ECE 497 - Senior Project Deliverables

Komatsu Sponsored - ECU Communication and Networking

Zach Oakes
Christian West

Advised by:
Aleksander Malinowski

November 9th, 2017

Table of Contents

Abstract
It would be advantageous for automotive companies producing heavy machine equipment to implement a robust ECU network that is “smart” enough to recognize what modules are connected to it along with their names, serial numbers, and what truck frame the network is on. Additionally, having the ability to handle memory retainment for the swapping of ECU’s, and detecting the addition of unauthorized modules. This is very appealing as this type of network can serve as a platform to monitor where and when a machine is and to collect data such as diagnostics and prognostics to potentially be sent to a central database for monitoring. It will also increase security ensuring that paid features will only be enabled on the intended machine. This project should deliver a prototype of a simulated ECM network that encompasses these features along with proposing alternative of different methods in order for these features to exist.

Introduction

Komatsu has asked us to develop a system that automatically keeps track of which truck an Electronic Control Unit (ECU) is installed on. Currently each ECU must be programmed with the identifier of all of the other ECUs on the network. Under the new system, each unit on the network would automatically acquire the identifiers of all the other devices on the network. Additionally if any device on the network knows the Frame Number (a unique identifier for each truck chassis, like a VIN for on-road vehicles) then it would broadcast this information over the network so all units would have a copy of this information. Further goals are to provide secure transmission of this information and – not strictly related to CAN communication – to have the frame number automatically detected, removing the possibility of accidently programming the units incorrectly.

Review of Literature and Prior work

Some of the stuff from online about how CAN Traffic works in general, stuff from the docs about how piCAN works in specific. Other companies probably have some prior work in a similar direction, but since it’s proprietary we won’t be able to find it. CAN High-Speed is defined by ISO11898-2 as a differential data bus, acting as can be seen in the following image:

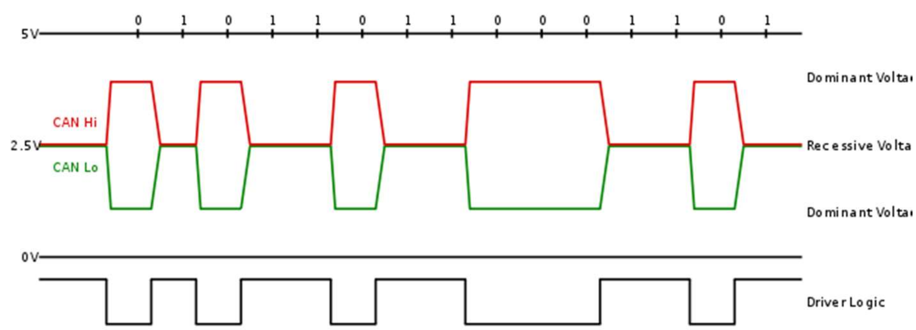

Figure 1: CAN Traffic

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 write 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.

Applicable Standards and Patents

The Physical and Datalink Layers (OSI Layers 1&2) are described by ISO11898-2. SAE J1939 is the most commonly used protocol for vehicular CAN communication, describing OSI Layers 3-5 and is what most of the existing communication on the network will be using. Whatever we use will have to be built so that it doesn’t interfere with already existing J1939 traffic. This said, J1939 is not a good choice for transporting our information because it is a published standard and therefore any information sent over it will not be secure.

Subsystem Level Functional Requirements with Specifications

The communication network topology utilized by Komatsu machines follow that of master and slave. Figure 2 below illustrates

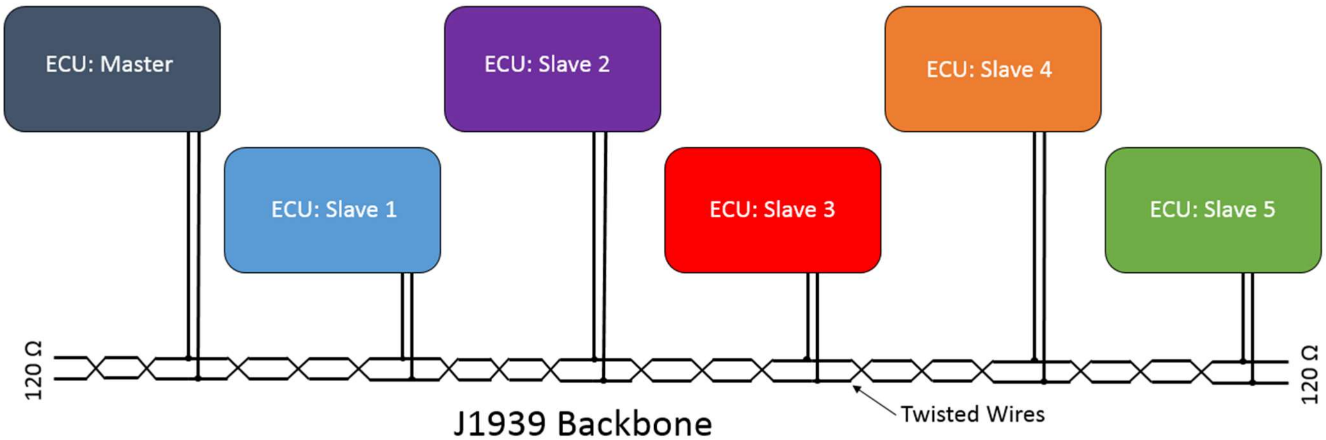

Figure 2: System Diagram of ECU Network

Each ECU will be simulated using raspberry pi's labeled where one will act as the Master ECU and the rest being the slave ECU's. The CAN network will connect through each ECU where the ends will be terminated by 120 Ohm resistors.

Engineering efforts completed to date

Two systems can send packets back and forth but don't yet respond automatically. Python Library seems to work. Next step is to get all the rest of the parts in and build up an application layer for handling data over the network. In terms of the OSI Model, we have Layers 1 - 3 working currently, and will need to build a secure way to send information between nodes (OSI layer 4).

Parts List

The hardware portion of our CAN network is made up for six (6) Raspberry Pi 3b units, equipped with the PiCAN2 CAN transceiver. The CAN modules are linked by twisted pairs of wires, as is common in automotive networks. At the moment we only have two of the PiCAN2 units, but more will be coming now that we have proof that they work. All the other equipment is used to power and interface to the Pis.

Table 1: Parts List ECU Network

Quantity	Description	Price	Ext. Price
6	Raspberry Pi3 Model B	$35.10	$ 210.60
12	8GB micro SD cards	$7.11	$ 85.32
1	USB HUB for Data	$24.99	$ 24.99
1	USB HUB for Power	$23.99	$ 23.99
1	3 foot USB 2.0 (6 Pack)	$8.99	$ 8.99
6	USB to TTL Serial	$6.99	$ 41.94
1	Ethernet Switch	$23.28	$ 23.28
1	Ethernet Cables 2 Feet (10 pack)	$13.99	$ 13.99
6	CAN Interface for Raspberry Pi	$47.95	$ 287.70
		Total	**$ 720.80**

Deliverables for ECE 499 Schedule

ECE499 tentative timeline (Spring 2018)

	Tuesday	Thursday
Jan 16, 18		ECE499 project work
Jan 23, 25	ECE499 project work	ECE499 project work
Jan 30, Feb 1	ECE499 project work	ECE499 project work
Feb 6, 8	ECE499 project work	ECE499 project work
Feb 13, 15	ECE499 project work	ECE499 project work Due: midpoint project progress self-check
Feb 20, 22	ECE499 project work	ECE499 project work
Feb 27, Mar 1	ECE499 project work	ECE499 project work
Mar 6, 8	ECE499 project work	ECE499 project work (Student Expo registration Due: Mar 9th Friday)
Mar 13, 15	Spring break (Mar 10 - 18)	
Mar 20, 22	ECE499 project work	ECE499 project work
Mar 27, 29	ECE499 project work	ECE499 project work Draft due: final report (Student Expo abstract Due: Mar 30th Friday)
Apr 3, 5	ECE499 project work	ECE499 project work (Poster print Due: Apr 5th Thursday)
Apr 10, 12	Poster presentation practice (Expo Poster setup Due: Apr 10th)	ECE499 project work (partially) (Expo Poster judging: Apr 12th) (Award ceremony: Apr 13th)
Apr 17, 19	Project demonstration Draft due: final presentation	Project demonstration
Apr 24, 26	Poster and final presentation practice	Poster and final presentation practice Class picture (Apr 26th Thursday) (weather permits, in front of Bradley Hall at noon) IAB poster session (Apr 27th Friday)
May 1, 3	Senior Project Conference and Reception? (April 28, Saturday) All deliverables (final presentation, final report, recorded final demo videos) completed and uploaded to the project website (May 1, Tuesday)	
May 8, 10		
Graduation reception May 11 Friday Graduation ceremony May 12 Saturday		

Figure 5: 499 Tentative Schedule

References

http://copperhilltech.com/pican2-controller-area-network-can-interface-for-raspberry-pi/
http://sgframework.readthedocs.io/en/latest/cantutorial.html
http://copperhilltech.com/content/CIA_article.pdf