

第 10 章 I/O 端口

目录

本章包括下列主题：

10.1 简介	10-2
10.2 I/O PORTx 控制寄存器	10-3
10.3 外设复用	10-5
10.4 电平变化通知 (CN) 引脚	10-7
10.5 休眠和空闲模式下的 CN 操作	10-8
10.6 寄存器	10-9
10.7 相关应用笔记	10-11
10.8 版本历史	10-12

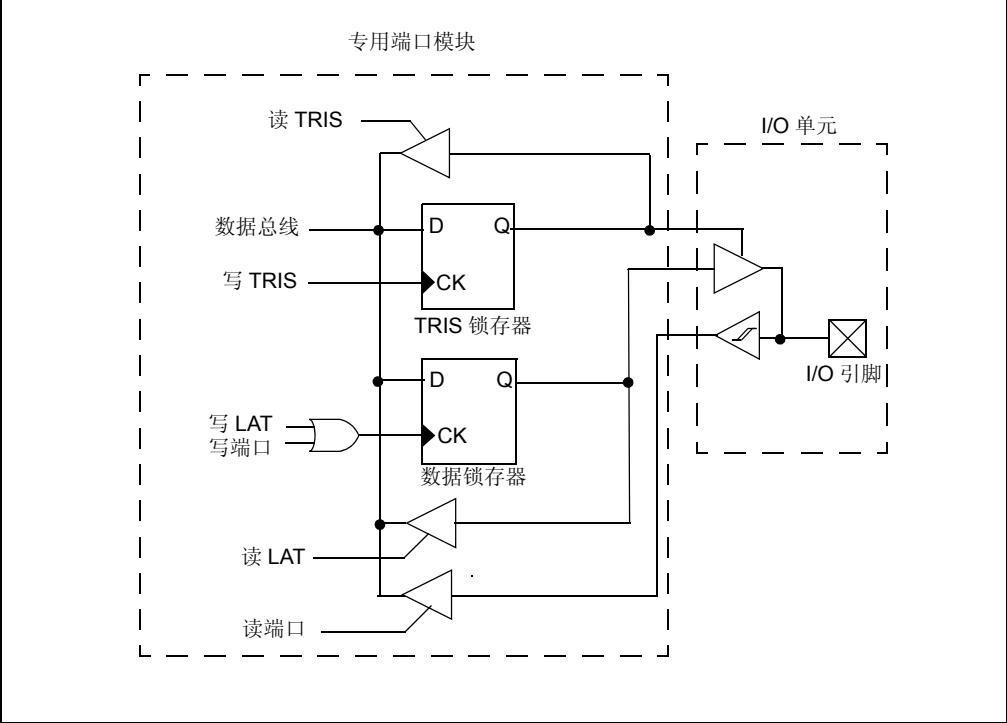
10.1 简介

本章提供关于 dsPIC33F 系列器件的 I/O 端口的信息。所有器件引脚（除 V_{DD}、V_{SS}、 $\overline{\text{MCLR}}$ 和 OSC1/CLKI 外），均为外设与通用 I/O 端口所共用。

通用 I/O 端口可供 dsPIC33F 监视和控制其他器件。大多数 I/O 引脚与备用功能复用。复用将取决于不同器件上的外设功能部件。一般来说，当某个外设正在工作时，其对应的引脚就不能被用作通用 I/O 引脚。

图 10-1 给出了典型 I/O 端口的框图。该框图没有考虑 I/O 引脚上可能复用的外设功能。

图 10-1: 专用端口结构框图



10.2 I/O PORTx 控制寄存器

所有 I/O 端口都有 4 个与该端口操作直接相关的寄存器，其中字母 “x” 表示特定的 I/O 端口：

- TRISx: 数据方向寄存器
- PORTx: I/O 端口寄存器
- LATx: I/O 锁存寄存器
- ODCx: 漏极开路控制寄存器

器件上的每个 I/O 引脚在 TRIS、PORT 和 LAT 寄存器中都分别有一个相关的位。

注： 端口和可用 I/O 引脚的总数将取决于不同的器件。在一个给定的器件中，可能并没有实现端口控制寄存器中的所有位。更多详细信息，请参见具体器件数据手册。

10.2.1 TRIS 寄存器

TRISx 寄存器控制位决定与 I/O 端口相关的各个引脚是输入引脚还是输出引脚。如果某个 I/O 引脚的 TRIS 位为 1，则该引脚是输入引脚。如果某个 I/O 引脚的 TRIS 位为 0，则该引脚被配置为输出引脚。这很好记，因为 1 很像 I（Input，输入），0 很像 O（Output，输出）。复位后，所有端口引脚都被定义为输入。

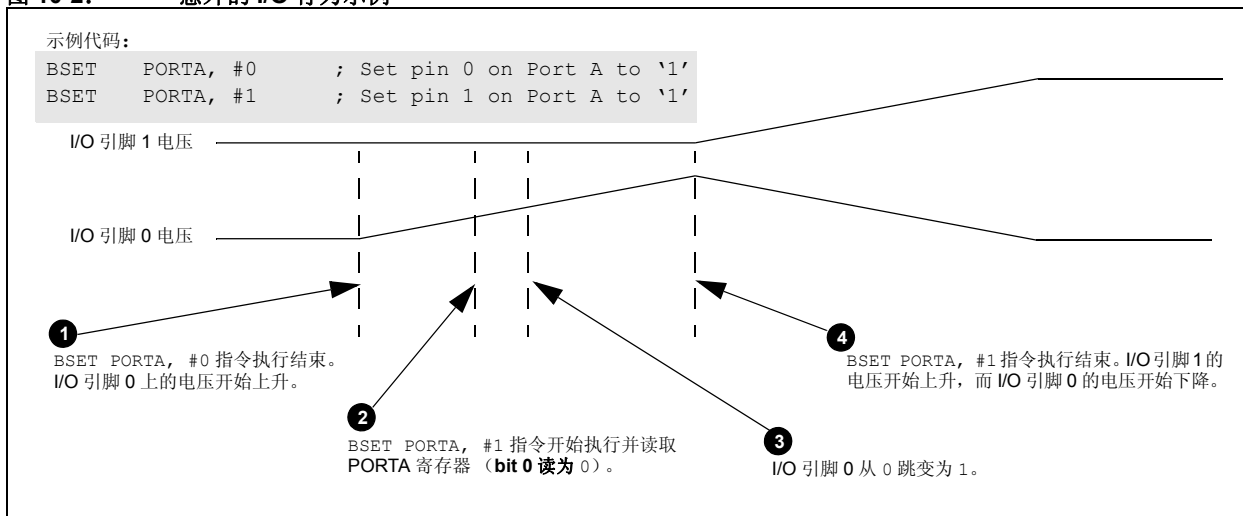
10.2.2 PORT 寄存器

通过 PORTx 寄存器访问 I/O 引脚上的数据。读 PORTx 寄存器是读取 I/O 引脚上的值，而写 PORTx 寄存器是将值写入端口数据锁存器。

很多指令，如 BSET 和 BCLR 指令，都是读 - 修改 - 写操作指令。因此，写一个端口就意味着读该端口引脚的电平，修改读到的值，然后再将改好的值写入端口数据锁存器。当与端口相关的 I/O 引脚被配置为输入，并且在 PORTx 寄存器上使用读 - 修改 - 写命令时，应当特别小心。如果某个配置为输入的 I/O 引脚在过了一段时间后变为输出引脚，则该 I/O 引脚上可能会输出一个意外值。产生这种情况的原因是读 - 修改 - 写指令读取了输入引脚上的瞬时值，并将该值装入了端口数据锁存器。

此外，如果在 I/O 引脚被配置为输出时在 PORTx 寄存器上使用了读 - 修改 - 写指令，根据器件速度和 I/O 容性负载的情况可能出现意外的 I/O 行为。图 10-2 所示为当用户应用程序试图对 PORTA 寄存器使用两条连续的读 - 修改 - 写指令将 PORTA 上 I/O 的 bit 0 和 bit 1 置 1 时，所出现的意外情况。当 CPU 速度很快并且 I/O 引脚上的容性负载很大时，示例代码的意外结果是只有 I/O 的 bit 1 被置 1。

图 10-2: 意外的 I/O 行为示例



执行第一条 BSET 指令时，它会向 PORTA 寄存器中的 bit 0 写入 1，这将导致引脚 0 上的电平开始上升为逻辑电平 1（见图 10-2 中的步骤 1）。但是，如果在引脚 0 上的电平达到逻辑 1 的门限值之前执行第二条 BSET 指令（图 10-2 中的步骤 3），则第二条 BSET（读 - 修改 - 写）指令读取的 bit 0 值为 0，然后它会将该值重新写入 PORTA 寄存器（图 10-2 中的步骤 2）。即，它从 PORTA 寄存器读取的值不是 0x0001，而是 0x0000，之后将它修改为 0x0002（而不是期望值 0x0003），并将该值重新写入 PORTA 寄存器。这将导致引脚 0 上的电压开始下降为逻辑 0，引脚 1 上的电压开始上升为逻辑 1（图 10-2 中的步骤 4）。

10.2.3 LAT 寄存器

与 I/O 引脚相关的 LATx 寄存器消除了可能在执行读 - 修改 - 写指令过程中发生的问题。读 LATx 寄存器将返回保存在端口输出锁存器中的值，而不是 I/O 引脚上的值。对与某个 I/O 端口相关的 LAT 寄存器进行读 - 修改 - 写操作，避免了将输入引脚值写入端口锁存器的可能性。写 LATx 寄存器与写 PORTx 寄存器的效果相同。

下面的示例使用 LATx 寄存器来设置两个 I/O 位。

例 10-1: 使用 LATx 寄存器设置 I/O 引脚

```
BSET    LATA, #0           ;Set pin 0 on Port A to '1'
BSET    LATA, #1           ;Set pin 1 on Port A to '1'
```

PORT 和 LAT 寄存器之间的差异可以归纳如下：

- 写 PORTx 寄存器就是将数据值写入端口锁存器。
- 写 LATx 寄存器就是将数据值写入端口锁存器。
- 读 PORTx 寄存器就是读取 I/O 引脚上的数据值。
- 读 LATx 寄存器就是读取保存在端口锁存器中的数据值。

对于特定器件无效的任何位及其相关的数据和控制寄存器都将被禁止。这意味着对应的 LATx 和 TRISx 寄存器以及端口引脚将读为零。

10.2.4 漏极开路控制寄存器

除 PORT、LAT 和 TRIS 寄存器用于数据控制外，每个端口引脚也可被单独地配置为数字输出或漏极开路输出。这是由与每个端口相关的漏极开路控制寄存器 ODCx 控制的。将其中的任何位置 1 即可将相应的引脚配置为漏极开路输出。

这种漏极开路特性允许通过使用外部上拉电阻，在所需的任意仅用作数字功能的引脚上产生高于 VDD（如 5V）的输出电压。与模拟功能复用的引脚不支持漏极开路 I/O 特性。所允许的最大漏极开路电压与最大 VIH 规范相同。端口引脚和外设配置都支持漏极开路输出特性。

10.3 外设复用

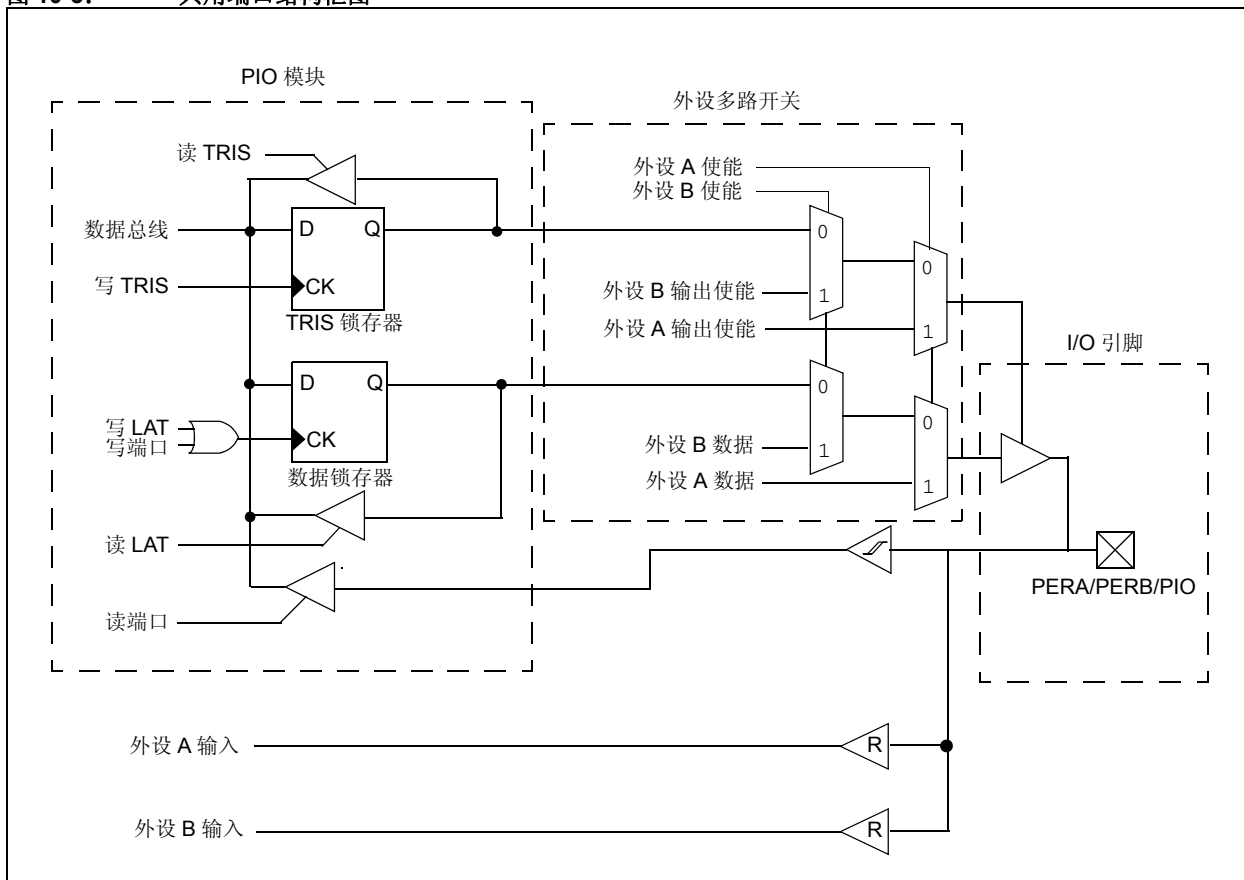
当某个外设使能时，相关引脚的输出驱动器通常由模块控制，而有些可由用户设置。术语“用户可设置”（User Settable）是指相关外设端口引脚的输出驱动器可由用户通过相关 TRISx SFR 配置。必须正确设置 TRISx 寄存器以使外设正常工作。对于用户可设置的外设引脚，实际的端口引脚状态总是可通过 PORTx SFR 读取。

可以通过输入数据路径读该 I/O 引脚，但 I/O 端口位的输出驱动器通常被禁止。

与另一个外设共用一个引脚的 I/O 端口总是服从于该外设。外设的输出缓冲数据和控制信号提供给一对多路开关。这对多路开关用于选择 I/O 引脚的输出数据和控制信号是属于外设还是相应的端口。图 10-3 所示为端口如何与其他外设共用，以及端口所连接的相关 I/O 引脚。

注： 一些端口与 ADC 模块引脚共用。要使用 I/O 端口功能，即使关闭了 ADC 模块，也必须将 AD1PCFG 和 AD2PCFG 寄存器中的相应位（如果有）设为 1。

图 10-3: 共用端口结构框图



10.3.1 I/O 与多个外设复用

对于有些 dsPIC33F 器件，尤其是那些 I/O 引脚数较少的器件，其每个 I/O 引脚可能要复用多个外设功能。图 10-3 所示为两个外设与同一个 I/O 引脚复用的示例。

I/O 引脚的名称定义了与该引脚相关的各个功能的优先级。如图 10-3 所示，概念化的 I/O 引脚与两个外设（外设 A 和外设 B）复用，并命名为 PERA/PERB/PIO。

已为 I/O 引脚选择了适当的名称，以便用户应用程序可以方便地识别分配给该引脚的功能的优先级。对于图 10-3 中的示例，外设 A 对引脚的控制具有最高优先权。如果外设 A 和外设 B 同时使能，外设 A 将控制 I/O 引脚。

10.3.1.1 软件输入引脚控制

分配给某个 I/O 引脚的一些功能可能是那些不控制引脚输出驱动器的输入功能。这类外设的一个示例就是输入捕捉模块。如果使用相应的 TRIS 控制位将与输入捕捉相关的 I/O 引脚配置为输出引脚，则用户可通过其相应的 PORT 寄存器手动影响输入捕捉引脚的状态。这种做法在有些情况下很有用，尤其适用于在没有外部信号连接到输入引脚时进行测试。

请参见图 10-3，外设多路开关的结构将决定外设输入引脚是否可以通过使用 PORT 寄存器用软件控制。当使能外设功能时，图中所示的概念化的外设会断开 I/O 引脚与端口数据的连接。

一般来说，以下外设允许通过 PORT 寄存器手动控制它们的输入引脚：

- 外部中断引脚
- 定时器时钟输入引脚
- 输入捕捉引脚
- PWM 故障引脚

大多数串行通信外设在使用时将完全控制 I/O 引脚，因此不能通过相应的 PORT 寄存器影响与该外设相关的输入引脚。这些外设包括：

- SPI
- I²C™
- UART
- ECAN™

注： 有一些外设可能在部分器件型号上并不提供。更多信息，请参见具体器件数据手册。

10.3.1.2 引脚控制概述

当某个外设使能时，相关引脚的输出驱动器通常由模块控制，而有些可由用户设置。术语“模块控制”（Module Control）是指相关端口引脚的输出驱动器被禁止，并且该引脚只能由外设控制和访问。

输入捕捉外设就是一个很好的用户可设置外设的例子。用户应用程序必须写入相关的 TRIS 寄存器，将输入捕捉引脚配置为输入。当输入捕捉使能时，由于 I/O 引脚电路仍然是激活的，可以使用软件通过以下方法来手动产生捕捉事件：使用相关 TRIS 寄存器将输入捕捉引脚配置为输出。然后，软件即可向相应的 LAT 寄存器中写入值，以对输入捕捉引脚进行内部控制并强制产生捕捉事件。

另一个例子是，可将一个 INTx 引脚配置为输出，然后通过写入相关的 LATx 位即可产生 INTx 中断（如果允许了中断）。

UART 就是一个模块控制外设的例子。当 UART 使能时，PORT 和 TRIS 寄存器不起作用，不能用于写 RX 和 TX 引脚。dsPIC33F 上提供的大多数通信外设都是模块控制外设。

例如，SPI 模块可配置为主模式，而主模式下只需用到 SDO 引脚。在这种情况下，清零（设置为逻辑 0）相关的 TRISx 位即可将 SDI 引脚配置为通用输出引脚。关于如何为模块配置引脚的更多信息，请参见具体模块。

当 CN 中断发生时，用户应用程序必须读与该 CN 引脚相关的 PORT 寄存器。这样做将清除不匹配条件，并设置 CN 逻辑以检测下一次引脚电平变化。可以将当前的端口值与上一次 CN 中断时得到的端口读出值比较，来确定发生了电平变化的引脚。

CN 引脚有最小输入脉冲宽度规范。更多详细信息，请参见器件数据手册的“电气规范”章节。

例 10-2: 配置和使用 CN 中断

```
void configure_CN(void)
{
    CNEN1bits.CN3IE = 1;           // Enable CN3 pin for interrupt detection
    IEC1bits.CNIE = 1;             // Enable CN interrupts
    IFS1bits.CNIF = 0;             // Reset CN interrupt
}

void __attribute__((__interrupt__)) _CNInterrupt(void)
{
    // Insert ISR code here

    IFS1bits.CNIF = 0;             // Clear CN interrupt
}
```

10.5 休眠和空闲模式下的 CN 操作

CN 模块在休眠或空闲模式下继续工作。如果使能的 CN 引脚之一改变了状态，IFSx 寄存器中的 CNIF 状态位将被置 1。如果 IECx 寄存器中的 CNIE 位被置 1，则器件将从休眠或空闲模式唤醒并恢复工作。

如果为 CN 中断分配的优先级等于或小于当前 CPU 的优先级，则器件会从紧随 SLEEP 或 IDLE 指令后的那条指令开始继续执行。

如果为 CN 中断分配的优先级大于当前 CPU 的优先级，则器件将从 CN 中断向量地址继续执行。

10.6 寄存器

10.6.1 电平变化通知寄存器

以下寄存器用于允许和禁止相应的 CN 中断和上拉电阻。

- **CNEN1**: 输入电平变化通知中断允许寄存器 1
- **CNEN2**: 输入电平变化通知中断允许寄存器 2
- **CNPU1**: 输入电平变化通知上拉使能寄存器 1
- **CNPU2**: 输入电平变化通知上拉使能寄存器 2

寄存器 10-1: **CNEN1**: 输入电平变化通知中断允许寄存器 1

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
CN15IE	CN14IE	CN13IE	CN12IE	CN11IE	CN10IE	CN9IE	CN8IE
bit 15							bit 8

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
CN7IE	CN6IE	CN5IE	CN4IE	CN3IE	CN2IE	CN1IE	CN0IE
bit 7							bit 0

图注:

R = 可读位

W = 可写位

U = 未实现位, 读为 0

-n = POR 时的值

1 = 置 1

0 = 清零

x = 未知

bit 15-0

CNxIE: 输入电平变化通知中断允许位

1 = 允许输入电平变化中断

0 = 禁止输入电平变化中断

寄存器 10-2: **CNEN2**: 输入电平变化通知中断允许寄存器 2

U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
—	—	—	—	—	—	—	—
bit 15							bit 8

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
CN23IE	CN22IE	CN21IE	CN20IE	CN19IE	CN18IE	CN17IE	CN16IE
bit 7							bit 0

图注:

R = 可读位

W = 可写位

U = 未实现位, 读为 0

-n = POR 时的值

1 = 置 1

0 = 清零

x = 未知

bit 15-8

未实现: 读为 0

bit 7-0

CNxIE: 输入电平变化通知中断允许位

1 = 允许输入电平变化中断

0 = 禁止输入电平变化中断

dsPIC33F 系列参考手册

寄存器 10-3: CNPU1: 输入电平变化通知上拉使能寄存器 1

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
CN15PUE	CN14PUE	CN13PUE	CN12PUE	CN11PUE	CN10PUE	CN9PUE	CN8PUE
bit 15							bit 8

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
CN7PUE	CN6PUE	CN5PUE	CN4PUE	CN3PUE	CN2PUE	CN1PUE	CN0PUE
bit 7							bit 0

图注:

R = 可读位

W = 可写位

U = 未实现位, 读为 0

-n = POR 时的值

1 = 置 1

0 = 清零

x = 未知

bit 15-0

CNxPUE: 输入电平变化通知上拉使能位

1 = 使能 CNx 引脚输入电平变化上拉

0 = 禁止 CNx 引脚输入电平变化上拉

寄存器 10-4: CNPU2: 输入电平变化通知上拉使能寄存器 2

U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
—	—	—	—	—	—	—	—
bit 15							bit 8

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
CN23PUE	CN22PUE	CN21PUE	CN20PUE	CN19PUE	CN18PUE	CN17PUE	CN16PUE
bit 7							bit 0

图注:

R = 可读位

W = 可写位

U = 未实现位, 读为 0

-n = POR 时的值

1 = 置 1

0 = 清零

x = 未知

bit 15-8

未实现: 读为 0

bit 7-0

CNxPUE: 输入电平变化通知上拉使能位

1 = 使能 CNx 引脚输入电平变化上拉

0 = 禁止 CNx 引脚输入电平变化上拉

10.7 相关应用笔记

本节列出了与手册本章内容相关的应用笔记。这些应用笔记可能并不是专为 dsPIC33F 产品系列而编写的，但其概念是相近的，通过适当修改并受到一定限制即可使用。当前与 I/O 端口模块相关的应用笔记包括：

标题

Implementing Wake-up on Key Stroke

应用笔记编号

AN552

注：如需获取更多 dsPIC33F 系列器件的应用笔记和代码示例，请访问 Microchip 网站（www.microchip.com）。

10.8 版本历史

版本 A（2007 年 2 月）

这是本文档的初始版本。

版本 B（2007 年 2 月）

对整篇文档进行了少量编辑。

版本 C（2008 年 9 月）

该版本包括以下更新：

- 寄存器：
 - 修正了 bit 15-0 的位说明（见寄存器 10-3）。
 - 修正了 bit 7-0 的位说明（见寄存器 10-4）。
- 对整篇文档进行了其他少量修正，如语言和格式的更新。