哈爾濱工業大學（深圳）
HARBIN INSTITUTE OF TECHNOLOGY

实验报告

课程名称：DSP 的原理与应用

实验名称：实验 3.1 指示灯实验

专业-班级：电气 1 班 学号：220330124 姓名：舒晟超

实验日期：2024 年 5 月 13 日

试验台号：

报告总分数：

教师评语：

助教签字：_____

教师签字：_____

日　　期：_____

一、实验目的

1．了解 ICETEK-F28335-AF 评估板在 TMS320F28335DSP 外部扩展存储空间上的扩展。
2．了解 ICETEK-F28335-AF 评估板上指示灯扩展原理。
3．学习在 C 语言中使用扩展的控制寄存器的方法。

二、实验过程

1．导入工程并导入硬件仿真文件。

2. 打开程序并推测实验现象；

3. 下载并运行程序，观察实验现象。

三、实验结果

1．导入工程并导入硬件仿真文件。

2. 阅读代码：

```
1.  #define LED (*(unsigned short int *)0x180000)
```

这段代码宏定义了板上指示灯控制寄存器的基地址 0x180000。对该基地址写入值（即对寄存器写入值）可以控制 D5（最高位）到 D2（最低位）的亮灭。

```
1.  for (;;)
2.  {
3.    for (i = 0; i <= 0xf; i++)
4.    {
5.      LED = i;
6.      Delay(612);
7.    }
8.    for (i = 0xf; i >= 0; i--)
9.    {
10.     LED = i;
11.     Delay(612);
12.   }
13. }
```

上述代码为主循环程序，首先向寄存器每隔 612 个间隔进行值的递增，此时 D5 到 D2 将通过亮灭指示当前 LED 寄存器内的二进制值（亮为 0，灭为 1，因此开始时为全亮）；当 LED 寄存器值到达 0xf（16）时，进行递减，以此进行循环。

3.编译下载并运行程序，观察现象如图：

拍摄不同时刻的 LED 状态，可以看到不同的 LED 状态，表示 LED 正在闪烁。实际上，四个 LED 灯闪烁的周期各不相同，高位的周期约为低位周期的两倍。

LED 闪烁现象的视频在附件中给出。

四、问题与思考

1、请粗略计算 Delay(612)延时函数的延时时间，写出计算过程。（1 分）

```
37 void Delay(unsigned int nDelay)
38 {
39     int i,j,k=0;
40     for(i=0;i<nDelay;i++)
41     {
42         for(j=0;j<1024;j++)
43         {
44             k++;
45         }
46     }
47 }
```

上述代码如果需要精确计算延时时间，首先需要查询在对应优化下执行一次机器指令需要的时间，再乘以1024*612 即可得到 Delay(612) 的延时时间。

实际应用中，通过观察 LED 的循环时间进行粗略推算：

观察蓝色 LED 的闪烁周期（为 16 个 Delay(612)），在 7s 时间内，蓝色 LED 闪烁约为 5 次，闪烁周期为 1.4s，故 Delay(612)约为 87.5ms。

哈爾濱工業大學（深圳）
HARBIN INSTITUTE OF TECHNOLOGY

实验报告

课程名称：DSP 的原理与应用

实验名称：实验 3.2 拨码开关控制实验

专业-班级：电气 1 班 学号：220330124 姓名：舒晟超

实验日期：2024 年 5 月 13 日

试验台号：

报告总分数：

教师评语：

助教签字：_____

教师签字：_____

日　　期：_____

一、实验目的

1．了解 ICETEK-F28335-AF 评估板在 TMS320F28335DSP 外部扩展存储空间上的扩展。
2．了解 ICETEK-F28335-AF 评估板上拨码开关扩展原理。
3．熟悉在 C 语言中使用扩展的控制寄存器的方法。

二、实验过程

1．导入工程并导入硬件仿真文件。

2. 打开程序并推测实验现象；

3. 下载并运行程序，观察实验现象。

三、实验结果

1．导入工程并导入硬件仿真文件。

2. 阅读代码：

```
1. #define LED (*(unsigned short int *)0x180000)
```

这段代码宏定义了板上指示灯控制寄存器的基地址 0x180000。对该基地址写入值（即对寄存器写入值）可以控制 D5（最高位）到 D2（最低位）的亮灭。

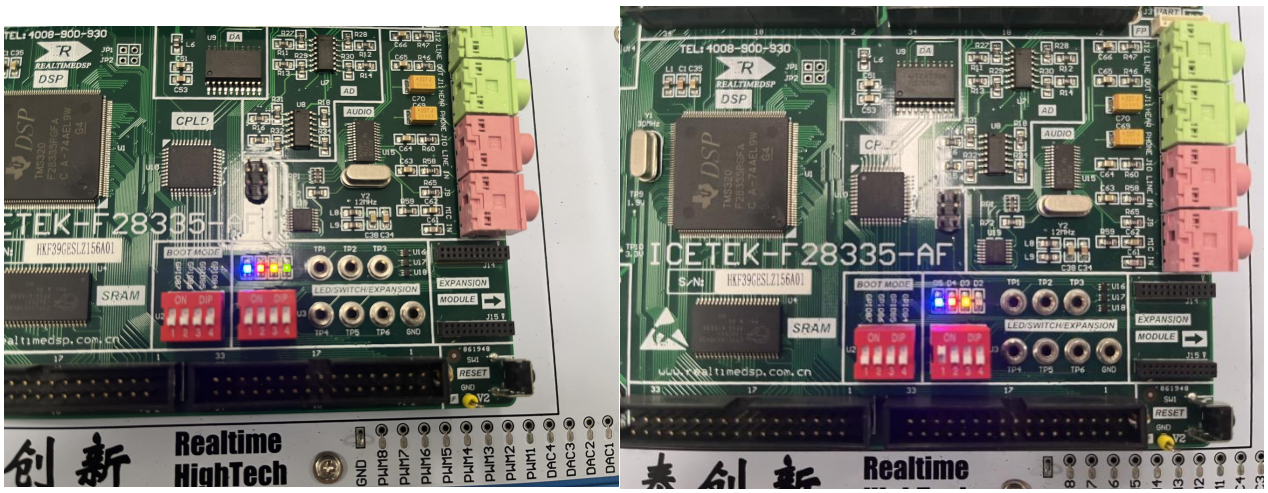
```
1.  #define DIP (*(unsigned short int *)0x180001)
```

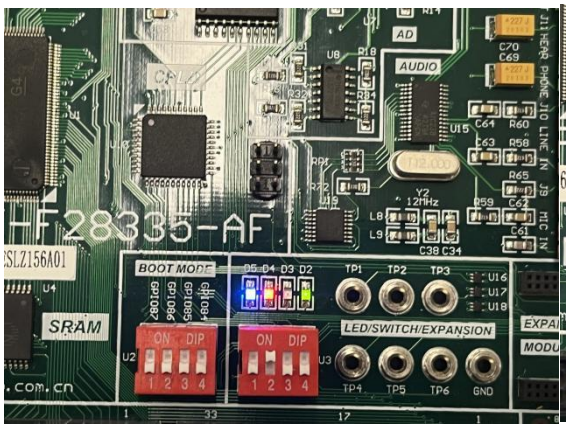
这段代码宏定义了板上 DIP 开关控制寄存器的基地址 0x180001。读取该地址的值可以获得四位拨码开关的状态（0 为闭合，1 为断开），（1 为低位，4 为高位）。

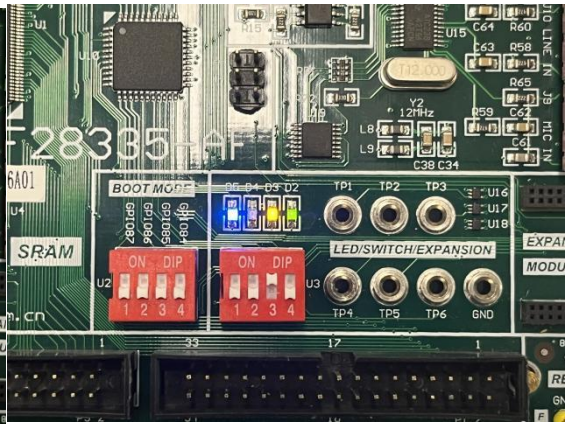
```
1.      while (1)
2.      {
3.          LED=DIP;
4.      }
```

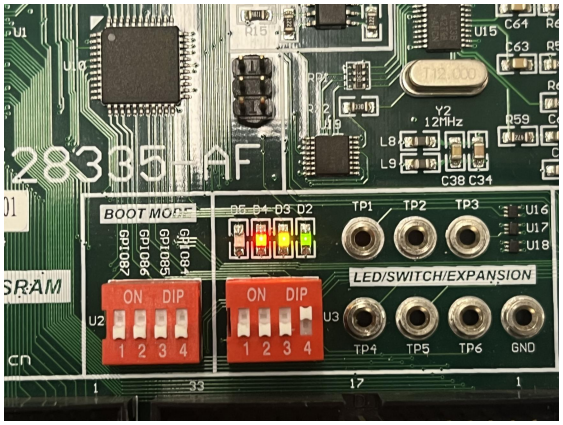
主程序中 LED 寄存器会实时更新为 DIP 寄存器的值，因此开关状态能够实时反映到 LED 上（开关闭合->DIP 对应位为 0->LED 寄存器对应位为 0->对应 LED 亮）

3.编译下载并运行程序，观察现象如图：





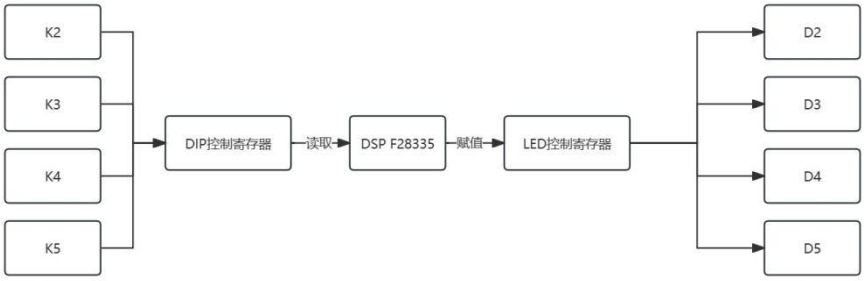




四、问题与思考

1、 请描述拨码开关为什么可以控制 LED 指示灯？其中的逻辑是什么？（1 分）

如下图所示，LED 寄存器会实时更新为 DIP 寄存器的值，因此开关状态能够实时反映到 LED 上（开关闭合->DIP 对应位为 0->LED 寄存器对应位为 0->对应 LED 亮）



哈爾濱工業大學（深圳）
HARBIN INSTITUTE OF TECHNOLOGY

实验报告

课程名称：DSP 的原理与应用

实验名称：实验 3.3 DSP 的定时器

专业-班级：电气 1 班　学号：220330124　姓名：舒晟超

实验日期：2024 年 5 月 13 日

试验台号：

报告总分数：

教师评语：

助教签字：_____

教师签字：_____

日　　期：_____

一、实验目的

1．通过实验熟悉 F28335 的定时器；
2．掌握 F28335 定时器的控制方法；
3．掌握 F28335 的中断结构和对中断的处理流程；
4．学会 C 语言中断程序设计，以及运用中断程序控制程序流程。

二、实验过程

1．导入工程并导入硬件仿真文件。

2. 打开程序并推测实验现象；

3. 下载并运行程序，观察实验现象。

三、实验结果

1．导入工程并导入硬件仿真文件。

2. 阅读代码：

```
1.  void main(void)
2.  {
3.      // DSP28335 初始化
4.      InitSysCtrl();
5.
6.      // GPIO(外部扩展模块)初始化
7.      InitXintf16Gpio();
8.
9.      // 禁止中断
10.     DINT;
11.
12.     // 初始化 PIE 中断控制器
13.     InitPieCtrl();
14.
15.     // 禁止 CPU 中断并复位 CPU 中断标志位
16.     IER = 0x0000;
17.     IFR = 0x0000;
18.
19.     // 初始化中断向量表
20.     InitPieVectTable();
21.
22.     // 设置定时器 0 的中断入口地址为中断向量表的 TINT0
23.     EALLOW;
24.     PieVectTable.TINT0 = &cpu_timer0_isr;
25.     EDIS;
26.
27.     // 初始化定时器
28.     InitCpuTimers();
29.
30.     // 根据不同 CPU 系统频率设定定时器中断周期
31. #if (CPU_FRQ_150MHZ)
```

```
// Configure CPU-Timer 0, 1, and 2 to interrupt every second:
// 150MHz CPU Freq, 1 second Period (in uSeconds)

   ConfigCpuTimer(&CpuTimer0, 150, 1000000);
   //ConfigCpuTimer(&CpuTimer1, 150, 1000000);
   //ConfigCpuTimer(&CpuTimer2, 150, 1000000);
#endif

#if (CPU_FRQ_100MHZ)
// Configure CPU-Timer 0, 1, and 2 to interrupt every second:
// 100MHz CPU Freq, 1 second Period (in uSeconds)

   ConfigCpuTimer(&CpuTimer0, 100, 1000000);
   //ConfigCpuTimer(&CpuTimer1, 100, 1000000);
   //ConfigCpuTimer(&CpuTimer2, 100, 1000000);
#endif

        // 初始化定时器0
        // 设置定时器的周期寄存器值为65535.
 CpuTimer0Regs.PRD.all=0xffff;
      // 设置定时器的计数寄存器值为0
 CpuTimer0Regs.TIM.all=0;
        // 设置定时器预定标计数器值为0
 CpuTimer0Regs.TPR.all=0;
 CpuTimer0Regs.TPRH.all=0;
        // 确保定时器 0 为停止状态
 CpuTimer0Regs.TCR.bit.TSS=1;
        //  定时器0设置为自由运行（仿真模式设置）
 CpuTimer0Regs.TCR.bit.SOFT=1;
 CpuTimer0Regs.TCR.bit.FREE=1;
        // 使能定时器0的重载
 CpuTimer0Regs.TCR.bit.TRB=1;
        // 使能定时器0的中断
 CpuTimer0Regs.TCR.bit.TIE=1;
        // 清零定时器0的中断计数值
 CpuTimer0.InterruptCount=0;

        // TSS 置1，开启定时器0
 startCpuTimer0();

   // 开启 CPU 第一组中断并使能第一组中断的第 7 个小中断，即定时器 0
   IER |= M_INT1;
   PieCtrlRegs.PIEIER1.bit.INTx7 = 1;

   // 使能总中断
   EINT;   // Enable Global interrupt INTM
   ERTM;   // Enable Global realtime interrupt DBGM

```

```
    for(;;)
    {
    }
}
```

中断程序如下：

```
interrupt void cpu_timer0_isr(void)
{
    // 进入中断，中断计数器递增
    CpuTimer0.InterruptCount++;

    // PIEACK 相应中断标志位清零
    PieCtrlRegs.PIEACK.all = PIEACK_GROUP1;

    // 定时器中断标志位清零
    CpuTimer0Regs.TCR.bit.TIF=1;

    // 使能重载
    CpuTimer0Regs.TCR.bit.TRB=1;

    // ncount 递增到194 时清零，清零时改变 LED 状态，LED 为递增计数，从 0 到 16 为一个循环。
    if(ncount==0)
    {
       LED=uLBD;
       uLBD++;
       uLBD%=16;
    }
    ncount++;
    ncount%=194;
}
```

3.编译下载并运行程序，观察现象如图：

将定时器的周期寄存器值设置为 0xffff。观察到 LED 闪烁。

将定时器的周期寄存器值设置为 0x0fff。观察到 LED 闪烁变快。

观测 LED 闪烁时间，基本上估计出的周期和计算结果相符。

LED 闪烁现象的视频在附件中给出。

四、问题与思考

1、 中断向量表中的 TINT0 中断向量里写的是什么？例程里设置的中断周期是多长时间，怎么计算的？LED 每隔多长时间变化一次，怎么计算的？（1.5 分）

（1）中断向量表中的 TINT0 中断向量内写的是定时器 0 的中断入口地址。

（2）例程设计的中断周期：

中断周期由下式给出：

$$T = \frac{(PRDH:PRD)*(TDDRH:TDDR+1)}{SYSCLKOUT}$$

PRDH:PRD 为周期寄存器值；TDDRH:TDD 为定时器预分频寄存器值，SYSCLKOUT 为 CPU 时钟频率。例程中，PRDH:PRD 为 65535，TDDRH:TDD 为 0，SYSCLKOUT 为 150MHz，得到 0.4369ms。

（3）LED 闪烁周期：ncount 递增到 194 时清零，清零时改变 LED 状态，故 LED 闪烁周期为 194 倍中断周期，为 84.75ms。

哈爾濱工業大學（深圳）
HARBIN INSTITUTE OF TECHNOLOGY

实验报告

课程名称：DSP 的原理与应用

实验名称：实验 3.4 单路，多路模数转换（AD）

专业-班级：电气 1 班 学号：220330124 姓名：舒晟超

实验日期：2024 年 5 月 13 日

试验台号：

报告总分数：

教师评语：

助教签字：_____

教师签字：_____

日　　期：_____

一、实验目的

1．通过实验熟悉 F28335 的定时器。
2．掌握 F28335 片内 AD 的控制方法。

二、实验过程

1．导入工程并导入硬件仿真文件，准备信号源进行 AD 输入，设置波形输出。

2. 打开程序并推测实验现象；

3. 下载并运行程序，观察实验现象。

三、实验结果

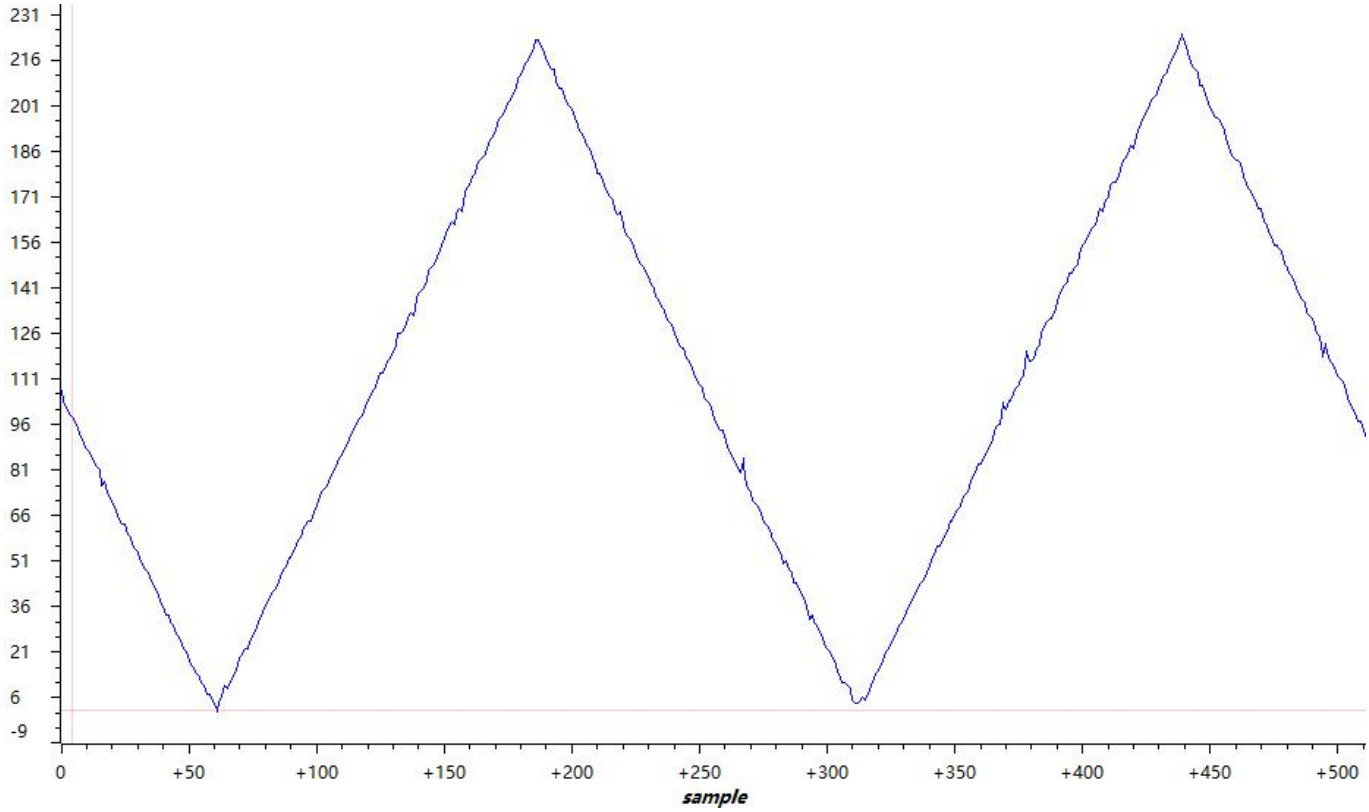
1．导入工程并导入硬件仿真文件。

2. 阅读代码：

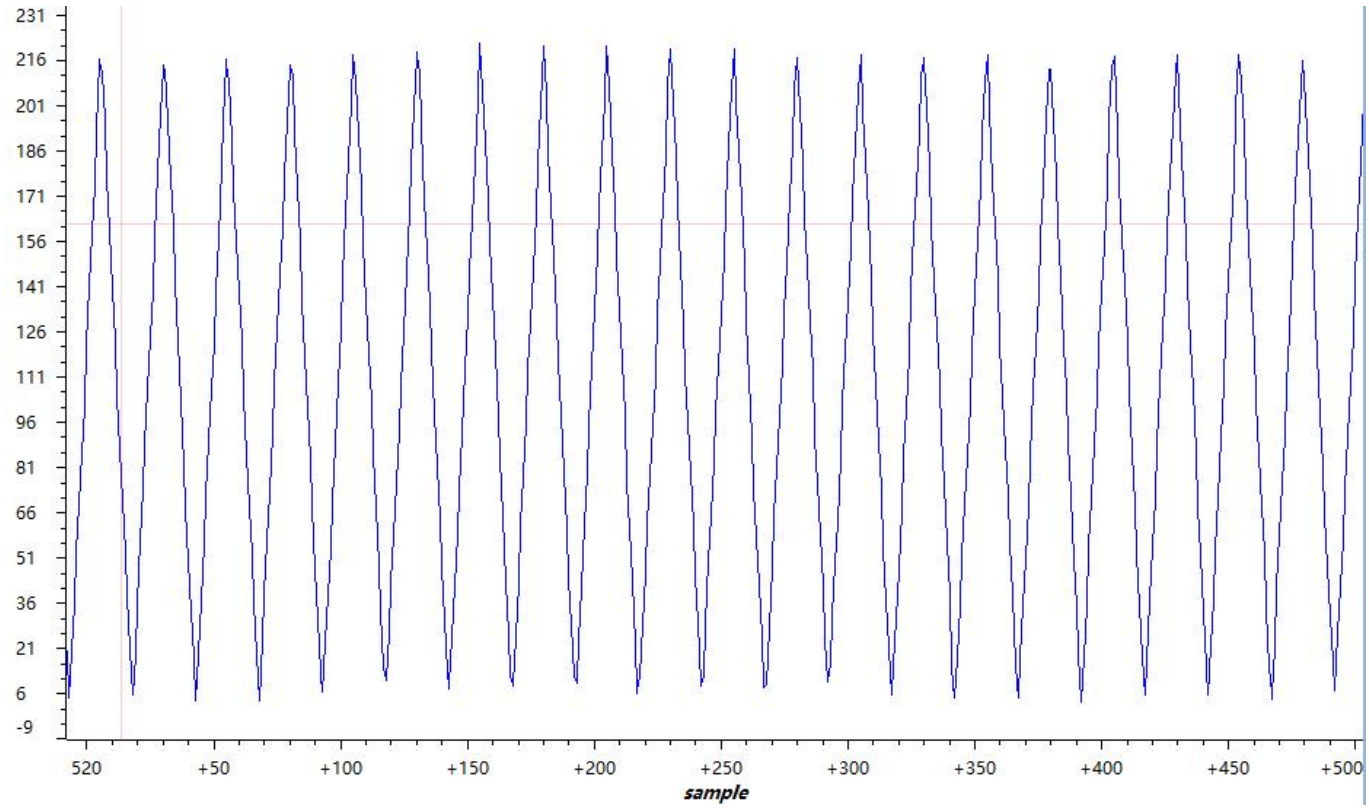
```
// TI 提供 ADC 初始化函数：开启 ADC 时钟，参考电压寄存器设置，延时
InitAdc();
// 顺序采样模式
AdcRegs.ADCTRL1.bit.ACQ_PS = ADC_SHCLK;
// ADC 工作 25M 下不分频
AdcRegs.ADCTRL3.bit.ADCCLKPS = ADC_CKPS;
// 排序器级联模式
AdcRegs.ADCTRL1.bit.SEQ_CASC = 1;
// 配置采样顺序
AdcRegs.ADCCHSELSEQ1.bit.CONV02 = 0x2;
AdcRegs.ADCCHSELSEQ1.bit.CONV03 = 0x3;
// 连续运行模式
AdcRegs.ADCTRL1.bit.CONT_RUN = 1;
// 启用序列器覆盖功能
AdcRegs.ADCTRL1.bit.SEQ_OVRD = 1;
// 最大有效通道数为 2
AdcRegs.ADCMAXCONV.bit.MAX_CONV1 = 0x1;
```

3. 连线，设置波形输入并加入断点，运行程序并观察现象

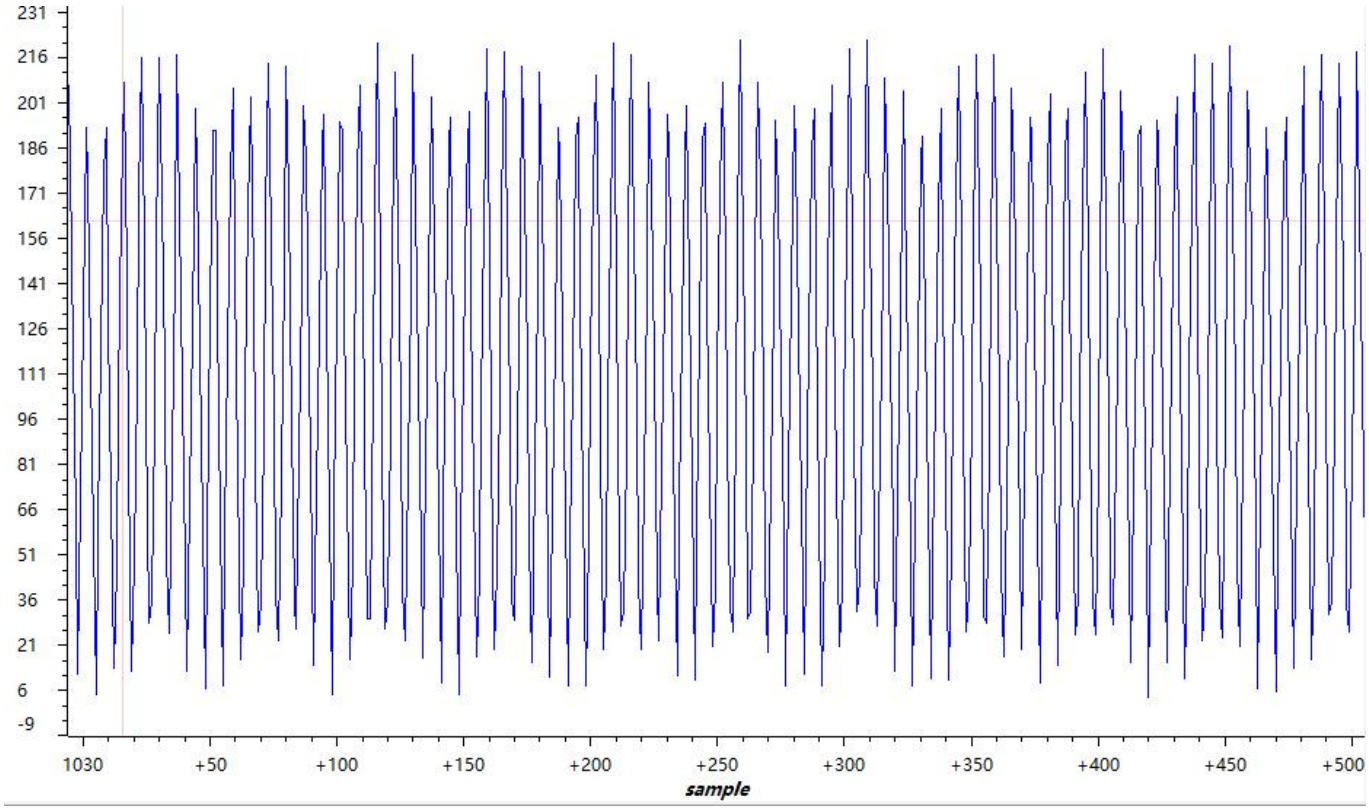
(1)100-1k


231
216
201
186
171
156
141
126
111
96
81
66
51
36
21
6
-9
0
+50
+100
+150
+200
+250
+300
+350
+400
+450
+500
sample

(2)1k-10k


231
216
201
186
171
156
141
126
111
96
81
66
51
36
21
6
-9
520
+50
+100
+150
+200
+250
+300
+350
+400
+450
+500
sample

(3)30k



四、问题与思考

1、 例程里为什么第一次要把 ADCRESULT 里的结果向右移动 4 位？ADC 转换之后得到的最大可能的数是多少？为什么例程里观察到的波形和上面的波形幅值不一样？（1.5 分）

（1）ADC 结果寄存器各位功能描述如下：

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0	X	X	X	X

F28335 内部 AD 只有 12 位，用 16 位的结果寄存器来存储，所以还有 4 位是保留位，得到的寄存器值需要右移 4 位方能得到采样结果。

（2）结果寄存器最大值为 4096，右移 4 位后最大值为 256.

（3）上述观察的波形未右移 4 位，观察到的值为 0-4096，例程波形已右移 4 位。