CAN communication Utilizing OBD-II on a Hybrid Electric Vehicle

Matthew Spencer

11/02/2020

CONTENTS

INTRODUCTION

This lab takes place at the Advanced Power Systems Research Center (APSRC) located in Calumet, Michigan next to the Houghton airport. An electric hybrid vehicle is used in conjunction with CAN Bus software to read and receive CAN messages in real-time from the vehicle's internal CAN Bus network. Students are tasked with sending request messages to receive the real-time vehicle information about its current operation, and then decoding the response using OBD-II PID protocol lookup.

MATERIALS

- Electric Hybrid Vehicle
- Windows 10 Laptop
- CANKing software
- USB-CAN cable

THEORY

Modern vehicles of today are outfitted with a CAN Bus interface for outside devices to connect to the network. Using a Windows 10 laptop installed with the CANKing software, a USB-CAN cable can be used to connect to the vehicle CAN Bus network to send and receive messages between the network and the laptop through the laptop's serial interface. [1]

By sending the properly formatted messages adhering to the OBD-II protocol onto the vehicle's CAN Bus network, specific vehicle information supplied by its sensors should be able to be received and requested by the CANKing software. The data values received should then be able to be properly converted and calculated using a OBD-II PID look-up table to retrieve the correct formulas and equations. [2]

PROCEDURE

The laptop is connected to the vehicle's CAN Bus network through the USB-CAN Bus serial cable. The CANKing software [Figure 1] is then started and configured to send and receive data.

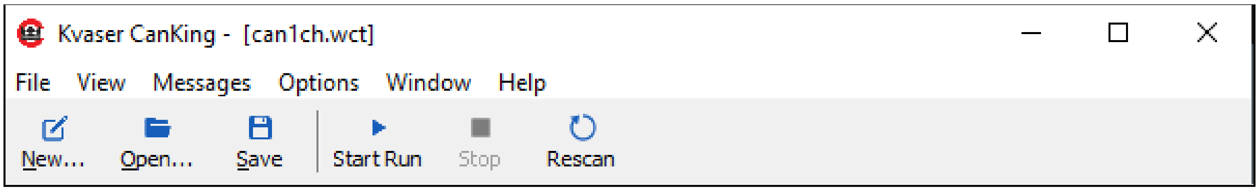
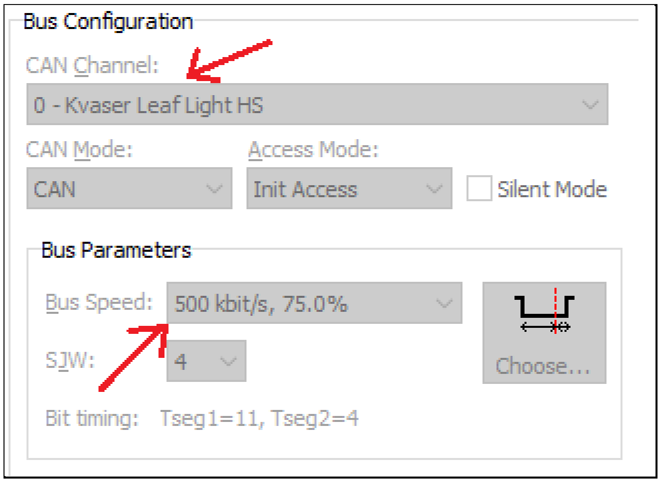


Figure 1: CANKing Software Menu Bar

The Bus rate for CAN 1's bus channel (0-Kvaser Leaf Light HS) is then set to 500kb/s Baud rate to successfully communicate on the bus. [Figure 2]



Bus Configuration

CAN Channel: 0 - Kvaser Leaf Light HS

CAN Mode: CAN

Access Mode: Init Access

□ Silent Mode

Bus Parameters

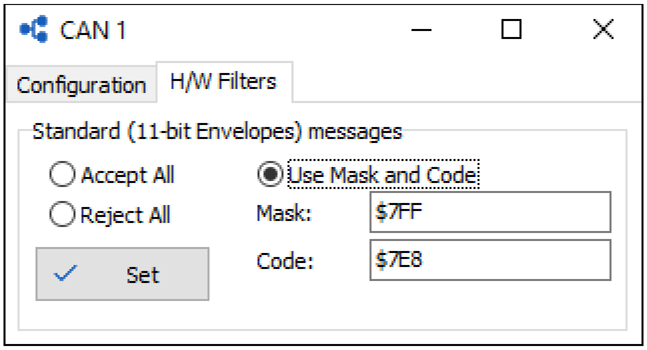
Bus Speed: 500 kbit/s, 75.0%

SJW: 4

Bit timing: Tseg1=11, Tseg2=4

Figure 2: CAN Bus Baud Rate Settings

The next step is to implement the correct mask and filter to correctly listen to just the requested messages sent out on the bus. This setting is changed under the H/W Filters tab. The needed messages have the one ID of 0x7E8, therefore the mask can be set to 0x7FF to allow all messages through to be compared to the filter (code) set to the needed ID 0x7E8. [Figure 3]



Standard (11-bit Envelopes) messages

○ Accept All

○ Reject All

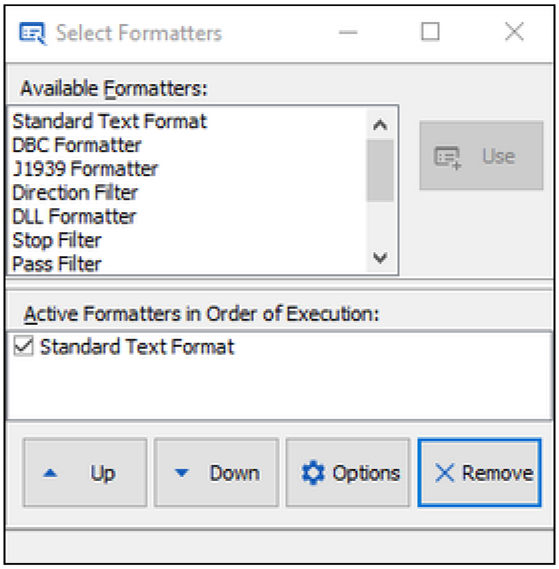
● Use Mask and Code

Mask: $7FF

Code: $7E8

Figure 3: Mask and Filter Settings

After successfully setting the mask and filter to the correct values, the formal is then set to be "Standard Text Format". [Figure 4] Under "Options", choose to display the output in hexadecimal.



Available Formatters:

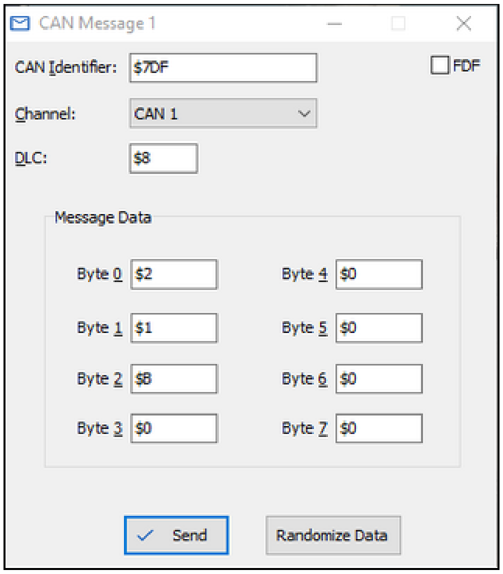
Standard Text Format
DBC Formatter
J1939 Formatter
Direction Filter
DLL Formatter
Stop Filter
Pass Filter

Active Formatters in Order of Execution:

☑ Standard Text Format

Figure 4: Format Settings

Finally, construct the desired messages to transmit onto the bus under the message window. Select which vehicle sensor to request data from by setting the "CAN Identifier" field, choose the channel to transmit on to be "CAN 1", set the data length code (DLC) to "8" to indicate that the message contains a total of 8 bytes, and then fill out the corresponding "Message Data" bytes for the desired vehicle information request. [Figure 5]



CAN Message 1

CAN Identifier: $7DF

□ FDF

Channel: CAN 1

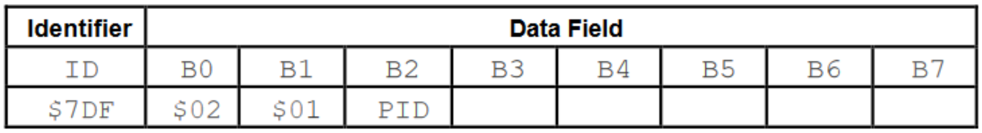
DLC: $8

Message Data

Byte 0	$2	Byte 4	$0
Byte 1	$1	Byte 5	$0
Byte 2	$8	Byte 6	$0
Byte 3	$0	Byte 7	$0

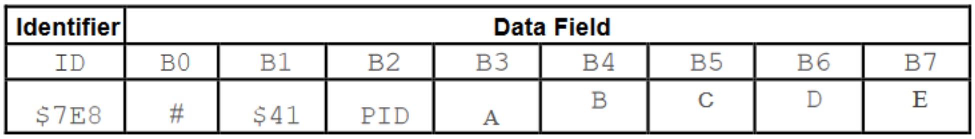
Figure 5: CAN Message Builder

Byte 0 indicates how many bytes are used in the data field, byte 1 indicates which mode is being used (transmitting), and the PID indicates which sensor we are requesting data from. [1] Another view of the CAN Bus transmitting message can be seen in Figure 6, and the receiving message in Figure 7. [1]



Identifier	Data Field							
ID	B0	B1	B2	B3	B4	B5	B6	B7
$7DF	$02	$01	PID					

Figure 6: Request PID Message Format



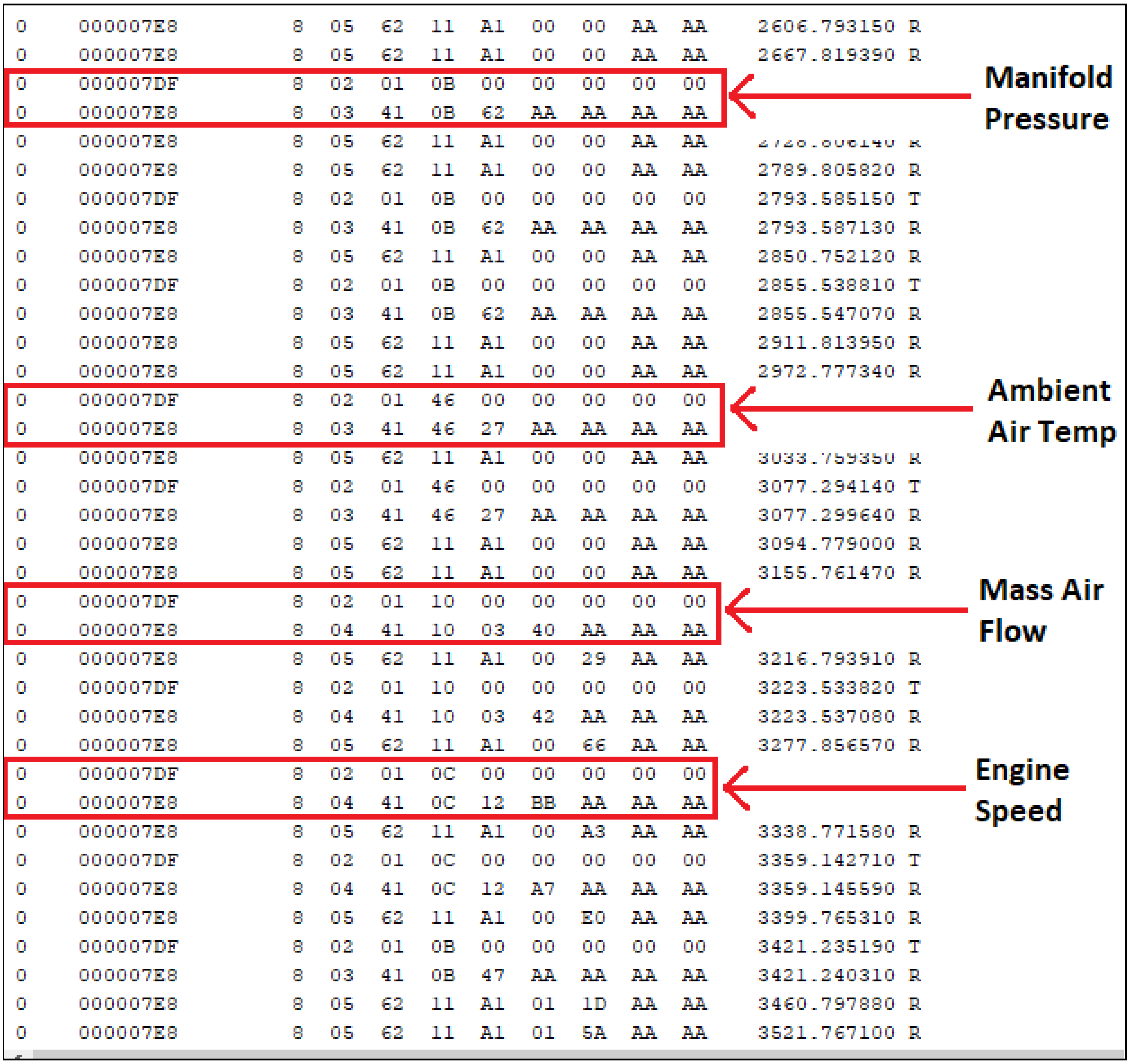
Identifier	Data Field							
ID	B0	B1	B2	B3	B4	B5	B6	B7
$7E8	#	$41	PID	A	B	C	D	E

Figure 7: Response PID Message Format

The message window is configured to request and record data from the vehicle's manifold pressure and ambient air temperature with the vehicle off, and then the mass air flow, and engine speed with it the engine turned on.

<u>RESULTS</u>

The following results can be seen in the output window taken from the CANKing software. [Figure 7] “0x7DF” indicates a sent message, and “0x7E8” indicates a received message. Any column with “AA” indicates that there is no usable data there. [1]



```
0     000007E8          8  05  62  11  A1  00  00  AA  AA   2606.793150 R
0     000007E8          8  05  62  11  A1  00  00  AA  AA   2667.819390 R
0     000007DF          8  02  01  0B  00  00  00  00  00
0     000007E8          8  03  41  0B  62  AA  AA  AA  AA
0     000007E8          8  05  62  11  A1  00  00  AA  AA   2728.806140 R
0     000007E8          8  05  62  11  A1  00  00  AA  AA   2789.805820 R
0     000007DF          8  02  01  0B  00  00  00  00  00   2793.585150 T
0     000007E8          8  03  41  0B  62  AA  AA  AA  AA   2793.587130 R
0     000007E8          8  05  62  11  A1  00  00  AA  AA   2850.752120 R
0     000007DF          8  02  01  0B  00  00  00  00  00   2855.538810 T
0     000007E8          8  03  41  0B  62  AA  AA  AA  AA   2855.547070 R
0     000007E8          8  05  62  11  A1  00  00  AA  AA   2911.813950 R
0     000007E8          8  05  62  11  A1  00  00  AA  AA   2972.777340 R
0     000007DF          8  02  01  46  00  00  00  00  00
0     000007E8          8  03  41  46  27  AA  AA  AA  AA
0     000007E8          8  05  62  11  A1  00  00  AA  AA   3033.759350 R
0     000007DF          8  02  01  46  00  00  00  00  00   3077.294140 T
0     000007E8          8  03  41  46  27  AA  AA  AA  AA   3077.299640 R
0     000007E8          8  05  62  11  A1  00  00  AA  AA   3094.779000 R
0     000007E8          8  05  62  11  A1  00  00  AA  AA   3155.761470 R
0     000007DF          8  02  01  10  00  00  00  00  00
0     000007E8          8  04  41  10  03  40  AA  AA  AA
0     000007E8          8  05  62  11  A1  00  29  AA  AA   3216.793910 R
0     000007DF          8  02  01  10  00  00  00  00  00   3223.533820 T
0     000007E8          8  04  41  10  03  42  AA  AA  AA   3223.537080 R
0     000007E8          8  05  62  11  A1  00  66  AA  AA   3277.856570 R
0     000007DF          8  02  01  0C  00  00  00  00  00
0     000007E8          8  04  41  0C  12  BB  AA  AA  AA
0     000007E8          8  05  62  11  A1  00  A3  AA  AA   3338.771580 R
0     000007DF          8  02  01  0C  00  00  00  00  00   3359.142710 T
0     000007E8          8  04  41  0C  12  A7  AA  AA  AA   3359.145590 R
0     000007E8          8  05  62  11  A1  00  E0  AA  AA   3399.765310 R
0     000007DF          8  02  01  0B  00  00  00  00  00   3421.235190 T
0     000007E8          8  03  41  0B  47  AA  AA  AA  AA   3421.240310 R
0     000007E8          8  05  62  11  A1  01  1D  AA  AA   3460.797880 R
0     000007E8          8  05  62  11  A1  01  5A  AA  AA   3521.767100 R
```

Figure 7: CANKing Software Output Window

Each sensor recorded is accompanied by its own corresponding table representing the sent and received CAN Bus message, [Table 1 - 4] equations [Equation 1 - 4], and therefore calculated to find the sensor's actual result. [Table 5]

Manifold Pressure: Engine OFF [Table 1]

ID	Data Field							
	B0	B1	B2	B3	B4	B5	B6	B7
0x7DF	02	01	0B	00	00	00	00	00
0x7E8	03	41	0B	62	AA	AA	AA	AA

Table 1: Mass Air Flow Data

$$Manifold\ Pressure\ =\ A\text{kpA}$$

Equation 1: Manifold Pressure [2]

Ambient Air Pressure: Engine OFF [Table 2]

ID	Data Field							
	B0	B1	B2	B3	B4	B5	B6	B7
0x7DF	02	01	46	00	00	00	00	00
0x7E8	03	41	46	27	AA	AA	AA	AA

Table 2: Mass Air Flow Data

$$Ambient\ Air\ Temp\ =\ A\ -\ 40\ ^{\circ}\text{C}$$

Equation 2: Ambient Air Temperature [2]

Mass Air Flow: Engine ON [Table 3]

ID	Data Field							
	B0	B1	B2	B3	B4	B5	B6	B7
0x7DF	02	01	10	00	00	00	00	00
0x7E8	03	41	10	03	40	AA	AA	AA

Table 3: Mass Air Flow Data

$$Mass\ Air\ Flow\ =\ \frac{256A+B}{100}\ \text{grams/sec}$$

Equation 3: Mass Air Flow [2]

Engine Speed: Engine ON [Table 4]

ID	Data Field							
	B0	B1	B2	B3	B4	B5	B6	B7
0x7DF	02	01	0C	00	00	00	00	00
0x7E8	03	41	0C	12	BB	AA	AA	AA

Table 4: Mass Air Flow Data

$$Engine\ Speed\ =\ \frac{256A+B}{4}\text{rpm}$$

Equation 4: Engine Speed [2]

Using Equations 1 - 4, the correct values for each received data message in Tables 1 - 4 can be calculated. The letter values received represent the data bytes for the equations, seen in Figure 7.

Table	PID	Equation (HEX)	Result (DEC)
1	0x0B	$A\ =\ 0x62$	98 kPa
2	0x46	$A\ -\ 40\ =\ 0x27\ -\ 4$	-1 °C
3	0x10	$\frac{256A+B}{100}\ =\ \frac{256(0x03)+0x}{100}$	8.32 grams/sec

4	0x0C	$\frac{256A+B}{4} = \frac{256(0x12)+0}{4}$	1198.75 rpm

Table 5: Data Result OBD-II Calculations

CONCLUSION

Using the CANKing software and the USB-CAN cable to connect the software to the vehicle's CAN Bus network, request messages were able to be sent to vehicle sensors to gather real-time data. This data was then able to be converted and read using the equations supplied by the OBD-II PID lookup table.

LAB QUESTIONS

Q1. Nearly every PID requires some sort of scaling and offset to determine its final value. Please, explain why.

> **A1.** Most PIDs require scaling and offsets as a form of security put in place by the vehicle manufacturer to prevent cyber-attacks. The CAN Bus network is available to anybody that has a device and materials similar to this experiment with the CANKing software. Therefore, any critical data that can be manipulated within the vehicle is protected by encoding and decoding it using a predetermined formula. [3]

Q2. In this lab you determined mass flow rate at idle. Using your knowledge of restrictions of emissions and after treatment, calculate the rate at which the vehicle consumes fuel at idle.

> **A2.** Mass Air Flow = 8.32 grams/sec
> Engine Speed = 1198.75/60 = 19.98 rps
> Fuel Consumed = 8.32*19.98= 166.23 grams/s

<u>REFERENCES</u>

1. A. Oliveira, "Lab 7 - CAN communication Utilizing OBD-II on a Hybrid Electric Vehicle", *Michigan Technological University*, Houghton, Michigan, United States, 2019.

2. "OBD-II PIDs", *Wikipedia*, 2020, [Online] Available: https://en.wikipedia.org/wiki/OBD-II_PIDs, [Accessed: Nov 2, 2020].

3. A. Oliveira, "EE5365 In-Vehicle Communications Lectures #16 & 17 - On Board Diagnosis (I and II)", Michigan Technological University, Houghton, Michigan, United States, 2019, [Online] Available: https://mtu.instructure.com/courses/1326161/files/89736150?module_item_id=16414488, [Accessed: Nov 2, 2020].