```
#include <16F877.h>
#fuses HS,NOWDT,NOPROTECT,NOLVP
#use delay(clock=4000000)
#use rtos(timer=0,minor_cycle=10ms)
//definition of input output pins
#bit Yes=0x08.2
#bit NO=0x08.3
#bit col1=0x06.7//PortB possede l'adresse 06
#bit col2=0x06.6
#bit col3=0x06.5
#bit col4=0x06.4
#bit row1=0x06.3
#bit row2=0x06.2
#bit row3=0x06.1
#bit row4=0x06.0
#bit pulse=0x07.0//PortC location adress is 07
#bit spkr=0x07.1
#bit vol=0x07.2
#bit pol3=0x07.3
#bit pol1=0x08.0
#bit moled=0x05.4
#bit tled=0x09.0
#bit mled=0x09.1
#bit lbled=0x09.2
#bit select=0x08.1
//file to control the LCD mode 4 bits
#include "klcdpicm.c"
```

```
//variable needed for the main programm
unsigned int voltage;
int32 timeout;
int32 error=0,error1=0;
unsigned int go=1,flashconfig=1,config=2;
int8 sem;

#task(rate=40ms,max=10ms)
void store_test(void);
#task(rate=40ms,max=10ms)
void pulse_test(void);
#task(rate=40ms,max=10ms)
void flash_test(void);
#task(rate=40ms,max=10ms)
void speaker_test(void);
#task(rate=40ms,max=10ms)
void volume_test(void);
#task(rate=40ms,max=10ms)
void LNR_test(void);
#task(rate=40ms,max=10ms)
void memory_test(void);
#task(rate=40ms,max=10ms)
void monitor_led_test(void);
#task(rate=40ms,max=10ms)
void talk_led_test(void);
#task(rate=40ms,max=10ms)
void mute_led_test(void);
```

```
#task(rate=40ms,max=10ms)

void low_battery_test(void);

#task(rate=40ms,max=10ms)

void polarity_polarityled_test(void);

#task(rate=40ms,max=10ms)

void short_circuit_test(void);

#task(rate=40ms,max=10ms)

void clavier_test(void);

#task(rate=40ms,max=10ms)

void type_clavier(void);



void store_test(void)

{

rtos_wait(sem);

PORT_B_PULLUPS(true);//make that this line is not commented otherwise keypad will not tested correctly


select=0;

//delay_ms(1000);

timeout=0xFFF60000;//timout for 10s


//row1=0;row2=1;row3=1;row4=1;

output_low(PIN_B3);output_high(PIN_B2);output_high(PIN_B1);output_high(PIN_B0);

lcd_putc("\fPress STORE");

while((col1)&&(++timeout!=0));

error<<=1;

if (timeout!=0)
```

```
lcd_putc("\fSTORE");
else
error+=1;
rtos_signal(sem);
}

void pulse_test(void)
{
rtos_wait(sem);
//delay_ms(1000);
timeout=0xFFF60000;//timout for 10s
lcd_putc("\fPress Pulse");
while((pulse)&&(++timeout!=0));
error<<=1;
if (timeout!=0)
lcd_putc("\fPULSE");
else
error+=1;

rtos_signal(sem);
}

void flash_test(void)
{
rtos_wait(sem);

//delay_ms(1000);
```

```
timeout=0xFFF60000;//timout for 10s
row1=1;row2=1;row3=1;row4=0;
lcd_putc("\fPress Flash");
switch(flashconfig)
{
case 1:
{
//Flash case of J100
while((col2)&&(++timeout!=0)&&(col3)&&(col4));
error<<=3;
if ((timeout!=0))
lcd_putc("\fFLASH");
if (!col3)
error+=1;
else if(!col4)
error+=2;
else if (timeout==0)
error+=4;
}break;
case 2:
{
//Flash case of J270
while((col3)&&(++timeout!=0)&&(col2)&&(col4));
error<<=3;
if (timeout!=0)
lcd_putc("\fFLASH");
if (!col2)
```

```
error+=1;
else if(!col4)
error+=2;
else if (timeout==0)

error+=4;
}break;
case 3:
{
//Flash case of J600
while((col4)&&(++timeout!=0)&&(col2)&&(col3));
error<<=3;
if (timeout!=0)
lcd_putc("\fFLASH");
if (!col2)
error+=1;
else if(!col3)
error+=2;
else if (timeout==0)
error+=4;
}break;
}//end of switch case

rtos_signal(sem);
}
void speaker_test(void)
{
```

```
rtos_wait(sem);


//delay_ms(1000);
timeout=0xFFF60000;//timout for 10s
lcd_putc("\fPress SPKR");
while((spkr)&&(++timeout!=0));
error<<=1;
if (timeout!=0)
lcd_putc("\fSPKR");
else
error+=1;
rtos_signal(sem);
}
void volume_test(void)
{
rtos_wait(sem);


//delay_ms(1000);
timeout=0xFFF60000;//timout for 10s
lcd_putc("\fPress VOL");
while((vol)&&(++timeout!=0));
error<<=1;
if (timeout!=0)
lcd_putc("\fVOL");
else
error+=1;
```

```
rtos_signal(sem);
}

void LNR_test(void)
{
rtos_wait(sem);
PORT_B_PULLUPS(false);
//delay_ms(1000);
timeout=0xFFF60000;//timout for 10s
//row1=0;row2=1;row3=1;row4=1;
select=1;
output_high(PIN_B3);output_low(PIN_B2);output_low(PIN_B1);output_low(PIN_B0);
lcd_putc("\fPress LNR");

while((!col1)&&(++timeout!=0));
error<<=1;
if (timeout!=0)
lcd_putc("\fLNR");
else
error+=1;
//part to check the diode is not short circuit
//RB1 should configured as input and pulled low Col1 should be driven High check the RB1 if Low no faults


/*set_tris_b(0xD2);
output_high(PIN_B7);
lcd_putc("\fKeep Holding\nLNR");
```

```
delay_ms(500);
error<<=1;
if(row1) error++;
while(1);
set_tris_b(0xF0);
row1=row2=row3=row4=1;*/

rtos_signal(sem);
}
void memory_test(void)
{
rtos_wait(sem);

PORT_B_PULLUPS(false);
//delay_ms(1000);
timeout=0xFFF60000;//timout for 10s
select=1;
row1=0;row2=0;row3=0;row4=1;
lcd_putc("\fPress MEM");
while((!col4)&&(++timeout!=0));
error<<=1;
if (timeout!=0)
lcd_putc("\fMEM");
else
error+=1;
rtos_signal(sem);
}
```

```
void monitor_led_test(void)

{

//Monitor LED test

rtos_wait(sem);

delay_ms(1000);

lcd_putc("\fTesting LEDs...");

moled=0;

set_adc_channel(3);

delay_us(10);

voltage = read_adc();

//error<<=1; this is the first test so no need for shifting the error

if ((voltage<=150)&&(voltage>=130))//real value read from an other test not as mentionned on the data



{

lcd_putc("\nMonitor Led OK");

}

else

error+=1;

rtos_signal(sem);

}

void talk_led_test(void)

{

rtos_wait(sem);

delay_ms(1000);

//lcd_putc("\fTesting LEDs...");

moled=1;tled=0;

//output_low(PIN_E0);
```

```
//lcd_putc("\nTalk LED");

set_adc_channel(2);

delay_us(10);

voltage = read_adc();

error<<=1;

if ((voltage<=170)&&(voltage>=150))//real value read from an other test not as mentionned on the data


{

lcd_putc("\fTesting LEDs...\nTalk Led OK");

}

else

error+=1;

rtos_signal(sem);

}

void mute_led_test(void)

{

rtos_wait(sem);

delay_ms(1000);

//lcd_putc("\fTesting LEDs...");

//tled=1;mled=0;

output_high(PIN_E0);output_low(PIN_E1);

//lcd_putc("\nMute LED");

set_adc_channel(1);

delay_us(10);

voltage = read_adc();

error<<=1;
```

```
if ((voltage<=150)&&(voltage>=130))//real value read from an other test not as mentionned on the data


{
lcd_putc("\fTesting LEDs...\nMute Led OK");
}
else
error+=1;
}
void low_battery_test(void)
{
//Low Battery Test
delay_ms(1000);
//lcd_putc("\fTesting LEDs...");
//tled=1;mled=0;
output_high(PIN_E1);output_high(PIN_E2);
//lcd_putc("\nLow Battery LED");
set_adc_channel(0);
delay_us(10);
voltage = read_adc();
error<<=1;
if ((voltage>=80)&&(voltage<=100))//real value read from an other test not as mentionned on the data


{
lcd_putc("\fTesting LEDs...\nLow Batt Led OK");
}
```

```
else
error+=1;
delay_ms(1000);
output_low(PIN_E2);
rtos_signal(sem);
}

void polarity_polarityled_test(void)
{
rtos_wait(sem);
lcd_putc("\fHold POL");
pol1=0;//Pol Red Led
pol3=1;
timeout=0xFFF60000;//timout for 10s
set_adc_channel(4);
delay_us(10);
voltage = read_adc();
while((++timeout!=0)&&(!((voltage>=115)&&(voltage<=135))))//real value read from an other test not as

{
set_adc_channel(4);
delay_us(10);
voltage = read_adc();
}
error<<=1;
if (timeout!=0)
lcd_putc("\nRed Pol led ok");
```

```
else

error+=1;

//Polarity Second Test

lcd_putc("\nReversing Polarity");

pol3=0;

pol1=1;

timeout=0xFFF60000;//timout for 10s

set_adc_channel(4);

delay_us(10);

voltage = read_adc();

while((++timeout!=0)&&(!((voltage>=105)&&(voltage<=125))))//real value read from an other test not as


{

set_adc_channel(4);

delay_us(10);

voltage = read_adc();

}

error<<=1;

if (timeout!=0)

lcd_putc("\nGreen Pol led ok");

else

error+=1;

rtos_signal(sem);

}

void short_circuit_test(void)

{

rtos_wait(sem);
```

```
PORT_B_PULLUPS(true);
set_tris_b(0xFF);
error1<<=1;
if (input_b()!=0xFF)
error1+=1;
set_tris_b(0xFD);
output_low(PIN_B1);
error1<<=1;
if(row4==0)//short circuit row4-row3
{
error1+=1;
}
error1<<=1;
if(row2==0)//short circuit Row2-Row3
{
error1+=1;
}
set_tris_b(0xF7);
output_low(PIN_B3);
error1<<=1;
if(row2==0)//short circuit row2-row1
{
error1+=1;
}
error1<<=1;
if(col4==0)//short circuit row1-col4
{
```

```
error1+=1;
}
set_tris_b(0xDF);
output_low(PIN_B5);
error1<<=1;
if(col4==0)//short circuit col3-col4
{
error1+=1;
}
error1<<=1;
if(col2==0)//short circuit co12-col3
{
error1+=1;
}
set_tris_b(0x7F);
output_low(PIN_B7);
error1<<=1;
if(col2==0)//short circuit col2-col1
{
error1+=1;
}
rtos_signal(sem);
}

void clavier_test(void)
{
//test the keypad buttons
```

```
//starting with 1,2...#
//1
//delay_ms(1000);
rtos_wait(sem);
timeout=0xFFF60000;//timout for 10s
//row1=0;row2=1;row3=1;row4=1;

output_low(PIN_B3);output_high(PIN_B2);output_high(PIN_B1);output_high(PIN_B0);
lcd_putc("\fPress 1");
while((col2)&&(++timeout!=0));
error<<=1;
if (timeout!=0)
lcd_putc("\f1");
else
{
error+=1;
}
//2
//delay_ms(1000);
timeout=0xFFF60000;//timout for 10s
//row1=0;row2=1;row3=1;row4=1;
output_low(PIN_B3);output_high(PIN_B2);output_high(PIN_B1);output_high(PIN_B0);
lcd_putc("\fPress 2");
while((col3)&&(++timeout!=0));
error<<=1;
if (timeout!=0)
lcd_putc("\f2");
```

```
else
{
error+=1;
}
//3
//delay_ms(1000);
timeout=0xFFF60000;//timout for 10s
//row1=0;row2=1;row3=1;row4=1;
output_low(PIN_B3);output_high(PIN_B2);output_high(PIN_B1);output_high(PIN_B0);
lcd_putc("\fPress 3");
while((col4)&&(++timeout!=0));
error<<=1;
if (timeout!=0)
lcd_putc("\f3");
else
{
error+=1;
}
//4
//delay_ms(1000);
timeout=0xFFF60000;//timout for 10s
//row1=1;row2=0;row3=1;row4=1;
output_high(PIN_B3);output_low(PIN_B2);output_high(PIN_B1);output_high(PIN_B0);
lcd_putc("\fPress 4");
while((col1)&&(++timeout!=0));
error<<=1;
if (timeout!=0)
```

```
lcd_putc("\f4");
else
{
error+=1;
}
//5
//delay_ms(1000);
timeout=0xFFF60000;//timout for 10s
row1=1;row2=0;row3=1;row4=1;
lcd_putc("\fPress 5");
while((col2)&&(++timeout!=0));
error<<=1;
if (timeout!=0)
lcd_putc("\f5");
else
{

error+=1;
}
//6
//delay_ms(1000);
timeout=0xFFF60000;//timout for 10s
row1=1;row2=0;row3=1;row4=1;
lcd_putc("\fPress 6");
while((col3)&&(++timeout!=0));
error<<=1;
if (timeout!=0)
```

```
lcd_putc("\f6");
else
{
error+=1;
}
//7
//delay_ms(1000);
timeout=0xFFF60000;//timout for 10s
row1=1;row2=0;row3=1;row4=1;
lcd_putc("\fPress 7");
while((col4)&&(++timeout!=0));
error<<=1;
if (timeout!=0)
lcd_putc("\f7");
else
{
error+=1;
}
//8
//delay_ms(1000);
timeout=0xFFF60000;//timout for 10s
//row1=1;row2=1;row3=0;row4=1;
output_high(PIN_B3);output_high(PIN_B2);output_low(PIN_B1);output_high(PIN_B0);
lcd_putc("\fPress 8");
while((col1)&&(++timeout!=0));
error<<=1;
if (timeout!=0)
```

```
lcd_putc("\f8");
else
{
error+=1;
}
//9
//delay_ms(1000);
timeout=0xFFF60000;//timout for 10s
//row1=1;row2=1;row3=0;row4=1;
output_high(PIN_B3);output_high(PIN_B2);output_low(PIN_B1);output_high(PIN_B0);
lcd_putc("\fPress 9");
while((col2)&&(++timeout!=0));
error<<=1;
if (timeout!=0)
lcd_putc("\f9");
else
{
error+=1;
}
//*
//delay_ms(1000);
timeout=0xFFF60000;//timout for 10s
//row1=1;row2=1;row3=0;row4=1;
output_high(PIN_B3);output_high(PIN_B2);output_low(PIN_B1);output_high(PIN_B0);
lcd_putc("\fPress *");

while((col4)&&(++timeout!=0));
```

```
error<<=1;
if (timeout!=0)
lcd_putc("\f*");
else
{
error+=1;
}
//0
//delay_ms(1000);
timeout=0xFFF60000;//timout for 10s
//row1=1;row2=1;row3=0;row4=1;
output_high(PIN_B3);output_high(PIN_B2);output_low(PIN_B1);output_high(PIN_B0);
lcd_putc("\fPress 0");
while((col3)&&(++timeout!=0));
error<<=1;
if (timeout!=0)
lcd_putc("\f0");
else
{
error+=1;
}
//#
//delay_ms(1000);
timeout=0xFFF60000;//timout for 10s
row1=1;row2=1;row3=1;row4=0;
lcd_putc("\fPress #");
while((col1)&&(++timeout!=0));
```

```
error<<=1;
if (timeout!=0)
lcd_putc("\f#");
else
{
error+=1;
}
rtos_signal(sem);
}
void type_clavier(void)
{
rtos_wait(sem);
while(No&Yes);
while(((!Yes)||(!No))&&go)
{
if (!Yes)
{
while(No)
{
switch(config)
{
case 1:
{
lcd_putc("\fConfigured to:\n");
lcd_putc("Europe:J100");
flashconfig=1;
}break;
```

```
case 2:
{
lcd_putc("\fConfigured to:\n");
lcd_putc("French:J270");
flashconfig=2;
}break;
case 3:
{
lcd_putc("\fConfigured to:\n");
lcd_putc("USA:J600");
flashconfig=3;
config=0;
}break;
}
config+=1;
while((!Yes)&&(No));
while((Yes)&&(No));
}}
if((!No))
{
go=0;
}}
//confirmation
delay_ms(1000);
switch(flashconfig)
{
case 1:
```

```
{
lcd_putc("\fConfirmed to:\n");
lcd_putc("Europe:J100");
}break;
case 2:
{
lcd_putc("\fConfirmed to:\n");
lcd_putc("French:J270");
}break;
case 3:
{
lcd_putc("\fConfirmed to:\n");
lcd_putc("USA:J600");
}break;
}
rtos_signal(sem);
}
void main(void)
{
timeout=0xFFF60000;//timout for 10s
SETUP_ADC_PORTS(AN0_AN1_AN2_AN3_AN4);
SETUP_ADC(ADC_CLOCK_INTERNAL);
PORT_B_PULLUPS(true);//make sur that this line is not commented otherwise keypad will not be tested
sem=1;
go=1;
select=0;
pol1=1;pol3=1;
```

```
set_tris_c(0x07);//RC0,RC1,RC2 confgured as inputs Pluse,SPKR,Vol
set_tris_a(0x2F);
set_tris_e(0x00);
moled=1;tled=1;mled=1;lbled=0;
lcd_init();
lcd_putc("Keypad Tester\n");
lcd_putc("Kacey LTD V1.2");
//default configuration is for Europe J100
delay_ms(1000);
lcd_putc("\fConfigured to:\n");
lcd_putc("Europe:J100");
//menu to change the default configuration
delay_ms(1000);
lcd_putc("\fYes:change config\nNo:Default config");
//Start testing
delay_ms(1000);
lcd_putc("\fYes:Start Test");
while(1)
while((!Yes)||(!No))
{
if (!Yes)

{ error1=0;
error=0;
lcd_putc("\fShort Circuit Test");
short_circuit_test();
if (error1!=0) goto aborting;
```

```
set_tris_b(0xF0);
row1=row2=row3=row4=1;
lcd_putc("\fShort circuit Test\n Finished");
 monitor_led_test();//First test
 talk_led_test();
 mute_led_test();
 low_battery_test();
 polarity_polarityled_test();
 clavier_test();
 volume_test();
 speaker_test();
 flash_test();
 pulse_test();
 LNR_test();
 memory_test();
 store_test();//last test
//Test Finished
lcd_putc("\fTested:");
if (error==0)
lcd_putc("PASS");
else lcd_putc("FAIL");
lcd_putc("\nPress Yes or No");
goto finish;
aborting: lcd_putc("\fSC Failure:\n");
lcd_putH(error1);
lcd_gotoxy(9,2);
lcd_putc(" ");
```

```
finish:;
}
else if(!No)
if (error!=0)
{
lcd_putc("\fFailure code:");
lcd_putc("\n");
lcd_putH(error);
lcd_gotoxy(9,2);
lcd_putc(" ");
}//Case of Displaying error code
}
rtos_run();
}//Main
```