The CAN Data Diode

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

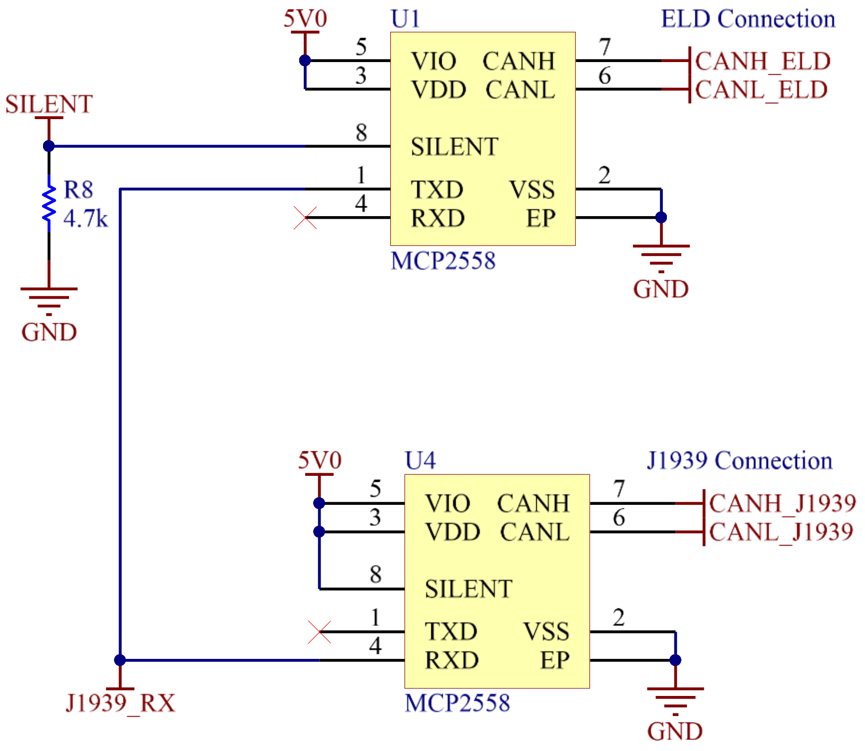
THE UNIVERSITY of TULSA
Student CyberTruck Experience

Problem Statement

- Additional devices, like Electronic Logging Devices (ELDs), are installed on the J1939 control network of trucks.
  - Provides Internet connections to heavy vehciles
  - 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



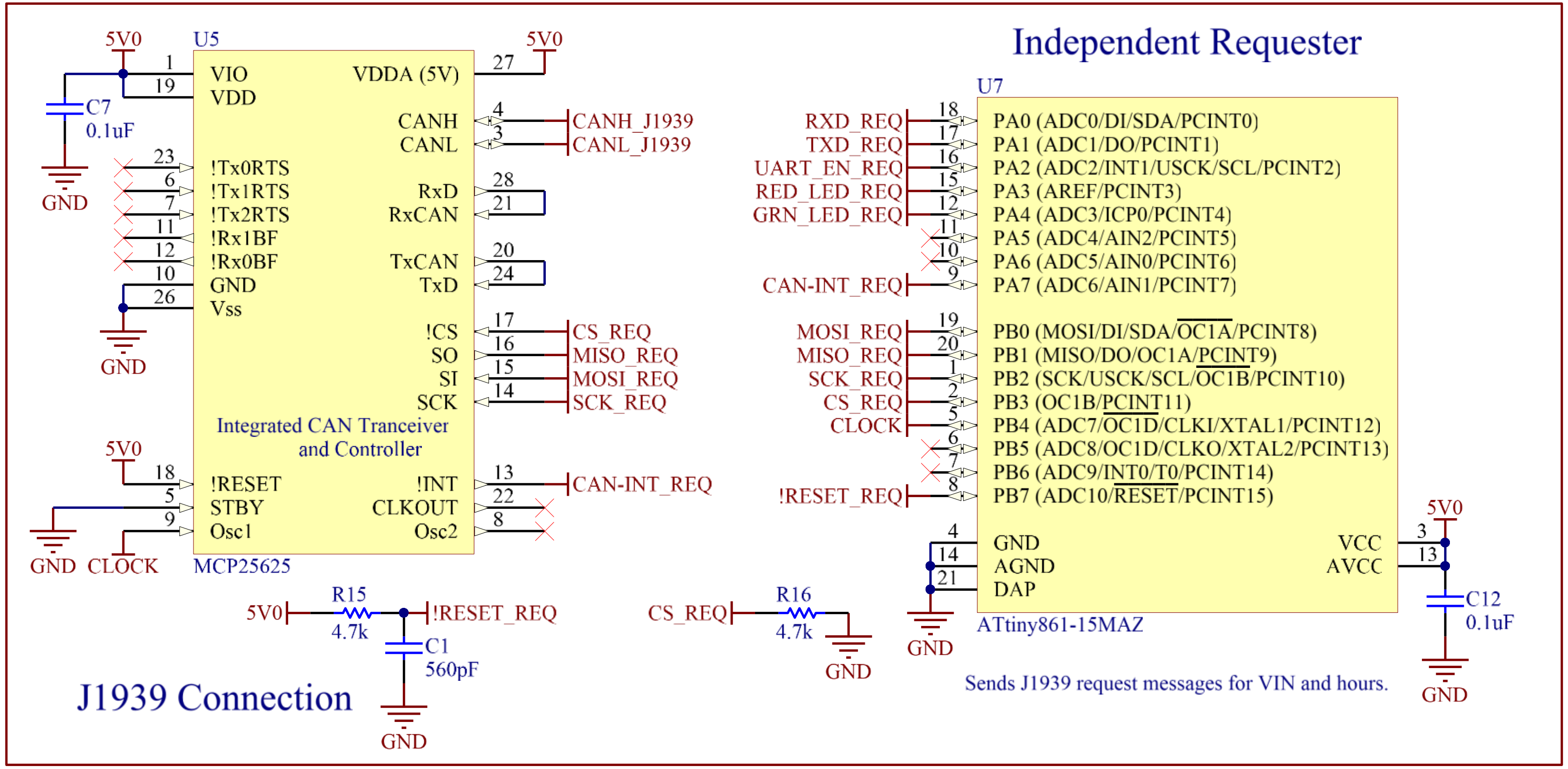
Pull SILENT High if too many CAN errors on ELD side.

CAN Transceivers for One Way Communication

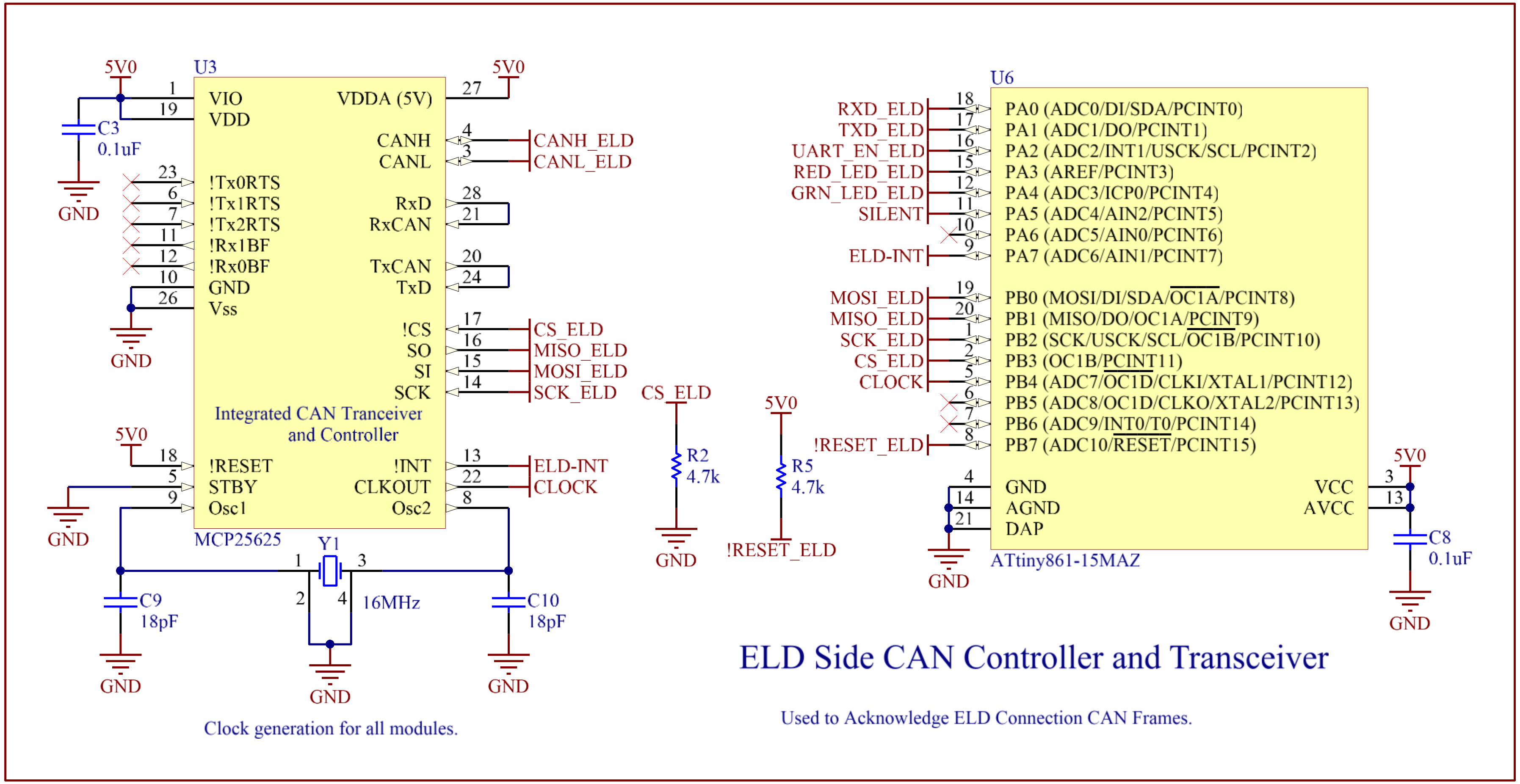
There is no physical path for the ELD Connection to write to the J1939 Connection.

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



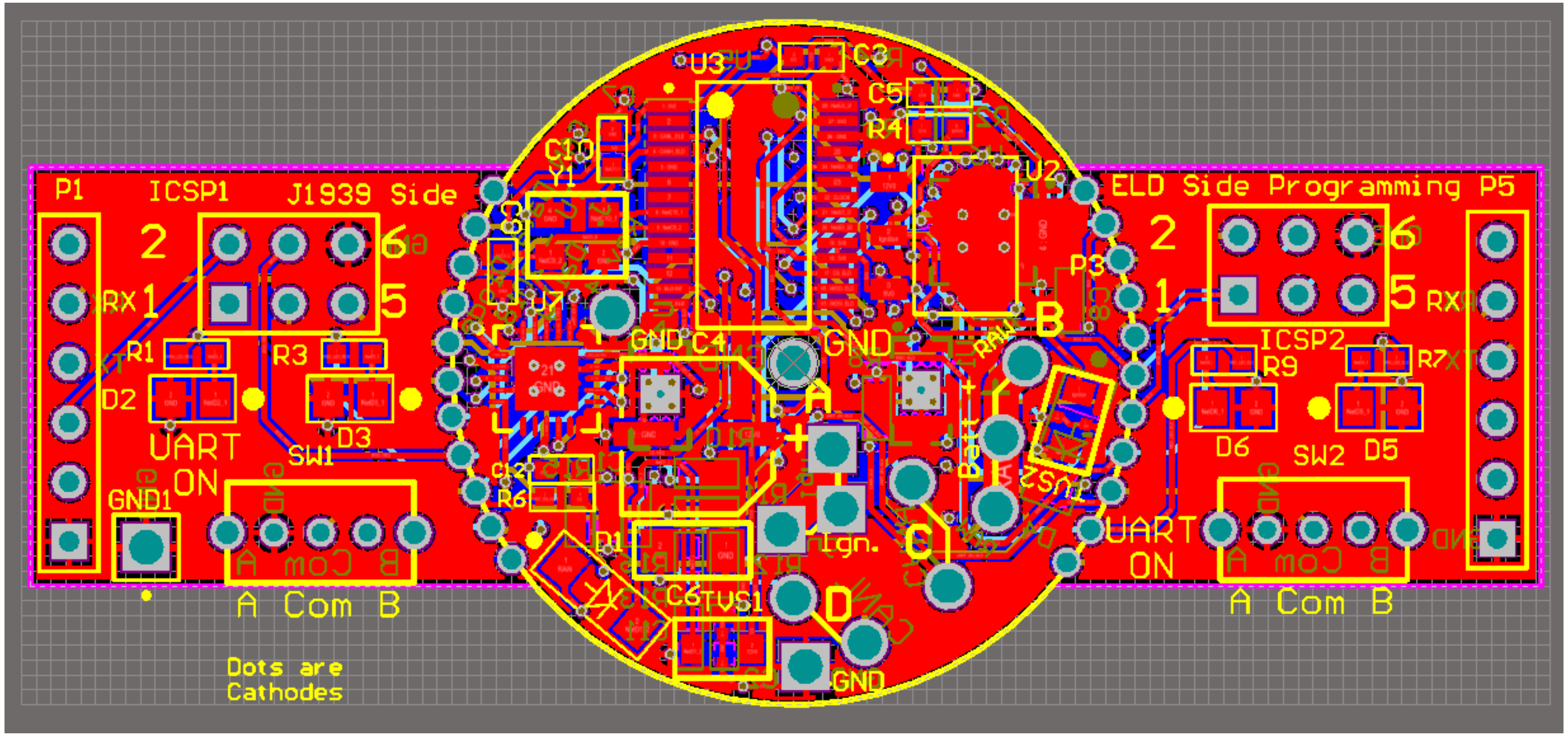
Clock generation for all modules.

ELD Side CAN Controller and Transceiver

Used to Acknowledge ELD Connection CAN Frames.

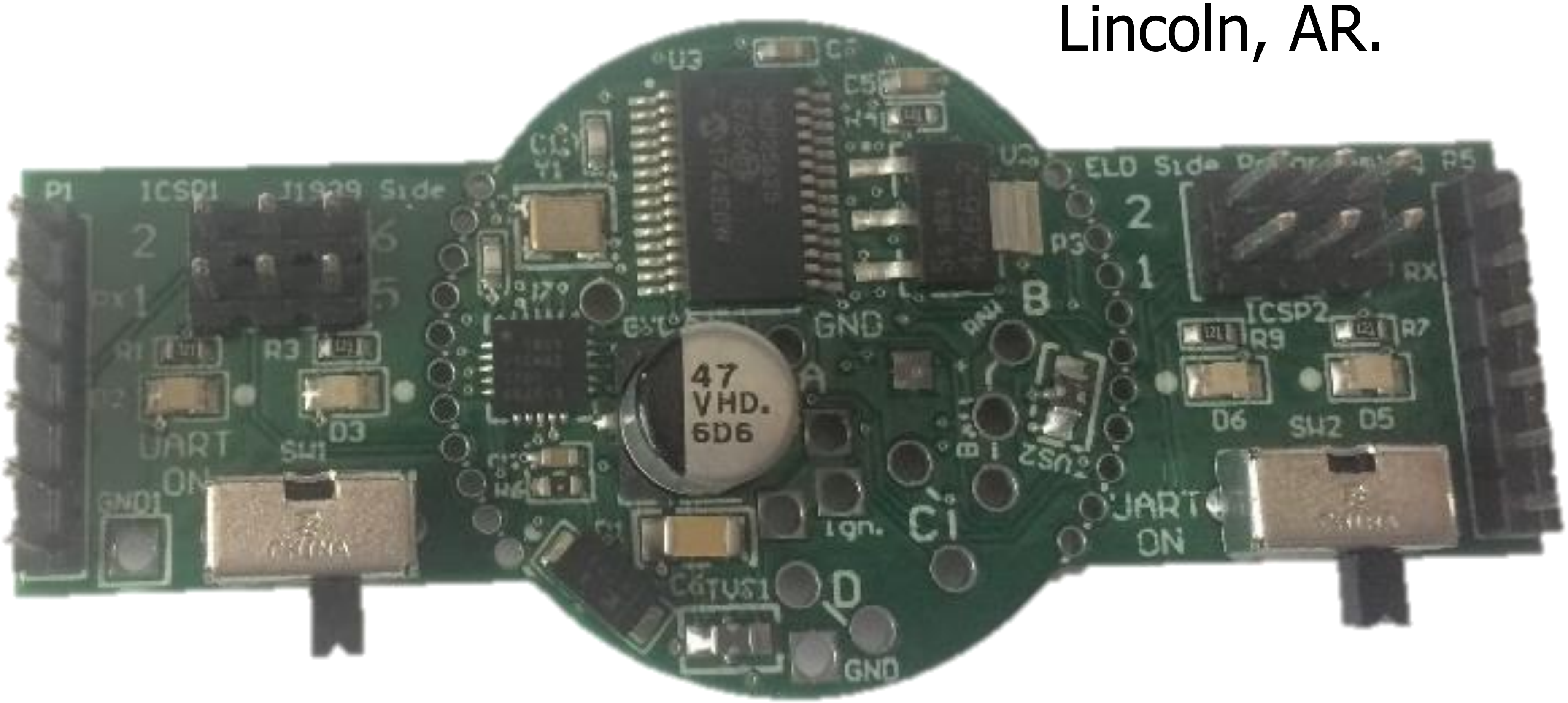
- Provide functionality to turn off the diode if the ELD sees CAN frame errors.
- Enable Auto baud detection

Prototype Design



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

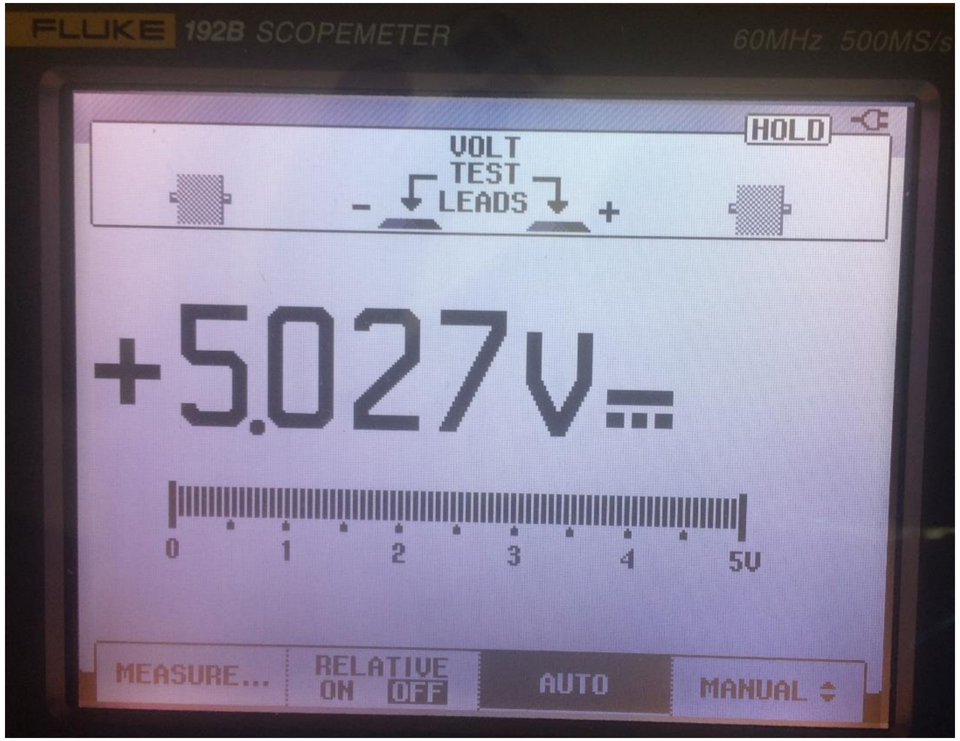
Prototype Hardware



- Produced by EMS Cable in Lincoln, AR.

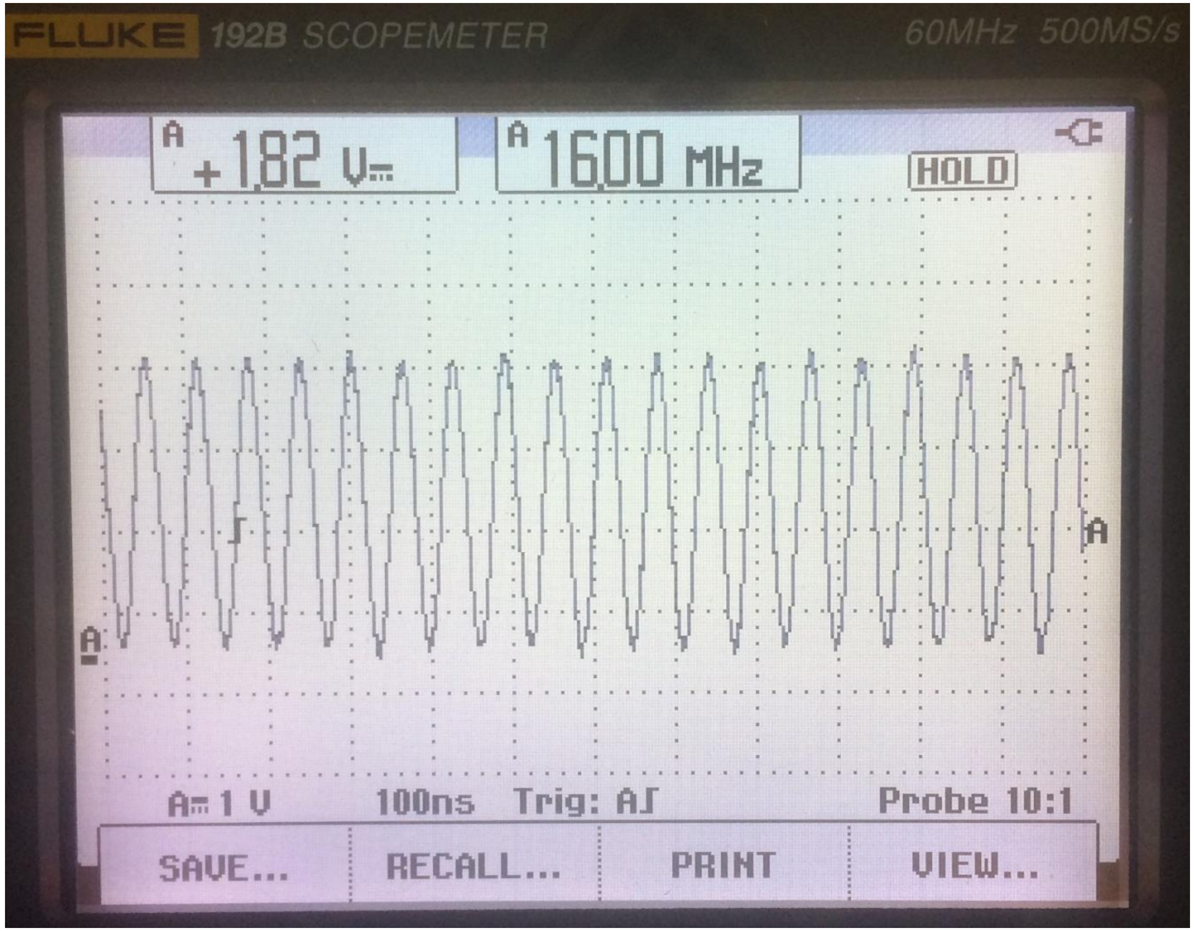
Functional Testing

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



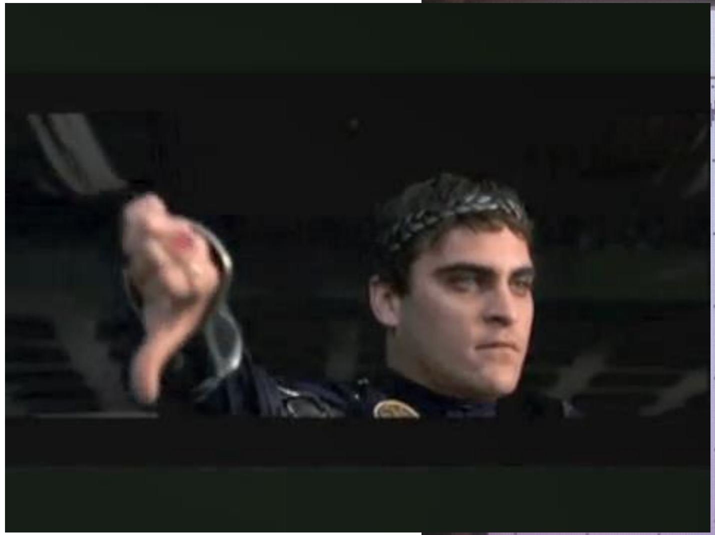
Functional Testing (cont.)

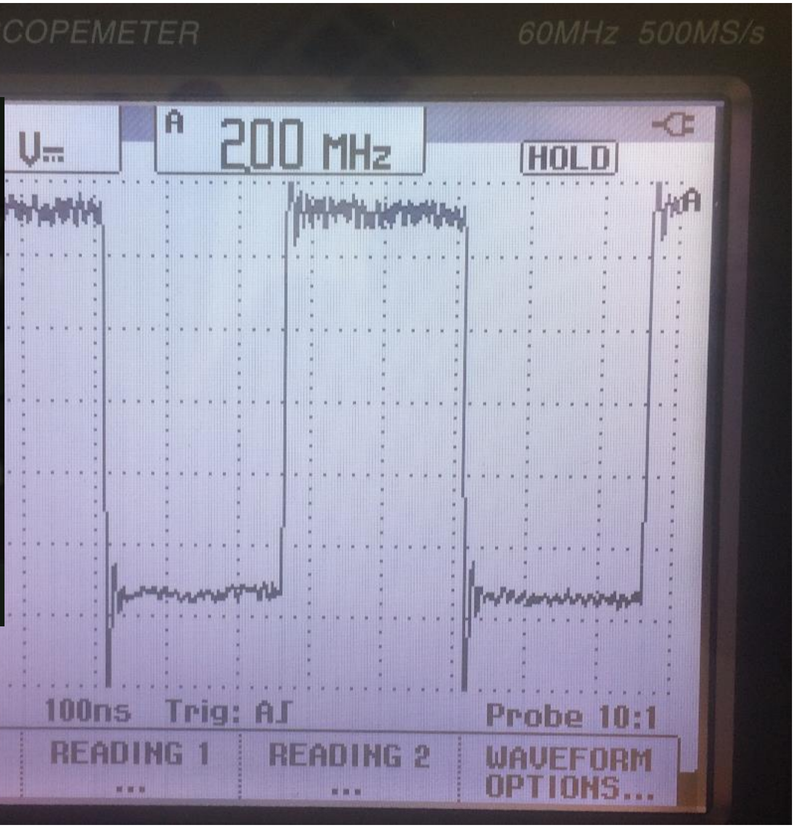
- Test 2: Clock Generation
  - Does the Oscillator Circuit produce 16MHz?
- Check the top of C9 or U3: pin 9



Functional Testing (cont.)

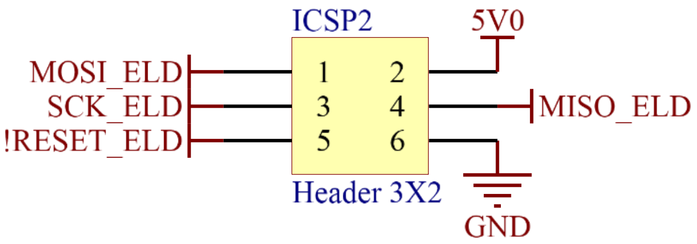
- Test 3: Do the other processors have 16Mhz?
- Test the following pins:
  - U6:5 (ATTiny)
  - U7:5 (ATTiny)
  - U5:9 (MCP25265)
- We have 2MHz present
  - Need to change prescaler in the CAN controller (U3) register.



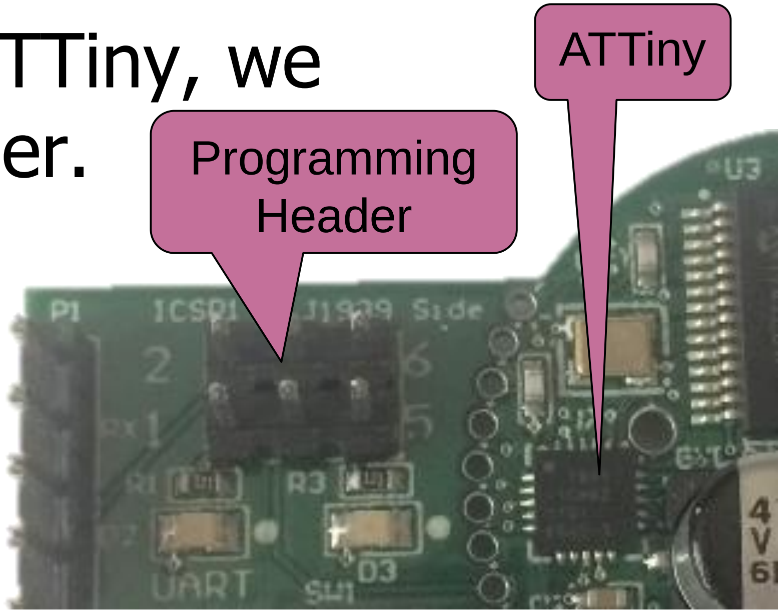


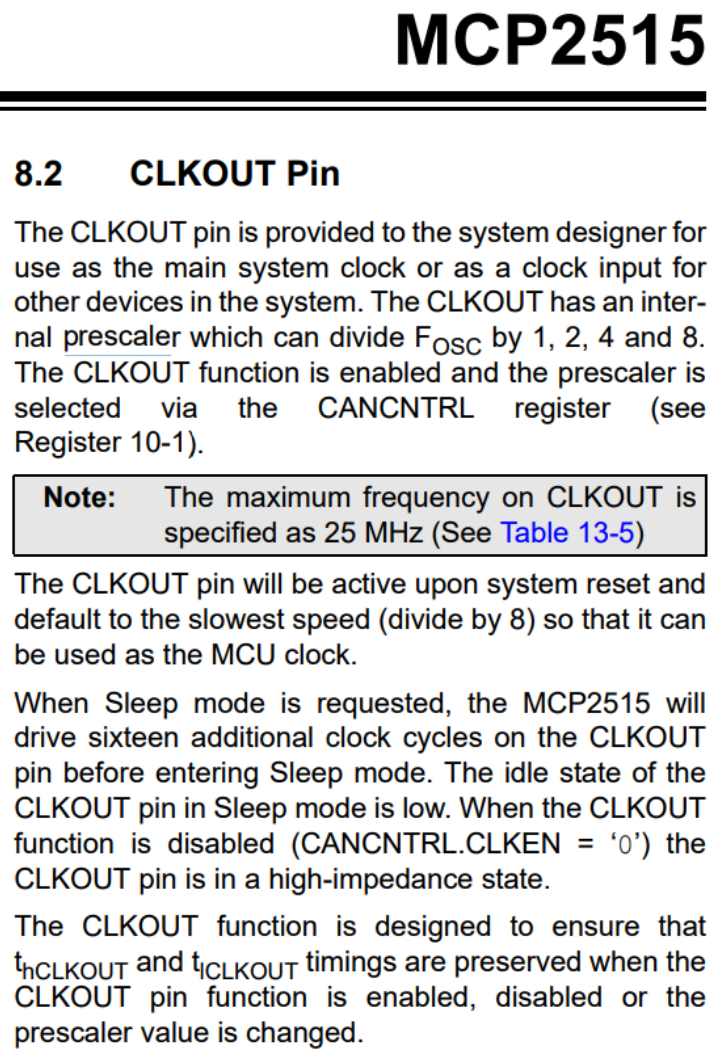
Programming the ATTiny

- 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.



Programming Header





MCP2515

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)

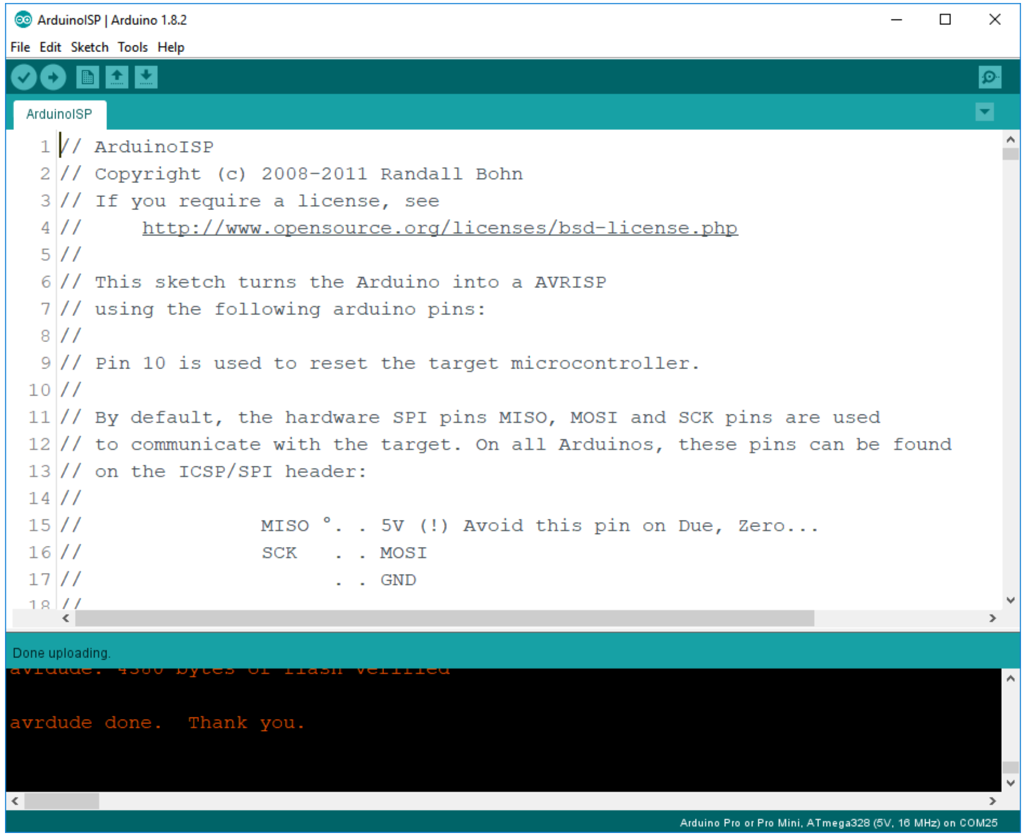
The CLKOUT pin will be active upon system reset and default to the slowest speed (divide by 8) so that it can be used as the MCU clock.

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 $t_{hCLKOUT}$ and $t_{lCLKOUT}$ 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



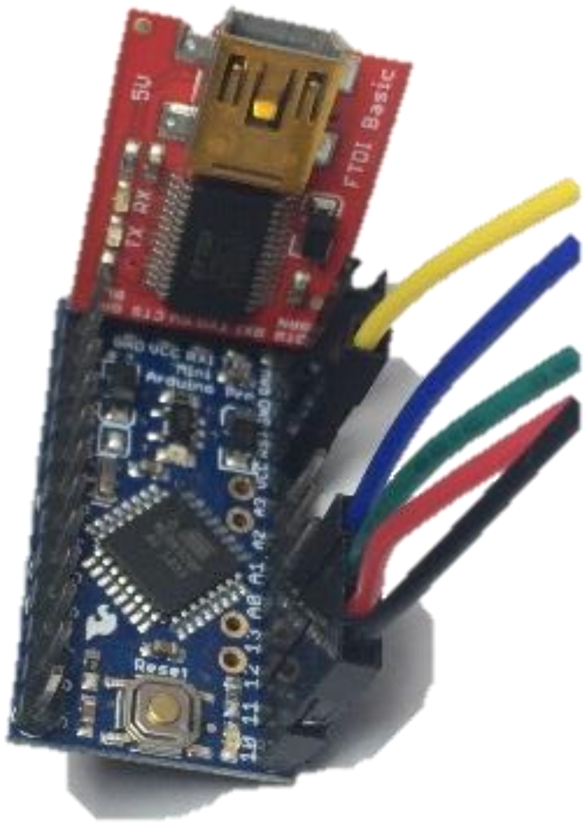
```
// ArduinoISP
// Copyright (c) 2008-2011 Randall Bohn
// If you require a license, see
//     http://www.opensource.org/licenses/bsd-license.php
//
// This sketch turns the Arduino into a AVRISP
// using the following arduino pins:
//
// Pin 10 is used to reset the target microcontroller.
//
// By default, the hardware SPI pins MISO, MOSI and SCK pins are used
// to communicate with the target. On all Arduinos, these pins can be found
// on the ICSP/SPI header:
//
//                MISO °. . 5V (!) Avoid this pin on Due, Zero...
//                SCK   . . MOSI
//                      . . GND
//
```

Done uploading.

```
avrdude: 4300 bytes of flash verified

avrdude done.  Thank you.
```

Arduino Pro or Pro Mini, ATmega328 (5V, 16 MHz) on COM25

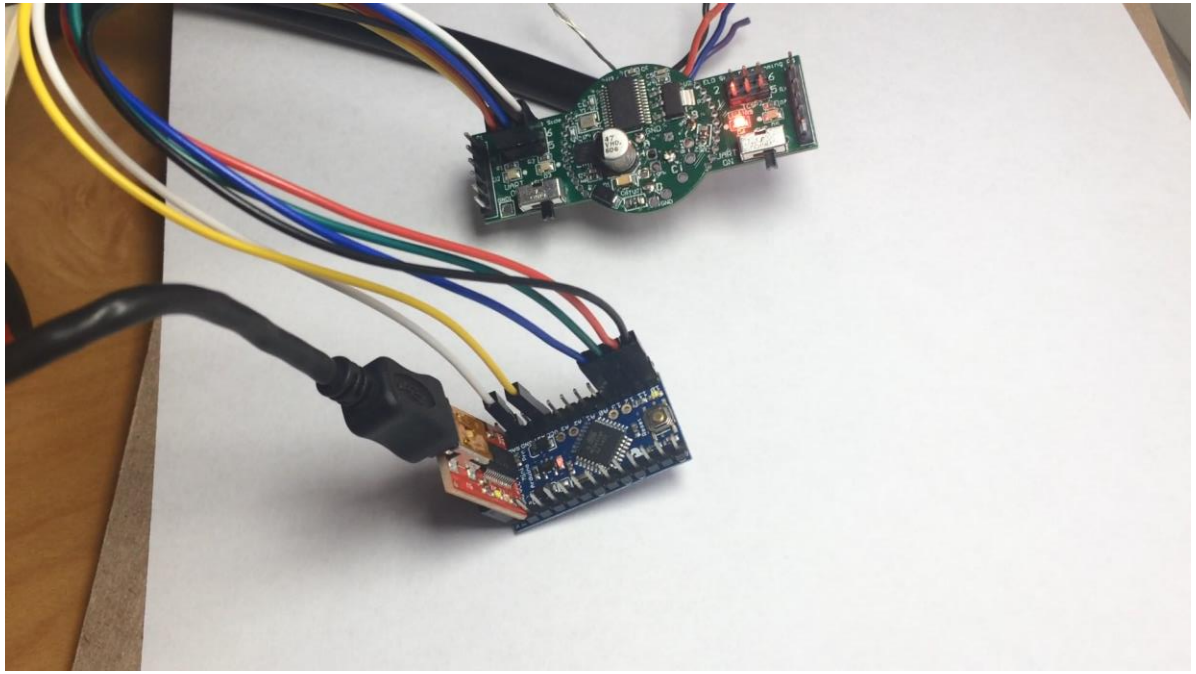
An ISP with an Arduino Pro Mini



Uploading a Program

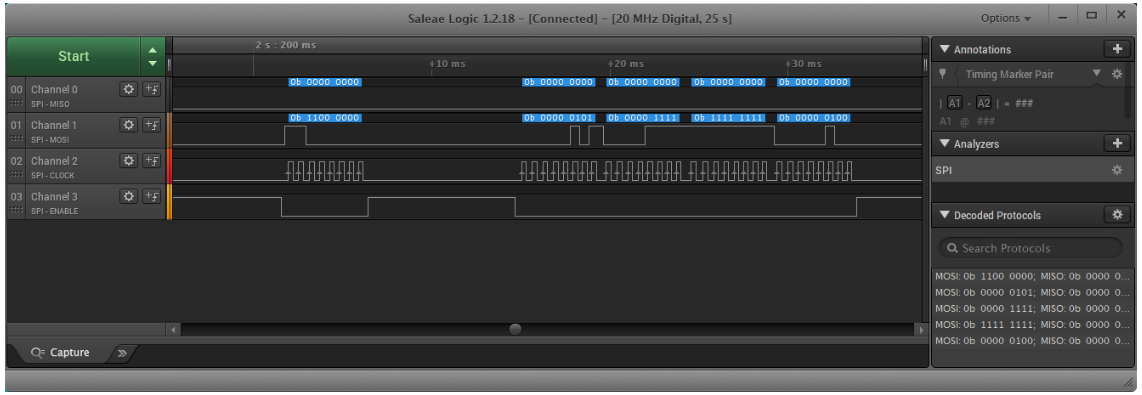
- Goal: write some basic code to ensure the programming process works.
- Result: Blinking LEDs!
- Next Steps:
  - Write a small SPI transfer function
  - Send the MCP CAN Controller commands to change the clockout prescaler.

Blinking LEDs



Arduino Bit-Bang SPI

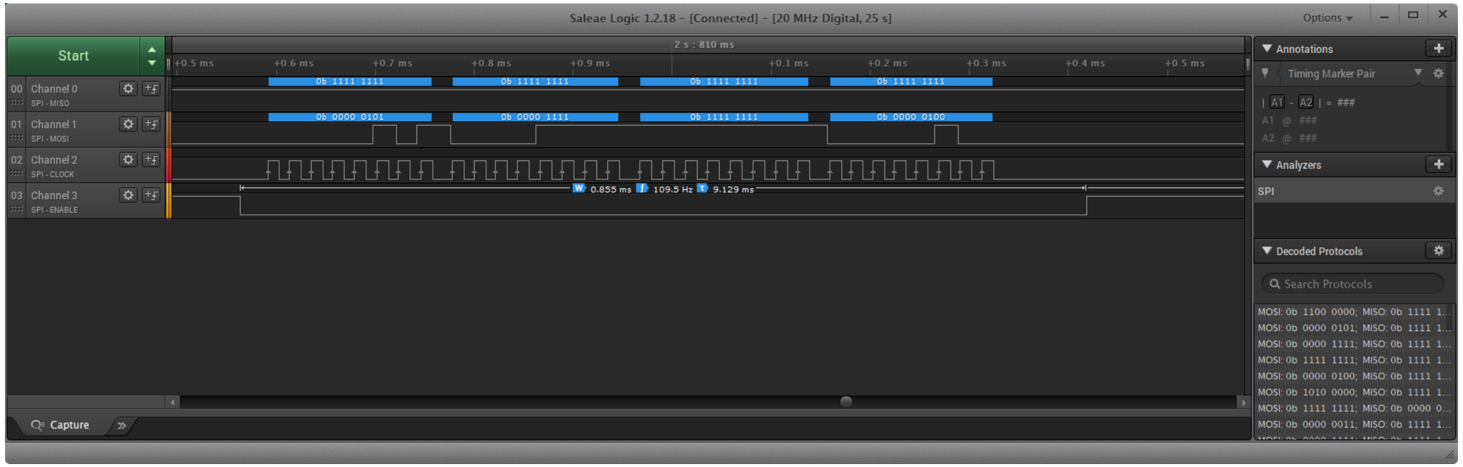
- Bit-bang with Ardunio digitalWrite and digitalRead: 13.26 ms for four bytes.
- Bitbank with direct port manipulation: 0.855 ms for four bytes.



Saleae Logic 1.2.18 - [Connected] - [20 MHz Digital, 25 s]

Decoded Protocols

```
MOSI: 0b 1100 0000; MISO: 0b 0000 0...
MOSI: 0b 0000 0101; MISO: 0b 0000 0...
MOSI: 0b 0000 1111; MISO: 0b 0000 0...
MOSI: 0b 1111 1111; MISO: 0b 0000 0...
MOSI: 0b 0000 0100; MISO: 0b 0000 0...
```



Saleae Logic 1.2.18 - [Connected] - [20 MHz Digital, 25 s]

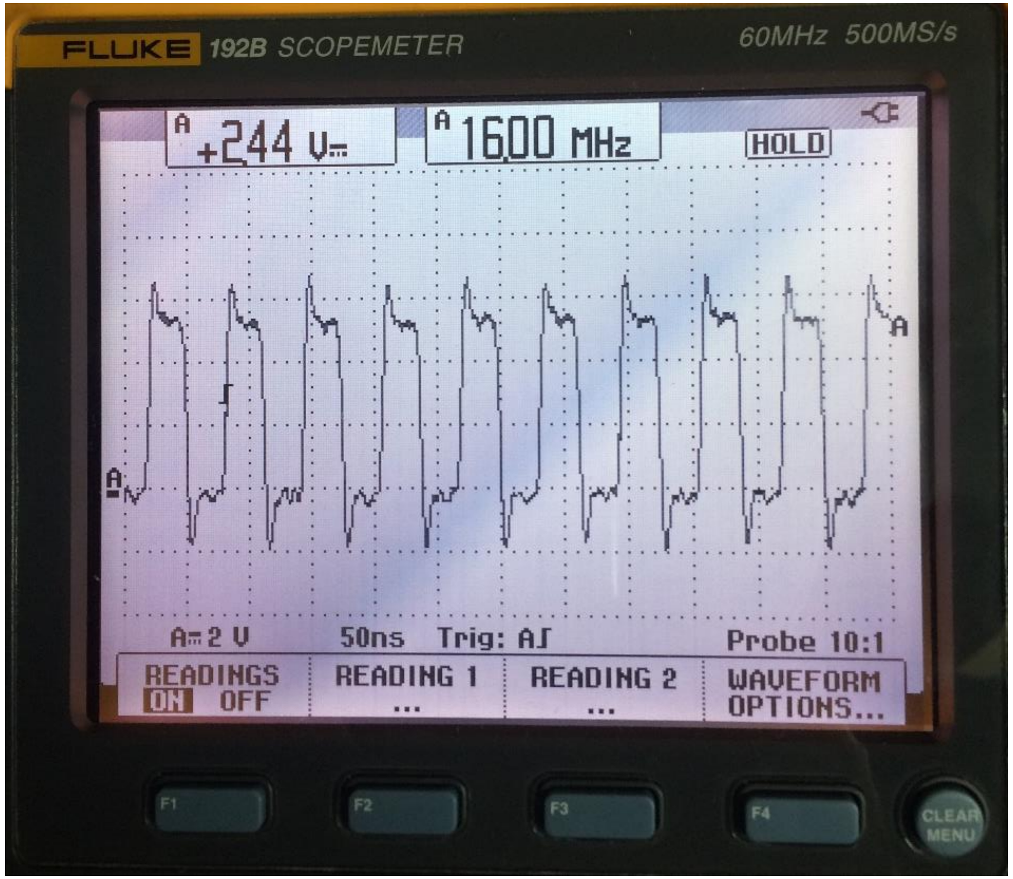
Decoded Protocols

```
MOSI: 0b 1100 0000; MISO: 0b 1111 1...
MOSI: 0b 0000 0101; MISO: 0b 1111 1...
MOSI: 0b 0000 1111; MISO: 0b 1111 1...
MOSI: 0b 1111 1111; MISO: 0b 1111 1...
MOSI: 0b 0000 0100; MISO: 0b 1111 1...
MOSI: 0b 1010 0000; MISO: 0b 1111 1...
MOSI: 0b 1111 1111; MISO: 0b 0000 0...
MOSI: 0b 0000 0011; MISO: 0b 1111 1...
```

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!

Test 4: Data Diode Function

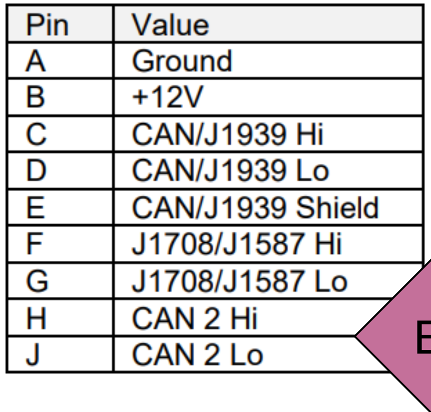
- Does CAN Data from J1939 show up on the ELD side?
- Does the device prevent CAN messages on the ELD side from being transmitted to the J1939 bus?
- Use BeagleBone Black with TruckCape for tests (2 channel)
  - Transmit messages using Linux SocketCAN "cangen"
  - Look for messages using "candump"

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

J1939 Connector

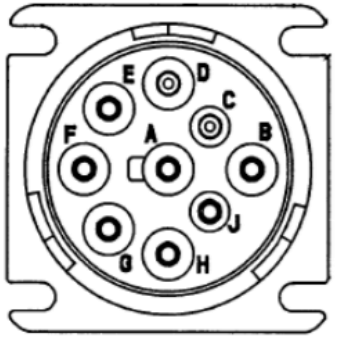
DG Technologies Product Pinouts and Industry Connectors Reference Guide

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



Pin	Value
A	Ground
B	+12V
C	CAN/J1939 Hi
D	CAN/J1939 Lo
E	CAN/J1939 Shield
F	J1708/J1587 Hi
G	J1708/J1587 Lo
H	CAN 2 Hi
J	CAN 2 Lo

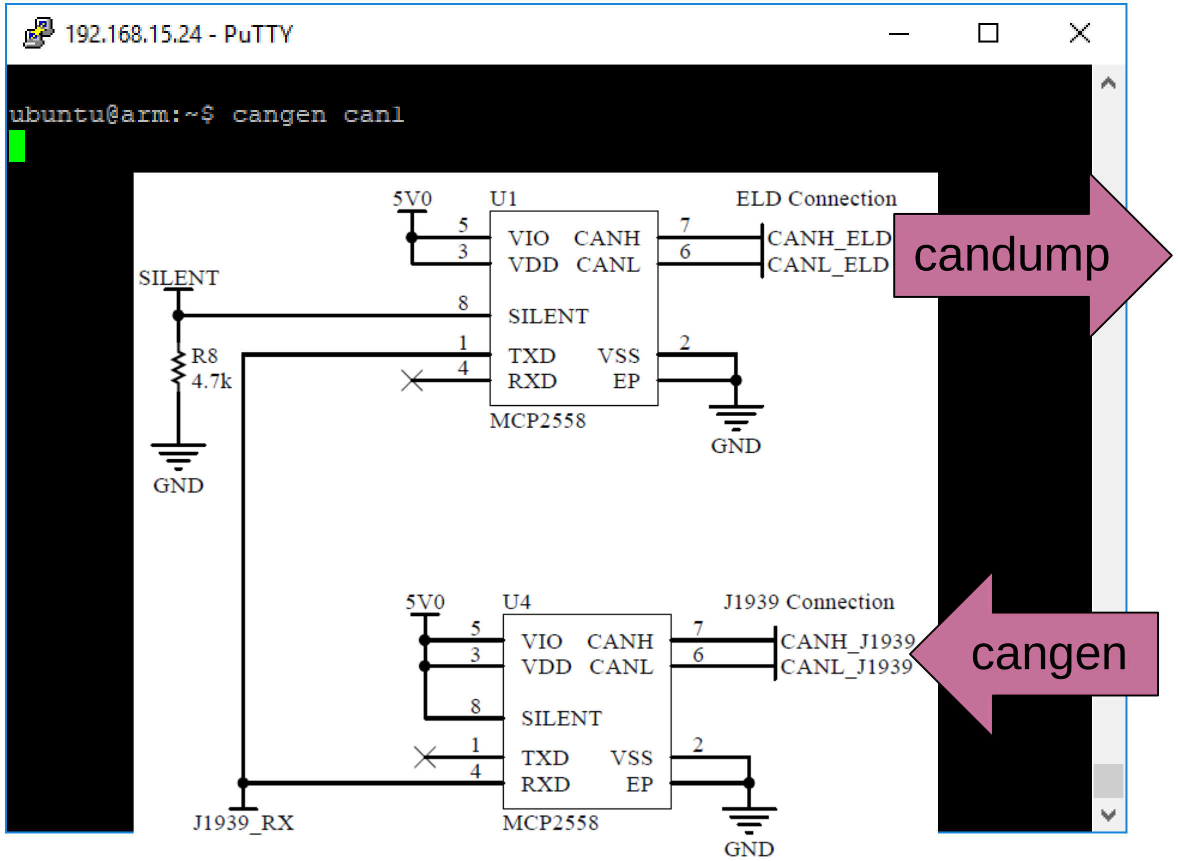
ELD



https://www.dgtech.com/wp-content/uploads/2016/04/Pinouts_ICR.pdf

Data Diode Test

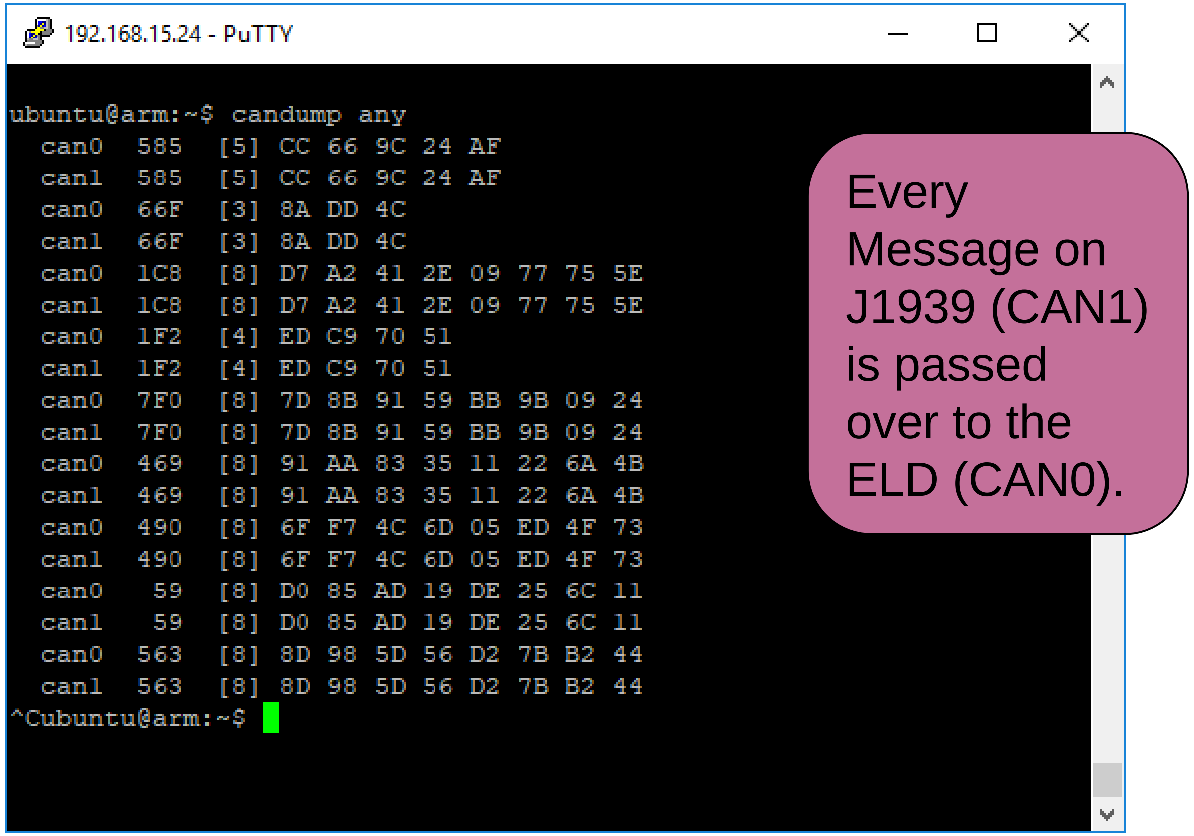
J1939 Side (CAN1) → → → → → ELD Side (CAN0)



```
ubuntu@arm:~$ cangen can1
```

candump

cangen

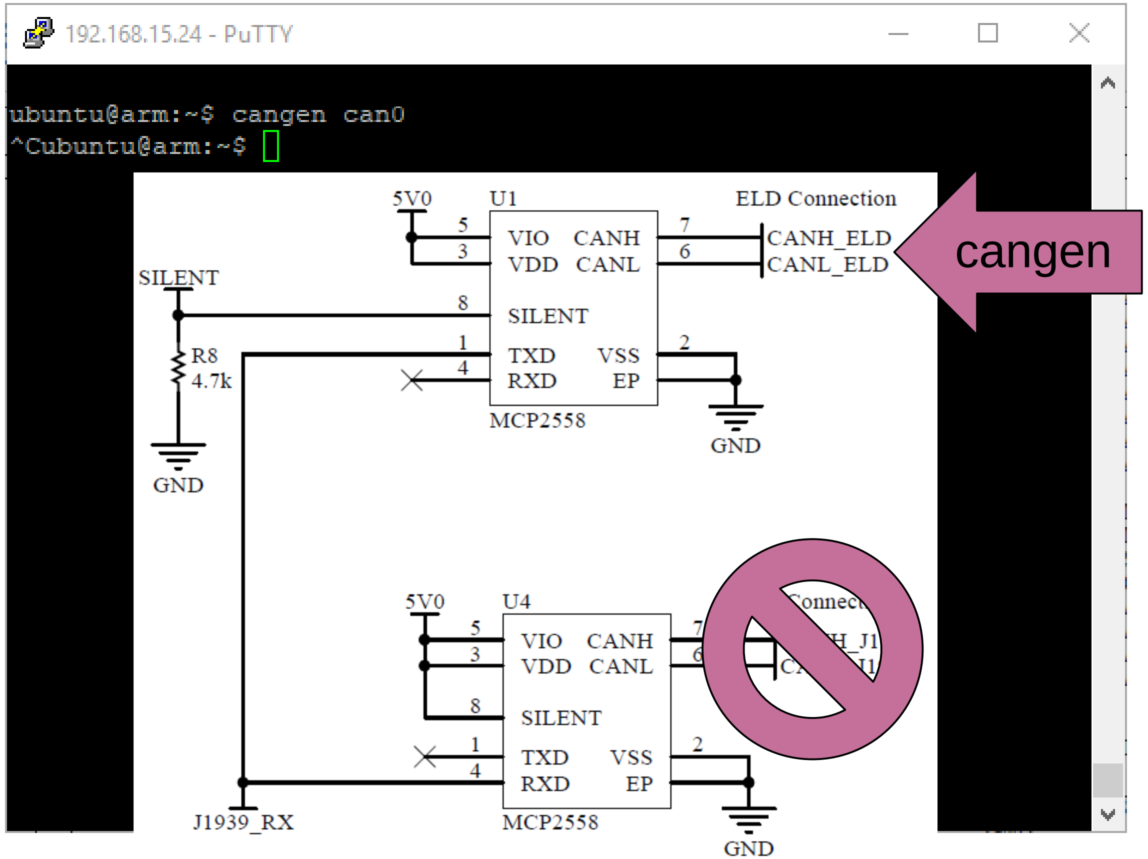


```
ubuntu@arm:~$ candump any
  can0  585   [5]  CC 66 9C 24 AF
  can1  585   [5]  CC 66 9C 24 AF
  can0  66F   [3]  8A DD 4C
  can1  66F   [3]  8A DD 4C
  can0  1C8   [8]  D7 A2 41 2E 09 77 75 5E
  can1  1C8   [8]  D7 A2 41 2E 09 77 75 5E
  can0  1F2   [4]  ED C9 70 51
  can1  1F2   [4]  ED C9 70 51
  can0  7F0   [8]  7D 8B 91 59 BB 9B 09 24
  can1  7F0   [8]  7D 8B 91 59 BB 9B 09 24
  can0  469   [8]  91 AA 83 35 11 22 6A 4B
  can1  469   [8]  91 AA 83 35 11 22 6A 4B
  can0  490   [8]  6F F7 4C 6D 05 ED 4F 73
  can1  490   [8]  6F F7 4C 6D 05 ED 4F 73
  can0   59   [8]  D0 85 AD 19 DE 25 6C 11
  can1   59   [8]  D0 85 AD 19 DE 25 6C 11
  can0  563   [8]  8D 98 5D 56 D2 7B B2 44
  can1  563   [8]  8D 98 5D 56 D2 7B B2 44
^Cubuntu@arm:~$
```

Every Message on J1939 (CAN1) is passed over to the ELD (CAN0).

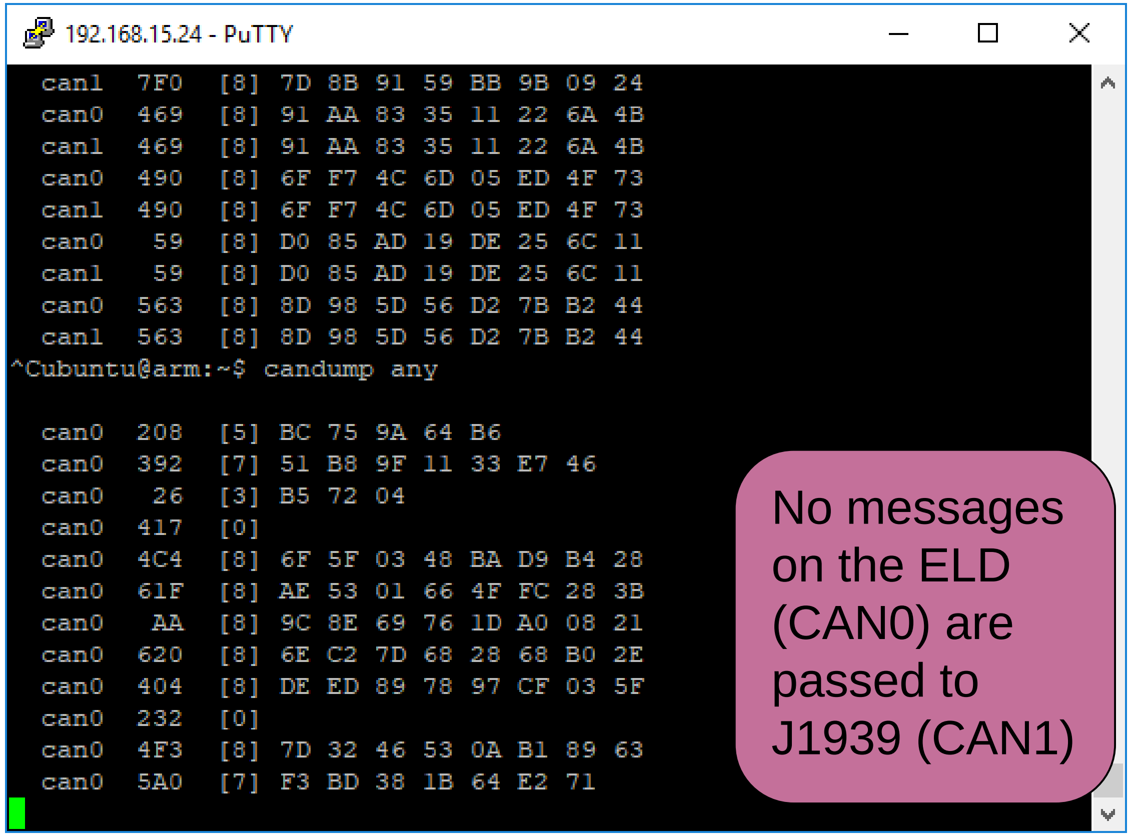
Data Diode Test

ELD Side (CAN0) → → → → → J1939 Side (CAN1)



```
ubuntu@arm:~$ cangen can0
^Cubuntu@arm:~$ 
```

cangen



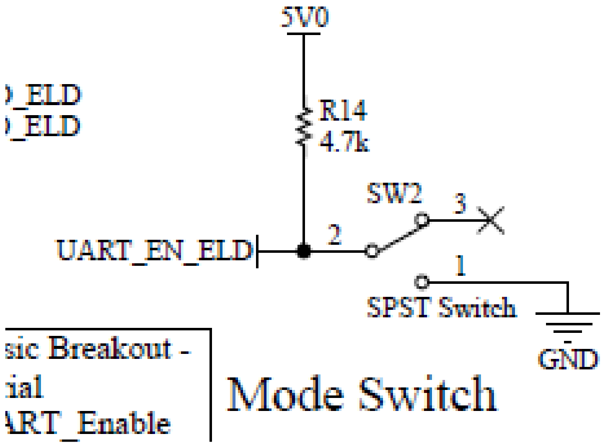
```
  can1  7F0   [8]  7D 8B 91 59 BB 9B 09 24
  can0  469   [8]  91 AA 83 35 11 22 6A 4B
  can1  469   [8]  91 AA 83 35 11 22 6A 4B
  can0  490   [8]  6F F7 4C 6D 05 ED 4F 73
  can1  490   [8]  6F F7 4C 6D 05 ED 4F 73
  can0   59   [8]  D0 85 AD 19 DE 25 6C 11
  can1   59   [8]  D0 85 AD 19 DE 25 6C 11
  can0  563   [8]  8D 98 5D 56 D2 7B B2 44
  can1  563   [8]  8D 98 5D 56 D2 7B B2 44
^Cubuntu@arm:~$ candump any

  can0  208   [5]  BC 75 9A 64 B6
  can0  392   [7]  51 B8 9F 11 33 E7 46
  can0   26   [3]  B5 72 04
  can0  417   [0]
  can0  4C4   [8]  6F 5F 03 48 BA D9 B4 28
  can0  61F   [8]  AE 53 01 66 4F FC 28 3B
  can0   AA   [8]  9C 8E 69 76 1D A0 08 21
  can0  620   [8]  6E C2 7D 68 28 68 B0 2E
  can0  404   [8]  DE ED 89 78 97 CF 03 5F
  can0  232   [0]
  can0  4F3   [8]  7D 32 46 53 0A B1 89 63
  can0  5A0   [7]  F3 BD 38 1B 64 E2 71
```

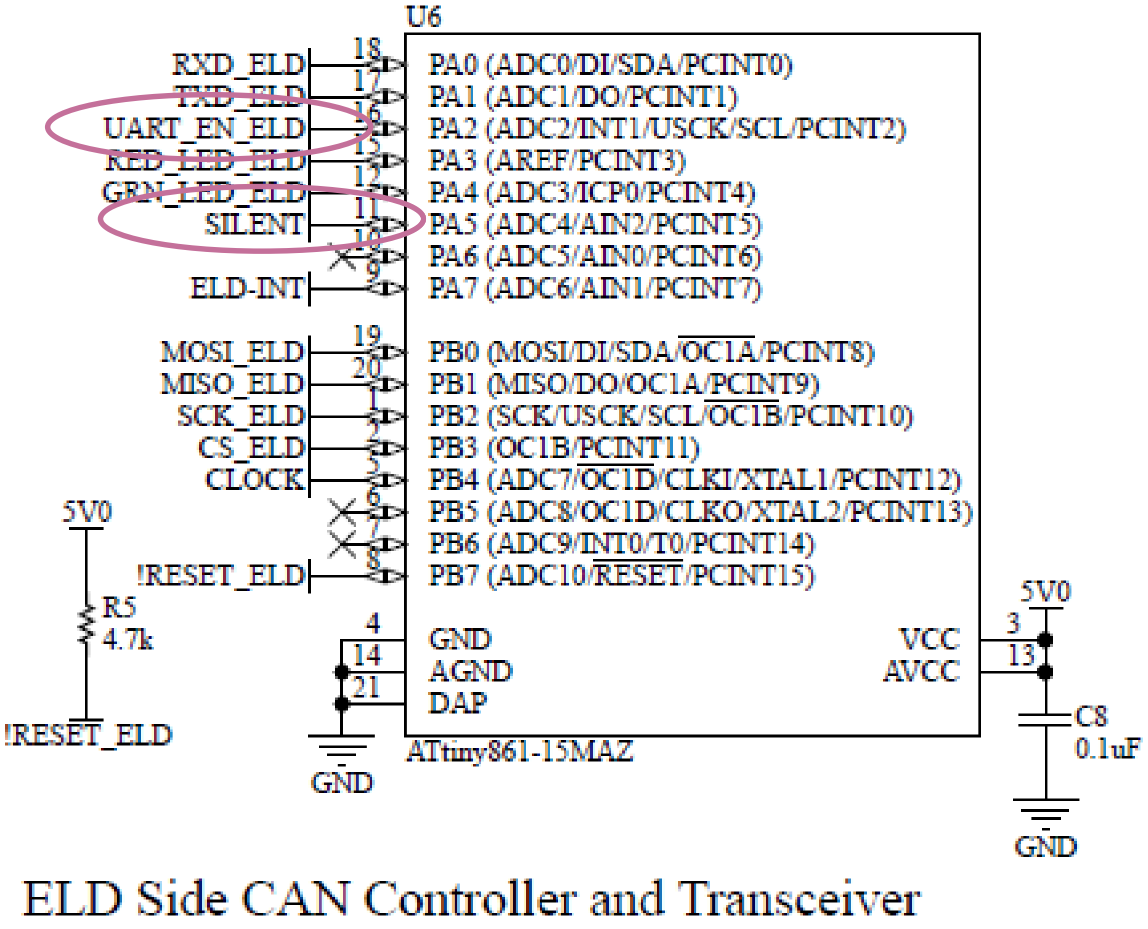
No messages on the ELD (CAN0) are passed to J1939 (CAN1)

Diode Functionality Works!

Switch ELD TX mode On/Off

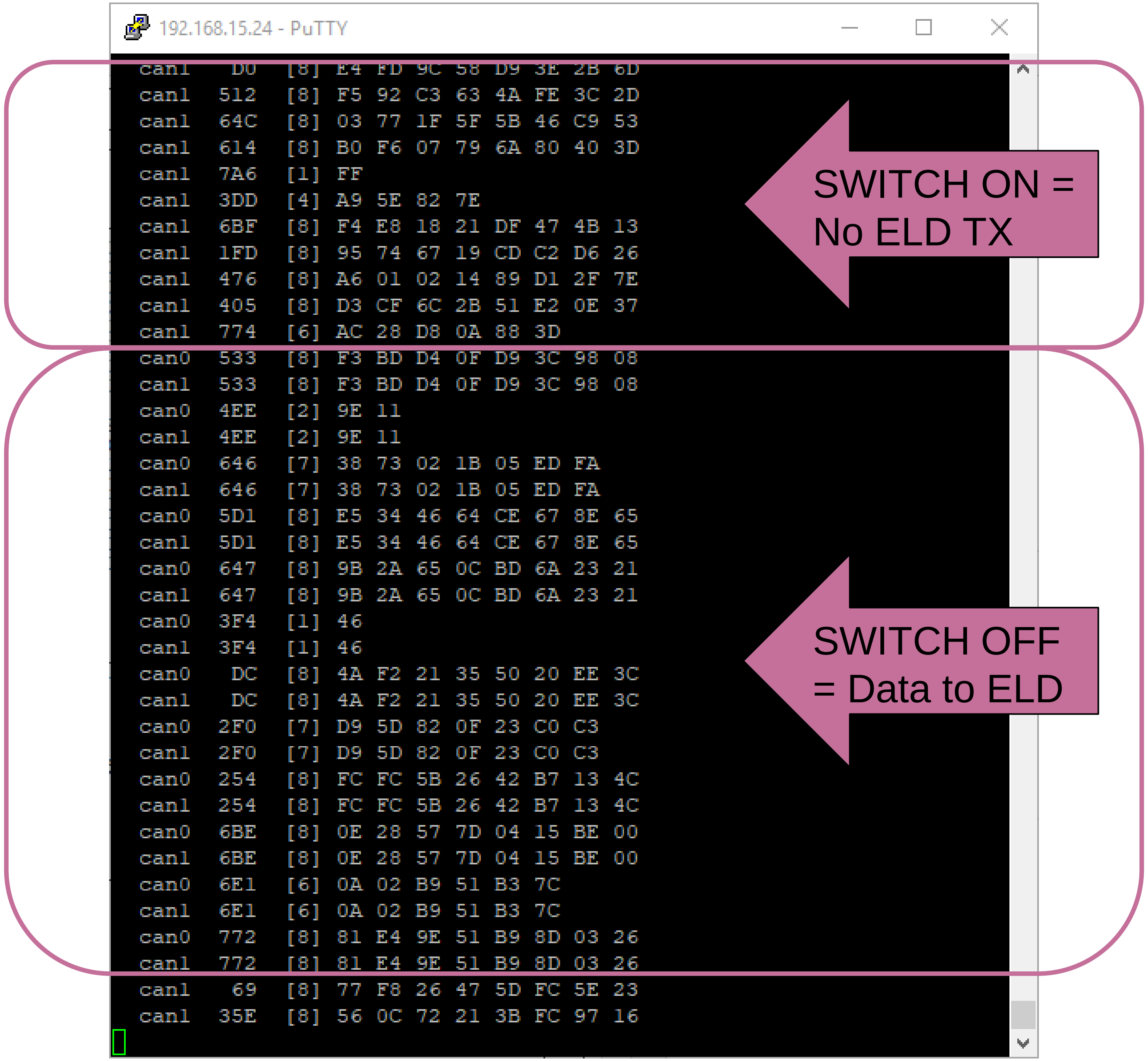
- Turn on and off the ELD Transceiver with the manual switch.
- Program the ATTiny to set the SILENT pin to the Switch Reading.





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.



```
  can1    D0   [8]  E4 FD 9C 58 D9 3E 2B 6D
  can1   512   [8]  F5 92 C3 63 4A FE 3C 2D
  can1   64C   [8]  03 77 1F 5F 5B 46 C9 53
  can1   614   [8]  B0 F6 07 79 6A 80 40 3D
  can1   7A6   [1]  FF
  can1   3DD   [4]  A9 5E 82 7E
  can1   6BF   [8]  F4 E8 18 21 DF 47 4B 13
  can1   1FD   [8]  95 74 67 19 CD C2 D6 26
  can1   476   [8]  A6 01 02 14 89 D1 2F 7E
  can1   405   [8]  D3 CF 6C 2B 51 E2 0E 37
  can1   774   [6]  AC 28 D8 0A 88 3D
  can0   533   [8]  F3 BD D4 0F D9 3C 98 08
  can1   533   [8]  F3 BD D4 0F D9 3C 98 08
  can0   4EE   [2]  9E 11
  can1   4EE   [2]  9E 11
  can0   646   [7]  38 73 02 1B 05 ED FA
  can1   646   [7]  38 73 02 1B 05 ED FA
  can0   5D1   [8]  E5 34 46 64 CE 67 8E 65
  can1   5D1   [8]  E5 34 46 64 CE 67 8E 65
  can0   647   [8]  9B 2A 65 0C BD 6A 23 21
  can1   647   [8]  9B 2A 65 0C BD 6A 23 21
  can0   3F4   [1]  46
  can1   3F4   [1]  46
  can0    DC   [8]  4A F2 21 35 50 20 EE 3C
  can1    DC   [8]  4A F2 21 35 50 20 EE 3C
  can0   2F0   [7]  D9 5D 82 0F 23 C0 C3
  can1   2F0   [7]  D9 5D 82 0F 23 C0 C3
  can0   254   [8]  FC FC 5B 26 42 B7 13 4C
  can1   254   [8]  FC FC 5B 26 42 B7 13 4C
  can0   6BE   [8]  0E 28 57 7D 04 15 BE 00
  can1   6BE   [8]  0E 28 57 7D 04 15 BE 00
  can0   6E1   [6]  0A 02 B9 51 B3 7C
  can1   6E1   [6]  0A 02 B9 51 B3 7C
  can0   772   [8]  81 E4 9E 51 B9 8D 03 26
  can1   772   [8]  81 E4 9E 51 B9 8D 03 26
  can1    69   [8]  77 F8 26 47 5D FC 5E 23
  can1   35E   [8]  56 0C 72 21 3B FC 97 16
```

SWITCH ON = No ELD TX

SWITCH OFF = Data to ELD

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