The CAN Data Diode

Hayden Allen, hayden-allen@utulsa.edu
Professor Jeremy Daily, jeremy-daily@utulsa.edu
Problem Statement

- Additional devices, like Electronic Logging Devices (ELDs), are installed on the J1939 control network of trucks.
  - Provides Internet connections to heavy vehicles
  - May not be secure
  - Trucks use J1939 for safety critical systems (i.e. brakes).
- Need a method to protect heavy vehicles while complying with the Federal Mandate.
Proposed Solution

- Use two transceivers back to back without connecting the received ELD connection to the transmitted J1939 connection.
- Physically isolates the J1939 network from the ELD
- No software control
  - Not hackable
Supporting Functionality

- Some elements are not broadcast without requests
- Need independent logic to ask for required ELD parameters like VIN, hours, etc.
Error Frame Detection

- Provide functionality to turn off the diode if the ELD sees CAN frame errors.
- Enable Auto baud detection
Prototype Design

- Circular section
  - Power
  - Transceivers (as the Diode)
  - Requester
  - ELD Responder
- Tabs
  - Programming
  - UART Connection
  - LED Feedback
Prototype Hardware

- Produced by EMS Cable in Lincoln, AR.
Functional Testing

- Test 1: The Smoke Test.
  - When the new device is plugged in, does the power circuit work?
Functional Testing (cont.)

- Test 2: Clock Generation
  - Does the Oscillator Circuit produce 16MHz?
- Check the top of C9 or U3: pin 9
Test 3: Do the other processors have 16Mhz?

Test the following pins:
- U6:5 (ATTiny)
- U7:5 (ATTiny)
- U5:9 (MCP25265)

We have 2MHz present
- Need to change prescaler in the CAN controller (U3) register.
Programming the ATTiny

- To affect frequency, we need to send commands.
- To send commands, we need to program the ATTiny.
- To program the ATTiny, we need a programmer.

---

**ATTiny Programming Header**

8.2 CLKOUT Pin

The CLKOUT pin is provided to the system designer for use as the main system clock or as a clock input for other devices in the system. The CLKOUT has an internal prescaler which can divide $F_{Osc}$ by 1, 2, 4 and 8. The CLKOUT function is enabled and the prescaler is selected via the CANCNTRL register (see Register 10-1).

*Note:* The maximum frequency on CLKOUT is specified as 25 MHz (See Table 13-5).

The CLKOUT pin will be active upon system reset and default to the slowest speed (divide by 8) so that it can be used as the MCU clock.

When Sleep mode is requested, the MCP2515 will drive sixteen additional clock cycles on the CLKOUT pin before entering Sleep mode. The idle state of the CLKOUT pin in Sleep mode is low. When the CLKOUT function is disabled (CANCNTRL.CLKEN = ‘0’), the CLKOUT pin is in a high-impedance state.

The CLKOUT function is designed to ensure that $t_{\text{CLKOUT}}$ and $t_{\text{CLKOUT}}$ timings are preserved when the CLKOUT pin function is enabled, disabled or the prescaler value is changed.
Making an In-System Programmer (ISP)

Software to make an ISP

An ISP with an Arduino Pro Mini
Uploading a Program

- Goal: write some basic code to ensure the programming process works.
- Result: Blinking LEDs!
- Next Steps:
  - Write a small SPI transfer function
  - Send the MCP CAN Controller commands to change the clockout prescaler.
Arduino Bit-Bang SPI

- Bit-bang with Arduino digitalWrite and digitalRead: 13.26 ms for four bytes.
- Bitbank with direct port manipulation: 0.855 ms for four bytes.

After power up, there is a RESET command (1 byte) followed by setting the Clock Pre-scaler Register (4 bytes).
Clock out at 16MHz

Tested CLOCK circuit

Test #3 Passed!

4/6/2018
Test 4: Data Diode Function

- Does CAN Data from J1939 show up on the ELD side?
- Does the device prevent CAN messages on the ELD side from being transmitted to the J1939 bus?
- Use BeagleBone Black with TruckCape for tests (2 channel)
  - Transmit messages using Linux SocketCAN “cangen”
  - Look for messages using “candump”

Use the J1939 CAN2 (Pins H and J on the 9-pin connector) as the ELD.
J1939 Connector

9-Pin Deutsch – Freightliner Cascadia (H,J Used for Dual CAN)

<table>
<thead>
<tr>
<th>Pin</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>Ground</td>
</tr>
<tr>
<td>B</td>
<td>+12V</td>
</tr>
<tr>
<td>C</td>
<td>CAN/J1939 Hi</td>
</tr>
<tr>
<td>D</td>
<td>CAN/J1939 Lo</td>
</tr>
<tr>
<td>E</td>
<td>CAN/J1939 Shield</td>
</tr>
<tr>
<td>F</td>
<td>J1708/J1587 Hi</td>
</tr>
<tr>
<td>G</td>
<td>J1708/J1587 Lo</td>
</tr>
<tr>
<td>H</td>
<td>CAN 2 Hi</td>
</tr>
<tr>
<td>J</td>
<td>CAN 2 Lo</td>
</tr>
</tbody>
</table>

Every Message on J1939 (CAN1) is passed over to the ELD (CAN0).
No messages on the ELD (CAN0) are passed to J1939 (CAN1)
Diode Functionality Works!
Switch ELD TX mode On/Off

- Turn on and off the ELD Transceiver with the manual switch.
- Program the ATTiny to set the SILENT pin to the Switch Reading.
Test #5: Use SILENT mode on ELD CAN Transceiver

Uses the same setup for test 4 with "cangen can1" producing messages on J1939. Those messages will always show up on can1 and only on can0 (ELD) when the switch is closed.
Next Steps

- Program the AVRs for their purpose
  - Autobaud
  - Requester
  - Error Detector (with hysteresis)
- Stress Test
  - High Voltage Input
  - High bitrates and bus load
- Check Analog Wave Form of CAN Signals

- Set proper bit timing
- Measure current draw
- Check for Acknowledgement bits on TXD pin of U3
- Program Halting of Requests on Command
- Document all the default register values