文档版本	V1.0
发布日期	20191108

APT32F172 ADC 应用开发指南

目录

1 概述

本文介绍了在APT32F172中使用ADC的应用范例。

2. 适用的硬件

该例程使用于 APT32F172 开发板 APT-DB172

3. 应用方案代码说明

基于 APT32F172 完整的库文件系统，可以很方便的对 ADC 进行配置。

ADC 注意事项：

1.12bitADC 转换速率不能超过 500KSPS;10bitADC 转换速率不能超过 1MSPS。

2.选择内部参考电压 Vref 需要接 104 电容到 GND。

3.1 ADC 单次转换模式

软件配置：
可在 apt32f172_initial.c 文件中进行初始化的配置；
开启内部主频 20MHz,并作为系统时钟。
使能 ADCIN10、 ADCIN11 通道,12BIT ADC,参考电压选择内部 2.048V,单次转换模式，PRLVAL=2， ADC 采样周期=3。
ADC 转换周期： F_ANA=PCLK/(2*PRLVAL)-> F_ANA=20M/(2*2)=0.2us
ADC 转换时间： ADC 采样周期+1 转换周期*12bit（或 10bit） +3 个处理结果周期=18 个转换周期=18*0.2us=277KSPS

```
/*************************************************************/
//ADC Functions
//EntryParameter:NONE
//ReturnValue:NONE
/*************************************************************/
void ADC_CONFIG(void)
{
ADC12_RESET_VALUE(); //ADC 所有寄存器复位赋值
ADC12_CLK_CMD(ADC_CLK_CR , ENABLE); //使能 ADC CLK
ADC12_Software_Reset(); //ADC 软件复位
/*************************************************************/
//GPIO Functions
//EntryParameter:NONE
//ReturnValue:NONE
```

```
ADC12_Configure_Mode(ADC12_12BIT , One_shot_mode ,0, 2 , 2);
//选择 12BIT ADC; 单次模式; 转换优先序列寄存器为 0; ADC_CLK=PCLK/2*2=0.2us; 转换序列个数为
2
ADC12_Configure_VREF_FVR(ADC12_FVR_2_048V); //使用内部为参考电压 2.048mV
ADC12_ConversionChannel_Config(ADC12_ADCIN10,ADC12_3CYCLES,ADC12_CV_RepeatNum1,A
DC12_AVGDIS,0);
//转换序列 0,选择 ADCIN10 通道, 3 个转换周期, 连续重复采样次数为 1,平均值计算禁止
ADC12_ConversionChannel_Config(ADC12_ADCIN11,ADC12_3CYCLES,ADC12_CV_RepeatNum1,A
DC12_AVGDIS,1);
//转换序列 1,选择 ADCIN11 通道, 3 个转换周期, 连续重复采样次数为 1,平均值计算禁止
ADC12_CMD(ENABLE); //使能 ADC 模块
ADC12_ready_wait(); //等待 ADC 模块配置完成
ADC12_Control(ADC12_START); //ADC 模块启动
}
/*****************************************************************************/
//APT32F172_init /
//EntryParameter:NONE /
//ReturnValue:NONE /
/*****************************************************************************/
void APT32F172_init(void)
{
SYSCON_WDT_CMD(DISABLE); //关闭 WDT
SYSCON->PCER0=0xFFFFFFF; //使能 IP
SYSCON->PCER1=0xFFFFFFF; //使能 IP
while(!(SYSCON->PCSR0&0x1)); //判断 IP 是否使能
SYSCON_Int_Enable(); //使能 SYSCON 中断向量
SYSCON->IECR=ISOSC_ST|IMOSC_ST|EMOSC_ST|SYSCLK_ST;
//使能 ISOSC 时钟稳定中断,使能 IMOSC 时钟稳定中断,使能 EMOSC 时钟稳定中断
CK_CPU_EnAllNormalIrq(); //打开全局中断
SYSCON_CONFIG(); //syscon 参数 初始化
GPIO_CONFIG(); //GPIO 初始化
ADC_CONFIG (); //ADC 初始化
}
/*****************************************************************/
//main
/*****************************************************************/
volatile U32_T R_ADC_Buf1, R_ADC_Buf2;
int main(void)
{
APT32F172_init();
while(1)
{
SYSCON_IWDCNT_Reload(); //清狗
ADC12_SEQEND_wait(0); //等待转换序列 0 转换完成
```

```
ADC12_SEQEND_wait(1); //读取转换序列 1 数据
R_ADC_Buf2= ADC0->DR[1]; //转换结果保存
ADC12_Control(ADC12_START); //ADC 模块启动
}
}
```

3.3 ADC 连续转换模式

软件配置：
开启内部主频 20MHz,并作为系统时钟。
使能 ADCIN10、 ADCIN11 通道,12BIT ADC,参考电压选择外部 Vref,连续转换模式，PRLVAL=2， ADC 采样周期=3。
ADC 转换周期： F_ANA=PCLK/(2*PRLVAL)-> F_ANA=20M/(2*2)=0.2us
ADC 转换时间： ADC 采样周期+1 转换周期*12bit（或 10bit） +3 个处理结果周期=18 个转换周期=18*0.2us=277KSPS

```
/*************************************************************
//ADC Functions
//EntryParameter:NONE
//ReturnValue:NONE
*************************************************************/
void ADC_CONFIG(void)
{
ADC12_RESET_VALUE(); //ADC 所有寄存器复位赋值
ADC12_CLK_CMD(ADC_CLK_CR , ENABLE); //使能 ADC CLK
ADC12_Software_Reset(); //ADC 软件复位
ADC12_Configure_Mode(ADC12_12BIT , One_shot_mode ,0, 2 , 2);
//选择 12BIT ADC; 单次模式; 转换优先序列寄存器为 0; ADC_CLK=PCLK/2*2=0.2us; 转换序列个数为 2
ADC12_Configure_VREF_FVR(ADC12_FVR_2_048V); //使用内部为参考电压 2.048mV
ADC12_ConversionChannel_Config(ADC12_ADCIN10,ADC12_3CYCLES,ADC12_CV_RepeatNum1,AD
C12_AVGDIS,0);
//转换序列 0,选择 ADCIN10 通道,3 个转换周期, 连续重复采样次数为 1,平均值计算禁止
ADC12_ConversionChannel_Config(ADC12_ADCIN11,ADC12_3CYCLES,ADC12_CV_RepeatNum1,AD
C12_AVGDIS,1);
//转换序列 1,选择 ADCIN11 通道,3 个转换周期, 连续重复采样次数为 1,平均值计算禁止
ADC12_CMD(ENABLE); //使能 ADC 模块
ADC12_ready_wait(); //等待 ADC 模块配置完成
ADC12_Control(ADC12_START); //ADC 模块启动
}
/*****************************************************************************/
//APT32F172_init /
//EntryParameter:NONE /
```

```
//ReturnValue:NONE /
/*******************************************************************************/
void APT32F172_init(void)
{
SYSCON_WDT_CMD(DISABLE); //关闭 WDT
SYSCON->PCER0=0xFFFFFFF; //使能 IP
SYSCON->PCER1=0xFFFFFFF; //使能 IP
while(!(SYSCON->PCSR0&0x1)); //判断 IP 是否使能
SYSCON_Int_Enable(); //使能 SYSCON 中断向量
SYSCON->IECR=ISOSC_ST|IMOSC_ST|EMOSC_ST|SYSCLK_ST;
//使能 ISOSC 时钟稳定中断,使能 IMOSC 时钟稳定中断,使能 EMOSC 时钟稳定中断
CK_CPU_EnAllNormalIrq(); //打开全局中断
SYSCON_CONFIG(); //syscon 参数 初始化 25 / 69
GPIO_CONFIG(); //GPIO 初始化
ADC_CONFIG (); //ADC 初始化
}
/*************************************************************/
//main
/*************************************************************/
volatile U32_T R_ADC_Buf1, R_ADC_Buf2;
int main(void)
{
APT32F172_init();
while(1)
{
SYSCON_IWDCNT_Reload(); //清狗
ADC12_SEQEND_wait(0); //等待转换序列 0 转换完成
R_ADC_Buf1= ADC0->DR[0]; //转换结果保存
```

```
ADC12_SEQEND_wait(1); //读取转换序列 1 数据
R_ADC_Buf2= ADC0->DR[1]; //转换结果保存
 }
}
```

3.3 EPWM 事件触发 ADC 单次转换模式

- PA1.4->ADC0， PA1.5->ADC1
- 选择内部 FVR=2.048V 作为参考电压
- ADC 采样触发通过 EPWM 事件触发
- ADC 采用单次触发模式
- 转换结果在中断中读取

```
/*****************************************************************/
//ADC Functions
//EntryParameter:NONE
//ReturnValue:NONE
/*****************************************************************/
void ADC_CONFIG(void)
{
ADC12_RESET_VALUE();//ADC 所有寄存器复位赋值
ADC12_CLK_CMD(ADC_CLK_CR , ENABLE); //使能 ADC CLK
ADC12_Software_Reset();//ADC 软件复位
ADC12_Configure_Mode(ADC12_12BIT , One_shot_mode ,0, 2 , 2); //选择 12BIT ADC; 单次模式; 转换优先
序
列寄存器为 0; ADC_CLK=PCLK/2*2=0.2us; 转换序列个数为 2
//ADC12_Configure_VREF_VDD(); //使用 VDD 为参考电压
ADC12_Configure_VREF_FVR(ADC12_FVR_2_048V);//使用内部为参考电压 2.048mV
//ADC12_Configure_VREF_EX(); //使用外部 Vref 为参考电压
ADC12_ConversionChannel_Config(ADC12_ADCIN0,ADC12_3CYCLES,ADC12_CV_RepeatNum1,ADC12_AVGDI
```

```
S,0);//转换序列 0,选择 ADCIN10 通道, 3 个转换周期, 连续重复采样次数为 1,平均值计算禁止
ADC0->SEQ[0] |=0x3<<13;//EPMW 触发 ADC 通道 0
ADC12_ConversionChannel_Config(ADC12_ADCIN1,ADC12_3CYCLES,ADC12_CV_RepeatNum1,ADC12_AVGDI
S,1);//转换序列 1,选择 ADCIN11 通道, 3 个转换周期, 连续重复采样次数为 1,平均值计算禁止
ADC0->SEQ[1] |=0x3<<13;//EPMW 触发 ADC 通道 0
ADC12_CMD(ENABLE); //使能 ADC 模块
ADC12_ready_wait();//等待 ADC 模块配置完成
//ADC12_Control(ADC12_START); /启动 ADC 转换
ADC12_ConfigInterrupt_CMD(ADC12_SEQ_END0,ENABLE);
ADC12_ConfigInterrupt_CMD(ADC12_SEQ_END1,ENABLE);
ADC_Int_Enable();//使能 ADC 中断向量
}
/*************************************************************/
//ADC Interrupt
//EntryParameter:NONE
//ReturnValue:NONE
/*************************************************************/
 void ADCIntHandler(void)
{
// ISR content ...
if((ADC0->SR&ADC12_SEQ_END0)==ADC12_SEQ_END0)
{
ADC0->CSR = ADC12_SEQ_END0;
R_ADC_Buf1=ADC0->DR[0];
}
if((ADC0->SR&ADC12_SEQ_END1)==ADC12_SEQ_END1)
{
ADC0->CSR = ADC12_SEQ_END1;
R_ADC_Buf2=ADC0->DR[1];
```

```
}
}
```

应用注意事项：

- 在使用 ADC 作为内部 FVR 做参考时,vref 脚位需要接外部 104 电容，不能做其他功能使用。

4. 程序下载和运行

1．将目标板与仿真器连接，分别为 VDD SCLK SWIO GND

2. 将需要采集 AD 的口连接到配置好的 ADC 口

3. 程序编译后仿真运行

4. R_ADC_Buf1 和 R_ADC_Buf2 两个变量得到的就是需要采集的 ADC 数据

5. 改版历史

版本	修改日期	修改概要
V1.0	2019-11-08	初版