PLC « CAN interface

for the
ZEUS *MicroVertex Detector (MVD)*
Cooling System

Henk Boterenbrood
NIKHEF, Amsterdam
February 2000

USER DOCUMENTATION Version 1.4

ABSTRACT:
*This document describes the **CANopen** interface to the **CAN**-module that provides an interface to the **PLC** that controls the **ZEUS Microvertex Detector** (MVD) **cooling system**. The **SPICAN** CAN-module equipped with this application firmware is referred to as **PLC-CAN** in this note.*

Contents

1 Introduction

The CAN-fieldbus network that interconnects (a number of) the MVD control systems and the *MVD-Controls Supervisor* is used by the *Supervisor* to monitor and control the various MVD subsystems. One of the subsystems connected to the CAN network is the PLC-controlled cooling system of the MVD frontend-electronics.

The PLC (SIEMENS S7) of the MVD cooling system does not have an interface to connect it directly to a CAN-bus. But it does have an RS232 interface, which is used to connect it to a so-called **SPICAN** module which features an RS232 interface as well as a CAN interface. The SPICAN module can thus provides an interface between the PLC and the CAN network, as illustrated below in Figure 1. Custom application firmware has been developed for the SPI-CAN module to provide a communication interface between RS232 and CAN-bus.

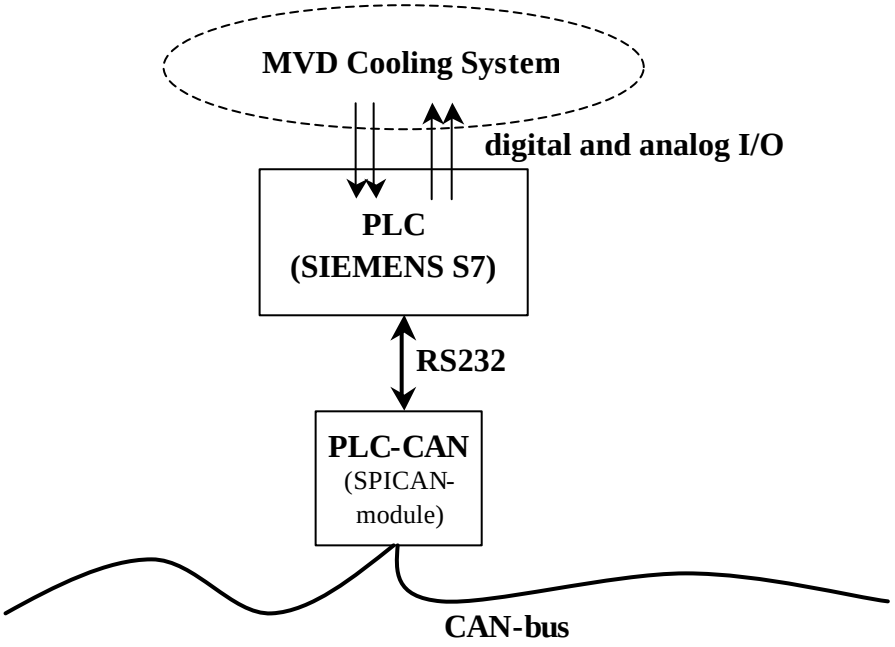


Figure 1. Interface of MVD Cooling System to CAN-bus.

A protocol has been agreed upon for transfer of data and commands via the RS232 connection between the application running on the SPICAN module (**PLC-CAN**) and the PLC application (see "
APPENDIX C Protocol on the PLC-CAN↔PLC" for a description). The **PLC-CAN** application maps messages to and from the PLC application onto appropriate CAN-messages. This document provides a detailed description of the interface to the PLC seen from the CAN-bus side.

The protocol used on the MVD CAN-network is ***CANopen*** ([1]). This document assumes that the user has at least a basic knowledge of this protocol.

2 Operation

The *MVD-Controls Supervisor* should be able to monitor the status of the MVD cooling system, to set a number of warning and error limits in the system and be able to assert general control over the PLC (such as: start / stop / reinitialise). All communication will take place using CAN-bus communication, i.e. the ***CANopen*** protocol, a standardized application layer protocol for CAN-bus ([1]).

All relevant PLC parameters are mapped by PLC-CAN to an object in the *CANopen* **Object Dictionary** (OD) in the "Manufacturer-Specific Profile Area", shown in **Table 3**.

By making the parameters accessible for reading and writing through the *CANopen* ***SDO*** (*Service Data Object*) mechanism and mapping appropriate parameters to *CANopen* ***PDO***s (*Process Data Objects*) a straightforward and *CANopen*-compliant interface to the cooling system PLC has been created.

One of the PLC-CAN's features is that it can autonomously monitor the PLC for status changes, which –if they occur– result in a CAN-message.

2.1 Initialisation

After power-up, watchdog reset, manual reset or *CANopen* initiated reset actions the PLC-CAN node sends a so-called *Bootup* message (defined by the *CANopen* standard) as soon as it has finished its initialization; this is a CAN-message with the following syntax:

PLC-CAN (NMT-Slave) ® Host (NMT-Master)

COB-ID	Byte 0
0x700 + *Node_ID*	0

In case of a watchdog or manual reset the Bootup message is followed by a *CANopen* Emergency message, as listed in the table in section 2.8.

2.2 PLC Parameter Block

The PLC parameters accessible via the CAN-bus are listed under Object Dictionary object 0x2000 in **Table 3**. The descriptions of the parameters in this table speak for itself for most of the parameters listed. Some of the parameters are read-only, others are read-write.

Some of the parameters require a more detailed description and/or subdivision into bits, which is provided in the tables below.

- **PLC Digital Inputs (OD index 0x2000, subindex 7)**
  - in error-free situations all (used) bits are 1; unused bits are 0
  - all bits are read-only

BIT	Description (when bit=1)
0	Coolant flow OKAY
1	Pump temperature OKAY
2	24VDC powersupply OKAY
3	No pressure wave **P1** detected
4	No pressure wave **P2** detected
5-15	*...Not used...*

- **PLC Error Input Override Setting (OD index 0x2000, subindex 41)**
  - when set to 1 the corresponding error-input is ignored by the PLC application
  - all (used) bits are read-write; unused bits are 0

BIT	Description
0	Coolant flow
1	Pump temperature (clixon)
2	24VDC powersupply
3	Pressure wave **P1**
4	Pressure wave **P2**
5-15	*...Not used...*

- **PLC Error Status (OD index 0x2000, subindex 9)**
  - only one error bit (bits 2-15) is set at a time
  - in case several errors occurred repeated resets of the PLC Error Status (by writing zero to the OD object) are needed until the error status is cleared: *Error* bit (bit 0) goes to 0
  - the cooling system can only be started when the *Error* bit (bit 0) is 0
  - the *interlock* bit (bit 1) is set to 0 only after the cooling system is started

BIT	Description	Access
0	***Error*** ("OR" of errors present in PLC)	RO
1	Interlock	RO
2	Coolant flow	RW
3	Coolant temperature (**T1**)	RW
4	Pump temperature (clixon)	RW
5	Humidity sensor (**H1**)	RW
6	Humidity sensor (**H3**)	RW
7	Pressure sensor (**P1**)	RW
8	Pressure sensor (**P2**)	RW
9	Airflow (**AF**)	RW
10	24VDC powersupply	RW
11	Pressure wave (**P1**)	RW
12	Pressure wave (**P2**)	RW
13	Humidity sensor (**H2**)	RW
14	*...Not used...*	RO
15	*...Not used...*	RO

- **PLC Warning Status** (OD index 0x2000, subindex 10):

BIT	Description
0	***Warning*** ("OR" of bits 1 to 15)
1	Coolant temperature (**T1**)
2	Pressure sensor (**P1**)
3	Pressure sensor (**P2**)
4	Humidity sensor (**H1**)
5	Humidity sensor (**H3**)
6	Airflow (**AF**)
7	Humidity sensor (**H2**)
8	*...Not used...*
9	*...Not used...*
10	*...Not used...*
11	*...Not used...*
12	*...Not used...*
13	*...Not used...*
14	Error in provided parameters
15	RS232 communication error

- **PLC Global Status** (OD index 0x2000, subindex 11):
  - When both bits 0 and 1 of this word are written as 1 the PLC switches to a so-called *de-ventilation* mode where the cooling system pump is running with all valves closed in order to expel any remaining air from the cooling liquid in the system.

BIT	Description
0	Cooling system ON(1) or OFF(0)
1	Deventilate (when system ON (bit 0))
2-15	*...Not used...*

- **PLC Interlock Delay (OD index 0x2000, subindex 40)**
  - The interlock delay is the time in seconds between the occurrence of an error condition in the cooling system and the moment the hardware interlock intervenes and systems get switched off or get taken into a safe state; this time allows other MVD control systems to shut down gracefully or take appropriate actions before the interlock is activated or even to prevent subsequent activation of the interlock.

2.3 Monitoring PLC Error, Warning and Global Statuses

Before PLC status and data monitoring can start the PLC-CAN node has to be set into *Operational* state using the following 2-databyte *CANopen **NMT*** message:

Host (NMT-Master) Ⓡ PLC-CAN (NMT-Slave)

COB-ID	Byte 0	Byte 1
0x000	1 (*Start_Remote_Node*)	<*Node-ID*> or 0 (all nodes in network)

There is no reply to this message.

PLC data that is subjected to regular monitoring can be read out using the *CANopen **PDO*** mechanism. A PDO message is a non-confirmed CAN-message with one sender and one or more receivers, containing no protocol overhead, only data (1 to 8 bytes). It is assumed that receivers of a PDO message know the meaning of the data content of a PDO message.

PLC-CAN can produce 2 different PDOs, which we will call here **PDO1** and **PDO2**. **PDO2** messages are described in the next section.

The PLC parameters *'Error Status'*, *'Warning Status'* and *'Global Status'* are found in the PLC parameter block in the Object Dictionary index 0x2000, subindices 9, 10 and 11. They can be included in the regular 'monitoring scan' of the PLC parameters as described in the next section.

However, to enable a more frequent check (or PLC-CAN autonomous check, see below) on the PLC status words alone, an extra PDO has been defined that contains only these status words (6 bytes in total): **PDO1**.

(The definition of the data content of **PDO1** (its 'mapping') can be found in the Object Dictionary at index 0x1A00).

A PLC-CAN **PDO1** CAN-message has 6 data bytes:

PLC-CAN Ⓡ Host

COB-ID	Byte					
	0	1	2	3	4	5
0x180 + *Node_ID*	*Error Status* (LSB)	*Error Status* (MSB)	*Warning Status* (LSB)	*Warning Status* (MSB)	*Global Status* (LSB)	*Global Status* (MSB)

This **PDO1** message can be requested by the host by sending a socalled *Remote Transmission Request* (***RTR***) for **PDO1**. The CAN *Remote Frame* that constitutes the RTR has no data bytes and looks like this:

Host Ⓡ PLC-CAN

COB-ID
0x180+*Node_ID*

Typically the PLC-CAN status words would be read out every second for example, and the PLC parameters from the previous section only every 30 seconds or so.

However, the preferred way is to set the **PDO1** object to *'event-triggered'*, meaning that as soon as PLC-CAN detects that one or more bits in the statuswords have changed it will send a PDO1 message. To achieve this PLC-CAN polls the PLC periodically for its status words, with a period that can be set in OD index 0x2002. This parameter is set in units of 100 ms: if set to 1, PLC-CAN polls the PLC with a frequency of 10 Hz, when set to 2 with a frequency of 5 Hz, when set to 10 with a frequency of 1 Hz, etc, etc, up to a maximum period of 25.5 seconds.

When OD index 0x2002 is set to 0, PLC-CAN does *not* poll the PLC statuswords and the host should periodically issue an ***RTR*** for **PDO1** as described above, if it wants to stay up-to-date on the PLC statuswords.

Polling (and thus sending of event-triggered PDO1s) only takes place when PLC-CAN is in *Operational* state (PLC-CAN state is controlled by *CANopen* **NMT** messages).

See "*APPENDIX C* Protocol on the PLC-CAN↔PLC RS232-connection" for more details on PLC polling and the communication between PLC-CAN and PLC.

2.4 Monitoring PLC Parameters

PDO2 messages are used to transfer other PLC data that is to be monitored on a regular basis. The data to be monitored are16-bit data and are numbered according to the subindices of PLC-CAN Object Dictionary index 0x2000 (**Table 3**). One **PDO2** CAN-message is used to transfer one 16-bit PLC-parameter preceeded by its Object Dictionary subindex. Thus every 'monitoring request' results in a series of PDO2 messages, one PDO2 for every PLC-parameter to be monitored.

The number of parameters listed at OD index 0x2000 to be monitored (starting from subindex 1) can –if required– be set to any value by writing to OD index 0x2001 (using the *CANopen* ***SDO*** mechanism). This parameter has a default value of 6, but could be increased to e.g. 15 if additional PLC-parameters are added to OD index 0x2000 at subindices 12 to 15, marked *reserved*. See **Table 3**.

A change to parameter 0x2001 can be made permanent by saving it to the PLC-CAN on-board non-volatile memory (see OD index 0x1010 in **Table 1**).

A PLC-CAN **PDO2** CAN-message has 3 data bytes::

PLC-CAN ® Host

COB-ID	Byte 0	Byte 1-2
0x280 + *Node_ID*	PLC Parameter Index	16-bit PLC Parameter

with:

PLC Parameter Index: runs from 1 to 8 (or to whatever value has been set in OD index 0x2001).

PLC Parameter: 16-bits value, LSB in byte 1, MSB in byte 2.

(The definition of the data content of **PDO2** (its 'mapping') can be found in the Object Dictionary at index 0x1A01).

A 'monitoring request' –as mentioned above– is either a ***SYNC*** message or an ***RTR*** (Remote Transmission Request) for **PDO2**. Whether PLC-CAN responds to either one depends on the configuration of its **PDO2** *transmission type* (OD index 0x1801, subindex 2):

- **PDO2 transmission type = 1**:
  after every socalled **SYNC** message issued on the CAN-bus PLC-CAN sends 6 (or…) **PDO2** messages, one message for every PLC parameter configured for monitoring.
  The SYNC message is a CAN-message with a fixed COB-ID and no data bytes:

  Host Ⓡ all (SYNC-)slave nodes

COB-ID
0x080

  Note that all nodes configured to respond to a SYNC will react to a SYNC message.

- **PDO2 transmission type = 254**:
  after every socalled *Remote Transmission Request* (***RTR***) for **PDO2** the node sends 8 PDO messages, one message for every PLC parameter configured for monitoring.
  The CAN *Remote Frame* that constitutes the RTR has no data bytes and looks like this:

  Host Ⓡ PLC-CAN

COB-ID
0x280+*Node_ID*

  Note that an RTR is sent to and received by only one particular node.

The **PDO2** *transmission type* can also be saved to the PLC-CAN on-board non-volatile memory if required (see section 2.7).

2.5 Setting PLC Error, Warning and Global Statuses

The PLC parameters *'Error Status'*, *'Warning Status'* and *'Global Status'* can be written to, using the *CANopen* **SDO** mechanism.

'Error Status' and *'Warning Status'* can be reset to zero by writing 0 to the corresponding OD objects. To reset for example the '*Error Status*' word the following CAN-message (***SDO** expedited transfer*) is to be sent:

Host Ⓡ PLC-CAN

COB-ID	Byte						
	0	**1**	**2**	**3**	**4**	**5**	**6-7**
0x600 + *Node_ID*	0x2B	0x00	0x20	0x09	0x00	0x00	–

If successful PLC-CAN will reply with the following CAN-message:

PLC-CAN Ⓡ Host

COB-ID	Byte						
	0	**1**	**2**	**3**	**4**	**5**	**6-7**
0x580 + *Node_ID*	0x60	0x00	0x20	0x09	–	–	–

Note that in case several errors are present multiple resets of the '*Error Status*' (by writing zero to the OD object) are needed until the error status is cleared and the cooling system can be started.

A similar command/reply message pair can be used to reset the *'Warning Status'* word, or to set or reset individual bits in the *'Global Status'* PLC-parameter (if their function is defined, see section 2.1).

SDO messages, *to* as well as *from* PLC-CAN, always contain 8 databytes and the mechanism to transfer data is '***SDO** expedited transfer*', meaning that the data is contained in the message and is 4 bytes or less in size. (A segmented transfer would be necessary if the data to be transferred contained more than 4 bytes). If for some reason a node cannot service an SDO it will respond with an '***SDO** Abort Domain Transfer*' message (see example in section 2.7).

2.6 Setting PLC Error and Warning Limits

The PLC error and warning limits can be read and written by accessing the appropriate subindices (16 to 47, see **Table 3**) of OD index 0x2000 using the *CANopen **SDO*** mechanism.
The values read and written are in ADC-counts. If physical values are required a conversion from or to physical values has to be done by the user or the host application.

To set for example "*P2 warning level minimum*" (subindex 25 = 0x19 hexadecimal) to a value of 10000 (0x2710 hexadecimal) the following CAN-message (***SDO** expedited transfer*) is to be sent:

Host Ⓡ PLC-CAN

COB-ID	**Byte**						
	0	**1**	**2**	**3**	**4**	**5**	**6-7**
0x600 + *Node_ID*	0x2B	0x00	0x20	0x19	0x10	0x27	–

If successful PLC-CAN will reply with the following CAN-message:

PLC-CAN Ⓡ Host

COB-ID	**Byte**						
	0	**1**	**2**	**3**	**4**	**5**	**6-7**
0x580 + *Node_ID*	0x60	0x00	0x20	0x19	–	–	–

To read for example "*T error level maximum*" (subindex 19 = 0x13 hexadecimal) the following CAN-message is to be sent:

Host Ⓡ PLC-CAN

COB-ID	**Byte**						
	0	**1**	**2**	**3**	**4**	**5**	**6-7**
0x600 + *Node_ID*	0x40	0x00	0x20	0x13	–	–	–

The reply CAN-message (***SDO** expedited transfer*) of PLC-CAN looks for example like this:

PLCCAN Ⓡ Host

COB-ID	Byte						
	0	**1**	**2**	**3**	**4**	**5**	**6-7**
0x580 + *Node_ID*	0x4B	0x00	0x20	0x13	0x34	0x12	–

In this case it means that the "*T error level maximum*" (shown in bytes 4 and 5) is equal to 0x1234 (= 4660 decimal).

SDO messages, to as well as from PLC-CAN, always contain 8 databytes and the mechanism to transfer data is '***SDO** expedited transfer*', meaning that the data is contained in the message and is 4 bytes or less in size. (A *segmented transfer* would be necessary if the data to be transferred contains more than 4 bytes). If for some reason a node cannot service an SDO it will respond with an '***SDO** Abort Domain Transfer*' message (see example in section 2.7).

2.7 Storing PLC-CAN Parameters

Some parameters can be stored permanently onboard the PLC-CAN node in non-volatile memory (an EEPROM) by writing the string "save" to OD index 0x1010. The *CANopen **SDO*** mechanism is used to accomplish this:

Host Ⓡ PLC-CAN

COB-ID	Byte							
	0	**1**	**2**	**3**	**4**	**5**	**6**	**7**
0x600 + *Node_ID*	0x23	0x10	0x10	1	0x73 ('s')	0x61 ('a')	0x76 ('v')	0x65 ('e')

with OD *index* 0x1010 in byte 1+2 and *subindex* 1 in byte 3.

The parameters stored are:

- OD index 0x1800, subindex 2 (PDO1 communication parameters).
- OD index 0x1801, subindex 2 (PDO2 communication parameters).
- OD index 0x2001 (number of PLC parameters to be monitored).
- OD index 0x2002 (PLC status polling period).

PLC error and warning limit settings are assumed to be stored by the PLC on the PLC.

If the store-operation succeeded the controller sends the following reply:

PLC-CAN Ⓡ Host

COB-ID	Byte						
	0	**1**	**2**	**3**	**4**	**5**	**6-7**
0x580 + *Node_ID*	0x60	0x10	0x10	1	–	–	–

If the store-operation did NOT succeed the controller sends the following reply (***SDO** Abort Domain Transfer*, error reason: 'hardware fault' (for details see [1])):

PLC-CAN Ⓡ Host

COB-ID	**Byte**							
	0	**1**	**2**	**3**	**4**	**5**	**6**	**7**
0x580 + *Node_ID*	0x80	0x10	0x10	1	0	0	6 (Error Code)	6 (Error Class)

Parameters can be reset to their default values (by invalidating the corresponding contents of the EEPROM) by writing to OD index 0x1011, using this time the string "load" in bytes 4 to 7 of the SDO message. Note that the default values take effect only after a subsequent reset of the node. Default values are listed in the Object Dictionary tables.

2.8 Emergency Objects

Emergency messages are triggered by the occurrence of an internal (fatal) error situation. An emergency CAN-message has the following general syntax:

PLC-CAN ® Host

COB-ID	Byte 0-1	Byte 2	Byte 3-7
0x080 + *Node_ID*	Emergency Error Code	Error Register (Object 0x1001)	Manufacturer specific error field

The following Emergency messages are defined for PLC-CAN:

Error Description	Emergency Error Code (byte 0-1)	Error Register bit (Object 1001H) (byte 2)	Manufacturer-specific Error Field (byte 3-7)
Watchdog or manual (front-panel) reset	0x6000	0x01	Byte 3,4,5,6: Manufacturer Device Name (Object Dictionary index 0x1008) Byte 7: 0
CAN-controller overrun: message lost	0x8100	0x10	Byte 3: 1 Byte 4: counter (modulo 256) Byte 5: CANSTA (CAN-controller status register) Byte 6,7: 0
CAN-controller error: communication error	0x8100	0x10	Byte 3: 2 Byte 4: counter (modulo 256) Byte 5: CANSTA (CAN-controller status register) Byte 6,7: 0
Local CAN message buffer overflow: message lost	0x8100	0x10	Byte 3: 3 Byte 4: counter (modulo 256) Byte 5: CANSTA (CAN-controller status register) Byte 6,7: 0
RS232: communication time-out	0xFF00	0x80	Byte 3: 1 Byte 4: parameter index for which timeout occurred Byte 5: number of received chars in RS232 buffer Byte 6,7: 0
RS232: unexpected parameter index	0xFF00	0x80	Byte 3: 2 Byte 4: parameter index expected Byte 5: parameter index received Byte 6,7: 0
RS232: PLC poll operation aborted	0xFF00	0x80	Byte 3: 0x10 Byte 4: poll error counter Byte 5,6,7: 0
EEPROM: write failed	0x5000	0x80	Byte 3: 4 Byte 4,5,6,7: 0
EEPROM: read CRC error	0x5000	0x80	Byte 3: 8 Byte 4,5,6,7: 0

Note that the Error Register (Object Dictionary index 0x1001) can have one or more of the bits shown above set, depending on the node's history of error occurrences. The table shows the Error Register bit that gets additionally set when described error occurs.

3 Object Dictionary

Table 1 to **Table 4** show in detail the Object Dictionary (**OD**) of the **PLC-CAN** *CANopen* application.

Column '**Attr**' shows the access rights attribute of an object: RO=read-only, RW=read-or-write, WO=write-only.

All entries in the OD are accessed using the *CANopen* ***SDO*** mechanism with *expedited transfer* (object data content always <= 4 bytes).

Communication Profile Area (PLC-CAN)						
Index (hex)	**Sub Index**	**Name**	**Data/ Object**	**Attr**	**Default (hex)**	**Comment**
1000	-	Device type	U32	RO	00000000	
1001	-	Error register	U8	RO	0	Error bits according to DS-301 (error status overview)
1002	-	Manufacturer status reg	U32	RO	0	*see below*
1004		#PDOs supported	Array			
	0	Total #PDOs supported	U32	RO	00000002	0 receive, 2 transmit PDO
	1	#PDOs sync	U32	RO	00000002	PDO after SYNC
	2	#PDOs async	U32	RO	00000002	PDO after RTR or 'event'
1008	-	Manufacturer device name	VisStr	RO	"SPIC"	= SPICAN module
100A	-	Manufacturer software version	VisStr	RO	"PC10"	MVD Cooling System PLC-to-CAN, Version 1.0
100B	-	Node identifier	U32	RO		set by frontpanel hex-switches
100E	-	Node Guarding COB-ID	U32	RO	0x700+ Node-ID	According to CANopen Predefined Connection Set
100F	-	#SDOs supported	U32	RO	00000001	0 client, 1 server SDO
1010		Store parameters	Array			Save stuff in onboard EEPROM
	0	Highest index supported	U8	RO	1	
	1	Save all parameters	U32	RW	1	read: 1 write "save": store all
1011		Restore default parameters	Array			Invalidate stuff in onboard EEPROM
	0	Highest index supported	U8	RO	1	
	1	Restore all parameters	U32	RW	1	read: 1 write "load": invalidate all stored

Table 1. Communication Profile Area of the CANopen Object Dictionary.

Communication Profile Area (PLC-CAN) *(continued…)*						
Index (hex)	**Sub Index**	**Name**	**Data/ Object**	**Attr**	**Default (hex)**	**Comment**
1800		1st Transmit PDO parameters	Record			Data type = PDOCommPar
	0	Number of entries	U8	RO	2	
	1	COB-ID used by PDO	U32	RO	180+ Node-ID	According to CANopen Predefined Connection Set
	2	Transmission type	U8	RW	FE	254: (manufctr–specific) get PLC statuswords at RTR (default) 1 : get PLC statuswords at SYNC
1801		2nd Transmit PDO parameters	Record			Data type = PDOCommPar
	0	Number of entries	U8	RO	2	
	1	COB-ID used by PDO	U32	RO	280+ Node-ID	According to CANopen Predefined Connection Set
	2	Transmission type	U8	RW	FE	254: (manufctr–specific) get PLC data at RTR (default) 1 : get PLC data at SYNC
1A00		1st Transmit PDO mapping	Record			Data type = PDOMapping
	0	Number of entries	U8	RO	3	
	1	Cooling System Error Status	U32	RO	20000910	OD-index 2000, sub-index 9; Size = 16 bits
	2	Cooling System Warning Status	U32	RO	20000A10	OD-index 2000, sub-index 10; Size = 16 bits
	3	Cooling System Global Status	U32	RO	20000B10	OD-index 2000, sub-index 11; Size = 16 bits
1A01		2nd Transmit PDO mapping	Record			Data type = PDOMapping
	0	Number of entries	U8	RO	2	
	1	Multiplexor 1	U32	RO	6F100108	OD-index 6F10, sub-index 1: *Multiplexor* 1 (see DSP-404); Size = 8 bits
	2	16-bit PLC parameter	U32	RO	2000FD10	OD-index 2000, sub-index 253: PLC parameter via a multiplexor; Size = 16 bits

Table 2. Communication Profile Area of the CANopen Object Dictionary (continued).

The **Manufacturer Status Register** (Object Dictionary index 0x1002), a 32-bit object, providing more specific status information about the PLC-CAN module, i.e the type of errors that have occurred (reported by Emergency Objects). The layout of this Register is as follows:

bits 31-5	**bit 4**	**bit 3**	**bit 2**	**bit 1**	**bit 0**
not used	PLC-status poll operation aborted (RS232)	PLC-CAN EEPROM-read CRC error	PLC-CAN EEPROM-write error	communication timeout (RS232)	unexpected parameter index received (RS232)

Manufacturer-specific Profile Area (PLC-CAN)						
Index (hex)	**Sub Index**	**Name**	**Data/ Object**	**Attr**	**Default (dec)**	**Comment**
2000		PLC parameter block	Array			
	0	Number of entries	U16	RO	53	
	1	Temperature (**T1**)	U16	RO		
	2	Pressure 1 (**P1**)	U16	RO		
	3	Pressure 2 (**P2**)	U16	RO		
	4	Hygrometer 1 (**H1**)	U16	RO		
	5	Hygrometer 3 (**H3**)	U16	RO		
	6	Airflow (**AF**)	U16	RO		
	7	Digital inputs	U16	RO		see section 2.2
	8	Hygrometer 2 (**H2**)	U16	RO		
	9	Error status	U16	RW		mapped into PDO1; see sect. 2.2
	10	Warning status	U16	RW		mapped into PDO1; see sect. 2.2
	11	Global status	U16	RW		mapped into PDO1; see sect. 2.2
	12	*reserved*	U16	RO	0	
	13	*reserved*	U16	RO	0	
	14	*reserved*	U16	RO	0	
	15	*reserved*	U16	RO	0	
	16	**T1** error level minimum	U16	RW		
	17	**T1** warning level minimum	U16	RW		
	18	**T1** warning level maximum	U16	RW		
	19	**T1** error level maximum	U16	RW		
	20	**P1** error level minimum	U16	RW		
	21	**P1** warning level minimum	U16	RW		
	22	**P1** warning level maximum	U16	RW		
	23	**P1** error level maximum	U16	RW		
	24	**P2** error level minimum	U16	RW		
	25	**P2** warning level minimum	U16	RW		
	26	**P2** warning level maximum	U16	RW		
	27	**P2** error level maximum	U16	RW		
						…continued in the next table…

Table 3. Manufacturer-specific Profile Area of the *CANopen* Object Dictionary for the PLC-CAN device.

Manufacturer-specific Profile Area (PLC-CAN) *(continued…)*						
Index (hex)	**Sub Index**	**Name**	**Data/ Object**	**Attr**	**Default**	**Comment**
2000						*..continued from previous table..*
	28	**H1** error level minimum	U16	RW		
	29	**H1** warning level minimum	U16	RW		
	30	**H1** warning level maximum	U16	RW		
	31	**H1** error level maximum	U16	RW		
	32	**H3** error level minimum	U16	RW		
	33	**H3** warning level minimum	U16	RW		
	34	**H3** warning level maximum	U16	RW		
	35	**H3** error level maximum	U16	RW		
	36	**AF** error level minimum	U16	RW		
	37	**AF** warning level minimum	U16	RW		
	38	**AF** warning level maximum	U16	RW		
	39	**AF** error level maximum	U16	RW		
	40	Interlock delay	U16	RW		in seconds [0, 200]
	41	Error input override	U16	RW		see section 2.2
	42	D**P1** error level maximum	U16	RW		
	43	D**P2** error level maximum	U16	RW		
	44	**H2** error level minimum	U16	RW		
	45	**H2** warning level minimum	U16	RW		
	46	**H2** warning level maximum	U16	RW		
	47	**H2** error level maximum	U16	RW		
2001		Max. PLC monitoring subindex for PDO2	U8	RW	8	Determines up to which subindex of Object 2000 is to be monitored using PDO2-RTR and/or SYNC (Access to other subindices using PDO1or SDO).
2002		PLC-status polling period for PDO1	U8	RW	0	Time between consecutive polls by PLC-CAN of PLC error and warning status, in units of 100 ms; '0' means: polling not enabled.

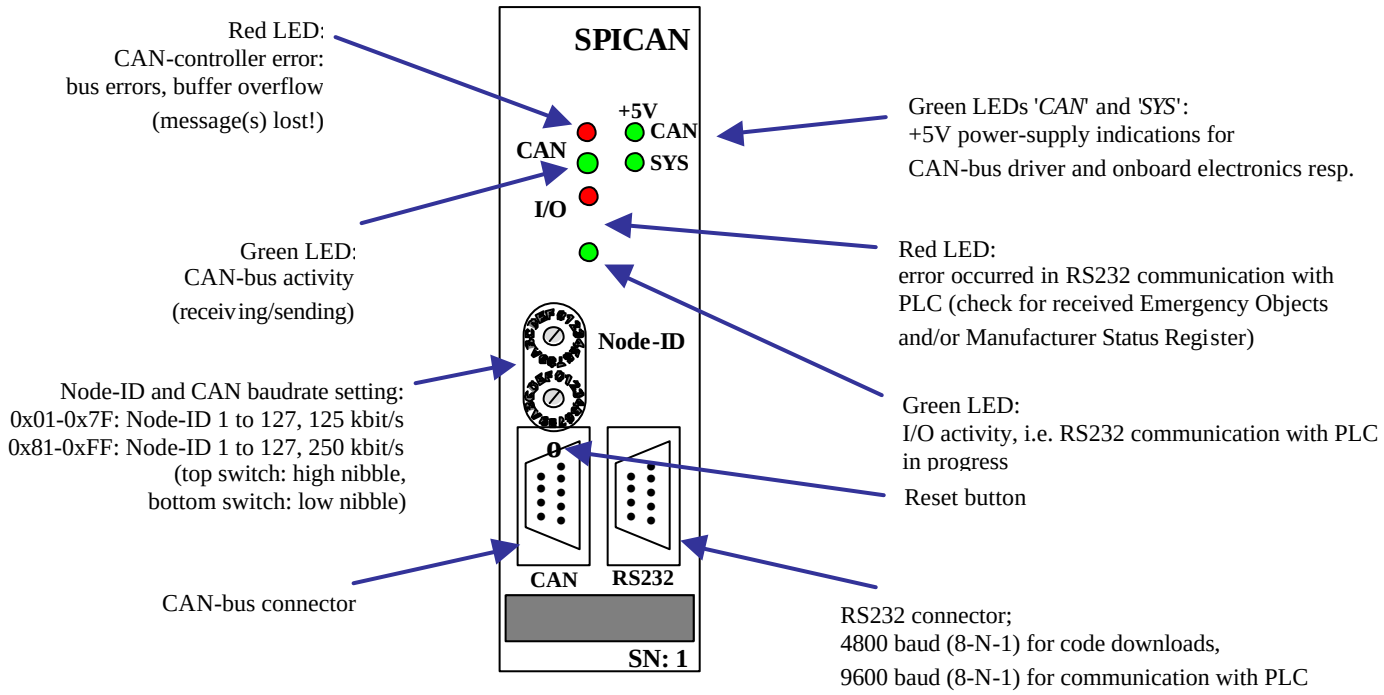
Table 4. Manufacturer-specific Profile Area of the *CANopen* Object Dictionary for the PLC-CAN device (continued).

References

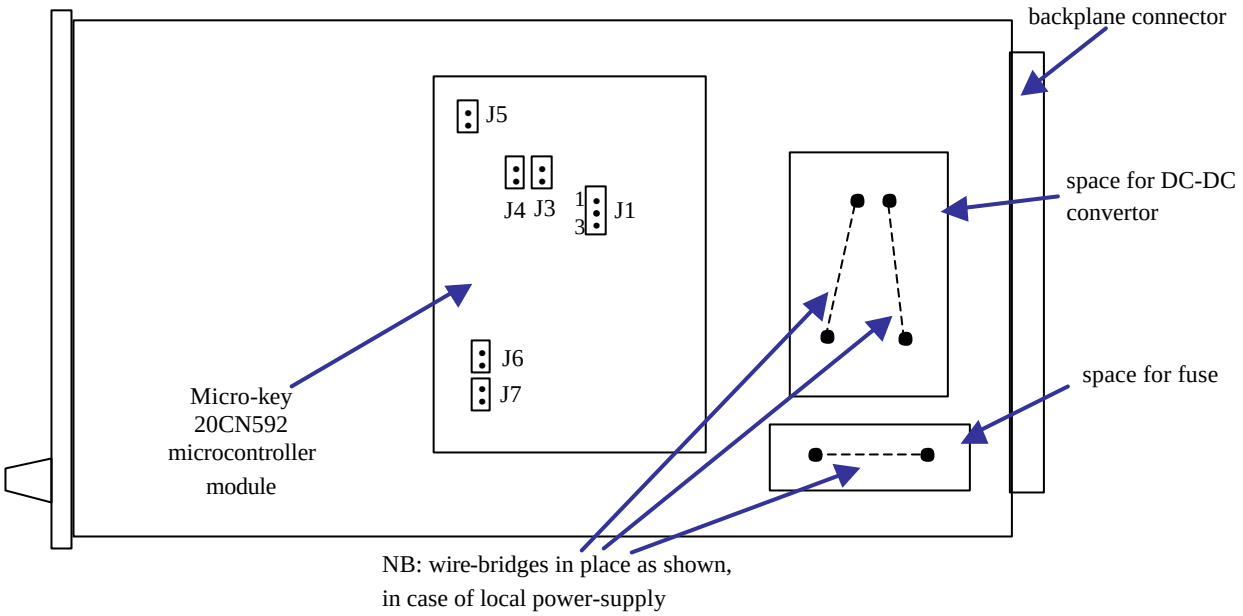
[1] H.Boterenbrood,
CANopen, high-level protocol for CAN-bus,
Version 3.0, NIKHEF, Amsterdam, March 20 2000.
(http://www.nikhef.nl/pub/departments/ct/po/doc/CANopen30.pdf).

APPENDIX A LEDs, Switches and Jumpers

SPICAN Frontpanel:



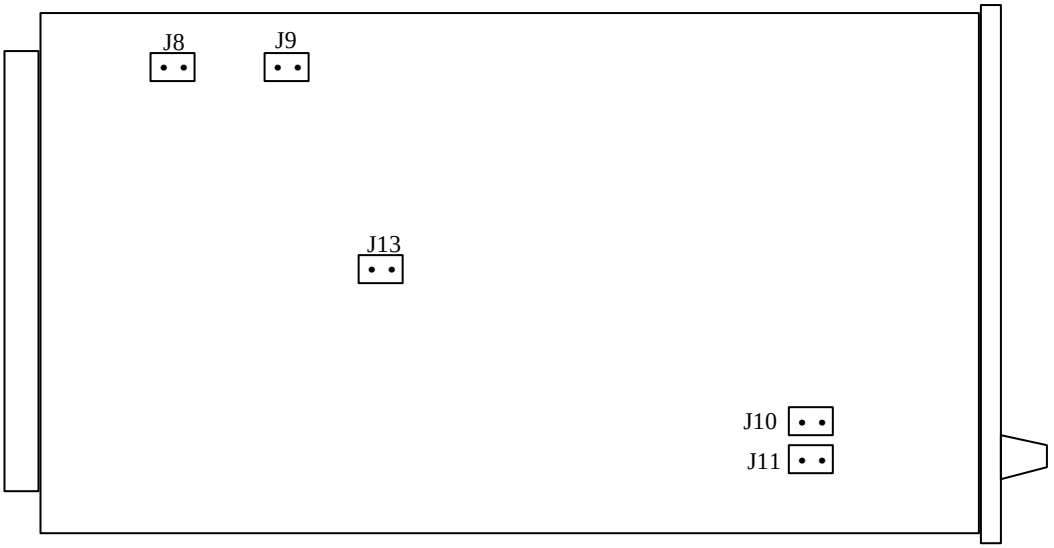
SPICAN component side:



J1: 80C592 internal watchdog (J1/1-2 closed: disabled; J1/2-3 closed: enabled).
J3: powerfail interrupt request via P1.0/#INT2 pin (open: interrupt disabled)
J4: external watchdog enable selector, 80C592 pin P1.1 to MAX691 (open watchdog disabled).
J5: reset jumper (closed: system will reset).

J6/J7: serial port signal connection/disconnection.

SPICAN solder side:



Power options:

Local power supply: via backplane
J8 + J9 closed, J10 + J11 open,
DC-DC convertor and fuse NOT placed, wire-bridges in place.

Ext. +5V power supply: via CAN-connector
J8 + J9 open, J10 + J11 closed,
DC-DC convertor NOT placed (wire-bridges in place), fuse placed.

Ext. +9…36V power-s.: via CAN-connector
J8 + J9 open, J10 + J11 closed,
DC-DC convertor and fuse placed.

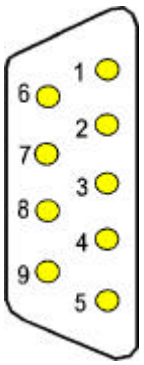
Additional power option:

Battery backup: J13 open.

APPENDIX B Connector Layout

9-pin D-sub *male* CAN-connector:

Pin	Signal
1	–
2	CAN_Low
3	V_gnd
4	–
5	–
6	–
7	CAN_High
8	–
9	V+



APPENDIX C **Protocol on the PLC-CAN « PLC RS232-connection**

The cooling system's PLC works autonomously and independently but can be controlled and monitored via its interface to the CAN-bus, formed by the PLC-CAN module and an RS232 connection between the PLC and the PLC-CAN. The RS232-connection is set to 9600 baud, 8 bits, 1 stop bit, no parity.

The protocol over the RS232-connection between the PLC-CAN module and the PLC is a simple command/reply type of protocol where the PLC-CAN module is the initiator of the command message: the PLC only generates a message in reply to a message from the PLC-CAN. The full list of messages is shown in **Table 5** (together with the preferred *CANopen* access mechanism for each individual parameter; for more details of this, see the tables containing the PLC-CAN Object Dictionary and the description of the PDOs).

The command message as well as the reply message consists of 3 bytes. The first byte contains a parameter-identifier in bit 0 to 6; bit 7 is 1 when the PLC-CAN requests a *write* access to the parameter and bit 7 is 0 when a *read* access is requested; the value of byte 0 of the reply must be identical in the reply from the PLC. This is checked by the PLC-CAN application for data that is subsequently put in a *CANopen PDO* message (before being put in the PDO the parameter-identifier is removed). When the PLC-CAN application detects such a mismatch in the parameter-identifier in the RS232 command and reply message an appropriate *CANopen* EMERGENCY message is sent (see section 2.8).

The parameter identifier is not checked for *CANopen SDO* access of PLC parameters; in this case it is the responsibility of the CAN host-application to check that the subindex of the object (which equals the 'parameter identifier') in the SDO-reply matches the one in the corresponding SDO-request.

Bytes 2 and 3 of the PLC-CAN command message either have a *don't-care* value (PLC-CAN requests a *read* access) or contain the parameter-value (16-bit, LSB first) to be written/set in the PLC.

Bytes 2 and 3 of the PLC reply message contains the current or newly written value of the parameter.

When a time-out occurs on reception of the reply from the PLC in response to a message from the PLC-CAN application an appropriate *CANopen* EMERGENCY message is sent (again see section 2.8) and PLC-CAN reinitializes its RS232 interface and buffers.
The time-out is set to 100 ms.

If a time-out or parameter-index mismatch occurs during the automatic PLC-status scanning operation of the PLC-CAN application the appropriate *CANopen* EMERGENCY is sent. The scanning operation is not suspended until a total of 50 such errors in a row occur (currently hardcoded…). A successful PLC-status scan decrements the error counter by one. This is to prevent a possible endless stream of error message when the connection to the PLC is permanently lost. The aborting of further PLC-status scanning is notified to the user by a separate *CANopen* EMERGENCY (again see section 2.8).

The scanning operation can be resumed by sending a *CANopen* NMT *Start-Remote-Node* message (the error counter is reset to zero).

PLC parameter	**Attr**	**Byte1 bit 7 (R/W)**	**Byte1 bit 6-0** (parameter-id)	**Byte 2+3** (parameter-data)		**(preferred) CANopen access mechanism**
Temperature (T)	RO	0	0x01	–	–	PDO2-Tx (RTR or SYNC)
Pressure 1 (P1)	RO	0	0x02	–	–	PDO2-Tx (RTR or SYNC)
Pressure 2 (P2)	RO	0	0x03	–	–	PDO2-Tx (RTR or SYNC)
Hygrometer 1 (H1)	RO	0	0x04	–	–	PDO2-Tx (RTR or SYNC)
Hygrometer 3 (H3)	RO	0	0x05	–	–	PDO2-Tx (RTR or SYNC)
Airflow (AF)	RO	0	0x06	–	–	PDO2-Tx (RTR or SYNC)
Digital Inputs	RO	0	0x07	–	–	PDO2-Tx (RTR or SYNC)
Hygrometer 2 (H2)	RO	0	0x08	–	–	PDO2-Tx (RTR or SYNC)
Error Status	R	0	0x09	–	–	PDO1-Tx (RTR or CoS)
“ (reset 1 error)	W	1	“	0	0	SDO
Warning Status	R	0	0x0A	–	–	PDO1-Tx (RTR or CoS)
“ (reset warnings)	W	1	“	0	0	SDO
Global Status	R	0	0x0B	–	–	PDO1-Tx (RTR or CoS)
“ (cooling on/off)	W	1	“	1 / 0	0	SDO
“ (deventilate on/off)	W	1	“	3 / 0	0	SDO
….reserved….			0x0C-0x0F	–	–	–
T1 min. error	R	0	0x10	–	–	SDO
“ “ “ (set)	W	1	“	LSB	MSB	SDO
T1 min. warning	RW	0 / 1	0x11	"	"	SDO
T1 max. warning	RW	0 / 1	0x12	"	"	SDO
T1 max. error	RW	0 / 1	0x13	"	"	SDO
P1 min. error	RW	0 / 1	0x14	"	"	SDO
P1 min. warning	RW	0 / 1	0x15	"	"	SDO
P1 max. warning	RW	0 / 1	0x16	"	"	SDO
P1 max. error	RW	0 / 1	0x17	"	"	SDO
P2 min. error	RW	0 / 1	0x18	"	"	SDO
P2 min. warning	RW	0 / 1	0x19	"	"	SDO
P2 max. warning	RW	0 / 1	0x1A	"	"	SDO
P2 max. error	RW	0 / 1	0x1B	"	"	SDO
H1 min. error	RW	0 / 1	0x1C	"	"	SDO
H1 min. warning	RW	0 / 1	0x1D	"	"	SDO
H1 max. warning	RW	0 / 1	0x1E	"	"	SDO
H1 max. error	RW	0 / 1	0x1F	"	"	SDO
H3 min. error	RW	0 / 1	0x20	"	"	SDO
H3 min. warning	RW	0 / 1	0x21	"	"	SDO
H3 max. warning	RW	0 / 1	0x22	"	"	SDO
H3 max. error	RW	0 / 1	0x23	"	"	SDO
AF min. error	RW	0 / 1	0x24	"	"	SDO
AF min. warning	RW	0 / 1	0x25	"	"	SDO
AF max. warning	RW	0 / 1	0x26	"	"	SDO
AF max. error	RW	0 / 1	0x27	"	"	SDO
						continued on next page

PLC parameter	Attr	Byte1 bit 7 (R/W)	Byte1 bit 6-0 (parameter-id)	Byte 2+3 (parameter-data)		CANopen access mechanism
Interlock delay	RW	0 / 1	0x28	"	"	SDO
Error Input Override	RW	0 / 1	0x29	"	"	SDO
delta-P1 max. error	RW	0 / 1	0x2A	"	"	SDO
delta-P2 max. error	RW	0 / 1	0x2B	"	"	SDO
H2 min. error	RW	0 / 1	0x2C	"	"	SDO
H2 min. warning	RW	0 / 1	0x2D	"	"	SDO
H2 max. warning	RW	0 / 1	0x2E	"	"	SDO
H2 max. error	RW	0 / 1	0x2F	"	"	SDO

Table 5. List of messages exchanged over RS232 between PLC-CAN and the PLC.
(RO = *Read-Only*, RW = *Read/Write*, '–' = 0 / *don't care*, LSB = *Least Significant Byte*, MSB = *Most Significant Byte*; RTR = *triggered by CAN Remote Transmission Request*, SYNC = *triggered by CANopen SYNC message*, CoS = *triggered by Change of State*).