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

// TP2\_BSE.c

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

// AUTH: LAJUGIE Rodolphe, CORNATON Maxime

// DATE:

//

// Target: C8051F02x

// Tool chain: KEIL Microvision5

//

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

// Fichiers d'entête

#include "intrins.h"

#include<c8051F020.h>

#include<c8051F020\_SFR16.h>

#include<TP2\_BSE\_Lib\_Config\_Globale.h>

#include<TP2\_BSE\_Lib\_Divers.h>

#include<TP2\_BSE\_Main.h>

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

// Déclaration des MACROS

#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 START 1

#define END 0

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

// Déclarations Registres et Bits de l'espace SFR

sbit LED = P1^6; // LED

sbit BP =P3^7; // Bouton Poussoir

sbit DECL\_EXTRN = P3^6;

sbit PROC\_USED = P3^2; //Passe a 1 durant les fcts d'inter.

sbit VISU\_INT7\_WIDTH = P2^4; //Interruption BP

sbit VISU\_INT\_TIMER2 = P3^5; //Signal impulsion tte les 10ms

bit STATE\_LED = LED\_BLINK;

int COUNT = 0;

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

// Variable globale

Event = PROCESSED;

void Config\_INT6(void);

void CFG\_Timer2\_BT (void);

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

// MAIN Routine

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

void main (void) {

// Configurations globales

Init\_Device();

// Configurations spécifiques

Config\_INT7(); // Configuration de INT7

Config\_INT6(); // Configuration de INT6

CFG\_Timer2\_BT();

// Fin des configurations

P2MDOUT |= (1<<4); //Config bit 2.4 en Push Pull

P74OUT |= (1<<5); //Config bit 6.4 en Push Pull

VISU\_INT7\_WIDTH = END;

P6 &= ~(1<<4); //VISU\_INT6\_WIDTH

P3MDOUT |= (1<<5);

VISU\_INT\_TIMER2 = 0;

EA = 1; // Validation globale des interruptions

// Boucle infinie

while(1)

{

// if (Event == TO\_BE\_PROCESSED)

// {

// Event = PROCESSED;

// STATE\_LED = !STATE\_LED;

// }

//

// if (STATE\_LED == LED\_BLINK)

// {

// LED = LED\_ON;

// Software\_Delay(2);

// LED = LED\_OFF;

// Software\_Delay(10);

// }

// else LED = LED\_OFF;

}

}

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

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\_WIDTH = START;

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

Event = TO\_BE\_PROCESSED;

VISU\_INT7\_WIDTH = END;

}

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

void Config\_INT6(void)

{

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

P3IF |= (1<<2); // IE6CF mis à 1 - sensibilité int6 front montant

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

{

P6 |= (1<<4); //VISU\_INT6\_WIDTH

P3IF ^= (1<<2); // IE6

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

Event = TO\_BE\_PROCESSED;

P6 &= ~(1<<4); //VISU\_INT6\_WIDTH

}

void CFG\_Timer2\_BT (void)

{

T2CON &= ~(1<<2); //Déctivation Timer2

CKCON |= (1<<5);

T2CON &= ~(1<<7); //Effacer le flag d’overflow TF2

T2CON &= ~(1<<6); //Effacer le flag d’overflow EXF2(pour éviter une interruption intempestive)

T2CON &= ~(1<<5); //On met à 0 RLCK0

T2CON &= ~(1<<4); //On met à 0 TLCK0

T2CON &= ~(1<<0); //Activation du rechargement

T2CON &= ~(1<<1); //Incrémentation de Timer2 selon la clock

T2CON &= ~(1<<3); //On ignore la trnasition sur front descendant

RCAP2H = 0xAE; //10ms sachant que clk = 2MHz/12Hz

RCAP2L = 0xE0; //2^16 - Nbr(10ms)

TL2 = 0x00;

TH2 = 0x00;

IP |= (1<<5);

IE |= (1<<5);

T2CON |= (1<<2);

}

void ISR\_INT\_TIMER2(void) interrupt 5

{

if (Event == TO\_BE\_PROCESSED)

{

Event = PROCESSED;

STATE\_LED = !STATE\_LED;

}

if (STATE\_LED == LED\_BLINK)

{

if(LED == LED\_ON){

if(COUNT == 1){

LED = LED\_OFF;

}else{

COUNT += 1;

}

}else{

if(COUNT == 10){

LED = LED\_ON;

COUNT = 0;

}else{

COUNT+=1;

}

}

}

else LED = LED\_OFF;

VISU\_INT\_TIMER2 = 1;

TF2 = 0;

VISU\_INT\_TIMER2 = 0;

}