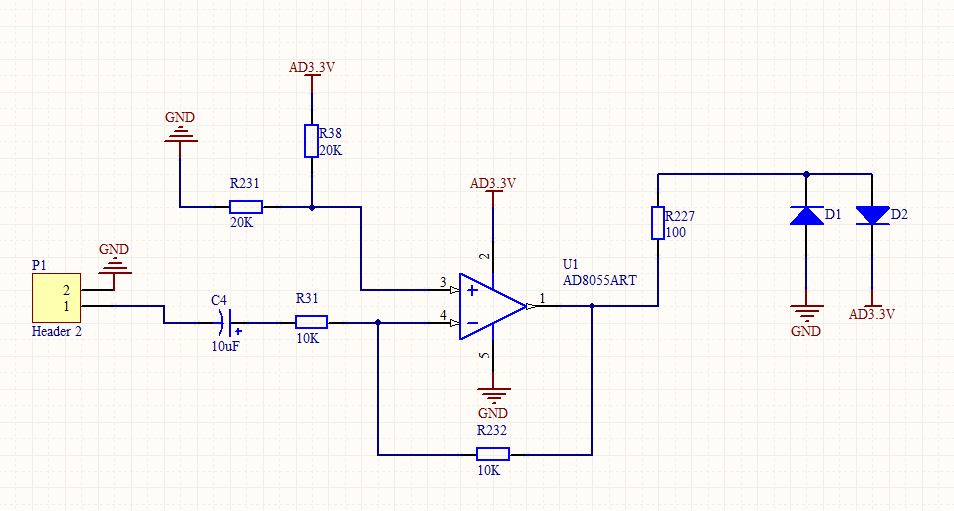
刚刚开始玩DSP，属于一个菜鸟。在网上找了个关于AD采样的程序，几经修改，终于可以在板子上跑起来了。不过，后来不管怎么优化程序，采样得到的波形总是不太如意，波形漂移很严重。

ADC.C 配置如下：  
for(i = 0;i<DELAY\_ADC\_POWERUP;i++){asm("  NOP ");}  
   AdcRegs.ADCTRL1.bit.RESET=1;//ADC模块软件复位位，复位整个ADC模块  
  for(i=0;i<10000;i++) {};  
   AdcRegs.ADCTRL1.bit.SUSMOD=3;//仿真挂起模式设置  
    AdcRegs.ADCTRL1.bit.ACQ\_PS=15;//获取窗口大小  
  AdcRegs.ADCTRL1.bit.CPS=0;//内核时钟预定标器  
  AdcRegs.ADCTRL1.bit.CONT\_RUN=0;//连续运行模式  
  AdcRegs.ADCTRL1.bit.SEQ\_CASC=1;//级联排序器操作位  
    
  for(i = 0;i<15;i++){asm("   NOP ");}  
  AdcRegs.ADCTRL3.bit.ADCBGRFDN = 3;  //Power up bandgap/reference circuity  
  for(i = 0;i<DELAY\_ADC\_POWERUP;i++){asm("   NOP ");}  //Delay defore powering up rest of ADC  
  AdcRegs.ADCTRL3.bit.ADCPWDN = 1;  
       AdcRegs.ADCTRL3.bit.ADCCLKPS =1;  //hspclk/16//内核时钟分频器  
  for(i = 0;i<DELAY\_ADC\_POWERUP;i++){asm("   NOP ");}   
  AdcRegs.ADCTRL3.bit.SMODE\_SEL=0; //采样模式选择  
      
        AdcRegs.ADCTRL2.bit.INT\_ENA\_SEQ1=1;//ESQ1中断使能位  
  AdcRegs.ADCTRL2.bit.RST\_SEQ1=1; //排序器1复位位  
     
  AdcRegs.ADCCHSELSEQ1.bit.CONV00 =0x09;   //ADCINB0输入通道选择  
   AdcRegs.ADCMAXCONV.all = 0;        //最大转换通道寄存器  
  AdcRegs.ADCTRL2.bit.SOC\_SEQ1=1;  
主函数为：  
#include "DSP281x\_Device.h"     // Headerfile Include File  
#include "DSP281x\_Examples.h"   // Examples Include File  
#include "IQmathLib.h"  
#include "pid\_reg3.h"  
unsigned int sampling[128];  
unsigned int sampling1[128];  
int flag1=0,m=0,n1=0;  
interrupt void  timer0(void);  
void main(void)  
{  
   InitSysCtrl();//系统初始化，主要对开门狗，时钟等模块进行初始化，以保证2812正常工作  
   InitGpio();//GPIO的初始化函数，只和GPIO相关  
   nitPeripherals();//外设初始化  
   DINT;//清空INTM全局使能  
   InitPieCtrl();//PIE初始化函数，和中断相关   IER = 0x0000;  
   IFR = 0x0000;  
   InitPieVectTable();//PIE中断向量表定义以及初始化  
   EALLOW;  
   PieVectTable.TINT0  = &timer0;   
   EDIS;  
   IER |= (M\_INT2|M\_INT1);  
   PieCtrlRegs.PIEIER1.bit.INTx7 = 1;    //Enable CPU定时器0  
   InitCpuTimers();//CPU定时器的初始化和配置函数，与CPU的定时器相关  
   ConfigCpuTimer(&CpuTimer0, 150, 19);  //19us  
   StartCpuTimer0();  
   EINT;   // 打开 INTM 全局使能  
   ERTM;   // 允许DEBUG中断  
  
interrupt void timer0(void)  
{         
  AdcRegs.ADCTRL2.bit.SOC\_SEQ1=1;//AD转换软件触发  
  while(AdcRegs.ADCST.bit.INT\_SEQ1==0){}  
  AdcRegs.ADCST.bit.INT\_SEQ1\_CLR=1;  
                sampling[flag1]=((AdcRegs.ADCRESULT0)>>4);  
  flag1 ++;  
         if(flag1==128)   
          {  
           flag1=0;  
           for(m=0;m<128;m++){sampling1[m]=sampling[m] ; }  
                        }  
CpuTimer0.InterruptCount++;  
CpuTimer0Regs.TCR.bit.TIF=1;//CPU定时器中断标志位  
PieCtrlRegs.PIEACK.all = PIEACK\_GROUP1;//PIE响应寄存器  
CpuTimer0Regs.TCR.bit.TRB=1;//CPU定时器重新装载位  
     
return;  
}

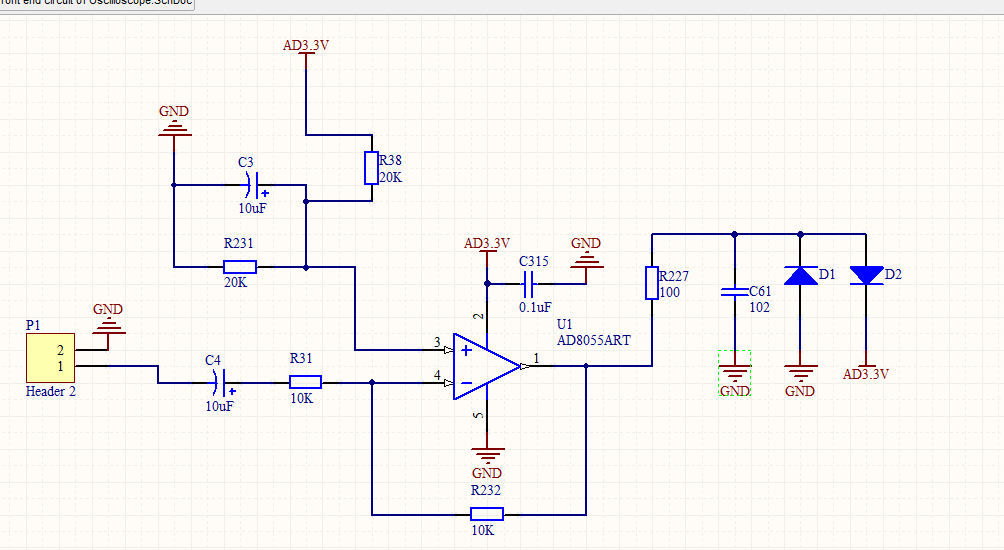
程序已经调到无法再动的程度了，最终我想到了会不会是硬件电路除了什么问题（哎，这时才想到硬件出问题，真得要好好检讨啊 - -！）。

这是刚开始我设计的采样电路：



现在看这个图真的有点无地自容的感觉。

这是我后来改了后的电路：



显而易见，问题就是出在滤波去耦电容上。（还有一点得非常注意，模拟地和数据地要隔离，千万不要嫌麻烦！！！！！！！！！！！！！！！！！！！）

接下来是CCS自带的图形绘制功能得到的A/D采样结果（幅度小于3V，频率为1khz的正弦波）：

