TINE-based <u>CANopen Manager</u>

Piotr Bartkiewicz MCS-1, DESY, Hamburg

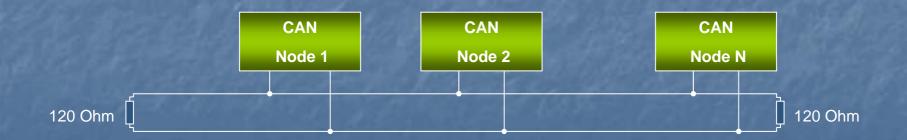
Agenda

CAN fieldbus in a very small nutshellSome CANopen protocol basic definitions:

- Object Dictionary
- NMT State Machine
- Message types: NMT, SDO, PDO, special messages
- Electronic data sheet and Device Configuration File
- Functionality of CANopen Managers (CiA DSP 302)TICOM
 - which functionality of CANopen Manager has been implemented?
 - connection to TINE
 - linking with user code
- Java TICOM Viewer
 - Live demonstration

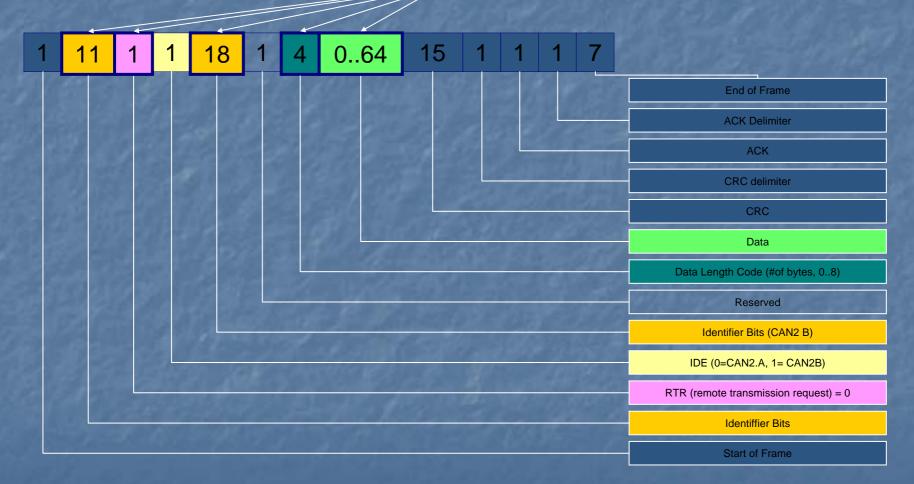
CAN - overview

- CAN: Controller Area Network
- Serial Bus, connects more devices (bus nodes) with only 2 wires
- Multimaster Capability (no master-slave architecture)
- Broadcast: nodes have no address, but messages have identifiers
- No limit for number of nodes due to protocol (only available driver electronics limit the number of nodes on the bus)
- Number of nodes can be dynamically changed (devices can be connected or disconnected when the bus operates)
- Very sophisticated error detection and error handling
- A lot of microcontrollers having a built-in CAN interface available at very low cost.



CAN frame

Only these bits are available for application software



CANopen: the application layer standard

CANopen specification:

application layer

Available are also other application layer standards like:

- DeviceNet
- J1939
- TTCAN
- **_** ...

CAN specification: physical and data link layer

Application Layer

Presentation Layer

Session Layer

Transport Layer

Network Layer

Data Link Layer

Physical Layer

Open Systems Interconnection Model

CANopen – cont. [2]

Very popular in Europe (transportation, medical, industrial machinery, building automation, military) - many devices "plug & play" are available "off-the-shelf"
 CANopen is "open" in three ways:

- The technology is open and does not require any payment or license fees
- Proprietary devices can be combined with CANopen devices into one network (just only ensure, there is no ID collisions)
- only a small set of mandatory functionality and huge set of optional functionality, even very simple microcontrollers can be CANopen compliant.

Software stacks available for reasonable price.

CANopen – cont. [3]

The CANopen standard is maintained and certified by the CAN in Automation (CiA) organization, Munich, Germany (www.can-cia.org)

- Online documentation (some available for members only)
- Conformance tests
- Conferences and trainings

DESY is the CiA member:

 the contact person at DESY-Hamburg site: Stefan May, MPS, tel. (040) 89984636 stefan.may@desy.de

CANopen Basics [1]: The Object Dictionary

- Each CANopen device has to implement the Object Dictionary
- Every process or communication parameter is identified by two numbers: 16-bit Index and 8-bit subindex.
- From the network point of view using an Index and Subindex is the only way to address the variable
- Some Indexes and Subindexes are reserved for predefined variables, which are mandatory for a CANopen node

Index (hex)	Sub-Index (hex)	Object Symboli c Name	Name	Туре	Attr Defaul value		Mandatory	
1000		VAR	Device Type	UNSIGNED32	RO	0	yes	
1001	-	VAR	Error register	UNSIGNED8	RO	0	yes	
1002		VAR	Manufacturer status register	UNSIGNED32	RO	0	no	
1003	-	ARRAY	Pre-defined error field	UNSIGNED32			no	
	0	VAR	Size	UNSIGNED8	RO	0	yes	
the second	1	VAR	Error field	UNSIGNED32	RO	12	No	
Darl	2	VAR	Error field	UNSIGNED32	RO		No	
							••••	
2000		VAR	Sollwert	REAL32	RW	0.0	199	

CANopen Basics [2]: The NMT State Machine

Each CANopen node can be in one of three states:
 Preoperational : node configuration
 Operational : normal operation
 Stopped : network maintenance

CANopen Basics [3]: Message types

NMT (Network Management)

- Used to :
 - drive individual nodes or entire network in pre-, operational, stopped state.
 - reboot individual node or all nodes
 - Report the state of the node:
 - Boot-up message: sent once, just after reboot
 - Heartbeat message: sent periodically, showing the NMT state

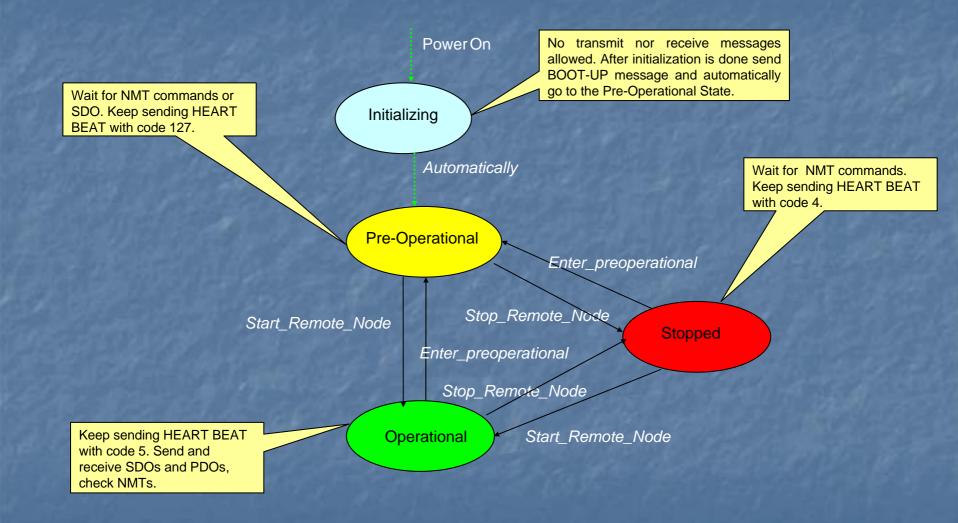
SDO (Service Data Object)

- Used to:
 - configure individual node (by writing to the Object Dictionary)
 - examine the node's configuration (by reading the Object Dictionary)
- Not recommended for process data transfer ("slow", protocol overhead)
- Available in pre- and operational state
- PDO (Process Data Object)
 - Used for exchanging process data between nodes, no CANopen protocol overhead
 - Available in operational state

Special messages:

- EMCY: emergency messages (extended error information)
- SYNC: triggering PDOs (triggering data capture and data telegrams)
- TIME: timestamps

CANopen NMT state machine



CANopen: EDS and DCF files

EDS (Electronic Data Sheet):

- Provides the documentation in a standardized way, as a file, similar to MS Windows *.ini file format
- Comes with each CANopen device as a description of Object Dictionary contents
- Used by CANopen monitors, analyzers, configuration tools, CANopen masters
- Common for all devices of the same type

DCF (Device Configuration File):

- the same format as EDS, but different usage
- stores individual setting for each device of given type, like: minimum, maximum, default value of a process data
- used for a CANopen Master to individually configure nodes

Functionality of CANopen Managers (CiA DSP 302)

NMT Master

- control and monitor state of devices
- TIME producer
 - generates messages containing time and date information
- SYNC producer
 - generates a SYNC message for PDOs triggering

Configuration Manager

 Carries out the device configuration process (by reading the appropriate DCF and sending sequence of SDOs) when a boot-up message arrives

SDO Manager

assigns "SDO channels", if more than one is needed

Optional:

- LSS (Layer Setting Service) Master
 - Dynamic assignment of devices ID and selection of bit rate

TICOM

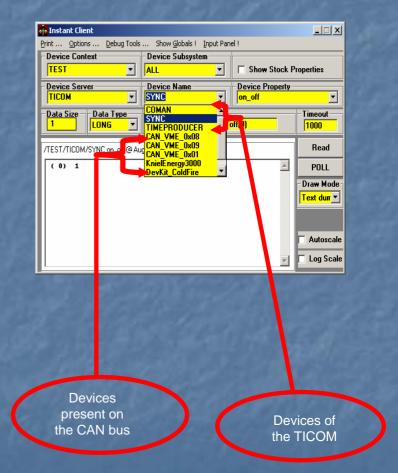
which CANopen Manager functionality has been implemented?

NMT Master TIME producer SYNC producer **Configuration Manager:** parsing DCF files for each node sending SDO sequences in order to configure devices Object Dictionary NOT implemented (yet?): SDO manager

TICOM: connection to TINE [1]

TICOM is a TINE server with:

- 3 default devices (always present)
 - **COMAN:** management of the entire bus:
 - Sending 'global' NMT messages (reboot, enter pre- and operational state, enter stop state)
 - Providing the bus statistics: how many devices, how many in which state
 - SYNC: switching SYNC on/off, defining message period
 - TIME: switching message sending on/off, defining message period, setting the current time
- CAN node image devices: devices corresponding to the physical devices connected to the CAN bus (list defined by the set of DCF files found in the TICOM directory)



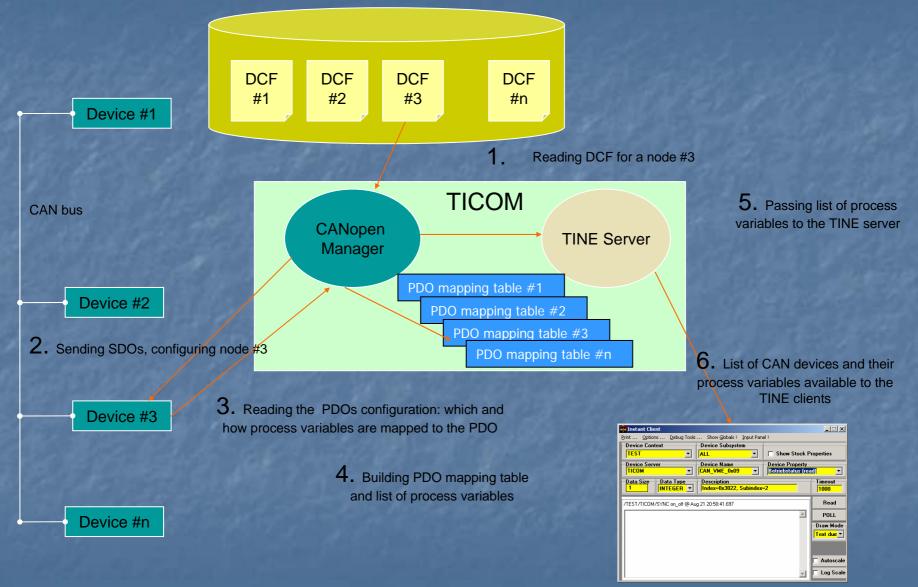
TICOM: connection to TINE [2]

- Each CAN Node Image device offers:
 - fixed properties (common for all CAN devices)
 - Basic device information: device type, manufacture data, soft- and hardware version etc.
 - Heartbeat (or node guard) control, NMT control and state readout
 - List of process variables mapped to PDOs (separately for RPDO and TPDO)
 - Simple access (RW) to the device's object dictionary
 - Dynamically built properties, which represent process variables (depending on CAN device)



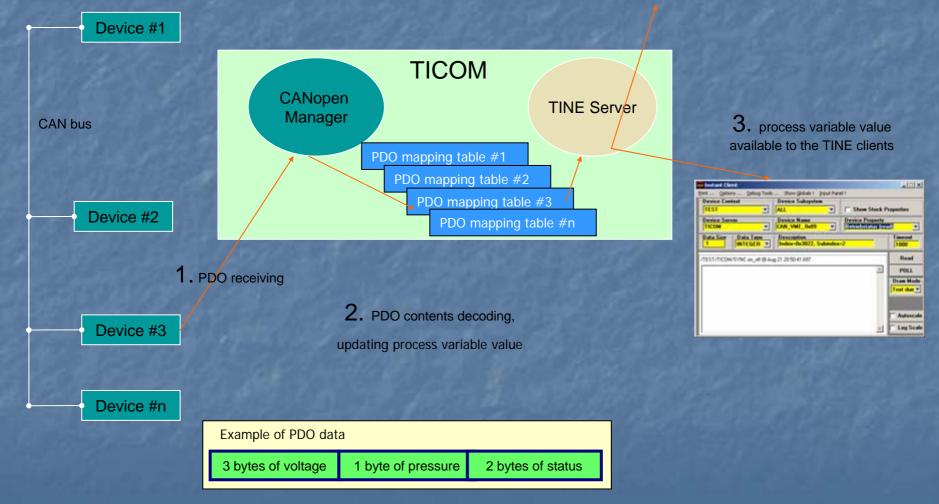
- Device Context TEST	•	Device Subsystem	Show Stock Properties
- Device Server - TICOM	_	Device Name CAN_VME_0x09	Device Property Betriebstatus (read)
	ata Type TEGER 🔽	Index=0x3022, Subinde	Code requested (write) ax= HbeatControl IdInfo Istwert set (read) NGuard_onOff NGuard_period NMState Draw Mo Text dun

TICOM to TINE: configuration data flow



TICOM to TINE: process data flow

ExecLink("/TEST/TICDEMO/CAN_VME_09", "Betriebstatus (read)", &dout, NULL, CA_READ)



TICOM: linking with user code

 TICOM activates periodically (~milliseconds) an empty 'UserCode' object: user can fill it with his own code.
 Set of APIs provides access to the TICOM devices and CAN Node Images, their Object Dictionaries etc.

Running the TICOM

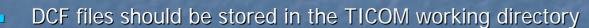
Currently available version for Linux (incl. ELINOS):

- Tested on a desktop PC and PC104
- So far works with:
 - PC104 CAN card
 - USB to CAN interface delivered by the PEAK Systems (the same driver)

usage: ticom [options...] options:

-n <number> : PEAK can device number (default: 32 (PCAN_USB) when omitted)
-d : run in debug mode
-s <name> : register as TINE server <name>
-h : help

ex: ticom – s TICDEMO





Java TICOM Viewer [1]

selection of TICOM servers

overview: how many devices, in which state etc.

- comfortable access to TICOM devices
- possibility to issue 'global' NMT commands for entire CANopen bus
- List of CAN devices
 - access to individual CANopen device:
 - NMT read/change state, heartbeat timing manipulation
 - Access (RW) to the Object Dictionary
 - Access to all process variables:
 - Plots for transmit PDOs
 - New value entry for read PDOs

Java TICOM Viewer [2]

IICOM Viewer, version: 1.0	I	🛃 TICOM Viewer, version: 1.0		- 🗆 🗵
Device Context: TEST 💌 Device Subsystem: ALL 💌 Device Server: TICOM		Device Context: TEST	Device Subsystem: ALL Device Server: TICOM	
CANagen bus SYNC producer The Producer CAN_WHE_0x03 CAN_WHE_0x03 KnieEnergy3000 DevKit_ColdFire CAN_WHE_0x07 DevKit_ColdFire CAN_WHE_0x07 Disspect: Z50000 bits/sec		CANopen bus Time Producer CAN_VME_Dx08 CAN_VME_Dx08 CAN_VME_Dx01 KnielEnergy3000 DevKit_ColdFire CAN_VME_Dx07	Current SYNC generator settings SYNC messages enabled: YES SYNC message period: 3000 ms Set Enable SYNC messages: SYNC message period: 50 SYNC message period: 500 1000 500 1000 500 Set	
status 1		Success		
The second s	_			

Java TICOM Viewer [3]

CANopen bus manipulation

🕌 TICOM ¥iewer, version: 1.0		- 🗆 🗵
Device Context: TEST	Device Subsystem: ALL 💌 Device Server: TICOM	
CANopen bus SYNC producer Time Producer CAN_VME_0x08 CAN_VME_0x08 CAN_VME_0x01 KinelEnergy3000 DevKit_ColdFire CAN_VME_0x07	Nodes state overview Total number of nodes: 6 in preoperational state: 4 in operational state: 0 in stopped state: 0 not responding (no heartbeat): 2 Set nodes state reset All nodes set preoperational state set operational state Set apped mode set stopped mode	
Success		

Java TICOM Viewer [4]

CANopen device basic information

STICOM Viewer, version: 1.0	0		STICOM Viewer, version: 1.0		<u>_</u> _×
Device Context: TEST	Device Subsystem: ALL Device Server: TICOM		Device Context: TEST	▼ Device Subsystem: ALL ▼ D	Device Server: TICOM
CANopen bus	Node Basic Information Node Data		CANopen bus SYNC producer Time Producer	Node Basic Information Node Data	
🖻 💼 Devices	Node name: CAN_VME_0x09		🗄 🖷 🧰 Devices	Node name:	DevKit_ColdFire
CAN_VME_0x08	Device type: 401		CAN_VME_0x08	Device type:	401
CAN_VME_0x01	Hardware version: 1.0.0		CAN_VME_0x01	Hardware version:	1.0.0
MielEnergy3000	Software version: 1.0.2		DevKit_ColdFire	Software version:	1.0.2
CAN_VME_0x07	Vendor ID: 0		CAN_VME_0x07	Vendor ID:	0
	Product code: 0			Product code:	0
	Revision number: 0			Revision number:	0
	Serial number: 0			Serial number:	0
	Manufacturer device name: CAN_VME			Manufacturer device name:	ColdFire Test Board
	NMT Counters Timestamp 0 Reset 6106 21.08.2007, 20:58:48 Preoperational 0 Operational 0 Operational 0 Stop No heartbeat Stop Current settings Enable: period: 0 period: 500 ms Set 0 500 1000 1500 2000			enabled: YES	0:58:48 Preoperational 1:56:01 Operational Stop
Success			Success		

Java TICOM Viewer [5]

access to the device's Object Dictionary and process variables

STICOM Viewer, version: 1.	0						
Device Context: TEST	Device Subsystem: ALL	Device Server: TICOM	-				
CANopen bus SYNC producer Time Producer CAN_VME_0x08 CAN_VME_0x09 CAN_VME_0x01 CAN_VME_0x09		a Data jubindex: 0x 0					
KnielEnergy3000	Read			Plot of ADC channel 1		X	
CAN_VME_0x07				Write		/TICOM/DevKit_ColdFire/ADC channel 1	
	Status:						
	Transmit Process Variables (TPDO)				Index: 0x2001 Subindex: 0x2 Data format: SHORT Current value: 180	350	
	Variable	Value	Update time	Plot	Updated at: 21.08.2007, 21:59:28.993	ΛΛ	
	ADC channel 0 ADC channel 1	500 179	21.08.2007, 21:59:47.811 21.08.2007, 21:59:47.811	show hide		250 A A	/`\
	ADC channel 1 ADC channel 2	11	21.08.2007, 21:59:48.707	show	[200	
	Button Counter 0	0	21.08.2007, 21:56:01.231	show			\
	Button Counter 1	0	21.08.2007, 21:56:01.231	show		150	
	Button Counter 2	0	21.08.2007, 21:56:01.235	show			
	Button Counter 3	1	21.08.2007, 21:56:01.235	show		100	
	-Read Process Variables	(RPDO) Value	Variable status	Action	Prink	50 0 5 10 15 8r Aur - 21 - 21 - 56 - 59 - FST - 2007	20 25 30 Seconds
	Led outputs	0		send		Ū	,
Success					X-exis time range:	0 600 1200 1800	2400 3000 3600