ICM-CANBUS

Inline Contamination Monitor

CAN-Bus Guide

MPFILTRI®

www.mpfiltri.co.uk

200.072-EN

Document Revision 0.7

1 Introduction

1.1 Overview

The ICM measures and quantifies the numbers of solid contaminants in Hydraulic, Lubrication and Transmission applications. The ICM is designed to be an accurate instrument for permanently installed applications utilising mineral oil as the operating fluid.

The unit can operate using any of the international standard formats ISO 4406:1999, NAS 1638, AS 4059E and ISO 11218.

The ICM incorporates a CAN-bus interface so that it can be connected to an existing CAN-bus network. This allows the ICM to be easily integrated into industrial, mobile, agricultural and military applications.

1.2 CAN-bus Standards

The ICM supports the major CAN-bus basic message format standards CAN 2.0A (11 bit identifiers) and CAN2.0B (29 bit identifiers).

J1939 and CanOpen are higher level protocols built on these basic standards. J1939 uses CAN2.0A, and CanOpen uses CAN2.0B. The ICM does not implement either of these protocols. Instead it defines a few CAN-bus messages to communicate data. However the message identifiers have been chosen so as to allow operation with both J1939 and CanOpen. Generally it should be possible to use the ICM with either, as well as other CAN-bus systems.

1.3 Installation Outline

- Perform a general installation and check of the ICM as detailed in the main product user guide.
- Perform a one-off general configuration check of the ICM using a PC running LPA-View[1]. This procedure is described in the main ICM user guide. You will need a suitable RS485 interface, such as the ICM-USBi.

[1] For example set it to test continuously and to automatically start testing on power-up

- Use the software to configure any CAN-bus specific parameters required by your CAN-bus network.[2]
- Connect the ICM to your CAN-bus network and provide a 24VDC power supply. See Figure 1.
- The ICM automatically emits the test result message after each set test interval.
- Configure your CAN-bus controller to listen for the messages configured above.

[2] For example configure a CAN-bus message ID and baud rate,

2 Electrical Connection

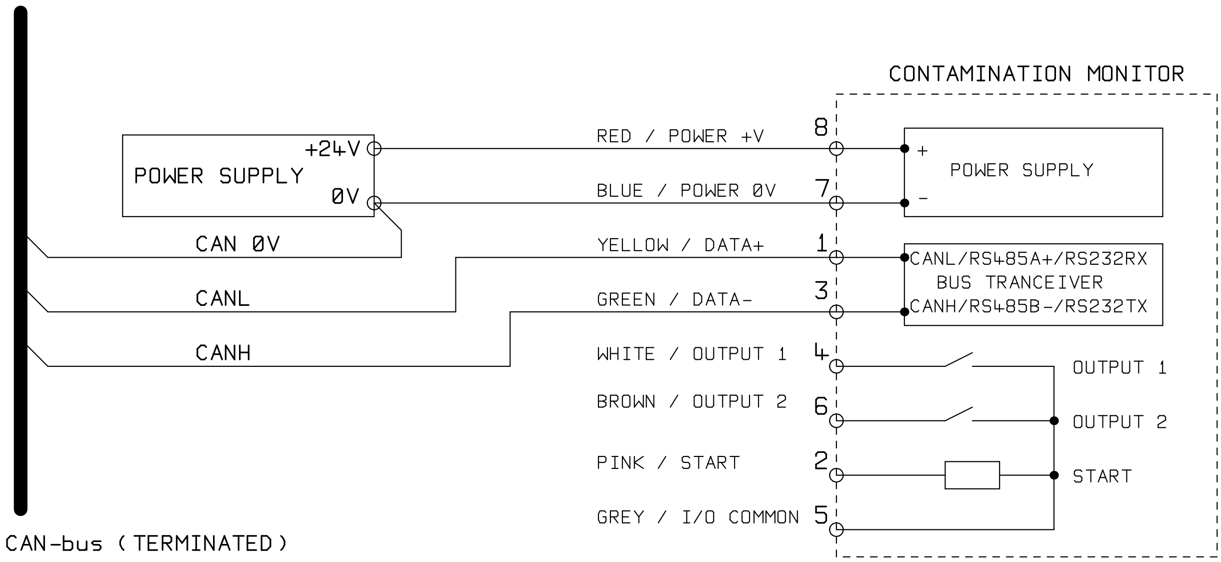

Figure 1 ICM Electrical Connection

The ICM requires a DC power supply and the two CAN-bus signals CANL and CANH, as shown in Figure 1. The numbers shown are the pin numbers of the circular connector that plugs into the ICM.

- The DC voltage is typically 24V, but can be 9-36V. See the ICM user guide for the precise range.
- Twisted pair cable should be used for the CAN-bus signals, for cable lengths over a few meters. The cables supplied with the ICM are twisted pair.
- CAN-bus *requires* the network to be terminated at each end. This must be done externally to the ICM.
- The CAN-bus signals CANL and CANH are referred to the system 0V supply. These should stay within the common mode range allowed by the ISO-11898-4 CAN-bus standard relative to the ICM 0V connection. This range is -2V to +7V. This can be normally be ensured by connecting together the ICM 0V and the 0V of the CAN-bus controller. The ``CAN 0V'' wire shown indicates this link.

(Not needed if both CAN-bus controller and ICM are connected to a vehicle chassis or otherwise ``Earthed''.)

- There are other wires available for switched alarm and start signals (optional). These are documented separately in the ICM user guide.

3 Configuration

3.1 Use PC Software for Configuration

The free LPA-View software package is needed in order to initially configure the ICM. Once configured, the unit can be left connected to the CAN-bus network.

The ICM was designed to be as flexible as possible. There are large number of options for setting operating modes, test result formats, alarm settings, downloading stored data etc.

The easiest approach is to use LPA-View to configure the test parameters and result format. Then the customer application only has to read the results.

The CAN-bus parameters are configured from the Communication Settings dialogue, accessed from the ICM Settings dialogue (see the main ICM user guide for more details).

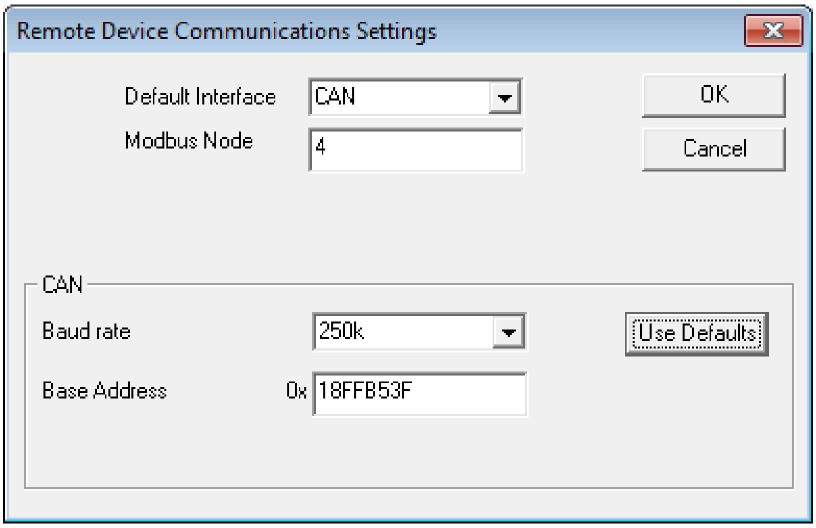

Figure 1 Communication Settings dialogue

The ICM can use either CAN 2.0A ``basic'' format with 11 bit identifiers, or CAN 2.0B ``extended frame'' format with 29 bit identifiers.

3.2 CAN 2.0B and J1939

The default 29 bit format is designed to be compatible with the J1939 standard. It should also be possible to use the ICM with any system that permits arbitrary raw CAN-bus 2.0B identifiers to be received.

The dialogue shows the default CAN-bus settings. The ICM transmits all data using a range of message identifiers starting at that selected. If the ``Use Defaults'' button is pressed, the program constructs an identifier suitable for J1939[3].

At the end of each test, the ICM will generate a ``test result codes'' message using the selected CAN-bus identifier.

On a J1939 network the test result will appear as PGN 0x00ff00. Users not using J1939 can simply listen for messages with the identifier shown in the dialogue, e.g. 0x18FFB53F.

3.3 CAN 2.0A and CanOpen

The 11 bit format is designed to be compatible with the CanOpen standard. It can also be used with any system that allows raw CAN-bus 2.0A identifiers to be received.

In order to use 11 bit identifiers (CAN 2.0A) set a value below 0x7ff for the ``Base Address''.

For a CanOpen network, use a base address of 0x182 for example. This will result in message IDs corresponding to the CanOpen ``pre-defined connection set''.

[3] i.e. This will use PGNs within the region allocated to proprietary applications, starting at 0x00FFB53F

4 CAN-bus Operations

4.1 CAN-bus Settings

CAN-bus physical layer	ISO-11898-2[4]
Protocol type	CAN 2.0B (29 bit identifiers)
	CAN 2.0A (11 bit identifiers)
Baud	User Set 1M/800k/500k/250k/125k/100k/50k/20k/10k
Identifier Range	User Set

4.2 Operation

Typically the installer will have configured the ICM to automatically start continuous testing. At the end of each set test interval (e.g. 2 minutes) the ICM will emit a CAN ``Result Codes'' message using the set CAN identifier (e.g. 0x18FFB53F using hexadecimal notation). So a typical CAN message might be:

	Byte							
Identifier	1	2	3	4	5	6	7	8
0x18FFB53F	12	8	2					

Details of the CAN-bus messages supported are in Appendix B.

[4] This is by far the most commonly used standard, as used in automotive and industrial applications

Example Walk-through

Real applications will generally have an existing CAN network, but in this chapter we show how the ICM can be connected to a PC using a USB:CAN adaptor.

The adaptor used in this example is the ``PCAN-USB'', available from Peak System Technik GmbH or a distributor.

We also need to make up a special cable to connect this to the ICM.

Figure I PCAN-USB CAN-bus to USB Adaptor

Equipment Required

- ICM with CAN-bus capability
- PCAN-USB USB:CAN Adaptor
- ICM-USBi interface for initial ICM setup
- PC with USB ports running Windows
- Special made-up CAN-bus cable detailed below
- 12 or 24 Volt DC power supply

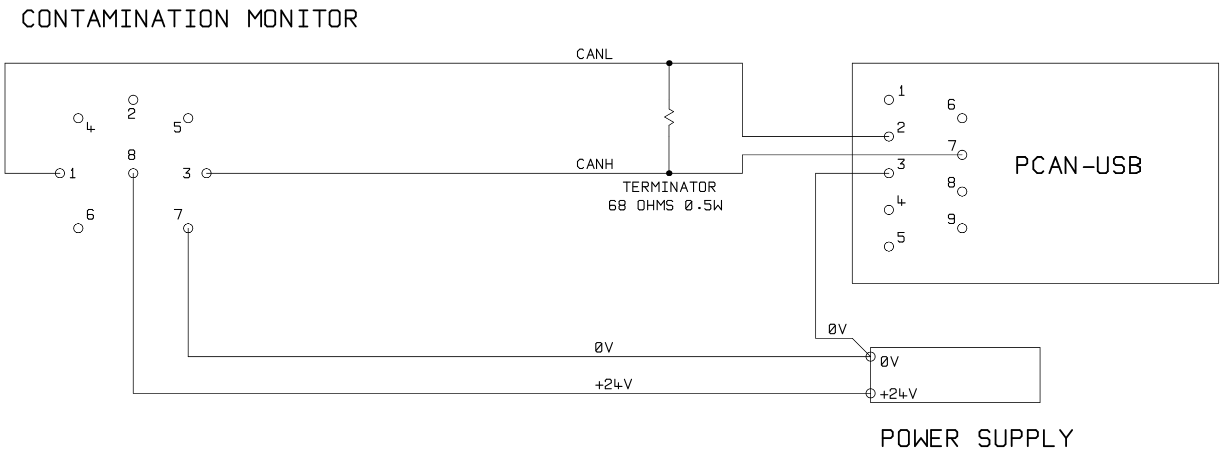

Figure II Connecting the ICM to the PCAN-USB Adaptor

The ``TERMINATOR'' resistor shown simulates the combined effect of the bus termination resistors normally used on either end of a CAN-bus network. It's value is not critical, anything from 50-150 ohms will work.

Initial Configuration

Initially we connect using the ICM-USBi interface so that the ICM can be comfortably configured using LPA-View. Detailed information is provided in the user guides but the general procedure is:

- Install LPA-View
- Plug in the ICM-USBi
- The ``Hardware Found'' wizard will appear. If you have an Internet connection you can let Windows Update install the driver, otherwise point the wizard to the drivers provided.
- Plug the ICM into the ICM-USBi
- Start LPA-View
- Select Tools/Remote Control to connect to the ICM.

Suggested General Settings

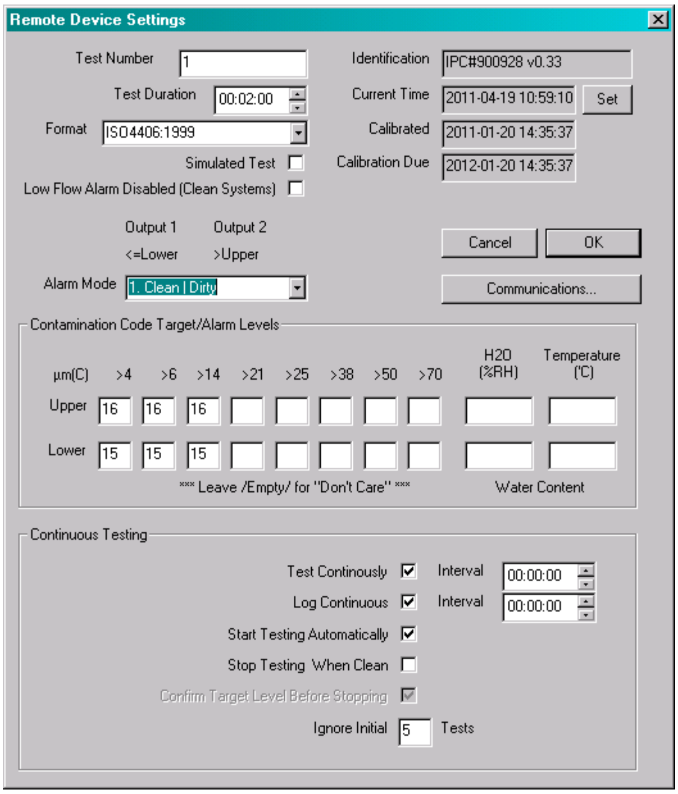

Figure III General Settings

Press the Settings button to open the Settings dialogue. The important settings for this walk-through are:

- Test Duration; 10 seconds
- Test Continuously: On, interval 0.
- Start Testing Automatically: On
- Stop Testing When Clean: Off
- Simulate Test: On[V]

Suggested Communication Settings

Press the ``Communications...'' button to open the Communications dialogue.

[V] This will cause fictitious test results to be generated in order to test communications and demonstrate the unit. Do not forget to turn this off before deploying in a real application!

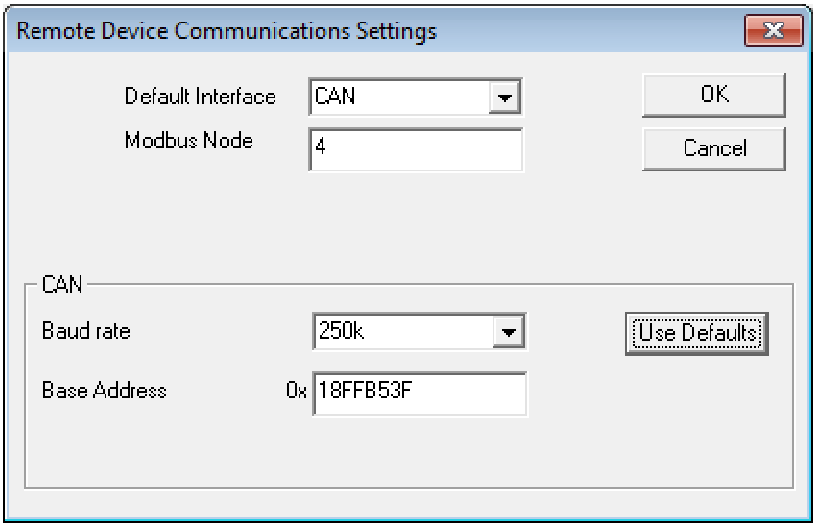

Figure IV Communication Settings

Select the Interface, Node Number and Baud Rate as shown, then press ``Use Defaults'' to assign the ``base address''. This will define the start of the block of CAN message identifiers used by the software (using a value compatible with the J1939 standard).

Press the OK button on the ``Communications Settings'' and ``Remote Device Settings'' dialogues. Leave the Remote Control dialogue open.

Now check that the ICM is now set to automatically perform tests:

- Unplug the ICM circular connector
- Plug it back in
- You should see the connection re-established on the Remote Control dialogue within a few seconds.
- A test should have been automatically started
- The tests should repeat every 10 seconds
- You should see a test result that starts high and decreases with each further test.

Close the Remote Control dialogue and quit the program. Unplug the ICM at the circular connector.

PCAN-USB Software

The PCAN-USB adaptor comes with a software CD. This includes a simple CAN-bus diagnostic utility called ``PCAN-View USB''. This should be installed from the CD.

Connect the ICM to the computer using the special made-up cable and the PCAN-USB. Power-up the ICM by turning on the power supply.

Upon connecting the PCAN-USB and starting PCAN-View, the Connect dialogue is presented.

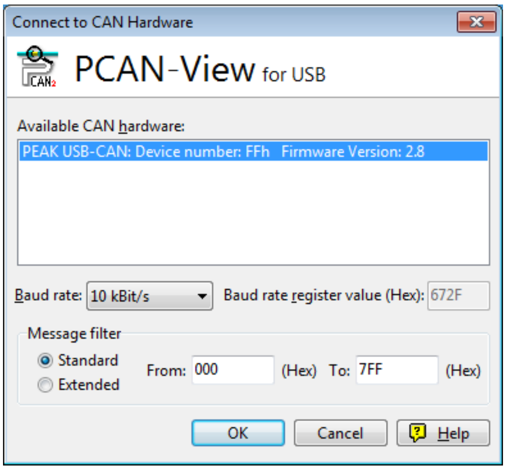

Figure V PCAN-View Connect dialogue

Select a baud rate to match that being used on the ICM, for example 250k. Select the ``Extended'' message filter (so that 29 bit identifiers are used). Press OK to go to the main PCAN-View screen.

Simulated Tests

Plug the ICM into its circular connector. It should power up and start performing a test.

If everything is working, after about 20 seconds you should see CAN messages similar to that shown below.

This shows the 2nd result received. The first 3 bytes 0x17, 0x15, 0x13 show the 3 ISO codes[VI]

[VI] The display is in hexadecimal (base 16) so the actual code are 23/21/19

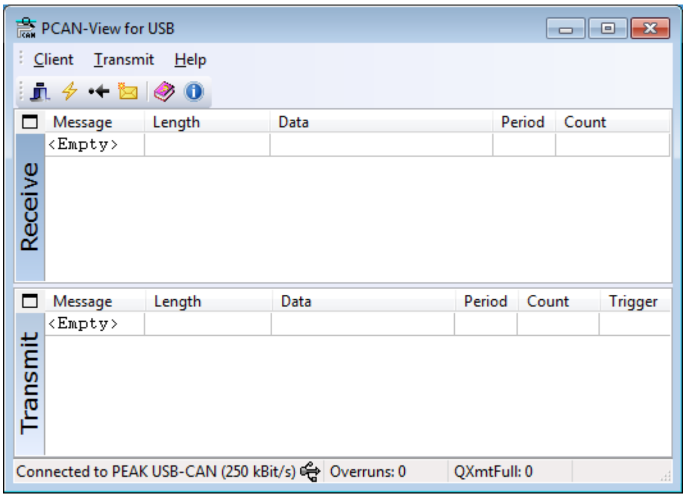

Figure VI PCAN-View Main Screen

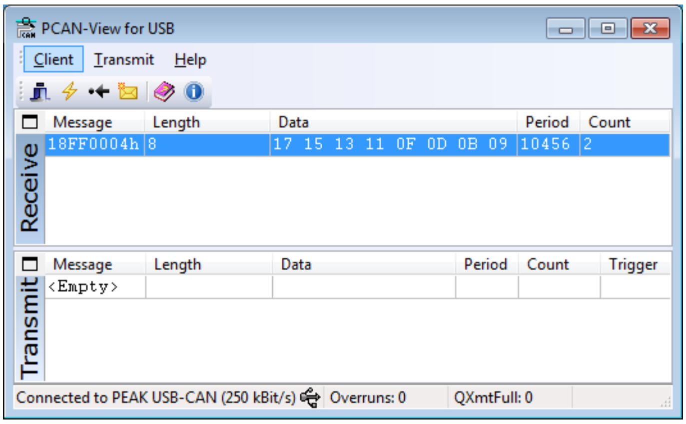

Figure VII Reception of a Test Result Codes Message

Messages

CAN2.0B and J1939

The ICM CAN-bus implementation is designed to be interoperable with J1939 networks. This is done by restricting CAN-bus message IDs to those within the proprietary ranges allocated by J1939. Advanced J1939 features have been avoided, so that customers not using J1939 will also be able to communicate using ``generic'' CAN-bus frames. For non-J1939 users the only requirement is that their network should support CAN2.0 (29 bit identifiers).

Broadcast messages use the J1939 PDU2 format. These are transmitted periodically to communicate the ICM status and the latest test results.

Peer-to-peer messages use the J1939 PDU1 format. These are used to *control* the ICM. These are generally optional; customers may opt to leave the ICM automatically testing and broadcasting results.

J1939 Parameters

Node Address (PDU1)	0x3F (J1939 ``Oil Sensor'')
Command & Configuration Message PGN	0xEF3F
Broadcast Messages PGNs	0xFFB5 - 0xFFB9
Default Broadcast Interval	1s
Data page	0
Priority	6
PDU format / PDU specific	Derived from PGN
Byte Endianness	All data is in little-endian byte order

Table I CAN-bus Parameters for J1939 Interoperation

Non-J1939 CAN2.0B Users

- Taken together these imply a generic CAN "base address" of 0x18FFB53F.
- Command and control messages can then be sent to CAN address 0x18EF3F00.

CAN2.0A and CanOpen

On CanOpen networks the results need to be transmitted as ``process data objects'' (PDOs) from the ``predefined connection set''. In order to do this, ensure that the set base address is equal to (0x180 + node number). For example, 0x182 to make the ICM node address 2.

CAN-bus Message List

The message ID numbers shown are examples only and are dependent on the set base address.

For CAN2.0A/CanOpen we have an example base address of 0x182. So you see ``2'' as the last digit of all the message IDs. CanOpen interprets this as the device node number.

For CAN2.0B/J1939 we have an example base address of 0x18FFB53F. The equivalent node number is ``3F'' so you see this appear in all the message IDs.

For other node numbers change the set base address value as required. CanOpen has node numbers from 0x01 to 0x7f. J1939 has node numbers from 0x01 to 0xff.

Parameter Name	CAN2.0A ID	CanOpen PDO	CAN2.0B ID	J1939 PGN
Result Codes	0x182	Transmit PDO 1	0x18FFB53F	0xFFB5
Status	0x282	Transmit PDO 2	0x18FFB63F	0xFFB6
Water Sensor	0x382	Transmit PDO 3	0x18FFB73F	0xFFB7
Commands	0x202	Receive PDO 1	0x18EF3F00	PDU1

Table II CAN-bus Messages

Message: Result Codes

This message is transmitted after each test.

The test result is expressed as a set of codes *in the selected Test Format (ISO4406, NAS1638 etc)*. The test result message is always 8 bytes long, with the result codes packed as follows:

Format:	ISO4406	AS4059E2	NAS1638/AS4059E1/ISO11218[VII]
Byte	Code	Class	Class
1	4µm(c)	Basic[VIII]	Basic
2	6µm(c)	-	-
3	14µm(c)	A	5-15µm
4	21µm(c)[IX]	B	15-25µm
5	25µm(c)	C	25-50µm
6	38µm(c)	D	50-100µm
7	50µm(c)	E	>100µm
8	70µm(c)	F	-

Special Values

The result codes use a few ``special'' values in order to represent codes that are not simple numbers.

The NAS1638 standard defines classes ``00'' and ``000'', these are classes ``cleaner'' than class 0. We represent these using signed integers of value -1 and -2 respectively.[X]

VII NAS1638, AS4059E Table 1, and ISO11218 standards produce identical numerical codes so they are listed together here.

VIII The ``Basic'' class is the highest of the individual size classes

IX ISO4406 only defines codes for the first three sizes 4,6 and 14µm(c). We extend the concept to cover the other sizes. This allows limits to be set on the number of large particles, even when using the ISO 4406 coding system.

X These will appear as 255 and 254 if read as unsigned integers.

Message: Status

The message is transmitted every 1 second so that it can be used as a ``heartbeat''. However if no test has been performed yet, the ICM will wait until it sees other CAN-bus activity before sending anything.

Byte	Bit	Length	Type	Item
1-4	1	32	unsigned	Test Number
5	1	8	unsigned	Status code
6	1	8	unsigned	Completion
7-8	1	16	bitmask	Status Flags

Test Number The current Test Number is an auto-incremented integer or can also be set as part of the Test Start command. This is used to distinguish tests / circuits.

Status Code This is a number used to indicate the current state of the ICM, or a fault code in the case of a problem being detected. The codes are listed in Table III. This allows a system to remotely monitor the ICM operation, if desired, allowing more specific diagnostics.[XI]

Completion A number between 0 and 100 indicating the progress of the test. This will increase from 0 to 100 during the set test time. It can be used to drive a progress indicator.

Status Flags This is a group of flags indicating test status.

XI The fault conditions are also indicated on the front panel LED.

XII User has not set tests to occur automatically

XIII User has set a non-zero test interval

XIV Or fluid is totally clean (no particle counts). Flow alarm can be turned off by user if this is a problem, for example cleaning rigs.

Value	Function	Comment
0	NOT READY	Unit is powering-up, or there is some problem
1	READY	Ready to start a test[XII]
2	TESTING	Test in progress
3	WAITING	Waiting between tests[XIII]
128	FAULT OPTICAL	LED failure / sensor blocked / filled with air
129	FAULT FLOW LOW	Flow too low for reliable test[XIV]
130	FAULT FLOW HIGH	
131	FAULT LOGGING	Fault with data logging
132	FAULT WATER SENSOR	Water sensor failure

Table III The STATUS CODE Parameter

Status Flags Bitmask

This represents the states of various items in a bitmask format.

Bit	Function	Comment
0	RESULT_VALID	Current result is valid
1	RESULT_NEW	A new result is available
2	RESULT_LOG	Current result should be logged
3	TESTING	Test in progress
4	COMPLETE	Test is complete
5	ALM_HI_COUNT	High particle count alarm
6	ALM_HI_H2O	High water content alarm
7	ALM_HI_TEMP	High Temperature alarm
8	ALM_LO_COUNT	Low count alarm
9	ALM_LO_H2O	Low water content alarm
10	ALM_LO_TEMP	Low temperature alarm
11	REMOTE_CONTROL	Unit is under remote control
12	IO_IP	Start signal input
13	IO_OP1	Alarm output 1
14	IO_OP2	Alarm output 2
15	UNUSED	Not Currently Used

Table IV Status Flags

Bits 0-2 are so that external equipment (for example LPA-View or a PLC/MMI) can display, update and log results intelligently.

Bits 3 and 4 can be used to monitor the test progress.

Bits 5-10 are used to generate alarms. Depending on the selected alarm mode, they operate the alarm relay output(s). But they can also be monitored directly by a PLC/MMI program and used to drive indicators, for example.

Bit 11 is used internally to detect that the ICM is being controlled by modbus (from a PLC or by LPA-View).

Finally bits 12-14 reflect the state of the ICM ``start signal'' input and alarm output relays.

Message: Water Sensor

Byte	Bit	Length	Type	Item
1	1	8	unsigned	RH%
2	1	8	signed	Temperature degrees C

Message: Commands

Various commands can be sent to the ICM via CAN-bus .

For J1939 networks Peer-to-Peer (PDU1) Messages are used.

For CanOpen networks Receive Process Data Objects are used.

Command

Byte	Bit	Length	Type	Item
1	1	8	unsigned	Command Byte (0x00)
2	1	8	enum	Command (0,1,2...)
3-6	1	32	unsigned	Parameter

Commands

enum	Function	Parameter
1	Start Test	None
9	Stop Test	–
13(0xd)	Start Test	Fixed Test Number
14	Format ISO4406	Set ISO4406 result format
15	Format NAS1638	Set NAS1638 result format
16	Format AS4059_E2	Set AS4059E Table 2 result format
17	Format AS4059_E1	Set AS4059E Table 1 result format
18	Format ISO11218	Set ISO11218 result format

Produced by MP Filtri UK

Revision 0.7

As a policy of continual improvement, MP Filtri UK reserve the right to alter specifications without prior notice.