浙江工業大學
ZHEJIANG UNIVERSITY OF TECHNOLOGY

DSP 原理及应用

实验 7：ADC 采样和信号滤波实验

实验报告

姓　　名：林宇航

班　　级：自动化 1901

学　　号：201906060308

学　　院：信息工程学院

设计日期　2022.4.26

实验七 ADC 采样和信号滤波实验

1. 实验目的

1）了解 TMS320F28335 片上外设 AD 转换器
2）熟悉片上 AD 的使用
3）利用片上 AD 进行采集

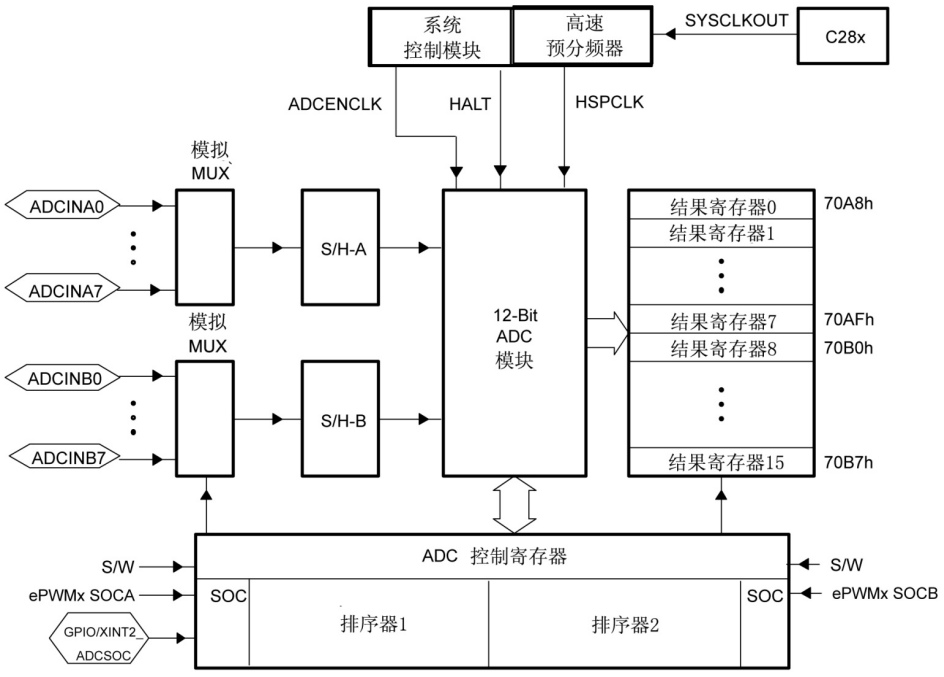
2. 实验主要内容

1）在 CCS 软件中，配置 adc 的各种属性并采集到模拟信号。
2）编写程序配置 ePWM，使得其以 100kHZ 的频率触发 adc 进行采样。
3）编写程序进行过采样，以及低通滤波。

3. 实验基本原理

1）adc 模块

模数转换模块 ADC 有 16 个通道，可配置为 2 个独立的 8 通道模块，服务于 ePWM 模块。两个独立的 8 通道模块也可以级联构成一个 16 通道模块。尽管在模数转换模块中有多个输入通道和两个排序器，但仅有一个转换器。ADC 模块的功能框图如图所示。



2）数字滤波器原理

一阶低通滤波器的公式为：

$$G(s)=\frac{y}{x}=\frac{1}{\frac{s}{\omega_c}+1}$$

将其离散化后，可得到：

$$y(k)=\frac{1}{1+T_s\omega_c}y(k-1)+\frac{T_s\omega_c}{1+T_s\omega_c}x(k)$$

根据需要采集的信号频率来计算其系数。

4. 实验过程和关键程序解读

1. 启动 CCS，进入 CCS 的操作环境，并导入 AD 工程。
2. 加载 AD 工程，添加 NewTargetConfiguration.ccxml 文件
3. 阅读源代码

1）初始化系统控制寄存器与要使用的 GPIO：

```
// Step 1. 初始化系统控制寄存器:
   InitSysCtrl();

// Step 2. 初始化IO口
   InitGpio();
```

2）PIE 相关初始化：

```
// Step 3.初始化PIE控制寄存器
   DINT;
   InitPieCtrl();

   IER = 0x0000;
   IFR = 0x0000;

// 初始化PIE中断向量表
   InitPieVectTable();
```

3）重映射中断向量表；

```
// 重映射中断服务函数
   EALLOW;  // 关闭寄存器写保护
   PieVectTable.ADCINT = &adc_isr;
   EDIS;    // 开启寄存器写保护
```

4）中断使能；

```
// Step 5. 中断使能
   PieCtrlRegs.PIEIER1.bit.INTx6 = 1;
   IER |= M_INT1; // 使能中断1
   EINT;          // 使能全局中断
   ERTM;          // 使能全局实时（realtime）中断
```

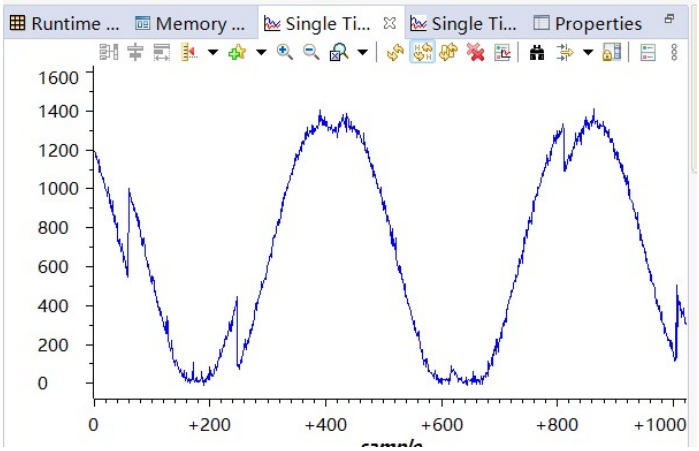
5）配置 adc 寄存器；

```
   AdcRegs.ADCTRL1.bit.ACQ_PS = ADC_SHCLK;  // 采样保持时钟
   AdcRegs.ADCTRL3.bit.ADCCLKPS = ADC_CKPS; // ADC采集时钟
   AdcRegs.ADCTRL1.bit.SEQ_CASC = 1;        // 0 双排序器; 1 级联运行
   AdcRegs.ADCTRL2.bit.EPWM_SOCA_SEQ1 = 1;  //允许EPWM触发SEQ1转换
   AdcRegs.ADCTRL2.bit.INT_ENA_SEQ1 = 0x1;  // 中断使能
   AdcRegs.ADCTRL2.bit.RST_SEQ1 = 0x1;
   AdcRegs.ADCCHSELSEQ1.bit.CONV00 = 0x6;   //打开通道1, 下同
   AdcRegs.ADCMAXCONV.bit.MAX_CONV1 = 15;   //转换次数 16
   AdcRegs.ADCTRL2.bit.SOC_SEQ1 = 0x1 ;     //软件触发
```

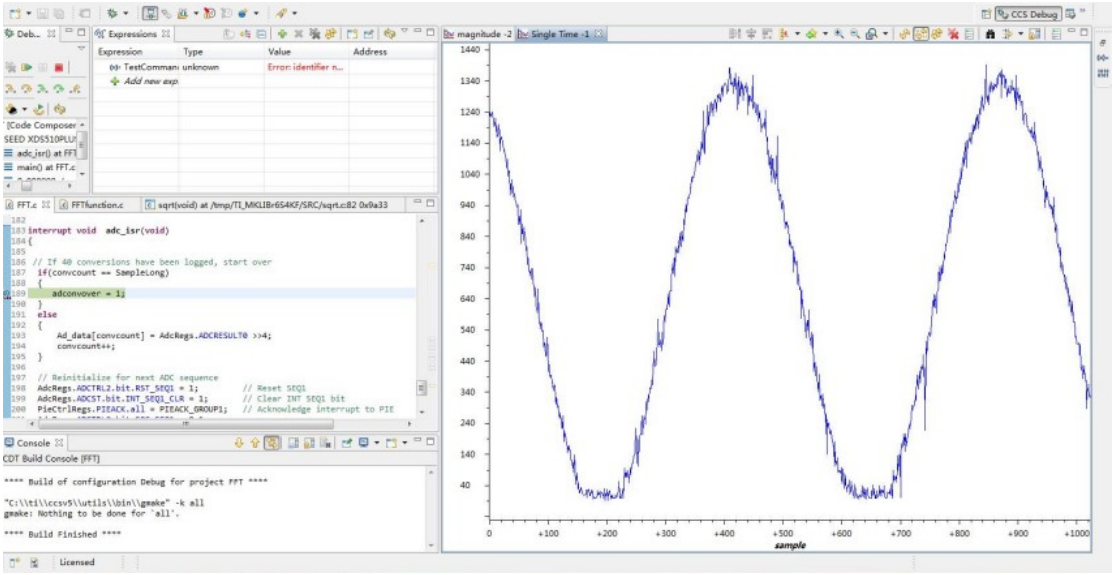
6）adc 中断服务函数

```
interrupt void  adc_isr(void)
{
  Voltage1[ConversionCount] = AdcRegs.ADCRESULT0 >>4;

  // 40次转换完成
  if(ConversionCount == 1024)
  {
     ConversionCount = 0;
  }
  else ConversionCount++;

  // Reinitialize for next ADC sequence
  AdcRegs.ADCTRL2.bit.RST_SEQ1 = 1;          // 复位排序器1
  AdcRegs.ADCST.bit.INT_SEQ1_CLR = 1;        // 清除中断标志位
  PieCtrlRegs.PIEACK.all = PIEACK_GROUP1;    // 响应中断
  AdcRegs.ADCTRL2.bit.SOC_SEQ1 = 0x1 ;       // 软件触发
  return;
}
```

其中值得注意的是在中断中也有软件触发，所以 DSP 不停得进入中断并触发采集。在实验箱上配置信号发送器输出正弦波，幅值 1V，频率 300Hz。将程序烧录进入后，进入 debug 模式，点击 Tool 选项中的 Graph，选择 signle Time 单变量观察，将记录 adc 采样结果的数组(Voltage1)地址写入，选择 16 位无符号数，选择数据量与显示的量为 1024，理论上可以看到有 3 个周期的正弦波。



这里波形出现撕裂的原因是数据变换太快，波形来不及显示，我们将断点打在中断中结束采集 1024 次的条件中即可观察到较为正常的波形：



7）配置 ePWM 触发 ADC

```
169    EPwm3Regs.ETSEL.bit.SOCAEN = 1;        // 使能epwm触发adc中断
170    EPwm3Regs.ETSEL.bit.SOCASEL = 4;       // 选择当前计数器递增至CMPA作为触发事件
171
172    EPwm3Regs.ETPS.bit.SOCAPRD = 1;        // 在第一个触发事件后即产生触发脉冲
173    EPwm3Regs.TBCTL.bit.CTRMODE = 0;       //增计数模式
174    EPwm3Regs.TBCTL.bit.HSPCLKDIV = 1;     //二分频
175    EPwm3Regs.TBCTL.bit.CLKDIV = 0;        //不分频
176    EPwm3Regs.TBPRD = 749;                 //设置周期    150M / 2 / 1 /  (749+1)
177    EPwm3Regs.CMPA.half.CMPA = EPwm3Regs.TBPRD / 2;  //占空比50%
```

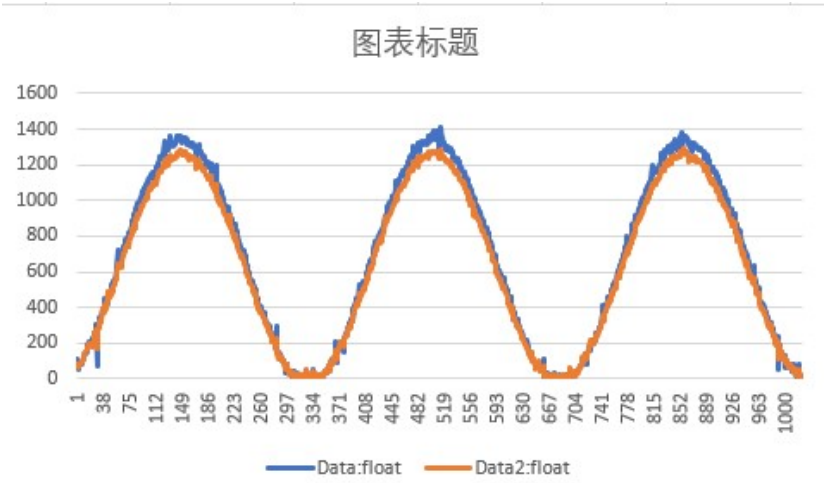
为了验证 ePWM 成功触发 adc，应关闭主循环与中断中的软件触发，结果也是能采集到与软件触发相似的波形，说明 ePWM 成功触发。

8）过采样

简单理解就是取 16 次的平均

```
186  Voltage1[ConversionCount] = AdcRegs.ADCRESULT0 >>4;
187  Voltage2[ConversionCount] = AdcRegs.ADCRESULT0 >>4/16 + AdcRegs.ADCRESULT1>>4/16+AdcRegs.ADCRESULT2>>4/16+AdcRegs.ADCRESULT3>>4/16
188          +AdcRegs.ADCRESULT4>>4/16+AdcRegs.ADCRESULT5>>4/16+AdcRegs.ADCRESULT6>>4/16+AdcRegs.ADCRESULT7>>4/16;
189  Voltage2[ConversionCount] += AdcRegs.ADCRESULT8>>4/16 + AdcRegs.ADCRESULT9>>4/16+AdcRegs.ADCRESULT10>>4/16+AdcRegs.ADCRESULT11>>4/1
190          +AdcRegs.ADCRESULT12>>4/16+AdcRegs.ADCRESULT13>>4/16+AdcRegs.ADCRESULT14>>4/16+AdcRegs.ADCRESULT15>>4/16;
```

除了观察波形外，CCS 还可以将数据导出，在波形图右键，选择 Data 选项，导出为 csv 文件，利用 excel 画图工具，生成波形图：



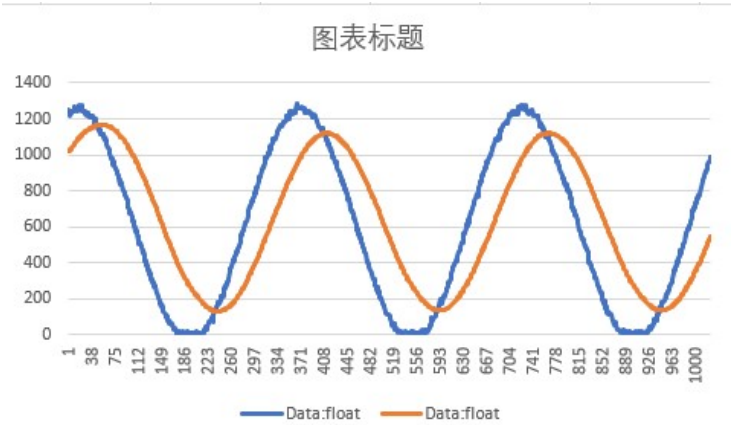
9）对 300Hz 信号进行低通滤波

经过计算可以得出，y(k-1)前的系数为 0.9814779794226468，x(k)即 1 - y(k-1)前的系数。

$$y(k)=\frac{1}{1+T_s\omega_c}y(k-1)+\frac{T_s\omega_c}{1+T_s\omega_c}x(k)$$

```
188  Voltage2[ConversionCount] = Voltage2[ConversionCount] /16;
189  if(ConversionCount>1)
190  {
191      Voltage2[ConversionCount] = k1 * Voltage2[ConversionCount-1] + k2 * Voltage1[ConversionCount];
192  }
193  else
194  {
195      Voltage2[ConversionCount] = k1 * Voltage2[1024] + k2 * Voltage1[ConversionCount];
196  }
```

用相同的方法导出数据，画图得：



很明显可以看出采样前的信号有较多的高频噪声，即毛刺，采样后就变得十分的平滑，利用电路原理中的低通滤波器也可以得到滤波后的信号幅值衰减到 0.707 倍，相位滞后 45°，所以实验结果是很完美的。

5. 实验总结与思考

这次实验是 DSP 最后的一次实验，做的量也相较以往的更多，我们深入实践了如何用 DSP 中的片上外设 AD 转换器来采集外部输入的信号，并且对信号做数字上的处理使其的特性更好，更接近真实的波形。想起了上学期的电子系统设计大作业，我们设计了一个语言存储与回放系统，将麦克风中的微小信号进行放大，滤波，加偏置传到单片机的 DA 管脚，再通过 AD 输出，经过一系列处理给扬声器，当时为了更好的还原声音，我们不停得加单片机的采样频率，做了这次实验又想到了可以不仅仅在模拟滤波上做文章，也可以在单片机里进行数字滤波，没准效果会更好。