

Application Note

I2C：什么是自动递增功能？

Duy Nguyen

摘要

本应用手册介绍不同类型的自动递增功能、如何执行递增以及使用自动递增功能的优势。

内容

1 什么是自动递增？	2
2 自动递增有何用途？	3
3 是否所有 I2C 器件都支持自动递增？	4
4 是否有不同类型的自动递增？	4
5 是否需要启用自动递增？	5
6 参考资料	6

插图清单

图 1-1. I2C 单次写入	2
图 1-2. I2C 单次读取	2
图 2-1. I2C 自动递增突发写入	3
图 2-2. I2C 自动递增突发读取	3
图 4-1. 自动递增 IO 扩展器	4
图 4-2. 示波器示例写入突发	5

商标

所有商标均为其各自所有者的财产。

1 什么是自动递增？

在 I2C 中，自动递增是指在不执行停止条件的情况下读取或写入连续的字节时寄存器地址自动增加到下一个字节（例如从寄存器 1 到寄存器 2）。

以前在 I2C 中将数据写入寄存器时，用户需要发送 I2C 目标地址，后跟寄存器编号，再然后是需要写入寄存器的数据。图 1-1 所示为 I2C 单个写入事务的一般示例。如果用户需要向 I2C 器件中的其他寄存器进行写入，则需要额外的写入事务。此过程图 1-1 与类似，但需要多个事务而不是一个事务。

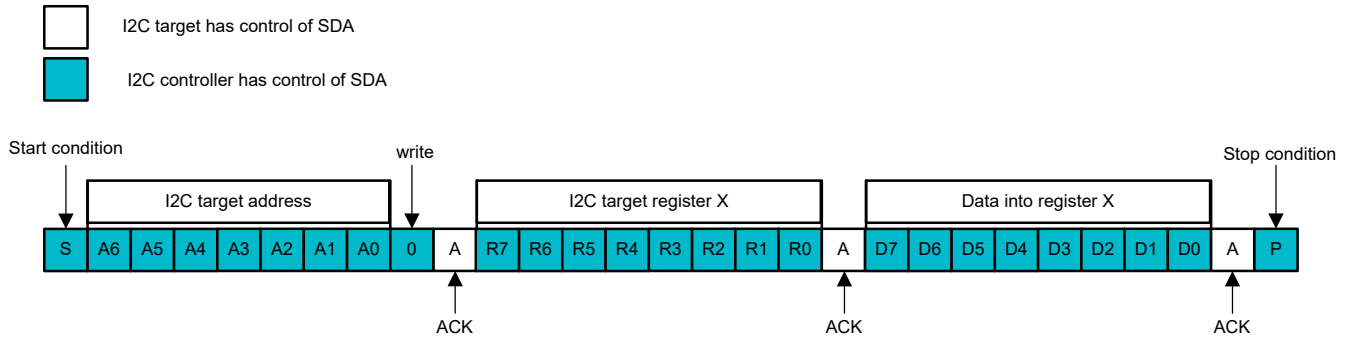


图 1-1. I2C 单次写入

对于读取事务，I2C 事务涉及更多，因为针对读取，需要事先写入以设置读取寄存器的指针或地址。图 1-2 展示了单个读取事务的一般示例，其中首先进行写入以设置需要读取的寄存器。

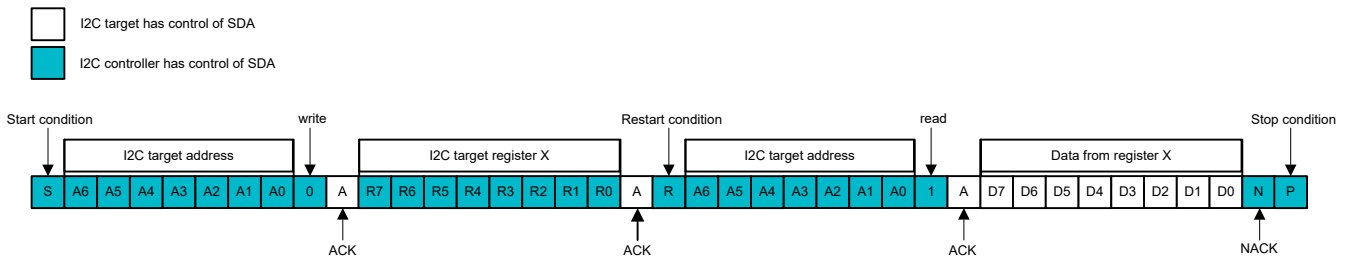


图 1-2. I2C 单次读取

2 自动递增有何用途？

使用自动递增功能允许用户在无需执行停止条件的情况下读取或写入数据。这样可以在 I2C 总线的初始化序列期间节省 I2C 控制器的时间和计算周期。图 2-1 展示了使用自动递增功能的突发写入示例。该示例展示了向 I2C 目标写入 4 个数据字节。每个字节的数据都从第一个寄存器（寄存器 0）开始写入堆栈中的下一个寄存器。发送到 I2C 目标的总字节数为 6。如果不使用自动递增功能，则执行同一事务所需的字节数总共可能为 12。此事务可以将所需的时间减半（针对该示例）。

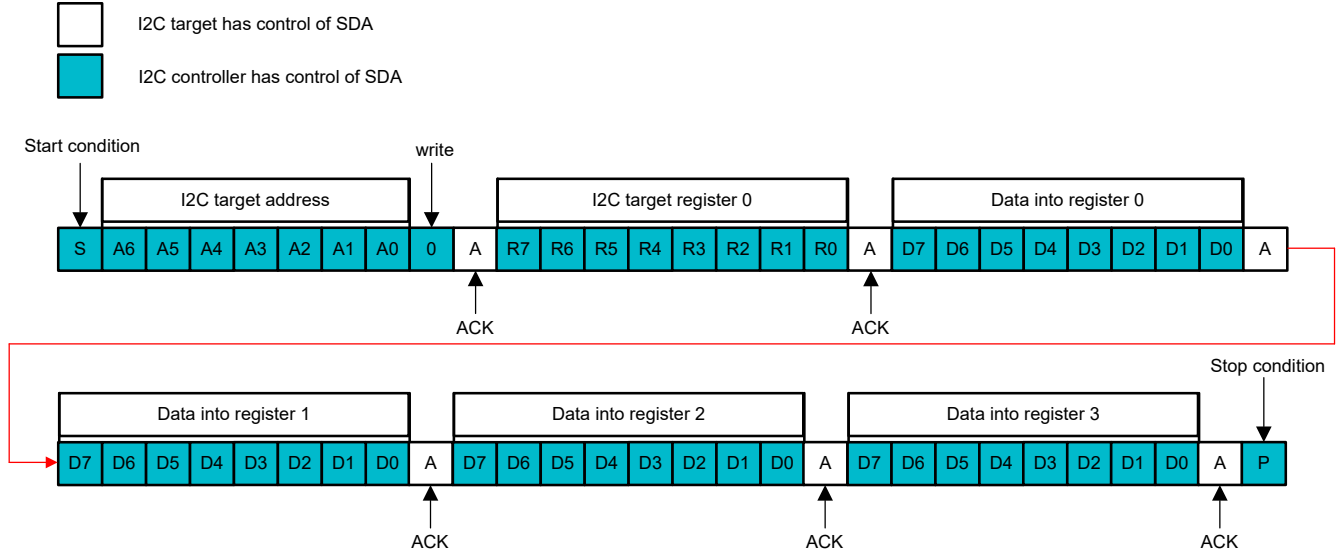


图 2-1. I2C 自动递增突发写入

图 2-2 展示了具有自动递增功能的读取事务示例。该示例显示起始寄存器为寄存器 5（不需要从寄存器 0 开始），并递增到寄存器 9。从 I2C 器件读取的总字节数为 5，执行此过程所需的字节数为 8。如果 I2C 控制器（或用户）选择了不使用自动递增功能，则需要额外设置 3 个字节来读取下一个寄存器（该示例中为寄存器 6）。此递增总共用到 8 个字节来读取 2 个字节的数据。由此可以清楚地看出，自动递增功能可为用户提供更高效的编程方法。

还有一点需要注意，在该示例中，如果在从寄存器 9 接收数据后发出了 ACK，并且寄存器 10 中有数据需要收集，则理论上 I2C 控制器可能会继续从该 I2C 目标读取字节。实际上，读取事务可以持续到控制器结束事务为止。

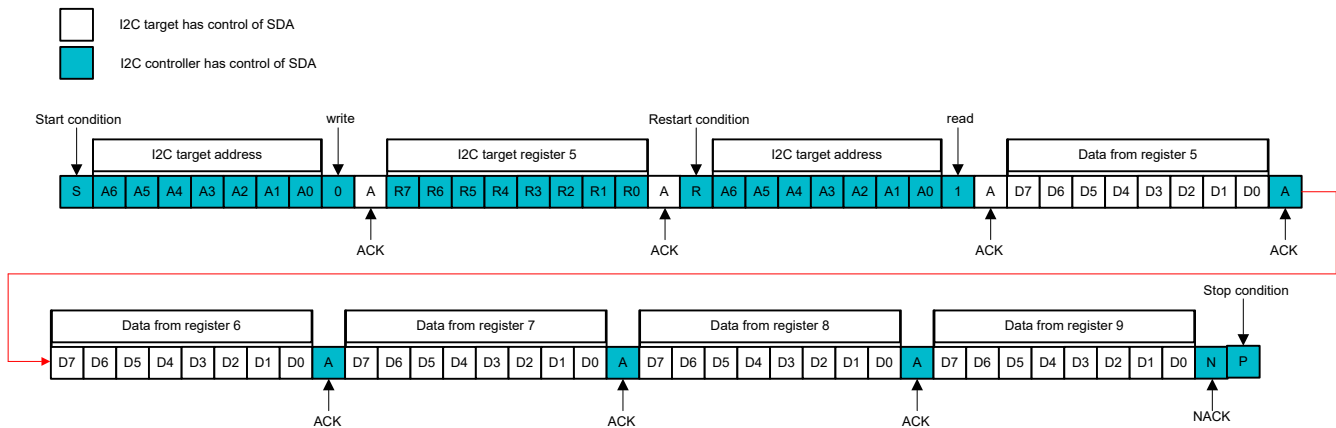


图 2-2. I2C 自动递增突发读取

3 是否所有 I2C 器件都支持自动递增？

不是，只有设计为包含自动递增功能的器件才支持该功能。自动递增功能在最初设计时并不属于 I2C 标准的一部分，旧的 I2C 器件可能不包括或不允许对寄存器进行连续写入或读取。

4 是否有不同类型的自动递增？

是的，有些器件只能在一组特定的寄存器内自动递增。例如，TI I2C IO 扩展器便是如此。对于具有 2 个端口（16 位扩展）的 IO 扩展器，可在一组端口寄存器内进行自动递增。图 4-1 展示了使用 TCAL6416 等 16 位 IO 扩展器的这种特殊自动递增情况的示例。如果在将寄存器设置为 0x02h（输出端口 0 寄存器）后执行连续写入，则会使得向寄存器写入的数据在寄存器 0x02h（输出端口 0 寄存器）和 0x03h（输出端口 1 寄存器）之间交替。

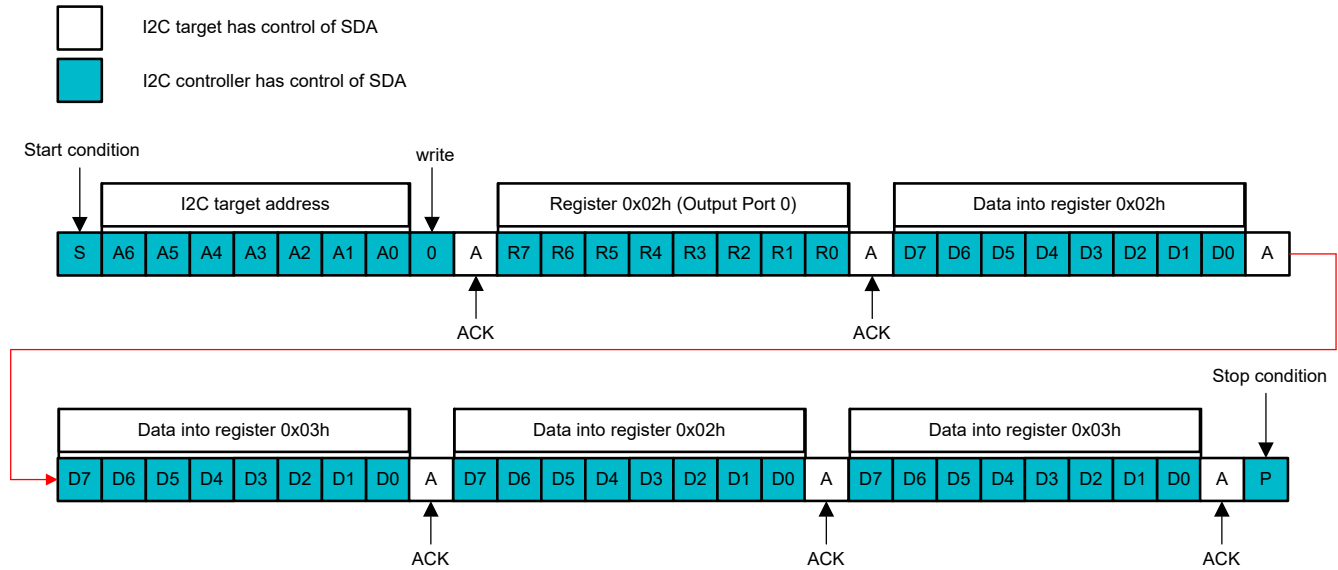


图 4-1. 自动递增 IO 扩展器

由于用户无法一次性向所有寄存器进行写入，因此这种类型的自动递增功能会产生一个需要多个 I2C 事务（具有启动/重启和/或停止条件）的初始化序列。虽然从初始化的角度来看这是一个缺点，但这种自动递增方法有一个优点。由于只能从一个设定端口进行写入或读取，因此可以在接收/输出有用数据时获得更好的延迟。

图 4-2 所示为一个 4 位 IO 扩展器 (TCA9536) 用于切换位 2 和位 3。自动递增功能用于在每次 ACK 后立即切换这些位。由于该器件只有一个端口，因此自动递增功能会持续递增到同一寄存器（输出寄存器 0x01h）中，因此每次写入新数据时，该器件都会在每次 ACK 时更新。这种方法降低了输出切换的延迟，因为发送寄存器字节 (0x01h) 后，该器件可以自由更改输出状态。另一种方法是始终通过发送器件地址、寄存器以及启动和停止条件之间的数据来向器件执行写入操作。

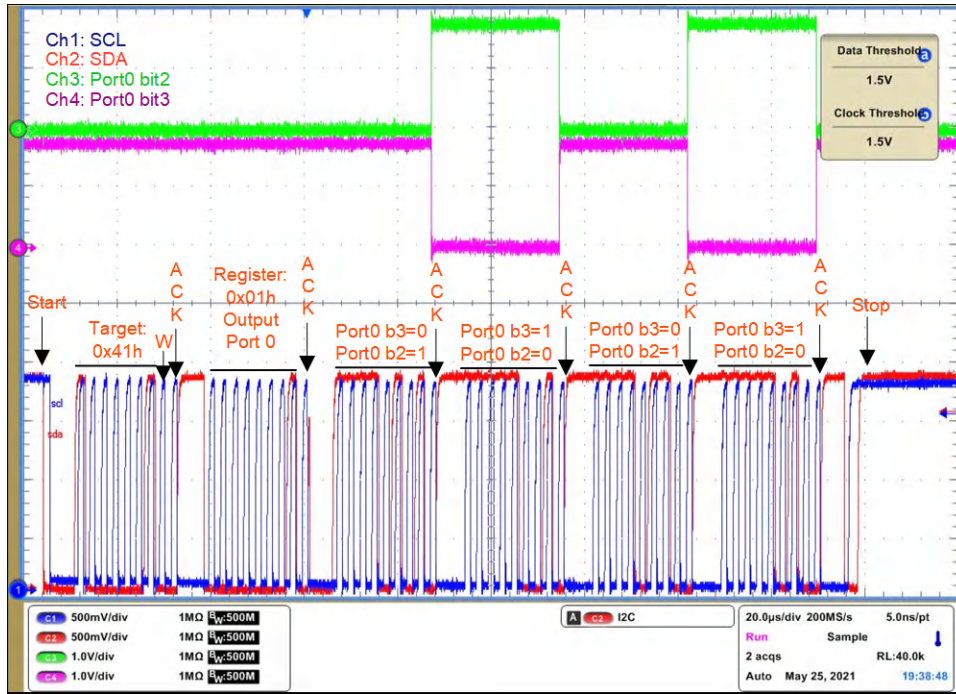


图 4-2. 示波器示例写入突发

从读取角度来看，需轮询 IO 扩展器输入的应用可以使这种自动递增变得高效，因为器件在返回到相关寄存器之前不需要在所有寄存器中进行递增。

5 是否需要启用自动递增？

视情况而定。在前面的示例中，TI 产品系列中的 IO 扩展器全部设计为在数据帧上发生连续写入或读取时自动递增或循环。其他 I2C 器件需要对某个寄存器进行写入来设置一个位，才能启用自动递增功能。这方面的一个相关示例是 TI 的键盘扫描仪 (TCA8418/TCA8418E)，必须将其寄存器 0x01h (配置寄存器) 中的位 7 设置为 1 才能启用该功能。与 I2C IO 扩展器不同，该器件的自动递增功能可在所有寄存器中完全循环，由于该器件总共有 46 个寄存器，因此节省了大量时间。

6 参考资料

- 德州仪器 (TI), [TCA8418 具有集成 ESD 保护的 I2C 控制型键盘扫描 IC](#) 数据表。
- 德州仪器 (TI), [TCA8418E 具有集成 ESD 保护的 I2C 控制型键盘扫描 IC](#) 数据表。
- 德州仪器 (TI), [TCA9536 具有配置寄存器的远程 4 位 I2C 和 SMBus I/O 扩展器](#) 数据表。
- 德州仪器 (TI), [TCAL6416 具有中断输出、复位和敏捷 I/O 配置寄存器的 16 位转换 I2C 总线、SMBus I/O 扩展器](#) 数据表。

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

Copyright © 2024，德州仪器 (TI) 公司