



南京理工大学

NANJING UNIVERSITY OF SCIENCE & TECHNOLOGY

# DSP 应用技术实验

## DSP 数据采集实验报告

作 者 : 周鹏 学 号 : 9181040G0740

同 组 人 : 杨霄宇 学 号 : 9181040G0736

同 组 人 : 许昕荣 学 号 : 9181040G1038

学 院 : 电子工程与光电技术学院

专 业 : 电子信息工程

班 级 : 电信 3 班

组 号 : 第二组 B6

题 目 : DSP 应用技术实验

DSP 数据采集实验报告

指 导 者 : 李彧晟

2021 年 04 月

# 目录

|                                     |    |
|-------------------------------------|----|
| 1 实验目的.....                         | 1  |
| 2 实验仪器.....                         | 1  |
| 2.1 实验仪器清单.....                     | 1  |
| 2.2 硬件连接示意图.....                    | 1  |
| 3 实验步骤及现象.....                      | 2  |
| 3.1 程序流程.....                       | 2  |
| 3.2 实验步骤.....                       | 3  |
| 3.3 编写数据存储代码.....                   | 4  |
| 3.3.1 数据存储的原理.....                  | 4  |
| 3.3.2 数据存储代码.....                   | 5  |
| 3.4 建立工程并运行、调试程序.....               | 5  |
| 3.5 修改采样频率并验证.....                  | 5  |
| 4 实验结果.....                         | 5  |
| 4.1 ADC 采样频率计算公式.....               | 5  |
| 4.2 信号波形存储地址及作图.....                | 6  |
| 4.3 观察不同频率的输出情况.....                | 6  |
| 4.4 ADC 采样频率的验证.....                | 7  |
| 4.5 ADC 采样频率的硬件验证.....              | 8  |
| 4.5.1 修改前的采样频率的验证.....              | 8  |
| 4.5.2 修改后的采样频率的验证.....              | 8  |
| 4.6 利用查询方式对 ADC 外设进行数据采样.....       | 9  |
| 4.7 将存储的采样数据保存到数据文件中.....           | 9  |
| 4.8 利用动态有效位 ENOB 进行测试数据平台的采集性能..... | 9  |
| 5 实验总结.....                         | 9  |
| 5.1 实验中遇到的问题及解决方法.....              | 9  |
| 5.2 实验心得体会.....                     | 10 |

## 1 实验目的

1. 熟悉 DSP 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 的中断的设置；
4. 掌握 TMS320F28335 的 ADC 模块的基本使用方法；
5. 熟悉 DSP 代码调试基本方法。

## 2 实验仪器

### 2.1 实验仪器清单

- |                              |    |
|------------------------------|----|
| 1. DSP 仿真平台（仿真器、DSP 实验箱、计算机） | 一套 |
| 2. 示波器                       | 一台 |
| 3. 信号发生器                     | 一台 |

### 2.2 硬件连接示意图

实验硬件连接大致如图 2.1 所示，F28335 的 ADC 原理如图 2.2 所示。

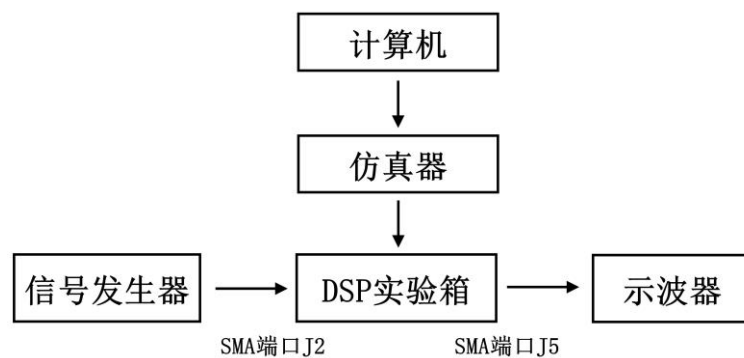


图 2.1 硬件连接示意图

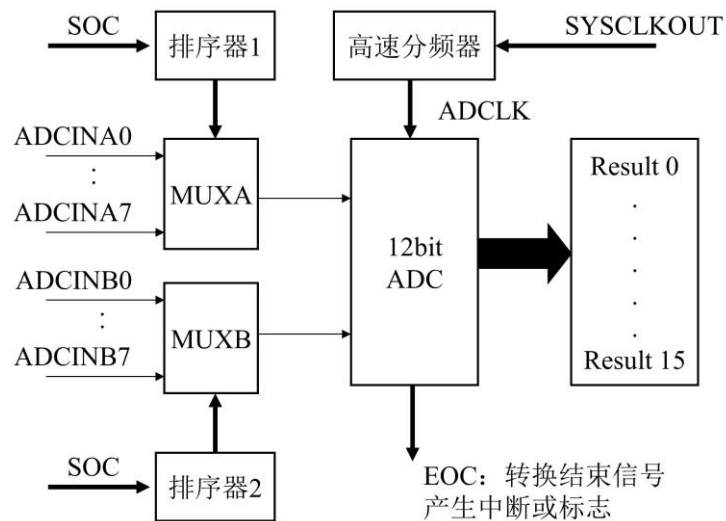


图 2.2 F28335 的 ADC 原理

### 3 实验步骤及现象

#### 3.1 程序流程

为实现 DSP 的数据采集存储以及模拟的还原，必须依赖于 ADC、DSP 以及 DAC 三大基本部件，而 TMS320F28335 芯片上集成了 ADC 模块，因此实现该功能较为简单，数据采集的工作可以由 DSP 单独完成，只需要对相关外设模块进行合理配置。模拟还原由实验箱中 DAC（AD9747）来完成。TMS320F28335 中的 ADC 模块与 DSP 内核之间的通信可以通过查询方式或中断方式，在此，我们采用 ADC 的中断功能实现数据的交换。

TMS320F28335 中 ADC 的转换频率和采样频率可以独立设置，分别位于 ADC 模块和 ePWM 的时间基准子模块中，因此要使 ADC 工作，必须掌握 ADC 模块和 ePWM 模块中的相关设置。

由此可得程序流程如图 3.1 所示。

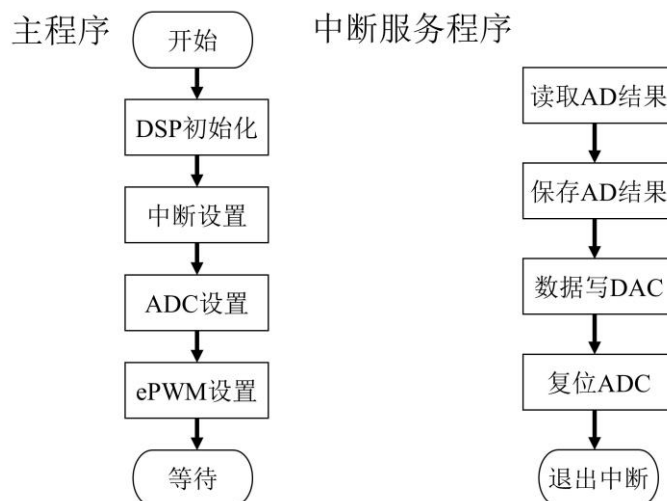


图 3.1 DSP 数据采集程序流程图

#### DSP 初始化

一般而言，DSP 要正常工作，必须首先设置时钟，时钟确定了 DSP 工作主频。

TMS320F28335 中时钟设置大致分为三个主要寄存器，它们分别是锁相环控制寄存器(PLLCCR)、外设时钟使能控制寄存(PCLKCR0, PCLKCR1, PCLKCR2)和外设时钟预定标设置寄存器(HISPCP、LOSPCP)。F28335 的时钟设置具体内容 参照课件 2.4 等相关资料。

### 模数转换模块(ADC)

TMS320F28335 内部有一个 16 通道、采样精度为 12bit 的 ADC 模块。这 16 通道可配置两个独立的 8 通道模块，具有同步采样和顺序采样模式，模拟输入范围 0~3V，最快转换时间为 80ns，具有多个触发源用于启动 AD 的转换，采用灵活的中断控制。

### ePWM 模块

TMS320F28335 中 ePWM 模块的事件可产生 ADC 转换启动脉冲信号 SOC，本次实验采用时间基准子模块的产生周期事件，通过事件触发子模块的设置来产生 ADC 转换启动脉冲信号 SOC。

### TMS320F28335 中断系统

TMS320F283x 的外设中断扩展(PIE)单元通过少量中断输入信号的复用来扩展大量的中断源，PIE 单元支持多达 96 个独立的中断，这些中断以 8 个为一组进行分类，每组中的所有中断共用一个 CPU 级中断(INT1~INT12)。96 个中断对应的中断向量表存储在专用 RAM 区域中。PIE 向量表用来存储系统中每个中断服务程序(ISR)的入口地址。一般来说，在设备初始化时就要设置 PIE 向量表，并可在程序执行期间根据需要对其进行更新。

在实验中，当我们设置 VMAP=1(ST1 寄存器的 bit3)，ENPIE=1(PIECTRL 寄存器的 bit0)后，TMS320F28335 的中断向量表地址范围 0x000D00~0x000DFF。例如 ADC 外设模块 SEQ1INT 中断向量地址是 0x000D40，SEQ2IN 中断向量地址是 0x000D42，ADCINT 中断向量地址为 0x000D4A(ADCINT 是 SEQ1INT 和 SEQ2INT 的逻辑或)。

要想正确使用中断，首先应该合理设置中断向量表，在对应地址填入中断服务子程序的入口地址。其次，必须对上述三个级别的中断作出正确的设置。比如实验中，要想实现 CPU 利用中断方式读取 ADC 的采样数据，可以使能 ADC 模块的中断 SEQ1INT，其次使能外设使能寄存器 PIEIER1.1，保证中断发生时 PIEACK1.1 位清零，最后使能 CPU 中断使能寄存器 IER 中的 INT1，以及全局中断使能位 INTM。这些工作必须在系统初始化时完成。退出中断服务程序前，清除 ADCST 中的 INT SEQ1 以及相应的 PIEACK<sub>x</sub>。

## 3.2 实验步骤

### 1. 设备检查

检查仿真器、F28335 DSP 教学实验箱、计算机之间的连接是否正确，打开计算机和实验箱电源。

### 2. 启动集成开发环境

点击桌面 CCS 5 的快捷方式，进入集成开发环境 CCS。

### 3. 新建工程

新建一个 DSP 工程，编辑源程序、配置命令等相关文件，并在工程中添加这些程序文件。

在源程序中，通过对中断、ADC 外设以及事件管理通用时钟的设置，利用中断方式读取 ADC 的采样结果，并用 DAC 实现模拟信号的还原。在程序中，开辟一段数据空间，用于保存 ADC 的采样结果，要求保存 1024 点数据，且该空间的数据不断刷新。

### 4. 建立工程 (Build)

建立工程 (build)，若出错，则根据错误提示，修改源程序文件或者配置命令文件，直至编译链接正确，生成可执行的 .out 文件。

### 5. 连接外部电路

打开信号源，产生一个合适的频率（ADC 的采样频率必须满足奈奎斯特采样定律），信号幅度控制在 0-3V 以内，验证后将信号通过接口输入到电路板上 SMA 端口 AD\_CHA（SMA 端口 AD\_CHA 对应 DSP 芯片 ADCINA1 引脚，SMA 端口 AD\_CHB 对应 DSP 芯片 ADCINB1 引脚）。打开示波器，将实验箱中的 SMA 接口 J5 输出到示波器上，并正确设置。

### 6. 调试程序

在工程中合理配置 ccxml 文件，打开实验箱电源，在主菜单下选择“Run → Debug”，若仿真器正确连接后，进入“CCS Debug”调试界面。

首先验证中断设置是否正确。可以在 ADC 中断服务程序的入口地址处添加断点，全速或者动画运行程序，检查程序计数器 PC 能否间隔性的停留在中断服务入口地址处。若能，说明中断设置基本正确。若以上步骤正确，其次，验证数据采集的正确性。程序连续运行一段时间后，暂停程序执行，打开图形显示功能，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。若上述步骤正确，则调节示波器，观察信号波形，是否为信号源的输入波形。若是，则实验调试结束。以上步骤如果出错，则可以利用各种调试手段，比如打开寄存器窗口、变量窗口等辅助手段，根据数值以及实验原理，查找错误原因，重新修改程序，直至正确为止。

### 7. 运行程序

若第 6 步正确，可去掉断点，重新全速运行程序。

连接实验箱 SMA 输出口 J5 至示波器，调节示波器，观察信号的输出。可以实时的改变信号源的输入信号（注意信号幅度不要随意修改，超出输入范围易烧毁实验电路），示波器上显示的波形亦会随之变化。

数据直通通道就是最简单的实时信号处理电路。

## 3.3 编写数据存储代码

### 3.3.1 数据存储的原理

中断服务程序触发的条件是 ADC 采样信号到来，原本的范例程序中直接将这个信号输送给 DA，实现数据实时输入输出。

将这个采集到的信号同时存入 SampleTable1 数据空间，可以实现数据保存。但每次中断到来，只能保存一个数据。因此需要设定一个数据存放位置指示变量 ConvCount，每次存放数据后，位置加一。当指示变量超过 SampleTable1 数据空

间的长度（1024 个值）后，将 ConvCount 归零，以实现 SampleTable1 数据空间中数据的不断更新。

### 3.3.2 数据存储代码

综合以上内容，线性调频信号查找表的产生代码如下：

```
347 interrupt void epwm1_timer_adc_isr(void) //中断函数
348 {}
349 //DA
350 xn= (AdcRegs.ADCRESULT1 & 0xFFFF0);
351 if(ConvCount<1024)
352 {
353     SampleTable1[ConvCount]=xn;
354     ConvCount++;
355 }
356 else
357 {
358     ConvCount=0;
359 }
360 *Da_out= xn ;
```

## 3.4 建立工程并运行、调试程序

连接信号发生器至教学实验箱 SMA 输入端口 J2、教学实验箱 SMA 输出端口 J5 至示波器，编译链接工程并进入调试调试界面，运行程序后，查看存储空间中时域波形。

## 3.5 修改采样频率并验证

阅读程序，发现主程序中的如下代码：

```
295 // Set Period for EPWM1
296 EPwm1Regs.TBPRD = 208; //设定时间基准器计数器的周期 208-fs 20kHz,139-f
297 EPwm1Regs.TBCTL.bit.CTRMODE = TB_COUNT_UPDOWN; //增减计数模式
298 // Setup Compare A = 2 TBCLK counts
299 EPwm1Regs.CMPA.half.CMPA = 2; //计数比较寄存器A CMPA 当前工作的CMPA的值不断
300 // Phase is 0 for Synchronization Event
301 EPwm1Regs.TBPHS.half.TBPHS = 0x0000; //TBCTR不装载相位寄存器TBPHS的值
302 // Clear TB counter
303 EPwm1Regs.TBCTR = 0x0000; //事件基准计数寄存器TBCTR 读取写到其中的TBCTR的值
```

可知修改 TBPRD 的值可以实现修改采样频率，修改值为 139，则理论采样频率变为 30kHz，通过软件和硬件的方法分别验证采样频率是否正确。

## 4 实验结果

### 4.1 ADC 采样频率计算公式

根据范例，发现 TB 计数模式为增减计数，故 ADC 采样频率公式如下：

$$T_{PWM1} = \frac{TBCLK}{TBPRD * 2 * 3} = \frac{25}{139 * 2 * 3} = 30KHz$$

程序如下图 4.1 所示：

```
// Set Period for EPWM1
EPwm1Regs.TBPRD = 139; //设定时间基准器计数器的周期 21
EPwm1Regs.TBCTL.bit.CTRMODE = TB_COUNT_UPDOWN; //增减计数模式
// Setup Compare A = 2 TBCLK counts
EPwm1Regs.CMPA.half.CMPA = 2; //计数比较寄存器A CMPA 当前工
```

图 4.1 采样频率程序

## 4.2 信号波形存储地址及作图

指出信号波形的存储地址，并作图显示。

如图 4.2 所示，可得到波形存储地址为 0x0000C040，利用 graph 工具绘图得到图 4.3。

| Expression         | Type               | Value           | Address         |
|--------------------|--------------------|-----------------|-----------------|
| Da_out             | unsigned int *     | 0x00200400      | 0x0000C004@Data |
| SampleTable1       | unsigned int[1024] | 0x0000C040@Data | 0x0000C040@Data |
| Add new expression |                    |                 |                 |

图 4.2 波形存储地址

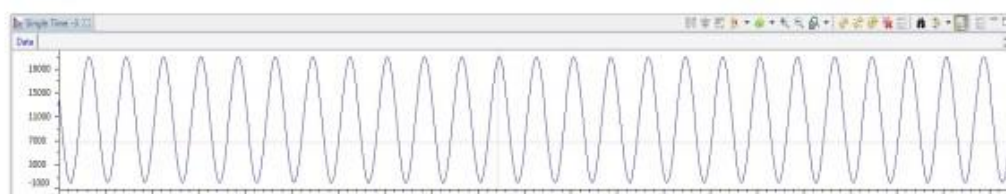


图 4.3 采样频率为 30kHz 时的波形

## 4.3 观察不同频率的输出情况

改变信号源的频率，观察示波器上输出。修改信号源频率，示波器输出结果如图 4.4 到图 4.7 所示。

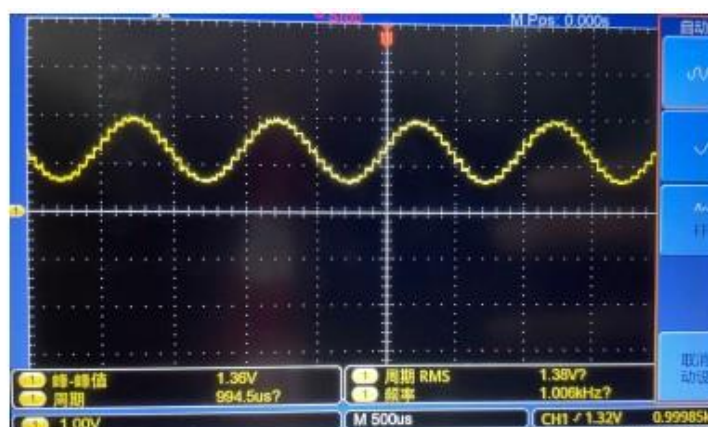


图 4.4 信号源频率 1kHz

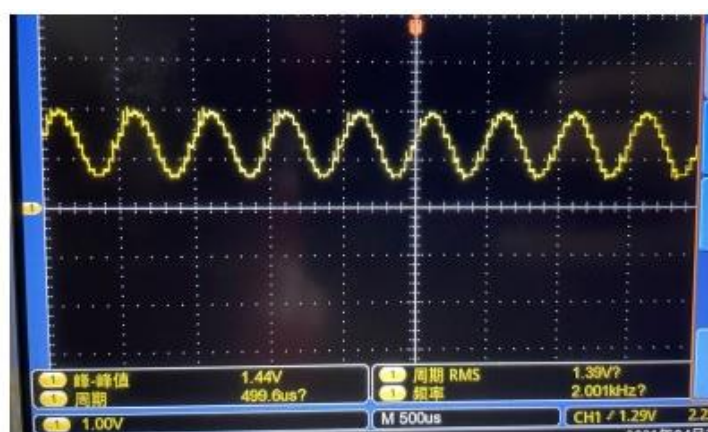


图 4.5 信号源频率 2kHz



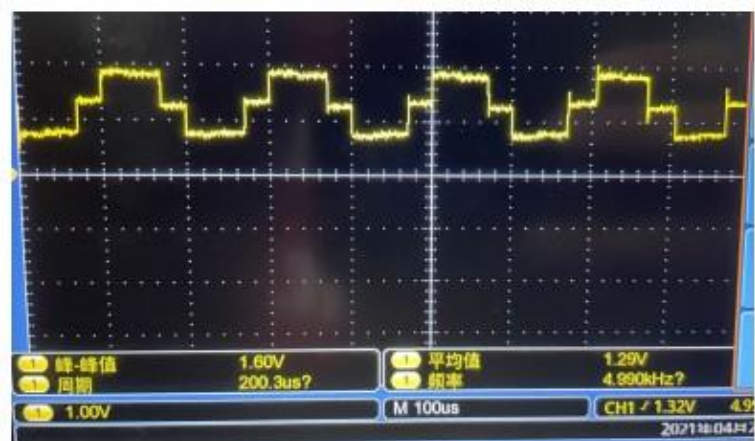


图 4.6 信号源频率 5kHz

## 4.4 ADC 采样频率的验证

### 4.4.1 修改前的采样频率的验证

在 graph 绘制的波形图中,统计一个周期内的点数,与信号源输入频率相乘,即可得到大致的 ADC 采样频率。

修改 ADC 采样频率前的波形图如图 4.3 所示,此时记录两个最高点所在位置为 537、568,输入频率为 1000Hz (如图 4.7 所示),则计算所得的采样频率为  $(568-537)*1000=31000\text{Hz}$ ,与 30kHz 的理论值接近。

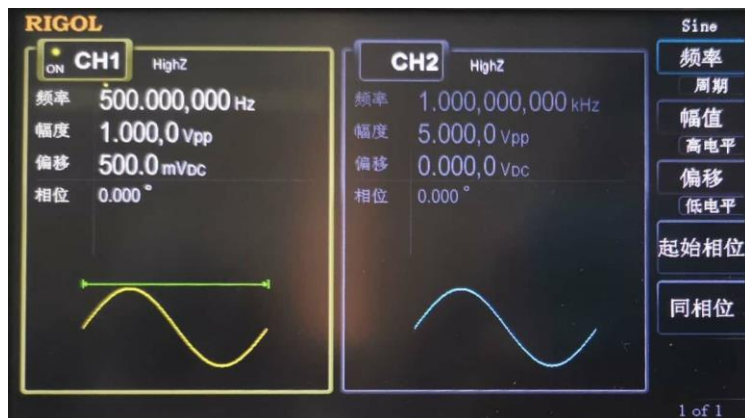


图 4.7 信号源输入频率

### 4.4.2 修改后的采样频率的验证

验证方法与修改前一致,此时存储空间内的波形如图 4. 所示,记录两个最高点的所在位置为 560、589,输入频率同样为 1000Hz,则计算所得的采样频率为  $(589-560)*1000=29000\text{Hz}$ ,与理论值 30kHz 接近。

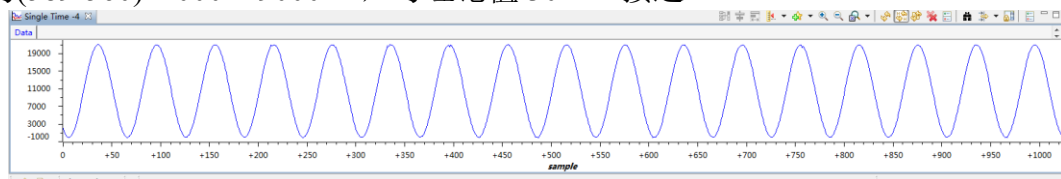


图 4.8 采样频率为 30kHz 时的波形

## 4.5 ADC 采样频率的硬件验证

### 4.5.1 修改前的采样频率的验证

在每次进入中断时，使 DA 高电平、下次输出低电平，如此往复，对应的代码如下：

```
346 unsigned int property=10000;
347 interrupt void epwml_timer_adc_isr(void)    //中断函数
348 {
349     //DA
350     /* xn= (AdcRegs.ADCRESULT1 & 0xFFF0);
351     if(ConvCount<1024)
352     {
353         SampleTable1[ConvCount]=xn;
354         ConvCount++;
355     }
356     else
357     {
358         ConvCount=0;
359     }
360     *Da_out= xn ;
361     */
362     *Da_out=property;
363     property=10000-property;
```

此时，可在示波器上观察到方波如图 4. 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  $10.00*2=20\text{kHz}$ ，与理论值一致。

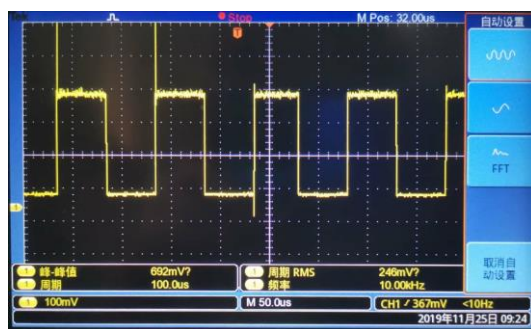


图 4.9 硬件验证 20kHz 采样频率时的示波器波形

### 4.5.2 修改后的采样频率的验证

验证方法与修改前一致，此时示波器波形如图 4.1 所示，则硬件验证的采样频率为  $15.37*2=30.74\text{kHz}$ ，与理论值接近。

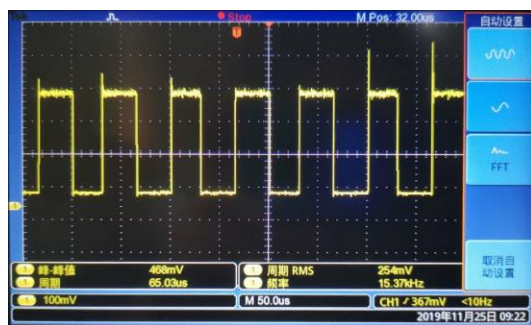


图 4.1 硬件验证 30kHz 采样频率时的示波器波形

## 4.6 利用查询方式对 ADC 外设进行数据采样

我们通过对采样程序的修改，完成了从中断方式采样到查询方式的转变，我们利用 SEQ1\_BSY 的标志位，通过这个标志位对采集状态进行判断，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 ;
        }
}
```

## 4.7 将存储的采样数据保存到数据文件中

在观察窗口有着 export 选项，单击选择即可导出.dat 文件。

## 4.8 利用动态有效位 ENOB 进行测试数据平台的采集性能

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： $ENOB = (SNR - 1.76)/6.02 \text{ dB}$

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。

# 5 实验总结

## 5.1 实验中遇到的问题及解决方法

### 1.存储空间中的数值没有变化

在第一次编写代码时，运行后发现 SampleTable1 对应的存储空间中的数值

没有发生实时的变化。查看赋值语句的修改代码后发现，是没有处理好位置指示变量 `ConvCount`。重新编写代码，当 `ConvCount` 大于 1023 后对其赋 0，重新编译链接，运行后数据可以实现实时变化。

## 2. 软件、硬件验证得到的采样频率与理论值误差极大

在第三次实验开始验证采样频率时，发现无论用硬件、还是软件验证，得到的采样频率均为 50kHz 左右，与理论值误差极大。后来，在第四次实验课开始，老师提示程序代码有误，修改 `AdcRegs.ADCTRL1.bit.CONT_RUN=1;` 语句的值为 0 后，验证的采样频率与理论值较符合。

## 3. 利用查询方式对 ADC 外设进行数据采样时，结构体不知道如何引用

根据实验讲义中的思考题目，我们想通过修改中断方式，改为查询方式对 ADC 采集的方式进行更改，在查阅 CSDN 中的相关资料，我们发现 `SEQ1_BSY` 这一标志位可以用来判断是否采集完成，从而，我们将中断中的相关代码进行赋值，加入到 `mian` 函数中，进行查询，由此，我们就只剩下一个 `if` 判断语句，由于我们不知道如何利用结构的引用对 `ADC.h` 头文件中的 `SEQ1_BSY` 这一标志位进行引用，我们首先根据其他类似语句的形式进行修改，尝试添加了引用链接，但是发现编译无法成功，之后我们询问了李老师，得到了正确的语句：`AdcRegs.ADCST.BIT.SEQ1_BSY==0`，在正确的语句下，我们重新修改得到了正确的答案。

## 5.2 实验心得体会

这次实验的主要目的在于数据采集，前半段实验都进行的很顺利，但是在验证采样频率时遇到了一点波折。由于平时都是测信号的频率，这次突然让验证采样频率，一时半会有点不知从何下手。但是和队友讨论最后发现其实验证的方法也很多，比如统计一个周期内的点数，与信号源输入频率相乘，又比如在进入中断时使 `Data_out` 高电平，结束中断时使 `Data_out` 低电平，测方波频率，即时采样频率的一半。

最后我们还考虑了从中断方式到查询方式的转变，但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。

最后很感谢李老师对我们每次实验的帮助，我们提出的各种方法，无论不可行，老师都会一点点向我们分析，让我们对 DSP 应用技术这门课有了更深的理解，更好地掌握了所学的内容。