TICOM

TINE-based CANopen Manager

Piotr Bartkiewicz

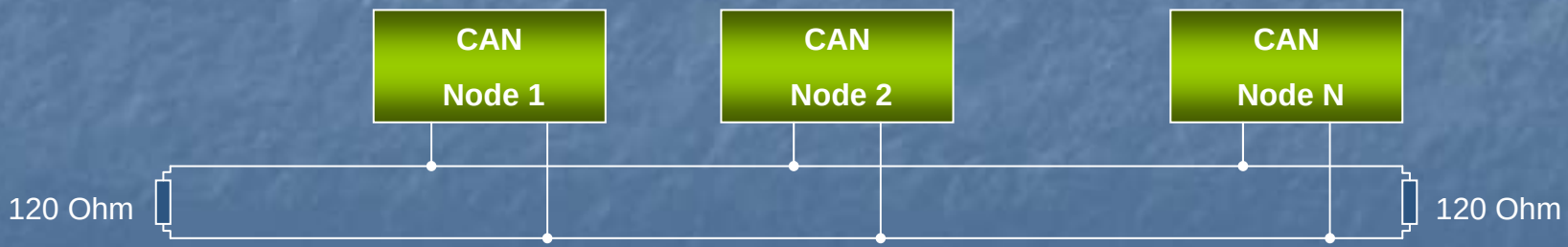
MCS-1, DESY,Hamburg

Agenda

- CAN fieldbus in a very small nutshell
- Some 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
1
11
1
1
18
1
4
0..64
15
1
1
1
7
End of Frame
ACK Delimiter
ACK
CRC delimiter
CRC
Data
Data Length Code (#of bytes, 0..8)
Reserved
Identifier Bits (CAN2 B)
IDE (0=CAN2.A, 1= CAN2B)
RTR (remote transmission request) = 0
Identiffier Bits
Start of Frame

CANopen: the application layer standard

CANopen specification:

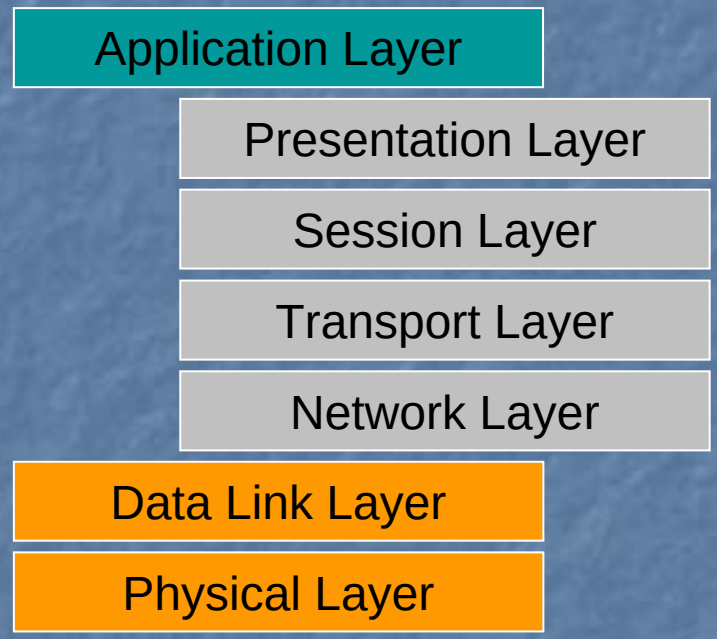
application layer

Available are also other application layer standards like:

- DeviceNet
- J1939
- TTCAN
- ...

CAN specification:

physical and data link layer



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 Symbolic Name	Name	Type	Attr	Default 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
	1	VAR	Error field	UNSIGNED32	RO		No
	2	VAR	Error field	UNSIGNED32	RO		No
....	....	....	....	.....	...	....	....
2000	-	VAR	Sollwert	REAL32	RW	0.0	

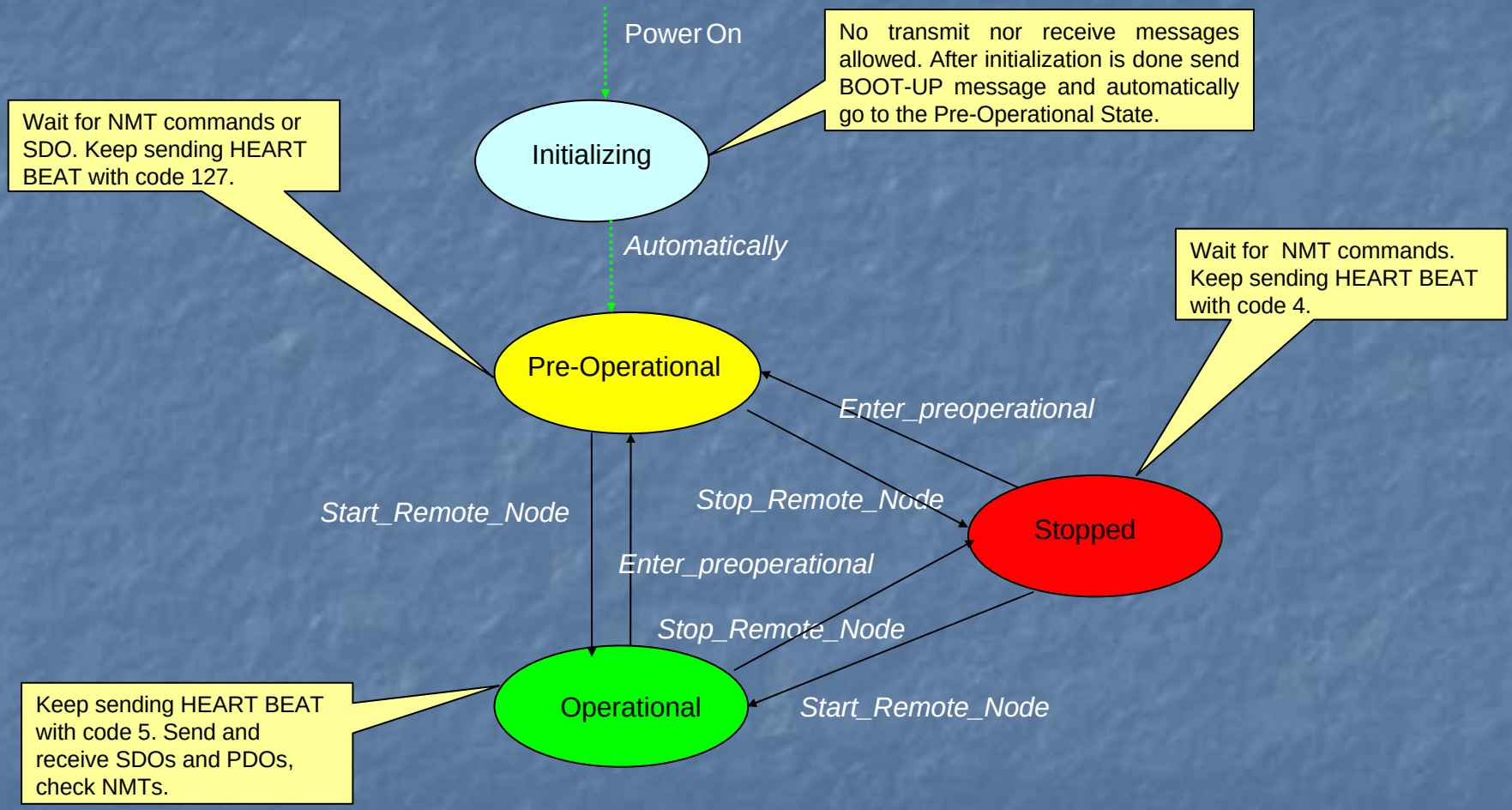
CANopen Basics [2]: The NMT State Machine

- 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

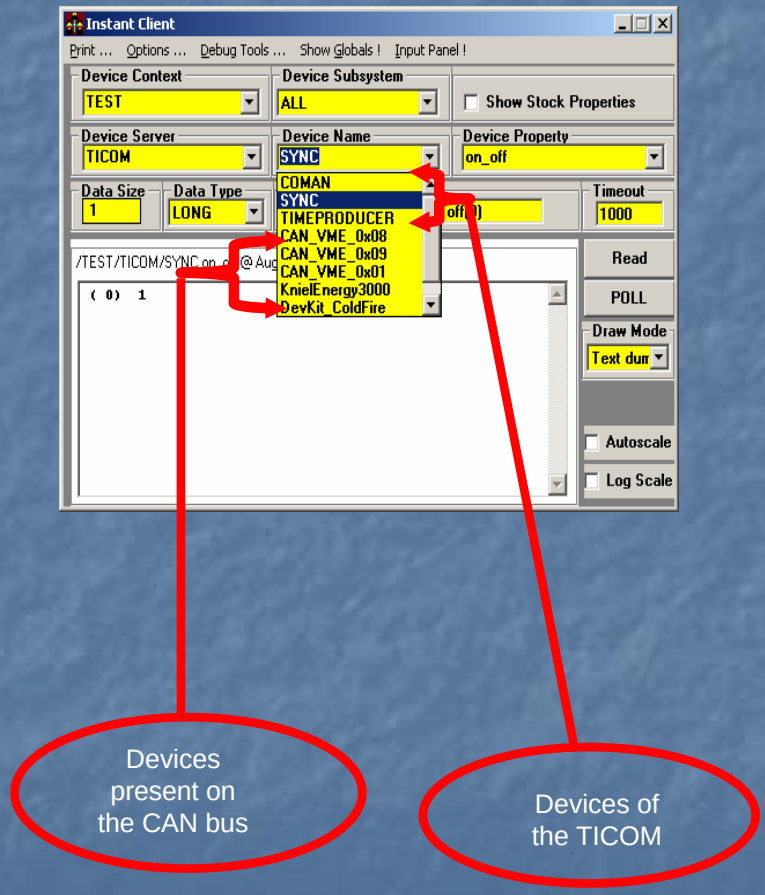
TICOM
which CANopen Manager functionality has been implemented?

- 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]

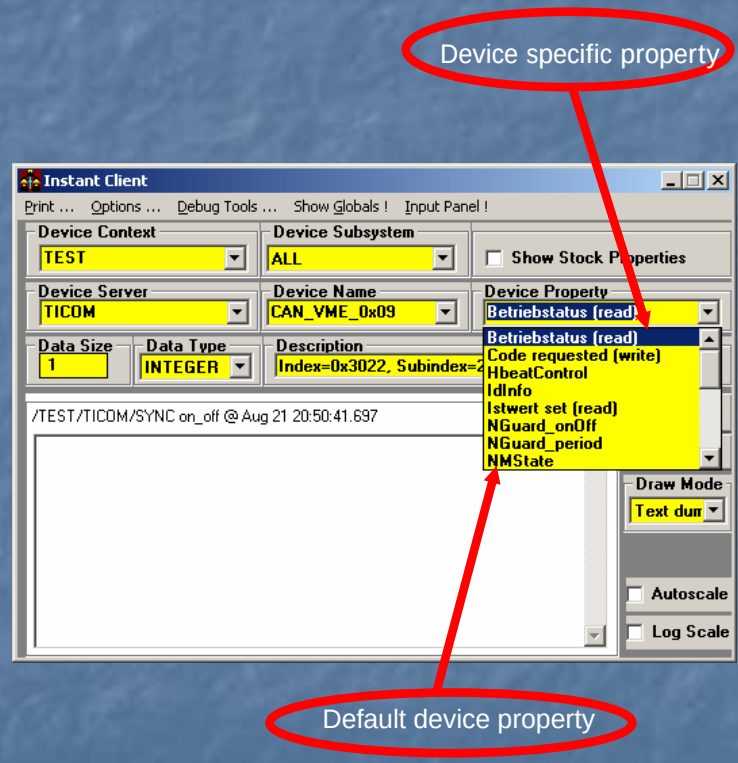
TICOM is a TINE server with:

- 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)

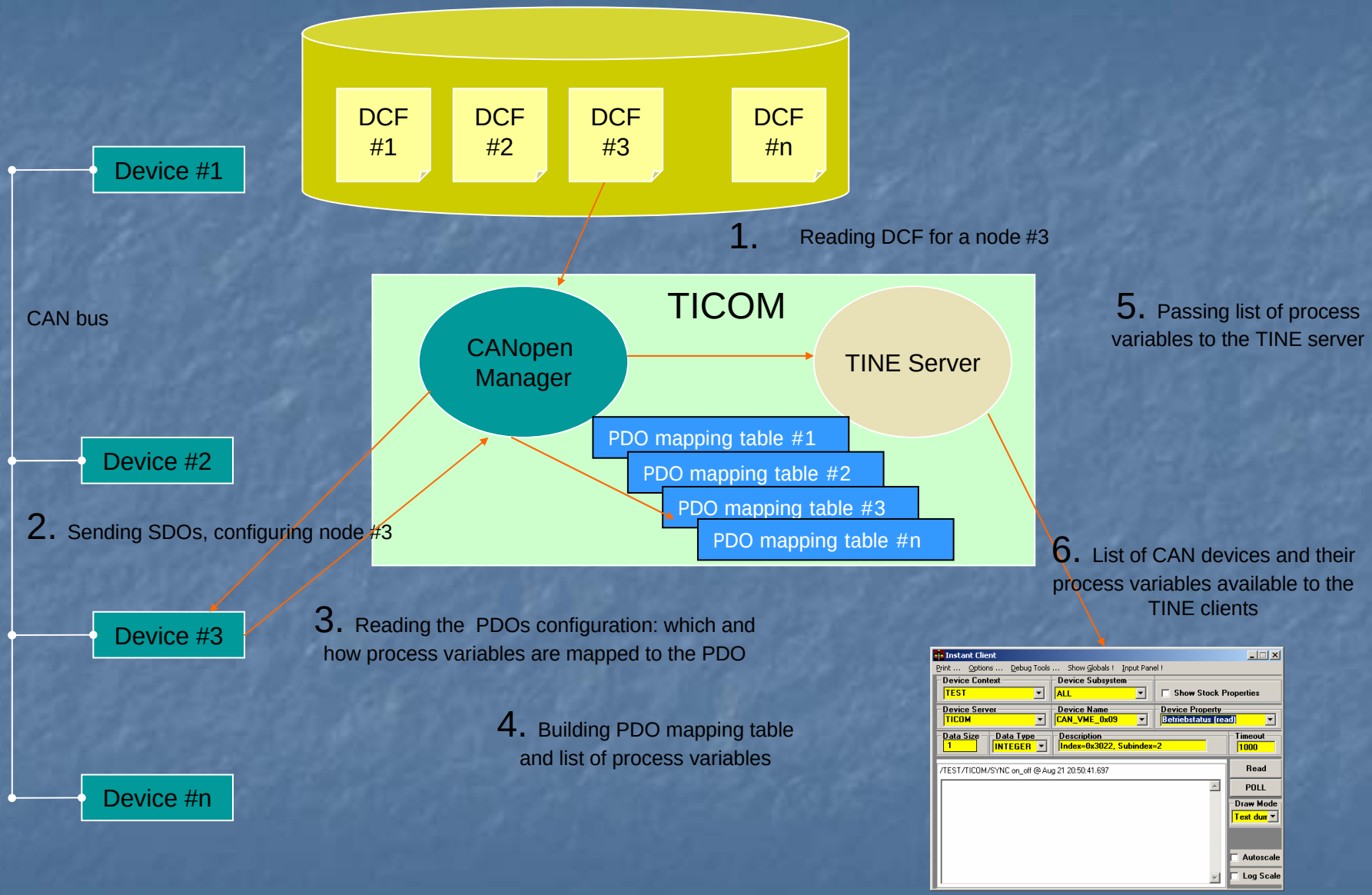


TICOM: connection to TINE [2]

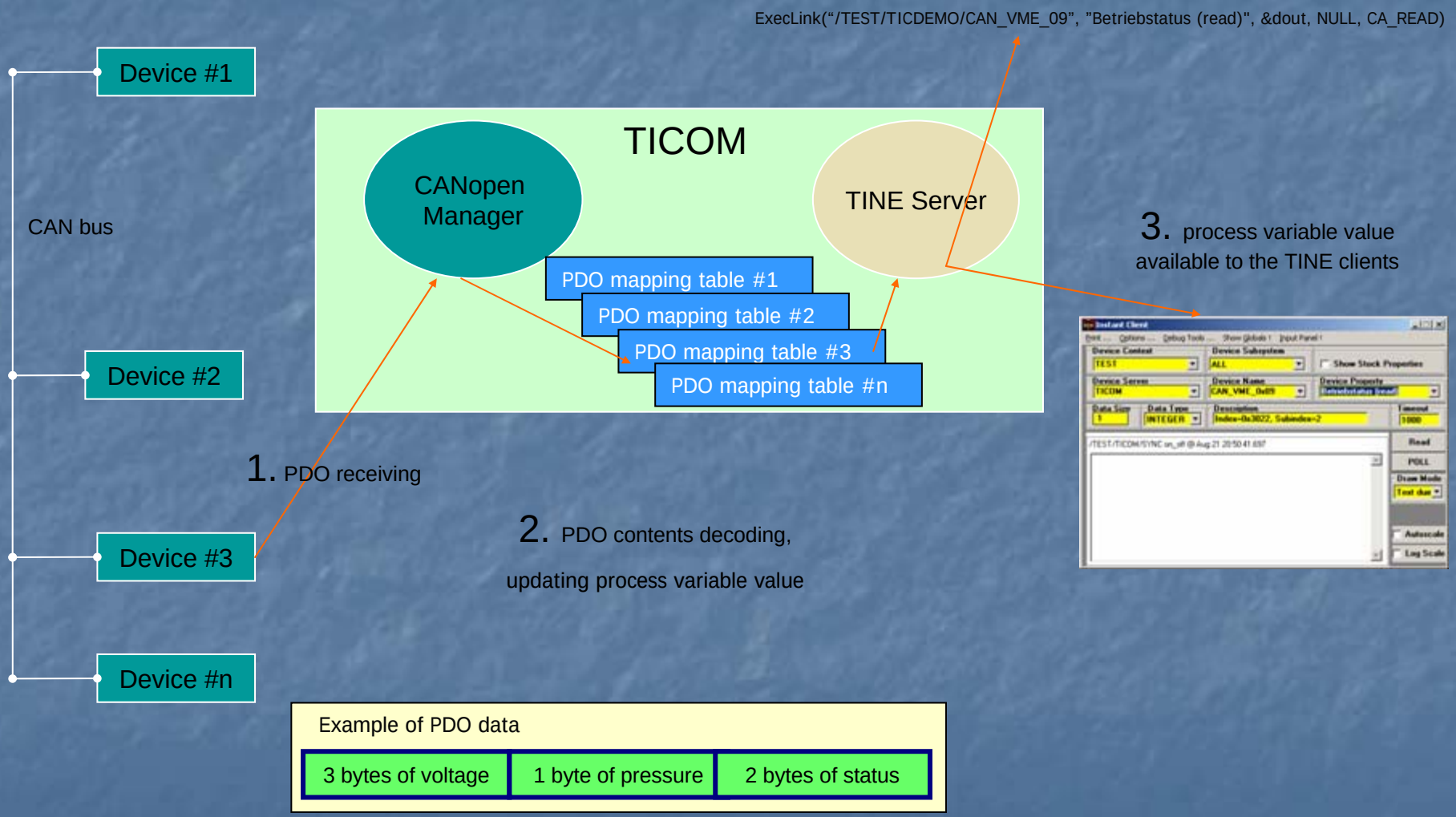
- Each CAN Node Image device offers:
  - fixed properties (common for all CAN devices)
    - Basic device information: device type, manufacture data, soft- and hardware version etc.
    - Heartbeat (or node guard) control, NMT control and state readout
    - List of process variables mapped to PDOs (separately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



TICOM to TINE: configuration data flow



TICOM to TINE: process data flow



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

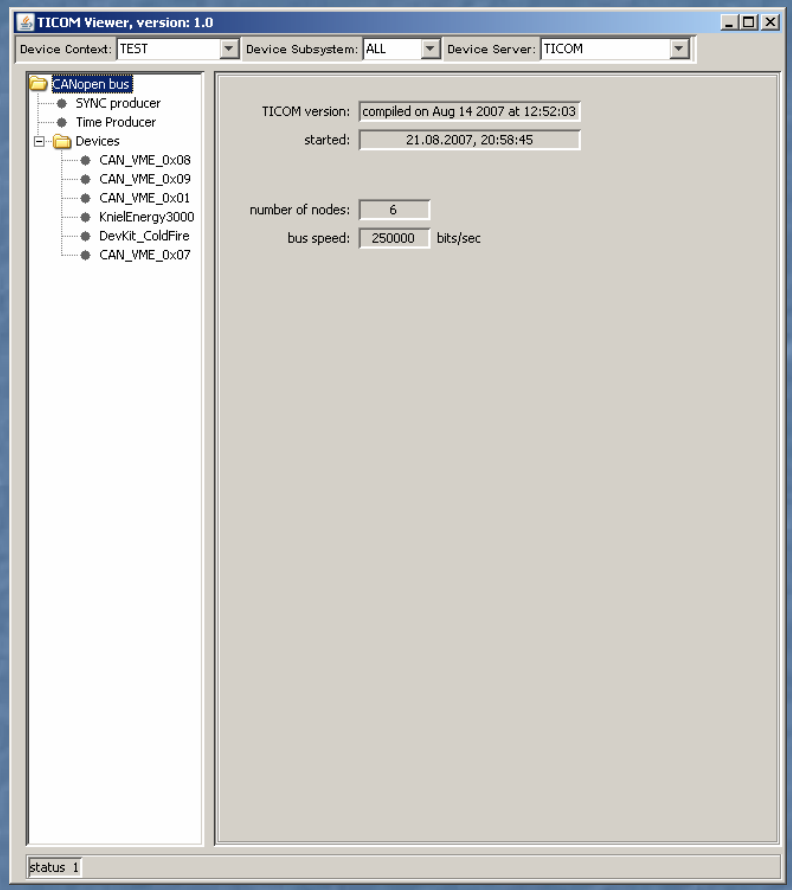
  ex: ticom – s TICDEMO

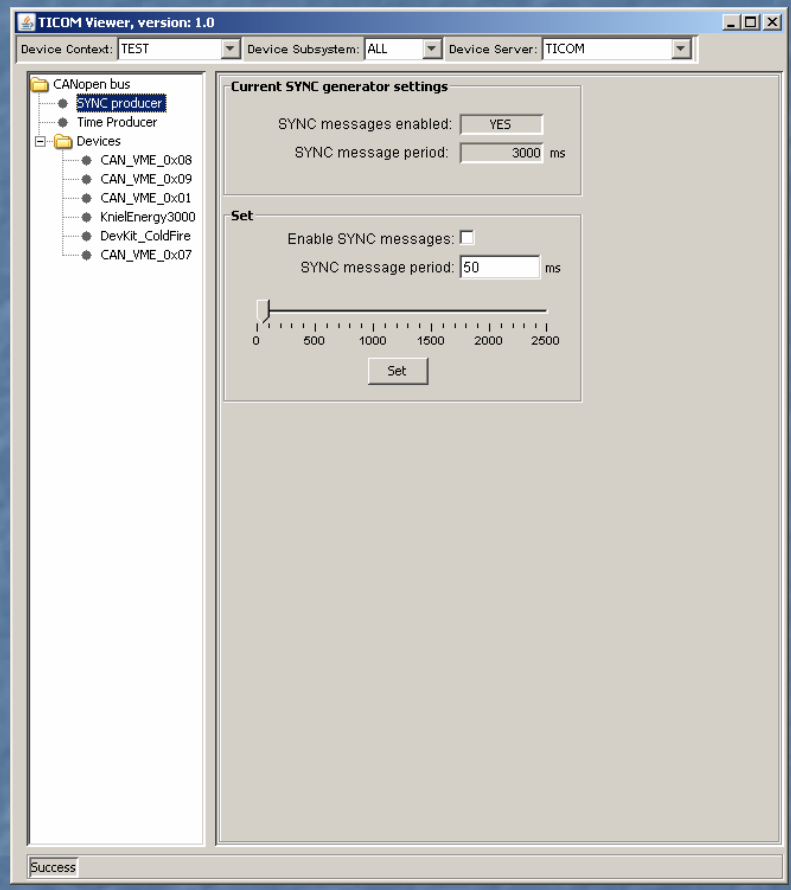
- DCF files should be stored in the TICOM working directory

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

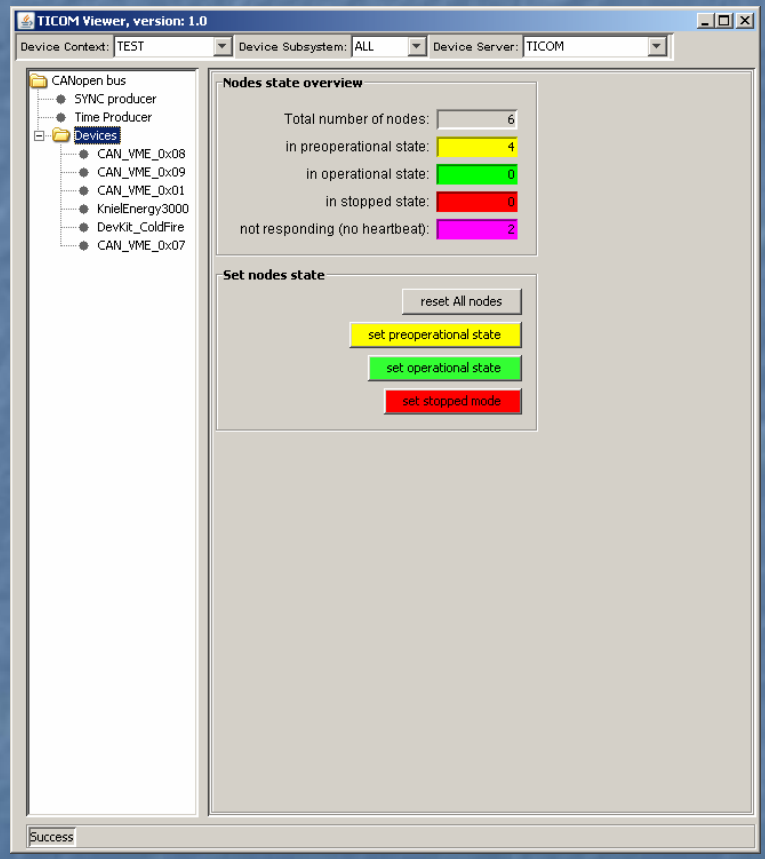
Java TICOM Viewer [2]





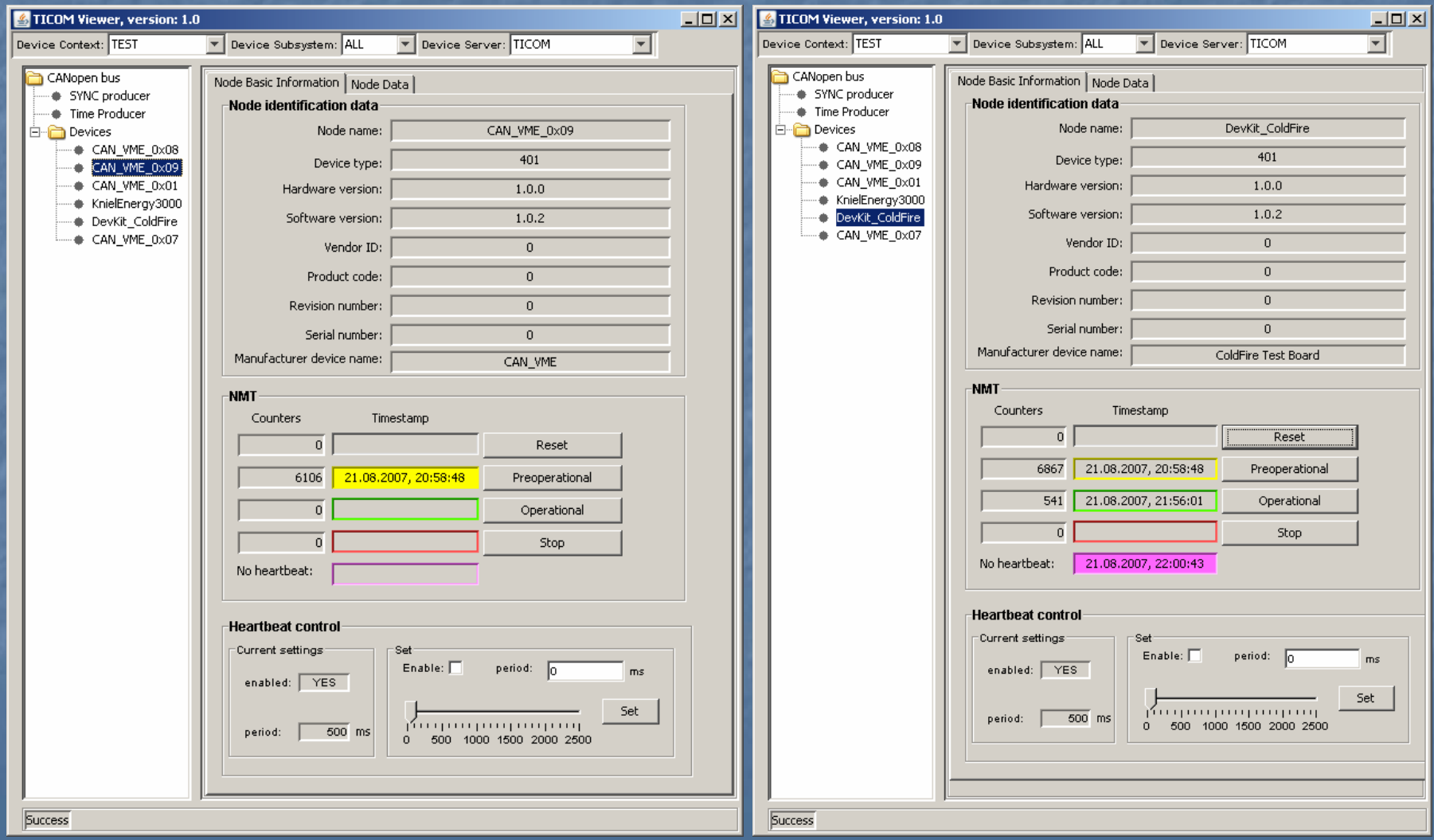
Java TICOM Viewer [3]

CANopen bus manipulation



Java TICOM Viewer [4]

CANopen device basic information



Java TICOM Viewer [5]

access to the device's Object Dictionary and process variables