Truck Duck User Guide

http://truckhacking.github.io

Six Q. Volts, Esq. Haystack McStuffins, PhD

August 5, 2016

1 Introduction

The Truck Duck is an add-on board for the BeagleBone platform originally released at DEFCON 24 in August of 2016. The board and its associated drivers and libraries allow for easily connecting to heavy vehicle networks. This can facilitate security analysis, device testing, digital forensics, and prototyping ECUs for heavy vehicles.

1.1 Document History

- Preliminary release (5-aug-2016): Posted for DEFCON 24

1.2 Suitability For Cars

The Truck Duck can be used as a general-purpose CAN reversing, emulation and development platform. Since it runs Linux and has quite a large number of resources compared to other tools, it would be well suited for the task. HOWEVER, very little (zero) testing has been done with using the truck duck on passenger vehicles. A screw-terminal to OBDII adapter board is on the github repo (Although this was design for trucks, so the labeled pinouts won't be exactly correct).

1.3 Board Revision History

- Rev 1 (21-mar-2014): Initial prototyping run for hardware development.
- Rev 2 (21-jun-2016): Changed power circuit, J1708 design, smaller board
- Rev 3 (7-jul-2016): DEFCON24 board. Added duck logo.

2 Features

2.1 Vehicle Networking

The Truck Duck features two CAN 2.0B (J1939) interfaces and two J1708-compatible transceivers. One CAN interface and one J1708 interfaces are brought out to a DB-15 connector on the board using a ”semi-standard” pin-out for heavy vehicle diagnostic adapter cables. The second CAN interface is wired through to the 15-pin connector in a normally unused position. All four interfaces are brought out to screw terminals.

2.2 Power

The Truck Duck is capable of running off vehicle power provided by a diagnostic link cable, or via the screw terminals. The board accepts 7-24V input, pulling a maximum of 1.5A. The board power supply has reverse polarity protection, over-current protection (PTC thermal fuse), and transient voltage suppression. 12v and 24v vehicles are supported, although 24v support may be tricky with a running vehicle since this may trip the transient voltage suppression when the alternator kicks on. We didn’t have access to a running 24v vehicle to test, so this hasn’t been completely tested.

2.3 Connectors

The Truck Duck board features three connectors: A 15-pin DSUB connector and two 5-pin screw terminals. All interfaces and power are hooked up to the screw terminals. The DSUB connector is a semi-standard pin-out for heavy vehicle diagnostic cables. One CAN/J1939 channel and one J1708/J1587 channel are broken out into positions to be compatible with commonly available cables. A second CAN channel is broken out on two normally unused pins on the DSUB connector.

3 BeagleBone

The Truck Duck is designed to work with the BeagleBone Black, originally produced by BeagleBoard.org. All testing was done with revision-C boards. While everything should work with the older BeagleBone White, and the BeagleBone Greeen and Green Wireless versions, this is untested.

3.1 SD Cards

We highly recommend running the Truck Duck operating system image from an SD card rather than booting from the on-board eMMC. The on-board storage can be a constraint in both speed and capacity, so we prefer to use high-quality SD cards. With a fast card, you can write the image in few minutes where as imaging the onboard eMMC can take up to an hour, and you will still need an SD card to do so. We’ve have tested with the following cards (beware of counterfeit cards):

- Samsung PRO MicroSDHC 16GB/32GB
- SanDisk Extreme Pro MicroSDHC 8GB/16GB/32GB
- Lexar 633X MicroSDHC Cards 16GB/32GB

Imaging the SD cards can be done with Win32DiskImager on Windows or the dd command on Mac/Linux. We highly recommend using a fast USB3-to-SD adapter to write the images, or it will be quite slow. Once the card is imaged, you will need to expand the size of the file system to fill the card you have so that you don't run out of space. This can be done with gparted on another Linux machine, or with fdisk and resize2fs on the BeagleBone itself.

3.2 Operating System

The Truck Duck runs a modified version of Ubuntu 14.04 using a 3.8 series kernel. The kernel has been updated to provide support for J1939 messages on top of CAN and the Programmable Real-time units (PRUs) on-board the processor are invoked for the J1708 interfaces. Several device-tree overlays have been used to handle the work of configuring the hardware for the interfaces. A start-up script configures the overlays for the interfaces and brings them up automatically. The default username for the image is "ubuntu" and the default password is "truckduck".

3.2.1 Baud Rates

The CAN/J1939 interfaces have adjustable baud rates up to 1Mbit/s. You can configure them as follows:

```
sudo ip link set can0 type can bitrate 250000
```

The J1708 interface speed is fixed speed at 9600 baud. The J1708 bus speed is not variable and is fixed in software due to timing constraints imposed by the standard.

4 Building the Board

While not something you would want to do as your first-ever soldering project, the Truck Duck can be hand assembled. Most of the passive components 0603 and there some simple fine-pitched parts (TSSOP and SOT-23-5). If you are not familiar with PCB fabrication, the boards can be bought in small quantity from a vendor like OSHPARK or another prototyping service by providing them the gerber files, which are available on the github.

Parts can be bought from a vendor like Digikey or Mouser. Links to create a cart with all of the components for enough to build one copy of the newest version of the board from both vendors can be found on the github. Assembling the board will require a reasonable soldering iron. If you need to buy one, a Hakko or Weller in the $100 range is good starting point if you plan on doing a reasonable amount of electronics assembly and hacking.

4.1 Testing the board

Included on the operating system image is a testing script in the home directory:

```
python3 test-device.py
```

You will need to hook up both CAN interfaces to each other and use terminating resistors and connect the J1708 interfaces together. No termination is required for the J1708 interfaces, they just need to be wired together. The test script will attempt to send data from one interface to another, and then back the other direction for both CAN and J1708. This ensures that send and receive on all four interfaces is working.