#include "QSKDefines.h"

#include "proto.h"

#include "extern.h"

void timer\_init(void); //function for initializing timer related registers

unsigned char owntime[]="00:00";//character string which shows the own board time count

unsigned char othertime[]="00:00";//character string which shows the other board time count

unsigned int send\_char\_count; //count the numbers of characters sent

unsigned int receive\_char\_count; // count the numbers of characters received

unsigned int seconds\_timer = FALSE; // flag when timer ends

unsigned int sw1\_pressed = FALSE; // flag for sw1

unsigned int sw2\_pressed = FALSE; // flag for sw2

void main(void){

unsigned int set\_flag =0; //local variable for selecting a particular case, when switch 1 is pressed

MCUInit();

InitDisplay("\tSample \n ");

InitUART();

timer\_init();

BNSPrintf(SERIAL\_FILE\_NUM, "\n\rSample\n\r");

ENABLE\_LEDS;

ENABLE\_SWITCHES;

\_asm (" fclr i"); // turn off interrupts

ta1ic |= 0x02; // Interrupt priority of timer1 interrupt set to 3

s0tic |= 0x05; // Interrupt priority of uart0 recieve interrupt set to 5

s0ric |= 0x06; // Interrupt priority of uart1 transmit interrupt set to 6

int0ic |= 0x04; // Interrupt priority of int0/sw2 interrupt set to 4

int1ic |= 0x03; // Interrupt priority of int1/sw1 interrupt set to 3

\_asm (" fset i"); // globally enable interrupts

while(1) {

if (sw1\_pressed == TRUE){

if (set\_flag==0) { //when stopwatch is started

ta0s = 1; //start the timers

ta1s = 1;

while(!S1);

set\_flag = 1;

}

else if (set\_flag==1){ //when stop watch is paused

ta0s = 0; //pause the timers

ta1s = 0;

while(!S1); //wait for s1 to go back up

set\_flag = 0;

}

else if (set\_flag==2){ //when stop watch is started after resetting

ta0 = 37499; //reload the timers

ta1=20;

ta0s = 1;

ta1s = 1;

while(!S1);

set\_flag = 1;

}

sw1\_pressed = FALSE;

}

if(sw2\_pressed == TRUE){ //when reset switch is pressed

ta0s = 0;

ta1s = 0;

ta2s = 0;

owntime[0]=0x30;

owntime[1]=0x30;

owntime[2]=0x30;

owntime[3]=0x30;

u0tb= owntime[0];

while(!S2);

set\_flag = 2;

sw2\_pressed = FALSE;

}

if(seconds\_timer == TRUE){ //when timer runs out

if (owntime[4]<=0x38){//when sec value is less than 9sec

owntime[4]= owntime[4]+1;

}

else if(owntime[4]>0x38 && owntime[3]<=0x34){ //when sec value is more than 9sec

owntime[4]=0x30;

owntime[3]= owntime[3]+1;

}

else if(owntime[4]>0x38 && owntime[3]>0x34){ //when sec value is more than 59sec

owntime[3]=0x30;

owntime[4]=0x30;

if(owntime[1]<=0x38){ //when min value is less than 60

owntime[1]= owntime[1]+1;

}

else if(owntime[1] > 0x38 && owntime[0] <= 0x34){

owntime[0] = owntime[0] + 1;

owntime[1] = 0x30;

}

else if(owntime[1]> 0x38 && owntime[0] > 0x34){

owntime[0] =0x30;

owntime[1] = 0x30;

}

}

seconds\_timer = FALSE;

disp\_time(1); //function that display own time on LCD line1

u0tb= ownitme[0];

}

if( receive\_char\_count == 0){ //to make sure othertime has beed received properly

disp\_time(2); //function that display other time on LCD line1

}

}