Socket-CAN Device Support for EPICS IOCs*

C. Burandt†, U. Bonnes, J. Enders, N. Pietralla
Institut für Kernphysik, Technische Universität Darmstadt, Germany

Socket-CAN Device Support
for EPICS IOCs*

C. Burandt†, U. Bonnes, J. Enders, N. Pietralla
Institut für Kernphysik, Technische Universität Darmstadt, Germany

*Work supported by DFG through CRC 634
†burandt@ikp.tu-darmstadt.de

CAN Bus
The In-house Developed Hardware Family
- magnet power supplies, low-level RF, multi-purpose measurement system, ...
- uniform firmware running on microcontrollers for all types of hardware
- connected via CAN bus to PC.

PC Interface Hardware
- PCI/PCIexpress slot cards used in PCs running the EPICS IOCs
- USB interface for on-site diagnostics

Linux Socket-CAN Network Stack

What it is
- part of the Linux kernel main line since version 2.6.25
- some manufacturers supply their own Socket-CAN device driver
- many CAN interface device drivers already included

How it works
- analogous to internet protocols like TCP/IP (protocol family)
- connection endpoint is represented as a BSD socket

Using tudSocketCan Device Support

How to use it
- identical to ethernet devices
- setup of virtual CAN device possible

Software Engineering Features
- approx. 1300 lines of plain C
- build dependencies are limited to header files which are part of the Linux kernel source or EPICS base
- not dependent of the PC CAN interface manufacturer in any way

Supported Record Types
- analog in/out
- long in/out
- binary in/out
- multi binary in/out
- multi binary direct in/out

Not Yet Supported Record Types
- different record types can be implemented in principle
- complex record types like string in and out require transmission of multiple CAN frames, which is different from the record types implemented so far

Debugging of Templates

Example
Request supply voltage by hand and observe the answer with candump and camonitor.

```
$ cathd /dev/hidraw0
$ camonitor
```

Summary

Experience
- first used for the digital low-level rf control system
- in a production environment for two years now
- growing number of different devices controlled through this device support

Future
- rewrite in C++ programming language is proposed
- C++ STL provides many useful components which allow for a much cleaner software design
- improve bus error detection and treatment
- support record types which require multiple CAN frames to be transmitted