ECM ENGINE CONTROL AND MONITORING

LambdaCANp

Lambda CAN Module

Instruction Manual

REV 1.31 9/11/14

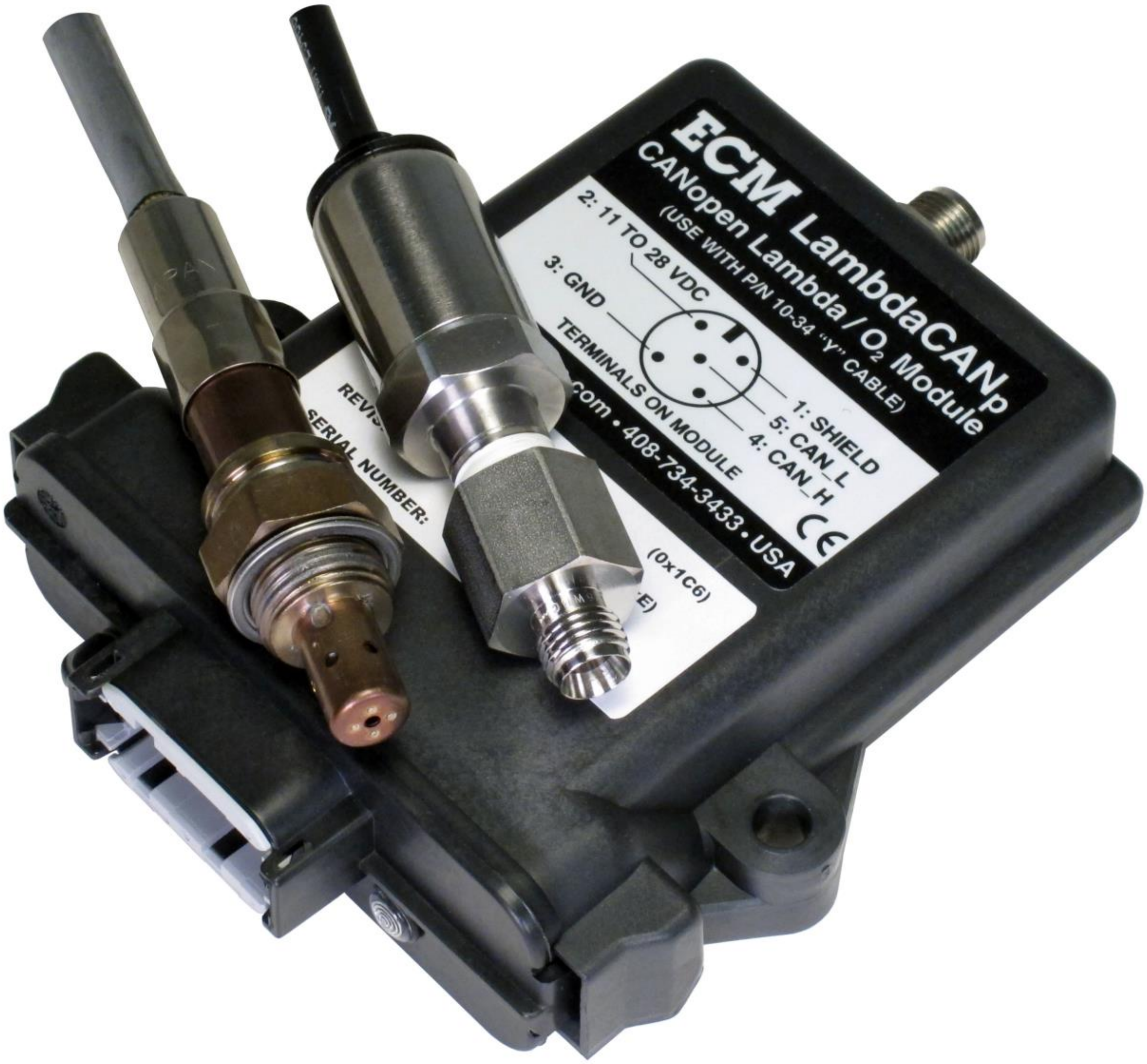
REVISION	DATE	DESCIPTION	SECTION
1.00	7/1/2011	Original Release	
1.20	2/2/2012	"ERFL" and "ERCD" changed to "UERF" and"UERC".	D
		Added new ECM OS Commands.	B
1.30	2/16/2012	Added CAN baudrate setting, Delta tables, and Pressure sensor calibration. Modified appendix G	8.12 - 8.17
1.31	9/11/2014	Added error code 0x63	A

LambdaCANpmanual 1.31.doc

ECM
586 Weddell Dr., Suite 2
Sunnyvale, CA 94089 USA

Tel: 408-734-3433
Fax: 408-734-3432
www.ecm-co.com

Contents


ECM LambdaCANp
CANopen Lambda / O2 Module
(USE WITH P/N 10-34 "Y" CABLE)
2: 11 TO 28 VDC
3: GND
1: SHIELD
5: CAN_L
4: CAN_H
TERMINALS ON MODULE
.com • 408-734-3433 • USA
(0x1C6)
SERIAL NUMBER:

1.0 Introduction and General Notes

1.1 Introduction

The LambdaCANp kit is a ceramic sensor-based Lambda, AFR, O_2, and pressure (optional) 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 LambdaCANp Kit consists of:

PART		P/N	QTY
1. LambdaCANp Control Module		02-08	1
2. LambdaCANp Sensor	(Kit P/N: LCAN-N)	05-01 (NTK 6mA) or	1
	(Kit P/N: LCAN-B2)	05-02 (BOSCH LSU4.2) or	
	(Kit P/N: LCAN-B9)	05-03 (BOSCH LSU4.9) or	
	(Kit P/N: LCAN-N4)	05-04 (NTK 4mA) or	
	(Kit P/N: LCAN-D)	05-06 (DELPHI OSL)	
3. Lambda 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. Lambda 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):

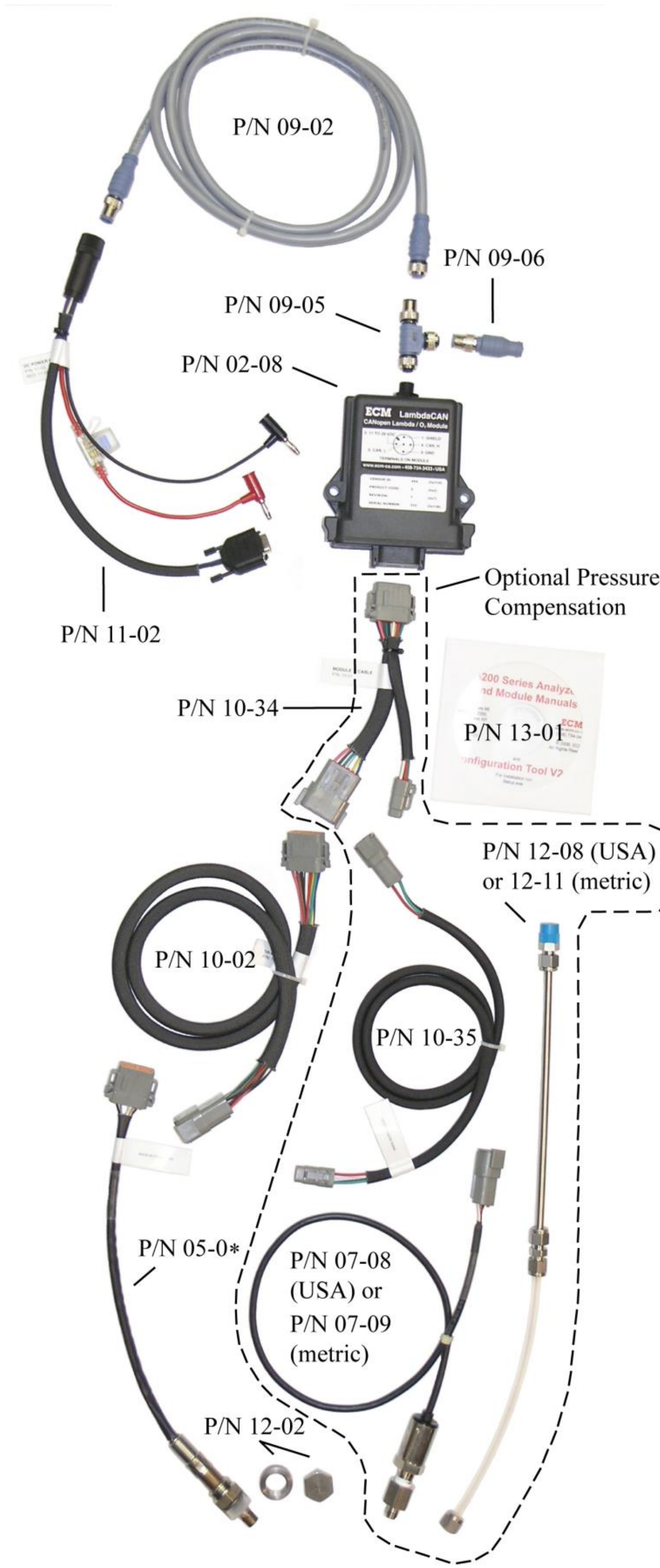
PART	P/N	QTY
10. Pressure Sensor, 0-75 psia, 517 kPa	07-08 (USA) or 07-09 (metric)	1
11. Pressure Extension Cable	10-35 (1m)	1
12. Pressure Sensor Tubing	12-08 (USA) or 12-11 (metric)	1
13. Module Y Cable	10-34	1

Optional Cables:

Lambda Cable	10-02 (1m) or 10-03 (2m)
Pressure Cable	10-35 (1m) or 10-36 (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 02-08
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 05-0*
P/N 07-08 (USA) or P/N 07-09 (metric)
P/N 12-02

3.0 Connecting the LambdaCANp module

Power and CAN connections to the module are made using the Eurofast 12mm connector on the module. The power input requirement is 11 to 30VDC at 12W (steady-state). At start-up, there can be a peak power draw of up to 40W. Multiple modules can be connected together. All modules are configured to broadcast CAN messages at the default CAN baud rate of 500kBits/s (to change the baud rate see section 8.12). The maximum distance between any two nodes on the CAN bus operating at a 500kBit/s baud rate is 100m. Each end of the CAN bus must have a terminating resistor of 121 Ohms.

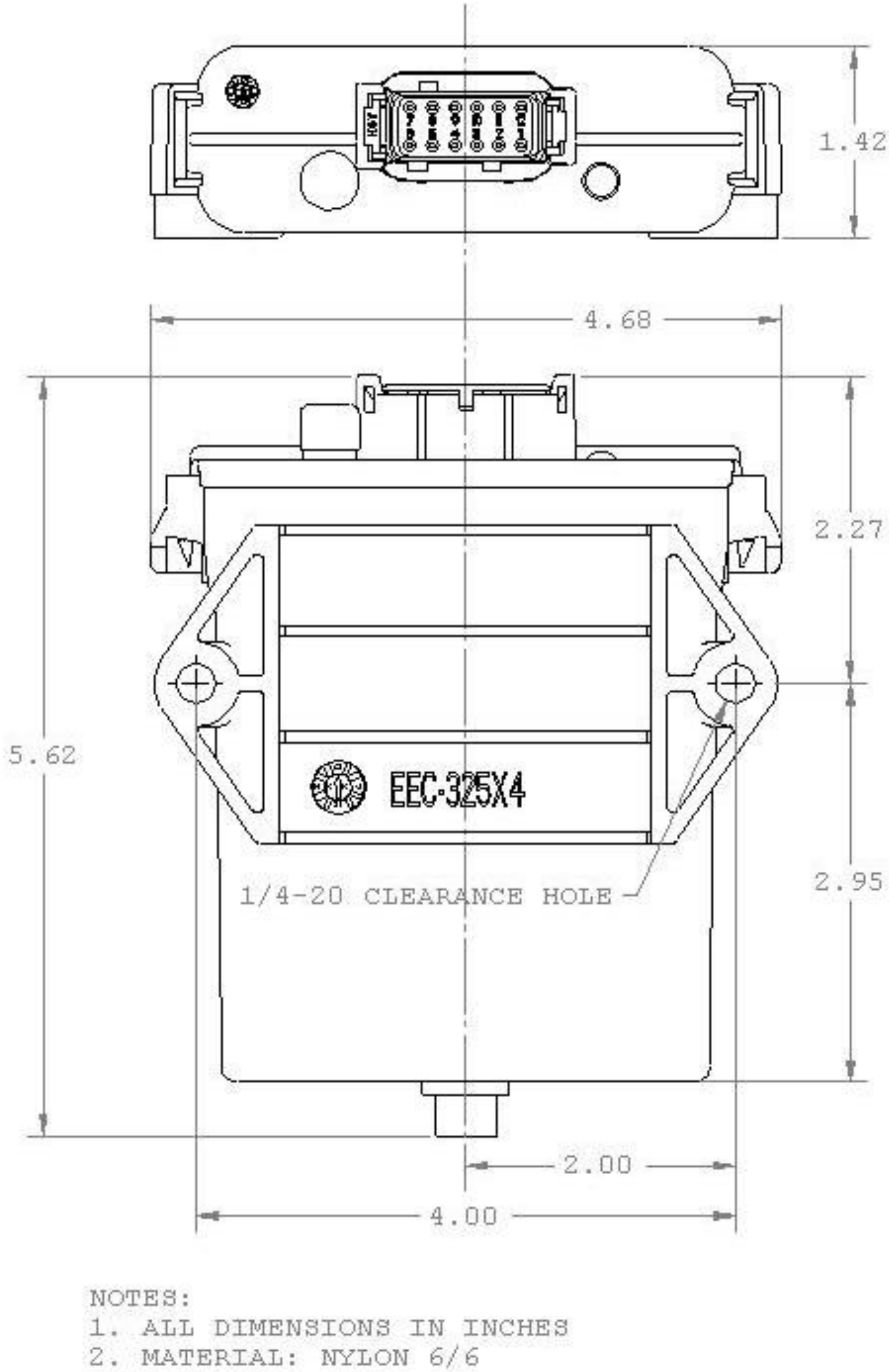
3.1 Application Notes

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 Lambda sensor supplied with LambdaCANp is already calibrated with the calibration residing in a memory chip in the sensor's connector. To compensate for sensor ageing, the sensor can be spanned using ambient air, (it is not necessary to zero the sensor). See section 8.1 for instructions on how to perform a span of the O_2 measurement.
3. Do not extend the wires between the Lambda 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 11 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.
4. The LambdaCANp 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)

3.2 Overall Dimensions



4.0 Getting Information from the LambdaCANp Module

As soon as power is attached to the LambdaCANp 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 lambda sensor has reached its operating temperature. This may take up to 30 seconds. Once the sensor is ready and the Lambda and O_2 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 the programmed CAN baud rate (see section 8.12). All CAN messages have an identifier (CANid) that is related to the Node ID (NID) of the particular module. As shipped, the Node ID 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=8)

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

L byte = Lambda sensor ECM Error Code (0x01 = Sensor Warm-up, 0x00 = Data valid etc, see Appendix A)
count = Sensor Warm-up countdown in seconds (active during ECM Error Code 0x01)
P byte = Pressure sensor ECM Error Code (0x00 = Data valid etc, see Appendix A)

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	Lambda (AFR/AFRstoic)				O_2 (%)			
TPDO2 CAN id	byte 0	byte 1	byte 2	byte 3	byte 4	byte 5	byte 6	byte 7
0x280+NID	AFR disabled				FAR disabled			
TPDO3 CAN id	byte 0	byte 1	byte 2	byte 3	byte 4	byte 5	byte 6	byte 7
0x380+NID	Pressure (mmHg) disabled				PHI disabled			
TPDO4 CAN id	byte 0	byte 1	byte 2	byte 3	byte 4	byte 5	byte 6	byte 7
0x480+NID	RPVS (ohms) disabled				VHCM (V) disabled			

The table above shows the default TPDO assignments. Note that only Lambda and O_2 are enabled (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 corresponds to a measured parameter (e.g. Lambda, AFR, O_2, FAR, PHI, etc). 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, Lambda and O_2.

TPDO1 CAN id	byte 0	byte 1	byte 2	byte 3	byte 4	byte 5	byte 6	byte 7
0x190	0x63	0xC6	0x99	0x3F	0xF2	0xFD	0x54	0x40

Lambda = 0x3F99C663 = 1.20137
O_2 = 0x4054FDF2 = 3.32800

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

5.0 Writing to the LambdaCANp Module (SDO Write)

Configuration of the LambdaCANp 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 LambdaCANp 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 LambdaCANp Module

Each LambdaCANp 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 (LambdaCANp = 0x0000000E) 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 LambdaCANp Module

There are several commands that can be used with the LambdaCANp 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 SPANing O_2

All calibration information, including sensor constants and the user-performed span, is stored in a memory chip located in the connector of the lambda sensor. The Lambda and O_2 values do not depend on what module the sensor is plugged into, they depend on the factory calibration and user span information stored in the sensor's connector that is attached to the sensor.

To compensate for sensor aging, the lambda sensor can be recalibrated using ambient air ("Air-Cal"). This recalibration affects O_2, Lambda, AFR, FAR, and PHI measurements. To recalibrate of the lambda sensor perform the following:

1. Put the lambda sensor and the pressure sensor (if so equipped) in ambient, stationary air for 20 minutes. Pressure during lambda sensor calibration is required if the calibration is to be pressure compensated. Record the O_2 value reported by the module.
2. Note that calibrating the sensor in the exhaust of an engine which has stopped running is not recommended due to the possibility of residual exhaust gases remaining in the exhaust system.
3. Calculate the $\%O_2$ in air; this is the "true/correct O_2 value". The $\%O_2$ in air with no humidity is 20.945. This percentage decreases with increased humidity. To calculate the $\%O_2$ in non-zero humidity air, refer to Appendix F.
4. Send the following three messages to the LambdaCANp module:

   i) SDO Write to OD address 0x5000 of the O_2 value reported by the module.
   ii) SDO Write to OD address 0x5001 the true/correct O_2 value.
   iii) OS Command to SPAN O_2

The first message contains the O_2 value reported by the LambdaCANp. 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 LambdaCANp (float)			

The second message contains the true value of O_2.

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 SPAN O_2.

CANid	byte 0	byte 1	byte 2	byte 3	byte 4	byte 5	byte 6	byte 7
0x600+NID	0x2F	0x23	0x10	0x01	0x0E			

Example: A module with NID = 0x02 has a lambda sensor which is currently transmitting a reading 19.5% O_2 (0x419C0000). The desired or true value is 20.95% (0x41A7999A). The following messages are sent to the module to 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 SPAN

The following command, when sent to the LambdaCANp module, erases the user O_2 SPAN information stored in the sensor's connector. Thus, the lambda 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	0x11			

8.3 Modifying the Pre-Broadcast Averaging of Data

The Lambda sensor output (Ip1) and pressure (P) 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 pressure and one α for Ip1. The defaults are 0.375 and are stored in non-volatile memory (EEprom) in the module. Note that O_2, Lambda, PHI, AFR and FAR are calculated from the Ip1 measurement and therefore are affected by the Ip1 averaging 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 Ip1
= 0x09 for Pressure (P)

α x 1000 = 1-1000 (0x0001 – 0x03E8)

Example: Set the α for Ip1 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 two averaging filters (alpha IP1 and alpha P) can be reset to factory default (0.375) 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 LambdaCANp module. This must be followed by a reset of the module.

First 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). Note that the data is sent LSB first. For example, if the serial number is 0x12345678, then byte1 = 0x78, byte2 = 0x56, byte3 = 0x34, byte 4 = 0x12.

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

If successful, the module will respond with the following message.

CAN id	byte 0
0x7E4	0x44

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	= 0x02 (2)
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	0x02	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 LambdaCANp 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 1^{st} PDO.
  (see Appendix D PROCESS DATA OBJECTS)
- iii) Enter the OD address of the 2^{nd} 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 AFR to TPDO2 for the module with NID = 0x02. (Pressure PDO OD Address = 0x2016, AFR PDO OD Address = 0x2018, 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	0x18	0x20
0x602	0x2F	0x01	0x1A	0x00	0x02			

8.10 Entering Fuel Constants

The fuel constants are entered as single precision floating point numbers (IEEE-754) into the module by performing a SDO Write to the following addresses.

	OD Address	Factory Default
H:C	0x500B	1.850
O:C	0x500C	0.0
N:C	0x500D	0.0

Example: Enter an H:C ratio (OD address 0x500B) of 1.9 (0x3FF33333) for the module with NID = 0x10.

CANid	byte 0	byte 1	byte 2	byte 3	byte 4	byte 5	byte 6	byte 7
0x610	0x23	0x0B	0x50	0x00	0x33	0x33	0xF3	0x3F

The module and sensor are compatible with hydrogen fuels. To configure the module for hydrogen fuel requires sending the ECM CANopen OS Command 0x19 (EnableH2Calc) via an SDO Write. The H:C, O:C, N:C constants are ignored until the hydrogen fuel mode is disabled.

Example: Enable hydrogen fuel compatibility for the module with NID = 0x03:

CANid	byte 0	byte 1	byte 2	byte 3	byte 4	byte 5	byte 6	byte 7
0x603	0x2F	0x23	0x10	0x01	0x19			

Example: Disable hydrogen fuel compatibility for the module with NID = 0x03:

CANid	byte 0	byte 1	byte 2	byte 3	byte 4	byte 5	byte 6	byte 7
0x603	0x2F	0x23	0x10	0x01	0x1A			

8.11 Factory Reset

Parameters that are stored in non-volatile memory (EEprom) can be reset to a standard configuration by issuing the ECM OS Command 0xDF (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	0xDF			

Issuing this command sets configuration and module parameters as follows:

1. Lambda sensor enabled
2. Lambda sensor 1wire memory enabled
3. Pressure sensor 1wire memory enabled
4. Expert mode disabled

5. Hydrogen fuel compatibility disabled
6. Lambda sensor Ip1 pressure compensation enabled
7. Fast Lambda sensor start enabled
8. Pre-broadcast averaging values reset to factory default (see section 8.3)
9. Delta O_2 and Delta Lambda tables cleared
10. TPDOs are reset to factory default (see section 4.1.3)
11. Fuel constants reset to factory defaults (see section 8.10)
12. TPDO Broadcast rate set to 5ms (see section 8.6)

Note that the following are **NOT** affected by a factory reset:

i) O_2 SPAN. To set the O_2 SPAN see section 8.2
ii) CAN baudrate. To set the CAN baudrate see section 8.12
iii) Node id (NID). To set the NID see section 8.5

8.12 Setting the CAN baudrate

To change the CAN baudrate, several messages must be sent to the LambdaCANp module. This must be followed by a reset of the module. The procedure described below will change the CAN baudrate on ALL modules. If there are multiple modules on the CAN bus during this procedure, all the modules must be communicating at the same baudrate.

First 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). Note that the data is sent LSB first. For example, if the serial number is 0x12345678, then byte1 = 0x78, byte2 = 0x56, byte3 = 0x34, byte 4 = 0x12.

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

If successful, the module will respond with the following message.

CAN id	byte 0
0x7E4	0x44

Send the command to configure the baud rate. This will apply to ALL modules that are set to LSS configuration mode.

CAN id	byte 0	byte 1	byte 2
0x7E5	0x13	0x00	Index

Where “Index” corresponds to the following baud rates:
0 = 1000kBbit/s
1 = 800kBbit/s (Not supported. Do not use.)
2 = 500kBbit/s (Max bus length 100m)
3 = 250kBbit/s (Max bus length 250m)
4 = 125kBbit/s (Max bus length 500m)
5 = Reserved
6 = 50kBbit/s
7 = 20kBbit/s
8 = 10kBbit/s

If successful, the module will respond with the following message:

CAN id	byte 0	byte 1	byte 2	byte 3	byte 4	byte 5	byte 6	byte 7
0x7E4	0x13	Error						

If Error = 0, then the baud rate configuration is successful. Otherwise, it has failed.

Send the command to activate the new baud rate.

CAN id	byte 0	byte 1	byte 2	byte 3	byte 4	byte 5	byte 6	byte 7
0x7E5	0x15	Delay (lsb first)						

Where delay is the millisecond time delay before the new baud rate is active. This insures that CAN messages stop transmitting on one baud rate before the new baud rate is active. This value is also sent LSB first. No reply message expected.

At this point, the new baud rate is active, and you will no longer be able to communicate on the old baud rate. Make sure you set your software to communicate on the new baud rate.

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

NEW BAUDRATE = 250kBits/s
DELAY = 2000mSec (0x07D0)

MESSAGE SENT

CAN id	byte 0	byte 1	byte 2
0x00	0x80	0x10	
0x7E5	0x40	0x01	
0x7E5	0x13	0x00	0x03
0x7E5	0x15	0xD0	0x07

MODULE REPLY

CAN id	byte 0	byte 1
0x7E4	0x44	
0x7E4	0x13	0x00

8.13 Modifying the Delta Lambda table

The Lambda measurement can be modified to match a user's desired calibration. A Delta Lambda table is stored in non-volatile memory (EEPROM) in the module and can be written and read by the user performing a SDO READ or SDO WRITE (see sections 5.0 and 6.0). The Lambda measurement is modified as follows:

$$LAM = LAMR + \Delta$$

Where:
LAM = users desired Lambda reading (PDO OD Address = 0x201B)
LAMR = the Lambda value calculated by the module (PDO OD Address = 0x2017)
Δ = is a positive or negative Lambda number added to LAMR

Up to 31 entries are available in the Delta Lambda table. The table entries must be monotonic and if fewer than 31 entries are used, the table must be appended with an end of table character (0x7FFF). The LAMR index table is located at ECM OD location 0x5015 and is formatted as signed 16bit integers and scaled by 1000 as shown below:

ADDRESS	SUBINDEX	SIZE	READ/WRITE	SCALING
0x5015	0x00	signed 16	RD/WR	Lambda * 1000
	0x01	signed 16	RD/WR	Lambda * 1000
	0x02	signed 16	RD/WR	Lambda * 1000
	…			
	0x1F	signed 16	RD/WR	Lambda * 1000

The corresponding Δvalue table is located at ECM OD location 0x5016 and is formatted as signed 16bit integers and scaled by 1000 as shown below.

ADDRESS	SUBINDEX	SIZE	READ/WRITE	SCALING
0x5016	0x00	signed 16	RD/WR	Δ * 1000
	0x01	signed 16	RD/WR	Δ * 1000
	0x02	signed 16	RD/WR	Δ * 1000
	…			
	0x1F	signed 16	RD/WR	Δ * 1000

Note that there must be a corresponding Δ entry for every Lambda index (LAMR). The Δ is linearly interpolated for LAMR values that lie within the table and is limited to the closest Δ value for LAMR values that fall outside the table. LAM is used in the calculations for AFR, FAR and PHI.

Example: Modify the Delta Lambda table so that a module with NID = 0x10 transmits LAM = 0.8 lambda when LAMR = 0.7, LAM = 1.0 lambda when LAMR = 1.03 and LAM = 1.5 lambda when LAMR = 1.5.

The Delta Lambda table using the formula Δ = Desired Lambda – LAMR is calculated below:

SUBINDEX	0x5016 (LAMR)	0x5017 (Δ)
0x00	0.7 * 1000 = 700 (0x02BC)	0.1 * 1000 = 100 (0x0064)
0x01	1.03 * 1000 = 1030 (0x0406)	-0.03 * 1000 = -30 (0xFFE2)
0x02	1.5 * 1000 = 1500 (0x05DC)	0 * 1000 = 0 (0x0000)
0x03	0x7FFF	0x7FFF

The CAN messages to write the Delta Lambda table in the module are as follows (see sections 5.0 and 6.0). Note that the data bytes are loaded least significant byte first (Intel format):

CAN id	byte 0	byte 1	byte 2	byte 3	byte 4	byte 5	byte 6	byte 7
0x610	0x2B	0x16	0x50	0x00	0xBC	0x02	0x00	0x00
0x610	0x2B	0x16	0x50	0x01	0x04	0x06	0x00	0x00
0x610	0x2B	0x16	0x50	0x02	0xDC	0x05	0x00	0x00
0x610	0x2B	0x16	0x50	0x03	0xFF	0x7F	0x00	0x00
0x610	0x2B	0x17	0x50	0x00	0x64	0x00	0x00	0x00
0x610	0x2B	0x17	0x50	0x01	0xE2	0xFF	0x00	0x00
0x610	0x2B	0x17	0x50	0x02	0x00	0x00	0x00	0x00
0x610	0x2B	0x17	0x50	0x03	0xFF	0x7F	0x00	0x00

8.14 Modifying the Delta O2 table

The $\%O_2$ measurement can be modified to match a user's desired calibration. A Delta O2 table is stored in non-volatile memory (EEPROM) in the module and can be written and read by the user performing a SDO READ or SDO WRITE (see sections 5.0 and 6.0). The $\%O_2$ measurement is modified as follows:

$$O2 = O2R + \Delta$$

Where:
O2 = users desired $\%O_2$ reading (PDO OD Address = 0x201C)
O2R = the $\%O_2$ value calculated by the module (PDO OD Address = 0x2001)
Δ = is a positive or negative $\%O_2$ number added to O2R

Up to 31 entries are available in the Delta O2 table. The table entries must be monotonic and if fewer than 31 entries are used, the table must be appended with an end of table character (0x7FFF). The O2R index table is located at ECM OD location 0x5013 and is formatted as signed 16bit integers and scaled by 100 as shown below:

ADDRESS	SUBINDEX	SIZE	READ/WRITE	SCALING
0x5013	0x00	signed 16	RD/WR	$\%O_2$ * 100
	0x01	signed 16	RD/WR	$\%O_2$ * 100
	0x02	signed 16	RD/WR	$\%O_2$ * 100
	…			
	0x1F	signed 16	RD/WR	$\%O_2$ * 100

The corresponding Δvalue table is located at ECM OD location 0x5014 and is formatted as signed 16bit integers and scaled by 1000 as shown below.

ADDRESS	SUBINDEX	SIZE	READ/WRITE	SCALING
0x5016	0x00	signed 16	RD/WR	Δ * 1000
	0x01	signed 16	RD/WR	Δ * 1000
	0x02	signed 16	RD/WR	Δ * 1000
	…			
	0x1F	signed 16	RD/WR	Δ * 1000

Note that there must be a corresponding Δ entry for every $\%O_2$ index (O2R). The Δ is linearly interpolated for O2R values that lie within the table and is limited to the closest Δ value for O2R values that fall outside the table.

Example: Modify the Delta O2 table so that a module with NID = 0x10 transmits O2 = -10 $\%O_2$ when O2R = -9.0, O2 = 0.3 $\%O_2$ when O2R = 0.

The Delta Lambda table using the formula $\Delta =$ Desired $\%O_2$ – O2R is calculated below:

SUBINDEX	0x5013 (O2R)	0x5014 (Δ)
0x00	-9.00 * 100 = -900 (0xFC7C)	-1.0 * 1000 = -1000 (0xFC18)
0x01	0 * 100 = 0 (0x0000)	0.3 * 1000 = 300 (0x012C)
0x03	0x7FFF	0x7FFF

The CAN messages to write the Delta O2 table in the module are as follows (see sections 5.0 and 6.0). Note that the data bytes are loaded least significant byte first (Intel format):

CAN id	byte 0	byte 1	byte 2	byte 3	byte 4	byte 5	byte 6	byte 7
0x610	0x2B	0x13	0x50	0x00	0x7C	0xFC	0x00	0x00
0x610	0x2B	0x13	0x50	0x01	0x00	0x00	0x00	0x00
0x610	0x2B	0x13	0x50	0x02	0xFF	0x7F	0x00	0x00
0x610	0x2B	0x14	0x50	0x00	0x18	0xFC	0x00	0x00
0x610	0x2B	0x14	0x50	0x01	0x2C	0x01	0x00	0x00
0x610	0x2B	0x14	0x50	0x02	0xFF	0x7F	0x00	0x00

8.15 Clearing the Delta Lambda and Delta O2 table

The Delta Lambda table can be “cleared” by writing 0x7FFF to OD location 0x5015 subindex 0 so that LAM = LAMR always. Also the Delta O2 table can be “cleared” by writing 0x7FFF to OD location 0x5013 subindex 0 so that O2 = O2R always.

Example: Clear the Delta Lambda table so that LAM = LAMR for a module with NID = 0x20.

CANid	byte 0	byte 1	byte 2	byte 3	byte 4	byte 5	byte 6	byte 7
0x620	0x2B	0x15	0x50	0x00	0xFF	0x7F	0x00	0x00

8.16 Modifying the Pressure Sensor Calibration

All calibration information, including factory pressure sensor calibration and user-performed calibration, are stored in a memory chip located in the connector of the pressure sensor. The reported pressure values do not depend on what module the pressure sensor is plugged into, they depend on the factory calibration and user calibration information stored in the sensor’s connector that is attached to the sensor.

To compensate for pressure sensor aging and/or drift, the user can recalibrate the pressure sensor and write these calibration values to a user specific location in the memory chip located in the connector of the pressure sensor. This recalibration affects the following PDOs: P, PKPA, PBAR, and PPSI (see Appendix D).

To recalibrate of the pressure sensor perform the following steps:

1. Install the pressure sensor in the calibration apparatus. Connect the pressure sensor to the LambdaCANp module using the cable p/n 10-34. Apply power to the LambdaCANp module. Allow the pressure sensor and module to warm-up for 10 minutes. For each pressure setting record the PVLT voltage value reported by the module.
2. Turn the sensor power off, issue ECM OS Command 0x08 (SensorOff, see Appendix B). Write the PVLT voltage values and pressure values to OD location 0x5059 subindices 0x50 to 0x6F (do not use subindex 0x5F). A maximum of 15 pairs of data are allowed. The PVLT voltage values are in volts and scaled by a factor of 1000. The pressure values are in mmHg and scaled by a factor of 10. The pressure is linearly interpolated for PVLT values that lie within the table and is limited to the closest pressure value for PVLT values that fall outside the table.
3. If fewer than 15 data pairs are required, append 0xFFFF as the last element of the table.
4. Update the user CAL YEAR by writing to OD location 0x5059 subindex 0x70 and the user CAL DATE by writing to OD location 0x5059 subindex 0x71.
5. Write the data to the pressure sensor memory by issuing ECM OS Command 0x5B and read the OS Status and OS Reply (ProgUserPcal, see Appendix B).
6. Turn the sensor power on, issue ECM OS Command 0x07 (SensorOn, see Appendix B).

The OD location 0x5059 is formatted as shown below. Note the scaling and the units of each entry. Also note that subinex 0x5F is reserved. Do not write to this location.

ADDRESS	SUBINDEX	SIZE	READ/WRITE	SCALING
0x5059	0x50	unsigned 16	RD/WR	PVLT 1 (Volts * 10000)
	0x51	unsigned 16	RD/WR	Pressure 1 (mmHg * 10)
	0x52	unsigned 16	RD/WR	PVLT 2 (Volts * 10000)
	0x53	unsigned 16	RD/WR	Pressure 2 (mmHg * 10)
	0x54	unsigned 16	RD/WR	PVLT 3 (Volts * 10000)
	0x55	unsigned 16	RD/WR	Pressure 3 (mmHg * 10)
	0x56	unsigned 16	RD/WR	PVLT 4 (Volts * 10000)
	0x57	unsigned 16	RD/WR	Pressure 4 (mmHg * 10)
	0x58	unsigned 16	RD/WR	PVLT 5 (Volts * 10000)
	0x59	unsigned 16	RD/WR	Pressure 5 (mmHg * 10)
	0x5A	unsigned 16	RD/WR	PVLT 6 (Volts * 10000)
	0x5B	unsigned 16	RD/WR	Pressure 6 (mmHg * 10)
	0x5C	unsigned 16	RD/WR	PVLT 7 (Volts * 10000)
	0x5D	unsigned 16	RD/WR	Pressure 7 (mmHg * 10)
	0x5E	unsigned 16	RD/WR	PVLT 8 (Volts * 10000)
	0x5F	unsigned 16	RD/WR	**Reserved**
	0x60	unsigned 16	RD/WR	Pressure 8 (mmHg * 10)
	0x61	unsigned 16	RD/WR	PVLT 9 (Volts * 10000)
	0x62	unsigned 16	RD/WR	Pressure 9 (mmHg * 10)
	0x63	unsigned 16	RD/WR	PVLT 10 (Volts * 10000)
	0x64	unsigned 16	RD/WR	Pressure 10 (mmHg * 10)
	0x65	unsigned 16	RD/WR	PVLT 11 (Volts * 10000)
	0x66	unsigned 16	RD/WR	Pressure 11 (mmHg * 10)
	0x67	unsigned 16	RD/WR	PVLT 12 (Volts * 10000)
	0x68	unsigned 16	RD/WR	Pressure 12 (mmHg * 10)
	0x69	unsigned 16	RD/WR	PVLT 13 (Volts * 10000)
	0x6A	unsigned 16	RD/WR	Pressure 13 (mmHg * 10)
	0x6B	unsigned 16	RD/WR	PVLT 14 (Volts * 10000)
	0x6C	unsigned 16	RD/WR	Pressure 14 (mmHg * 10)
	0x6D	unsigned 16	RD/WR	PVLT 15 (Volts * 10000)
	0x6E	unsigned 16	RD/WR	Pressure 15 (mmHg * 10)
	0x6F	unsigned 16	RD/WR	**Reserved**
	0x70	unsigned 16	RD/WR	CAL YEAR (year * 1)
	0x71	unsigned 16	RD/WR	CAL DATE (lower byte = day, upper byte = month)

Example: Modify the user pressure data using the data below for a module with NID = 0x10.

PVLT	Pressure
1.0V * 10000 = 10000 (0x2710)	500mmHg * 10 = 5000 (0x1388)
2.5V * 10000 = 25000 (0x61A8)	1000mmHg * 10 = 10000 (0x2710)
4.0V * 10000 = 40000 (0x9C40)	2000mmHg * 10 = 20000 (0x4E20)
0xFFFF	0xFFFF

The CAN messages to write the PVLT vs Pressure table in the module are as follows (see sections 5.0 and 6.0). Note that the data bytes are loaded least significant byte first (Intel format):

CAN id	byte 0	byte 1	byte 2	byte 3	byte 4	byte 5	byte 6	byte 7
0x610	0x2B	0x59	0x50	0x50	0x10	0x27	0x00	0x00
0x610	0x2B	0x59	0x50	0x51	0x88	0x13	0x00	0x00
0x610	0x2B	0x59	0x50	0x52	0xA8	0x61	0x00	0x00
0x610	0x2B	0x59	0x50	0x53	0x10	0x27	0x00	0x00
0x610	0x2B	0x59	0x50	0x54	0x40	0x9C	0x00	0x00
0x610	0x2B	0x59	0x50	0x55	0x20	0x4E	0x00	0x00
0x610	0x2B	0x59	0x50	0x56	0xFF	0xFF	0x00	0x00
0x610	0x2B	0x59	0x50	0x57	0xFF	0xFF	0x00	0x00

8.17 Restoring the Pressure Sensor Calibration to Factory Default

To revert back to the factory pressure sensor calibration issue ECM OS Command 0x59. Connect the pressure sensor to the module using cable p/n 10-34. Apply power to the module and issue the ECM OS Command as shown below.

CANid	byte 0	byte 1	byte 2	byte 3	byte 4	byte 5	byte 6	byte 7
0x600+NID	0x2F	0x23	0x10	0x01	0x59			

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	Lambda Error lo	Lambda Error hi	ECM Aux	Pressure Error lo	Pressure Error hi

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

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	IP1 out of range (>12.5mA or <-12.5mA)
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)

Appendix B: ECM CANopen OS Commands

A user-specific CANopen OS Command to the LambdaCANp module is sent using an SDO expedited write message in the following form. These commands apply only to the LambdaCANp 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
SpanO2	0x0E	Set span for O2, requires write to OD 0x5000 (old value) and 0x5001 (desired value)	
		defZeroSpanSuccessful	0x00
		defSpanInvalidNegativeSlope	0xFB
		defSpanTooCloseToOffset	0xFC
		defSenModNotReady	0xFD
		defZeroSpanDataInvalid	0xFE
		defOWZeroSpanWrFail	0xFF
ResetO2	0x11	Resets the span for %O2 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 %O2 to factory value (alpha=48)	
		defAlphaOK	0x00
ExpertModeDisable	0x16	This command removes the unit from expert mode	None
EnableH2Calc	0x19	Use H2 formula for lambda calculation	None
DisableH2Calc	0x1A	Use std formula for lambda calculation	None
EnableIP1Pcomp	0x1B	Compensate IP for Pressure	None
DisableIP1Pcomp	0x1C	Do not compensate IP for Pressure (PCF=1.0)	None

COMMAND	VALUE	DESCRIPTION	REPLY
ResetDeltaO2Table	0x1D	Set all delta %O2 values to 0	None
ResetDeltaLambdaTable	0x1E	Set all delta Lambda values to 0	None
ResetTPDOs	0x1F	Set all TPDOs as delivered from ECM	None
FastSensorStart	0x20	Use sensor start parameters in 1wire memory	None
SlowSensorStart	0x21	Sensor start draws <1A max	None
DisableTPDOCOBreset	0x22	TPDO CANids set by user	None
EnableTPDOCOBreset	0x23	TPDO CANids default to CANopen standard	None
Enable1WPress	0x52	Use pressure data in sensor connector	None
Disable1WPress	0x53	Use N and C stored in module EEprom	None
FactRstPcal	0x59	Factory reset the Pressure cal table stored in sensor OW memory	
		defOWWriteOK	0x00
		defOWWriteFail	0xFF
ForcePOWRead	0x5A	Force the module to read the pressure 1wire memory (OWPEnabled)	
		defOWReadSuccessfully	0x00
		defOWInvalidSenType	0xFD
		defOWZeroSpanDataCRCFail	0xFE
		defOWReadError	0xFF
ProgUserPcal	0x5B	Program the user calibration in the pressure sensor memory	
		defOWWriteOK	0x00
		defOWWriteFail	0xFF
FactoryReset	0xDF	Set all EE values to a standard configuration.	None

Appendix C: Heartbeat

A Heartbeat message is transmitted every 0.5 seconds by the LambaCAN 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							

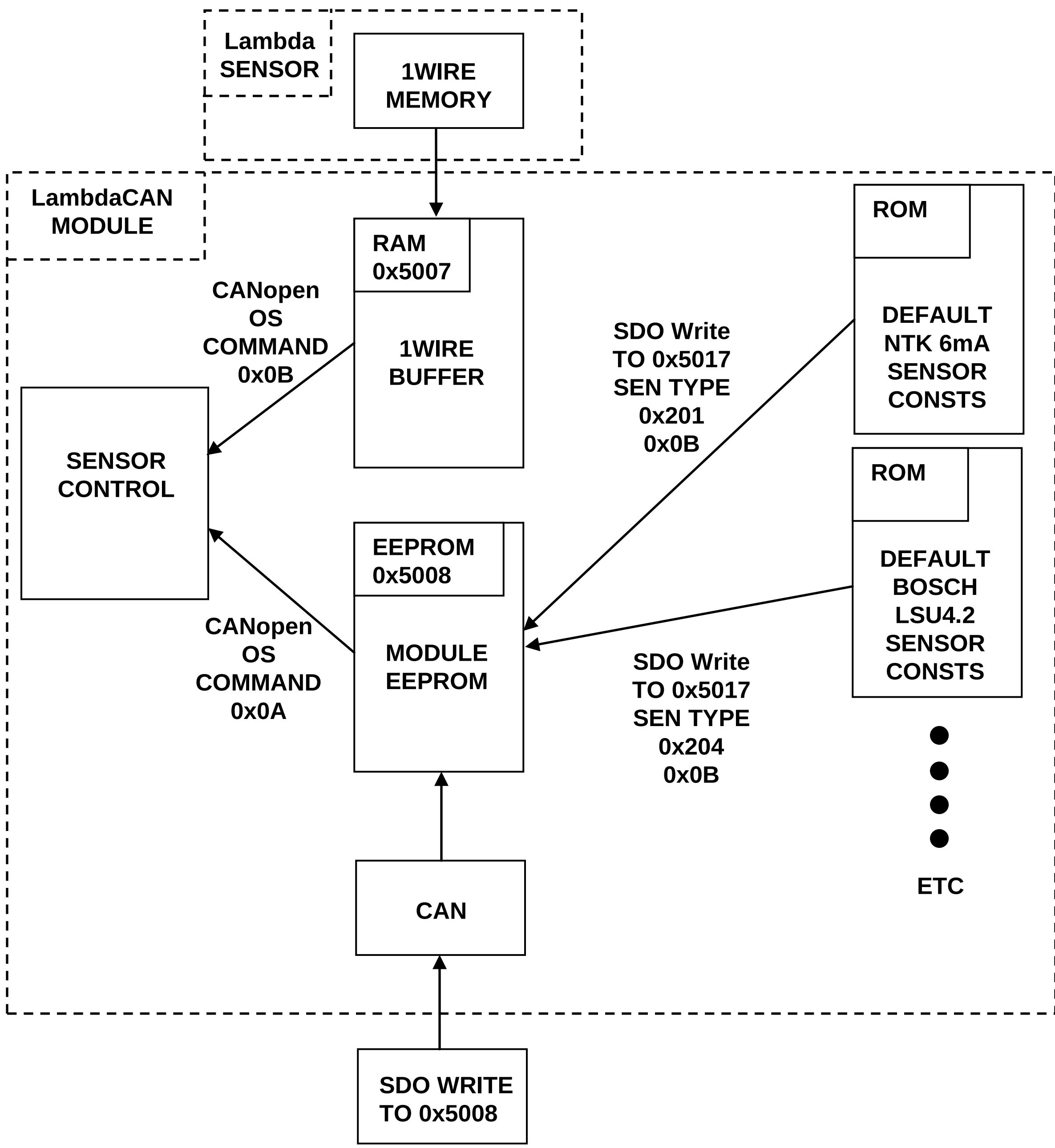
NMT state	
0x00	Boot-up
0x04	Stopped
0x05	Operational
0x7F	Pre-operational

Appendix D: Process Data Objects

Address	**Type**	**Symbol**	**Description**
0x2000	single float		
0x2001	single float	O2R	PDO Data O2real (%)
0x2002	single float	IP1	PDO Data Ip1 (A)
0x2003	single float		
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
0x2007	single float	VP1P	PDO Data VP+ (V) * 1000
0x2008	single float		
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	IP1R	PDO Data Ip1raw (bits)
0x200D	single float	PR16	PDO Data 16bit Praw (bits)
0x200E	single float	UERF	PDO Data Lambda Error bit flags
0x200F	single float	UERC	PDO Data ECM CANOpen Lambda Error Code
0x2010	single float	PR10	PDO Data 10bit Praw (bits)
0x2011	single float	PCF	PDO Data PCF (Pressure Correction Factor) * 10000
0x2012	single float		
0x2013	single float		
0x2014	single float		
0x2015	single float		
0x2016	single float	P	PDO Data Pressure (mmHg)
0x2017	single float	LAMR	PDO Data LAMBDAreal
0x2018	single float	AFR	PDO Data AFR
0x2019	single float	PHI	PDO Data PHI
0x201A	single float	FAR	PDO Data FAR
0x201B	single float	LAM	PDO Data LAMBDA
0x201C	single float	O2	PDO Data O_2 (%)
0x201D	single float	IP1X	PDO Data Non pressure compensated Ip1 (A)
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)
0x2022	single float	PERF	PDO Data Pressure Error bit flags
0x2023	single float	PERC	PDO Data ECM CANOpen Pressure Error Code

Appendix E: Sensor Constants

The Lambda sensor can be controlled using the sensor constants stored in an EEprom memory chip (1WIRE) located in the sensor connector or using a generic set stored in EEprom in the module itself (EEPROM). The factory default configuration automatically loads the sensor constants stored in the sensor memory (1WIRE) every time the sensor is connected to the module or the module is powered on. The module can be configured to use generic sensor constants which can be modified by the user by issuing the proper ECM CANopen OS Command (see Appendix B). The memory locations are organized as shown below:



The procedure for using the generic sensor constants is as follows:

1. Disable the 1WIRE memory in the sensor connector by issuing ECM CANopen OS Command 0x0A.
2. Load the appropriate sensor constants from ROM into EEPROM by writing the sensor type to OD location 0x5017.
3. Modify sensor constants (if necessary) by writing the new value to the appropriate OD location (0x5008).

The OD locations for using the generic sensor constants are as follows:

ADDRESS	**SUBINDEX**	**SIZE**	**READ/WRITE**	**DESCRIPTION**
0x5008	0x00	unsigned 16	RD/WR	SENSOR TYPE
	0x01	unsigned 16	RD/WR	SERIAL NUMBER (0xFFFF)
	0x02	unsigned 16	RD/WR	
	0x03	unsigned 16	RD/WR	
	0x04	unsigned 16	RD/WR	
	0x05	unsigned 16	RD/WR	
	0x06	unsigned 16	RD/WR	
	0x07	unsigned 16	RD/WR	
	0x08	unsigned 16	RD/WR	
	0x09	unsigned 16	RD/WR	
	0x0A	unsigned 16	RD/WR	MAX VH (Volts * 100)
	0x0B	unsigned 16	RD/WR	MIN VH (Volts * 100)
	0x0C	unsigned 16	RD/WR	NOMINAL VH (Volts * 100)
	0x0D	unsigned 16	RD/WR	INCREMENT VH every .3sec (Volts * 100)
	0x0E	unsigned 16	RD/WR	START VH RAMP at (Volts * 100)
	0x0F	unsigned 16	RD/WR	
	0x10	unsigned 16	RD/WR	1-WIRE FORMATTING REVISION
	0x11	unsigned 16	RD/WR	MASK ERROR CODE 0x41, 0x51 for (sec *100)
	0x12	unsigned 16	RD/WR	
	0x13	unsigned 16	RD/WR	
	0x14	unsigned 16	RD/WR	%O2 WHEN SENSOR NEW (% * 1000)
	0x15	unsigned 16	RD/WR	
	0x16	unsigned 16	RD/WR	
	0x17	unsigned 16	RD/WR	
	0x18	unsigned 16	RD/WR	
	0x19	unsigned 16	RD/WR	
	0x1A	unsigned 16	RD/WR	
	0x1B	unsigned 16	RD/WR	
	0x1C	unsigned 16	RD/WR	
	0x1D	unsigned 16	RD/WR	
	0x1E	unsigned 16	RD/WR	
	0x1F	unsigned 16	RD/WR	
	0x20	unsigned 16	RD/WR	O2 SPAN SLOPE (O2M1*65536 + O2M0 = single precision float)
	0x21	unsigned 16	RD/WR	
	0x22	unsigned 16	RD/WR	O2 SPAN Y INTERCEPT (O2B1*65536 + O2B0 = single precision float)
	0x23	unsigned 16	RD/WR	
	0x24	unsigned 16	RD/WR	O2 ZERO/SPAN X INTERCEPT (O2X1*65536 + O2X0 = single precision float)
	0x25	unsigned 16	RD/WR	

	0x26	unsigned 16	RD/WR	
	0x27	unsigned 16	RD/WR	
	0x28	unsigned 16	RD/WR	
	0x29	unsigned 16	RD/WR	
	0x2A	unsigned 16	RD/WR	
	0x2B	unsigned 16	RD/WR	
	0x2C	unsigned 16	RD/WR	
	0x2D	unsigned 16	RD/WR	
	0x2E	unsigned 16	RD/WR	RICH PCF/LEAN PCF (* 1000)
	0x2F	unsigned 16	RD/WR	
	0x30	unsigned 16	RD/WR	RPVS LOW threshold (ohms * 10)
	0x31	unsigned 16	RD/WR	RPVS HIGH threshold (ohms * 10)
	0x32	unsigned 16	RD/WR	RPVS TARGET (ohms * 10)
	0x33	unsigned 16	RD/WR	SAMPLE TIME (sec * 1000)
	0x34	unsigned 16	RD/WR	PI INTEGRATION TIME (sec * 1000)
	0x35	unsigned 16	RD/WR	PROPORTIONAL GAIN Kp (V/ohm * 1000)
	0x36	unsigned 16	RD/WR	INTERGRAL GAIN Ki (V/ohms * 1000)
	0x37	unsigned 16	RD/WR	Ip AIR NEW (mA * 1000)
	0x38	unsigned 16	RD/WR	Ip ZERO NEW (mA * 1000 + 32768)
	0x39	unsigned 16	RD/WR	
	0x3A	unsigned 16	RD/WR	
	0x3B	unsigned 16	RD/WR	HYDROGEN GAIN mA/%H2/mA/%O2 * 10000
	0x3C	unsigned 16	RD/WR	HYDROGEN OFFSET mA/%O2 * 10000 + 32768
	0x3D	unsigned 16	RD/WR	
	0x3E	unsigned 16	RD/WR	
	0x3F	unsigned 16	RD/WR	
0x5017	0x00	unsigned 16	RD/WR	EEPROM SENSOR TYPE A write to this location with sensor type value loads sensor constants for that type into EE memory (EEAuintOW[]). If 1 wire memory is disabled (ECM CANopen OS Cmd 0x0A) module will use these sensor constants for sensor control. = 0x0201 (513) NTK 6mA UEGO = 0x0202 (514) NTK 4mA UEGO = 0x0204 (516) BOSCH LSU4.2 UEGO = 0x0205 (517) BOSCH LSU4.9 UEGO = 0x0206 (518) DELPHI OSL UEGO

Example: Disable 1WIRE sensor memory (ECM CANopen Command 0x0A), load sensor constants for a BOSCH LSU 4.9 UEGO (SDO Write 0x0205 to OD address 0x5017), and modify the RVS TARGET to 150 ohms (SDO Write to 150*10 = 0x05DC to OD address 0x5008, subindex 0x32) for the module with NID = 0x07:

CANid	byte 0	byte 1	byte 2	byte 3	byte 4	byte 5	byte 6	byte 7
0x607	0x2F	0x23	0x10	0x01	0x0A			
0x607	0x2B	0x17	0x50	0x00	0x05	0x02		
0x607	0x2B	0x08	0x50	0x32	0xDC	0x05		

Appendix F: Calculating the %O_2 in Air

The oxygen concentration in dry air (zero humidity) is 20.945 and decreases with increasing humidity. The %O2 in air can be calculated from the barometric pressure (Pb), the relative humidity (Rh), and the saturated water vapor pressure (Pws) by using the following formula:

%O2 = 20.945% x (Pb - Pws x (Rh/100)) / Pb

The saturated water vapor pressure (Pws) is a function of the ambient temperature (Ta) and is given in the table below.

Ta (°C)	0	1	2	3	4	5	6	7	8	9
	Pws (mm Hg)									
0	4.579	4.926	5.294	5.685	6.101	6.543	7.013	7.513	8.045	8.609
10	9.209	9.844	10.518	11.231	11.987	12.788	13.634	14.530	15.477	16.477
20	17.535	18.650	19.827	21.068	22.377	23.756	25.209	26.739	28.349	30.043
30	31.824	33.695	35.663	37.729	39.898	42.175	44.563	47.067	49.692	52.442
40	55.324	58.340	61.500	64.800	68.260	71.880	75.650	79.600	83.710	88.020
50	92.510	97.200	102.09	107.20	112.51	118.04	123.80	129.82	136.08	142.60
60	149.38	156.43	163.77	171.38	179.31	187.54	196.09	204.96	214.17	223.73
70	233.70	243.90	254.60	265.70	277.20	289.10	301.40	314.10	327.30	341.00
80	355.10	369.70	384.90	400.60	416.80	433.60	450.90	468.70	487.10	506.10
90	525.76	546.05	566.99	588.60	610.90	633.90	657.62	682.07	707.27	733.24

Example: At Ta = 19°C, Pws = 16.477mmHg.

Appendix G: Module EIB mode and Stand-alone Mode

The LambdaCANp 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 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.

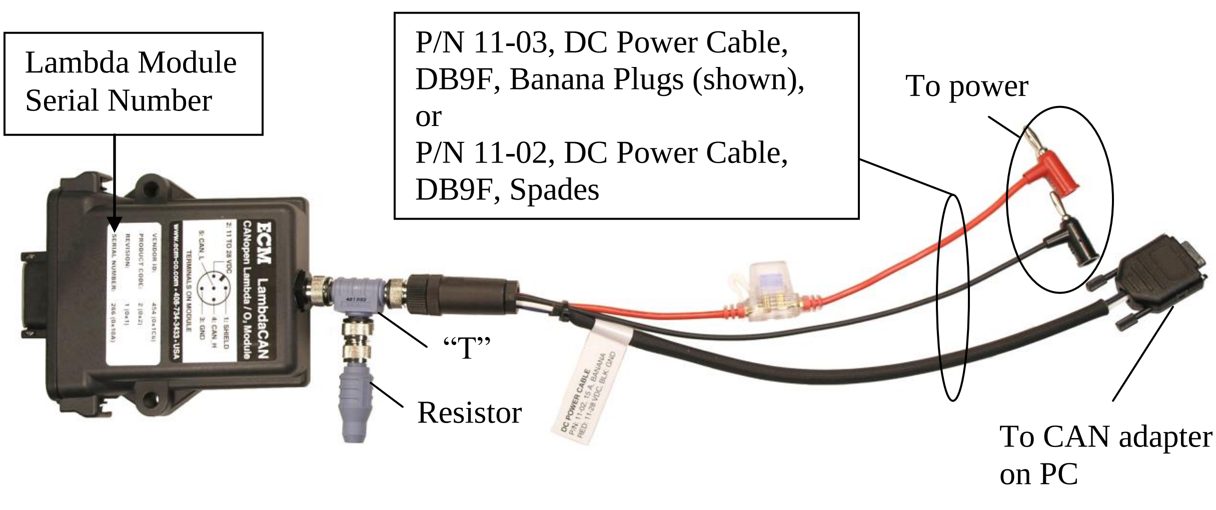
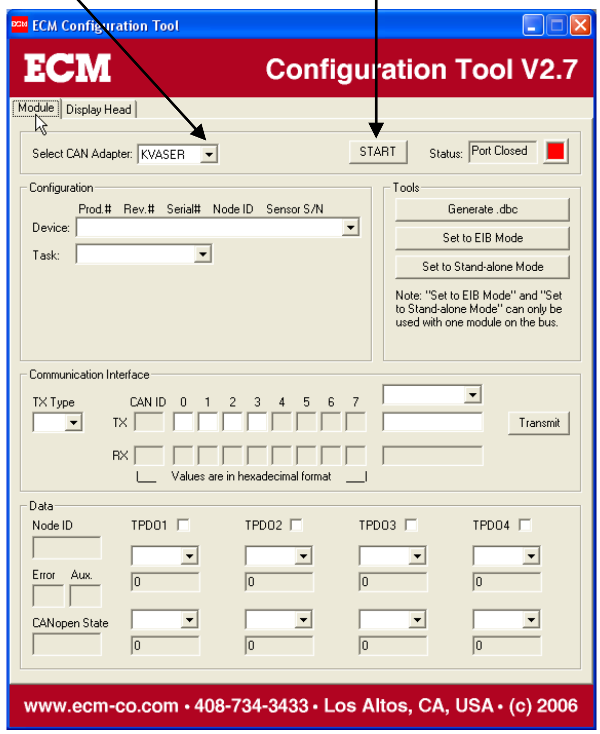
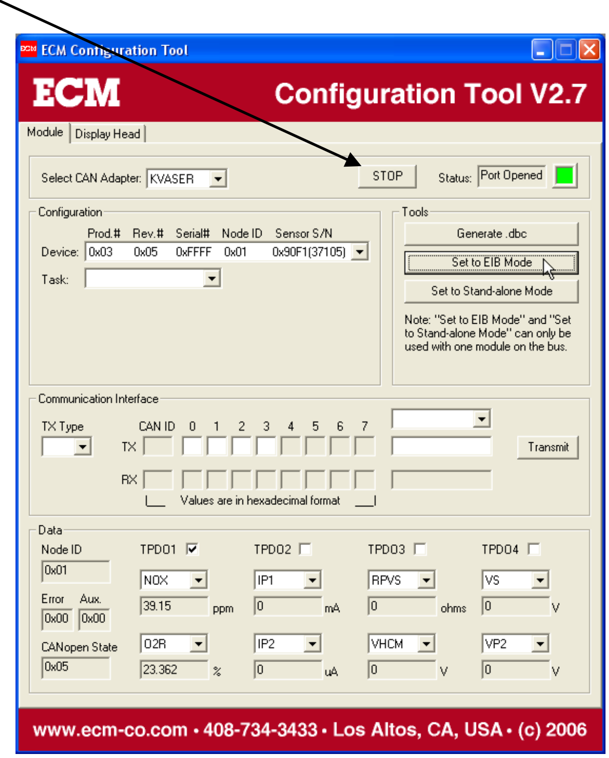


Figure B1: LambdaCAN Module prepared for Reprogramming

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

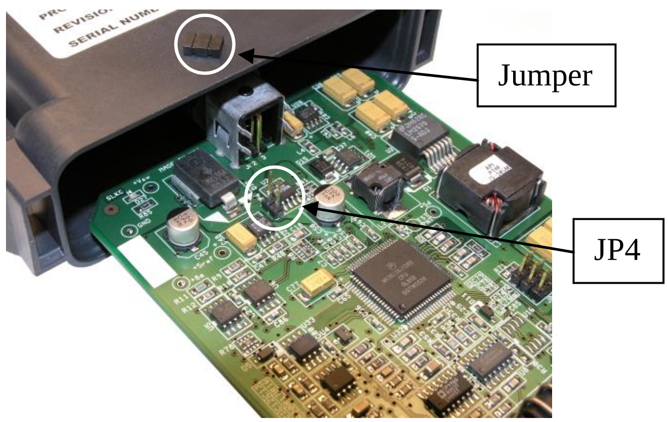


3. Click on the "Set to EIB Mode". Wait for "Done" Message.
Stop communication and exit program.



5. 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 installed on the EIB bus 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 CAN.

Appendix H: 5200 Series Instruments Parts List

01 Display Heads

01-01 NOx 5210
01-02 Lambda 5220
01-03 EGR 5230
01-04 dashCAN
01-05 dashCAN+
01-06 dashCANc

02 CAN Modules

02-01 LambdaCAN
02-02 NOxCAN
02-03 NOxCANg
02-04 LambdaCANc
02-05 appsCAN
02-06 baroCAN
02-07 NOxCANt
02-08 LambdaCANp
02-09 LambdaCANd
02-10 gpCAN
02-11 CO/CO2CAN

03 Modules

03-01 Lambda Sensor Simulator, SIM300
03-02 NOx Sensor Simulator, SIM400
03-03 Ceramic Sensor Heater, BTU200
03-04 NOxg Sensor Simulator
03-05 NOxt Sensor Simulator
03-06 LambdaCANp Sensor Simulator

04 Power Supplies

04-01 AC/DC Power Supply, Universal, 24VDC @ 4.2A
04-02 Vboost Supply, 10~14VDC to 24VDC @ 14.5A
04-03 30A AC/DC Power Supply, 15V, 120VAC
04-04 15A AC/DC Power Supply, 15V, 120VAC
04-05 60A AC/DC Power Supply, PWR60, 15V, 120VAC

05 Linear O_2 Sensors

05-01 NTK 6 mA
05-02 Bosch LSU4.2
05-03 Bosch LSU4.9
05-04 NTK 4 mA
05-05 Bosch LSU4.2, Type P
05-06 Delphi OSL
05-07 NTK 4mA Cofired (ZFAS-U2)
05-08 Bosch LSU4.9, Type P
05-09 Bosch ADV
05-10 NTK, 6mA, Type P
05-11 Bosch LSU4.2, Type PI (Intake)
05-12 CO, CO2

06 NOx Sensors

06-01 NTK
06-02 NGK
06-03 Calibrate NOx Sensor
06-04 Cal Sheet with NOx Sensor
06-05 NTK (Type 2), (use with NOxCANt)

07 Sensors

07-01 Pressure, 0-75 psia, ¼” tube fitting, (USA)
07-02 Pressure, 0-517 kPa, 6mm fitting, (Metric)
07-03 Pressure, Type P, 0-75 psia, ¼” tube fitting, (USA)
07-04 Pressure, Type P, 0-517 kPa, 6mm fitting, (Metric)
07-05 Pressure, Type KP, 0-100 psia, 1/4", (USA)
07-06 Pressure, Type KP, 0-689 kPa, 6mm, (Metric)
07-07 RH (Humidity) Sensor
07-08 Pressure (bCAN, LCANp), 0-75 psia, 1/4", (USA)
07-09 Pressure (bCAN, LCANp), 0-517 kPa, 6mm, (Metric)
07-10 Pressure (bCAN, LCANp), Type KP, 0-75 psia, 1/4", (USA)
07-11 Pressure (bCAN, LCANp), Type KP, 0-517 kPa, 6mm, (Metric)

08 Actuators

08-01 Ceramic Sensor Heater Mount

09 Eurofast Cables, Ts, Termination Resistors, and Connectors

09-01 4 m Eurofast 12 mm Cable
09-02 2 m Eurofast 12 mm Cable

09-03/n “n” m Eurofast 12 mm Cable
09-04 Flexi-Eurofast Cable, 0.3 m
09-05 Eurofast “T”
09-06 Eurofast Termination Resistor
09-07 Eurofast Male Connector
09-08 8 Channel Eurofast Hub Block
09-09 Minifast Termination Resistor (for Hub Block)

10 Sensor Cables

10-01 Module Y Cable
10-02 1 m Lambda Cable
10-03 2 m Lambda Cable
10-04 1 m Pressure Cable
10-05 2 m Pressure Cable
10-06 Adapter to Lambda Wires (NTK wire colors)
10-07 Adapter to NTK, 6 mA, JAMP
10-08 Adapter to LSU4.2
10-09 Adapter to LSU4.9
10-10 Adapter to Lambda Wires (Bosch wire colors)
10-11 Adapter to NTK NOx Wires (NTK wire colors)
10-12 Adapter to Pressure Sensor Wires
10-13 Adapter to use VW LSU4.9 with LCAN
10-14 Adapter to use P/N 2400E-1 sensor (CPC) with LCAN
10-15 Adapter to use JAMP (NTK 4 mA) with LCAN
10-16 Adapter to use P/N 2400E-1S sensor (Fischer) with LCAN
10-17 Adapter to use P/N 1001A-2 (Deutsch) with LCAN
10-21 Module Y Cable (for all except LambdaCANp)
10-22 1.5m Cable to LSU4.2 for LambdaCANc
10-23 1.5m Cable to LSU4.9 for LambdaCANc
10-24 1.5m Cable to ADV for LambdaCANc
10-25 1.5m Cable to NTK UEGO for LambdaCANc
10-26 1m Humidity Cable (bCAN), (6 term.)
10-27 2m Humidity Cable (bCAN), (6 term.)
10-28 Adapter to use Sumitomo NTK UEGO sensor with LCAN
10-29 Adapter to use ETAS LA3 (LSU4.2) sensor with LCAN
10-30 Module Y Cable (for baroCAN only)
10-31 1m Extension Cable for 12 terminal Deutsch
10-32 2m Extension Cable for 12 terminal Deutsch
10-33 1.5m Cable to OSL for LambdaCANc
10-34 Module Y Cable (for LambdaCANp only)
10-35 1m Pressure Cable (bCAN, LCANp only), (8 term.)
10-36 2m Pressure Cable (bCAN, LCANp only), (8 term.)
10-37 3m Lambda/NOx/baro Cable, (12 term.)
10-38 3m Pressure Cable (bCAN, LCANp only), (8 term.)

11 Cables

11-01 DC Power Cable, DB9F, Spades
11-02 DC Power Cable, DB9F, Banana Plugs
11-03 DB9M to CSM Lemo F Adapter
11-04 DB9M to ETAS Lemo Adapter
11-05 Female Eurofast to DB9F
11-06 Male Eurofast to DB9F (to add others to end of CAN bus)
11-07 Not assigned
11-08 2 m Key-on Cable
11-09 2 m Heater Cable
11-10 2 m Heater Power Cable
11-11 Heater Power Hookup
11-12 Simulator Lambda Adapter Cable
11-13 Simulator NOx Adapter Cable
11-14 BNC to Banana Cable
11-15 DC Power Cable, Spades
11-16 DC Power Cable, Banana Plugs
11-17 Deutsch DTM3M to DB9F
11-18 3 m, DB9 Cable, M-F
11-19 Heater to EIB Power Adapter
11-20 25' DB9 M-F Cable
11-21 SIM-200 Calibration Kit
11-22 Left (gray) appsCAN Connector with 300mm Pigtail Wires
11-23 Right (blk) appsCAN Connector with 300mm Pigtail Wires
11-24 Connector Kit: 2 connectors, 24 terminals, 12 plugs
11-25 Male Eurofast to Braided Shield Ground
11-26 Boom Box Cable for CAN Products (80' CAN, 30' Power)
11-27 Boom Box to Hub Springy Cable (1m relaxed, 2m stretched)

12 Mounting Panels, Bosses, and Hardware

12-01 19” Rackmount Panel. Holds up to 4 Display Heads.
12-02 18 mm x 1.5 mm Mild Steel (MS) Boss and Stainless Steel (SS) Plug
12-03 18 mm x 1.5 mm SS Boss and SS Plug
12-04 18 mm x 1.5 mm Tall Aluminum (Al) Boss, Copper (Cu) Gasket, and Al Plug
12-05 ¼” NPT MS Boss and Brass Plug
12-06 ¼” NPT SS Boss and Brass Plug
12-07 ¼” NPT Al Boss and Brass Plug
12-08 Pressure Line Assembly, ¼” tubing/hose/fittings, 19”, (USA)
12-09 Inconel Shield
12-10 18 mm Cu Gasket
12-11 Pressure Line Assembly, 6 mm tubing/hose/fittings, 483 mm, (Metric)
12-12 ¼” ISO tapered MS Boss and Brass Plug

12-13 ¼” ISO tapered SS Boss and Brass Plug
12-14 ¼” ISO tapered Al Boss and Brass Plug
12-15 15 A fuse
12-16 Bifurcated Intake Sample Probe, 8mm
12-17 Replacement Bifurcated Tube, 8mm
12-18 Aluminum Sensor Mounting Block
12-19 Individual Cylinder Exhaust Probe (USA)
12-20 Individual Cylinder Exhaust Probe (Metric)
12-21 PS Rolling Cart to Support 8 LCAN or NCANs (Pwr & Suc)
12-22 Sampling-Tyle Exhaust Probe (USA)
12-23 Sampling-Tyle Exhaust Probe (Metric)
12-24 Small Heated Aluminum Sensor Heater Block
12-25 1/4" UNC Module Stacking Standoff
12-26 Small Aluminum Sensor Mounting Block
12-27 Copper Gasket for 20mm x 1.5mm Boss and Plug
12-28 20mm x 1.5mm SS Boss and Plug for NGK NOx
12-29 18mm x 1.5mm (male) to 1/4" NPT (female)
12-30 Carrying Case, Medium
12-31 Aluminum Sensor Mounting Block, 20mm & 18mm
12-32 Small Aluminum Sensor Mounting Block, 20mm & 18mm
12-33 Pressure Line Assembly (for baroCAN), 1/4", (USA)
12-34 Presure Line Assembly (for baroCAN), 6mm, (Metric)
12-35 Carrying Case for SIM300
12-36 Carrying Case for SIM400
12-37 Carrying Case for SIM500
12-38 Carrying Case for SIM600
12-39 Carrying Case for SIM700
12-40 Individual Cylinder Exhaust Probe, 20mm Sensor (USA)
12-41 Individual Cylinder Exhaust Probe, 20mm Sensor (Metric)

13 Software and Media

13-01 5200 Series Analyzer and Module Manuals and Configuration Software, CD
13-02 Kvaser Leaf Light CAN Adapter
13-Product Name (Manual)

14 Tools

14-01 18 mm x 1.5 mm Tap
14-02 18 mm x 1.5 mm Die
14-03 ¼” NPT Tap
14-04 ¼” ISO Tapered Tap
14-05 Antiseize
14-06 Metal Brush

Appendix I: 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
EGR 4830 Analyzer
NOx 5210 NOx Analyzer
Lambda 5220 Lambda Analyzer
EGR 5230 EGR Analyzer
LambdaCAN, LabmdaCANc, LambdaCANd, LambdaCANp Lambda modules
NOx1000, NOxCAN, NOxCANg, NOxCANt NOx modules
CO/CO_2CAN module
baroCAN Module
dashCAN, dashCAN+
SIM300, SIM400, SIM500, SIM600, SIM700
BTU200 Ceramic Sensor Heater

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 22, 2006