**電工實驗（四）**

**微處理器實驗4**

**ADC**

**第十四組**

**許銘森B063011058**

程式碼

1. ADC

//ADC標頭檔

#ifndef \_\_ADCAGENT\_H\_\_

#define \_\_ADCAGENT\_H\_\_

#include "M451Series.h"

extern uint32\_t timecount;

void ADC\_Initial(void);

void ADC\_Task(void);

uint8\_t ADC\_GetTemperature(void);

uint8\_t ADC\_GetVR(void);

#endif /\* \_\_ADCAGENT\_H\_\_ \*/

//ADC標頭檔C檔

#include "ADCAgent.h"

uint8\_t ADCState;

uint16\_t TempBuf;

uint16\_t VRBuf;

void ADC\_Initial(void){ //ADC初始化函式

CLK->APBCLK0 |= CLK\_APBCLK0\_EADCCKEN\_Msk; //ADC 時脈 啓用

CLK->CLKDIV0 &= ~CLK\_CLKDIV0\_EADCDIV\_Msk; //ADC 時脈 除頻

CLK->CLKDIV0 |= 0 << CLK\_CLKDIV0\_EADCDIV\_Pos; //(div + 1)

EADC->CTL |= EADC\_CTL\_ADCEN\_Msk; //ADC 啓用

EADC->CTL &= ~EADC\_CTL\_DIFFEN\_Msk;

EADC->CTL &= ~EADC\_CTL\_SMPTSEL\_Msk;

EADC->CTL |= EADC\_CTL\_SMPTSEL8; /ADC 取樣時間

EADC->SCTL[0] &= ~EADC\_SCTL\_TRGSEL\_Msk;

EADC->SCTL[0] |= EADC\_SOFTWARE\_TRIGGER;

EADC->SCTL[0] &= ~EADC\_SCTL\_CHSEL\_Msk;

EADC->SCTL[0] |= 6 << EADC\_SCTL\_CHSEL\_Pos;

SYS->GPB\_MFPH &= ~SYS\_GPB\_MFPH\_PB9MFP\_Msk; //set channel 6 pin(PB9) as analog input

SYS->GPB\_MFPH |= SYS\_GPB\_MFPH\_PB9MFP\_EADC\_CH6;

PB->DINOFF |= BIT9; //disable digital input path

SYS->IVSCTL |= SYS\_IVSCTL\_VTEMPEN\_Msk; //啓用溫度感應器

ADCState = 0; //inititial adc agent state

}

void ADC\_Task(void){

static uint32\_t ADCOldCount = 0;

if((uint32\_t)(timecount - ADCOldCount) < 10)

return;

ADCOldCount = timecount;

switch(ADCState){

case 0:/\* 程式觸發VR和溫度感應器\*/

EADC->SWTRG |= (BIT0 | BIT17);

ADCState = 1;

break;

case 1:/\* 等待所有轉換完成 \*/

if(EADC->SWTRG == 0){

ADCState = 2;

}

break;

case 2:/\*獲取數據\*/

VRBuf = (EADC->DAT[0] & 0xFFF);

TempBuf = (EADC->DAT[17] & 0xFFF);

ADCState = 0;

break;

default:

ADCState = 0;

}

}

uint8\_t ADC\_GetTemperature(void){/\*讀取溫度數據\*/

/\* Vtemp(mV) = Gain(mV/C) \* Temperature(C) + Offset(mV)

\*\* Gain = 1.672(mV/C)

\*\* Offset = 740 (mV)

\*\* Vref+ = Vcc = 3.3V

\*\* Vref- = 0V

\*\* Resolution = 2^12 = 4096LSB

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\* Vtemp(mV) = (3300(mV) - 0(mV))\*(Dtemp(LSB) / 4096(LSB))

\*\* Temperature(C) = (Vtemp(mV) - Offset(mV)) / Gain(mV/C)

\*\* = (( (3300 - 0) \* (Dtemp / 4096)) - 740) / - 1.672

\*\* = ((3300 \* Dtemp / 4096) - 740) / -1.672

\*\* = (740 - (3300 \* Dtemp / 4096)) / 1.672

\*\* = (740000 - (3300000 \* Dtemp / 4096)) / 1.672

\*/

return (740000 - (TempBuf \* (3300000 >> 12))) / 1672;

}

uint8\_t ADC\_GetVR(void){/\*讀取VRBuf 數據\*/

return VRBuf >> 4;

}

2.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

#include "ADCAgent.h"//加入自訂ADC的標頭檔

uint32\_t timecount;

uint8\_t dir;//方向

uint32\_t speed;//速度

uint8\_t mtr\_speed;//速度（rpm）

int DisplayMode;//顯示模式

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

void DisplayTask (void);

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

ADC\_Initial（）；//呼叫ADC初始化函式

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

dir = 1;

speed = 0;

while(1){

if(Btn\_IsOneShot(0x01) == 0x01){

DisplayMode=1;//1=顯示溫度

//clear one-shot flag

Btn\_OneShotClear(0x01);

}

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

DisplayMode=2;//2=顯示速度

Btn\_OneShotClear(0x02);

}

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

DisplayMode=3;//3=顯示方向

dir=!dir;

Btn\_OneShotClear(0x04);

//Btn\_OneShotClear(0x04);

}

/\*將類比轉換為數位\*/

ADC\_Task();

mtr\_speed=ADC\_GetVR();

speed=StepMtr\_RPMtoD(mtr\_speed);

DisplayTask();/\* 寫進7\_seg buffer \*/

StepMtr\_Task(dir,speed);//把方向速度給馬達

Btn\_Task();

\_7Seg\_Task();/\* 7 segment 輸出 \*/

}

}

void DisplayTask (void){

static uint32\_t Dpoldcount=0;

if((uint32\_t)timecount-Dpoldcount<1000)return;

Dpoldcount=timecount;

switch(DisplayMode)

{case 1://1=顯示溫度

\_7Seg\_WriteBuf((ADC\_GetTemperature()/10)%10, ADC\_GetTemperature()%10);

break;

case 2://2=顯示速度

\_7Seg\_WriteBuf((mtr\_speed/10)%10,mtr\_speed%10);

break;

case 3://3=顯示方向

\_7Seg\_WriteBuf(dir/10,dir);

break;

default:

\_7Seg\_WriteBuf((ADC\_GetTemperature()/10)%10, ADC\_GetTemperature()%10);

}

}

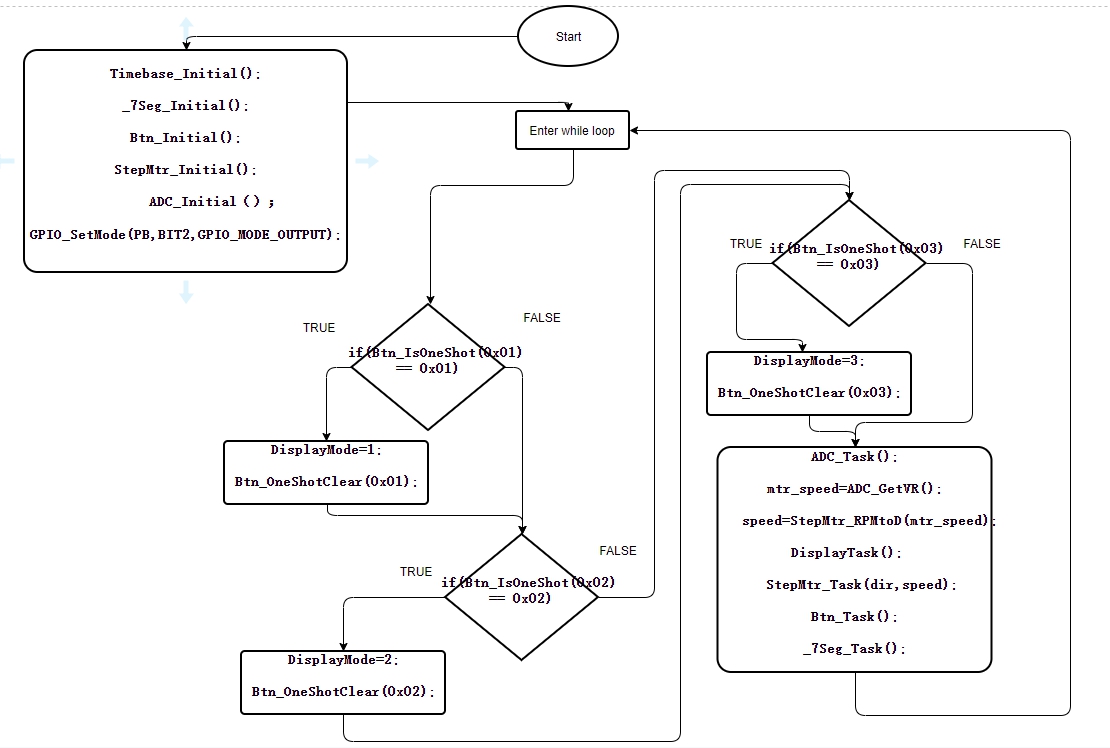
uint32\_t StepMtr\_RPMtoD(uint8\_t rpm)

{

return rpm?6000/rpm:0;

}

程式碼流程



心得

這次實驗是第四次使用微控器，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。這次不同的是有ADC，將類比轉換為數位，也是用state machine 的概念去控制轉換訊號，等待所有轉換完成才讀取溫度數據。這次實驗也不難，主要目的為結合上三次實驗的概念再加以利用，實驗過程大致順利。