



复旦微电子

FM33G0xx

低功耗系列 MCU

应用笔记

ADC 应用说明

AN0007

V1.2



本资料是为了让用户根据用途选择合适的上海复旦微电子集团股份有限公司（以下简称复旦微电子）的产品而提供的参考资料，不转让属于复旦微电子或者第三者所有的知识产权以及其他权利的许可。

在使用本资料所记载的信息最终做出有关信息和产品是否适用的判断前，请您务必将所有信息作为一个整体系统来进行评价。

采购方对于选择与使用本文描述的复旦微电子的产品和服务全权负责，复旦微电子不承担采购方选择与使用本文描述的产品和服务的责任。除非以书面形式明确地认可，复旦微电子的产品不推荐、不授权、不担保用于包括军事、航空、航天、救生及生命维持系统在内的，由于失效或故障可能导致人身伤亡、严重的财产或环境损失的产品或系统中。

未经复旦微电子的许可，不得翻印或者复制全部或部分本资料的内容。

今后日常的产品更新会在适当的时候发布，恕不另行通知。在购买本资料所记载的产品时，请预先向复旦微电子在当地的销售办事处确认最新信息，并请您通过各种方式关注复旦微电子公布的信息，包括复旦微电子的网站(<http://www.fmsh.com/>)。

如果您需要了解有关本资料所记载的信息或产品的详情，请与上海复旦微电子集团股份有限公司在当地的销售办事处联系。

商 标

上海复旦微电子集团股份有限公司的公司名称、徽标以及“复旦”徽标均为上海复旦微电子集团股份有限公司及其分公司在中国的商标或注册商标。

上海复旦微电子集团股份有限公司在中国发布，版权所有。

联系方式：

电表产品应用：

邢杰： xingjie@fmsh.com.cn TEL: 13916427310

陈钊： chenzhao@fmsh.com.cn TEL: 18616125501

水气热表及智能家居：

朱发旺： zhufawang@fmsh.com.cn TEL: 17749796664

姜涛： jiangtao@fmsh.com.cn TEL: 18701992908

超高频900M及物联网相关：

王晓腾： wangxiaoteng@fmsh.com.cn TEL: 13585663727

王天纵： wangtianzong@fmsh.com.cn TEL: 18221803903

资料下载及交流：

开发者论坛： <http://www.fmdevelopers.com.cn>



目 录

1	说明	1
2	原理	1
2.1	说明	1
2.2	时钟	1
2.3	电压测量	2
2.3	温度测量	4
3	参考示例	4
3.1	ADC 配置说明	4
3.2	ADC 参考示例	4
3.3	ADC 如何在低功耗模式下使用	7
4	示例验证	8
5	注意事项	10
	附录	11
	版本信息	15
	上海复旦微电子集团股份有限公司销售及 服务网点	16



图片目录

图 2-1	ADC 时钟输入	1
图 2-2	ADC 斜率和截距参数	3
图 2-3	ADC 电压计算	3
图 3-1	ADC 实现步骤	5
图 3-2	ADC 查询示例	5
图 3-3	ADC 查询示例程序	6
图 4-1	示例 demo 板	8
图 4-2	示例仿真	9

1 说明

本文档为为 FM33G0 系列低功耗 MCU 的应用笔记，用于 ADC 使用的说明和使用方法。FM33G0 系列是复旦微电子公司开发的低功耗 MCU 芯片，请联系复旦微电子公司提供更多相关文档支持设计开发。

2 原理

2.1 说明

FM33G0 带有一个 1 阶 Σ - Δ ADC，分辨率 11bits。支持 8 个外部输入通道和数个内部通道，可用于转换外部电压和片上温度传感器等信号。其特点为：

- 工作电压为 2.2~5.5V
- 低功耗
- 单次转换时间典型值为 2ms（工作时钟 1MHz），降低精度可达到最低 512us

2.2 时钟

ADC 工作时钟由 RCHF 分频得到，如图 2-1 所示。

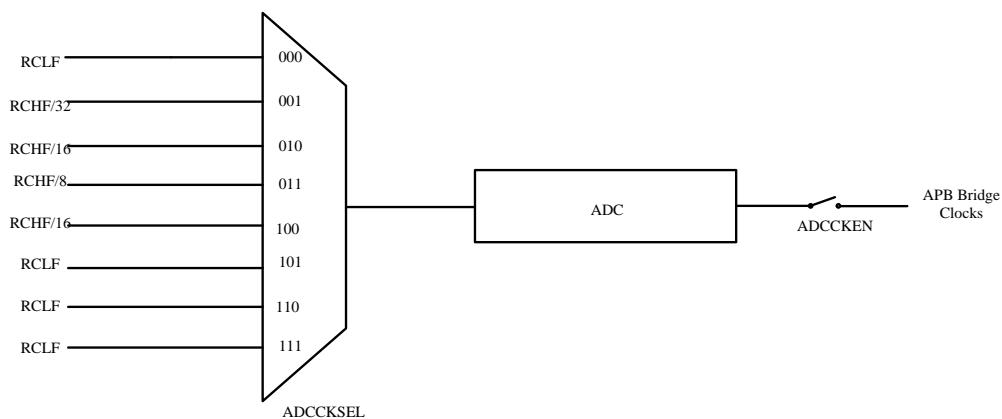


图 2-1 ADC 时钟输入

ADC 的工作时钟推荐小于 1M，超过 1M 采样精度将受到影响，时钟寄存器如附录表 1 所示。

2.3 电压测量

理想情况下，ADC 的电压输入和数字量输出成线性关系，根据输出的数字量可以直接计算出输入电压大小。假设输出数字量为 $DATA_{adc}$ ，输入电压值为 V_{adc} ，则有：

$$DATA_{adc} = k * V_{adc} + DATA_{offset}$$

其中 k 为输入输出曲线斜率，理想情况下，该值可以为任意两点间的斜率， $DATA_{offset}$ 为当输入电压为 0 时的输出数字量。当 k 和 $DATA_{offset}$ 确定后，可以根据输出的数字量计算出输入电压大小：

$$V_{adc} = \frac{DATA_{adc}}{k} - \frac{DATA_{offset}}{k} = slope * DATA_{adc} + V_{offset}$$

其中

$$slope = \frac{1}{k}$$

$$V_{offset} = -\frac{DATA_{offset}}{k}$$

对于 FM33G0 来说， $slope$ 和 V_{offset} 的值由参考电压决定。ADC 的 ADC_TRIM 寄存器，用来调整 ADC 转换时间，ADC_TRIM 值越低，转换之间越快，相应的电压分辨率降低。典型值 ADC_TRIM 为 0x3FF，11bit 分辨率。ADC_TRIM 为 0x1FF，10bit 分辨率。ADC_TRIM 为 0x0FF，9bit 分辨率。

ADC_TRIM 为 0x3FF 测量电压范围 0~电源电压、ADC 时钟 1MHz、转换时间 2ms

ADC_TRIM 为 0x1FF 测量电压范围 0~电源电压、ADC 时钟 1MHz、转换时间 1ms

ADC_TRIM 为 0x0FF 测量电压范围 0~电源电压、ADC 时钟 1MHz、转换时间 512us

ADC_TRIM 程序中的选择

```
void ADC_Init_TsetExVolt(void)
{
    //使用简单函数配置
    RCC_PERCLK_SetableEx(ANACCLK, ENABLE); //模拟电路总线时钟使能
    RCC_PERCLK_SetableEx(ADCCLK, ENABLE); //ADC时钟使能
    RCC_PERCLKCON2_ADCKSEL_Set(RCC_PERCLKCON2_ADCKSEL_RCHFIDIV16); //ADC工作时钟配置，不可高于1M
    ANAC_ADC_Channel_SetEx(CH_IN5); //ADC输入通道选择

    ANAC_ADCTRIM_Write(const_adc_TrimV_3FF); //adc频率1M 时 计算时间2ms
    // ANAC_ADCTRIM_Write(const_adc_TrimV_1FF); //adc频率1M 时 计算时间1ms
    // ANAC_ADCTRIM_Write(const_adc_TrimV_0FF); //adc频率1M 时 计算时间0.5ms

    ANAC_ADCCON_ADC_IE_Setable(DISABLE); //中断禁止
    ANAC_ADCCON_ADC_EN_Setable(DISABLE); //ADC关闭
}
```

需要注意的是等效参考电压可超过电源电压，但是输入电压不可超过电源电压，等效参考电压超过电源电压时，输入电压范围：0~电源电压。

芯片出厂时已经把 ADC_TRIM = 0x3FF、0x1FF、0xFF 时的 ADC 斜率和截距写到芯片参数区，如图 2-2 所示，ADC 斜率为 *slope* 扩大 1000 倍的数据，截距为 V_{offset} 扩大 100 倍的数据。

#define const_adc Slope_3FF	3FF	((uint16_t *) (0x1FFFC54)))	//ADC斜率, 除1000后使用, 单位约2.2mV/1sb
#define const_adc Offset_3FF		((int16_t *) (0x1FFFC56)))	//ADC截距, 除100后使用, 单位mV
#define const_adc Slope_1FF	1FF	((uint16_t *) (0x1FFFD50)))	//ADC斜率, 除1000后使用, 单位约4.4mV/1sb
#define const_adc Offset_1FF		((int16_t *) (0x1FFFD52)))	//ADC截距, 除100后使用, 单位mV
#define const_adc Slope_0FF	0FF	((uint16_t *) (0x1FFFD58)))	//ADC斜率, 除1000后使用, 单位约8.8mV/1sb
#define const_adc Offset_0FF		((int16_t *) (0x1FFFD5A)))	//ADC截距, 除100后使用, 单位mV

图 2-2 ADC 斜率和截距参数

为了对 ADC 输入输出测试曲线更好的进行拟合，可以用如图 2-3 程序进行计算输入电压：

```

/*ADC 电压计算函数
  输入：AD值、电源电压
  输出：电压@mV
*/
//被测电压低于100mV后测试结果不太准确
//被测电压超过4.4V后测试结果不太准确
//被测电压不可超过电源电压
//被测电压不可超过4.5V
/*****
ADC电压计算函数
功能：计算ADC测量电压值
输入：ADC值、电源电压
输出：电压值
*****/
uint32_t ANAC_ADC_VoltageCalc(uint32_t fADCDData,uint8_t Vdd)
{
    uint32_t fVolt = 0;
    uint32_t AdcTrim = 0;
    AdcTrim = ANAC_ADCTRIM_Read();

    if((AdcTrim == 0x0ff)&&(const_adc_Slope_0FF > 8000) && (const_
    {
        //待生成测试数据
    }

```

ADC采样值 (指向 fADCDData)

Vdd 填 3 或 5, 代表 3.xV 系统, 5V 系统 (指向 Vdd)

图 2-3 ADC 电压计算

程序中 fADCDData 为 ADC 采样值，Vdd 为系统电压，3.xV 系统填 3，5V 系统填 5。电压值通过返回值返回。需要注意的是，被测电压低于 100mV 及高于 4.4V 测试结果都不太准确，因此被测电压需在 100mV~4.4V 之间。

2.3 温度测量

FM33G0 除了可以进行外部电压测量，其内部还集成了温度传感器，可以进行温度的测量，测量原理为温度传感器根据温度的变化在内部输出不同的模拟量，ADC 对模拟量进行采集转化为数字量。在使用温度测量时，调校值 $ADC_TRIM = 0x640$ 。

芯片出厂时已经对温度传感器进行了校准，校准数据存储在参数区：

校准温度：30℃

校准标志地址：0x1FFFC90 (unsigned int) ，固定为 0x1E00

校准数据地址：0x1FFFC92 (unsigned int) ，30℃时的温度 ADC

温度斜率：5.02℃/LSB。

温度的计算公式为：

$$Temperature = (DATA_{adc} - DATA_{30}) / 5.02 + 30.0$$

式中 $Temperature$ (℃) 为当前温度值， $DATA_{adc}$ 为当前温度 AD 值， $DATA_{30}$ 为 30℃ 时的温度 AD 值，该数值可以从地址 0x1FFFC92 读到。

3 参考示例

3.1 ADC 配置说明

ADC 采集需要的配置为：

- ADC 引脚初始化
- ADC 时钟配置
- ADC 通道选择
- ADC 使能

3.2 ADC 参考示例

ADC 模块建议的实现思路如图 3-1 所示。

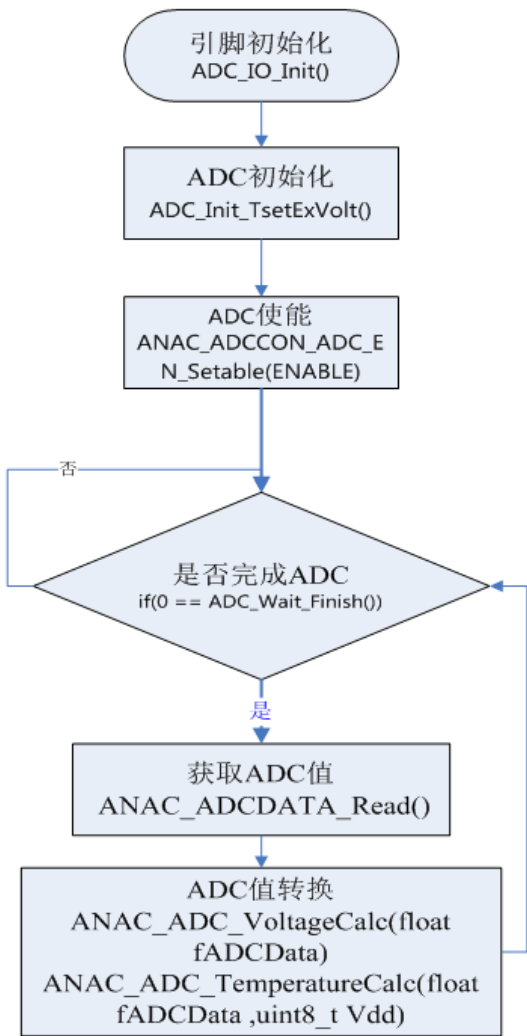


图 3-1 ADC 实现步骤

根据上图实现步骤，实现的 ADC 查询示例如图 3-2 所示。

ADC查询	2018/9/4 16:41	文件夹
ADC中断	2018/9/5 13:49	文件夹
PDR&BOR下电复位	2018/9/4 16:54	文件夹
SVD常使能查询	2018/9/4 19:46	文件夹
SVD常使能中断	2018/9/4 19:48	文件夹
XTLF停振检测	2018/9/4 19:49	文件夹
比较器查询	2018/9/4 19:35	文件夹
比较器中断	2018/9/4 19:42	文件夹

图 3-2 ADC 查询示例

示例程序部分如图 3-3 所示，本程序必须进行的配置有：



- ADC 引脚的配置: AnalogIO(GPIOF, GPIO_Pin_6);该函数将 PF6 配置为模拟输入模式, 此时 PF6 的电压作为 ADC 外部输入电压, 而温度传感器由于是内部集成, 不需要配置输入引脚。
- ADC 时钟配置: RCC_PERCLKCON2_ADCCCKSEL_Set (RCC_PERCLKCON2_ADCCCKSEL_RCHFDDIV16);该函数选择 ADC 的工作时钟为 RCHF16 分频。RCC_PERCLK_SetableEx(ADCCLK, ENABLE); 该函数使能 ADC 时钟。
- ADC 通道选择: 在测量外部电压时, ANAC_ADC_Channel_SetEx(CH_IN5) 函数用以选择通道 5, 测量温度时, ANAC_ADC_Channel_SetEx(CH_PTAT) 函数用以选择温度测量通道。
- ADC 使能: ANAC_ADCCON_ADC_EN_Setable(ENABLE);该函数用以开启 ADC 模块。

```
#include "define_all.h"

struct ADC_Struct
{
    uint16* Bufu16;    //数据缓冲区地址    //注意内存对齐
    uint08 SampleNeed; //需要采样的数据个数
    uint08 SampleCount; //已经采样的数据个数
};

struct ADC_Struct ADC_Struct;

void ADC_IO_Init(void)
{
    AnalogIO( GPIOF, GPIO_Pin_6);    //CH_IN5    PF6
}

void ADC_Init_TsetExVolt(void)
{
    //使用简单函数配置
    RCC_PERCLK_SetableEx(ANACCLK, ENABLE);    //模拟电路总线时钟使能
    RCC_PERCLK_SetableEx(ADCCLK, ENABLE);    //ADC时钟使能
    RCC_PERCLKCON2_ADCCCKSEL_Set(RCC_PERCLKCON2_ADCCCKSEL_RCHFDDIV16); //ADC工作时钟配置, 不可高于1M
    ANAC_ADC_Channel_SetEx(CH_IN5);    //ADC输入通道选择

    ANAC_ADCTRIM_Write(const_adc_TrimV_3FF);    //adc频率1M 时 计算时间2ms
    // ANAC_ADCTRIM_Write(const_adc_TrimV_1FF);    //adc频率1M 时 计算时间1ms
    // ANAC_ADCTRIM_Write(const_adc_TrimV_0FF);    //adc频率1M 时 计算时间0.5ms

    ANAC_ADCCON_ADC_IE_Setable(DISABLE);    //中断禁止
    ANAC_ADCCON_ADC_EN_Setable(DISABLE);    //ADC关闭
}

void ADC_Init_TsetTemperature(void)
{
    //使用简单函数配置
    RCC_PERCLK_SetableEx(ANACCLK, ENABLE);    //模拟电路总线时钟使能
    RCC_PERCLK_SetableEx(ADCCLK, ENABLE);    //ADC时钟使能
    RCC_PERCLKCON2_ADCCCKSEL_Set(RCC_PERCLKCON2_ADCCCKSEL_RCHFDDIV16); //ADC工作时钟配置, 不可高于1M

    ANAC_ADC_Channel_SetEx(CH_PTAT);    //ADC输入通道选择温度传感器
    ANAC_ADCCON_ADC_IE_Setable(DISABLE);    //中断禁止
    ANAC_ADCCON_ADC_EN_Setable(DISABLE);    //ADC关闭
}

uint08 ADC_Wait_Finish(void)
{
    Do_DelayStart();
    {
        if(SET == ANAC_ADCIF_ADC_IF_Chk()) return 0;
    }While_DelayMsEnd(8*clkmode); //等待8ms

    return 1; //超时
}
```

图 3-3 ADC 查询示例程序



3.3 ADC 如何在低功耗模式下使用

ADC 正常采样没法边休眠边工作。所以只能休眠唤醒后，打开工作，工作完后关闭、再进休眠。工作流程可以参考下方：

1、上电初始化 ADC

```
Test_ADC_ExVolt();//ADC 外部输入通道测试
```

2、为保证低功耗模式下功耗最低，ADC 使能必须在休眠前关闭

```
ANAC_ADCINSEL_BUFEN_Setable(DISABLE); //关闭 ADC bufen
```

```
ANAC_ADCCON_ADC_EN_Setable(DISABLE); //ADC 关闭
```

3、进入低功耗模式

```
Test_Sleep();//进入休眠 此函数可以在 sleep 例程找到
```

4、唤醒后打开 ADC 使能

```
ANAC_ADCINSEL_BUFEN_Setable(ENABLE); //打开 ADC bufen
```

```
ANAC_ADCCON_ADC_EN_Setable(ENABLE); //ADC 启动
```

5、扔掉第一个 ADC 转换值，因为第一个 ADC 转换值可能是错误的

```
ADC_Wait_Finish(); //丢弃第一个转换结果
```

```
ANAC_ADCIF_ADC_IF_Clr();//清除中断标志
```

6、获取第二个 ADC 转换值，第二个转换值是正确的。

```
if(0 == ADC_Wait_Finish()) //等待转换完成
```

```
{
```

```
    ADCData[i] = ANAC_ADCDATA_Read(); //读取 AD 值
```

```
}
```

4 示例验证

如图 4-1 所示的示例 demo 板，箭头所示分别为 PF6 和 3.291V（实测）的 VDD，将 PF6 作为外部通道输入 VDD 电压，让程序进入仿真状态，如如 4-2 所示，设置断点，当程序执行到图中断点 1 处，可以得到 fVoltage 值为 3276.01，单位是 mV，接近于 3.291V，当程序执行到断点 2 出，可以得到 fTemperature 为 24.19，单位是℃，与当前环境温度较为符合，因此验证成功。

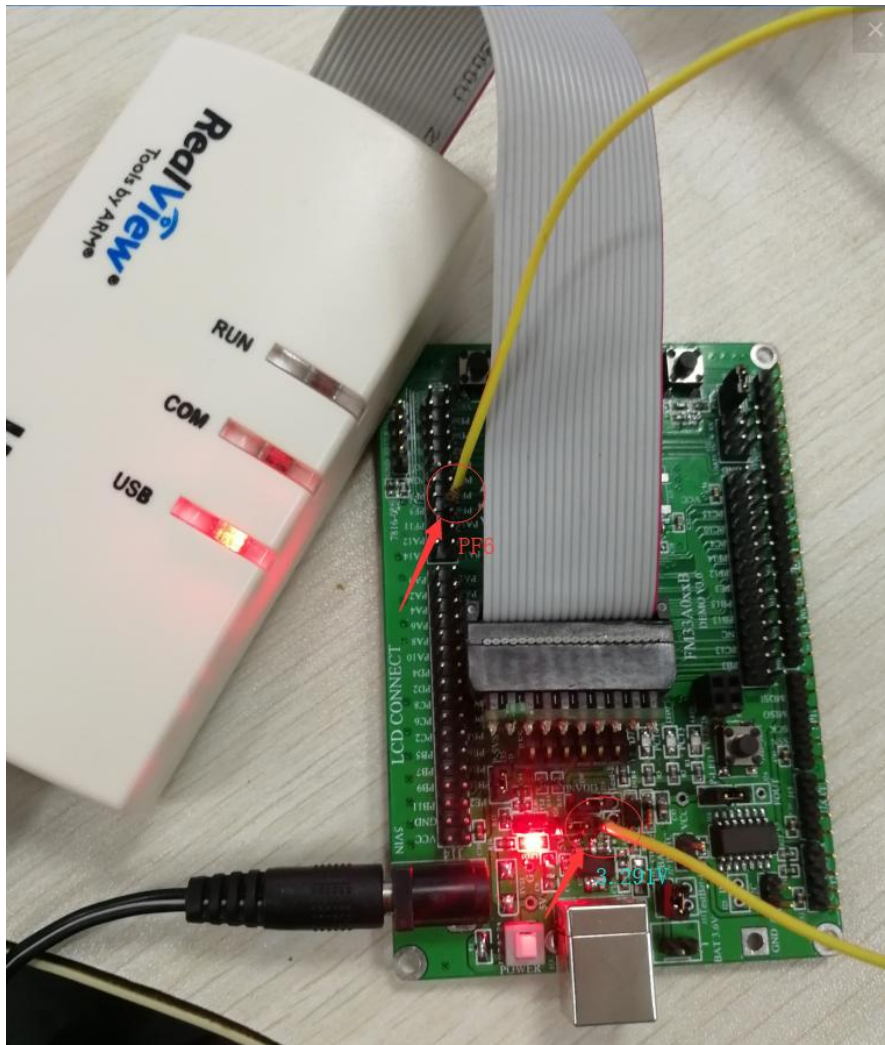


图 4-1 示例 demo 板

```

56 //ADC外部输入通道测试
57 uint32 Test_ADC_ExVolt(void)
58 {
59     uint16 ADCData[8];
60     uint08 i;
61     volatile uint32 fVlotage = 0;
62     uint32 fTempADC = 0;
63
64     ADC_IO_Init();
65     ADC_Init_TsetExVolt(); //IO口配置为ADC输入
66                             //配置ADC
67     ANAC_ADCCON_ADC_EN_Setable(ENABLE); //ADC启动
68
69     ADC_Wait_Finish();
70     ANAC_ADCIF_ADC_IF_Clr(); //丢弃第一个转换结果
71                             //清除中断标志
72
73     fTempADC = 0;
74     for(i=0; i<8; i++)
75     {
76         ANAC_ADCIF_ADC_IF_Clr(); //清除中断标志
77
78         ADCData[i] = 0;
79         if(0 == ADC_Wait_Finish()) //等待转换完成
80         {
81             ADCData[i] = ANAC_ADCDATA_Read(); //读取AD值
82             fTempADC += ADCData[i];
83         }
84         else
85         {
86             break;
87         }
88     }
89     if( i == 8 )
90     {
91         fTempADC = fTempADC/8;
92     }
93     fVlotage = ANAC_ADC_VoltageCalc(fTempADC,5); //AD值转换为电压,电源电压为5V
94     return fVlotage;
95 }
96
97
98 //ADC温度传感器通道测试
99 float Test_ADC_Temperature(void)
100 {
101     uint16 ADCData[8];
102     uint08 i;
103     volatile float fTemperature = 0;
104     float fTempADC = 0;
105
106     ADC_Init_TsetTemperature(); //配置ADC
107
108     ANAC_ADCCON_ADC_EN_Setable(ENABLE); //ADC启动
109
110     ADC_Wait_Finish();
111     ANAC_ADCIF_ADC_IF_Clr(); //丢弃第一个转换结果
112                             //清除中断标志
113
114     fTempADC = 0;
115     for(i=0; i<8; i++)
116     {
117         ANAC_ADCIF_ADC_IF_Clr(); //清除中断标志
118
119         ADCData[i] = 0;
120         if(0 == ADC_Wait_Finish()) //等待转换完成
121         {
122             ADCData[i] = ANAC_ADCDATA_Read(); //读取AD值
123             fTempADC += ADCData[i];
124         }
125         else
126         {
127             break;
128         }
129     }
130
131     if( i == 8 )
132     {
133         fTempADC = fTempADC/8.0;
134     }
135     fTemperature = ANAC_ADC_TemperatureCalc(fTempADC, 5); //AD值转换为温度
136     return fTemperature;
137 }
138
139
140
141

```

图 4-2 示例仿真



5 注意事项

1、该系列芯片采样时间：ADC 采用的是 1 阶 sigma-delta 方式，单次转换需要 256~2048 个时钟周期，ADC 工作频率 512KHz 或 1MHz，1MHz 时 ADC 温度采样时间为 3.2ms，电压采样时间为 2ms。

2、ADC 的参考电压 VREF 为 1.23V，ADC 取得是 4 倍的 VREF，即 4.92V。但是 4 倍的 VREF 不是直接产生的物理量，而是通过电容比例折算的，所以实际上 ADC 里没有 4.92V 这个电压，4.92V 是个折算值。所以 ADC 最高能转换 4.92V 的电压（理论上），一般情况下，让转换电压范围处于 1.0V~4.3V，这样比较准。另外需要注意的是，该系列芯片不支持输入比电源还高的电压，所以如果 VDD 是 3.0V，那么 ADC 转换的电压必须低于 3.0V。

3、经过测试发现，该系列芯片在使用 ADC 的电压测量功能，并且 BUFSEL(ADC 通道控制寄存器) 的 BUFEN

，芯片在进入 SLEEP 模式后会多出 10uA 左右的功耗。所以在芯片进入 SLEEP 模式前，把 BUFEN 关掉可以减少 10uA 的功耗。



附录

表 1 外设时钟控制寄存器 2

名称	PERCLKCON2							
地址	0x4000022C							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	—							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	—					ADCCKSEL		
位权限	U-0					R/W-110		
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	—							ADCCCKEN
位权限	U-0							R/W-0
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	WWDTCCKEN	RAMBISTC KEN	FLSEPCKE N	DMACKEN	LCDCKEN	AESCKEN	TRNGCKEN	CRCCCKEN
位权限	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
Bit	助记符		功能描述					
31:19	—		RFU: 未实现, 读为 0					
18:16	ADCCKSEL		ADC 工作时钟选择 000: RCLF 001: RCHF/32 010: RCHF/16 011: RCHF/8 100: RCHF/16 101/110/111: RCLF					
15:9	—		RFU: 未实现, 读为 0					
8	ADCCCKEN		ADC 时钟使能, 高使能					
7	WWDTCCKEN		WWDT 时钟使能, 高使能					
6	RAMBISTCKEN		RAMBIST 时钟使能, 高使能					
5	FLSEPCKEN		FLSC (Flash 擦写控制器) 时钟使能, 高使能					
4	DMACKEN		DMA 时钟使能, 高使能					
3	LCDCKEN		LCD 时钟使能, 高使能					



2	AESCKEN	AES 时钟使能, 高使能
1	TRNGCKEN	RNG 时钟使能, 高使能
0	CRCKEN	CRC 时钟使能, 高使能

表 2 ADC 控制寄存器通道

名称	ADCCTL							
地址	0x4001282C							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							
位权限	U-0							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	ADC_IE	-					ADC_VANA_EN	ADC_EN
位权限	R/W-0	U-0					R/W-0	R/W-0

Bit	助记符	功能描述
31:8	--	RFU: 未实现, 读为 0
7	ADC_IE	ADC 中断使能
6:2	--	RFU: 未实现, 读为 0
1	ADC_VANA_EN	外部电压通道使能 0: ADC 用作温度传感器 1: ADC 用于测量外部电压
0	ADC_EN	ADC 使能信号 0: ADC 不使能 1: ADC 使能

表 3 模拟通道 Buffer 控制寄存器

名称	ADC 输入通道选择寄存器							
地址	0x40012828							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16



位名	—							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	—							
位权限	U-0							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	—		BUFEN	BUFBYP	BUFSEL			
位权限	U-0		R/W-0	R/W-0	R/W-0000			
Bit	助记符		功能描述					
31:6	—		RFU：未实现，读为 0					
5	BUFEN		ADC 输入通道 Buffer 使能					
4	BUFBYP		ADC 输入 Buffer Bypass 使用 ADC 测量外部信号输入时，不要 Bypass Buffer 使用 ADC 测量电源电压时，必须将此位置 1					
3:0	BUFSEL		ADC 输入通道选择 0000-0101：保留，禁止使用 0110：VDD 主电源 0111：保留 1000：ADC_IN1（PC12） 1001：ADC_IN2（PC13） 1010：ADC_IN3（PD0） 1011：ADC_IN4（PD1） 1100：ADC_IN5（PF6） 1101：ADC_IN6（PC15） 1110：ADC_IN7（PB2） 1111：ADC_IN8（PB3）					

表 4 ADC 中断位控制寄存器

名称	ADCCTL							
地址	0x4001282C							
位	Bit31	Bit30	Bit29	Bit28	Bit27	Bit26	Bit25	Bit24
位名	-							
位权限	U-0							
位	Bit23	Bit22	Bit21	Bit20	Bit19	Bit18	Bit17	Bit16
位名	-							
位权限	U-0							
位	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
位名	-							



位权限	U-0							
位	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
位名	ADC_IE	-					ADC_VANA_EN	ADC_EN
位权限	R/W-0	U-0					R/W-0	R/W-0
Bit	助记符		功能描述					
31:8	--		RFU: 未实现, 读为 0					
7	ADC_IE		ADC 中断使能					
6:2	--		RFU: 未实现, 读为 0					
1	ADC_VANA_EN		外部电压通道使能 0: ADC 用作温度传感器 1: ADC 用于测量外部电压					
0	ADC_EN		ADC 使能信号 0: ADC 不使能 1: ADC 使能					



版本信息

版本号	发布日期	更改说明
1.0	2018.10	首次发布
1.1	2019.06	按最新例程 ADC_TRIM 更新
1.2	2019.08	修正 ADC_TRIM 的错误说明



上海复旦微电子集团股份有限公司销售及服务中心

上海复旦微电子集团股份有限公司

地址：上海市国泰路 127 号 4 号楼

邮编：200433

电话：(86-021) 6565 5050

传真：(86-021) 6565 9115

上海复旦微电子（香港）股份有限公司

地址：香港九龙尖沙咀东嘉连威老道 98 号东海商业中心 5 楼 506 室

电话：(852) 2116 3288 2116 3338

传真：(852) 2116 0882

北京办事处

地址：北京市东城区东直门北小街青龙胡同 1 号歌华大厦 B 座 423 室

邮编：100007

电话：(86-10) 8418 6608

传真：(86-10) 8418 6211

深圳办事处

地址：深圳市华强北路 4002 号圣廷苑酒店世纪楼 1301 室

邮编：518028

电话：(86-0755) 8335 0911 8335 1011 8335 2011 8335 0611

传真：(86-0755) 8335 9011

台湾办事处

地址：台北市 114 内湖区内湖路一段 252 号 12 楼 1225 室

电话：(886-2) 7721 1889

传真：(886-2) 7722 3888

新加坡办事处

地址：237, Alexandra Road, #07-01, The Alexcior, Singapore 159929

电话：(65) 6472 3688

传真：(65) 6472 3669

北美办事处

地址：2490 W. Ray Road Suite#2 Chandler, AZ 85224 USA

电话：(480) 857-6500 ext 18

公司网址：<http://www.fmsh.com/>

