the module circuit board
IP1R	Ip1raw (bits)	Lambda sensor pumping current (unsigned integer format)
PR16	Praw16 (bits)	16 bit Pressure sensor output voltage (unsigned integer format)
ERFL	Error bit flags (bits)	Module error flags (unsigned long format)
ERCd	ECM CANopen Error Code	ECM CANopen Error Code
PR10	Praw10 (bits)	10 bit Pressure sensor output voltage (unsigned integer format)
PCF	Pressure Correction Factor	Lambda sensor pressure compensation correction factor x 10000
PCFE		ECM diagnostic parameter
O2E		ECM diagnostic parameter
IP1E		ECM diagnostic parameter
PE		ECM diagnostic parameter
P	P (mmHg)	Pressure sensor measured pressure (absolute) in mmHg
LAMR	LAMBDAreal	Lambda before addition of Delta Lambda Table
AFR	Air-Fuel Ratio	Air-Fuel ratio calculated using LAMBDA
PHI	PHI	PHI = 1/LAMBDA
FAR	FAR*10000	FAR = (1/AFR) * 10000
LAM	LAMBDA	Lambda after addition of Delta Lambda Table
O2	O2 (%)	%O2 after addition of Delta O2 Table
IP1X	Ip1 non Pcomp (A)	Non-pressure compensated Lambda sensor pumping current
PVLP	P (V)	Raw volts from pressure sensor
PKPA	P (kPa)	Pressure sensor measured pressure (absolute) in kPa
PBAR	P (bar)	Pressure sensor measured pressure (absolute) in bar
PPSI	P(psi)	Pressure sensor measured pressure (absolute) in psi
PERF	Pressure error bit flags	Pressure sensor bit flags (LambdaCANp only)
PERC	CANopen error code	ECM CANopen pressure error code (LambdaCANp only)

Table 2: LambdaCAN* Parameter List

Producing a .dbc File

A .dbc file describes to the device receiving data from one or more LambdaCAN* modules what is in the data packages. For each module, the packages will contain data for the parameters selected in the activated TPDOs and an error code. The Configuration Software has a tool called “Generate .dbc…” that will generate a .dbc file for all the LambdaCAN* modules on a CAN bus. Make sure that each module is configured as desired and that all modules are on the bus before the “Generate .dbc…” button is pushed. Data package information from all the modules is stored in the one .dbc file produced.

Programs importing the .dbc file and applying it to the CAN data transmitted by the modules will see data, etc identified as follows:

Data: **name_nid[units]**

where: name = parameter name. See Table 2 for LambdaCAN*
nid = node id of module in hex
units = units of parameter

for example: O2_0X01[%] which is the %O2 measured by module with nid 0X01

Error code: **ECM_Error_Code_nid**

where nid = node id of module hex
error code is in hex and given in Table 3

for example: ECM_Error_Code_0x11

Auxiliary: **ECM_Auxiliary _time[sec]**

where: time = decrementing countdown to module activation in hex

for example: ECM_Auxiliary_0X12[sec]

ECM ERROR CODE	**LED ACTION**	**DESCRIPTION OF ERRORS**
0x0000	Grn ON	All OK, (green led constantly on)
0x0001	Flash Grn 10Hz	Sensor warm-up period
0x0002	Grn/Both/Red 2s	Power on reset/ Init hardware
0x0011	Pulse Red 1x/2s	16b ADC failed to init
0x0012	Pulse Red 1x/2s	+Vsw shorted
0x0013	Red ON	Sensor turned off (red led constantly on)
0x0014	Pulse Red 1x/2s	HTR open
0x0015	Pulse Red 1x/2s	HTR shorted
0x0021	Pulse Red 2x/2s	1wire bus shorted
0x0022	Pulse Red 2x/2s	No 1wire present
0x0023	Pulse Red 2x/2s	CRC16 error
0x0024	Pulse Red 2x/2s	Invalid 1wire parameter (sensor type)
0x0025	Pulse Red 2x/2s	1-wire data format not compatible (old rev)
0x0031	Pulse Red 3x/2s	+Vsw < 6 for > 7sec
0x0032	Pulse Red 3x/2s	+Vsw > 30V
0x0041	Pulse Red 4x/2s	VS too high
0x0051	Pulse Red 5x/2s	RVS too high
0x0052	Pulse Red 5x/2s	(VHcommanded - VHactual) > 0.5V for > 10sec
0x0061	Pulse Red 6x/2s	VP+ > 6V
0x0062	Pulse Red 6x/2s	VP+ < 2V
0x0063	Pulse Red 6x/2s	VP2 out of range (NOxCAN* only)
0x0064	Pulse Red 6x/2s	0.15V > VS+ > 0.75V for 7.5 sec.
0x0065	Pulse Red 6x/2s	User data (span) in 1wire corrupted (user must set new span)
0x00A1	N/A	Invalid software state
0x00B1	N/A	CAN overrun
0x00B2	N/A	CAN passive mode
0x00B3	N/A	CAN heartbeat error
0x00B4	N/A	CAN recover bus off
0x00B5	N/A	CAN Tx CanId collision
0x00B6	N/A	Serial overrun
0x00FF	Both ON	Module powering down within 500ms
ECM AUX	N/A	Sensor Warm-up count down in seconds (active during ECM Error Code 0x0001)

Table 3: LambdaCAN and NOxCAN* Error Codes List

Using the dashCAN* Display

The dashCAN* display (see cover and below) is a small (105 mm x 63 mm x 63 mm, for dashCAN and dashCAN2 or 105 mm x 63 mm x 165 mm, for dashCAN+), display for CAN networks containing ECM *CAN modules. dashCAN has two displays and no analog outputs. dashCAN2 has two displays and two programmable analog outputs. dashCAN+ has two displays and six programmable analog outputs. dashCAN* comes with a two meter cable and a "T" (P/N 09-05). Simply attach dashCAN* to the CAN bus and parameters from two ECM *CAN modules can be displayed and converted to analog outputs (dashCAN2 or dashCAN+ only). The top display and half the analog outputs can be assigned to one module and the bottom display and the other half of the analog outputs can be assigned to the same or another module. Multiple dashCAN* displays can be attached to the CAN bus.

dashCAN* has two modes of operation: RUN (when measurements are displayed) and SYS (where dashCAN* is set-up). The SYS key toggles between the modes.

While in RUN mode:

i. If the ↑ button is pressed, the displays will show the serial numbers of the modules assigned to the displays (a module is assigned to the top display and analog outputs 1, 2, 3 (for dashCAN+) or analog output 1 (for dashCAN2), and a module is assigned to the bottom display and analog outputs 4, 5, 6 (for dashCAN+) or analog output 2 (for dashCAN2)).
ii. If the ↓ button is pressed, the displays will show the parameter names assigned to the displays. See Table 2.
iii. If the ENT button is pressed, the displays will show the units of the parameters. "PCTG" is %. "dIM" means dimensionless (ex. for AFR, FAR, PHI, Lambda).

In RUN mode, four things other than data can be displayed:

i. "ERR" and "####" where "####" is an error code. See Table 3.
ii. "…." which means that a module has not been assigned to that display.
iii. "----" which means that dashCAN* has an internal problem.
iv. "XXXX" which means that dashCAN* is not receiving any data from the module assigned to that display.

When first entering SYS mode, either “MOd” will be on the top display or “LOCK” will be on the bottom display. If “MOd” is displayed, the ↑ and ↓ keys will roll through the setup options (see Table 4). First the options for the module assigned to the top display are shown on the top display, followed by identical options for the module assigned to the bottom display, ending with the global CONF (Configuration) setup. Pressing the ENT key will select the displayed setup option and allow its programming.

If “LOCK” is displayed, the dashCAN* has been locked and its setup cannot be changed until it is unlocked. Appendix G describes how to LOCK and unlock dashCAN*.

Setup Option	**Level 1**	**Function**
MOd		Select module s/n to be assigned to the display. Default is NONE.
RATE		Set parameter averaging rate. Range 0.001 to 1.000 Default is 1.000. 1.000 means no averaging.
AOUT (dashCAN+ only)		
	A1	Program analog output 1 from module assigned to top display
	A2	Program analog output 2 from module assigned to top display
	A3	Program analog output 3 from module assigned to top display
	A4	Program analog output 4 from module assigned to bottom display
	A5	Program analog output 5 from module assigned to bottom display
	A6	Program analog output 6 from module assigned to bottom display
AOUT (dashCAN2 only)		
	A1	Program analog output 1 from module assigned to top display
	A2	Program analog output 2 from module assigned to bottom display
dISP		Select parameter. Note: Parameters available are those contained in TPDOs programmed to be transmitted from the module (programmed using the Configuration Tool).
CONF	LEdS	dashCAN+ only. Set display intensity. Default is 3333. dashCAN and dashCAN2 have a knob for display intensity.
	1V4V	dashCAN2 and dashCAN+ only. Command 1V and 4V outputs.
	LOCK	Lock and Unlock Display for Programming

MOd, RATE, AOUT, and dISP appear on the top display for the module assigned to the top display and then on the bottom display for the module assigned to the bottom display. CONF just appears on the bottom display and is for global dashCAN* setup. All entries must be followed by pressing the ENT key.

Table 4: Menu Tree for dashCAN*

MOd (Module) Setup Option

In MOd setup, the serial number of the module assigned to the top or bottom display is entered. The serial number is written on a label on the module. The module assigned to the top display will be assigned the first half of the analog outputs. The module assigned to the bottom display will be assigned to the second half of the analog outputs. The same module can be assigned to both displays or different modules can be assigned to each display.

After entering MOd (i.e. press ENT when "MOd" is displayed), the serial numbers of the available modules will be displayed. Select using ↑ and ↓ followed by the ENT key.

RATE Setup Option

Data is transmitted from modules at the broadcast rate and the programmed averaging that was programmed using the Configuration Tool. This transmitted data can then be further averaged before being displayed on the displays. Separate averaging can be programmed for the top display and the bottom display. RATE does not affect the analog outputs.

The averaging is programmed with values from 0.001 (heavy averaging) to 1.000 (no averaging). The default is 1.000. The averaging is performed as follows:

$$\mathbf{DisplayedValue_t = \alpha \times Parameter_t + (1 - \alpha) \times DisplayedValue_{t-1}}$$

where:

$DisplayedValue_t$ = the new displayed value

α = The user-programmable averaging.
Range: 0.001 (heavy averaging) to 1.000 (no averaging).

$Parameter_t$ = the latest value transmitted by the module

$DisplayedValue_{t-1}$ = the previous displayed value

AOUT (Analog Outputs) Setup Option (dashCAN+ and dashCAN2 Only)

The dashCAN2 display head has two 0 to 5V programmable analog outputs. The dashCAN+ display head has six 0 to 5V programmable analog outputs. The analog outputs are updated at the module Broadcast Rate (see Table 1). This can be as fast as every 5 ms. Keep in mind that the data may be averaged (see **Set Averaging Filters** in Table 1) before being broadcast by the module.

Parameter information from the module assigned to the top display (see MOd) can be sent to analog output 1 (for dashCAN2) or 1, 2, and 3 (for dashCAN+). Parameter information from the module assigned to the bottom display can be sent to analog output 2 (for dashCAN2) or 4, 5, and 6 (for dashCAN+). Only parameters selected as active TPDOs for that module (using the Configuration Tool) can be output.

Here is an example of setting analog output 2 for a dashCAN+ (i.e. A2):

1. Press the SYS key so that “MOd” appears on the top display.
2. Press the ↓ key until “AOUT” is on the top display. Then press the ENT key.
3. Press the ↓ key until “A2” (analog output 2) is on the display. Then press the ENT key.
4. Press the ↑ and ↓ key until the parameter (see Table 2) that will drive A2 is displayed. Then press the ENT key.
5. When 0V is displayed, press ENT. Using the ↑, ↓, and ENT keys, set the parameter value that you want to result in an analog output voltage of 0V on analog output 2. The first time you do this, it may be a little tricky. You are setting one digit at a time and for some numbers, the display will shift to the left so you can set the right-most digits. If you get into trouble when programming, press the SYS key twice to exit and re-enter setup to try again.
6. When 5V is displayed, press ENT. Using the ↑, ↓, and ENT keys, set the parameter value that you want to result in an analog output voltage of 5V on analog output 2.
7. When “AOUT” is displayed, press SYS to return to RUN mode.

If in the above example analog output 4 was being programmed, AOUT, A4, 0V, 5V, and your entries would be shown on the bottom display.

dISP (Display) Setup Option

The dashCAN* display head has two displays, the top display and the bottom display.

In dISP setup, a parameter from the module assigned to the display (see MOd) can be shown on the display. Only parameters selected as active TBDOs for that module (using the Configuration Tool) can be displayed.

Here is an example of setting the parameter to be displayed on the top display:

1. Press the SYS key so that “MOd” is displayed.
2. Press the ↓ key until “dISP” is on the top display. Then press the ENT key.
3. Press the ↑ and ↓ key until desired parameter name is displayed. See Table 2. Then press the ENT key.
4. Press SYS to return to RUN mode.

CONF (Configure) Setup Option

CONF setup appears at the end of the setup list on the bottom display. To enter CONF, press the SYS key so that “MOd” appears on the top display, press the ↓ key until “CONF” appears on the bottom display, and then press the ENT key. CONF is for global dashCAN* setup.

◆ **LEdS (dashCAN+ only)**

The display intensity is programmable. Press the ENT key when “LEdS” appears on the bottom display, press the ↑ or ↓ keys until the display intensity is suitable, press ENT, and press SYS to return to RUN mode.

◆ **1V4V (dashCAN2 and dashCAN+ only)**

"1V4V" commands the analog outputs to go to 1V then 4V. This is used to verify that the data acquisition system reading the analog outputs is correctly reading them.

◆ **LOCK**

"LOCK" locks the MOd, RATE, AOUT, dISP, and LEdS setup. This stops unauthorized modification of the display. Refer to Appendix G for more information.

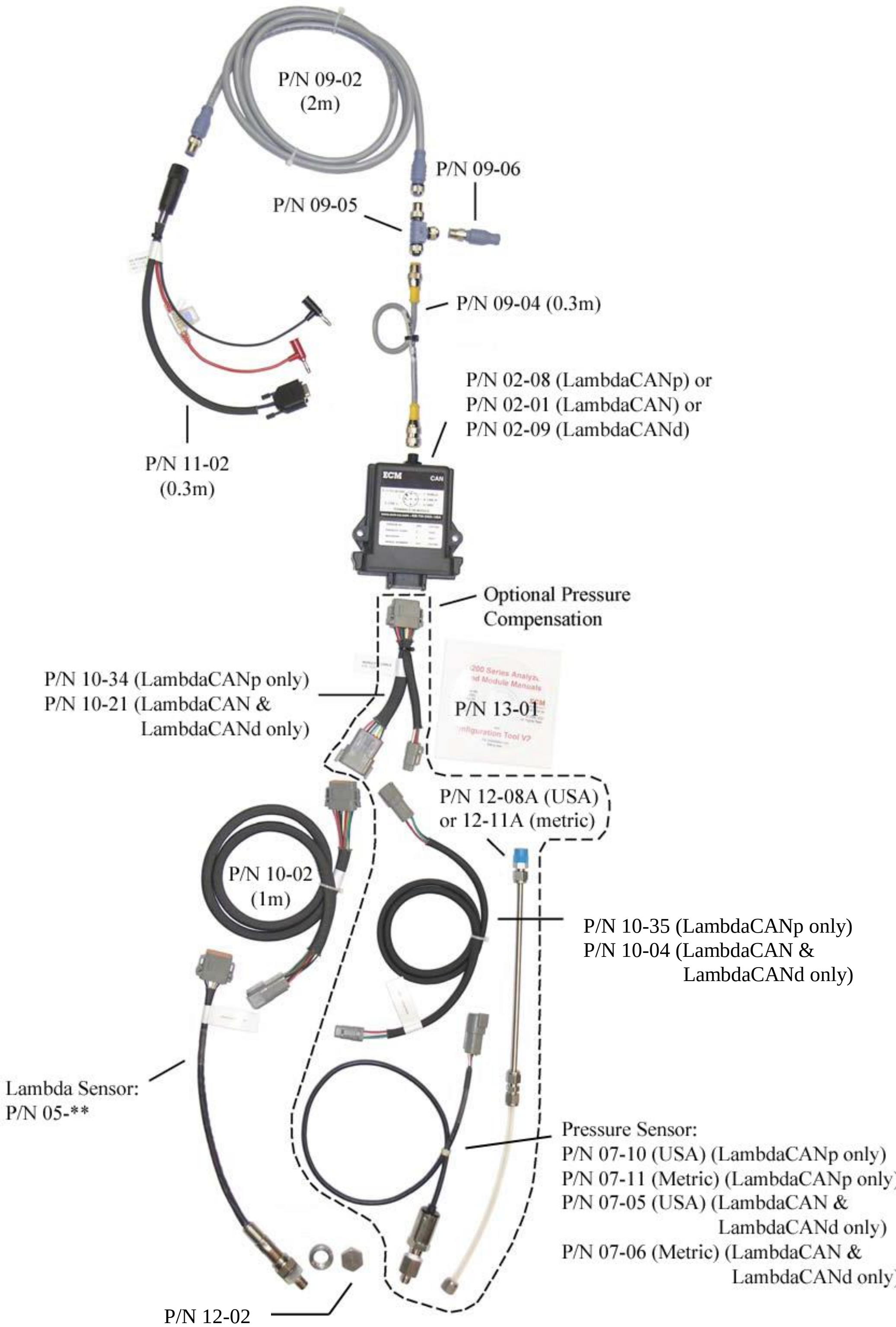
Using the Lambda Sensor Simulator

The Lambda Sensor Simulator outputs currents to simulate a lambda sensor and voltages to simulate a pressure sensor. It is plugged into a LambdaCAN* module instead of the lambda and pressure sensors. When plugged in, if the LambdaCAN* module is operating properly, it will report the correct currents (i.e. lambdas) and voltages (i.e. pressures) output by the simulator.

The simulator is a useful tool to check out a system where there is a problem and it is not known if the sensor, cable, module, or data acquisition system receiving the CAN data from the LambdaCAN* module is the cause of the problem. The simulator checks out everything but the sensors, however once everything else checks out okay, the focus can be put on the sensors. A properly operating LambdaCAN* module monitors lambda sensor condition and can calibrate the lambda sensor. An external pressure source is required to check out the pressure sensor.

Lambda Sensor Simulators can be returned to ECM on a schedule (1 year recommended) for recalibration.

Appendix A: LambdaCAN* Kit Contents



The LambdaCAN* Kit consists of:

	Description	P/N	Quantity
1.	LambdaCANp Control Module LambdaCANd Control Module LambdaCAN Control Module	02-08 or 02-09 or 02-01	1
2.	LambdaCAN Sensor	05-01 (NTK 6mA) or Others Available	1
3.	Lambda Sensor Extension Cable	10-02 (1m)	1
4.	Flexi-Eurofast Cable	09-04 (0.3m)	1
5.	Eurofast "T"	09-05	1
6.	Eurofast Terminating Resistor	09-06	1
7.	2m Eurofast 12mm Cable	09-02	1
8.	DC Power Cable, DB9F, Banana	11-02	1
9.	Lambda Sensor Boss & Plug (18mm x 1.5mm)	12-02	1
10.	Manuals and Configuration Software CD	13-01	1

Optional Pressure Compensation (P-comp) Kit:

	Description	P/N	Quantity
1.	Pressure Sensor, 0-75 psia, 517 kPa For LambdaCANp: 07-10 (USA) or 07-11 (Metric) For LambdaCANd and LambdaCAN: 07-05 (USA) or 07-06 (Metric)		1
2.	Pressure Sensor Extension Cable For LambdaCANp: For LambdaCANd and LambdaCAN:	 10-35 (1m) 10-04 (1m)	1
3.	Pressure Sensor Tubing (686mm)	12-08A (1/4", USA) or 12-11A (6mm, Metric)	1
4.	Module Y Cable For LambdaCANp: For LambdaCANd and LambdaCAN:	10-34 10-21	1

Optional Cables:

	Description	P/N	Quantity
1.	Lambda Sensor Extension Cable	10-03 (2m)	1
2.	Lambda Sensor Extension Cable with Midpoint: Lemo connectors (2 pieces, A & B)	10-42A (1.5m) Controller Side 10-42B (1.5m) Sensor Side	1
3.	Pressure Sensor Extension Cable For LambdaCANp: For LambdaCANd and LambdaCAN:	 10-36 (2m) 10-05 (2m)	1

Optional Power Supplies:

	Description	P/N	Quantity
1.	AC/DC Power Supply, Universal 24VDC @ 4.2A (requires P/N 11-17 Deutsch DTM3M to DB9F)	04-01	1
2.	Vboost Supply, 10-14VDC to 24VDC @ 14.5A	04-02	1
Optional USB-to-CAN Adapter:		13-02	1
Optional Displays:			
1.	dashCAN (just dual display)	01-04	1
2.	dashCAN2 (dual display, 2 analog outputs)	01-08	1
3.	dashCAN+ (dual display 6 analog outputs)	01-05	1
Optional Sensor Simulator:			
1.	For LambdaCANp	SIM700	1
	For LambdaCANd and LambdaCAN	SIM300	

For a full list of options for LambdaCAN*, visit www.ecm-co.com

Appendix B: Module Stand-alone Mode and EIB Mode

CAN data from LambdaCAN* modules can either be taken directly from the modules themselves or from the CAN port of display heads connected to the modules. When data is taken directly from one or more modules, each module must in Stand-alone mode. When data is taken from one or more display heads of a Lambda 5220, or EGR 5230 analyzer, each module must be in EIB mode.

Therefore, the module must be properly configured in Stand-alone mode or EIB mode depending on how it will be used. When LambdaCAN* modules are sold alone, they are delivered in Stand-alone mode. When LambdaCAN* modules are sold as part of a Lambda 5220 or EGR 5230 analyzer, they are delivered in EIB mode.

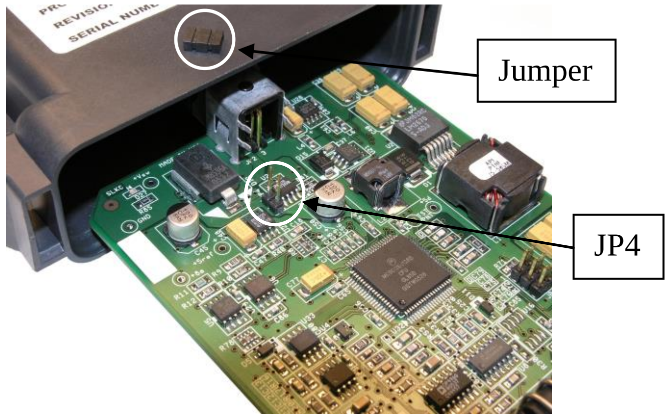
To convert from one mode to the other requires software reprogramming of the module followed by the installation (to set to Stand-alone) or removal (to set to EIB) of a jumper inside the module.

◆ **To convert a module from EIB to Stand-alone Mode**

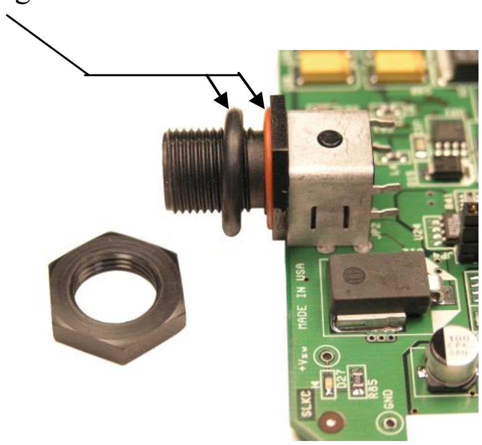
1. Take the nut off the end of the module. Use an 18mm socket without the wrench.

2. Release the two tangs at each side of the module.

3. Slide the PCB out. Install a jumper on JP4.



4. Make sure both O-rings are on the threaded connector.



5. Slide the PCB into the enclosure until the two tangs "click".

6. Put the nut on and tighten ONLY ½ turn from where it is seated. If this nut is tightened too much, the connector will crack and the enclosure will not be sealed.

7. Connect the module to a power supply and a PC (via a CAN communication adapter) using the cabling shown. A sensor does not have to be connected to the module. Note that only one module is connected and a display head is not involved.

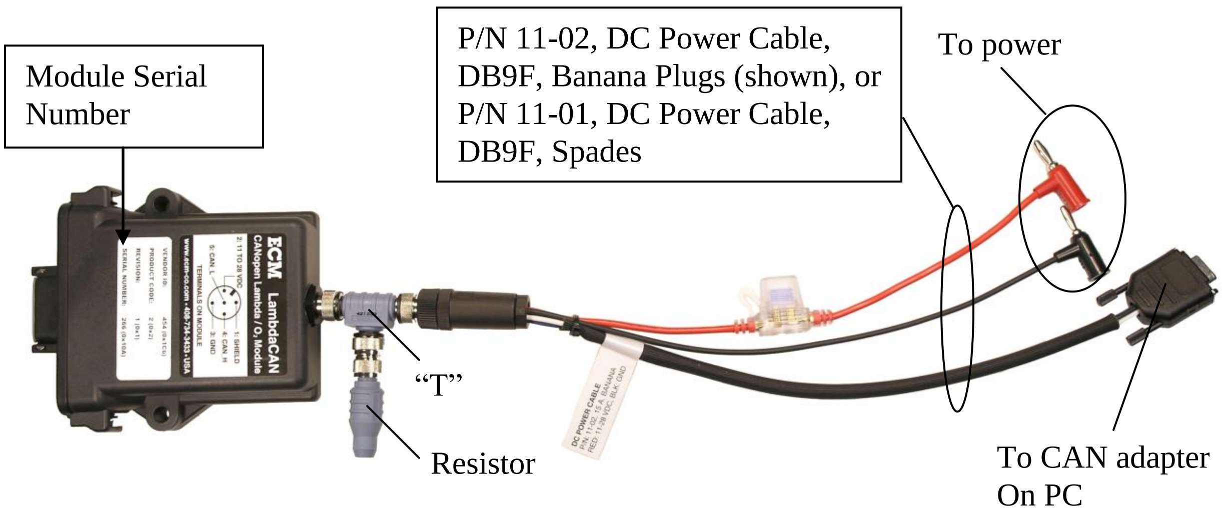
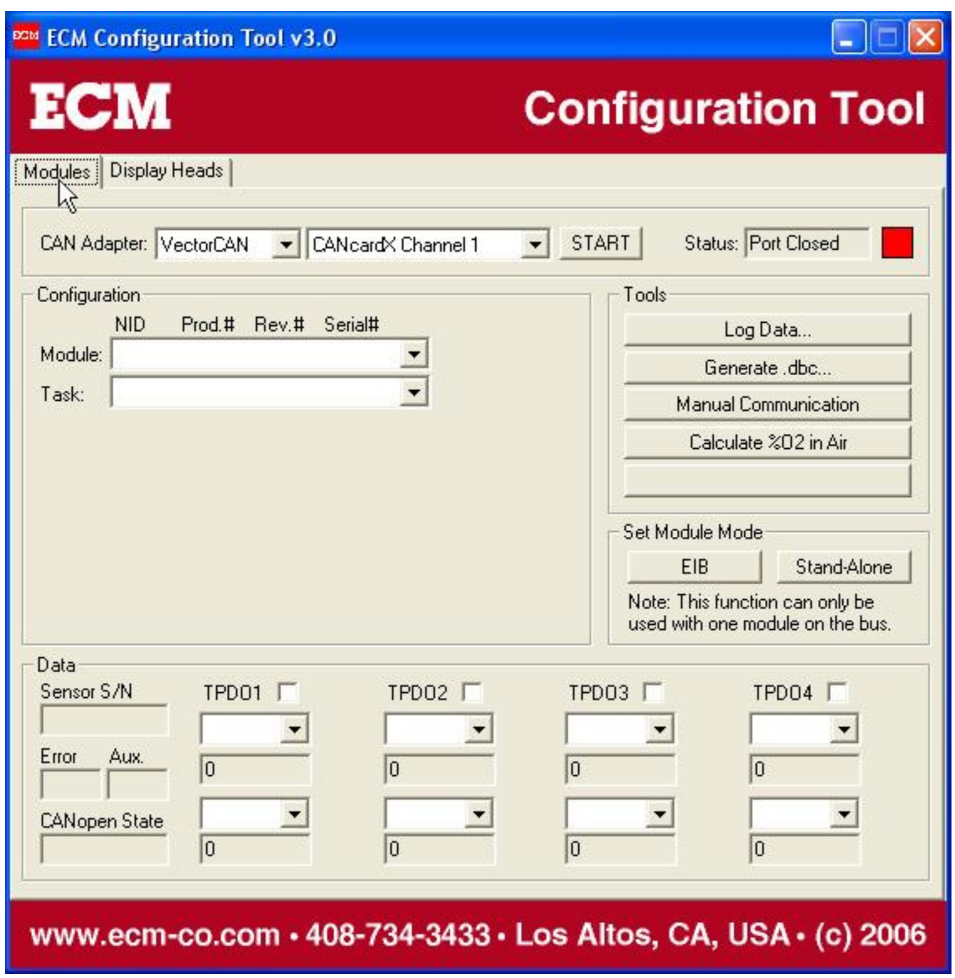


Figure A1: Module prepared for Reprogramming

8. Start the Configuration Tool (software). Click on the "Module" tab. Select the CAN adapter being used. Then start the communication.



9. Click on the "Set to Stand-Alone Mode". Wait for "Done" Message.
   Stop communication and exit program. The module is in Stand-alone mode.

◆ **To convert a module from Stand-alone Mode to EIB Mode**

1. Use the Configuration Tool (software) to "Set to EIB Mode".
2. Remove the jumper on JP4 in the lambda module.

Appendix C: Calculating the %O_2 in Air

The Configuration Tool Software has a routine to calculate the %O_2 in air. If the software is not available, the below may be used.

The oxygen concentration in dry air (zero humidity) is 20.945 and decreases with increasing humidity. The %O_2 in air can be calculated from the barometric pressure (P_b, in mmHg), the relative humidity (Rh), and the saturated water vapor pressure (P_{ws}, in mmHg) by using the following formula:

$$\%O_2 = 20.945\% \times (P_b - P_{ws} \times (Rh/100)) / P_b$$

The saturated water vapor pressure (P_{ws}) is a function of the ambient temperature (T_a) and is given in the table below. For example, at 21 °C, P_{ws} = 18.65 mmHg.

T_a (°C)	0	1	2	3	4	5	6	7	8	9
	P_{ws} (mm Hg)									
0	4.579	4.926	5.294	5.685	6.101	6.543	7.013	7.513	8.045	8.609
10	9.209	9.844	10.518	11.231	11.987	12.788	13.634	14.530	15.477	16.477
20	17.535	18.650	19.827	21.068	22.377	23.756	25.209	26.739	28.349	30.043
30	31.824	33.695	35.663	37.729	39.898	42.175	44.563	47.067	49.692	52.442
40	55.324	58.34	61.50	64.8	68.26	71.88	75.65	79.60	83.71	88.02
50	92.51	97.2	102.09	107.2	112.51	118.04	123.80	129.82	136.08	142.60
60	149.38	156.43	163.77	171.38	179.31	187.54	196.09	204.96	214.17	223.73
70	233.7	243.9	254.6	265.7	277.2	289.1	301.4	314.1	327.3	341.0
80	355.1	369.7	384.9	400.6	416.8	433.6	450.9	468.7	487.1	506.1
90	525.76	546.05	566.99	588.60	610.90	633.9	657.62	682.07	707.27	733.24

1 mmHg = 0.01934 lbf/in2 = 1 torr

Appendix D: Interpreting Data from LambdaCAN* Modules

◆ Comparing to Spindt and Brettschneider Calculations

LambdaCAN* calculates Lambda, AFR, FAR, and PHI numbers comparable to those determined by mass flowrates of air and fuel into the engine. Lambda calculations based on gas-bench analyzer data (i.e. measurements of CO, CO2, etc) do not always give these same values. This fact is well documented in the literature.

However, some organizations prefer to call their specific (and often unique) Spindt or Brettschneider calculations of Lambda the "true" values. To satisfy such needs, ECM has provided the Delta Lambda Table and Delta O2 Table features.

◆ Before and After Catalyst Measurements

The Lambda sensor operates on a diffusion mechanism. Molecules leaving the combustion chamber (O2, CO, CO2, H2O, H2, HC, NOx, N2, etc) diffuse into the sensor where oxidation, oxygen liberating, and oxygen pumping occurs. O2, Lambda, AFR, FAR, and PHI are determined from the oxygen pumping rate and assumptions about the combustion products entering the sensor. One such assumption is that the exhaust exists in a certain chemical composition. Exhaust gases entering and exiting a catalytic converter have different chemical compositions. LambdaCAN* assumes that the gases are engine-out (not catalyst-out). Therefore measurements made at the exit of a catalyst will require a Delta O2 Table and a Delta Lambda Table correction to be accurate. This is sometimes called "H2 Correction".

◆ Wet versus Dry Measurements

Percentages of components in the exhaust of an engine are expressed as percentages (molecule count or volume) or ppm. The numerator used to calculate these percentages and ppm contains the molecule-of-interest count and the denominator contains the total number of molecules in the sample containing the "count". One of the molecules produced by the process of combustion, and in significant quantities (ex. 15%), is H20.

Percentages and ppm calculated by LambdaCAN* consider the water molecules in the denominator and are called "wet" percentages (or ppm). This only makes sense since the sensors are directly in the exhaust where the H2O is present. In contrast to this, classical gas-bench analyzers almost always remove the water before the gas sample reaches the analyzers. This is because the analyzers cannot tolerate condensed water. Therefore, gas-bench analyzers will report O2 percentages and NOx ppms as "dry" percentages (or ppm) (i.e. without H2O molecules counted in the denominator).

Therefore, when comparing "wet" (or "true") O2 and NOx data from LambdaCAN* to "dry" O2 and NOx data from gas-bench analyzers, realize that the dry percentages and ppm will be 5% to 10% higher than the wet (or true) readings. The wet to dry conversion for %O2 is a function of %O2 (or Lambda) and can be performed by using a Delta O2 Table.

◆ Equilibrium versus Non-Equilibrium O2

Gases exiting the combustion chamber of an Otto or Diesel cycle engine are not at chemical equilibrium. As the exhaust valve opens, the gas temperature and pressure drop so quickly that many chemical reactions still occurring are slowed down or "frozen" at non-equilibrium values (for the specific temperature and pressure they are at). This degree of chemical non-equilibrium is affected by exhaust valve timing and ignition timing. The lambda sensor is hot and highly catalytic and will bring the exhaust they sample closer to chemical equilibrium. The result of this is that the %O2 measured by the lambda sensor will be close to equilibrium levels (typically 0.5% lower than actual engine-out). For example, at Lambda = 1 conditions, the %O2 reported by the lambda sensor will be 0 (the chemical equilibrium %O2 value) as opposed to the actual frozen equilibrium %O2 value of approximately 0.5%. Keep this in mind when comparing gas-bench measured %O2 with lambda sensor-measured O2. This difference can be corrected for by a Delta O2 Table.

◆ Lambda Sensor-Measured O2 versus NOx Sensor-Measured O2

Both the lambda sensor and the NOx sensor will report %O2. However, the NOx sensor will not measure %O2 (and hence Lambda, AFR, FAR, and PHI) as accurately as the Lambda sensor. This is because in the lambda sensor, the electrodes measuring %O2 are pure platinum. Pure platinum electrodes give best %O2 measuring performance. In the NOx sensor, the %O2 measuring electrodes have some gold in them which keeps the electrodes from destroying the NOx in the exhaust. NOx is measured in a second set of electrodes downstream of the %O2 measuring electrodes. If the first set of electrodes were pure platinum, the NOx would not reach the second set.

◆ Pressure

The main source of error influencing O2 (Lambda, AFR, FAR, PHI) measurements is exhaust pressure. This error is more significant the further from Lambda = 1 the exhaust stoichiometry gets. Pressure compensation (P-comp) data is stored in the memory chip of every Lambda and NOx sensor. ECM offers P-comp kits for all of its LambdaCAN* kits. It is highly recommended that pressure compensation be used with any lean burn, HCCI, or diesel engine.

◆ Comparing to an ETAS LA4 Meter

Many ETAS meters with new sensors will not correctly show the %O2 when the sensor is held in air. The actual %O2 in air depends on the humidity and will be almost always less than 20.945%. Since the measurement of %O2 is the fundamental mechanism by which the wideband sensor determines Lambda, the effect that this error has on meter accuracy is obvious. This %O2 measurement error will have a negligible effect at Lambda=1 but the effect on Lambda will increase the further Lambda gets from Lambda=1.

All lambda meters should read the correct %O2 when the sensor is held in air. Due to sensor aging and changes in ambient pressure (due to weather and altitude changes), this may not be the case. However, LambdaCAN*'s "Span O2" feature allows to user to restore the system's accuracy.

The O2 span is performed by holding the sensor in air and entering the actual %O2 in air. However, if you are comparing a LambdaCAN* to an ETAS meter and you want the LambdaCAN* to match the ETAS meter, then you must span the LambdaCAN* to whatever the ETAS meter says the %O2 is in air – even if it is wrong.

Therefore, if you want a LambdaCAN* to match an ETAS meter, whether or not the ETAS meter reads the correct %O2 in air, do the following:

1. Make certain that the "LA42.TXT" (for the LSU4.2 sensor) or "LA49.TXT" (for the LSU4.9 sensor) curve is downloaded to the LambdaCAN* module as the "Delta Lambda Table". This is for the case of matching the "Default Curve" in the ETAS meter. Other curves are available and the user can create his own curve.

2. Turn on the ETAS meter and hold the sensor in air. Read the %O2 on the ETAS meter. This value should be 20.945 or less since 20.945 is pure air with no humidity. We have seen values of over 22%.

3. O2 span the LambdaCAN* to the %O2 given on the ETAS meter even if it is wrong.

This procedure is to be used only if you want the LambdaCAN* to match a specific ETAS meter and sensor combination. Normally, the LambdaCAN* are spanned to the correct %O2 in air.

Appendix E: Setting Up ETAS INCA for ECM Modules

Hardware Setup: Using ETAS ES591.1

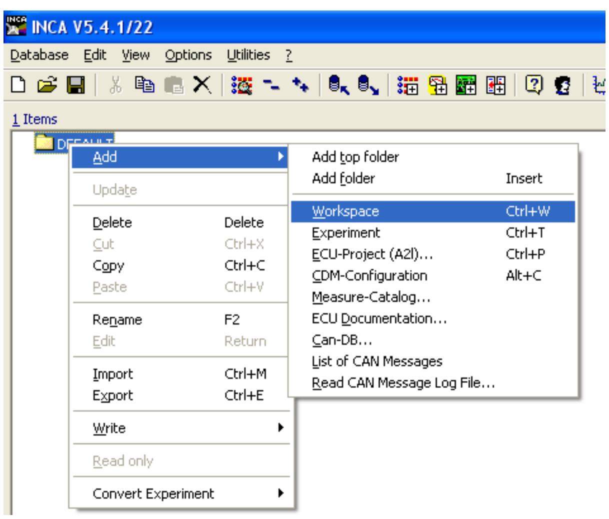
1. Connect the power port to a power source between 6V and 32V.
2. Connect the Ethernet port directly to the Ethernet port on your PC. This port does not use an internet/intranet connection like a router.
3. Connect either the CAN1 or CAN2 port to a CAN network (i.e. ECM modules or display heads).

Software Setup: Using ETAS INCA V5.4.1, Hotfix 22, GM Install

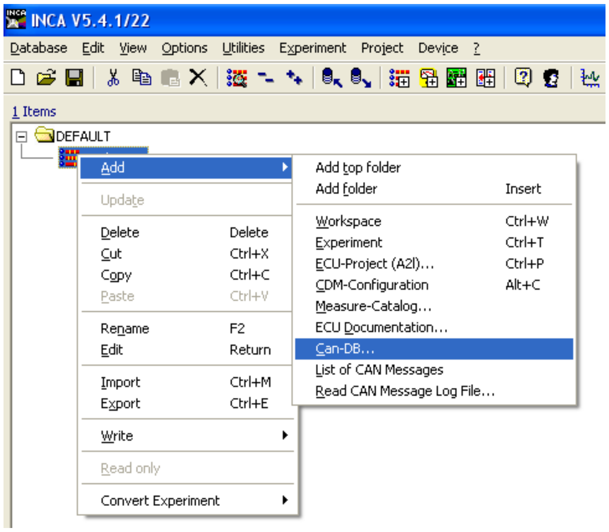
1. Double click the INCA V5.4 icon to open the software.

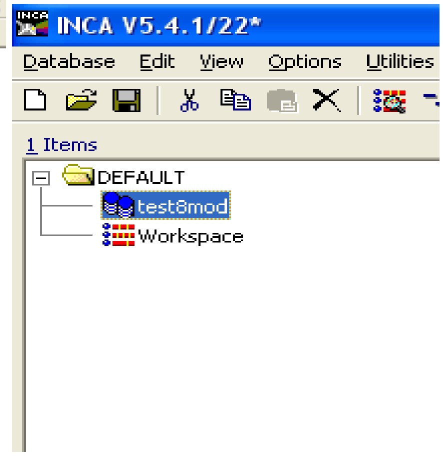
2. **Create a new Database**. In the Database menu, select New. Give your database a name (i.e. a folder name). In INCA, a Database means the current working directory. Each project is created in a unique directory. When INCA is opened, it will default to the last Database that was used.

3. **Add a new Workspace**. Right click on the "DEFAULT" folder icon, select Add > Workspace. You can rename it to whatever you want.

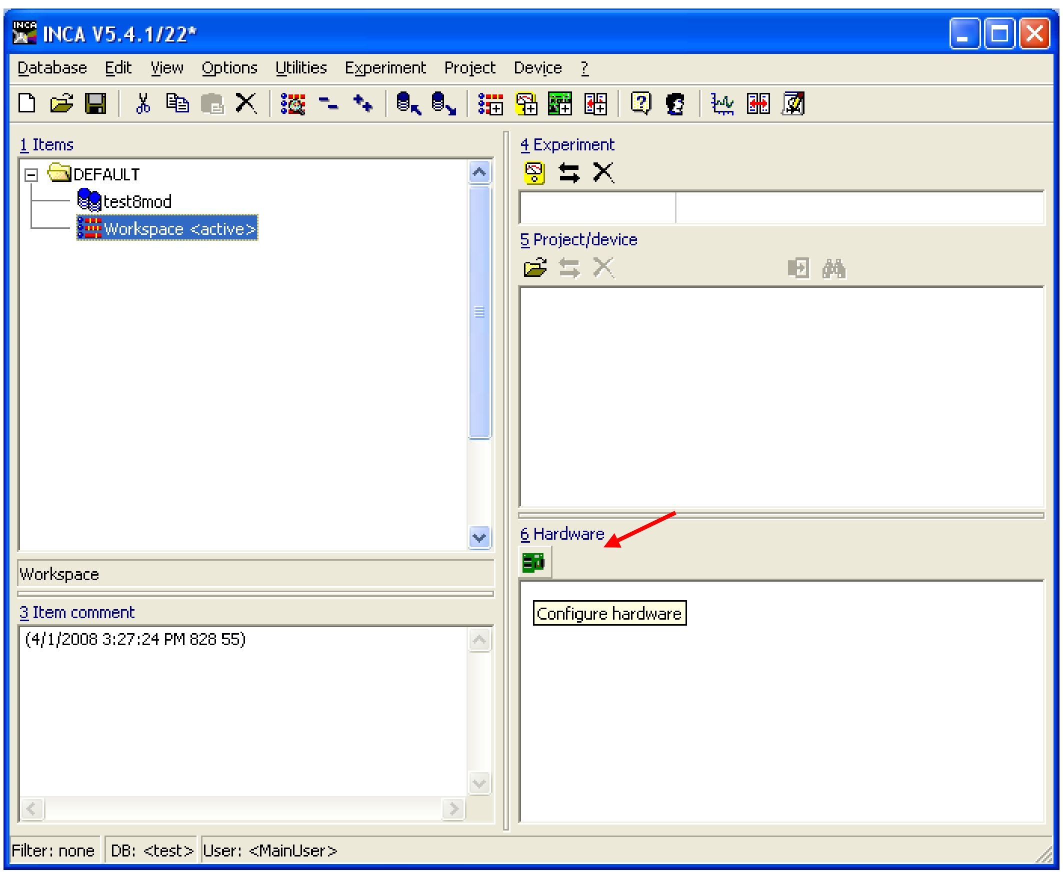


4. **Add a new dbc file for your project.** Right click on the workspace you created in step 3, select Add > Can-DB. Browse to your dbc file and click open. In this example, we are using a file named test8mod.dbc. An INCA log window will pop up. You can ignore this.

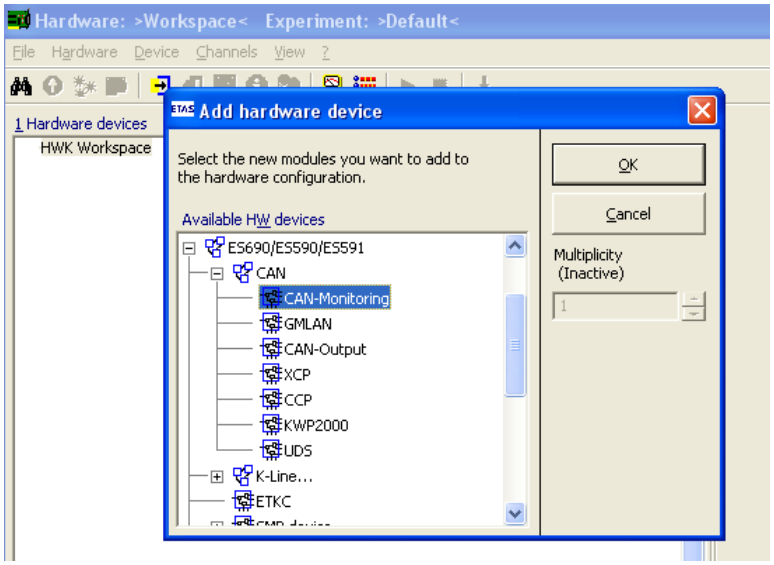




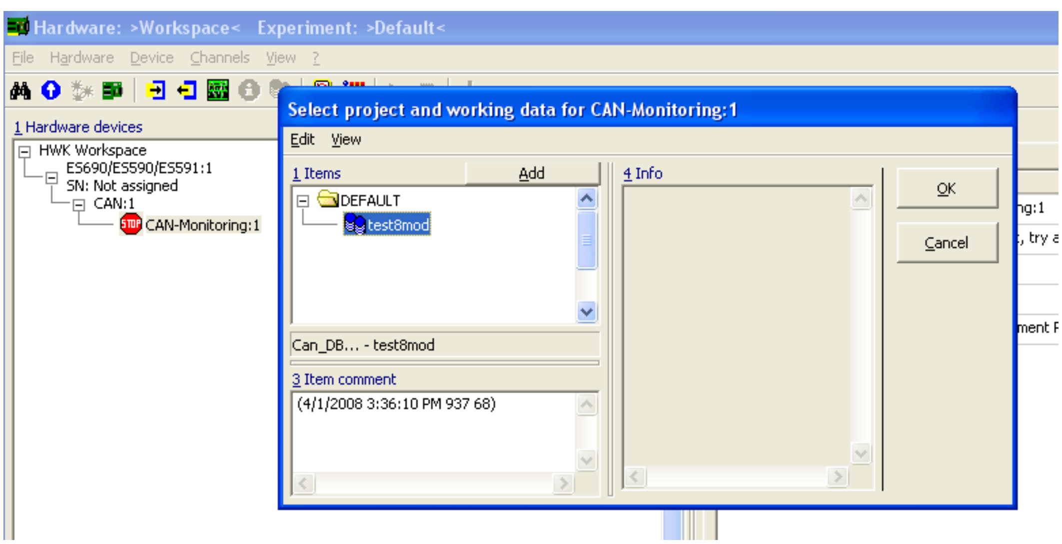
5. **Configure the hardware.** Click on the icon for the workspace you created in step 3. Open the Hardware Configuration icon under the section text “6. Hardware”. A hardware configuration window will open.



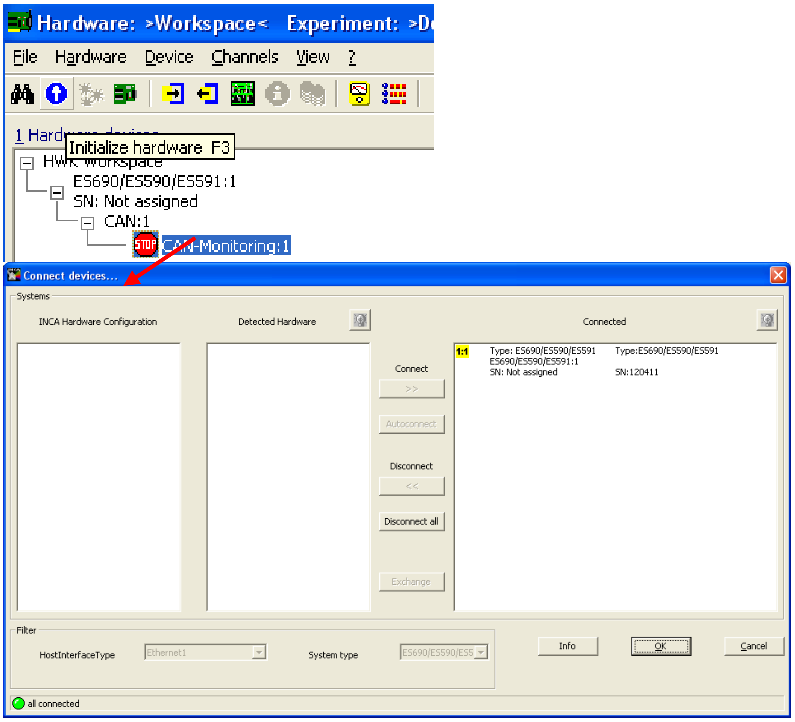
6. **Select the hardware.** In the hardware configuration window, right click the "HWK Workspace" listed under the section text "1. Hardware Devices", and select Insert. Select the ETAS device you wish to use. In this example, we are using an ETAS ES591.1. Expand the selection tree by clicking the "+" next to the hardware device model. Expand the CAN selection and select CAN-Monitoring. Click OK.



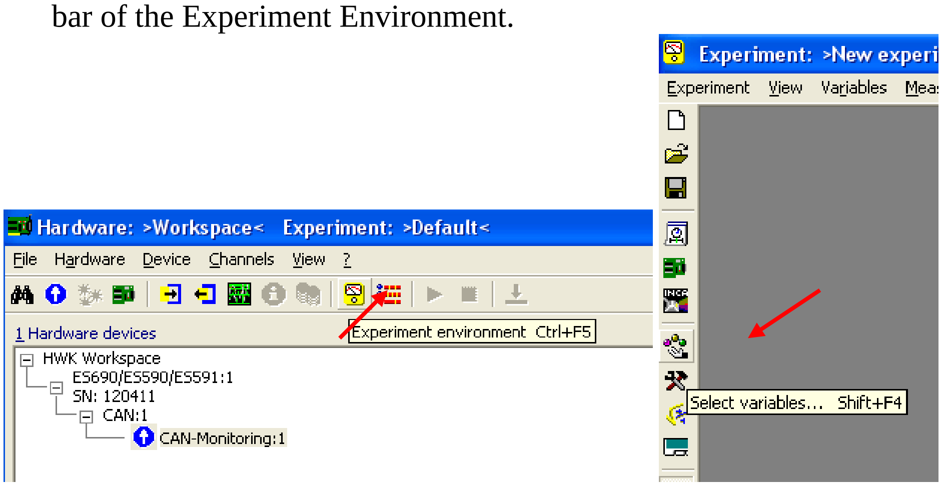
7. **Associate the dbc.** When you clicked OK in the last step, another window will pop up that will allow you to select a dbc that you have added to your workspace from step 4. Expand the selection tree, select your dbc file, and click OK.



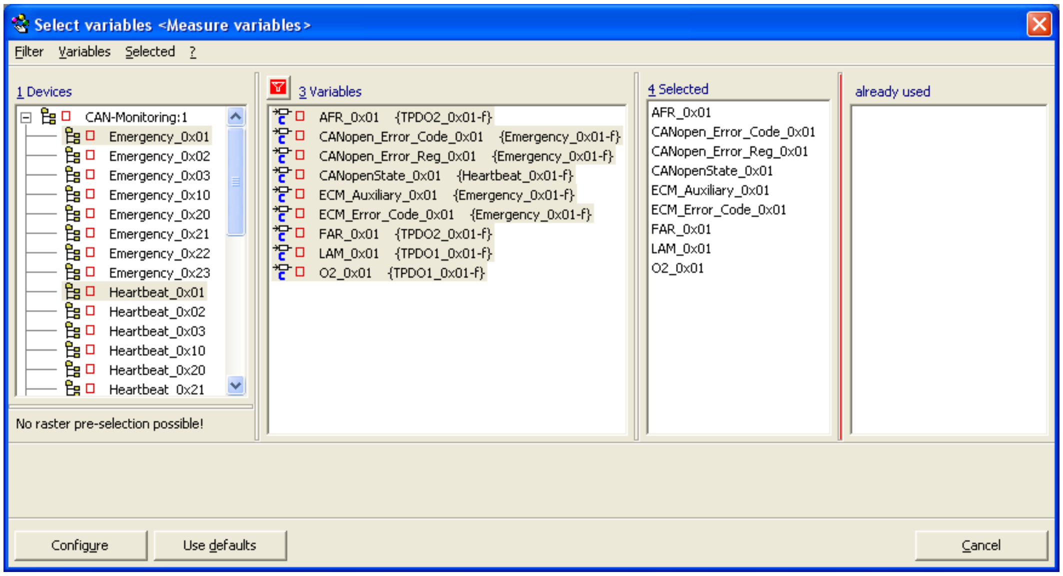
8. **Initialize hardware.** The hardware is currently stopped, as indicated by the red stop sign icon next to the selected hardware. You must initialize it before you can use it to collect data. Click on the Initialize Hardware button on the upper tool bar and wait for the hardware to complete its initialization. Another window will pop up to confirm the device to connect to. Click OK.



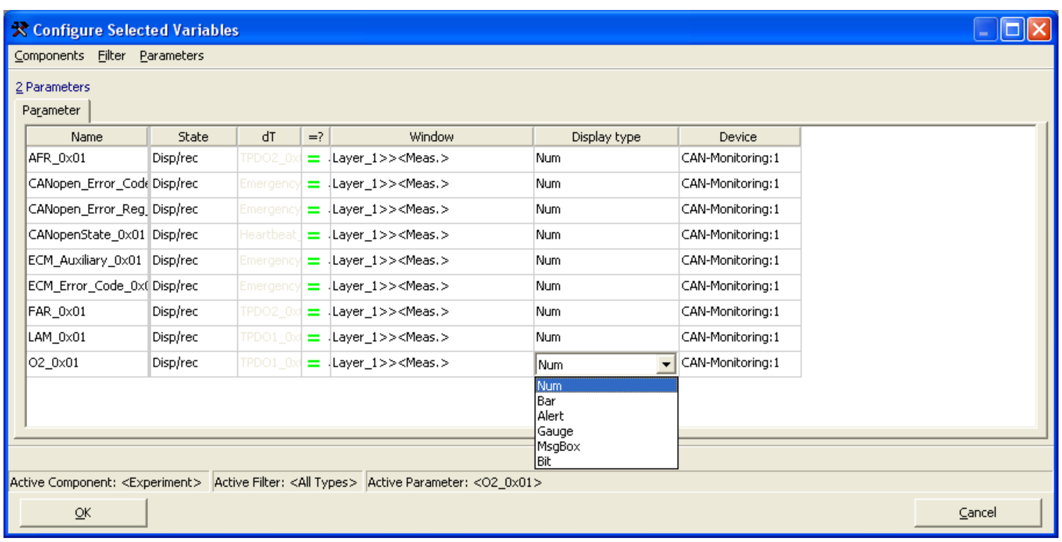
9. **Open an Experiment Environment.** Click on the Experiment Environment button on the upper tool bar to open an Experiment Environment. The Experiment Environment is where you can setup the monitoring of the CAN bus. By default, the Experiment Environment will be blank. You must select the variables from the dbc file that you wish to monitor. Click on the Select Variables icon in the left hand tool bar of the Experiment Environment.



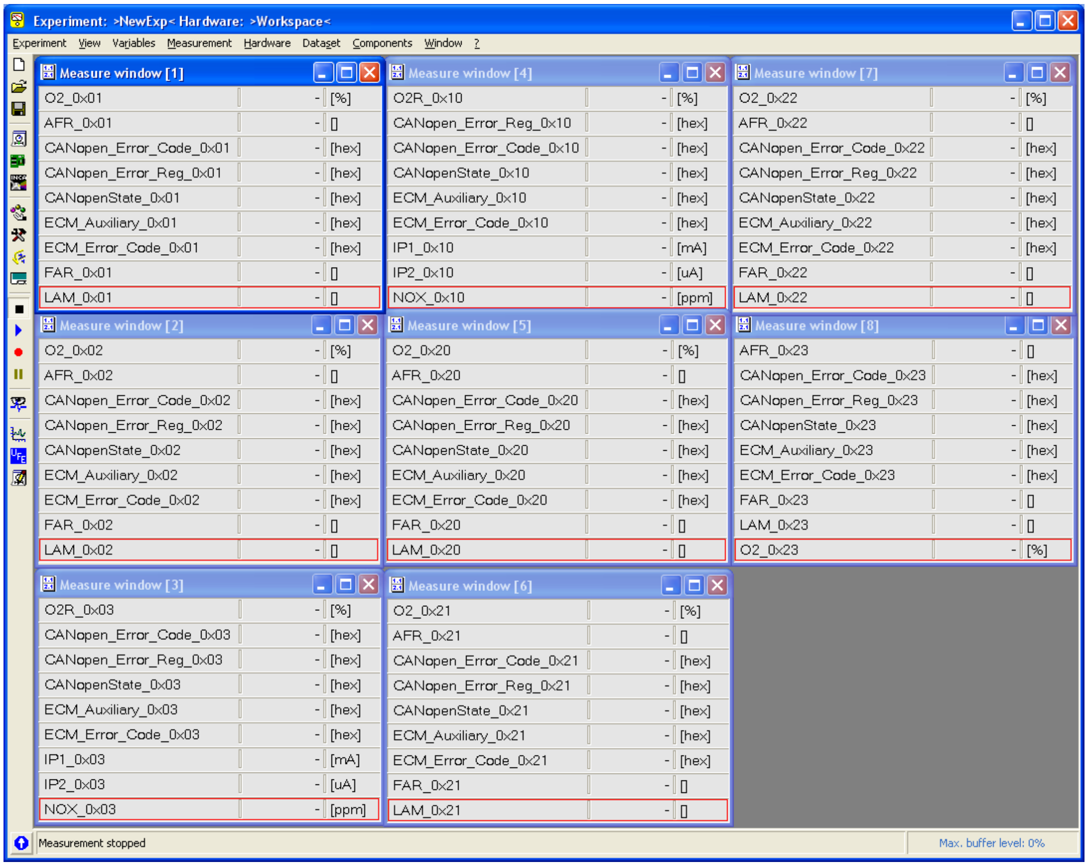
10. **Select and Configure Variables.** Select the variables that you wish to monitor in the Experiment Environment. These variables names are based on the data found in the dbc file. Click Configure.



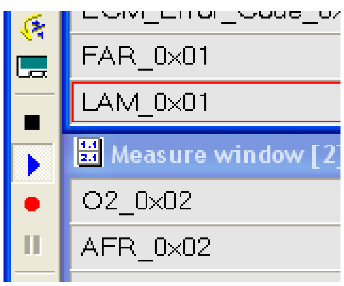
11. Another window will pop up to configure each selected variable. You can configure, for each variable, whether to record or simply display the data, how the data will be displayed (graphs, charts, gauges, numeric, etc.). When complete, click OK. We have left all configurations at default for this example.



12. A new sub-window will be added to the Experiment Environment. You do not need to select all the variables you want to monitor all at once. You can click on the Select Variables icon again at a later time to add more variables. Each set of variables you add will be placed in a new sub-window unless it is configured to join an existing sub-window. In this example, we have created a sub-window for each of the eight modules in the dbc file.



13. **Start CAN monitoring.** Right now there is no data displayed. That is because the CAN monitoring is stopped. To begin CAN monitoring, click on the Start Visualization icon (blue triangle) on the left hand tool bar. To stop CAN monitoring, click the Stop Measuring icon (black square) on the left hand tool bar. To begin recording the data, click on the Start Recording icon (red circle) on the left hand tool bar.



Appendix F: Setting Up ATI Vision for ECM Modules

NOTE: While shown here for a single LambdaCAN* module, the same procedure applies for any of ECM's CAN-based devices (ex. NOxCAN) as well as for multiple modules simultaneously connected on the same bus.

Introduction

Connecting ECM LambdaCAN hardware to ATI VISION software is simple and does not require any third-party software interface. Using the ECM Configuration Tool software to produce a .dbc database file, and the ATI VISION CANMonitor interface, any available hardware CAN interface can be used to read LambdaCAN data.

Hardware Setup

A typical hardware configuration is shown in Figure 1. In this example, a Kvaser Leaf Light CAN-USB adapter is used. Other supported adapters have a similar procedure. Connect the DB9 CAN connector of the LambdaCAN to the PC to CAN adapter. Supply 11-28V DC (5A min. supply) to the LambdaCAN.

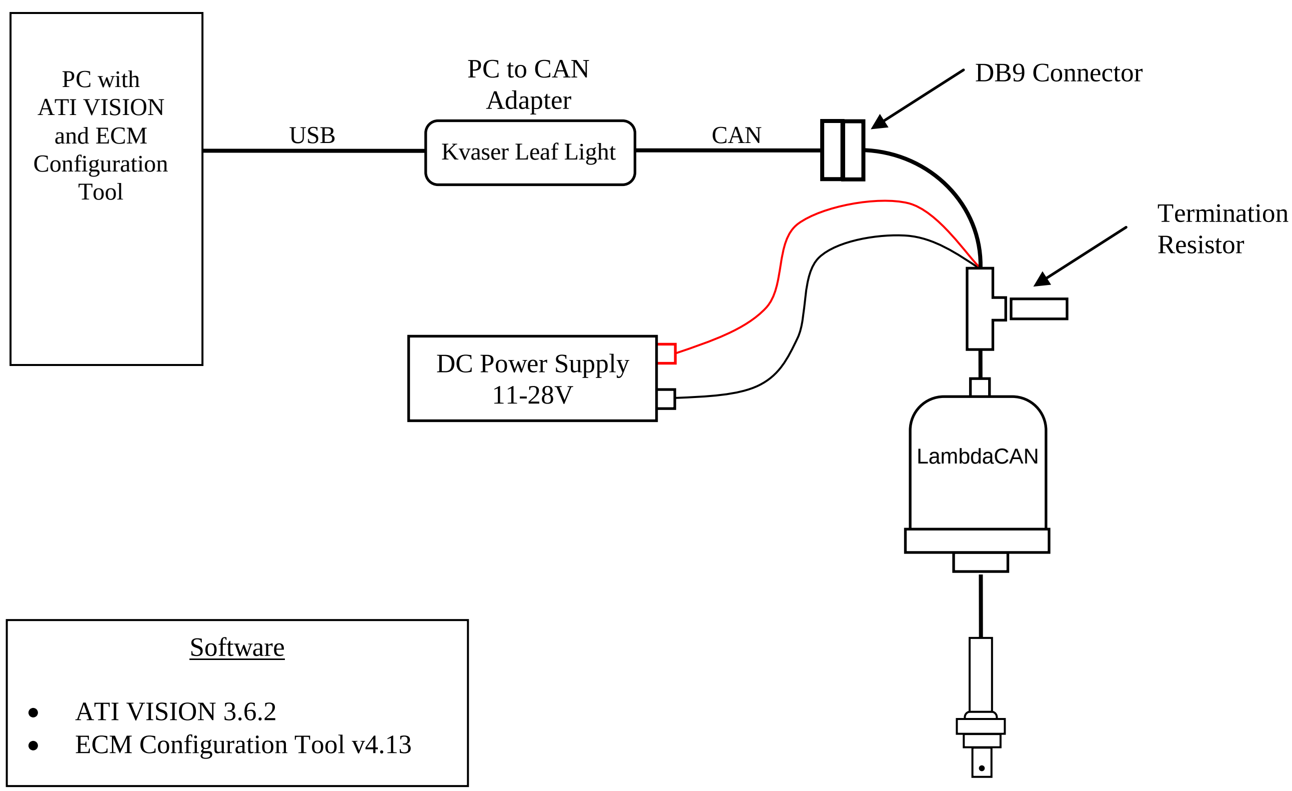


Figure 1: Equipment Schematic Layout

Creating a .dbc File

The ECM Configuration Tool is used to create a .dbc database file for describing the CAN messages broadcast from the LambdaCAN. All ECM products with a CAN interface use the CANopen protocol at 500kHz by default. To generate a .dbc file using ECM Configuration Tool:

1. Connect hardware as shown in Figure 1. Ensure LambdaCAN bi-color LED indicator near sensor connector is visible (green during normal operation, flashing red without sensor attached).
2. Run ECM Configuration Tool software, and select the Modules tab (or the Analyzers tab if connecting to 5200 series analyzers).
3. Select CAN adapter from drop down menus as shown in Figure 2, and click START.
4. After LambdaCAN module(s) have initialized, select desired parameters to transmit from the TPDO drop down menus for each module.
5. Click Generate .dbc, and save this file in a location such as the VISION Projects folder.
6. Click STOP to end CAN connection.

NOTE: Whenever TPDO's are modified, a new .dbc file must be created.

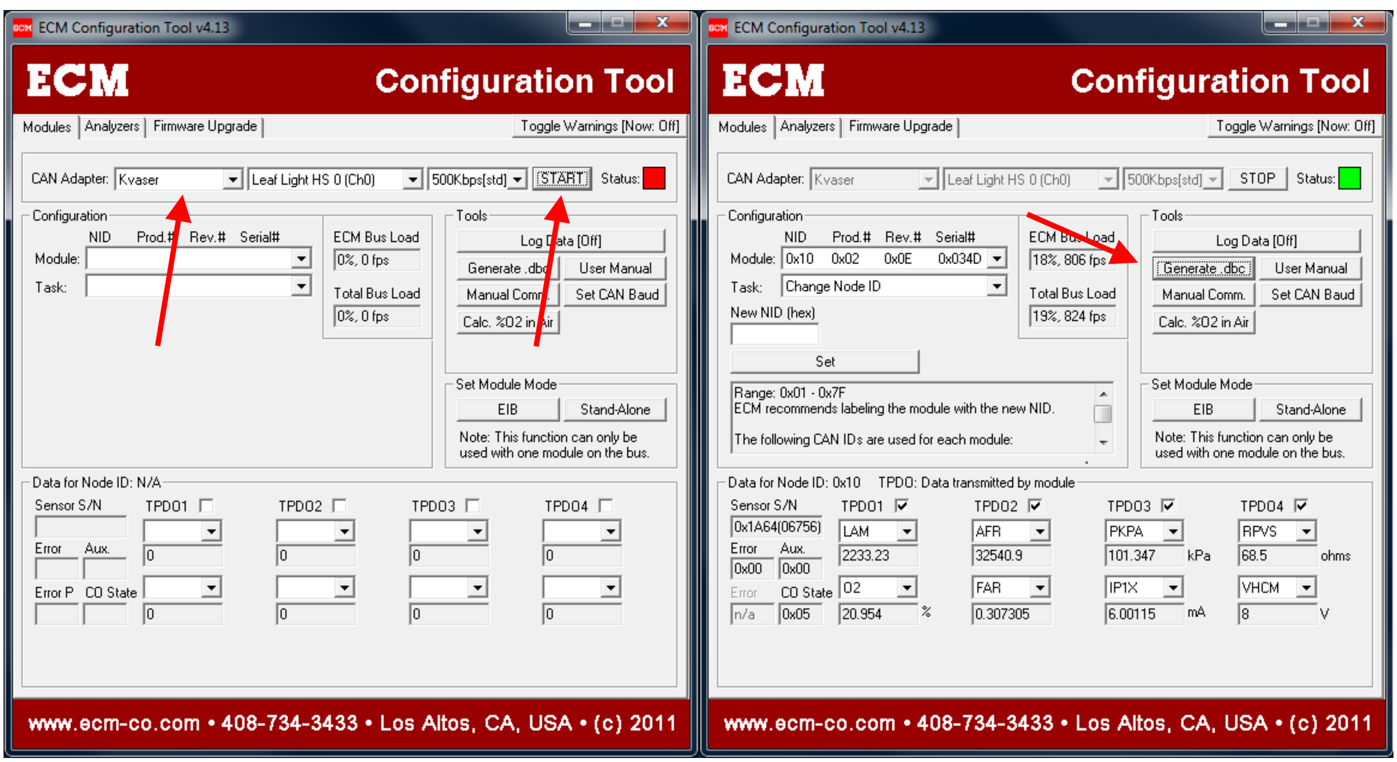
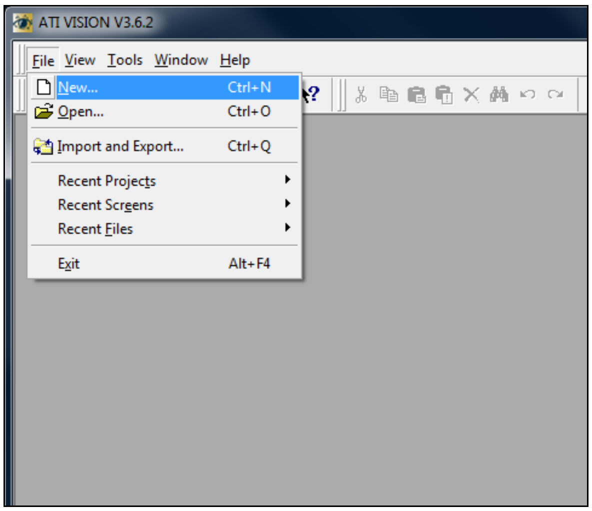


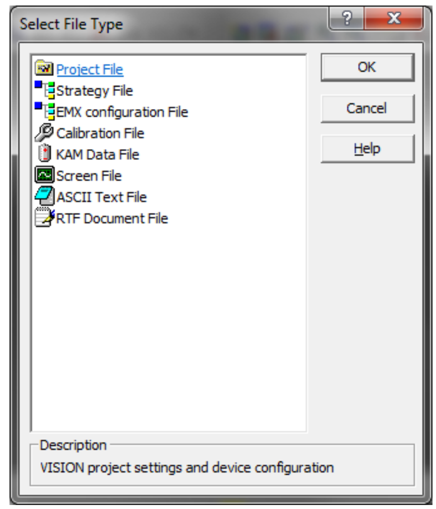
Figure 2: ECM Configuration Tool

Setup CANMonitor using ATI VISION

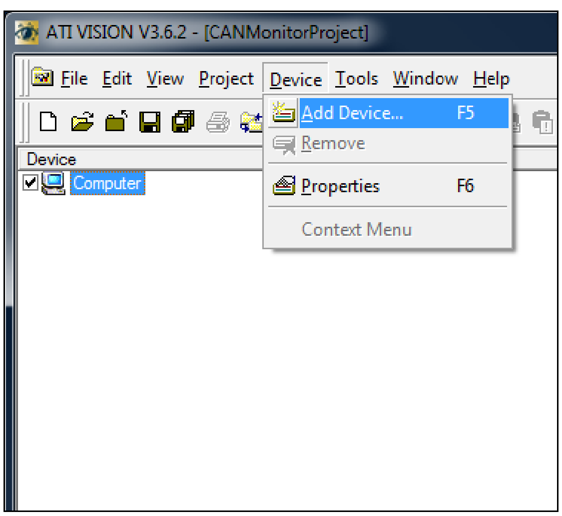
ATI VISION CANMonitor provides a method of reading general purpose information from any available CAN channel. The .dbc file generated by the ECM Configuration Tool is used to describe the format of the information available to VISION. To setup a CANMonitor in ATI VISION:

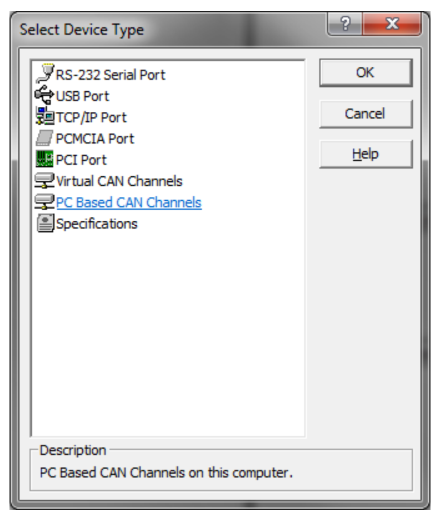
1. Run ATI VISION and open an existing Project File or create a new one by clicking File > New > Project File. In this example the Project has been named CANMonitorProject.



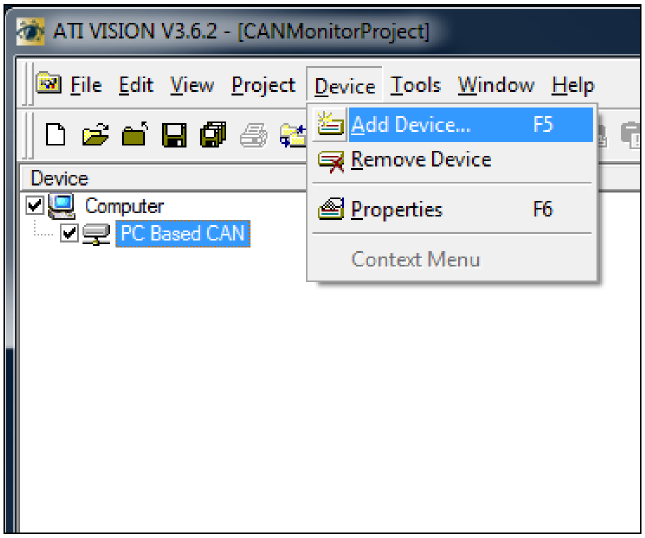


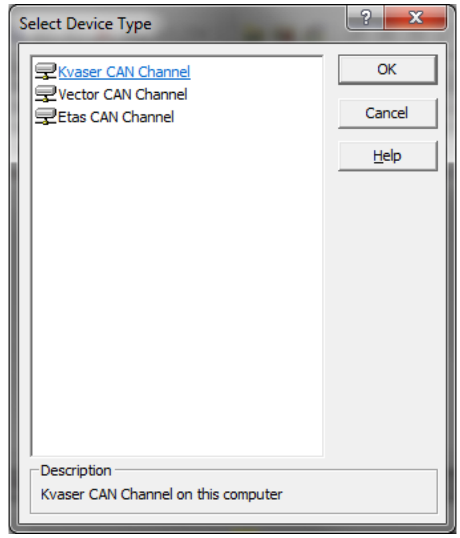
2. Add a Device by clicking Device → Add Device, select PC Based CAN Channels from the list.



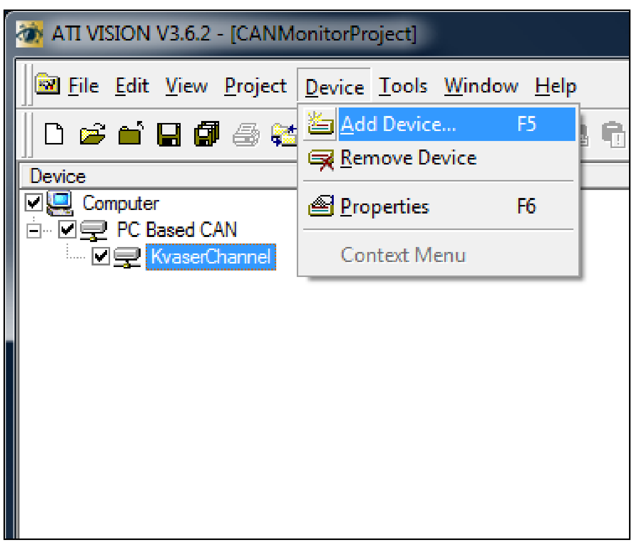


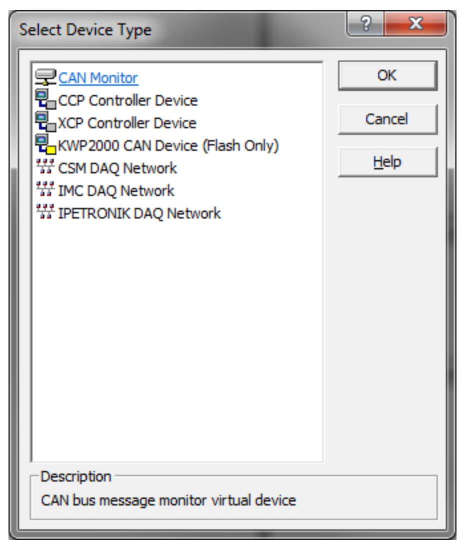
3. Add a physical hardware device by clicking Device → Add Device, and select Kvaser CAN Channel.



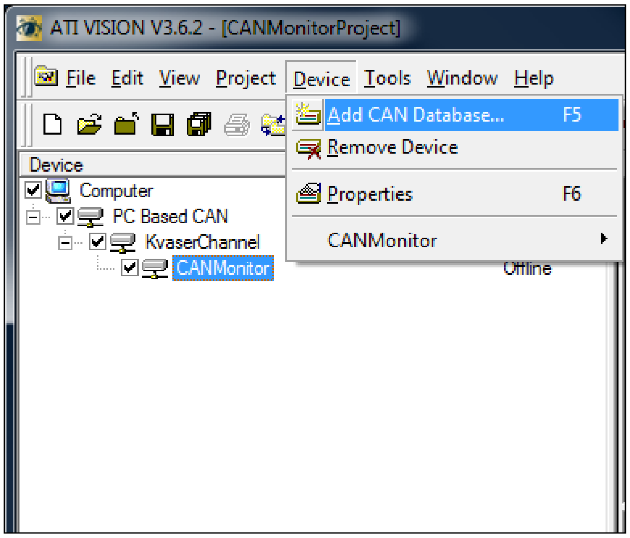


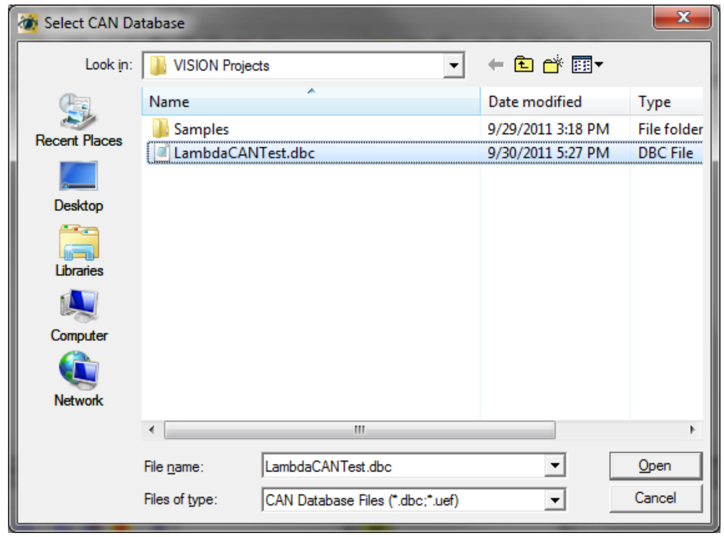
4. Select a CANMonitor device by again clicking Device → Add Device, and select CANMonitor.



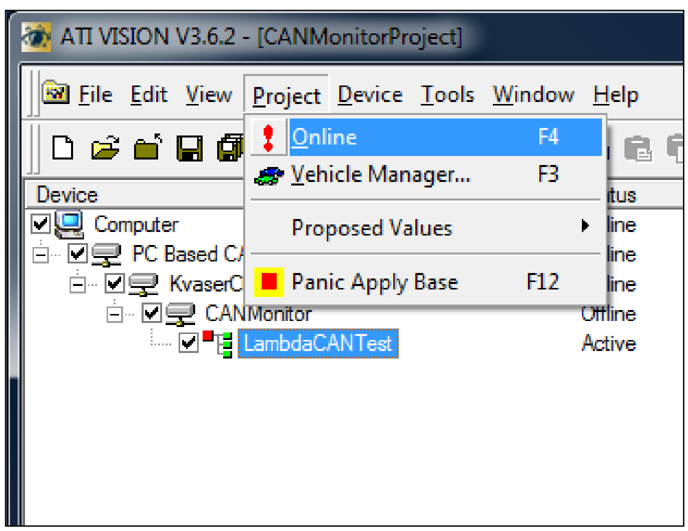


5. Add the .dbc file generated from the ECM Configuration Tool to CANMonitor by clicking Device → Add CAN Database and browsing to the previously created .dbc file.

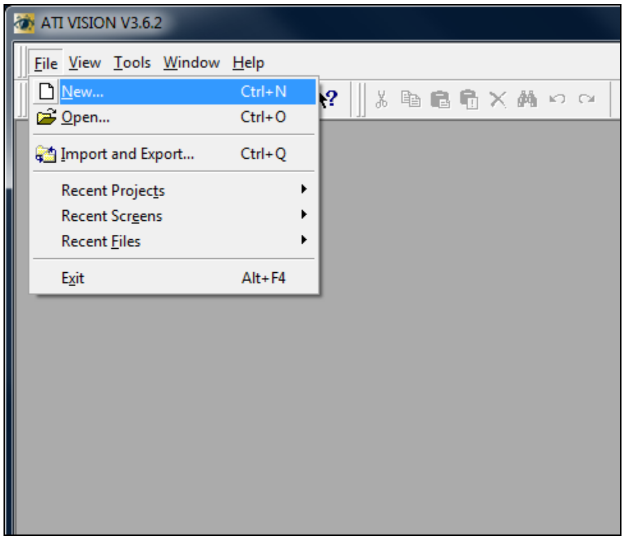


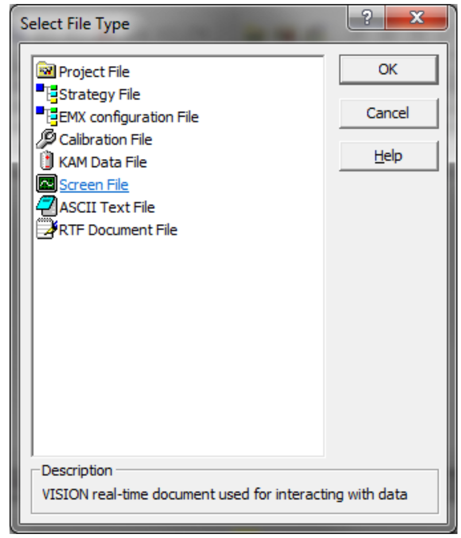


6. Enable the hardware by clicking Project → Online. The status of all of the devices should now show a Status of Online, and a value should appear in the Data Rate column of the Project window.

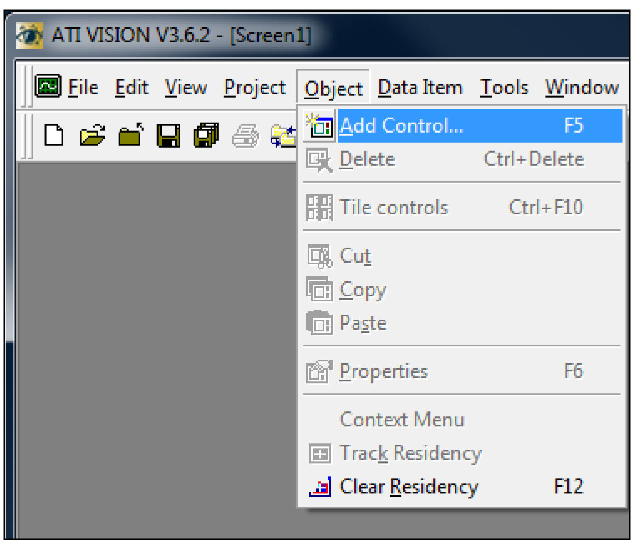


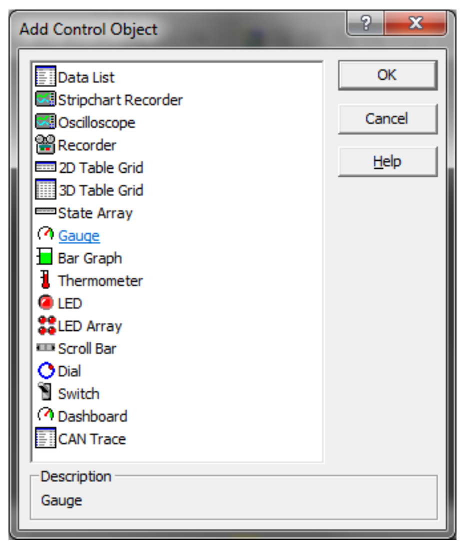
7. To view data, create a new Screen File and add a Control. Click File → New → Screen File



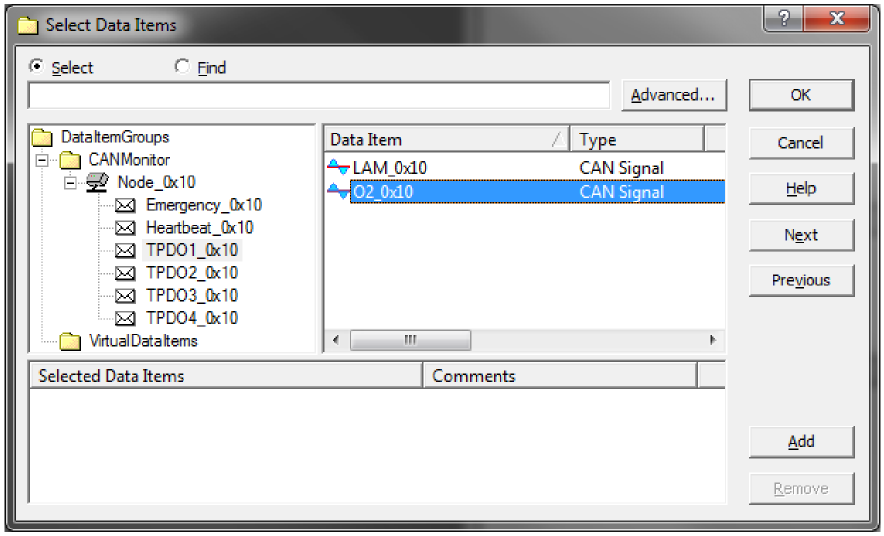


8. Select Object → Add Control → Gauge

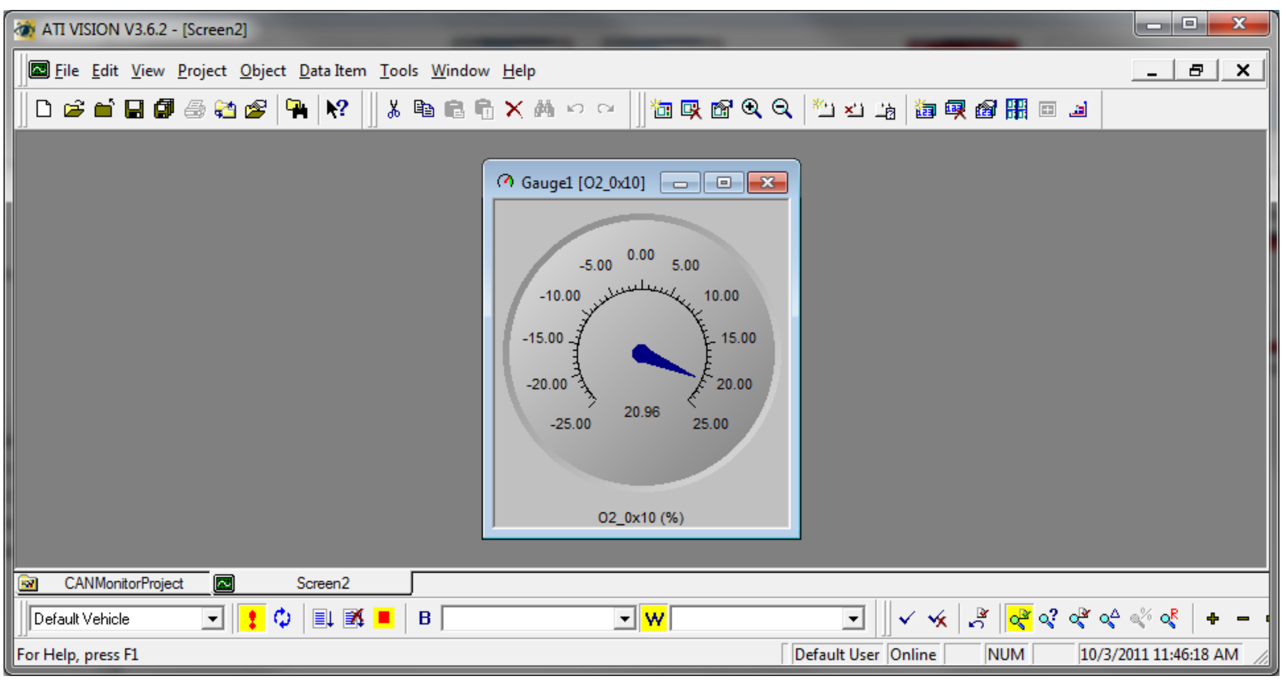




9. In the Select Data Items window open the CANMonitor file tree to view all of the available signals. Here the O2% from Node 0x10 has been selected. Click OK to add the Data Item to the Control.



10. Data should be visible on the gauge.



Appendix G: LOCKing and unLOCKing dashCAN*

When dashCAN* is locked, its setup cannot be modified.

◆ To LOCK dashCAN*

1. Press SYS until “MOd” is displayed.
2. Press ↓ until “CONF” is displayed. Then press ENT.
3. Press ↓ until “LOCK” is displayed. Then press ENT.
4. “50” will be displayed. Press ↑ until “60” is displayed. Then press ENT.
   dashCAN is now LOCKed.

◆ To unLOCK dashCAN*

1. Press SYS until “LOCK” is displayed. Then press ENT.
2. “50” will be displayed. Press ↑ until “60” is displayed. Then press ENT.
   dashCAN is now unLOCKed.

If an unauthorized person learns that 60 is the key number, contact ECM.

ECM ENGINE CONTROL AND MONITORING

5191 Lafayette St.
Santa Clara, CA 95054
Phone: (408) 734-3433
FAX: (408) 734-3432
Email: sales@ecm-co.com
Web: www.ecm-co.com

EC DECLARATION OF CONFORMITY

We declare under our sole responsibility that the products:

AFM1540 Lambda Module
AFM1600L, AFM1600D Modules
DIS1000 Display head
EGR 4830 Analyzer
NOx 5210 NOx Analyzer
Lambda 5220 Lambda Analyzer
EGR 5230 EGR Analyzer
NOx/NH3 5240 NOx and NH3 Analyzer
LambdaCAN, LambdaCANc, LambdaCANd, LambdaCANp Lambda Modules
NOxCAN, NOxCANg, NOxCANt NOx Modules
NOx1000 NOx Module
baroCAN Module
dashCAN, dashCANc, dashCAN+, dashCAN2
appsCAN
SIM300, SIM400, SIM500, SIM600, SIM700, SIM800
BTU200

To which this declaration relates are in conformity with the essential requirements of the following standards:
EN61326: 1997/A2: 2001 (Class A & Annex A)
EN61010-1: 2001 (Electrical Safety)

And therefore conform to the requirements of the following directives:
89/336/EEC Electromagnetic Compatibility (EMC)
72/23/EEC Low Voltage Directive (LVD)

Ronald S. Patrick
Vice President Sales
January 27, 2014

ECM ENGINE CONTROL AND MONITORING
Los Altos, CA 94023-0040 • USA • (408) 734-3433 • Fax: (408) 734-3432 • www.ecm-co.com