#include "HT68F0021.h"

#define INIT\_LED1() (\_pac0 = \_pa0 = 0)

#define SET\_LED1(X) (\_pa0 = (X))

#define INIT\_LED2() (\_pac1 = \_pa1 = 0)

#define SET\_LED2(X) (\_pa1 = (X))

#define INIT\_LED3() (\_pac7 = \_pa7 = 0)

#define SET\_LED3(X) (\_pa7 = (X))

#define INIT\_LED4() (\_pac2 = \_pa2 = 0)

#define SET\_LED4(X) (\_pa2 = (X))

#define INIT\_LED5() (\_pac6 = \_pa6 = 0)

#define SET\_LED5(X) (\_pa6 = (X))

#define RESET\_LEDS() (\_pa0 = \_pa1 = \_pa7 = \_pa2 = \_pa6 = 0)

#define INT\_ISR\_ENABLE() (\_inte = 1)

#define INT\_ISR\_DISABLE() (\_inte = 0)

#define INT\_SET\_FLAG() (\_intf = 1)

#define INT\_CLEAR\_FLAG() (\_intf = 0)

#define EMI\_ENABLE() (\_emi = 1)

#define EMI\_DISABLE() (\_emi = 0)

#define INIT\_SWITCH() (\_pac5 = \_papu5 = 1)

unsigned char state;

unsigned char got\_intt;

volatile int timer\_cnt;

volatile int prev\_timer\_cnt;

volatile int button\_down;

volatile int sleep;

#pragma vector isr\_4 @ 0x04

void isr\_4(){

//\_delay(65000);

if(\_pa5 == 0) {

prev\_timer\_cnt = timer\_cnt = -1;

button\_down = 1;

}/\* else if(timer\_cnt - prev\_timer\_cnt > 3 && button\_down == 1){

button\_down = 0;

prev\_timer\_cnt = 0;

state = 6;

got\_intt = 1;

timer\_cnt = 0;

}\*/ else {

button\_down = 0;

if(sleep == 0){

got\_intt = 1;

} else {

sleep = 0;

}

prev\_timer\_cnt = timer\_cnt = -1;

}

} // external ISR

#pragma vector isr\_c @ 0x0C

void isr\_c(){

timer\_cnt ++;

if((timer\_cnt >= 300) || ((timer\_cnt > 2 && timer\_cnt < 10) && (\_pa5 == 0))) {

sleep = 1;

state = 6;

got\_intt = 1;

timer\_cnt = -1;

} else {

button\_down = 0;

}

} // timer isr

void mydelay(unsigned int times){

while(times--) \_delay(65000);

}

unsigned char leds[5];

void soft\_pwm(unsigned int l) {

unsigned char i,j;

for(i=0; i<l; i++){

for(j=0; j<255; j++){

SET\_LED1(leds[0] > j);

SET\_LED2(leds[1] > j);

SET\_LED3(leds[2] > j);

SET\_LED4(leds[3] > j);

SET\_LED5(leds[4] > j);

if(got\_intt != 0) {

return;

}

//\_delay(65000);

}

}

}

void main()

{

unsigned char leds2[5] = {0,0,0,0,0};

unsigned char step = 0;

unsigned char i = 0;

unsigned char j = 0;

unsigned char k = 0;

unsigned char stretch = 0;

unsigned char step5\_seq[6] = {36, 42, 49, 17, 10, 4};

INIT\_LED1();

INIT\_LED2();

INIT\_LED3();

INIT\_LED4();

INIT\_LED5();

//\_ints1 = 0; \_ints0 = 1; //rising edge trigger external interrupt

\_ints1 = 1; \_ints0 = 1; //rising edge and falling edge triggers external interrupt

\_intps = 0;

INT\_CLEAR\_FLAG();

INT\_ISR\_ENABLE();

EMI\_ENABLE();

\_intf = 0;

\_inte = 1;

\_emi = 1;

//////////////////////// Timer registers ///////////////////////////////////////////

\_tmr = 0;

// interrupt registers

\_emi = 1;

//\_tbe = 1;

\_te = 1;

// TMRC register

\_tm1 = 1;

\_tm0 = 0;

\_ton = 0; // disable timer

\_tpsc0 = 1;

\_tpsc1 = 1;

\_tpsc2 = 1;

\_ton = 1;

//////////////////////// Timer registers ///////////////////////////////////////////

INIT\_SWITCH();

\_wdtc = 0xA8;

//\_pas04 = 0;

//\_pawu5 = 0;

//\_inte = 0;

state = 0;

got\_intt = 0;

prev\_timer\_cnt = -1;

button\_down = 0;

timer\_cnt = -1;

sleep = 0;

RESET\_LEDS();

for(step = 0; step < 5; step ++){

leds[step] = 0;

}

while(1) {

if(got\_intt != 0){

mydelay(20);

state += 1;

if(state > 5) {

state = 0;

}

RESET\_LEDS();

for(step = 0; step < 5; step ++){

leds[step] = 0;

}

i = j = 0;

stretch = 1;

got\_intt = 0;

// button\_down = 0;

}

switch(state) {

case 0:

RESET\_LEDS();

break;

case 1:

//soft\_pwm(&leds);

//mydelay(2);

SET\_LED1(1);

SET\_LED2(1);

SET\_LED3(1);

SET\_LED4(1);

SET\_LED5(1);

break;

case 2:

stretch = 3;

case 3:

if(state == 3) {

stretch = 1;

}

for(k = 0; k < 248; k += 4){

for(i = 0; i < 5; i++) {

leds[i] = k;

}

soft\_pwm(stretch);

if(got\_intt != 0){

RESET\_LEDS();

break;

}

}

for(i = 0; i < 5; i++) {

leds[i] = k;

}

soft\_pwm(stretch);

for(k = 255; k > 5; k -= 4){

for(i = 0; i < 5; i++) {

leds[i] = k;

}

soft\_pwm(stretch);

if(got\_intt != 0){

RESET\_LEDS();

break;

}

}

break;

case 4:

for(k = 0; k < 240; k += 10){

leds[i] = k;

if(j != i) {

leds[j] = 255 - k;

}

soft\_pwm(1);

if(got\_intt != 0){

break;

}

}

if(got\_intt != 0){

break;

}

if(j != i) {

leds[j] = 0;

}

leds[i] = 255;

soft\_pwm(100);

j = i++;

if(i >= 5) {

i = 0;

}

break;

case 5:

for(k=0; k < 6; k++) {

j = (step5\_seq[k] & 32) >> 5;

for(i = 0; i <= 253; i += 2) {

for(step = 0; step < 5; step ++){

if((step5\_seq[k] & (1 << step)) > 0) {

leds[step] = (j == 1) ? i : (255 - i);

}

}

soft\_pwm(1);

if(got\_intt != 0){

break;

}

}

for(step = 0; step < 5; step ++){

if((step5\_seq[k] & (1 << step)) > 0) {

leds[step] = (j == 1) ? 255 : 0;

}

}

soft\_pwm(5);

if(got\_intt != 0){

break;

}

}

break;

}

}

After the power supply is applied to the microcontroller, it will wait for SW1 to pushed.

The product follows different LED sequence as follows

1.       **Pattern 1:**1st time SW1 is pressed => All LED’s are ON continuously

2.       **Pattern 2:**2nd time SW1 is pressed => All LED’s are blinking with gradual ON to OFF. All LED’s start from OFF condition and slowly go to max brightness and then slowly reduces brightness until it reaches zero. Basically LED will have a smooth transition (DIM in and DIM out). LED will reach full brightness in 2 sec and will reduce to zero brightness in another 2 sec. Total time as 4 sec.

3.       **Pattern 3:**3rd time SW1 is pressed => All LED’s will turn ON for 1sec and then will turn OFF for 1 sec. Instantaneously ON and OFF.

4.       **Pattern 4: 4**th time SW1 is pressed => LED are running in pattern as follows. LED brightness is gradually increasing until full ON and then gradually decreasing until OFF

         All LED’s are OFF

         LED 1 (OFF to ON in 1 sec)

         LED 2 (OFF to ON in 1 sec) & LED 1 (ON to OFF in 1 sec)

         LED 3 (OFF to ON in 1 sec) & LED 2 (ON to OFF in 1 sec)

         LED 4 (OFF to ON in 1 sec) & LED 3 (ON to OFF in 1 sec)

         LED 5 (OFF to ON in 1 sec) & LED 4 (ON to OFF in 1 sec)

         LED 1 (OFF to ON in 1 sec) & LED 5 (ON to OFF in 1 sec)

         Pattern continuous as long as switch is not pressed

5.       **Pattern 5: 5**th time SW1 is pressed => All transition in this mode are gradual or DIM i.e. LED will turn on slowly by changing brightness. The pattern is as follows.

         All LED’s are OFF

         LED 3 (OFF – ON in 1 sec)

         LED 2 and LED 4 (OFF – ON in 1 sec)

         LED 1 and LED 5 (OFF – ON in 1 sec)

         LED 1 and LED 5 (ON – OFF in 1 sec)

         LED 2 and LED 4 (ON – OFF in 1 sec)

         LED (ON – OFF in 1 sec) at this points all LED’s are OFF)

         Pattern continuous as long as switch is not pressed

6.       6th time SW1 is pressed => All LED’s are OFF

7.       7th time SW1 is pressed => LED’s follow pattern 1 as described in pattern 1 above

**Switch functionality**

1.       Pattern

a.       First time pressed after power ON, LED’s will follow pattern 1

b.      Second time switch is pressed, LED’s will follow pattern 2

c.       Third time switch is pressed, LED’s will follow pattern 3

d.      Fourth time switch is pressed, LED’s will follow pattern 4

e.      Fifth time switch is pressed, LED’s will follow pattern 5

f.        Sixth time the switch is pressed, all LED’s turn OFF. Pattern continues us if switch is pressed again

2.       Auto OFF

a.       Auto OFF in 15 minutes. This is applicable only when LED’s are ON and following the pattern as described.