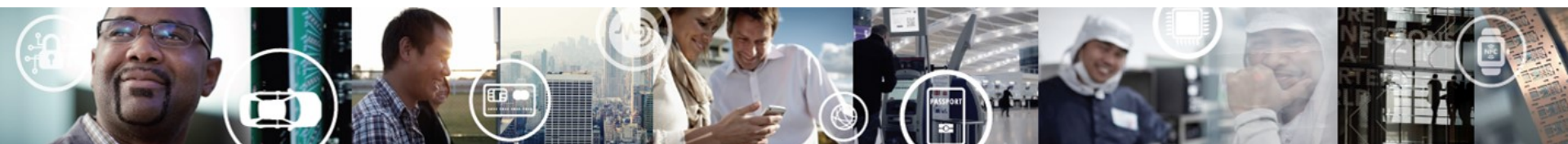


LPC82X 培训资料

开关矩阵

MAY, 2016



EXTERNAL USE



SECURE CONNECTIONS
FOR A SMARTER WORLD

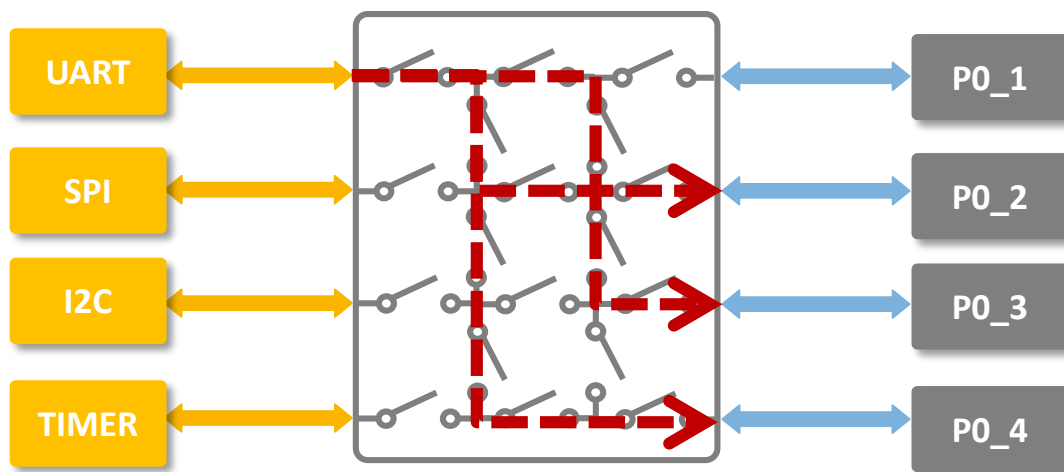
内容

- 开关矩阵 简介
- 开关矩阵 配置及寄存器
- 开关矩阵 LPCOPEN API
- 开关矩阵 物理特性
- 开关矩阵 软件配置工具
- 开关矩阵 手动配置

开关矩阵 简介

LPC82x 开关矩阵 简介

- 开关矩阵可分配的引脚和功能
 - 除了电源和地管脚以外，任意外部管脚均可被SWM分配功能
 - 可分配的功能包括 UART、SPI、I2C、SCT、比较器输出、CLKOUT、逻辑匹配输出
- 特定功能引脚仅可被管理（不能被分配）
 - 如外部晶振引脚、比较器输入口、ADC采样、复位口、SWD调试接口、GPIO口



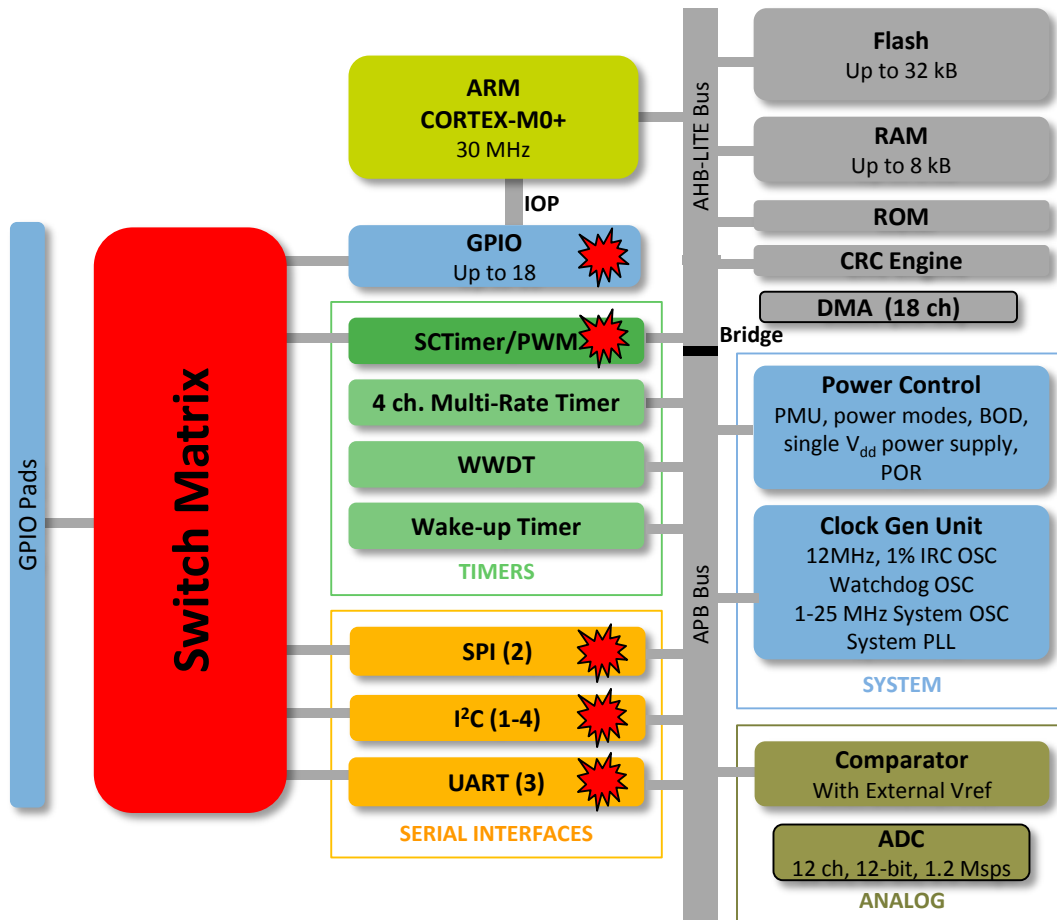
开关矩阵功能示意图

LPC82x 开关矩阵 优势

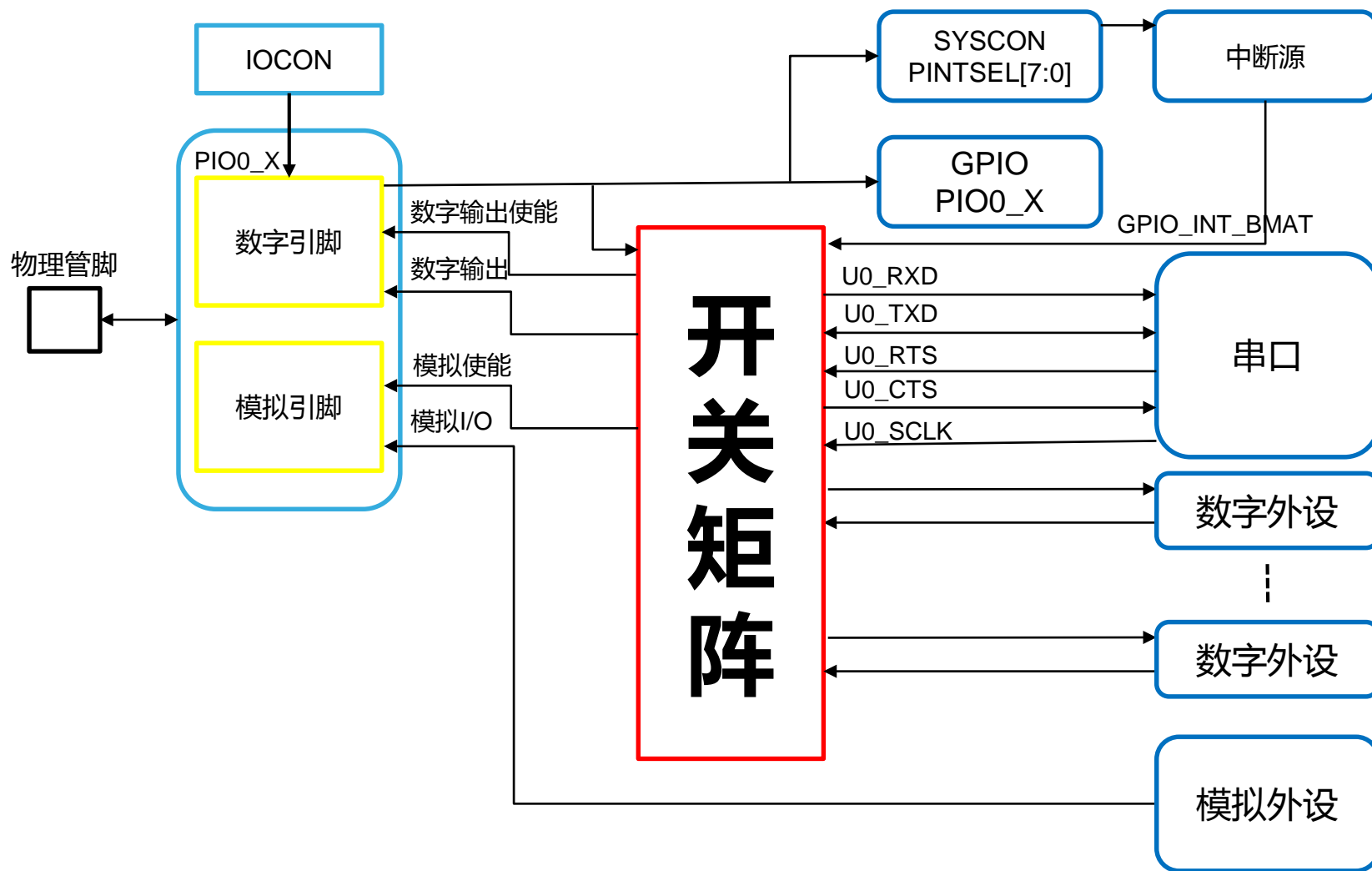
- 开关矩阵可以将外设功能灵活的分配到指定的管脚
- 解决原理图设计错误
 - 在原理图管脚功能设计有误的情况下，可以通过软件重新分配外设功能
- 优化硬件布线设计
 - 根据布线需求合理的分配外设功能到各个管脚
- 实时动态改变管脚上的外设功能
- 合理的情况下分配多个外设功能到指定的管脚

LPC82x 开关矩阵 在芯片上的位置

- 开关矩阵可以管理以下外设
 - GPIO
 - SCT定时器
 - SPI
 - I2C
 - UART
- 开关矩阵配置好后，不需要时钟源

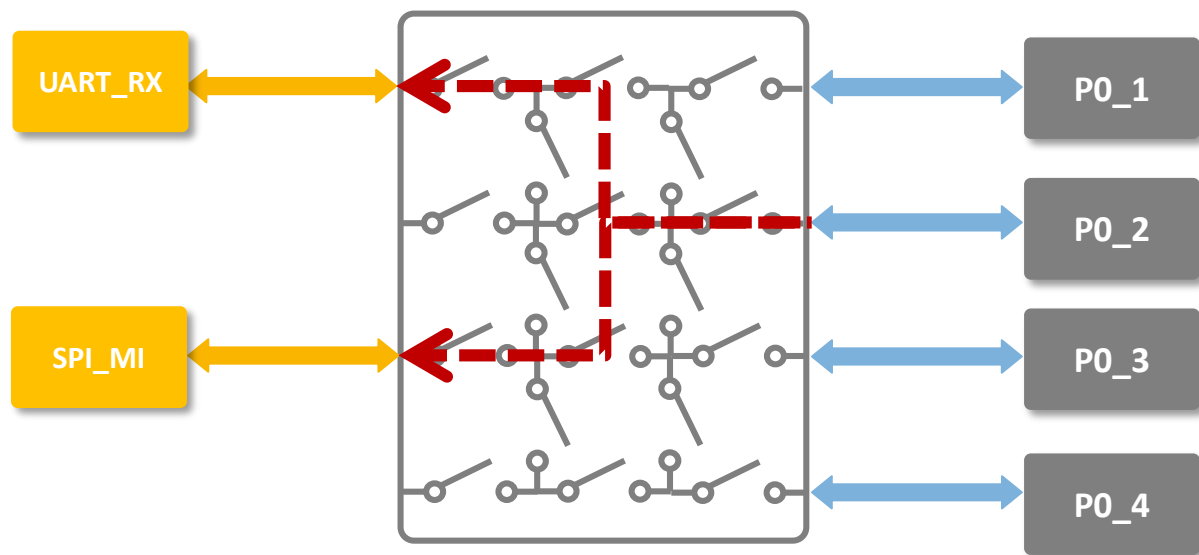


LPC82x 开关矩阵 功能框图



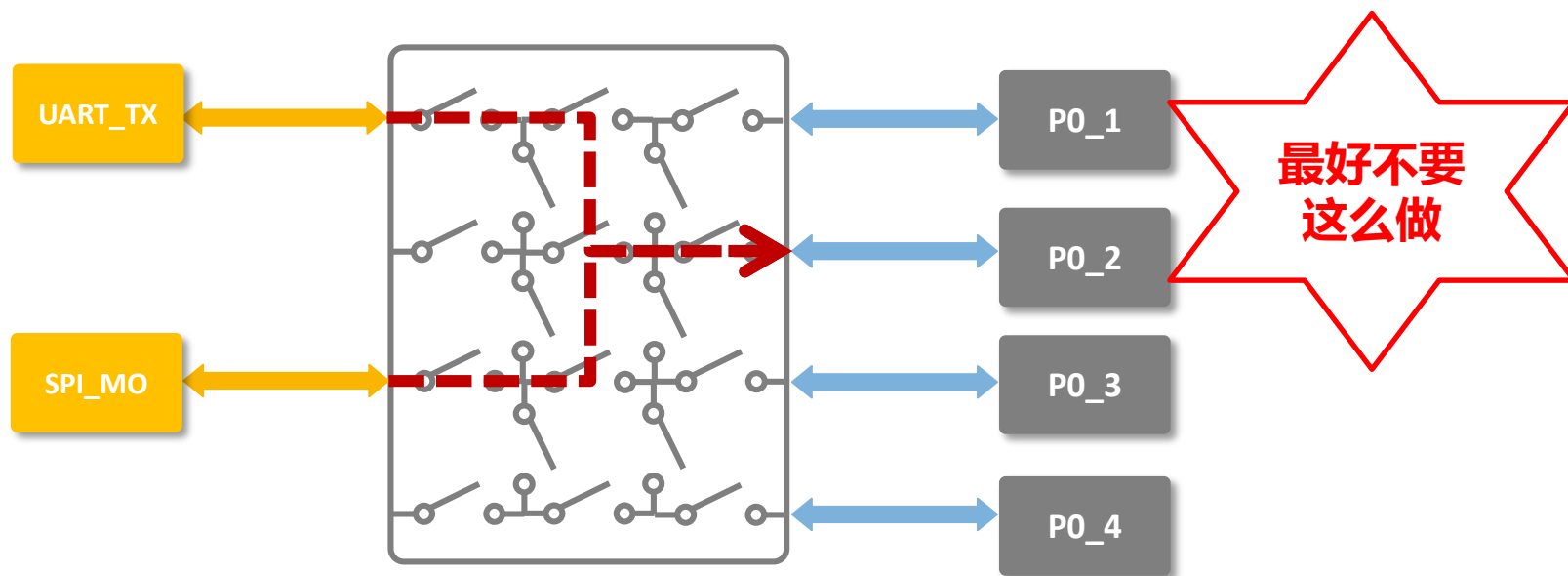
LPC82x 配置多种外设功能到同一个管脚

- 初始化分配好后，在接下来的程序中仍然可以修改开关矩阵的配置
- 多种输入功能分配到一个管脚
 - 最优的解决方式是根据用户的需求分时切换不同的功能到指定的I/O管脚



LPC82x 配置多种外设功能到同一个管脚

- 多项输出功能分配到一个管脚
 - 开关矩阵是一组复用的逻辑关系，不允许多路信号驱动一个输出缓冲
 - 如果使能了多组输出到一个管脚，则信号会被混合在一起



开关矩阵 配置及寄存器

设置开关矩阵的主要步骤



LPC82x 开关矩阵 基础设置

- 默认状态，只有RESET和串口0被分配到了对应的管脚上
 - 调试无关的引脚，默认状态下均为GPIO
 - 在配置I/O管脚上的功能前，需要通过PINENABLE0寄存器先禁用，然后再分配功能
 - 一旦I/O管脚被设置为ADC功能，则对应的数字功能自动从该管脚上移除
- 在配置开关矩阵的寄存器时，是需要打开系统时钟的
 - 在设置完开关矩阵后，关闭其时钟源可以降低系统的功耗
 - 锁住I/O配置可以保护设置
 - SYSAHBCLKCTRL（第7位）中矩阵开关时钟的控制位
- 在使能或者使用外设前，需要先通过设置PINASSIGN[0:11]寄存器配置好外设对应的管脚
- 注意：如果分配外设功能到SWD相关的引脚，会禁用SWD功能



LPC82x 管脚分配 寄存器 PINASSIGN[0:11]

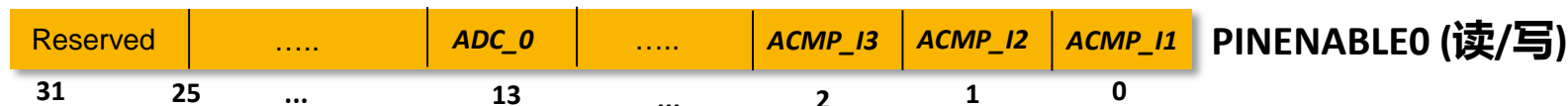
- 配置外设功能连接到对应I/O管脚
- CMSIS 标准函数名称: **LPC_SWD->PINASSIGNn**
- LPCOPEN API 名称: **Chip_SWM_MovablePinAssign**
- 举例:



位	功能	值	描述	复位
0:7	UART0_TXD_O	0..28	分配UART0_TXD功能。数值代表着分配该功能到该值的I/O管脚上。 PIO0_0 = 0(0x00), ..., PIO0_28 = 28(0x1C)。	0xFF
8:15	UART0_RXD_I	0..28	分配UART0_RXD功能。数值代表着分配该功能到该值的I/O管脚上。 PIO0_0 = 0(0x00), ..., PIO0_28 = 28(0x1C)。	0xFF
16:23	UART0_RTS_O	0..28	分配UART0_RTS_O功能。数值代表着分配该功能到该值的I/O管脚上。 PIO0_0 = 0(0x00), ..., PIO0_28 = 28(0x1C)。	0xFF
24:31	UART0_CTS_I	0..28	分配UART0_CTS_I功能。数值代表着分配该功能到该值的I/O管脚上。 PIO0_0 = 0(0x00), ..., PIO0_28 = 28(0x1C)。	0xFF

LPC82x 管脚功能使能 寄存器 PINENABLE0

- 特定功能可通过该寄存器禁用或使能并连接到特定的I/O管脚
- CMSIS 标准函数名称: **LPC_SWD->PINENABLE0**
- LPCOPEN API 名称: **Chip_SWM_EnableFixedPin**
- 举例:



位	功能	值	描述	复位
0	ACMP_I1		ACMP_I1模拟比较器输入1	1
		0	禁用	
		1	使能	
1	ACMP_I2		ACMP_I2模拟比较器输入2	1
		0x0	禁用	
		0x1	使能	
...

LPC82x 可以被开关矩阵分配的外设功能-1

- 每组外设配置寄存器均有对应的详细描述
- 参考 LPC82x数据手册 第7章 Table4

外设功能名称	类型	描述	SWM 分配寄存器
U0_TXD	输出	USART0 数据 发送	PINASSIGN0
U0_RXD	输入	USART0 数据 接收	PINASSIGN0
U0_RTS	输出	USART0 数据 发送请求	PINASSIGN0
U0_CTS	输入	USART0 数据 接收请求	PINASSIGN0
U0_SCLK	输入/输出	USART0 同步模式 串行时钟输入或输出	PINASSIGN1
U1_TXD	输出	USART1 数据 发送	PINASSIGN1
U1_RXD	输入	USART1 数据 接收	PINASSIGN1
U1_RTS	输出	USART1 数据 发送请求	PINASSIGN1
U1_CTS	输入	USART1 数据 接收请求	PINASSIGN2
U1_SCLK	输入/输出	USART1 同步模式 串行时钟输入或输出	PINASSIGN2
U2_TXD	输出	USART2 数据 发送	PINASSIGN2
U2_RXD	输入	USART2 数据 接收	PINASSIGN2
U2_RTS	输出	USART2 数据 发送请求	PINASSIGN3
U2_CTS	输入	USART2 数据 接收请求	PINASSIGN3
U2_SCLK	输入/输出	USART2 同步模式 串行时钟输入或输出	PINASSIGN3
SPIO_SCK	输入/输出	SPIO 时钟	PINASSIGN3
SPIO_MOSI	输入/输出	SPIO 主发从收	PINASSIGN4
SPIO_MISO	输入/输出	SPIO 从发主收	PINASSIGN4
SPIO_SSEL0	输入/输出	SPIO 片选0	PINASSIGN4
SPIO_SSEL1	输入/输出	SPIO 片选1	PINASSIGN4

LPC82x 可以被开关矩阵分配的外设功能-2

外设功能名称	类型	描述	SWM 分配寄存器
SPI0_SSEL2	输入/输出	SPI0 片选2	PINASSIGN5
SPI0_SSEL3	输入/输出	SPI0 片选3	PINASSIGN5
SPI1_SCK	输入/输出	SPI1 时钟	PINASSIGN5
SPI1_MOSI	输入/输出	SPI1 主发从收	PINASSIGN5
SPI1_MISO	输入/输出	SPI1 从发主收	PINASSIGN6
SPI1_SSEL0	输入/输出	SPI1 片选0	PINASSIGN6
SPI1_SSEL1	输入/输出	SPI1 片选1	PINASSIGN6
SCT_PIN0	输入	SCT定时器 输入0	PINASSIGN6
SCT_PIN1	输入	SCT定时器 输入1	PINASSIGN7
SCT_PIN2	输入	SCT定时器 输入2	PINASSIGN7
SCT_PIN3	输入	SCT定时器 输入3	PINASSIGN7
SCT_OUT0	输出	SCT定时器 输出0	PINASSIGN7
SCT_OUT1	输出	SCT定时器 输出1	PINASSIGN8
SCT_OUT2	输出	SCT定时器 输出2	PINASSIGN8
SCT_OUT3	输出	SCT定时器 输出3	PINASSIGN8
SCT_OUT4	输出	SCT定时器 输出4	PINASSIGN8
SCT_OUT5	输出	SCT定时器 输出5	PINASSIGN9
I2C1_SDA	输入/输出	I2C1 数据	PINASSIGN9
I2C1_SCL	输入/输出	I2C1 时钟	PINASSIGN9
I2C2_SDA	输入/输出	I2C2 数据	PINASSIGN9
I2C2_SCL	输入/输出	I2C2 时钟	PINASSIGN10
I2C3_SDA	输入/输出	I2C3 数据	PINASSIGN10
I2C3_SCL	输入/输出	I2C3 时钟	PINASSIGN10
ADC_PINTRIG0	输入	ADC外部触发输入0	PINASSIGN10
ADC_PINTRIG1	输入	ADC外部触发输入1	PINASSIGN11
ACMP_0	输出	模拟比较器 输出	PINASSIGN11
CLKOUT	输出	时钟信号输出	PINASSIGN11
GPIO_INT_BMAT	输出	逻辑匹配输出	PINASSIGN11

开关矩阵的物理属性

LPC82x 开关矩阵 功耗

- 参考 LPC82x数据手册 第12章 Table10

外设	典型功耗 μA			注释
	MCU主频			
	N/A	12Mhz	30Mhz	
开关矩阵	-	59	145	

- 在配置开关矩阵的寄存器时，是需要打开系统时钟的
- 在设置完开关矩阵后，关闭其时钟源可以降低系统的功耗

开关矩阵 图形配置工具

LPC82x 开关矩阵配置工具



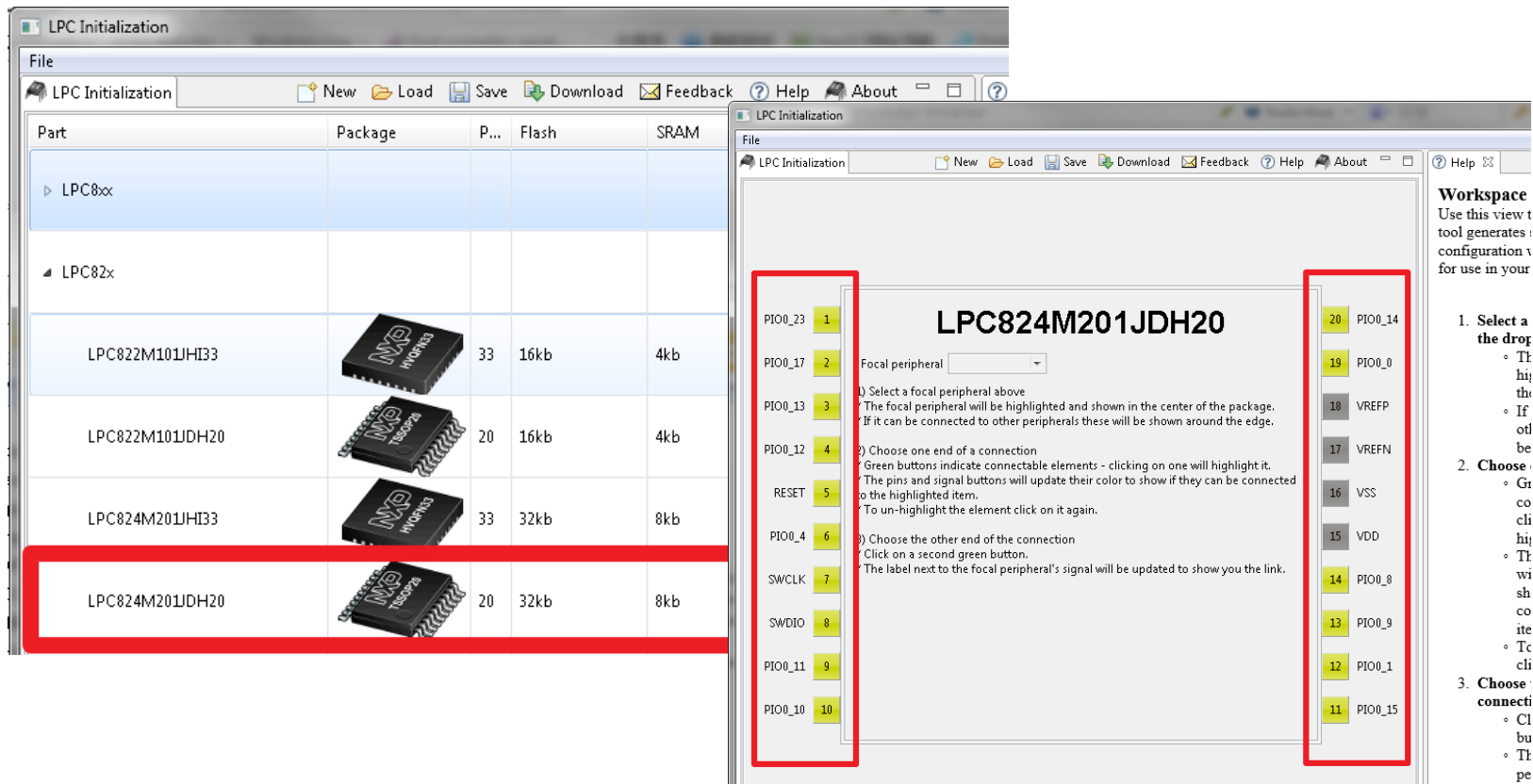
- 配置非常简单易用！
- 用户可以使用图形界面工具生成配置文件，或者手工通过软件配置
- 通过以下链接下载LPC82x开关矩阵图形配置工具
- 开关矩阵 配置工具，支持网页版，Windows，Linux和MAC
- 本地版的配置工具，需要预先安装[JAVA](#)

<https://www.lpcware.com/content/nxpfile/lpc820-pinmuxing-tool>

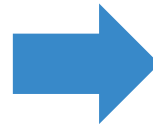
Utility	Downloads/More information	Function
LPCScript	Downloads for Windows, Linux or MacOS	LPC18xx and LPC43xx OTP and Flash Programming. Now also provides functionality to program LPC-Link2 and LPCXpresso V2/V3 firmware(replacing LCT)
LPC Initializer and Pinmux	Web app Windows:32-bit 64-bit Linux:32-bit 64-bit Mac OS X 64-bit	Pin configuration tool. Supports:LPC8xx, LPC11E6x, LPC11U6x, LPC15xx, LPC541xx

LPC82x 使用开关矩阵配置工具-1

- 第一步：选择对应封装的器件，选择完器件/封装后，会显示管脚默认分配的功能。



- 第二步：点击“Focal Peripheral”选择需要配置的外设，这里我们以USART1功能为例，选择好外设后，会弹出需要定义的功能。



LPC82x 使用开关矩阵配置工具-3

- 第三步：鼠标选择需要配置的引脚功能（选中后，功能绿色方格会变成橙色）。然后，鼠标选择需要分配到的I/O管脚，即可。这里，以USART1_RXD功能分配到PIO0_14为例。



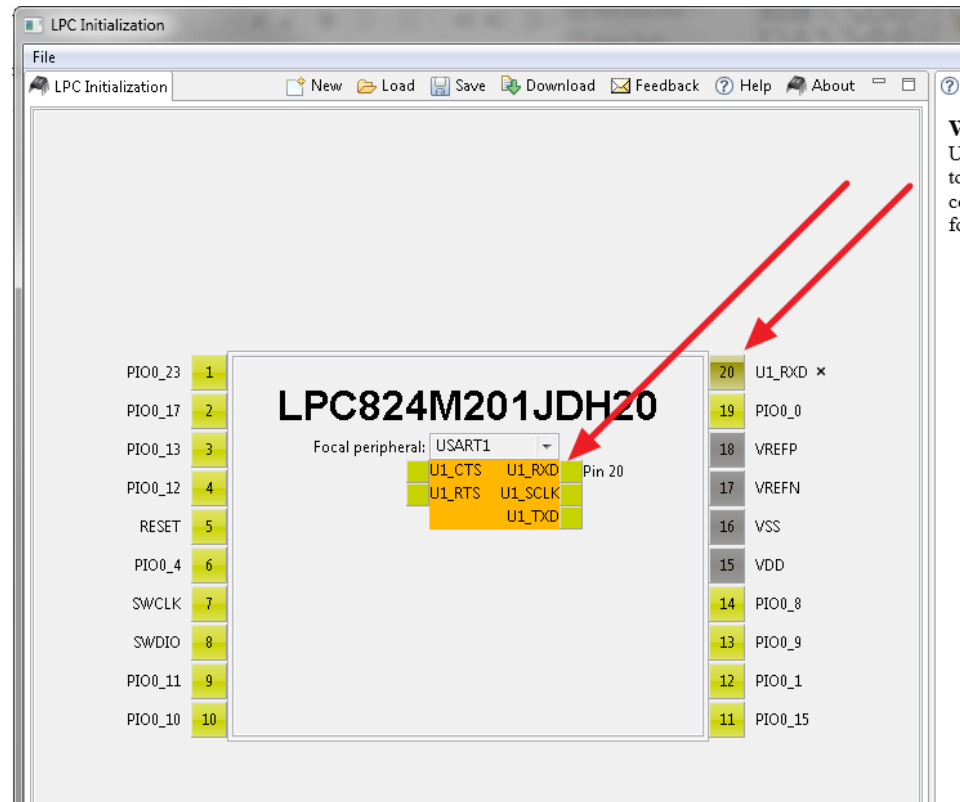
LPC82x 使用开关矩阵配置工具-4

- 第三步也可以先选择需要设置的管脚，然后在这里，以 USART1_RXD功能分配到PIO0_14为例。



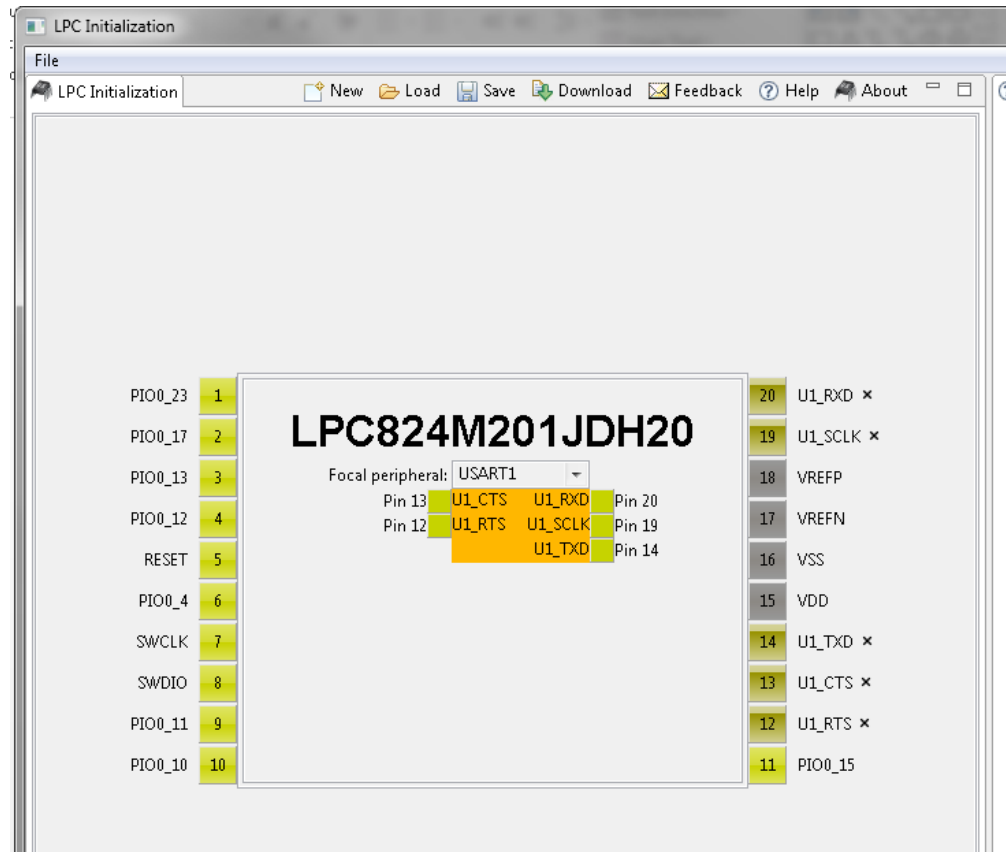
LPC82x 使用开关矩阵配置工具-5

- 此时，U1_RXD功能引脚就被分配到原先的PIO0_14的位置



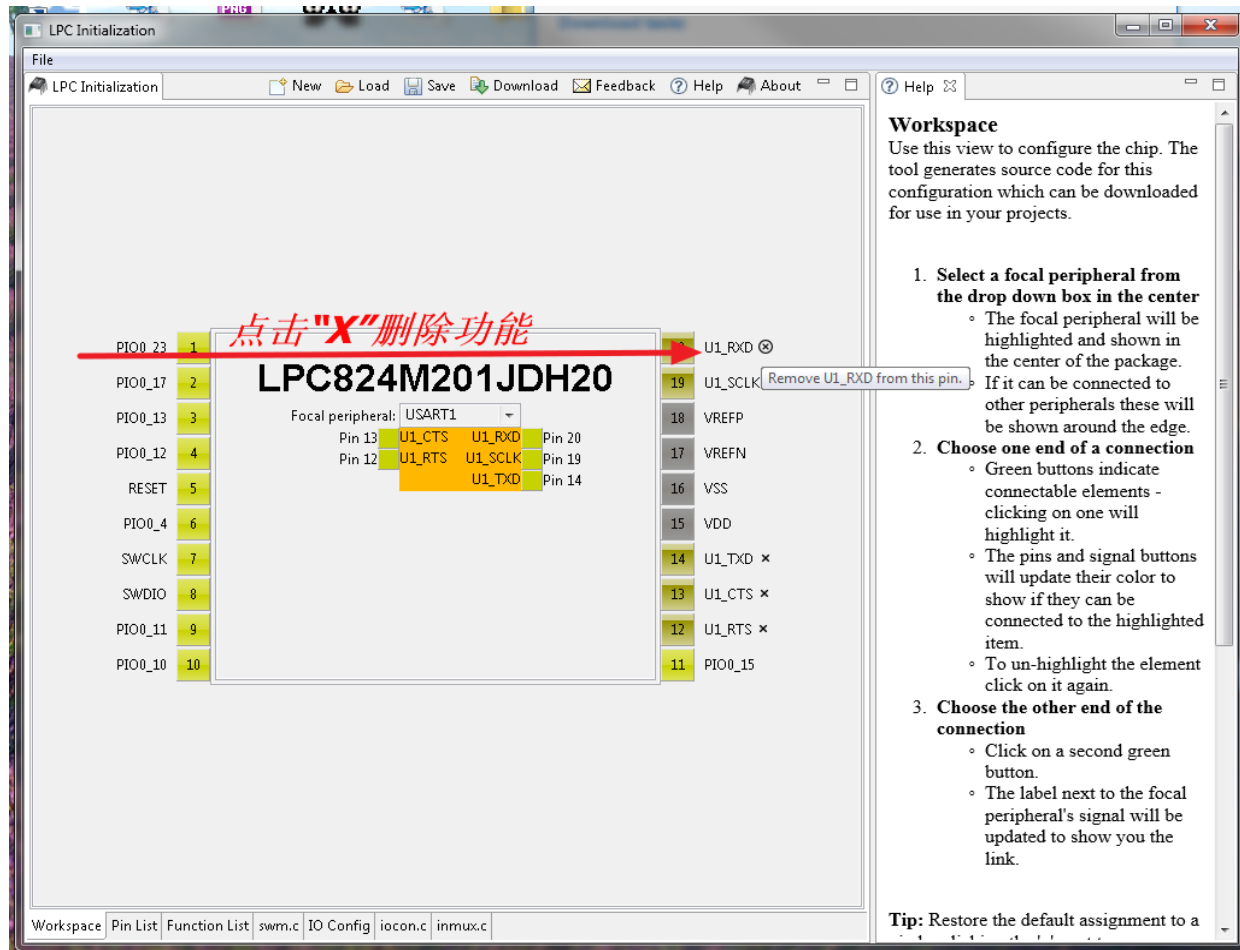
LPC82x 使用开关矩阵配置工具-6

- 第四步，按照前述步骤三，分配好用户所需要的功能。



LPC82x 使用开关矩阵配置工具-7

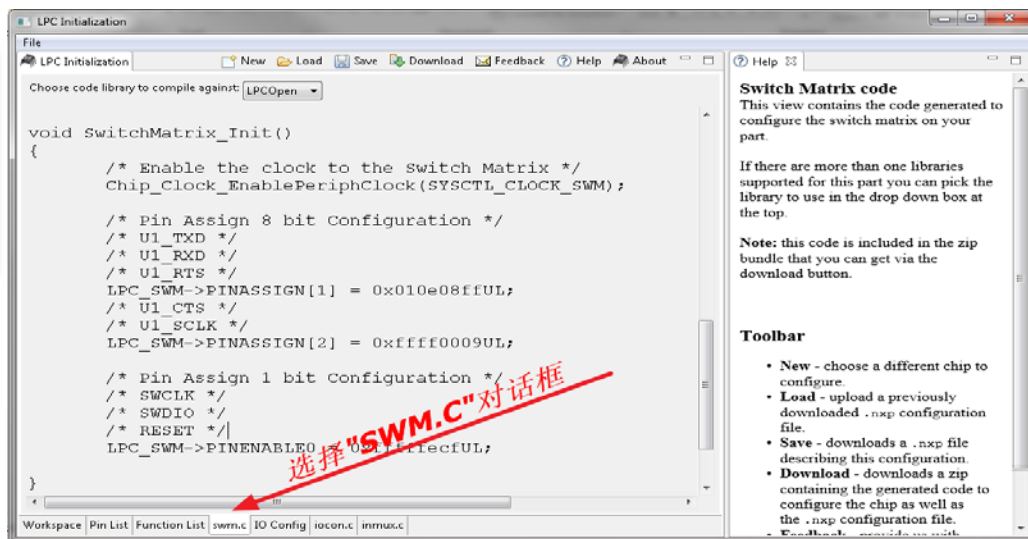
- 若需要删除分配的功能，则点击对应管脚上的“X”图标即可



LPC82x 使用开关矩阵配置工具-8

- 用户选择“SWM.C”对话框后，可以找到最终生成的初始化程序
- 直接复制到用户的代码中并调用即可配置好开关矩阵
- 别忘了关闭开关矩阵的时钟源，以降低系统功耗!
 - Chip_Clock_DisablePeriphClock(SYSCTL_CLOCK_SWM)

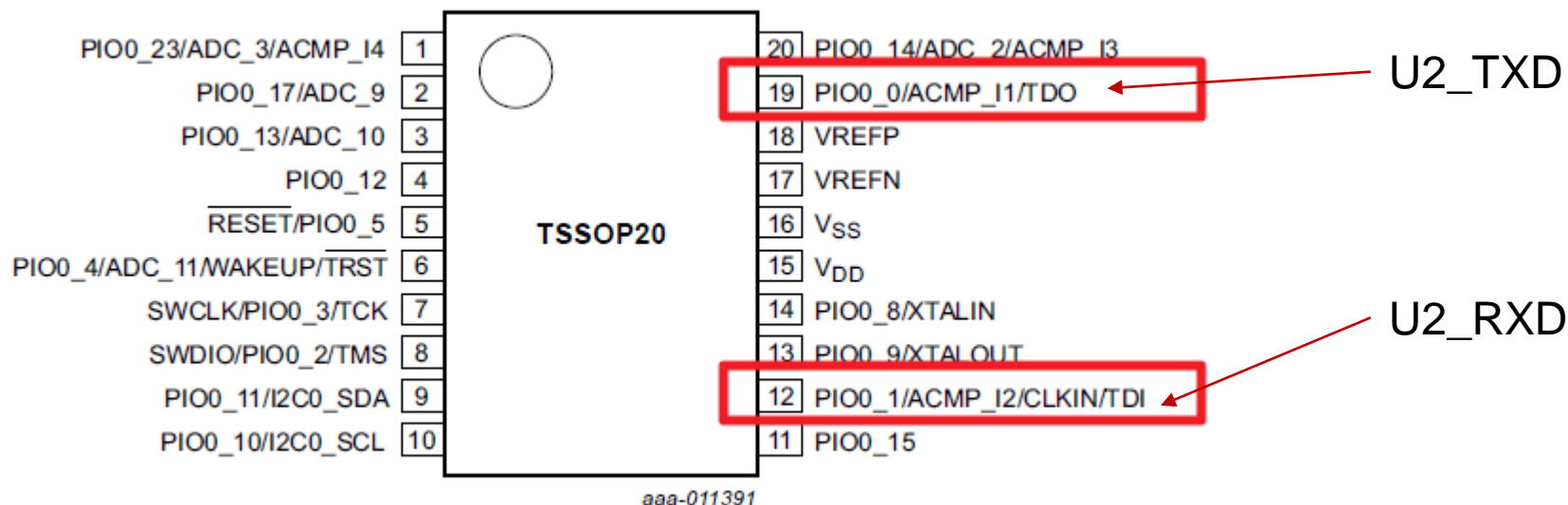
提示：为了降低功耗，用户在配置完开关矩阵后可以关闭其时钟源



开关矩阵 手动配置

LPC82x 手动配置开关矩阵-1

- 本章节介绍手工配置开关矩阵。开关矩阵相关寄存器比较简单，PINASSIGN[0:11]寄存器组和PINENABLE0寄存器;此外，用户还需要配置SYSAHBCLKCTRL的第7位使能和禁用开关矩阵。
- 我们以配置PIO0_0为U2_TXD和PIO0_19位U2_RXD为例:



LPC82x 手动配置开关矩阵-2

- 第一步：找对对应功能的PINASSIGN[0:11]寄存器。本例，U2_TXD和U2_RXD均对应PINASSIGN2。

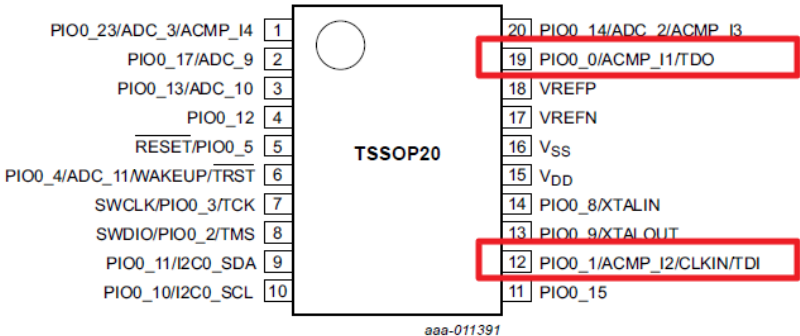
7.5.3 Pin assign register 2

Table 69. Pin assign register 2 (PINASSIGN2, address 0x4000 C008) bit description

Bit	Symbol	Description	Reset value
7:0	U1_CTS_I	U1_CTS function assignment. The value is the pin number to be assigned to this function. The following pins are available: PIO0_0 (= 0) to PIO0_28 (= 0x1C).	0xFF
15:8	U1_SCLK_IO	U1_SCLK function assignment. The value is the pin number to be assigned to this function. The following pins are available: PIO0_0 (= 0) to PIO0_28 (= 0x1C).	0xFF
23:16	U2_TXD_O	U2_TXD function assignment. The value is the pin number to be assigned to this function. The following pins are available: PIO0_0 (= 0) to PIO0_28 (= 0x1C).	0xFF
31:24	U2_RXD_I	U2_RXD function assignment. The value is the pin number to be assigned to this function. The following pins are available: PIO0_0 (= 0) to PIO0_28 (= 0x1C).	0xFF

U2_TXD

U2_RXD



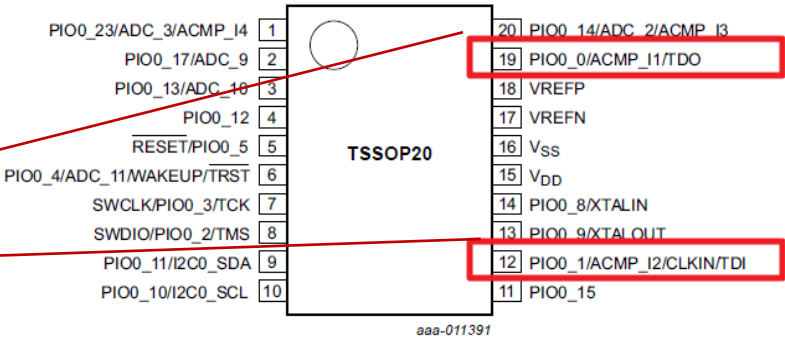
LPC82x 手动配置开关矩阵-3

- 第二步：根据需求分配的I/O管脚，设置PINASSIGN[0:11]寄存器。

7.5.3 Pin assign register 2

Table 69. Pin assign register 2 (PINASSIGN2, address 0x4000 C008) bit description

Bit	Symbol	Description	Reset value
7:0	U1_CTS_I	U1_CTS function assignment. The value is the pin number to be assigned to this function. The following pins are available: PIO0_0 (= 0) to PIO0_28 (= 0x1C).	
15:8	U1_SCLK_IO	U1_SCLK function assignment. The value is the pin number to be assigned to this function. The following pins are available: PIO0_0 (= 0) to PIO0_28 (= 0x1C).	
23:16	U2_TXD_O	U2_TXD function assignment. The value is the pin number to be assigned to this function. The following pins are available: PIO0_0 (= 0) to PIO0_28 (= 0x1C).	
31:24	U2_RXD_I	U2_RXD function assignment. The value is the pin number to be assigned to this function. The following pins are available: PIO0_0 (= 0) to PIO0_28 (= 0x1C).	



- 配置PINASSIGN2寄存器如下：

```
LPC_SWM->PINASSIGN2=(LPC_SWM->PINASSIGN2 & ~(0xFF<<16)) | (0<<16); //P0.0 = U2TxDO
LPC_SWM->PINASSIGN2=(LPC_SWM->PINASSIGN2 & ~(0xFF<<24)) | (1<<24); //P0.1 = U2RxDO
```



LPC82x 手动配置开关矩阵-4

- 最终配置源码

```
Chip_Clock_EnablePeriphClock(SYSCTL_CLOCK_SWM); // 打开 开关矩阵 时钟源
LPC_SWM->PINASSIGN2=(LPC_SWM->PINASSIGN2 & ~(0xFF<<16)) | (0<<16); // 设置 P0.0 = U2TxD0
LPC_SWM->PINASSIGN2=(LPC_SWM->PINASSIGN2 & ~(0xFF<<24)) | (1<<24); // 设置 P0.1 = U2RxD0
Chip_Clock_DisablePeriphClock(SYSCTL_CLOCK_SWM); // 关闭 开关矩阵 时钟源
```



SECURE CONNECTIONS
FOR A SMARTER WORLD