USER MANUAL FOR THE

CANbus PRESSURE TRANSDUCER

SERIES 9000

Part Number: 562293
Issue: A

CONTENTS

1. OPERATING & INSTALLATION INSTRUCTIONS

IMPORTANT NOTE: All GEMS Pressure and Level & Flow Products are designed and manufactured in accordance with Sound Engineering Practice as defined by the Pressure Equipment Directive 97/23/EC. Pressure transducer products designed to meet the highest risk category "IV" of the Pressure Equipment Directive are clearly marked on the label by "CE0086". Compliance is achieved through modules "B+D". No other products should be used as "Safety Accessories" as defined by the PED, Article 1, Paragraph 2.1.3.

INTRODUCTION

Series 9000 are fluid pressure measuring transducers in which a four active arm Wheatstone bridge of thin film gauges, integral with a beam structure which in turn is connected to a pressure summing diaphragm, is used to convert fluid pressure into a proportional electrical signal. In built intelligence in the form of micro-controller provides a digital representation of the measured pressure, with all errors due to temperature and non-linearity accurately corrected. The 9000 output signal is in the form of a CANopen communications protocol (see getting started).

Series 9000 conform with the essential protection requirements of the EMC Directive 89/336/EEC amended by certified type testing to EN 61000-6-4 and EN 61000-6-2.

Conformity with the requirements of the CE mark only applies when the installation conditions described in these instructions have been met. For units supplied without a cable assembly connection to the transducer must be accomplished using Gems Sensors approved cable. See APPROVED CABLE section.

All instruments conform to the appropriate specifications and/or drawings applicable and have been subjected to relevant strict quality control procedures.

HAZARDOUS PRODUCTS

The Consumer Protection Act of 1987, Section 6 of the Heath and Safety at Work Act 1974 and the Control of Substances Hazardous to Health Regulations 1988 require that we advise recipients and users of our products of any potential hazards associated with their storage, handling or use.

The products which our Company supplies may be classified as Electrical, Electro-Mechanical and Electronic equipment.

These products are tested and supplied in accordance with our published specifications or individual special requirements that are agreed in writing at time of order. They are constructed so as not to affect adversely the safety of persons and property when properly installed, maintained and used by qualified personnel, in the applications for which they were designed and manufactured

ACTION ON RECEIPT

* Check details on Calibration Certificate agree with data etched on transducer body.
* Check accessories supplied include:

  Calibration Certificate (or QA Certificate)
  Seal, Dowty Bonded
  Mating Connector (where applicable)

GENERAL

- Transducer should not be subjected to greater than the maximum allowable pressure range / Temperature (T.S) as defined on the transducer label.
- Transducer should not be subjected to mechanical impact.
- In the event of fire the end user must ensure that the system pressure is vented to a safe area.
- The effects of decomposition of unstable fluids should be considered by the user when placing this device in service.
- The pressure transducer has no means of draining or venting, this must be performed by another component in the end users system.
- Pressure range must be compatible with the maximum pressure being measured.
- Pressure media must be compatible with the transducer wetted parts which are:-

  Ranges 1 and 1.6 bar, 15 and 30 psi : 15-7 Mo plus 17-4 PH stainless steels
  Ranges 2.5 bar and 60 psi and above : 17-4 PH stainless steel
  Pressure connection codes K or L : Inconel alloy 625

- Exposed end of cable must be kept free from moisture.
- Liquid must not be allowed to freeze in the pressure port.

Full specifications for all products available on request from our Service Department.

MECHANICAL INSTALLATION

Pressure Connections: G¼ internal pressure connection to BS2779 as standard. Alternative fitted as specified at time of order.

Pressure couplings screwed into G¼ pressure ports should have a maximum thread engagement 13 mm and **UNDER NO CIRCUMSTANCES** be allowed to touch the pressure sensitive diaphragm. Pressure couplings should be sealed against the outer face at the pressure port entry using bonded seal washer such as:-

Part Number	Description
234646-0002	G¼ bonded seal up to +100°C operation. Dowty Ref: 400-021-4490-02
499207-0002	G¼ bonded seal up to +200°C operation. Dowty Ref: 300-021-0967-02
499207-0006	G1/8 bonded seal up to +200°C operation. Dowty Ref: 300-020-0967-02

Refer to Dowty for external pressure ratings.

Mounting: The pressure transducer is designed to be attached by the coupling thread only. Omni-directional. To fit, use a 22.2 mm (7/8 inch) AF spanner on the hexagon provided and apply maximum torque of 27 Nm (20 lbf-ft). The Customer must ensure that the pressure seal is suitable for the application. If in doubt contact Gems Sensors.

Effects of Pressure Transients: If it is suspected that adverse transient pressure pulses are present for any reason, then fit a restrictor such as our part number 466175-0000 (depending upon pressure adaptor).

Vibration: Where present, mount in a saddle clamp such as part number 499877-1000 (material: Polypropylene) or 499877-1001 (material: Polyamide). Position the clamp as close to the pressure port as practicable.

Effects of Heat: Avoid mounting the transducer near a source of heat which is liable to create a temperature gradient across the instrument. If this is unavoidable, use a heat shield to deflect uneven radiated heat or wrap the transducer in glass fibre insulation so that an even temperature is assumed throughout.

High Temperature Pressure Measurement: Possible using a length of piping or a "syphon" to isolate the instrument from the hot media. Since there is no flow, the temperature drop along the tubing is considerable and usually a relatively short length is sufficient to bring the pressure media temperature within acceptable limits for the instrument. Alternatively a temperature isolator, part number 558564-0001, can be fitted (G¼ connection only). Pressure media must not be allowed to solidify in the tube and/or pressure port.

ELECTRICAL INSTALLATION

All types include suppression devices providing transient protection. The 9000 requires a d.c. power supply between 7V and 30V. The maximum current should be limited via a suitable power supply or fuse to 1A.

For all types conformity with the requirements of the CE mark only applies when connection is made with Gems Sensors approved cable, See APPROVED CABLE section, and the screen of that cable is connected to a reliable earthing point at the instrumentation end.

Types 90XXX-XX-M/P2-X: These types are capable of being immersed to a maximum depth of 200m WG.

Types 90XXX-XX-L2-X: These types are fitted with a M12 x 1, 5 pin receptacle.

ELECTRICAL REQUIREMENTS

Electrical Code	Supply Volts	Output	CAN Shield	CAN +ve Supply	CAN -ve Supply	CAN Hi	CAN Lo
L	7 to 30	CANbus	Pin 1	Pin 2	Pin 3	Pin 4	Pin 5
M	7 to 30	CANbus	Green	Red	White	Yellow	Blue

APPROVED CABLES

Gems Sensors uses cable comprising 6 colour-coded cores, with a central vent tube, enclosed by an aluminium/polyester screen where the screen is in intimate contact with a separate drain wire. The outer sheath can be of various material, depending upon application and operating temperature, standard Polyurethane (immersible, +50°C). Other cables available on request for operation at higher temperatures or in corrosive media.

OPERATION

Having installed the transducers as instructed they are ready for use. The transducer should not be removed whilst the system is at pressure. Before applying power, check that the correct polarity and excitation levels are being applied. See ELECTRICAL REQUIREMENTS.

Compensated Temperature Range : -40°C to +85°C

Operable Temperature Range: Electrical Connector Code L -40°C to +85°C
Electrical Connector Code M -20°C to +50°C (Standard Polyurethane Cable)

OPERATIONAL LIFE

Limited to 100 million cycles to maximum allowable pressure.

CALIBRATION

Transducers are calibrated to the datum requested at time of order; this can be identified by the fifth entry in the stock numbering code as follows:-

G - gauge datum vented to atmosphere via the electrical connection

A - absolute datum

S - sealed reference; reference side of the instrument is sealed and the output electrically adjusted to zero with 1013.25 mbar applied to pressure port

GETTING STARTED

See section 2.1.

The 9000 operates according to the CANOPEN protocols:-

DS301
DSP305
DSP404
D2P303-2

Alternatively a Starter Kit, including PC Software and a USB-CAN interface is available part number 562321.

MAINTENANCE

Routine Inspection: Not required except for periodic inspection of the cable and moulding to ensure that these are neither damaged nor softened by incompatible liquid.

CAUTION

CARE MUST BE TAKEN NOT TO TOUCH THE PRESSURE SENSITIVE DIAPHRAGM WHILST CLEANING THE PRESSURE PORT. FAILURE TO OBSERVE THIS PRECAUTION CAN CAUSE IRREPARABLE DAMAGE.

WARRANTY

The Company warrants its products to be free from defects in material and workmanship in normal use and service for a period of two years from date of shipment. The Company reserves the right and option to refund the purchase price in lieu of repair or replacement upon evaluation of the returned original part. Modification, misuse, attempted repair by others, improper installation or operation shall render this guarantee null and void. The Company makes no warranty of merchantability or fitness for a part or purpose.

SERVICING

The transducer cannot be repaired locally and if damaged should be returned to ourselves at the address shown below or to accredited dealers when a replacement/repair is required:

Gems Sensors Lennox Road Basingstoke Hants. RG22 4AW	Gems Sensors Inc 1 Cowles Road Plainville, CT 06062 U.S.A.

RETURN TO FACTORY

PLEASE NOTE: To comply with Health and Safety requirements, the instrument must be clean and safe to handle and accompanied by a formal statement to that effect duly signed by an authorised officer of the Company.

Any instrument returned without certification will be quarantined and no action will occur until cleared. It may ultimately be returned to you and subject to a transportation charge.

2. INSTALLATION

2.1 Getting Started

The series 9000 is provided with the CANopen default connection set shown in Table 2. In order to operate in the host application then it is necessary to ensure:

i) The bit rate of the series 9000 matches that of the network to which it is being connected.

ii) The Node-ID of the series 9000 is not the same as any other node on the CANbus network.

The configuration of both of these parameters is performed using the layer settings services. It is common to configure these parameters prior to connecting the series 9000 with other devices on the CANbus via a configuration program. The PC application in the rapid development kit provides a facility for such configuration using a low cost USB to CAN interface.

2.1.1 Configuring the Bit Rate

To configure the bit rate follow steps 1 to 7.

1. Connect the series 9000 to the configuration program & apply excitation to the series 9000. The series 9000 will transmit the boot message shown in Figure 1.

Figure 1
Boot Message

ID	Byte 0
701_{16}	00_{16}

2. Set the 9000 to the stopped state by sending the stop remote node NMT command shown in figure 2.

Figure 2
Stop Remote Node Command

ID	Byte 0
02_{16}	01_{16}

3. Send the switch mode global command, shown in Figure 3, to set the 9000 to configuration mode.

Note:
Byte 0 is the command specifier, a value of 04_{16} being the switch mode global service.

Byte 1 is the mode selected, 0=switches to operation mode; 1=switches to configuration mode.

Figure 3
Switch Mode Global Command

ID	Byte 0	Byte 1
$7E5_{16}$	04_{16}	01_{16}

4. Send the configure bit timing command, shown in Figure 4. Byte 1 should be chosen from Table 1 to provide the required bit rate. The 9000 will respond as shown in Figure 5.

Figure 4
Configure Bit Timing Command

ID	Byte 0	Byte 1
$7E5_{16}$	13_{16}	4_{16}

Note: A value of 4 in byte 1 will provide a bit rate of 125 kbps.

Figure 5
9000 Response to Configure Bit Timing Command

ID	Byte 0	Byte 1
$7E4_{16}$	0_{16}	0_{16}

Note: A value of 1 in byte 0 indicates the bit rate is not supported.

Table 1
Available Bit Rates

Bit Rate	Table Index
1000 Kbit	0
800 Kbit	Not supported
500 Kbit	2
250 Kbit	3
125 Kbit	4
100 Kbit	5
50 Kbit	6
20 Kbit	7
10 Kbit	Not supported

5. Send the activate bit timing command, shown in figure 6 to re-start with the selected bit rate. The switch delay is an Intel format unsigned integer number that represents a time delay in mS, e.g. $3E8_{16}$ = 1000mS. This time delay is used for two periods on the 9000. Firstly the new bit timing will not be applied until the delay is expired, secondly no messages shall be sent by the 9000 until the delay has expired once more.

Figure 6
Activate Bit Timing Command

ID	Byte 0	Byte 1	Byte 2
	Command Specifier	Switch Delay	
$7E5_{16}$	15_{16}	$E8_{16}$	03_{16}

6. Before storing the configuration then the Node-ID can also be configured, refer to 2.1.2. Send the store configuration command, shown in Figure 7, to store the new bit timing in a non volatile manner. This command should now be transmitted with the new bit rate. Figure 8 shows the 9000 response.

Figure 7
Store Configuration Command

ID	Byte 0
$7E5_{16}$	17_{16}

Figure 8
9000 Response to Store Configuration Command

ID	Byte 0	Byte 1
$7E4_{16}$	0_{16}	0_{16}

Note: A value of 2 in byte 0 indicates a storage media access error.

7. Send the switch mode global command, shown in Figure 9, to set the 9000 to operation mode.

Note: Byte 0 is the command specifier, a value of 04_{16} being the switch mode global service.

Byte 1 is the mode selected, 0=switches to operation mode; 1=switches to configuration mode.

Figure 9
Switch Mode Global Command

ID	Byte 0	Byte 1
$7E5_{16}$	04_{16}	00_{16}

2.1.2 Configuring the Node-ID

The Node-ID is used to provide unique communication object identifiers for each CANopen device on the network. Messages are prioritised on the network so that the lowest identifier will gain access to the bus first. Configuring of the Node-ID can be performed in-situ on the target network using the switch mode selective command, but is more commonly configured prior to connection using the switch mode global command.

To configure the Node-ID then with the 9000 in the configuration mode, refer to 2.1.1, the configure Node-ID command shown in Figure 10 must be sent. The 9000 response is shown in Figure 11.

Figure 10
Configure Node-ID

ID	Byte 0	Byte 1
$7E5_{16}$	11_{6}	$0A_{16}$

Figure 11
9000 Response to Configure Node-ID

ID	Byte 0	Byte 1
$7E5_{16}$	0_{16}	0_{16}

Note: A value of 1 in byte 0 indicates the Node-ID is out of range.

3. CANOPEN IMPLEMENTATION

3.1 Communications

The series 9000 implements the standard CANOpen communications protocol. Table 2 shows the CANOpen standards supported. These standards are available from CAN in Automation www.can-cia.de.

Table 2
Supported CANOpen Standards

CANopen CIA Standard	Version	Description
DS301	4.02	Application Layer & Communication Profile
DSP 305	1.0	CANopen Layer Setting Services & Protocol
DSP 404	1.0	Measuring Devices & Closed Loop Controllers
DRP-303-2	1.1	Representation of SI Units and Prefixes

3.1.1 Default Settings for CANopen Communication

The series 9000 utilises the CANopen predefined connection set, Table 3 summarises these default settings.

Table 3
Series 9000 Default Settings

Parameter	Default Value	Description
Bit rate	250 Kbps	The bit rate that is used for CAN communications.
Node-ID	1	The Node-Identifier used for CANopen communications.
LSS Master	$7E5_{16}$	The identifier of the layer settings service master (host).
LSS Slave	$7E4_{16}$	The identifier of the layer settings service slave (series 9000).
COB-ID NMT	00_{16}	The communications object identifier of the Network Management Object
COB-ID SYNC	80_{16}	The communications object identifier of the Synchronisation Object.
COB-ID EMERGENCY	80_{16} + Node-ID	The communications object identifier of the Emergency Object.
COB-ID PDO	180_{16} + Node-ID	The communications object identifier of the Process Data Object.
COB-ID SDO	600_{16} + Node-ID	The communications object identifier of the Service Data Object.
BOOTUP / HEARTBEAT COB-ID	700_{16}+ Node-ID	The communications object identifier for the heartbeat object.

3.2 Network Management

The 9000 is a CANopen NMT slave, Figure 12 shows the state machine that is implemented. Table 4 shows the types of communication support available in each state. Automatically after power on the 9000 will enter the pre-operational state.

Figure 12
Network Management State Machine

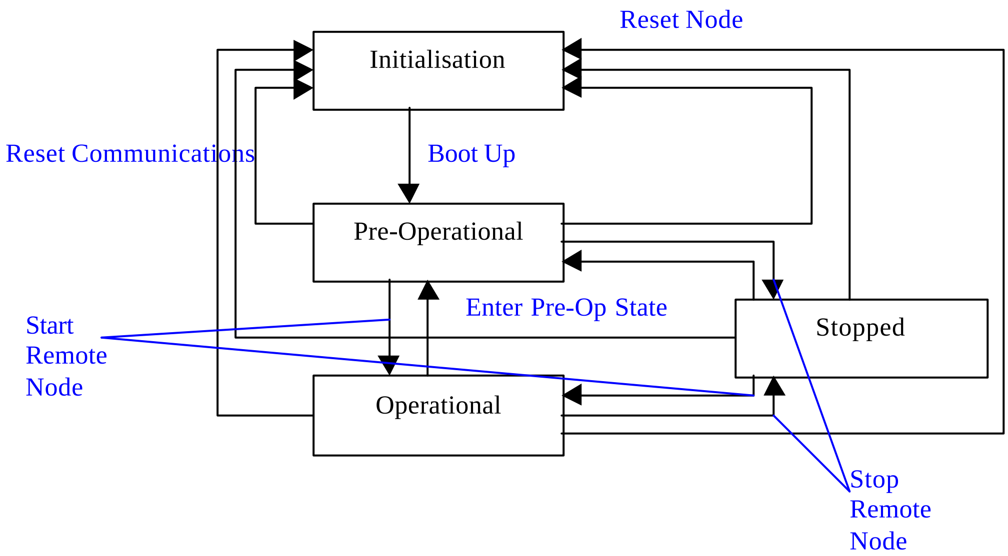

Table 4
Network Management States and Communications Support

	Initialising	Pre-Operational	Operational	Stopped
PDO			✓	
SDO		✓	✓	
Synchronisation		✓	✓	
Emergency		✓	✓	
Boot-Up	✓			
NMT		✓	✓	✓
LSS				✓

3.2.1 Boot Up Protocol

During the initialisation state the transducer internally reads the object dictionary from non-volatile memory. On completion the pre-operational state is entered & the boot-up message, shown in Figure 13, is transmitted to signal the transducer has entered the pre-operational state. The 9000 can be configured to skip the boot-up message and enter the operational state when excitation is applied via object 2813_{16}, accessible via the service data object.

Figure 13
Boot-Up Message

ID	Byte 0
700_{16} + Node-ID	
701_{16}	00_{16}

3.2.2 Entering Operational State

After power up the 9000 will autonomously reach the pre-operational state. In this state the 9000 can be configured using the SDO. For example a filter could be applied to the pressure reading, or the pressure reading and maximum pressure reading could be mapped to the PDO. When configured the start remote node command, shown in Figure 14 should be sent to the 9000. The 9000 will then enter the operational state, allowing readings to be obtained from the PDO.

Figure 14
Start Remote Node

ID	Byte 0	Byte 1
	Cmd	**Node-ID**
00_{16}	01_{16}	00_{16}

Note: A Node-ID of 00 will start all node on a network.

3.2.3 Stop Remote Node

The stop remote node command, shown in figure 15, causes the 9000 to enter the stopped state. In the stopped state the LSS are available.

Figure 15
Stop Remote Node

ID	Byte 0	Byte 1
	Cmd	**Node-ID**
00_{16}	02_{16}	01_{16}

Note: byte 1 should be set to the value of the Node-ID of the 9000.

3.2.4 Enter Pre-Operational State

Figure 16
Enter Pre-operational State

ID	Byte 0	Byte 1
	Cmd	**Node-ID**
00_{16}	80_{16}	01_{16}

Note: byte 1 should be set to the value of the Node-ID of the 9000.

3.2.5 Reset Node

The reset node command, shown in Figure 17, causes the 9000 to re-load the whole object dictionary from non-volatile memory.

Figure 17
Reset Node Command

ID	Byte 0	Byte 1
	Cmd	Node-ID
00_{16}	81_{16}	01_{16}

3.2.6 Reset Communications

The reset communications command, shown in Figure 18, causes the 9000 to re-load the communications area of the object dictionary from non-volatile memory.

Figure 18
Reset Communications Command

ID	Byte 0	Byte 1
	Cmd	Node-ID
00_{16}	81_{16}	01_{16}

3.3 Heartbeat Protocol

The 9000 can provide a heartbeat message at a pre-determined period, this can be used by the NMT master to monitor that the 9000 exists on the network. The heartbeat protocol is implemented in favour of the node guarding protocol as the heartbeat protocol provides error control without the need for remote frames. The 9000 will send a heartbeat message every n mS, where n is the heartbeat producer time available at object 1017_{16}. If the heartbeat producer time is zero then no heartbeat message will be transmitted.

Figure 19
9000 Heartbeat Message

ID	Byte 0
700_{16} + Node-ID	Cmd
701_{16}	NMT state

Where valid values for NMT state are:-

0: BOOTUP

4: STOPPED

5: OPERATIONAL

127: PRE-OPERATIONAL

3.4 SDO Access to the Object Dictionary

The 9000 allows only expedited SDO access to the object dictionary. This means that all reads or writes, to or from, the object dictionary take the form of a single CAN frame containing the command and data, followed by a single CAN frame containing the 9000 response and data.

3.4.1 Expedited SDO Upload – Object Dictionary Read

The expedited SDO request is made from the Client to the 9000 (server). Figure 20 shows the format of an expedited SDO frame, whilst Figure 21 shows an example of reading the device type from object 1000_{16}.

Figure 20
Expedites SDO Upload Request Frame Contents

Byte 0		Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
CCS 7..5	X 4	Index LSB	Index MSB	Sub Index	Data	Data	Data	Data

Where:-

CCS = Client Command Specifier – a value of 2 indicates the initiate SDO Upload command.

X = reserved always set to zero.

Figure 21
Example Read from Object Dictionary

ID	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
600_{16} + Node-ID	Control	Index		Sub Index	Reserved			
601_{16}	40_{16}	00_{16}	10_{16}	00_{16}				

9000 Response

ID	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
580_{16} + Node-ID	Control	Index		Sub-Index	Data			
581_{16}	42_{16}	00_{16}	10_{16}	00_{16}	94_{16}	01_{16}	02_{16}	00_{16}

3.4.2 Expedited SDO Download – Object Dictionary Write

Figure 22 shows the format of an expedited SDO download request frame, whilst Figure 23 shows an example of writing the heartbeat time to object 1017_{16}.

Figure 22
Expedited SDO Download Request Frame Format

Byte 0					Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
CCS 7..5	X 4	N 3-2	E 1	S 0	Index LSB	Index MSB	Sub Index	Data	Data	Data	Data

Where:-

CCS: Client Command Specifier – a value of 1 indicates the initiate SDO download request.
X: reserved always set to zero.
N: only valid if e=1 & s=1, otherwise 0. If valid indicates the number of bytes that do not contain data.
E: transfer type. 1=expedited, 0=normal.
S: size indicator. 0=data size not indicated, 1=data size indicated.

Figure 23
Example Write to Object Dictionary

SDO Request

ID	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
600_{16} + Node-ID	Control	Index		Sub-Index	1000 mS		Reserved	
					LSB	MSB		
601_{16}	22_{16}	17_{16}	10_{16}	00_{16}	$E8_{16}$	03_{16}		

9000 Response

ID	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
580_{16} + Node-ID	Control	Index		Sub-Index	Reserved			
581_{16}	60_{16}	17_{16}	10_{16}	00_{16}	00_{16}	00_{16}	00_{16}	00_{16}

3.4.3 SDO Abort

The 9000 is capable of detecting errors within the SDO protocol, examples of such errors are:

- Index/Sub Index does not exist in the object dictionary
- The range of the uploaded/downloaded data is not valid
- Size mismatch between object dictionary entry & the number of bytes actually transferred.

If an error is detected then the 9000 replies with an SDO abort message. This message contains a 4 byte abort code in bytes 4 to 7. Table 5 details the supported abort codes and their meaning. Figure 24 shows an example of the abort response when access to an object with the incorrect sub-index is attempted.

Table 5
Supported SDO Abort Codes

Abort Code	Meaning
0504 001_{16}	Client/Server command specifier not valid or unknown
0601 0001_{16}	Attempt to read a write only object
0601 0002_{16}	Attempt to write a read only object
0602 0000_{16}	Object does not exist in object dictionary
0604 0041_{16}	Object can not be mapped to the PDO
0604 0042_{16}	The number & length of object to be mapped would exceed the PDO length
0607 0010_{16}	Data type does not match, length of service parameter does not match
0609 0011	Sub-Index does not exist
0609 00 30	Value of parameter exceeded (only for write access)
0800 0000	General error
0800 0020	Data can't be transferred or stored to the application

Figure 24
SDO Access to Invalid Sub-Index

SDO Request

ID	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
600_{16} + Node-ID	Control	Index		Sub-Index	Reserved			
601_{16}	40_{16}	00_{16}	10_{16}	01_{16}				

9000 Response

ID	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
580_{16} + Node-ID	Control	Index		Sub-Index	Data			
581_{16}	80_{16}	00_{16}	10_{16}	01_{16}	11_{16}	00_{16}	09_{16}	06_{16}

3.5 Process Data Object

The process data object is used to transmit the process data variable in real time. The default operation of the 9000 is to transmit the pressure and temperature in 16 bit signed integer format.

3.5.1 PDO Mapping

The 9000 allows different objects to be mapped into the process data object. This allows the user to configure the data they will receive in the PDO during real time data acquisition. For example pressure and temperature in signed integer format or pressure and temperature in floating point format. Figure 25 shows the PDO mapping record format, that defines the PDO mapping object, 1A00. Figure 26 shows the default mapping of the 9000. Table 6 shows the objects that may be mapped to the PDO.

Figure 25
Format of PDO Mapping Record

Sub Index		**Field in Record**		**Data Type**
0_{16}		Number of Objects Mapped to PDO		U8
1_{16}		1st Object to be mapped		U32 Intel
2_{16}		2nd Object to be mapped		U32 Intel
3_{16}		3rd Object to be mapped		U32 Intel
4_{16}		4th Object to be mapped		U32 Intel
Format of U32 (Sub Index 1_{16} to 4_{16})				
MSB$_{(31-24)}$	**(23-16)**	**(15-8)**		**LSB$_{(7-0)}$**
Index (16 bits Intel)		Sub-index (8 bits)		Object Length (8 bits)

Figure 26
9000 Default PDO Map

Object Number	**Sub Index**	**Value**	**Description**
$1A00_{16}$	00_{16}	02_{16}	Two objects are mapped
$1A00_{16}$	01_{16}	10013071_{16}	1st object is field value stored at 7130 sub index 1, length 16 bits.
$1A00_{16}$	02_{16}	10023071_{16}	2nd object is field value stored at 7130 sub index 2, length 16 bits.,
$1A00_{16}$	01_{16}	00000000_{16}	Not Mapped.
$1A00_{16}$	02_{16}	00000000_{16}	Not Mapped.

Table 6
Objects that can be Mapped to the PDO

Object	**Sub Index**	**Description**
6130_{16}	1_{16}	Pressure Reading REAL32
6130_{16}	2_{16}	Temperature Reading REAL32
7130_{16}	1_{16}	Pressure Reading INT16
7130_{16}	2_{16}	Temperature Reading INT16
4130_{16}	1_{16}	Pressure Reading Max REAL32
4130_{16}	2_{16}	Temperature Reading Max REAL32
4131_{16}	1_{16}	Pressure Reading Min REAL32
4131_{16}	2_{16}	Temperature Reading Min REAL32
5130_{16}	1_{16}	Pressure Reading VISIBLE STRING
5130_{16}	2_{16}	Temperature Reading VISIBLE STRING

3.5.2 PDO Transmission Type

The PDO transmission type is used to control how the PDO data is updated and transmitted over the CAN network. The 9000 supports a number of different transmission types that can be configured by writing the appropriate value to the PDO communications object, 1800_{16}, sub-index 2_{16}. Table 7 summarises the supported transmission types.

Table 7
Supported PDO Transmission Types

Transmission Type	Description
0	Acyclic, synchronous transmission. The 9000 will transmit a PDO message on reception of the SYNC message, providing the event timer has expired.
1-240	Cyclic, Synchronous transmission. The 900 will transmit the PDO message on reception of the n^{th} SYNC message, where n is 1 to 240.
254	Asynchronous transmission. The 9000 will transmit the PDO message at the time interval defined by the event time. The PDO transmission is not related to the SYNC message.

3.6 Emergency Object

The 9000 will transmit an emergency object if the internal error state changes, that is an error is detected or a source of error is removed. Figure 27 shows the format of the emergency object, whilst figure 28 shows the supported error codes.

Figure 27
Emergency Object Frame Format

ID	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
80 + Node-ID	Emergency Error Code		Error Register (Object 1001_{16})	Manufacturer Specific Error Field				

Figure 28
Supported Error Codes

Emergency Error Code	Error Register	Error Description
2201	C0	Sensor Failure
6101	81	Floating Point Maths Error
6102	81	Non volatile memory error.
8120	88	CAN in Error Passive Mode.
8140	88	Recovered from bus off.

4. OPERATION

The operation of the 9000 is configurable. All configuration relating to the process variables and their processing is accessible by SDO access to the object dictionary. The supported object dictionary is provided in Appendix 1. This section describes the basic operation and configuration choices available.

4.1 Obtaining a Pressure and Temperature Reading

The pressure and temperature are available, via the PDO, in decimalised integer format. The default operation of the 9000 is to reply with the PDO every time a SYNC message is received. Figure 29 shows the sequence of messages necessary to obtain a reading when excitation is applied to the 9000. It can be interpreted as follows:

1. 9000 transmits boot up message to PC on identifier 701.
2. Host application transmits start remote node command.
3. When SYNC command is transmitted, by the host, the 9000 responds with the PDO containing the pressure and temperature.

Figure 29
Obtaining a Pressure Reading

PC / Host Application									Direction	9000								
ID	**Byte 0**	**Byte 1**	**Byte 2**	**Byte 3**	**Byte 4**	**Byte 5**	**Byte 6**	**Byte 7**		**ID**	**Byte 0**	**Byte 1**	**Byte 2**	**Byte 3**	**Byte 4**	**Byte 5**	**Byte 6**	**Byte 7**
Boot up message from 9000									←	701	00							
00	01	00							→	NMT Command – Start Remote Node								
80									→	SYNC								
Process Data Object Bytes 0-1 = Pressure, bytes 2-3 = Temperature									←	181	02	00	E1	00	00	00	00	00
80									→	SYNC								
Process Data Object Bytes 0-1 = Pressure, bytes 2-3 = Temperature									←	181	02	00	E1	00	00	00	00	00

4.2 Interpreting & Changing the Data Formats

The 9000 is capable of providing the pressure or temperature readings in three different data formats.

4.2.1 Decimalised Integer

This is the default data format shown in Figure 29. The pressure and temperature data are signed integer values of Intel format, that is the byte order is LSB-MSB, and the bit order is msb-lsb. So referring to Figure 29 the integer pressure value is 2 and the integer temperature value is 225. The actual value of the pressure reading and temperature reading, depends on the number of decimal digits that should be used with these integer values. The decimal digits used can be read from the 9000 using an SDO access to object 6132, Figure 30 shows an example of the communications when reading the decimal digits.

Figure 30
Reading the Decimal Digits

PC / Host Application									**Direction**	**9000**								
ID	**Byte 0**	**Byte 1**	**Byte 2**	**Byte 3**	**Byte 4**	**Byte 5**	**Byte 6**	**Byte 7**		**ID**	**Byte 0**	**Byte 1**	**Byte 2**	**Byte 3**	**Byte 4**	**Byte 5**	**Byte 6**	**Byte 7**
601	40	32	61	1	0	0	0	0	→	SDO read from object 6132, sub-index 1 = decimal digits for pressure.								
9000 reply, byte 4 indicates 3 decimal digits									←	581	42	32	61	1	03	00	00	00
601	40	32	61	2	0	0	0	0	→	SDO read from object 6132, sub-index 2 = decimal digits for temperature.								
9000 reply, byte 4 indicates 1 decimal digit									←	581	42	32	61	1	03	00	00	00

So using the decimal digits value, the pressure reading is 0.002 and the temperature reading 22.5.

4.2.2 Floating Point

The 9000 also supports the IEEE-754-1985 single precision floating point number format, shown in Figure 31.

Figure 31
Floating Point Number Format

S – sign	**E – exponent**	**F - fraction**
b31	b30…..b23	b22…..b0
0	100 0000 1	100 1000 0000 0000 0000 0000
Number $= (-1)^{s}.2^{E-127}.(1+F) =$ 6.25		

Note: that the byte order when using floating point data is LSB – MSB.

To change the PDO so that the pressure and temperature readings are provided in floating point format it is necessary to write to the PDO mapping objects defining the object that should be placed in the process data. Figure 32 shows the communications required to map the floating point values to the PDO and obtain readings in floating point format.

Figure 32
Changing the PDO to Contain Floating Point Data

PC / Host Application									Direction	9000								
ID	**Byte 0**	**Byte 1**	**Byte 2**	**Byte 3**	**Byte 4**	**Byte 5**	**Byte 6**	**Byte 7**		**ID**	**Byte 0**	**Byte 1**	**Byte 2**	**Byte 3**	**Byte 4**	**Byte 5**	**Byte 6**	**Byte 7**
601	22	00	1A	0	0	0	0	0	→	Write zero to the mapping object 1A00, sub-index 0 to clear the existing PDO map.								
Confirmation data written									←	581	60	00	1A	00	00	00	00	00
601	22	00	1A	1	20	1	30	61	→	Write to sub-index 1 with object number, sub-index & length of the AI_Process Value object for pressure.								
Confirmation data written									←	581	60	00	1A	1	00	00	00	00
601	22	00	1A	2	20	2	30	61	→	Write to sub-index 2 with object number, sub-index & length of the AI_Process Value object for pressure.								
Confirmation data written									←	581	60	00	1A	2	00	00	00	00
601	22	00	1A	0	2	00	00	00	→	Write to sub-index 0 with the number of objects written.								
Confirmation data written									←	581	60	00	1A	0	00	00	00	00
601	22	00	18	1	81	01	00	80	→	Write to the PDO communications object, sub-index 1 to set the PDO COB-ID & make the PDO valid.								
Confirmation data written									←	581	60	00	18	00	00	00	00	00
80									→	SYNC								
9000 response, bytes 0 to 3 = FP pressure = 0.002, bytes 4 to 7 = FP temperature = 22.5.									←	181	6F	12	03	3B	00	00	B4	41

4.2.3 **Visible String**

The 9000 also supports a visible string data type. The data is interpreted as ISO 646-19743(E) 7 bit coded characters. The ASCII table is available from www.asciitable.com . Figure 33 shows the communications required to map the ascii values to the PDO and obtain readings in floating point format. Note that only one visible string can be mapped to the PDO because eight bytes are required to obtain a reading.

4.2.4 **Mapping Other Objects into the PDO**

Figure 32 and Figure 33 are examples of mapping other objects into the PDO. The basic sequence of events that is necessary to map new objects into the PDO is:-

a) Clear the existing object map by writing 0 to object 1A00, sub-index 0.

b) Write the new object map into sub-indexes 1 to 4 of object 1A00.

c) Write the number of objects mapped to 1A00, 0.

d) Write the PDO communications object with the PDO COB-ID and making the PDO valid.

When mapping new objects care must be taken not to map objects whose number of data bytes, when summed, is greater than eight. Table 8 shows the objects that can be mapped to the PDO.

Table 8
Objects that can be Mapped

Object	Sub-Index	Data Type	Description	Data that must be written to 1A00
4130	1	REAL32	Maximum Pressure Value	20013041
4130	2	REAL32	Maximum Temperature Value	20023041
4131	1	REAL32	Minimum Pressure Value	20013141
4131	2	REAL32	Minimum Temperature Value	20023141
5130	1	STRING	Pressure Value	40013051
5130	2	STRING	Temperature Value	40023051
6130	1	REAL32	Pressure Value	20013061
6130	2	REAL32	Temperature Value	20023061
7130	1	INTEGER	Pressure Value	10023071
7130	2	INTEGER	Temperature Value	10023071

Figure 33
Changing the PDO to Contain a Visible String

PC / Host Application									Direction	9000								
ID	**Byte 0**	**Byte 1**	**Byte 2**	**Byte 3**	**Byte 4**	**Byte 5**	**Byte 6**	**Byte 7**		**ID**	**Byte 0**	**Byte 1**	**Byte 2**	**Byte 3**	**Byte 4**	**Byte 5**	**Byte 6**	**Byte 7**
Example for Pressure String																		
601	22	00	1A	0	0	0	0	0	→	Write zero to the mapping object 1A00, sub-index 0 to clear the existing PDO map.								
Confirmation data written									←	581	60	00	1A	00	00	00	00	00
601	22	00	1A	1	40	1	30	51	→	Write to sub-index 1 with object number, sub-index & length of the AI_Process Value object for pressure.								
Confirmation data written									←	581	60	00	1A	1	00	00	00	00
601	22	00	1A	0	1	00	00	00	→	Write to sub-index 0 with the number of objects written.								
Confirmation data written									←	581	60	00	1A	0	00	00	00	00
601	22	00	18	1	81	01	00	80	→	Write to the PDO communications object, sub-index 1 to set the PDO COB-ID & make the PDO valid.								
Confirmation data written									←	581	60	00	18	00	00	00	00	00
80									→	SYNC								
9000 response, bytes 0 to 8 = ASCII pressure = "0.002"									←	181	30	2E	30	30	32	20	20	20
80									→	SYNC								
Example for Temperature																		
601	22	00	1A	0	0	0	0	0	→	Write zero to the mapping object 1A00, sub-index 0 to clear the existing PDO map.								
Confirmation data written									←	581	60	00	1A	00	00	00	00	00
601	22	00	1A	1	40	2	30	51	→	Write to sub-index 1 with object number, sub-index & length of the AI_Process Value object for temperature.								
Confirmation data written									←	581	60	00	1A	1	00	00	00	00
601	22	00	1A	0	1	00	00	00	→	Write to sub-index 0 with the number of objects written.								
Confirmation data written									←	581	60	00	1A	0	00	00	00	00
601	22	00	18	1	81	01	00	80	→	Write to the PDO communications object, sub-index 1 to set the PDO COB-ID & make the PDO valid.								
Confirmation data written									←	581	60	00	18	00	00	00	00	00
80									→	SYNC								
9000 response, bytes 0 to 8 = ASCII pressure = "22.5"									←	181	32	32	2E	35	20	20	20	20

4.3 Changing the Engineering Units

Object 6131 controls the physical units of the process value. When the units are changed then the 9000 automatically applies the units conversion to the values of all objects that are related to the process value, these related objects are identified in table 9.

Table 9
Objects Affected by Unit Changes

Object	Parameter
4130	AI_Maximum
4131	AI_Minimum
5130	AI_Process_Value String
6124	AI_Input_Offset
6130	AI_Process_value Real32
6132	AI_Decimal_Digits
6148	AI_Span_Start
6149	AI_Span_End
7130	AI_Process_Value Integer

The units of pressure are changed by writing to sub-index 1 of object 6131, whilst the units of temperature are changed by writing to sub-index 2. For pressure an additional object 2032 is provided to support non-SI units. If a non-SI pressure unit is required then it is necessary to write the floating point number value that represents the multiplier for the required unit, as shown in equation 1, to object 2032. Table 10 shows the SI units supported, whilst Figure 34 shows an example of changing units.

Equation 1 : Pressure Units Conversion

NewUnit * value = 1 bar

Figure 34
Changing Units

PC / Host Application									Direction	9000								
ID	**Byte 0**	**Byte 1**	**Byte 2**	**Byte 3**	**Byte 4**	**Byte 5**	**Byte 6**	**Byte 7**		**ID**	**Byte 0**	**Byte 1**	**Byte 2**	**Byte 3**	**Byte 4**	**Byte 5**	**Byte 6**	**Byte 7**
Example for Pressure SI Unit																		
601	22	31	61	01	00	00	22	02	→	Write to Object containing Physical Units of Process Value, setting units to hPa.								
Confirmation data written									←	581	60	31	61	01	00	00	00	00
Example for non SI Unit																		
601	22	32	20	00	66	34	8D	3D	→	Write to user pressure units object with multiplier required for psi.								
Confirmation data written									←	581	60	32	20	00	00	00	00	00
601	22	31	61	01	00	00	A0	00	→	Write to Object containing Physical Units of Process Value, setting units to user scaling, i.e. object 2032.								
Confirmation data written									←	581	60	31	61	01	00	00	00	00
Example for Temperature Units																		
601	22	32	61	02	00	00	A0	00	→	Write to Object containing Physical Units of Process Value, setting temperature units to non-SI, °F.								
Confirmation data written									←	581	60	32	61	02	00	00	00	00

Table 10
Supported Units

Unit	CANOpen Value (b7-b0 b15-b8 b23-b16 b31-b24)
bar	00 00 4E 00
mbar	00 00 4E FD
Pa	00 00 22 00
HPa	00 00 22 02
KPa	00 00 22 03
MPa	00 00 22 06
Non-SI	00 00 A0 00
°C	00 00 2D 00

4.4 Configuring the Transmission of Readings

The 9000 continually samples the pressure and temperature signals. These signals are used to calculate the compensated pressure and temperature readings. Three different methods are available for obtaining these reading via the PDO:

1) The PDO is transmitted with every n^{th} SYNC pulse received, where n is a value between 1 and 240.

2) The PDO is transmitted when a SYNC pulse is received, providing the 9000 event timer has expired.

3) The PDO is transmitted when the 9000 event timer expires. The transmission is not synchronous, that is it does not require a SYNC pulse.

The methods of PDO transmission are controlled by the transmission type and event timer, located in the PDO Communications Object (1800), sub-indexes 2 and 5 respectively. Figure 35 shows examples of setting up the different transmission types.

Figure 35
Setting up PDO Transmission

PC / Host Application									Direction	9000								
ID	**Byte 0**	**Byte 1**	**Byte 2**	**Byte 3**	**Byte 4**	**Byte 5**	**Byte 6**	**Byte 7**		**ID**	**Byte 0**	**Byte 1**	**Byte 2**	**Byte 3**	**Byte 4**	**Byte 5**	**Byte 6**	**Byte 7**
Example for Synchronous Transmission on Every 5th SYNC Object																		
601	22	00	18	02	05	00	00	00	→	Write to PDO Communications Object with transmission type of 5.								
Confirmation data written, PDO will now be transmitted on the reception of the 5th SYNC message.									←	581	60	00	18	02	00	00	00	00
Example for Synchronous Transmission Combined with Event Timer																		
601	22	00	18	05	88	13	00	00	→	Write to PDO Communications Object Event Timer with value of 5000 mS.								
Confirmation data written									←	581	60	00	18	05	00	00	00	00
601	22	00	18	02	00	00	00	00	→	Write to PDO Communications Object with transmission type of 0.								
Confirmation data written, PDO will now be transmitted on the reception of a SYNC message, providing 5 seconds has expired.									←	581	60	31	61	01	00	00	00	00
Example for Asynchronous Transmission																		
601	22	00	18	05	88	13	00	00	→	Write to PDO Communications Object Event Timer with value of 5000 mS.								
Confirmation data written									←	581	60	00	18	05	00	00	00	00
601	22	00	18	02	FE	00	00	00	→	Write to PDO Communications Object with transmission type of 254.								
Confirmation data written, PDO will now be transmitted at the expiry of the event time, that is 5 second intervals.									←	581	60	00	18	02	00	00	00	00

4.5 <u>Adding an Offset to the Measurements</u>

The pressure or temperature reading can be offset by a value using the analogue input offset object, 6124. The offset is always applied in the units currently set on the 9000, the units can be queried by reading object 6131. Figure 36 shows an example of off-setting pressure and temperature.

Figure 36
Off-Setting the Pressure and Temperature

PC / Host Application									Direction	9000								
ID	**Byte 0**	**Byte 1**	**Byte 2**	**Byte 3**	**Byte 4**	**Byte 5**	**Byte 6**	**Byte 7**		**ID**	**Byte 0**	**Byte 1**	**Byte 2**	**Byte 3**	**Byte 4**	**Byte 5**	**Byte 6**	**Byte 7**
Off Setting the Pressure Reading by 10 bar																		
601	40	31	61	01	00	00	00	00	→	Read the units of pressure from object 6131, sub-index 1.								
Units = 004E0000 = bar									←	581	42	31	61	01	00	00	4E	00
601	22	24	61	01	00	00	20	41	→	Write to off set object with floating point value of 10.0.								
Confirmation data written, 9000 will now add 10 to the pressure reading.									←	581	60	31	61	01	00	00	00	00
Off Setting the Temperature Reading by 10°C																		
601	40	31	61	02	00	00	00	00	→	Read the units of temperature from object 6131, sub-index 2.								
Units = 002D0000 = °C									←	581	42	31	61	02	00	00	2D	00
601	22	24	61	02	00	00	20	41	→	Write to off set object with floating point value of 10.0.								
Confirmation data written, 9000 will now add 10 to the temperature reading.									←	581	60	31	61	02	00	00	00	00

4.6 Advanced Functions

The 9000 supports a wide variety of further functionality via SDO access to it's object dictionary. Appendix 1 details all of the objects supported.

4.6.1 Access Rights for the Object Dictionary

All objects are accessible to the user, however the objects located in the experienced user area can not be written to unless the correct PIN is first written to object 20FF.

4.6.2 Detecting Overflow and Underflow Conditions

Objects 2006, 2007, 2008 and 2009 provide a count value indicating the number of times the process value has exceeded the limits determined by the span start and span end of the 9000. The span start and span end objects (6148 and 6149) may be written by the user allowing any system limits to be used. The overflow/underflow count value is reset by writing "zero" to the overflow control object 2803.

4.6.3 Network Initialisation

The 9000 can be programmed to enter the operational NMT state when excitation is applied, without transmitting a boot up message. This allows previously configured devices to be installed on a network without the need for any NMT commands. To skip the boot up then 0 must be written to object 2813.

4.6.4 Boot Loading

The 9000 is in system programmable, allowing software update in the field. In order to re-program the 9000 then the Atmel FLIP program is required, along with an appropriate CAN interface. The method for re-programming is :-

1. Write to object 2813, the string ‘boot’.
2. Remove / reapply the 9000 excitation.
3. Start the Atmel FLIP programming tool and select microcontroller T89C51CC01.
4. Select the appropriate CAN interface and connect to the node.
5. Load the new software, a hex file provided by Gems Sensors.
6. Blank check, program and verify the 9000.
7. Clear the BSB and write to the special status registers.
8. Disconnect the node, the 9000 is re-programmed.

5. MAINTENANCE AND SUPPORT

The 9000 provides excellent long term stability. To facilitate a planned calibration schedule then the 9000 provides access to the next calibration date and last calibration date.

5.1 Calibration

5.1.1 Reading the Calibration Dates

To determine if a calibration is required then it is necessary to read the next calibration date as shown in the Figure 37.

Figure 37
Querying the Calibration Dates

PC / Host Application									Direction	9000								
ID	**Byte 0**	**Byte 1**	**Byte 2**	**Byte 3**	**Byte 4**	**Byte 5**	**Byte 6**	**Byte 7**		**ID**	**Byte 0**	**Byte 1**	**Byte 2**	**Byte 3**	**Byte 4**	**Byte 5**	**Byte 6**	**Byte 7**
601	40	00	28	01	00	00	00	00	→	Read Object 2800, sub-index 1 – last calibration day and month.								
Calibration day 28, Calibration Month 01									←	581	42	00	28	01	32	38	30	31
601	40	00	28	02	00	00	00	00	→	Read Object 2800, sub-index 2 – last calibration year.								
Calibration Year 2002. Therefore the last calibration was 28th January 2002									←	581	42	00	28	02	32	30	30	32
601	40	01	28	01	00	00	00	00	→	Read Object 2801, sub-index 1 – next calibration day and month.								
Calibration day 28, Calibration Month 01									←	581	42	00	28	01	32	38	30	31
601	40	01	28	02	00	00	00	00	→	Read Object 2800, sub-index 2 – next calibration year.								
Calibration Year 2003. Therefore the next calibration is due on 28th January 2003									←	581	42	00	28	02	32	30	30	33

5.1.2 Performing a Calibration

The calibration of the 9000 should be periodically measured against a known reference standard. The 9000 allows for adjustment of the zero point, the stability of the sensitivity of the sensor is such that gain adjustments are not necessary. An example of zero correction is shown in Figure 38.

Figure 38
Correcting the Zero Point

PC / Host Application									Direction	9000								
ID	**Byte 0**	**Byte 1**	**Byte 2**	**Byte 3**	**Byte 4**	**Byte 5**	**Byte 6**	**Byte 7**		**ID**	**Byte 0**	**Byte 1**	**Byte 2**	**Byte 3**	**Byte 4**	**Byte 5**	**Byte 6**	**Byte 7**
601	22	24	61	01	00	00	00	00	→	Write 0 to Object 6124, sub-index 1 to ensure that no offset is being applied to the pressure reading.								
Confirmation of Data Written.									←	581	60	24	61	01	00	00	00	00
APPLY ZERO PRESSURE																		
601	22	25	61	01	7A	65	72	6F	→	Write "zero" to Object 6125, sub-index to perform an auto zero on the pressure reading.								
Confirmation zero has been applied. The 9000 will apply a correction to make the pressure reading 0.									←	581	60	25	61	01	00	00	00	00
601	22	10	10	01	73	61	76	65	→	Write "save" to Object 1010, sub-index 1 to store the new zero point.								
Confirmation Save has been done.									←	581	60	10	10	01	00	00	00	00
601	22	FF	20	00	PIN				→	Write to Object 20FF to provide experienced user access to the object dictionary.								
Confirmation of Data Written									←	581	60	FF	20	00	00	00	00	00
601	22	00	28	01	32	38	30	31	→	Write Object 2800, sub-index 1 with the last calibration date; ddmm; e.g. '2801'								
Confirmation of Data Written									←	581	60	00	28	01	00	00	00	00
601	22	00	28	02	32	30	30	33	→	Write Object 2800, sub-index 2 with the last calibration year; yyyy; e.g. '2003'								
Calibration Year 2002. Therefore the last calibration was 28th January 2002									←	581	60	00	28	02	00	00	00	00
601	22	01	28	01	32	38	30	31	→	Write Object 2801, sub-index 1 with the next calibration date; ddmm; e.g. '2801'								
Confirmation of Data Written									←	581	60	01	28	01	00	00	00	00
601	22	01	28	02	32	30	30	34	→	Write Object 2801, sub-index 2 with the next calibration year; yyyy; e.g. '2004'								
Confirmation of Data Written									←	581	60	01	28	02	00	00	00	00

Note: PIN is the pin number required for write access to the object dictionary in the experienced user area, refer to 5.6.1.

APPENDIX 1

9000 OBJECT DICTIONARY

Object	COB-ID	Sub Index	Data Type	Access
COMMUNICATIONS PROFILE OBJECTS 1000-1FFF				
Device Type	1000	0	UNSIGNED32	RD
Error Register	1001	0	UNSIGNED8	RD
Pre-Defined Error Field	1003	0	UNSIGNED8	RD/WR
		1	UNSIGNED32	RD
		2	UNSIGNED32	RD
		3	UNSIGNED32	RD
		4	UNSIGNED32	RD
Manufacturer	1009	0	STRING	RD
Store Parameters	1010	0	UNSIGNED32	RD
		1	UNSIGNED32	RD/WR
Restore Default Parameters	1011	0	UNSIGNED32	RD
		1	UNSIGNED32	RD/WR
Producer Time /	1017	0	UNSIGNED16	RD/WR
Identity Object	1018	0	UNSIGNED8	RD
		1	UNSIGNED32	RD
		2	UNSIGNED32	RD
		3	UNSIGNED32	RD
		4	UNSIGNED32	RD
Server SDO Comms Parameter	1200	0	UNSIGNED8	RD
		1	UNSIGNED32	RD
		2	UNSIGNED32	RD
Tx PDO Comms Parameter	1800	0	UNSIGNED8	RD
		1	UNSIGNED32	RD/WR
		2	UNSIGNED8	RD/WR
		3	Not Supported	
		4	Not Supported	
		5	UNISGNED16	RD/WR
TX PDO Mapping Parameter	1A00	0	UNSIGNED8	RD/WR
		1	UNSIGNED32	RD/WR
		2	UNSIGNED32	RD/WR
		3	UNSIGNED32	RD/WR
		4	UNSIGNED32	RD/WR
MANUFACTURER SPECIFIC PROFILE – USER AREA 2000-27FF				
Manufacturer	2000	0	STRING	RD
Manufacture Date	2001	0	UNSIGNED8	RD
		1	STRING	RD
		2	STRING	RD
Pressure Datum	2002	0	CHAR	RD
Software Number	2003	0	STRING	RD
Gems Calibration	2004	0	UNSIGNED8	RD
		1	STRING	RD
		2	STRING	RD
		3	REAL32	RD
		4	REAL32	RD
		5	REAL32	RD
		6	REAL32	RD
		7	REAL32	RD
		8	REAL32	RD
		9	REAL32	RD

Object	COB-ID	Sub Index	Data Type	Access
		A	REAL32	RD
		B	REAL32	RD
		C	REAL32	RD
		D	REAL32	RD
		E	REAL32	RD
		F	REAL32	RD
		10	UNSIGNED32	RD
Seconds in	2005	0	UNSIGNED32	RD
Pressure Span	2006	0	UNSIGNED16	RD
Pressure Span Underflow	2007	0	UNSIGNED16	RD
Temperature Span Overflow	2008	0	UNSIGNED16	RD
Temperature Span Underflow	2009	0	UNSIGNED16	RD
Calibrated Minimum	200A	0	REAL32	RD
Calibrated Maximum Pressure	200B	0	REAL32	RD
Analogue Input Process Value Filter	2030	0	UNSIGNED8	RD
		1	REAL32	RD/WR
		2	REAL32	RD/WR
Maximum & Minimum Control	2031	0	UNSIGNED8	RD
		1	UNSIGNED32	WR
		2	UNSIGNED32	WR
Non SI Units Conversion	2032	0	REAL32	RD/WR
Object Access	20FF	0	UNSIGNED32	WR
MANUFACTURERS PROFILE AREA- EXPERIENCED USER AREA 2800-2FFF				
Last Calibration Date	2800	0	UNSIGNED8	RD
		1	STRING	RD/WR
		2	STRING	RD/WR
Next Calibration date	2801	0	UNSIGNED8	RD
		1	STRING	RD/WR
		2	STRING	RD/WR
Update Rate	2802	0	UNSIGNED32	RD/WR
Overflow Control	2803	0	UNSIGNED8	RD
		1	STRING	RD/WR
		2	STRING	RD/WR
Boot Load	2810	0	UNSIGNED32	WR
Network	2813	0	UNSIGNED8	RD/WR
Calculations	2814	0	UNISGNED8	RD/WR
MANUFACTURERS PROFILE AREA – MAPABLE OBJECTS 4000-4FFF				
Analogue Input Process Value Maximum	4130	0	UNSIGNED8	RD
		1	REAL32	RD
		2	REAL32	RD
Analogue Input Process Value Minimum	4131	0	UNSIGNED8	RD
		1	REAL32	RD
		2	REAL32	RD

Object	COB-ID	Sub Index	Data Type	Access
Analogue Input Process Value String	5130	0	UNSIGNED8	RD
		1	STRING	RD
		2	STRING	RD
CANOpen PROFILE – DSP404 6000-9FFF				
Analogue Input Input Offset Value	6124	0	UNSIGNED8	RD
		1	REAL32	RD/WR
		2	REAL32	RD/WR
Analogue Input Auto Zero	6125	0	UNSIGNED8	RD
		1	UNSIGNED32	WR
		2	UNSIGNED32	WR
Analogue Input Process Value	6130	0	UNSIGNED8	RD
		1	REAL32	RD
		2	REAL32	RD
Physical Units of Process Value	6131	0	UNSIGNED8	RD
		1	UNSIGNED32	RD/WR
		2	UNSIGNED32	RD/WR
Process Value Decimal Digits	6132	0	UNSIGNED8	RD
		1	UNSIGNED8	RD/WR
		2	UNSIGNED8	RD/WR
Process Value Span Start	6148	0	UNSIGNED8	RD
		1	REAL32	RD/WR
		2	REAL32	RD/WR
Process Value Span End	6149	0	UNSIGNED8	RD
		1	REAL32	RD/WR
		2	REAL32	RD/WR
Analogue Input Field Value	7100	0	UNSIGNED8	RD
		1	UNSIGNED16	RD
		2	UNSIGNED16	RD
		3	UNSIGNED16	RD
Analogue Input Process Value	7130	0	UNSIGNED8	RD
		1	SIGNED16	RD
		2	SIGNED16	RD

REVISION HISTORY SHEET

ISSUE	DATE	PARA.		REVISED
A	March 2003		NEW ISSUE	