Pgm1

#include <stdio.h>

#include "NUC1xx.h"

#include "Driver\DrvSYS.h"

#include "Driver\DrvGPIO.h"

#include "NUC1xx-LB\_002\LCD\_Driver.h"

int main (void)

{

int32\_t number;

char TEXT0[16]="SmplKeypad";

char TEXT1[16];

UNLOCKREG(); // unlock register for programming

DrvSYS\_Open(48000000);// set System Clock to run at 48MHz

LOCKREG(); // lock register from programming

// Initialize LEDs (four on-board LEDs below LCD panel)

Initial\_panel();

clr\_all\_panel();

print\_lcd(0,TEXT0); // print title

while (1) // forever loop to keep flashing four LEDs one at a time

{

number=DrvGPIO\_GetPortBits(E\_GPA);

sprintf(TEXT1,"%x",number); // print scankey input to string

print\_lcd(1,TEXT1);

if(number==0xfffe)

print\_lcd(2,"A0");

else if(number==0xfffd)

print\_lcd(2,"A1");

else if(number==0xfffb)

print\_lcd(2,"A2");

else if(number==0xfff7)

print\_lcd(2,"A3");

else if(number==0xffef)

print\_lcd(2,"A4");

else if(number==0xffdf)

print\_lcd(2,"A5");

else if(number==0xffbf)

print\_lcd(2,"A6");

else if(number==0xff7f)

print\_lcd(2,"A7");

else if(number==0xfeff)

print\_lcd(2,"A8");

}

}

Pgm2

// Smpl\_GPIO\_EINT1 : External Interrupt pin to trigger interrupt //on GPB15, then Buzz INT1(GPB.15) pin INT0(GPB.14) pin

#include <stdio.h>

#include "NUC1xx.h"

#include "Driver\DrvGPIO.h"

#include "Driver\DrvSYS.h"

// External Interrupt Handler (INT button to trigger GPB15)

void EINT1Callback(void)

{

DrvGPIO\_ClrBit(E\_GPA,14); // GPB11 = 0 to turn on Buzzer

DrvSYS\_Delay(1000000); // Delay

DrvGPIO\_SetBit(E\_GPA,14); // GPB11 = 1 to turn off Buzzer

DrvSYS\_Delay(10000); // Delay

}

int main (void)

{

UNLOCKREG();

DrvSYS\_SetOscCtrl(E\_SYS\_XTL12M, 1); // external 12MHz Crystal

//DrvSYS\_Delay(5000); // delay for stable clock

DrvSYS\_SelectHCLKSource(0); // clock source = 12MHz Crystal

LOCKREG();

DrvGPIO\_Open(E\_GPA, 14, E\_IO\_OUTPUT); // initial GPIO pin GPB11 for controlling Buzzer

//0 External Interrupt

DrvGPIO\_Open(E\_GPB, 15, E\_IO\_INPUT); // configure external interrupt pin GPB15

DrvGPIO\_EnableEINT1(E\_IO\_BOTH\_EDGE, E\_MODE\_EDGE, EINT1Callback); // configure external interrupt

while(1)

{

}

}

Prg3

//

// Smpl\_ADC\_PWM : ADC7 to read VR1 resistance value, PWM0 output to control LED (GPA12)

//

#include <stdio.h>

#include "NUC1xx.h"

#include "LCD\_Driver.h"

void Delay(int count)

{

while(count--)

{

// \_\_NOP;

}

}

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

MAIN function

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

int32\_t main (void)

{

//Enable 12Mhz and set HCLK->12Mhz

char adc\_value[15]="ADC Value:";

UNLOCKREG();

SYSCLK->PWRCON.XTL12M\_EN = 1;

SYSCLK->CLKSEL0.HCLK\_S = 0;

LOCKREG();

InitPWM();

InitADC();

Initial\_panel(); //call initial pannel function

clr\_all\_panel();

/\* Synch field transmission & Request Identifier Field transmission\*/

while(1)

{

while(ADC->ADSR.ADF==0);

ADC->ADSR.ADF=1;

PWMA->CMR0=ADC->ADDR[7].RSLT<<4;

Show\_Word(0,11,' ');

Show\_Word(0,12,' ');

Show\_Word(0,13,' ');

sprintf(adc\_value+4,"%d",ADC->ADDR[7].RSLT);

print\_lcd(0, adc\_value);

Delay(20000);

ADC->ADCR.ADST=1;

}

}

Pgm 4

//

// Smpl\_GPIO\_LED1 : GPC12--15 GPA 12\_14 to control on-board LEDs

// low-active output to control Red LEDs

//

#include <stdio.h>

#include "NUC1xx.h"

#include "Driver\DrvGPIO.h"

#include "Driver\DrvSYS.h"

void Init\_LED() // Initialize GPIO pins

{

DrvGPIO\_Open(E\_GPC, 12, E\_IO\_OUTPUT); // GPC12 pin set to output mode

DrvGPIO\_Open(E\_GPC, 13, E\_IO\_OUTPUT); // Goutput Hi to turn off LED

DrvGPIO\_Open(E\_GPC, 14, E\_IO\_OUTPUT);

DrvGPIO\_Open(E\_GPC, 15, E\_IO\_OUTPUT);

}

int main (void)

{

UNLOCKREG(); // unlock register for programming

DrvSYS\_Open(48000000);// set System Clock to run at 48MHz

// 12MHz crystal input, PLL output 48MHz

LOCKREG(); // lock register from programming

Init\_LED(); // Initialize LEDs (four on-board LEDs below LCD panel)

while (1) // forever loop to keep flashing four LEDs one at a time

{

DrvGPIO\_SetPortBits(E\_GPC,0xffff0fff); // output Low to turn on LED

DrvSYS\_Delay(300000); // delay

DrvGPIO\_SetPortBits(E\_GPC,0xffffffff) ; // output Hi to turn off LED

DrvSYS\_Delay(300000); // delay

}

}

Pgm5

// Smpl\_7seg\_ADC7 : ADC7 to read and display on lcd

//

#include <stdio.h>

#include "NUC1xx.h"

#include "Driver\DrvSYS.h"

#include "Seven\_Segment.h"

#include "DrvADC.h"

#include "LCD\_Driver.h"

int32\_t main (void)

{ uint16\_t value;

char TEXT[16];

UNLOCKREG();

SYSCLK->PWRCON.XTL12M\_EN = 1; //Enable 12Mhz and set HCLK->12Mhz

SYSCLK->CLKSEL0.HCLK\_S = 0;

LOCKREG();

Initial\_panel(); // initialize LCD pannel

clr\_all\_panel(); // clear LCD panel

print\_lcd(0,"variable reistor");

DrvADC\_Open(ADC\_SINGLE\_END,ADC\_SINGLE\_OP , 0x40,INTERNAL\_HCLK , 1);

while(1)

{

DrvADC\_StartConvert(); // start A/D conversion

while(DrvADC\_IsConversionDone()==FALSE);

value = ADC->ADDR[6].RSLT & 0xFFF;

sprintf(TEXT,"Value: %d",value); // convert ADC0 value into text

print\_lcd(1, TEXT); // output TEXT to LCD

}

}

Pgm6

//

// Smpl\_GPIO\_EINT1 : External Interrupt pin to trigger lcd on interrupt //on GPB15,

#include <stdio.h>

#include "NUC1xx.h"

#include "Driver\DrvGPIO.h"

#include "Driver\DrvSYS.h"

// External Interrupt Handler (INT button to trigger GPB15)

void EINT1Callback(void)

{

print\_lcd(1,"hello");

print\_lcd(2,"A0");

print\_lcd(3,"A01");

print\_lcd(4,"A03");

}

int main (void)

{

UNLOCKREG();

DrvSYS\_SetOscCtrl(E\_SYS\_XTL12M, 1); // external 12MHz Crystal

//DrvSYS\_Delay(5000); // delay for stable clock

DrvSYS\_SelectHCLKSource(0); // clock source = 12MHz Crystal

LOCKREG();

Initial\_panel();

clr\_all\_panel();

//0 External Interrupt

DrvGPIO\_Open(E\_GPB, 15, E\_IO\_INPUT); // configure external interrupt pin GPB15

DrvGPIO\_EnableEINT1(E\_IO\_BOTH\_EDGE, E\_MODE\_EDGE, EINT1Callback); // configure external interrupt

while(1)

{

}

}

Pgm7

//

// Smpl\_GPIO\_EINT1 : External Interrupt pin to trigger interrupt //on GPB15, then Buzz INT1(GPB.11)

#include <stdio.h>

#include "NUC1xx.h"

#include "Driver\DrvGPIO.h"

#include "Driver\DrvSYS.h"

// External Interrupt Handler (INT button to trigger GPB15)

void EINT1Callback(void)

{

DrvGPIO\_ClrBit(E\_GPB,11); // GPB11 = 0 to turn on Buzzer

DrvSYS\_Delay(1000000); // Delay

DrvGPIO\_SetBit(E\_GPB,11); // GPB11 = 1 to turn off Buzzer

DrvSYS\_Delay(10000); // Delay

}

int main (void)

{

UNLOCKREG();

DrvSYS\_SetOscCtrl(E\_SYS\_XTL12M, 1); // external 12MHz Crystal

//DrvSYS\_Delay(5000); // delay for stable clock

DrvSYS\_SelectHCLKSource(0); // clock source = 12MHz Crystal

LOCKREG();

DrvGPIO\_Open(E\_GPB, 11, E\_IO\_OUTPUT); // initial GPIO pin GPB11 for controlling Buzzer

//0 External Interrupt

DrvGPIO\_Open(E\_GPB, 15, E\_IO\_INPUT); // configure external interrupt pin GPB15

DrvGPIO\_EnableEINT1(E\_IO\_BOTH\_EDGE, E\_MODE\_EDGE, EINT1Callback); // configure external interrupt

while(1)

{

}

}

Pgm8

//

// smpl\_GPIO\_Interrupt

//

// GPA15 to input interrupt

// GPD15 to input interrupt

#include <stdio.h>

#include "NUC1xx.h"

#include "Driver\DrvUART.h"

#include "Driver\DrvGPIO.h"

#include "Driver\DrvSYS.h"

#include "LCD\_Driver.h"

volatile uint32\_t irqA\_counter = 0;

volatile uint32\_t irqE\_counter = 0;

void GPIOAB\_INT\_CallBack(uint32\_t GPA\_IntStatus, uint32\_t GPB\_IntStatus)

{

if ((GPA\_IntStatus>>15) & 0x01) irqA\_counter++;

print\_lcd(3,"GPA interrupt !!");

}

void GPIOCDE\_INT\_CallBack(uint32\_t GPC\_IntStatus, uint32\_t GPD\_IntStatus, uint32\_t GPE\_IntStatus)

{

if ((GPE\_IntStatus>>15) & 0x01) irqE\_counter++;

print\_lcd(3,"GPC interrupt !!");

}

int32\_t main()

{

char TEXT[16];

UNLOCKREG();

SYSCLK->PWRCON.XTL12M\_EN=1;

DrvSYS\_Delay(5000); // Waiting for 12M Xtal stalble

SYSCLK->CLKSEL0.HCLK\_S=0;

LOCKREG();

// setup GPA15 & GPD15 to get interrupt input

DrvGPIO\_Open(E\_GPA,15,E\_IO\_INPUT);

DrvGPIO\_Open(E\_GPE,15,E\_IO\_INPUT);

DrvGPIO\_EnableInt(E\_GPA, 15, E\_IO\_RISING, E\_MODE\_EDGE);

DrvGPIO\_EnableInt(E\_GPE, 15, E\_IO\_RISING, E\_MODE\_EDGE);

DrvGPIO\_SetDebounceTime(5, 1);

DrvGPIO\_EnableDebounce(E\_GPA, 15);

DrvGPIO\_EnableDebounce(E\_GPE, 15);

DrvGPIO\_SetIntCallback(GPIOAB\_INT\_CallBack, GPIOCDE\_INT\_CallBack);

Initial\_panel();

clr\_all\_panel();

print\_lcd(0,"Smpl\_GPIO\_Intr");

while(1)

{

sprintf(TEXT,"IRQ\_A: %d",irqA\_counter);

print\_lcd(1, TEXT);

sprintf(TEXT,"IRQ\_E: %d",irqE\_counter);

print\_lcd(2, TEXT);

}

}

Pgm9

// Smpl\_GPIO\_EINT1 : External Interrupt pin to trigger interrupt //on GPB15, then Buzz INT1(GPB.15) pin INT0(GPB.14) pin

#include <stdio.h>

#include "NUC1xx.h"

#include "Driver\DrvGPIO.h"

#include "Driver\DrvSYS.h"

// External Interrupt Handler (INT button to trigger GPB15)

void EINT1Callback(void)

{

DrvGPIO\_ClrBit(E\_GPA,14); // GPB11 = 0 to turn on Buzzer

DrvSYS\_Delay(1000000); // Delay

DrvGPIO\_SetBit(E\_GPA,14); // GPB11 = 1 to turn off Buzzer

DrvSYS\_Delay(10000); // Delay

}

int main (void)

{

UNLOCKREG();

DrvSYS\_SetOscCtrl(E\_SYS\_XTL12M, 1); // external 12MHz Crystal

//DrvSYS\_Delay(5000); // delay for stable clock

DrvSYS\_SelectHCLKSource(0); // clock source = 12MHz Crystal

LOCKREG();

DrvGPIO\_Open(E\_GPA, 14, E\_IO\_OUTPUT); // initial GPIO pin GPB11 for controlling Buzzer

//0 External Interrupt

DrvGPIO\_Open(E\_GPB, 15, E\_IO\_INPUT); // configure external interrupt pin GPB15

DrvGPIO\_EnableEINT1(E\_IO\_BOTH\_EDGE, E\_MODE\_EDGE, EINT1Callback); // configure external interrupt

while(1)

{

}

}

\

Pgm10

//

// Smpl\_GPIO\_EINT1 : External Interrupt pin to trigger interrupt //on GPB15, then Buzz INT1(GPB.15) pin INT0(GPB.14) pin

#include <stdio.h>

#include "NUC1xx.h"

#include "Driver\DrvGPIO.h"

#include "Driver\DrvSYS.h"

// External Interrupt Handler (INT button to trigger GPB15)

void EINT1Callback(void)

{

DrvGPIO\_ClrBit(E\_GPA,12); // GPA12 = Blue, 0 : on, 1 : off

DrvGPIO\_ClrBit(E\_GPA,13); // GPA13 = Green, 0 : on, 1 : off

DrvGPIO\_ClrBit(E\_GPA,14); // GPA14 = Red, 0 : on, 1 : off

DrvSYS\_Delay(1000000);

DrvGPIO\_SetBit(E\_GPA,12); // GPA12 = Blue, 0 : on, 1 : off

DrvGPIO\_SetBit(E\_GPA,13); // GPA13 = Green, 0 : on, 1 : off

DrvGPIO\_SetBit(E\_GPA,14); // GPA14 = Red, 0 : on, 1 : off

}

void Init\_LED()

{

// initialize GPIO pins

DrvGPIO\_Open(E\_GPA, 12, E\_IO\_OUTPUT); // GPA12 pin set to output mode

DrvGPIO\_Open(E\_GPA, 13, E\_IO\_OUTPUT); // GPA13 pin set to output mode

DrvGPIO\_Open(E\_GPA, 14, E\_IO\_OUTPUT); // GPA14 pin set to output mode

// set GPIO pins output Hi to disable LEDs

DrvGPIO\_SetBit(E\_GPA, 12); // GPA12 pin output Hi to turn off Blue LED

DrvGPIO\_SetBit(E\_GPA, 13); // GPA13 pin output Hi to turn off Green LED

DrvGPIO\_SetBit(E\_GPA, 14); // GPA14 pin output Hi to turn off Red LED

}

int main (void)

{

UNLOCKREG();

DrvSYS\_SetOscCtrl(E\_SYS\_XTL12M, 1); // external 12MHz Crystal

//DrvSYS\_Delay(5000); // delay for stable clock

DrvSYS\_SelectHCLKSource(0); // clock source = 12MHz Crystal

LOCKREG();

// initial GPIO pin GPB11 for controlling Buzzer

//0 External Interrupt

DrvGPIO\_Open(E\_GPB, 15, E\_IO\_INPUT); // configure external interrupt pin GPB15

DrvGPIO\_EnableEINT1(E\_IO\_BOTH\_EDGE, E\_MODE\_EDGE, EINT1Callback); // configure external interrupt

while(1)

{

}

}

or

//

// Smpl\_GPIO\_RGBled : GPA12,13,14 output control RGB LED

// output low to enable LEDs

#include <stdio.h>

#include "NUC1xx.h"

#include "Driver\DrvGPIO.h"

#include "Driver\DrvUART.h"

#include "Driver\DrvSYS.h"

// Initial GPIO pins (GPA 12,13,14) to Output mode

void Init\_LED()

{

// initialize GPIO pins

DrvGPIO\_Open(E\_GPA, 12, E\_IO\_OUTPUT); // GPA12 pin set to output mode

DrvGPIO\_Open(E\_GPA, 13, E\_IO\_OUTPUT); // GPA13 pin set to output mode

DrvGPIO\_Open(E\_GPA, 14, E\_IO\_OUTPUT); // GPA14 pin set to output mode

// set GPIO pins output Hi to disable LEDs

DrvGPIO\_SetBit(E\_GPA, 12); // GPA12 pin output Hi to turn off Blue LED

DrvGPIO\_SetBit(E\_GPA, 13); // GPA13 pin output Hi to turn off Green LED

DrvGPIO\_SetBit(E\_GPA, 14); // GPA14 pin output Hi to turn off Red LED

}

int main (void)

{

UNLOCKREG(); // unlock register for programming

DrvSYS\_Open(48000000); // set System Clock to run at 48MHz (PLL with 12MHz crystal input)

LOCKREG(); // lock register from programming

Init\_LED();

while (1)

{

// GPA12 = Blue, 0 : on, 1 : off

// GPA13 = Green, 0 : on, 1 : off

// GPA14 = Red, 0 : on, 1 : off

// set RGBled to Blue

DrvGPIO\_ClrBit(E\_GPA,12); // GPA12 = Blue, 0 : on, 1 : off

DrvGPIO\_SetBit(E\_GPA,13);

DrvGPIO\_SetBit(E\_GPA,14);

DrvSYS\_Delay(1000000);

// set RGBled to Green

DrvGPIO\_SetBit(E\_GPA,12);

DrvGPIO\_ClrBit(E\_GPA,13); // GPA13 = Green, 0 : on, 1 : off

DrvGPIO\_SetBit(E\_GPA,14);

DrvSYS\_Delay(1000000);

// set RGBled to Red

DrvGPIO\_SetBit(E\_GPA,12);

DrvGPIO\_SetBit(E\_GPA,13);

DrvGPIO\_ClrBit(E\_GPA,14); // GPA14 = Red, 0 : on, 1 : off

DrvSYS\_Delay(1000000);

// set RGBled to off

DrvGPIO\_SetBit(E\_GPA,12); // GPA12 = Blue, 0 : on, 1 : off

DrvGPIO\_SetBit(E\_GPA,13); // GPA13 = Green, 0 : on, 1 : off

DrvGPIO\_SetBit(E\_GPA,14); // GPA14 = Red, 0 : on, 1 : off

DrvSYS\_Delay(1000000);

}

}

Pgm11

//

// Smpl\_7seg : counting from 0 to 9999 and display on 7-segment LEDs

//

#include <stdio.h>

#include "NUC1xx.h"

#include "DrvSYS.h"

#include "Seven\_Segment.h"

#include "Driver\DrvGPIO.h"

// display an integer on four 7-segment LEDs

void seg\_display(int16\_t value)

{

int8\_t digit;

digit = value / 1000;

close\_seven\_segment();

show\_seven\_segment(3,digit);

DrvSYS\_Delay(5000);

value = value - digit \* 1000;

digit = value / 100;

close\_seven\_segment();

show\_seven\_segment(2,digit);

DrvSYS\_Delay(5000);

value = value - digit \* 100;

digit = value / 10;

close\_seven\_segment();

show\_seven\_segment(1,digit);

DrvSYS\_Delay(5000);

value = value - digit \* 10;

digit = value;

close\_seven\_segment();

show\_seven\_segment(0,digit);

DrvSYS\_Delay(5000);

}

int32\_t main (void)

{

int32\_t i =0;

UNLOCKREG();

DrvSYS\_Open(48000000);

LOCKREG();

while(i<10000)

{

seg\_display(i); // display i on 7-segment display

DrvSYS\_Delay(10000); // delay for keeping display

i++; // increment i

}

}

Pgm12