
第 30 章 带外设引脚选择的 I/O 端口

目录

本章包括下列主题：

30.1 简介	30-2
30.2 I/O 端口控制寄存器	30-3
30.3 外设复用	30-5
30.4 外设引脚选择	30-7
30.5 电平变化通知 (CN) 引脚	30-14
30.6 休眠和空闲模式下的 CN 操作	30-15
30.7 寄存器	30-16
30.8 相关应用笔记	30-34
30.9 版本历史	30-35

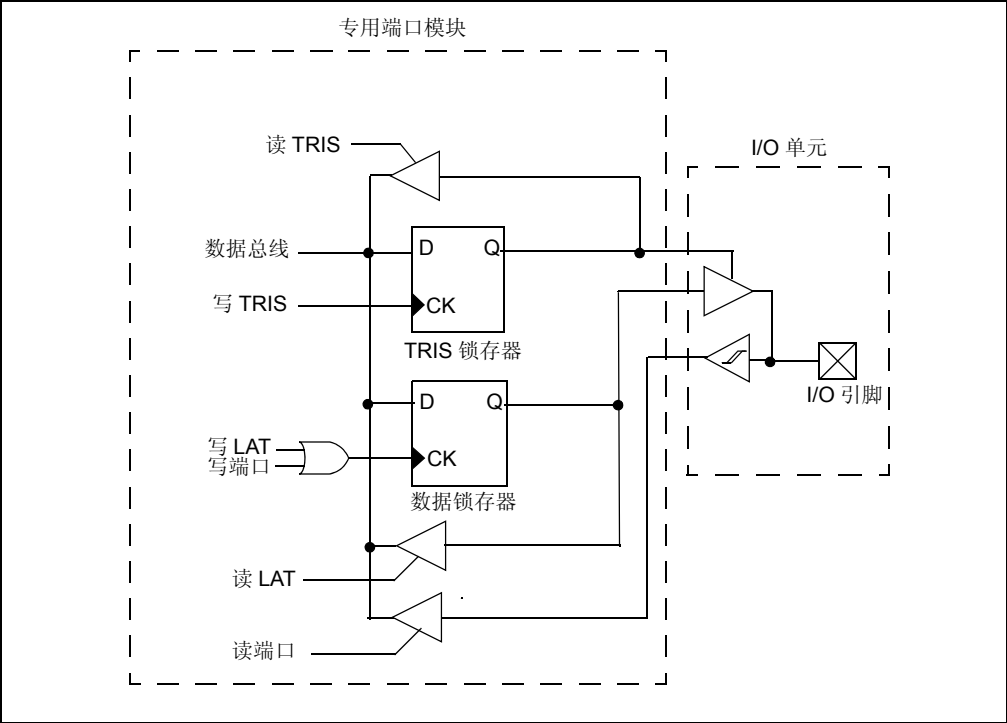
30.1 简介

本章提供关于 dsPIC33F 系列器件的带外设引脚选择的 I/O 端口的信息。所有的器件引脚（除 VDD、VSS、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 30-1 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 30-1: 专用端口结构框图



30.2 I/O 端口控制寄存器

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。

30.2.1 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。

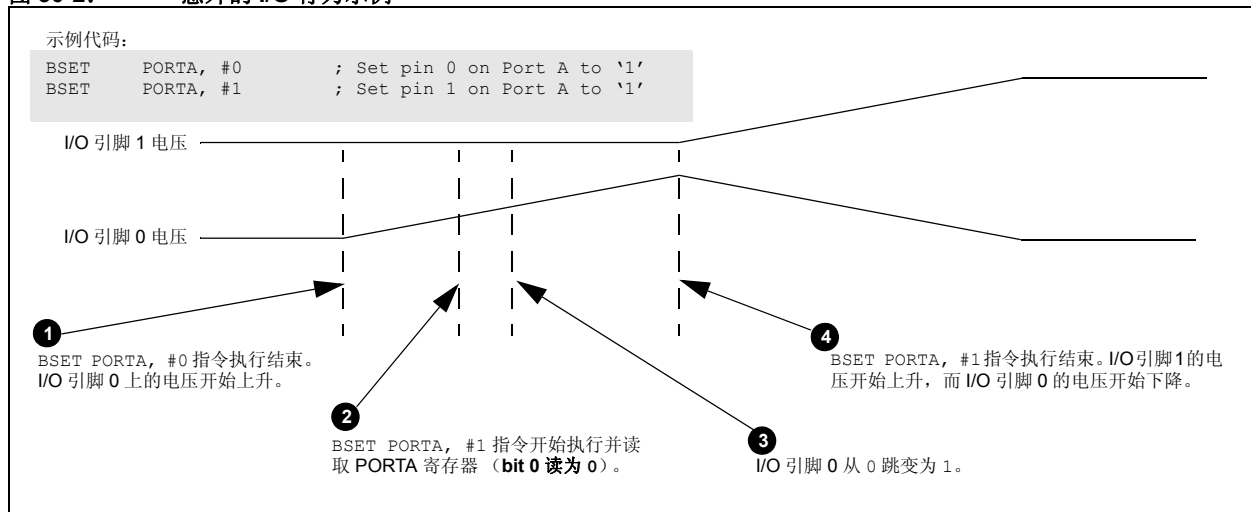
30.2.2 PORT 寄存器

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 30-2 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 30-2: 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 30-2 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 30-2 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 30-2 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电平 0，引脚 1 上的电压开始上升为逻辑电平 1（见图 30-2 中的步骤 4）。

30.2.3 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 30-1: 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'

PORTx 和 LATx 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。

30.2.4 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。）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。

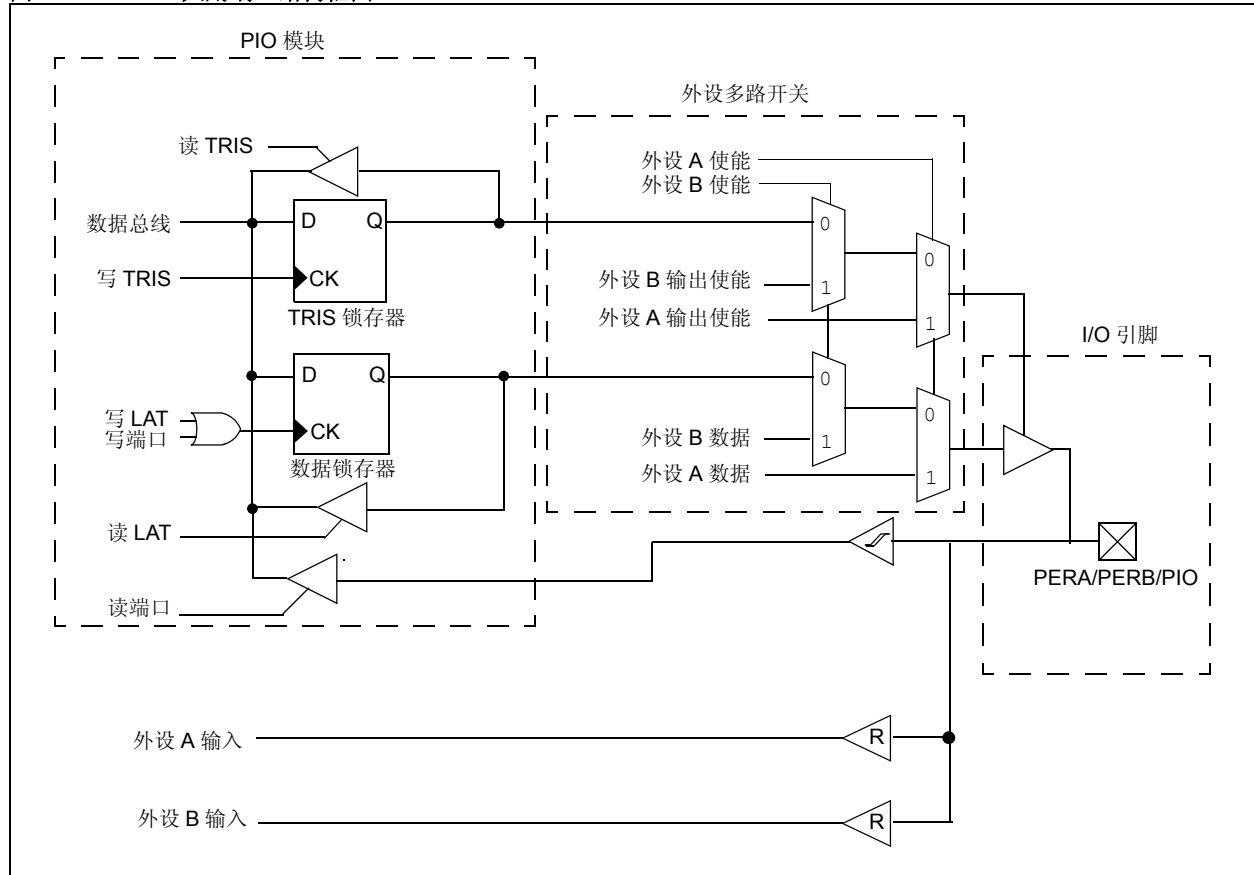
30.3 外设复用

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。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 30-3 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 30-3: 共用端口结构框图



30.3.1 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 30-3 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 30-3 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，以使用户应用程序可以方便地确定分配给该引脚的功能的优先级。对于图 30-3 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。

30.3.1.1 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 30-3，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™
- DCI
- UART
- ECAN™
- QEI

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。

30.3.1.2 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。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 特殊功能寄存器（Special Function Register, SFR）在软件中配置。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。

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，不能用于读或写 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，请参见具体模块章节。

30.4 外设引脚选择

通用器件的主要挑战是提供尽可能多的外设功能部件，同时将其与 I/O 引脚功能的冲突降到最小。在低引脚数器件上，这一挑战更为严峻。在需要多个外设复用一個引脚的应用中，要在应用程序代码中进行变通比较困难，换句话说彻底重新设计可能是唯一的选择。

外设引脚选择配置提供了这些选择的替代方法，使得用户可以**在较宽的 I/O 引脚范围内选择和配置外设功能部件**。通过增加特定器件上可用的引脚排列选项，用户可以让器件更好地适合他们的整个应用，而不是通过修改应用来适应器件。

外设引脚选择配置功能对固定的一部分数字 I/O 引脚进行操作。**用户可以将大多数数字外设的输入和 / 或输出独立地映射到这些 I/O 引脚中的任何一个**。外设引脚选择通过软件来执行，通常不需要对器件进行再编程。一旦建立外设引脚选择，就同时包含了硬件保护，以防止对外设映射的意外或错误更改。

30.4.1 可用的引脚

外设引脚选择功能可在最多 16 个引脚的范围内使用。可用引脚的数目取决于特定器件及其引脚数。**支持外设引脚选择功能的引脚在它们的引脚全称中包含名称“RPn”，其中“RP”表示可重映射的外设，“n”是可重映射的引脚编号**。

30.4.2 可用的外设

外设引脚选择管理的外设都是**仅数字功能的外设**。这些外设包括一般串行通信（UART 和 SPI）、通用定时器时钟输入、与定时器相关的外设（输入捕捉和输出比较）以及电平变化中断输入。

相比较而言，一些仅数字功能的外设模块不能使用外设引脚选择功能。这是因为外设功能需要特定端口上的特殊 I/O 电路，且不能很容易地连接到多个引脚。这些模块包括 I²C。类似的要求排除了所有带模拟输入的模块，例如 A/D 转换器。

注： 关于外设所支持的外设引脚选择的具体列表，请参见器件数据手册。

可重映射和不可重映射外设之间的主要差异在于可重映射外设与默认的 I/O 引脚无关。必须始终在使用外设前将其分配给特定的 I/O 引脚。相反，不可重映射外设始终在默认引脚上可用，假设该外设有效且与其他外设没有冲突。

当给定 I/O 引脚上的可重映射外设有效时，它的优先级高于所有其他数字 I/O 和与该引脚相关的数字通信外设。优先级与被映射外设的类型无关。可重映射外设的优先级永远不会高于与该引脚相关的任何模拟功能。

30.4.3 控制外设引脚选择

外设引脚选择功能由两组 SFR 控制：一组映射外设输入，另一组映射外设输出。因为它们分别是分别控制的，所以可以不受限制地将特定外设的输入和输出（如果外设同时具有输入和输出）配置在任何可选择的功能引脚上。

外设与外设可选择引脚之间的关系用两种不同的方式进行处理，取决于被映射的是输入还是输出。

30.4.3.1 输入映射

外设引脚选择选项的输入在外设基础上进行映射。即，与外设相关的控制寄存器指示要被映射的引脚。RPINRx 寄存器用于配置外设输入映射（见寄存器 30-5 至寄存器 30-17）。每个寄存器包含 5 位位域组，每组都与可重映射外设之一相关。用适当的 5 位值编程给定外设的位域，会将具有对应值的 RPn 引脚映射到该外设。对于任何给定的器件，任何位域的值的有效范围与器件所支持的外设引脚选择的数目相对应。

例如，图 30-4 给出了 U1RX 输入的可重映射引脚选择的图示。

图 30-4: U1RX 的可重映射输入

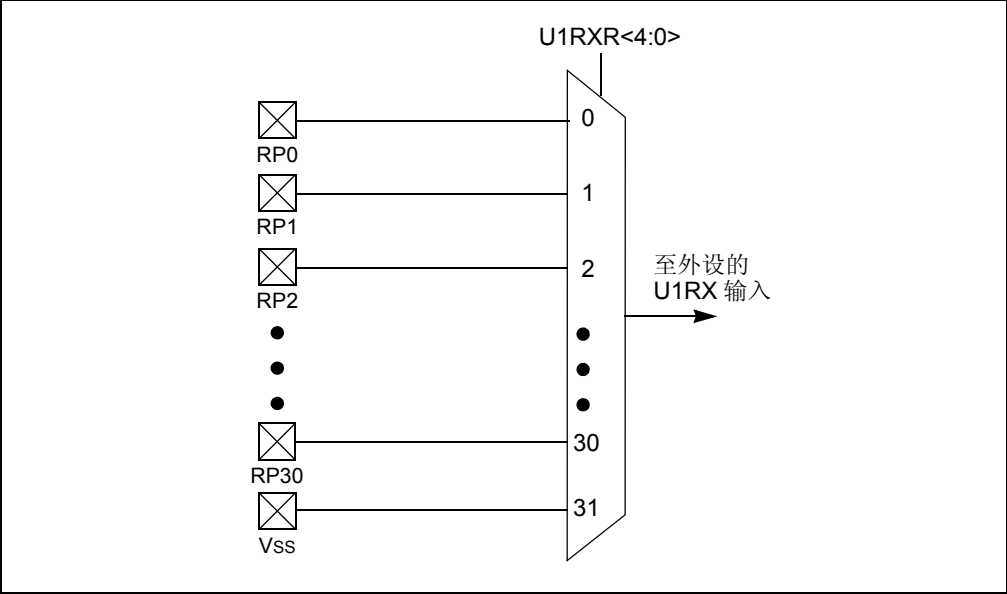


表 30-1: 可选择的输入源（将输入映射到功能）

输入名称 ⁽¹⁾	功能名称	寄存器	配置位
外部中断 1	INT1	RPINR0	INT1R<4:0>
外部中断 2	INT2	RPINR1	INT2R<4:0>
Timer2 外部时钟	T2CK	RPINR3	T2CKR<4:0>
Timer3 外部时钟	T3CK	RPINR3	T3CKR<4:0>
输入捕捉 1	IC1	RPINR7	IC1R<4:0>
输入捕捉 2	IC2	RPINR7	IC2R<4:0>
输入捕捉 7	IC7	RPINR10	IC7R<4:0>
输入捕捉 8	IC8	RPINR10	IC8R<4:0>
输出比较故障 A	OCFA	RPINR11	OCFAR<4:0>
PWM1 故障	FLTA1	RPINR12	FLTA1R<4:0>
PWM1 故障	FLTA2	RPINR13	FLTA2R<4:0>
QEI A 相	QEA	RPINR14	QEAR<4:0>
QEI B 相	QEB	RPINR14	QEBR<4:0>
QEI 索引	INDX	RPINR15	INDXR<4:0>
UART1 接收	U1RX	RPINR18	U1RXR<4:0>
UART1 允许发送	U1CTS	RPINR18	U1CTSR<4:0>
SPI1 数据输入	SDI1	RPINR20	SDI1R<4:0>
SPI1 时钟输入	SCK1	RPINR20	SCK1R<4:0>
SPI1 从选择输入	SS1	RPINR21	SS1R<4:0>

注 1: 除非另外声明，否则所有输入都使用施密特输入缓冲器。

注: 表 30-1 和图 30-4 提供了通用器件的可选择输入源的示例。更多信息，请参见具体器件数据手册。

30.4.3.2 输出映射

与输入相比，外设引脚选择选项的输出在引脚基础上进行映射。在这种情况下，与特定引脚相关的控制寄存器指示要被映射的外设输出。RPORx 寄存器用于控制输出映射。像 RPINRx 寄存器一样，每个寄存器包含 5 位位域组，每组都与一个 RPn 引脚相关（见寄存器 30-18 至寄存器 30-25）。位域的值与外设之一相对应，并且该外设的输出被映射到引脚（见表 30-2 和图 30-5）。空输出与输出寄存器的复位值 0 相关。这样做可确保在默认情况下，可重映射输出保持与所有输出引脚之间的断开状态。

图 30-5: RPn 的可重映射输出的复用

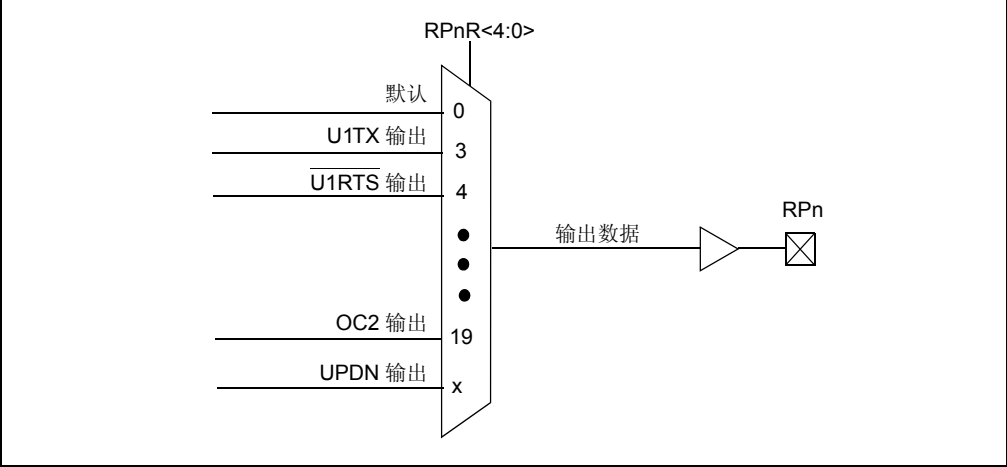


表 30-2: 可重映射引脚 (RPn) 的输出选择

功能	RPnR<4:0>	输出名称
NULL	00000	RPn 连接到默认端口引脚
U1TX	00011	RPn 连接到 UART1 发送
U1RTS	00100	RPn 连接到 UART1 请求发送
SDO1	00111	RPn 连接到 SPI1 数据输出
SCK1OUT	01000	RPn 连接到 SPI1 时钟输出
SS1OUT	01001	RPn 连接到 SPI1 从选择输出
OC1	10010	RPn 连接到输出比较 1
OC2	10011	RPn 连接到输出比较 2
UPDN	11010	RPn 连接到 QE1 方向 (UPDN) 状态

注: 图 30-5 和表 30-2 提供了通用器件的示例。更多信息, 请参见具体器件数据手册。

30.4.3.3 映射限制

外设选择引脚的控制机制不局限于固定外设配置的小范围内。在任何外设映射 SFR 之间没有互锁或硬件强制的锁定。也就是说, 任何或所有 RPn 引脚上的外设映射的任何组合都是可能的。这包括外设输入和输出到引脚的多对一或一对多映射。从配置观点来看, 这种映射在技术上是可能的, 但从电气观点来看可能不受支持。

30.4.4 控制配置更改

由于可以在运行时更改外设的映射, 因此必须对外设重映射设置一些限制条件以防止意外更改配置。所有 dsPIC33F 器件都具有 3 个功能以防止对外设映射的更改:

- 控制寄存器锁定序列
- 连续状态监视
- 配置位重映射锁定

30.4.4.1 控制寄存器锁定

在正常工作时，不允许写入 RPINRx 和 RPORx 寄存器。尝试写入操作看似正常执行，但实际上寄存器的内容保持不变。要更改这些寄存器，必须用硬件进行解锁。寄存器锁定由 IOLOCK 位 (OSCCON<6>) 控制。将 IOLOCK 置 1 可防止对控制寄存器的写操作；将 IOLOCK 清零则允许写操作。

要置 1 或清零 IOLOCK，必须执行特定的命令序列：

1. 将 0X46 写入 OSCCON<7:0>
2. 将 0X57 写入 OSCCON<7:0>
3. 执行对 IOLOCK 清零（或置 1）的单个操作

IOLOCK 会保持一种状态直到被更改。这允许对所有的外设引脚选择这样进行配置：在对所有控制寄存器的更新后紧跟一个解锁序列，然后用第二个锁定序列将 IOLOCK 置 1。

注： MPLAB® C30 提供了用于解锁 OSCCON 寄存器的内建 C 语言函数：

```
__builtin_write_OSCCONL(value)
__builtin_write_OSCCONH(value)
```

更多信息，请参见 MPLAB 帮助文件。

30.4.4.2 连续状态监视

除了防止直接写操作，RPINRx 和 RPORx 寄存器的内容一直由影子寄存器通过硬件进行监视。如果任何寄存器发生了意外更改（例如 ESD 或其他外部事件引起的干扰），将会触发配置不匹配复位。

30.4.4.3 配置位引脚选择锁定

为了进一步确保安全，可以将器件配置为防止对 RPINRx 和 RPORx 寄存器进行多于一次写会话。IOL1WAY (FOSC<5>) 配置位会阻止 IOLOCK 位在置 1 后被清零。

在默认（未编程）状态下，IOL1WAY 被置 1，将用户限制为只能进行一次写会话。对 IOL1WAY 编程可允许用户（通过对解锁序列的正确使用）对外设引脚选择寄存器不受限制的访问。

30.4.5 外设引脚选择的注意事项

在应用设计中使用控制外设引脚选择功能有一些可能被大多数用户忽略的注意事项。对于几个只能作为可重映射外设的常见外设尤其如此。

主要的注意事项是在器件的默认（复位）状态下，外设引脚选择在默认引脚上不可用。特别是，由于所有 RPINRx 寄存器复位为全 1，所有 RPORx 寄存器复位为全 0，这意味着所有外设引脚选择输入连接到 Vss，而所有外设引脚选择输出处于断开状态。这种情况要求用户在执行任何其他应用程序代码前，必须用适当的外设配置初始化器件。由于 IOLOCK 位在解锁状态下复位，因此在器件复位结束后不必执行解锁序列。然而，基于应用安全考虑，在写入控制寄存器后最好将 IOLOCK 置 1 并锁定配置。

由于解锁序列对时序的要求很严格，它必须作为汇编语言程序以与更改振荡器配置相同的方式执行。如果应用程序是用 C 语言或其他高级语言编写的，则解锁序列应通过写行内汇编代码来执行。

选择配置需要查看所有外设引脚选择及其引脚分配，尤其是那些不会在应用中使用的外设。在所有情况下，必须完全禁止未用的引脚可选择外设。未用的外设应将它们的输入分配给未用的 RPN 引脚功能。带有未用 RPN 功能的 I/O 引脚应被配置为空外设输出。

外设到特定引脚的分配不会自动执行引脚的 I/O 电路的任何其他配置。理论上，这意味着将引脚可选择输出加到引脚，当驱动输出时，引脚可能会意外驱动现有的外设输入。用户必须熟悉共用同一个可重映射引脚的其他固定外设的行为，了解何时使能或禁止它们。为安全起见，共用同一个引脚的固定数字外设在不使用时应被禁止。

根据这些概念，配置特定外设的可重映射引脚不会自动开启该外设功能。必须将外设特别配置为工作并使能，好像是连接到固定引脚一样。这部分在应用程序代码中的位置（紧跟器件复位和外设配置，或在主应用程序内）取决于外设及其在应用中的使用。

最后的注意事项是，外设引脚选择功能既不会改写模拟输入，也不会将带模拟功能的引脚重新配置为数字 I/O。如果器件复位时引脚被配置为模拟输入，则使用外设引脚选择时必须明确将其重新配置为数字 I/O。

例 30-2 给出了使用 UART1 进行带流控制的双向通信的配置。使用了以下输入和输出功能：

- 输入功能：U1RX 和 $\overline{\text{U1CTS}}$
- 输出功能：U1TX 和 $\overline{\text{U1RTS}}$

例 30-2: 配置 UART1 输入和输出功能

```

//*****
// Unlock Registers
//*****
__builtin_write_OSCCONL(OSCCON & ~(1<<6));

//*****
// Configure Input Functions
// (See Table 30-1)
//*****
// Assign U1Rx To Pin RP0
//*****
RPINR18bits.U1RXR = 0;

//*****
// Assign U1CTS To Pin RP1
//*****
RPINR18bits.U1CTSR = 1;

//*****
// Configure Output Functions
// (See Table 30-2)
//*****
// Assign U1Tx To Pin RP2
//*****
RPOR1bits.RP2R = 3;

//*****
// Assign U1RTS To Pin RP3
//*****
RPOR1bits.RP3R = 4;

//*****
// Lock Registers
//*****
__builtin_write_OSCCONL(OSCCON | (1<<6));

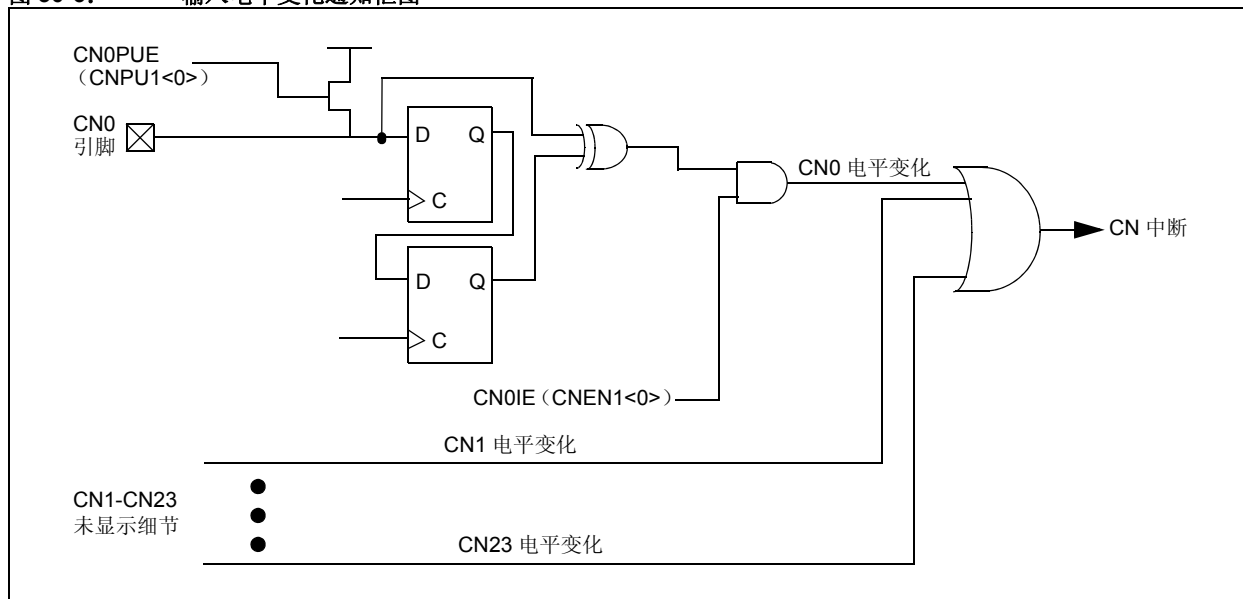
```

30.5 电平变化通知 (CN) 引脚

电平变化通知 (Change Notification, CN) 引脚使 dsPIC33F 器件能够向处理器发出中断请求, 以响应所选择的输入引脚上的状态变化。可以选择 (使能) 最多 24 个输入引脚产生 CN 中断。可用的 CN 输入引脚总数取决于所选的 dsPIC33F 器件。更多详细信息, 请参见器件数据手册。

图 30-6 给出了 CN 硬件的基本功能。

图 30-6: 输入电平变化通知框图



30.5.1 CN 控制寄存器

有 4 个与 CN 模块相关的控制寄存器: CNEN1、CNEN2、CNPU1 和 CNPU2。

CNEN1 和 CNEN2 寄存器包含 CNxIE 控制位, 其中 “x” 表示 CN 输入引脚的编号。要让某个 CN 输入引脚中断 CPU, 必须将其 CNxIE 位置 1。

CNPU1 和 CNPU2 寄存器包含 CNxPUE 控制位。每个 CN 引脚连接一个弱上拉器件, 该器件可通过 CNxPUE 控制位使能或禁止。弱上拉器件充当连接到该引脚的电流源, 并且当连接了按钮或键盘设备时, 有了弱上拉器件即可不需要外部电阻。关于 CN 上拉器件的电流规范, 请参见器件数据手册的“电气规范”章节。

30.5.2 CN 配置和操作

CN 引脚配置如下:

1. 通过将 TRISx 寄存器中的相关位置 1, 确保 CN 引脚配置为数字输入引脚。
2. 通过将 CNEN1 和 CNEN2 寄存器中的相应位置 1, 允许所选 CN 引脚的中断。
3. 通过将 CNPU1 和 CNPU2 寄存器中的相应位置 1, 开启所选的 CN 引脚的弱上拉器件 (如需要)。
4. 清零 IFSx 寄存器中的 CNIF 中断标志。
5. 使用 IPCx 寄存器中的 CNIP<2:0> 控制位为 CN 中断选择所需的中断优先级。
6. 使用 IECx 寄存器中的 CNIE 控制位来允许 CN 中断。

当 CN 中断发生时，用户应用程序应该读与该 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 30-3: 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
}
```

30.6 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。

30.7 寄存器

30.7.1 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

30.7.2 外设引脚选择寄存器

以下寄存器用于配置 dsPIC33F 器件引脚的输入和输出功能：

- **RPINR0**：外设引脚选择输入寄存器 0
- **RPINR1**：外设引脚选择输入寄存器 1
- **RPINR3**：外设引脚选择输入寄存器 3
- **RPINR7**：外设引脚选择输入寄存器 7
- **RPINR10**：外设引脚选择输入寄存器 10
- **RPINR11**：外设引脚选择输入寄存器 11
- **RPINR12**：外设引脚选择输入寄存器 12
- **RPINR13**：外设引脚选择输入寄存器 13
- **RPINR14**：外设引脚选择输入寄存器 14
- **RPINR15**：外设引脚选择输入寄存器 15
- **RPINR18**：外设引脚选择输入寄存器 18
- **RPINR20**：外设引脚选择输入寄存器 20
- **RPINR21**：外设引脚选择输入寄存器 21
- **RPOR0**：外设引脚选择输出寄存器 0
- **RPOR1**：外设引脚选择输出寄存器 1
- **RPOR2**：外设引脚选择输出寄存器 2
- **RPOR3**：外设引脚选择输出寄存器 3
- **RPOR4**：外设引脚选择输出寄存器 4
- **RPOR5**：外设引脚选择输出寄存器 5
- **RPOR6**：外设引脚选择输出寄存器 6

注： 一些 dsPIC33F 器件可能具有额外的 RPORx（外设引脚选择输出）寄存器。更多信息，请参见具体器件数据手册。
--

寄存器 30-1: **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 30-2: **CNEN2**: 输入电平变化通知中断允许寄存器 2

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-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 30-3: 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 = 使能输入电平变化上拉
0 = 禁止输入电平变化上拉

寄存器 30-4: CNPU2: 输入电平变化通知上拉使能寄存器 2

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-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 = 使能输入电平变化上拉
0 = 禁止输入电平变化上拉

寄存器 30-5: RPINR0: 外设引脚选择输入寄存器 0

U-0	U-0	U-0	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
—	—	—	INT1R<4:0>				
bit 15							bit 8

U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
—	—	—	—	—	—	—	—
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 15-13 未实现: 读为 0

bit 12-8 **INT1R<4:0>**: 将外部中断 1 (INTR1) 分配给对应 RPn 引脚的位

11111 = 输入连接到 Vss

11110 = 输入连接到 RP30

.

.

.

00001 = 输入连接到 RP1

00000 = 输入连接到 RP0

bit 7-0 未实现: 读为 0

寄存器 30-6: RPINR1: 外设引脚选择输入寄存器 1

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

U-0	U-0	U-0	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
—	—	—	INT2R<4:0>				
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 15-5 未实现: 读为 0

bit 4-0 **INT2R<4:0>**: 将外部中断 2 (INTR2) 分配给对应 RPn 引脚的位

11111 = 输入连接到 Vss

11110 = 输入连接到 RP30

.

.

.

00001 = 输入连接到 RP1

00000 = 输入连接到 RP0

dsPIC33F 系列参考手册

寄存器 30-7: **RPINR3: 外设引脚选择输入寄存器 3**

U-0	U-0	U-0	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
—	—	—	T3CKR<4:0>				
bit 15							bit 8
U-0	U-0	U-0	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
—	—	—	T2CKR<4:0>				
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 15-13

未实现: 读为 0
- bit 12-8

T3CKR<4:0>: 将 Timer3 外部时钟 (T3CK) 分配给对应 RPn 引脚的位

11111 = 输入连接到 Vss

11110 = 输入连接到 RP30

.

.

.

00001 = 输入连接到 RP1

00000 = 输入连接到 RP0
- bit 7-5

未实现: 读为 0
- bit 4-0

T2CKR<4:0>: 将 Timer2 外部时钟 (T2CK) 分配给对应 RPn 引脚的位

11111 = 输入连接到 Vss

11110 = 输入连接到 RP30

.

.

.

00001 = 输入连接到 RP1

00000 = 输入连接到 RP0

寄存器 30-8: **RPINR7: 外设引脚选择输入寄存器 7**

U-0	U-0	U-0	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
—	—	—	IC2R<4:0>				
bit 15							bit 8
U-0	U-0	U-0	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
—	—	—	IC1R<4:0>				
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 15-13

未实现: 读为 0
- bit 12-8

IC2R<4:0>: 将输入捕捉 2 (IC2) 分配给对应 RPn 引脚的位

11111 = 输入连接到 Vss

11110 = 输入连接到 RP30

.

.

.

00001 = 输入连接到 RP1

00000 = 输入连接到 RP0
- bit 7-5

未实现: 读为 0
- bit 4-0

IC1R<4:0>: 将输入捕捉 1 (IC1) 分配给对应 RPn 引脚的位

11111 = 输入连接到 Vss

11110 = 输入连接到 RP30

.

.

.

00001 = 输入连接到 RP1

00000 = 输入连接到 RP0

寄存器 30-9: **RPINR10: 外设引脚选择输入寄存器 10**

U-0	U-0	U-0	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
—	—	—	IC8R<4:0>				
bit 15							bit 8

U-0	U-0	U-0	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
—	—	—	IC7R<4:0>				
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 15-13

未实现: 读为 0
- bit 12-8

IC8R<4:0>: 将输入捕捉 8 (IC8) 分配给对应 RPn 引脚的位

11111 = 输入连接到 Vss

11110 = 输入连接到 RP30

.

.

.

00001 = 输入连接到 RP1

00000 = 输入连接到 RP0
- bit 7-5

未实现: 读为 0
- bit 4-0

IC7R<4:0>: 将输入捕捉 7 (IC7) 分配给对应 RPn 引脚的位

11111 = 输入连接到 Vss

11110 = 输入连接到 RP30

.

.

.

00001 = 输入连接到 RP1

00000 = 输入连接到 RP0

寄存器 30-10: RPINR11: 外设引脚选择输入寄存器 11

U-0	U-0	U-0	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
—	—	—	—	—	—	—	—
bit 15						bit 8	
U-0	U-0	U-0	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
—	—	—	OCFAR<4:0>				
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 15-5 未实现: 读为 0

bit 4-0 OCFAR<4:0>: 将输出比较 A (OCFA) 分配给对应 RPn 引脚的位

11111 = 输入连接到 Vss

11110 = 输入连接到 RP30

.

.

.

00001 = 输入连接到 RP1

00000 = 输入连接到 RP0

寄存器 30-11: RPINR12: 外设引脚选择输入寄存器 12

U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
—	—	—	—	—	—	—	—
bit 15						bit 8	
U-0	U-0	U-0	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
—	—	—	FLTA1R<4:0>				
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 15-5 未实现: 读为 0

bit 4-0 FLTA1R<4:0>: 将 PWM1 故障 (FLTA1) 分配给对应 RPn 引脚的位

11111 = 输入连接到 Vss

11110 = 输入连接到 RP30

.

.

.

00001 = 输入连接到 RP1

00000 = 输入连接到 RP0

寄存器 30-12: **RPINR13: 外设引脚选择输入寄存器 13**

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

U-0	U-0	U-0	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
—	—	—	FLTA2R<4:0>				
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 15-5 **未实现:** 读为 0

bit 4-0 **FLTA2R<4:0>:** 将 PWM2 故障 ($\overline{\text{FLTA2}}$) 分配给对应 RPn 引脚的位

11111 = 输入连接到 Vss

11110 = 输入连接到 RP30

.

.

.

00001 = 输入连接到 RP1

00000 = 输入连接到 RP0

寄存器 30-13: **RPINR14: 外设引脚选择输入寄存器 14**

U-0	U-0	U-0	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
—	—	—	QEBR<4:0>				
bit 15							bit 8
U-0	U-0	U-0	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
—	—	—	QEAR<4:0>				
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 15-13

未实现: 读为 0
- bit 12-8

QEBR<4:0>: 将 B (QEB) 分配给对应 RPn 引脚的位

11111 = 输入连接到 Vss

11110 = 输入连接到 RP30

.

.

.

00001 = 输入连接到 RP1

00000 = 输入连接到 RP0
- bit 7-5

未实现: 读为 0
- bit 4-0

QEAR<4:0>: 将 A (QEA) 分配给对应 RPn 引脚的位

11111 = 输入连接到 Vss

11110 = 输入连接到 RP30

.

.

.

00001 = 输入连接到 RP1

00000 = 输入连接到 RP0

dsPIC33F 系列参考手册

寄存器 30-14: **RPINR15: 外设引脚选择输入寄存器 15**

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

U-0	U-0	U-0	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
—	—	—	INDXR<4:0>				
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 15-5 **未实现:** 读为 0
- bit 4-0 **INDXR<4:0>:** 将 QEI 索引（INDX）分配给对应 RPn 引脚的位
 - 11111 = 输入连接到 Vss
 - 11110 = 输入连接到 RP30
 - .
 - .
 - .
 - 00001 = 输入连接到 RP1
 - 00000 = 输入连接到 RP0

寄存器 30-15: **RPINR18: 外设引脚选择输入寄存器 18**

U-0	U-0	U-0	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
—	—	—	U1CTSR<4:0>				
bit 15							bit 8
U-0	U-0	U-0	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
—	—	—	U1RXR<4:0>				
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 15-13

未实现: 读为 0
- bit 12-8

U1CTSR<4:0>: 将 UART1 允许发送 (**U1CTS**) 分配给对应 RPn 引脚的位

11111 = 输入连接到 Vss

11110 = 输入连接到 RP30

.

.

.

00001 = 输入连接到 RP1

00000 = 输入连接到 RP0
- bit 7-5

未实现: 读为 0
- bit 4-0

U1RXR<4:0>: 将 UART1 接收 (**U1RX**) 分配给对应 RPn 引脚的位

11111 = 输入连接到 Vss

11110 = 输入连接到 RP30

.

.

.

00001 = 输入连接到 RP1

00000 = 输入连接到 RP0

寄存器 30-16: **RPINR20: 外设引脚选择输入寄存器 20**

U-0	U-0	U-0	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
—	—	—	SCK1R<4:0>				
bit 15							bit 8

U-0	U-0	U-0	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
—	—	—	SDI1R<4:0>				
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 15-13

未实现：读为 0
- bit 12-8

SCK1R<4:0>：将 SPI1 时钟输入（SCK1IN）分配给对应 RPn 引脚的位
11111 = 输入连接到 Vss
11110 = 输入连接到 RP30
.
.
.
00001 = 输入连接到 RP1
00000 = 输入连接到 RP0
- bit 7-5

未实现：读为 0
- bit 4-0

SDI1R<4:0>：将 SPI1 数据输入（SDI1）分配给对应 RPn 引脚的位
11111 = 输入连接到 Vss
11110 = 输入连接到 RP30
.
.
.
00001 = 输入连接到 RP1
00000 = 输入连接到 RP0

寄存器 30-17: **RPINR21: 外设引脚选择输入寄存器 21**

U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
—	—	—	—	—	—	—	—
bit 15				bit 8			
U-0	U-0	U-0	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
—	—	—	SS1R<4:0>				
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 15-15 未实现: 读为 0

bit 4-0 **SS1R<4:0>**: 将 SPI1 从选择输入 (SS1IN) 分配给对应 RPn 引脚的位

11111 = 输入连接到 Vss

11110 = 输入连接到 RP30

.

.

.

00001 = 输入连接到 RP1

00000 = 输入连接到 RP0

dsPIC33F 系列参考手册

寄存器 30-18: **RPOR0: 外设引脚选择输出寄存器 0**

U-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	—	RP1R<4:0>				
bit 15							bit 8
U-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	—	RP0R<4:0>				
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 15-13 未实现: 读为 0
- bit 12-8 **RP1R<4:0>**: 将外设输出功能分配给 RP1 输出引脚的位 (请参见表 30-2 了解外设功能编号)
- bit 7-5 未实现: 读为 0
- bit 4-0 **RP0R<4:0>**: 将外设输出功能分配给 RP0 输出引脚的位 (请参见表 30-2 了解外设功能编号)

寄存器 30-19: **RPOR1: 外设引脚选择输出寄存器 1**

U-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	—	RP3R<4:0>				
bit 15							bit 8
U-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	—	RP2R<4:0>				
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 15-13 未实现: 读为 0
- bit 12-8 **RP3R<4:0>**: 将外设输出功能分配给 RP3 输出引脚的位 (请参见表 30-2 了解外设功能编号)
- bit 7-5 未实现: 读为 0
- bit 4-0 **RP2R<4:0>**: 将外设输出功能分配给 RP2 输出引脚的位 (请参见表 30-2 了解外设功能编号)

寄存器 30-20: RPOR2: 外设引脚选择输出寄存器 2

U-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	—	RP5R<4:0>				
bit 15							bit 8

U-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	—	RP4R<4:0>				
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 15-13 未实现: 读为 0

bit 12-8 **RP5R<4:0>**: 将外设输出功能分配给 RP5 输出引脚的位 (请参见表 30-2 了解外设功能编号)

bit 7-5 未实现: 读为 0

bit 4-0 **RP4R<4:0>**: 将外设输出功能分配给 RP4 输出引脚的位 (请参见表 30-2 了解外设功能编号)

寄存器 30-21: RPOR3: 外设引脚选择输出寄存器 3

U-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	—	RP7R<4:0>				
bit 15							bit 8

U-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	—	RP6R<4:0>				
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 15-13 未实现: 读为 0

bit 12-8 **RP7R<4:0>**: 将外设输出功能分配给 RP7 输出引脚的位 (请参见表 30-2 了解外设功能编号)

bit 7-5 未实现: 读为 0

bit 4-0 **RP6R<4:0>**: 将外设输出功能分配给 RP6 输出引脚的位 (请参见表 30-2 了解外设功能编号)

dsPIC33F 系列参考手册

寄存器 30-22: **RPOR4: 外设引脚选择输出寄存器 4**

U-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	—	RP9R<4:0>				
bit 15							bit 8

U-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	—	RP8R<4:0>				
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 15-13 **未实现:** 读为 0
- bit 12-8 **RP9R<4:0>:** 将外设输出功能分配给 RP9 输出引脚的位 (请参见表 30-2 了解外设功能编号)
- bit 7-5 **未实现:** 读为 0
- bit 4-0 **RP8R<4:0>:** 将外设输出功能分配给 RP8 输出引脚的位 (请参见表 30-2 了解外设功能编号)

寄存器 30-23: **RPOR5: 外设引脚选择输出寄存器 5**

U-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	—	RP11R<4:0>				
bit 15							bit 8

U-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	—	RP10R<4:0>				
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 15-13 **未实现:** 读为 0
- bit 12-8 **RP11R<4:0>:** 将外设输出功能分配给 RP11 输出引脚的位 (请参见表 30-2 了解外设功能编号)
- bit 7-5 **未实现:** 读为 0
- bit 4-0 **RP10R<4:0>:** 将外设输出功能分配给 RP10 输出引脚的位 (请参见表 30-2 了解外设功能编号)

寄存器 30-24: RPOR6: 外设引脚选择输出寄存器 6

U-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	—	RP13R<4:0>				
bit 15							bit 8

U-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	—	RP12R<4:0>				
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 15-13 **未实现:** 读为 0

bit 12-8 **RP13R<4:0>:** 将外设输出功能分配给 RP13 输出引脚的位 (请参见表 30-2 了解外设功能编号)

bit 7-5 **未实现:** 读为 0

bit 4-0 **RP12R<4:0>:** 将外设输出功能分配给 RP12 输出引脚的位 (请参见表 30-2 了解外设功能编号)

寄存器 30-25: RPOR7: 外设引脚选择输出寄存器 7

U-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	—	RP15R<4:0>				
bit 15							bit 8

U-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	—	RP14R<4:0>				
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 15-13 **未实现:** 读为 0

bit 12-8 **RP15R<4:0>:** 将外设输出功能分配给 RP15 输出引脚的位 (请参见表 30-2 了解外设功能编号)

bit 7-5 **未实现:** 读为 0

bit 4-0 **RP14R<4:0>:** 将外设输出功能分配给 RP14 输出引脚的位 (请参见表 30-2 了解外设功能编号)

注: 一些 dsPIC33F 器件可能具有额外的 RPORx (外设引脚选择输出) 寄存器。更多信息, 请参见具体器件数据手册。

30.8 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。当前与带外设引脚选择的 I/O 端口相关的应用笔记包括：

标题	应用笔记编号
Implementing Wake-Up on Key Stroke	AN552

<p>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)。</p>
--

30.9 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 6 月）

该版本包括以下更新：

- **例 30-2：“配置 UART1 输入和输出功能”** 已经用全新内容更新了。
- 对整篇文档进行了其他少量修正，如语言和格式的更新。

注: