

White Paper

Seamless Integration of Dual-port SocketCAN Solution

The Dual-port Solution is a CANbus expansion card based on a modified SocketCAN driver that allows for connection to two separate CAN networks. SocketCAN allows for easy integration of any Linux-based CAN device. This basically enables a plug-and-play CAN solution with several advantages over traditional Linux drivers.

Introduction

Traditionally, not many CAN devices run on Linux operating systems. For those that do however, there have been a tendency to run on simple and limited character drivers. SocketCAN offers a better solution that is based on TCP/IP characteristics. It enables several processes to be run simultaneously and allows for greater portability.

However, the basic SocketCAN driver still has certain limitations. The driver only supports one CAN port from the device. For any applications that need more ports, additional modules would have to be installed, or extensive modifications would have to be made to the API.

The Dual-port Solutions make these modifications to the source code allowing for two independent ports. This paper will dive deeper into the differences between character drivers and SocketCAN, as well the advantages of the Dual-port Solution.

Background

CAN is a message-based, automotive protocol that allows microcontrollers to communicate without the need of a host computer. The signal is sent through a single twisted wire that connects to various devices within the vehicle. It is also widely used in automation and the embedded industry.

SocketCAN is a CAN socket application programming interface (API) consisting of a set of CAN drivers and networking stacks for Linux. It is open source and is based on the Berkeley socket API.

Before SocketCAN was available, applications running on Linux had simple character device drivers. The problem with these character drivers is that they will only allow one process to access the device at any given instant. Changing the CAN device will also mean having to change the driver and consequently reprogram applications to fit with the new settings.

SocketCAN solves these issues by employing a network-like model (similar to TCP/IP) that allows for many applications to simultaneously access the CAN device. The TCP/IP similarity allows users familiar with network programming to more easily learn about SocketCAN. As can be seen in figure 1, the protocol family, which provides a Socket Layer that connects to user space, is built on the Linux network layer, while network device drivers for the CAN controller hardware interact with the Linux network layer.

A network example

Using a character driver would be similar to being unable to run any other application connected to the internet when running a web browser. With a network driver however, the user can run a web browser, download files and run an anti-virus program, all at the same time.

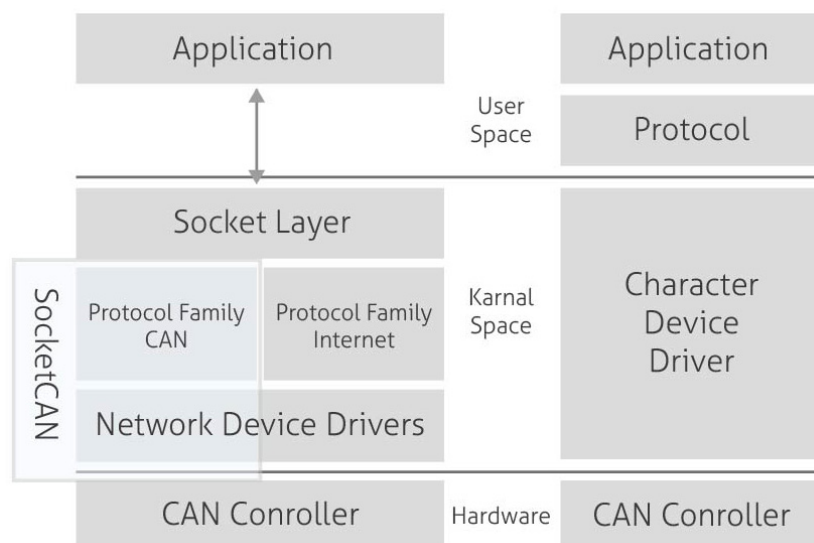


Figure 1: SocketCAN vs Character Driver

Dual-port Solution vs Standard SocketCAN

With standard SocketCAN solution, only one CANbus port is available. Any further expansion could be both costly and time-consuming as the system integrator would have to modify the driver, or install a second module.

The Dual-port Solution comprises a CANbus expansion card with a modified SocketCAN API (see figure 2). Building on the standard SocketCAN open source code, this miniature form factor provides two ports that can link to up to two CANbus networks. This is made possible by altering the SocketCAN driver and daemon.

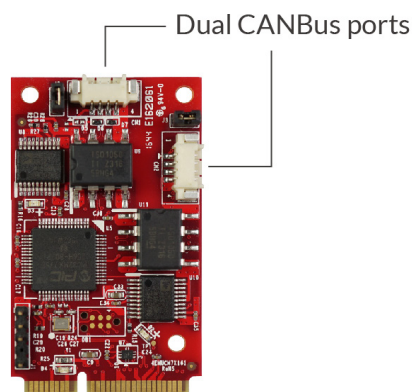


Figure 2: Dual-port CANbus Module

Dual ports offer redundancy, share flow and the option to connect to independent CANbus networks.

Switching to Dual-port Solution

There are two scenarios to consider when switching to a Dual-port Solution:

Scenario 1 : Original setup without SocketCAN

If the original setup does not use standard SocketCAN, then the applications would have to be reprogrammed as they are not compatible with the SocketCAN interface.

Scenario 2 : Original setup with SocketCAN

The Dual-port Solution is compatible with any application that is already using SocketCAN. This means that adding it is a simple plug-and-play installation.

Conclusion

The advantages of SocketCAN for Linux based CANbus applications are obvious. A character driver can only run a single process at a time and there are also compatibility issues with as they all hold different interfaces towards the application layer.

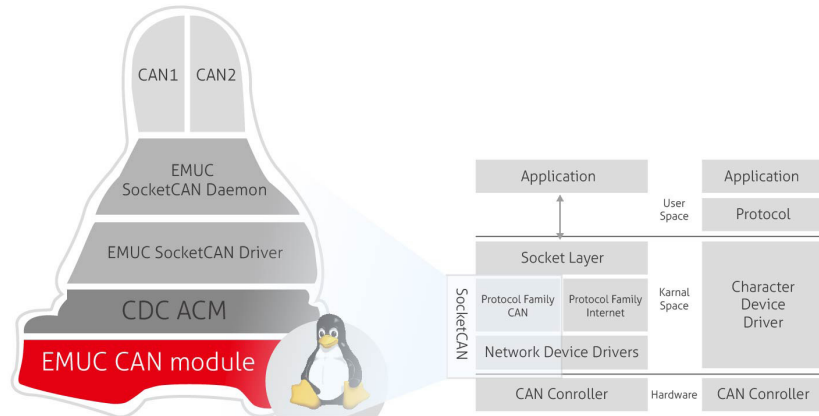
However, there are still limitations to SocketCAN, as it only supports one CANbus port. With two separate ports, the Dual-port Solution allows for greater flexibility, redundancy and customization for Linux-based CAN devices.

The Innodisk Solution

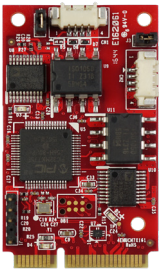
Wide Compatibility

The Dual-port Solution is compatible with ARM architecture and the following Linux operating systems:

- Ubuntu
- Fedora
- Debian
- CentOS
- QNX



EMUC-B202



Rugged Hardware Design

- Complies with EN61000-4-5 2.5kV Surge protection
- Complies with IEC 60950-1:2005 + A1: 2009 + A2:2013 2.5kV HiPOT protection
- Complies with EN61000-4-2 (ESD) Air-15kV, Contact-8kV
- Supports use in temperatures from -40°C to 85°C

Innodisk Corporation

5F., NO. 237, Sec. 1, Datong Rd., Xizhi Dist., New Taipei City, 221, Taiwan

Tel : +886-2-7703-3000

Fax : +886-2-7703-3555

E-Mail : sales@innodisk.com

innodisk

Copyright © July 2017 Innodisk Corporation. All rights reserved. Innodisk is a trademark of Innodisk Corporation, registered in the United States and other countries. Other brand names mentioned herein are for identification purposes only and may be the trademarks of their respective owner(s).