I. Appendix:

```
#include <hidef.h>      /* common defines and
macros */
#include "derivative.h"  /* derivative information
*/
#include "SCI.h"

char string[20];
int val;
int i;
int mode = 0;
float value;

unsigned short angle;
void setCLK8(void);
void delay1ms(unsigned int k);

//---------------------OutCRLF---------------------
-
// Output a CR,LF to SCI to move cursor to a new
line
// Input: none
// Output: none
// Toggle LED each time through the loop

void OutCRLF(void){
  SCI_OutChar(CR);
  SCI_OutChar(LF);
  PTJ ^= 0x20;          // toggle LED D2
}

void main(void) {
    setCLK8();
    DDRJ = 0xFF;

    //analog to digital configuration
    ATDCTL1 = 0x25;       //set for 10 bit
resolution
    ATDCTL3 = 0x88;       //right justified,one
sample per sequence
    ATDCTL4 = 0x03;       //ATD clock =
8MHZ/2*(3+1)=1MHZ
    ATDCTL5 = 0x25;       //continuous conversion
for one channel(channel 5)

    //Port configuration
    DDR1AD = 0x1F;  //set the PAD 0~4 channel of
AD to be output 5 in total
    DDR0AD = 0x0F;  //set the last 4(8,9,10,11)
channel of AD to be output   4 in total
    ATDDIEN = 0x0000;
    PER1AD = 0x00;        //disable pull-up
registers for input pints (A0-A3)

    //interrupt configuration
    TSCR1 = 0x90;
    TSCR2 = 0x00;
    TIOS = 0xFC;
    PERT = 0x03;
    TCTL3 = 0x00;
    TCTL4 = 0x0A;
    TIE = 0x03;

    //mode configuration
    DDRT = 0x00;

    //enable interrupt
    EnableInterrupts;
    SCI_Init(9600);
    SCI_OutString("Zeyang Wen, 400139518");
    mode=0;
      for(;;){
        if(mode % 2 == 1){
           PTJ ^= 0x01;
           val=0;
           for(i=0;i<10;i++){
           val=ATDDR0+val;
           delay1ms(10);
           }
           val=val/10;

           if(val<=530){
            val=530;
           }

           SCI_OutUDec(val);
           OutCRLF();
           delay1ms(10)

           value= ((val * 3.3/1023)-1.71)/0.38;
angle=(value+(value*value*value)/6+(value*value*va
lue*value*value)*3/40+(value*value*value*value*val
ue*value*value)*15/336)*(18000/314);
           if(angle>90){
            angle=90;
           }

           SCI_OutUDec(angle);
           OutCRLF();
          if(PTT==0xC3){
            PT0AD=angle/10;
            PT1AD=angle%10;
          }
          if(PTT==0xC7){
           if(angle>=0 && angle<5){
            PT1AD=0x00;
            PT0AD=0x00;
           }
           if(angle>=5 && angle<15){
            PT1AD=0x01;
            PT0AD=0x00;
           }
           if(angle>=15 && angle<25){
            PT1AD=0x03;
            PT0AD=0x00;
           }
           if(angle>=25 && angle<35){
            PT1AD=0x07;
            PT0AD=0x00;
           }
           if(angle>=35 && angle<45){
            PT1AD=0x0F;
            PT0AD=0x00;
           }
           if(angle>=45 && angle<55){
            PT1AD=0x0F;
            PT0AD=0x01;
           }
           if(angle>=55 && angle<65){
            PT1AD=0x0F;
            PT0AD=0x03;
           }
           if(angle>=65 && angle<75){
            PT1AD=0x0F;
            PT0AD=0x07;
           }
           if(angle>=75 && angle<85){
            PT1AD=0x0F;
            PT0AD=0x0F;
           }
           if(angle>=85 && angle<90){
            PT1AD=0x1F;
            PT0AD=0x0F;
           }
          }
```

```
      }
    }
}

interrupt  VectorNumber_Vtimch1 void
ISR_Vtimch1(void){
    unsigned int temp;
    mode++;
    temp = TC1;
}


void setCLK8(void){
  CPMUPROT = 0;      //disable clock write
protection
  CPMUCLKS = 0x80;       //set PLLSEL=1
  CPMUOSC = 0x80;        // set OSCE=1
  CPMUREFDIV = 0x41;     //set reference frequency
to 8/2=4MHZ
  CPMUSYNR = 0x05;       //set VCOCLK frequency to
2*4*(5+1)=48MHZ
  CPMUPOSTDIV = 0x02;    //set pll frequency to
48/(2+1)=16MHZ
  while(CPMUFLG == 0);  //wait for pll to engage
  CPMUPROT = 1;          //enable clock write
protection
}


void delay1ms(unsigned int k){
  int i;                 //loop contorl variable
  TSCR1 = 0x90;          //enable timer
  TSCR2 = 0x00;          //prescaler=1
  TIOS |= 0x01;
  TC0 = TCNT + 8000;
  TIE = 0x00;
  for(i=0;i<k;i++){
    while(!(TFLG1_C0F));
    TC0 += 8000;
  }
  TIOS &= ~0x01;
  TIE = 0x03;
}
```