**電工實驗（四）**

**微處理器實驗五**

**系統整合實驗**

**第十四組**

**許銘森B063011058**

程式碼

**1. Include**

#include <stdio.h>

#include "M451Series.h"//加入自訂M451的標頭檔

#include "TimebaseAgent.h"//加入自訂時間的標頭檔

#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

#include "ADCAgent.h"//加入自訂ADC的標頭檔

**2. Define**

#define TASK\_CLKSHOW 0

#define TASK\_CLKSET 1

#define TASK\_MTRSHOW 2

#define TASK\_MTRSET 3

#define TASK\_MTRADC 4

#define STATE\_CLKSHOW\_SEC 0

#define STATE\_CLKSHOW\_MIN 1

#define STATE\_CLKSHOW\_HRS 2

#define STATE\_CLKSET\_ON 0

#define STATE\_CLKSET\_OFF 1

#define STATE\_MTRSHOW\_SPEED 0

#define STATE\_MTRSHOW\_DIR 1

#define STATE\_MTRSHOW\_TEMP 2

#define STATE\_MTRSET\_ON 0

#define STATE\_MTRSET\_OFF 1

#define STATE\_MTRADC\_SPEED 0

#define STATE\_MTRADC\_DIR 1

#define STATE\_MTRADC\_TEMP 2

uint32\_t timecount;

uint8\_t Task\_ID;

uint8\_t mtr\_set;

uint8\_t Task\_ClkShow\_State;//CLK顯示state

uint8\_t Task\_ClkSet\_State;//CLK調整state

uint8\_t Task\_MtrShow\_State;//馬達顯示state

uint8\_t Task\_MtrSet\_State;//馬達調整state

uint8\_t Task\_MtrADC\_State;//馬達analog —》Digital state

uint8\_t \_7seg\_lo;

uint8\_t \_7seg\_hi;

uint8\_t mtr\_speed; //rpm

uint8\_t mtr\_dir;

uint32\_t mtr\_d;

uint8\_t stop;

int8\_t clk\_sec;

int8\_t clk\_min;

int8\_t clk\_hr;

void DisplayTask(uint8\_t hi, uint8\_t lo);

void ClkShow(void);

void ClkSet(void);

void MtrShow(void);

void MtrSet(void);

void MtrADC(void);

void CLOCK(void);

uint32\_t StepMtr\_RPMtoD(uint8\_t rmp);

**3.main**

int \_\_main(){

Timebase\_Initial();//呼叫各元件初始化函式

\_7Seg\_Initial();

Btn\_Initial();

StepMtr\_Initial();

ADC\_Initial();

clk\_sec = 0;

clk\_min = 0;

clk\_hr = 0;

\_7seg\_hi = 0;

\_7seg\_lo = 0;

mtr\_dir = 1;

mtr\_speed = 25;

stop = 0;

mtr\_d = StepMtr\_RPMtoD(mtr\_speed);

Task\_ID = TASK\_CLKSHOW;

Task\_ClkShow\_State = STATE\_CLKSHOW\_SEC;

Task\_ClkSet\_State = STATE\_CLKSET\_ON;

Task\_MtrShow\_State = STATE\_MTRSHOW\_SPEED;

Task\_MtrSet\_State = STATE\_MTRSET\_ON;

Task\_MtrADC\_State = STATE\_MTRADC\_SPEED;

while(1){

switch(Task\_ID)//五個模式用ID切換

{

case TASK\_CLKSHOW:

ClkShow();

break;

case TASK\_CLKSET:

ClkSet();

break;

case TASK\_MTRSHOW:

MtrShow();

CLOCK();//數時間

break;

case TASK\_MTRSET:

MtrSet();

CLOCK();

break;

case TASK\_MTRADC:

MtrADC();

CLOCK();

break;

default:

Task\_ID = TASK\_CLKSHOW;

}

DisplayTask(\_7seg\_hi, \_7seg\_lo); //10Hz update rate

\_7Seg\_Task(); //1kKz switch rate

ADC\_Task(); //ADC conversion

StepMtr\_Task(mtr\_dir, mtr\_d); //Step motor output

Btn\_Task(); //Button scan

}

}

**4.DisplayTask//**七段顯示器設定

void DisplayTask(uint8\_t hi, uint8\_t lo){

static uint32\_t DpOldCount = 0;

if((uint32\_t)(timecount - DpOldCount) < 1000)//100ms

return;

DpOldCount = timecount;

\_7Seg\_WriteBuf(hi, lo);

}

**5.CLOCK**//計時間

void CLOCK(void)

{

static uint32\_t ClkOldCount = 0;

/\* Clock tick \*/

if((uint32\_t)(timecount - ClkOldCount) >= 10000)

{

if(++clk\_sec >= 60){

clk\_sec = 0;

if(++clk\_min >= 60){

clk\_min = 0;

if(++clk\_hr >= 24)

clk\_hr = 0;

}

}

ClkOldCount = timecount;

}

}

**6.Clkshow**

void ClkShow(void){

CLOCK();

/\* Button control \*/

if(Btn\_IsDown(0x01) == 0x01)//同時按兩個按鈕

{

if(Btn\_IsOneShot(0x02) == 0x02)//按1 2號按鈕

{

Task\_ID = TASK\_MTRSHOW; //TASK\_MTRSHOW

Btn\_OneShotClear(0x02);

}

if(Btn\_IsOneShot(0x04) == 0x04){//按1 4號按鈕

Task\_ID = TASK\_CLKSET; //TASK\_CLKSET

Btn\_OneShotClear(0x04);

}

if(Btn\_IsOneShot(0x08) == 0x08){//按1 8號按鈕

Btn\_OneShotClear(0x08);

}

}else{

if(Btn\_IsOneShot(0x02) == 0x02){//按2號按鈕

Task\_ClkShow\_State = STATE\_CLKSHOW\_HRS;//state為顯示小時

Btn\_OneShotClear(0x02);

}

if(Btn\_IsOneShot(0x04) == 0x04){//按4號按鈕

Task\_ClkShow\_State = STATE\_CLKSHOW\_MIN;//state為顯示分

Btn\_OneShotClear(0x04);

}

if(Btn\_IsOneShot(0x08) == 0x08){//按8號按鈕

Task\_ClkShow\_State = STATE\_CLKSHOW\_SEC;//state為顯示秒

Btn\_OneShotClear(0x08);

}

}

/\* 7segment control \*/

switch(Task\_ClkShow\_State){//根據ClkShow\_State顯示時間單位

case STATE\_CLKSHOW\_SEC:

\_7seg\_hi = (clk\_sec / 10) % 10;

\_7seg\_lo = clk\_sec % 10;

break;

case STATE\_CLKSHOW\_MIN:

\_7seg\_hi = (clk\_min / 10) % 10;

\_7seg\_lo = clk\_min % 10;

break;

case STATE\_CLKSHOW\_HRS:

\_7seg\_hi = (clk\_hr / 10) % 10;

\_7seg\_lo = clk\_hr % 10;

break;

default:

Task\_ClkShow\_State = STATE\_CLKSHOW\_SEC;

}

}

**7.ClkSet**

void ClkSet(void){

static uint32\_t ClkOldCount = 0;

if((uint32\_t)(timecount - ClkOldCount) >= 3000){

Task\_ClkSet\_State = !Task\_ClkSet\_State;

ClkOldCount = timecount;

}//調整閃爍快慢

if(Btn\_IsDown(0x01) == 0x01)//同時按兩個按鈕

{

if(Btn\_IsOneShot(0x02) == 0x02){

Btn\_OneShotClear(0x02);

}

if(Btn\_IsOneShot(0x04) == 0x04){

Btn\_OneShotClear(0x04);

}

if(Btn\_IsOneShot(0x08) == 0x08){

Btn\_OneShotClear(0x08);

}

}

else

{

if(Btn\_IsOneShot(0x02) == 0x02)//按2號按鈕

{

switch(Task\_ClkShow\_State)//根據ClkShow\_State加時間

{

case STATE\_CLKSHOW\_SEC:

if(++clk\_sec >= 60)

clk\_sec = 0;

break;

case STATE\_CLKSHOW\_MIN:

if(++clk\_min >= 60)

clk\_min = 0;

break;

case STATE\_CLKSHOW\_HRS:

if(++clk\_hr >= 24)

clk\_hr = 0;

break;

}

Btn\_OneShotClear(0x02);

}

if(Btn\_IsOneShot(0x04) == 0x04)//按4號按鈕

{

switch(Task\_ClkShow\_State)//根據ClkShow\_State減時間

{

case STATE\_CLKSHOW\_SEC:

if(--clk\_sec < 0)

clk\_sec = 59;

break;

case STATE\_CLKSHOW\_MIN:

if(--clk\_min < 0)

clk\_min = 59;

break;

case STATE\_CLKSHOW\_HRS:

if(--clk\_hr < 0)

clk\_hr = 23;

break;

}

Btn\_OneShotClear(0x04);

}

if(Btn\_IsOneShot(0x08) == 0x08)//按8號按鈕

{

Task\_ID = TASK\_CLKSHOW;//跳回顯示時間模式

Btn\_OneShotClear(0x08);

}

}

/\* 7segment control \*/

switch(Task\_ClkSet\_State){//根據ClkSet\_State顯示時間單位

case STATE\_CLKSET\_ON:

switch(Task\_ClkShow\_State){

case STATE\_CLKSHOW\_SEC:

\_7seg\_hi = (clk\_sec / 10) % 10;

\_7seg\_lo = clk\_sec % 10;

break;

case STATE\_CLKSHOW\_MIN:

\_7seg\_hi = (clk\_min / 10) % 10;

\_7seg\_lo = clk\_min % 10;

break;

case STATE\_CLKSHOW\_HRS:

\_7seg\_hi = (clk\_hr / 10) % 10;

\_7seg\_lo = clk\_hr % 10;

break;

default:

Task\_ClkShow\_State = STATE\_CLKSHOW\_SEC;

}

break;

case STATE\_CLKSET\_OFF:

\_7seg\_hi = 10;

\_7seg\_lo = 10;

break;

default:

Task\_ClkSet\_State = STATE\_CLKSET\_ON;

}

}

**8.馬達顯示**//原理如Clk\_show

void MtrShow(void){

static uint32\_t MtrShowOldCount = 0;

if((uint32\_t)(timecount - MtrShowOldCount) < 1000)//100 ms

return;

MtrShowOldCount = timecount;

if(Btn\_IsDown(0x01) == 0x01)

{

if(Btn\_IsOneShot(0x02) == 0x02){

Task\_ID = TASK\_MTRADC;//切換模式TASK\_MTRADC

Btn\_OneShotClear(0x02);

}

if(Btn\_IsOneShot(0x04) == 0x04){

Task\_ID = TASK\_MTRSET;//切換模式TASK\_MTRSET

Btn\_OneShotClear(0x04);

}

if(Btn\_IsOneShot(0x08) == 0x08){

mtr\_d= 0;//馬達速度爲零

Btn\_OneShotClear(0x08);

}

}

else

{

if(Btn\_IsOneShot(0x02) == 0x02)

{

Task\_MtrShow\_State=STATE\_MTRSHOW\_SPEED;//state為顯示速

Btn\_OneShotClear(0x02);

}

if(Btn\_IsOneShot(0x04) == 0x04)

{

Task\_MtrShow\_State=STATE\_MTRSHOW\_DIR;//state為顯示方向

Btn\_OneShotClear(0x04);

}

if(Btn\_IsOneShot(0x08) == 0x08)

{

Task\_MtrShow\_State=STATE\_MTRSHOW\_TEMP;//state為顯示溫度

Btn\_OneShotClear(0x08);

}

}

/\* 7segment control \*/

switch(Task\_MtrShow\_State){//根據MtrShow\_State顯示馬達參數

case STATE\_MTRSHOW\_SPEED:

\_7seg\_hi = (mtr\_speed / 10) % 10;

\_7seg\_lo = mtr\_speed % 10;

break;

case STATE\_MTRSHOW\_DIR:

\_7seg\_hi = 0;

\_7seg\_lo = mtr\_dir % 10;

break;

case STATE\_MTRSHOW\_TEMP:

\_7seg\_hi =(ADC\_GetTemperature()/10)%10;

\_7seg\_lo = ADC\_GetTemperature()%10;

break;

default:

Task\_MtrShow\_State = STATE\_MTRSHOW\_SPEED;

}

}

**9.馬達設定**//原理如Clkset

void MtrSet(void){

static uint32\_t MtrSetOldCount = 0;

if((uint32\_t)(timecount - MtrSetOldCount) >= 3000)//100 ms

{

Task\_MtrSet\_State =!Task\_MtrSet\_State;

MtrSetOldCount = timecount;

}//調整閃爍快慢

if(Btn\_IsDown(0x01) == 0x01)

{

if(Btn\_IsOneShot(0x02) == 0x02){

Btn\_OneShotClear(0x02);

}

if(Btn\_IsOneShot(0x04) == 0x04){

Btn\_OneShotClear(0x04);

}

if(Btn\_IsOneShot(0x08) == 0x08){

Btn\_OneShotClear(0x08);

}

}

else

{

if(Btn\_IsOneShot(0x02) == 0x02)

{

if(Task\_MtrShow\_State==STATE\_MTRSHOW\_SPEED)//如果顯示速度

{

mtr\_speed++;//加速

\_7seg\_hi = (mtr\_speed / 10) % 10;

\_7seg\_lo = mtr\_speed % 10;

Btn\_OneShotClear(0x02);

}

if(Task\_MtrShow\_State==STATE\_MTRSHOW\_DIR)//如果顯示方向

{

mtr\_dir=1;//換向

Btn\_OneShotClear(0x02);

}

}

if(Btn\_IsOneShot(0x04) == 0x04)

{

if(Task\_MtrShow\_State==STATE\_MTRSHOW\_SPEED)//如果顯示速度

{

mtr\_speed--;//減速

\_7seg\_hi = (mtr\_speed / 10) % 10;

\_7seg\_lo = mtr\_speed % 10;

Btn\_OneShotClear(0x04);

}

if(Task\_MtrShow\_State==STATE\_MTRSHOW\_DIR)//如果顯示方向

{

mtr\_dir=0;

Btn\_OneShotClear(0x04);

}

}

if(Btn\_IsOneShot(0x08) == 0x08)

{

Task\_ID = TASK\_MTRSHOW;//切換模式TASK\_MTRSHOW

Btn\_OneShotClear(0x08);

}

}

mtr\_d = StepMtr\_RPMtoD(mtr\_speed);

/\* 7segment control \*/

switch(Task\_MtrSet\_State){//根據MtrSet\_State顯示馬達參數

case STATE\_MTRSET\_ON:

switch(Task\_MtrShow\_State){

case STATE\_MTRSHOW\_SPEED:

\_7seg\_hi = (mtr\_speed / 10) % 10;

\_7seg\_lo = mtr\_speed % 10;

break;

case STATE\_MTRSHOW\_DIR:

\_7seg\_hi = 0;

\_7seg\_lo = mtr\_dir % 10;

break;

case STATE\_MTRSHOW\_TEMP:

\_7seg\_hi =(ADC\_GetTemperature()/10)%10;

\_7seg\_lo = ADC\_GetTemperature()%10;

break;

default:

Task\_ClkShow\_State = STATE\_MTRSHOW\_SPEED;

}

break;

case STATE\_MTRSET\_OFF:

\_7seg\_hi = 10;

\_7seg\_lo = 10;

break;

default:

Task\_MtrSet\_State = STATE\_MTRSET\_ON;

}

}

**9.馬達ADC模式**

void MtrADC(void){

static uint32\_t SpeedUpdateOldCount = 0;

static uint8\_t select\_speed ;

select\_speed = mtr\_speed;

if((uint32\_t)(timecount - SpeedUpdateOldCount) >= 1000)//100 ms

{

if(Btn\_IsDown(0x01) == 0x01)

{

if(Btn\_IsOneShot(0x02) == 0x02){

Task\_ID = TASK\_CLKSHOW;//切換模式TASK\_MTRSHOW

Btn\_OneShotClear(0x02);

}

if(Btn\_IsOneShot(0x04) == 0x04){

mtr\_dir = (mtr\_dir)? 0 : 1;//切換馬達方向

Btn\_OneShotClear(0x04);

}

if(Btn\_IsOneShot(0x08) == 0x08){

stop = stop? 0 : 1;

select\_speed = (stop)? 0 : mtr\_speed;//按著馬達停

Btn\_OneShotClear(0x08);

}

}

else

{

if(Btn\_IsOneShot(0x02) == 0x02)

{

Task\_MtrADC\_State=STATE\_MTRADC\_SPEED;

Btn\_OneShotClear(0x02);

}

if(Btn\_IsOneShot(0x04) == 0x04)

{

Task\_MtrADC\_State=STATE\_MTRADC\_DIR;

Btn\_OneShotClear(0x04);

}

if(Btn\_IsOneShot(0x08) == 0x08)

{

Task\_MtrADC\_State=STATE\_MTRADC\_TEMP;

Btn\_OneShotClear(0x08);

}

}

mtr\_speed = ADC\_GetVR();//用可變電阻控制馬達速度

mtr\_d = StepMtr\_RPMtoD(select\_speed);

SpeedUpdateOldCount = timecount;

}

/\* 7segment control \*/

switch(Task\_MtrADC\_State){//根據MtrADC\_State顯示馬達參數

case STATE\_MTRADC\_SPEED:

\_7seg\_hi = (mtr\_speed / 10) % 10;

\_7seg\_lo = mtr\_speed % 10;

break;

case STATE\_MTRADC\_DIR:

\_7seg\_hi = 0;

\_7seg\_lo = mtr\_dir % 10;

break;

case STATE\_MTRADC\_TEMP:

\_7seg\_hi =(ADC\_GetTemperature()/10)%10;

\_7seg\_lo = ADC\_GetTemperature()%10;

break;

default:

Task\_MtrADC\_State = STATE\_MTRADC\_SPEED;

}

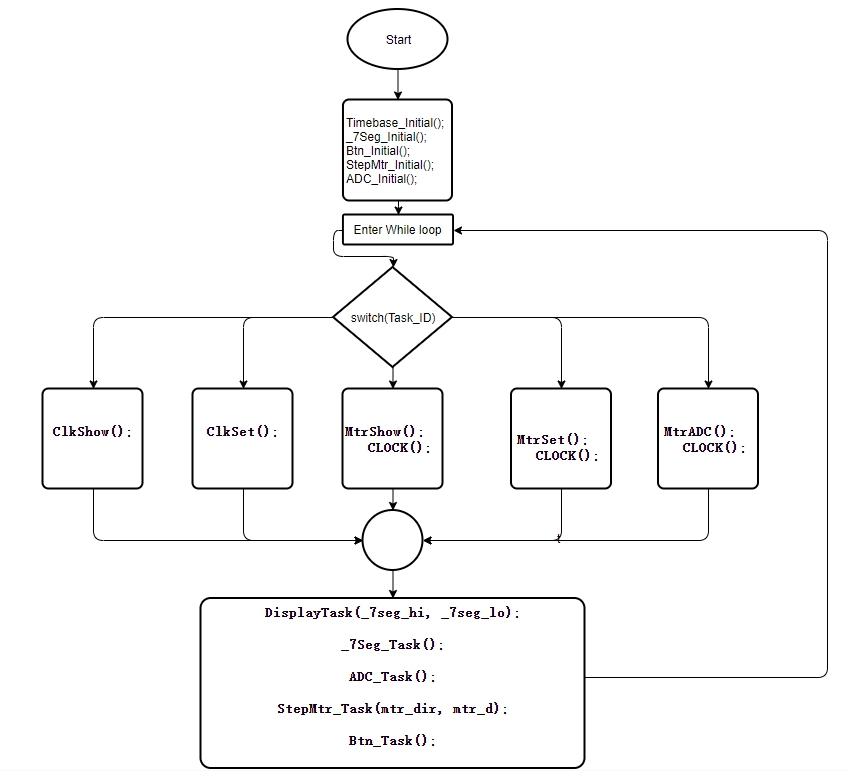
}

uint32\_t StepMtr\_RPMtoD(uint8\_t rmp)

{

return rmp ? 6000/rmp : 0;

}

程式碼流程

心得

這次實驗是第五次使用微控器，和上次一樣用到中斷處理程式與時鐘

，再結合state machine 的概念去控制馬達。這次主要是把上幾次實驗的功能一次做出，code很長需要仔細思考，把注解寫出來會比較清楚。