Application Note

用于控制器局域网的可配置错误发生器

Joseph Casuga and Hareesh Janakiraman *Applications Engineering - C2000 Real-Time Microcontrollers*

摘要

尽管控制器局域网 (CAN) 是在 30 多年前推出的（主要用于汽车应用），但发展势头仍然强劲，现已进入工业自动化、航空航天和医疗电子等其他应用领域。其中许多应用对安全至关重要，需要一个框架来模拟和分析网络中可能发生的不同类型的错误条件。虽然目前有许多廉价的 CAN 总线分析仪可用，但它们只提供一些基本的错误生成能力。例如，它们可以提供在指定位置生成错误标志的能力，但仅此而已。已针对一些学术环境开发了基于 FPGA 的错误发生器。本报告中提供了两种不同的方法（通用输入/输出 (GPIO) 方法和 LabVIEW™ 方法），它们在可能引入错误的精确位置方面具有更好的可配置性。

本文档中概述的 GPIO 方法适用于在 C2000Ware 中具有 DCAN 模块和 Driverlib 框架的所有 C2000™ 实时微控制器。目前，这包括以下器件：TMS320F2837xD、TMS320F2837xS、TMS320F2807x、TMS320F28004x、TMS320F2838x 和 TMS320F28002x。

代码示例在 TMS320F280049 器件上进行了测试；但是，可以轻松调整这些示例，以使其在任何具有 DCAN 模块的 C2000 器件上运行。本文档中描述的工程文件和示例作为 C2000Ware 的一部分提供。

可以从以下 URL 下载本文档所述的 LABVIEW 文件：https://www.ti.com/cn/lit/zip/spracq3。

内容

1 引言.....2
2 帧生成 – GPIO/CCS 方法.....3
3 帧生成 – LabVIEW 方法.....5
3.1 设置过程.....5
3.2 输入窗口.....5
3.3 输出窗口.....5
4 坐标系.....7
5 错误生成.....10
5.1 GPIO/CCS 方法.....10
5.2 LabVIEW 方法.....11
6 仿真错误帧.....12
7 参考文献.....16

插图清单

图 2-1. 帧生成 – GPIO/CCS 方法.....4
图 3-1. 帧生成 – LabVIEW 方法.....6
图 4-1. 未引起错误的基准数据帧（标准 MSGID），DLC = 4.....7
图 4-2. 未引起错误的基准远程帧（标准 MSGID），DLC = 4.....7
图 4-3. 未引起错误的基准远程帧（标准 MSGID），DLC = 0.....8
图 4-4. 未引起错误的基准数据帧（扩展 MSGID），DLC = 4.....8
图 4-5. 未引起错误的基准远程帧（扩展 MSGID），DLC = 4.....8
图 4-6. 未引起错误的基准远程帧（扩展 MSGID），DLC = 0.....9
图 5-1. 使用 GPIO/CCS 方法生成的错误.....10
图 5-2. 使用 LabVIEW 方法生成的错误.....11
图 6-1. 未引起错误的扩展 ID（LEC = 0；无错误）.....12
图 6-2. SRS 位发生翻转的扩展 ID（1 到 0）.....12
图 6-3. IDE 位发生翻转的扩展 ID.....13

图 6-4. RTR 位发生翻转的扩展 ID..... 13
图 6-5. r1 位发生翻转的扩展 ID..... 14
图 6-6. r0 位发生翻转的扩展 ID..... 14
图 6-7. MSGID 的 Bit0 发生翻转的扩展 ID（LEC = 6；CRC 错误）..... 15
图 6-8. Data0 的 Bit0 发生翻转的扩展 ID（LEC = 6；CRC 错误）..... 15
图 6-9. CRC 的 Bit0 发生翻转的扩展 ID（LEC = 6；CRC 错误）..... 15
图 6-10. 没有位填充的扩展 ID（LEC = 1；填充错误）..... 16

表格清单

表 1-1. 生成的 CAN 帧..... 2

商标

1 引言

本应用报告的目的是提供一个易于使用的硬件和软件框架，以生成和分析 CAN 总线中的不同类型错误。下文介绍了两种不同的方法：

- GPIO 方法：可以在任何适用的 C2000 器件上运行的测试用例。如有需要，这可以提供 GPIO 引脚的可见性。只需可正常使用的 C2000 目标板和 Code Compose Studio™ (CCS) IDE。
- LabVIEW 方法：如果需要集成到更大的测试装置（独立于 C2000 目标），此方法很有用。这需要节 3 中概述的硬件。

对于这两种方法，具有内置 CAN 总线触发/解码功能的示波器都是必不可少的。

本文档中的所有仿真波形都是在仿真 CAN 传输功能的 GPIO 引脚上捕获的。无法看到 CAN 接收器节点检测到错误并破坏正在运行的帧所产生的影响，因为波形无法反映真正的 CAN 总线活动，而只是对 CAN 功能进行仿真。出于这个原因，在示波器捕获中可以看到完整的“CAN 波形”。表 1-1 中显示的 CAN 帧是使用 GPIO 和 LabVIEW 方法生成的。这些模拟帧由 CAN 总线分析仪监控。分析器对帧的正确解释验证了这些帧是否以一致方式正确生成。

表 1-1. 生成的 CAN 帧

帧类型		ARBID	DLC	D0	D1	D2	D3	CRC
ID	远程请求							
标准	否	0x45B	4	95	1A	23	45	0x5AD8
标准	是	0x45B	4	不适用	不适用	不适用	不适用	0x238C
标准	是	0x45B	0	不适用	不适用	不适用	不适用	0x7B43
扩展	否	0x1914A75B	4	95	1A	23	45	0x4101
扩展	是	0x1914A75B	4	不适用	不适用	不适用	不适用	0x4EB3
扩展	是	0x1914A75B	0	不适用	不适用	不适用	不适用	0x167C

2 帧生成 – GPIO/CCS 方法

本节介绍了如何使用 GPIO/CCS 方法配置错误生成。这应该与图 2-1 结合使用。

- **配置消息长度和比特率**
  - 配置消息长度（以字节为单位）(DLC)
  - 配置比特率（以 bps 为单位）

硬件设置过程：

CAN_GPIO_MODE

1. 选择将生成 CAN 位流的 GPIO。
2. 将所选 GPIO 的跳线连接至所选 CANRX 引脚。
3. GPIO 和 CANRX 引脚之间直接连接。不涉及收发器。

CAN_DATALBCK_MODE

1. 无需外部连接。
2. CANTX 数据通过内部连接直接进入 CANRX 缓冲区。

- **配置 CAN 数据帧**
  - 输入所需的仲裁 ID（消息 ID）
  - 输入要传输的所需数据
  - 选择数据帧或远程请求

CCS-使用 Watch Expression 监控 CAN 帧相关变量

- 可以通过 CCS 中的 *Expressions*（表达式）窗口监控使用多项式 0xC599 生成的 15 位 CAN CRC。
- 也可以在 CCS 的 *Expressions*（表达式）窗口中监控填充位出现的次数。

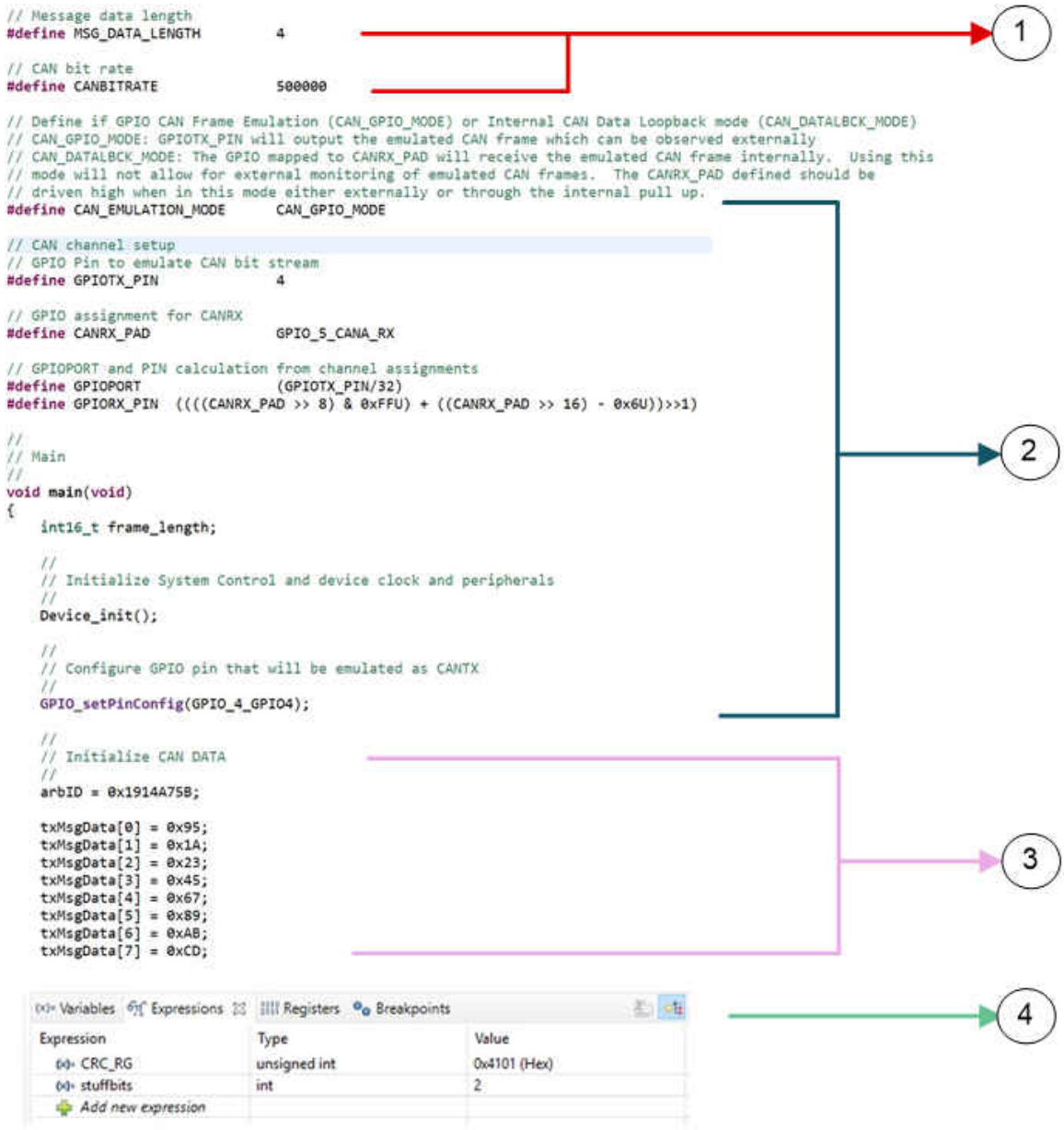

图 2-1. 帧生成 – GPIO/CCS 方法

3 帧生成 – LabVIEW 方法

本节演示了使用“*GenerateCANStream.vi*”通过 LabVIEW 方法生成 CAN 帧涉及到的步骤。还介绍了 Labview vi 中的各种指标和可配置选项。

3.1 设置过程

1. 获得一个任意波形发生器（本例中使用的是 Agilent AG33250；A33xxx 系列的任何 AWG 都应该可以无缝工作）、LabVIEW 开发软件包（2016 或更高版本）和一个使 LabVIEW 能够控制任意波形发生器 (AWG) 的 USB 转 GPIB 接口。
2. 解压“*GenerateCANStream.vi*”并将其保存至运行 LabVIEW 开发软件的本地计算机。
3. 使用 USB 转 GPIB 接口将 AWG 连接至计算机。将 AWG 的输出连接至 CAN 收发器的 CANTX 侧或直接连接到 CAN 模块的 CANRX，以响应生成的错误。

3.2 输入窗口

图 3-1 显示了 *GenerateCANStream.vi* 中的各个窗口。输入窗口带有红色边框。

- 错误生成：
  - 此时，将 Forcing Error 值设置为“None”。稍后在生成包含错误的 CAN 帧时配置此区域。
- 配置 AWG（波形发生器属性）：
  - 设置适当的 GPIB 仪器地址。
  - 使用 I/O 电平和增益将 I/O 电平和增益设置为所需的级别。在连接到 CAN 模块的 CANRX 引脚或收发器的 CANTX 引脚之前，使用示波器验证这些级别以确保其正确性。
  - 配置比特率。
- 配置 CAN 数据帧：
  - 输入所需的仲裁 ID（消息 ID）。
  - 配置消息长度（以字节为单位 - DLC）。
  - 输入要传输的所需数据。
  - 选择数据帧或远程请求。
- 按下 vi 前面板上的“Run”（运行）按钮以开始生成 CAN 帧。

3.3 输出窗口

在 LabVIEW GUI 中，输出窗口带有蓝色边框。

- 比特流数据：
  - 完整的比特流数据可用于位填充和非位填充字符串。
  - 另外还计算不同字段的索引或标记以补充数据流字符串。
- 帧信息：
  - 标识消息 ID 是标准的还是扩展的。
  - 使用多项式 0xC599 提供计算得出的 15 位 CRC。
  - 提供发生位填充的实例数。
- 状态：
  - 显示以下任一情况下 SRS、IDE、RTR 和保留位的位值：正常 CAN 帧生成，或在错误生成模式下翻转这些位中的任何位之时。亮绿色代表隐性 (1) 状态，而深绿色代表显性 (0) 状态。
  - 在错误生成模式中，具有预期位翻转的相应字段以亮绿色突出显示。

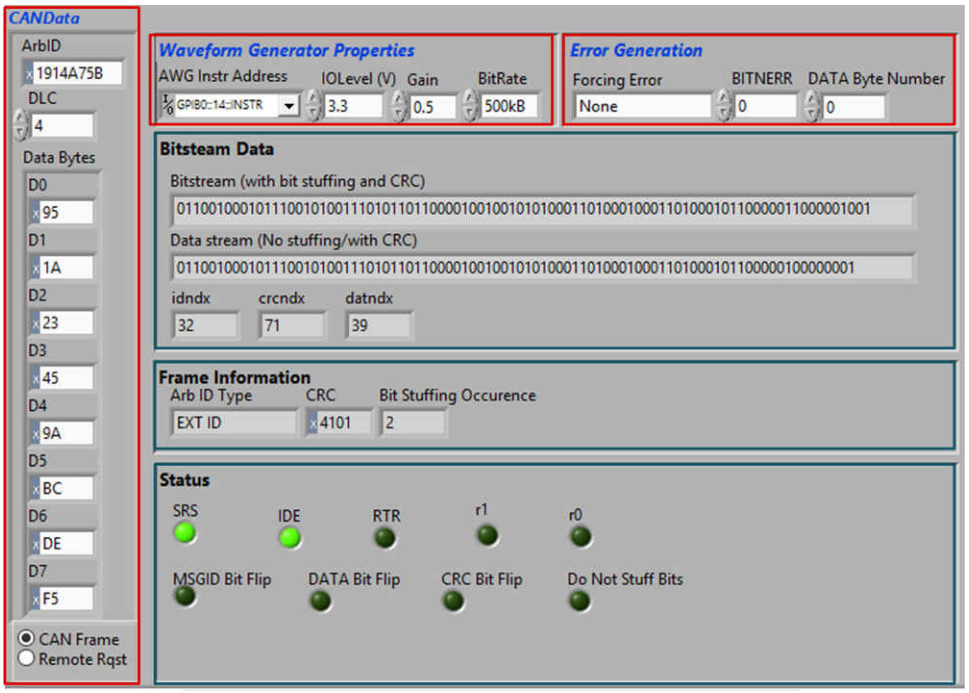

图 3-1. 帧生成 – LabVIEW 方法

4 坐标系

图 4-1 至图 4-6 描绘了根据表 1-1 生成的坐标系。所有坐标系均是使用基于 USB 的 CAN 总线分析仪生成的。

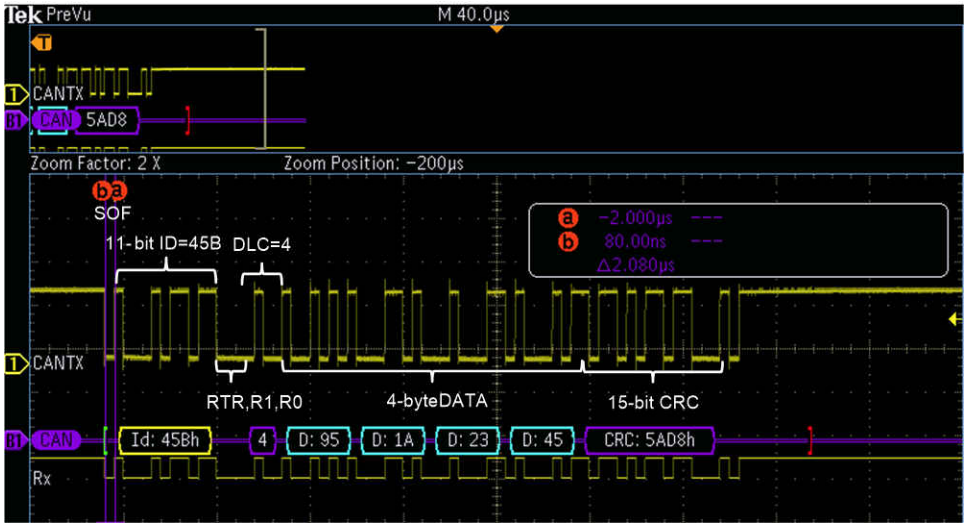

图 4-1. 未引起错误的基准数据帧（标准 MSGID），DLC = 4

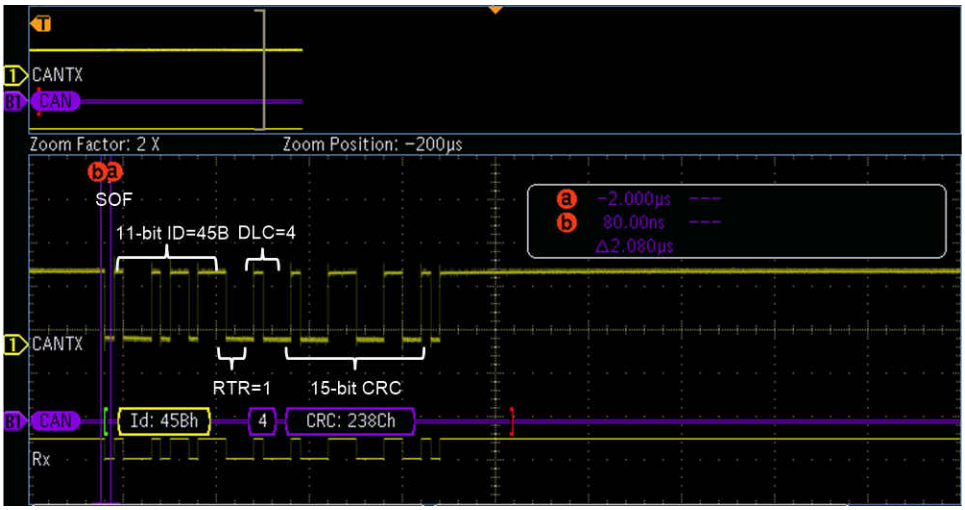

图 4-2. 未引起错误的基准远程帧（标准 MSGID），DLC = 4

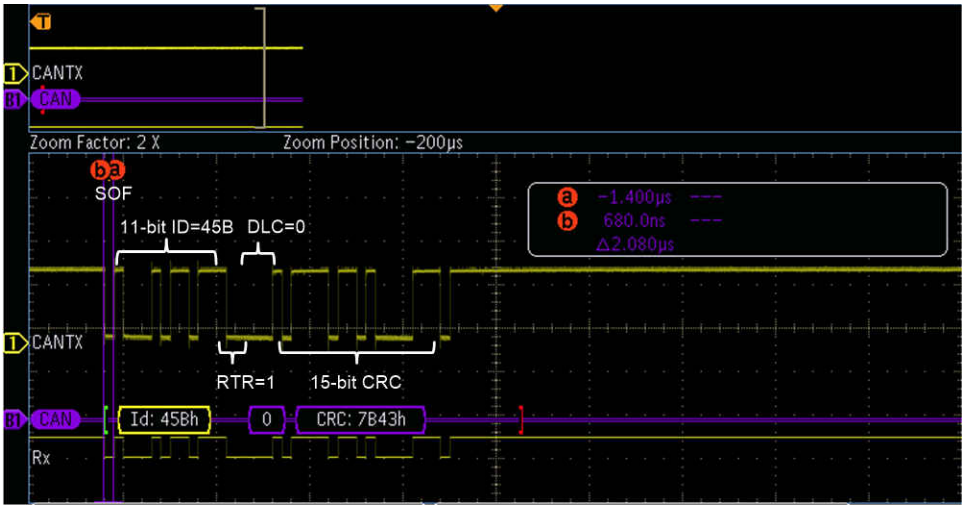

图 4-3. 未引起错误的基准远程帧（标准 MSGID），DLC = 0

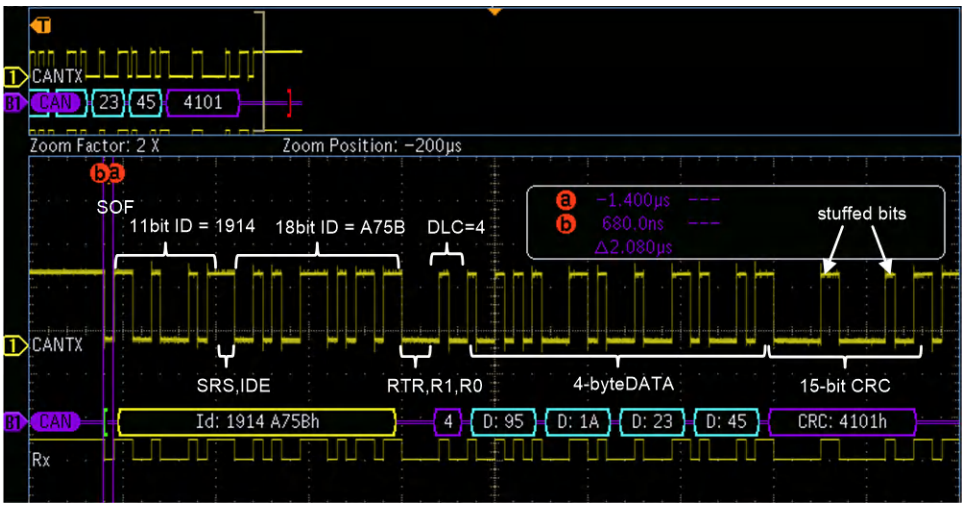

图 4-4. 未引起错误的基准数据帧（扩展 MSGID），DLC = 4

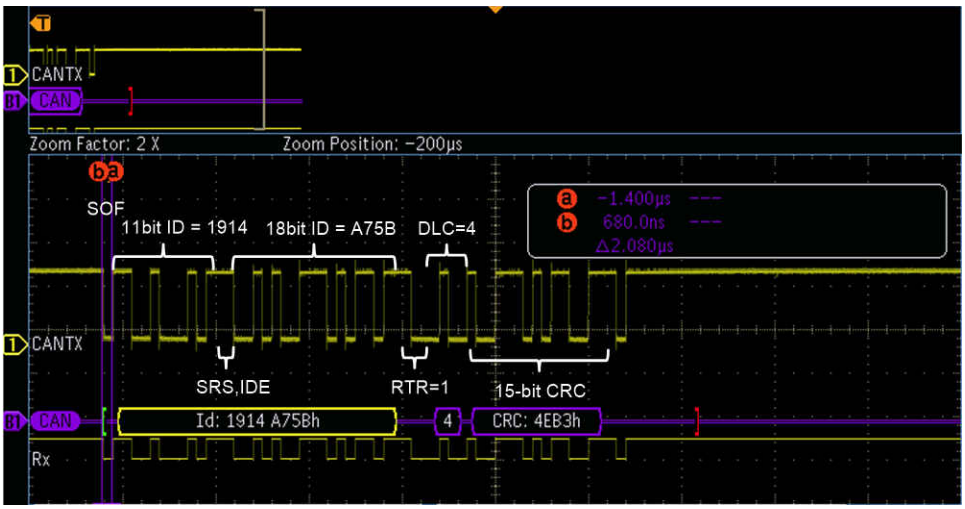

图 4-5. 未引起错误的基准远程帧（扩展 MSGID），DLC = 4

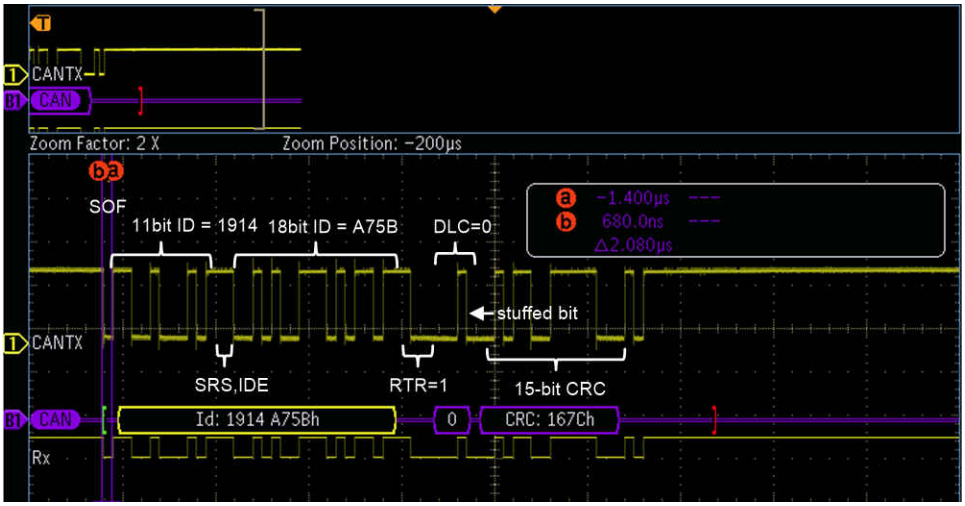

图 4-6. 未引起错误的基准远程帧（扩展 MSGID），DLC = 0

5 错误生成

本节介绍了如何生成不同类型的 CAN 总线错误。

5.1 GPIO/CCS 方法

图 5-1 演示了使用 GPIO/CCS 方法生成的错误。

- CCS 工程的 *#define* 标头中列出了可以生成的错误类型。
- 若要生成特定错误，请根据错误类型更新函数调用 *generateCANFrame()*。默认情况下，NO_ERR 是函数调用的第一个参数，这意味着不会产生错误。函数 *generateCANFrame()* 的最后两个参数分别是 BITNERR（位位置）和 DATABYTENUM（字节数）。“位位置”用于 MSGID_ERR、DATA_ERR 和 CRC_ERR。“字节数”仅与 DATA_ERR 一同使用。
- 错误定义的含义：
  - FF_SRS_ERR – 翻转 SRS 位
  - FF_IDE_ERR – 翻转 IDE 位
  - FF_RTR_ERR – 翻转 RTR 位
  - FF_R1_ERR – 翻转预留位“r1”
  - FF_R0_ERR – 翻转预留位“r0”
  - MSGID_ERR – 翻转 MSGID 中的“bit position”（位位置）
  - DATA_ERR – 翻转数据字段中“byte number”（字节数）指向的“bit position”（位位置）
  - CRC_ERR – 翻转 CRC 字段中的“bit position”（位位置）
  - STUFF_BITS_ERR – 当出现连续 5 个相同状态的位时，该函数不会生成填充位。

```
..
// CAN error definitions
// Use any of these defines for ERR_CFG
//
//No Error is generated for this define
//
#define NO_ERR                  0
//
//Bit 0/ CRC Error is generated for this Error define
//
#define FF_SRS_ERR              1
//
//Form Error is generated for this Error define
//
#define FF_IDE_ERR              2
//
//Form Error is generated for this Error define
//
#define FF_RTR_ERR              3
//
//Bit 0/ CRC Error is generated for this Error define
//
#define FF_R1_ERR               4
//
//Bit 0/ CRC Error is generated for this Error define
//
#define FF_R0_ERR               5
//
//Bit 0/ CRC Error is generated for this Error define
//
#define MSGID_ERR               6
//
//Bit 0/ CRC Error is generated for this Error define
//
#define DATA_ERR                8
//
//Bit 0/ CRC Error is generated for this Error define
//
#define CRC_ERR                 9
//
//Form Error is generated for this Error define
//
#define STUFF_BITS_ERR          10

frame_length = generateCANFrame(NO_ERR, CAN_FRAME, 0, 0, GPIOTX_PIN);
```

图 5-1. 使用 GPIO/CCS 方法生成的错误

5.2 LabVIEW 方法

图 5-2 演示了使用 LabVIEW 方法生成的错误。

- 在 *Error Generation*（错误生成）区域中，点击 *Forcing Error*（强制错误）条目以显示下拉选项。选择所需的错误类型。
- *BITnERR* 用于 MSGID_ERR、DATA_ERR 和 CRC_ERR，而 *DATA Byte Number*（数据字节数）仅用于 DATA_ERR。
- 错误生成下拉选项一目了然。
- 若要翻转 MSGID、DATA 和 CRC 字段中的位，请使用 *BITnERR* 中的位值。
- *BITnERR* 和 *DATA Byte Number*（数据字节数）字段中的值用于确定位位置和发生位翻转的字节数（如果在下拉菜单中选择了“DATA”）。

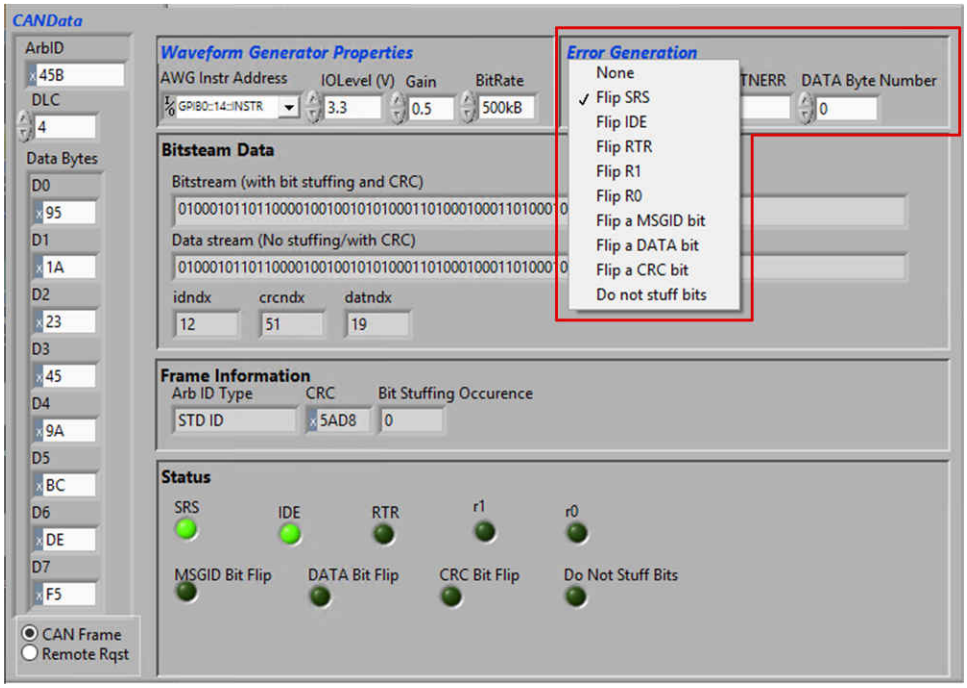

图 5-2. 使用 LabVIEW 方法生成的错误

6 仿真错误帧

本节介绍了仿真错误帧。下图显示了当 C2000 器件中的 CAN 模块配置为接收节点时 GPIO（或 LabVIEW）方法带来的各种错误。在接收模块上检查 CANES 寄存器中的最后一个错误代码 (LEC) 值，以说明 CAN 模块如何解释由错误引起的 CAN 帧。

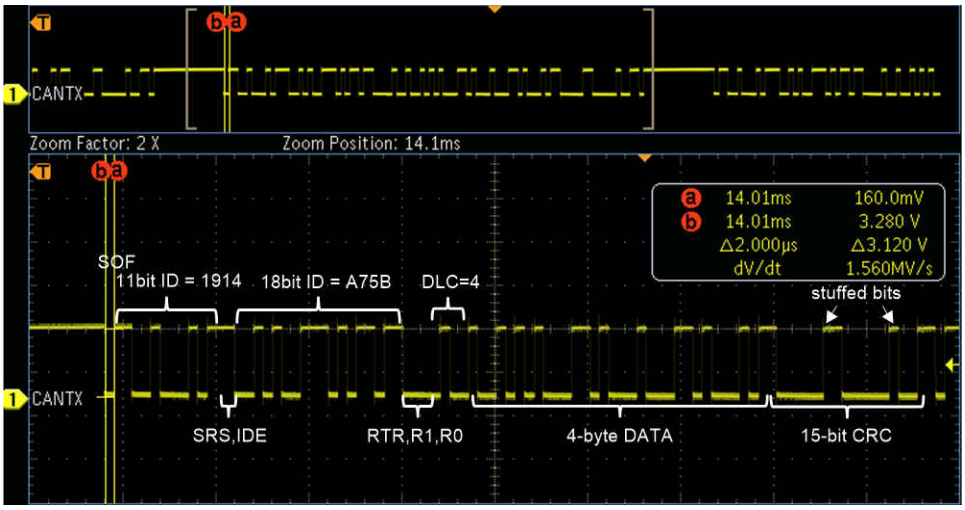

图 6-1. 未引起错误的扩展 ID（LEC = 0；无错误）

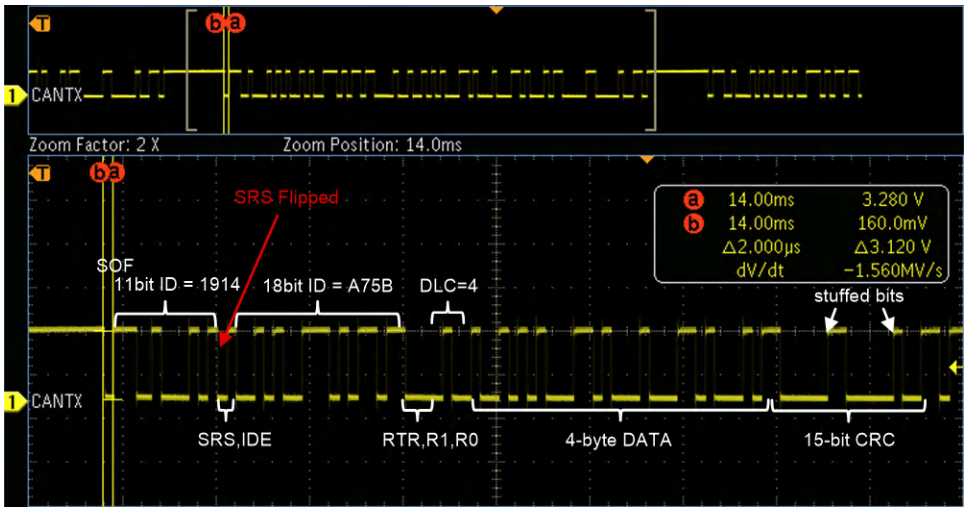

图 6-2. SRS 位发生翻转的扩展 ID（1 到 0）

备注

LEC = 6；标记了 CRC 错误而非格式错误，因为接收器接受 1 或 0 作为保留位。

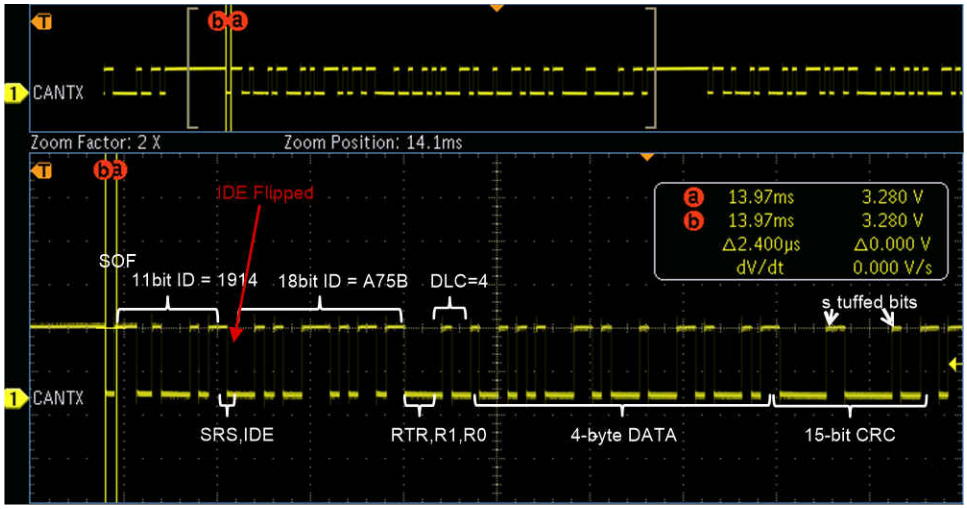

图 6-3. IDE 位发生翻转的扩展 ID

备注

LEC = 2；格式错误，因为接收器期望 IDE 为 1。

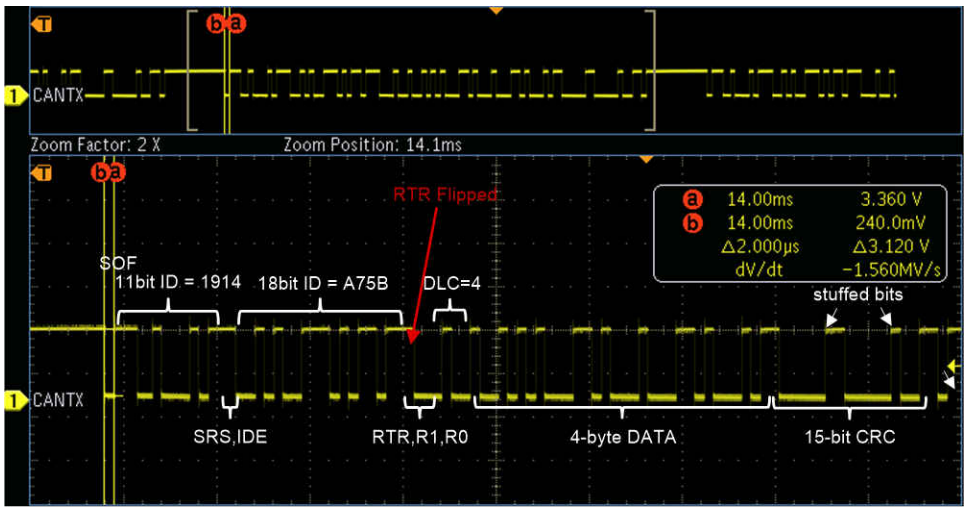

图 6-4. RTR 位发生翻转的扩展 ID

备注

LEC = 2；格式错误，因为接收器期望“远程帧”，但数据字节破坏了固定格式的位字段，如 CRC 定界符和 EOF。

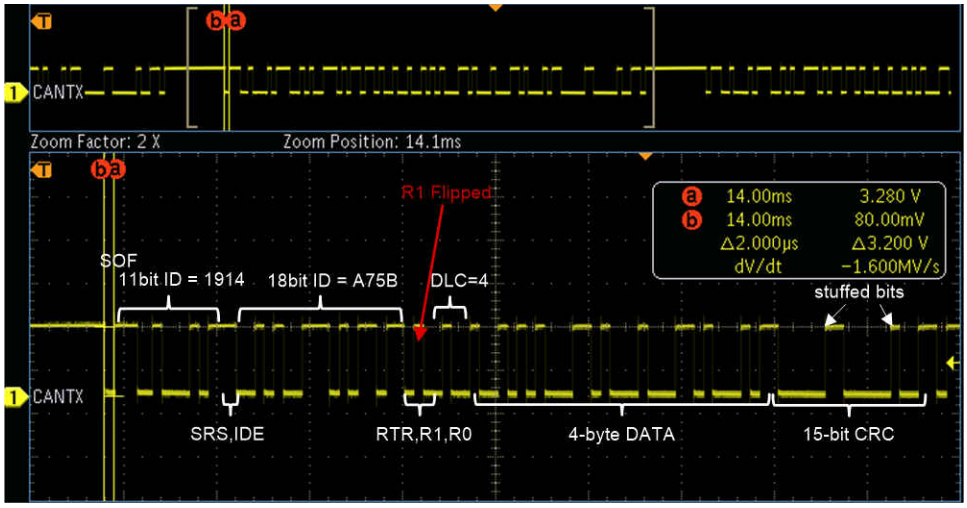

图 6-5. r1 位发生翻转的扩展 ID

备注

LEC = 6；CRC 错误，因为接收器将接受值为 0 或 1 的 r1。但是，计算得出的 CRC 不匹配。

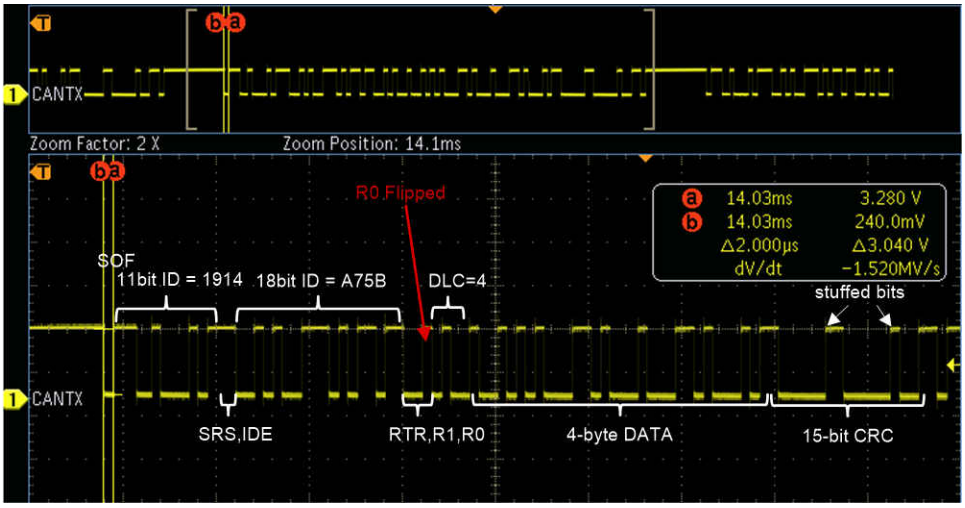

图 6-6. r0 位发生翻转的扩展 ID

备注

LEC = 6；CRC 错误，因为接收器将接受值为 0 或 1 的 r0。但是，计算得出的 CRC 不匹配。

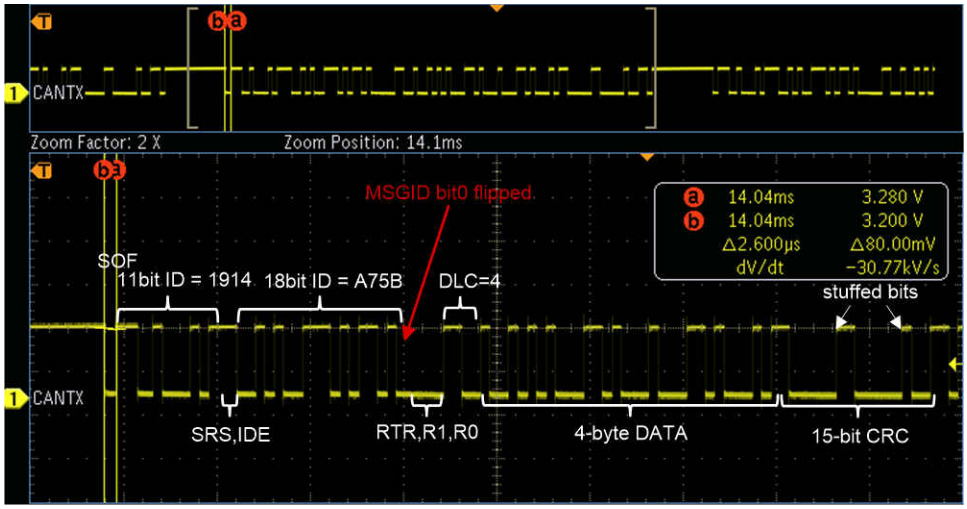

图 6-7. MSGID 的 Bit0 发生翻转的扩展 ID（LEC = 6；CRC 错误）

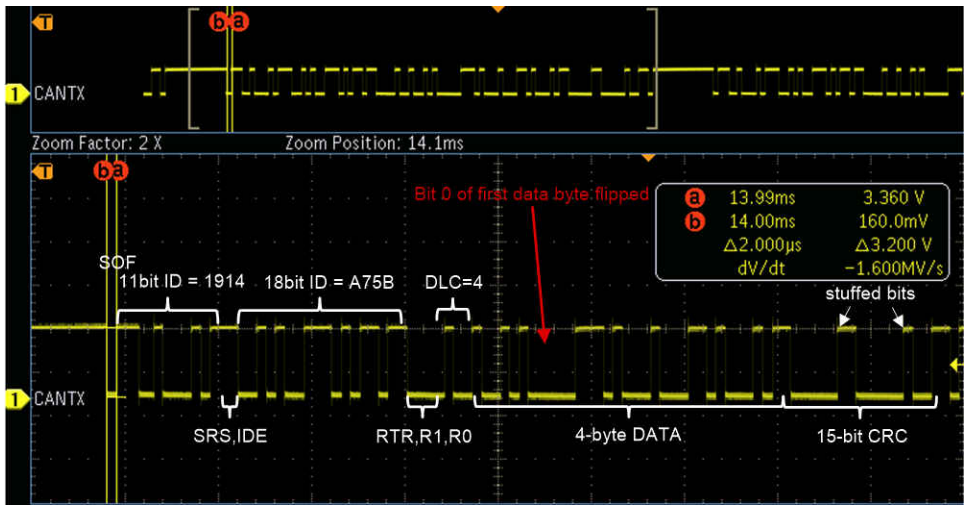

图 6-8. Data0 的 Bit0 发生翻转的扩展 ID（LEC = 6；CRC 错误）

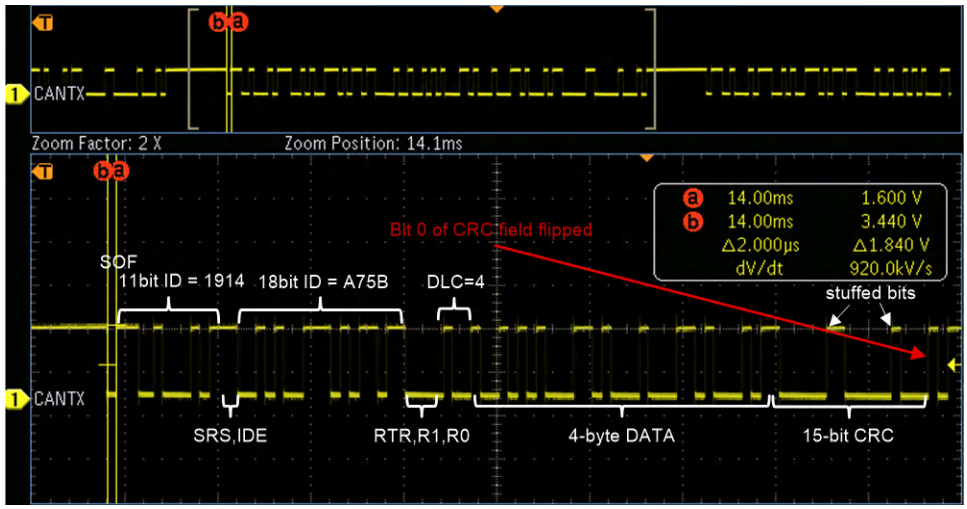

图 6-9. CRC 的 Bit0 发生翻转的扩展 ID（LEC = 6；CRC 错误）

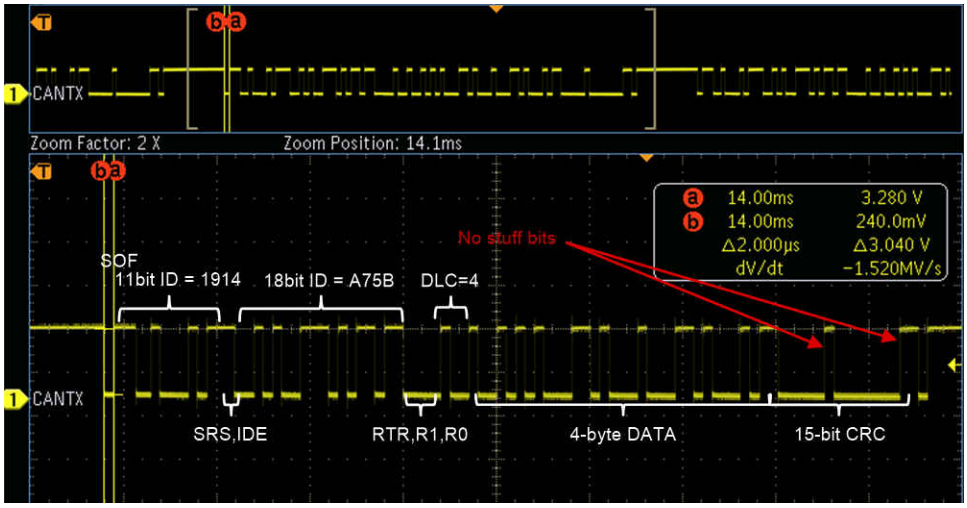

图 6-10. 没有位填充的扩展 ID（LEC = 1；填充错误）

7 参考文献

- 德州仪器 (TI)：*控制器局域网 (CAN) 简介*
- 德州仪器 (TI)：*控制器局域网物理层要求*
- 德州仪器 (TI)：*调试控制器局域网 (CAN) 物理层的基础知识*
- 德州仪器 (TI)：*用于 CAN 位定时参数的计算器*
- 德州仪器 (TI)：*3.3V CAN（控制器局域网）收发器概述*
- 德州仪器 (TI)：*使用无扼流圈收发器简化 CAN 总线实施*
- 德州仪器 (TI)：*CAN 总线连接的关键间距*
- 德州仪器 (TI)：*使用 TI 的 SN65HVD1050 收发器改善 CAN 网络安全性*
- 德州仪器 (TI)：*CAN 总线上的消息优先级反转*
- 德州仪器 (TI)：*使用片上零引脚振荡器的 Piccolo MCU CAN 模块操作*
- 德州仪器 (TI)：*《C2000 实时控制 MCU 外设参考指南》*
- 德州仪器 (TI)：*TMS320x28xx eCAN 的编程示例*
- 德州仪器 (TI)：*DCAN 模块的编程示例和调试策略*

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