**電工實驗（四）**

**微處理器實驗2**

**中斷處理程式與時鐘**

**第十四組**

**許銘森B063011058**

程式碼

1. 主程式

#include "myM451Header.h"//加入自訂M451的標頭檔

#include "SevenSegmentAgent.h"//加入自訂七段顯示器 的標頭檔

#include "M451Series.h"//加入M451的標頭檔

void TMR0\_Initial(void);//宣告 TMR0初始化函式

uint32\_t timecount;

void \_7Seg\_Task(void);//宣告七段顯示器運作函式

void Clock\_Initial(void);//宣告Clock初始化函式

void Clock\_TimeTick(void);//宣告Clock運作函式

void Clock\_TimeWrite(void);//宣告Clock寫入函式

uint8\_t second;

uint8\_t minute;

uint8\_t hour;

uint8\_t display;//顯示

uint8\_t mode;//顯示時間單位模式

int \_\_main(){

TMR0\_Initial();//呼叫 TMR0初始化函式

\_7Seg\_Initial();//呼叫七段顯示器初始化函式

Clock\_Initial();//呼叫Clock初始化函式

Second=0；

minute=0;

hour=0;

PE\_MODE &= ~(0x3 << 4);//PE2為輸入，mode2因爲是兩位元0x3要《《4

PE\_MODE |= (0x0 << 4);//PE\_MODE暫存器的BIT5和BIT4必須設為0和0

PA\_MODE &= ~(0x3 << 16);

PA\_MODE |= (0x0 << 16);

PB\_MODE &= ~(0x3 << 12);

PB\_MODE |= (0x0 << 12);

mode=0;

while(1){//一直運作

if(mode==0)//顯示時間運作模式為0時display=second

{

display=second;

}else if(mode==1)

{

display=minute;

}else if(mode==2)

{

display=hour;

}

Clock\_TimeTick();

Clock\_TimeWrite(); //寫入output buffer

\_7Seg\_Task();//掃描七段顯示器

if(PE2==0)//輸入PE2==0顯示時間運作模式為0

{

mode=0

}else if(PA8==0)

{

mode=1;

}else if(PB6==0)

{

mode=2;

}

}

}

2. Timer0

void TMR0\_IRQHandler(void)

{

timecount++;//中斷開始時timecount++

myTMR0\_TIF |= myTMR0\_TIF\_MASK; //clear interrupt flag by set to 1.

}

void TMR0\_Initial(void){

myCLK\_APBCLK0 |= myTMR0\_CLKEN\_MASK; //timer0 clock enable

myCLK\_CLKSEL1 &= ~myTMR0\_CLKSEL\_MASK; //select timer0 clock source

myCLK\_CLKSEL1 |= 0x7 << myTMR0\_CLKSEL\_POS; //22.1184MHz

myTMR0\_CTL &= ~myTMR0\_PRESCL\_MASK;

myTMR0\_CTL |= 0xF6 << myTMR0\_PRESCL\_POS; //除頻除數=254

myTMR0\_CTL &= ~myTMR0\_OPMODE\_MASK;

myTMR0\_CTL |= 0x01 << myTMR0\_OPMODE\_POS; //0x01 for periodic mode

myTMR0\_CMP = 9; //除頻除數=9

myTMR0\_CTL |= myTMR0\_INTEN\_MASK; //enable interrupt

myTMR0\_CTL |= myTMR0\_CNTEN\_MASK; //start timer

myNVIC\_ISER2 |= myTMR0\_ISER2\_MASK; //enable NVIC timer0 interrupt

timecount = 0;

}

4. 七段顯示器

void \_7Seg\_Task(void){

static uint32\_t \_7SegOldCount = 0;

if((uint32\_t)(timecount - \_7SegOldCount) > 0){

\_7Seg\_Switch();

\_7Seg\_Output();

\_7SegOldCount = timecount;

}

}

1. 七段顯示器 2-1//定義七段顯示器

#ifndef \_\_SEVENSEGMENTAGENT\_H\_\_

#define \_\_SEVENSEGMENTAGENT\_H\_\_

#include "myM451Header.h"

#define SEG\_A PB11

#define SEG\_B PB12

#define SEG\_C PB13

#define SEG\_D PB14

#define SEG\_E PB15

#define SEG\_F PB5

#define SEG\_G PD11

#define SEG\_H PF2

#define SEG0\_EN PD8

#define SEG1\_EN PC8

extern uint8\_t \_7Seg\_OutputBuf[2];

void \_7Seg\_Initial(void);

void \_7Seg\_Switch(void);

void \_7Seg\_Output(void);

#endif /\* \_\_SEVENSEGMENTAGENT\_H\_\_ \*/

1. 七段顯示器 2-2

#include "SevenSegmentAgent.h"

uint8\_t \_7Seg\_OutputBuf[2];

uint8\_t BufSel;

const uint8\_t \_7SegPatTable[] = { 0xC0, //7-Segment shows 0

0xF9, //7-Segment shows 1

0xA4, //7-Segment shows 2

0xB0, //7-Segment shows 3

0x99, //7-Segment shows 4

0x92, //7-Segment shows 5

0x82, //7-Segment shows 6

0xD8, //7-Segment shows 7

0x80, //7-Segment shows 8

0x98, //7-Segment shows 9

0xFF};

void \_7Seg\_Initial(void){

GPIO\_Mode\_Set(&myPB\_MODE, 11, GPIO\_OUTPUT\_MODE);

GPIO\_Mode\_Set(&myPB\_MODE, 12, GPIO\_OUTPUT\_MODE);

GPIO\_Mode\_Set(&myPB\_MODE, 13, GPIO\_OUTPUT\_MODE);

GPIO\_Mode\_Set(&myPB\_MODE, 14, GPIO\_OUTPUT\_MODE);

GPIO\_Mode\_Set(&myPB\_MODE, 15, GPIO\_OUTPUT\_MODE);

GPIO\_Mode\_Set(&myPB\_MODE, 5, GPIO\_OUTPUT\_MODE);

GPIO\_Mode\_Set(&myPD\_MODE, 11, GPIO\_OUTPUT\_MODE);

GPIO\_Mode\_Set(&myPF\_MODE, 2, GPIO\_OUTPUT\_MODE);

GPIO\_Mode\_Set(&myPD\_MODE, 8, GPIO\_OUTPUT\_MODE);

GPIO\_Mode\_Set(&myPC\_MODE, 8, GPIO\_OUTPUT\_MODE);

BufSel = 0;

\_7Seg\_OutputBuf[0] = 0xFF;

\_7Seg\_OutputBuf[1] = 0xFF;

}

void \_7Seg\_Switch(void){

BufSel ^= 1;

}

void \_7Seg\_Output(void){

uint8\_t OutputNumber;

uint8\_t OutputData;

OutputNumber = BufSel ? \_7Seg\_OutputBuf[1] : \_7Seg\_OutputBuf[0];

if(OutputNumber >= 10)

OutputNumber = 10;

OutputData = \_7SegPatTable[OutputNumber];

SEG0\_EN = 0;

SEG1\_EN = 0;

SEG\_A = (OutputData & (1 << 0)) ? 1 : 0;

SEG\_B = (OutputData & (1 << 1)) ? 1 : 0;

SEG\_C = (OutputData & (1 << 2)) ? 1 : 0;

SEG\_D = (OutputData & (1 << 3)) ? 1 : 0;

SEG\_E = (OutputData & (1 << 4)) ? 1 : 0;

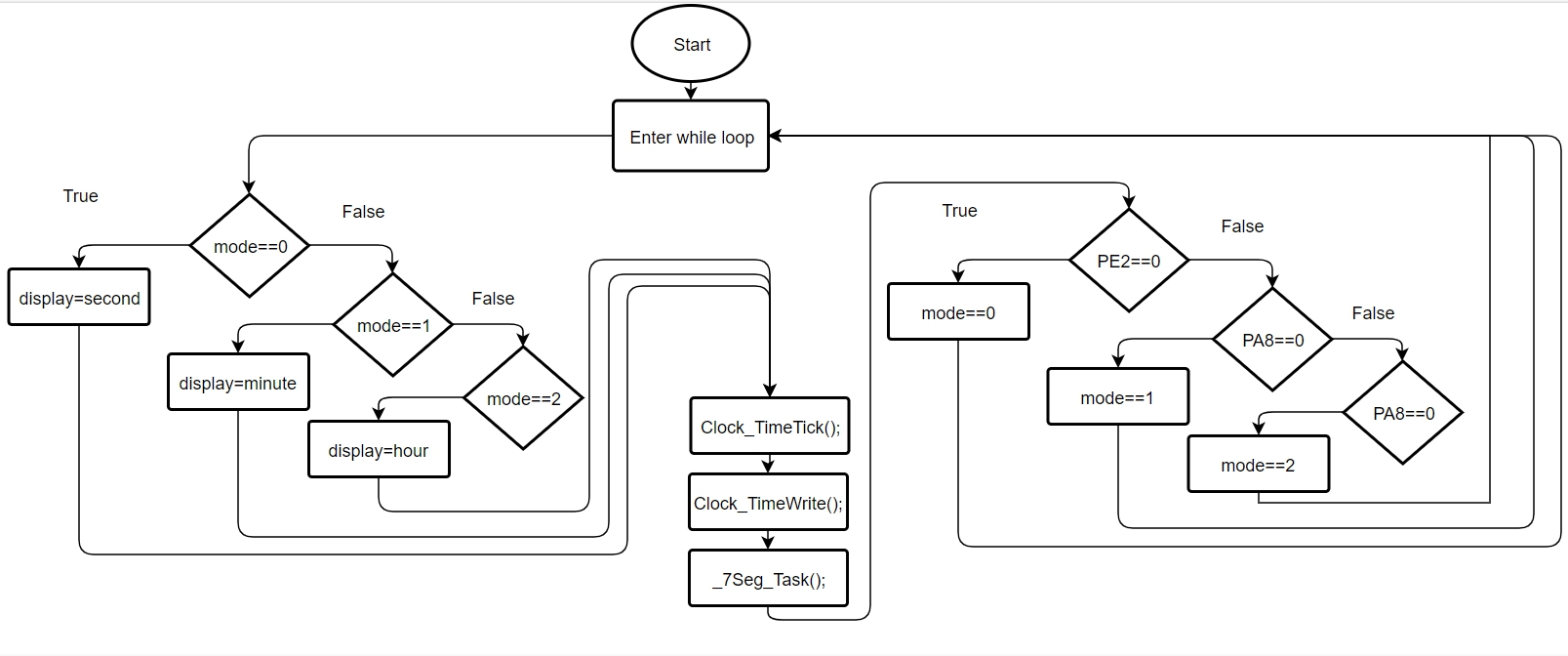
SEG\_F = (OutputData & (1 << 5)) ? 1 : 0;

SEG\_G = (OutputData & (1 << 6)) ? 1 : 0;

SEG\_H = (OutputData & (1 << 7)) ? 1 : 0;

SEG0\_EN = BufSel ? 1 : 0;

SEG1\_EN = BufSel ? 0 : 1;

****程式碼流程

心得

這次實驗是第二次使用微控器，和上次一樣用到c語言去設定M451的腳位輸入輸出，其中要了解mode的設定方法，設定時皆需以兩位移位。這次實驗的重點時要把Clock和interrupt的概念弄清楚。中斷處理程式是時鐘的實行方法，設定中斷頻數便可以用time count數數，Clock就可以實現。這次實驗也不難，主要目的為熟悉軟體操作如何做出跟Clock有關的的IO，然而在做實驗時code寫成按住才顯示時間，后來加了mode和display做顯示時間單位模式和顯示變數，問題就解決了。