# Assembly of a DEF Emulator

Table of Contents

***This documentation and software is distributed in the hope that it will be useful, but***

***WITHOUT ANY WARRANTY; without even the implied warranty of***

***MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.***

## Background

This document and software was developed to deal with the failing of DEF sensor packs (also known as DEF heads or DEF headers) that are used on motorhome chassis that have left, and will continue to leave, motorhomes in an unsafe situation in a derated performance mode limiting speed to 5 mph. The emulated data traffic will make the Cummins engine control module (ECM) believe that all is well in order to avoid the derate condition such that the motorhome can be safely driven to a service center for repair. This does not disable any of the OEM emissions controls. The operator is responsible for maintaining the DEF fluid level and quality according to the manufacturer specifications at all times that the vehicle is in operation.

This condition has been prevalent in late 2020 and 2021, which along with COVID made DEF sensor replacement packs hard to acquire and often took several months. Motorhomes built on Spartan chassis utilizing the Shaw Development DEF sensor was the primary chassis having failures but it has also occurred on Freightliner and other chassis.

## Overview of Solution

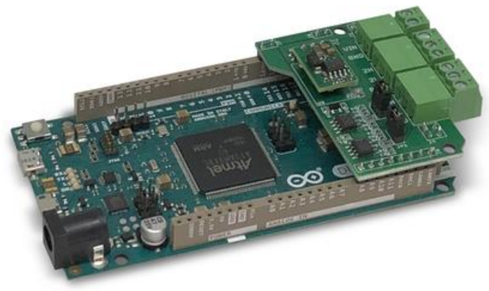
A motorhome has several J1939 controller area network (CAN) buses. One such bus includes the various emission related sensors and controls and is connected directly to the Cummins ECM. The DEF sensor made by Shaw Development (as well as other manufacturers) utilizes a Deutsch DT04-4P connector with 4 pins. The mating connector on the J1939 bus is a Deutsch DT06-4S. The connector has four pins which are CAN-H, CAN-L, 12V, and Ground. The *typical* wire coloring for these are CAN-H (yellow), CAN-L (green), 12V (red), and Ground (black), but can vary depending on manufacturer. You must always verify your connections and don’t rely on the wire color.

The software that has been developed for the emulator device runs equally well on an Arduino Due, Arduino UNO or an Arduino Mega2560 device. The Arduino boards have on-board voltage regulation so they will operate off of nominal 12V from the vehicle connector so a separate external power supply is not required. It is recommended that you use Arduino brand boards.

This emulator has been shown to work on Cummins ISL9 and ISB6.7 engines with  the CM2350 ECM.  This emulator will NOT work on older DEF heads that do not use J1939 communications or those without a 4-pin connector.

# Hardware

The current hardware options to construct a DEF Head Emulator are all based on Arduino platforms. There are three specific Arduino development boards that have been successfully tested, the Arduino UNO, ATMega2560 and DUE. In addition to the development board, additional parts are required to interface with the CAN (Controller Area Network) on your coach. This takes the form of an add-on circuit board and is commonly known as a “Shield” in Arduino-speak. The Arduino DUE is available from Copperhill Technologies with the CAN shield already attached, so at this time is probably the easiest variant to put together for those not familiar with the technology. Cost of the board with CAN shield pre-installed is currently about $100 delivered. Arduino DUE with dual CAN bus and Extended Range Power Supply.



The UNO and the Mega 2560 both require the user to buy a Seeed Studios CANBUS V.2 shield which will need to be plugged into existing headers on the Arduino development board to form a two board “piggyback” or “sandwich”.

In addition to the development board and the CANBUS interface device, a special connector and a cable are also required. The specific parts required are specified in the next section. At a minimum, the assembly process will involve cutting a cable, stripping insulation off the wires and inserting bare wire ends into screw terminals (no soldering required). An enclosure of some sort is also required as the electronics should not be exposed to moisture. The total cost of the board and cable assembly (not including an enclosure) is about $125 including shipping. We will describe in detail the steps to assemble the DUE Hardware and program the DEF Emulator software. We will use the Arduino DUE platform with the Copperhill Technologies CANBUS transceiver and we will describe the few differences that apply to the UNO and Mega2650 with the  Seeed Studios CANBUS V2 shield board.

### IMPORTANT NOTE APPLICABLE TO ALL HARDWARE OPTIONS:

Whichever hardware option you select, all of them will come with what is known as a “Termination Resistor” installed or enabled. This resistor MUST be disabled before the device is connected to the live vehicle CANBUS network. On the DUE with the CopperHill CANBUS interface you must remove a jumper to disable the terminator. On the UNO or Mega2560 with the Seeed Studio CANBUS you will need to use a sharp utility knife to cut a trace on the Seeed circuit board to disable the resistor.

## Required Components

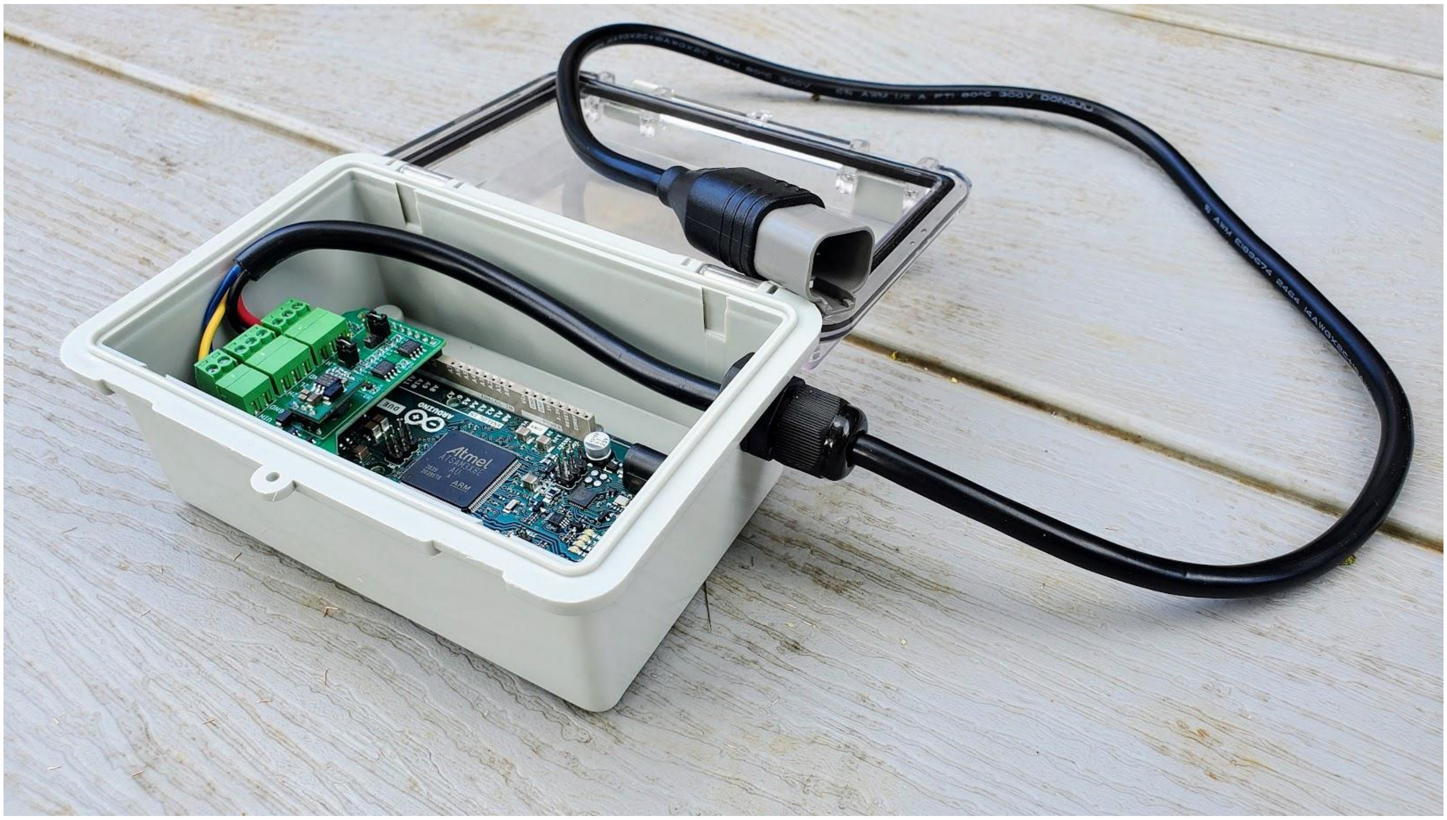
**Arduino DUE:**

- **Development Board** - Arduino DUE with dual CAN bus and Extended Range Power Supply. ***Be absolutely sure you order the optional Extended Range Power Supply option by clicking the “Yes” selection for the option on the ordering page!***

- **DT 4 pin Cable Assembly** - this is a cable and connector that will connect the DUE to the CAN on your coach at the DEF tank.
    - Option 1 - Recommended - this cable + molded connector is a bit more expensive but will save you some time and a few additional parts. The 1m (~39") version is plenty long for most applications.
    - Option 2 - If desired a prewired DT 4 pin connector and separate cable can be utilized.
        - Prewired DT 4 pin connector - order the 16 gauge version.
        - Length of 4 conductor cable - this is only if you want to place the emulator farther away from the 4 pin DEF Sensor connector on the vehicle wiring harness than the prewired connector with pigtail allows for. Any 4 conductor cable will probably work unless you use more than a few (like 4 or 5) feet. Longer than that should probably use "twisted-pair" cable. Sourcing the cable is up to the user.
        - Adhesive lined heat shrink crimp-on butt connectors or electrical tape
    - **Enclosure** - there are many options for a case, at this time there is no perfect choice. In a pinch a food storage container with a snap on lid or even a ziplock bag and some duct tape can be utilized, however the DEF tank area is exposed to indirect weather, so for maximum protection a waterproof case is recommended. The cases listed below have been proven to fit the DUE hardware.
    - LMioEtool Waterproof IP65 Electrical Project Enclosure 5.9 x 3.9 x 2.8inch (shown in photo below)
    - https://www.amazon.com/dp/B07C97BBMX?psc=1&ref=ppx_yo2_dt_b_product_details

- **Waterproof Cable Gland** - seals the connecting cable where it enters the case, they work best with round cables like the one specified. The PG-9 size is the best fit for the DT-4 cable assembly listed under Option 1 above, however the PG-11 size gland will also fit. Note for this part to work properly a case with flat, smooth sides is required.
- Magnets or Velcro to temporarily affix the completed enclosure inside the DEF tank compartment.
- Small screws or double sided tape to affix the board to the inside of the case.

NOTE: All of the descriptions for the enclosure and mounting are just suggestions as a starting point. You are free to use whatever you like from a custom-made weatherproof enclosure all the way down to a Zip-Loc freezer bag and some duct tape. The same goes for the cable and connector. As long as you end up with a Deutsch DT04-4P connector with the right 4 pins connected to the right 4 places on your hardware, how you get there is entirely up to you.



## Assembling Components

### Arduino DUE:

Tools Required:

- Wire strippers.
- “Jewelers” ~ 3/32” or smaller flat blade screwdriver for the screw terminals on the CAN shield.
- If building a waterproof assembly, a ~9/16” drill bit is required to install the PG-9 waterproof gland in the plastic case. A step type drill bit works well for this.
- A utility knife or X-acto for cutting terminator circuit board trace

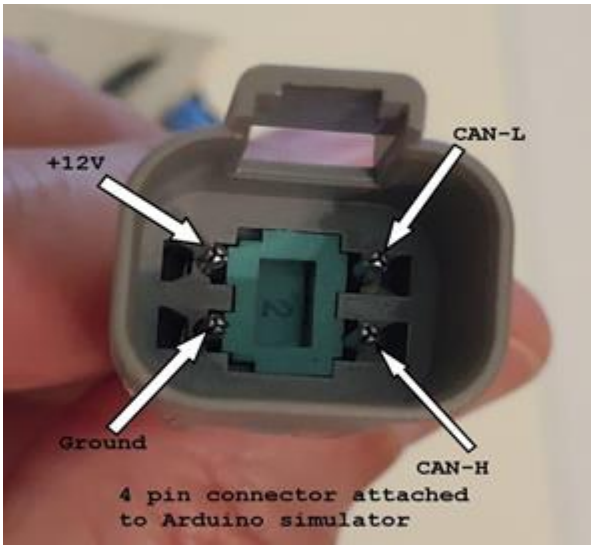
*Note: Leave the DUE loose inside the case until the software installation has been completed. The software installation requires access to the micro-USB connector on the DUE, but that connection is inaccessible while the DUE is seated inside the case.*

Assembly:

The DUE ships with a plastic base for protection. Leave it in place unless your mounting plans require it to be removed. The CopperHill CANBUS Board should already be installed but if it doesn’t just carefully mount it on the end of the DUE board by lining the pins on the CANBUS board up with the header sockets provided on the DUE and seat it firmly. The green screw terminals should be on the opposite end from the DUE power connection.

### Cable and Connector

The cable and connector are self-explanatory. You need a cable with the 4-pin Deutsch Plug attached. You must identify how the  wires are connected to which pins. The wires must be positively identified so you know which wire goes to each connector on the board assembly. The wiring connections  on the 4 pin DT-4 cable connector are: +12v, ground, CAN-H and CAN-L.

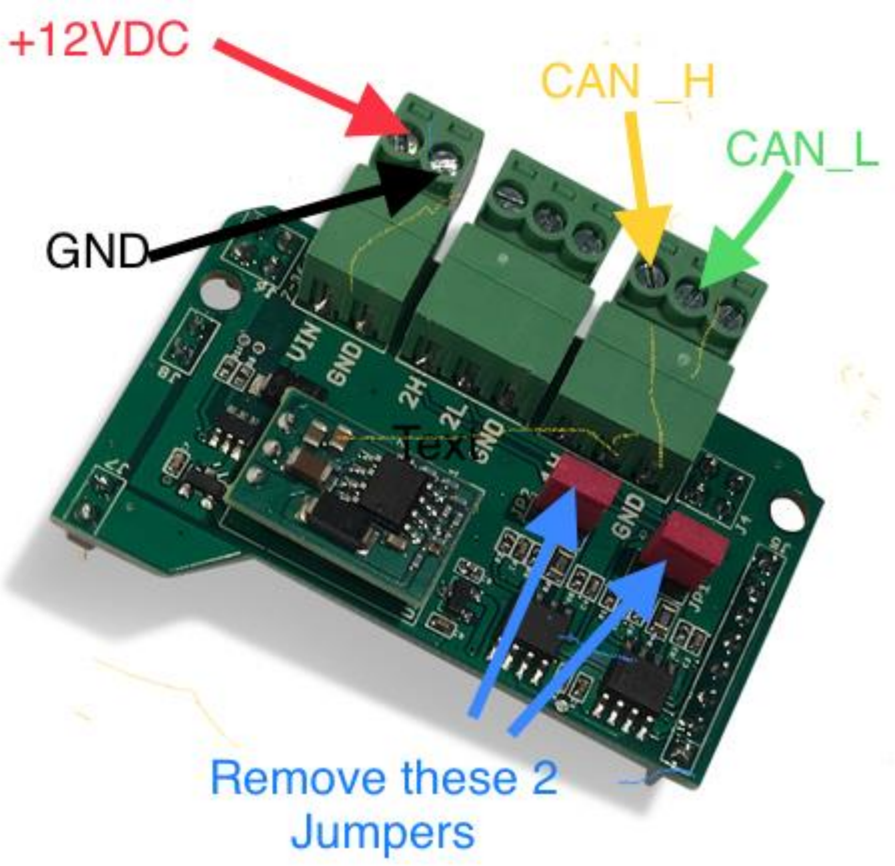


4 pin connector attached to Arduino simulator

The wires associated with these pins (above picture) are connected to the DUE via the screw terminals.

Strip about 3/16 inch of insulation off each wire end and connect to the  screw terminals.  If the wire is too large, simply cut off a few strands.  Make sure there are no stray wires sticking out.  The wires must be fed through the gland nut and case before connecting to the DUE.

Here is a picture of where each wire from the plug should be connected on the DUE assembly. ***Remember to remove the 2 jumpers to disable the termination resistors***



**IF You Have an UNO or Mega 2560 Board Then The Following Differences Should Be Noted. The Procedures Are Slightly Different**

[Amazon link to Aurduino UNO]()

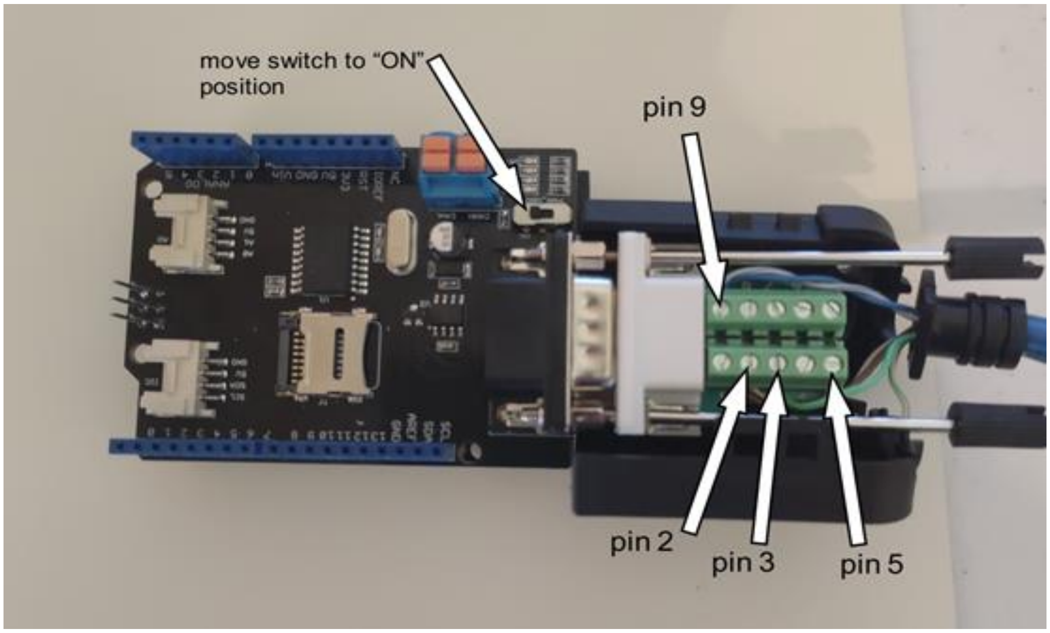
[Amazon link to Arduino Mega 2560]()

[Amazon link to Seeed Canbus V2]()

First difference is that the UNO and the Mega use a Seeed Studios CANBUS V2 board that is mounted on top of the main board. This means the cable connections will be made to a DB9 Receptacle that mates to the DB9 connector on the Seeed board. (see picture below). Just assemble your 4-pin plug and cable and verify the correct wires for each of the connections and wire the DB9 Receptacle onto the end of your cable as indicated in the picture.

Second difference is the Seeed board has a tiny on/off switch that should be set to “ON”. This allows the board to be powered from the DB9 connector and it should be in the position shown in the picture (see picture)

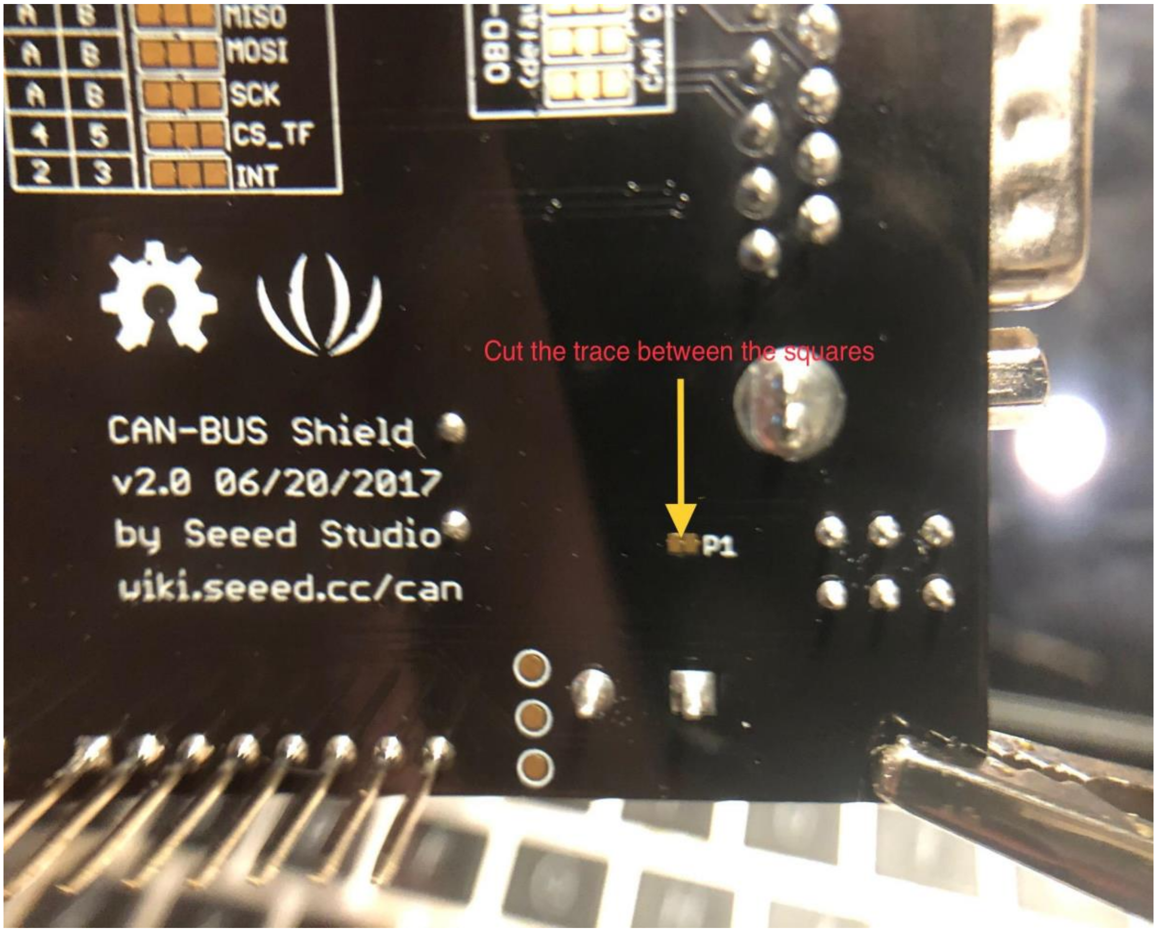
Pin 2 = ground   Pin 3 = CAN-H    Pin 5 = CAN-L    Pin 9 = +12v



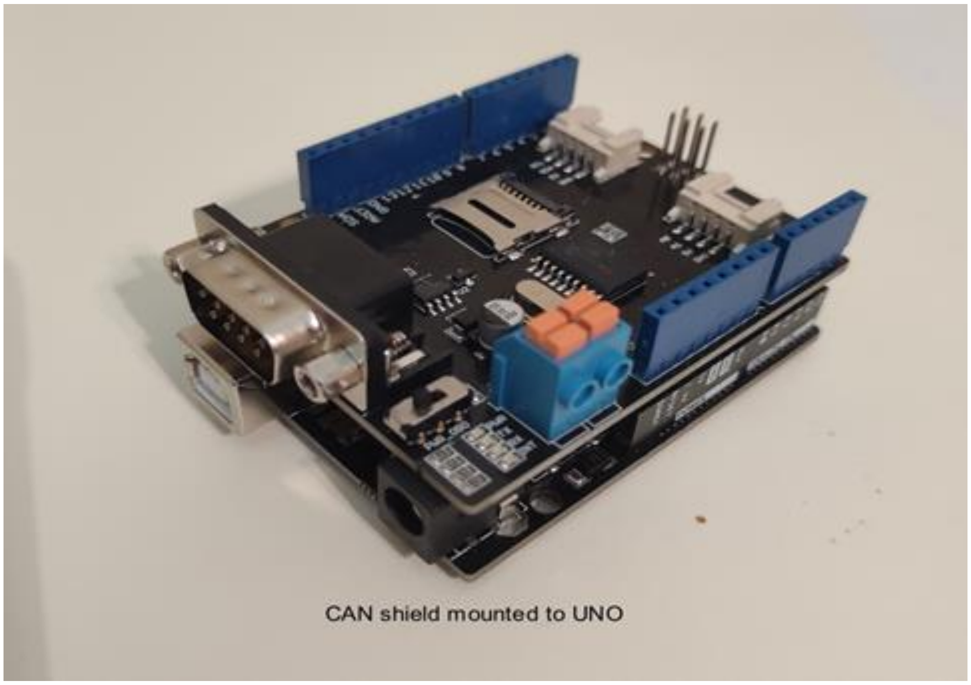
YIOVVOM DB9 Breakout Connector to Wiring Terminal RS232 D-SUB Male Serial Adapters Port Breakout Board Solder-Free Module with case( Female Serial Adapter)

Make sure to move the switch on the CAN shield to the “ON” position.

The termination resistor on  the CAN Bus Shield must be disabled by cutting a circuit board trace  with a utility or X-acto knife. The trace to cut is labeled “P1” and is located on the opposite side from where the components are mounted on the Seeed CANBUS interface (see photo). It won’t take much pressure with the blade but make sure the trace is completely cut. Use a magnifying glass or better still, use an ohmmeter to verify that the resistance between the two CANBUS connectors is significantly higher than 120 ohms. Just measure between the 2 orange and blue spring-loaded test terminals on the Seeed board.

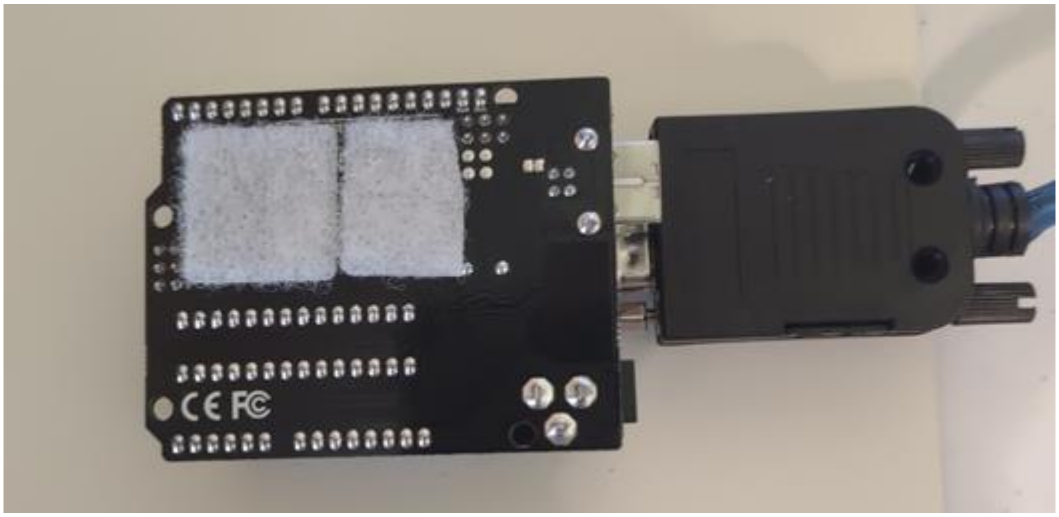


The CAN shield needs to be mounted on top of the UNO or Mega2560 board. The DB9 connector on the Seeed should be at the same end as the power socket and USB connectors on the Arduino board. There is only one way for it to attach but you must be very careful to line the pins on the shield with the socket holes on the microcontroller board. The pins are very easy to bend and not very easy to straighten back out.

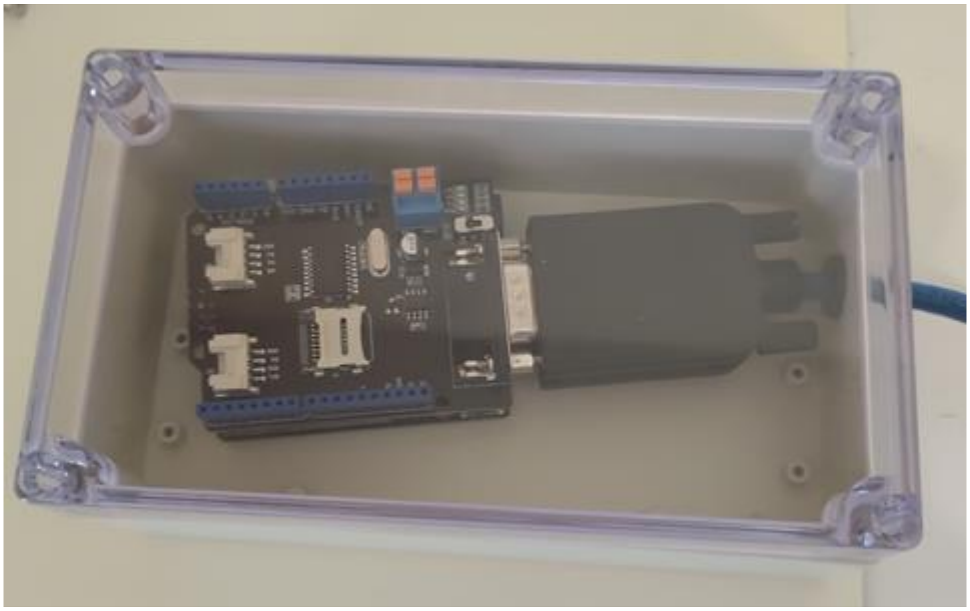


CAN shield mounted to UNO

The finished board assembly can be secured in the box bottom using Velcro.



Completed assembly:



**NOTE: before mounting in the enclosure box, the Arduino needs to be programmed! Please see instructions in the Loading Software section.**

## Overview

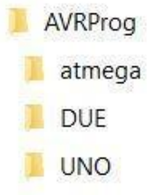
## Loading Software onto Arduino

It is assumed that someone attempting to construct one will have a modest knowledge of a Windows based computer. The initial work was completed on a Windows 10 based machine, but it is expected that a Windows 7 or XP based machine will also work. The computer will need to have at least 1 available USB 2/3 port.

The software for the Arduino can be downloaded from: https://github.com/flyboy013/DEF_Sensor_Emulator_Firmware.

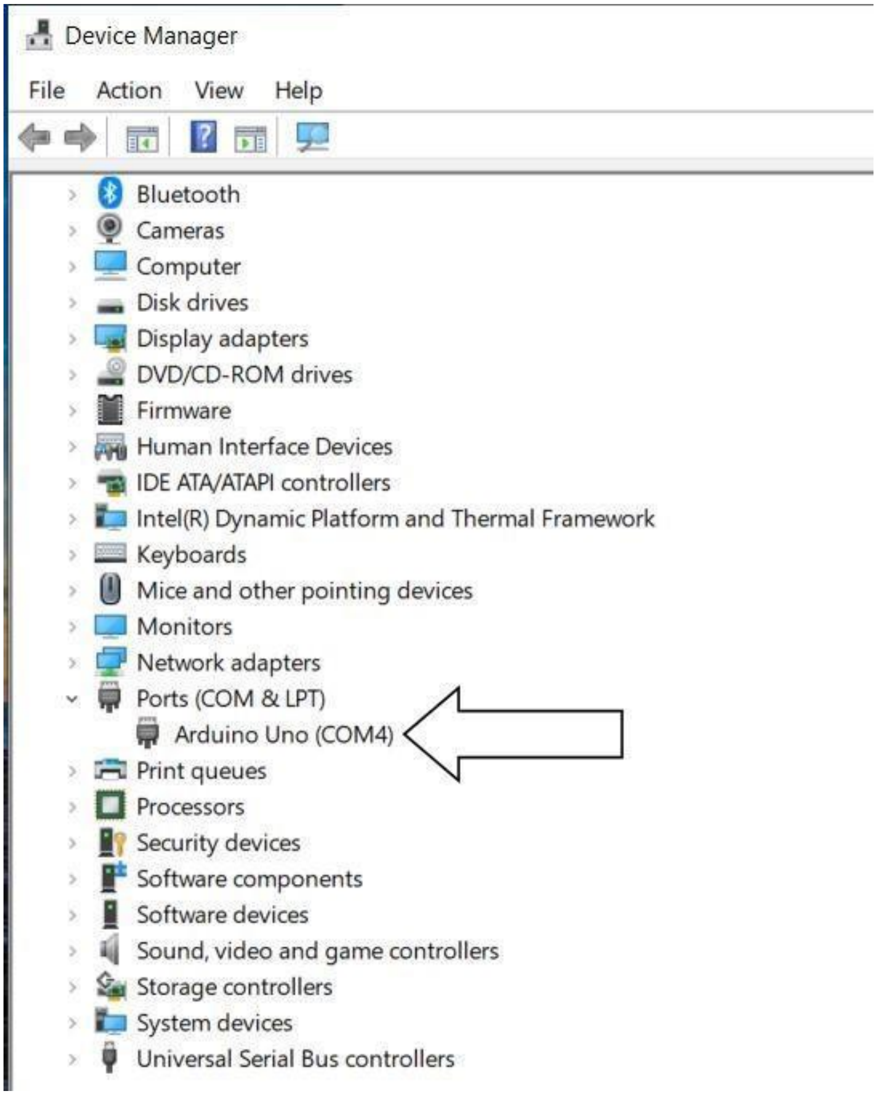
The download file (DEF Emulator Install Files (x32).exe) is a self-extracting file that will include the compiled Arduino software appropriate for your hardware configuration ( .bin or .hex) and files for actually programming the Arduino. It is not necessary to install any additional software from Arduino or anywhere else.

Save the file somewhere convenient on your computer. The Windows Desktop will do nicely. After the file is downloaded, just double-click the file and it will automatically create the required directory structure on your C drive and copy all necessary files into their appropriate locations. It will prompt you to verify the destination directory, just accept the suggested default. You may see warnings from your anti-virus software on your Windows computer and you may have to deal with those to allow the installation to run. The downloaded installation package will create the following directory structure in the root of your C:\ drive:



- AVRProg
  - atmega
  - DUE
  - UNO

Using an appropriate USB cable, attach your Arduino board to your computer. Please note that the DUE has 2 USB connections, the one nearest the black power connector is known as the “Programming Port” and it is the one you should use. Note that the preceding sentence applies *only* to the DUE. Some LEDs will light up on the board showing power is connected via the USB cable. Open the Windows “Device Manager” tool and expand the “Ports (COM & LPT)”. You will need to make note of the number of the COM port being used by the Arduino. Note that the Port may also be shown as a “USB Serial Device (COM xx)”.

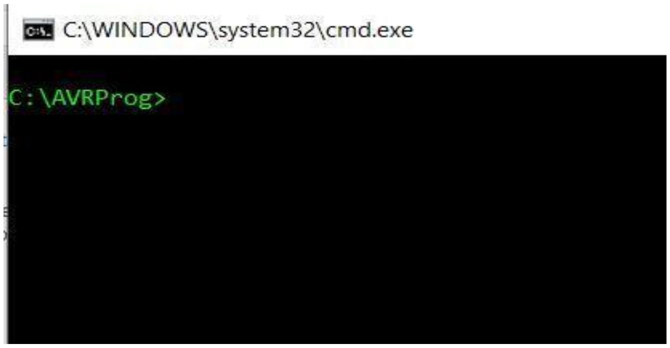


Bluetooth
Cameras
Computer
Disk drives
Display adapters
DVD/CD-ROM drives
Firmware
Human Interface Devices
IDE ATA/ATAPI controllers
Intel(R) Dynamic Platform and Thermal Framework
Keyboards
Mice and other pointing devices
Monitors
Network adapters
Ports (COM & LPT)
Arduino Uno (COM4)
Print queues
Processors
Security devices
Software components
Software devices
Sound, video and game controllers
Storage controllers
System devices
Universal Serial Bus controllers

Close the Device Manager window.

Open the Windows CMD program ( on Win10 right click the windows icon at lower left, select run, type CMD in the box and hit OK). A “CMD” window should open.

In the CMD window type: *cd\AVRProg* and then hit the enter key. The prompt should now indicate: C:\AVRProg.



```
C:\AVRProg>
```

The file “program.bat” will be used to upload the compiled Arduino software to your board. Program.bat takes 2 inputs to operate correctly: the board type and the COM port number. The board type can be UNO, DUE or atmega. You can use all upper-case or all lower-case but you cannot mix upper and lower cases. The COM port number will be 1,2,3 etc (just use the number part. Don’t type the ‘COM’ OR ’USB‘ part) that you found from Device Manager.

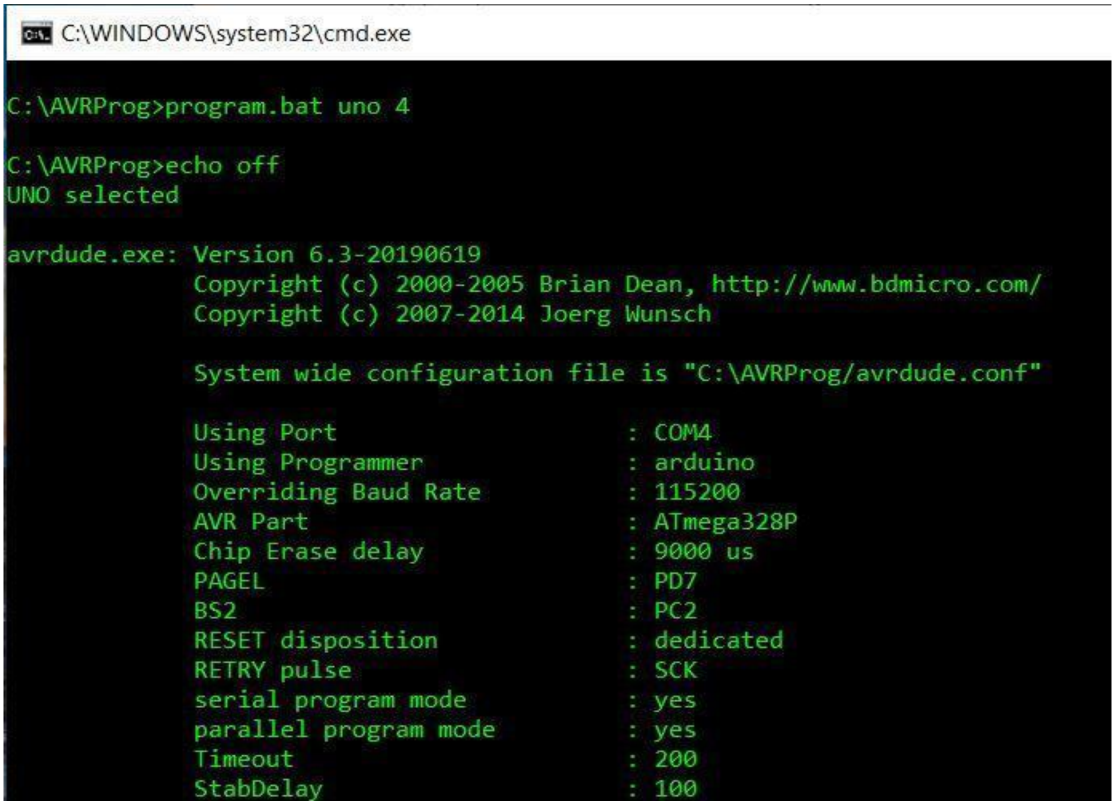
As an example to program a UNO on COM port 4 use:

```
Program.bat  uno  4
```

For a DUE on port 3 use:

```
Program.bat  due  3
```

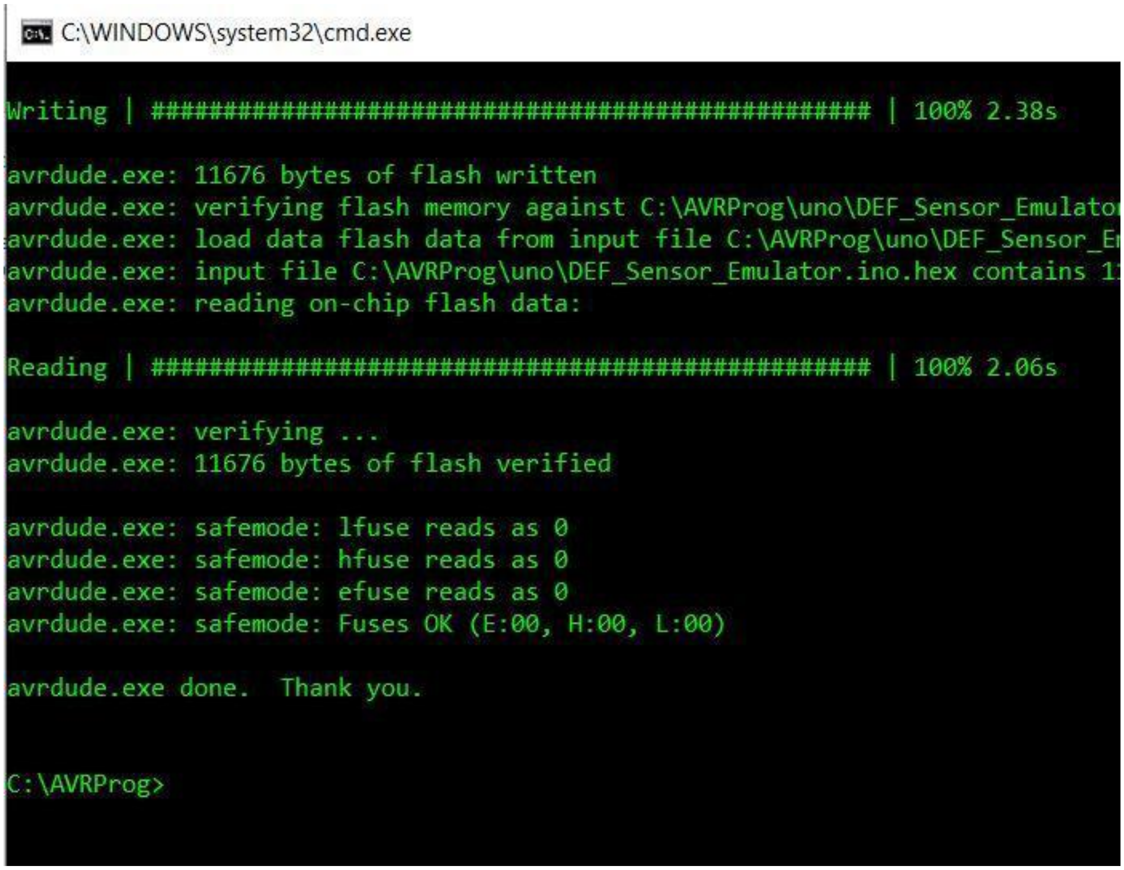
Type the required command in the CMD window and press the “enter” key. A lot of information will scroll by on the CMD window. As an example for a UNO:



C:\WINDOWS\system32\cmd.exe

```
C:\AVRProg>program.bat uno 4

C:\AVRProg>echo off
UNO selected

avrdude.exe: Version 6.3-20190619
             Copyright (c) 2000-2005 Brian Dean, http://www.bdmicro.com/
             Copyright (c) 2007-2014 Joerg Wunsch

             System wide configuration file is "C:\AVRProg/avrdude.conf"

             Using Port                    : COM4
             Using Programmer              : arduino
             Overriding Baud Rate          : 115200
             AVR Part                      : ATmega328P
             Chip Erase delay              : 9000 us
             PAGEL                         : PD7
             BS2                           : PC2
             RESET disposition             : dedicated
             RETRY pulse                   : SCK
             serial program mode           : yes
             parallel program mode         : yes
             Timeout                       : 200
             StabDelay                     : 100
```

And at the completion:



C:\WINDOWS\system32\cmd.exe

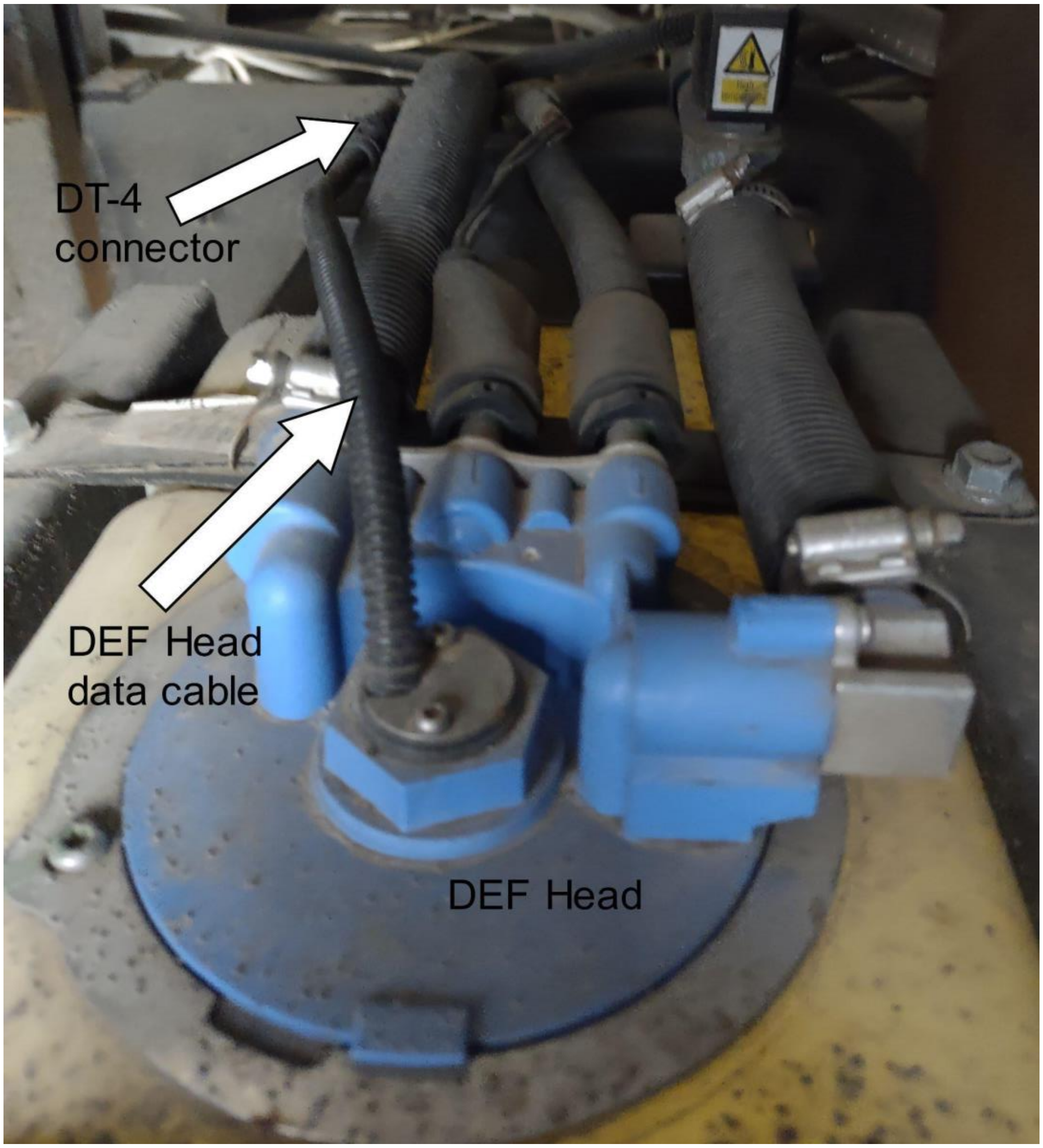
```
Writing | ################################################## | 100% 2.38s

avrdude.exe: 11676 bytes of flash written
avrdude.exe: verifying flash memory against C:\AVRProg\uno\DEF_Sensor_Emulator
avrdude.exe: load data flash data from input file C:\AVRProg\uno\DEF_Sensor_E
avrdude.exe: input file C:\AVRProg\uno\DEF_Sensor_Emulator.ino.hex contains 1
avrdude.exe: reading on-chip flash data:

Reading | ################################################## | 100% 2.06s

avrdude.exe: verifying ...
avrdude.exe: 11676 bytes of flash verified

avrdude.exe: safemode: lfuse reads as 0
avrdude.exe: safemode: hfuse reads as 0
avrdude.exe: safemode: efuse reads as 0
avrdude.exe: safemode: Fuses OK (E:00, H:00, L:00)

avrdude.exe done.  Thank you.


C:\AVRProg>
```

The Arduino board should now be programmed.

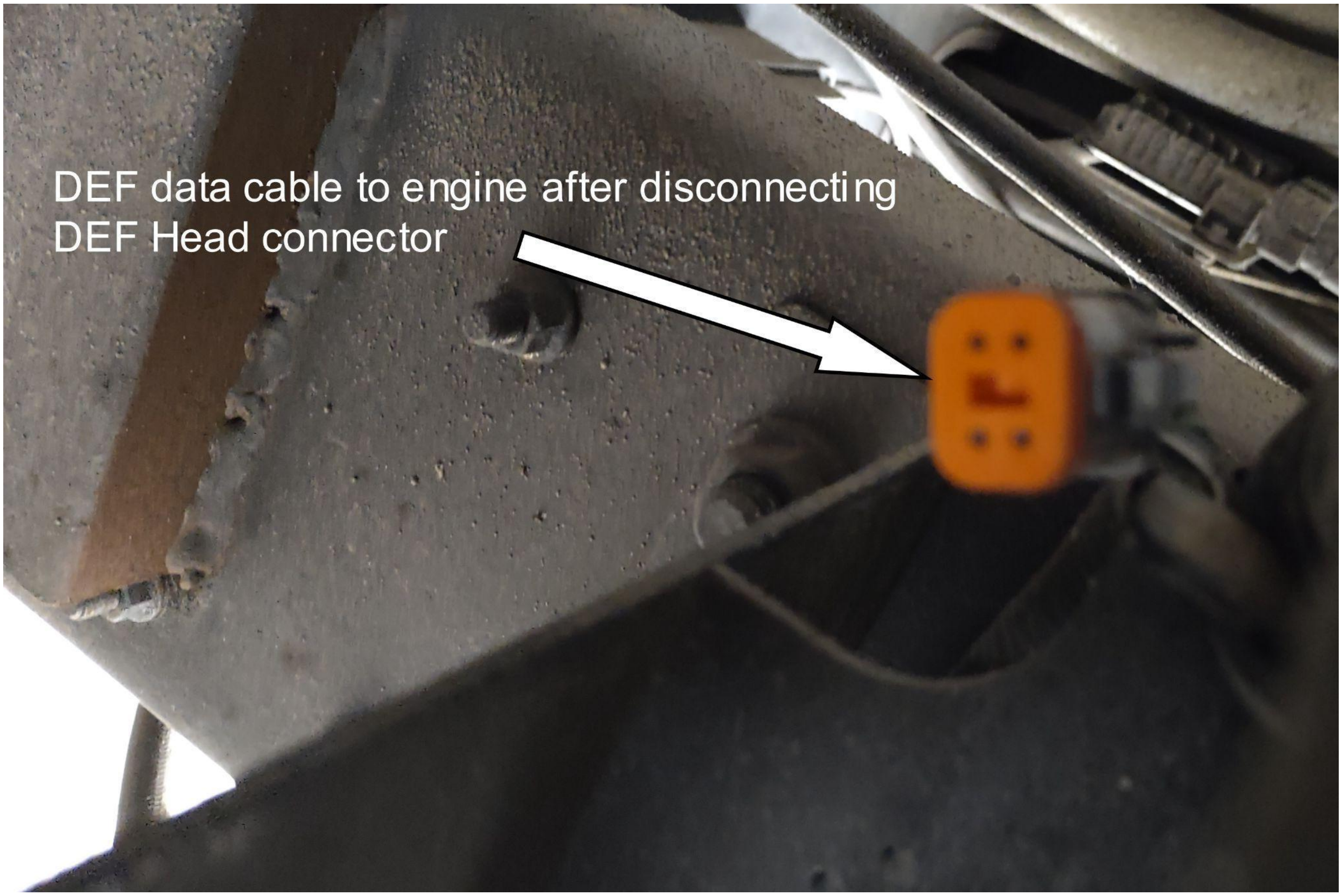
# Connecting to the Vehicle

***MAKE SURE THE IGNITION IS TURNED OFF BEFORE PROCEEDING. YOU SHOULD ONLY CONNECT THE EMULATOR TO THE VEHICLE FOR TESTING OR IN THE EVENT OF A DEF SENSOR FAILURE. YOU SHOULD HAVE EITHER THE OEM SENSOR CONNECTED OR THE EMULATOR CONNECTED BUT NEVER BOTH.***

The DEF Emulator is connected to the Engine ECU in place of the DEF Head. Locate the DEF tank on your coach. The DEF Head is located on top of the tank. There should be a cable running from the DEF Head to a connector attached to the Engine's emissions data bus. You may have to cut a zip tie that is holding the wiring to gain enough slack to be able to access the connector.



Follow the data cable to its connector. There is a release tab on the connector that must be pressed while pulling the connector "apart". The piece you need to press on to release the latch is actually on the connector that is on the wiring harness side, not on the connector attached to the sensor. Do not force it and don't pull on the wires. Grasp the connector bodies and work them apart. With the connector disconnected, you will see the mating connection going to the engine.



This is where you plug the DEF Emulator into. Align the 4-pin plug from the DEF emulator with the 4-socket connector from the wiring harness and make sure that the latching mechanisms on the two connectors are aligned. Firmly press the connectors together. You should feel a slight click when the connectors are properly connected. You can then either mount your emulator box on the top of the DEF tank or possibly in an adjacent compartment if your cable is long enough. Be sure to secure the case.

Several “fault codes” have been associated with the DEF Head failure. The most common codes are:

- 1761
- 3031
- 3364

After installation of the Emulator, the codes need to be cleared. In most newer coaches the codes can be cleared by following this procedure:

- Turn the key to ON, wait for your warning test lights to go out and then start the engine.
- Idle your engine for 5 minutes (minimum).
- Turn engine OFF Leave off for 90 seconds
- Repeat the above 3 steps 2 more times in sequence. (a total of 3 times)

- On the 4th time the codes should be cleared.

This procedure has worked on 2019-2021 coaches with a bad DEF head and the emulator installed. There has been an older 2016 coach where the 3031 code would not clear and had to be cleared with a reset tool such as “Blue Fire” or Nexus.

# Operating With the Emulator Installed

Once the Emulator is installed properly and you have verified that it is operating as intended as evidenced by the original fault codes being reset within the first few trip cycles, there are no special operating requirements. You can operate the vehicle normally, with one exception. ***You are responsible*** for maintaining a proper supply of DEF by manually inspecting the level of DEF in the tank as often as is necessary to ensure that an adequate volume of quality DEF is present in the DEF tank during operation.

# User Responsibilities To Ensure Continued Emissions Control Functionality

Installing this device will NOT interfere with the operation of any factory installed Emission Control System. This is not a “Delete Device” in any way. This device **only** enables the avoidance of unwarranted engine performance derate based on the ECM acting on faulty data from a failed DEF Sensor. ***IT IS THE USER’S RESPONSIBILITY TO MAINTAIN AN ADEQUATE SUPPLY OF QUALITY DEF FLUID IN THE TANK AT ALL TIMES***. The vehicle will continue to consume DEF at a normal rate, however YOU MUST BE AWARE that the instrument panel DEF level gauge will no longer necessarily reflect the true level of DEF in the tank. If you allow the fluid to run out or if you allow anything other than approved, high quality DEF into the tank you WILL encounter several severe Emission Fault warnings and the ECM will force deration to 5 MPH and the vehicle will not recover until repairs are made. You may also cause serious damage to one or more of the emissions control system components.