**電工實驗（四）**

**微處理器實驗3**

**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 "M451Series.h"//加入自訂M451的標頭檔

#include "TimebaseAgent.h"

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

#include "ButtonAgent.h"//加入自訂按鈕的標頭檔

#include "StepMotorAgent.h"//加入自訂馬達的標頭檔

uint32\_t timecount;

uint8\_t dir;//方向

uint8\_t speed;//速度

uint8\_t oldspeed;//停下來前的速度

int flag;//記錄有沒有停下來過 1：有 0：沒有

int \_\_main(){

Timebase\_Initial();//呼叫 Timebase初始化函式

\_7Seg\_Initial();//呼叫七段顯示器初始化函式

Btn\_Initial();//呼叫按鈕初始化函式

StepMtr\_Initial();//呼叫馬達初始化函式

GPIO\_SetMode(PB,BIT2,GPIO\_MODE\_OUTPUT);//呼叫 LED初始化函式

dir = 1;

speed = 25;

\_7Seg\_WriteBuf(2,5);

flag=0;

while(1){

if(Btn\_IsDown(0xF) == 0x03)//兩個按鈕0x03和0x04同時按，LED亮

PB2 = 0;

else

PB2 = 1;

if(flag==1 && Btn\_IsOneShot(0x04) == 0x04)//馬達停下來再按0x04按鈕，轉回停下來前的速度

{

speed=oldspeed;

flag=0;

\_7Seg\_WriteBuf(1000/speed/10,1000/speed%10);

Btn\_OneShotClear(0x04);

}

/\* if the button pressed \*/

if(Btn\_IsOneShot(0x01) == 0x01){//按0x04按鈕加速

speed++;

//write to 7seg output buffer

\_7Seg\_WriteBuf(1000/speed/10,1000/speed%10);

//clear one-shot flag

Btn\_OneShotClear(0x01);

}

if(Btn\_IsOneShot(0x02) == 0x02){//按0x04按鈕減速

if(speed)

speed--;

//write to 7seg output buffer

\_7Seg\_WriteBuf(1000/speed/10,1000/speed%10);

Btn\_OneShotClear(0x02);

}

if(Btn\_IsOneShot(0x04) == 0x04){//按0x04按鈕停下

oldspeed=speed;//把停下來前的速度存起來

speed=0;

flag=1;

\_7Seg\_WriteBuf(1000/speed/10,1000/speed%10);

Btn\_OneShotClear(0x04);

}

if(Btn\_IsOneShot(0x08) == 0x08){//按0x03按鈕改變轉向

if(dir==0)

{

dir=1;

}

else if(dir==1)

{

dir=0;

}

//write to 7seg output buffer

\_7Seg\_WriteBuf(1000/speed/10,1000/speed%10);

Btn\_OneShotClear(0x08);

}

/\* Step motor output \*/

StepMtr\_Task(dir, speed);//把方向速度給馬達

/\* Scan button \*/

Btn\_Task();

/\* 7 segment output \*/

\_7Seg\_Task();

}

}

1. 步進馬達
2. #ifndef \_\_STEPMOTORAGENT\_H\_\_//定義步進馬達

#define \_\_STEPMOTORAGENT\_H\_\_

#include "M451Series.h"

#define MTR\_A PE11

#define MTR\_B PE10

#define MTR\_A\_BAR PE13

#define MTR\_B\_BAR PE12

extern uint32\_t timecount;

void StepMtr\_Initial(void);

void StepMtr\_Task(uint8\_t dir, uint32\_t d);

#endif /\* \_\_STEPMOTORAGENT\_H\_\_ \*/

void StepMtr\_Task(uint8\_t dir, uint32\_t d){//步進馬達運作函式

static uint32\_t StepMtrOldCount = 0;

if((uint32\_t)(timecount - StepMtrOldCount) < d)

return;

StepMtrOldCount = timecount;

if(d ==0 )

StepMtrState = 4;

switch(StepMtrState){

case 0://state 0輸出1001

MTR\_A = 1;

MTR\_B = 0;

MTR\_A\_BAR = 0;

MTR\_B\_BAR = 1;

StepMtrState=dir?1:3;//dir為1時，順向。0時，逆向。

break;

case 1://state 1輸出1100

MTR\_A = 1;

MTR\_B = 1;

MTR\_A\_BAR = 0;

MTR\_B\_BAR = 0;

StepMtrState=dir?2:0;

break;

case 2://state 2輸出0110

MTR\_A = 0;

MTR\_B = 1;

MTR\_A\_BAR = 1;

MTR\_B\_BAR = 0;

StepMtrState=dir?3:1;

break;

case 3://state 3輸出0011

MTR\_A = 0;

MTR\_B = 0;

MTR\_A\_BAR = 1;

MTR\_B\_BAR = 1;

StepMtrState=dir?0:2;

break;

case 4://state 4輸出0000

MTR\_A = 0;

MTR\_B = 0;

MTR\_A\_BAR = 0;

MTR\_B\_BAR = 0;

StepMtrState=dir?0:4;

break;

default://default輸出0000

StepMtrState = 4;

}

}

3. 按鍵

#ifndef \_\_BUTTONAGENT\_H\_\_//定義按鍵

#define \_\_BUTTONAGENT\_H\_\_

#include "M451Series.h"

#define SW1 PE2

#define SW2 PA8

#define SW3 PB6

#define SW4 PB7

#define BTN\_STATE\_UP 0

#define BTN\_STATE\_BOUNCING 1

#define BTN\_STATE\_ONESHOT 2

#define BTN\_STATE\_DOWN 3

#define DEBOUNCEDELAY 100

extern uint32\_t timecount;

void Btn\_Initial(void);

void Btn\_Task(void);

uint8\_t Btn\_IsDown(uint8\_t btnSelMask);

uint8\_t Btn\_IsOneShot(uint8\_t btnSelMask);

void Btn\_OneShotClear(uint8\_t btnSelMask);

#endif /\* \_\_BUTTONAGENT\_H\_\_ \*/

#include "ButtonAgent.h"

uint8\_t BtnDownFlag;

uint8\_t BtnOneShot;

uint8\_t BtnState[4];

uint32\_t BtnOldCount[4];

volatile uint32\_t\* pSW[4];

1. void Btn\_Initial(void){//步進馬達運作函式

GPIO\_SetMode(PE, BIT2, GPIO\_MODE\_INPUT);//呼叫按鍵初始化函式

GPIO\_SetMode(PA, BIT8, GPIO\_MODE\_INPUT);

GPIO\_SetMode(PB, BIT6, GPIO\_MODE\_INPUT);

GPIO\_SetMode(PB, BIT7, GPIO\_MODE\_INPUT);

BtnDownFlag = 0;

BtnOneShot = 0;

pSW[0] = &SW1;

pSW[1] = &SW2;

pSW[2] = &SW3;

pSW[3] = &SW4;

for(uint8\_t i = 0; i < 4 ; i++){

BtnState[i] = BTN\_STATE\_UP;

BtnOldCount[i] = 0;

}

}

void Btn\_Task(void){

for(uint8\_t i = 0 ; i < 4 ; i++){

switch(BtnState[i]){

case BTN\_STATE\_UP://按鍵為沒按的狀態

if(\*pSW[i] == 0){

BtnState[i] = BTN\_STATE\_BOUNCING;

}

BtnOldCount[i] = timecount;

BtnDownFlag &= ~(0x01 << i);

BtnOneShot &= ~(0x01 << i);

break;

case BTN\_STATE\_BOUNCING://按鍵彈跳中

if(\*pSW[i] == 1){

BtnState[i] = BTN\_STATE\_UP;

}else if((uint32\_t)(timecount - BtnOldCount[i]) > DEBOUNCEDELAY){

BtnState[i] = BTN\_STATE\_ONESHOT;

}

break;

case BTN\_STATE\_ONESHOT://按鍵的oneshot訊號發出

BtnState[i] = BTN\_STATE\_DOWN;

BtnOneShot |= 0x01 << i;

break;

case BTN\_STATE\_DOWN://按鍵為按的狀態

if(\*pSW[i] == 1){

BtnState[i] = BTN\_STATE\_UP;

}else {

BtnDownFlag |= 0x01 << i;

}

break;

default:

BtnState[i] = BTN\_STATE\_UP;

}//switch(BtnState[i])

}//for(0:3)

}

uint8\_t Btn\_IsDown(uint8\_t btnSelMask){

return BtnDownFlag & btnSelMask;

}

uint8\_t Btn\_IsOneShot(uint8\_t btnSelMask){

return BtnOneShot & btnSelMask;

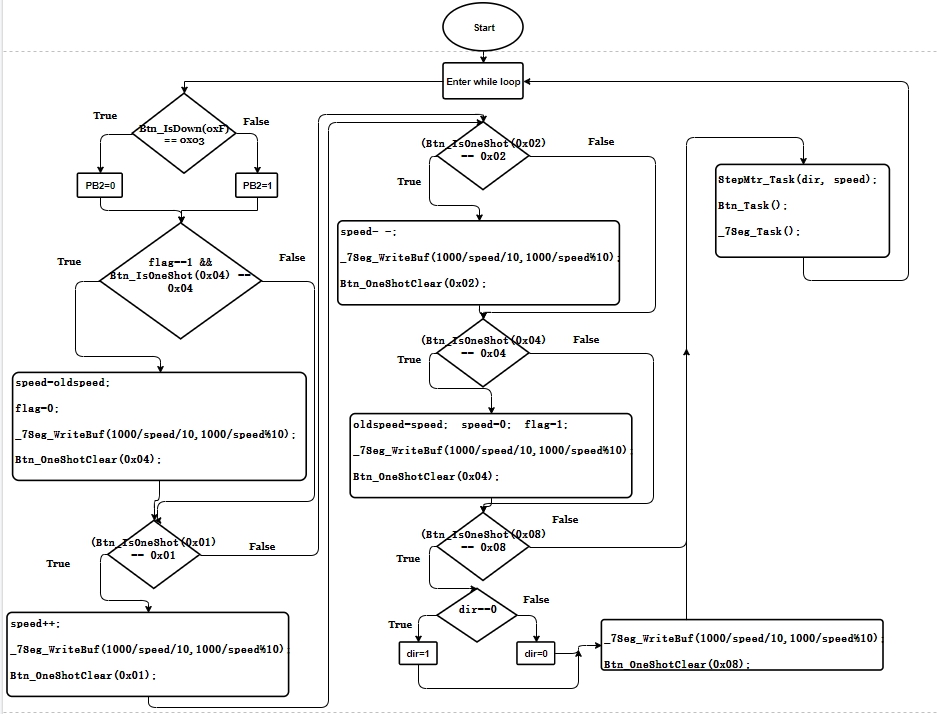
}

void Btn\_OneShotClear(uint8\_t btnSelMask){

BtnOneShot &= ~btnSelMask;

}

程式碼流程



心得

這次實驗是第三次使用微控器，和上次一樣用到c語言去設定M451的腳位輸入輸出，但是這次有寫好的標頭檔，不用另外再設定。這次實驗的重點時要把Clock和interrupt的概念弄清楚，再結合state machine 的概念去控制馬達。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。

state是控制馬達的轉動波形，有五個state。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及運用，實驗過程大致順利。