微控制器

實驗九

電子計算機(中斷式 Interrupt)

班級：機械 1A

學號：108303013

姓名：黃鉦淳

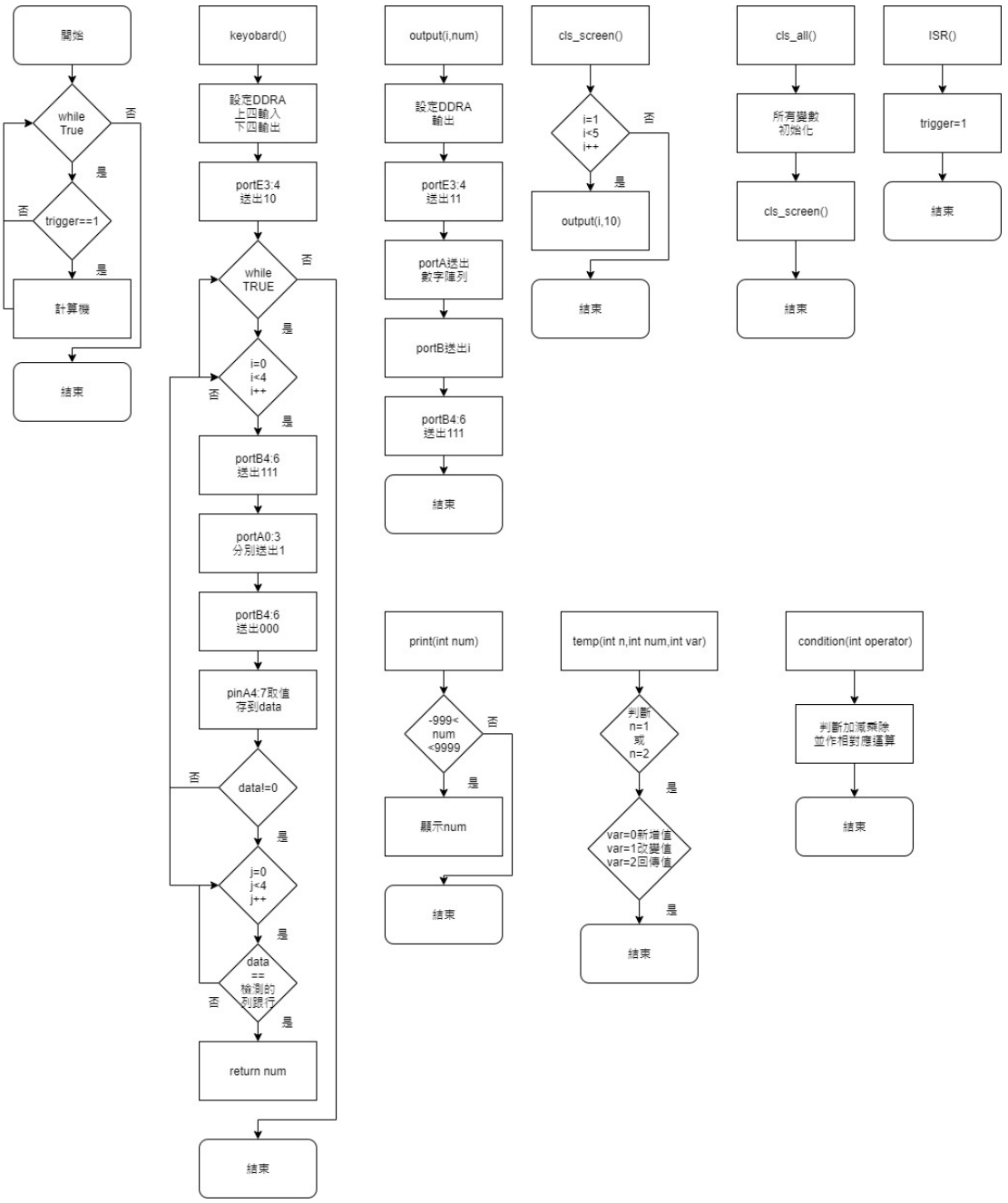
日期：108/12/16

微控制器工作日誌

實驗　　年　月　日

<table>
<tr><td>組別</td><td></td><td>姓名</td><td></td><td>學號</td><td></td></tr>
<tr><td colspan="2">實驗起始時間</td><td colspan="2"></td><td rowspan="2">費時</td><td rowspan="2"></td></tr>
<tr><td colspan="2">實驗結束時間</td><td colspan="2"></td></tr>
<tr><td>所遭遇問題</td><td colspan="5"></td></tr>
<tr><td>解決方法</td><td colspan="5"></td></tr>
<tr><td>完成項目及心得．</td><td colspan="5"></td></tr>
<tr><td>調查</td><td colspan="2">□是否有看課程講解影片
是否實用？有何建議？</td><td colspan="3">□是否有看實驗教學影片
是否實用？有何建議？</td></tr>
</table>

一、流程圖


開始
while True
否
是
trigger==1
否
是
計算機
結束
keyobard()
設定DDRA 上四輸入 下四輸出
portE3:4 送出10
while TRUE
否
是
i=0 i<4 i++
否
是
portB4:6 送出111
portA0:3 分別送出1
portB4:6 送出000
pinA4:7取值 存到data
data!=0
否
是
j=0 j<4 j++
是
data == 檢測的列跟行
否
是
return num
結束
output(i,num)
設定DDRA 輸出
portE3:4 送出11
portA送出 數字陣列
portB送出i
portB4:6 送出111
結束
cls_screen()
i=1 i<5 i++
否
是
output(i,10)
結束
cls_all()
所有變數 初始化
cls_screen()
結束
ISR()
trigger=1
結束
print(int num)
-999< num <9999
否
是
顯示num
結束
temp(int n,int num,int var)
判斷 n=1 或 n=2
是
var=0新增值 var=1改變值 var=2回傳值
是
結束
condition(int operator)
判斷加減乘除 並作相對應運算
結束

二、程式碼

```
#include <stdio.h>
#include "c4mlib.h"
#include <avr/iom128.h>

int operator1 = -1, operator2 = 0, temp_operator = 0, press_operator = 0, count = 0, press_num = 0;
long sum = 0;
int trigger = 0;

int keyboard()
{
    DIO_fpt(200, 0xff, 0, 0x0f); //DDRA setting
    DIO_fpt(4, 0x18, 3, 1);      //portE3:4 sending H,L

    int data = 0, value[4] = {0x08, 0x04, 0x02, 0x01}, num[4][4] = {{0, 1, 4, 7}, {'A', 2, 5, 8}, {'B', 3,
6, 9}, {'F', 'E', 'D', 'C'}};
    int get = -1, up = 0;

    for (int i = 0; i < 4; i++)
    {
        DIO_fpt(1, 0x70, 4, 0x07);         //portB sending 00 為H//Inverter L
        DIO_fpt(0, 0x0f, 0, value[3 - i]); //portA0:3 分別 sending H
        DIO_fpt(1, 0x70, 4, 0);            //CLK改變
        DIO_fgt(100, 0xf0, 4, &data);

        if (data != 0)
        {
            up = 1;
            for (int j = 0; j < 4; j++) //下降緣//怎麼不知道信號 是說有讀到值但是讀到之後就退出了
            {
                if (data == value[j])
                {
                    get = num[i][j];
                }
            }
        }
    }
```

```
    DIO_fpt(1, 0x70, 4, 0x07);
    DIO_fpt(0, 0x0f, 0, 0xff);
    DIO_fpt(1, 0x70, 4, 0);
    DIO_fgt(100, 0xf0, 4, &up);

    if (up != 0 && get > -1)
    {
        return get;
    }

    else if (up == 0)
    {
        return -1;
    }
}

void output(int i, int num) //i為陣列燈1,2,3,4
{
    DIO_fpt(200, 0xff, 0, 0xff);                                                    //DDRA
setting
    DIO_fpt(4, 0x18, 3, 0x03);                                                      //portE3:4
sending H,H
    DIO_fpt(1, 0x70, 4, 0x07);                                                      //portB4:6
sending O1:4為H
    int number[12] = {0x01, 0x4F, 0x12, 0x06, 0x4C, 0x24, 0x20, 0x0F, 0x0, 0x04, 0xff, 0xfe}; //10全暗 11
一槓

    DIO_fpt(0, 0xff, 0, number[num]); //印出數字
    DIO_fpt(1, 0x70, 4, i);           //portB O1:4分別為L
    DIO_fpt(1, 0x70, 4, 0x07);        //portB4:6 sending O2:4為H
}

void cls_screen()
{
    for (int i = 1; i < 5; i++)
    {
        output(i, 10);
```

```
    }
}

void cls_all()
{
    cls_screen();
    temporary(1, 0, 1); //var=0時新增值 1改變值 2就是回傳值
    temporary(2, 0, 1);
    print(0);
    operator1 = -1;
    press_num = sum = operator2 = temp_operator = count = 0;
}

void print(int num) //判斷位數並印出
{
    int bulb[4] = {10, 10, 10, 10}, len = 0, neg = 0, temp;

    if (num != 0)
    {
        temp = num;

        while (temp)
        {
            len++;
            temp /= 10;
        }

        if (num < 0)
        {
            temp = num * (-1);
            neg = 1;
        }
        else
        {
            temp = num;
            neg = 0;
        }
```

```
        cls_screen();

        for (int i = 4; i > 4 - len; i--)
        {
            bulb[i] = temp % 10;
            output(i, bulb[i]);
            temp /= 10;
        }

        if (neg == 1)
        {
            output(4 - len, 11);
        }
    }

    else
    {
        cls_screen();
        output(4, 0);
    }
}

int temporary(int n, int num, int var)
{
    static int a[4] = {0}, b[4] = {0};
    int temp;

    switch (n)
    {
    case 1:
        if (var == 0) //var=0時新增值 1改變值 2就是回傳值
        {
            a[0] = a[1];
            a[1] = a[2];
            a[2] = a[3];
            a[3] = num;
        }
```

```
        else if (var == 1)
        {
            temp = num;

            for (int i = 3; i > -1; i--)
            {
                a[i] = 0;
                a[i] = temp % 10;
                temp /= 10;
            }
        }

        return 1000 * a[0] + 100 * a[1] + 10 * a[2] + a[3];

    case 2:
        if (var == 0)
        {
            b[0] = b[1];
            b[1] = b[2];
            b[2] = b[3];
            b[3] = num;
        }

        else if (var == 1)
        {
            temp = num;

            for (int i = 3; i > -1; i--)
            {
                b[i] = 0;
                b[i] = temp % 10;
                temp /= 10;
            }
        }

        return 1000 * b[0] + 100 * b[1] + 10 * b[2] + b[3];
    }
}
```

```
int condition(int operator)
{
    int DNE = 0;
    long a, b, temp;

    switch (operator)                                   //F  70 +
    {                                                   //E  69 -
    case 43:                                            //D  68 *
        sum = temporary(1, 0, 2) + temporary(2, 0, 2); //C  67 /
        break;

    case 45:
        sum = temporary(1, 0, 2) - temporary(2, 0, 2);
        break;

    case 42:

        a = temporary(1, 0, 2);
        b = temporary(2, 0, 2);
        sum = a * b;
        break;

    case 47:

        if (temporary(2, 0, 2) == 0)
        {
            DNE = 1;
        }
        else
        {
            sum = temporary(1, 0, 2) / temporary(2, 0, 2);
        }
        break;
    }

    printf("sum=%ld\n", sum);
```

```
    if (sum > 9999 || sum < -999 || DNE == 1)
    {

        for (int i = 1; i < 5; i++)
        {
            output(i, 11);
        }
        while (keyboard() != 65 || keyboard() == -1)
        {
            keyboard();
        }

        cls_all();
    }

    return sum;
}

ISR(INT5_vect)
{
    trigger = 1;
}

int main() //不能輸入0
{
    C4M_DEVICE_set();
    DIO_fpt(201, 0xff, 0, 0xff); //DDRB setting
    DIO_fpt(204, 0xff, 0, 0xdf); //DDRE setting PE5為接收

    int get = -1, in = 0;
    long temp_num, temp;

    EICRB = 0x0c; //設定上升緣觸發 依照實驗講義上面寫的 不同的觸發方式不同
    EIMSK = 0x20; //INT5設定
    sei();        //新講義

    print(0);
    printf("-----start!-----\n");
```

```
while (1)
{

    if (keyboard() == -1)
    {
        in = 0;
    }

    if (trigger == 1)
    {
        trigger -= 1;
        in++;

        if (in == 1 && keyboard() != -1)
        {
            get = keyboard();
            printf("Press %d\n", get);

            if (press_operator == 1 && temporary(2, 0, 2) != 0)
            {
                cls_screen();
            }

            if (temporary(2, 0, 2) == 0 && operator1 == 0 && get > -1 && get < 10)
            {
                cls_all();
            }

            if (get > -1 && get < 10 && count < 4)
            {
                press_operator = count = 0;

                if (operator1 == -1)
                {
                    print(temporary(1, get, 0));
                    temp_num = temporary(1, 0, 2);
                }
```

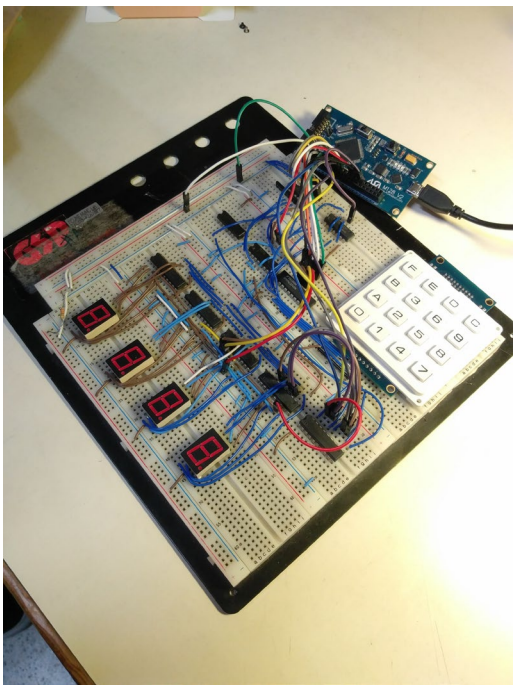
```
    else
    {
        print(temporary(2, get, 0));
        temp_num = temporary(2, 0, 2);
    }
    temp = temp_num;

    while (temp)
    {
        count++;
        temp /= 10;
    }
}

else if (get == 65) //A  65
{
    cls_all();
}

else if (get > 66)
{
    count = 0;
    press_operator = 1;

    switch (get)
    {
    case 70:
        operator1 = 43;
        break;

    case 69:
        operator1 = 45;
        break;

    case 68:
        operator1 = 42;
        break;
```

```
    case 67:
        operator1 = 47;
        break;
    }

    if (temporary(1, 0, 2) != 0 && temporary(2, 0, 2) == 0)
    {
        press_num = 1;
    }
    else
    {
        press_num = 2;
    }

    if (press_num == 1)
    {
        temp_operator = operator1;
    }
    else
    {
        sum = condition(temp_operator);
        print(sum);
        temporary(1, sum, 1);
        temporary(2, 0, 1);
        temp_operator = operator1;
    }
}
else if (get == 66) //按下B 等於
{
    if (operator1 == 0)
    {
        temporary(2, temp_num, 1);

        sum = condition(temp_operator);
        print(sum);
    }
    else if (operator1 > 40)
```

```
                    {
                        cls_screen();

                        sum = condition(operator1);
                        print(sum);

                        operator1 = operator2 = 0;
                        temp_num = temporary(2, 0, 2);
                    }

                    temporary(1, sum, 1);
                    temporary(2, 0, 1);
                }
                printf("1)%4d\n2)%4d\n\n", temporary(1, 0, 2), temporary(2, 0, 2));
                printf("press_num=%d operator1=%d operator2=%d\n", press_num, operator1, operator2);
            }
        }
    }
    return 0;
}
```

三、實驗數據

1.實驗照片



三、實驗問題與討論

1. 在本次實驗中你學到了什麼？

中斷式如何使用

2. 若改變INT5為INT6要有那些變動?

改成ISR(INT6_vect){}還有EIMSK=0x40

3. 為何中斷服務常式一開始就要禁能中斷，若不如此會發生什麼狀況?

ASA板不知道訊號代表中斷

4. 若將中斷改為準位觸發會發生什麼狀況?

達到準位時就會觸發，例如1就會啟動中斷式。邊緣觸發則是分為上升緣或下降緣

5. 請問你，是否可能不將掃描程式寫在中斷中? 如果可以應如何寫，有何優缺點?

是，如上程式碼。若是將掃描程式寫在中斷，缺點是造成中斷式運作時間過長而影響到後面的函式。如上程式碼就可避免中斷式過長，僅改變廣域變數trigger=1