



6. 所解析的信号将会被显示出来, 如图 9-6 所示。

1199	246 μ s	ECU Message H5 CAN 1	102	8	00213000000000	H5 CAN
Signal 1					0.000 V [0]	
Signal 2					13.482 V [213]	
Signal 3					0.000 V [0]	
Signal 4					0.000 V [0]	
1200	249.765 ms	ECU Message H5 CAN 2	101	8	00000000000000	H5 CAN
Signal 5					0.000 V [0]	
Signal 6					0.000 V [0]	
Signal 7					0.000 V [0]	
Signal 8					0.000 V [0]	
1201	248 μ s	ECU Message H5 CAN 1	102	8	00210000000000	H5 CAN
Signal 1					0.000 V [0]	
Signal 2					13.406 V [210]	
Signal 3					0.000 V [0]	

图 9-6 报文视图中解析的信号

9.4 neoMOTE 使用

概述

该应用说明涵盖了如何设置和使用 neoMOTE 在它的屏幕上显示 J1979 数据, 同时也包括如何使用设备的按钮和蜂鸣器功能。

使用 Vehicle Spy 部分

- 报文编辑器
- Function Block
- CoreMini 控制台

需要的硬件

neoVI MOTE, ValueCAN 或者 neoVI

本示例中的文件

neoMOTEapp.zip

9.4.1 neoMOTE 使用 -第 1 部分: 报文

1. 打开 Vehicle Spy

第一步和其它 Vehicle Spy 工程一样, 是打开 Vehicle Spy 并登录。

2. 创建 Rx 报文

示例文件中为 J1979 诊断请求定义了 4 条 Rx 报文。您可创建这些报文或者如果您想查看不同的信号, 可以创建其它报文。

3. 创建脚本型 Function Block

第一个脚本将为 neoVI MOTE 中的按钮进行编程, 如图 9-7 所示。



1	If	{Switch 1 (Value) :neo0-sw0-0-index(0)}=1	// if button 1 is pushed
2	Function Block Action	Start RPM	// start rpm function block
3	Function Block Action	Stop Fuel Pressure	// stops fuel pressure function block if running
4	Function Block Action	Stop Throttle Position	// stops throttle position function block if running
5	Else If	{Switch 2 (Value) :neo0-sw1-0-index(0)}=1	// if button 2 is pushed
6	Function Block Action	Start Fuel Pressure	// starts fuel pressure function block
7	Function Block Action	Stop Throttle Position	// stops throttle position function block
8	Function Block Action	Stop RPM	// stops rpm function block
9	Else If	{Switch 3 (Value) :neo0-sw2-0-index(0)}=1	// if button 3 is pushed
10	Function Block Action	Start Throttle Position	// starts throttle position function block
11	Function Block Action	Stop Fuel Pressure	// stops fuel pressure function block
12	Function Block Action	Stop RPM	// stops rpm function block
13	End If		// ends if statements
14			

图 9-7 neoVI MOTE 中按钮进行编程

9.4.2 neoMOTE 使用 -第 2 部分：在屏幕上显示

1. 更多的脚本型 Function Block

在本步骤中，我们将指定一个信号到按钮 1, 2 和 3 上面。按钮 4 作为一个 Home 按钮，停止任何执行的 Function Block 并且在屏幕上显示“neoMOTE”。为了实现这个，为每个按钮创建新的 Function Block，分别为：RPM，Fuel Pressure，Throttle Position 和 Home。

2. 在 neoVI MOTE 屏幕上显示

这个利用下面显示的 Log Data 命令来完成。利用 fx 按钮，如图 9-8 所示，选择信号或者只需输入文本。确定勾选了“Evaluate as text”项，这将告知软件该表达式是文本类型而非等式类型，如图 9-9 所示。

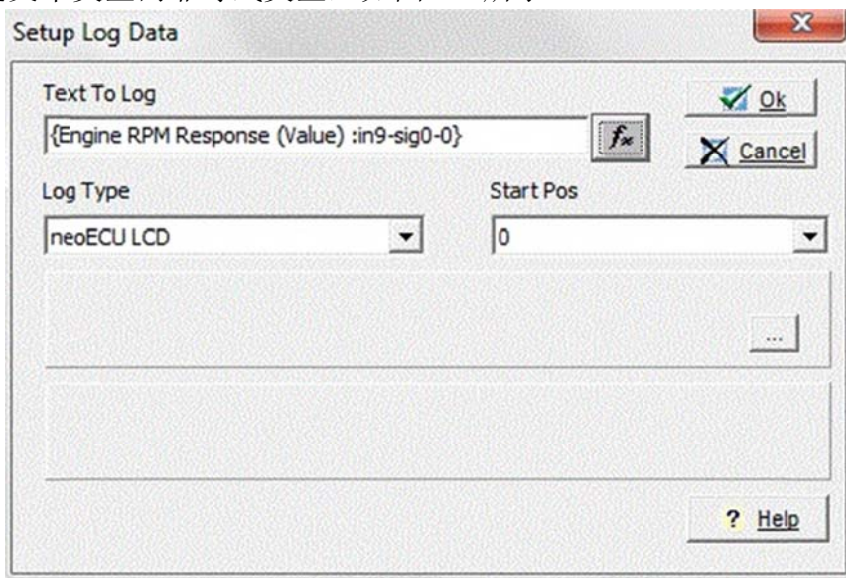


图 9-8 Setup Log Data 界面

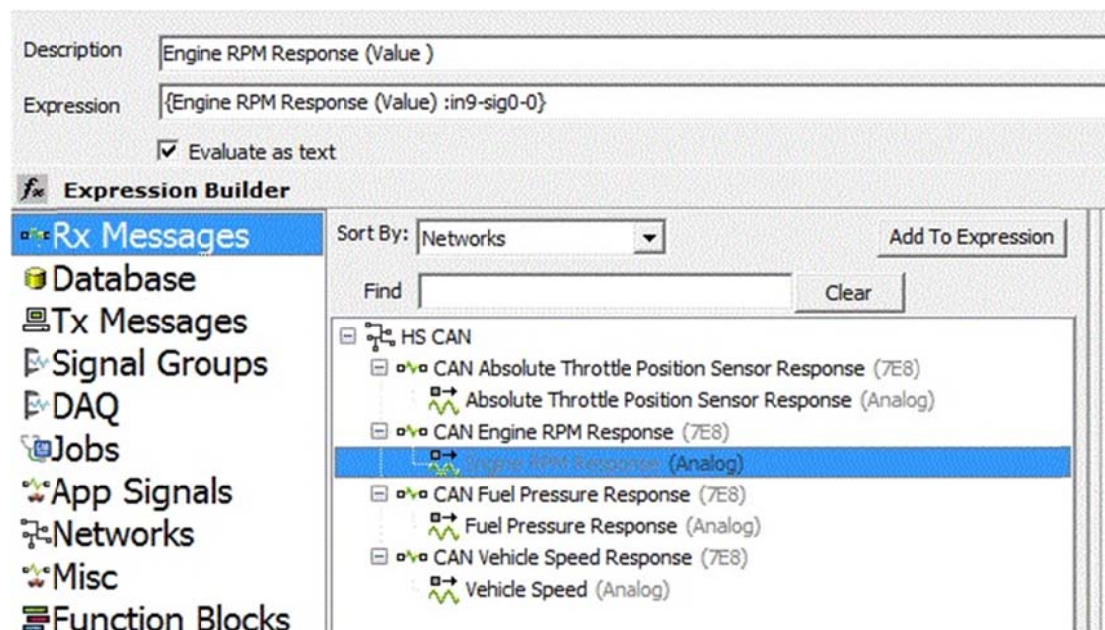


图 9-9 设置表达式

3. 创建 Function Block

下面图 9-10 中，分别显示了 RPM、Fuel Pressure 和 Throttle Position 三个 Function Block，都是用了 Log Data 命令。

Step	Description	Value	Comment
1	Log Data	LCD: {Engine RPM Response (Value) :in9-sig0-0}	// displays engine RPMs
2			

Step	Description	Value	Comment
1	Log Data	LCD: {Fuel Pressure Response (Value) :in7-sig0-0}	// displays Fuel Pressure
2			

Step	Description	Value	Comment
1	Log Data	LCD: {Absolute Throttle Position Sensor Response (Value) :in14-sig0-0}	// displays Throttle position
2			

图 9-10 RPM、Fuel Pressure 和 Throttle Position 三个 Function Block

9.4.3 neoMOTE 使用 -第 3 部分: 使用蜂鸣器

1. Function Block 脚本控制蜂鸣器

在 Function Block 中，我们使用“if”语句等待车辆速度值达到 100 km/h，然后设置蜂鸣器发出频率为 500Hz 的声音，然后使用“Else If”语句在速度低于 100 km/h 时再关掉蜂鸣器。如图 9-11 所示。

Step	Description	Value	Comment
1	If	{Vehicle Speed (Value) :in10-sig0-0}>100	// waits for vehicle speed to be >100 kmph
2	Set Value	{PWM Output 1 (PWM Frequency (Hz)) :neo0-po0-1-index(0)} = 500	// beeps when speed is above 100
3	Else If	{Engine RPM Response (Value) :in9-sig0-0}<100	
4	Set Value	{PWM Output 1 (PWM Frequency (Hz)) :neo0-po0-1-index(0)} = 0	// no beep when speed is less than 100
5	End If		

图 9-11 控制蜂鸣器的 Function Block

为了设置蜂鸣器，在表达式生成器中，进入“Physical IO”，然后定位到“PWM Output 1”，它是 neoVI MOTE 的蜂鸣器输出。为了设置该参数，确保您选择了底



部窗口中的“PWM Frequency (Hz)”属性，如图 9-12 所示。

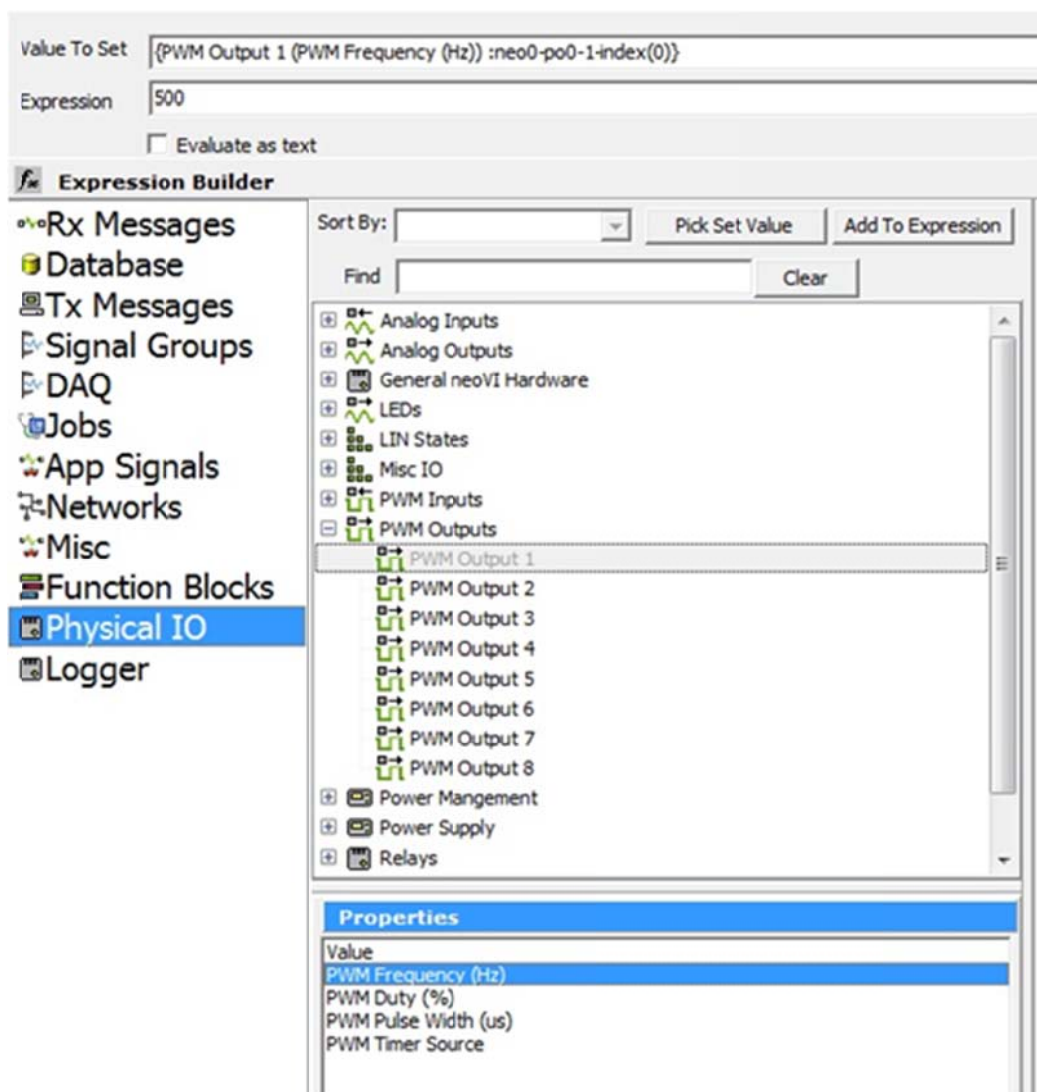


图 9-12 设置 neoVI MOTE 的蜂鸣器输出频率

9.4.4 neoMOTE 使用 -第 4 部分：Home 屏幕及结论

1. Home 屏幕的脚本型 Function Block

对于最后一个按钮的功能，我们设置一个 Home 屏幕，将会关闭所有正在运行的 Function Block 并且在屏幕上显示“neoMOTE”。脚本如图 9-13 所示。

Step	Description	Value	Comment
1	If	{Switch 4 (Value) :neo0-sw3-0-index(0)}=1	// if button 4 is pushed
2	Function Block Action	Stop RPM	// the following 3 steps stop all running function blocks
3	Function Block Action	Stop Fuel Pressure	
4	Function Block Action	Stop Throttle Position	
5	Log Data	LCD: neoMOTE	// displays neoMOTE on screen
6	End If		

图 9-13 Home 屏幕的 Function Block 脚本

2. 最后一步-加载 CoreMini

准备加载 CoreMini 至 neoVI MOTE 中。



结论: 这里总结了有关 neoVI MOTE 简单应用的应用说明。现在可用您的 neoVI MOTE 来读取 J1979 信号并显示它了。

9.5 VehicleScape 单机记录

概述

本应用说明描述了如何使用 VehicleScape DAQ 的单机记录功能, 自动地创建需要的 Function Block。本说明将复习如何使用这些设置工具, 以及如何上传 Function Block 至记录仪 (Logger) 或 neoVIMOTE (如需要) 中。

使用 Vehicle Spy 部分

-VehicleScape

需要的硬件

neoVI FIRE, neoVI RED, neoVI Yellow

本示例中的文件

neoVIMOTE Default CoreMini.vs3 (included in the Vehicle Spy Install)

9.5.1 单机记录-第 1 部分: 报文采集

1. 打开 Vehicle Spy

第一步和其它 Vehicle Spy 工程一样, 是打开 Vehicle Spy 并登录。

2. 打开 VehicleScape

VehicleScape 位于 Measurement 下面。这里应用说明忽略了平台和信号选择步骤。

3. 打开设置

从 VehicleScape 中选择设置标签。在这些标签中, 选择“Standalone Logging”标签。您将使用该视图完成在本教程中大部分内容。

4. 选择记录的内容

您可选择两者中其一: 记录所有的总线通信报文 (如图 9-14 中①所示) 或者只记录您选择的报文 (如图 9-14 中②所示)。如果您选择只记录选择的报文, 您将需要在“Channels”标签中设置这些报文。勾选“Enable Decimation”选项 (如图 9-14 中③所示) 将强制记录设备以某一速率只记录最近接收到的报文。该选项将以降低数据分辨率而增加记录的时间。通过对收到的数据以设置的速率进行采样, 而不是记录每一条收到的数据。