



APPLICATION NOTE

PAN2416 芯片使用注意事项

Rev 1.6

Panchip Microelectronics

www.panchip.com

修订历史

版本	修订日期	描述
V1.6	2022-6-7	更新定时器使用和其它注意事项
V1.5	2021-11-11	部分程序优化，新增烧录器软件部分使用说明
V1.4	2020-4-3	新增低功耗 RC0 配置说明和部分程序优化
V1.3	2018-12-19	5325 的 IO 配置问题
V1.2	2018-08-27	新增中断和滚码描述
V1.1	2018-06-11	新增 2416Vref 描述
V1.0	2018-04-04	初始版本创建

版权所有©

上海磐启微电子有限公司

本资料内容为上海磐启微电子有限公司在现有数据资料基础上慎重编制而成，本资料中所记载的实例以正确的使用方法和标准操作为前提，使用方在应用该等实例时需充分考虑外部诸条件，上海磐启微电子有限公司不担保或确认该等实例在使用方的适用性、适当性或完整性，上海磐启微电子有限公司亦不对使用方因使用本资料所有内容而可能或已经带来的风险或后果承担任何法律责任。基于使本资料的内容更加完善等原因，上海磐启微电子有限公司保留未经预告的修改权，使用方如需获得最新的产品信息，请随时与上海磐启微电子有限公司联系。

目录

第 1 章 概述	4
第 2 章 差异对比说明	5
2.1 芯片封装	5
2.2 内部存储	5
2.3 内部基准电压（ADC 的固定电压通道）	5
2.4 引脚 RC0 输出模式	6
2.5 RF 晶振串联电阻	6
2.6 特殊寄存器配置对比	6
2.6.1 1Mbps 模式	7
2.6.2 250Kbps 模式	8
第 3 章 使用注意事项	9
3.1 编译软件	9
3.2 软件设计	9
3.3 烧录滚码	13
3.4 离线烧录	14

第 1 章 概述

PAN2416 系列芯片提供 SOP16 和 SOP14 两种封装，其中 SOP16 封装与 MX5325 芯片 pin to pin 兼容，MX5325 芯片与 PAN2416 芯片两者之间最主要的区别是 MX5325 芯片提供了 4KB 的 FLASH 和 64Bytes 的 EEPROM，而 PAN2416 提供的是更低成本的 4K*16bit 的 OTP 存储空间，未提供 EEPROM。

在进行方案设计时，考虑到调试的便捷性，可先使用 MX5325 芯片进行前期调试、样品试制等工作，整体验证通过后再使用 PAN2416 芯片进行替换。

本文档主要对 PAN2416 芯片与 MX5325 芯片之间的具体差异进行了说明，同时整理了 PAN2416 芯片使用时的一些注意事项。

第 2 章 差异对比说明

本章节主要对 PAN2416 芯片与 MX5325 芯片在芯片封装、内部存储、内部基准电压、软硬件设计等方面的具体差异进行说明。

2.1 芯片封装

表 1-1 封装类型对比

芯片型号 封装类型	芯片型号	
	PAN2416	MX5325
SOP14	有	无
SOP16	有	有

2.2 内部存储

表 1-2 内部存储对比

芯片型号 内存类型	芯片型号	
	PAN2416	MX5325
RAM	176Bytes	344Bytes
OTP	4K*16Bit	—
FLASH	—	4KBytes
EEPROM	—	64Bytes

在使用 MX5325 进行程序仿真时，注意不要使用超过 176Bytes 的 RAM 空间，否则会在移植程序至 PAN2416 时出现严重错误；

2.3 内部基准电压（ADC 的固定电压通道）

表 1-3 内部基准电压对比

芯片型号	内部基准电压
PAN2416	1.2V
MX5325	0.6V

请注意，PAN2416 内部基准源读取时请保证 **RF 工作在 ST3 模式**（即进入 TX/RX 模式），若 RF 工作在休眠模式，此时内部基准源读取结果为 0。MX5325 无此问题。

芯片引脚设置为输入后，若引脚电平 $< 1.6V$ ，则判定为低电平。

2.4 引脚 RC0 输出模式

表 1-4 引脚 RC0 输出模式对比

芯片型号	RC0 输出模式
PAN2416	开漏输出(输出高电平需外接上拉电阻，同时设置 VPP_OUT 为 ENABLE)
MX5325	推挽输出

RC0 在作为输出口时，需要外部加上拉电阻操作才能输出高。RC0 在做低功耗休眠过程时，需要配置其为输出模式，TRISC=0;若外围接了上拉，比如 LED 灯，则内部配置为 RC0=1;若外部浮空或者接地了，则配置 RC0=0。

Eg:

```
TRISC = 0x00;
TRISB = 0x00;
RC0=0;
RF_SLEEP();
system_to_sleep();
```

具体操作见第三章第六点。

2.5 RF 晶振串联电阻

表 1-5 晶振电路对比

芯片型号	晶振电路是否串联电阻
PAN2416	不需要
MX5325	需串联 510R 电阻

2.6 特殊寄存器配置对比

软件配置芯片内部 **RF 部分功能寄存器**时，需要注意 PAN2416（OTP 芯片）芯片与 MX5325（flash 版本芯片）芯片部分配置有所不同。需要在 rf297L.h 文件中对“#define IC_TYPE_MX5325”，开启该宏为仿真芯片，关闭后为 OTP 芯片。

```

29  ////////////////////////////////////////
30  ////////////////////////////////////////
31  //      芯片型号一定要注意切换，MX5325和PAN2416 RF寄存器配置和功率值不一样      //
32  ////////////////////////////////////////
33  //define      IC_TYPE_MX5325      //选择芯片型号，禁掉则为PAN2416配置
34  ////////////////////////////////////////
35

```

2.6.1 1Mbps 模式

表 1-6 1Mbps 模式寄存器推荐配置

寄存器名称	地址	PAN2416	MX5325
DEMOCAL	0x19	0x01	0x01
RF_CAL2	0x1A	0x45, 0x21, 0x3F, 0x2D, 0x5C, 0x40	0x45, 0x21, 0xeF, 0x2c, 0x5a, 0x40
DEMOCAL2	0x1B	0x0B, 0xDF, 0x02	0x0B, 0xDF, 0x02
RF_CAL	0x1E	0x16, 0x33, 0x27	0xF6, 0x37, 0x5D
BB_CAL	0x1F	0x0A, 0x6D, 0x67, 0x9C, 0x46	0x12, 0xED, 0x67, 0x9C, 0x46

2.6.2 250Kbps 模式

表 1-7 250Kbps 模式寄存器推荐配置

寄存器名称	地址	PAN2416	MX5325
DEMOCAL	0x19	0x03	0x1F
RF_CAL2	0x1A	0xD5, 0x21, 0x3B, 0x2D, 0x5C, 0x40	0xD5, 0x21, 0xEB, 0x2C, 0x5A, 0x40
DEM_CAL2	0x1B	0x0B, 0xDF, 0x02	0x0B, 0xDF, 0x02
RF_CAL	0x1E	0x06, 0x33, 0x00	0xF6, 0x37, 0x5D
BB_CAL	0x1F	0x12, 0xEC, 0x6F, 0xA9, 0x46	0x12, 0xEC, 0x6F, 0xA1, 0x46

第 3 章 使用注意事项

3.1 编译软件

- 1、在开始使用 CMS_IDE_Vx.xx 开发工具之前，必须要点击“帮助”后选择“一键注册软件”，该操作可以在编译程序时优化 ROM 占用空间，避免烧录带读取滚码的程序时出错。
- 2、如果出现无法编译，或者 IDE 注册失败的情况，请关闭电脑防火墙和杀软后再做尝试；
- 3、使用该编译软件进行调试时，仅支持最多 4 个断点，不支持单步调试；使用断点仿真时，工程所在路径不能为中文路径。

3.2 软件设计

- 1、软件中操作芯片进入休眠的指令应为 `asm("sleep");`
- 2、MX5325 与 PAN2416 芯片在软件配置 RF 部分参数和功率时有所区别，使用说明参考 2.6 章节；
- 3、WDT(watchdog timer)默认是打开的，配置工程选项时需要注意；在正常使用过程时，务必将其设置为 Disable，若要使用看门狗，则软件中配置。

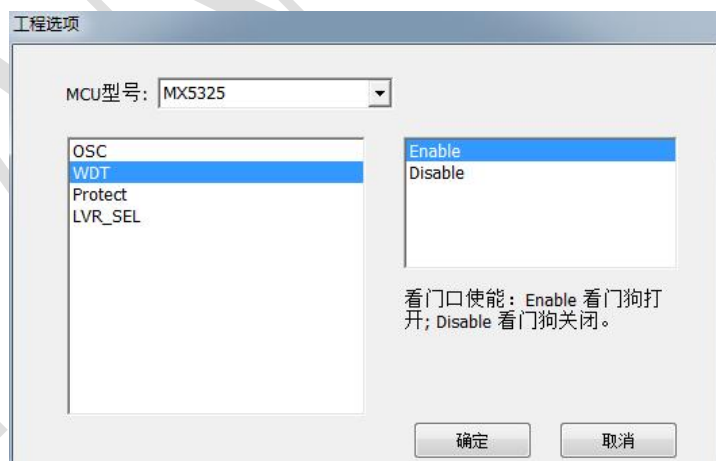


图 3-1 WDT 默认配置

- 4、ADC 上电初始化后需延时至少 100ms 等待稳定，防止多路 ADC 互相干扰。另外需将 ADC 通道引脚配置为数字输入模式而非模拟输入模式；

5、PAN2416AF 芯片（SOP14 封装）不能通过 ADC 采集外部引脚上的电压，对于需要实现低电压检测报警功能的应用，可以通过特定代码库读取芯片内部固定的基准电压 V_{ref} 来进行判断，因为 ADC 的参考电压始终由 VDD 和 GND 提供，VDD 变化时 ADC 采集到的 V_{ref} 的值也是变化的，由此可推出 VDD 的电压值。

6、芯片进入休眠和 PB 口电平中断唤醒程序示例如下：

//需要在初始化 IO 口时,配置 PB 口的电平中断,如下图;

PORTB 电平变化中断寄存器 IOCB(96H)

96H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
IOCB	IOCB7	IOCB6	IOCB5	IOCB4	IOCB3	IOCB2	IOCB1	----
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	----
复位值	0	0	0	0	0	0	0	----

Bit7-0 IOCB<7:1>: PORTB 的电平变化中断控制位

1=允许电平变化中断
0=禁止电平变化中断

```
void mcu_sleep(){
```

//弱 RC0 部分未使用，芯片配置选项选 VPPOUT 为 disanble 则无需下面阴影部分代码。

```
    TRISC = 0x00; //RC0 休眠配置输出状态
```

```
    RC0=0;// RC0 外部浮空
```

```
    //RC0=1;//RC0 外部接了上拉电阻，比如加入了 LED 灯。
```

```
    deep_sleep_rf();
```

```
    PORTB_INT_Init(0xfe);//设置 PB1~PB7 均能按键唤醒
```

```
    asm("nop");asm("nop");
```

```
    asm("SLEEP"); //至此芯片进入休眠
```

```
    asm("nop");asm("nop");asm("nop");asm("nop");
```

```
    weakup_rf();
```

```
}
```

其中 weakup_rf()函数和 deep_sleep_rf()函数代码示例如下：

```
void weakup_rf(void){
```

```
    unsigned char reg_config;
```

```
    reg_config = RF_ReadReg(CONFIG);
```

```
RF_WriteReg(W_REGISTER + CONFIG, reg_config | 0x02);  
CE_HIGH;  
delay_ms(10);  
}  
void deep_sleep_rf(void){  
    unsigned char reg_config;  
    reg_config = RF_ReadReg(CONFIG);  
    RF_WriteReg(W_REGISTER + CONFIG, reg_config & ~(0x02));  
    CE_LOW;  
}
```

休眠配置时，没用到的 IO 建议都改为输出低模式。休眠只能用 IO 唤醒或者 WDT 唤醒;详细使用说明在 CMS89F22X 用户手册 P39.

7、芯片进入中断程序示例如下：

```
void interrupt ISR(void)  
{  
    unsigned char inside_intterrupt_enable, peripheral_intterrupt_enable;  
    unsigned char inside_intterrupt_touch, peripheral_intterrupt_touch;  
    //读取中断标志  
    inside_intterrupt_enable = INTCON & 0x38;  
    peripheral_intterrupt_enable = PIE1 & 0x43;  
    inside_intterrupt_touch = INTCON & 0x07;  
    peripheral_intterrupt_touch = PIR1 & 0x43;  
    //跳转到中断函数  
    if(inside_intterrupt_touch & PORTB_CHANGE_ISR_TOUCH) { //RBIF  
        PORTB_CHANGE_ISR();  
    }  
    if(inside_intterrupt_touch & INT0_ISR_TOUCH){ //INTF  
        INT0_ISR();  
    }  
    if(inside_intterrupt_touch & TIMER0_ISR_TOUCH) { //T0IF
```

```
TIMER0_ISR();  
  
}  
  
if(peripheral_interrupt_touch & TIMER1_ISR_TOUCH) { //TMR1IF  
    TIMER1_ISR();  
}  
  
if(peripheral_interrupt_touch & TIMER2_ISR_TOUCH) { //TMR2IF  
    TIMER2_ISR();  
}  
  
if(peripheral_interrupt_touch & ADC_ISR_TOUCH) { //ADIF  
    ADC_ISR();  
}  
}
```

注意没有开启的中断，一定要把“需要判断哪些中断标志位”置 0。防止中断误判！

8、5325 的 IO 配置问题：

TRIC 寄存器不能进行位操作（位操作包含各种逻辑运算，位置 0，位置 1 等），只能整体赋值。（例如 `TRISC1 = 0; TRISC0 = 0;` 可能会导致 IO 口输出异常）

如果一定要进行位运算的话可定义一个缓存寄存器，操作完成后将缓存寄存器的值整体赋给 TRISC 的方式。

注意：目前这个 IO 配置异常问题存在 5325 芯片，2416 芯片没有存在该问题，操作 IO 配置寄存器，建议整体赋值。

9、VPP_OUTEN 的配置：

Disable 为关闭 RC0 的 IO 功能，enable 为开启 RC0 的 IO 功能。

10、timer 的时钟说明：

Timer 应用中，建议使用 1M 时钟速率（即 8 分频），1us 时钟计数周期。次配置有两个好处：①时钟精度提高；②提高系统稳定性。

11、 上电延时等待问题：

Demo 里面均配置了 500ms 的上电延时操作，主要考虑到应用中存在大电容充放电的情况；如果 LVR 配置 1.8V，上电过程从 1.8->2.2V 这个区间初始化射频会异常，固上电这加入了一段延时；若实际使用场景下不存在 1.8->2.2v 区间电压上升慢的情况，就不加上电等待。

3.3 烧录滚码

- 1、使用“CMS Writer”软件烧录 PAN2416 芯片滚码时，先打开需要烧录的 .cmx 文件，单击“设置”后选择“自动增加模式”，设置好长度和地址等参数。需要注意的是，地址设置时必须选择 .cmx 文件中空白的位置，否则会导致程序运行异常。
- 2、滚码存储的地址必须和代码中的地址保持一致,否则滚码无法正确读取。



图 3-2 烧录滚码配置

3、 程序上读取滚码的程序：

/*各滚码的地址之间必须要相隔 2 个字节，以防覆盖，应先用烧录软件打开待烧录程序，然后找到未使用的空间，选取滚码的存放地址，读滚码的函数示例如下，非标准 C 语言格式，请勿改动*/

```
unsigned char get_roll0(void) @0x8d1 //(0x8d1 为滚码地址一)
```

```
{  
    return (0xff);  
}  
  
unsigned char get_roll1(void) @0x8e1 //滚码地址二  
{  
    return (0xff);  
}  
  
unsigned char get_roll2(void) @0x8f1 //滚码地址三  
{  
    return (0xff);  
}
```

/*如滚码读取已严格按流程操作仍失败,可能是 IDE 在电脑上注册不成功,电脑里面有装其他的 PIC 单片机编译器可能会出现提示成功但是未成功的情况,此类情况下建议换台电脑或系统再进行操作*/

4、烧录器的数码管显示

若增加烧录滚码选项,则烧录器的数码管显示是滚码“rXXX”,r 代表滚码标志,XXX 代表滚码低字节。

若没有增加烧录滚码选项,则烧录器的数码管显示是代码的 checksum。

3.4 离线烧录

如果需要使用离线烧录功能,必须先使用烧录器连接电脑在线烧录一遍程序,然后将烧录器重新上电,才能进入离线烧录模式;