

FT61F02X

ADC Application note

目录

1. 10bit 模/数转换器(ANALOG TO DIGITAL CONVERTER, ADC).....	3
1.1. ADC 相关寄存器汇总.....	4
1.2. ADC 配置.....	5
1.2.1. ADC 触发.....	6
1.2.2. ADC 中止转换	7
1.2.3. 中断	7
1.3. ADC 采样时间.....	7
1.4. ADC 最短采样时间	8
1.5. ADC 转换步骤示例	8
2. 应用范例.....	10
联系信息	14

FT61F02x ADC 应用

1. 10bit 模/数转换器(ANALOG TO DIGITAL CONVERTER, ADC)

ADC 模块可将模拟输入信号转换成 10-bit 的数字信号。ADC 可在不同的时钟速度下运行，并且在高达 500 kHz 的时钟速度(即 43 kHz 的采样率，23 μ s/采样)下仍具有真正的 10-bit 精度。

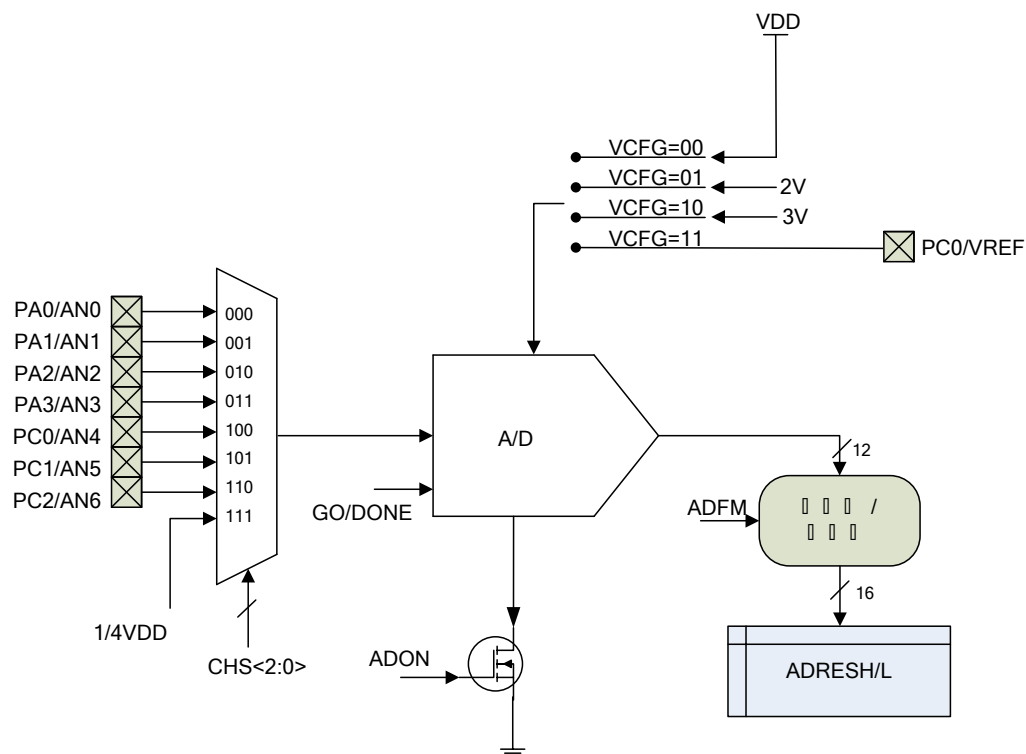


图 1-1 ADC 结构框图

模拟输入信号可以为 7 个 I/O(ANx)通道之一或 1 个内部通道(Internal 1/4VDD)。ADC 由指令或 ECCP 特殊事件触发器触发。在触发和 ADC 采样之间可增加延时。

当 ADC 转换完成时，将置位相应的中断标志位，并可触发中断和/或从睡眠中唤醒。

ADC 参考电压($V_{ADC-REF}$) 通过指令选择为 V_{DD} ，2 个内部参考电压(2V, 3V) 之一，或通过 I/O 输入外部参考电压。

ADC 不需要校准。另外，ADC 转换过程在后台运行，期间 CPU 可执行其他指令。

如果 ADC 需要在 SLEEP 下保持运行，且其转换时钟源为 Sysclk 或其分频时，则需通过另外使能所选时钟源为 Sysclk 的 Timers，来使系统时钟 Sysclk 在 SLEEP 时保持运行。当 ADC 的时钟源为 LIRC 时，进入 SLEEP 后 LIRC 将自动开启。

当 ADC 配置为硬件触发(ECCP 特殊事件触发器)时，GO/DONE 由硬件触发事件直接置位并启动 A/D 转换，软件置位 GO/DONE 将被忽略。ECCP 特殊事件触发器可在软件不干预的情况下周期性地对 ADC 测量。发生触发事件时，Timer1 计数器复位为零。特殊事件触发器的使用并不确保正常 ADC 定时，用户必须确保满足 ADC 定时要求。更多信息请参见[第 4 节“增强型捕捉/比较/PWM \(ECCP\)”](#)。

1.1. ADC 相关寄存器汇总

名称	状态	寄存器	地址	复位值
GIE	全局中断 1 = 使能 (PEIE, ADCIE 适用) 0 = 全局关闭 (唤醒不受影响)	INTCON[7]	0x0B 0x8B 0x10B	RW-0
PEIE	外设总中断 1 = 使能 (ADCIE 适用) 0 = 关闭 (无唤醒)	INTCON[6]		RW-0
ADIE	ADC 转换完成中断 1 = 使能 0 = 关闭 (无唤醒)	PIE2[1]	0x8D	RW-0
ADIF	ADC 转换完成中断标志位 1 = Yes (锁存) 0 = No	PIR2[1]	0x0D	RW-0

表 1-1 ADC 中断使能和状态位

名称	状态	寄存器	地址	复位值
ADRESL	<u>ADC 转换结果低有效位 (LSB)</u> ADFM=0: ADRESL[7:6] = 低 2 位 (其余为“0”) ADFM=1: ADRESL[7:0] = 低 8 位	ADRESL[7:0]	0x9E	RW-xxxx xxxx
ADRESH	<u>ADC 转换结果高有效位 (MSB)</u> ADFM=0: ADRESH[7:0] = 高 8 位 ADFM=1: ADRESH[1:0] = 高 2 位 (其余为“0”)	ADRESH[7:0]	0x1E	RW-xxxx xxxx
ADFM	<u>A/D 转换结果格式 (参阅 “ADRESH”)</u> 1 = 右对齐 0 = 左对齐	ADCON0[7]	0x1F	RW-0
VCFG	<u>V_{ADC-REF} (参考电压)</u> 00 = V _{DD} 10 = 内部 3V 01 = 内部 11 = 外部参考电压(I/O) 2V 注: PC0 作为外部参考输入 V _{REF} 必须设为模拟管脚	ADCON0[6:5]		RW-00
CHS	<u>ADC 模拟输入通道</u> 000 = AN0 100 = AN4 001 = AN1 101 = AN5 010 = AN2 110 = AN6 011 = AN3 111 = (内部 1/4 V _{DD})	ADCON0[4:2]		RW-000
GO/DONE	<u>ADC 转换启动和状态位</u> 1 = 由软件, ECCP1 启动 A/D 转换 (转换完成后自动清零) 0 = 转换完成 / 未进行转换	ADCON0[1]		RW-0
ADON	1 = ADC 使能 0 = ADC 关闭 (无电流消耗)	ADCON0[0]		RW-0

名称	状态	寄存器	地址	复位值
DIVS	<u>ADC 分频时钟源选择</u> 1 = LIRC 0 = SysClk	ADCON1[7]		RW-0
ADCS	<u>ADC 转换时钟源</u> DIVS = 0 DIVS = 1 000 = SysClk/2 000 = LIRC/2 001 = SysClk/8 001 = LIRC/8 010 = SysClk/32 010 = LIRC/32 100 = SysClk/4 100 = LIRC/4 101 = SysClk/16 101 = LIRC/16 110 = SysClk/64 110 = LIRC/64 x11 = LIRC x11 = LIRC 注: LIRC = 32kHz 或 256kHz, 取决于 LFMOD 的值	ADCON1[6:4]	0x9F	RW-000
ANSEL	<u>引脚模拟选择位</u> 1 = 模拟输入 0 = 数字 IO 注: 当 ADC 配置为采样内部 1/4 VDD 通道时, 需设置 ANSEL[7] = 1	ANSEL[7:0]	0x91	RW-1111 1111

表 1-2 ADC 相关用户寄存器

名称	地址	bit 7	bit 6	bit 5	bit 4	bit 3	Bit 2	bit 1	bit 0	复位值
ADRESL	0x9E	A/D 转换结果低有效位								xxxx xxxx
ADRESH	0x1E	A/D 转换结果高有效位								xxxx xxxx
ADCON0	0x1F	ADFM	VCFG<1:0>		CHS<2:0>			GO/DONE	ADON	0000 0000
ADCON1	0x9F	DIVS	ADCS<2:0>			—				0000 ... 000 ... 000

表 1-3 ADC 相关用户寄存器地址

1.2. ADC 配置

配置 ADC 包括以下设置 (更改配置时需设置 ADON=0 以关闭 A/D 转换或外部触发):

- 通道选择
- ADC 参考电压
- ADC 转换时钟源
- 转换结果格式
- 触发源
- 响应 (中断设置)

通道选择 – CHS 寄存器决定将哪个通道连接到用于 ADC 转换的采样保持电路。相应的 I/O 需通过设置 TRISx = 1 和 ANSELx = 1 来配置成模拟输入。

ADC 参考电压 ($V_{\text{ADC-REF}}$) – ADC 以参考电压作为相对值来测量输入模拟电压: V_{REF} 。负参考电压始终为参考地, 正参考电压可以选择为:

- V_{REF} 可选 VDD
- 内部参考电压 2V
- 内部参考电压 3V
- 外部参考电压 (V_{REF} 为 PC0)

ADC 转换时钟选择 – ADC 可通过指令选择 13 种时钟频率 (参阅 “ADCS”, 表 1-2):

- DIVS = 0 时为 SysClk/N 或 LIRC; DIVS = 1 时为 LIRC/N; N = 1, 2, 4, 8, 16, 32, 64
- LIRC (256 kHz 或 32 kHz, 参阅 “LFMOD”, 错误!未找到引用源。)

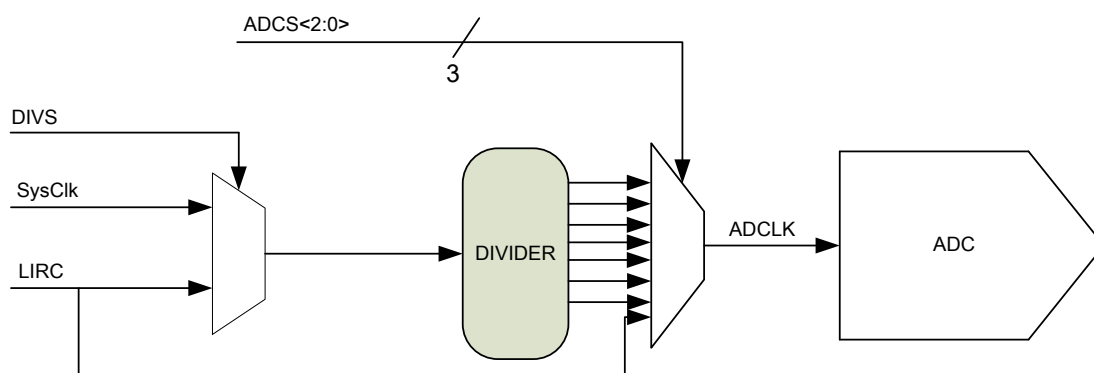


图 1-2 ADC 时钟配置

转换结果格式 – A/D 转换结果可储存为左对齐或右对齐两种格式(参阅 “ADFM”, 表 1-2)。

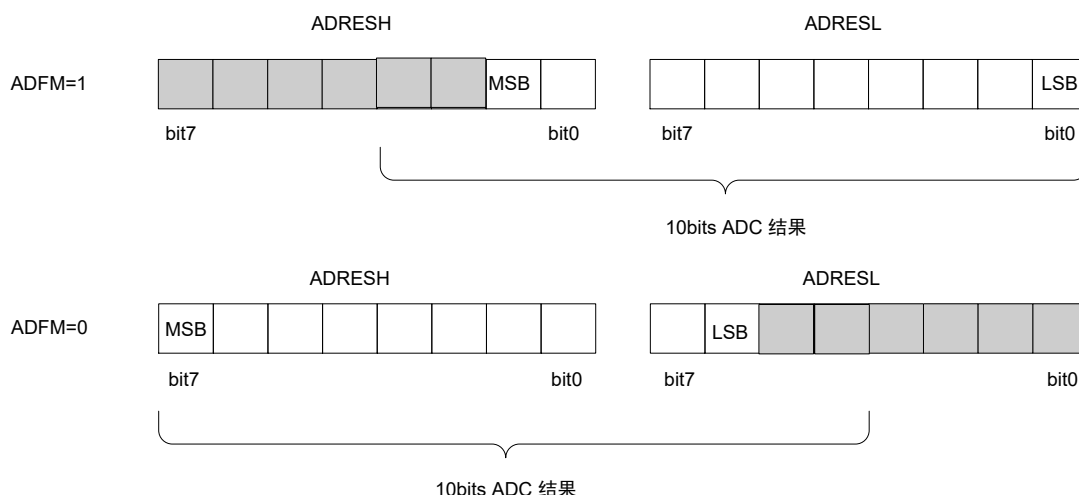


图 1-3 ADC 转换结果格式

1.2.1. ADC 触发

ADC 转换可由指令(GO/DONE= 1)、ECCP1 特殊事件触发器触发。GO/DONE 由指令置位后立即启动 A/D 转换。当 ECCP1 选择为特殊事件触发模式下, TMR1H/L 与 CCPR1H/L 寄存器发生匹配时, 则产生特殊事件触发输出, 启动一次 ADC 转换。

注: ADC 转换完成前将忽略新的触发条件。

1.2.2. ADC 中止转换

有时需中止 ADC 转换，比如需启动新的 ADC 采样时。

- 可通过软件设置 GO/DONE = 0 来中止 ADC。
- 当选择特殊事件触发时，必须通过关闭 ADC 模块(ADON = 0)来中止 ADC。
- 当 ADC 转换被中止时，ADRESH 和 ADRESL 不会被更新，而是保持前一次的转换结果值。
- 系统复位时，由于相应的寄存器被复位，因此 ADC 将中止，且 ADC 模块被关闭。

1.2.3. 中断

ADC 模块在发生下列事件时将置位相应的中断标志位：

- ADC 转换完成 (ADCIF)

中断模块有其相应的中断使能位(ADCIE)，和更高层级的外设总中断(PEIE)，以及最高级别的全局中断(GIE)。

无论中断使能位是否打开，发生中断事件时都将置位相应的中断标志位。是否触发中断和/或从睡眠中唤醒则取决于相应的使能控制位(GIE, PEIE, ADCIE)。

1.3. ADC 采样时间

采样时间，即采样保持时间，必须足够长以保证内部 ADC 电压稳定在输入通道电压的 0.01%误差以内，以达到 10bit 的精度(0.097%)。采样时间和外部串联电阻的关系如下(表 1-4)：

$$T_{ACQ} > 0.16 \times R (\mu s); R \text{ 的单位为 } k\Omega.$$

当采样时间 T_{ACQ} 为 $2\mu s$ 时，外部串联电阻必须 $\leq 12.5k\Omega$ 。如果使用更大的串联电阻，则 T_{ACQ} 将成比例增加。结点漏电流限制了允许使用的最大串联电阻值。对于 5nA 的结点漏电流，在 50 kΩ 的串联电阻上将产生 0.25mV (2V 参考电压的 0.0125%)的压降。而当温度超过 100°C 时，结点漏电流将大幅提高。因此，串联电阻越小越好。

串联电阻值	T_{ACQ}
> 50 kΩ	(不推荐)
50 kΩ	$\geq 8.0 \mu s$
25 kΩ	$\geq 4.0 \mu s$
12 kΩ	$\geq 2.0 \mu s$
< 12 kΩ	$\geq 2.0 \mu s$

表 1-4 不同的外部串联电阻与最短 T_{ACQ} 的对应关系

采样保持时间即为内部 ADC 观测输入通道电压的时间。

采样保持时间的开始 = 通道切换后或 ADC 稳定后，以时间较迟者为准。

采样保持时间的结束 = 延迟结束后的 0 到 1 个 T_{AD} 时间内。

1.4. ADC 最短采样时间

T_{AD} 为 ADC 的时钟周期。转换 10 位(bit) 需 $11.5 T_{AD}$ ，同步需 $2 - 3 \times T_{AD}$ ，完整的 10-bit 转换所需最短时间：

$$T_{ACQ} + (2 + 11.5) \times T_{AD} = T_{ACQ} + 13.5 T_{AD}$$

可保证真正 10-bit 精度的最高转换采样率约为 34 kHz (或~29 μ s/采样)。

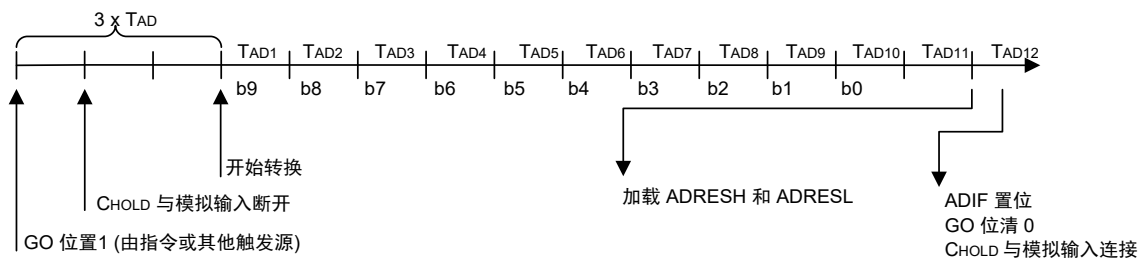


图 1-4 模数转换 T_{AD} 周期

1.5. ADC 转换步骤示例

设置 ADC：

1. 配置端口：
 - a. 设置 $TRISx = 1$ ，禁止引脚输出驱动
 - b. 设置 $ANSELx = 1$ ，关闭数字输入、弱上拉和弱下拉功能
2. 配置 ADC 模块：
 - a. 选择 ADC 转换时钟源
 - b. 选择 ADC 参考电压
 - c. 选择 ADC 触发条件：GO/DONE 或 CCP1 特殊事件触发
 - d. 选择转换结果格式
3. 配置 ADC 中断(可选)：
 - a. 使能 ADC 中断
 - b. 使能外设总中断
 - c. 关闭全局中断(如需执行中断服务程序则使能)
4. 打开 ADC 模块。然后等待所需 ADC 稳定时间 T_{ST} (~15 μ s)，当 $V_{ADC-REF}$ 选择内部参考电压时，则需等待内部参考电压的稳定时间 T_{VRINT} (参阅“ T_{VRINT} ”，[章节 错误!未找到引用源。](#)) 和 T_{ST} 时间的较长者，即 $\max(T_{VRINT}, T_{ST})$ 。

至此，ADC 已准备好对不同的通道进行采样。对输入通道采样时：

1. ADC 输入选择为需测量的通道 (参阅“CHS”)。
2. 如有必要，需清除 ADC 转换完成中断标志位。

3. 对采样时间 T_{ACQ} 有最低要求, T_{ACQ} 需足够长以保证内部 ADC 输入电容充满至输入通道电压的 0.01% 误差以内。另外取决于触发类型, 切换通道后或 ADC 稳定后(以时间较迟者为准)可能会有一定的延迟再触发。
 - a. 对于软件触发, 需要额外的 T_{ACQ} 时间。
5. 等待所需的延迟后, 由指令置位 GO/DONE, 或等待 CCP1 特殊事件触发自动置位 GO/DONE, 以启动 A/D 转换。
6. 通过以下方式等待 ADC 转换完成:
 - a. 查询 GO/DONE 位
 - b. 等待 ADC 中断 (使能中断时)
7. 读取 ADC 转换结果
8. 如有必要, 清除 ADC 转换完成中断标志位。

注:

1. 虽然 GO/DONE 和 ADON 在同一个寄存器(ADCON0)中, 但不应同时设置。
2. ADC 转换过程中或等待特殊事件触发时, 不可更改配置。建议在 ADON = 0 时进行更改。

以下为 ADC 程序示例 (输入采样通道为 PA0, ADC 时钟为 LIRC):

```

BANKSEL ADCON1
LDWI B'00110000'           ; ADC LIRC clock
STR ADCON1

BANKSEL TRISA
BSR TRISA, 0               ; Set PA0 to input
BANKSEL ANSEL
BSR ANSEL, 0               ; Set-PA0 to analog

BANKSEL ADCON0
LDWI B'10000001'           ; Right justify,
STR ADCON0                 ;  $V_{DD}$ , Vref, AN0, On

LCALL StableTime           ; ADC stable time
LCALL SampleTime           ; Acquisition delay,  $T_{ACQ}$ 

BSR ADCON0, GO              ; Start conversion
BTSC ADCON0, GO             ; Conversion done?
LJUMP $-1                  ; No, test again

BANKSEL ADRESH;
LDR ADRESH, W               ; Read upper 2 bits
STR RESULTHI               ; Store in GPR space
BANKSEL ADRESL;
LDR ADRESL, W               ; Read lower 8 bits
STR RESULTLO               ; Store in GPR space
  
```

2. 应用范例

```

/* 文件名: TEST_61F02x_ADC.C
* 功能:    FT61F02x-ADC 功能演示
* IC:      FT61F023 SOP16
* 晶振:    16M/2T
* 说明:    程序根据 AN2(PA2)口的电压来调节 PWM3P(PC4)的占空比
*          AN2 脚电压越高(0-VDD)PWM3(PC2)占空比越大(2K,0-99%)
*
*          FT61F023  SOP16
*          -----
* VDD-----|1(VDD) (VSS)16|-----GND
* NC-----|2(PA7)  (PA0)15|-----NC
* NC-----|3(PA6)  (PA1)14|-----NC
* NC-----|4(PA5)  (PA2)13|-----AN2
* NC-----|5(PC3)  (PA3)12|-----NC
* NC-----|6(PC2)  (PC0)11|-----NC
* NC-----|7(PA4)  (PC1)10|-----NC
* NC-----|8(PC5)  (PC4)09|----PWM3
*          -----
*/
#include "SYSCFG.h"
//*****宏定义*****
#define uchar      unsigned char
#define uint       unsigned int

//PWM 引脚输入输出控制
#define PWM3Dir   TRISC4

volatile uint     TestADC;
/*-----
* 函数名: POWER_INITIAL
* 功能:   上电系统初始化
* 输入:   无
* 输出:   无
*-----*/
void POWER_INITIAL (void)
{
    OSCCON = 0B01110001;          //IRCF=111=16MHz/2T=8MHz,0.125µs
    INTCON = 0;                  //暂禁止所有中断

    PORTA = 0B00000000;
    TRISA = 0B00000100;          //PA 输入输出 1-输入 0-输出
                                //PA2-输入

    PORTC = 0B00000000;

```

```

TRISC = 0B00000000;          //PC 输入输出 1-输入 0-输出
WPUA = 0;                     //禁止所有 PA 口上拉
WPUC = 0;                     //禁止所有 PC 口上拉

OPTION = 0B00001000;          //Bit3=1,WDT MODE,PS=000=WDT RATE 1:1
MSCKCON = 0B00000000;
//Bit6: 0-禁止 PA4, PC5 稳压输出
//Bit5: 0-TIMER2 时钟为 Fosc
//Bit4: 0-禁止 LVR
CMCON0 = 0B00000111;          //关闭比较器, CxIN 为数字 IO 口
}
/*-----
* 函数名: DelayUs
* 功能: 短延时函数 --16M-2T--大概快 1%左右.
* 输入: Time 延时时间长度 延时时长 Time μs
* 输出: 无
-----*/
void DelayUs(unsigned char Time)
{
    unsigned char a;
    for(a=0;a<Time;a++)
    {
        NOP();
    }
}
/*-----
* 函数名: ADC_INITIAL
* 功能: ADC 初始化
* 输入: 无
* 输出: 无
-----*/
void ADC_INITIAL (void)
{
    ADCON1 = 0B01100000;
    //Bit7: DIVS=0,时钟选 FOSC
    //Bit[6:4]: ADCS[2:0]=110,分频 FOSC/64

    ADCON0 = 0B10001001;
    //Bit7: ADFM=1,结果右对齐
    //Bit[6:5]: VCFG 参考电压
    // 00-参考电压 VDD
    // 01-参考电压内部 2V
    // 10-参考电压内部 3V
    // 11-参考电压 Vref //Bit[4:2]:CHS=010-选择 AN2 通道

```

```

//Bit1:  GO,AD 转换状态位
//Bit0:  ADON=1,ADC 使能
ANSEL = 0B00000100;          //使能 AN2 为模拟输入
}
/*-----
* 函数名:  GET_ADC_DATA
* 功能:   读取通道 ADC 值
* 输入:   AN_CN 通道序号
* 输出:   INT 类型 AD 值(单次采样无滤波)
*-----*/
uint GET_ADC_DATA (uchar AN_CH)
{
    uchar  i;
    uchar  ADCON0Buff;
    uint    tBuffer = 0;
    uint    ADC_DATA=0;

    ADCON0Buff = ADCON0 & 0B11100011;//清空通道值

    AN_CH <=<=2;
    ADCON0Buff |=  AN_CH;          //Bit[4:2]:CHS=010,选择 AN2 通道

    ADCON0 = ADCON0Buff;          //重新加载通道值
    DelayUs(20);
    GO_DONE = 1;                  //启动 ADC
    while( GO_DONE==1 );          //等待 ADC 转换完成

    ADC_DATA = ADRESH;
    ADC_DATA <=<= 8;
    ADC_DATA |= ADRESL;           //10Bit ADC 值整合
    tBuffer = ADC_DATA;
    return  tBuffer;
}
/*-----
* 函数名:  PWM_INITIAL
* 功能:   PWM 初始化
* 设置周期 = 2^TMRxPS*2^NBit*TPxCK
*           = 2^0*2^8*[(T3CKDIV+1)/16000000]
*           = 1*256*[(30+1)/16000000] = 0.496ms
* 设置脉宽 = 2TMRxPS*(PRx)*TPxCK
*           = 2^0*128*[(T3CKDIV+1)/16000000]
*           = 1*128*[(30+1)/16000000] = 0.248ms
*-----*/
void PWM_INITIAL (void)

```

```
{
    PWM3Dir = 1;                      //PWM3 输出 PIN 暂为输入模式

    PWM3CR0 = 0B00110010;
    //Bit7:  中断选择位 0-计数器溢出时产生中断
    //Bit[6:4]:周期位选择 011-8 位
    //Bit[3:1]:时钟选择 001-内部 RC 快时钟/(T3CKDIV+1)
    //Bit0:   PWM/BUZZER 选择 0-PWM 输出

    PWM3CR1 = 0B10000000;
    //Bit7:   1-TMR3 为 PWM/BUZZER 模式
    //Bit6:   0-PWM3 为高电平有效
    //Bit[5:3]: 000-PWM3 预分频比设置为 1:1
    //Bit2:   0-暂关 TMR3
    //Bit1:   0-禁止 TMR3 中断
    //Bit0:   0-TMR3 中断标志位只读
    TMR3H = 0;
    T3CKDIV = 30;
    PR3L = 128;                      //占空比 50%
}
/*-----
* 函数名: main
* 功能:   主函数
* 输入:   无
* 输出:   无
-----*/
void main()
{
    POWER_INITIAL();                //系统初始化
    ADC_INITIAL();                  //ADC 初始化
    PWM_INITIAL();                  //PWM 初始化
    TMR3ON = 1;                     //T3 开启 PWM3
    PWM3Dir = 0;                    //PWM3 PIN 设为输出模式允许 PWM 输出
    while(1)
    {
        TestADC = GET_ADC_DATA(2); //通道 2AD 值
        PR3L = TestADC >> 2; //将 AD 值赋值给 PWM 的 PR3L 调节 PWM3P 输出的占空比
        NOP();
        NOP();
    }
}
```

联系信息

Fremont Micro Devices Corporation

#5-8, 10/F, Changhong Building
Ke-Ji Nan 12 Road, Nanshan District,
Shenzhen, Guangdong, PRC 518057

Tel: (+86 755) 8611 7811

Fax: (+86 755) 8611 7810

Fremont Micro Devices (HK) Limited

#16, 16/F, Block B, Veristrong Industrial Centre,
34-36 Au Pui Wan Street, Fotan, Shatin, Hong Kong SAR

Tel: (+852) 2781 1186

Fax: (+852) 2781 1144

<http://www.fremontmicro.com>

* Information furnished is believed to be accurate and reliable. However, Fremont Micro Devices Corporation assumes no responsibility for the consequences of use of such information or for any infringement of patents of other rights of third parties, which may result from its use. No license is granted by implication or otherwise under any patent rights of Fremont Micro Devices Corporation. Specifications mentioned in this publication are subject to change without notice. This publication supersedes and replaces all information previously supplied. Fremont Micro Devices Corporation products are not authorized for use as critical components in life support devices or systems without express written approval of Fremont Micro Devices Corporation. The FMD logo is a registered trademark of Fremont Micro Devices Corporation. All other names are the property of their respective owners.