LED 数据手册 LED V 2.00

Copyright © 2005-2014 Cypress Semiconductor Corporation. All Rights Reserved.

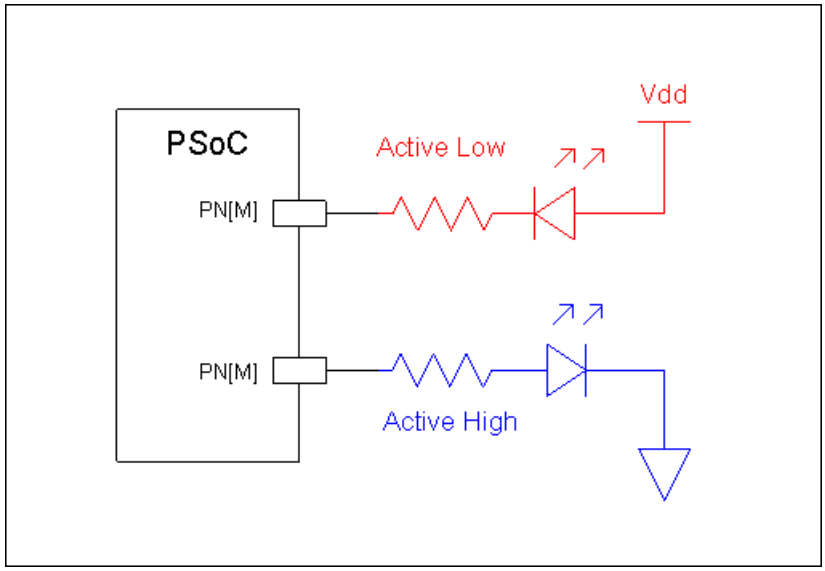
资源	PSoC® 模块			API 存储器 （字节）		引脚数量 （针对每个外部 I/O）
	数字	模拟 CT	模拟 SC	闪存	RAM	
所有 PSoC 器件	0	0	0	40	1	1

特性与概述

- 支持高电平有效和低电平有效电路
- 支持与系统的映像寄存器结合使用
- 功能 （切换、反转以及 GetState）

LED 用户模块仅拥有少量的功能，用于控制 LED 或由 “On” 和 “Off” 状态控制的简单器件。

图 1. LED 框图



功能说明

通过 LED 用户模块，您不再需要考虑映像寄存器但仍能够方便地打开或关闭引脚。

如果项目使用了端口上控制引脚的 LED 用户模块 （其他 LED 用户模块也共同使用该端口），那么必须避免进行直接写入 PRTxDR 的操作。应该将映像寄存器用于阻止 LED 用户模块执行的错误操作。

放置

可以将 LED 放置在任何一个 I/O 引脚上。特定引脚的驱动模式被自动设置为 Strong （强驱动）。

参数与资源

端口

选择 LED 将连接的端口。

引脚

选择 LED 将连接的引脚。

驱动

对 LED 用户模块进行配置，用以驱动 “ 高电平有效 ” 或 “ 低电平有效 ” 的配置。在 “ 高电平有效 ” 配置中， LED 的阳极通过一个电阻器与 PSoC 引脚连接，同时 LED 的阴极被连接到 Vss。在 “ 低电平有效 ” 配置中， LED 的阴极通过一个电阻器与 PSoC 引脚连接，同时 LED 的阳极被连接到 Vss。

应用程序接口 （**API**）

所提供的应用程序接口 （API）函数作为用户模块的一部分，设计人员通过它可以采用更高级的方式来处理模块。本节指定了每个函数的接口以及 “include” 文件中所提供的相关常量。

注意：

这些内容同所有用户模块 API 中的一样，可以通过调用 API 函数更改 A 和 X 寄存器的值。如果调用后需要使用 A 和 X 的值，则调用函数需要保留调用前 A 和 X 的值。选择该 “ 寄存器易失 ” 策略可以提高效率，自 PSoC Designer 1.0 版起已强制使用该策略。 C 编译器会自动遵循该要求。汇编语言程序员也必须确保他们的代码遵守该策略。虽然某些用户模块 API 函数可以保持 A 和 X 不变，但无法保证它们将来也会如此。

对于大型储存器模型器件，调用程序需要保留 CUR_PP、 IDX_PP、 MVR_PP 以及 MVW_PP 等寄存器中的所有值。尽管目前某些寄存器未被修改，但无法保证在将来的版本中也会如此。

此处介绍的是提供给 LED 用户模块的 API 函数：

LED_Start， LED_Stop

说明：

这两个函数都能实现同一个目的：关闭 LED。

C 原型：

```
void  LED_Start(void)
void  LED_Stop(void)
```

汇编程序：

```
lcall  LED_Start
lcall  LED_Stop
```

返回值：

无

其他影响：

无

LED_Switch

说明：

打开或关闭 LED。

C 原型：

```
void  LED_Switch(BYTE bOnOff)
```

汇编程序：

```
Mov  a,0x01    ; Turn on LED
lcall   LED_Switch
```

参数：

bOnOff: 0 = Off, Non-Zero = On

返回值：

无

其他影响：

无

LED_On

说明：

打开 LED。

C 原型：

```
void  LED_On(void)
```

汇编程序：

```
lcall   LED_On
```

参数：

无

返回值：

无

其他影响：

无

LED_Off

说明：

关闭 LED。

C 原型：

```
void  LED_Off(void)
```

汇编程序：

```
lcall  LED_Off
```

参数：

无

返回值：

无

其他影响：

无

LED_Invert

说明：

转换 LED 的状态。如果 LED 已经打开，那么它将被关闭；如果 LED 已经关闭，那么它将被打开。

C 原型：

```
void  LED_Invert(void)
```

汇编程序：

```
lcall  LED_Invert
```

参数：

无

返回值：

无

其他影响：

无

LED_GetState

说明：

返回 LED 的状态。

C 原型：

```
BYTE  LED_GetState(void)
```

汇编程序：

```
lcall  LED_GetState
mov   [myLED_State],A  ; Place result in location myLED_State
```

参数：

无

返回值：

返回 LED 的状态。如果 LED 处于关闭状态，将返回 “0”。如果 LED 处于打开状态，将返回 “1”。下面显示的是使用于 C 和 ASM 的一些符号名。

符号名	数值
LED_ON	1
LED_OFF	0

其他影响：

无

示例固件源代码

下面是使用汇编代码编写的示例项目：

```
;;; Sample ASM Code for the LED User Module
;;;
;;; Turn off LED at start up then turn it on, WOW!
;;;

include "m8c.inc"       ; part specific constants and macros
include "PSoCAPI.inc"   ; PSoC API definitions for all User Modules

area text(ROM,REL)
export _main

_main:

   mov  a,0x00          ; Turn Off LED
   call  LED_Switch
                        ; Do user stuff
   mov  A,0x01          ; Turn On LED
   call  LED_Switch
```

下面是使用 C 语言代码编写的示例项目：

```
//--------------------------------------------------------
//  Sample C Code for the LED
//  Turn LED off then, then stay in a loop and invert
//  its state.
//
//
//-------------------------------------------------------------------------
#include <m8c.h>          // part specific constants and macros
#include "PSoCAPI.h"      // PSoC API definitions for all User

void main(void)
{
   LED_Start();

   LED_Switch(1);         // Turn on LED

   while(1) {
      LED_Invert();       // Flash LED
}
}
```

配置寄存器

无

版本历史记录

版本	修订人	说明
1.2	DHA	添加了版本历史记录。
1.3	DHA	改进了引脚的选择，用以排除被使用的引脚。
1.40	DHA	添加了 DRC 的警告信息。当多个 LED 用户模块共同使用同一个引脚时，将出现此警告信息。

注意： PSoC Designer 5.1 介绍了所有用户模块数据手册中的 “ 版本历史记录 ”。本数据手册详细介绍了当前和先前的用户模块版本之间存在的区别。

Copyright © 2005-2014 赛普拉斯半导体公司。此处所包含的信息可能会随时更改，恕不另行通知。除赛普拉斯产品内嵌的电路外，赛普拉斯半导体公司不对任何其他电路的使用承担任何责任。也不会根据专利权或其他权利以明示或暗示的方式授予任何许可。除非与赛普拉斯签订明确的书面协议，否则赛普拉斯产品不保证能够用于或适用于医疗、生命支持、救生、关键控制或安全应用领域。此外，对于合理预计会发生运行异常和故障并对用户造成严重伤害的生命支持系统，赛普拉斯将不批准将其产品用作此类系统的关键组件。若将赛普拉斯产品用于生命支持系统，则表示制造商将承担因此类使用而招致的所有风险，并确保赛普拉斯免于因此而受到任何指控。

PSoC Designer™ 和 Programmable System-on-Chip™ 是赛普拉斯半导体公司的商标， PSoC® 是赛普拉斯半导体公司的注册商标。此处引用的所有其他商标或注册商标归其各自所有者所有。

所有源代码 （软件和 / 或固件）均归赛普拉斯半导体公司 （赛普拉斯）所有，并受全球专利法规 （美国和美国以外的专利法规）、美国版权法以及国际条约规定的保护和约束。赛普拉斯据此向获许可者授予适用于个人的、非独占性、不可转让的许可，用以复制、使用、修改、创建赛普拉斯源代码的派生作品、编译赛普拉斯源代码和派生作品，并且其目的只能是创建自定义软件和 / 或固件，以支持获许可者仅将其获得的产品依照适用协议规定的方式与赛普拉斯集成电路配合使用。除上述指定用途外，未经赛普拉斯的明确书面许可，不得对此类源代码进行任何复制、修改、转换、编译或演示。

免责声明：赛普拉斯不针对该材料提供任何类型的明示或暗示保证，包括 （但不仅限于）针对特定用途的适销性和适用性的暗示保证。赛普拉斯保留在不另行通知的情况下对此处所述材料进行更改的权利。赛普拉斯不对此处所述之任何产品或电路的应用或使用承担任何责任。对于合理预计可能发生运转异常和故障，并对用户造成严重伤害的生命支持系统，赛普拉斯不授权将其产品用作此类系统的关键组件。若将赛普拉斯产品用于生命支持系统，则表示制造商将承担因此类使用而导致的所有风险，并确保赛普拉斯免于因此而受到任何指控。

产品使用可能受适用于赛普拉斯软件许可协议的限制。