ECM ENGINE CONTROL AND MONITORING

NH_3CAN

NH_3 CAN Module

Instruction Manual

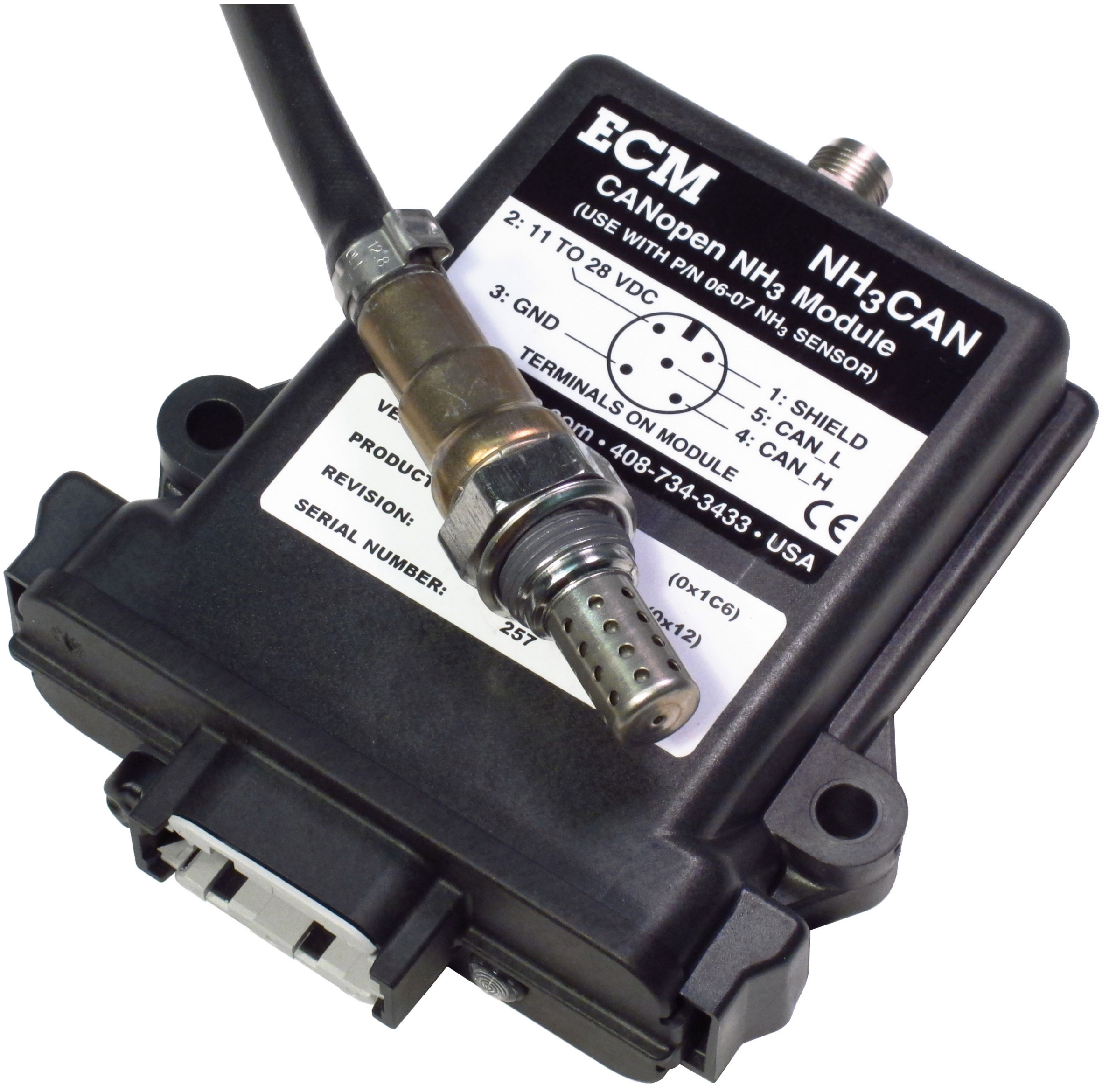
REV 1.00 2/5/2015

NH3CANmanual 1.00.doc

REVISION	DATE	DESCIPTION	SECTION
1.00	2/5/2015	Original Release	

ECM
(408) 734-3433
www.ecm-co.com

Contents


ECM
NH$_3$CAN
CANopen NH$_3$ Module
(USE WITH P/N 06-07 NH$_3$ SENSOR)
2: 11 TO 28 VDC
3: GND
1: SHIELD
5: CAN_L
4: CAN_H
TERMINALS ON MODULE
408-734-3433 • USA
PRODUCT
REVISION:
SERIAL NUMBER:
(0x1C6)
257

1.0 Introduction and General Notes

1.1 Introduction

The NH_3CAN kit is a ceramic sensor-based NH_3, measurement system that communicates and is configured via the CANopen protocol. Its primary application is for the analysis of combustion systems and their after-treatment.

1.2 General Notes

1. All numbers are decimal unless preceded by the “0x” suffix which denotes a hexadecimal value: e.g. 0x0F = 15.
2. 1 byte contains 8 bits.
3. One “unsigned character” contains 1 byte (unsigned 8) and has a range of 0x00 – 0xFF (0 – 255).
4. One “string” contains 1 byte (unsigned 8) and has a range of 0x00 – 0xFF (0-255). The number represents an ASCII character.
5. One “unsigned integer” contains 2 bytes (unsigned 16) and has a range of 0x0000 – 0xFFFF (0 – 65535).
6. One “unsigned long” contains 4 bytes (unsigned 32) and has a range of 0x00000000 – 0xFFFFFFFF (0 – 4294967295).
7. One “single float” contains 4 bytes (single float) that represent a decimal number using the IEEE-754 standard.
8. A “lo” byte and “high” byte can be combined as follows to form a 2 byte unsigned integer: e.g. lo byte = 0x10 (16), hi byte = 0x1F (31), 2 byte integer = 0x1F10 = 31*256+16 = 7952
9. “lo” byte can also be referred to as the least significant byte (LSB).
10. If a subindex value for an Object Dictionary (OD) is unspecified, it is assumed to be 0x00.
11. Data value boxes that are shown as blank are reserved; do not use these locations.
12. All messages on the CAN bus must have a unique identifier which is referred to as “CANid” in this manual but can also be referred to as “COB ID, communication object identifier”.

2.0 Parts

The NH_3CAN Kit consists of:

PART	P/N	QTY
1. NH_3CAN Control Module	02-12	1
2. NH_3 Sensor	06-07	1
3. NH_3 Extension Cable	10-02 (1m)	1
4. Eurofast "T"	09-05	1
5. Eurofast Terminating Resistor	09-06	1
6. 2m Eurofast 12mm Cable	09-02	1
7. DC Power Cable, DB9F, Banana	11-02	1
8. NH_3 Sensor Boss & Plug (18mm x 1.5mm)	12-02	1
9. Manuals and Configuration software CD	13-01	1

Optional Pressure Compensation (add /P to kit part number):

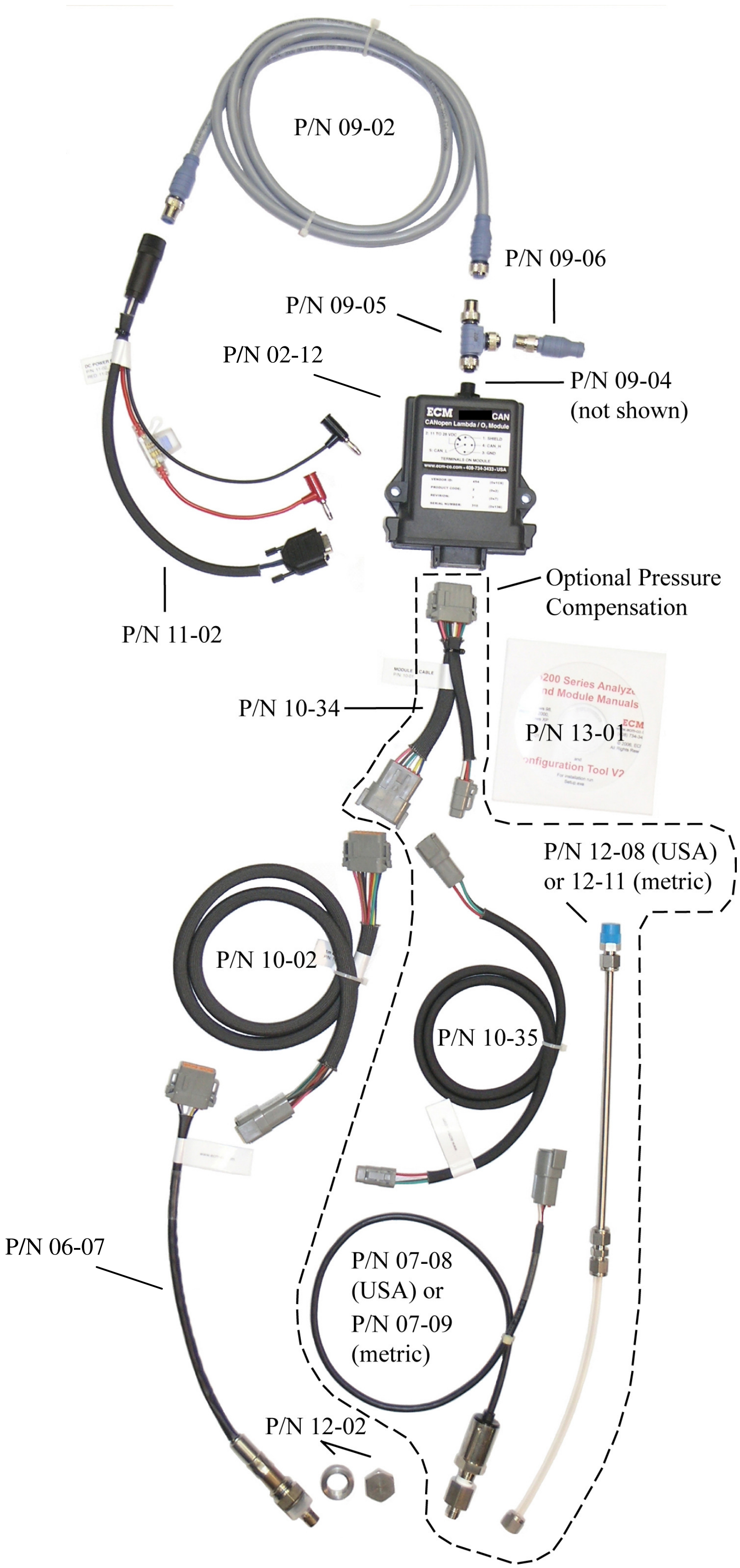
10. Pressure Sensor, 0-75 psia, 517 kPa	07-01 (USA) or 07-02 (metric)	1
11. Pressure Extension Cable	10-04 (1m)	1
12. Pressure Sensor Tubing	12-08 (USA) or 12-11 (metric)	1
13. Module Y Cable	10-01	1

Optional Cables:

NH_3 Cable	10-02 (1m) or 10-03 (2m) 10-37 (3m)
Pressure Cable	10-04 (1m) or 10-05 (2m)
DC Power Cable, DB9F, Spades	11-01

Optional Power Supplies:

AC/DC Power Supply, Universal 24VDC @ 4.2A (requires p/n 11-17 Deutsch DTM3M to DB9F)	04-01
Vboost Supply, 10-14VDC to 24VDC @ 14.5A	04-02


P/N 09-02
P/N 09-06
P/N 09-05
P/N 02-12
P/N 09-04
(not shown)
Optional Pressure
Compensation
P/N 11-02
P/N 10-34
P/N 13-01
P/N 12-08 (USA)
or 12-11 (metric)
P/N 10-02
P/N 10-35
P/N 06-07
P/N 07-08
(USA) or
P/N 07-09
(metric)
P/N 12-02

3.0 Connecting the NH_3CAN module

Power and CAN connections to the module are made using the Eurofast 12mm connector on the module. The power input requirement is 12 to 30VDC at 1.5A (steady-state). At start-up, there can be a peak current of up to 5A. Multiple modules can share the CAN bus. All modules are configured to broadcast CAN messages at the CAN baud rate of 500K. The maximum distance between any two nodes on the CAN bus at this baud rate is 100m. Each end of the CAN bus must have a terminating resistor of 121 Ohms.

3.1 Application Notes

1. Configuration software (ECM Configuration Tool) for the module is located on the CD. This software allows the setup, configuration, monitoring, and recording of data using supported CAN adapters.
2. The NH_3CAN is calibrated for NH_3 between 0 and 100ppm. The user can zero and span the NH_3 measurement. The user zeros and spans can also be cancelled, returning to the factory calibration. The factory calibration and the user zeros and spans are stored in a memory chip located in the connector of the NH_3 sensor. No matter what the NH_3 sensor is plugged into, it will read the same NH_3 value since the factory calibration, user zeros, and user spans travel with the sensor.
3. The main disturbance affecting the accuracy of the NH_3 is the temperature of the NH_3 sensor body. Thus the sensor should be zero'd and span'd under thermal conditions (i.e. exhaust pipe temperatures) close to what it will experience in service. During start-up, when the sensor body temperature is constant, the NH_3 value output by the module can be used. Typically, this takes 5 minutes when the sensor is in the exhaust of a running engine, as much as 30min when hanging in free air or in a cold manifold.
4. Do not exceed 500°C exhaust gas temperature. Short excursions up to 700°C are allowable.
5. **It is best to zero and span the NH_3 sensor in the exhaust of an engine and in comparison to a reference NH_3 measurement. If you zero and span the NH_3 measurement in a vessel fed by gases from tanks, you will need to heat the vessel walls to the temperature of the exhaust pipe the sensor will be used in, and will need to include H_2O and O_2 in the gases in the concentration(s) the sensor will see in the exhaust of the engine.**
6. Do not extend the wires between the NH_3 sensor and the control module. You can lengthen the power wires on the DC Power Cable (P/N: 11-01 or 11-02) but use large gauge wire and make sure that the voltage at the power terminals of the supplied harness is at least 14 V when the sensor is being heated. You can lengthen the CAN communication wires using Eurofast 12mm cable. Eurofast 12mm cable was designed specifically for CAN communication and along with additional "Tees", allows you to easily build long and reliable CAN networks.
7. The NH_3CAN broadcasts several messages on the CAN bus using the CANopen protocol. Each message has an identifying number known as the CAN identifier (CANid). Since multiple modules can be placed on the same CAN bus, each module on the bus also has an identifying number known as the node identifier (NID). The allowable range for the NID is 0x01 to 0x7F. When connecting other non-ECM devices on the same CAN bus, ensure that the following CANids are not used:

Message type	CANid (hex)
NMT	0x00
Emergency	0x80 + NID
TPDO1	0x180 + NID
TPDO2	0x280 + NID
TPDO3	0x380 + NID
TPDO4	0x480 + NID
SDO Tx	0x580 + NID
SDO Rx	0x600 + NID
Heartbeat	0x700 + NID
LSS	0x7E4, 0x7E5

Note this list applies to EACH ECM module on the CAN bus.

4.0 Getting Information from the NH_3CAN Module

As soon as power is attached to the NH_3CAN module, it will perform a POWER ON/RESET sequence during which the bi-color LED will display a 2 second GREEN/BOTH/RED pattern. After the POWER ON/RESET sequence is finished, the LED will flash green at 10Hz until the NH_3 sensor has reached its operating temperature. This may take up to 30 seconds. Once the sensor is ready and the NH_3 data is valid, the bi-color LED will display GREEN continuously.

If there is an error, the LED color will change to RED and flash a number of times every 2 seconds indicating an error code. This error code will also be transmitted on the CAN bus. See ECM Error codes in Appendix A.

Approximately 5 seconds after power is applied, the unit will start broadcasting CAN messages at a CAN baud rate of 500kbps. All messages are related to the Node ID (NID) of the particular module. As shipped, the NID is pre-assigned and is written on a label above the LED. The NID can be changed using the supplied configuration software.

4.1 CANopen Message Types

i) HEARTBEAT (Broadcast rate = 0.5sec, DLC=1)

CAN id	byte 0	byte 1	byte 2	byte 3	byte 4	byte 5	byte 6	byte 7
0x700+NID	value							

value = NMT STATE (see Appendix C)

ii) ERROR (Broadcast rate = 0.250sec, DLC=6)

CAN id	byte 0	byte 1	byte 2	byte 3	byte 4	byte 5	byte 6	byte 7
0x80+NID	0x00	0xFF	0x81	lo byte	0x00	count		

lo byte = ECM Error Code (0x01 = Sensor Warm-up, 0x00 = Data valid, see Appendix A)
count = Sensor Warm-up countdown in seconds (active during ECM Error Code 0x01)

iii) TRANSMIT PROCESS DATA OBJECT [TPDO] (Broadcast rate = 0.005sec, DLC=8)

TPDO1 CAN id	byte 0	byte 1	byte 2	byte 3	byte 4	byte 5	byte 6	byte 7
0x180+NID	NH3 (ppm)				MODE (hex)			
TPDO2 CAN id	byte 0	byte 1	byte 2	byte 3	byte 4	byte 5	byte 6	byte 7
0x280+NID	CEL1 (mV)				CEL2 (mV)			
TPDO3 CAN id	byte 0	byte 1	byte 2	byte 3	byte 4	byte 5	byte 6	byte 7
0x380+NID	RCL				SCF			
TPDO4 CAN id	byte 0	byte 1	byte 2	byte 3	byte 4	byte 5	byte 6	byte 7
0x480+NID	RPVS (ohms)				VHCM (V)			

The table above shows the default TPDO assignments. (see sections 8.7 Enable TPDO, 8.8 Disable TPDO and 8.9 TPDO MAPPING).

Each module can transmit up to four TRANSMIT PROCESS DATA OBJECTS (TPDO) at the programmed TPDO broadcast rate (see section 8.6 to determine minimum broadcast rate). A TPDO contains two data values; each data value corresponds to a measured parameter (e.g. NH_3). These data values are referred to as PROCESS DATA OBJECTS (PDO). Each PDO is a single precision 32 bit floating point number that conforms to the IEEE-754 standard. All TPDO data is transmitted on the CAN bus least significant byte first (Intel format).

The NID, TPDO Broadcast rate and TPDO mapping can be changed by the user.

Example: The following data was transmitted by the module with NID = 0x10 on TPDO1 and contains 2 PDOs, NH_3 and MODE.

TPDO1 CAN id	byte 0	byte 1	byte 2	byte 3	byte 4	byte 5	byte 6	byte 7
0x190	0x00	0x80	0x4A	0x43	0x78	0x42	0x00	0x00

NH3 = 0x434A8000 = 202.5
MODE = 0x00004278 = 62

Configuring which PDOs are transmitted in a particular TPDO is also known as TPDO MAPPING and can be set by the user (see 8.9 TPDO MAPPING).

5.0 Writing to the NH_3CAN Module (SDO Write)

Configuration of the NH_3CAN module is performed by writing to the Object Dictionary (OD) and by issuing ECM CANopen OS Commands (OS Command). Both of these actions are implemented using a Service Data Object Expedited Write (SDO Write). The format is as follows:

SDO Write Tx CAN id	byte 0	byte 1	byte 2	byte 3	byte 4	byte 5	byte 6	byte 7
0x600+NID	Size	OD lo	OD hi	Sub	Data0	Data1	Data2	Data3

Size = 0x2F (1 byte write)
0x2B (2 byte write)
0x23 (4 byte write)

OD lo = low byte of OD address
OD hi = hi byte of OD address
Sub = Subindex of OD address
Data0 always contains the Least Significant Byte (LSB) of the data to be written to the OD.

A SDO Write will generate the following reply:

SDO Write Rx CAN id	byte 0	byte 1	byte 2	byte 3	byte 4	byte 5	byte 6	byte 7
0x580+NID	0x60	OD lo	OD hi	Sub				

Example: Write a 2 byte integer = 0x204 to OD address 0x5017 subindex 0 in the module with NID = 0x10

SDO Write Tx CAN id	byte 0	byte 1	byte 2	byte 3	byte 4	byte 5	byte 6	byte 7
0x610	0x2B	0x17	0x50	0x00	0x04	0x02		

The module will reply as follows:

SDO Write Rx CAN id	byte 0	byte 1	byte 2	byte 3	byte 4	byte 5	byte 6	byte 7
0x590	0x60	0x17	0x50	0x00				

6.0 Reading from the NH_3CAN Module (SDO Read)

During configuration it may be necessary to read certain locations in the Object Dictionary (OD). The format for a Service Data Object Read (SDO Read) is as follows:

SDO Read Tx CAN id	byte 0	byte 1	byte 2	byte 3	byte 4	byte 5	byte 6	byte 7
0x600+NID	0x40	OD lo	OD hi	Sub				

OD lo = low byte of OD address
OD hi = hi byte of OD address
Sub = Subindex of OD address

A SDO Read will generate the following reply:

SDO Read Rx CAN id	byte 0	byte 1	byte 2	byte 3	byte 4	byte 5	byte 6	byte 7
0x580+NID	Size	OD lo	OD hi	Sub	Data0	Data1	Data2	Data3

Size = 0x4F (1 byte response)
0x4B (2 byte response)
0x43 (4 byte response)

OD lo = low byte of OD address
OD hi = hi byte of OD address
Sub = Subindex of OD address
Data0 always contains the Least Significant Byte (LSB) of the data present at the OD address.

Example: Read OD address 0x5008 subindex 0x32 in the module with NID = 0x10

SDO Write Tx CAN id	byte 0	byte 1	byte 2	byte 3	byte 4	byte 5	byte 6	byte 7
0x610	0x40	0x08	0x50	0x32				

The module will reply as follows:

SDO Write Rx CAN id	byte 0	byte 1	byte 2	byte 3	byte 4	byte 5	byte 6	byte 7
0x590	0x4B	0x08	0x50	0x32	0xBC	0x02		

OD address 0x5008, subindex 0x32 of the module with NID = 0x10 contains the 2 byte value 0x2BC

7.0 Identifying the NH_3CAN Module

Each NH_3CAN module can be uniquely identified by reading the following four parameters in the OD:

i) Vendor ID (0x000001C6) located at OD address 0x1018, subindex 0x01 (4 byte integer/unsigned 32)
ii) Product Code (NH_3CAN = 0x00000012) located at OD address 0x1018 subindex 0x02 (4 byte integer/unsigned 32)
iii) Revision Number located at OD address 0x1018, subindex 0x03 (4 byte integer/unsigned 32)
iv) Serial Number located at OD address 0x1018, subindex 0x04 (4 byte integer/unsigned 32)

Furthermore, the hardware and software revision number can be found at the following locations:

i) Hardware Revision is located at OD address 0x1009, subindex 0x00 (4 byte string)
ii) Software Revision is located at OD address 0x100A, subindex 0x00 (4 byte string)

8.0 Commands to the NH_3CAN Module

There are several commands that can be used with the NH_3CAN module. The commands are executed by performing an SDO Write to OD address 0x1023, subindex 0x01. A list of the commands can be found in Appendix B.

8.1 ZEROing and SPANing

All calibration information, including sensor constants and the user-performed span, is stored in a memory chip located in the connector of the NH_3 sensor. The NH_3 value does not depend on what module the sensor is plugged into, they depend on the factory calibration and user zero and span information stored in the sensor's connector that is attached to the sensor.

To compensate for sensor aging, the NH_3 sensor can be recalibrated. To perform a recalibration of the NH_3 measurement, three messages must be sent to the NH_3CAN. The procedure is as follows:

i) SDO Write to OD address 0x5000 of the current NH_3 value output by the module.
ii) SDO Write to OD address 0x5001 of the true/correct NH_3 value.
iii) OS Command to ZERO/SPAN NH_3.

The first message contains the NH_3CAN output for NH_3. Bytes 4-7 contain a single precision 32bit floating point value that conforms to the IEEE-754 standard. It is loaded least significant byte first (Intel format).

CANid	byte 0	byte 1	byte 2	byte 3	byte 4	byte 5	byte 6	byte 7
0x600+NID	0x23	0x00	0x50	0x00	value output by NH_3CAN (float)			

The second message contains the true value of NH_3.

CANid	byte 0	byte 1	byte 2	byte 3	byte 4	byte 5	byte 6	byte 7
0x600+NID	0x23	0x01	0x50	0x00	true value (float)			

The third message sent is the OS Command to ZERO NH_3, or SPAN NH_3.

To ZERO NH_3 send:

CANid	byte 0	byte 1	byte 2	byte 3	byte 4	byte 5	byte 6	byte 7
0x600+NID	0x2F	0x23	0x10	0x01	0x0F			

To SPAN NH_3 send:

CANid	byte 0	byte 1	byte 2	byte 3	byte 4	byte 5	byte 6	byte 7
0x600+NID	0x2F	0x23	0x10	0x01	0x10			

Example: A module with NID = 0x02 has a NH_3 sensor which is currently transmitting a reading 19.5ppm NH_3 (0x419C0000). The desired or true value is 20.95ppm (0x41A7999A). The following messages are sent to the module perform the SPAN:

CANid	byte 0	byte 1	byte 2	byte 3	byte 4	byte 5	byte 6	byte 7
0x602	0x23	0x00	0x50	0x00	0x00	0x00	0x9C	0x41
0x602	0x23	0x01	0x50	0x00	0x9A	0x99	0xA7	0x41
0x602	0x2F	0x23	0x10	0x01	0x0E			

If the ECM Error Code is from 0x10 to 0x3F inclusive, there is a problem with the module and/or the sensor memory. Therefore the SPAN will be ignored. A successful SPAN can be determined by reading locations 0x5000 and 0x5001 in the OD by performing an SDO Read. If the SPAN was successful those locations will read as 99999.0. There is an OS Command Reply which can be read as well to determine if the SPAN was successful (see Appendix B).

8.2 Canceling (Erasing) the ZERO and SPAN

The following command, when sent to the NH_3CAN module, erases the user NH_3 ZERO and SPAN information stored in the sensor's connector. Thus, the NH_3 sensor reverts to the factory calibration:

CANid	byte 0	byte 1	byte 2	byte 3	byte 4	byte 5	byte 6	byte 7
0x600+NID	0x2F	0x23	0x10	0x01	0x12			

8.3 Modifying the Pre-Broadcast Averaging of Data

The NH_3 sensor output (CEL1 and CEL2) voltage data is averaged prior to being broadcast. Each is averaged by the module every 5ms independent of the TPDO broadcast rate (see section 8.6). The averaging filter (α) can range from 0.001 (heavy averaging) to 1.000 (no averaging). The averaging filters (also called recursive averaging filters or digital low-pass filters) are used as follows:

$$AvgData_{n+1} = \alpha \text{ x } Data_n + (1 - \alpha) \text{ x } AvgData_n$$

Where:
$AvgData_{n+1}$ = Current averaged data value broadcast by the module.
α = User-programmable averaging filter, also called "alpha". (Range 0.001 to 1.000)
$Data_n$ = Current data value measured by the module.
$AvgData_n$ = Previous averaged data value broadcast by the module 5ms ago.

There is one α for CEL1 and one α for CEL2. The defaults are 0.375 and are stored in non-volatile memory (EEprom) in the module. Note that NH_3 is calculated from both the CEL1 and CEL2 measurements and therefore are affected by both α averaging values. For best performance set both alphas to the same value.

The α value is loaded into the module as a scaled (x1000) unsigned 16bit integer sent least significant byte (LSB) first (Intel format). This value is written to OD address 0x5012 by performing an SDO Write. Note that the subindex determines which parameter is affected and averaging values beyond the range specified are limited to the appropriate maximum or minimum.

CANid	byte 0	byte 1	Byte 2	byte 3	byte 4	byte 5	byte 6	byte 7
0x600+NID	0x2B	0x12	0x50	subindex	α x 1000 lo byte	α x 1000 hi byte		

subindex = 0x08 for CEL1
= 0x09 for CEL2

α x 1000 = 1-1000 (0x0001 – 0x03E8)

Example: Set the α for CEL1 to 0.256 for the module with NID = 0x05. Multiply 0.256 x 1000 = 256 (0x0100).

CANid	byte 0	byte 1	byte 2	byte 3	byte 4	byte 5	byte 6	byte 7
0x605	0x2B	0x12	0x50	0x08	0x00	0x01		

8.4 Returning the Pre-Broadcast Averaging to Factory Default

The averaging values can be reset to factory default (375) by issuing the ECM OS Command 0x15 (see Appendix B).

CANid	byte 0	byte 1	byte 2	byte 3	byte 4	byte 5	byte 6	byte 7
0x600+NID	0x2F	0x23	0x10	0x01	0x15			

8.5 Changing the NID

The Node ID (NID) can be programmed from 0x01 to 0x7F (1 to 127). To change the NID, several messages must be sent to the NH_3CAN module. This must be followed by a reset of the module (that can be performed three different ways; see the following).

Start by sending the following message to place the module into pre-operational mode.

CAN id	byte 0	byte 1
0x00	0x80	NID

The next message(s) place the module(s) into LSS (Layer Select Services) configuration mode. If there is only one CANopen module on the CAN bus this process requires only one message. If there are several CANopen modules on the same CAN bus the specific module must be identified using Product Code, Revision Number and Serial Number, (these can be found on a white label placed on the top of the plastic enclosure).

MULTIPLE MODULES ON BUS

CAN id	byte 0	byte 1	byte 2	byte 3	byte 4
0x7E5	0x04	0x00			
0x7E5	0x40	0xC6	0x01	0x00	0x00
0x7E5	0x41	Product Code			
0x7E5	0x42	Revision Number			
0x7E5	0x43	Serial Number			

SINGLE MODULE ON BUS

CAN id	byte 0	byte 1
0x7E5	0x04	0x01

The module will reply with byte 0 = 0x44 on CAN id 0x7E4 if it enters LSS configuration mode successfully.

The next message sent contains the new NID as an unsigned hexadecimal character.

CAN id	byte 0	byte 1
0x7E5	0x11	new NID

The module will reply with byte 0 = 0x11 and byte 1 = 0x00 on CAN id 0x7E4 indicating a successful NID change.

The last message sent takes the module out of configuration mode.

CAN id	byte 0	byte 1
0x7E5	0x04	0x00

After the NID has been successfully changed, the module enters pre-operational mode and does not broadcast data. The module can be returned to broadcast mode 1 of 3 ways:

i) Power-cycle the module by disconnecting and reconnecting the power.
ii) A second method is to send a command instructing the module to perform a hard reset (similar to power-cycling the module but software controlled).

CAN id	byte 0	byte 1
0x00	0x81	NID

iii) A third method is to send a command instructing the module to reset the CAN interface only.

CAN id	byte 0	byte 1
0x00	0x82	NID

Example: Change the NID for the following module with **multiple modules** on the CAN bus.

CURRENT NID = 0x10 (16)
PRODUCT CODE = 0x03 (3)
REVISION NUMBER = 0x03 (3)
SERIAL NUMBER = 0x192 (402)
NEW NID = 0x1A (26)

MESSAGE SENT

CAN id	byte 0	byte 1	byte 2	byte 3	byte 4
0x00	0x80	0x10			
0x7E5	0x04	0x00			
0x7E5	0x40	0xC6	0x01	0x00	0x00
0x7E5	0x41	0x03	0x00	0x00	0x00
0x7E5	0x42	0x03	0x00	0x00	0x00
0x7E5	0x43	0x92	0x01	0x00	0x00
0x7E5	0x11	0x1A			
0x7E5	0x04	0x00			
0x00	0x82	0x1A			

MODULE REPLY

CAN id	byte 0	byte 1
0x7E4	0x44	
0x7E4	0x11	0x00

Example: Change the NID for the **only CANopen module** on the CAN bus.

CURRENT NID = 0x10 (16)
NEW NID = 0x1A (26)

MESSAGE SENT

CAN id	byte 0	byte 1
0x00	0x80	0x10
0x7E5	0x40	0x01
0x7E5	0x11	0x1A
0x7E5	0x04	0x00
0x00	0x82	0x1A

MODULE REPLY

CAN id	byte 0	byte 1
0x7E4	0x44	
0x7E4	0x11	0x00

8.6 Changing the TPDO Broadcast Rate

The data broadcast rate can be programmed from 5 ms to 65535 ms and applies to all TPDOs that have been enabled (see section 8.7). It is an unsigned 16bit integer (2 bytes) written least significant byte (LSB) first (Intel format) to OD address 0x1800, subindex 0x05. The format of the SDO Write to the LambdaCAN module is as follows:

CANid	byte 0	byte 1	byte 2	byte 3	byte 4	byte 5	byte 6	byte 7
0x600+NID	0x2B	0x00	0x18	0x05	Broadcast rate lo	Broadcast rate hi		

Example: Set TPDO broadcast rate to 500 ms (0x01F4) for the module with NID = 0x0F (15).

CANid	byte 0	byte 1	byte 2	byte 3	byte 4	byte 5	byte 6	byte 7
0x60F	0x2B	0x00	0x18	0x05	0xF4	0x01		

There is a minimum broadcast rate that is dependent on the number of modules transmitting on the CAN bus and how many TPDOs have been enabled for each module. If the broadcast rate is too fast the ECM Configuration Tool will not be able to identify or configure any of the modules. The formula for calculating the minimum broadcast rate is as follows:

Minimum Broadcast Rate (ms) > Total number of TPDOs for all modules x 0.3125

Example: There are 8 modules on the CAN bus.
NID 0x01 has 3 TPDOs enabled
NID 0x02 has 1 TPDOs enabled
NID 0x03 has 4 TPDOs enabled
NID 0x04 has 2 TPDOs enabled
NID 0x05 has 4 TPDOs enabled
NID 0x06 has 4 TPDOs enabled
NID 0x07 has 4 TPDOs enabled
NID 0x08 has 4 TPDOs enabled

Minimum Broadcast Rate (ms) = (3 + 1 + 4 + 2 + 4 + 4 + 4 + 4) x 0.3125 = 8.125ms. Since the broadcast rate is valid only in increments of 1ms, round 8.125ms up to the next integer value; 9ms. Therefore no module can have a TPDO broadcast rate less than 9ms.

8.7 Enable Transmit Process Data Object (TPDO)

There are four TPDOs, each can be individually enabled to transmit the mapped PDO data at the broadcast rate. The following OD addresses are required to enable each TPDO.

TPDO	EnableOD Address	Transmit CANid
TPDO1	0x1800	0x180 + NID
TPDO2	0x1801	0x280 + NID
TPDO3	0x1802	0x380 + NID
TPDO4	0x1803	0x480 + NID

To enable a TPDO perform a SDO Write to the Enable OD Address for that particular TPDO as follows:

CANid	byte 0	byte 1	byte 2	byte 3	byte 4	byte 5	byte 6	byte 7
0x600+NID	0x23	EnableOD Address lo	EnableOD Address hi	0x01	Transmit CANid lo	Transmit CANid hi	0x00	0x40

Example: Enable TPDO4 for the module with NID = 0x20, (EnableOD Address = 0x1803, Transmit CANid = 0x480 + 0x20 = 0x4A0).

CANid	byte 0	byte 1	byte 2	byte 3	byte 4	byte 5	byte 6	byte 7
0x620	0x23	0x03	0x18	0x01	0xA0	0x04	0x00	0x40

8.8 Disable Transmit Process Data Object (TPDO)

The following OD addresses are required to disable each TPDO.

TPDO	EnableOD Address	Transmit CANid
TPDO1	0x1800	0x180 + NID
TPDO2	0x1801	0x280 + NID
TPDO3	0x1802	0x380 + NID
TPDO4	0x1803	0x480 + NID

To disable a TPDO perform a SDO Write to the Enable OD Address for that particular TPDO as follows:

CANid	byte 0	byte 1	byte 2	byte 3	byte 4	byte 5	byte 6	byte 7
0x600+NID	0x23	EnableOD Address lo	EnableOD Address hi	0x01	Transmit CANid lo	Transmit CANid hi	0x00	0xC0

Example: Enable TPDO1 for the module with NID = 0x10, (EnableOD Address = 0x1800, Transmit CANid = 0x180 + 0x10 = 0x190).

CANid	byte 0	byte 1	byte 2	byte 3	byte 4	byte 5	byte 6	byte 7
0x620	0x23	0x00	0x18	0x01	0x90	0x01	0x00	0xC0

8.9 Transmit Process Data Object Mapping (TPDO MAPPING)

Each TPDO transmits two PROCESS DATA OBJECTS (PDOs). Which PDOs are transmitted by the module in a particular TPDO can be configured by the user.

Configuring a TPDO is a 4 step process:

i) Write a 0 to the TPDO Configuration OD Address, subindex 0x00.
ii) Enter the OD address of the 1st PDO.
(see Appendix D PROCESS DATA OBJECTS)
iii) Enter the OD address of the 2nd PDO.
iv) Enter the number of PDOs in the TPDO.

Also, the following information is required to successfully map a TPDO.

TPDO	ConfigOD Address	EnableOD Address	Transmit CANid
TPDO1	0x1A00	0x1800	0x180 + NID
TPDO2	0x1A01	0x1801	0x280 + NID
TPDO3	0x1A02	0x1802	0x380 + NID
TPDO4	0x1A03	0x1803	0x480 + NID

Write a 0 to the TPDO Configuration OD Address, subindex 0x00 by performing a SDO Write as follows:

CANid	byte 0	byte 1	byte 2	byte 3	byte 4	byte 5	byte 6	byte 7
0x600+NID	0x2F	ConfigOD Address lo	ConfigOD Address hi	0x00	0x00			

Configure the 1st PDO by performing a SDO Write follows:

CANid	byte 0	byte 1	byte 2	byte 3	byte 4	byte 5	byte 6	byte 7
0x600+NID	0x23	ConfigOD Address lo	ConfigOD Address hi	0x01	0x20	0x00	PDO OD Address lo	PDO OD Address hi

Configure the 2nd PDO by performing a SDO Write follows:

CANid	byte 0	byte 1	byte 2	byte 3	byte 4	byte 5	byte 6	byte 7
0x600+NID	0x23	ConfigOD Address lo	ConfigOD Address hi	0x02	0x20	0x00	PDO OD Address lo	PDO OD Address hi

Enter the number of PDOs in the TPDO by performing a SDO Write as follows:

CANid	byte 0	byte 1	byte 2	byte 3	byte 4	byte 5	byte 6	byte 7
0x600+NID	0x2F	ConfigOD Address lo	ConfigOD Address hi	0x00	0x02			

Example: Map the PDO for Pressure (mmHg) and NH_3 to TPDO2 for the module with NID = 0x02. (Pressure PDO OD Address = 0x2016, NH_3 PDO OD Address = 0x201C, ConfigOD Address for TPDO2 = 0x1A01)

CANid	byte 0	byte 1	byte 2	byte 3	byte 4	byte 5	byte 6	byte 7
0x602	0x2F	0x01	0x1A	0x00	0x00			
0x602	0x23	0x01	0x1A	0x01	0x20	0x00	0x16	0x20
0x602	0x23	0x01	0x1A	0x02	0x20	0x00	0x1C	0x20
0x602	0x2F	0x01	0x1A	0x00	0x02			

Appendix A: Error Codes and Error Register

CAN id	byte 0	byte 1	byte 2	byte 3	byte 4	byte 5	byte 6	byte 7
0x80+NID	Error Code lo	Error Code hi	Error Register	ECM Error lo	ECM Error hi	ECM Aux		

ERROR CODE	**DESCRIPTION OF ERRORS**
0x0000 – 00FF	No error or error reset
0x1000 – 10FF	Generic
0x2000 – 20FF	Current
0x2100 – 21FF	Current – Device inputs
0x2200 – 22FF	Current – Inside the module
0x3000 – 30FF	Voltage
0x3100 – 31FF	Voltage – Main voltage
0x3200 – 32FF	Voltage – Inside the module
0x3300 – 33FF	Voltage – Output
0x4000 – 40FF	Temperature
0x4100 – 41FF	Temperature – Ambient
0x4200 – 42FF	Temperature – Device
0x5000 – 50FF	Device Hardware
0x6000 – 60FF	Device Software
0x6100 – 61FF	Device Software – Internal
0x6200 – 62FF	Device Software – User
0x6300 – 63FF	Device Software – Data set
0x7000 – 70FF	Additional Modules
0x8000 – 80FF	Monitoring
0x8100 – 81FF	Monitoring – Communication
0x8110	Monitoring – CAN Overrun (objects lost)
0x8120	Monitoring – CAN in error passive mode
0x8130	Monitoring – Node Guarding or Heartbeat Error
0x8140	Monitoring – Recovering from bus off
0x8150	Monitoring – Transmit COB ID collision
0x8200 – 82FF	Protocol
0x8210	Protocol – PDO not processed due to length error
0x8220	Protocol – PDO length exceeded
0x9000 – 90FF	External
0xF000 – F0FF	Additional functions
0xFF00 – FFFF	Device specific
ERROR REGISTER	
BIT	**DESCRIPTION**
0	Generic Error
1	Current
2	Voltage
3	Temperature
4	Communication Error
5	Device profile defined error
6	Reserved (always 0)
7	Manufacturer Specific Error

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 > 32V
0x0041	Pulse Red 4x/2s	VS too high
0x0051	Pulse Red 5x/2s	RVS to high
0x0052	Pulse Red 5x/2s	(VHcommanded - VHactual) > 0.5V for > 10sec
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
0x00B7	N/A	Can overrun Lss
0x00B8	N/A	Can overrun Sdo
0x00B9	N/A	Can overrun Rx
0x00BA	N/A	Can overrun ECT5
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)

Appendix B: ECM CANopen OS Commands

A user-specific CANopen OS Command to the NH_3CAN module is sent using an SDO expedited write message in the following form. These commands apply only to the NH_3CAN module and are listed on the following page:

CANid	byte 0	byte 1	byte 2	byte 3	byte 4	byte 5	byte 6	byte 7
0x600+NID	0x2F	0x23	0x10	0x01	Command			

Issuing a SDO Read of OD address 0x1023, subindex 0x02 will indicate the status of the command.

CANid	byte 0	byte 1	byte 2	byte 3	byte 4	byte 5	byte 6	byte 7
0x600+NID	0x40	0x23	0x10	0x02				

The module will reply as follows:

CANid	byte 0	byte 1	byte 2	byte 3	byte 4	byte 5	byte 6	byte 7
0x580+NID	0x4F	0x23	0x10	0x02	Status			

The values that may be returned are listed below.

Status	
0x00	Last command completed. No error occurred. No reply.
0x01	Last command completed. No error occurred. The reply can now be read.
0x02	Last command completed. Error occurred. No reply.
0x03	Last command completed. Error occurred. The reply can now be read.
0x04 - FE	Reserved
0xFF	Command is executing.

If there is a reply it can read using an SDO Read of OD address 0x1023, subindex 0x03.

CANid	byte 0	byte 1	byte 2	byte 3	byte 4	byte 5	byte 6	byte 7
0x600+NID	0x40	0x23	0x10	0x03				

The reply value will be located in byte 4 of the response to the SDO Read.

CANid	byte 0	byte 1	byte 2	byte 3	byte 4	byte 5	byte 6	byte 7
0x580+NID	0x4F	0x23	0x10	0x03	Reply			

The reply values and what they indicate are listed on the following page. Commands that are written in *ITALICS* are valid only in Expert Mode.

Command	Value	Description	Reply
SensorOn	0x07	Turn on sensor power	None
SensorOff	0x08	Turn off sensor power	None
OWDisable	0x0A	Ignore 1wire memory (OWDisabled) and use EE memory values, unit subsequently performs defForceOW/EERead	None
OWEnable	0x0B	Use 1wire memory values (OWEnabled), unit subsequently performs defForceOW/EERead	None
ForceOW/EERead	0x0C	Force the module to read the 1wire memory (OWEnabled) **OR** read the EEprom (OWDisabled)	
		defOWReadSuccessfully	0x00
		defEEReadSuccessfully	0x01
		defOWInvalidSenType	0xFD
		defOWZeroSpanDataCRCFail	0xFE
		defOWReadError	0xFF
ZeroNH3	0x0F	Set zero for NH3, requires write to OD 0x5000 (old value) and 0x5001 (desired value)	
		defZeroSpanSuccessful	0x00
		defSenModNotReady	0xFD
		defZeroSpanDataInvalid	0xFE
		defOWZeroSpanWrFail	0xFF
SpanNH3	0x10	Set span for NH3, requires write to OD 0x5000 (old value) and 0x5001 (desired value)	
		defZeroSpanSuccessful	0x00
		defSpanInvalidNegativeSlope	0xFB
		defSpanTooCloseToOffset	0xFC
		defSenModNotReady	0xFD
		defZeroSpanDataInvalid	0xFE
		defOWZeroSpanWrFail	0xFF
ResetNH3	0x12	Resets the zero and span for NH3 back to factory values (M=1, b=0, x=0)	
		defZeroSpanSuccessful	0x00
		defSenModNotReady	0xFD
		defZeroSpanDataInvalid	0xFE
		defOWZeroSpanWrFail	0xFF
ResetAllFilters	0x15	Resets alpha for recursive average of NH3 to factory value (alpha=375)	
		defAlphaOK	0x00
ExpertModeDisable	0x16	This command removes the unit from expert mode.	None
ResetDeltaNH3Table	0x1D	Set all delta %O2 values to 0.	None

ResetTPDOs	0x1F	Set all TPDOs as delivered from factory.	None
FastSensorStart	0x20	Use sensor start parameters in 1wire memory	None
SlowSensorStart	0x21	Sensor start draws <1A max	None
FactoryReset	0xDF	Set all EE values to standard configuration	None
ExpertModeEnable	0xE0	This command places the unit in expert mode (password required)	None
HtrPidEnable	0xE1	Enable heater PID control	None
HtrPidDisable	0xE2	Disable heater PID control	None
RvsEnable	0xE3	Enable RVS measurement	None
RvsDisable	0xE4	Disable RVS measurement	None
UseCalibratedValues	0xE5	Use calibrated ADC/DAC values	None
UseUnCalibratedValues	0xE6	Use raw uncalibrated ADC/DAC values	None
UseProgrammedFilter	0xE7	Use programmed recursive average filter constants (std)	None
UseCalibrationFilter	0xE8	Use slow recursive average filter constants (cal)	None

Appendix C: Heartbeat

A Heartbeat message is transmitted every 0.5 seconds by the NH_3CAN module. During normal operation the module is in operational mode (NMT state = 0x05).

CAN id	byte 0	byte 1	byte 2	byte 3	byte 4	byte 5	byte 6	byte 7
0x700+NID	NMT state							

NMT state	
0x00	Boot-up
0x04	Stopped
0x05	Operational
0x7F	Pre-operational

Appendix D: Process Data Objects

Address	Type	Symbol	Description
0x2001	single float	NH3R	PDO Data NH3real (ppm)
0x2002	single float	CEL1	PDO Data Cell1 Voltage (mV)
0x2003	single float	CEL2	PDO Data Cell2 Voltage (mV)
0x2004	single float	RPVS	PDO Data RPVS (ohms) * 1000
0x2005	single float	VHCM	PDO Data VH Commanded (V) * 1000
0x2006	single float	VS	PDO Data VS (V) * 1000
0x2009	single float	VSW	PDO Data Vsw (V) * 1000
0x200A	single float	VH	PDO Data VH Measured (V) * 1000
0x200B	single float	TEMP	PDO Data Board Temperature (deg C) * 100
0x200C	single float	C1R	PDO Data Cell1 raw (bits)
0x200D	single float	C2R	PDO Data Cell2 raw (bits)
0x200E	single float	ERFL	PDO Data Error bit flags
0x200F	single float	ERCD	PDO Data ECM CANOpen Error Code
0x2010	single float	PR10	PDO Data Praw (bits)
0x2016	single float	P	PDO Data Pressure (mmHg)
0x2017	single float	LAMR	PDO Data LAMBDAreal
0x2018	single float	MODE	PDO Data Measurement Mode
0x2019	single float	RCL	PDO Data RCL
0x201A	single float	SCF	PDO Data SCF
0x201C	single float	NH3	PDO Data NH3 (ppm)
0x201E	single float	PVLT	PDO Data Pressure (V)
0x201F	single float	PKPA	PDO Data Pressure (kPa)
0x2020	single float	PBAR	PDO Data Pressure (bar)
0x2021	single float	PPSI	PDO Data Pressure (psi)

Appendix E: Module EIB mode and Stand-alone Mode

The NH_3CAN module can be used in conjunction with a display head (EIB mode) or on its own (Stand-alone mode). When delivered to be used with a display head, the module is setup in EIB mode. When delivered to be used alone, the module is setup in Stand-alone mode.

In EIB mode, the module is polled by the display head and the data transmission is synchronized on the CAN bus via the CANopen protocol. The module must be in EIB mode when on the EIB bus with a display head. When in Stand-alone Mode, the module broadcasts its data on the 500kbps CAN bus at the programmed broadcast rate (see section 4.0 and section 8.6).

The module must be properly configured in EIB mode or Stand-alone mode depending on how it will be used.

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

<u>To convert a module from Stand-alone to EIB Mode</u>

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

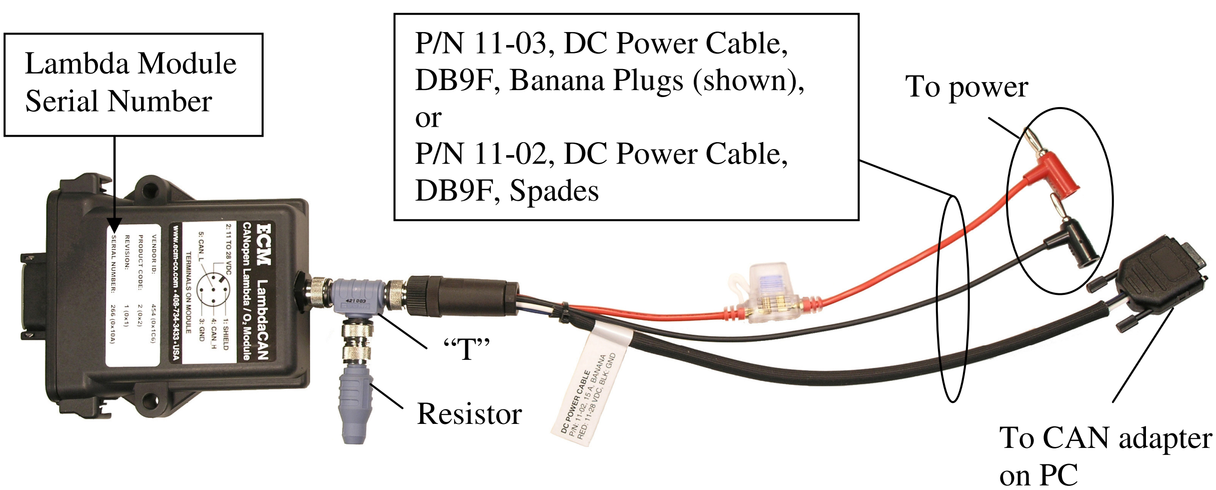
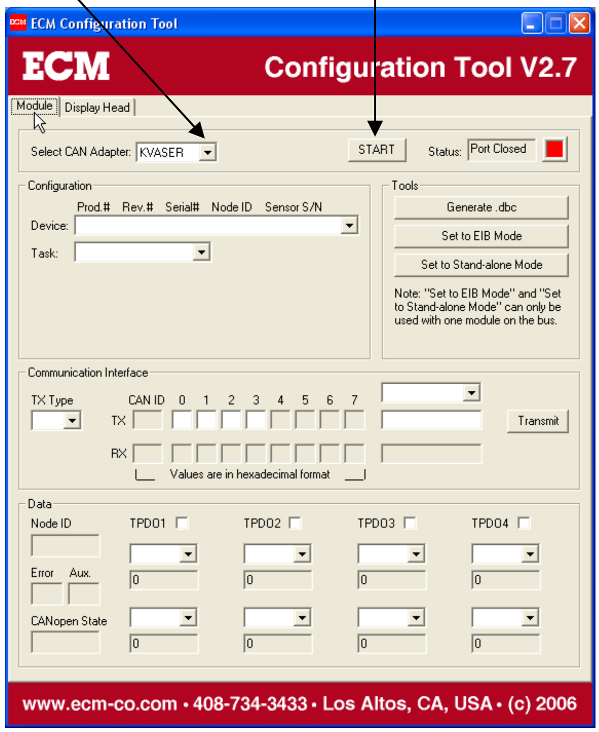
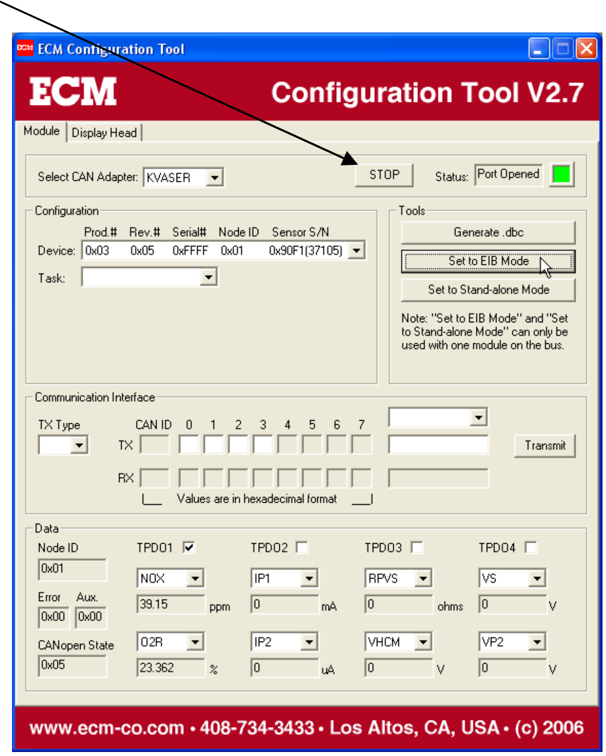


Figure B1: LambdaCAN Module prepared for Reprogramming

2. Install and start the Configuration Tool (software). Click on the “Module” tab. Select the CAN adapter being used. Then start the communication.

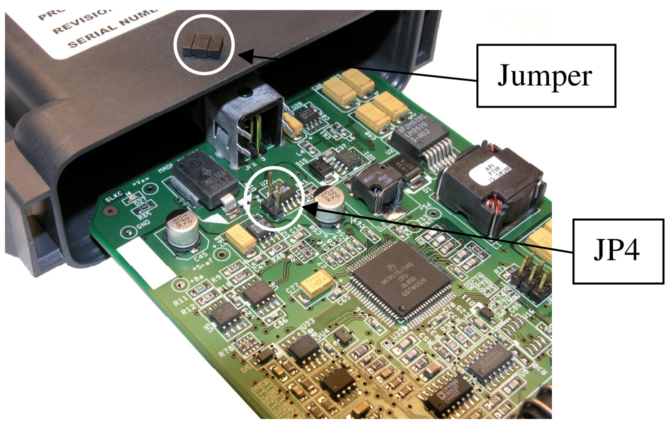


3. Click on the “Set to EIB Mode”. Wait for “Done” Message. Stop communication and exit program.



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

6. Slide the PCB out. Remove the jumper from JP4. You can hang it on one pin of JP4 when “off”.



7. Slide the PCB into the enclosure until the two tangs “click”.
8. The lambda module is now in EIB mode and can be on the EIB with a display.

<u>To convert a module from EIB to Stand-alone Mode</u>

The process is similar to the previously-described procedure. Note that in EIB mode, the module will not show up in the device list of the configuration program.

1. Use the Configuration Tool (software) to “Set to Stand-alone Mode”.
2. Install the jumper on JP4 in the lambda module.
3. In Stand-alone Mode, the module will continuously broadcast data via 500kbps CAN.

Appendix F: Warranty and Disclaimers

WARRANTY

The products described in this manual, with the exception of the lambda and pressure sensors, are warranted to be free from defects in material and workmanship for a period of 365 days from the date of shipment to the buyer. Within the 365 day warranty period, we shall at our option repair such items or reimburse the customer the original price of such items which are returned to us with shipping charges prepaid and which are determined by us to be defective. This warranty does not apply to any item which has been subjected to misuse, negligence or accident; or misapplied; or modified; or improperly installed.

The lambda and pressure sensors are considered an expendable part and as such cannot be covered by a warranty.

This warranty comprises the sole and entire warranty pertaining to the items provided hereunder. Seller makes no other warranty, guarantee, or representation of any kind whatsoever. All other warranties, including but not limited to merchantability and fitness for purpose, whether express, implied, or arising by operation of law, trade usage, or course of dealing are hereby disclaimed.

The warranty is void if the display head is opened.

LIMITATION OF REMEDY

Seller's liability arising from or in any way connected with the items sold and/or services provided shall be limited exclusively to repair or replacement of the items sold or refund of the purchase price paid by buyer, at seller's sole option. In no event shall seller be liable for any incidental, consequential or special damages of any kind or nature whatsoever, including but not limited to lost profits arising from or in any way connected with items sold and/or services provided to buyer, whether alleged to arise from breach of contract, express or implied warranty, or in tort, including without limitation, negligence, failure to warn or strict liability. In no event shall the company's liability to buyer arising out of or relating to the sale of any product or service exceed the purchase price paid by buyer to the company for such product or service.

PRODUCT CHANGES

We reserve the right to discontinue a particular product or to make technical design changes at any time without notice.

ECM ENGINE CONTROL AND MONITORING

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

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 meter
DIS1000 Display head
Lambda 5220 Lambda meter
NH3 5210 NH3 meter
EGR 5230 EGR meter
LambdaCAN Lambda module
LambdaCANp Lambda module
NOxCANt NOx module
NH3CAN NH3 module
NOxCANg NOx module
BTU200 Ceramic Sensor Heater
EGR 4830 Analyzer
SIM300
SIM400

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)

RSPatrick

Ronald S. Patrick
Vice President Sales
February 5, 2015