2.8 教程 5：请求并保存故障码（DTC）

概述

本教程将会显示如何利用 Function Block 和 Graphical Panel 使用诊断（Diagnostic）任务服务。结果使用图形面板上的按钮，来请求诊断故障码并将其保存到一个文件。本示例的副本“DTCRequestAndSave”可在 Vehicle Spy 登录屏幕中的“Examples”标签中找到。

为了实现此目标，教程 5 将会引领您学习如下内容：

1. 创建一条 DTC 请求服务；
2. 创建一个 Function Block 请求信息并保存数据；
3. 为终端用户做一个图形面板；
4. 查询并查看数据记录。

2.8.1 第 1 部分-创建诊断任务

第 1 部分将以创建一个诊断任务，请求故障码开始。

1. 登录 Vehicle Spy

打开 Vehicle Spy 并选择登录名后，使用主菜单依次选择“File”->“New”。在本教程中，用户名比之前的要重要，因为采集的数据将会保存在选择的用户数据目录之下。

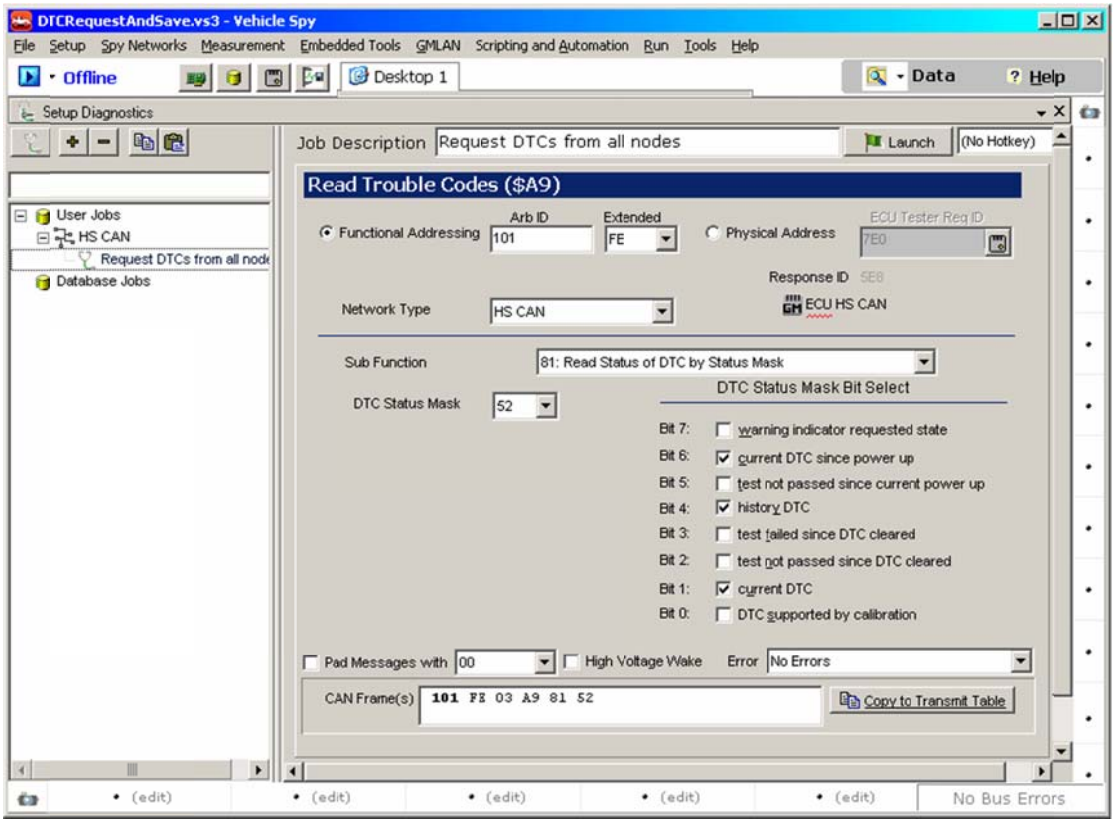

图 2-54 设置诊断任务在 HS CAN 上请求故障码

2. 创建请求 DTC 任务

下一步就是创建诊断任务读取 DTC。选择“Vehicle Networks”->“Diagnostics

Setup”打开设置诊断窗口。使用“+”按钮添加一条 GMLAN $A9 的验证诊断代码任务。（注意：本示例可适用于任何诊断服务。）

3. 确认设置并编辑描述

在本示例中，默认设置就可以了。该诊断任务将在采用 GMLAN 协议车辆上向所有高速 CAN 网络上运行的电控单元（ECU）发送 DTC 请求。更改任务描述为“Request DTCs from all nodes”。设置应如图 2-54 所示。

2.8.2 第 2 部分-创建 Function Block 发送请求并保存数据

1. 打开 Function Block 视图

下一步是创建一个 Function Block 控制诊断服务并保存数据。通过“Scripting and Automation”->“Function Blocks”打开 Function Block 窗口。

2. 创建一个 Function Block

单击“+”按钮并选择“Script”。创建一个新的脚本型 Function Block。更改 Function Block 的描述为“Check Codes and Log Them to a File”类似的描述。

3. 写脚本

现在需要输入脚本。本任务需要 4 步。所需要的步骤以及描述如下表 2-1 所示。对于“Diag Job Action”，双击“Value”单元选择诊断任务启动以及如何启动。为了有关该命令的更多信息，请见“Diag Job Action Command”帮助文档。

表 2-1 任务脚本

Step	Command	Value	Reason for Step
1	Diag Job Action	Start Request DTCs from all nodes	Starts, or launches, the diagnostic service.
2	Diag Job Action	Wait until Complete Request DTCs from all nodes	Waits at this step until the selected diagnostic service is finished. When completed, continues to the next step.
3	Diag Job Action	Save Request DTCs from all nodes	Saves the received data and message traffic to a log file.
4	Stop	n/a	Ends this function block.

4. 配置启动设置

在“Start”标签，选择“Manual Start”，使用本教程下一部分设置的图形面板启动 Function Block。

2.8.3 第 3 部分-制作图形面板

让我们通过创建一个图形面板简化用户发送请求故障码的操作。

1. 打开图形面板

在“Measurement”->“Graphical Panels”下面打开图形面板编辑器。这是您发挥创造性的机会，制作图形面板没有什么正确或错误的方式。

2. 创建面板

本面板中，我们将从屏幕底部选择 3 个控件进行设置：

第 1 个是 Function Block Button 启动 Function Block。需要设置的按钮关键属性如下：

FunctionBlock：Check Codes and Log Them to a file

FBlockAction：0-Start

每次按钮按下之后，将会启动 Function Block。

下一步我们添加一个 LED，让用户了解故障码请求正在进行中。绘制一个 LED 并设置如下属性：

Signal：在“**Jobs**”下创建的诊断任务，并选择“**Is Running**”属性。

最后，使用一个 Text Display，利用“**Caption**”属性添加一些按钮作用的描述。图 2-55 显示了完成后的样子。

图 2-55 HS CAN 总线上向所有电控单元发送请求故障码的图形面板完成示例

2.8.4 第 4 部分-请求、保存并查看故障码（DTC）

1. 启动面板

为了使用本示例，将附带的硬件连接到车辆或者带有高速 CAN 网络的 ECU 并启动 Vehicle Spy。单击图形面板上的按钮启动创建的 Function Block，LED 将会改变颜色。当它变暗后，采集的数据就被保存至文件中。

2. 查看保存的数据

数据采集完成之后，它在哪里呢？单击右上角的“Data”按钮，窗口浏览器打开显示数据目录，该目录是针对在第 1 部分中使用的登录名创建的。数据目录中保存的文件名字由诊断服务、日期和时间构成。例如，“Request DTCs from all nodes 9-04-2009 3-18-00 am.csv”。Excel 或者其它的电子表格编辑器可以打开这类文件。

2.8.5 第 5 部分-结论

本示例演示了如何联合使用 Vehicle Spy 不同的部件，完成自动化任务。利用 Function Block，创建简答的脚本就可以完成通常的任务。图形面板可以用于控制 Function Block 以及显示状态或者采集的数据。Function Block 和图形面板可设计成您所需要的复杂化或者简单化，关键是它们能够完成需要的任务。

2.9 教程 6：DPS 自动化编程

概述

本教程的目的是演示如何利用 Function Block 和图形面板（Graphical Panel）的诊断功能，使用 GM 的 DPS 应用程序来对 ECU 进行编程。最终的结果是在一个图形面板上有一个按钮，该按钮将一个接一个地烧写 2 个档案文件。本教程采用如同教程 5 一样的基本步骤，唯一不同的是使用的诊断服务。本示例的副本“DPS Automation Example”可在 Vehicle Spy 登录屏幕中的“Examples”标签中找到。

为了实现此目标，教程 6 将会引领您学习如下内容：

1. 创建 2 条 DPS 编程任务；
2. 创建 1 个 Function Block 执行该服务；
3. 为终端用户做一个图形面板；
4. 使用创建的控件。

2.9.1 第 1 部分-创建 DPS 编程任务

1. 登录 Vehicle Spy

打开 Vehicle Spy 并选择一个登录名之后，利用主菜单栏依次选择“File”->“New”。

2. 创建 1 个 DPS 诊断任务

通过主菜单栏依次选择“Vehicle Networks”->“Diagnostics Setup”打开设置诊断窗口。使用“+”按钮添加一条 GMLAN DPS 编程任务。

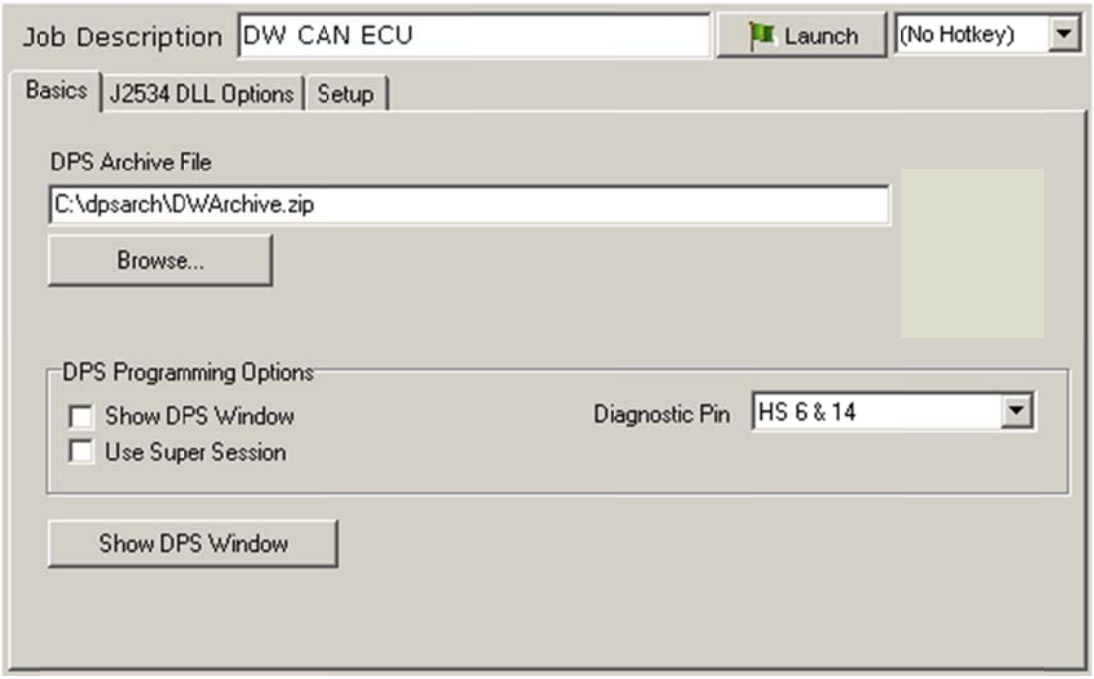

图 2-56 DPS 编程“Basic”标签

3. 编辑描述与基本设置

首先更改任务描述。在本教程中，将设置多个 DPS 编程任务。如果不更改默

认的描述，将很难区分它们。剩下的重要设置是在“Basic”标签中，如图 2-56 所示，确认要编程的 DPS 档案文件以及所在的诊断 PIN 管脚。

4. 确认 J2534 动态链接库选项

在“J2534 DLL Option”标签中，如图 2-57 所示，配置 J2534 通信的一些选项。正常情况下，默认的就可以了。

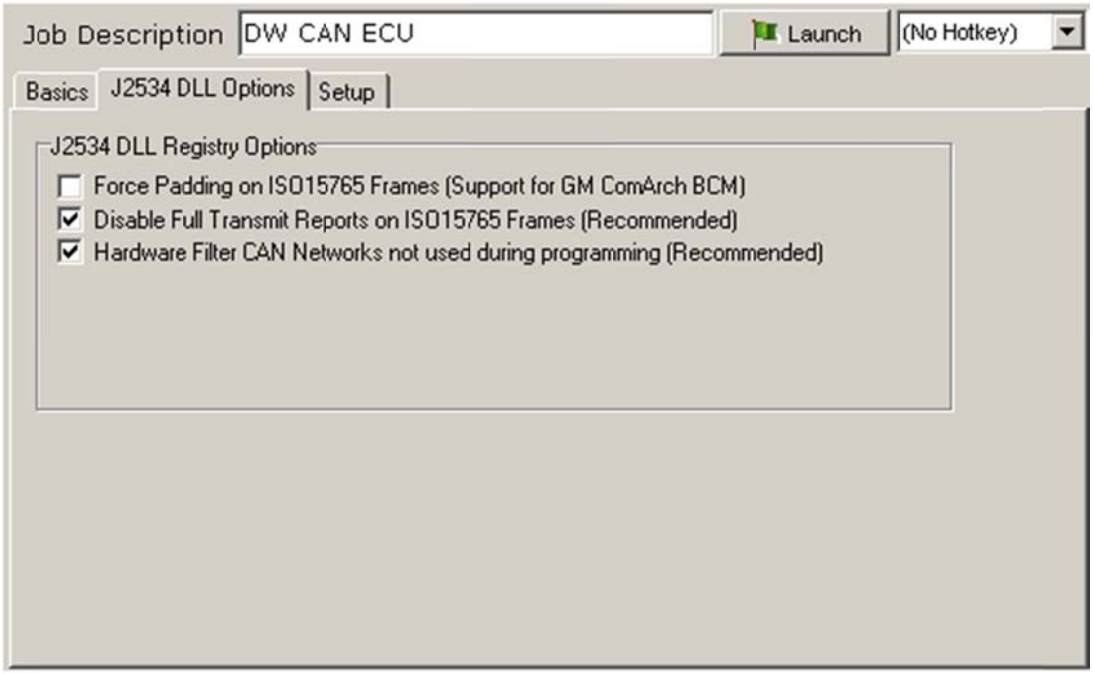

图 2-57 DPS 编程“J2534 DLL Options”标签

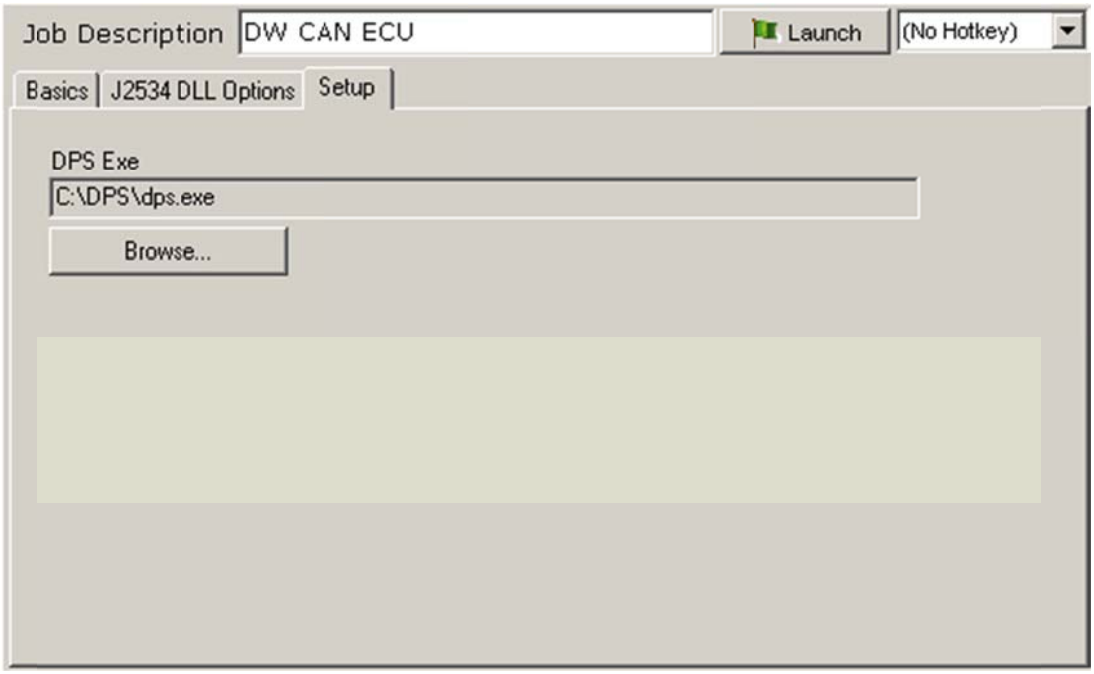

图 2-58 使用“Setup”标签选择主要的 DPS 执行文件位置

5. 确认了“DPS.exe”位置

在“Setup”标签中的“DPS Exe”区域，如图 2-58 所示，指定“DPS.exe”文件的位置。如果 DPS 安装在另外一个不同的目录中，单击“Browse”按钮并选择

合适的目录。

6. 创建另外一个任务

添加第 2 个 DPS 编程任务来对第 2 个 ECU 进行编程。如果直接使用完成的示例，为两个诊断任务选择合适的 DPS 档案文件。

2.9.2 第 2 部分-创建 Function Block 执行编程任务

1. 打开 Function Block 视图

下一步是创建一个 Function Block，控制已经创建好了的诊断任务。依次单击“Scripting and Automation”->“Function Block”打开 Function Block 窗口。

2. 创建一个 Function Block

单击“+”按钮并选择“Script”，创建一个新的脚本型 Function Block。更改 Function Block 的“Description”为描述性的内容，如“Program ECUs”。

3. 写入脚本

现在需要输入脚本。对于本任务需要创建 6 个步骤，所需的步骤及其描述见表 2-2 所示。对于“Diag Job Action”，双击“Value”单元格选择执行的诊断任务以及如何执行。更多信息请参见诊断相关章节。

表 2-2 任务脚本

Step	Command	Value	Reason for Step
1	Diag Job Action	Start DW CAN ECU	Starts, or launches, the diagnostic service.
2	Diag Job Action	Wait until Complete DW CAN ECU	Waits at this step until the selected diagnostic service is finished. When completed, continues to the next step.
3	Wait For	5.000 Sec	Waits for 5 seconds. This is not needed, but provides a delay so script progress can be monitored by the user.
4	Diag Job Action	Start SW CAN ECU	Starts, or launches, the diagnostic Service.
5	Diag Job Action	Wait until Complete SW CAN ECU	Waits at this step until the selected diagnostic service is finished. When completed, continues to the next step.
6	Stop	n/a	Ends this function block.

4. 配置启动设置

在“Start”标签上选择“Manual Start”，让脚本型 Function Block 利用图形面板控件来启动，具体设置将在本教程的下一部分介绍。

2.9.3 第 3 部分-创建图形面板

让我们通过创建一个图像面板使 DPS 编程刷新更简单化！

1. 打开图像面板

通过“Measurement”->“Graphical Panel”打开图形面板编辑器。这就是您发挥创造性的时候了，创建一个图像面板也没有正确或者错误的方式。

2. 设置面板

在该面板中，我们将从屏幕底部的选择 7 个控件。其中一些需要放置多个，每一个对应一个 DPS 编程任务。

第 1 个是 Function Block Button，用于启动 Function Block。按钮必须要设置的属性是：

FunctionBlock：Program ECU

FBlockAction：0-Start

其它需要设置的控件及属性如下所列：

Text Display

Signal：选择“Jobs”，然后选择第 1 个任务的“Log”以及“Value”属性。（确认“Evaluate as text”被勾选。）

Text Display

Signal：选择“Jobs”，然后选择第 2 个任务的“Log”以及“Value”属性。（确认“Evaluate as text”被勾选。）

Bargraph

Signal：选择“Jobs”，然后选择第 1 个任务的“Percent”以及“Value”属性。

Min：0

Bargraph

Signal：选择“Jobs”，然后选择第 2 个任务的“Percent”以及“Value”属性。

Min：0

LED

Signal：选择“Jobs”，然后选择第 1 个任务的“Is Successful”属性。

LED

Signal：选择“Jobs”，然后选择第 2 个任务的“Is Successful”属性。

利用上述设置，当单击 Function Block Button 后，将启动 DPS 编程任务。Text Display 将会显示当前的记录文本。Bargraph 将显示编程的进度，当编程完成之后，LED 等将会点亮。图 2-59 显示了完成后的示例。

图 2-59 带有 2 个 ECU 的 DPS 编程图形面板完成示例

2.9.4 第 4 部分-使用 DPS 编程图像面板

1. 使面板工作起来

为了使用该示例，连接相关硬件到车辆或者 ECU 后，启动 Vehicle Spy。单击图形面板上的按钮启动创建的 Function Block。屏幕上会跳出一些来自 DPS 的窗口。Vehicle Spy 将控制这些窗口，在它们完成之后将其关闭。图像面板上的控件将显示进度以及编程事件的各种记录。当第 1 个 ECU 完成之后，第 2 个将会启动编程。

2.9.5 第 5 部分-DPS 自动化编程结论

本教程展示了如何利用点击 1 个按钮，实现刷新多个 ECU。同时也展示了如何联合 Vehicle Spy 不同的组件完成自动化的任务。（与教程 5 相似。）

2.10 教程 7：诊断

概述

本教程将诠释一些诊断任务的工作原理以及作用。本示例中将会涵盖如下内容：

- 根据读取的 DID，写入 DID
- 读取通过动态 DPID 创建的 DPID

在开始本示例之前，假设前面的示例都已完成并且您已经具备了 GMLAN 相关的基本知识。如有车辆或者测试平台将会很有帮助，但这也不是必需的。重要的是理解介绍的概念。

2.10.1 第 1 部分-读取 DID 设置

在本示例中，从读取 DID 服务获取的 DID 信息传给写入 DID 服务。该应该可用于读取 VIN 码，刷新 ECU，然后将 VIN 码写回至 ECU。

1. 登录 Vehicle Spy

打开 Vehicle Spy 并选择登录名后，利用主菜单栏，依次选择“File”->“New”。

2. 创建一条读取 DID 任务

下一步是创建诊断任务读取 DID。依次选择“Vehicle Network”->“Diagnostic Setup”打开设置诊断窗口。利用“+”按钮添加一条 GMLAN“$1A”读取 DID 任务。

3. 编辑读取 DID 任务

下一步设置读取 DID 服务。如果您连接了一辆车或者测试平台，来读取或写入 DID 信息，使用该设置。如果您只是跟着学习，本示例将设置读取 VIN 码。选择一个 ECU 的物理地址（如果不从数据库中选择，还需要网络）以及 DID 码。对于 VIN，选择$90。如图 2-60 所示。

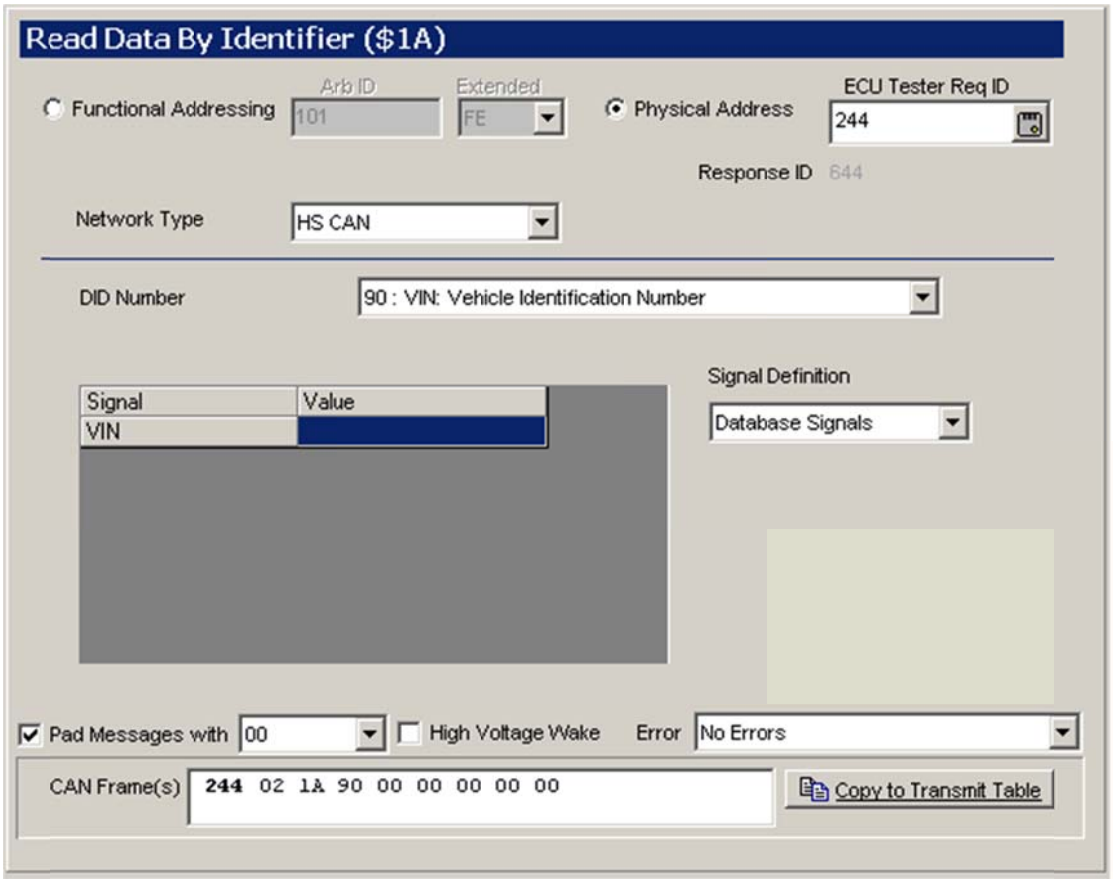

图 2-60 读取 DID 诊断任务的设置

2.10.2 第 2 部分-写入 DID 设置

1. 创建一条写入 DID 任务

现在可以创建写入 DID 服务（$3B）。利用“+”按钮添加一条 GMLAN $3B 的

写入 DID 任务。

2. 选择需要写入的 DID 码

下一步选择哪一个 DID 码需要写入。同样，这里再次使用 VIN 码作为例子。如果您连接到一辆车或测试平台，使用同第 1 部分读取 DID 相同的 DID 码。选择一个 ECU 的物理地址（如果不从数据库中选择，还需要网络）以及 DID 码。对于 VIN，选择$90。

3. 选择写入的数据

最后设置写入 DID 与读取 DID 之间的链接关系。在 DID 码选择下面，您可选择输入需要写入的数据。选择“Enter From Read DID”，然后会出现一个标有“Select Read DID Service”的下拉菜单。从下拉菜单中找到要关联的读取 DID 服务并选择它。在本示例中只有一个服务可选。如图 2-61 所示。

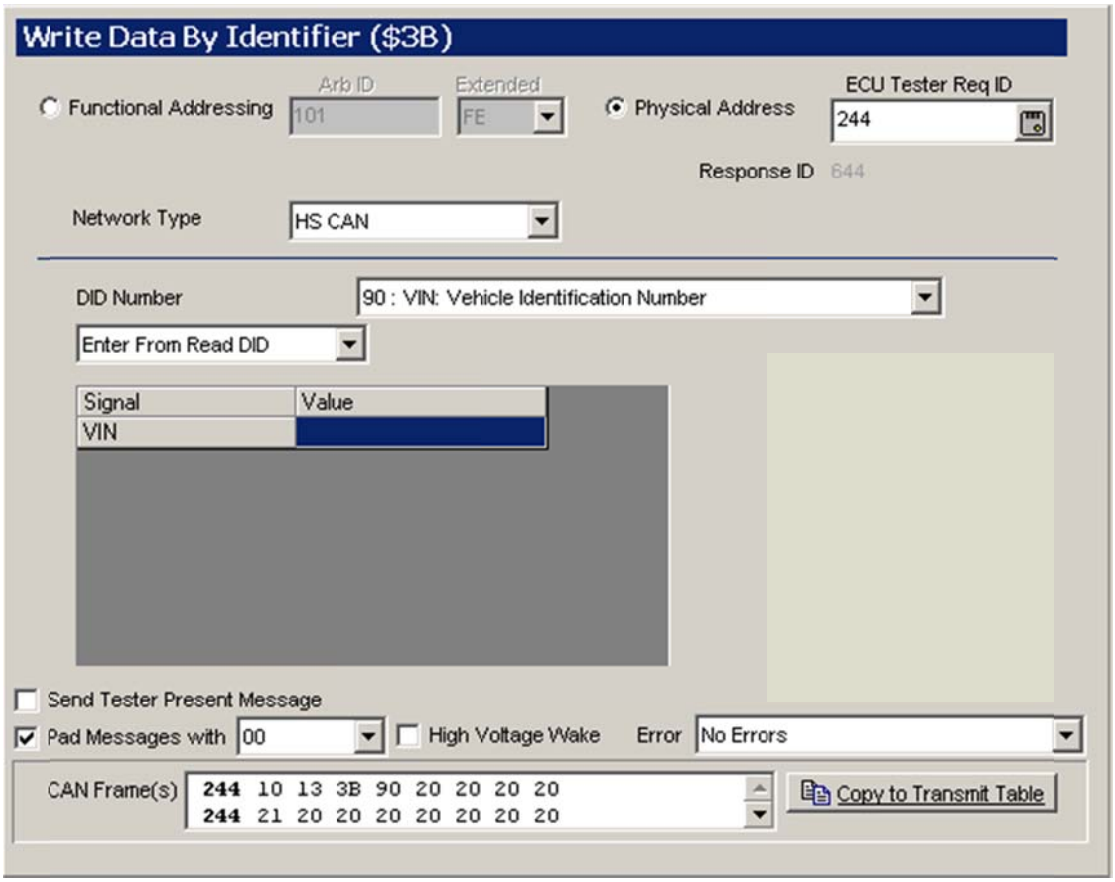

图 2-61 从读取 DID 获取数据的写入 DID 诊断任务设置

2.10.3 第 3 部分-写入读取的 DID

1. 执行读取 DID 任务

在诊断窗口中（“Vehicle Network”-> “Diagnostics”），执行读取 DID 服务。如果您连接着硬件，并且该硬件连接至车上或者测试平台上，您将会接收到请求的数据。

2. 执行写入 DID 任务

现在执行写入 DID 服务。该服务将想 ECU 中由读取 DID 服务提供的数据写入 VIN 码。如图 2-62 所示。

这里只是显示了一个简单的例子。可以选择不同的 ECU 进行写入 DID 服务。

本例子还可做些变化，如在读取 DID 和写入 DID 之间包含一个 DPS 刷新动作。这样当刷新完成之后，VIN 码以及其它参数可被写回至 ECU 中。

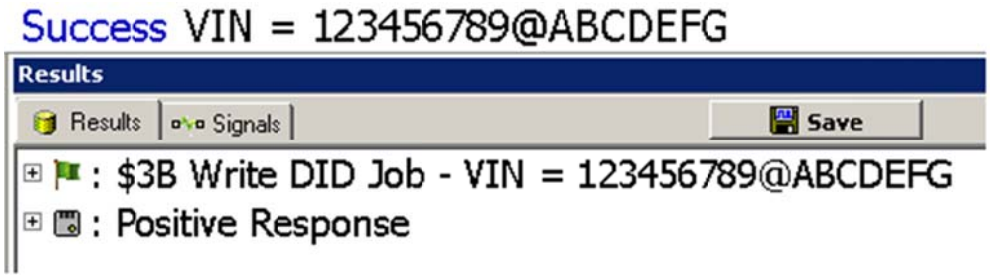

图 2-62 写入 DID 任务的正确响应

2.10.4 第 4 部分-从动态 DPID 请求 PID

在第 4 和 5 部分，将使用动态 DPID 过程中请求 PID。通过使用诊断服务定义动态 DPID（$2C）和读取 DPID（$AA）来完成。

1. 创建 PID

跟前面一样，本示例可以工作于硬件上。如果没有硬件，也过跟着理解该过程。第 1 步是确保获取数据的 ECU 所对应的 PID 已被定义。该信息可在“Vehicle Networks”->“ECUs”下面输入。选择 ECU 后，单击“PIDs”标签并输入 PID 数据。如需更多信息请参见 ECU 视图相关帮助主题。

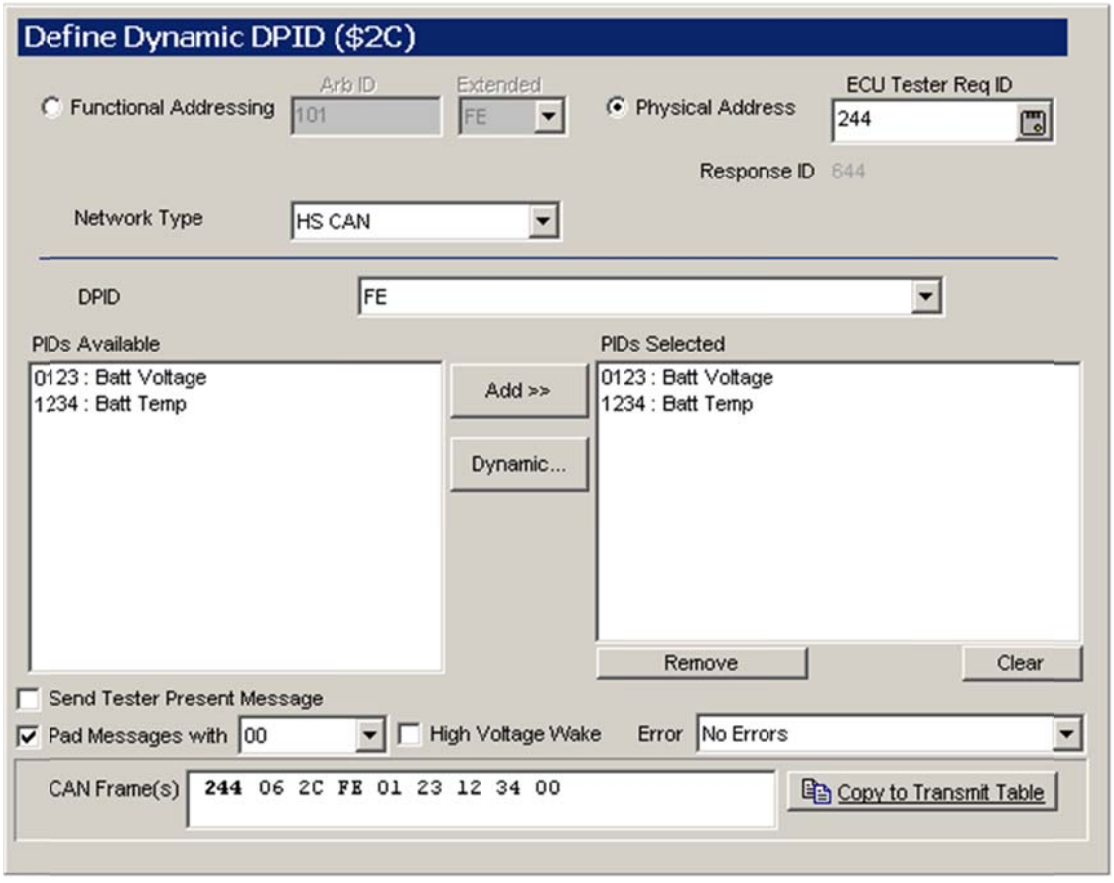

图 2-63 GMLAN 动态 DPID（$2C）设置示例

2. 建立动态 DPID

下一步创建动态的 DPID 请求 PID 数据。在设置诊断窗口（“Vehicle Networks” -> “Diagnostics Setup”）中添加一条 GMLAN（$2C）动态定义报文任务。选择 ECU 以及网络类型。从“PIDs Available”列表中选择一个 PID 并单击“Add >>”按钮将其放置到动态 DPID 中。如图 2-63 示例。

2.10.5 第 5 部分-请求动态 DPID

1. 建立动态 DPID 请求

最后的设置是请求 DPID。利用通过 Packet Identifier 读取数据服务（$AA），可获得 DPID 数据。在设置诊断时添加一条“Read Data By Packet Identifier”（$AA）服务。接着，设置 ECU ID 以及网络。保留设置“Sub Function”为“Send One Response”。为了将动态 DPID 连接到读取 DPID 服务上，单击“Dynamic…”按钮并选择第 4 部分中的动态 DPID，如图 2-64 所示。

2. 请求 PID

现在所有需要的就是请求 PID 的设置。在设置诊断窗口（“Vehicle Networks” -> “Diagnostics”）中，执行动态 DPID 服务。如果接收到正确的响应，接着执行 DPID 服务。PID 中被请求的数据将会显示出来。注意 Vehicle Spy 已经使用诊断数据库对 PID 进行解码，该数据库已经将 ECU 视图下面设置内容的进行了解析。

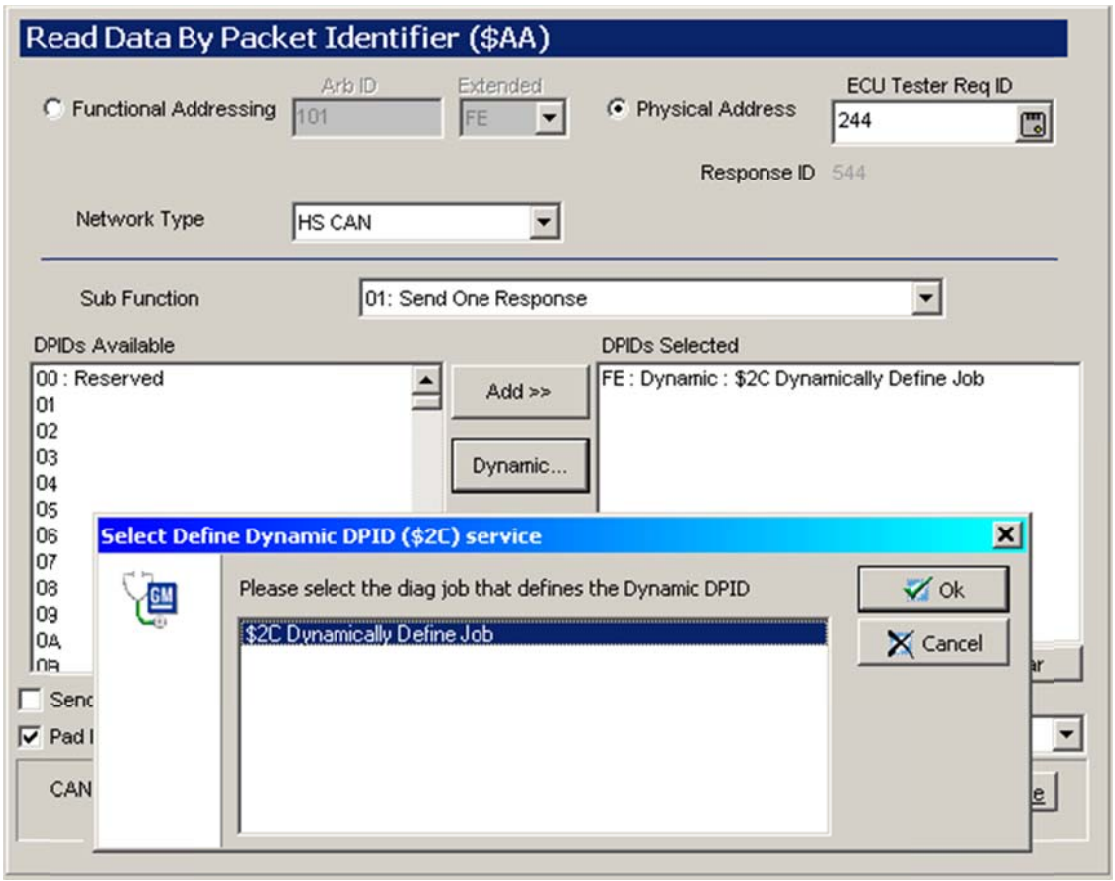

图 2-64 GMLAN 通过设置请求动态 DPID 的 Packet Identifier（$AA）读取数据示例

2.10.6 第 6 部分-结论

本教程描述了如何将诊断服务连接在一起完成不同的任务。可以使用 Function Block 来自动发送不同的服务并对回来的响应做出回应。