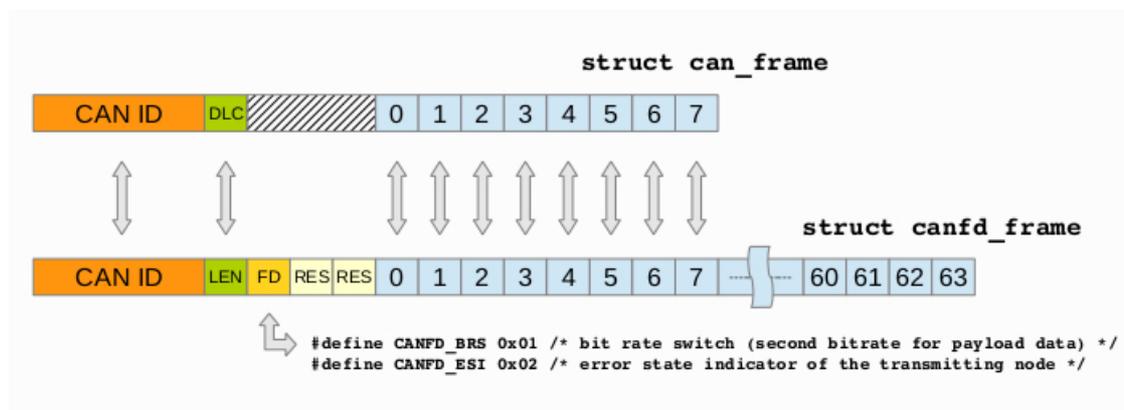


Linux 3.6 supports CAN FD

The CAN FD (CAN with flexible data-rate) capable data structures and programming interfaces have been released for the Linux CAN sub-systems. This enables CAN application programmers to implement and run CAN FD applications on virtual CAN FD interfaces.

ON JUNE 19th 2012 THE LINUX NETWORK MAINTAINER David S. Miller pulled a set of six source code patches into the networking repository, which will be integrated in Linux version 3.6. The CAN FD patches from Oliver Hartkopp (Volkswagen, Germany) have been reviewed by the Linux CAN community and the sixth revision of these patches was finally approved. The integrated functionality to handle CAN FD frames defines the programming interfaces for application programmers as well as for CAN driver developers (when real CAN FD controllers become available). To preserve the binary compatibility for existing Linux CAN applications the socket programming interface has been extended by a CAN FD option, which is disabled by default.

A CAN FD aware application may enable the CAN FD support on a per-socket basis, which allows sending and receiving CAN FD frames as well as "normal" CAN frames on this socket. The data structure for the CAN frame with its eight bytes of payload data was formerly assumed to be a fix point in CAN programming. With the introduction of CAN FD the payload data may consist of up to 64 bytes. In order to preserve the easy handling of CAN frames for application programmers a similar data structure for CAN FD frames has been defined:



The CAN FD data structure has a backward compatible layout, which allows processing all types of CAN frame. When a "normal" CAN frame content is read into the CAN FD structure, it can be accessed as a CAN FD frame. The CAN payload data length 'len' becomes a linear value from 0 to 64, which allows to preserve the known programming concepts, e.g. for loop programming statements. The mapping of the payload length to the DLC (data length code) field is supported by dedicated helper functions and is done on the CAN controller driver level only. This prevents the application programmer from cumbersome and error-prone mapping efforts. Currently, CAN FD applications and tools may be programmed and tested with the upgraded Virtual CAN (FD) interfaces only. When the real CAN FD controllers are released to the public, a second bit-rate configuration for CAN interfaces will be added to the Linux CAN driver infrastructure as well as the possibility to switch the then available CAN FD modes.