BRADLEY University

KOMATSU®

CAN Communication Development & Testing Platform

Christian M. West, Zachary R. Oakes Advised by Dr. Aleksander Malinowski

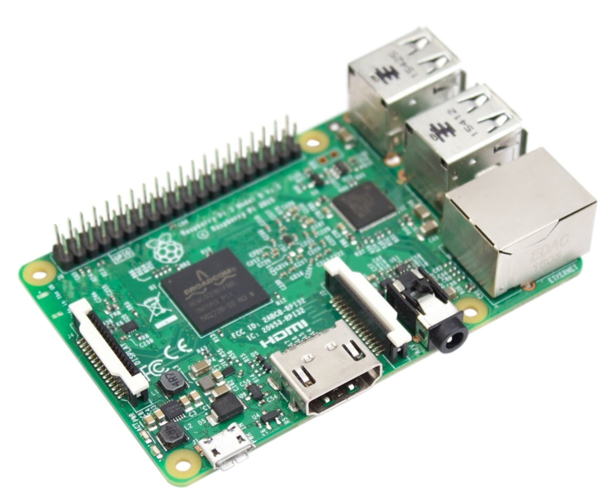
Department of Electrical & Computer Engineering

Abstract

Controller Area Network (CAN) communication is standard throughout the automotive and construction equipment industries due to its low-cost, durability, and structure that enables prioritized real time communication. Test benches in these industries commonly use CAN bus technology to detect faults in existing products and prototypes that are still under development. This project aims to create a low-cost and easily-repurposable testing and development platform for academic projects relating to CAN communication. It consists of six network endpoints implemented using single-board computers with CAN shields, along with an Ethernet switch. This topology also allows for direct access to each device, enabling a developer to easily control them over Ethernet. The result of this project yields a viable low cost testing and development bench for CAN network communication made from common off-the-shelf components.

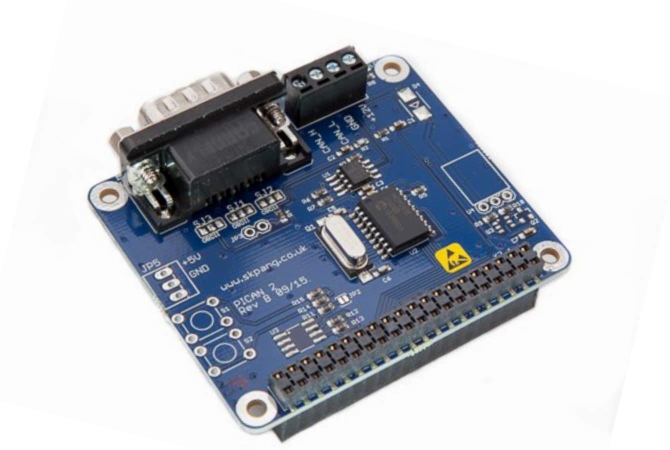
Hardware

The Raspberry Pi 3b is a single-board computer system. There are a total of 6 boards, each paired with a PiCAN2 that serve as an ECU (Electronic Computer Unit). The Raspberry Pi is widely available and is an efficient component in terms of cost for processing power.



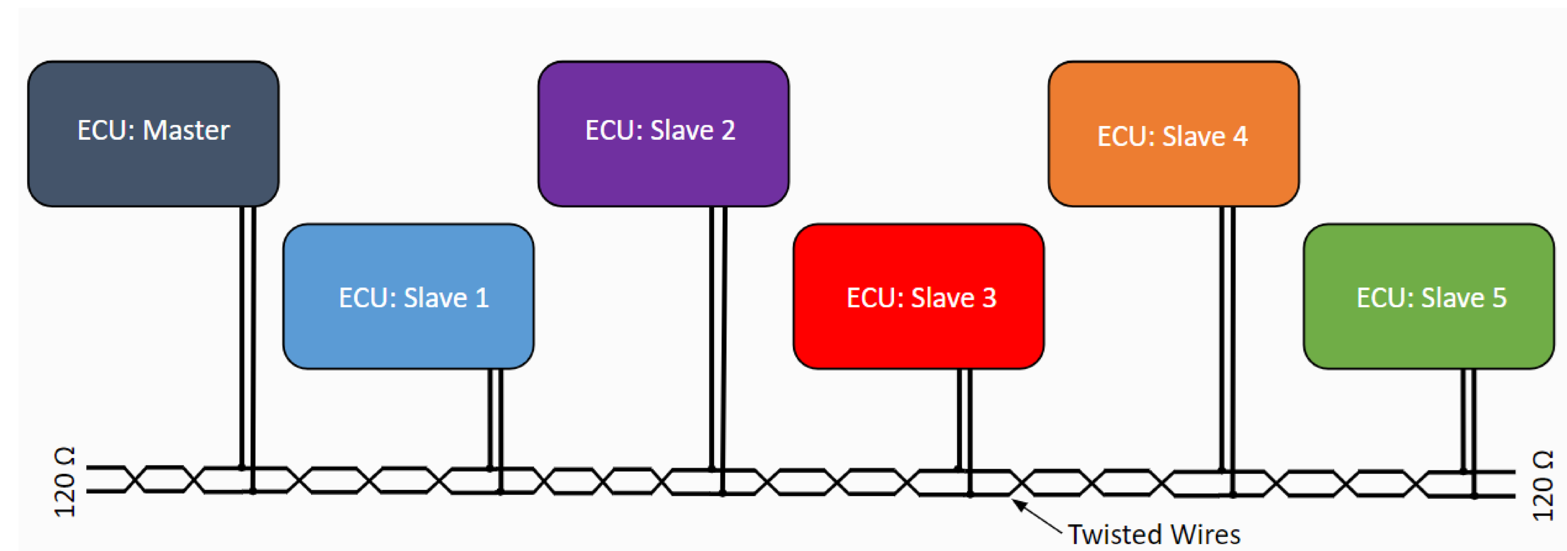
Raspberry Pi 3b - Single-Board Computer

There are a total of six PiCAN 2 boards to pair with each Raspberry Pi. Developed by Copperhill technologies, the PiCAN 2 provides Controller Area Network (CAN) Bus capabilities for the Raspberry Pi. It utilizes an onboard Microchip MCP2515 CAN controller with MCP25151 CAN transceiver. Connections are made via DB9 or 3—way screw terminal.

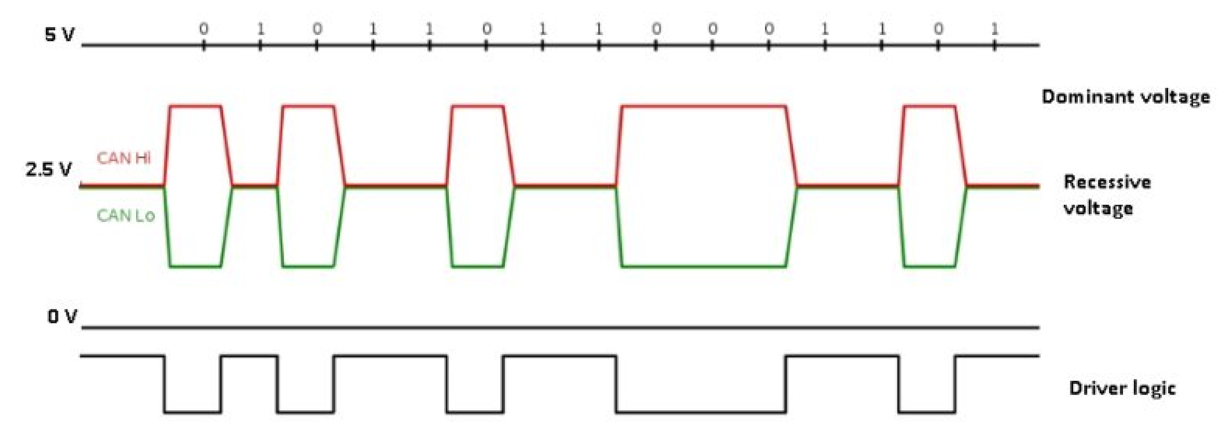


PiCAN 2 - CAN Interface for Raspberry Pi 2/3

CAN Communication

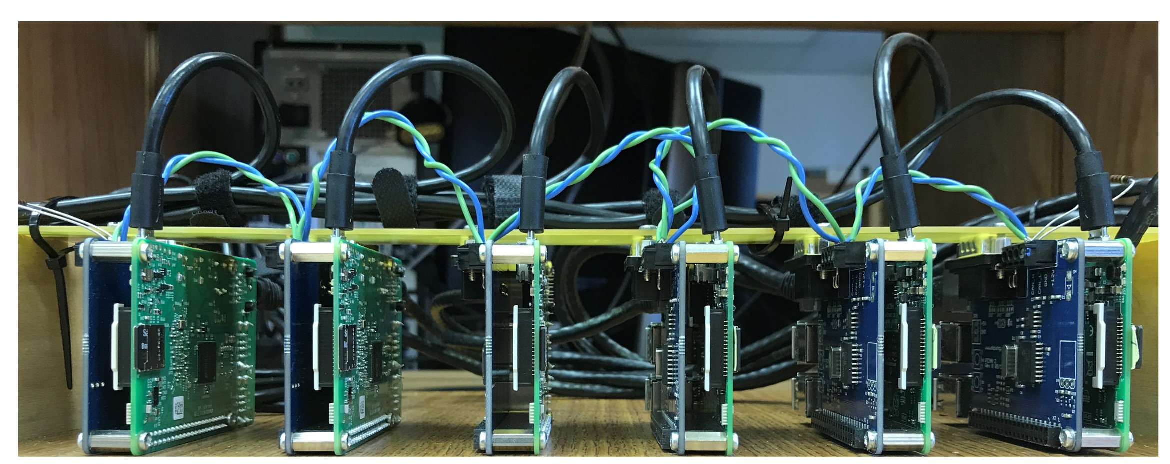


The structure of this testbench follows that of a Master-Slave topology as seen above. Each ECU is simulated using a Raspberry Pi equipped with the PiCAN2; one serves as the Master ECU and the rest as slave ECUs. The CAN backbone connects to each module in parallel with each end terminated by 120 Ohm resistors.



High-speed CAN signaling per ISO11898-2

From an idle voltage of 2.5 V, CAN_H and CAN_L pins are driven apart for logical 0 (dominant state) and back together for logical 1 (recessive state). If the bus is recessive, any device on the bus can assert dominant. This behavior is how messages are acknowledged: the sending unit writes recessive for one cycle and the receiving unit sends back dominant. If the bus still reads recessive at this time, then the transmitting unit thinks that transmit failed since it hasn't been acknowledged.



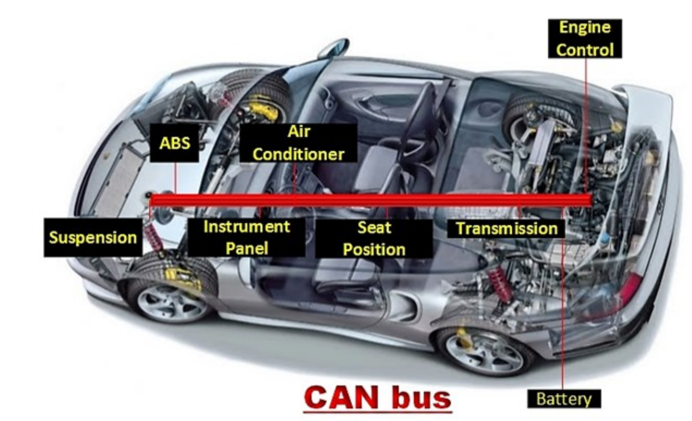
Simulated ECU Network Testbench

The figure above illustrates the components used to simulate an ECU network. Each ECU consists of a Raspberry Pi equipped with PiCAN 2. CAN backbone is created using a twisted pair of wires where CAN_H is blue and CAN_L is green. Termination resistance at each end using 120 Ω (±10%) resistors.

Industry Use

Automotive and Transport Industry

- Automobile Companies
  - Mercedes-Benz
- Construction/Mining Equipment Manufacturers
  - Komatsu/John Deere/Caterpillar
- Aerospace
  - SOFIA Boeing 747SP



Medical Equipment– Vital operating room components

- OR lights and tables
- Endoscope lights and cameras
- Insufflators
- X-ray machines
- Ultrasound machines
- Video recorders/printers

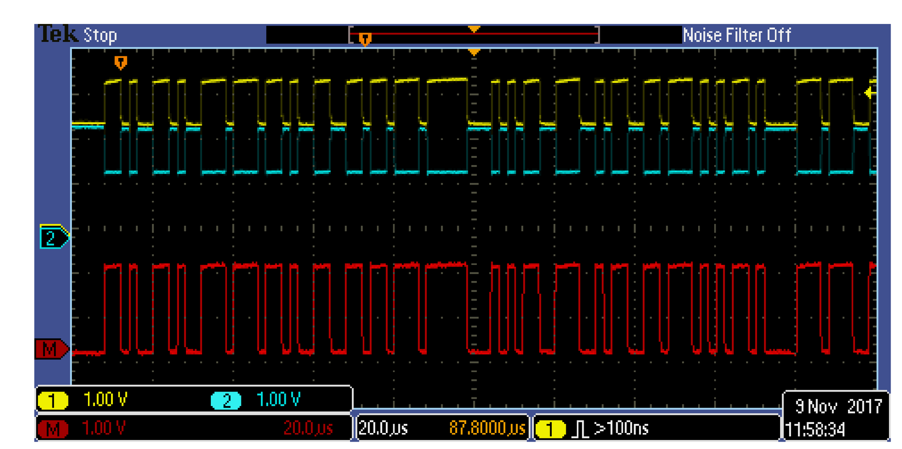
Conclusions

Accomplished:

- Fully realized CAN Communication testbench
  - Built the physical layer of a CAN Communication testbench
  - Established communication between Master and Slave modules
- Software in place for each module to send and extract data on the network

Future work:

- Implement a method for data encryption
- Simulate a "lossy" network to validate information redundancy



Transmitted CAN Communication

Using an oscilloscope, this data capture shows a packet sent from the Master to the rest of the network. The data contained within is: 123#456789abcdef0123.

References

1. http://copperhilltech.com/pican-2-can-interface-for-raspberry-pi-2-3/
2. https://en.wikipedia.org/wiki/Cisco_Systems
3. https://i1.wp.com/gadgetynews.com/wp-content/uploads/2018/03/raspberry-pi-3b--e1521128910887.png
4. http://resources.infosecinstitute.com/future-now-car-hacking/
5. https://www.testandmeasurementtips.com/exploring-canbus-oscilloscope/