PROJECT CODE:

#include <htc.h>

#include "lcd16.h"

void txs(unsigned char val)

{

int i;

while(!TXIF)

continue;

TXREG=val;

for(i=0;i<1200;i++)**;**

}

void delay2()

    {

long i;

    for(i=0;i<10000;i++);

    }

void delay3()

    {

    long i;

    for(i=0;i<58500;i++);

    }

unsigned char rxs(void)

{

}

int c=0;

while(!RCIF)

    {

         c++;

             if(c>5000)

                break;

}

return RCREG;

}

unsigned char val[6],flagx=0,val1,val2,val3,val4,val5;

int j=0;

long x,setp;

int sp,sp1,spt;

    unsigned int d=0,e=0,f=0,flgg=0;

    unsigned int m=0;

void Delay()

{

     long i;

     for(i=0;i<5000;i++)

{

        }

}

void Delay2()

{

long i;

for(i=0;i<100000;i++)

        {

        }

}

void sendtx(unsigned char val[],unsigned char length)

    {

     int i;

     for(i=0;i<length;i++)

        {

            txs(val[i]);

        }

    }

/\* Sample code to set up the A2D module \*/

void init\_a2d(void){

    ADCON0=0;   // select Fosc/2

    ADCON1=2;   // select left justify result. A/D port configuration 0

ADON=1;     // turn on the A2D conversion module

}

/\* Return an 8 bit result \*/

unsigned char read\_a2d(unsigned char channel){

    channel&=0x07;  // truncate channel to 3 bits

    ADCON0&=0xC5;   // clear current channel select

    ADCON0|=(channel<<3);   // apply the new channel select

    Delay();

    GO=1;   // initiate conversion on the selected channel

    while(GO)continue;

    return(ADRESH); // return 8 MSB of the result

}

long ll;

unsigned long rval=1,rvalt=1,cc,val6,val7,val8,val9;

int c,fg,fgg;

char oc2,oc1;

static void interrupt isr(void)         // Here be interrupt function - the name is

{

   if(INTF)      // Was this a timer overflow?

      {

      val6++;

      INTF = 0;      // Clear interrupt flag, ready for next

      }

}

int oc,oc3;

char va2;

void main()         //Main entry

{

CMCON=0x07;

    RCSTA=0x90;                             // receive enabling with the same speed

    TXSTA =(0x24);                  // High speed selection baud rate 9600

    SPBRG = 64;

    TRISD=0x00;

    TRISE=0x00;

    Lcdinit();

init\_a2d();

    TRISB=0xFF;

    TRISC=0x00;

    TRISD=0x00;

    TRISE=0x00;

    nRBPU=0;

    //GIE=1;

    //PEIE=1;

    //INTE=1;

    while(1)

    {

va2=read\_a2d(0);

    if(va2<10)

{

        if(oc2==0)

        {

        oc2=1;

        if(val2<200)

        val2++;

        }

    }

    else

    {

        oc2=0;

    }

va2=read\_a2d(1);

    if(va2<10)

    {

        if(oc1==0)

        {

        oc1=1;

        if(val3<200)

val3++;

        }

    }

    else

    {

        oc1=0;

    }

va2=read\_a2d(2);

    if(va2<10)

    {

        if(oc3==0)

        {

        oc3=1;

        if(val8<200)

        val8++;

        }

    }

    else

    {

        oc3=0;

    }

    Lcdcmd(0x80);

    lcddata('F');

    lcddata(' ');

    lcddata((val2/100)+0x30);

    lcddata(((val2%100)/10)+0x30);

    lcddata(((val2%10)/1)+0x30);

    Lcdcmd(0x86);

    lcddata((val3/100)+0x30);

    lcddata(((val3%100)/10)+0x30);

    lcddata(((val3%10)/1)+0x30);

    Lcdcmd(0x8B);

    lcddata((val8/100)+0x30);

lcddata(((val8%100)/10)+0x30);

    lcddata(((val8%10)/1)+0x30)

    Delay();

    val1=!RB0;

    Lcdcmd(0xC0);

    lcddata('G');

    lcddata(' ');

    lcddata((val1/100)+0x30);

    lcddata(((val1%100)/10)+0x30);

    lcddata((val1%10)+0x30);

    Delay();

    val5=!RB1;

    Lcdcmd(0xC6);

    lcddata((val5/100)+0x30);

    lcddata(((val5%100)/10)+0x30);

    lcddata((val5%10)+0x30);

    Delay();

    val9=!RB2;

    Lcdcmd(0xCB);

    lcddata((val9/100)+0x30);

    lcddata(((val9%100)/10)+0x30);

    lcddata((val9%10)+0x30);

txs('A');

    txs((val2/100)+0x30);

    txs(((val2%100)/10)+0x30);

    txs((val2%10)+0x30);

    txs('B');

    txs((val3/100)+0x30);

    txs(((val3%100)/10)+0x30);

    txs((val3%10)+0x30);

    txs('C');

txs((val8/100)+0x30);

    txs(((val8%100)/10)+0x30);

    txs((val8%10)+0x30);

    txs('D');

    txs((val4/100)+0x30);

    txs(((val4%100)/10)+0x30);

    txs((val4%10)+0x30);

    txs('E');

rval=RCREG;

RCIF=0;

OERR=0;

CREN=0;

CREN=1;

if(rval=='A')

    {

        RC0=1;

    }

    if(rval=='B')

    {

        RC0=0;

    }

    if(rval=='C')

    {

        RC1=1;

    //RC2=0;

    //RC3=0;

    }

    if(rval=='D')

    {

        RC1=0;

    }

if(rval=='E')

    {

        RC2=1;

    }

if(rval=='F')

    {

        RC2=0;

    }

//  if (oc==0)

//      {

//          txs('\*');

//          if(val2<2)

//          sendtx("F1:NF",5);

//          else

//          sendtx("F1:FL",6);

//          if(val3<2)

//       sendtx("\_F2:NF",6);

//          else

//          sendtx("\_F2:FL",6);

//          if(val8<2)

//          sendtx("\_F3:NF",6);

//          else

//          sendtx("\_F3:FL",6);

//          if(val1==0)

//          sendtx("\_G1:NO",6);

//          else

//          sendtx("\_G1:AB",6);

//          if(val5==0)

//          sendtx("\_G2:NO",6);

//          else

//          sendtx("\_G2:AB",6);

//          if(val9==0)

//          sendtx("\_G3:NO",6);

//          else

//          sendtx("\_G3:AB",6);

//          txs('#');

// oc=10;

    oc-

}

}