



# DSP 应用实验报告

## 实验十一：DSP 数据采集

院    系：电子工程与光电技术学院

专    业：电子信息工程

姓    名：赵婧萱

学    号：920104330118

指导老师：李戡晟

2023 年 5 月 25 日

# 目录

11.1 实验目的 .....	3
11.2 实验仪器 .....	3
11.3 实验内容 .....	3
11.4 实验步骤 .....	3
11.5 实验思考 .....	11
11.6 实验总结及问题 .....	15
11.7 实验体会 .....	16

# 实验十一：DSP 数据采集

## 11.1 实验目的

1. 熟悉 DSP 的软硬件开发平台
2. 掌握 TMS320F28335 的 ePWM 中时间基准子模块和事件触发子模块的基本使用方法
3. 熟悉 TMS320F28335 的中断设置
4. 掌握 TMS320F28335 的 ADC 模块的基本使用方法
5. 掌握代码调试的基本方法

## 11.2 实验仪器

计算机，TMS320F28335 DSP 教学实验箱，XDS510 USB 仿真器，示波器，信号源

## 11.3 实验内容

建立工程，编写 DSP 的主程序，对工程进行编译、链接，利用现有 DSP 实验平台实现数据的采集、存储以及模拟还原，并采取多种方法予以验证。

## 11.4 实验步骤

1. 连接 TMS320F28335 教学实验箱至计算机。打开计算机和实验箱电源，将 SMA 输出端口 J5 连接至示波器，信号发生器连接至 SMA 输入端口 J2；
2. 点击桌面 CCSv5 快捷方式，启动 CCS 集成开发环境；
3. 导入范例工程文件“Lab\_11”至目录，完成各项设置后运行程序；
  - a) 数据存储的原理：

阅读源程序，使用中断方式完成 ADC 内部数据读取；程序开头由预留存

放采样数据的数组和计数变量，如图 1 所示：

```

    Uint16 SampleTable1[BUF_SIZE]={0};

    Uint16 ConvCount;
  
```

图 1：采样数据数组和计数变量

在中断服务程序读取 AD 结果之后，需实时保存 AD 结果以便后续数据写 DAC。

中断服务程序触发的条件是 ADC 采样信号到来，原先范例程序中直接将信号输送给 DA，实现数据实时的输入输出。

将采集到的信号同时存入 SampleTable1 数据空间，可以实现数据保存。然而下一个中断到来时会覆盖原存储数据，因此设定了一个数据存放位置指示变量 ConvCount，每次存放数据后，变量地址自动加 1。当指示变量超过 SampleTable1 数据空间的长度（1024 点）后，置零 ConvCount，以实现 SampleTable1 数据空间中的数据不断更新；

综合以上内容，我们在中断函数内编写程序，将采集到的数据存放在数组内保存起来，具体代码如图 2 所示：

```

345 interrupt void epwm1_timer_adc_isr(void)    //中断函数
346 {
347
348 //DA
349   xn= (AdcRegs.ADCRESULT1 & 0xFFF0);
350   if(ConvCount<1024)
351   {
352     SampleTable1[ConvCount] = xn;
353     ConvCount++;
354   }
355   else
356   {
357     ConvCount=0;
358   }
359
360 /*   xn= (AdcRegs.ADCRESULT1 & 0xFFF0);
361     SampleTable1[ConvCount] = xn;
362     ConvCount++;
363     */
364   *Da_out= xn ;
  
```

图 2：中断保存采样数据程序代码

#### 4. 查询空间地址

修改好程序后进行编译，已知采集到的数据将存储在 SampleTable1 中，为观察其起始地址，将其添加至观察窗口，如图 3 所示：

Expression	Type	Value	Address
> SampleTable1	unsigned int[1024]	0x0000C040@Data	0x0000C040@Data
> *init_zone7	void (*)()	0x0099BF	
> *init_mcbasp_spi	void (*)()	0x0099FA	
> *mcbasp_write	void (*) (unsigned short)	0x009A32	
> *init_AD9747	void (*)()	0x009A3B	
> *main	void (*)()	0x00998B	
> *Da_out	unsigned int *	0x00200400	0x0000C006@Data

图 3：数据存储空间地址

由此可知，通过变量观察窗口，观察到变量 SampleTable1 的起始地址为 0x0000C040@data。

## 5. 验证数据采集的正确性

利用“graph”图形工具，查看存储空间中保存的时域波形，是否为信号源输出的信号波形。其参数设置如图 4 所示：

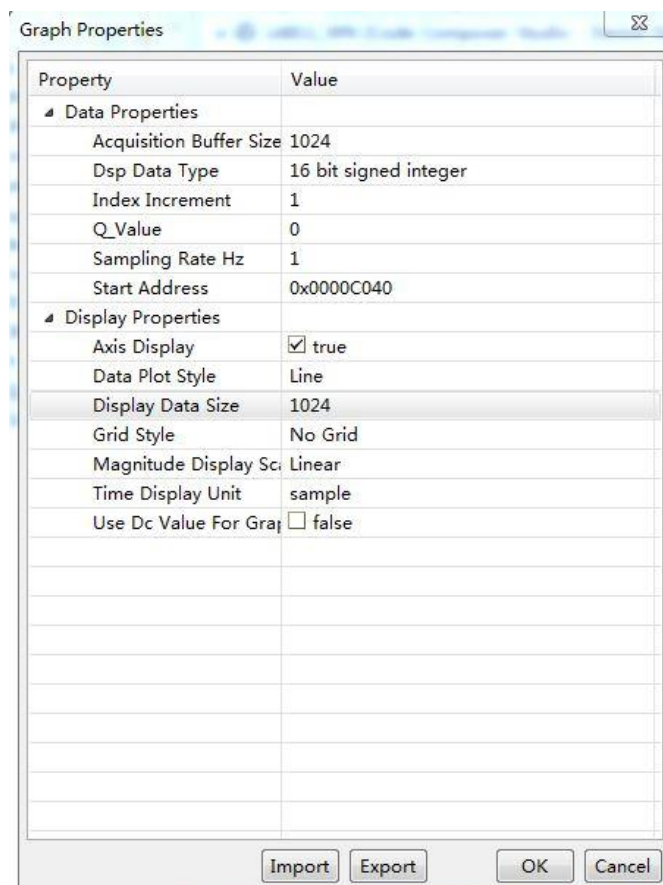


图 4：图形参数设置

令输入信号频率为 2kHz，得到采集的数据作图如图 5 所示：

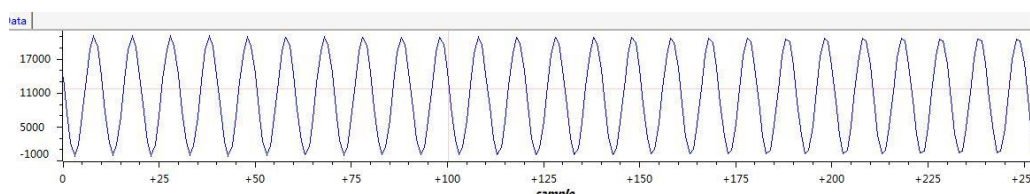


图 5：输入信号 2kHz 时存储 20kHz 波形

## 6. 示波器输出波形

连接示波器，Autoset 后将显示对应波形，如图 6 所示：

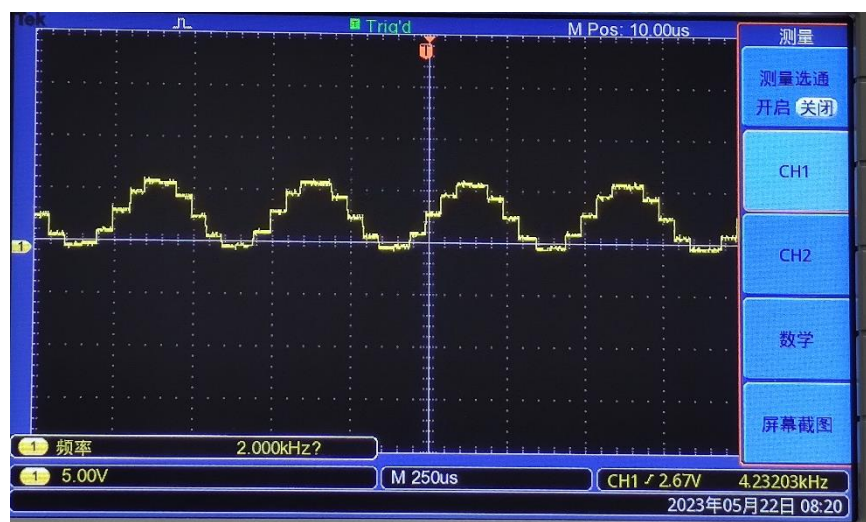


图 6: 示波器显示 2kHz 波形

## 7. 修改 ADC 采样频率

```
// Set Period for EPWM1
EPwm1Regs.TBPRD = 208; //设定时间基准器计数器的周期
EPwm1Regs.TBCTL.bit.CTRMODE = TB_COUNT_UPDOWN; //增减计
// Setup Compare A = 2 TBCLK counts
EPwm1Regs.CMPA.half.CMPA = 2; //计数比较寄存器A CMPA
// Phase is 0 for Synchronization Event
```

图 7: 控制 ADC 采样频率的寄存器

如图 7 所示, 阅读程序可知, 在函数 `void InitEPwm1Parameters(void)` 中设置采样频率。范例程序中, TB 计数模式为增减计数, 且每三次事件将产生一次采样中断, 其计算公式为:

$$T_{PWM1} = \frac{TBCLK}{(TBPRD * 2 * 3)} = \frac{25}{208 * 2 * 3} = 0.02\text{MHz} = 20\text{kHz}$$

故改变 TBPRD 即可改变采样频率, 不同的 TBPRD 对应的采样频率如表 1 所示:

TBPRD	Fs (kHz)
208	20
139	30
149	27.9
104	40

表 1: TBPRD 对应采样频率

其中, 高速时间基准时钟预分频位为 010b, 即六分频, 时间基准时钟预分频位为 000b, 即一分频, 故



$$TBCLK = \frac{SYSCLKOUT}{(HSPCLKDIV * CLKDIV)} = \frac{150}{6 * 1} = 25\text{MHz}$$

修改采样频率为 40KHZ (图 8、9):

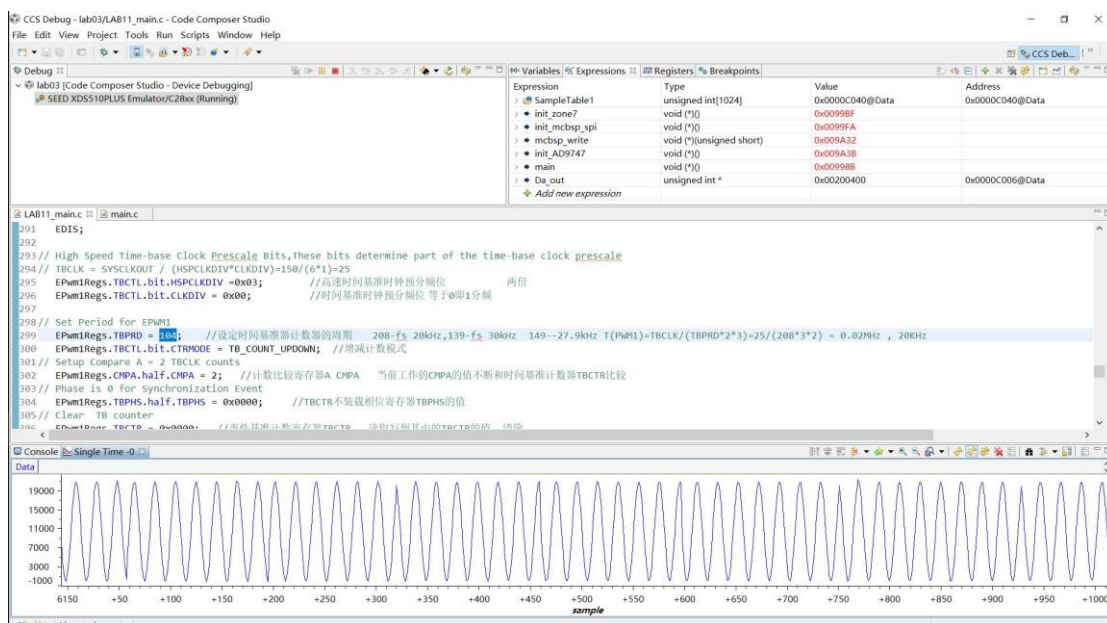


图 8: 修改采样频率为 40KHZ 的程序改动

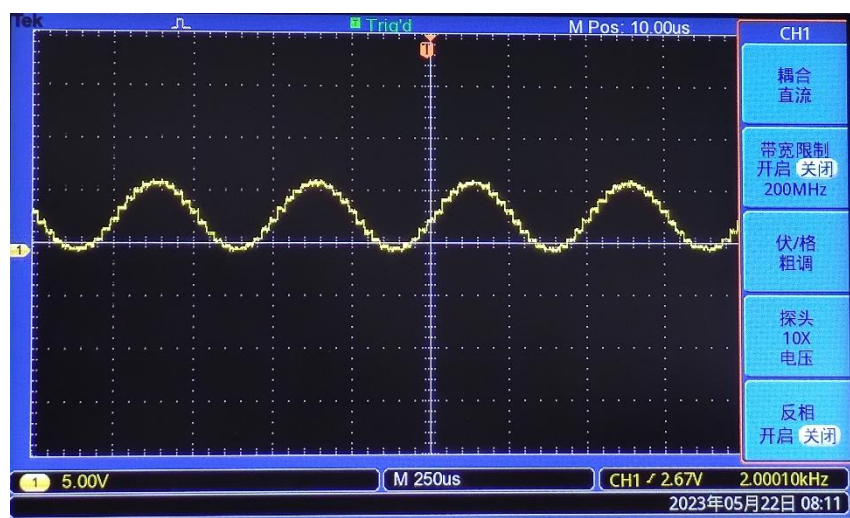


图 9: 修改采样频率为 40KHZ 的输出波形

修改采样频率为 10KHZ (图 10、11):

```
// Set Period for EPWM1
EPwm1Regs.TBPRD = 416; //设定时间基准器
EPwm1Regs.TBCTL.bit.CTRMODE = TB_COUNT_U
// Setup Compare A = 2 TBCLK counts
EPwm1Regs.CMPA.half.CMPA = 2; //计数比
```

图 10: 修改采样频率为 10KHZ 的程序改动



图 11: 修改采样频率为 10KHZ 的输出波形

## 8. 观察采样频率变化下的不同输出波形

改变信号源频率，示波器输出结果如下列图所示：

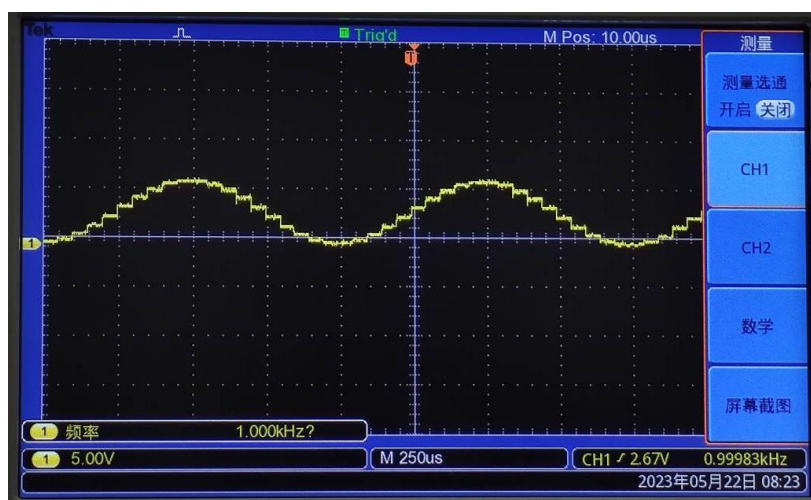


图 12: 信号源输入频率 1kHz

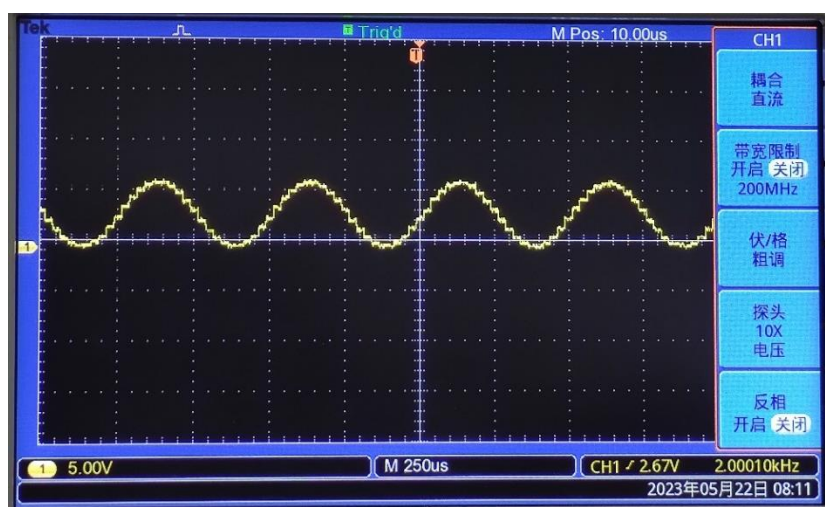


图 13: 信号源输入频率 2kHz



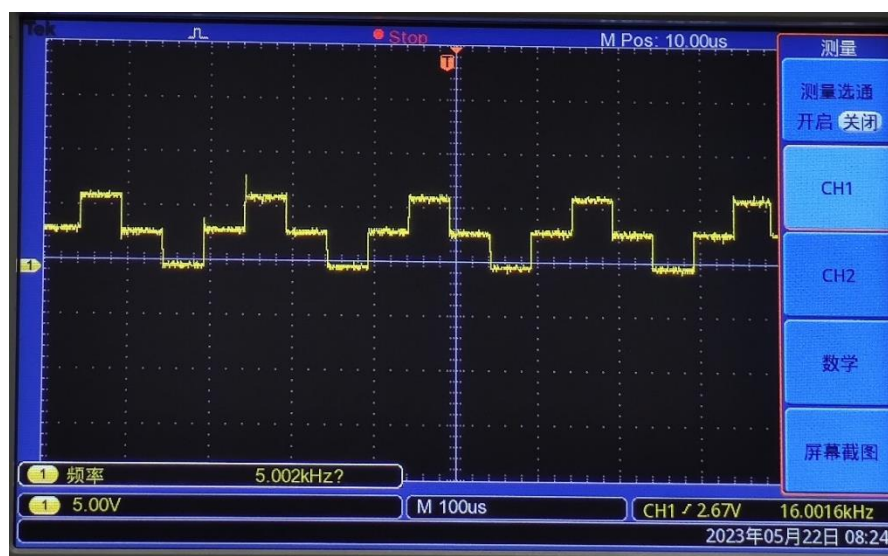


图 14：信号源输入频率 5kHz

## 9. ADC 采样频率的软件验证

修改前：在 graph 绘制的波形图中，统计一个周期内点数，将其与信号源输入频率相乘，即可求得大致的 ADC 采样频率。本次验证中，我们采用 2kHz 作为输入频率，测得波形图如图 15 所示：

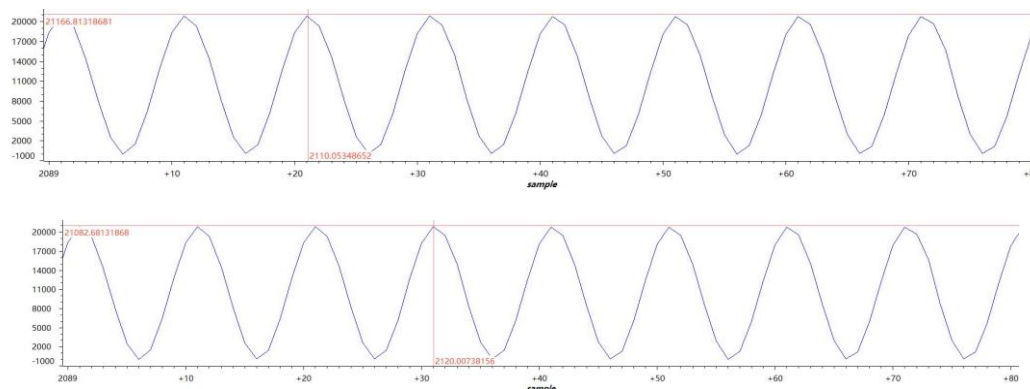


图 15：信号源输入参数设置

在图 15 的波形图中记录相邻两个最高点所在位置分别为 2110、2120，则计算所得的采样频率为  $(2120 - 2110) * 2k = 20kHz$ ，与理论值一致；

修改后：改变 TBPRD 的值，验证方法同理于上文，此时存储空间内的波形如图 16 所示：

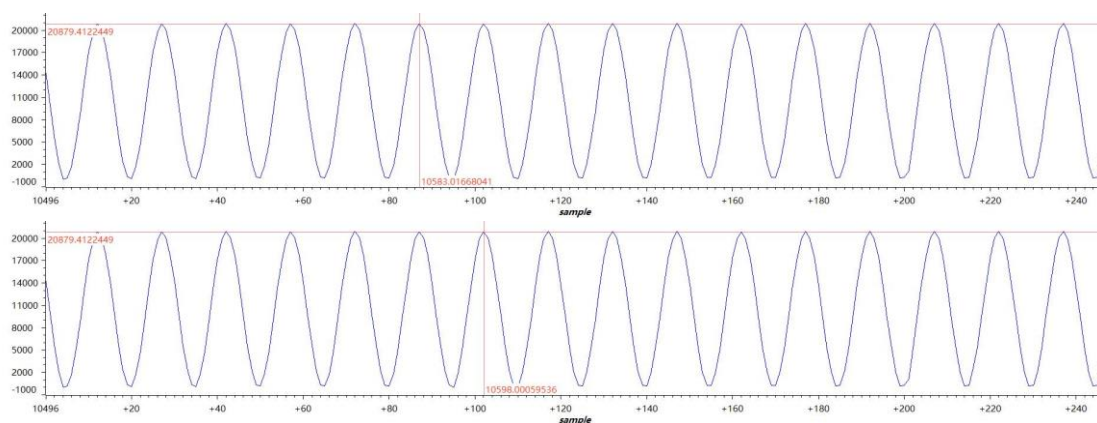


图 16: 采样频率为 30kHz 的波形

在图 16 的波形图中记录相邻两个最高点所在位置分别为 10583、10598，则计算所得的采样频率为  $(10598-10583) \times 2k=30\text{kHz}$ ，与理论值一致；

据此，软件验证部分完成，未发生明显问题。

## 10. ADC 采样频率的硬件验证

通过硬件验证，需要在每次进入中断时，使 DA 高低电平互换，如此往复，形成方波。因此，我们在中断语句段中删除存储数据模块，替换为方波程序，具体如图 17 所示：

```

353 unsigned int k=10000;
354 interrupt void epwm1_timer_adc_isr(void)    //中断函数
355 {
356
357     /*CPLD_CNTR = 3;
358     /*CPLD_LED = 3;
359
360
361 //DA
362 /* xn= (AdcRegs.ADCRESULT1 & 0xFFF0);
363 if(ConvCount<1024)
364 {
365     SampleTable1[ConvCount]=xn;
366     ConvCount++;
367 }
368 else
369     ConvCount=0;
370 *Da_out= xn ;
371 */
372 *Da_out=k;
373 k=10000-k;
374 // Reinitialize for the next ADC Sequence

```

图 17: 产生高低电平的方波程序代码

修改前：编译后运行程序，在示波器上观察到方波如图 18 所示，方波的频率是采样频率的一半，即硬件验证的采样频率为： $10.00 \times 2=20.00\text{kHz}$ ，与理论值在误差范围内一致；

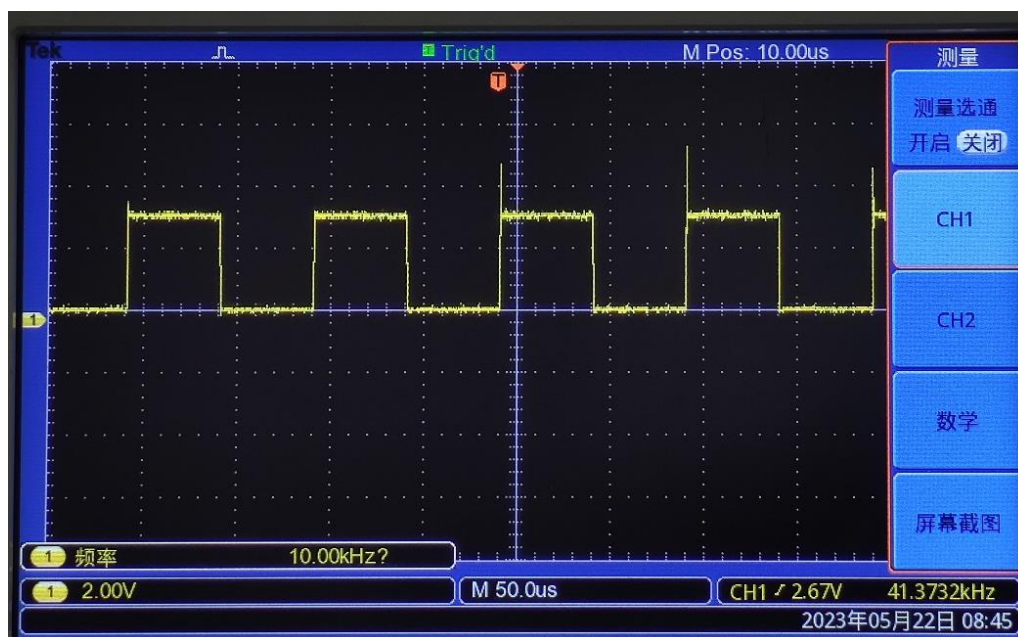


图 18: 硬件验证 20kHz 采样频率的示波器波形

修改后: 验证方法与“修改前”一致, 此时波形如图 19 所示。硬件验证的采样频率为  $15.36 \times 2 = 30.72\text{kHz} \approx 30\text{kHz}$ , 与理论值相近。

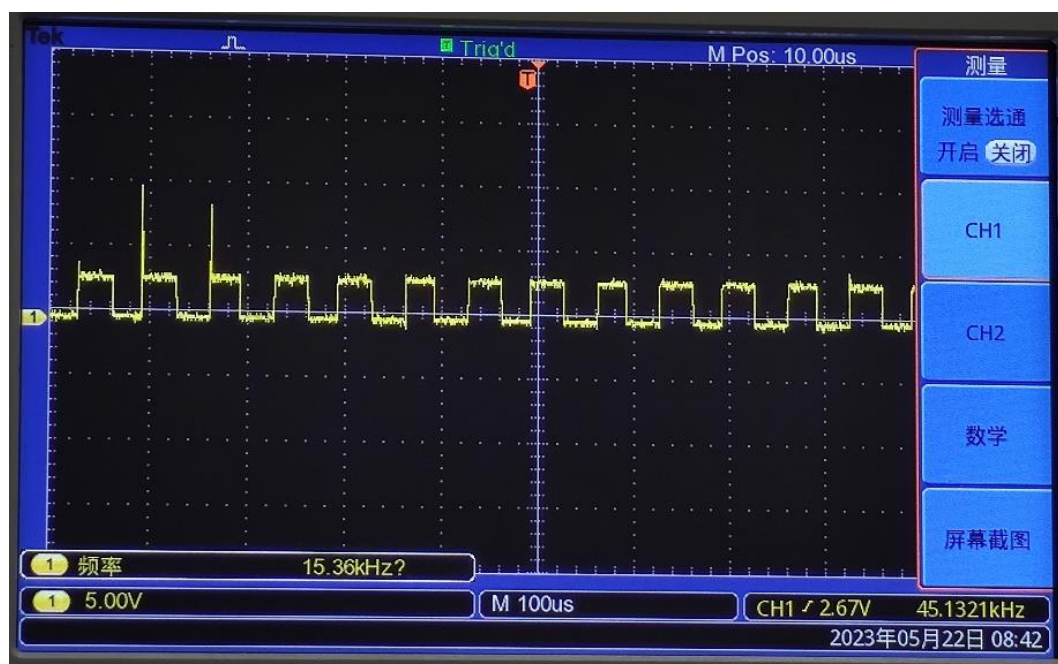


图 19: 硬件验证 30kHz 采样频率的示波器波形

## 11.5 实验思考

1. 根据范例, 写出各外设初始化的寄存器、数值及含义。

## (1) TBCTL 寄存器

字段	数值	含义
HSPCLKDIV	3	TBCLK=SYSCLKOUT/(HSPCLKDIV*CLKDIV) 共同决定 TB 模块预分频。
CLKDIV	0	HSPCLKDIV (高速基准时钟分频位) 011=/6 CLKDIV (TB 时钟预定标) 000=/1 (默认)
PRDL	TB_SHADOW (0)	周期映射装载: 0=在 CTR=0 时装载 1=立即装载
SYNCOSEL	TB_SYNC_DISABLE (3)	同步信号输出选择: 00=EPWMxSYNCl 01=CTR=0 10=CTR=CMPr 11=禁止输出 SyncOut
CTRMODE	TB_COUNT_UPDOWN (2)	计数模式: 00=增计数 01=减计数 10=增减计数 11=停止计数 (默认)
PHSEN	TB_DISABLE (0)	相位使能位: 0=禁止 1=当同步信号 EPWMxSYNCl 输入或当软件强制同步事件发生时, TBCTR 加载相位寄存器 TBPHS 的值

表 2: TBCTL 寄存器

## (2) TBPRD 寄存器

字段	数值	含义
TBPRD	208	设定时间基准器计数器的周期 设定 TB 计数器的计数最大值为 208（增减计数模式下，0→208→0）

表 3：TBPRD 寄存器

## (3) ETSEL 寄存器

字段	数值	含义
SOCAEN	1	使能 ePWMxSOCA 脉冲
SOCASEL	2	TBCTR=TBPRD 时产生 ePWMxSOCA
INTSEL	ET_CTR_PRD (2)	TBCTR=TBPRD 时产生 ePWMxSOCA 使能时基计时器等同周期
INTEN	1	使能 ePWMx_INT 产生

表 4：ETSEL 寄存器

## (4) ETPS 寄存器

字段	数值	含义
SPCAPRD	3	在第三个事件产生 ePWMxSOCA 脉冲
INTPRD	ET_3RD (3)	在第三个事件产生中断

表 5：ETPS 寄存器

2. 观察输入信号与示波器显示信号、存储器中存储波形信号幅度的差异，解释差异产生的原因。

原因可能是输入信号幅度是有正负的，而此处 AD 为 12 位无符号数，低 4 位



舍去为 0，在高位扩展的过程中出现幅值的变化；同时，当信号通过硬件时也会产生一些误差，造成波形的差异性。

### 3. 除了上述粗略验证 ADC 采样频率以外，思考其他测试采样频率的方法手段。

答：可以采用硬件验证方法。具体操作在上文

### 4. 除了中断方式，DSP 内核还可以采用查询方式获取 ADC 外设的采样数据。

若采用查询方式，则需要查询哪些标志位，给出程序流程。

答：在查询方式下工作时，是读外设的标志位。若无效则继续读标志位，有效则往下执行相应程序，执行完后继续读标志位。本实验中，由于只用到了 ADC1 通道，那么需要查询的是 EOSBUF1 和 SEQ1BSY，以分别判断 SEQ1 的序列缓冲是否结束以及 SEQ1 是否正忙。SEQ1\_BSY 的标志位如果为 0，则采集完成，可以进行下一步的程序，反之。如果为 1，则采集未完成，不能进行下一步，具体程序如下：

```
while(1){
    if(AdcRegs.ADCST.bit.SEQ1_BSY==0)
        //if(AdcRegs.==0)
        {
            xn= (AdcRegs.ADCRESULT1 & 0xFFF0);
            if(ConvCount<1024)
            {
                SampleTable1[ConvCount]=xn;
                ConvCount++;
            }
            else
            {
                ConvCount=0;
            }
            *Da_out=xn ;
        }
}
```

### 5. 如何将存储的采样数据保存到数据文件中，并利用动态有效位 ENOB 测试方法分析实验平台数据采集的性能

在观察窗口有着 export 选项，单击选择即可导出.dat 文件。

查阅资料得到：ENOB 是有效位数，对应于 AC 输入，是一项有关转换器对于交流信号的非线性性能指标，表示一个 ADC 在特定输入频率和采样率下的动态性能；

具体计算公式为： $ENOB = (SNR - 1.76) / 6.02 \text{ dB}$

可以利用以下代码计算均方根噪声（RMS Noise）和信噪比（SNR），然后使用这些值来计算 ENOB。

```
1. #define VREF 3.3 // 参考电压
2. #define LSB (VREF / pow(2, 12)) // 最低有效位
3. // 计算均方根噪声 (RMS Noise)
4. float sumSquared = 0;
5. for (int i = 0; i < MAX_SAMPLES; i++) {
6.     float voltage = sampleData[i] * LSB;
7.     sumSquared += voltage * voltage;
8. }
9. float rmsNoise = sqrt(sumSquared / MAX_SAMPLES);
10. // 计算信噪比 (SNR)
11. float snr = 20 * log10(VREF / rmsNoise);
12. // 计算 ENOB
13. float enob = (snr - 1.76) / 6.02;
14. printf("ENOB: %.2f\n", enob);
15.
```

一般来说，ENOB 做到 8dB 就比较符合工程需要了，经过计算我们采集的数据在 6dB 左右。

## 6. 体现系统的实时性

实时性是指信号处理算法所需的时间，算法时间越长，输出波形与输入波形的时间差越大，经过我们验证，当输入波形连续改变时，输出波形立即也随之变化，时间差为毫秒级别，这就体现了系统的实时性。

## 11.6 实验总结及问题

1. 连接示波器后波形不为完整正弦波，小于 0 的部分为直线。

我们发现由于存储数值为 16 位有符号数，而 graph 绘图参数设定为 16 位无符号数，造成波形的错误。于是我们在信号源参数上设置了  $500\text{mV}_{\text{DC}}$  的偏移，经验证这样输出波形完整。

2. graph 绘图工具波形在某些峰值不为标准正弦波。

由于 graph 记录的是以一段距离存储的数据波形，若运行时间太久，后采样数据就会覆盖先前数据而产生重叠现象。于是我们运行一段时间后暂停程

序，得到了较为完整的正弦波形显示。

### 3. 软件验证过程中采样频率与理论值相差较大

通过数点法估计采样频率时发现得到的采样频率为几十 kHz，不符合 20kHz。在分析代码后，我们修改了 `AdcRegs.ADCTRL1.bit, CONT_RUN` 语句值为 0。该语句使 ADC 工作在连续转换模式。当接收到 EOS 信号后，排序器的动作依赖于 `SEQ-OVRD`，如果值为 0，则排序器回到起始状态 `CONV00`；如果值为 1，排序器不再复位。

## 11.7 实验体会

本次实验，对整个 AD 的工作流程和采样频率计算有更深一步的理解，并且也自己通过修改寄存器的数值改变采样频率；虽然实验中对系统实时性的验证并未体现。实验过程中，在示波器输出和 graph 绘制波形的部分拖延了大量时间，具体问题体现在 11.6 中；

在尝试查询方式的过程中，由于第一次使用这种方法，很不熟练，对于结构体的操作也不太熟练，但在摸索过程中对寄存器的各个标志位有了更深的理解，同时也对如何引用标志位进行状态判断有了更深的认识。

在 ADC 采样频率硬件验证过程中，我们利用每次中断进行过程中交替赋予 AD 高、低电平的转换，则示波器上测试的方波频率为采样频率的一半这一方法；实际上若设定程序，在中断服务程序开始时输出高电平，结束时输出低电平，则经过示波器输出显示后，高电平部分为 AD 采样时间，输出信号的周期则为采样时间间隔。整个数据采集的程序设定只需要在数据存储后加数组保存依次输出，为后续 FIR 滤波器的设计奠定基础。