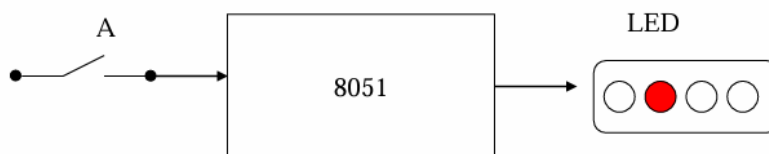


Lab04 Final Report

B1029009 簡鈺晴

- Problem description

- Control the LED display by pressing **a** button
 - One button only!



- Only **C programming** is allowed for this lab!
- Basic Part: (25%)
 - **Two buttons**, one for 1-bit left shift and one for 1-bit right shift
 - Initialize the LED with one bit set to 1, and others to 0
- Advanced Part: (65%)
 - The LED runs 4 modes (Just like what we did before)
 - As **a button** is pressed, the mode changes. Mode-changing sequence: Mode 0 → Mode 1 → Mode 2 → Mode 3 → Mode 0 ... (**Mode 0-3: change LED pattern every sec.** Refer to Lab2-3)
 - Hint: use the **timer interrupt** to make two I/O devices work simultaneously

- Code and explanations

Default: basic and bonus just different in time setting
(Left is basic, Right is bonus)

```
//set port address
void port_Configuration (){
    XBR2 = 0xc0;
    P1MDIN = 0xff;
    P2MDOUT = 0xff;
} //end of function Port_Configuration

void default_Config (){
    //turn-off watch-dog timer
    // disable watchdog timer
    WDTCON = 0xde;
    WDTCON = 0xad;

    //initialize SFR setup page
    SFRPAGE = CONFIG_PAGE;

    port_Configuration ();

    //set to normal mode
    SFRPAGE = LEGACY_PAGE;
} //end of function Default_Config

void Port_Configuration (){
    XBR2 = 0xc0;
    P1MDIN = 0xff;
    P2MDOUT = 0xff;
} //end of function Port_Configuration

void Timer_Configuration (){
    TMOD = 0x01;
    TCON = 0x10;
    CKCON = 0x10;

    IE = 0x82;
    TLO = 0;
    TH0 = 0;
} //end of function Timer_Configuration

void default_Config (){
    //turn-off watch-dog timer
    // disable watchdog timer
    WDTCON = 0xde;
    WDTCON = 0xad;

    OSCICN = 0x83;
    CLKSEL = 0x00;
    //initialize SFR setup page
    SFRPAGE = CONFIG_PAGE;
    Port_Configuration ();
    //set to normal mode
    SFRPAGE = LEGACY_PAGE;
    Timer_Configuration ();
} //end of function Default_Config
```

Button Detect:(Left is basic, Right is bonus)

```

void button_detect () {
    char key_hold;
    int key_release;
    int count;
    do {
        key_hold = patten01 || patten02;
    } while (!key_hold);

    if (patten01) {
        mode = 0;
    } else {
        mode = 1;
    }
    key_release = 0;
    count = 1000;
    while (!key_release) {
        key_hold = patten01 || patten02;
        if (key_hold) {
            count = 50;
        } else {
            count--;
            if (count == 0) key_release = 1;
        }
    }
    //Stage 2: wait for key released
} //end of function button_detect ()

void mode_change() {
    default_Config ();
    button_detect ();
} //end of function main

```

```

void button_detect () {
    char key_hold;
    int key_release;
    int count;

    do {
        key_hold = pattern;
    } while (!key_hold);

    //Stage 2: wait for key released
    key_release = 0;
    count = 50;
    while (!key_release) {
        key_hold = pattern;
        if (key_hold) {
            count = 50;
        } else {
            count--;
            if (count == 0) key_release = 1;
        }
    }
    //Stage 2: wait for key released
} //end of function button_detect ()

```

We just that timer count to 50, then what for the key_release to be 1. The state will start again until the button push again. At the basic of lab, we have add mode change to separate the button mode. The difference between them is the button number.

Main: (Left is basic, Right is bonus)

```

int main() {
    int status;
    status = 1;
    P2 = 0;
    while(1) {
        mode_change();
        P2 = status;

        if (mode == 0) {
            if (P2 == 128) {
                status = 1;
            } else {
                status = status << 1;
            }
            P2 = status;
        } //end while (patten01)

        if (mode == 1) {
            if (P2 == 1) {
                status = 128;
            } else {
                status = status >> 1;
            }
            P2 = status;
        } //end while (patten01)
    }
}

```

```

int main() {
    default_Config ();
    mode = 0;
    status = 1;
    P2 = 1;
    while(1) {
        button_detect ();
        //P2 = status;
        if (mode == 3) {
            mode = 0;
            status = 1;
        }
        else mode++;
    }
}

```

We use mode record the current state. And we just control the mode. And the difference in basic and bonus is we let the LED state change. One is at main function, another is at Timer_ISR which can that the LED light shift right/left or blink one second.

Timer_ISR:

```
void Timer0_ISR () interrupt 1{
    time_count++;
    if (time_count==4) {
        time_count = 0;
        if (mode == 0) {
            if (P2 == 128){
                status = 1;
            } else {
                status = status << 1;
            }
        } //end while (patten01)
        if (mode == 1) {
            if (P2 == 1){
                status = 128;
            } else {
                status = status >> 1;
            }
        } //end while (patten01)
        if (mode == 2) {
            if (status) status = 0;
            else status = 85;
        }
        if (mode == 3) {
            if (status) status = 0;
            else status = 170;
        }
    }
    P2 = status;
    TH0 = 0;
    TL0 = 0;
} //end of function Timer0_ISR
```

We use time_count to count 4 cycle equal to one second. So when button didn't pressed, the mode won't be change. Also, we have think of some special status happened, like when P2 is at 128 when the LED is shift right.

● Difficulties you've encountered and your solutions

For this lab we encounter two problems. First, when we try to run basic code on C8051, then the light start randomly blink in P2, so we ask to some assistance find that the board maybe have problem, so when we change the new board the code can run correctly. And then we use bonus code and this time light can't blink or move, at last we find the time config setting is at CONFIG PAGE(F), but it is done at LEGACY PAGE(0), so compiler cannot know how our code run, so when we change the page, all of the functions can work.

● Discussions or what you have learned

這次的實驗我學習到如何在 C 的環境中撰寫 Assembler code 成 C code、設定各種腳位以及執行 ISR 指令，和許多 debug 的方法，希望下次的實驗可以不用再換板子了!!! 其實 C8051 板子才是最主要的問題。