CANopen communication driver for VT series W

ESA elettronica S.p.A.
Via Padre Masciadri, 4/A
I – 22066 Mariano Comense (CO)
Tel. ++ 39.031757400 Fax ++ 39.031751777
web: www.esahmi.com email: info@esahmi.com

ESA elettronica S.p.A.
Via Padre Masciadri, 4/A
I – 22066 Mariano Comense (CO)
Tel. ++ 39.031757400 Fax ++ 39.031751777
web: www.esahmi.com email: info@esahmi.com

1. General

1.1 Document revision

Edition	Description	Author
10/2001	First issue : release of CANopen Master driver.	J. Quercia
06/2002	Paragraph 4 added.	J. Quercia
05/2003	“Protocol errors” paragraph added	J. Quercia

1.2 About this document

This document describes the functionalities of the ESA CANopen Master driver, released for use with ESA VT series W terminals.
It is assumed that the basic operating functions of the CANopen communication profile are known and available as reference documentation.

1.3 Reference documentation

CiA – CAN specification version 2.0, part A
CiA – CAN specification version 2.0, part B
CiA – Draft standard 301 version 4.0, “Application layer and communication profile”.

1.4 Definitions and abbreviations

CAN: controller area network. An internal standardized serial bus system.

COB: communication object. A unit of transportation in a CAN network. A COB can contain at most 8 bytes of data.

COB-ID: COB identifier. Each COB is uniquely identified in a CAN network by a number.

DEVICE: a generic equipment able to communicate via the CAN bus using the CANopen communication profile.

NMT: network management. Through NMT services, nodes are initialised, started, monitored, reset or stopped.

OBJECT DICTIONARY: a group of objects (parameters, variables) that belong to a device. Each object is accessible via the network using an index-subindex address format.

PDO: process data object. One of the COB types. The PDO performs the real-time data transfer.

RPDO: receive PDO. Through this PDO the device receives real-time data.

SDO: service data object. One of the COB types. The SDO provides access to the object dictionary of a device.

SYNC: synchronization object. It provides the basic network clock.

TPDO: transmit PDO. Through this PDO the device transmits real-time data.

VT: ESA operator terminal panel series W.

VTWIN: the software developed by ESA that allows the user to create the application for the VT.

2. CAN communication

2.1 Hardware interfaces

The VT can communicate in a CAN bus either using a built-in CAN port or through a VTCAN adapter.
The following VT models have the built-in CAN port: VT50, VT60, VT150W, VT300W, VT505W, VT555W, VT585W.
The VTCAN adapter can be connected to the serial ports of all the VT models, via a RS232 line. The adapter provides the connection to the CAN bus.

2.2 CAN frame type

The driver supports the CAN standard frames with 11-bit identifier field.

2.3 Communication parameters

The communication parameters setting is made using the VTWIN software. The following parameter are definable:

- transmission rates (250, 500, 800, 1000 kbit/s);
- device address (1 – 127);
- boot-up time (0 – 65000 msec);
- sync time (0 – 65000 msec);
- cycle time (0 – 65000 msec).

The device address represents the node number of the slave device(s) connected to the VT.
The boot-up time represents the time that, after power-on or reset, the VT waits for before broadcasting the start remote node message.
The sync time, if different from 0, represents the period of the VT SYNC object transmission.
The cycle time, if different from 0, represents the period of the VT cyclic PDO transmission.

2.4 Pre-defined connection set

A default identifier allocation scheme is defined.
The following table shows the pre-defined connection set. It has to be seen from the device point of view. So the transmit objects are the object transmitted by the device to the VT and the receive objects are those received by the device from the VT.

Object	COB-ID
NMT	0
SYNC	128 (80h)
TPDO1	385 (181h) – 511 (1FFh)
RPDO1	513 (201h) – 639 (27Fh)
TPDO2	641 (281h) – 767 (2FFh)
RPDO2	769 (301h) – 895 (37Fh)
TPDO3	897 (381h) – 1023 (3FFh)
RPDO3	1025 (401h) – 1151 (47Fh)
TPDO4	1153 (481h) – 1279 (4FFh)
RPDO4	1281 (501h) – 1407 (57Fh)
SDO (tx)	1409 (581h) – 1535 (5FFh)
SDO (rx)	1537 (601h) – 1663 (67Fh)

The default COB-ID is automatically assigned by the VTWIN software. The user can modify the identifier of SDOs and PDOs.

3. CAN protocol

3.1 Data types

The following data types are supported for SDOs and PDOs:

- Integer8;
- Integer16;
- Integer24;
- Integer32;
- String.

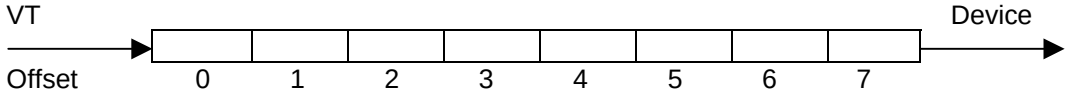
For numerical data types the encoding is little endian style. Integers can be signed or unsigned.

3.2 PDO protocol

The driver supports three types of PDO objects: cyclic synchronous, cyclic asynchronous and event-triggered PDOs.
The cyclic synchronous PDOs are cyclically transmitted with relation to the SYNC object. The cyclic asynchronous PDOs are cyclically transmitted without any relation to the SYNC object, with the period specified in the Cycle Time parameter.
The event-triggered PDOs are transmitted in the event of a data-change.

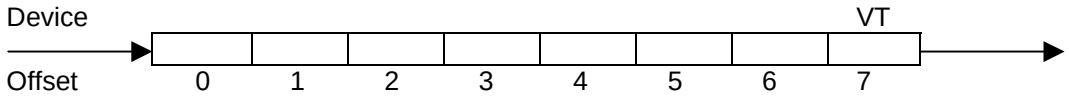
Write PDO protocol

The write PDO protocol allows to write data in the PDOs, using an offset that points to the byte(s) to be written. The write operation is allowed only for RPDOs.



Read PDO protocol

The read PDO protocol allows the VT to read data from the PDOs, using an offset that points to the byte(s) to be read.

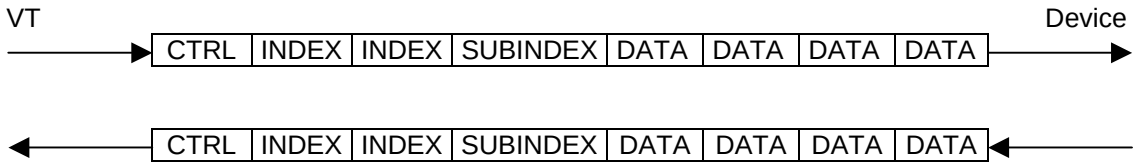


3.3 SDO protocol

The driver supports the SDO download and upload protocols in expedited transfer mode.
The SDO segment/block download and upload protocols are not supported.

SDO download protocol

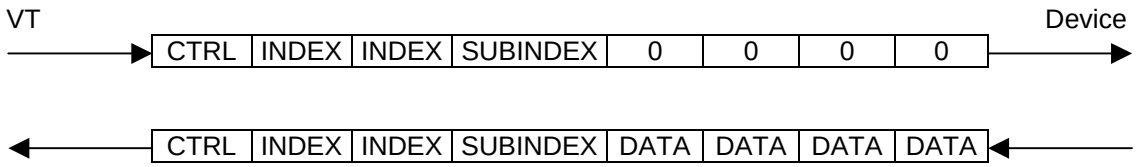
The SDO download protocol allows to write data in the SDOs and, consequently, in the object dictionary of a device. The Initiate SDO Download protocol is used. The type of transfer, the data length, the index-subindex address and the data are indicated to the device.



The device returns the indication of success or failure of the request in the control byte. In case of success the data bytes are set to 0, otherwise the error response is indicated.

SDO upload protocol

The SDO upload protocol allows the VT to read data from the device's object dictionary. The Initiate SDO Upload protocol is used. The type of transfer, the data length, the index-subindex address are indicated to the device.

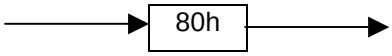


The device returns the indication of success or failure of the request in the control byte. In case of success the data bytes contain the value requested, otherwise the error response is indicated.

3.4 SYNC protocol

The driver broadcasts the SYNC object periodically, with the period specified in the Sync Time parameter. If this parameter is set to 0, the SYNC object is not broadcasted.

VT Device(s)

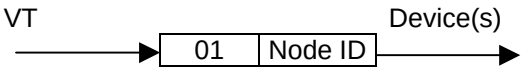


3.5 NMT protocol

The NMT protocol allows the master node to control the state of the slave. The slave can have the following states: Stopped, Operational, Pre-Operational, Initialising.
The driver supports the following NMT services: Start Remote Node, Stop Remote Node, Enter Pre-Operational, Reset Node, Reset Communication.

Start Remote Node protocol

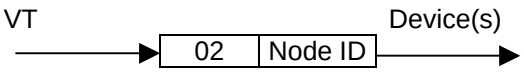
This protocol implements the Start Remote Node service.



The Node ID byte indicates the node number of the slave. If set to 0, the protocol addresses all the slaves.

Stop Remote Node protocol

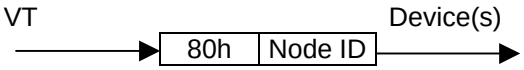
This protocol implements the Stop Remote Node service.



The Node ID byte indicates the node number of the slave. If set to 0, the protocol addresses all the slaves.

Enter Pre-Operational protocol

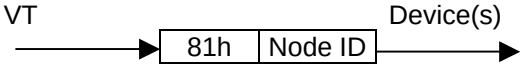
This protocol implements the Enter Pre-Operational service.



The Node ID byte indicates the node number of the slave. If set to 0, the protocol addresses all the slaves.

Reset Node protocol

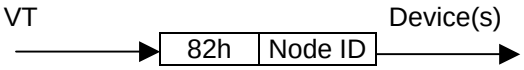
This protocol implements the Reset Node service.



The Node ID byte indicates the node number of the slave. If set to 0, the protocol addresses all the slaves.

Reset Communication protocol

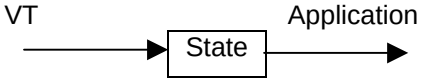
This protocol implements the Reset Communication service.



The Node ID byte indicates the node number of the slave. If set to 0, the protocol addresses all the slaves.

3.6 VT state protocol

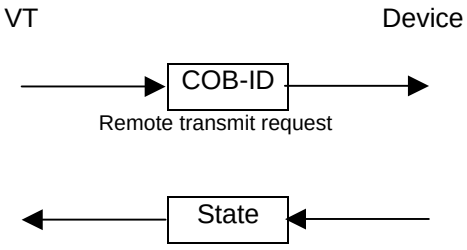
The VT state protocol indicates the VT state to the application.



State can assume the following values:
0 = Operational
1 = Pre-Operational
2 = Bus warning
3 = Bus off

3.7 Device state protocol

The Device state service allows the VT to read the device state.
This service is implemented using the Node Guarding protocol.



COB-ID = 700h + Node number

State can assume the following values:
4 = Stopped
5 = Operational
127 = Pre-Operational

ESA elettronica S.p.A.
Via Padre Masciadri, 4/A
I – 22066 Mariano Comense (CO)
Tel. ++ 39.031757400 Fax ++ 39.031751777
web: www.esahmi.com email: info@esahmi.com

4. Protocol errors

The VT reports the detected protocol errors in its service page.
Each message and its correspondendent meaning are described below.

PR ERR : a protocol time-out has occurred while the VT was communicating with the device;
SDO ERROR : the VT is requesting an incorrect SDO index or subindex.
INIT ERR : a protocol time-out has occurred during the communication initialize;
OPEN ERR : a protocol time-out has occurred while the VT is opening the communication channel;
DSBLE ERR : error during the disable of the CAN communication processor.
RESET ERR : error during the reset of the CAN communication processor.

If a VTCAN adapter is used then the VT will show the following error messages:

READ ERR : a protocol time-out has occurred while the VT was requesting data from the device;
WRITE ERR : a protocol time-out has occurred while the VT was sending data to the device;
INIT ERR : a protocol time-out has occurred during the communication initialize;
OPEN ERR : a protocol time-out has occurred while the VT is opening the communication channel;
CLOSE ERR : error during the disable of the CAN communication processor.

5. VTWIN application interface

5.1 Data areas

The VTWIN software allows the user to create the following type of application variables:

- Command;
- PDO 1 Receive;
- PDO 1 Receive Integer24;
- PDO 1 Transmit;
- PDO 1 Transmit Integer24;
- PDO 2 Receive;
- PDO 2 Receive Integer24;
- PDO 2 Transmit;
- PDO 2 Transmit Integer24;
- PDO 3 Receive;
- PDO 3 Receive Integer24;
- PDO 3 Transmit;
- PDO 3 Transmit Integer24;
- PDO 4 Receive;
- PDO 4 Receive Integer24;
- PDO 4 Transmit;
- PDO 4 Transmit Integer24;
- SDO Integer8;
- SDO Integer16;
- SDO Integer24;
- SDO Integer32;
- Status variable.

5.1.1 Command area

The Command area is a write only area that allows the terminal to send the following NMT commands to the device:

- Start remote node;
- Start remote nodes;
- Stop remote node;
- Stop remote nodes;
- Enter pre-operational node;
- Enter pre-operational all;
- Reset node;
- Reset all nodes;
- Reset node communication;
- Reset all nodes communication.

5.1.2 PDO 1/2/3/4 Receive area

The PDO Receive area is a read/write area that allows the terminal to send real time data to the device. The following data types are available:

- Integer8;
- Integer16;
- Integer32;

Via Padre Masciadri, 4/A
I – 22066 Mariano Comense (CO)
Tel. ++ 39.031757400 Fax ++ 39.031751777
web: www.esahmi.com email: info@esahmi.com

- String.

The “offset” field must be specified. It indicates the PDO byte from which the terminal starts to read the value.

5.1.3 PDO 1/2/3/4 Receive Integer24 area

The PDO Receive Integer24 area is a read/write area that allows the terminal to send real time data to the device using the 24-bit format.
The following data types are available:

- Integer24.

The “offset” field must be specified. It indicates the PDO byte from which the terminal starts to read the value.

5.1.4 PDO 1/2/3/4 Transmit area

The PDO Transmit area is a read only area that allows the terminal to receive real time data from the device.
The following data types are available:

- Integer8;
- Integer16;
- Integer32;
- String.

The “offset” field must be specified. It indicates the PDO byte from which the terminal starts to read the value.

5.1.5 PDO 1/2/3/4 Transmit Integer24 area

The PDO Transmit Integer24 area is a read only area that allows the terminal to receive real time data from the device using the 24-bit format.
The following data types are available:

- Integer24.

The “offset” field must be specified. It indicates the PDO byte from which the terminal starts to read the value.

5.1.6 SDO Integer8 area

The SDO Integer8 area is a read/write area that allows the terminal to exchange SDO data with the device using the 8-bit format.
The following data types are available:

- Integer8;
- String.

The “index” and “subindex” fields must be specified.

5.1.7 SDO Integer16 area

The SDO Integer16 area is a read/write area that allows the terminal to exchange SDO data with the device using the 16-bit format.
The following data types are available:

- Integer16;
- String.

The “index” and “subindex” fields must be specified.

5.1.8 SDO Integer24 area

The SDO Integer24 area is a read/write area that allows the terminal to exchange SDO data with the device using the 24-bit format.
The following data types are available:

- Integer24;
- String.

The “index” and “subindex” fields must be specified.

5.1.9 SDO Integer32 area

The SDO Integer32 area is a read/write area that allows the terminal to exchange SDO data with the device using the 32-bit format.
The following data types are available:

- Integer32;
- String.

The “index” and “subindex” fields must be specified.

5.1.10 Status Variable area

The Status Variable area is a read only area that allows the terminal to display its status or the device status.
The following data types are available:

- Terminal CAN status;
- Device CAN status.

The terminal CAN status variable can assume the following values:
- 0: terminal is in OPERATIONAL state;
- 1: terminal is in PRE-OPERATIONAL state;
- 2: terminal is in WARNING state;
- 3: terminal is in BUS-OFF state.

The device CAN status variable can assume the following values:
- 4 : device is in STOPPED state;
- 5 : device is in OPERATIONAL state;
- 127: device is in PRE-OPERATIONAL state.