//-----------------------------------------------------------------------------

// TP2\_BSE.c

//-----------------------------------------------------------------------------

// AUTH: LAJUGIE Rodolphe, CORNATON Maxime

// DATE: 29/11/2022

//

// Target: C8051F02x

// Tool chain: KEIL Microvision5

//

//-----------------------------------------------------------------------------

// Fichiers d'entête

void Oscillator\_Init(void);

void Config\_Timer4\_CounterBase(void);

#include "intrins.h"

#include<c8051F020.h>

#include<c8051F020\_SFR16.h>

#include<TP3\_BSE\_Lib\_Config\_Globale.h>

#include<TP3\_BSE\_Lib\_Divers.h>

#include<TP3\_BSE\_Main.h>

//-----------------------------------------------------------------------------

// Déclaration des MACROS

#define SYSCLK 2000000L

#define LED\_ON 1

#define LED\_OFF 0

#define LED\_BLINK 0

#define BP\_ON 0

#define BP\_OFF 1

#define TO\_BE\_PROCESSED 1

#define PROCESSED 0

#define SET\_VISU\_INT6 P6 |= (1<<4)

#define RESET\_VISU\_INT6 P6 &= ~(1<<4)

//-----------------------------------------------------------------------------

// Déclarations Registres et Bits de l'espace SFR

sbit LED = P1^6; // LED

sbit BP =P3^7;

sbit VISU\_INT7 = P2^4;

sbit VISU\_INT\_TIMER2 = P3^5;

sbit SIGN\_OUT = P1^1;

//-----------------------------------------------------------------------------

// Variable globale

bit Event = PROCESSED;

//-----------------------------------------------------------------------------

// MAIN Routine

//-----------------------------------------------------------------------------

void main (void) {

// Configurations globales

Oscillator\_Init();

Init\_Device();

Modif\_Cfg\_Globale ();

// Configurations spécifiques

Config\_INT7(); // Configuration de INT7

Config\_INT6(); // Configuration de INT6

Config\_Timer2\_TimeBase();

Config\_Timer4\_CounterBase();

// Fin des configurations

EA = 1; // Validation globale des interruptions

// Boucle infinie

while(1);

}

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

void Config\_INT7(void)

{

P3IF &= ~(1<<7); // IE7 mis à 0 pending flag de INT7 effacé

P3IF &= ~(1<<3); // IE7CF mis à 0 - sensibilité int7 front descendant

EIP2 &= ~(1<<5); // PX7 mis à 0 - INT7 priorité basse

EIE2 |= (1<<5); // EX7 mis à 1 - INT7 autorisée

}

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

void ISR\_INT7 (void) interrupt 19

{

VISU\_INT7 = 1;

P3IF &= ~(1<<7); // IE3 mis à 0 - remise à zéro du pending flag de INT7 effacé

Event = TO\_BE\_PROCESSED;

VISU\_INT7 = 0;

}

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

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

void Config\_INT6(void)

{

P3IF &= ~(1<<7); // IE6 mis à 0 pending flag de INT6 effacé

P3IF &= ~(1<<2); // IE6CF mis à 0 - sensibilité int6 front descendant

EIP2 &= ~(1<<4); // PX6 mis à 0 - INT6 priorité basse

EIE2 |= (1<<4); // EX6 mis à 1 - INT6 autorisée

}

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

void ISR\_INT6 (void) interrupt 18

{

SET\_VISU\_INT6;

P3IF &= ~(1<<6); // IE6 mis à 0 - remise à zéro du pending flag de INT6 effacé

P3IF ^= (1<<2); // Action sur IE6CF - Commutation Front montant / Front Descendant

Event = TO\_BE\_PROCESSED;

RESET\_VISU\_INT6;

}

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

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

void Config\_Timer2\_TimeBase(void)

{

CKCON &= ~(1<<5); // T2M: Timer 2 Clock Select

// CLK = sysclk/12TR2 = 0; //Stop Timer

TF2 = 0; // RAZ TF2

EXF2 = 0; // RAZ EXF2

RCLK0 = 0;

TCLK0 = 0;

CPRL2 = 0; // Mode AutoReload

EXEN2 = 0; // Timer2 external Enable Disabled

CT2 = 0; // Mode Timer

RCAP2L = 0x00;

RCAP2H = 0xB8;

T2 = RCAP2;

TR2 = 1; // Timer2 démarré

PT2 = 1; // Priorité Timer2 Haute

ET2 = 1; // INT Timer2 autorisée

}

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

void ISR\_Timer2 (void) interrupt 5

{

static char CP\_Cligno;

static bit STATE\_LED = LED\_BLINK;

VISU\_INT\_TIMER2 = 1;

CP\_Cligno++;

if (CP\_Cligno > 11) CP\_Cligno = 0;

if (TF2 == 1)

{

TF2 = 0;

if (Event == TO\_BE\_PROCESSED)

{

Event = PROCESSED;

STATE\_LED = !STATE\_LED;

}

if (STATE\_LED == LED\_BLINK)

{

if (CP\_Cligno < 2) LED = LED\_ON;

else LED = LED\_OFF;

}

else LED = LED\_OFF;

}

if (EXF2 == 1)

{

EXF2 = 0;

}

VISU\_INT\_TIMER2 = 0;

}

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

void Modif\_Cfg\_Globale (void)

{

P0MDOUT &= ~(1<<4); //Open-Drain pour T2

P0 |= (1<<4);

P0MDOUT &= ~(1<<5); //Open-Drain pour T2EX

P0 |= (1<<5);

P0MDOUT &= ~(1<<6); //Open-Drain pour T4

P0 |= (1<<6);

P0MDOUT &= ~(1<<7); //Open-Drain pour T4EX

P0 |= (1<<7);

P1MDOUT |= (1<<0); //Push -Pull pour SYSCLK

P1 |= (1<<0);

XBR1 |= (1<<5); //T2

XBR1 |= (1<<6); //T2EX

XBR2 |= (1<<3); //T4

XBR2 |= (1<<4); //T4EX

XBR1 |= (1<<7); //SYSCLK

P1MDOUT |= (1<<1);

P1 &= ~(1<<1);

}

//Modification de la CLK Systeme:

void Oscillator\_Init()

{

int i = 0;

OSCXCN = 0x67; // Crystal oscillator Mode – f> 6,7Mhz

for (i = 0; i < 3000; i++); // Wait 1ms for initialization

while ((OSCXCN & 0x80) == 0); // On attend que l’oscillateur à quartz soit stable

OSCICN = 0x0C; // Basculement sur l’oscillateur externe

}

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

void Config\_Timer4\_CounterBase(void)

{

T4CON &= ~(1<<7);

T4CON &= ~(1<<6);

T4CON &= ~(1<<5);

T4CON &= ~(1<<4);

T4CON &= ~(1<<3);

T4CON &= ~(1<<0);

T4CON |= (1<<1); //Moder Timer

RCAP4H = 0xFF;

RCAP4L = 0x9C;

T4 = RCAP4;

T4CON |= (1<<2);

EIP2 |= (1<<2);

EIE2 |= (1<<2);

}

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

void ISR\_Timer4 (void) interrupt 16

{

T4CON &= ~(1<<7);

SIGN\_OUT = !SIGN\_OUT;

}