

Application Note

使用 TMP1826 和 TM4C129x 微控制器为 1-Wire® 协议实现枚举功能



摘要

1-Wire® 总线用于要求低功耗通信和减少引脚数的系统。此应用报告介绍了 1-Wire® 通信协议、用于 Tiva™ C 系列 (TM4C129x) 微控制器中 1-Wire® 控制器的 TivaWare™ C 系列 API，以及使用二进制树搜索的枚举算法示例。此示例代码中枚举 1-Wire® 器件所用的器件是 TMP1826 具有 2Kbit EEPROM 的 $\pm 0.3^{\circ}\text{C}$ 精度数字温度传感器。

本应用报告中讨论的源代码可从[此处](#)下载。此外，为 TMP1826 单线温度传感器实现主机控制器可提供 Tiva C 系列微控制器的仿真主机示例，这些控制器使用 GPIO、UART 或 SPI 外设时没有专用的 1-Wire® 控制器。

内容

1 1-Wire® 简介.....	2
1.1 总线要求.....	2
1.2 供电.....	3
2 功能说明.....	3
2.1 1-Wire® 信号传输.....	3
2.2 1-Wire® 器件的地址格式.....	5
2.3 1-Wire® 总线中的典型通信流程.....	6
3 TivaWare C 系列中可用于 1-Wire® 模块的函数.....	6
4 枚举.....	7
4.1 旧搜索算法.....	7
4.2 快速搜索算法.....	9
5 总结.....	9
6 参考文献.....	9
7 修订历史记录.....	10

插图清单

图 1-1. 总线拓扑.....	2
图 2-1. 总线中至少有一个器件时的复位序列总线时序.....	3
图 2-2. 写入逻辑 0 总线时序.....	4
图 2-3. 写入逻辑 1 总线时序.....	4
图 2-4. 读取逻辑 1 总线时序.....	5
图 2-5. 读取逻辑 0 总线时序.....	5
图 2-6. 1-Wire® 总线中的典型通信流程.....	6
图 4-1. 使用二进制搜索树的枚举.....	8

表格清单

表 2-1. 复位信号传输说明和实现.....	3
表 2-2. 写入逻辑 0 位信号传输说明和实现.....	3
表 2-3. 写入逻辑 1 信号传输说明和实现.....	4
表 2-4. 读取位信号传输说明和实现.....	4
表 2-5. 1-Wire® 目标器件的地址格式.....	5
表 4-1. 搜索算法.....	7

商标

Tiva™ and TivaWare™ are trademarks of Texas Instruments.

1-Wire® is a registered trademark of Maxim Integrated Products, Inc.

所有商标均为其各自所有者的财产。

1 1-Wire® 简介

1-Wire® 通信总线旨在连接温度传感器和非易失性存储器器件（例如 **TMP1826**），可通过一根线支持通信和电力输送。此系统适用于低速和低功耗通信器件。共有两种可用速度模式：标准速度和过驱速度。标准速度可达到的数据速率通常为 8.33 kbit/s，而过驱速度通信模式可达到 90 kbits/s。

此协议使用单一数据线进行一个器件与另一个器件之间的数据传输。该总线为半双工，因此数据可以双向（但不能同时双向）移动。如需要，还可额外使用一根线给器件上电。

该协议支持总线有一个（单点）或多个目标器件（多点）。总线上还有单一控制器，用于控制总线的信息传输。控制器会启动数据线上的所有传输。数据只能在控制器和目标器件之间传输，数据不能在器件之间传输。

此协议不需要时钟，因为每个目标器件均由内部振荡器计时，该振荡器与总线的下降沿同步。传输一个字节时，首先传输最低有效位。

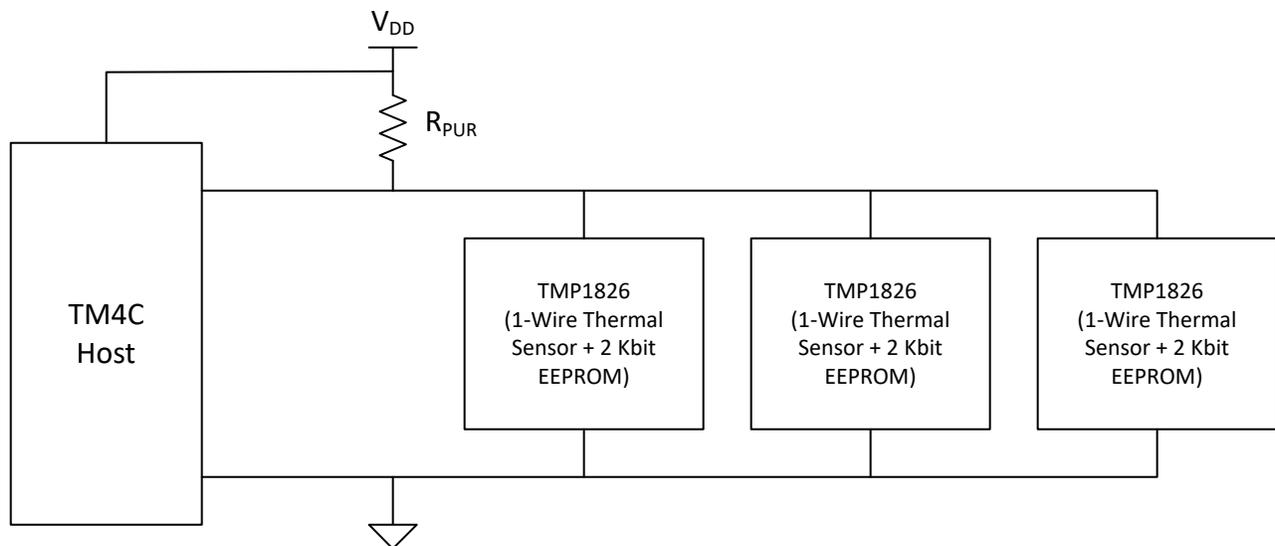


图 1-1. 总线拓扑

1.1 总线要求

每个输出引脚必须为开漏，还必须将弱上拉电阻连接到信号，如果至少有一个器件将总线驱动为低电平，总线将被驱动为低电平。此协议支持在总线上的两个器件间传输数据，而其他器件处于空闲状态。上拉的强度由用户根据以下因素决定：

- 如果不需要高数据速率，且系统的迹线长度较短，则由外部供电的器件的上拉值可能在 10K 或以上。
- 如果数据速率高（特别是在过驱模式下），或系统的迹线长度较长，由外部供电的器件的上拉值可能少于 1K。
- 寄生供电的器件在被选中后，可能需要运行驱动器，使终端设备有足够的能源执行终端操作。

1.2 供电

TMP1826 这类目标器件可通过两种模式供电：

- 电源供电：目标器件上的专用 V_{DD} 引脚用于为总线供电。如果应用可在目标器件附近支持附加电源或探头解决方案中的附加电缆，可使用此拓扑。
- 总线供电：在此模式下，器件由数据线供电。所有目标器件都必须包含内部电容器，用于在总线空闲时存储能源，并由上拉电阻器上拉。存储的能源然后可由器件在活动通信期间使用。

2 功能说明

2.1 1-Wire® 信号传输

数据线上可能涉及的四种信号传输类型为：

- 具有复位脉冲和复位应答 (ATR) 的复位序列：复位脉冲用于将所有器件置于已知状态。目标器件通过发送 ATR 信号确认其存在，方法是将线路保持为低电平。主机控制器对总线进行采样，如果总线为低电平，则至少存在一个目标器件。

表 2-1. 复位信号传输说明和实现

运行	说明	实施
复位	复位 1-Wire® 总线目标器件并准备向它们发出命令。	将总线驱动为低电平 480 μ s，以复位所有器件。主机将在接下来的 240 μ s 内对总线进行采样，同时以复位应答 (ATR) 为目标。

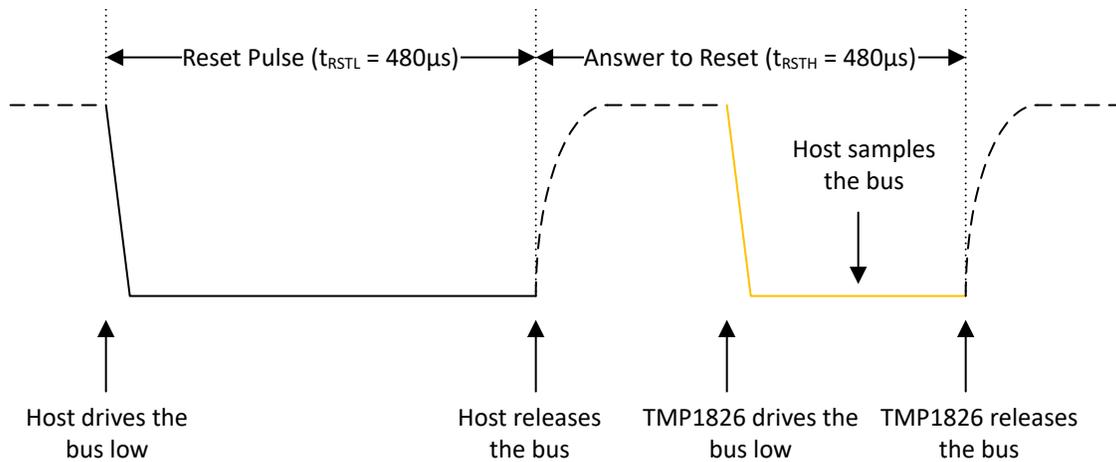


图 2-1. 总线中至少有一个器件时的复位序列总线时序

- 在总线中写入逻辑 0

表 2-2. 写入逻辑 0 位信号传输说明和实现

运行	说明	实施
写入逻辑 0	发送 0 位到 1-Wire® 目标器件	将总线驱动为低电平 60 μ s

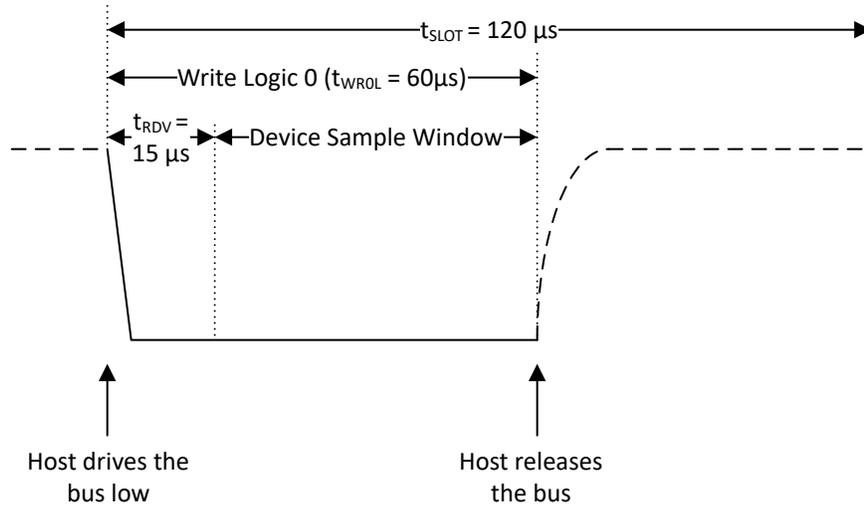


图 2-2. 写入逻辑 0 总线时序

- 在总线中写入逻辑 1

表 2-3. 写入逻辑 1 信号传输说明和实现

运行	说明	实施
写入 1 位	发送 1 位到 1-Wire® 目标器件	将总线驱动为低电平 < 15μs。典型时间约为 6μs。释放总线直至下降沿后 60μs。

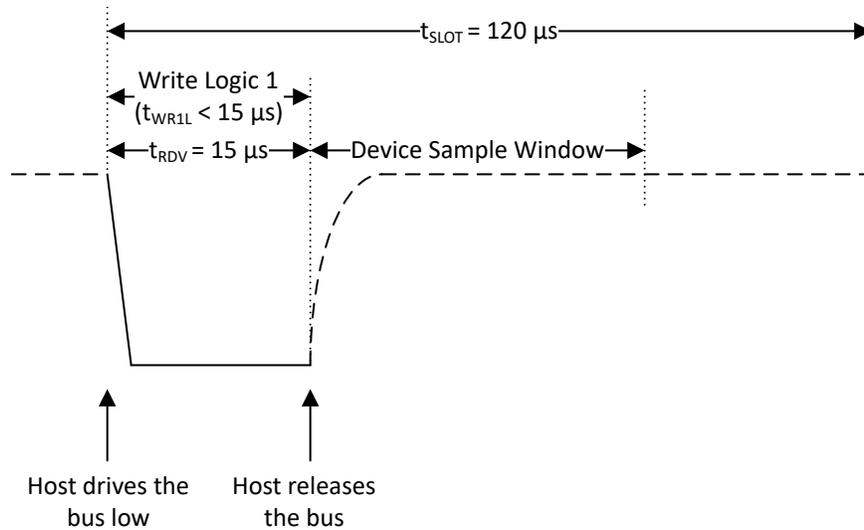


图 2-3. 写入逻辑 1 总线时序

- 读取位：从目标器件读取一位。读取位信号传输与写入“1”信号传输类似，区别在于主机控制器执行读取而不是写入操作。

表 2-4. 读取位信号传输说明和实现

运行	说明	实施
读取位	从 1-Wire® 目标器件读取一位	将总线驱动为低电平，从 1μs 到 15μs。在下降沿后 15μs 时对总线进行采样，以从目标器件读取位。

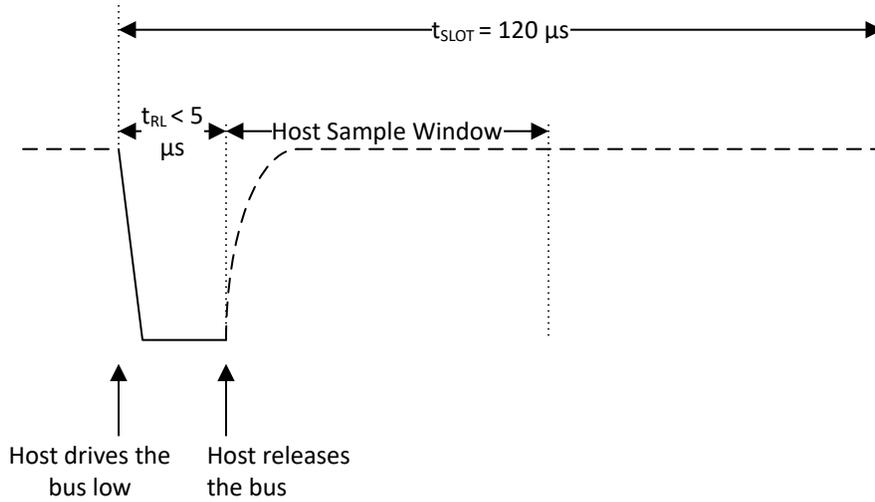


图 2-4. 读取逻辑 1 总线时序

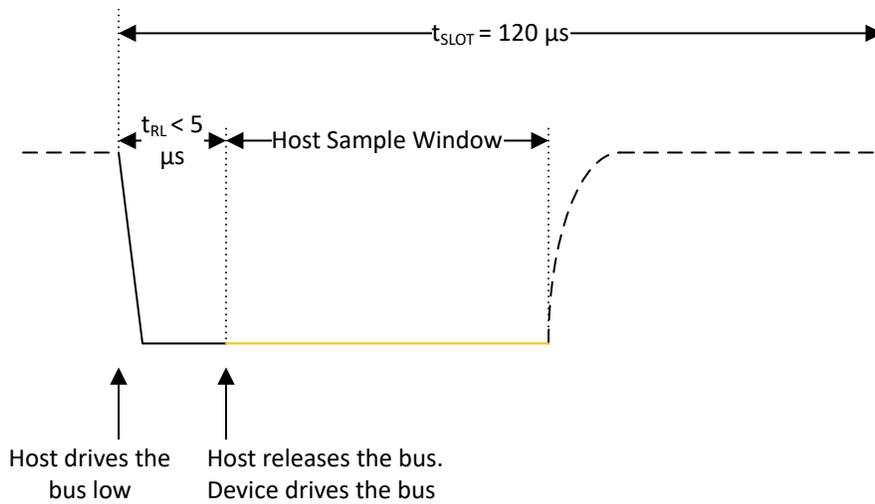


图 2-5. 读取逻辑 0 总线时序

2.2 1-Wire® 器件的地址格式

德州仪器 (TI) 的 1-Wire® 目标器件中在器件中存储着唯一一个 64 位地址，也称为器件地址。该地址的 8 个最低有效位代表器件的系列代码。接下来的 48 位为器件的序列号。8 个最高有效位提供根据 56 个最低位生成的 CRC。

表 2-5. 1-Wire® 目标器件的地址格式

	64 位器件地址	
8 位 CRC	48 位序列号	8 位系列代码
63:56 位	55:8 位	7:0 位

备注

CRC 用于检查数据是否已正确接收。Tiva C 系列器件未在硬件中实现 CRC，因此需要软件实现。

2.3 1-Wire® 总线中的典型通信流程

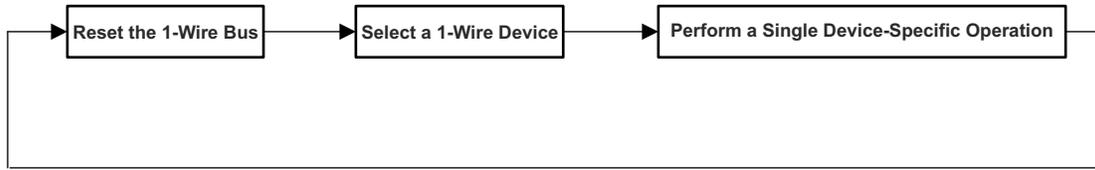


图 2-6. 1-Wire® 总线中的典型通信流程

- 从复位序列开始
- 如果主机必须确定总线中有哪些目标器件，则应执行搜索，以检测器件的 64 位器件地址。
- 在对器件执行操作之前，必须使用 ROM 命令对器件进行配置和/或选择。一些可用的功能 ROM 命令为：
 - **Read Address [0x33]**：只用于总线中有单个器件的情况。此命令用于读取总线中出现的唯一器件的 64 位器件地址。
 - **Match Address [0x55]**：此命令后跟 64 位器件地址，选择具有匹配地址的器件。所有其他器件会等待下次复位脉冲。
 - **Search Address [0xF0]**：需要用此命令来获取多个器件的 64 位器件地址，它通知器件主机将要进行搜索。然后将进行搜索，方法是从器件读取 ROM 编号的一位及其补码，并发回一个相应位。有关详细信息，请参阅节 4。如果目标器件的位与主机发送的位相同，则保持活动状态，其他器件等待下次复位
 - **Skip Address [0xCC]**：可以确定器件地址，主机无需知道 64 位器件地址。如果要为所有器件提供通用命令，此命令会很有用。
 - **Overdrive Skip Address [0x3C]**：此命令仅用于单点。此命令与 Skip Address 命令相同，区别在于只有可以在过驱模式下运行的器件保持运行，并进入过驱模式。无法在过驱模式下运行的器件等待下次复位。
 - **Overdrive Match Address [0x69]**：此命令与 Match Address 命令相同，区别在于只有可以在过驱模式下运行的器件才能匹配。所有其他器件会等待下次复位。
- 选择所需器件后，可发出特定于器件的命令，以执行所需操作。
- 通常在每次操作后，将发出复位脉冲。

3 TivaWare C 系列中可用于 1-Wire® 模块的函数

备注

ui32Base 应包含 1-Wire® 模块的基址。

OneWireBusReset(uint32_t ui32Base)；此函数的作用是在 1-Wire® 总线中发出复位命令；它不会等待复位完成。

OneWireBusStatus(uint32_t ui32Base)；此函数的作用是检索总线状态，用于确定总线是否已准备好执行操作。如果主机正在执行任何操作，状态为忙碌；如果主机未执行操作，状态为空闲。

OneWireDataGet(uint32_t ui32Base, uint32_t *pui32Data)；此函数的作用是等待事务（如有）完成并从 1-Wire® 接口检索数据。数据存储在 pui32Data 提供的地址中。

OneWireDataGetNonBlocking(uint32_t ui32Base, uint32_t *pui32Data)；此函数的作用是从 1-Wire® 接口检索数据。如有正在进行的事务，将返回 0xffffffff。数据存储在 pui32Data 提供的地址中。

OneWireInit(uint32_t ui32Base, uint32_t ui32InitFlags)；此函数的作用是初始化 1-Wire® 模块。ui32InitFlags 参数包含初始化标志。

OneWireIntClear(uint32_t ui32Base, uint32_t ui32IntFlags)；此函数的作用是清除 1-Wire® 模块中必要的中断源。

OneWireIntDisable(uint32_t ui32Base, uint32_t ui32IntFlags)；此函数的作用是禁用 1-Wire® 模块中必要的中断源。

OneWireIntEnable(uint32_t ui32Base, uint32_t ui32IntFlags)；此函数的作用是启用 1-Wire® 模块中必要的中断源。

OneWireIntRegister(uint32_t ui32Base, void(*pfnHandler)(void)); 此函数的作用是为 1-Wire® 模块注册一个中断处理程序。

OneWireIntUnregister(uint32_t ui32Base); 此函数的作用是为 1-Wire® 模块取消注册一个中断处理程序。

OneWireIntStatus(uint32_t ui32Base, bool bMasked); 此函数的作用是获取当前中断状态。如果 bMasked 为 true (真), 则可获得屏蔽中断状态。如果 bMasked 为 false (假), 则可获得原始中断状态。

OneWireTransaction(uint32_t ui32Base, uint32_t ui32OpFlags, uint32_t ui32Data, uint32_t ui32BitCnt); 此函数的作用是执行总线中的 1-Wire® 协议传输。ui32BitCnt 用于配置要发送或接收的位数。写入的数据由 ui32Data 指定。ui32OpFlags 参数用于定义要执行的操作 (复位、读取、写入)。

备注

有关 API 的更多信息可在 ([TivaWare™ 外设驱动程序库用户指南](#)) 中找到。

4 枚举

如果总线中有多个器件, 主机必须知道 64 位器件地址, 以匹配必要的器件。执行枚举可以获得器件在总线中的 64 位地址, 从而知道有哪些可用的器件类别。

搜索地址操作的传统模式在节 4.1 中介绍, 需要迭代执行两个步骤:

1. 从器件中读取两位 (实际位的值及其补码)
2. 在枚举算法中写入定义搜索路径的适当位。相应位等于写入位的器件保持有效状态, 而其余器件进入空闲状态, 等待下一个复位命令。

全部 64 个位都需要历经此周期, 在迭代结束时可组成 64 位器件地址。

TMP1826 支持快速搜索操作 (在节 4.2 中介绍), 显著简化了 64 位器件地址

4.1 旧搜索算法

搜索算法使用二进制搜索树。在每个节点, 算法可选择由“0”或“1”代表的路径。表 4-1 中介绍了在第 1 步中得到的两个位和第 2 步所选路径之间的关系。

表 4-1. 搜索算法

实际位读取值	补码位读取值	结论	要选择的途径
0	0	多个器件有一个对应的 0 位和一个对应的 1 位	这属于冲突情况, 需要决定选择哪条路径
1	0	只有一个器件在对应位的位置为 1	选择路径 1
0	1	只有一个器件在对应位的位置为 0	选择路径 0
1	1	总线中无器件	结束搜索

只有出现冲突时才需要做出决定。在其他三种情况下, 要选择的途径已进行定义。图 4-1 展示了 4 位搜索的算法流程。软件中的以下变量是搜索的关键。

- *i32LastConflictZeroBitNumber* 变量存储上一冲突节点的迭代次数, 查找当前 ROM 编号时所选路径为“0”。
- *li32ConflictBitNumber* 变量存储上一个 ROM 编号的 last_conflict_zero 值。
- *ui32BitNumber* 变量提供 ROM 编号中所考虑位的位置。

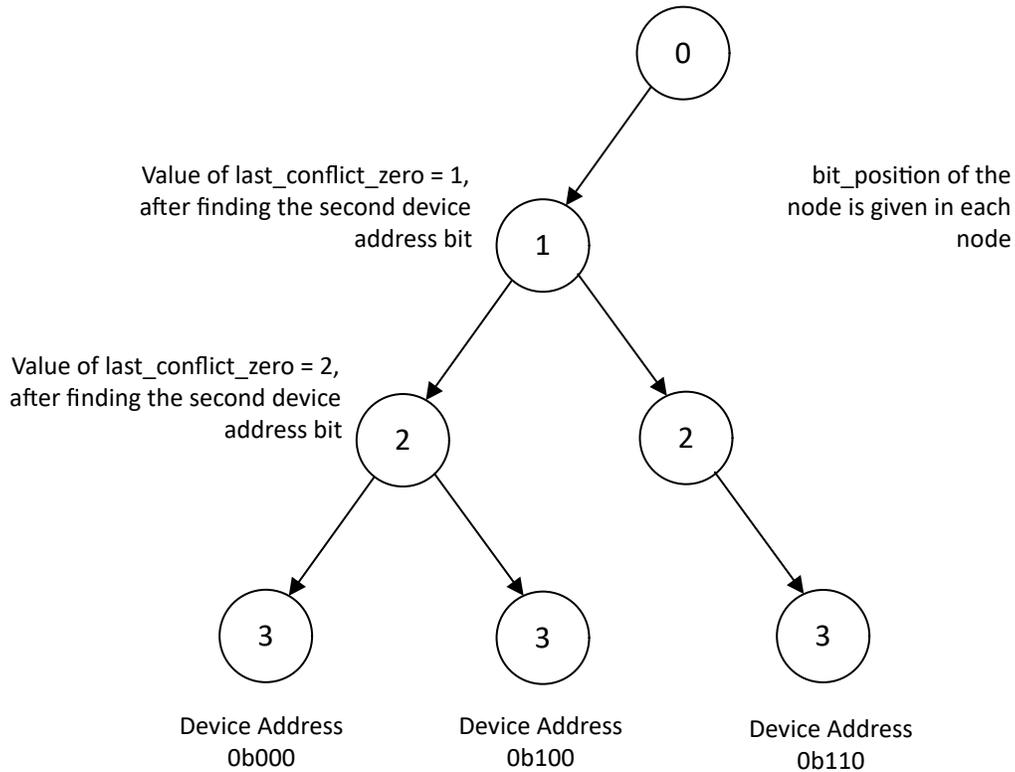


图 4-1. 使用二进制搜索树的枚举

4.1.1.3 位搜索算法的步骤

1. 复位总线并寻找 ATR 响应。如果总线中无器件，则结束流程。
2. 如果收到 ATR 响应，则发送 Search ROM 命令。
3. 从总线中的器件读取一位。
4. 从器件中读取第 3 步中的位的补码。
5. 检查是否两个位都为 1。如果为“是”，则结束此流程。如果为“否”，则继续。
6. 检查第一个读取位是否为 0，第二个读取位是否为 1。如果为“是”，则在总线中写入 0 并转至第 14 步。如果为“否”，则继续。
7. 检查第一个读取位是否为 1，第二个读取位是否为 0。如果为“是”，则转至第 9 步。如果为“否”，则继续。
8. 检查 *ui32BitNumber* 是否等于 *i32ConflictBitNumber*。如果为“是”，则继续。如果为“否”，则转至第 10 步
9. 在总线中写入 1。转至第 14 步。
10. 检查 *ui32BitNumber* 是否小于 *i32ConflictBitNumber*。如果为“是”，则继续。如果为“否”，则转至第 12 步。
11. 检查上一 ROM 编号的 *ui32BitNumber* 中的位是否等于 1。如果为“是”，则转至第 9 步。如果为“否”，则继续。
12. 在总线中写入 0。
13. 用 *ui32BitNumber* 更新 *i32LastConflictZeroBitNumber* 的值。
14. 检查 *ui32BitNumber* 是否等于 63。如果为“是”，则转至第 1 步。如果为“否”，则转至第 3 步。

4.2 快速搜索算法

快速搜索算法使用 TMP1826 器件进行仲裁。主机发出 SKIP ADDRESS 命令，为 ARB_MODE 写入配置寄存器 - 2 位，为 11b。执行上一步后，它可发出 SEARCH ADDRESS 命令，读取一位器件地址。总线中的所有器件都传输各自的地址：逻辑 1 或 0，并监控 SDQ 线路，以读回传输的内容。由于逻辑 0 是主导值，发送逻辑 1 的器件将离开总线，不再参与当前的地址搜索操作迭代。

主机将从赢得总线仲裁的器件读取下 1 位器件地址。此流程会继续，直到只有一个器件成功发送其 64 位器件地址，这时器件会在其状态寄存器中设置一个成功标志。主机然后会发送另一条 SEARCH ADDRESS 命令，除了已完成枚举的器件之外，所有其他器件均会重新加入地址枚举的流程。

相比较早的 1-Wire® 器件的旧搜索算法，快速搜索算法的速度加快 3 倍，节省了内存并降低了实现二进制树搜索所需的处理成本。有关所述方法和重新枚举的更多详情，请参阅[具有 2Kbit EEPROM 的单线 ±0.3°C 精度温度传感器数据表](#)。

5 总结

此应用报告概述了 1-Wire® 协议，并提供了有关主机如何使用二进制树搜索来识别总线中 TMP1826 的示例。

6 参考文献

- 德州仪器 (TI) : [TMP1826 具有 2Kbit EEPROM 的单线、±0.3°C 精度数字温度传感器](#)
- 德州仪器 (TI) : [TivaWare™ 外设驱动程序库用户指南](#)
- 德州仪器 (TI) : [为 TMP1826 单线温度传感器实现主机控制器](#)
- [TivaWare™ C 系列软件](#)

7 修订历史记录

Changes from Revision D (June 2022) to Revision E (January 2025) Page

- 更新了整个文档中的表格、图、商标和交叉参考的编号格式..... 1
 - 将文档标题从“使用 TM4C129x 微控制器实现 TMP1826 的单线枚举”更改为“使用 TMP1826 和 TM4C129x 微控制器实现 1-Wire® (单线) 协议的枚举” 1
-

Changes from Revision C (January 2018) to Revision D (June 2022) Page

- 通篇将“通用单线器件”更改为“TMP1826”。 2
-

重要通知和免责声明

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 产品的相关应用。严禁以其他方式对这些资源进行复制或展示。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
版权所有 © 2025，德州仪器 (TI) 公司