Introduction

Libre Car Control is an opensource car control effort to enable control of modern cars facilitate the development of non-proprietary vehicle technology. It provides an interface with modern vehicle's control systems.

Developers can formulate control commands like those which determine steering wheel torque sensor, throttle position sensor, and brake position sensor. This low-level interface means that it offers full-range control to develop/hack ADAS features.

Overview

Libre Car Control allows developers to:

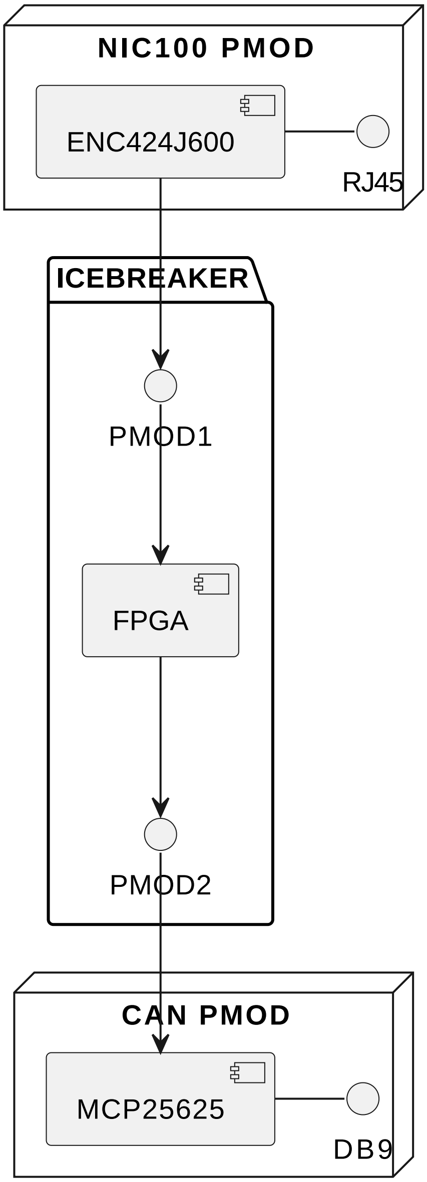
- Implement a simple service over Ethernet
- Convert this service to signal in Hardware
- Use this signal in control bus of vehicle CAN network

Product Objectives

- Read control messages from the Vehicle’s CAN Bus via Ethernet network
- Send control commands to the Vehicle’s CAN Bus via Ethernet network

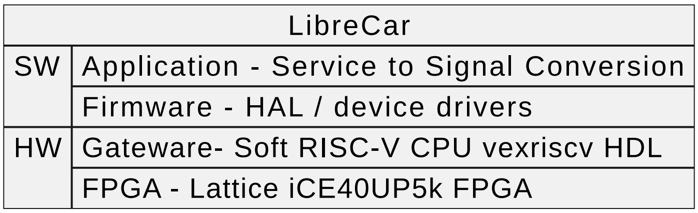
Control commands can be published to control bus in the form of Ethernet packet with pub/sub model from a node executing a path planning algorithm.

System Architecture



Components

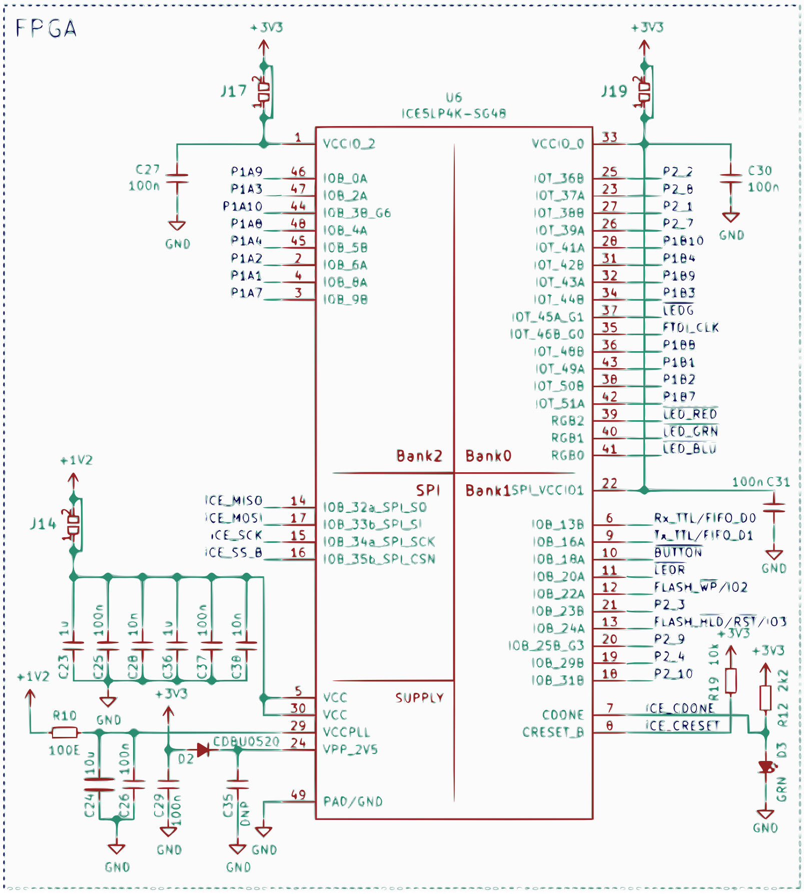
The main components of system architecture are broken down in Software and Hardware.



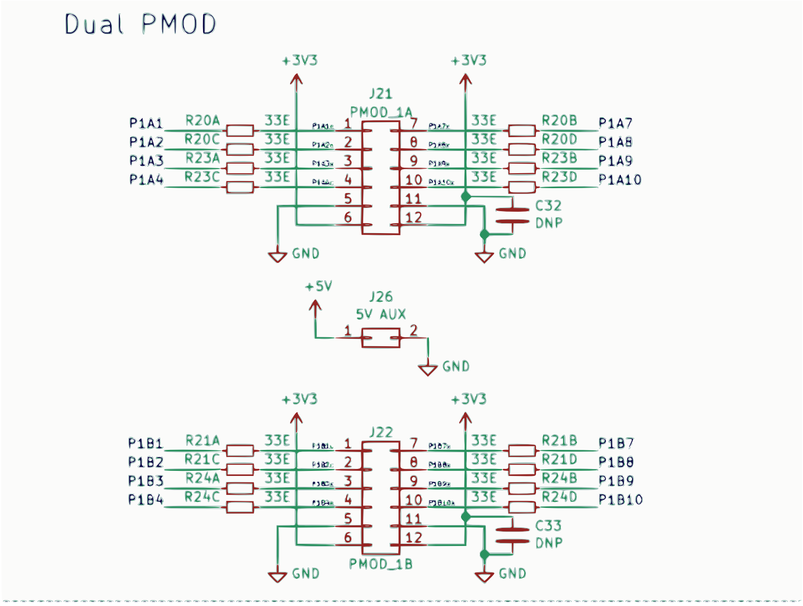
LibreCar	
SW	Application - Service to Signal Conversion
	Firmware - HAL / device drivers
HW	Gateware- Soft RISC-V CPU vexriscv HDL
	FPGA - Lattice iCE40UP5k FPGA

Hardware

iCE40 FPGA chip has severfal Pmod interface (peripheral module interface) for connecting peripheral modules.

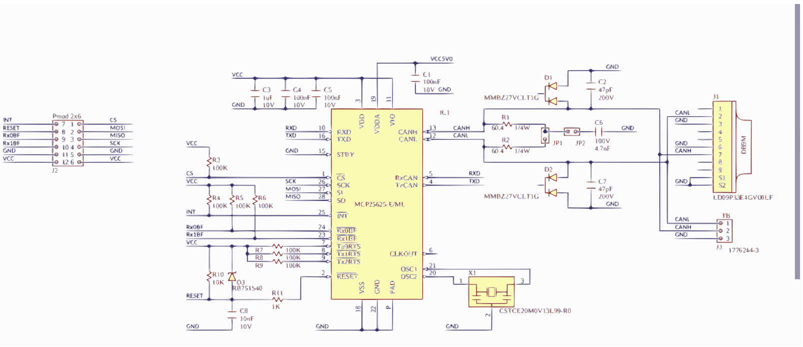


Pmod CAN and Pmod NIC100 are directly connected to FPGA via this interface :



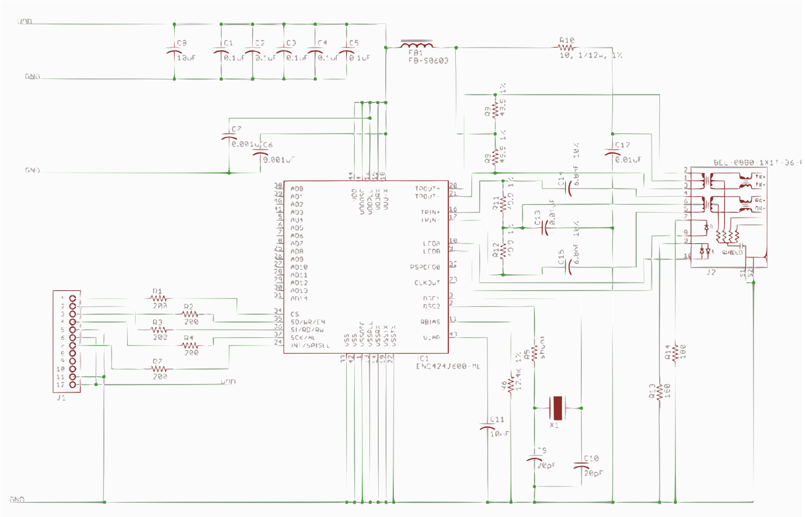
CAN

Pmod CAN is a CAN 2.0B controller with an integrated transceiver. The embedded Microchip MCP25625 chip connects directly to the physical CAN Bus.



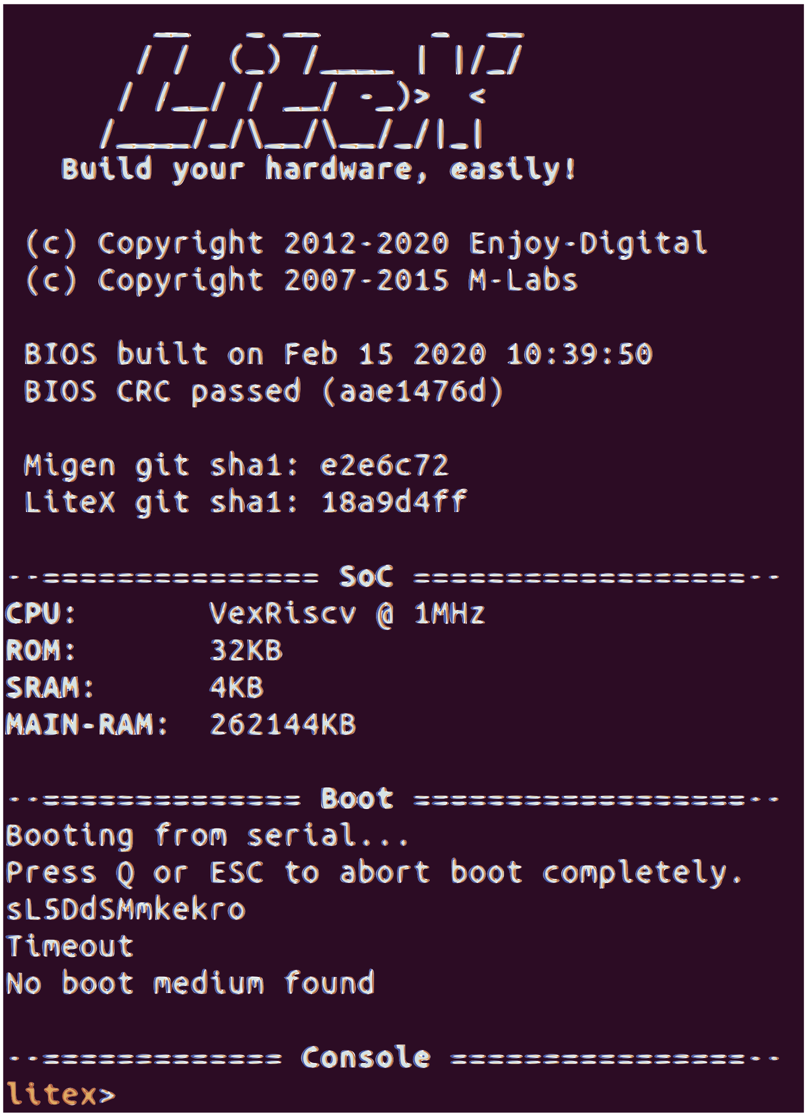
Ethernet

The PmodNIC100 utilizes Microchip's ENC424J600 to provide both MAC and PHY support to enable Ethernet functionality at data rates up to 10 Mbit/s.



Gateware

We use Litex framework to program VexRiscv-SMP CPU Risc-V SOC on the iCEBreaker FPGA that can be programmed from C language.



```
        __   _ __      _  __
       / /  (_) /____ | |/_/
      / /__/ / __/ -_)>  <
     /____/_/\__/\__/_/|_|
   Build your hardware, easily!

 (c) Copyright 2012-2020 Enjoy-Digital
 (c) Copyright 2007-2015 M-Labs

 BIOS built on Feb 15 2020 10:39:50
 BIOS CRC passed (aae1476d)

 Migen git sha1: e2e6c72
 LiteX git sha1: 18a9d4ff

--================ SoC ==================--
CPU:       VexRiscv @ 1MHz
ROM:       32KB
SRAM:      4KB
MAIN-RAM:  262144KB

--=============== Boot ==================--
Booting from serial...
Press Q or ESC to abort boot completely.
sL5DdSMmkekro
Timeout
No boot medium found

--============= Console ================--
litex>
```

Firmware

C language based device driver is used to program SPI interface that connects to CAN bus via MCP25625 and Ehternet via ENC424J600.

Application

Service to signal conversion layer is written to trasmit Ethernet based SOMEIP protototocol to CAN based signals.

Core APIs

- HelloWorldPublisher - Publish a message from a node to the Vehicle bus.
- HelloWorldSubscriber - Subscribe a node to messages on the Vehicle bus.
- Publish & Subscribe - Use publish/subscribe routines.
- Sample Application - Create an application node.

Device APIs

- Ethernet Reader - Read data from a network device.
- Ethernet Writer - Write data to a network device.
- CAN Reader - Read data from a CAN device.
- CAN Writer - Write data to a CAN device.