Application Note

在 EK-TM4C123GXL LaunchPad 上使用 Edde Flex CAN 控制器

摘要

该应用报告演示了如何使用 TM4C CAN 模块进行简单的发送和接收操作。它还包括实现基于 CAN 的引导加载程序所需的代码和实用程序。

本应用中所讨论的工程配套资料和源代码可从以下 URL 下载：http://www.ti.com/cn/lit/zip/spna244。

内容

商标

1 引言

TM4C MCU 提供广泛的有线连接外设，包括仅需要外部 CAN 收发器的集成 CAN 总线。德州仪器 (TI) 的低成本评估平台，例如 EK-TM4C123GXL LaunchPad™ 评估套件（用于基于 Arm® Cortex®-M4F 的微控制器），可以对许多器件功能进行快速且经济实惠的评估。但是，LaunchPad 缺少所需的板载 CAN 收发器。Edde Flex CAN BoosterPack 提供了 EK-TM4C123GXL 通过 CAN 进行通信所需的收发器。

2 为 TI TIVA C 安装 Edde Flex CAN Booster Pack

在 EK-TM4C123GXL Launchpad 上安装 Edde Flex CAN Booster Pack 非常简单。请注意，为了正确对齐，Booster Pack 上的丝印文字与 LaunchPad 上的丝印文字方向相反。有关正确的方向，请参阅图 2-1。

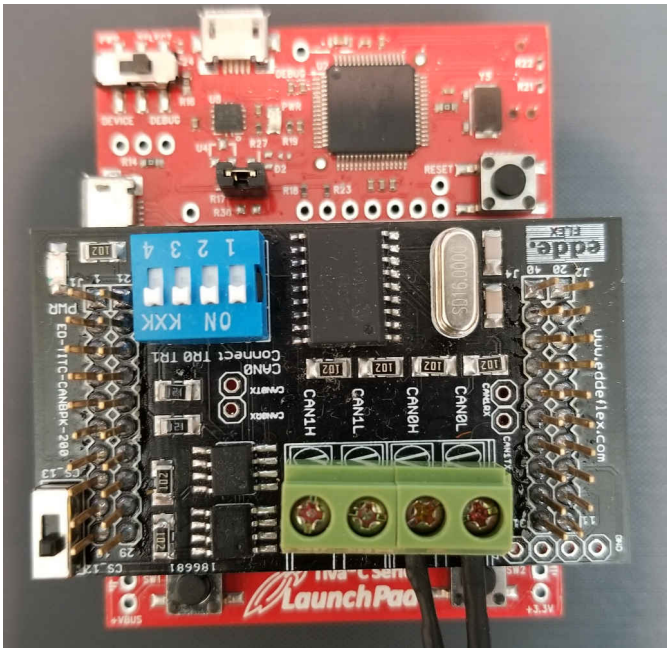

图 2-1. LaunchPad 和 BoosterPack 的方向

当开关 1 和 2 处于“打开”位置时，PE4 和 PE5 将连接到 Booster Pack CAN 收发器的发送和接收引脚。处于“打开”位置的开关 3 在 CAN0H 和 CAN0L 之间连接一个 120 Ω 终端电阻。

除了内部环回示例外，所有示例都需要一个正常工作的 CAN 网络。可以使用两个 EK-TM4C123GXL 构建一个简单的 CAN 网络，每个 EK-TM4C123GXL 都带有一个 Edde Flex Booster Pack，如图 2-2 所示。请注意，两个系统之间存在三个连接，即 CAN0H、CAN0L 和 GND。

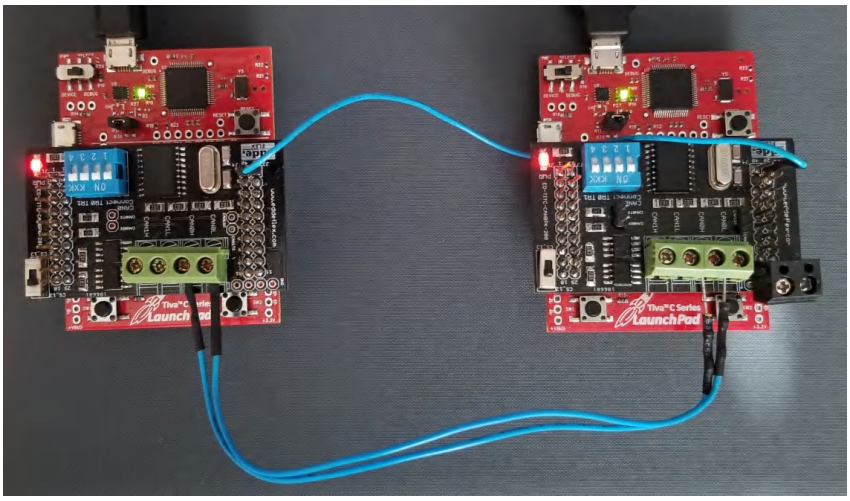

图 2-2. 两器件 CAN 网络

3 下载并导入 CAN 主机示例

本应用报告附加了七个 Code Composer Studio™ (CCS) 工程示例作为配套资料。可通过以下 URL 下载这些示例：http://www.ti.com/cn/lit/zip/spna244。用户可以解压缩该工程，或将其保留为 zip 格式。这两种格式都可以导入 CCS。

1. 若要将该工程导入 CCS，请首先选择“File”（文件）→“Import”（导入）。

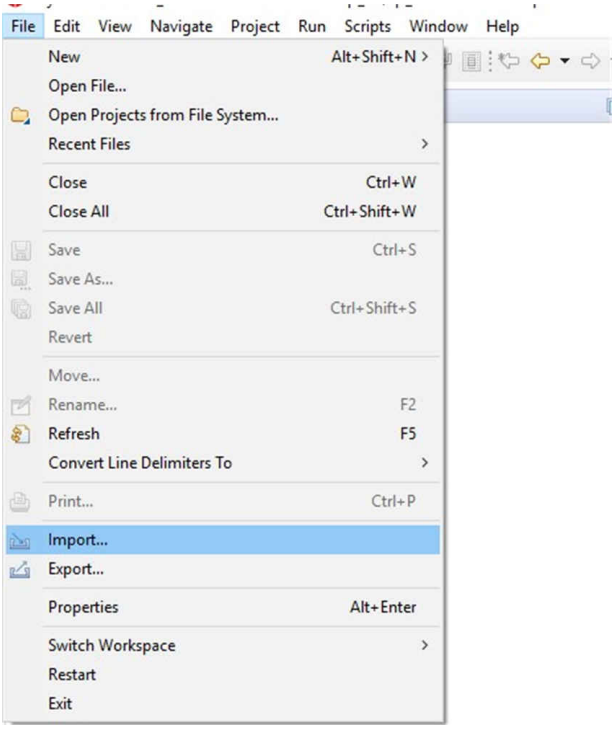

图 3-1. 导入 CCS 工程步骤 1

2. 选择“CCS Projects”（CCS 工程）以导入示例，然后点击“Next”（下一步）。

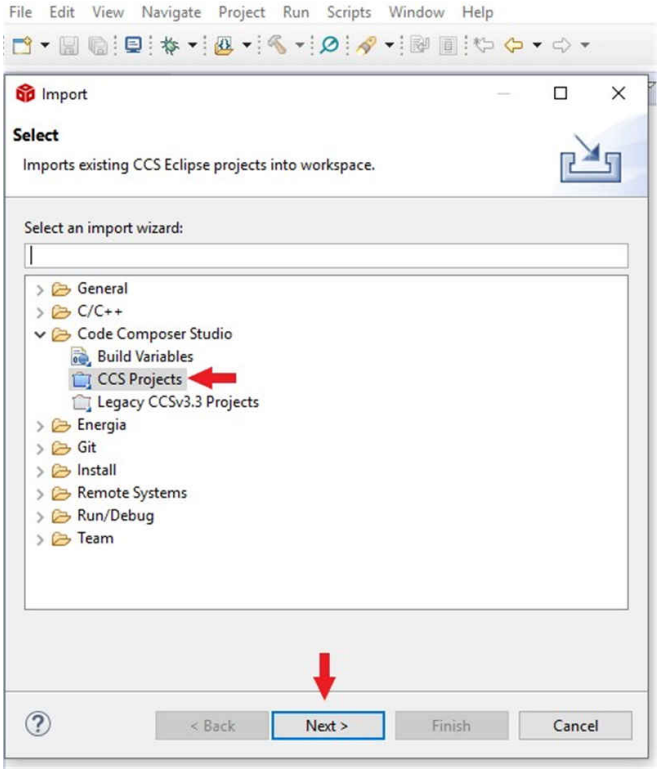

图 3-2. 导入 CCS 工程步骤 2

3. 提供解压缩工程（选择第一个单选按钮）或直接从 zip 文件（选择第二个单选按钮）导入。点击“Copy projects into workspace”（将工程复制到工作区）。

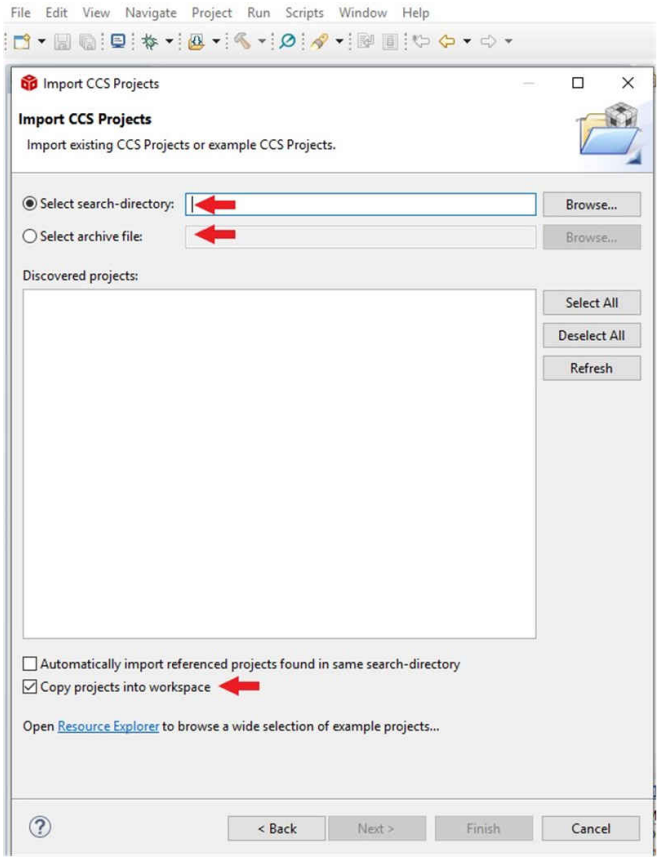

图 3-3. 导入 CCS 工程步骤 3

4. 提供工程路径后，将显示七个已发现的工程。首先点击“Select All”（全选）按钮，然后点击“Finish”（完成）按钮完成导入。

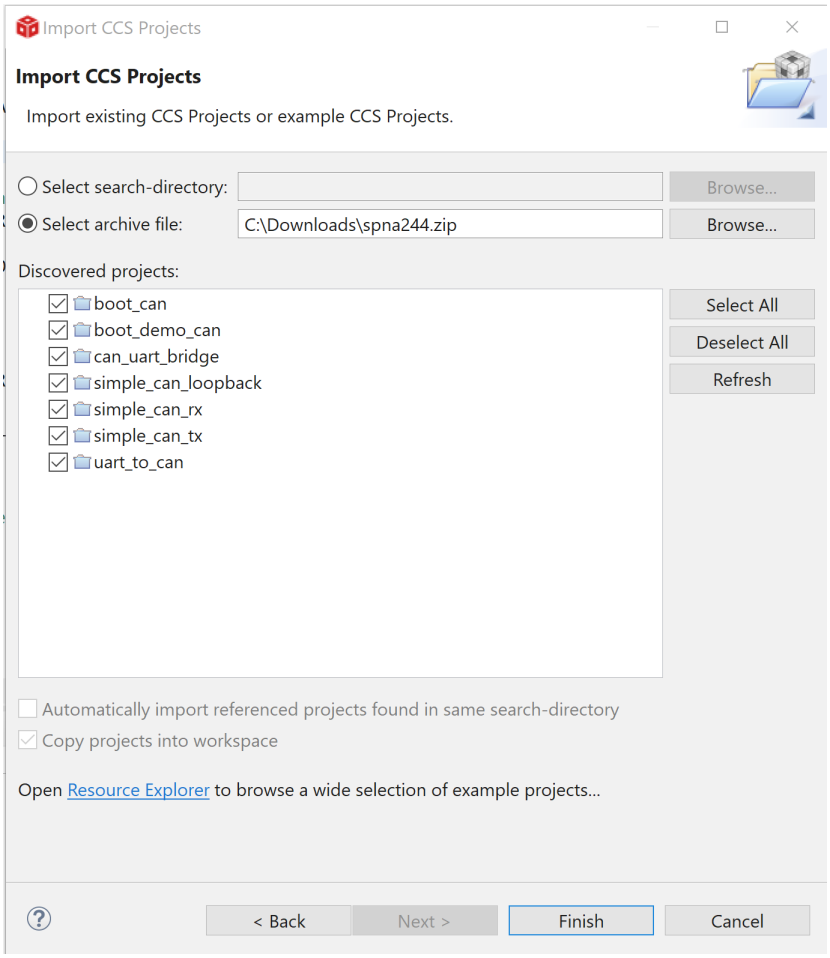

图 3-4. 导入 CCS 工程步骤 4

这些工程是使用 CCS 9.3 版和德州仪器 (TI) ARM 编译器工具 18.12.6 版构建的。

还有一个用于 PC 的可执行程序 (SendCANID.exe) 以及三个与 CAN 引导加载程序演示程序一同使用的 .bat 文件（StopBlue.bat、StopGreen.bat 和 StopRed.bat）。这些文件应该从 .zip 文件中提取并放置在可以执行它们的目录中。

4 用于中断且经过修改的 CAN.C

TivaWare 2.2.0.295 及更早版本中的驱动程序库文件 can.c 版本以及基于 ROM 的版本，针对大多数功能使用接口一 (IF1) 寄存器，并使用接口二 (IF2) 寄存器接收消息 (CANMessageGet())。在主应用程序代码中以及在中断例程或 RTOS 的多个线程中使用 CAN 功能时，这会产生问题。在中断或任务切换期间，接口寄存器的一致性无法保持。本应用手册中包含的更新版 can.c 在 CPU 未处于异常状态时对所有功能使用 IF1 寄存器，并在 CPU 处于异常状态时使用 IF2 寄存器。这简化了使用中断为 CAN 编写代码的过程。使用 RTOS 时，所有 CAN 功能都应在一个线程或一个中断例程中。

位于最新 TivaWare 安装程序的 driverlib 文件夹中的 can.c 文件可以替换为所提供的 can.c 文件并重建驱动程序库，或者可以将所提供的 can.c 文件直接添加到工程中。

5 示例工程

这些示例使用以下外设和 I/O 信号。用户必须查看它们并根据自有电路板的需要进行更改：

- CAN0 外设
- GPIO 端口 E 外设（适用于 CAN0 引脚）
- CAN0RX - PE4

- CAN0TX - PE5

使用以下 UART 信号。用户必须查看它们并根据自有电路板的需要进行更改：

- GPIO 端口 A 外设（适用于 UART0 引脚）
- UART0RX - PA0
- UART0TX - PA1

5.1 带有中断的内部环回 (simple_can_loopback)

在这些示例中，只有此示例可以在没有 CAN 收发器或 CAN 网络的情况下在 Launchpad 上运行。在环回模式下，如果配置了额外的邮箱来接收 CAN 模块发送的消息，则允许 CAN 模块接收该消息。同样在环回模式下，模块会忽略确认位的缺失。CAN TX 引脚处于活动状态，可以在示波器或逻辑分析仪上查看。此示例中的 CAN 帧如下所示，缺少的确认位以红色圈出。

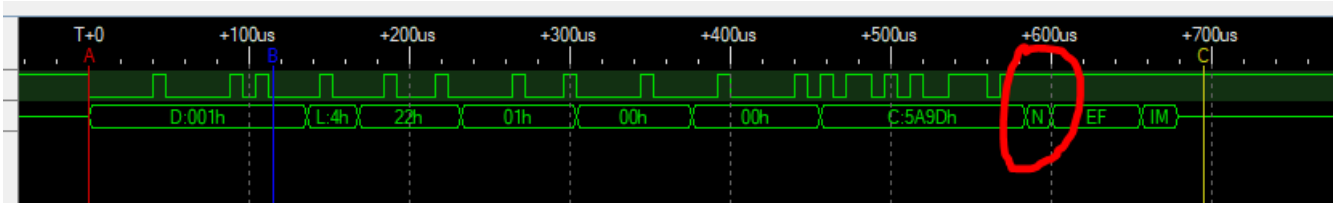

图 5-1. 逻辑分析仪波形

发送的消息是包含递增模式的 4 字节消息。发送的值和接收的消息在 UART 中回显。在本例中，CAN0 波特率为 125000，UART0 波特率为 115200。

此示例使用以下中断处理程序。若要在自有应用中使用此示例，用户必须将此中断处理程序添加到向量表中。

- CANIntHandler - CAN 0 中断处理程序

5.2 简单 CAN 传输 (simple_can_tx)

此示例显示了 CAN 的基本设置，以便在 CAN 总线上传输消息。将 CAN 外设配置为传输具有特定 CAN ID 的消息。然后，使用简单延时计时环路每秒传输一次消息。发送的消息是包含递增模式的 4 字节消息。CAN 中断处理程序用于确认消息传输并计算已发送的消息数。发送的消息和消息计数显示在 UART 中。

此示例使用以下中断处理程序。若要在自有应用中使用此示例，用户必须将此中断处理程序添加到向量表中。

- CANIntHandler - CAN 0 中断处理程序

5.3 简单 CAN 接收 (simple_can_rx)

此示例显示了 CAN 的基本设置，以便从 CAN 总线接收消息。将 CAN 外设配置为接收具有任何 CAN ID 的消息，然后通过 UART 将消息内容打印到控制台。

此示例使用以下中断处理程序。若要在自有应用中使用此示例，用户必须将此中断处理程序添加到向量表中。

- CANIntHandler - CAN 0 中断处理程序

5.4 CAN UART 桥接器 (can_uart_bridge)

此示例将 UART 连接到 CAN 总线。它在 UART0 上接收数据，然后在 CAN0 上发送出去，在 CAN0 上接收数据，然后在 UART0 上发送出去。对于此示例，将 UART0 配置为 115200 波特（在 *UartFunctions.h* 中定义），将 CAN0 配置为 1M 波特（在 *CanFunctions.h* 中定义）。在这些速率下，UART 上连续接收的数据将消耗 17% 的 CAN 总线带宽。

UART 或 CAN 接收到的数据由中断例程处理。接收到的数据被放入其各自的循环缓冲区。两个缓冲区都是使用 256 字节的大小定义的（在 *main.h* 中定义）。当 UART 接收到连续数据时，UART 中断例程在接收到四个字节（FIFO 填满一半）后触发。然后收集数据，直到接收到八个字节，然后再放入循环缓冲区。通过在 CAN 数据包中支持多达 8 个字节，最大限度地提升了 CAN 传输效率。当 UART 接收终止或暂停时，会产生接收超时 (RT) 中断。在这种情况下，UART 接收 FIFO 中的所有字节都被传输到循环缓冲区，并发送一个小于 8 个字节的 CAN 帧。

CAN 传输的消息和 CAN 接收的消息的仲裁 ID 相同，均为 0x101（在 *CanFunctions.h* 中定义）。这可以让运行同一软件的两个器件通过 CAN 网络将数据从一个串行端口传输到另一个串行端口。网络上有两个以上的器件运行同一软件，一个串行端口接收到的数据通过 CAN 网络传输，然后输出至所有其他器件的串行端口。

此示例使用以下中断处理程序。若要在自有应用中使用此示例，用户必须将这些中断处理程序添加到向量表中。

- CAN0IntHandler - CAN 0 中断处理程序
- UART0IntHandler - UART 0 中断处理程序

6 CAN 引导加载程序

6.1 引导加载程序配置

工程“boot_can”是 TivaWare 2.2.0.295“boot_loader”目录的副本和“bl_config.h”经修改的副本。对“bl_config.h”所做的修改将晶振频率设置为 16 MHz 并启用 CAN 引导加载程序。将 CAN 波特率设为 125K，将引脚 PE4 和 PE5 配置为 CAN 引脚。必须使应用程序代码偏移，才能将起始地址设为 0x1000。

6.2 引导演示程序 (boot_demo_can)

该工程包含一个示例应用程序代码，该代码链接到地址 0x1000，并与 CAN 引导加载程序兼容。此应用使三色 LED 闪烁并查找特定的 CAN 消息，从而使其跳回 CAN 引导加载程序。存在名为“BLUE”、“GREEN”和“RED”的三种配置，而非“调试”配置。这三个配置目录都有一个“boot_demo_can.bin”文件。在看到 ID 为 0x1F028000 且第一个数据字节为 0x04 的 CAN 帧时，“BLUE”目录中的版本会使蓝色 LED 闪烁并跳回引导加载程序。在看到 ID 为 0x1F028000 且第一个数据字节为 0x08 的 CAN 帧时，“GREEN”目录中的版本会使绿色 LED 闪烁并跳回引导加载程序。在看到 ID 为 0x1F028000 且第一个数据字节为 0x02 的 CAN 帧时，“RED”目录中的版本会使红色 LED 闪烁并跳回引导加载程序。有关如何使用 CAN 引导加载程序和这些演示程序的说明，请参阅节 6.4。

6.3 用于实现 LM Flash Programmer 支持的 UART 转 CAN 桥接器

6.3.1 CCS 演示程序 (uart_to_can)

这是一个 Code Composer Studio 工程，用于执行 TM4C 器件中的代码，它允许将来自 LM Flash Programmer 的串行端口命令桥接到可供 CAN 引导加载程序使用的 CAN 帧。LM Flash Programmer 是德州仪器 (TI) 提供的实用程序，可用于对 TM4C 微控制器上的闪存进行编程。它可以与 TM4C 器件上的串行引导加载程序、以太网引导加载程序或 USB DFU 引导加载程序进行通信。没有可与 TM4C 器件上的 CAN 引导加载程序进行对话的等效工具。

使用 uart_to_can 程序将提供一种使用 LM Flash Programmer 对运行 CAN 引导加载程序的 TM4C 微控制器进行闪存编程的方法。可以使用 JTAG 或串行引导加载程序将此应用代码编程到器件中，然后安装在 CAN 网络上。然后，它将使用 CAN0 将其在 UART0 上接收到的串行引导加载程序命令转换为 CAN 引导加载程序协议。有关使用此应用程序代码的示例，请参阅节 6.4。

6.3.2 SendCANID PC 程序

这是一个可以在 PC 上运行的命令行实用程序。它用于向运行 uart_to_can 软件的器件发送串行流。这会使器件在 CAN 总线上发出帧。CAN 总线上的器件可以使用该帧将该器件置于引导加载程序模式。

从 PC 命令行提示符或批处理文件执行此程序的格式为：

C:\>SendCANID COMnn 0xXXXXXXXX 0xXX

- nn = COM 端口号 (1 – 99)
- 0xXXXXXXXX = 用以切换到引导加载程序的命令的 CAN 仲裁 ID
- 0xXX = 发送以识别选择了哪个器件的 0 到 8 个十六进制字节

6.4 使用 CAN 引导加载程序

将目标文件“uart_to_can\debug\uart_to_can.out”编程到 CAN 网络上的 EK-TM4C123GXL 板上。确定将哪个 COM 端口分配给该板。将目标文件“boot_can\debug\boot_can.out”编程到 CAN 网络上的一个或多个其他 EK-TM4C123GXL Launchpad。Code Composer Studio 一次只能识别一个 Launchpad，因此可能需要单独连接到每个 Launchpad。所有 Launchpad 均完成编程后，即可在不使用 Code Composer Studio 的情况下运行演示。

带有“uart_to_can.out”的 Launchpad 应连接到 PC，以便访问其串行端口。其他 Launchpad 可以连接到 PC 或壁式 USB 充电器，仅用于供电。

打开 LM Flash Programmer 并将其配置为可在 115200 波特下使用 UART 接口。它应该使用通过“uart_to_can.out”程序连接到 Launchpad 的 COM 端口。

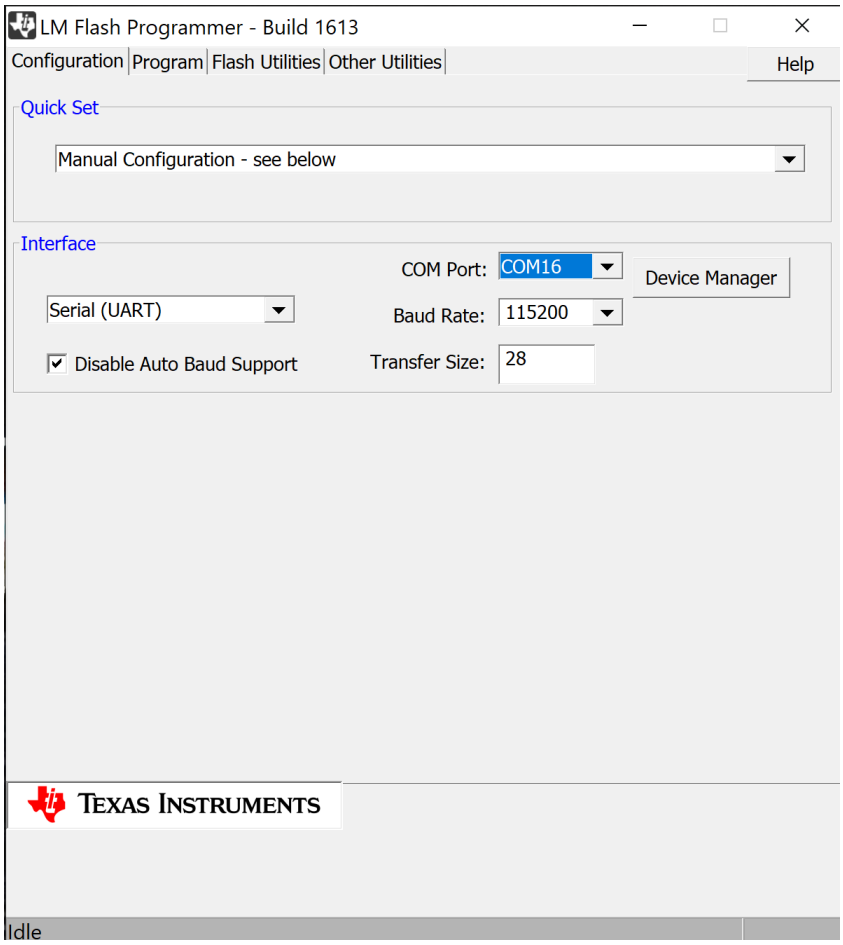

图 6-1. 配置 LM Flash Programmer

在 LM Flash Programmer 上，选择“Program”选项卡，然后浏览到“boot_demo_can\RED\boot_demo_can.bin”文件。将“Program Address Offset:”设置为 0x1000。按“Program”按钮。

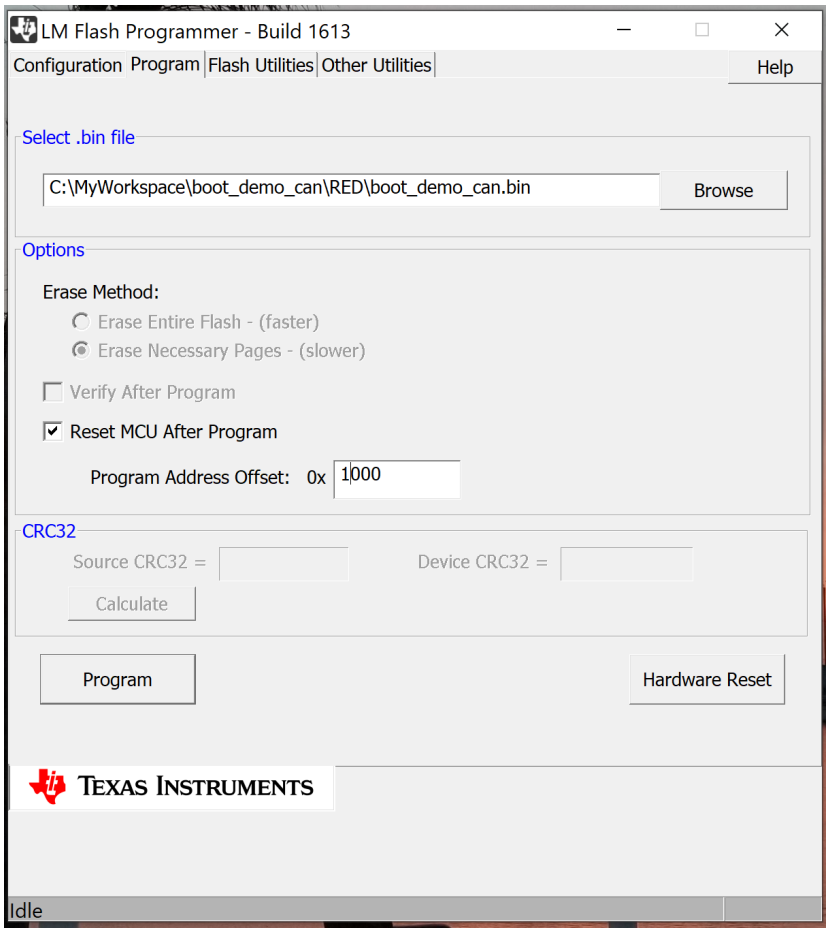

图 6-2. 使用 LM Flash Programmer 进行编程

编程完成后，用户将看到带有引导加载程序的板上的三色 LED 指示灯呈红色闪烁。该器件现在正在运行通过 CAN 引导加载程序编程的应用程序代码。此应用程序代码使三色 LED 呈红色闪烁，但它也在寻找特定的 CAN 帧，从而使 CPU 跳回 CAN 引导加载程序。仲裁 ID 为 0x1F028000 且第一个数据字节等于 0x2 的 CAN 帧将使该单元返回到 CAN 引导加载程序。

在 PC 上打开命令窗口并切换到包含示例工程的目录。该目录中有一个 PC 可执行程序“SendCANID.exe”和三个 .bat 文件。在命令窗口中输入“StopRed COMnn”，将 COM 端口号替换为“nn”。用户应看到与下文所示相似的结果，并且红色 LED 应停止闪烁。

```
C:\MyWorkspace>StopRed COM16
C:\MyWorkspace>SendCANID COM16 0x1F028000 0x02
Program to select CAN ID for download, Version 1.00
Opened serial port COM16 successfully
Sent ID: 0x1F028000 Data: 0x02
Received: 0xCC
C:\MyWorkspace>
```

现在 LaunchPad 已恢复为执行 CAN 引导加载程序，用户可以使用 LM FlashProgrammer 更改应用程序代码。尝试编程“boot_demo_can\BLUE\boot_demo_can.bin”。

使用此示例应用程序代码和引导加载程序代码，可以将 CAN 总线上的单个单元置于 CAN 引导加载程序模式并进行更新。CAN 总线上的每个器件都可以一次更新一个。can_to_uart 代码不支持同时更新多个器件。为此，需要修改 can_to_uart 代码和 CAN 引导加载程序，以支持对每个 CAN 命令的多个状态响应。

作为一项额外的功能，如果“uart_to_can”程序从“SendCANID”程序接收到 ID 0xFFFFFFFF 和数据 0x00，则“uart_to_can”程序将跳转到其 ROM 串行引导加载程序，从而允许“uart_to_can”程序被更新或覆盖。

重要声明和免责声明

TI“按原样”提供技术和可靠性数据（包括数据表）、设计资源（包括参考设计）、应用或其他设计建议、网络工具、安全信息和其他资源，不保证没有瑕疵且不做出任何明示或暗示的担保，包括但不限于对适销性、某特定用途方面的适用性或不侵犯任何第三方知识产权的暗示担保。

这些资源可供使用 TI 产品进行设计的熟练开发人员使用。您将自行承担以下全部责任：(1) 针对您的应用选择合适的 TI 产品，(2) 设计、验证并测试您的应用，(3) 确保您的应用满足相应标准以及任何其他功能安全、信息安全、监管或其他要求。

这些资源如有变更，恕不另行通知。TI 授权您仅可将这些资源用于研发本资源所述的 TI 产品的应用。严禁对这些资源进行其他复制或展示。您无权使用任何其他 TI 知识产权或任何第三方知识产权。您应全额赔偿因在这些资源的使用中对 TI 及其代表造成的任何索赔、损害、成本、损失和债务，TI 对此概不负责。

TI 提供的产品受 TI 的销售条款或 ti.com 上其他适用条款/TI 产品随附的其他适用条款的约束。TI 提供这些资源并不会扩展或以其他方式更改 TI 针对 TI 产品发布的适用的担保或担保免责声明。

TI 反对并拒绝您可能提出的任何其他或不同的条款。

邮寄地址：Texas Instruments, Post Office Box 655303, Dallas, Texas 75265