

## 1.Keypad

```
// Initial C Source File
#include "xstdsys.h"

unsigned int i,delay_ms,segval;
unsigned char index;
unsigned char lcdval,row,keyscan,keyret,keynum=0;
unsigned char keypress,scanret = 0xFF;

// This is the keypress data you need to show Hexto7Segment
unsigned char seg7[] =
{0x3f,0x06,0x5b,0x4f,0x66,0x6d,0x7d,0x07,0x7f,0x67,0x77,0x7c,0x39,0x
5e,0x79,0x71,0x00,0x00,0x00};
// This is the keyboard scan codes to output
unsigned char scan[] = {0xEF,0xDF,0xBF,0x7F,0x00} ;
// This is the response when user presses a key for each row
unsigned char keycode[] = {0xEE,0xED,0xEB,0xE7,0xDE,0xDD,0xDB,0xD7,
0xBE,0xBD,0xBB,0xB7,0x7E,0x7D,0x7B,0x77,0x00};

void InitLPC(void)
{
    //Note H1 is used for MKBD and H2 for 7 seg display for
    ease of programming
    PINSEL0 = 0x00L;
    g_pGPIO0->IODIR = 0xFFFFFFFF0;
}

void Delay(unsigned int dms)
{
    delay_ms = dms;
    while(delay_ms > 0)
    {
        delay_ms--;
    }
}

void GetKey()
{
    row=0;
    while(1)
    {
        g_pGPIO0->IOCLR = 0xFF;
        row &= 0x3;
        keyscan=scan[row];
        g_pGPIO0->IOSET = keyscan;
        Delay(2);
        keyret = g_pGPIO0->IOPIN;
        if (keyscan != keyret)
            break;
        row++;
    }
    for(i=0;i<0x10;i++)
    {
        if(keycode[i]==keyret)
```

```

        keynum=i;
    }
    // Connect 7 Segment on H2 with select pin grounded
    g_pGPIO0->IOCLR = 0xFF00;
    segval = seg7[keynum];
    segval <= 8;
    g_pGPIO0->IOSET = segval;
}

void main(void)
{
    InitLPC();
    index=0;
    //PORTD=seg7[keynum];
    while(1)
    {
        GetKey();
    }
}

```

## 2.LCD

#include "xstdsys.h"  
 //Please note that you have to connect H1 to LCD data and H2 to LCD control

```

unsigned int cmd8[] = {0X38,0x38,0x0E,0x02,0x01,0x00};
unsigned int msg[] =
{'H','e','l','l','o',0x20,'A','D','M',0x20,0x00};
unsigned int lcdval,index,delay_ms;

```

```

void InitLPC(void)
{
    PINSEL0 = 0x00L;
    g_pGPIO0->IODIR = 0xFFFFFFFF;
}

```

```

void Delay(unsigned int dms)
{
    delay_ms = dms;
    while(delay_ms > 0)
    {
        delay_ms--;
    }
}

```

```

void InitLCD()
{
    index=0;
    lcdval=cmd8[index];
    while(lcdval !=0x0)
    {
        g_pGPIO0->IOSET = lcdval;
        lcdval |= 0x400;
        g_pGPIO0->IOSET = lcdval;
        Delay(500);
    }
}

```

```

        g_pGPIO0->IOCLR=0xFFFF;
        index++;
        lcdval=cmd8[index];
    }

}

void ShowMsg()
{
    index=0;
    lcdval=msg[index];
    while(lcdval !=0x0)
    {
        g_pGPIO0->IOSET = lcdval;
        lcdval |= 0x500;
        g_pGPIO0->IOSET = lcdval;
        Delay(500);
        g_pGPIO0->IOCLR=0xFFFF;
        index++;
        lcdval=msg[index];
    }
}

```

```

void main(void)
{
    InitLPC();
    index=0;
    while(1)
    {
        InitLCD();
        ShowMsg();
        Delay(5000);
    }
}

```

### 3. Blinking leds

```

/// Initial C Source File
#include "xstdsys.h"
// Connect From P0.0 ,P0.8 ,P0.16 ,P0.24 to Led
// Define all your variables here
unsigned int led_val,delay_ms;
unsigned int
mvrigh[]={0x80808080,0x40404040,0x20202020,0x10101010,0x08080808,0x
04040404,0x02020202,0x01010101,0x00};

unsigned char index;

void InitLPC(void)
{
    PINSEL0 = 0x00L;
    g_pGPIO0->IODIR = 0xFFFFFFFF;
}

```

```

}
void Delay(unsigned int dms)
{
    delay_ms = dms;
    while(delay_ms > 0)
        delay_ms--;
}
main()
{
    index=0;
    InitLPC();
    while(1)
    {
        index &= 0x7;
        led_val = mvright[index++];
        g_pGPIO0->IOSET =led_val;
        Delay(20000);
        g_pGPIO0->IOCLR=0xFFFFFFFF;
    }
}

```

## 4. Counter

```

#include "lpc214x.h"
#include "stdint.h"
#define IO1 0x10000
#define IO2 0x20000
#define IO3 0x40000
#define IO4 0x80000
#define IOX 0xF0000
#define IOXcl 0xFFFF

//Multiplexed 7segment Display
int count=0x0000;
unsigned int d0,d1,d2,d3;
unsigned char seg[] =
{0x3f,0x06,0x5b,0x4f,0x66,0x6d,0x7d,0x07,0x7f,0x67,0x77,0x7c,0x39,0x5e,
0x79,0x71,0x00};

void init_gpio()
{
    PINSEL0 = 0x00000000;
    PINSEL1 = 0x00000000;
    PINSEL2 = 0x00000000;

    IO0DIR = 0xFFFFFFFF;
    IO1DIR = 0xFFFFFFFF;
}

void delay()
{

```

```

int c = 100000;
while(c) //while count is more than zero loop
{
    c--;
}

void show_disp()
{
    //Digit 3
    d3 = count & 0x0F000;
    d3 >>= 12;
    IO0CLR = IOXcl;
    IO0SET= seg[d3]; //Willdisplay data 1 on 7seg
    IO1SET = IOX;      //ALL display are OFF
    IO1CLR = IO4;      //Display1 is made on
    delay();
    IO1SET = IOX;      //ALL display are OFF

    //Digit 2
    d2 = count & 0x0F00;
    d2 >>= 8;
    IO0CLR = IOXcl;
    IO0SET= seg[d2]; //Willdisplay data 2 on 7seg
    IO1SET = IOX;      //ALL display are OFF
    IO1CLR = IO3;      //Display1 is made on
    delay();
    IO1SET= IOX;      //ALL display are OFF

    //Digit 1
    d1 = count & 0x00F0;
    d1 >>= 4;
    IO0CLR = IOXcl;
    IO0SET = seg[d1]; //Willdisplay data 3 on 7seg

    IO1SET = IOX;      //ALL display are OFF
    IO1CLR = IO2;      //Display1 is made on
    delay();

    IO1SET = IOX;      //ALL display are OFF

    //Digit 0
    d0 = count & 0x000F;
    IO0CLR = IOXcl;
    IO0SET = seg[d0]; //Will display data 4 on 7seg
    IO1SET = IOX;      //ALL display are OFF
    IO1CLR = IO1;      //Display1 is made on
    delay();
    IO1SET = IOX;      //ALL display are OFF
}

int main( void )
{

    init_gpio();
    while(1)

```

```

    {
        show_disp();
        count++;
        count &= 0xFFFF;
    }
}

```

## 5. DAC programs

### //Sine

```

#include "lpc214x.h"
#include "stdint.h"
void delay_ms(uint16_t j)
{
    uint16_t x,i;
    for(i=0;i<j;i++)
    {
        for(x=0; x<6000; x++);    /* loop to generate 1 milisecond delay
with Cclk = 60MHz */
    }
}
int main (void)
{
    uint16_t value;
    uint16_t i = 0;
    uint16_t sintable[64]=      {512,562,611,660,707,753,796,836,
873,907,937,963,984,1001,1013,1021,
1023,1021,1013,1001,984,963,937,907,
873,836,796,753,707,660,611,562,
512,461,412,363,316,270,227,187,
150,116,86,60,39,22,10,2,
0,2,10,22,39,60,86,116,
150,187,227,270,316,363,412,461 };

    PINSEL1 = 0x00080000; /* P0.25 as DAC output */
    IO0DIR = 0xFFFFFFFF; /* Input pins for switch. P0.8 sine, P0.9
triangular, P0.10 sawtooth, P0.11 square */
    while(1){
        while(i<64)
        {

            value=(sintable[i]*50);

            DACR=value;
            delay_ms(1);
            i++;
        }
        i=0;
    }
}

```

## //triangle

```
#include "lpc214x.h"
#include "stdint.h"
void delay_ms(uint16_t j)
{
    uint16_t x,i;
    for(i=0;i<j;i++)
    {
        for(x=0; x<6000; x++);    /* loop to generate 1 milisecond delay
with Cclk = 60MHz */
    }
}
```

```
int main (void)
{
    uint16_t value;
    uint16_t i = 0;

    PINSEL1 = 0x00080000; /* P0.25 as DAC output */
    IO0DIR = 0xFFFFFFFF; /* Input pins for switch. P0.8 sine, P0.9
triangular, P0.10 sawtooth, P0.11 square */
    while(1)
    {
        i=0;
        while(i!=1023)
        {
            DACR=i*50;
            i++;
        }

        i=1023;
        while(i!=0)
        {
            DACR=i*50;
            i--;
        }
    }
}
```

## //SAWTOOTH

```
#include "lpc214x.h"
#include "stdint.h"
void delay_ms(uint16_t j)
{
    uint16_t x,i;
    for(i=0;i<j;i++)
    {
        for(x=0; x<6000; x++);    /* loop to generate 1 milisecond delay
with Cclk = 60MHz */
    }
}
```

```
int main (void)
{
    uint16_t value;
    uint16_t i = 0;
```

```

    PINSEL1 = 0x00080000; /* P0.25 as DAC output */
    IO0DIR = 0xFFFFFFFF; /* Input pins for switch. P0.8 sine, P0.9
triangular, P0.10 sawtooth, P0.11 square */
    while(1)
    {
        i=0;
        while(i!=1023)
        {
            DACR=i*50;
            i++;
        }
        DACR=0;
    }
}

```

## //SQUARE

```

#include "lpc214x.h"
#include "stdint.h"
void delay_ms(uint16_t j)
{
    uint16_t x,i;
    for(i=0;i<j;i++)
    {
        for(x=0; x<6000; x++); /* loop to generate 1 milisecond delay
with Cclk = 60MHz */
    }
}
int main (void)
{
    uint16_t value;
    uint16_t i = 0;

    PINSEL1 = 0x00080000; /* P0.25 as DAC output */
    IO0DIR = 0xFFFFFFFF; /* Input pins for switch. P0.8 sine, P0.9
triangular, P0.10 sawtooth, P0.11 square */
    while(1)
    {
        DACR=1023*50;
        delay_ms(10);
        DACR=0;
        delay_ms(10);
    }
}

```

## Factorial – thumb

```

NAME main
PUBLIC __iar_program_start
SECTION datas:CONST(2)

```



```

num
    dc32 0x05
fact
    ds32 1

    SECTION .intvec:CODE:NOROOT (2)
__iar_program_start
main
    ldr r6,num
    ldr r0,=thumbcode
    blx r0
    str r6,fact

stop b stop

    code16
thumbcode
    mov r4,r6
    loop sub r4,#1
    beq here
    mul r6,r4
    b loop
here
    bx lr
END

```

## Sum of squares

```

    NAME main
    PUBLIC __iar_program_start
    SECTION datas:CONST(2)

num
    dc32 0x0A
sum
    ds32 1

    SECTION .intvec:CODE:NOROOT (2)
__iar_program_start
main
    ldr r6,num
    ldr r0,=thumbcode
    mov lr,pc
    bx r0
    str r1,sum

stop b stop
    code16
thumbcode
    mov r4,r6
    mov r1,#0x00
loop
    mul r6,r4
    add r1,r1,r6
    sub r4,#1

```

```

        mov r6,r4
        bne loop
here     bx lr
end

```

## Smallest and Largest

```

NAME     main
PUBLIC   __iar_program_start
SECTION  .data:CONST:NOROOT (2)

num
        dc32 0x03,0x01,0x09,0x04,0x08,0x0a
numend
        dc32 0x00
len
        dc32 (numend-num)/4
SECTION  .intvec:CODE:NOROOT (2)

__iar_program_start
main
        code32
                ldr r1,=num
                ldr r2,len
                sub r2,r2,#1
                ldr r3,num ; smallest
                ldr r4,num;largest
                ldr r0,=thumbcode
                mov lr,pc
                bx r0

stop    b stop
        code16
thumbcode
loop    add r1,#4
                ldr r5,[r1,#0]
                cmp r5,r3
                bpl subss
                mov r3,r5
                ;b l1
subss   cmp r5,r4
                bmi l1
                mov r4,r5
l1      sub r2,#1
                cmp r2,#0
                bne loop
                bx lr

        END

```