/\*

\*/

#include <stdio.h>

#include <stdlib.h>

#include <mcs51reg.h>

#include <stdint.h>

#include <mcs51/8051.h>

#include <at89c51ed2.h>

#define CKCON0\_X2 0x1F;

int number=0;

\_sdcc\_external\_startup()

{

AUXR |= 0x0C;

return 0;

}

/\* Function to take input from keyboard.\*/

int getchar(void)

{

while(!RI)

{

;

}

RI=0;

return SBUF;

}

/\* Function to print on terminal. \*/

int putchar(int x)

{

while(!TI){

;

}

SBUF=x;

TI=0;

return x;

}

/\* Function to take number input.\*/

uint16\_t input\_num()

{

\_\_xdata uint16\_t number=0;

\_\_xdata uint8\_t count=0;

\_\_xdata uint8\_t input\_digit=0;

\_\_xdata uint8\_t input\_digits[5];

while(input\_digit!=13)

{

input\_digit=getchar();

if(input\_digit==8)

{

putchar(8);

count--;

}

else if((input\_digit<='9') && (input\_digit>='0'))

{

putchar(input\_digit);

input\_digits[count]=input\_digit-'0';

count++;

}

}

for(int i=0;i<count;i++)

{

number=number\*10;

number=number+input\_digits[i];

}

printf\_tiny("\n\r entered input:%d\n\r",number);

return number;

}

void init()

{

CKCON0 = CKCON0\_X2; //FOR X2 MODE

IEN0 |= 0x80; //TIMER 1 OVERFLOW INTERRUPT ENABLED

TMOD |= 0x20; //TIMER 1, MODE 2

SCON |= 0x50; //8 BIT, 1 STOP , REN ENABLED

TCON |= 0x40; //START TIMER1

TH1 = 0xFD; //To set baud rate to 9600

TI = 1;

}

void start\_pwm()

{

CMOD = 0x02; //FCLKPERIPH/2

CL = 0x00; //PCA TIMER

CH = 0x00;

CCAP1L = 0xab;

CCAP1H = 0xab;

CCON = 0x40; // SET CR- Counter run bit

CCAPM1 = 0x42; // RUN 8-BIT PWM FUNCTION

}

void stop\_pwm()

{

CMOD = 0x02; //FCLKPERIPH/2

CL = 0x00; //PCA TIMER

CH = 0x00;

CCAP1L = 0xff;

CCAP1H = 0xff;

CCON = 0x40; // SET CR- Counter run bit

CCAPM1 = 0x00; // STOP 8-BIT PWM FUNCTION

}

void main(void)

{

init();

menu:

printf\_tiny("\n\r\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*MENU\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\r");

printf\_tiny("\n\r 1. Run PWM \n\r");

printf\_tiny("\n\r 2. Stop PWM \n\r");

printf\_tiny("\n\r 3. Set FCLK PERIPH at the minimum frequency supported by the CKRL register \n\r");

printf\_tiny("\n\r 4. Set FCLK PERIPH at the maximum frequency supported by the CKRL register \n\r");

printf\_tiny("\n\r 5. Enter Idle mode \n\r");

printf\_tiny("\n\r 6. Enter Power Down mode \n\r");

printf\_tiny("\n\r 7. Enter High speed mode \n\r");

printf\_tiny("\n\r 8. Watchdog mode \n\r");

printf\_tiny("\r\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\r");

printf\_tiny("\n\r Created by Gauriech Ishaan Pilla.\n\r");

while(1)

{

printf\_tiny("\n\r Enter a number from menu or press 9 to see menu again \n\r");

number=input\_num();

putchar(number);

if(number==1)

{

printf\_tiny("\n\rPWM ON\n\r");

start\_pwm();

}

else if(number==2)

{

printf\_tiny("\n\rPWM OFF\n\r");

stop\_pwm();

}

else if(number ==3)

{

printf\_tiny("\n\rMinimum Frequency\n\r");

CKRL =0x00; //Clock Reload Register to 0 for minimum frequency

}

else if(number ==4)

{

printf\_tiny("\n\rMaximum Frequency\n\r");

CKRL =0xFF; //Clock Reload Register to FF for maximum frequency

}

else if(number ==5)

{

printf\_tiny("\n\rIdle mode\n\r");

IEN0 |= 0x01; //External Interrupt enabled

IPL0 |= 0x01;

CMOD &= 0x7F;

PCON |= 0x01; // PCON set to idle mode

}

else if(number ==6)

{

printf\_tiny("\n\rPower down mode\n\r");

PCON |=0x02; // PCON set to power down mode

}

else if(number ==7)

{

printf\_tiny("\n\r High-Speed mode\n\r");

CMOD = 0x02; //FCLKPERIPH/2

CCAP0L = 0x7F;

CCAP0H = 0x7F;

CCON = 0x40; // SET CR- Counter run bit

CCAPM0 |= 0x4C; // Run high speed mode

}

else if(number ==8)

{

printf\_tiny("\n\r Watch-dog mode\n\r");

// watchdog\_switch();

// watchdog();

}

else if(number ==9)

{

goto menu;

}

}

}

----------------

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* File name : main.c

\* Description : ESD lab 3 part 3 supplemental sdcc code. This code first accepts a 1 digit input from the user. Then performs

\* the relevant function assigned to it.

\*

\* Press 1 to Run PWM

\* Press 2. Stop PWM

\* Press 3 to Set FCLK PERIPH at the minimum frequency supported by the CKRL register

\* Press 4 to Set FCLK PERIPH at the maximum frequency supported by the CKRL register

\* Press 5 to Enter Idle mode

\* Press 6 to Enter Power Down mode

\* Press 7 to Enter High speed mode

\* Press 8 to Watchdog mode

\*

\*

\* Author: Gauriech Ishaan Pilla.

\* Tools : CODEBLOCKS IDE.

\* Date: 3/11/2022.

\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\* -------------------------------------------------- \*/

// INCLUDES HERE

/\* -------------------------------------------------- \*/

#include <stdio.h>

#include <stdlib.h>

#include <at89c51ed2.h>

#include <mcs51reg.h>

#include <stdint.h>

#include <mcs51/8051.h>

/\* -------------------------------------------------- \*/

// DECLARING VARIABLES

/\* -------------------------------------------------- \*/

int number=0;

\_\_xdata uint8\_t watchdog\_flag = 0;

/\* -------------------------------------------------- \*/

// SDCC EXTRERNAL STARTUP

/\* -------------------------------------------------- \*/

\_sdcc\_external\_startup()

{

AUXR |= 0x0C;

return 0;

}

/\* -------------------------------------------------- \*/

// PUTCHAR FUNCTION

/\* -------------------------------------------------- \*/

int putchar (int c)

{

while (!TI);

SBUF = c; // load serial port with transmit value

TI = 0; // clear TI flag

return c;

}

/\* -------------------------------------------------- \*/

// GETCHAR FUNCTION

/\* -------------------------------------------------- \*/

int getchar (void)

{

while (!RI);

RI = 0; // clear RI flag

return SBUF; // return character from SBUF

}

/\* -------------------------------------------------- \*/

// NUMBER INPUT FUNCTION

/\* -------------------------------------------------- \*/

uint16\_t num\_input()

{

//initialize the variables for digit store

\_\_xdata uint8\_t digit = 0;

\_\_xdata uint8\_t digit\_store[10];

\_\_xdata uint8\_t counter = 0;

\_\_xdata uint16\_t number = 0;

//while digit is not 13

while(digit!=13)

{

//store the character

digit=getchar();

//check if it is digit and store the digit into the array in decimal form

if((digit >= '0') && (digit <= '9'))

{

putchar(digit);

digit\_store[counter]=digit-'0';

counter++;

}

//check if user entered backspace

else if(digit==8) //check for backspace

{

putchar(8); //print backspace

putchar(32); //print space

putchar(8); //print backspace

counter--;

}

}

printf\_tiny("\n\r");

//calculated the number from the array

for(int buffer\_number=0; buffer\_number<counter; buffer\_number++)

{

number\*= 10;

number+= digit\_store[buffer\_number];

}

return number;

}

/\* -------------------------------------------------- \*/

// HARDWARE INITIALIZATION

/\* -------------------------------------------------- \*/

void init()

{

CKCON0 = 0x1F; //FOR X2 MODE

IEN0 |= 0x80; //TIMER 1 OVERFLOW INTERRUPT ENABLED

TMOD |= 0x20; //TIMER 1, MODE 2

SCON |= 0x50; //8 BIT, 1 STOP , REN ENABLED

TCON |= 0x40; //START TIMER1

TH1 = 0xFD; //To set baud rate to 9600

TI = 1;

}

/\* -------------------------------------------------- \*/

// RUN PWM FUNCTION

/\* -------------------------------------------------- \*/

void init\_pwm()

{

CMOD = 0x02; //FCLKPERIPH/2

CL = 0x00; //PCA TIMER

CH = 0x00;

CCAP1L = 0xbf;

CCAP1H = 0xbf;

CCON = 0x40; // SET CR- Counter run bit

CCAPM1 = 0x42; // RUN 8-BIT PWM FUNCTION

}

/\* -------------------------------------------------- \*/

// STOP PWM FUNCTION

/\* -------------------------------------------------- \*/

void stop\_pwm()

{

CMOD = 0x02; //FCLKPERIPH/2

CL = 0x00; //PCA TIMER

CH = 0x00;

CCAP1L = 0xff;

CCAP1H = 0xff;

CCON = 0x40; // SET CR- Counter run bit

CCAPM1 = 0x00; // STOP 8-BIT PWM FUNCTION

}

/\* -------------------------------------------------- \*/

// WATCHDOG SWITCH

/\* -------------------------------------------------- \*/

void watchdog\_switch(void)

{

if(watchdog\_flag==0)

{

printf\_tiny("watchdog\_on");

watchdog\_flag=1;

}

else

{

printf\_tiny("watchdog\_off");

watchdog\_flag=0;

}

}

/\* -------------------------------------------------- \*/

// WATCHDOG

/\* -------------------------------------------------- \*/

void watchdog(void)

{

\_\_xdata uint32\_t value=0;

if(watchdog\_flag==1)

{

CCAP4L = 0x00;

CCAP4H = 0xFF;

CMOD |= 0x40; // Watchdog Timer enabled

CCON = 0x40; // SET CR- Counter run bit

CCAPM4 |= 0x4C; // Run Watchdog Timer

}

else

{

CMOD = 0x00; // Watchdog Timer disabled

}

return;

}

/\* -------------------------------------------------- \*/

// MAIN

/\* -------------------------------------------------- \*/

void main(void)

{

init();

menu:

printf\_tiny("\n\r\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*MENU\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\r");

printf\_tiny("\n\r 1. Run PWM \n\r");

printf\_tiny("\n\r 2. Stop PWM \n\r");

printf\_tiny("\n\r 3. Set FCLK PERIPH at the minimum frequency supported by the CKRL register \n\r");

printf\_tiny("\n\r 4. Set FCLK PERIPH at the maximum frequency supported by the CKRL register \n\r");

printf\_tiny("\n\r 5. Enter Idle mode \n\r");

printf\_tiny("\n\r 6. Enter Power Down mode \n\r");

printf\_tiny("\n\r 7. Enter High speed mode \n\r");

printf\_tiny("\n\r 8. Watchdog mode \n\r");

printf\_tiny("\r\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\r");

printf\_tiny("\n\r Created by Gauriech Ishaan Pilla.\n\r");

while(1)

{

printf\_tiny("\n\r Enter a number from menu or press 9 to see menu again \n\r");

number=num\_input();

putchar(number);

if(number==1)

{

printf\_tiny("\n\rPWM ON\n\r");

init\_pwm();

}

else if(number==2)

{

printf\_tiny("\n\rPWM OFF\n\r");

stop\_pwm();

}

else if(number ==3)

{

printf\_tiny("\n\rMinimum Frequency\n\r");

CKRL =0x00; //Clock Reload Register to 0 for minimum frequency

}

else if(number ==4)

{

printf\_tiny("\n\rMaximum Frequency\n\r");

CKRL =0xFF; //Clock Reload Register to FF for maximum frequency

}

else if(number ==5)

{

printf\_tiny("\n\rIdle mode\n\r");

IEN0 |= 0x01; //External Interrupt enabled

IPL0 |= 0x01;

CMOD &= 0x7F;

PCON |= 0x01; // PCON set to idle mode

}

else if(number ==6)

{

printf\_tiny("\n\rPower down mode\n\r");

PCON |=0x02; // PCON set to power down mode

}

else if(number ==7)

{

printf\_tiny("\n\r High-Speed mode\n\r");

CMOD = 0x02; //FCLKPERIPH/2

CCAP0L = 0x7F;

CCAP0H = 0x7F;

CCON = 0x40; // SET CR- Counter run bit

CCAPM0 |= 0x4C; // Run high speed mode

}

else if(number ==8)

{

printf\_tiny("\n\r Watch-dog mode\n\r");

watchdog\_switch();

watchdog();

}

else if(number ==9)

{

goto menu;

}

}

}