//Seven Segment Display Program:

//P0.19 Data pin of 1st shift register

//P0.20 Clock pin of shift registers, make 1 to 0

//P0.30 Strobe pin of shift registers: 1 to 0

#include <stdio.h>

#include <lpc214x.h>

#define LED\_OFF (IO0SET = 1U << 31)

#define LED\_ON (IO0CLR = 1U << 31)

#define PLOCK 0x00000400

void delay\_ms(unsigned int j);

void SystemInit(void);

unsigned char getAlphaCode(unsigned char alphachar);

void alphadisp7SEG(char \*buf);

unsigned int adc(int no,int ch);

void runDCMotor(int direction,int dutycycle);

long int ans = 1234;

char buf[5];

int main()

{

IO0DIR |= 1U << 31 | 1U << 19 | 1U << 20 | 1U << 30 ; // to set as o/ps

LED\_ON; // make D7 Led on .. just indicate the program is running

while(1)

{

ans = adc(1,3);

sprintf(buf,"%05lu",ans);

alphadisp7SEG(&buf[0]);

runDCMotor(1,ans/10);

delay\_ms(100);

}

}

unsigned char getAlphaCode(unsigned char alphachar)

{

switch (alphachar)

{

// dp g f e d c b a - common anode: 0 segment on, 1 segment off

case 'f': return 0x8e;

case 'i': return 0xf9;

case 'r': return 0xce;

case 'e':return 0x86; // 1000 0110

case 'h':return 0x89;

case 'l': return 0xc7;

case 'p':return 0x8c;

case '0': return 0xc0;

case '1': return 0xf9;

case '2': return 0xa4;

case '3': return 0xb0 ;

case '4': return 0x99;

case '5': return 0x92;

case '6': return 0x82;

case '7': return 0xf8;

case '8': return 0x80;

case '9': return 0x90;

case ' ': return 0xff;

//simmilarly add for other digit/characters

default : break;

}

return 0xff;

}

void alphadisp7SEG(char \*buf)

{

unsigned char i,j;

unsigned char seg7\_data,temp=0;

for(i=0;i<5;i++) // because only 5 seven segment digits are present

{

seg7\_data = getAlphaCode(\*(buf+i)); //instead of this look up table can be used

//to shift the segment data(8bits)to the hardware (shift registers) using Data,Clock,Strobe

for (j=0 ; j<8; j++)

{

//get one bit of data for serial sending

temp = seg7\_data & 0x80; // shift data from Most significan bit (D7)

if(temp == 0x80)

IOSET0 |= 1 << 19; //IOSET0 | 0x00080000;

else

IOCLR0 |= 1 << 19; //IOCLR0 | 0x00080000;

//send one clock pulse

IOSET0 |= 1 << 20; //IOSET0 | 0x00100000;

delay\_ms(1);

IOCLR0 |= 1 << 20; //IOCLR0 | 0x00100000;

seg7\_data = seg7\_data << 1; // get next bit into D7 position

}

}

// send the strobe signal

IOSET0 |= 1 << 30; //IOSET0 | 0x40000000;

delay\_ms(1); //nop();

IOCLR0 |= 1 << 30; //IOCLR0 | 0x40000000;

return;

}

void delay\_ms(unsigned int j)

{

unsigned int x,i;

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

{

for(x=0; x<10000; x++);

}

}

unsigned int adc(int no,int ch)

{

// adc(1,4) for temp sensor LM34, digital value will increase as temp increases

// adc(1,3) for LDR - digival value will reduce as the light increases

// adc(1,2) for trimpot - digital value changes as the pot rotation

unsigned int val;

PINSEL0 |= 0x0F300000; /\* Select the P0\_13 AD1.4 for ADC function \*/

/\* Select the P0\_12 AD1.3 for ADC function \*/

/\* Select the P0\_10 AD1.2 for ADC function \*/

switch (no) //select adc

{

case 0: AD0CR=0x00200600|(1<<ch); //select channel

AD0CR|=(1<<24); //start conversion

while((AD0GDR& (1U<<31))==0);

val=AD0GDR;

break;

case 1: AD1CR=0x00200600|(1<<ch); //select channel

AD1CR|=(1<<24); //start conversion

while((AD1GDR&(1U<<31))==0);

val=AD1GDR;

break;

}

val=(val >> 6) & 0x03FF; // bit 6:15 is 10 bit AD value

return val;

}

void runDCMotor(int direction,int dutycycle)

{

IO0DIR |= 1U << 28; //set P0.28 as output pin

PINSEL0 |= 2 << 18; //select P0.9 as PWM6 (option 2)

if (direction == 1)

IO0SET = 1 << 28; //set to 1, to choose anti-clockwise direction

else

IO0CLR = 1 << 28; //set to 0, to choose clockwise direction

PWMPCR = (1 << 14); // enable PWM6

PWMMR0 = 1000; // set PULSE rate to value suitable for DC Motor operation

PWMMR6 = (1000U\*dutycycle)/100; // set PULSE period

PWMTCR = 0x00000009; // bit D3 = 1 (enable PWM), bit D0=1 (start the timer) PWMLER = 0X70; // load the new values to PWMMR0 and PWMMR6 registers

PWMLER = 0X70;

}