PN# tn133-5.doc - 07/08/2008 - Ver. 1.05

Connecting UniOP to CANopen

This Technical Note contains all the information required to connect the UniOP panels as devices in a CANopen and to take advantage from the advanced data access options supported by the UniOP CANopen communication driver.

Important: *this Technical Note applies to the CANopen communication driver identified by the name 'CANopen' and included in the Designer file UPLC107.DLL. This communication driver requires an operator panel of hardware type –0045 or newer equipped with a communication module type TCM09.*

Contents

1. Introduction

The UniOP CANopen communication driver lets you add one or more UniOP operator panels as devices in a CANopen network.
UniOP has been equipped with a special object model that lets you take advantage from the advanced user interface features of the UniOP products, while retaining the simple networking concepts supported by the CANopen network.
The basic idea is to center the exchange of data between the controller and the device on a Local Database defined in the UniOP device. The controller will access the Local Database by means of properly defined PDO's and SDO's. UniOP can access the Local Database directly.
This Technical Note will describe in detail how to implement this connection.

2. UniOP Interface to the CANopen Network

To connect UniOP to the CANopen network a subset of the CIA Work Draft 403 has been implemented. According to the classification introduced in this draft profile, UniOP operates as a 'more skilled simple HMI'.
PDO's are available to enable the controller to access the data resident in the HMI device. Definition of the communication objects can be found in the draft document DSP-403 V. 1.0.

The UniOP local database is an array of 254 individually addressable double word variables; variables can be accessed for reading and writing by the controller using the appropriate read and write objects.

The system architecture is shown in Figure 1.

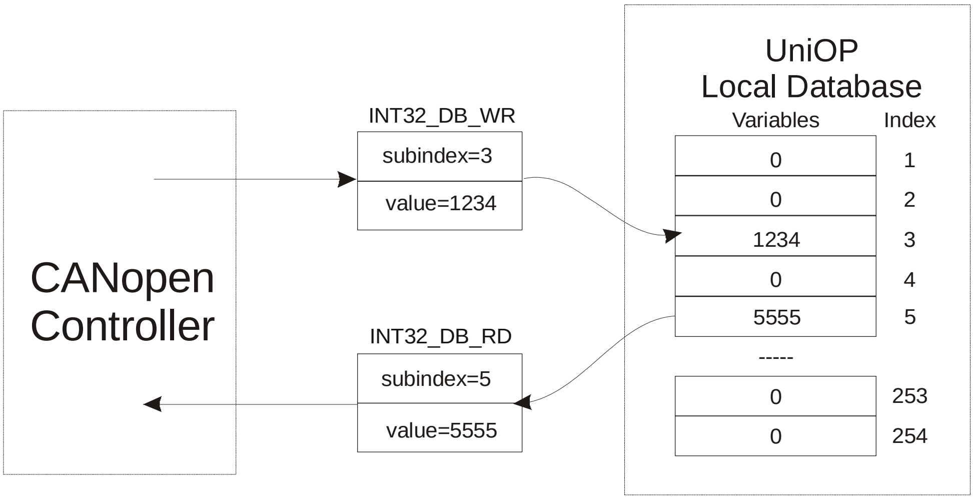
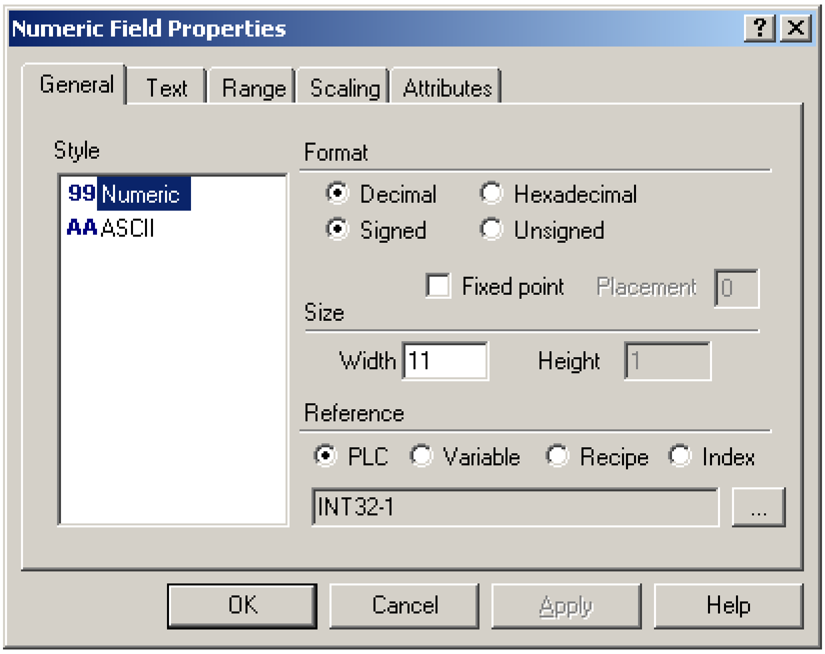


Figure 1 – Communication Model

2.1 The Local Database

The UniOP device contains an internal database composed of an array of 254 doubleword variables. Each variable in the Local Database can be individually addressed by the UniOP firmware. The UniOP application can reference variables in the Local Database to display them on the screen and/or to change their value with the data entry procedure.
Note also that all of the UniOP functions (Keyboard Macro Editor, alarms, RDA, mailbox, data transfer) will work on the Local Database.
The figure below shows how the address of a variable in the Local Database can be specified in the Designer software.



Style: 99 Numeric, AA ASCII

Format: ● Decimal ○ Hexadecimal ● Signed ○ Unsigned

□ Fixed point Placement 0

Size: Width 11 Height 1

Reference: ● PLC ○ Variable ○ Recipe ○ Index

INT32-1 ...

Figure 2 – Entering the address of a variable

The content of the Local Database is initialized to all zeroes when UniOP starts. It is responsibility of the CANopen controller to fill it with all the necessary values.

Whenever UniOP changes the value of a Variable in the Local Database, the CANopen controller is notified of the change with the second Transmit PDO (new data ready).
Another PDO message with the New Data Read flag reset to zero will be sent after a SDO read operation has been issued by the master to read the changed value.

PDO messages are generated also when the RDA area contents changes; please note that changes in the Status area are not notified via any PDO message; Status area is excluded from the the PDO mechanism notification in order to reduce the system communication overhead due to the Real Time Clock information that is changing once per second.

PDO messages are generated in the same way when Mailbox changes when its contents changes due to any write session started by UniOP.

Note: *SDO write sessions do not generate the corresponding PDO message for notification.*

A special mechanism for handshaking handling is implemente starting from CANopen driver DLL V5.05 – Drv. V4.09 and it can be enabled in Controller Setup dialog box checking “Enable Handshaking with Client on Write operations”.

In case the UniOP is performing multiple write operations on the Internal Database, due for instance to a Recipe Download action, the Client would be normally notified with a sequence of PDO messages, one next to the other, containing each one the offset of the modified variable.ù

Note: *a Write Operation can be classified as a “Multiple” one when it is longer than 4 bytes.*

To better handle this case, a new handshaking option has been implemented.

If the handshaking is enabled, the PDO notification flow will be as shown in the table:

	Action on HMI	Message to Client from HMI	Message from Client to HMI
Time	Multiple Write Operation Starting from Variable **n** to Variable **m**		
	Timeout Timer is Started	PDO: SubIdx = **n**, NewDataReady = **1**	
	Timeout Timer is Stopped		SDORead: SubIdx = **n**
	Timeout Timer is Started	PDO: SubIdx = **n+1**, NewDataReady = **1**	
	Timeout Timer is Stopped		SDORead: SubIdx = **n+1**
	Timeout Timer is Started	PDO: SubIdx = **n+2**, NewDataReady = **1**	
	Timeout Timer is Stopped		SDORead: SubIdx = **n+2**
	…	…	
	…		…
	Timeout Timer is Started	PDO: SubIdx = **m**, NewDataReady = **1**	
	Timeout Timer is Stopped		SDORead: SubIdx = **m**
		PDO: SubIdx = **m**, NewDataReady = **0**	

Any new PDO message will be sent to the Client only after an SDORead message over the changed Index (same SubIndex) is received from the client.
The Handshaking is dependent on a timeout timer restarted at each new PDO. The Timeout time is programmable in the Controller Setup dialog as shown in Figure 3.

2.2 ASCII Data Presentation

Information in the internal data base can be represented on screen also in ASCII format.

Note: *The support for ASCII data presentation has been introduced in the CANopen driver starting from version DLL V5.04 – Drv. V3.06*

Since the internal integer database is composed of double words, each of them can store up to 4 ASCII characters.

Address Index	ASCII	Hexadecimal
0	ABCD	0x44434241
1	EFGH	0x48474645

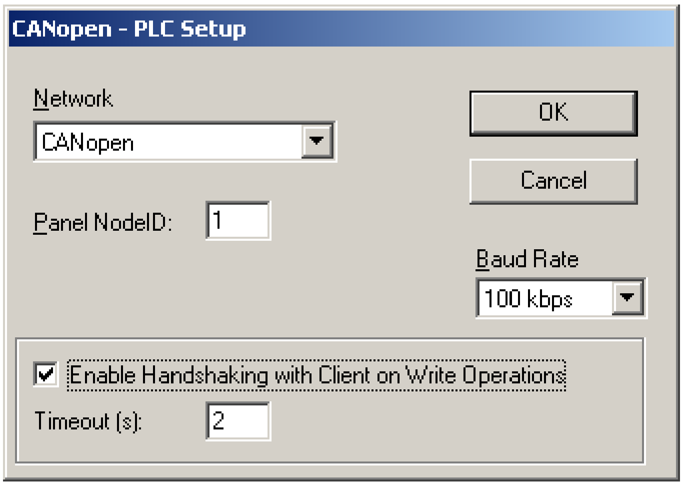
Table 1

Table 1 describes the equivalence between the hexadecimal and the ASCII representation. The least significant byte of the double word contains the ASCII code of the first character of the string.
To display a string in an ASCII field programmed in the project file, a SDO write operation has to be used. Because current implementation of the CANopen protocol does not support the “block transfers”, a maximum of 4 bytes (one double word) at a time can be written to the internal panel database. Strings longer than 4 characters need more than one SDO write operation.

3. Configuring UniOP for CANopen with Designer

To configure a MMI panel for use with the CANopen protocol with the Designer, follow the procedure described in this chapter.

1) Select the option 'Project/Change Communication Driver' and choose 'CANopen'



CANopen - PLC Setup

Network: CANopen

Panel NodeID: 1

Baud Rate: 100 kbps

☑ Enable Handshaking with Client on Write Operations

Timeout (s): 2

Figure 3 - Controller Setup

2) A UniOP panel in a CANopen network is a device that exchanges data with a single controller. Every UniOP in the CANopen network must be assigned its own unique Node address. You can do this by entering a number in the 'Panel NodeID' edit box in the Controller Setup dialog box.

3) Automatic recognition of the baud rate is not supported. Communication speeds up to 1 Mb are supported. Enter the appropriate baud rate in the 'Baud Rate' list box in the Controller Setup dialog box.

3) The Real Time Clock information in the Reserved Data Area (RDA) is coded in BCD

4) The Current Page and the Page Request registers in the RDA are coded in binary.

Note that you can attach more than one UniOP to a single CANopen controller.

Appendix 1 - Communication Error Codes

Current communication status is displayed in the System Menu of the UniOP.
The error status is described by a message and a numeric error code. The number shows the code of the current communication error and, if the communication is correct, the code of the last error encountered. When the error code 0 is shown, it means there have been no communication errors since this system start-up.

Code	Description	Notes
0	No error	There are no communication errors and there have been no errors since start-up.
05	Timeout	The controller has stopped sending NodeGuarding messages) after initiating the NodeGuarding procedure or the Timer of the Handshaking protocol is elapsed
07	General communication error	Internal software error

Table 2 – Communication Error Codes

Appendix 2 - Technical Data and Connection Information

The main technical information on the UniOP CANopen interface using TCM09 is shown in the table below:

Baud rate	100 Kb, 125 Kb, 250 Kb, 500 Kb, 800 Kb and 1 Mb
Slave address	Software configurable
Optical insulation	Yes
Communication port	9 pin female sub-D type

The Aux Port is used for the CANopen communication. When the TCM09 communication module is plugged, the Aux Port becomes a CANopen interface.

Use cable drawing CA149 as a reference for connecting of the UniOP panel to the CANopen network.

Software Requirements and Compatibility

This version of CANopen communication driver is contained in the Designer DLL file UPLC107.DLL. The initial release level is 3.03 for the communication driver and 5.02 for the DLL (both version numbers are displayed in the Change Controller Driver dialog box of the Designer software).
A communication module of type TCM09 is required. The UniOP panel must have hardware type – 0045 and firmware version number 4.10 or higher to support the TCM09 modules.

Appendix 3 - The Object Model

UniOP has been modelled as a ‘More skilled simple HMI’ according to the classification introduced in the CiA draft standard DSP-403 V.1.0.

Note that object numbers are not always consistent with specifications in the draft standard.

The supported objects are:

1st Transmit PDO	not implemented
2nd Transmit PDO	(fixed mapping to Object 9000h)

SDO Server permits the access to the following objects:

DEVICE_TYPE	0x1000	READ
ERROR_REGISTER	0x1001	READ
NUMBER_OF_PDOS	0x1004	READ
DEVICE_NAME	0x1008	READ
DEVICE_SW_VERSION	0x100A	READ
NODE_ID	0x100B	READ
NODE_GUARD_TIME	0x100C	READ/WRITE (default value 0)
LIFETIME_FACTOR	0x100D	READ/WRITE (default value 0)
FIRST_RX_PDOPAR	0x1400	NOT IMPLEMENTED
SECOND_RX_PDOPAR	0x1401	NOT IMPLEMENTED
FIRST_RX_PDOMAP	0x1600	NOT IMPLEMENTED
SECOND_RX_PDOMAP	0x1601	NOT IMPLEMENTED
FIRST_TX_PDOPAR	0x1800	NOT IMPLEMENTED
SECOND_TX_PDOPAR	0x1801	READ
FIRST_TX_PDOMAP	0x1A00	NOT IMPLEMENTED
SECOND_TX_PDOMAP	0x1A01	READ
SERVER_SDO_PAR	0x1200	READ
DIG_IN_MODULE	0x6000	NOT IMPLEMENTED
DIG_OUT_MODULE	0x6200	NOT IMPLEMENTED
INT32_DB_RD	0x6700	READ
INT32_DB_WR	0x8100	WRITE
FLOAT_DB_RD	0x7100	READ
FLOAT_DB_WR	0x8500	WRITE

- The CAN master must send a NMT Start Operation broadcast command. UniOP needs this command to go to operational state.

- The controller may use Node Guarding with UniOP. Upon reset Node Guarding checking is disabled in UniOP. Node Guarding is enabled after receiving the first Node Guarding message. After that, if controller stops doing Node Guarding, then UniOP will report communication error.