Progressive Automotive X Prize
CAN bus to CAN bus translator – “Babel Can”

Installation and Configuration

April 25, 2010

REV A

2G ENGINEERING
www.2g-eng.com | info@2g-eng.com

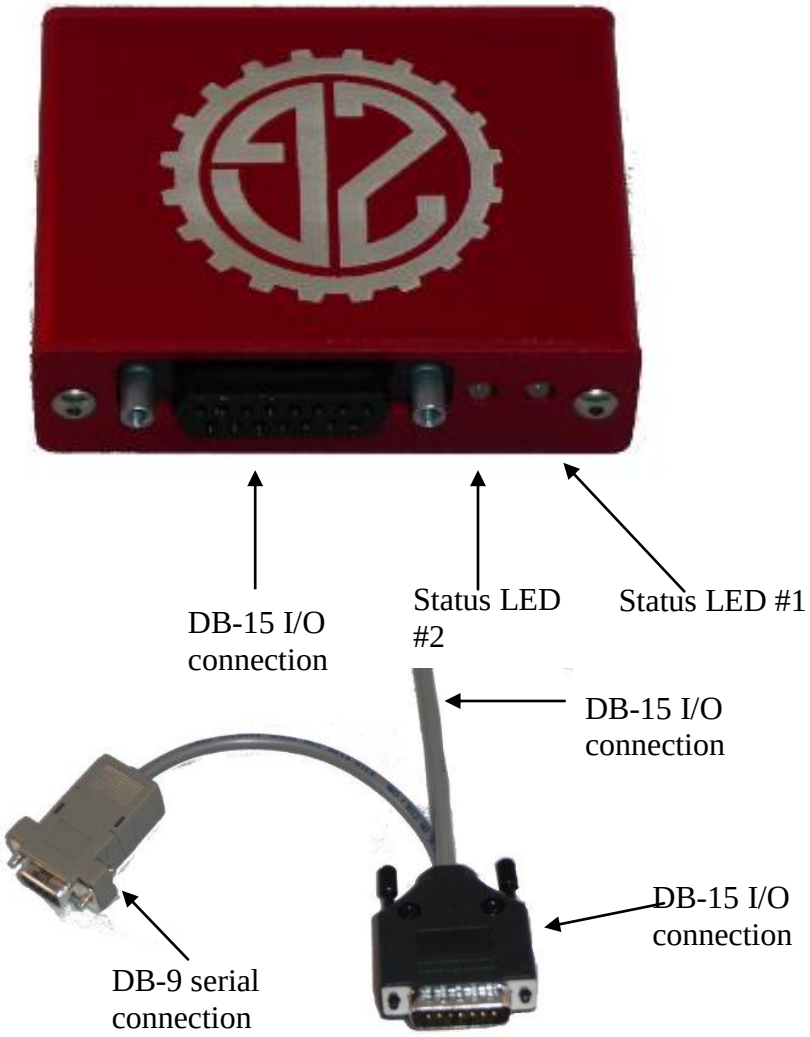
Table of Contents:

Overview

The 2G.133 Babel Can system is designed to help the teams in the PIAXP competition with interfacing there onboard vehicle CAN bus systems with the Morey DAS systems fixed input ranges.

Operational ranges:
Voltage: 5.5V-30V DC
Current: < 80ma
Operating Temperature: -25 C – 85C

Controller Connections:



Connections:

DESCRIPTION: DB-15 I/O connections to Babel CAN unit			
Pin	Description	Color	Length
1	Power in	Red	12
2	GND	Black	12
3	CAN1 LOW	Blue	12
4	CAN1 HI	Brown	12
5	CAN2 LOW	White	12
6	CAN2 HI	Green	12
13	GND	Black (DB-9)	6
14	TX ->OUT	Green (DB-9)	6
15	RX <- IN	Red (DB-9)	6

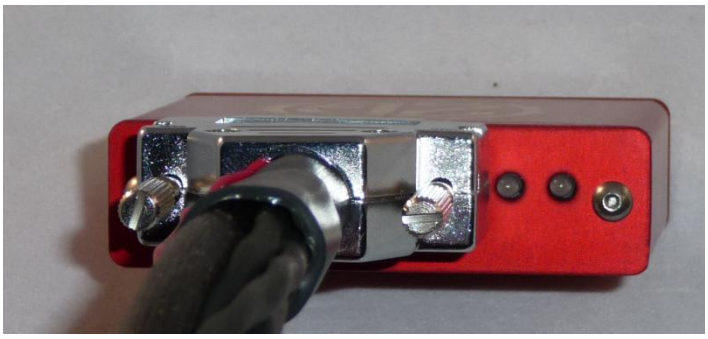
Connections Continued:

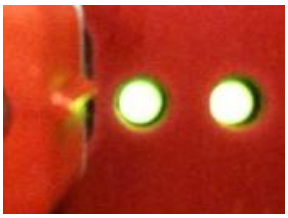
DESCRIPTION: DB-9 serial Connection to PC			
Pin	Description	Color	Length
2	TX ->OUT	Green (DB-15)	6
3	RX <- IN	Red (DB-15)	6
5	GND	Black (DB-15)	6

Status LED’s #1 and #2 on 2G.133 Babel CAN System:

LED #1: Located on right side of unit, has 3 states.
OFF: Unit is not powered.
RED: Unit has not seen any valid traffic on Morey CAN bus interface, CAN bus error. Check termination resistors, and CAN signal wiring. Check other device’s power
GREEN: Unit is correctly connected to CAN bus and is receiving valid traffic on the CAN BUS.

LED #2: Located on left side, LED closest to DB15 connector, has 3 states.
OFF: Unit is not powered.
RED: Unit has not seen any valid traffic on vehicle CAN bus interface, CAN bus error. Check termination resistors, and CAN signal wiring. Check other device’s power.
GREEN: Unit is correctly connected to CAN bus and is receiving valid traffic on the CAN BUS.


Unit is unpowered: Both LED’s are OFF.


Unit is powered on and both LED #2 (LEFT) and #1 (RIGHT) are Green. Unit is operating normally.


Unit is powered on: LED #2 is Red, improper communications on CAN #2
LED#1 is Green, valid CAN connection to the DAS bus (Morey unit).

Unit is powered on: LED #2 is Red, and LED#1 is RED, means bad CAN bus connections on both bus’s

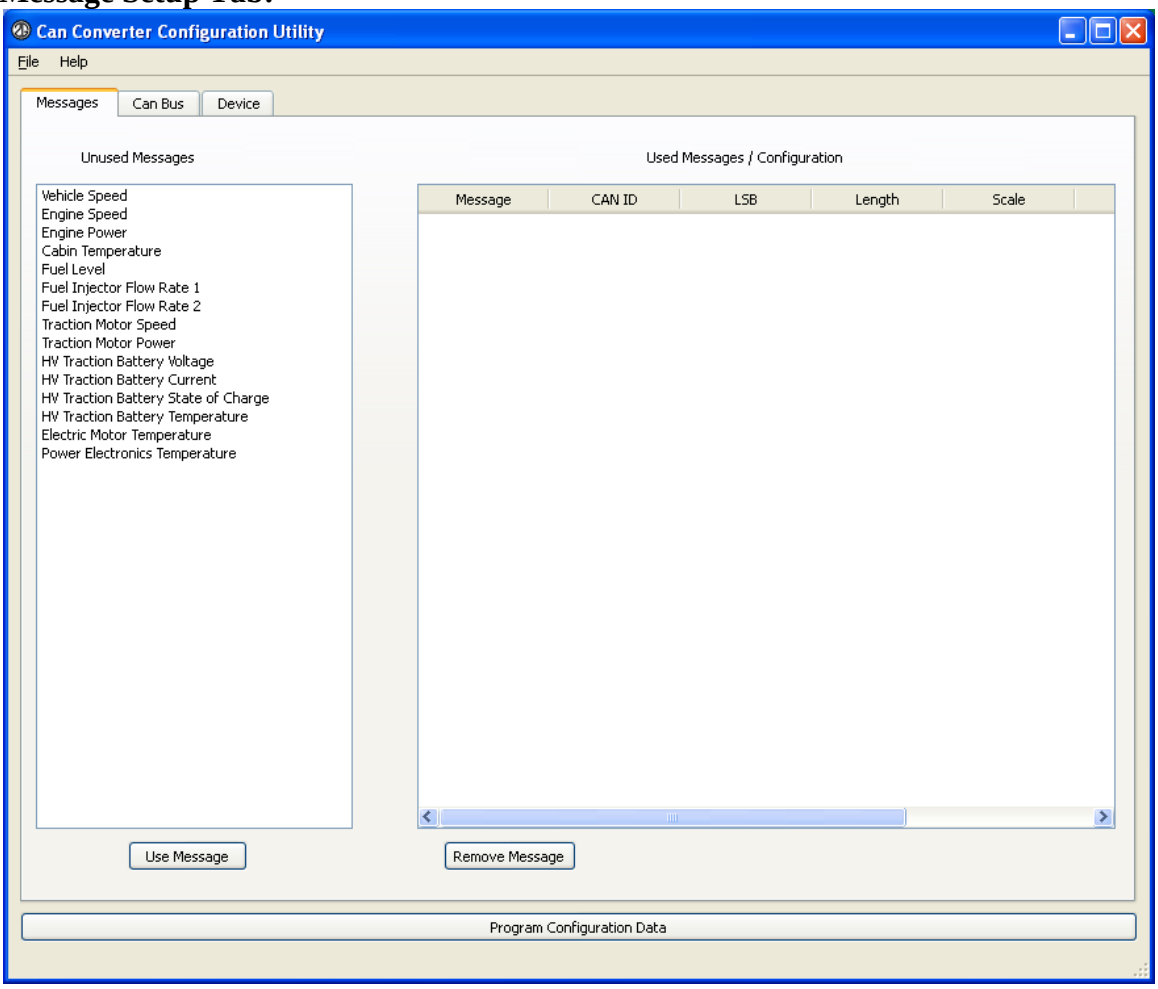
Common Can Bus Errors:

Common CAN bus Errors (LED is red):

- A CAN bus must have two powered devices on it. Check to make sure a second device is connected and powered.
- A properly functioning CAN bus must have two termination resistors placed at opposite ends of the bus. Please refer to the Morey MT-30 users manual for an overview of CAN bus connections and wiring configurations.
- There are two signals that are used for can communication. CAN HI and CAN LOW, make sure they are not reversed. Each device’s CAN hi wire should be connected to the CAN hi wire of the other devices on the bus. The same for CAN low.
- The CAN devices are not set to communicate at the same speed. Check that all the devices that you would like to have communicating on the CAN bus are configured for the same BAUD rate.

Program Overview:

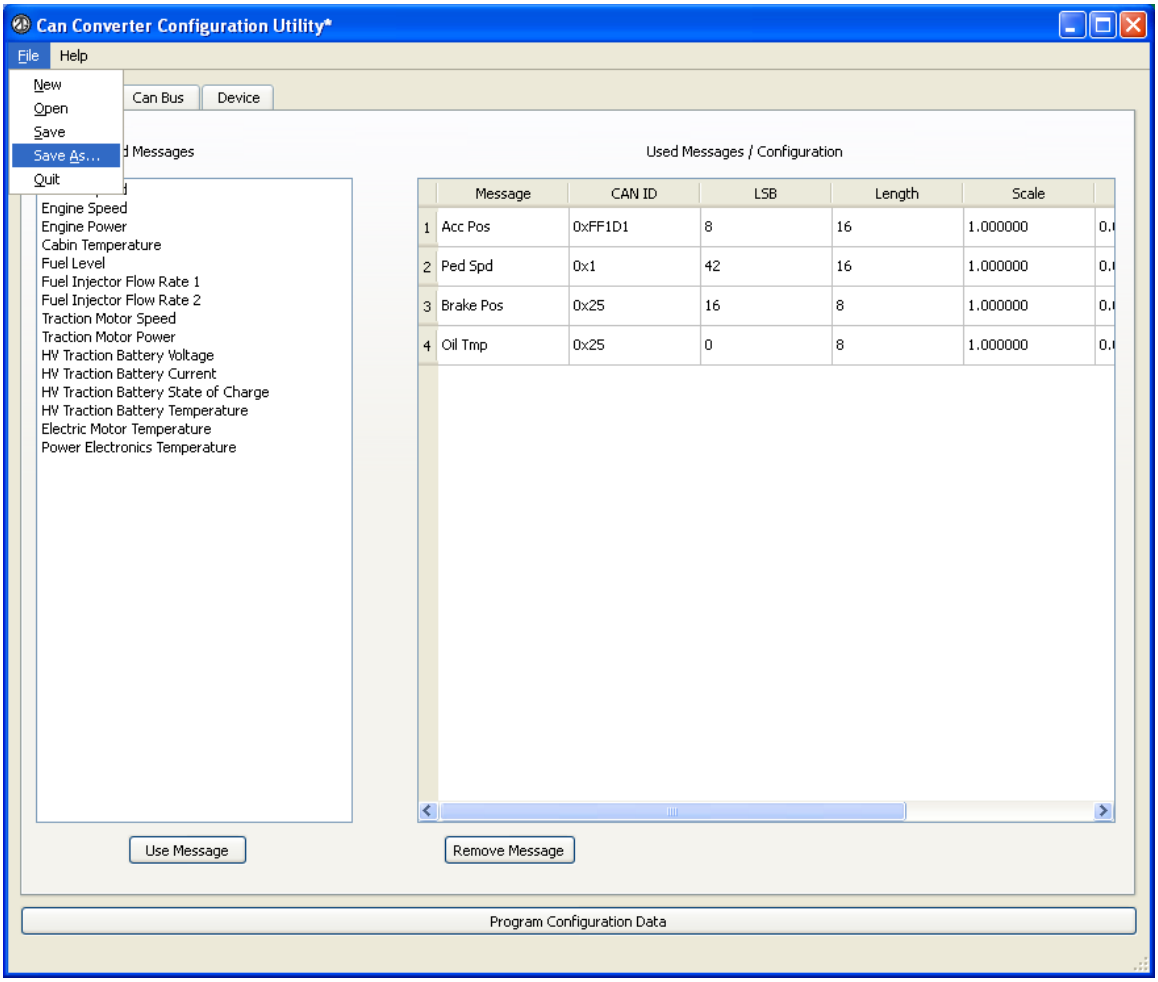
Message Setup Tab:



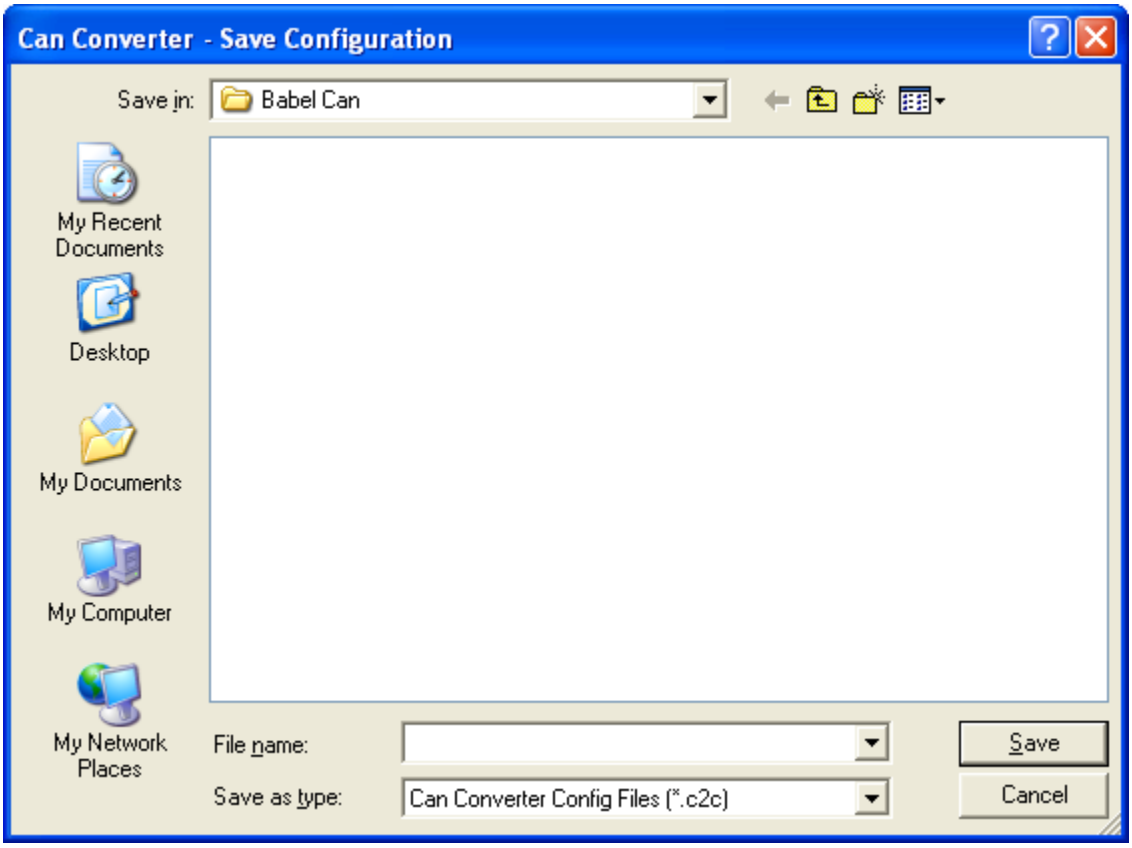
This is the starting TAB of the application. On this page you define all of the messages you want to send to the Morey DAS system.

The list on the left is the messages that are yet to be defined. On the right side of the menu is the configuration parameters for translating the message from one bus to the other.

Saving Your Configuration-example:

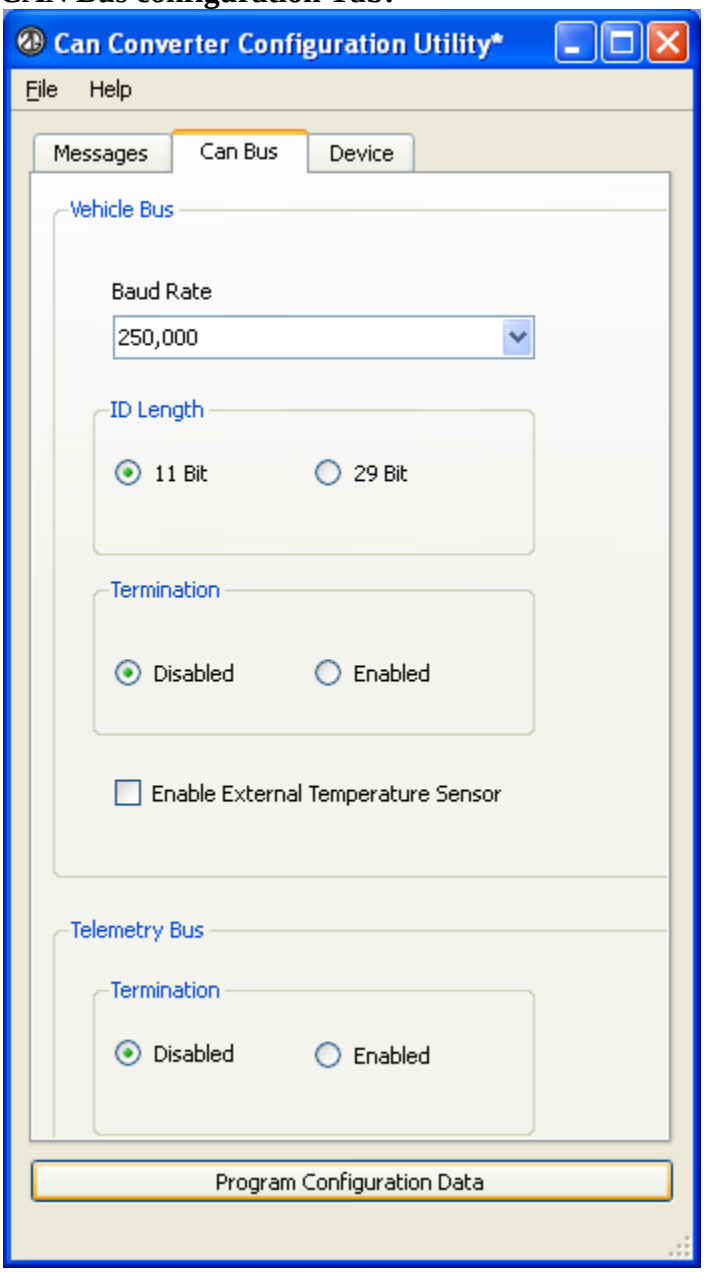


You can save and retrieve your specific vehicle CAN configuration from the file menu. Make sure you keep track of where you save the files and save an updated version as you fine tune your configuration for your application.

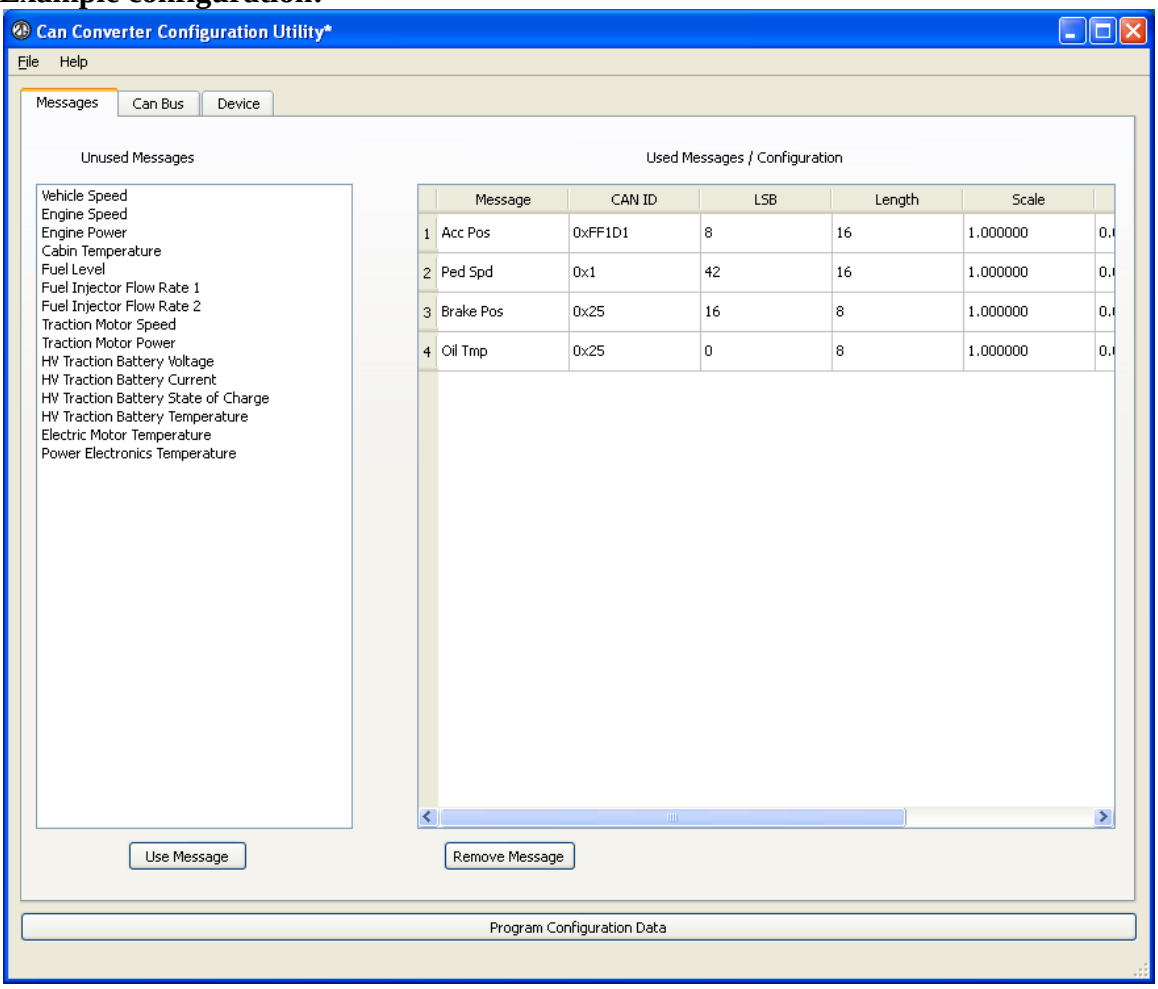


The save menu will default to the program instillation directory.

CAN Bus configuration Tab:



Example configuration:



Let’s do an example: CAN packets are made up of a data section called a ‘Payload’ it can be from 1-8 bytes in length. Below is an example of data sent on a CAN ID of 0xFF1D1.

8 byte payload CAN packet								
Location	1	2	3	4	5	6	7	8
Data	D1	10	1D	3	0	76	DC	28

The data in the message above can be represented by the number (in HEX) 0xD1101D030076DC28.

The Babel Can configuration message is below:

	Message	CAN ID	LSB	Length	Scale	
1	Acc Pos	0xFF1D1	8	16	0.062500	0.

At this point we have isolated the data we want to translate over to the Morey DAS system.

Babel Can Parameters:
The Babel Can configuration program has 2 main parameters that are used for getting the important data from the payloads:

1. **LSB:**

This parameter “least significant bit”, tells the program where in the CAN ‘payloads’ lower boundary is. There are 8 bits in a byte, so in the example bellow we want to shift the data 1 byte to the right. The number **0xD1101D030076DC28** becomes: **0xD1101D030076DC.~~28~~** (Note the missing lowest byte).

0xD1101D030076DC28 -> 0xD1 (Least significant bit is bit 56)
0xD1101D030076DC28 -> 0xD110 (Least significant bit is bit 48)
0xD1101D030076DC28 -> 0xD1101D (Least significant bit is bit 40)
0xD1101D030076DC28 -> 0xD1101D03(Least significant bit is bit 32)
0xD1101D030076DC28 -> 0xD1101D0300(Least significant bit is bit 24)
0xD1101D030076DC28 -> 0xD1101D030076 (Least significant bit is bit 16)
0xD1101D030076DC28 -> 0xD1101D030076DC (Least significant bit is bit 8)
0xD1101D030076DC28 -> 0xD1101D030076DC28 (Least significant bit is bit 0)

Example: Given a packet with a 6 byte payload of: 0x1234567890AB. **What will the value be of we use an LSB of 24**
(LSB 0) 0x1234567890AB -> 0x1234567890AB
(LSB 8) 0x1234567890AB -> 0x1234567890
(LSB 16) 0x1234567890AB -> 0x12345678
(LSB 24)0x1234567890AB -> 0x123456
(LSB 32) 0x1234567890AB -> 0x1234
(LSB 40) 0x1234567890AB -> 0x12

2. **Length**:

The length field tells how many bit’s of information we want to get from the ‘payload’ that we shifted with the LSB parameter in part #1. In this case we want 16 bits (2 bytes). Now we take the lowest 16 bits of information from the data remaining and we get
0x76DC (example 1)
0x3456 (example 2)

Other Babel Can Parameters:
The next important parameter to look at is **Scale.**

Scale:
This entry is used to convert the accelerator pedal position 0%-100% represented by a 16 bit number (0-0xFFFF hex, or 0-65535 decimal) to a 12 bit number as defined in the Morey DAS system manual. The Das system uses 12 bit numbers to represent the data--0-0x0FFF hex or 0-4095 decimal.

The ratio between these two number scales is 4096/65536 = .0635 so we enter that into the ‘scale factor’ so when the CAN message says throttle at 50% it will output a message

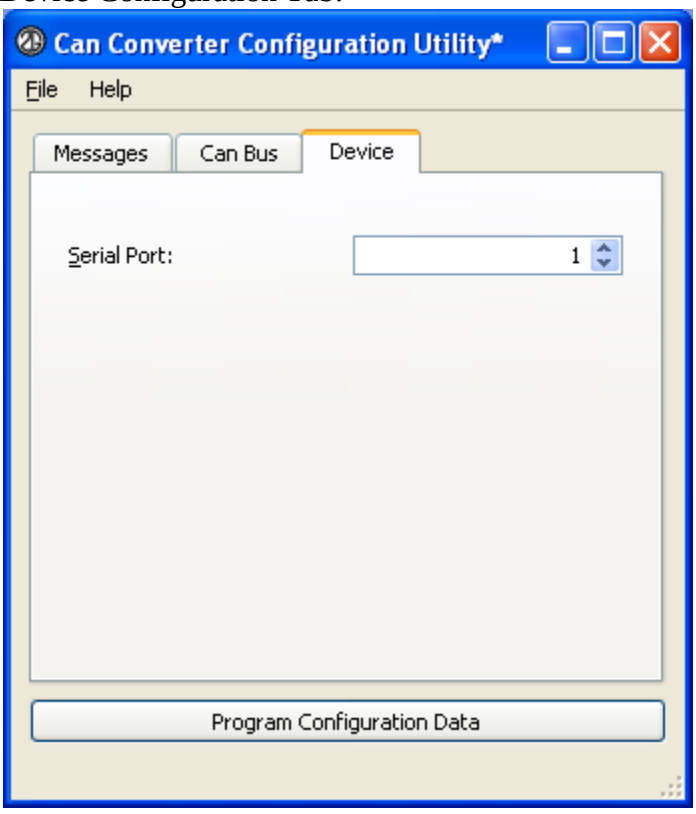
containing 0x07FF for throttle position. Since this number is the wrong scale for the Morey unit, we need to scale it by multiplying it by .0625 to get 0x07F.

Byte Order:
The Byte Order parameter is used when the systems communicate across deferent architecture types. In many case’s leaving it set to 8 bit word, little endian will work. (Little endian - Intel format, vs Big endian - Motorola Format)

If you are dealing with 16 bit or 32 bit numbers and they are of the little endian type you need to set that in the Byte Order. If the input data is of the big endian format, you can leave it set on 8 bit word, little endian.

The big endian configuration is a for a special case in this software release and should not be used unless you are dealing with 32 bit numbers and the 8 bit configuration does not work.

Device Configuration Tab:



The Babel can module connects through a serial port on a PC. If there is no serial port on your computer you will need to get a SB to serial adaptor to connect with the Babel can model. You may need to look in device manager to make sure you have the correct communications port selected.

Contact information:

2G Engineering LLC
2870 Terra Court Suit 3
Sun Prairie, WI 53590
608-628-8941
hglenn@2g-eng.com

Revision history:

Rev - : Initial Release
Rev A: Added Byte order comments. Added another example case for the LSB configuration. 4-27-10 (HAL)