

Introduction to Embedded System Design

Lab Report

- Lab date : 2023-5-17 (year-month-day)
- Group number : _____
- Group members: (student ID) (name)

410921365 陳祁鎔

410921304 張芯翎

1. Lab Title:

倒計時器

2. Lab Goal:

結合七段顯示器和蜂鳴器，實現倒數計時器的功能。

3. Lab Description and Steps:

1. 首先使用者按下要倒數的數字，限制兩位數，七段顯示器會顯示該數字，並且蜂鳴器會響兩聲。

2. 開始倒計時後，七段顯示器會顯示還剩多少時間。

3. 當數到零後，七段顯示器會顯示 0 且蜂鳴器會響三聲，代表時間到。

4. 按下 reset 即可重新開始。

4. Code:

```
#include <stdio.h>
#include "NUC100Series.h"
#include "MCU_init.h"
#include "SYS_init.h"
#include "Seven_Segment.h"
```

```
#include "LCD.h"
#include "Scankey.h"

volatile uint8_t ledState[4] = {0, 0, 0, 0};
int count = 0;
void TMR0_IRQHandler(void)
{
    ledState[0] = ~ledState[0]; // changing ON/OFF state
    if (ledState[0])
        PC12 = 0;
    else
        PC12 = 1;
    TIMER_ClearIntFlag(TIMER0); // Clear Timer0 time-out interrupt flag
}

void TMR1_IRQHandler(void)
{
    ledState[1] = ~ledState[1]; // changing ON/OFF state
    if (ledState[1])
        PC13 = 0;
    else
        PC13 = 1;
    TIMER_ClearIntFlag(TIMER1); // Clear Timer1 time-out interrupt flag
}

void TMR2_IRQHandler(void)
{
    ledState[2] = ~ledState[2]; // changing ON/OFF state
    if (ledState[2])
        PC14 = 0;
    else
        PC14 = 1;
    TIMER_ClearIntFlag(TIMER2); // Clear Timer2 time-out interrupt flag
}
```

```
void TMR3_IRQHandler(void)
{
    ledState[3] = ~ledState[3]; // changing ON/OFF state
    if (ledState[3])
    {
        PC15 = 0;
        count--;
    }
    else
        PC15 = 1;
    TIMER_ClearIntFlag(TIMER3); // Clear Timer3 time-out interrupt flag
}

void Init_Timer0(void)
{
    TIMER_Open(TIMER0, TMR0_OPERATING_MODE, TMR0_OPERATING_FREQ);
    TIMER_EnableInt(TIMER0);
    NVIC_EnableIRQ(TMR0_IRQn);
    TIMER_Start(TIMER0);
}

void Init_Timer1(void)
{
    TIMER_Open(TIMER1, TMR1_OPERATING_MODE, TMR1_OPERATING_FREQ);
    TIMER_EnableInt(TIMER1);
    NVIC_EnableIRQ(TMR1_IRQn);
    TIMER_Start(TIMER1);
}

void Init_Timer2(void)
{
    TIMER_Open(TIMER2, TMR2_OPERATING_MODE, TMR2_OPERATING_FREQ);
    TIMER_EnableInt(TIMER2);
    NVIC_EnableIRQ(TMR2_IRQn);
    TIMER_Start(TIMER2);
}
```

```
}
```

```
void Init_Timer3(void)
```

```
{
```

```
    TIMER_Open(TIMER3, TMR3_OPERATING_MODE, TMR3_OPERATING_FREQ);
```

```
    TIMER_EnableInt(TIMER3);
```

```
    NVIC_EnableIRQ(TMR3_IRQn);
```

```
    TIMER_Start(TIMER3);
```

```
}
```

```
void Buzz(int8_t no)
```

```
{
```

```
    while (no != 0)
```

```
    {
```

```
        PB11 = 0;
```

```
        CLK_SysTickDelay(50000);
```

```
        PB11 = 1;
```

```
        CLK_SysTickDelay(50000);
```

```
        no--;
```

```
    }
```

```
}
```

```
void Init_Buzz(void)
```

```
{
```

```
    GPIO_SetMode(PB, BIT11, GPIO_PMD_OUTPUT);
```

```
    PB11 = 1;
```

```
}
```

```
void Display_7seg(uint16_t value)
```

```
{
```

```
    uint8_t digit;
```

```
    digit = value / 1000;
```

```
    CloseSevenSegment();
```

```
    ShowSevenSegment(3, digit);
```

```
    CLK_SysTickDelay(5000);
```

```
value = value - digit * 1000;
digit = value / 100;
CloseSevenSegment();
ShowSevenSegment(2, digit);
CLK_SysTickDelay(5000);
```

```
value = value - digit * 100;
digit = value / 10;
CloseSevenSegment();
ShowSevenSegment(1, digit);
CLK_SysTickDelay(5000);
```

```
value = value - digit * 10;
digit = value;
CloseSevenSegment();
ShowSevenSegment(0, digit);
CLK_SysTickDelay(5000);
```

```
}
```

```
int main(void)
```

```
{
```

```
    int i, j;
```

```
    uint16_t key;
```

```
    SYS_Init(); // Intialize System/Peripheral clocks & multi-function I/Os
```

```
    Init_Buzz();
```

```
    GPIO_SetMode(PC, (BIT12 | BIT13 | BIT14 | BIT15), GPIO_MODE_OUTPUT); // set LED GPIO pin
```

```
    OpenKeyPad(); // for keypad
```

```
    OpenSevenSegment();
```

```
    for (i = 0; i < 2; i++)
```

```
    {
```

```
        key = ScanKey();
```

```
        while (!key)
```

```
            key = ScanKey();
```

```
        count = count * 10 + key;
```

```
        CLK_SysTickDelay(250000);
    }
    Buzz(2);
    Init_Timer3();
    while (1)
    {
        for (i = 0; i < 10; i++)
            if (ScanKey())
                count = 0;
        for (i = 0; i < 15; i++)
            Display_7seg(count);
        if (count == 0)
        {
            Buzz(3);
            break;
        }
    }
}
```

5. Lessons:

這次的實驗相比前一次的要簡單許多，比較困難部分在於要抓頻率，剛好對到頻率才能把數字輸入進去，因此用了 CLK 延遲 250000，才能抓到輸入進去的數字。