

CC2530 寄存器配置说明

ZigBee 的基础实验 (1)

P12/P13	CC2530 I/O口
UART CTS/L	P0.4
BUTTON1/LED4	P0.1
UART RX/L	P0.2
UART TX/L	P0.3
L MODE	P0.0
LED2/IR OUT	P1.1
KEY LEVEL	P0.6
LED0/L BLA	P0.7
FLASH CS	P1.3
LED1/IR IN	P1.0
DEBUG DD	P2.1
DEBUG DC	P2.2
CSN/LED3	P1.4
SCLK	P1.5
MOSI	P1.6
MISO	P1.7
LCD CS/BUTTON0	P1.2
KEY MOVE/LED1	P2.0
UART RTS/L	P0.5

这是飞比 FB2530EB V2.0 提供的芯片 I/O 对应表

```
*more607
*2011/11/17 22:13
*飞比 CC2530EB 模块
*/
#include <ioCC2530.h>
#define uint unsigned int
#define uchar unsigned char
//定义控制灯的端口
#define RLED P1_0 //定义 LED1 为 P1.0 口控制
#define GLED P1_1 //定义 LED2 为 P1.1 口控制
#define YLED P1_4 //定义 LED3 为 P1.4 口控制
#define BLEDP0_1 //定义 LED4 为 P0.1 口控制
```

```

#define S1 P0_1 //定义 S1 为 P0.1 口控制（注意：因为端口复用所以需要设置 P0DIR，
                //在程序中复用比较难，所以本程序就不用来做按键了）
#define S2 P0_3 //定义 S2 为 P0.3 口控制（我将 P10 的针脚接到 P14 针脚上，所以是 P0.3 口）
#define S6 P1_2 //定义 S6 为 P1.2 口控制
//函数声明
void InitIO(void);          //初始化 LED 控制 IO 口函数
void InitKey(void);         //初始化按键
void keyScan(void);         //按键输入
//全局变量
int times;                  //计数器
void InitIO(void) //初始化 IO 口程序
{
    P1DIR |= 0x13; //P1_0、P1_1、P1_4 定义为输出
    P0DIR |= 0x02; //P0_1 定义为输出
    RLED = 1;
    GLED = 1;
    YLED = 1;
    BLED = 1;        //将 4 盏 LED 灯都打开
}
void InitKey(void) //初始化按键
{
    P1SEL &= 0xFB; //定义为输入
    P1DIR &= 0xFB; //按钮 s6 的
    P1INP |= 0x06; //拉高电压

    P0SEL &= 0xFB; //定义为输入
    P0DIR &= 0xFB; //按钮 s6 的
    P0INP |= 0x06; //拉高电压
}
void keyScan(void)
{
    if(S6 == 0)
        times++; //增加值
    while(S6 == 0);

    if(S2 == 0)
        times=0; //清空值
    while(S2 == 0);
}
void main(void)
{
    times = 0;
    InitIO(); //初始化

```

```

while(1)          //死循环让循环内的代码不断执行
{
    keyScan();
    if(times>4)
        times = 0;
    if(times == 0)//灯全灭
    {
        RLED = 0;
        GLED = 0;
        YLED = 0;
        BLED = 0;
    }
    if(times == 1)//亮一灯
    {
        RLED = 1;
        GLED = 0;
        YLED = 0;
        BLED = 0;
    }
    if(times == 2)//亮两个灯
    {
        RLED = 1;
        GLED = 1;
        YLED = 0;
        BLED = 0;
    }

    if(times == 3)//亮三个灯
    {
        RLED = 1;
        GLED = 1;
        YLED = 1;
        BLED = 0;
    }

    if(times == 4)//全亮
    {
        RLED = 1;
        GLED = 1;
        YLED = 1;
        BLED = 1;
    }
}
}

```

来自: http://blog.163.com/j_evil/blog/static/163211317201161211362979/数据手册

POSEL (P1SEL 相同): 各个 I/O 口的功能选择, 0 为普通 I/O 功能, 1 为外设功能

D7	D6	D5	D4	D3	D2	D1	D0
P0_7 功能	P0_6 功能	P0_5 功能	P0_4 功能	P0_3 功能	P0_2 功能	P0_1 功能	P0_0 功能

P2SEL: (D0 到 D2 位) 端口 2 功能选择和端口 1 外设优先级控制
 什么是外设优先级: 当 PERCFG 分配两个外设到相同的引脚时, 需要设置这两个外设的优先级, 确定哪一个外设先被响应

D7	D6	D5	D4	D3	D2	D1	D0
未用	0: USART 0 优先 1: USART 1 优先	0: USART 1 优先 1: 定时器 3 优先	0: 定时器 1 优先 1: 定时器 4 优先	0: USART 0 优先 1: 定时器 1 优先	P2_4 功能选择	P2_3 功能选择	P2_0 功能选择

CFG: 设置部分外设的 I/O 位置, 0 为默认 I 位置 1, 1 为默认位置 2

D7	D6	D5	D4	D3	D2	D1	D0
未用	定时器 1	定时器 3	定时器 4	未用	未用	USART1	USART0

PODIR (P1DIR 相同): 设置各个 I/O 的方向, 0 为输入, 1 为输出

D7	D6	D5	D4	D3	D2	D1	D0
P0_7 方向	P0_6 方向	P0_5 方向	P0_4 方向	P0_3 方向	P0_2 方向	P0_1 方向	P0_0 方向

P2DIR : D0~D4 设置 P2_0 到 P2_4 的方向 D7、D6 位作为端口 0 外设优先级的控制

D7	D6	D5	D4	D3	D2	D1	D0
X	X	未使用	P2_4 方向	P2_3 方向	P2_2 方向	P2_1 方向	P2_0 方向

D7D6	意义
00	第 1 优先级: USART 0 第 2 优先级: USART 1 第 3 优先级: 定时器 1
01	第 1 优先级: USART 1

	第 2 优先级: USART 0 第 3 优先级: 定时器 1
10	第 1 优先级: 定时器 1 通道 0-1 第 2 优先级: USART 1 第 3 优先级: USART 0 第 4 优先级: 定时器 1 通道 2 - 3
11	第 1 优先级: 定时器 1 通道 2-3 第 2 优先级: USART 0 第 3 优先级: USART 1 第 4 优先级: 定时器 1 通道 0 - 1

P0INP (P1INP 意义相似) : 设置各个 I/O 口的输入模式, 0 为上拉/下拉, 1 为三态模式

D7	D6	D5	D4	D3	D2	D1	D0
P0_7 模式	P0_6 模式	P0_5 模式	P0_4 模式	P0_3 模式	P0_2 模式	P0_1 模式	P0_0 模式

需要注意的是: P1INP 中, 只有 D7~D2 分别设置对应 I/O 口的输入模式。D1D0 两位无作用。

P2INP: D0~D4 控制 P2_0~P2_4 的输入模式, 0 为上拉/下拉, 1 为三态;

D5~D7 设置对 P0、P1 和 P2 的上拉或下拉的选择。0 为上拉, 1 为下拉;

D7	D6	D5	D4	D3	D2	D1	D0
端口 2 选择	端口 1 选择	端口 0 选择	P2_4 模式	P2_3 模式	P2_2 模式	P2_1 模式	P2_0 模式

P0IFG (P1IFG 相同) : 终端状态标志寄存器, 当输入端口有中断请求时, 相应的标志位将置 1。

D7	D6	D5	D4	D3	D2	D1	D0
P0_7	P0_6	P0_5	P0_4	P0_3	P0_2	P0_1	P0_0

P0IEN (P1IEN 相同) : 各个控制口的中断使能, 0 为中断禁止, 1 为中断使能。

D7	D6	D5	D4	D3	D2	D1	D0
P0_7	P0_6	P0_5	P0_4	P0_3	P0_2	P0_1	P0_0

P2IFG: D0~D4 为 P2_0~P2_4 的中断标志位

D5 为 USD D+中断状态标志, 当 D+线有一个中断请求未决时设置该标志, 用于检测 USB 挂起状态事件。当 USB 控制器没有挂起时不设置该标志。

D7	D6	D5	D4	D3	D2	D1	D0
未用	未用	USB D+	P2_4	P2_3	P2_2	P2_1	P2_0

P2IEN: D0~D4 控制 P2_0~P2_4 的中断使能
D5 控制 USB D+的中断使能

D7	D6	D5	D4	D3	D2	D1	D0
未用	未用	USB D+	P2_4	P2_3	P2_2	P2_1	P2_0

PICTL: D0~D3 设置各个端口的中断触发方式，0 为上升沿触发，1 为下降沿触发。
D7 控制 I/O 引脚在输出模式下的驱动能力。选择输出驱动能力增强来补偿引脚 DVD 压，确保在较
低的电压下的驱动能力和较高电压下相同。0 为最小驱动能力增强。1 为最大驱动能力增强。

D7	D6	D5	D4	D3	D2	D1	D0
I/O 驱动能力	未用	未用	未用	P2_0~P2_4	P1_4~P1_7	P1_0~P1_3	P0_0~P0_7

IEN0: 中断使能 0, 0 为中断禁止，1 为中断使能

D7	D6	D5	D4	D3	D2	D1	D0
总中断 EA	未用	睡眠定时器中断	AES 加密/解密中断	USART1 RX 中断	USART0 RX 中断	ADC 中断	RF TX/RF FIFO 中断

IEN1:中断使能 1, 0 为中断禁止，1 为中断使能

D7	D6	D5	D4	D3	D2	D1	D0
未用	未用	端口 0	定时器 4	定时器 3	定时器 2	定时器 1	DMA 传输

IEN2: 中断使能 2, 0 为中断禁止，1 为中断使能

D7	D6	D5	D4	D3	D2	D1	D0
未用	未用	看门狗定时器	端口 1	USART1 TX	USART0 TX	端口 2	RF 一般中断

T1CTL: 定时器 1 的控制，D1D0 控制运行模式，D3D2 设置分频划分值

D7	D6	D5	D4	D3D2	D1D0
未用	未用	未用	未用	00: 不分频 01: 8 分频 10: 32 分频 11: 128 分频	00: 暂停运行 01: 自由运行, 反复从 0x0000 到 0xffff 计数 10: 模计数, 从 0x000 到 T1CC0 反复计数 11: 正计数/倒计数, 从 0x0000 到 T1CC0 反复计数并且从 T1CC0 倒计数到 0x0000

T1STAT: 定时器 1 的状态寄存器, D4~D0 为通道 4~通道 0 的中断标志, D5 为溢出标志位, 当计数到最大值时, 溢出标志位置 1。

D7	D6	D5	D4	D3	D2	D1	D0
未用	未用	溢出中断	通道 4 中断	通道 3 中断	通道 2 中断	通道 1 中断	通道 0 中断

T1CCTL0~T1CCTL4: 定时器 1 通道 0~通道 4 的工作方式设置。D1D0 为捕捉模式选择: 00 为不捕捉, 01 为上升沿捕获, 10 为下降沿捕获, 11 为上升或下降沿都捕获。

D2 位为捕获或比较的选择, 0 为捕获模式, 1 为比较模式。D5D4D3 为比较模式的选择: 000 为发生比较时输出端清 0, 001 为发生比较时输出端清 0, 010 为比较时输出翻转, 其他模式较少使用。

D7	D6	D5D4D3	D2	D1D0
未用	未用	比较模式	捕获/比较	捕捉模式

IRCON: 中断标志 4, ; 0 为无中断请求。1 为有中断请求。

D7	D6	D5	D4	D3	D2	D1	D0
睡眠定时器	必须为 0	端口 0	定时器 4	定时器 3	定时器 2	定时器 1	DMA 完成

T3CTL/T4CTL: 定时器 3 或定时器 4 的方式控制寄存器。D7D6D5 设置分频: 000 为无分频、001 为 2 分频、011 为 8 分频、100 为 16 分频、101 为 32 分频、110 为 64 分频, 111 为 128 分频。D4 为启动/停止工作。D3 位为中断使能位, 0 为禁止, 1 为使能, 默认为 1; D2 为复位, 置 1 时定时器复位。D1D0 为计数模式选择: 该位与 T1CTL 的 D1D0 位意义相同。

D7D6D5	D4	D3	D2	D1D0
分频	启动定时器	溢出中断	清除计数器	计数模式

T3CCTL0/T3CCTL1/T4CCTL0/T4CCTL1: 定时器 3 或定时器 4 的通道 0 和通道 1 的方式控制, D6 为该通道的中断使能位, 0 为禁止, 1 为使能, 默认为 1; D5~D0 与 T1CCTL0 相同

D7	D6	D5D4D3	D1	D0
未用	中断使能	比较模式	捕获/比较	捕捉模式

TIMIF：定时器 1 的溢出中断屏蔽与定时器 3、4 的中断标志。D6 为定时器 1 的溢出中断屏蔽，0 为屏蔽，1 为不屏蔽。D5~D0 为定时器 3 和 4 中各个通道的中断标志。

D7	D6	D5	D4	D3	D2	D1	D0
未用	T1 溢出中断使能	T4 通道 1 中断标志	T4 通道 0 中断标志	T4 溢出标志中断标志	T3 通道 1	T3 通道 0	T3 溢出中断标志

CLKCONCMD：时钟频率控制寄存器。

D7	D6	D5~D3	D2~D0
32KHZ 时间振荡器选择	系统时钟选择	定时器输出标记	系统主时钟选择

D7 位为 32KHZ 时间振荡器选择，0 为 32KRC 震荡，1 为 32K 晶振。默认为 1。
D6 位为系统时钟选择。0 为 32M 晶振，1 为 16M RC 震荡。当 D7 位为 0 时 D6 必须为 1。
D5~D3 为定时器输出标记。000 为 32MHZ，001 为 16MHZ，010 为 8MHZ，011 为 4MHZ，100 为 2MHZ，101 为 500KHZ，111 为 250KHZ。默认为 001。需要注意的是：当 D6 为 1 时，定时器频率最高可采用频率为 16MHZ。
D2~D0：系统主时钟选择：000 为 32MHZ，001 为 16MHZ，010 为 8MHZ，011 为 4MHZ，100 为 2MHZ，101 为 500KHZ，111 为 250KHZ。当 D6 为 1 时，系统主时钟最高可采用频率为 16MHZ。

CLKCONSTA：时间频率状态寄存器。

D7	D6	D5~D3	D2~D0
当前 32KHZ 时间振荡器	当前系统时钟	当前定时器输出标记	当前系统主时钟

D7 位为当前 32KHZ 时间振荡器频率。0 为 32KRC 震荡，1 为 32K 晶振。
D6 位为当前系统时钟选择。0 为 32M 晶振，1 为 16M RC 震荡。
D5~D3 为当前定时器输出标记。000 为 32MHZ，001 为 16MHZ，010 为 8MHZ，011 为 4MHZ，100 为 2MHZ，101 为 500KHZ，111 为 250KHZ。
D2~D0 为当前系统主时钟。000 为 32MHZ，001 为 16MHZ，010 为 8MHZ，011 为 4MHZ，100 为 2MHZ，101 为 500KHZ，111 为 250KHZ。

UOCSR：USART0 控制与状态；

D7	D6	D5	D4	D3	D2	D1	D0
模式选择	接收器使能	SPI 主/从模式	帧错误状态	奇偶错误状态	接受状态	传送状态	收发主动状态

D7 为工作模式选择，0 为 SPI 模式，1 为 USART 模式
D6 为 UART 接收器使能，0 为禁用接收器，1 为接收器使能。
D5 为 SPI 主/从模式选择，0 为 SPI 主模式，1 为 SPI 从模式。
D4 为帧错误检测状态，0 为无错误, 1 为出现出错。
D3 为奇偶错误检测，0 为无错误出现，1 为出现奇偶校验错误。
D2 为字节接收状态，0 为没有收到字节，1 为准备好接收字节。
D1 为字节传送状态，0 为字节没有被传送，1 为写到数据缓冲区的字节已经被发送。
D0 为 USART 接收/传送主动状态，0 为 USART 空闲，1 为 USART 忙碌。

U0GCR: USART0 通用控制寄存器;

D7	D6	D5	D4~D0
SPI 时钟极性	SPI 时钟相位	传送位顺序	波特率指数值

D7 为 SPI 时钟极性：0 为负时钟极性，1 为正时钟极性；
D6 为 SPI 时钟相位：
D5 为传送为顺序：0 为最低有效位先传送，1 为最高有效位先传送。
D4~D0 为波特率设置：

波特率	指数值	小数部分
2400	6	59
4800	7	59
9600	8	59
14400	8	216
19200	9	59
28800	9	216
38400	10	59
57600	10	216
76800	11	59
115200	11	216
230400	12	216

U0BAUD: 波特率控制小数部分。（取值参考上表）