ECM ENGINE CONTROL AND MONITORING

Configuring and Calibrating CO/CO2CAN Systems using the Configuration Tool (V4+) and Using the dashCAN* Display

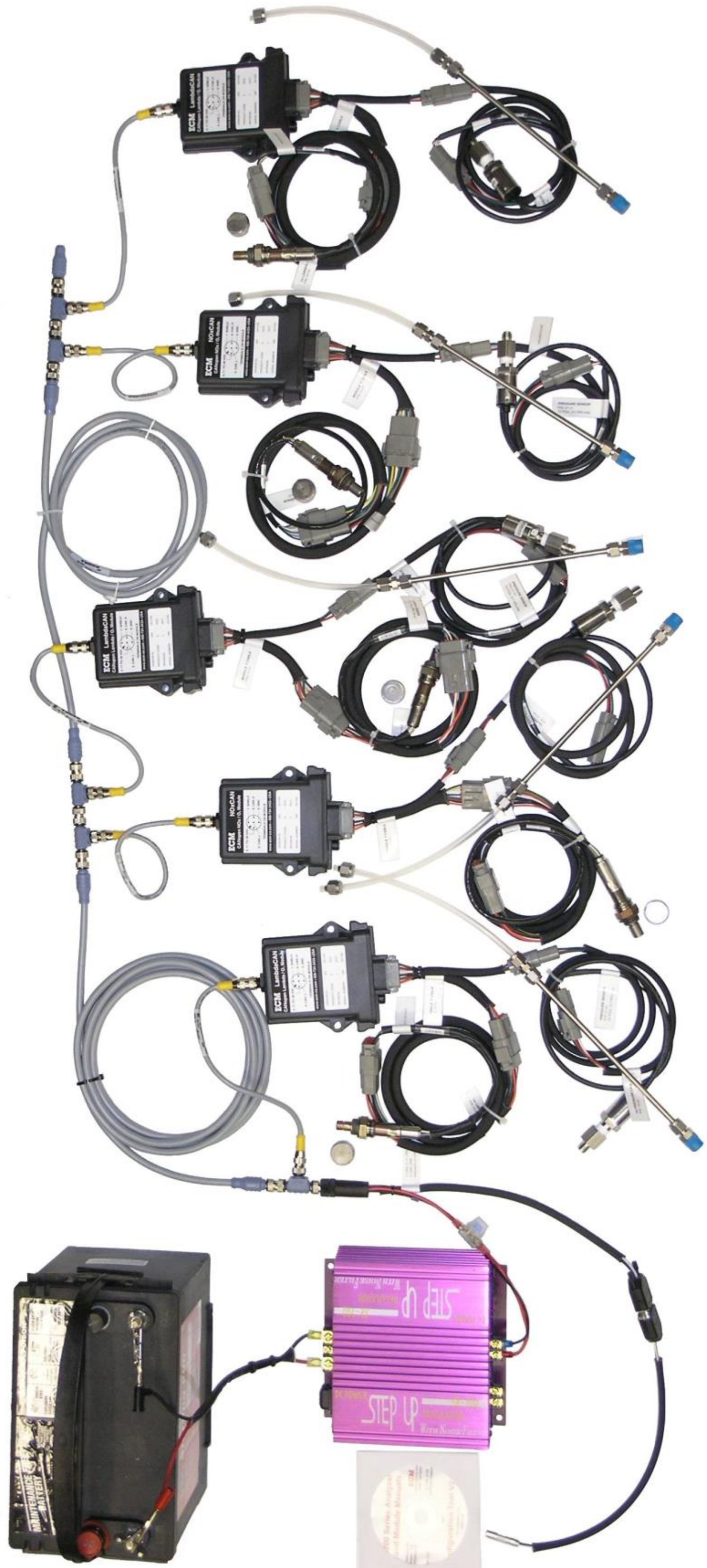
Important Note

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 CO/CO2CAN measurement module has to be set up to send the data required and the receiving device (i.e. data acquisition software) has to know what is being sent. The setting up modules and the production of a .dbc file used by the receiving device to understand the data sent is performed using the supplied Configuration Tool.

8-1-2018

Printed in the United States of America.

Table of Contents



Complex measurement systems can be easily built with LambdaCAN*, NOxCAN*, CO/CO2CAN, and baroCAN modules. Here is a five-channel lambda, O_2, and NOx pressure-compensated, in-vehicle system.

Introduction

The Configuration Tool runs on your PC and uses a CAN communication device to communicate with one or more ECM *CAN modules (i.e. CO/CO2CAN, LambdaCAN*, NOxCAN*, baroCAN) or analyzers (i.e. 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 CO/CO2CAN modules.

The Configuration Tool supports four CAN communication devices: Kvaser (recommended), 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. The Configuration Tool is delivered on a CD.

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 all 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 the conflicting 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 (see Appendix B).

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:

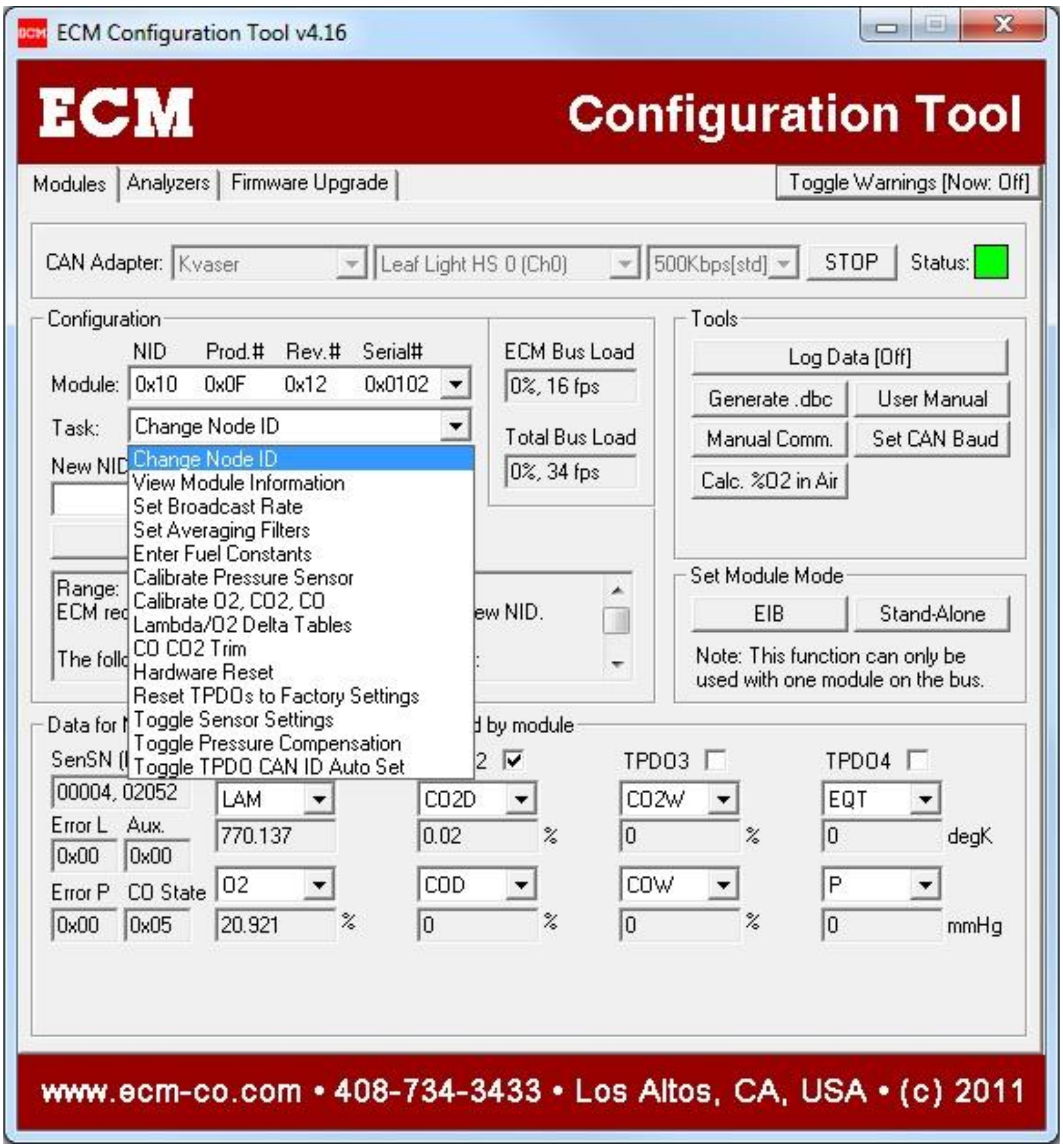
1. Configure a module. This includes calibrating a sensor attached to the module.
2. Look at data coming from that module in real-time and optionally log it.
3. Produce a .dbc file to be used by your data acquisition program.

Alternatively, 1. and 2. (above) can be performed by user-written CAN communication with the module. For information on how to do this and further detailed information about the CO/CO2CAN module, refer to the CO/CO2CAN 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 CO/CO2CAN modules are listed in Table 1.



Change Node ID: Allowable range 0x01 to 0x7F (hex). When you assign a Node ID (NID), the following CANs **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 everything but pressure (denoted as "Lambda Alpha") and one for pressure-related data (denoted as "Pressure Alpha"). 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 alphas are 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: This table contains the voltage versus pressure data for the pressure sensor (if connected). This data is stored in a memory chip in the pressure sensor's connector and is user-reprogrammable.

Calibrate O2, CO2, CO: User enters displayed %O2 (TPDO data) and the actual %O2 of the ambient air in which the COCO2 sensor is hanging. The %O2 in ambient air is determined using an ECM baroCAN device or the "Calc. %O2 in Air" feature in the Configuration Tool. The calibration data for the sensor is stored in a memory chip in the sensor's connector and is user-reprogrammable.

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

CO CO2 Trim: This task modifies the calculated CO and CO2. There are two levels of trim: basic and advanced. Trims are stored in the module, not the sensor and must be downloaded to every module that is to use them.

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 ceramic 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), CO, and CO2 data. Default: On.

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

Table 1: Task List for CO/CO2CAN Modules

Calibration of the COCO2 Sensor

Due to the method of operation of the COCO2 sensor, calibration is remarkably simple via the **Calibrate O2, CO2, CO** task and is performed using ambient air. Modifications of this calibration are made using the **Lambda/O2 Delta Tables** task (for Lambda, AFR, FAR, PHI, and %O2) and the **CO CO2 Trim** task (for CO and CO2).

All sensors are calibrated 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.

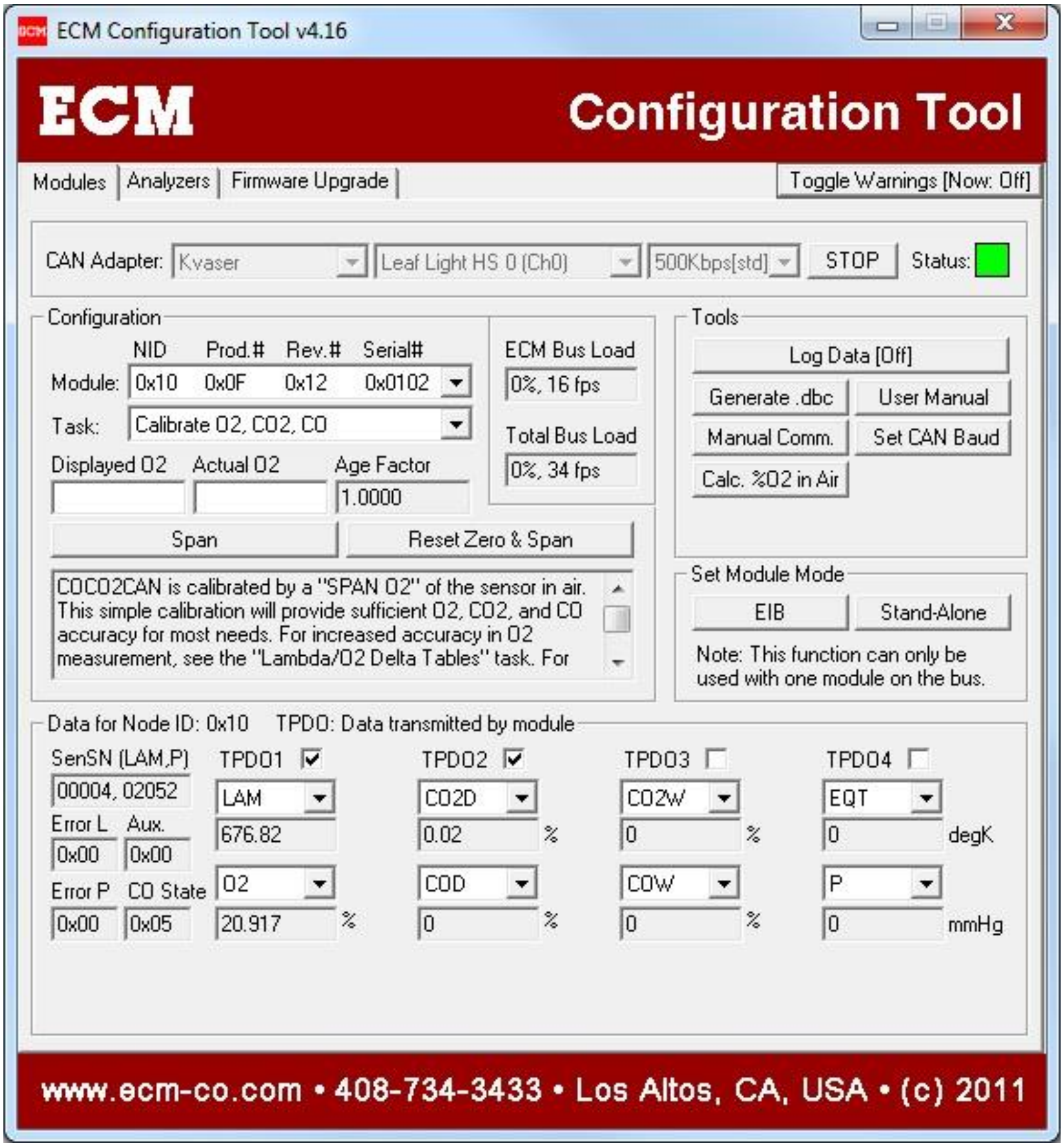
Calibration information (both factory calibration and user calibration) for the COCO2 sensor is stored in a memory chip in the sensor's connector. Therefore, the sensor does not have to be used with the module it was calibrated with. Once calibrated, the sensor can be sent to another site to be used. 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.

The **Lambda/O2 Delta Tables** and **CO CO2 Trim** are stored in the CO/CO2CAN control module (not in the sensor's memory chip). Therefore, they stay in the module in which they were programmed. If they are modified from the factory default, these modifications must also be made in all other CO/CO2CAN control modules.

Calibration of the COCO2 Sensor

The **Calibrate O2, CO2, CO** task is used to calibrate the COCO2 sensor. To perform the calibration:

1. Put the sensor and the pressure sensor (if equipped) in ambient, stationary air. The pressure during calibration is required if the sensor is to be pressure compensated.
2. The sensor should be left 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 or other reasons, take longer to stabilize.
3. Use ECM's baroCAN to measure the %O2 in air.
4. Alternatively to 3 (above), 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 "Calc. %O2 in Air" tool in the Configuration Software. 20.7 is a common number.
5. Select O2 as a TPDO parameter.
6. Select the **Calibrate O2, CO2, CO** task. Enter the displayed (as the TPDO) O2 and the actual %O2 (as determined in 3 or 4 above), then click on "Span".



The Lambda and O2 Delta Tables

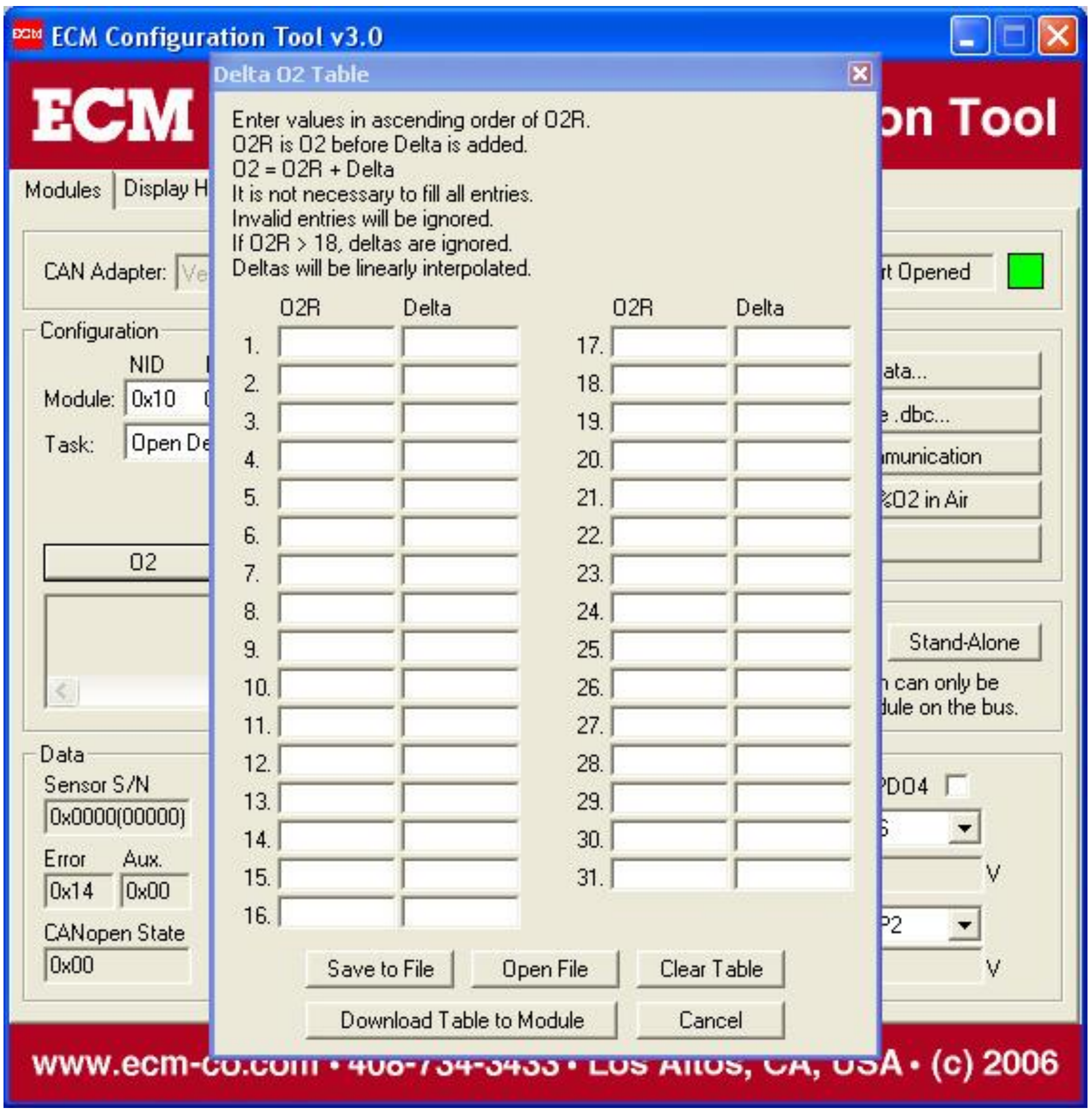
The %O2 calculated by CO/CO2CAN modules is denoted as "O2R" and is a wet (i.e. water taken into consideration in %O2 calculation), at chemical-equilibrium (i.e. not frozen-equilibrium) %O2 measurement (see Appendix C). The Lambda (AFR, FAR, PHI) calculated by the CO/CO2CAN module is denoted as "LAMR" 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 CO/CO2CAN 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 CO/CO2CAN module for it to be used. Unlike the sensor calibration information 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 CO/CO2CAN 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 CO/CO2CAN 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 CO/CO2CAN modules match default ETAS LA4 data (see Appendix C). Unlike the sensor calibration information (i.e. zero, span) which is stored in a memory chip in the sensor's connector, the Delta Lambda Table is stored in the module.

The CO CO2 Trim

The **CO CO2 Trim** task modifies the CO and CO2 numbers reported by the CO/CO2CAN module. This can be used to match the CO/CO2CAN output to another CO and CO2 measurement device (typically a gas analysis bench).

There are two levels of CO CO2 Trims: Basic and Advanced.

◆ CO and CO2 Trim (Basic)

In the Basic CO and CO2 trim, the user is allowed to enter gains and offsets to modify the reported CO and CO2 values and to change the bench gas temperature. The bench gas temperature is required for an accurate conversion from the wet CO and CO2 values determined by the CO/CO2CAN to the equivalent dry values (at the bench gas temperature).

The gains and offsets entered act on both the wet and dry CO (COw, COd) and CO2 (CO2w, CO2d) values according to the formulas:

$$\text{CO (corrected)} = m1 \times \text{CO} + b1$$

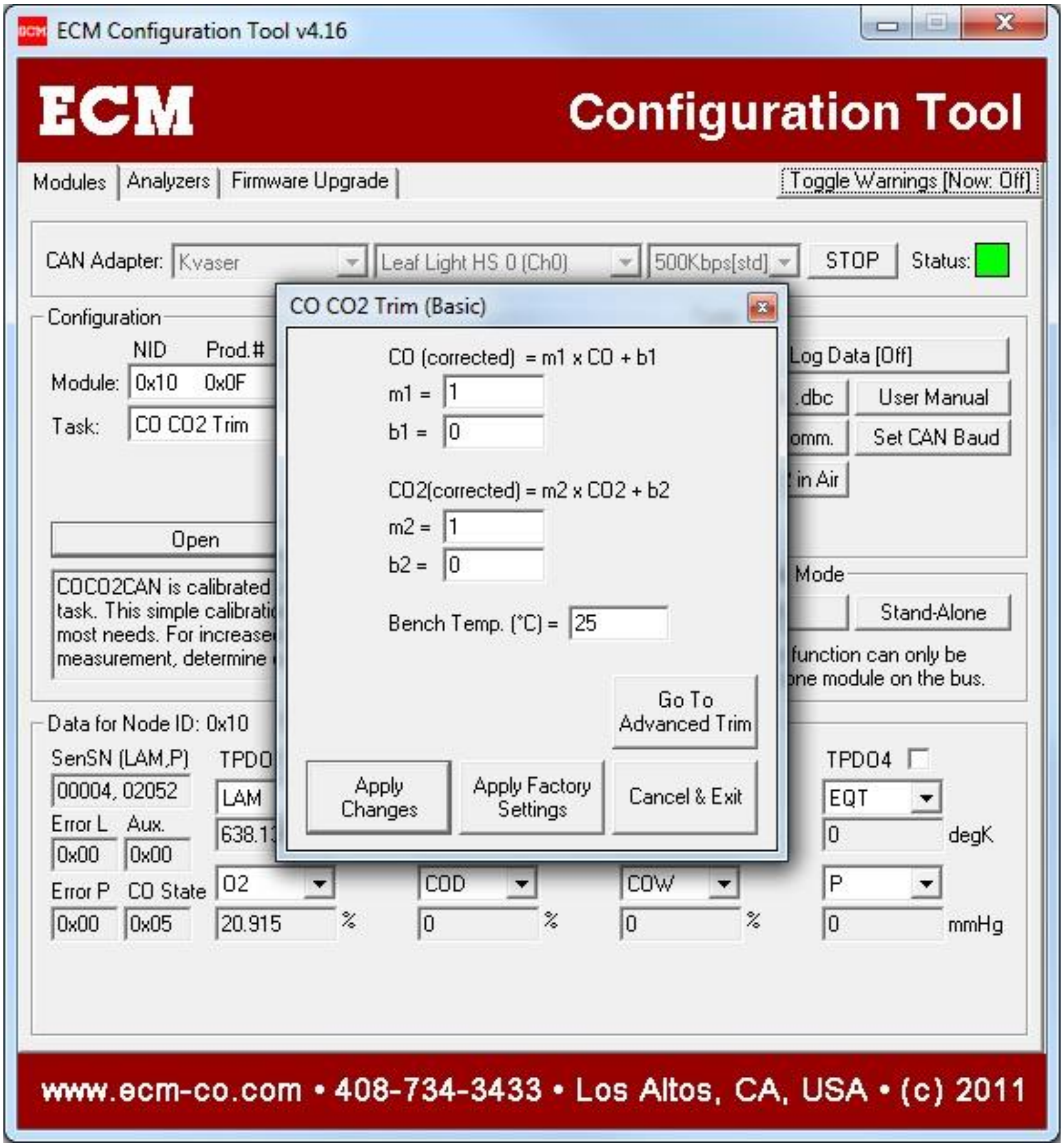
$$\text{CO2 (corrected)} = m2 \times \text{CO2} + b2$$

m1 and m2 are set at 1 and b1 and b2 are set at 0 as defaults.

The following procedure is recommended to determine the gains (m1, m2) and offsets (b1, b2) to match another CO and CO2 measurement device.

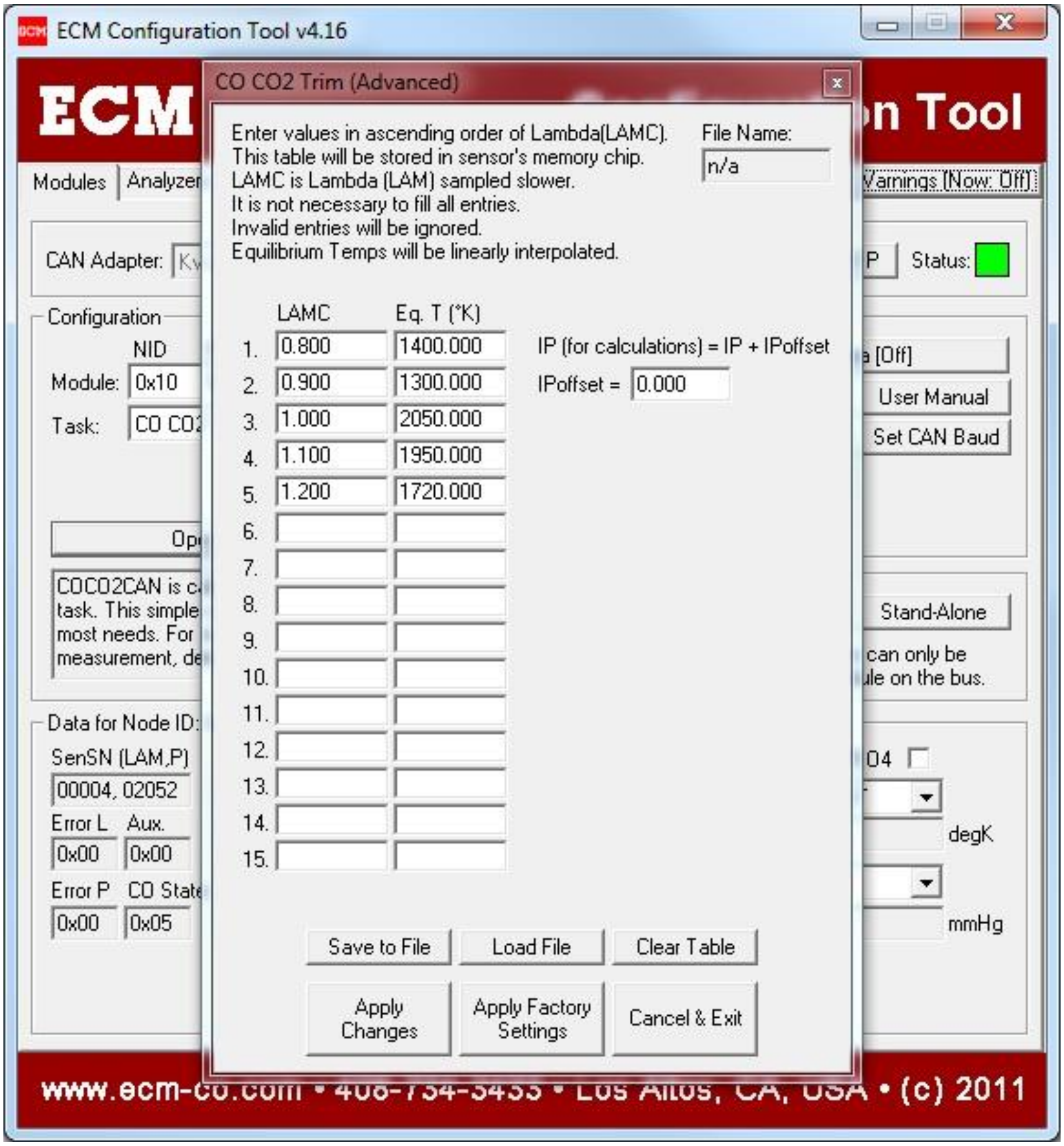
1. Perform the **Calibrate O2, CO2, CO** task
2. Enter the Bench Temperature
3. Set m1 =1, m2 = 1, b1 =0, b2 = 0 (Factory Calibration)
4. Click the "Apply Changes" button
5. Operate the engine and collect CO and/or CO2 values from CO/CO2CAN and the other measurement device. Note that gases from an engine, not calibration gases from tanks, must be used.
6. Use this data to perform a least-squares fit for m1, m2, b1, and b2.

7. Enter these values into the CO CO2 Trim (Basic) screen and click on the "Apply Changes" button.
8. Note that the O2 CO CO2 calibration results are stored in the sensor's connector but the bench temperature and the m1, m2, b1, and b2 values are stored in the CO/CO2CAN control module. Thus if that sensor is to be used in another module, the bench temperature and m1, m2, b1, and b2 values determined must be downloaded into the other module using the Configuration Tool.



◆ CO and CO2 Trim (Advanced)

The advanced CO and CO2 trim consists of modifying the equilibrium temperature table and the IPoffset value. To do this is a more complex procedure. Contact ECM for more information.



Recommended Setup

CO/CO2CAN was designed for engine-out (as opposed to catalyst-out) applications. However for some cases, the reduction in accuracy when the sensor is located after the catalyst may be acceptable. As-delivered, CO/CO2CAN is programmed for engine-out applications. For post-catalyst applications, various parameters should be changed for best results. Below are given the setups recommended for these two applications.

◆ **Engine-Out Applications**

Broadcast rate: 200ms
Averaging Filters: Alpha Lambda=0.375, Alpha Pressure=0.375
Fuel Constants: H:C=1.85, O:C=0.0, N:C=0.0
Delta Lambda Table: use file EngineOutLambdaDelta.txt
Delta O2 Table: none
CO CO2 Trim: Bench Temperature 5 °C, use file EngineOutCOCO2.txt

◆ **Catalyst-Out Applications**

Broadcast rate: 200ms
Averaging Filters: Alpha Lambda=1.0, Alpha Pressure=0.375
Fuel Constants: H:C=2.18, O:C=0.0, N:C=0.0
Delta Lambda Table: use file CatalystOutLambdaDelta.txt
Delta O2 Table: none
CO CO2 Trim: Bench Temperature 5 °C, use file CatalystOutCOCO2.txt

The files EngineOutLambdaDelta.txt, EngineOutCOCO2.txt, CatalystOutLambdaDelta.txt, CatalystOutCOCO2.txt are located on the supplied CD (or memory stick). The contents of the files are shown on the next page.

EngineOutLambdaDelta.txt

```
0.4500 0.0659
0.5000 0.0536
0.5500 0.0430
0.6000 0.0358
0.6500 0.0306
0.7000 0.0275
0.7500 0.0275
0.8000 0.0272
0.8500 0.0266
0.9000 0.0253
0.9500 0.0175
1.0000 0.0075
1.2500 0.0200
1.5000 0.0364
1.7500 0.0566
2.0000 0.0802
2.2500 0.1075
2.5000 0.1387
2.7500 0.1739
3.0000 0.2135
3.2500 0.2579
3.5000 0.3076
3.7500 0.3629
4.0000 0.4244
4.2500 0.4924
4.5000 0.5675
4.7500 0.6503
5.0000 0.7414
5.2500 0.8415
5.5000 0.9515
5.7500 1.0726
```

CatalystOutLambdaDelta.txt

```
0.4500 0.2129
0.5000 0.1867
0.5500 0.1623
0.6000 0.1412
0.6500 0.1221
0.7000 0.1052
0.7500 0.0913
0.8000 0.0771
0.8500 0.0627
0.9000 0.0475
0.9500 0.0258
1.0000 0.0020
1.2500 0.0205
1.5000 0.0445
1.7500 0.0723
2.0000 0.1036
2.2500 0.1385
2.5000 0.1774
2.7500 0.2202
3.0000 0.2674
3.2500 0.3195
3.5000 0.3769
3.7500 0.4398
4.0000 0.5090
4.2500 0.5846
4.5000 0.6673
4.7500 0.7578
5.0000 0.8565
5.2500 0.9643
5.5000 1.0819
5.7500 1.2107
```

EngineOutCOCO2.txt

```
0.000000
0.000000
0.800 1960.000
0.900 2005.000
1.000 2200.000
1.100 2005.000
1.200 1960.000
```

CatalystOutCOCO2.txt

```
0.000000
0.000000
0.800 1950.000
1.200 1950.000
```

Selecting What Data is to be Sent (TPDOs)

Data sent from CO/CO2CAN 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 CO/CO2CAN module is given in Table 2.

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

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.

LAMC is LAM at synchronized to the CO and CO2 measurements. It is used in the **CO CO2 Trim (Advanced)**.

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)	COCO2 sensor internal VS cell resistance x 1000
VHCM	VH Commanded (V)	Desired heater voltage commanded by the module
VS	VS (V)	COCO2 sensor internal VS cell voltage
VP1P	VP+ (V)	COCO2 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)	COCO2 sensor pumping current (unsigned integer format)
PR16	Praw16 (bits)	16 bit Pressure sensor output voltage (unsigned integer format)
URFL	Error bit flags (bits)	Module error flags (unsigned long format)
URCd	ECM CANOpen Error Code	ECM CANOpen Error Code
PR10	Praw10 (bits)	10 bit Pressure sensor output voltage (unsigned integer format)
PCF	Pressure Correction Factor	COCO2 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	LAMBDA real	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 COCO2 sensor pumping current
PVLT	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	Error bit flags (bits)	Pressure sensor error flags (unsigned long format)
PERC	CANopen error code	Pressure sensor error code
CO2w	**%CO2 wet**	**%CO2 (wet) after effect of CO CO2 Trim**
COw	**%CO wet**	**%CO (wet) after effect of CO CO2 Trim**
CO2d	**%CO2 dry**	**%CO2 (dry) after using Bench Temperature and CO CO2 Trim**
COd	**%CO dry**	**%CO (dry) after using Bench Temperature and CO CO2 Trim**
LAMC	Lambda	LAMBDA synchronized to the CO and CO2 measurements
EQT	T (°K)	Equilibrium Temperature (°K)

Table 2: CO/CO2CAN Parameter List
(Highlighted parameters are the ones most commonly used.)

Producing a .dbc File

A .dbc file describes to the device receiving data from one or more ECM *CAN 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 ECM *CAN 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 CO/CO2CAN parameters.
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]

PRESSURE ERROR CODE	**LAMBDA ERROR CODE**	**LED ACTION (Lambda Only)**	**DESCRIPTION OF ERRORS**
0x0000	0x0000	Grn ON	All OK, (green led constantly on)
n/a	0x0001	Flash Grn 10Hz	Lambda sensor warm-up period
n/a	0x0002	Grn/Both/Red 2s	Power on reset/ Init hardware
n/a	0x0011	Pulse Red 1x/2s	16b ADC failed to init
n/a	0x0012	Pulse Red 1x/2s	+Vsw shorted
0x0013	0x0013	Red ON	Sensor turned off (red led constantly on)
0x0014	0x0014	Pulse Red 1x/2s	Sensor not present/ HTR open
n/a	0x0015	Pulse Red 1x/2s	HTR shorted
0x0021	0x0021	Pulse Red 2x/2s	1wire bus shorted
0x0022	0x0022	Pulse Red 2x/2s	No 1wire present
0x0023	0x0023	Pulse Red 2x/2s	CRC16 error
0x0024	0x0024	Pulse Red 2x/2s	Invalid 1wire parameter (sensor type)
0x0025	0x0025	Pulse Red 2x/2s	1-wire data format not compatible (old rev)
n/a	0x0031	Pulse Red 3x/2s	+Vsw < 6 for > 7sec
n/a	0x0032	Pulse Red 3x/2s	+Vsw > 32V
n/a	0x0041	Pulse Red 4x/2s	VS too high
n/a	0x0051	Pulse Red 5x/2s	RVS too high
n/a	0x0052	Pulse Red 5x/2s	(VHcommanded - VHactual) > 0.5V for > 10sec
n/a	0x0061	Pulse Red 6x/2s	VP+ > 6V
n/a	0x0062	Pulse Red 6x/2s	VP+ < 2V
n/a	0x0063	Pulse Red 6x/2s	
n/a	0x0064	Pulse Red 6x/2s	0.25V > VS+ > 0.75V
n/a	0x0065	Pulse Red 6x/2s	User data (span) in 1wire corrupted (user must set new span)
n/a	0x00A1	n/a	Invalid software state
n/a	0x00B1	n/a	CAN overrun
n/a	0x00B2	n/a	CAN passive mode
n/a	0x00B3	n/a	CAN heartbeat error
n/a	0x00B4	n/a	CAN recover bus off
n/a	0x00B5	n/a	CAN Tx CanId collision
n/a	0x00B6	n/a	Serial overrun
n/a	0x00B7	n/a	Can overrun Lss
n/a	0x00B8	n/a	Can overrun Sdo
n/a	0x00B9	n/a	Can overrun Rx
n/a	0x00BA	n/a	Can overrun ECT5
n/a	0x00FF	Both ON	Module powering down within 500ms
n/a	**ECM AUX**	n/a	Lambda sensor warm-up count down in seconds (active during ECM Error Code 0x0001)

Table 3: CO/CO2CAN Error Codes List (in Hexidecimal)

Using the dashCAN* Display

The dashCAN display (see cover and below) is a small (105 mm x 63 mm x 63 mm), two-channel remote display for CAN networks containing ECM *CAN modules. dashCAN+ is a slightly larger display (105 mm x 63 mm x 165 mm) with the addition of six programmable analog outputs. dashCAN* (i.e. dashCAN or dashCAN+) comes with a two meter cable and a “T” (P/N 09-05). Simply attach dashCAN* to the CAN bus and any two parameters being transmitted from ECM *CAN modules can be displayed and any six parameters can be converted to analog outs (dashCAN+ only). dashCAN* can display parameters from the same module or two different modules. 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.
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 upper display or “LOCK” will be on the lower display. If “MOd” is displayed, the ↑ and ↓ keys will roll through the setup options (see Table 4). First the options for the upper channel are shown on the upper display, followed by identical options for the lower channel on the lower 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 F describes how to LOCK and unlock dashCAN*.

Setup Option	Level 1	Function
MOd		Select module s/n. Default is NONE.
RATE		Set parameter averaging rate. Range 0.001 to 1.000 Default is 1.000
AOUT (dashCAN+ only)		
	A1 (upper channel)	Program analog output 1
	A2 (upper channel)	Program analog output 2
	A3 (upper channel)	Program analog output 3
	A4 (upper channel)	Program analog output 4
	A5 (upper channel)	Program analog output 5
	A6 (upper channel)	Program analog output 6
dISP		Select parameter. Note: Parameters available are those contained in TPDOs programmed using Configuration Software.
CONF	LEdS	Set display intensity. Default is 3333.
	LOCK	Lock and Unlock Display for Programming

MOd, RATE, AOUT, and dISP appear on the upper display for the upper channel and on the lower display for the lower channel. CONF just appears on the lower display and is for global dashCAN* setup. All entries must be followed by pressing the ENT key.

Table 5: Menu Tree for dashCAN*

MOd (Module) Setup Option

In MOd setup, the serial number of the module assigned to the upper or lower channel is entered. The serial number is written on a label on the module. The module assigned to the upper channel will send information to the upper display and the module assigned to the lower channel will send information to the lower display. The same module can be assigned to both channels or different modules can be assigned to each channel.

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 Software. This transmitted data can then be further averaged before being displayed on the displays. Separate averaging can be programmed for the upper display and the lower display.

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:

$$\textbf{DisplayedValue}_t = \alpha \textbf{ x Parameter}_t + (1 - \alpha) \textbf{ x DisplayedValue}_{t-1}$$

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

The selected display averaging does not affect the module's CAN transmission rate or averaging.

AOUT Setup Option (dashCAN+ Only)

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.

Parameter information from the module assigned to the upper channel can be sent to analog outputs 1, 2, and 3. Parameter information from the module assigned to the lower channel can be sent to analog outputs 4, 5, and 6.

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

1. Press the SYS key so that "MOd" is displayed.
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 will be shown on the bottom display.

dISP (Display) Setup Option

In dISP setup, the parameters to be displayed are selected. Only parameters selected in active TBDOs by the Configuration Software can be displayed.

Here is an example of setting the parameter to be displayed on the upper 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 ↓ 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 lower display. To enter CONF, press the SYS key so that "MOd" appears on the upper display, press the ↓ key until "CONF" appears on the bottom display, and then press the ENT key. CONF is for global dashCAN setup.

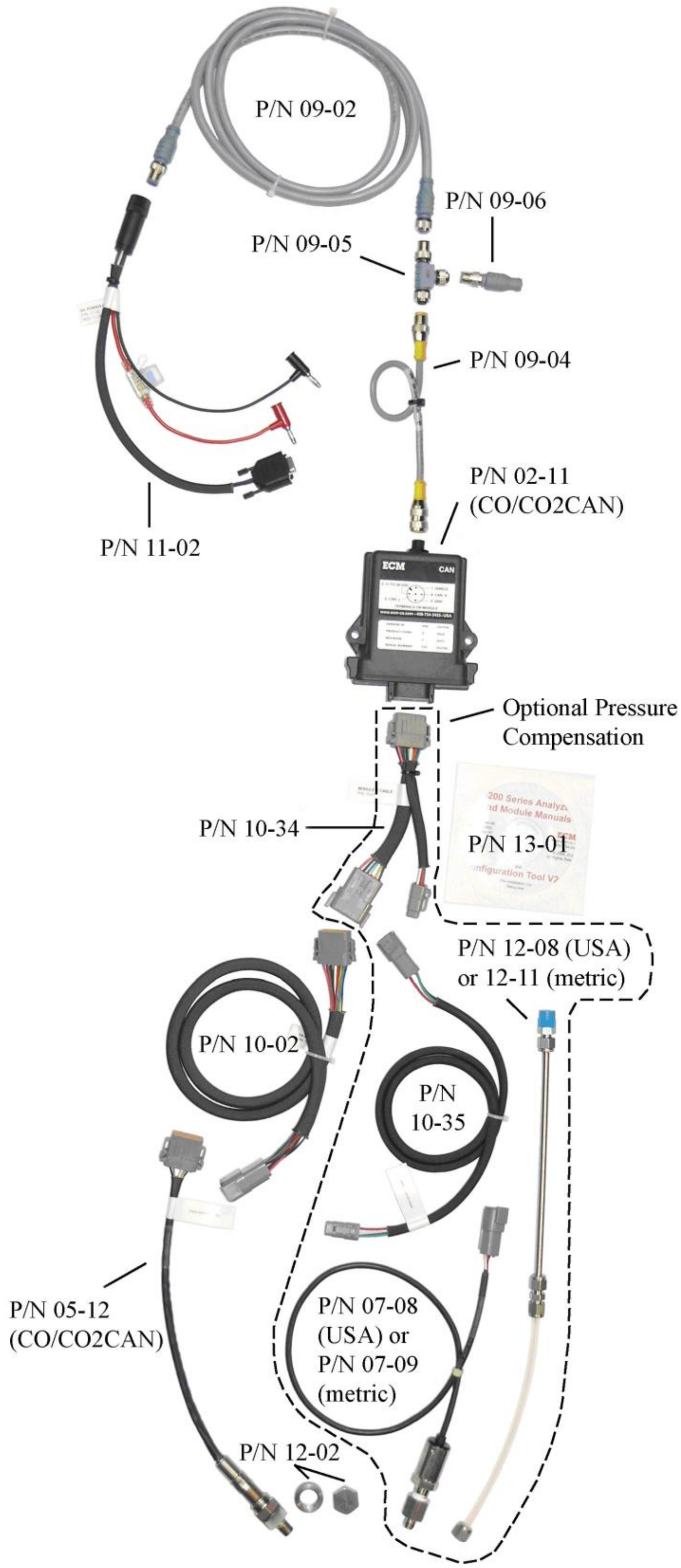
◆ LEdS

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

◆ LOCK

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

Appendix A: CO/CO2CAN Kit Contents



The CO/CO2CAN Kit consists of:

	Description	P/N	Quantity
1.	CO/CO2CAN Control Module	02-011	1
2.	COCO2 Sensor	05-12	1
3.	Lambda 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 (add /P to kit part number):

1.	Pressure Sensor, 0-75 psia, 517 kPa	07-08 (USA) or 07-09 (metric)	1
2.	Pressure Extension Cable	10-35 (1m)	1
3.	Pressure Sensor Tubing	12-08 (USA) or 12-11 (metric)	1
4.	Module Y Cable	10-34	1

Optional Cables:

1.	Lambda Cable	10-03 (2m)	1
2.	Pressure Cable	10-36 (2m)	1
3.	DC Power Cable, DB9F, Spades	11-01	1

Optional Power Supplies:

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

Appendix B: Module Stand-alone Mode and EIB Mode

CAN data from ECM *CAN 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 an EGR 5210, 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 *CAN modules are sold alone, they are delivered in Stand-alone mode. When *CAN modules are sold as part of a NOx 5210, Lambda 5220, or EGR 5230 analyzer, they are delivered in EIB mode.

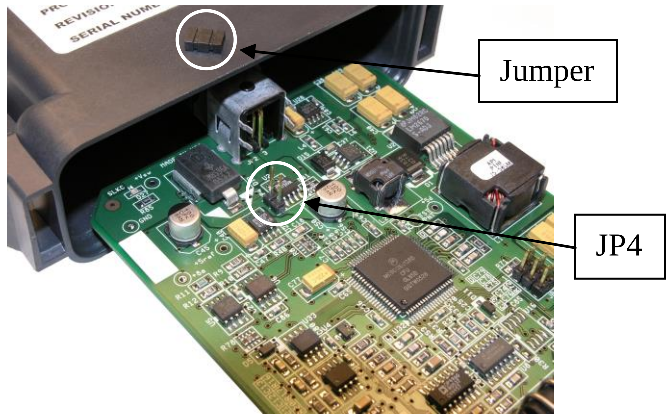
To convert from one mode to the other requires software reprogramming of the module followed by the installation (to set to Stand-alone) or removal (to set to EIB) of a jumper inside the module.

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

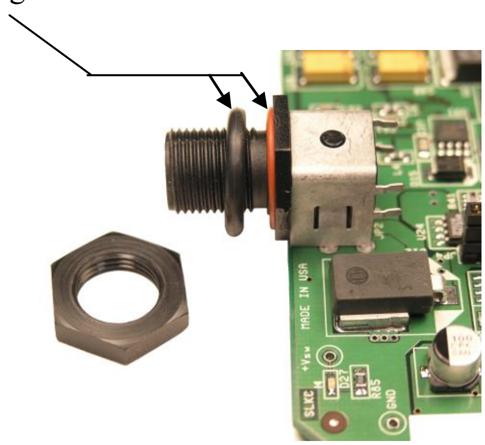
1. Take the nut off the end of the module. Use an 18mm socket without the wrench.

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.

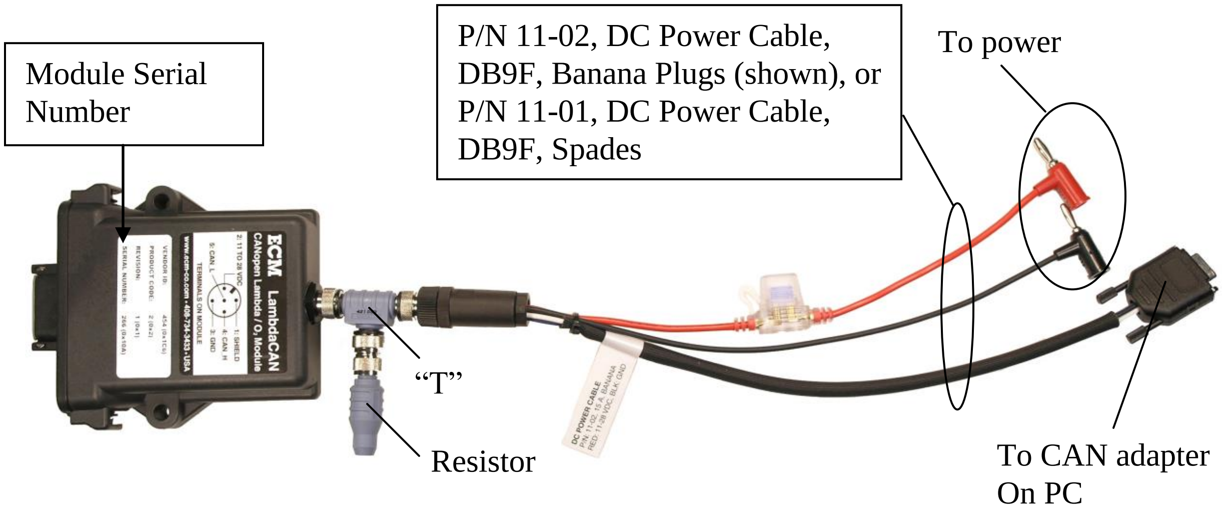
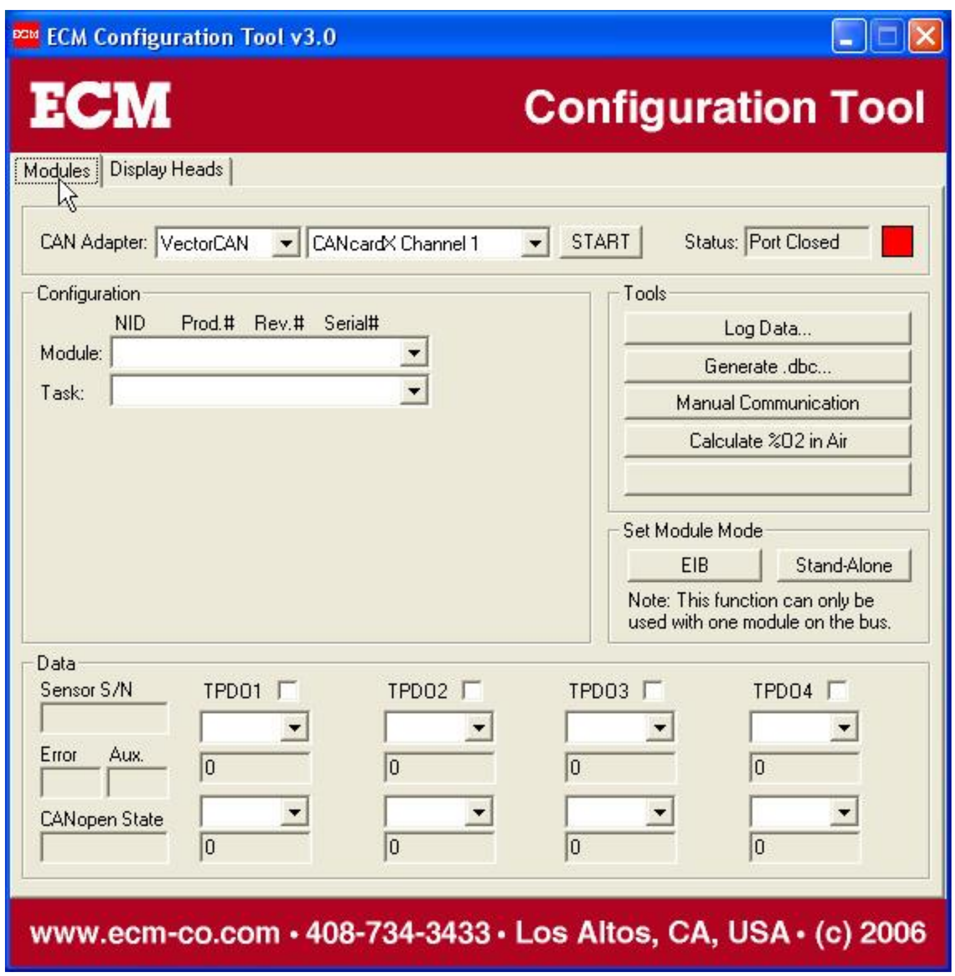


Figure A1: Module prepared for Reprogramming

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: Interpreting O2 and Lambda (AFR, FAR, PHI) Data from CO/CO2CAN Modules

◆ Comparing to Spindt and Brettschneider Calculations

CO/CO2CAN 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 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 COCO2 sensors operate 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. CO/CO2CAN 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 CO/CO2CAN 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 CO/CO2CAN 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 COCO2 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 COCO2 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 COCO2 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 COCO2 sensor-measured O2. This difference can be corrected for by a Delta O2 Table.

◆ **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. All that is required is the addition of a calibrated pressure sensor to the module activate the pressure compensation. 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, CO/CO2CAN'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 CO/CO2CAN to an ETAS meter and you want the CO/CO2CAN to match the ETAS meter, then you must span the CO/CO2CAN to whatever the ETAS meter says the %O2 is in air – even if it is wrong.

Therefore, if you want a CO/CO2CAN 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 CO/CO2CAN 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 CO/CO2CAN to the %O2 given on the ETAS meter even if it is wrong.

This procedure is to be used only if you want the CO/CO2CAN to match a specific ETAS meter and sensor combination. Normally, the CO/CO2CAN are spanned to the correct %O2 in air.

Appendix D: Setting Up ETAS INCA for ECM Modules

Hardware Setup: Using ETAS ES591.1

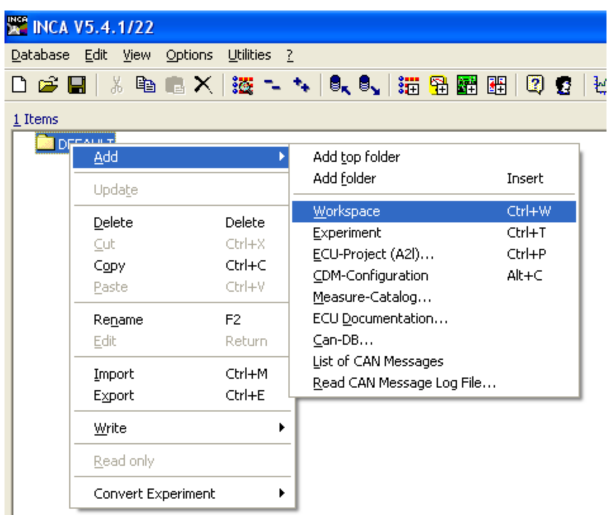
1. Connect the power port to a power source between 6V and 32V.
2. Connect the Ethernet port directly to the Ethernet port on your PC. This port does not use an internet/intranet connection like a router.
3. Connect either the CAN1 or CAN2 port to a CAN network (i.e. ECM modules or display heads).

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

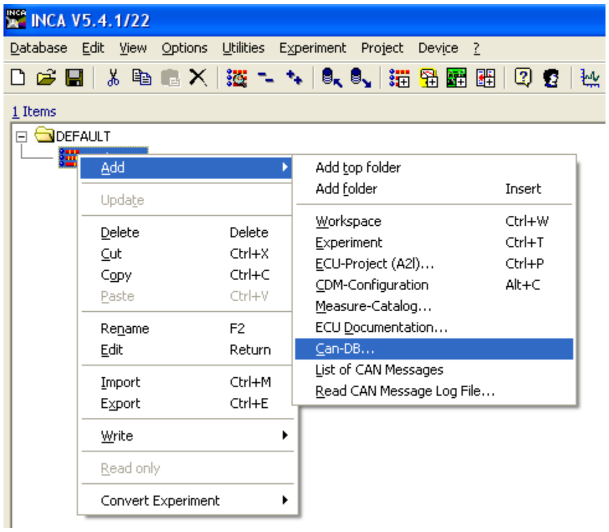
1. Double click the INCA V5.4 icon to open the software.

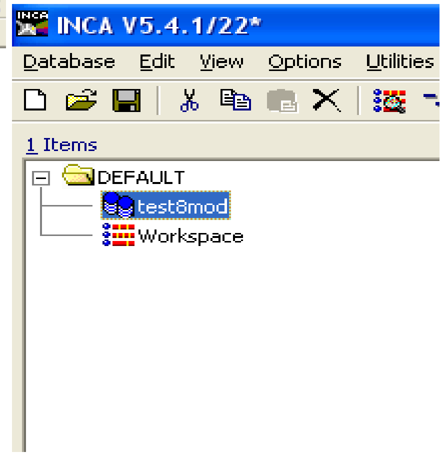
2. **Create a new Database**. In the Database menu, select New. Give your database a name (i.e. a folder name). In INCA, a Database means the current working directory. Each project is created in a unique directory. When INCA is opened, it will default to the last Database that was used.

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

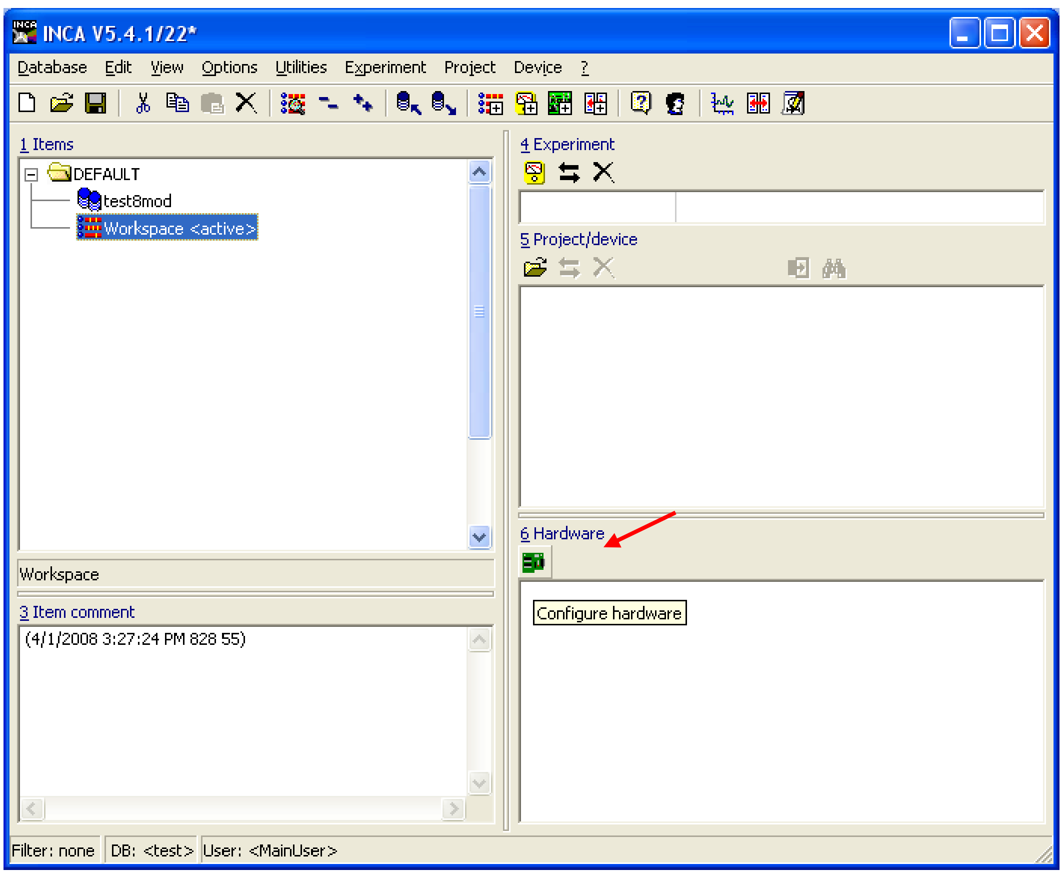


4. **Add a new dbc file for your project.** Right click on the workspace you created in step 3, select Add > Can-DB. Browse to your dbc file and click open. In this example, we are using a file named test8mod.dbc. An INCA log window will pop up. You can ignore this.

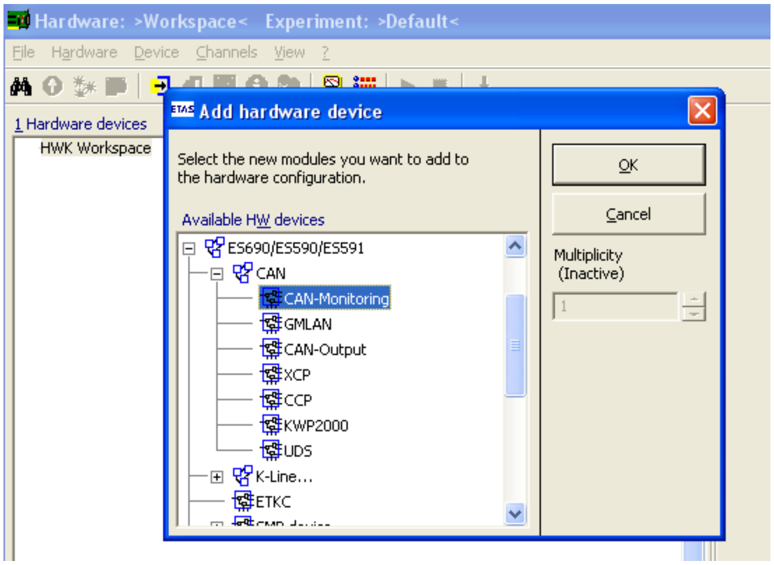




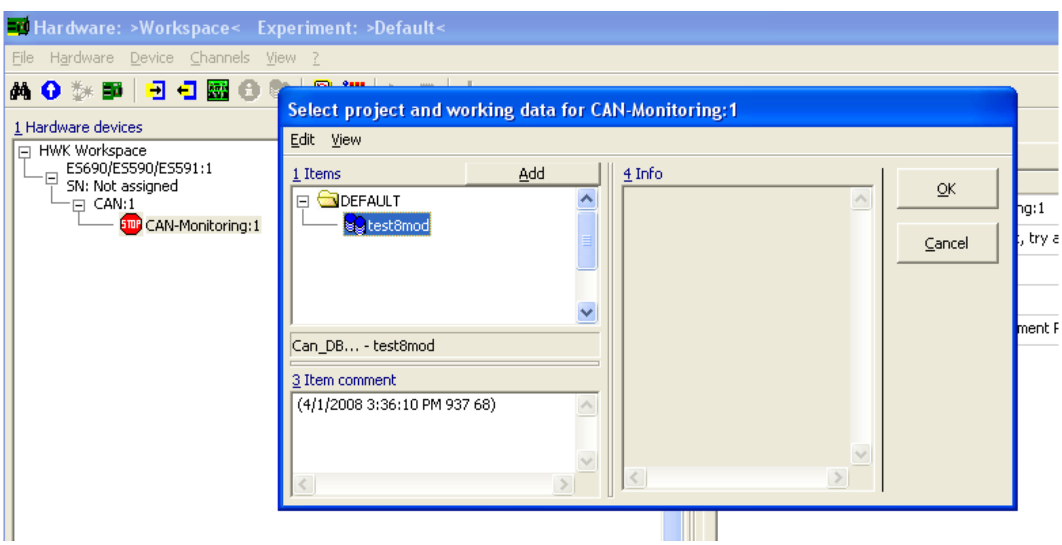
5. **Configure the hardware.** Click on the icon for the workspace you created in step 3. Open the Hardware Configuration icon under the section text "6. Hardware". A hardware configuration window will open.



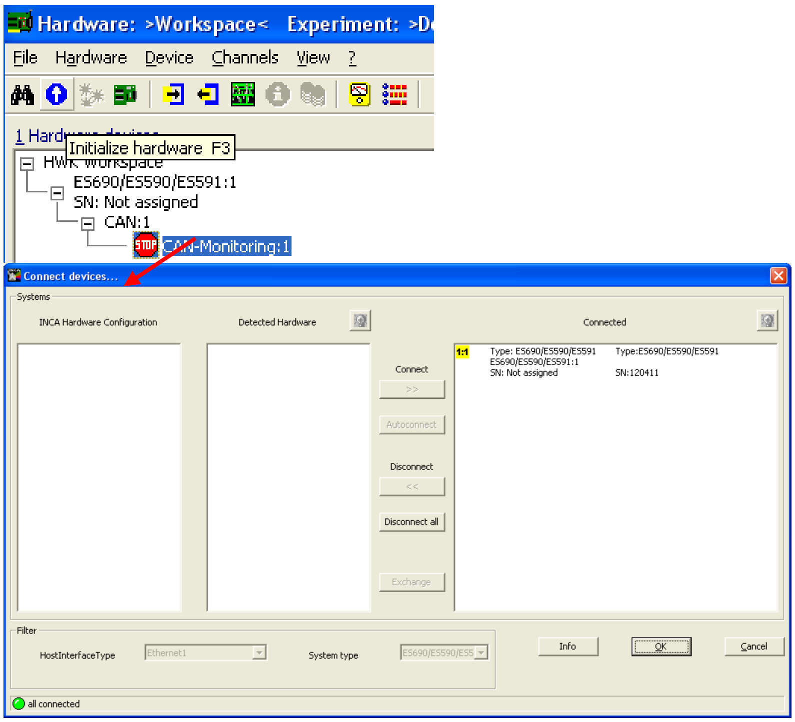
6. **Select the hardware.** In the hardware configuration window, right click the "HWK Workspace" listed under the section text "1. Hardware Devices", and select Insert. Select the ETAS device you wish to use. In this example, we are using an ETAS ES591.1. Expand the selection tree by clicking the "+"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



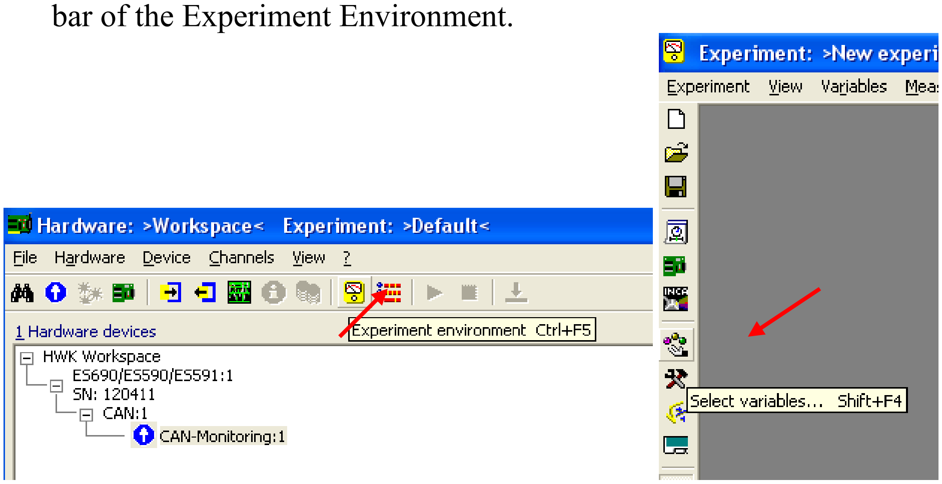
7. **Associate the dbc.** When you clicked OK in the last step, another window will pop up that will allow you to select a dbc that you have added to your workspace from step 4. Expand the selection tree, select your dbc file, and click OK.



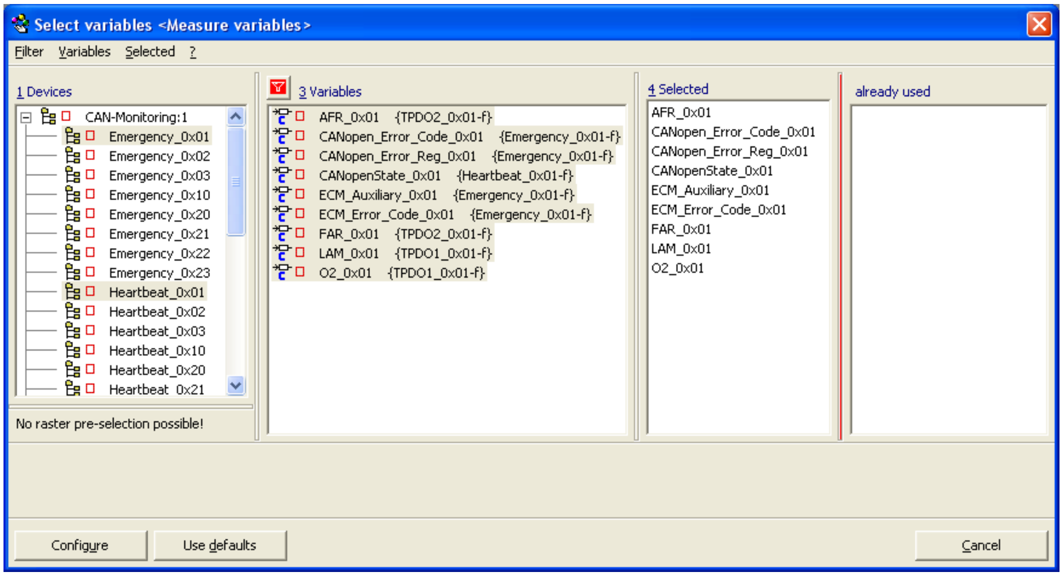
8. **Initialize hardware.** The hardware is currently stopped, as indicated by the red stop sign icon next to the selected hardware. You must initialize it before you can use it to collect data. Click on the Initialize Hardware button on the upper tool bar and wait for the hardware to complete its initialization.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



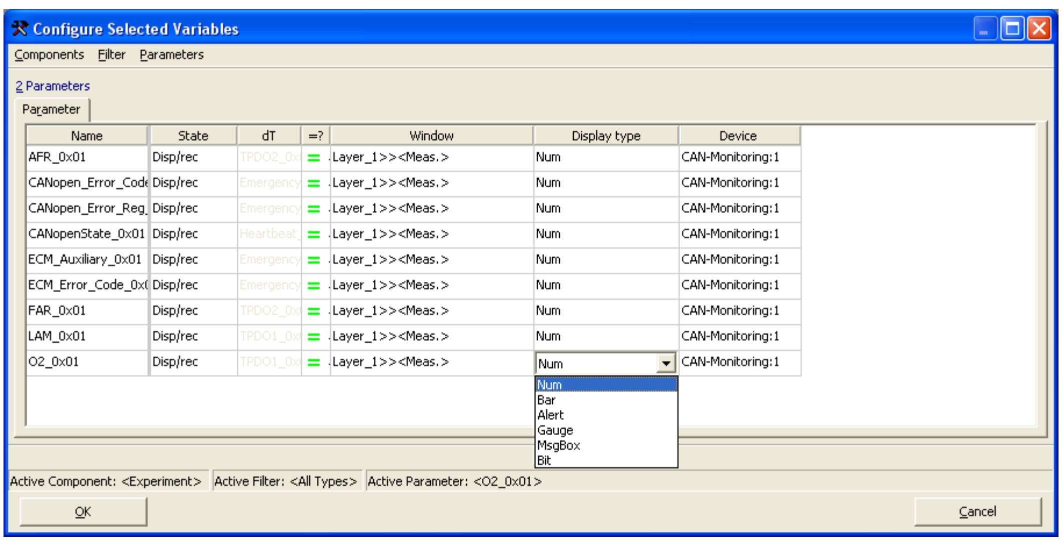
9. **Open an Experiment Environment.** Click on the Experiment Environment button on the upper tool bar to open an Experiment Environment. The Experiment Environment is where you can setup the monitoring of the CAN bus. By default, the Experiment Environment will be blank. You must select the variables 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.



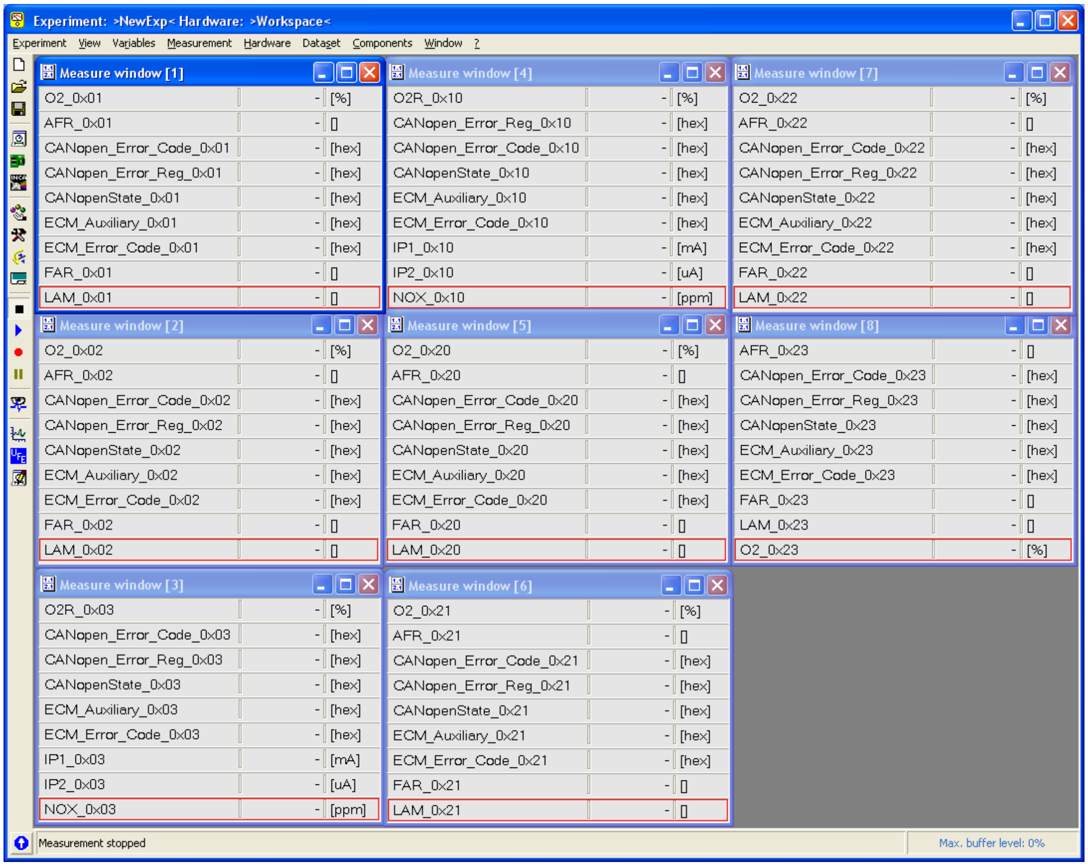
10. **Select and Configure Variables.** Select the variables that you wish to monitor in the Experiment Environment. These variables names are based on the data found in the dbc file. Click Configure.



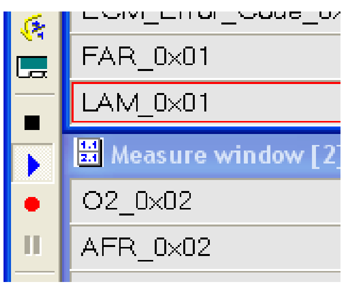
11. Another window will pop up to configure each selected variable. You can configure, for each variable, whether to record or simply display the data, how the data will be displayed (graphs, charts, gauges, numeric, etc.). When complete, click OK. We have left all configurations at default for this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

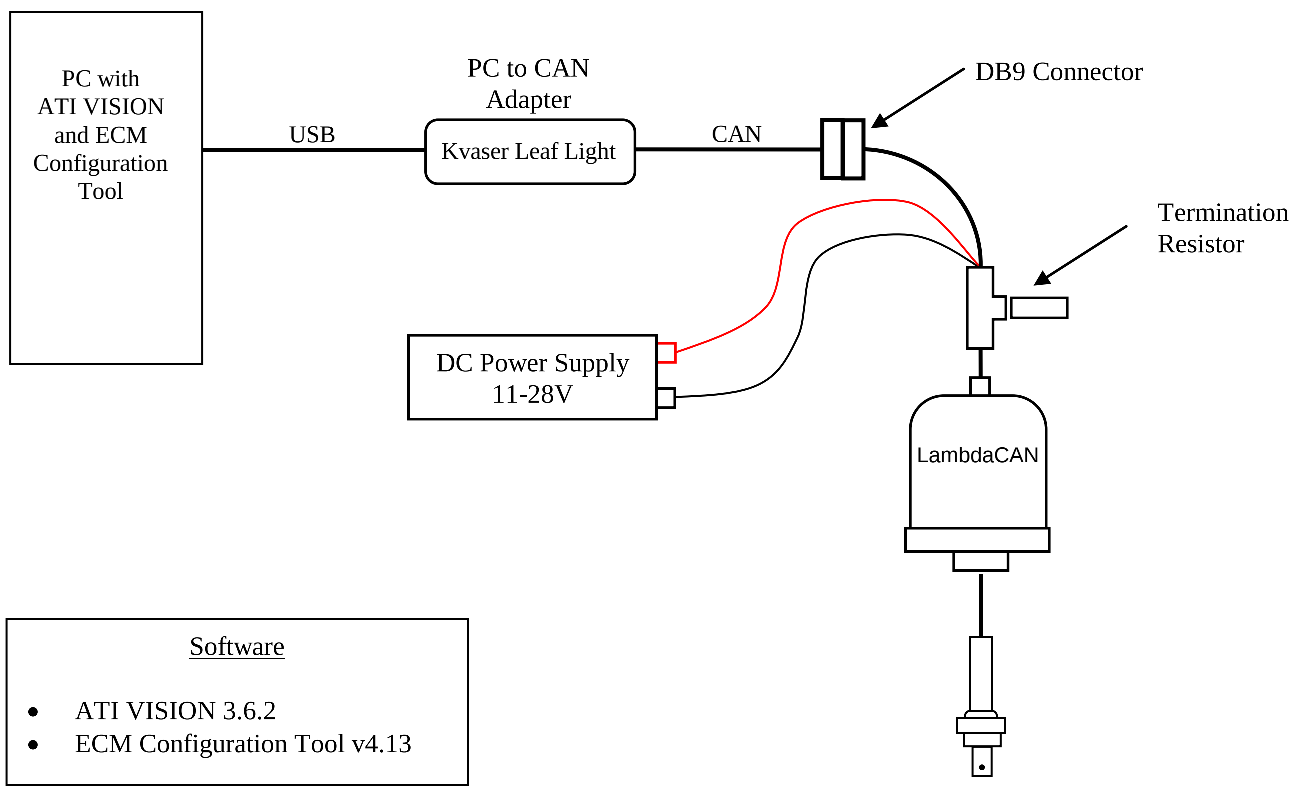


Figure 1: Equipment Schematic Layout

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.

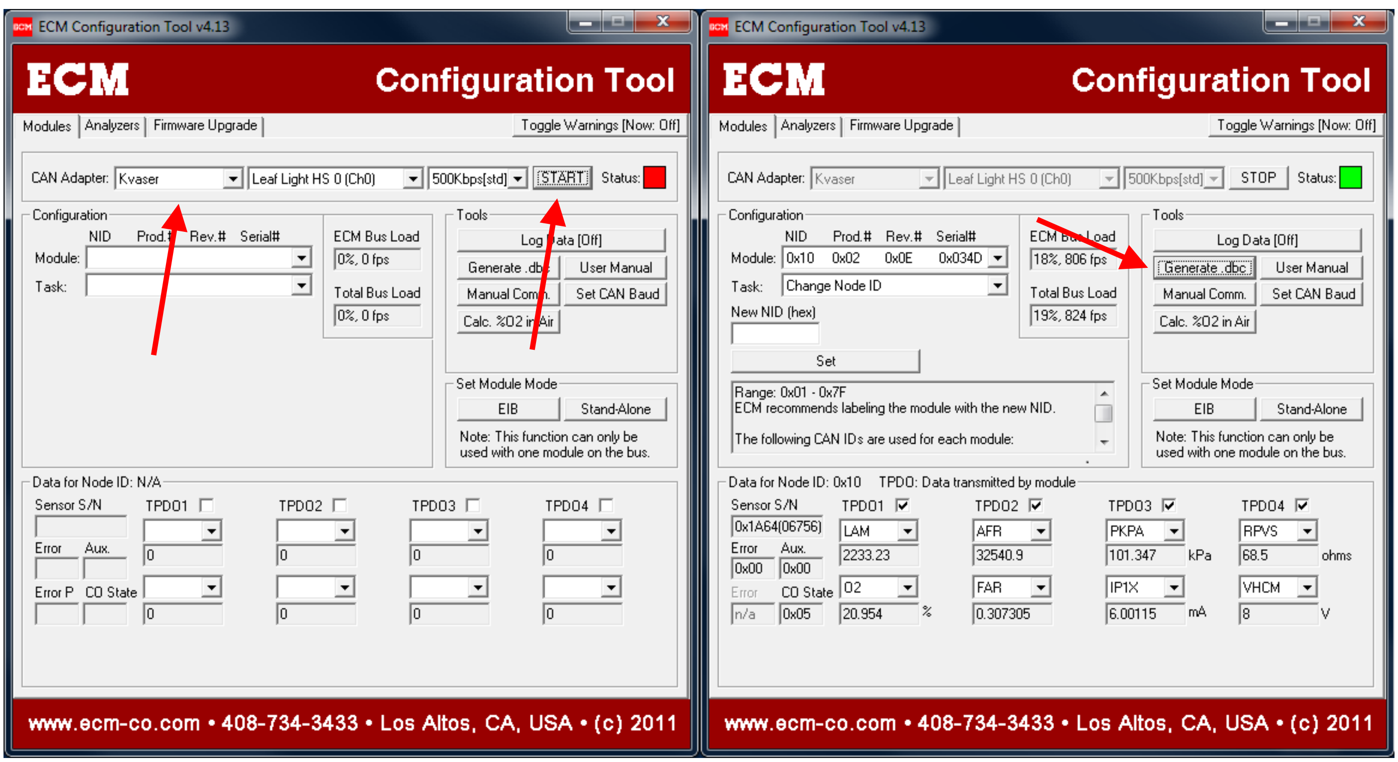
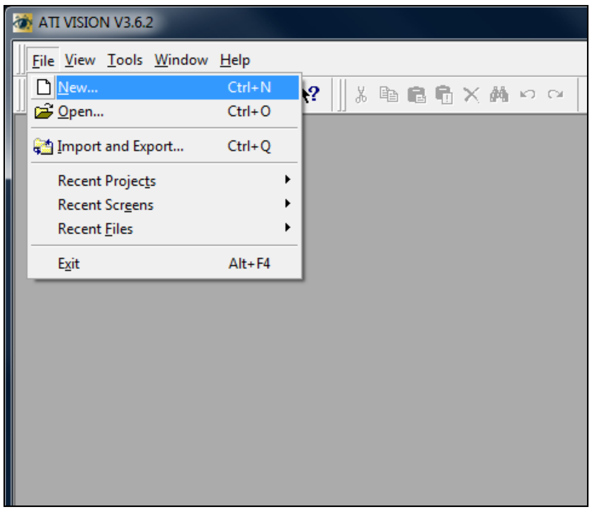


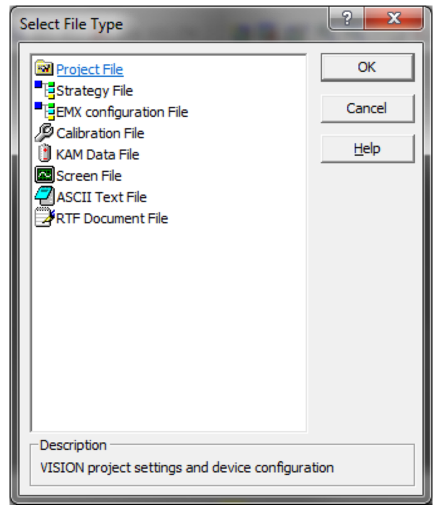
Figure 2: ECM Configuration Tool

Setup CANMonitor using ATI VISION

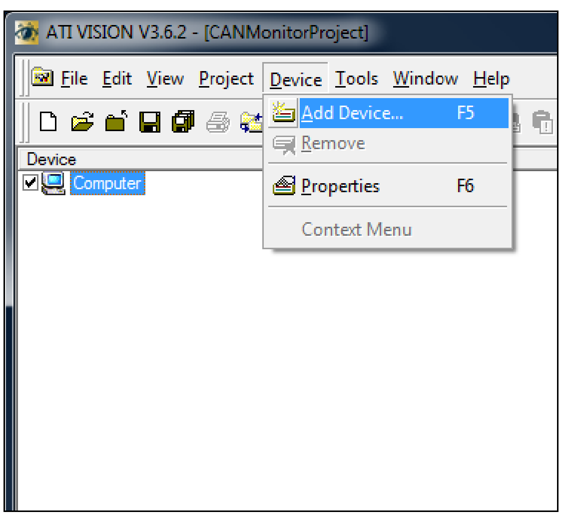
ATI VISION CANMonitor provides a method of reading general purpose information from any available CAN channel. The .dbc file generated by the ECM Configuration Tool is used to describe the format of the information available to VISION. To setup a CANMonitor in ATI VISION:

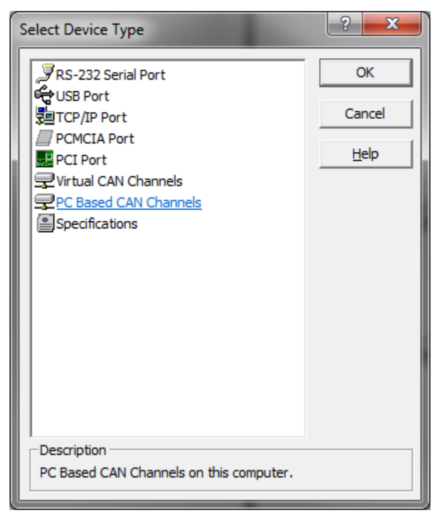
1. Run ATI VISION and open an existing Project File or create a new one by clicking File → New → Project File. In this example the Project has been named CANMonitorProject.



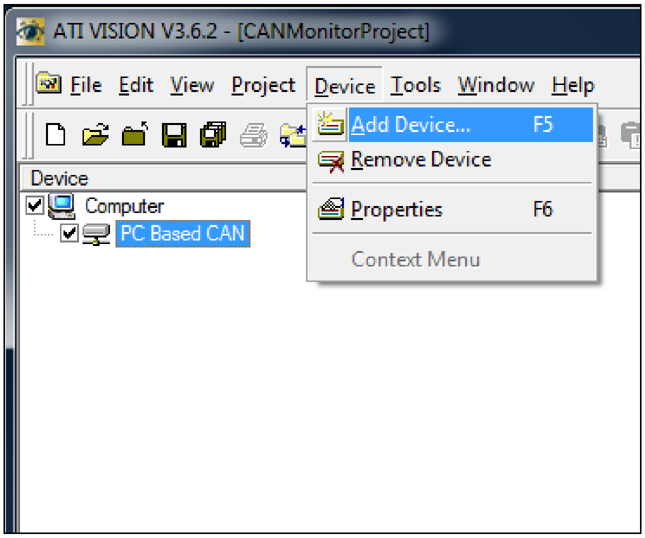


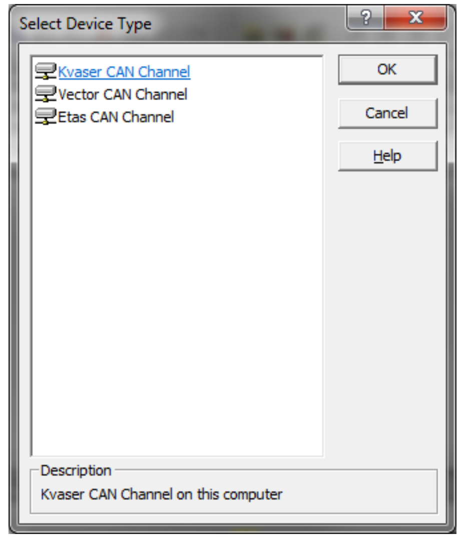
2. Add a Device by clicking Device → Add Device, select PC Based CAN Channels from the list.



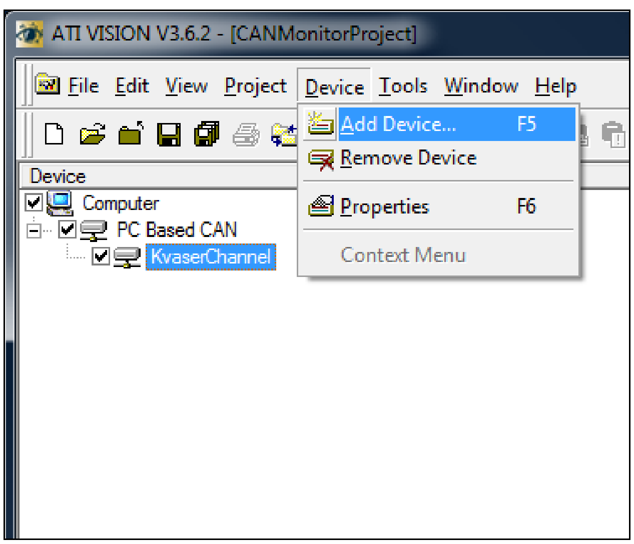


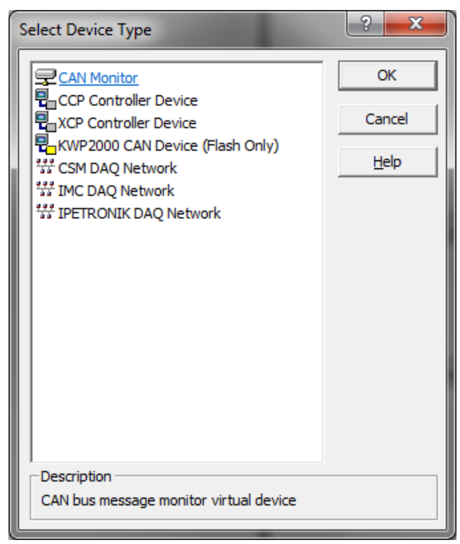
3. Add a physical hardware device by clicking Device → Add Device, and select Kvaser CAN Channel.



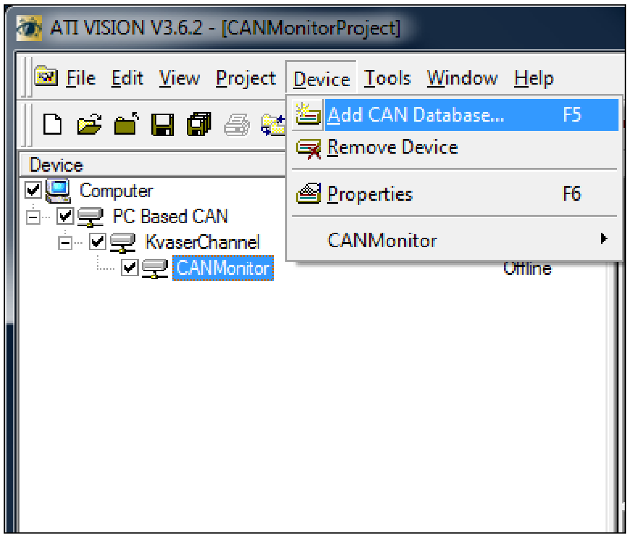


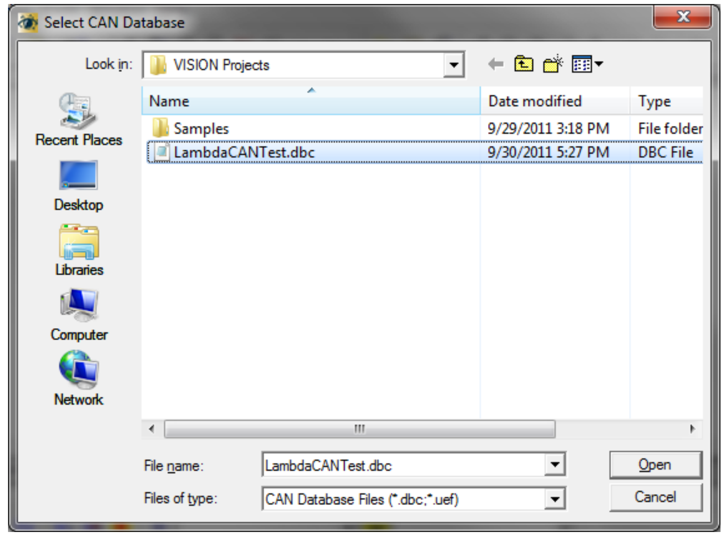
4. Select a CANMonitor device by again clicking Device → Add Device, and select CANMonitor.



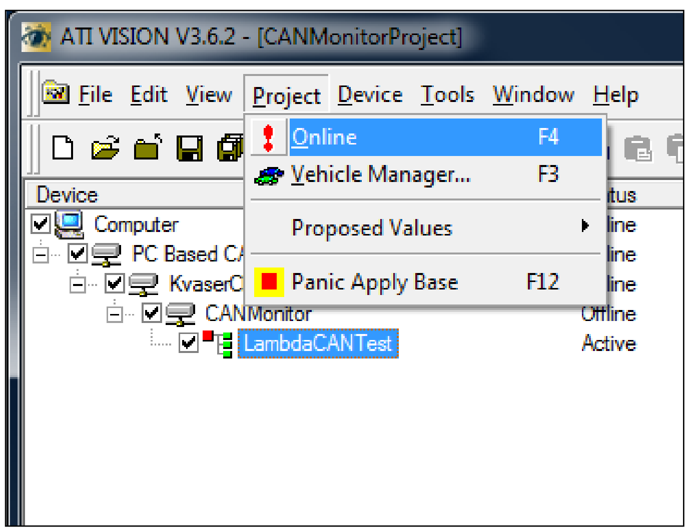


5. Add the .dbc file generated from the ECM Configuration Tool to CANMonitor by clicking Device → Add CAN Database and browsing to the previously created .dbc file.

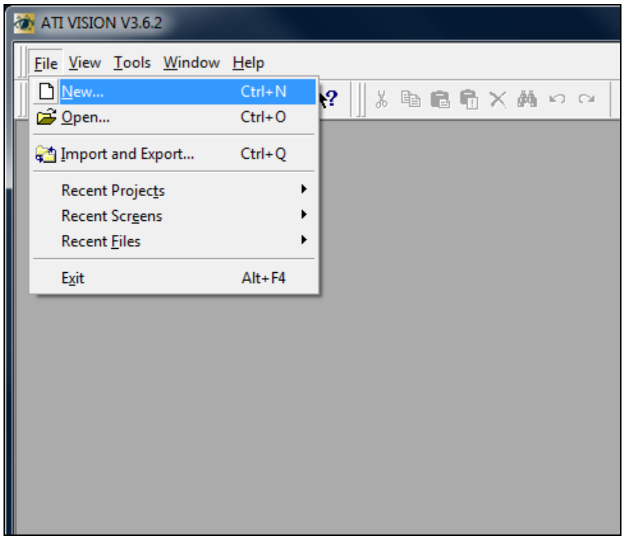


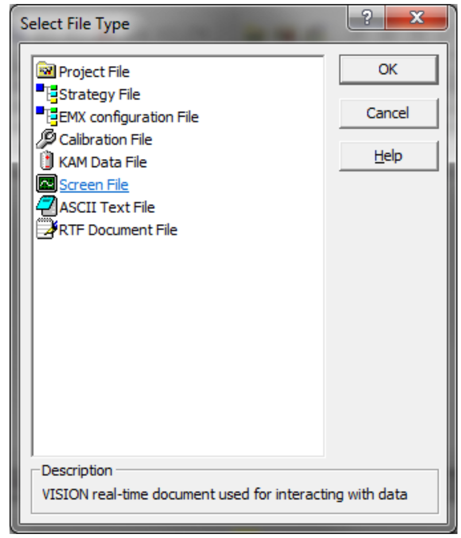


6. Enable the hardware by clicking Project → Online. The status of all of the devices should now show a Status of Online, and a value should appear in the Data Rate column of the Project window.

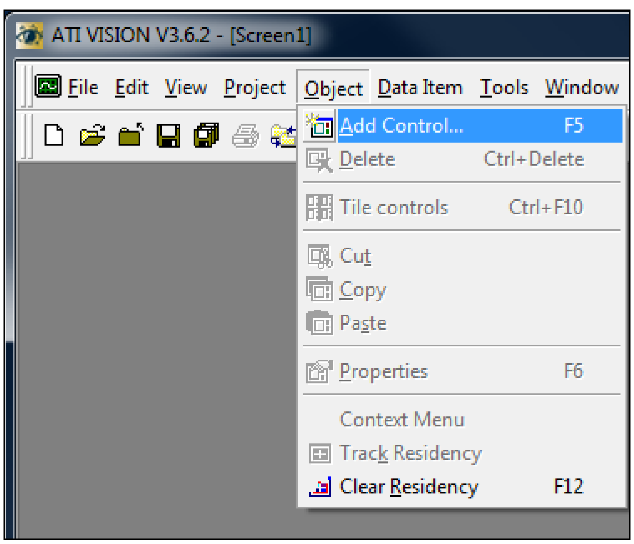


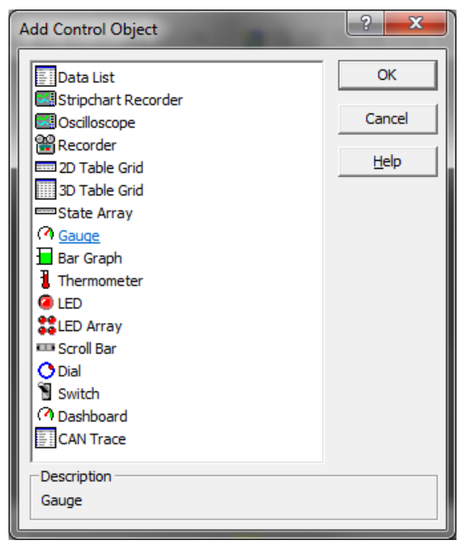
7. To view data, create a new Screen File and add a Control. Click File → New → Screen File



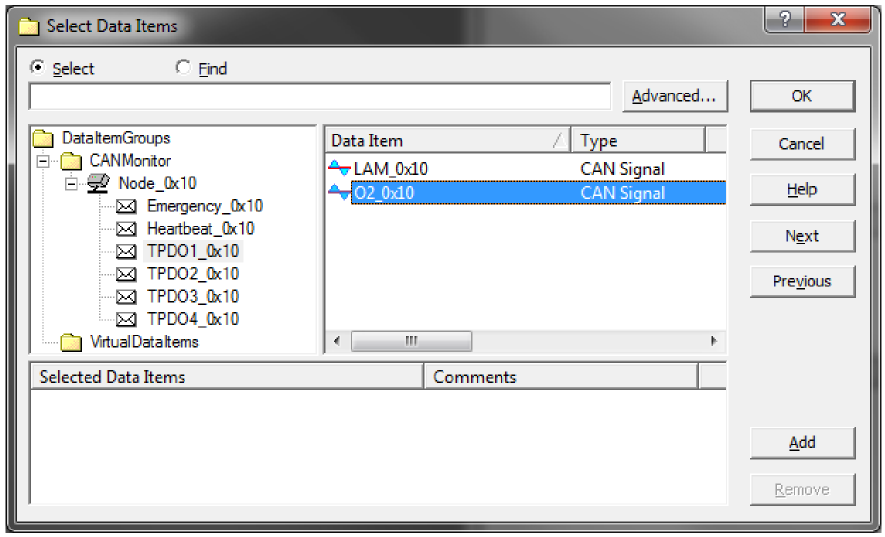


8. Select Object → Add Control → Gauge

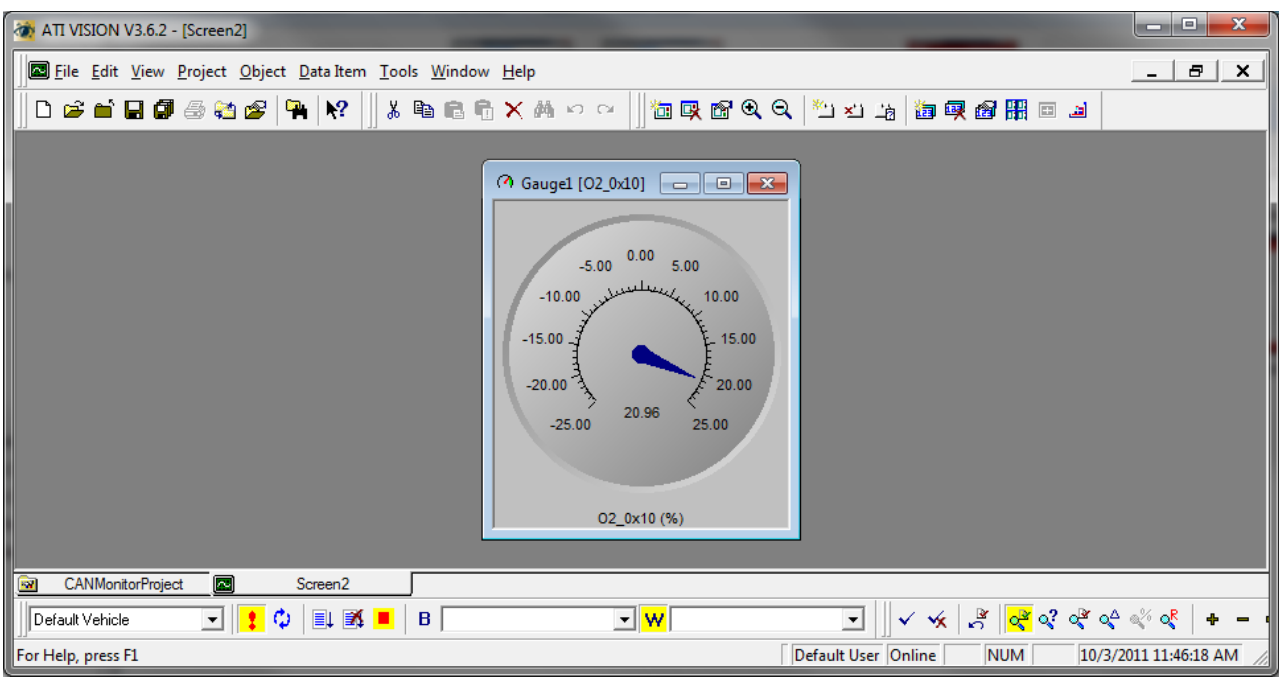




9. In the Select Data Items window open the CANMonitor file tree to view all of the available signals. Here the O2% from Node 0x10 has been selected. Click OK to add the Data Item to the Control.



10. Data should be visible on the gauge.



Appendix F: 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

586 Weddell Drive, Suite 2
Sunnyvale, CA 94089-2134
USA
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
AFM1600 Lambda and O_2 Analyzer
DIS1000 Display head
EGR 4830 Analyzer
NOx 5210 NOx Analyzer
Lambda 5220 Lambda Analyzer
EGR 5230 EGR Analyzer
LambdaCAN, LambdaCANc, LambdaCANd, LambdaCANp Lambda Modules
NOxCAN, NOxCANg, NOxCANt NOx Modules
NOx1000 NOx Module
CO/CO2CAN Module
baroCAN Module
dashCAN, dashCAN+
SIM300, SIM400, SIM500, SIM600, SIM700
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
February 20, 2012

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