

微控制器

實驗八

電子計算機(輪詢式)

班級：機械 1A

學號：108303013

姓名：黃鉦淳

日期：108/12/1

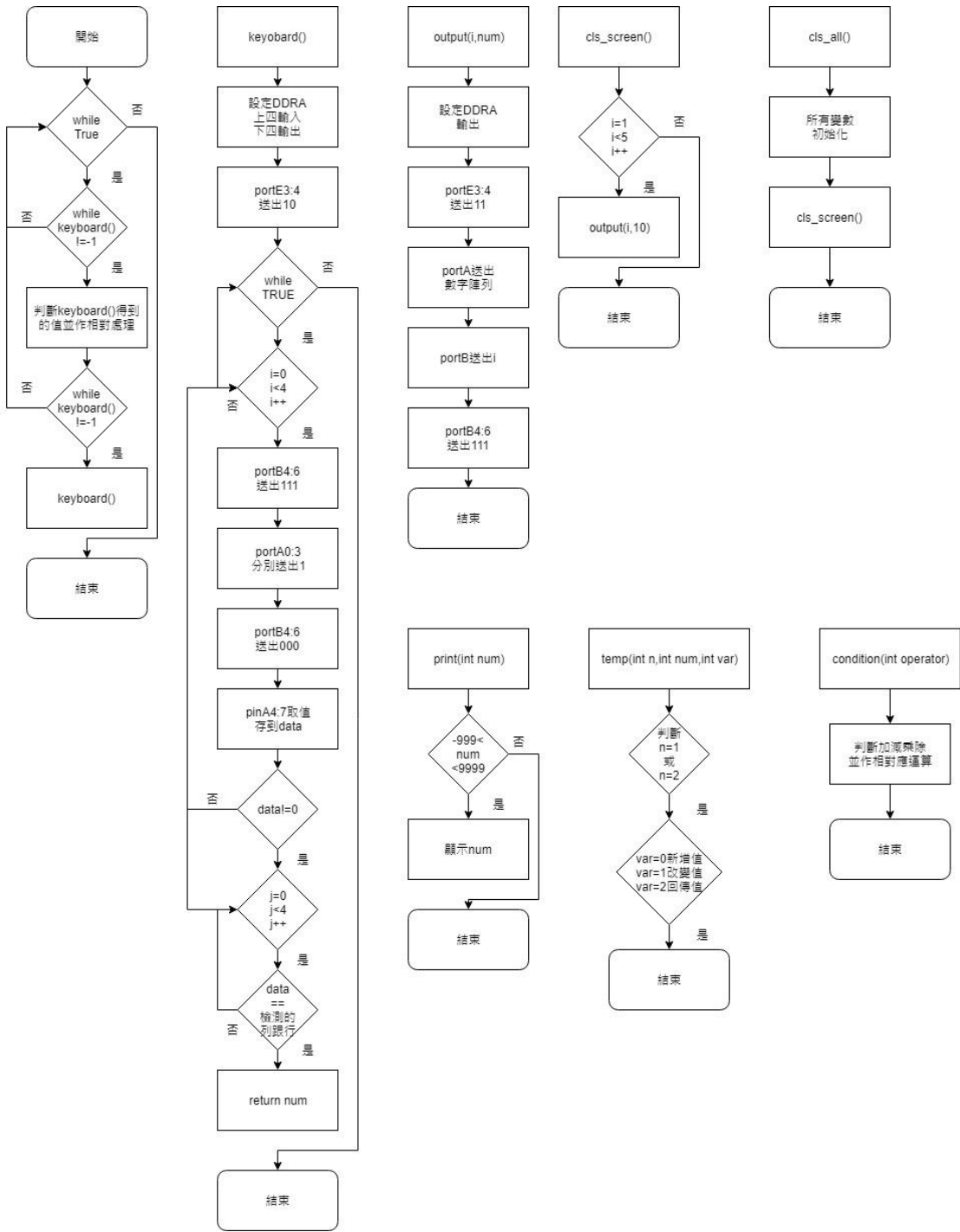
微控制器工作日誌

實驗

年 月 日

組 別		姓 名		學 號	
實驗起始時間				費 時	
實驗結束時間					
所 遭 遇 問 題					
解 決 方 法					
完 及 成 心 項 得 目					
調 查	<input type="checkbox"/> 是否有看課程講解影片 是否實用？有何建議？		<input type="checkbox"/> 是否有看實驗教學影片 是否實用？有何建議？		

一、流程圖



二、程式碼

```
#include <stdio.h>

#include "c4mlib.h"

int operator1 = -1, operator2 = 0, temp_operator1 = 0, count = 0, press_operator = 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'}};

    while (1)
    {
        for (int i = 0; i < 4; i++)
        {
            DIO_fpt(1, 0x70, 4, 0x07);          //portB sending O0 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)
            {
                for (int j = 0; j < 4; j++)
                {
                    if (data == value[j])
                    {
                        return num[i][j];
                    }
                }
            }
        }
        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()
{
    temp(1, 0, 1); //var=0時新增值 1改變值 2就是回傳值
    temp(2, 0, 1);

    cls_screen();
    print(0);

    operator1 = -1;
    operator2 = temp_operator1 = count = 0;
}

void print(int num) //判斷位數並印出
{

```

```
int bulb[4] = { 10, 10, 10, 10 }, len = 0, temp_num, neg = 0;
```

```
if (num > 9999 || num < -999)
```

```
{
```

```
    for (int i = 1; i < 5; i++)
```

```
    {
```

```
        output(i, 11);
```

```
    }
```

```
    while (keyboard() != 65 || keyboard() == -1)
```

```
    {
```

```
        keyboard();
```

```
    }
```

```
    cls_all();
```

```
}
```

```
else if (num != 0)
```

```
{
```

```
    temp_num = num;
```

```
    while (temp_num)
```

```
    {
```

```
        len++;
```

```
        temp_num /= 10;
```

```
    }
```

```
    if (num < 0)
```

```
    {
```

```
        temp_num = num * (-1);
```

```
        neg = 1;
```

```
    }
```

```
    else
```

```
    {
```

```
        temp_num = num;
```

```
        neg = 0;
```

```
    }
```

```

        cls_screen();

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

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

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

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

    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 = num;

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

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

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

```



```
}
```

```
int condition(int operator)
```

```
{
```

```
    int sum;
```

```
    switch (operator)                                //F 70 +
```

```
    {                                                //E 69 -
```

```
    case 43:                                          //D 68 *
```

```
        sum = temp(1, 0, 2) + temp(2, 0, 2); //C 67 /
```

```
        break;
```

```
    case 45:
```

```
        sum = temp(1, 0, 2) - temp(2, 0, 2);
```

```
        break;
```

```
    case 42:
```

```
        sum = temp(1, 0, 2) * temp(2, 0, 2);
```

```
        break;
```

```
    case 47:
```

```
        if (temp(2, 0, 2) == 0)
```

```
        {
```

```
            cls_all();
```

```
            print(10000);
```

```
            sum = 0;
```

```
        }
```

```
        else
```

```
        {
```

```
            sum = temp(1, 0, 2) / temp(2, 0, 2);
```

```
        }
```

```
        break;
```

```
    }
```

```
    return sum;
```

```
}
```

```
int main()
```

```
{
```

```

C4M_DEVICE_set();

DIO_fpt(201, 0xff, 0, 0xff); //DDRB setting
DIO_fpt(204, 0xff, 0, 0xff); //DDRE setting


int get, sum = 0, temp_num;


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


while (1)
{
    while (keyboard() != -1)
    {
        if (press_operator == 1 && temp(2, 0, 2) != 0)
        {
            cls_screen();
        }

        if (temp(2, 0, 2) == 0 && operator1 == 0 && keyboard() < 10)
        {
            cls_all();
        }

        get = keyboard();
        printf("Enter=%d\n", get);

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

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

            else
            {

```

```

        print(temp(2, get, 0));
        temp_num = temp(2, 0, 2);
    }

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

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

```

```

    }

    temp_operator1 = operator1;

    if (temp(2, 0, 2) != 0)
    {
        sum = condition(operator1);
        print(sum);
        if (sum > 9999 || sum < -999)
        {
            sum = 0;
            operator1 = -1;
        }
        temp(1, sum, 1);
        temp(2, 0, 1);
    }
}

else if (get == 66) //按下B 等於
{
    operator2 = 61;
}

if (operator2 == 61)
{
    if (operator1 == 0)
    {
        temp(2, temp_num, 1);

        sum = condition(temp_operator1);
        print(sum);
        if (sum > 9999 || sum < -999)
        {
            sum = 0;
            operator1 = -1;
        }

        operator2 = 0;
    }
}

```

```

    }

    else if (operator1 > 40 && operator1 < 50)
    {
        cls_screen();

        sum = condition(operator1);
        print(sum);
        if (sum > 9999 || sum < -999)
        {
            sum = 0;
            operator1 = -1;
        }
        operator1 = operator2 = 0;

        temp_num = temp(2, 0, 2);
    }
    temp(1, sum, 1);
    temp(2, 0, 1);
}

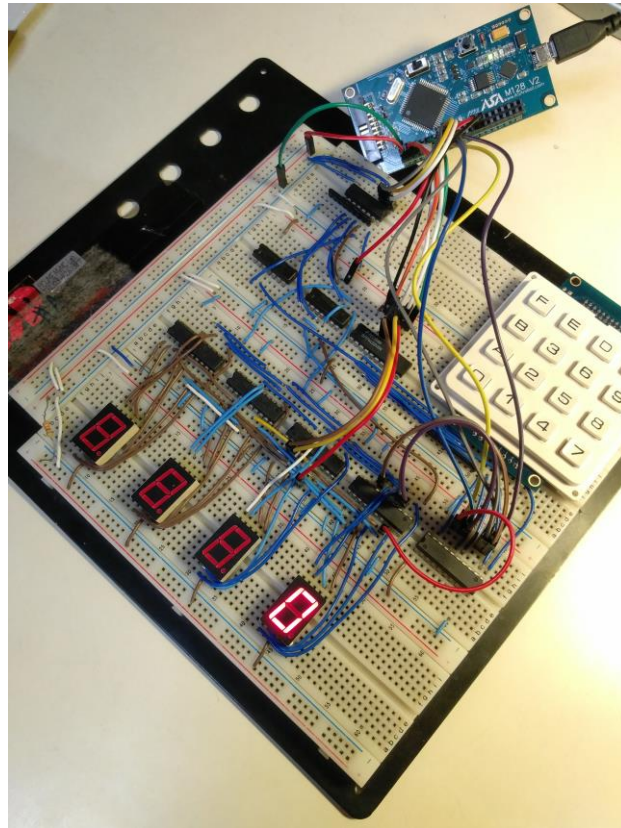
while (keyboard() != -1)
{
    keyboard();
}

printf("First var=%4d,Second var=%4d\n\n", temp(1, 0, 2), temp(2, 0, 2));
}
}
return 0;
}

```

三、實驗數據

1. 實驗照片



三、實驗問題

1、在本次實驗中你學到了什麼？

如何使用printf()除錯

四、實驗討論

請討論各個函式之間，互傳變數的永久性，以及區域/廣域性？

區域性即為在該函式內有效，一旦離開函式該變數即無法使用，並釋放該變數的記憶體

廣域性即為全域變數，在整個檔案內皆有效，從頭到尾都可改變，並不會釋放該變數的記憶體

Static變數離開函式之後不會釋放出記憶體，保留離開時的資料