MS Micro Controller

MS Designed By Origin-gd Tech

MS83Fxx02 应用文档

ADC 模块应用

8/14/16 引脚 8 位 FLASH 单片机

目录

ADC 模块应用

1. ADC 模块的特性：

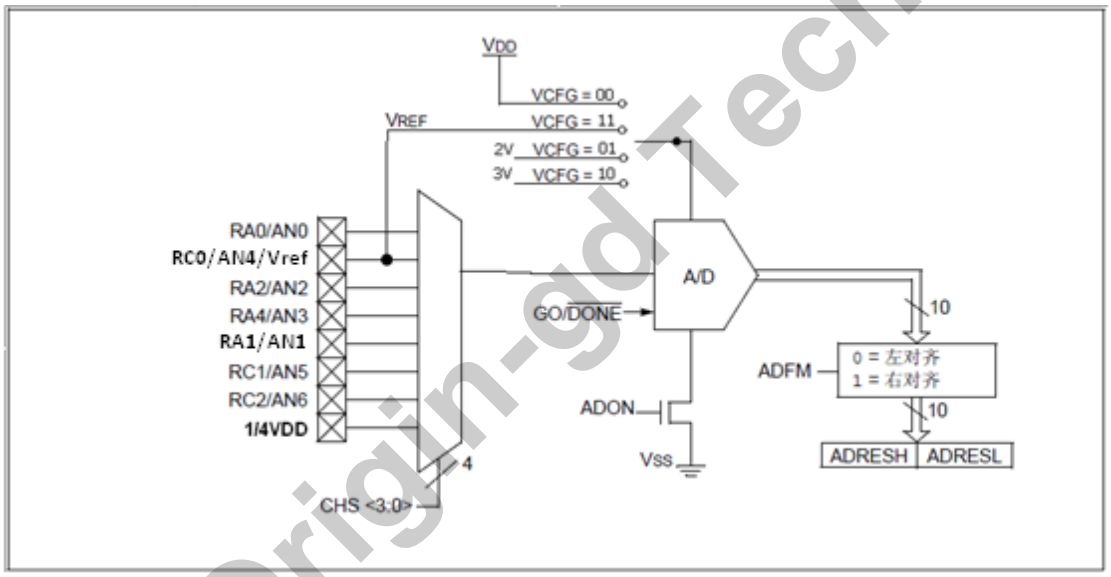
模数转换器(Analog-to-Digital Converter,ADC)可将模拟输入信号转换为相应的10位二进制表征值。该系列器件采用多个模拟输入复用到一个采样保持电路。采样保持电路的输出与转换器的输入相连接。转换器通过逐次逼近法产生10位二进制值，并将转换结果保存在ADC结果寄存器(ADSESH:ADSESL)中。

ADC参考电压可用软件选择为VDD、施加在外部参考引脚上的电压、内部固定参考电压2V或内部固定参考电压3V。

注：选择内部固定参考电压2V时，VDD电压必须大于2.5V。选择内部固定参考电压3V时，VDD电压必须大于3.5V。

ADC可在转换完成时产生中断。改中断可用于将器件从休眠中唤醒。

ADC框图：



2. ADC 的配置

配置和使用 ADC 时，必须考虑以下功能：

- 端口配置
- 通道选择
- ADC参考电压的选择
- ADC转换时钟源
- 中断控制
- 转换结果的格式

1).端口配置

ADC可用于把模拟型号转换为数字信号，转换模拟信号时，应将相关的TRIS和ANSEL位置1，将I/O引脚应配置为模拟输入功能。比如要将RA0引脚设置作为AD转换引脚，须将TRISA0置1和ANSEL0置1。

注：如果选择内部1/4VDD通道(通道7)，必须将ANSLE7置1。如果将模拟口定义为数字输入的引脚上存在模拟电压，可导致输入缓缓从其传导过大的电流。

2).通道选择

ADCON0寄存器的CHS位决定将哪个通道连接到采样保持电路。

改变通道时，开始下一次转换需要一个延时。建议延时20us至30us。

3).ADC参考电压

ADCON0寄存器的VCFG位提供对正参考电压的控制。正参考电压可以是VDD、可以是外部电源、内部固定参考电压2V或内部固定参考电压3V。负参考电压始终连接到参考地。

注：选择内部固定参考电压2V时，VDD电压必须大于2.5V。选择内部固定参考电压3V时，VDD电压必须大于3.5V。

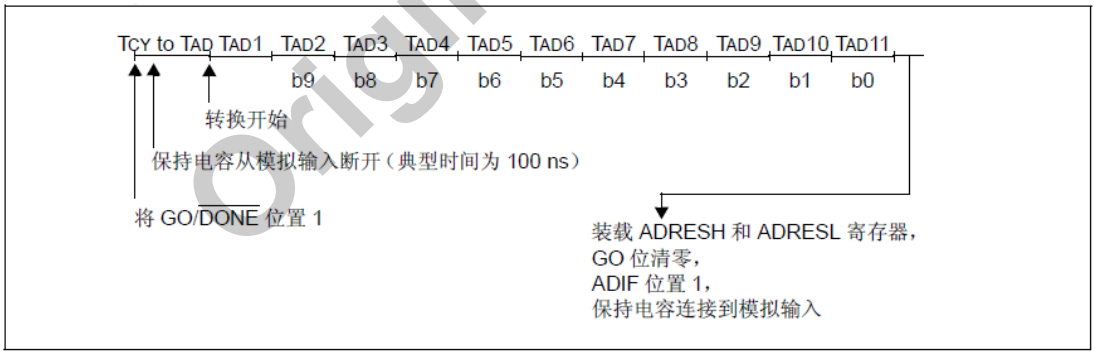
4).转换时钟

转换时钟源可通过ADCON1寄存器的ADCS位用软件选择。有以下13种选择：

- SYSCLK/2
- SYSCLK/4
- SYSCLK/8
- SYSCLK/16
- SYSCLK/32
- SYSCLK/64
- LFINTOSC/2
- LFINTOSC/4
- LFINTOSC/8
- LFINTOSC/16
- LFINTOSC/32
- LFINTOSC/64
- FRC(内部慢时钟振荡器)

完成一位(bit)的转换时间定义为T_{AD}。完成10位转换需要11.5个T_{AD}周期。

模数转换T_{AD}周期：



进行正确的转换必须满足相应的T_{AD}规范。

注：除非使用的是F_{RC}，否则任何系统时钟频率的变化均会改变ADC时钟频率，这将对ADC结果产生负面影响。F_{RC}可以是256KHz或者是32KHz，取决于LFMOD为何值。

ADC时钟周期(T_{AD})——器件工作频率，表：

ADC时钟周期(T_{AD})		器件频率(F_{OSC})			
ADC时钟源	ADCS[2:0]	16MHz	8MHz	4MHz	1MHz
F_{OSC}/2	000	125ns	250ns	500ns	2.0us
F_{OSC}/4	100	250ns	500ns	1.0us	4.0us
F_{OSC}/8	001	500ns	1.0us	2.0us	8.0us
F_{OSC}/16	101	1.0us	2.0us	4.0us	16.0us
F_{OSC}/32	010	2.0us	4.0us	8.0us	32.0us
F_{OSC}/64	110	4.0us	8.0us	16.0us	64.0us
F_{RC}	x11	2~6us	2~6us	2~6us	2~6us

注：非阴影区表示的值已经违反了最小T_{AD}时间要求。

5).中断

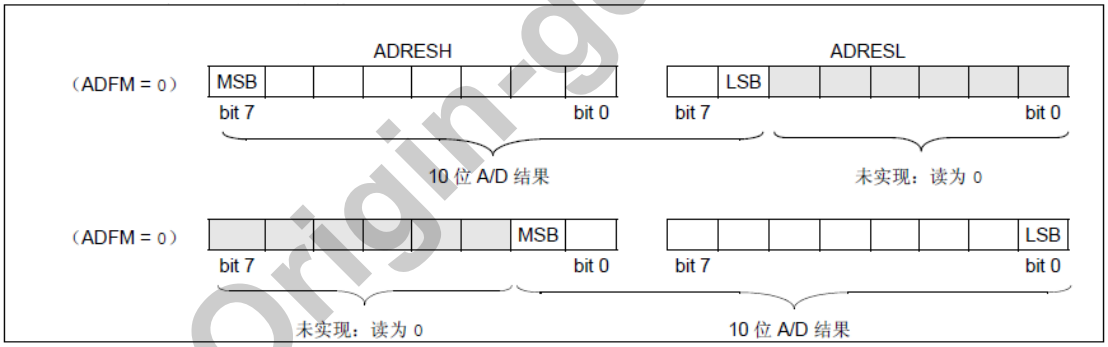
ADC模块可使中断在模数转换完成时产生。ADC中断标志为PIR1寄存器中的ADIF位。ADC中断使能为PIE1寄存器中的ADIE位。ADIF位必须用软件清零。

注：无论ADC中断是否被允许，ADIF位在每次转换完成时均置1。

器件工作或处于休眠状态时均可产生中断。如果器件处于休眠状态，中断将唤醒器件。从休眠唤醒时，始终执行SLEEP指令后的那条指令。如果用户试图唤醒器件并恢复顺序执行代码，必须禁止全局中断。如果允许全局中断，代码执行将先转至中断服务程序。

6).转换结果的格式

10位AD转换结果有两种格式，即左对齐和右对齐。ADCON0寄存器的ADFM位控制输出格式。两种输出结果格式图：



3. ADC 的工作原理

1).启动转换

要使能 ADC 模块，必须将 ADCON0 寄存器的 ADON 位置 1。将 ADCON0 寄存器的 GO/DONE 位置 1 将启动模数转换。

2).转换完成

转换完成时，ADC 模块将：

- 将GO/DONE位清零
- 将ADIF标志位置1
- 用新的转换结果更新ADRESH:ADRESL寄存器

3).终止转换

如果转换必须在完成前被终止，可用软件将 GO/DONE 清零。ADRESH:ADRESL 寄存器不会被未完成的模数转换采样值更新。相反，ADRESH:ADRESL 这对寄存器将保持先前转换的值。此外，启动另一次采集前，需等待 $2T_{AD}$ 的延时。延时后，所选通道的输入采集将自动启动。

注：器件复位将强制所有寄存器回到其复位状态。这样，ADC 模块就被关闭，并且任何待处理的转换均被终止。

4).休眠模式下 ADC 的工作

ADC 模块可在休眠期间工作。这要求将 ADC 时钟源置与 F_{RC} 选项。选定 F_{RC} 时钟源后，ADC 将再等待一条指令后才开始转换。这使 SLEEP 指令得以执行，从而降低转换期间的系统噪声。如果允许 ADC 中断，转换完成后器件将从休眠唤醒。

如果 ADC 时钟源不是，执行一条 SLEEP 指令将使当前转换中止。

注：如果有用到 SLEEP 模式但没有用到 ADC 模块，请在 SLEEP 前确保 ADCON0 寄存器的 ADON 位为 0，才能让 ADC 模块功耗降至最低。

4. 特殊事件触发器

ECCP 特殊事件触发器可在软件不干预的情况下周期地进行 ADC 测量。发生触发事件时，GO/DONE 位由硬件置 1，Timer1 计数器复位为零。

特殊时间触发器的使用并不确定正常 ADC 定时。确保满足 ADC 定时要求是用户的责任。

5. AD 转换步骤

以下是使用 ADC 进行模数转换的步骤示例：

a. 配置端口：
   - 禁止引脚输出驱动器(见TRIS寄存器)
   - 将引脚配置为模拟
b. 配置 ADC 模块：
   - 选择ADC转换时钟
   - 配置参考电压
   - 选择ADC输入通道
   - 选择转换结果的格式
   - 打开ADC模块
c. 配置 ADC 中断（可选）：
   - 将ADC中断标志清零
   - 允许ADC中断
   - 允许外设中断
   - 允许全局中断
d. 等待所需的采集时间
e. 将 GO/DONE 置 1 启动转换
f. 通过以下情况之一等待 ADC 转换完成：
   - 查询GO/DONE位
   - 等待ADC中断(允许中断时)
g. 读取 ADC 结果
h. 将 ADC 中断标志清零(允许中断时)

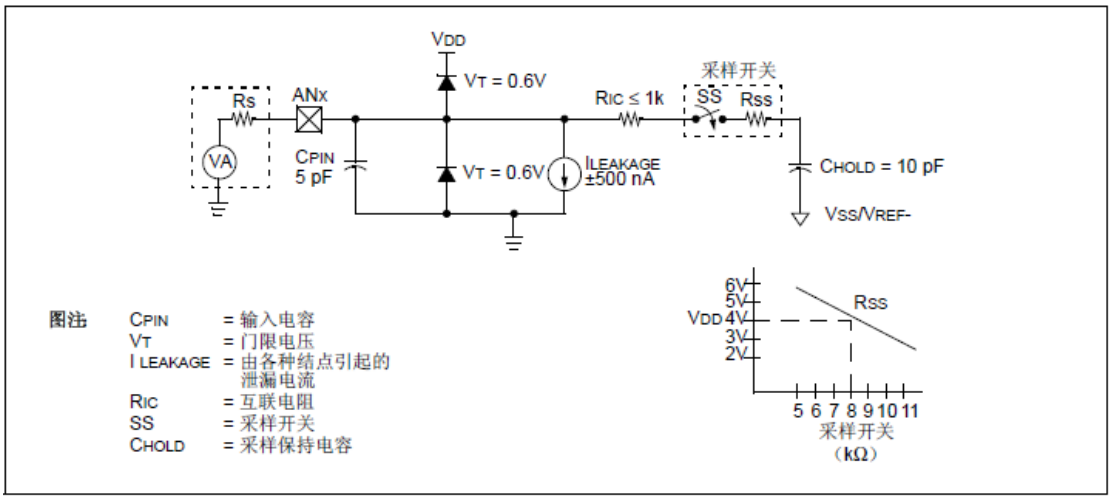
注：如果用户试图将器件从休眠中唤醒并顺序执行代码，则可禁止全局中断。

6. AD 采集要求

为了使 ADC 达到规定的精度，必须使充电保持电容(CHOLD)充满至输入通道的电平。源阻抗(R_S)和内部采样开关(R_{SS})阻抗直接影响 CHOLD 的充电时间。采样开关(R_{SS})阻抗随器件电压(VDD)的变化而变化。建议信号源的最大阻抗为 10kΩ 。采集时间随着源阻抗的降低而缩短。在选择(或

改变)模拟输入通道后，必须在开始转换前完成采集。因此开启通道后，必须要一个延时。

模拟输入模型：



7. 相关寄存器

1).ADRESH 寄存器，ADFM=0

Bit	7	6	5	4	3	2	1	0
Name	ADRES9	ADRES8	ADRES7	ADRES6	ADRES5	ADRES4	ADRES3	ADRES2
Reset	x	x	x	x	x	x	x	x

Bit7~Bit0：ADC 结果寄存器高 8 位

2).ADRESL 寄存器，ADFM=0

Bit	7	6	5	4	3	2	1	0
Name	ADRES1	ADRES0	-	-	-	-	-	-
Reset	x	x	-	-	-	-	-	-

Bit7~Bit6：ADC 结果寄存器低 2 位

Bit5~Bit0：保留，不使用

3).ADRESH 寄存器，ADFM=1

Bit	7	6	5	4	3	2	1	0
Name	-	-	-	-	-	-	ADRES9	ADRES8
Reset	-	-	-	-	-	-	x	x

Bit7~Bit2：保留，不使用

Bit1~Bit0：ADC 结果寄存器高 2 位

4).ADRESL 寄存器，ADFM=1

Bit	7	6	5	4	3	2	1	0
Name	ADRES7	ADRES6	ADRES5	ADRES4	ADRES3	ADRES2	ADRES1	ADRES0
Reset	x	x	x	x	x	x	x	x

Bit7~Bit0：ADC 结果寄存器低 8 位

5).ADCON0 寄存器

Bit	7	6	5	4	3	2	1	0
Name	ADFM	VCFG1	VCFG0	CHS2	CHS1	CHS0	GO/DONE	ADON
Reset	0	0	0	0	0	0	0	0

Bit7：AD转换结果格式选择位

1：右对齐

0：左对齐

Bit6~Bit5：参考电压选择位

11：V_{REF}引脚

10：内部固定3V参考电压

01：内部固定2V参考电压

00：VDD

注：选择内部固定参考电压2V时，VDD电压必须大于2.5V。选择内部固定参考电压3V时，VDD电压必须大于3.5V。RC0设置为模拟管脚才可以作为外部参考V_{REF}输入。

Bit4~Bit2：模拟通道选择位

000：AN0

001：AN1

010：AN2

011：AN3

100：AN4

101：AN5

110：AN6

111：内部1/4VDD

注：选择内部1/4VDD通道时，必须将ANSLE7置1。

Bit1：AD转换状态位位

1：AD转换正在进行。将本位置1启动一次AD转换。AD转换完成时此位由硬件自动清零

0：AD转换完成或不在进行中

Bit0：ADC使能位

1：使能ADC

0：禁止ADC，不消耗工作电流

注：低功耗环境下，如果没有用到ADC模块，请确保ADON位为0

6).ADCON1 寄存器

Bit	7	6	5	4	3	2	1	0
Name	DIVS	ADCS2	ADCS1	ADCS0	-	-	-	-
Reset	0	0	0	0	-	-	-	-

Bit7：AD分频时钟源选择位

1：分频时钟选择慢时钟

0：分频时钟选择F_{OSC}

Bit6~Bit4：AD转换时钟选择位

当DIVS为0时：

000：F_{OSC}/2

001：F_{OSC}/8

010：F_{OSC}/32

x11：F_{RC}(时钟来自内部振荡器，32KHz或者256KHz)

100：F_{OSC}/4

101：F_{OSC}/16

110：F_{OSC}/64

当DIVS为1时：

000：LFINTOSC/2

001：LFINTOSC/8

010：LFINTOSC/32

x11：F_{RC}(时钟来自内部振荡器，32KHz或者256KHz)

100：LFINTOSC/4

101：LFINTOSC/16

110：LFINTOSC/64

Bit3~Bit0：未实现，读为0

7).ANSEL 寄存器

Bit	7	6	5	4	3	2	1	0
Name	ANSEL7	ANSEL6	ANSEL5	ANSEL4	ANSEL3	ANSEL2	ANSEL1	ANSEL0
Reset	1	1	1	1	1	1	1	1

Bit7~Bit0：模拟选择位

在AN[7:0]引脚上分别进行模拟或者数字功能的选择

1：模拟输入 ，引脚被分配为模拟输入

0：数字IO，引脚被分配给端口或特殊功能

注：当 ADC 配置为采样 1/4VDD 通道时，ANSEL7 必须设置为 1。将某引脚设置为模拟输入将自动禁止数字输入电路、弱上拉以及电平变化中断。相应 TRIS 位必须设置为输入模式以允许改引脚的电压进行外部控制。

8).PORTA 寄存器

Bit	7	6	5	4	3	2	1	0
Name	RA7	RA6	RA5	RA4	RA3	RA2	RA1	RA0
Reset	x	x	x	x	x	x	x	x

Bit7~Bit0：RA[7:0] I/O引脚位

1： RAx引脚电平＞V_{IH}

0： RAx引脚电平＜V_{IL}

注：当 MCLRE 为 1 时，RA5 为外部复位引脚，否则 RA5 为通用的 I/O 口

9).PORTC 寄存器

Bit	7	6	5	4	3	2	1	0
Name	-	-	RC5	RC4	RC3	RC2	RC1	RC0
Reset	-	-	x	x	x	x	x	x

Bit7~Bit6：未实现，读为0

Bit5~Bit0：RC[5:0] I/O引脚位

1： RCx引脚电平＞V_{IH}

0： RCx引脚电平＜V_{IL}

10).TRISA 寄存器

Bit	7	6	5	4	3	2	1	0
Name	TRISA7	TRISA6	TRISA5	TRISA4	TRISA3	TRISA2	TRISA1	TRISA0
Reset	1	1	1	1	1	1	1	1

Bit7~Bit0：RA[7:0]引脚方向控制位

1： RAx引脚配置为输入(三态)

0： RAx引脚配置为输出

11).TRISC 寄存器

Bit	7	6	5	4	3	2	1	0
Name	-	-	TRISC5	TRISC4	TRISC3	TRISC2	TRISC1	TRISC0
Reset	-	-	1	1	1	1	1	1

Bit7~Bit6：未实现，读为0

Bit5~Bit0：RC[5:0]引脚方向控制位

1： RCx引脚配置为输入(三态)

0： RCx引脚配置为输出

12).INTCON 寄存器

Bit	7	6	5	4	3	2	1	0
Name	GIE	PEIE	T0IE	INTE	PAIE	T0IF	INTF	PAIF
Reset	0	0	0	0	0	0	0	0

Bit7：全局中断使能位

1：使能全局中断

0：关闭全局中断

Bit6：外设中断使能位

1：使能外设中断

0：关闭外设中断

Bit5：Timer0定时器溢出中断使能位

1：使能Timer0中断

0：关闭Timer0中断

Bit4：外部中断使能位

1：使能PA2/INT外部中断

0：关闭PA2/INT外部中断

Bit3：PORTA端口状态变化中断使能位

1：使能PORTA端口状态变化中断使能

0：关闭PORTA端口状态变化中断使能

Bit2：Timer0定时器溢出中断标志位

1：Timer0寄存器溢出（必须软件清零）

0：Timer0寄存器无溢出

Bit1：外部中断标志位

1：PA2/INT外部中断发生（必须软件清零）

0：PA2/INT外部中断无发生

Bit0： PORTA端口状态改变中断标志位

1：至少有一个 PORTA 口产生状态改变

0：没有一个 PORTA 口产生状态改变

13).PIR1 寄存器

Bit	7	6	5	4	3	2	1	0
Name	EEIF	CKMEAIF	-	C2IF	C1IF	OSFIF	TMR2IF	TMR1IF
Reset	0	0	-	0	0	0	0	0

Bit7：EEIF写中断标志位

1：EE写操作完成（必须软件清零）

0：EE写操作未完成

Bit6：快时钟测量慢时钟操作完成中断标志位

1：快时钟测量慢时钟操作完成（必须软件清零）

0：快时钟测量慢时钟未完成

Bit5：保留位

Bit4：比较器2中断标志位

1：比较器2输出发生了中断

0：比较器2输出未发生中断

Bit3：比较器1中断标志位

1：比较器1输出发生了中断

0：比较器1输出未发生中断

Bit2：振荡器故障中断标志位

1：系统振荡器发生故障，时钟输入切换为INTOSC

0：系统时钟运行正常

Bit1：Timer2与PR2比较相等中断标志位

1：timer2的值等于PR2（必须软件清零）

0：timer2的值不等于PR2

Bit0：Timer1 溢出中断标志位

1：timer1发生了溢出

0：timer1未发生溢出

14).PIE1 寄存器

Bit	7	6	5	4	3	2	1	0
Name	EEIE	CKMEAIE	-	C2IE	C1IE	OSFIE	TMR2IE	TMR1IE
Reset	0	0	-	0	0	0	0	0

Bit7：EEPROM写中断使能位

1：使能EE写操作完成中断

0：关闭EE写操作完成中断

Bit6：快时钟测量慢时钟操作完成中断使能位

1：使能快时钟测量慢时钟操作完成中断

0：关闭快时钟测量慢时钟操作完成中断

Bit5：保留位

Bit4：比较器2中断使能位

1：使能比较器2中断

0：关闭比较器2中断

Bit3：比较器1中断使能位

1：使能比较器1中断

0：关闭比较器1中断

Bit2：振荡器故障中断使能位

1：使能振荡器故障中断

0：关闭振荡器故障中断

Bit1：Timer2与PR2比较相等中断使能位

1：使能timer2的值等于PR2中断

0：关闭timer2的值等于PR2中断

Bit0：Timer1 溢出中断使能位

1：使能Timer1溢出中断

0：关闭Timer1溢出中断

8. 应用范例 1——AN0 的使用

```
/************************************************************
*文件名:MS83F_ADC_TEST_1.C
*功能: MS83Fxx02 的 ADC 通道 0 应用演示
*器件型号:MS83F1602
*振荡器:内部 RC 4MHz 4T
*引脚定义:
*                    --------------------------
*   VDD----------|1(VDD)    (GND)16|------------GND
*   NC-----------|2(RA7)     (PA0)15|------------AD0
*   NC-----------|3(PA6)     (PA1)14|------------LED
*   NC-----------|4(PA5)     (PA2)13|------------NC
*   NC-----------|5(PC3)     (PA3)12|------------NC
*   NC-----------|6(PC2)     (PC0)11|------------NC
*   NC-----------|7(PA4)     (PC1)10|------------NC
*   NC-----------|8(PC5)     (PC4)09|------------NC
*                    --------------------------
*                    MS83F1602 SOP16
*说明:每 5 毫秒检测 ADC 通道 0 一次.参考电压为 VDD
       判断 AD 转换到的值如果大于 512,RA1 输出高电平
       否则输出低电平。
       即如果 VDD=5V,
       通道 0 输入电压如果低于 2.5V,RA1 输出低电平
       通道 0 输入电压如果高于 2.5V,RA1 输出高电平
       注：这是一个简答的 ADC 例程,实际应用需要多次采样求平均等滤波
************************************************************/
#include  "syscfg.h";
#include "MS83Fxx02.h";

#define _XTAL_FREQ   4000000 //4T,此定义详解请看 MS_Q&A_Sheet.pdf 文档

#define LED   RA1

unsigned int Buff_ADC=0;
/*=====================================================
*函数名:DEVICE_INIT
*功能:上电器件初始化
*输入参数:无
*返回参数:无
=====================================================*/
void DEVICE_INIT(void)
{
     OSCCON = 0B01010001;     //Bit7 >>>    LFMOD=0 >>> WDT 振荡器频率=32KHz
```

```
                                //Bit6:4 >>> IRCF[2:0]=101 >>>  内部 RC 频率=4MHz
                                //Bit0   >>> SCS=1         >>>  系统时钟选择为内部振荡器
    MSCKCON = 0B00000000;   //Bit6   >>>   VREG_OE=0      >>>  禁止稳压输出
                                  //Bit5   >>> T2CKSRC=0   >>> Timer2 时钟源为系统时钟
                                 //Bit4    >>> SLVREN=0    >>>  关闭 LVR
    INTCON = 0B00000000;    //暂禁止所有中断
    CMCON0 = 0B00000111;   //关闭比较器
    PORTA = 0B00000000;
    TRISA = 0B11111101;     //RA1 设置为输出,其他 PORTA 口设置为输入
    WPUA = 0B11111100;      //开启 RA2~RA7 的内部弱上拉
    PORTC = 0B00000000;
    TRISC = 0B11111111;     //所有 RC 口设置为输入
    WPUC = 0B11111111;      //开启 RC0~RC5 的内部弱上拉
    OPTION = 0B00000000;   //bit7=0,开启 PORTA 内部上拉总闸
}

/*=====================================================
*函数名:ADC_INIT
*功能:比较器初始化
*输入参数:无
*返回参数:无
=====================================================*/
void ADC_INIT(void)
{
    ANSEL = 0B00000001;      //设置 RA0 为模拟输入口
    ADCON1 = 0B01100000;    //DIVS=0,时钟选 FOSC
                             //ADCS[2:0]=110,分频 FOSC/64
    ADCON0 = 0B10000000;     //B7,ADFM=1,结果右对齐
                             //B6:5,VCFG=00,参考电压 VDD
                             //B6:5,VCFG=01,参考电压内部 2V
                             //B6:5,VCFG=10,参考电压内部 3V
                             //B6:5,VCFG=11,参考电压 Vref
                             //B4:2,CHS=000,选择 AN0 通道
                             //B1,GO,AD 转换状态位
                             //B0,ADON=1,ADC 使能
}

/*=====================================================
*函数名:GET_ADC_VALUE
*功能:开启 AD 转换,参考电压为 VDD
*输入参数:ADC 通道数
*返回参数:采样到的 AD 值
=====================================================*/
unsigned int GET_ADC_VALUE(unsigned char ChannelNO)
```

```
{
    unsigned int TempADCBuffer=0;

    ADCON0 = (ChannelNO<<2);   //设置通道
    ADCON0 |= 0b10000001;      //开启 ADC 电路
    __delay_us(20);            //等待采集到电压
    GO_DONE = 1;               //开启转换
    while(GO_DONE==1) CLRWDT();//等待转换完成
    TempADCBuffer = ADRESH;
    TempADCBuffer = (TempADCBuffer<<8)|ADRESL;
    ADON = 0;
    return(TempADCBuffer);
}

/*=====================================================
*函数名:main
*功能:主函数
*输入参数:无
*返回参数:无
=====================================================*/
void main(void)
{
    DEVICE_INIT();        //器件初始化
    ADC_INIT();           //ADC 初始化
//  ENABLE_INTERRUPT();
    while(1)
    {
        CLRWDT();
        __delay_ms(5);
        Buff_ADC = GET_ADC_VALUE(0);
        if(Buff_ADC>=512)
        {
            LED = 1;
        }
        else
        {
            LED = 0;
        }
    }
}
```

9. 应用范例 2——内部 1/4VDD 的使用

```
/************************************************************
*文件名:MS83F_ADC_TEST_2.C
*功能:MS83Fxx02 的 ADC 内部 1/4VDD 通道应用演示
*器件型号:MS83F1602
*振荡器:内部 RC 4MHz 4T
*引脚定义:
*                    ------------------------
*   VDD----------|1(VDD)    (GND)16|------------GND
*   NC-----------|2(RA7)     (PA0)15|------------LED
*   NC-----------|3(PA6)     (PA1)14|------------NC
*   NC-----------|4(PA5)     (PA2)13|------------NC
*   NC-----------|5(PC3)     (PA3)12|------------NC
*   NC-----------|6(PC2)     (PC0)11|------------NC
*   NC-----------|7(PA4)     (PC1)10|------------NC
*   NC-----------|8(PC5)     (PC4)09|------------NC
*                    ------------------------
*                  MS83F1602 SOP16
*说明:利用内部 1/4VDD 通道和内部固定参考电压检测 MCU 的 VDD 脚电压。
       程序中,选择内部固定参考电压为 2V.
       采样值如果大于 512,RA0 输出高电平
       采样值如果小于 512,RA0 输出低电平
       可以换算到如果 VDD 小于 4V，RA0 输出低电平
       大于 4V，RA0 输出高电平
       即:((512/1023)*2V)*4=4V
       这个内部通道可以检测单节锂电的电压,且可以做到没有漏电流
       因为内部 1/4 分压电阻网络的漏电可以通过 ADON 置 0 来断开。
       注意:选择固定参考电压 2V，VDD 电压必须高于 2.5V
       选择固定参考电压 3V，VDD 电压必须高于 3.5V
************************************************************/
#include  "syscfg.h";
#include "MS83Fxx02.h";

#define _XTAL_FREQ    4000000 //4T,此定义详解请看 MS_Q&A_Sheet.pdf 文档

#define LED   RA0

unsigned int Buff_ADC=0;
/*=================================================
*函数名:DEVICE_INIT
*功能:上电器件初始化
*输入参数:无
*返回参数:无
```

```
==================================================*/
void DEVICE_INIT(void)
{
    OSCCON = 0B01010001;    //Bit7      >>>   LFMOD=0 >>> WDT 振荡器频率=32KHz
                            //Bit6:4 >>> IRCF[2:0]=101 >>>  内部 RC 频率=4MHz
                            //Bit0    >>> SCS=1         >>>  系统时钟选择为内部振荡器
    MSCKCON = 0B00000000;   //Bit6      >>>   VREG_OE=0      >>>  禁止稳压输出
                            //Bit5    >>> T2CKSRC=0   >>> Timer2 时钟源为系统时钟
                            //Bit4    >>> SLVREN=0    >>>  关闭 LVR
    INTCON = 0B00000000;    //暂禁止所有中断
    CMCON0 = 0B00000111;    //关闭比较器
    PORTA = 0B00000010;
    TRISA = 0B11111110;     //RA0 设置为输出,其他 PORTA 口设置为输入
    WPUA = 0B11111110;      //开启 RA1~RA7 的内部弱上拉
    PORTC = 0B00000000;
    TRISC = 0B11111111;     //所有 RC 口设置为输入
    WPUC = 0B11111111;      //开启 RC0~RC5 的内部弱上拉
    OPTION = 0B00000000;    //bit7=0,开启 PORTA 内部上拉总闸
}

/*==================================================
*函数名:ADC_INIT
*功能:比较器初始化
*输入参数:无
*返回参数:无
==================================================*/
void ADC_INIT(void)
{
    ANSEL = 0B10000000;     //设置内部 1/4VDD 通道位模拟输入口
    ADCON1 = 0B01100000;    //DIVS=0,时钟选 FOSC
                              //ADCS[2:0]=110,分频 FOSC/64
    ADCON0 = 0B10000000;    //B7,ADFM=1,结果右对齐
                              //B6:5,VCFG=00,参考电压 VDD
                              //B6:5,VCFG=01,参考电压内部 2V
                              //B6:5,VCFG=10,参考电压内部 3V
                              //B6:5,VCFG=11,参考电压 Vref
                              //B4:2,CHS=000,选择 AN0 通道
                              //B1,GO,AD 转换状态位
                              //B0,ADON=1,ADC 使能
}

/*==================================================
*函数名:GET_ADC_VALUE
*功能:开启 AD 转换,参考电压为内部固定参考电压 2V
```

```
*输入参数:ADC 通道数
*返回参数:采样到的 AD 值
==========================================================*/
unsigned int GET_ADC_VALUE(unsigned char ChannelNO)
{
    unsigned int TempADCBuffer=0;

    ADCON0 = (ChannelNO<<2);   //设置通道
    ADCON0 |= 0b10100001;      //开启 ADC 电路
    __delay_us(20);            //等待采集到电压
    GO_DONE = 1;               //开启转换
    while(GO_DONE==1) CLRWDT();//等待转换完成
    TempADCBuffer = ADRESH;
    TempADCBuffer = (TempADCBuffer<<8)|ADRESL;
    ADON = 0;
    return(TempADCBuffer);
}

/*==========================================================
*函数名:main
*功能:主函数
*输入参数:无
*返回参数:无
==========================================================*/
void main(void)
{
    DEVICE_INIT();        //器件初始化
    ADC_INIT();           //ADC 初始化
    while(1)
    {
        CLRWDT();
        __delay_ms(5);
        Buff_ADC = GET_ADC_VALUE(7);
        if(Buff_ADC>=512)
        {
            LED = 1;
        }
        else
        {
            LED = 0;
        }
    }
}
```

10. 联系我们

深圳市粤原点科技有限公司
SHENZHEN ORIGIN-GD TECH CO.,LTD
WEB:www.origin-gd.com
TEL: 0755-83666320
FAX: 0755-83666329
PHONE: 18344146830 13510476700 13902985185
FAE QQ: 2850507666
ADDRESS: 广东省深圳市龙岗区坂田街道环城南路 5 号坂田国际中心 E 栋 605 室