EIE3810 Microprocessor System Design Laboratory

**Laboratory Report #5**

Name: 关嘉俊

Student ID: 116010060

Date: April 15th

The Chinese University of Hong Kong, Shenzhen

 Experiment A: e.g. Learn how to set up the TIM3 first, and make use of the interrupt handler for controlling the LED0.

 Experiment B: e.g. Similarly, learn how to set up the TIM4, which could control the LED1 together with the TIM3.

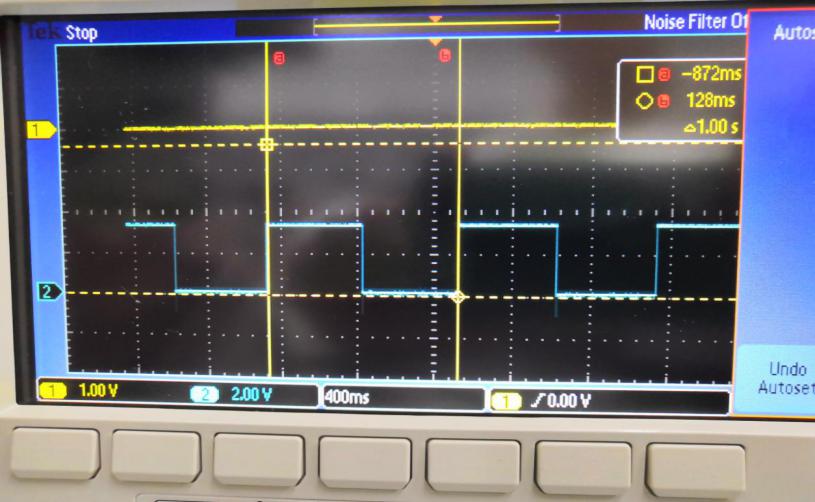
 Experiment C: e.g. Instead of using two timers for two individual processes, we setup SYSTICK with a 10ms periodic tick to control the two LEDs,

 Experiment D: e.g. First set up the related register, including the compare register, and then drive an LED with PWM.

 Experiment E: e.g. Achieve reading function of the JOYPAD key input based on TIM3 interrupt handler.

**I. Experiment A**

1. Basic procedure: 1. Type the program code and download it in the board. 2.Measure the DS0 flashing period with an oscilloscope 3.Try to change other flashing frequency.
2. Raw Data:



In the figure we can see that, for the blue line, its period is 1s, the timer interrupt set up the DS0 successfully.

1. Question:

For the LED flashing period, it is equal to 72M/pre-scalar value+1/reload-value+1\*2

1. Source Code:

#include "stm32f10x.h"

#include "EIE3810\_LED.h"

#include "EIE3810\_clock.h"

void EIE3810\_Key2\_EXTIInit(void);

void EIE3810\_TIM3\_Init(u16 arr, u16 psc);

void EIE3810\_NVIC\_SetPriorityGroup(u8 prigroup);

int main(void)

{

EIE3810\_clock\_tree\_init();

EIE3810\_LED\_Init();

EIE3810\_NVIC\_SetPriorityGroup(5); //Set PRIGROUP

EIE3810\_TIM3\_Init(4999,7199); //set arr to 4999 and psc to 7199 to flash in 1Hz

//EIE3810\_TIM3\_Init(2499,7199);//set arr to 2499 and psc to 7199 to flash in 2Hz

while(1)

{

;

}

}

void EIE3810\_TIM3\_Init(u16 arr, u16 psc)

{

//TIM3

RCC->APB1ENR|=1<<1;//enable TIM#

TIM3->ARR=arr;//set TIM3 auto-reload register

TIM3->PSC=psc;//set prescaler register

TIM3->DIER|=1<<0;//TIM3 update interrup enable

TIM3->CR1|=0x01;//couter enable

NVIC->IP[29]=0x45;//set the priority of TIM3 interrupt to 0100

NVIC->ISER[0]=(1<<29);//enable interrupt #29

}

void EIE3810\_NVIC\_SetPriorityGroup(u8 prigroup)

{

u32 temp1, temp2;

temp2 = prigroup&0x00000007;

temp2 <<=8;

temp1 = SCB->AIRCR;

temp1 &= 0x0000F8FF;

temp1 |= 0x05FA0000;

temp1 |= temp2;

SCB->AIRCR = temp1;

}

void TIM3\_IRQHandler(void)

{

if (TIM3->SR & 1<<0)//if update interrupt pending

{

GPIOB->ODR ^=1<<5;//toggle LED 0 by XOR a 1

}

TIM3->SR &= ~(1<<0);//get rid of the pending

}

**II. Experiment B**

1. Procedure: 1.Set up TIM4 to control DS1 to have a flashing rate of 4Hz. Keep the TIM3’s work. 2.Comple and run the program. 3)Achieve that the DS0 is on when DS1 is off
2. Raw Data:

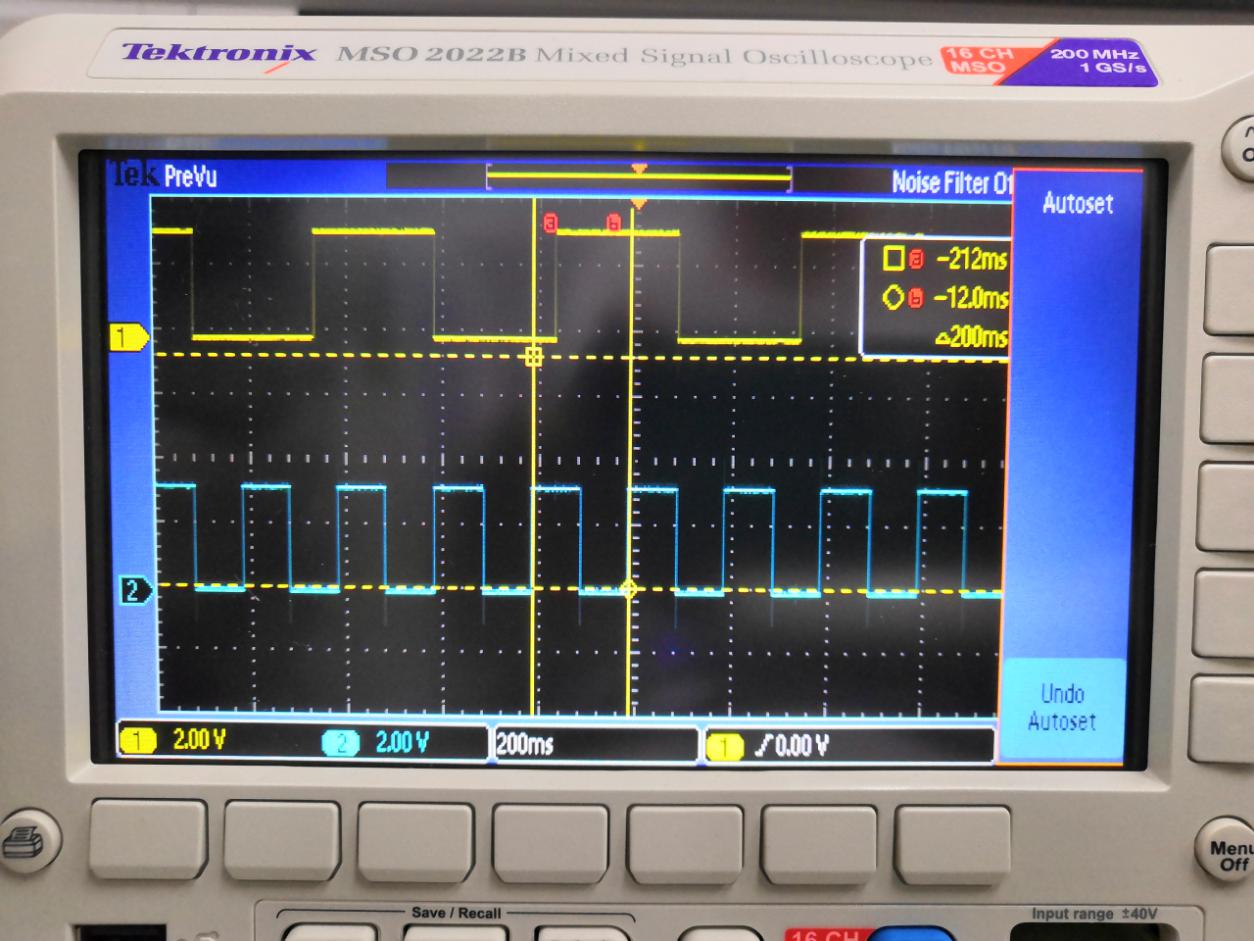
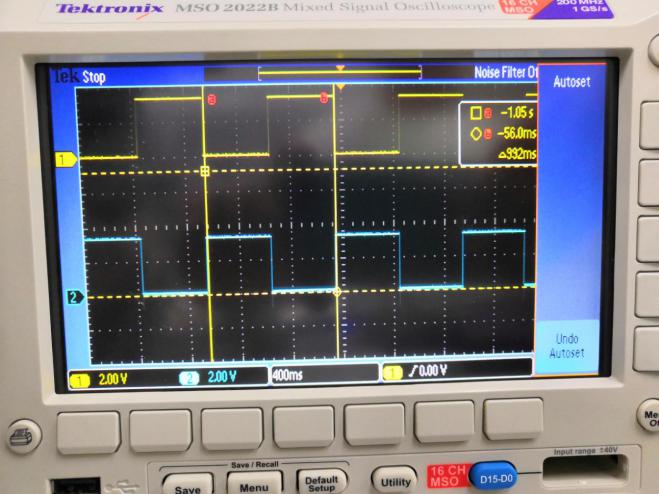
** **

Figure 1 Ds1flash with 4Hz, Ds0 flash with 1 Hz Figure 2 DS0 is on when Ds1 is off

1. Source Code

#include "stm32f10x.h"

#include "EIE3810\_LED.h"

#include "EIE3810\_clock.h"

void EIE3810\_Key2\_EXTIInit(void);

void EIE3810\_TIM3\_Init(u16 arr, u16 psc);

void EIE3810\_NVIC\_SetPriorityGroup(u8 prigroup);

void EIE3810\_TIM4\_Init(u16 arr, u16 psc);

int main(void)

{

EIE3810\_clock\_tree\_init();

EIE3810\_LED\_Init();

EIE3810\_NVIC\_SetPriorityGroup(5); //Set PRIGROUP

EIE3810\_TIM3\_Init(4999,7199); //set arr to 4999 and psc to 7199 to flash in 1Hz

//EIE3810\_TIM3\_Init(2499,7199);//set arr to 2499 and psc to 7199 to flash in 2Hz

EIE3810\_TIM4\_Init(1249,7199);

while(1)

{;

}

}

void EIE3810\_TIM3\_Init(u16 arr, u16 psc)

{

//TIM3

RCC->APB1ENR|=1<<1;//enable TIM3

TIM3->ARR=arr;//set TIM3 auto-reload register

TIM3->PSC=psc;//set prescaler register

TIM3->DIER|=1<<0;//TIM3 update interrup enable

TIM3->CR1|=0x01;//couter enable

NVIC->IP[29]=0x45;//set the priority of TIM3 interrupt to 0100

NVIC->ISER[0]=(1<<29);//enable interrupt #29}

void EIE3810\_TIM4\_Init(u16 arr, u16 psc)

{

//TIM3

RCC->APB1ENR|=1<<2;//enable TIM3

TIM4->ARR=arr;//set TIM3 auto-reload register

TIM4->PSC=psc;//set prescaler register

TIM4->DIER|=1<<0;//TIM3 update interrup enable

TIM4->CR1|=0x01;//couter enable

NVIC->IP[30]=0x45;//set the priority of TIM3 interrupt to 0100

NVIC->ISER[0]=(1<<30);//enable interrupt #29

}

void EIE3810\_NVIC\_SetPriorityGroup(u8 prigroup)

{

u32 temp1, temp2;

temp2 = prigroup&0x00000007;

temp2 <<=8;

temp1 = SCB->AIRCR;

temp1 &= 0x0000F8FF;

temp1 |= 0x05FA0000;

temp1 |= temp2;

SCB->AIRCR = temp1;

}

void TIM3\_IRQHandler(void)

{

if (TIM3->SR & 1<<0)//if update interrupt pending

{

GPIOB->ODR ^=1<<5;//toggle LED 0 by XOR a 1s

}

TIM3->SR &= ~(1<<0);//get rid of the pending

}

void TIM4\_IRQHandler(void)

{

if (TIM4->SR & 1<<0)//if update interrupt pending

{

GPIOE->ODR ^=1<<5;//toggle LED1 by XOR a 1

}

TIM4->SR &= ~(1<<0);//get rid of the pending

}

/\*\* For the additional experiment, the code is shown as followed:

void TIM4\_IRQHandler(void)

{

if (TIM4->SR & 1<<0)

{

if ((GPIOB-> ODR & 0x0020) == 0x0020)

{

GPIOE->BRR = 1<<5;

}

else{

GPIOE->BSRR = 1<<5;

}

}

}\*\*/

**III. Experiment C**

1. Procedure: (1) Instead of using two timers, we apply one timer and achieve multi-tasking by the 24-bit timer SYSTICL (2) Type and complete the function that achieve the goal. Include it into the main.c. (3) Modify the SysTick\_Handler in the stm32f10x\_it.c (4) Mody the code to achieve the function
2. Raw Data:

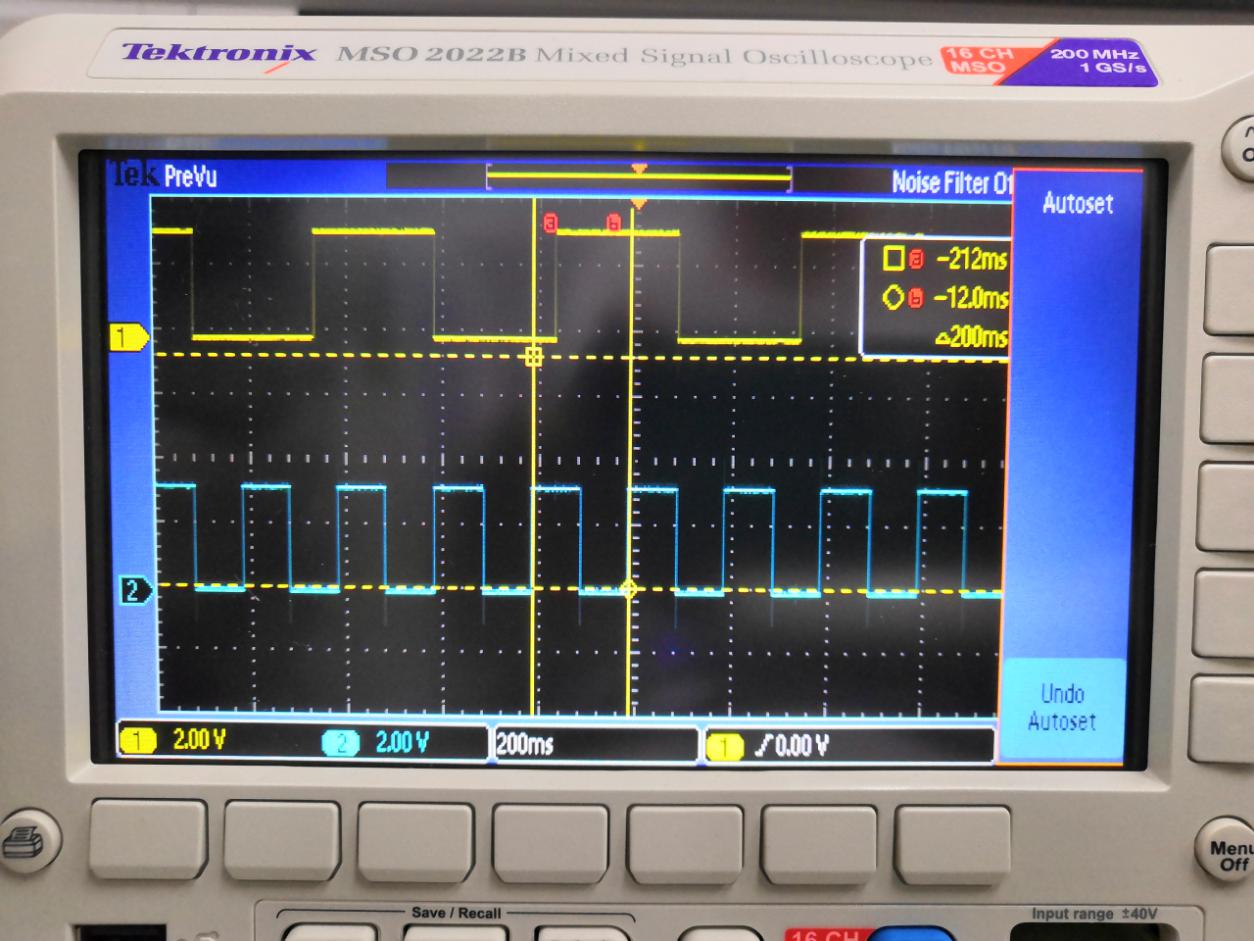
** **

Figure 3 Ds1 flash 5 times one second Figure 4 Ds0 flash w times one second

1. Source Code\

#include "stm32f10x.h"

#include "stm32f10x\_it.h"

#include "EIE3810\_LED.h"

#include "EIE3810\_clock.h"

u8 task1HeartBeat;

u8 LED0couter = 0;

u8 LED1couter = 0;

void EIE3810\_SYSTICK\_Init(void);

void EIE3810\_Key2\_EXTIInit(void);

void EIE3810\_NVIC\_SetPriorityGroup(u8 prigroup);

int main(void){

task1HeartBeat=0;

EIE3810\_clock\_tree\_init();

EIE3810\_SYSTICK\_Init();

EIE3810\_NVIC\_SetPriorityGroup(5);

EIE3810\_LED\_Init();

while(1){

if (task1HeartBeat>=10){

task1HeartBeat = 0;

if (LED0couter==10){

GPIOB->ODR ^=1<<5;//toggle LED 0 by XOR a 1

LED0couter = 0;

}

if (LED1couter==25){

GPIOE->ODR ^=1<<5;//toggle LED 0 by XOR a 1

LED1couter = 0;

}

LED0couter++;

LED1couter++;

}

}

}

void EIE3810\_SYSTICK\_Init()

{

//SYSTICK

SysTick->CTRL = 0;//Clear SysTick->CTRL setting

SysTick->LOAD = 8999;//tick every 9000 clock pulses,72000000/8/9000=1000

SysTick->CTRL = 0x00010003; //COUNTFLAG set to 1; clocksource=0,FCLK/8;

//counter enable with pending

}

void EIE3810\_NVIC\_SetPriorityGroup(u8 prigroup)

{

u32 temp1, temp2;

temp2 = prigroup&0x00000007;

temp2 <<=8;

temp1 = SCB->AIRCR;

temp1 &= 0x0000F8FF;

temp1 |= 0x05FA0000;

temp1 |= temp2;

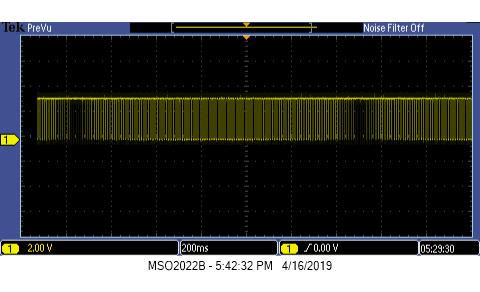
SCB->AIRCR = temp1;

}

**(D）**Question: In order to change the SysTick frequency to 200Hz, we need to change the pre-scalar value or counter value. In this experiment, the original frequency is 100Hz. Therefore, we need to set SysTick\_LOAD register. The original setting is 0xFF015F8F. The last 5 bits stands for the pre-scalar value 0x15F8F = 89999. So the frequency = 72000000/((89999+1)\*8) = 100Hz. In order to increasing the frequency to 200Hz, we need to change the pre-scalar value to 44999 = 0xAFC8. So the register should change to 0xFF00AFC8.

**IV. Experiment D**

1. Procedure: 1) Type EIE3810\_TIM3\_PWMInit() into the main.c, we will use a PWM signal that is generate by TIM3 to drive LED DS0 with various brightness. 2)Measure the output of the LED DS0 with an oscilloscope.
2. Raw Data:



1. Question: When LED0\_PWM\_VAL is 0, LED0 is turned off because the brightness is 0. In oscilloscope, the signal of PB5 is always high.When LED0\_PWM\_VAL is 9999, LED brightness is set to 9999, so the LED is very shining. The signal in oscilloscope is almost in low voltage with only little high pulse.
2. Source Code

#include "stm32f10x.h"

#include "stm32f10x\_it.h"

#include "EIE3810\_LED.h"

#include "EIE3810\_clock.h"

#define LED0\_PWM\_VAL TIM3->CCR2

void Delay(u32);

void EIE3810\_TIM3\_PWMInit(u16 arr, u16 psc);

int main(void)

{

u16 LED0PWMVal=0;

u8 dir=1;

EIE3810\_clock\_tree\_init();

EIE3810\_LED\_Init();

EIE3810\_TIM3\_PWMInit(9999, 71);//100Hz

while(1)

{

Delay(1500);

if (dir) LED0PWMVal++; //increae the value which is going to be loaded CCR2

else LED0PWMVal--;//decrease the value which is going to be loaded in CCR2

if (LED0PWMVal > 5000) dir=0;//start to decrease

if (LED0PWMVal ==0) dir=1;//start to increase

LED0\_PWM\_VAL = LED0PWMVal;//set the value of TIM\_CCR2

}

}

void Delay(u32 count)

{

u32 i;

for (i=0;i<count;i++);

}

void EIE3810\_TIM3\_PWMInit(u16 arr, u16 psc)

{

//load TIM3\_CCR2 to actual compare 2 register

RCC->APB2ENR |=1<<3;//enable GPIOB

GPIOB->CRL &= 0xFF0FFFFF;//clear bit5

GPIOB->CRL |= 0x00B00000;//set PB5 to 1011, 50MHz alternate function output Push-pull

RCC->APB2ENR |= 1<<0;//enable AIFO

AFIO->MAPR &= 0xFFFFF3FF;//clear bit TIM3\_REMAP which is at [11:10]

AFIO->MAPR |= 1<<11;//set TIM3\_REMAP to '10'

RCC->APB1ENR |= 1<<1;//enable TIM3

TIM3->ARR = arr;//set auto-reload register

TIM3->PSC = psc;//set prescaler register

TIM3->CCMR1 |= 7<<12;//set output compare 2 mode as 0111.

TIM3->CCMR1 |= 1<<11;//set OC2PE to 1, to enable output compare 2 preload

TIM3->CCER |= 1<<4;//enable compare 2 output

TIM3->CR1 = 0x0080;//enable auto-reload preload

TIM3->CR1 |= 1<<0; //enable counter

}

**V. Experiment E**

1. Procedure: (1) Complete the JOYPAD initiation function and reading function. (2) Modify the handler to connect the JOYPAD with the TIM3 at a frequency of 100Hz.
2. Raw Data:



Figure 5 Pressng start would show on the LCD screen



Figure 6 Pressing down would also show on the screen

1. Source Code:

#include "stm32f10x.h"

#include "stm32f10x\_it.h"

#include "EIE3810\_LED.h"

#include "EIE3810\_Clock.h"

#include "EIE3810\_TFTLCD.h"

void Delay(u32);

void JOYPAD\_Init(void);

void JOYPAD\_Delay(u16 t);

u8 JOYPAD\_Read(void);

void EIE3810\_TIM3\_Init(u16 arr, u16 psc);

void EIE3810\_NVIC\_SetPriorityGroup(u8 prigroup);

int main(void){

EIE3810\_clock\_tree\_init();

EIE3810\_LED\_Init();

EIE3810\_TFTLCD\_Init();

EIE3810\_NVIC\_SetPriorityGroup(5);

JOYPAD\_Init();

EIE3810\_TIM3\_Init(99, 7199);//100Hz

EIE3810\_TFTLCD\_DrawAll(0,0,WHITE);

while(1)

{;}

void JOYPAD\_Init(void)

{

RCC->APB2ENR|=1<<3; //Enable GPIOB

RCC->APB2ENR|=1<<5; //Enable GPIOE

GPIOB->CRH&=0XFFFF00FF; //Clear the bit of the pin11, pin10

GPIOB->CRH|=0X00003800; //Set the pin10 as the Input with pull-ip mode,

//and the pin11 with the general purpose output push-pull

GPIOB->ODR|=3<<10; //initialize PB10 PB11 to 1

GPIOD->CRL&=0XFFFF0FFF; //Clear the bit3

GPIOD->CRL|=0X00003000; //Set up pin bit3 as the input with pull-up/pull-down

GPIOD->ODR|=1<<3; //Enable the input pull-up mode

}

void JOYPAD\_DELAY(u16 t)

{while(t--);}

u8 JOYPAD\_Read(void)

{

vu8 temp=0;

u8 t;

GPIOB->BSRR |= 1<<11; //Set PB11 to high at the start of reading

Delay(80); //Delay for a while

GPIOB->BSRR |= 1<<27; //Set PB27 to high at the start of reading

for(t=0;t<8;t++)

{

temp>>=1; //Move the temp to one right digit

if((((GPIOB->IDR)>>10)&0x01)==0) temp|=0x80; //detect a low voltage of PB10 and set bit 7-t to 1

GPIOD->BSRR |= (1<<3);

Delay(80);//generate a high voltage in PD3 lasted delay(80)

GPIOD->BSRR |=(1<<19);

Delay(80);//generate a low voltage in PD3 lasted delay(80)

}

return temp;

}

void EIE3810\_TIM3\_Init(u16 arr, u16 psc)

{

//TIM3

RCC->APB1ENR|=1<<1;//enable TIM#

TIM3->ARR=arr;//set TIM3 auto-reload register

TIM3->PSC=psc;//set prescaler register

TIM3->DIER|=1<<0;//TIM3 update interrup enable

TIM3->CR1|=0x01;//couter enable

NVIC->IP[29]=0x45;//set the priority of TIM3 interrupt to 0100

NVIC->ISER[0]=(1<<29);//enable interrupt #29

}

void EIE3810\_NVIC\_SetPriorityGroup(u8 prigroup)

{

u32 temp1, temp2;

temp2 = prigroup&0x00000007;

temp2 <<=8;

temp1 = SCB->AIRCR;

temp1 &= 0x0000F8FF;

temp1 |= 0x05FA0000;

temp1 |= temp2;

SCB->AIRCR = temp1;

}

void TIM3\_IRQHandler(void)

{

u8 temp = 0;

if (TIM3->SR & 1<<0){//if update interrupt pending

temp = JOYPAD\_Read();

if ((temp)&0x01){

EIE3810\_TFTLCD\_DrawAll(0,0,WHITE);

EIE3810\_TFTLCD\_ShowChar(200,200,'A',BLUE,WHITE);

}

if ((temp>>1)&0x01){

EIE3810\_TFTLCD\_DrawAll(0,0,WHITE);

EIE3810\_TFTLCD\_ShowChar(200,200,'B',BLUE,WHITE);

}

if ((temp>>2)&0x01){

EIE3810\_TFTLCD\_DrawAll(0,0,WHITE);

EIE3810\_TFTLCD\_ShowChar(200,200,'S',BLUE,WHITE);

EIE3810\_TFTLCD\_ShowChar(210,200,'E',BLUE,WHITE);

EIE3810\_TFTLCD\_ShowChar(220,200,'L',BLUE,WHITE);

EIE3810\_TFTLCD\_ShowChar(230,200,'E',BLUE,WHITE);

EIE3810\_TFTLCD\_ShowChar(240,200,'C',BLUE,WHITE);

EIE3810\_TFTLCD\_ShowChar(250,200,'T',BLUE,WHITE);

}

if ((temp>>3)&0x01){

EIE3810\_TFTLCD\_DrawAll(0,0,WHITE);

EIE3810\_TFTLCD\_ShowChar(200,200,'S',BLUE,WHITE);

EIE3810\_TFTLCD\_ShowChar(210,200,'T',BLUE,WHITE);

EIE3810\_TFTLCD\_ShowChar(220,200,'A',BLUE,WHITE);

EIE3810\_TFTLCD\_ShowChar(230,200,'R',BLUE,WHITE);

EIE3810\_TFTLCD\_ShowChar(240,200,'T',BLUE,WHITE);

}

if ((temp>>4)&0x01){

EIE3810\_TFTLCD\_DrawAll(0,0,WHITE);

EIE3810\_TFTLCD\_ShowChar(200,200,'U',BLUE,WHITE);

EIE3810\_TFTLCD\_ShowChar(210,200,'P',BLUE,WHITE);

}

if ((temp>>5)&0x01){

EIE3810\_TFTLCD\_DrawAll(0,0,WHITE);

EIE3810\_TFTLCD\_ShowChar(200,200,'D',BLUE,WHITE);

EIE3810\_TFTLCD\_ShowChar(210,200,'O',BLUE,WHITE);

EIE3810\_TFTLCD\_ShowChar(220,200,'W',BLUE,WHITE);

EIE3810\_TFTLCD\_ShowChar(230,200,'N',BLUE,WHITE);

}

if ((temp>>6)&0x01){

EIE3810\_TFTLCD\_DrawAll(0,0,WHITE);

EIE3810\_TFTLCD\_ShowChar(200,200,'L',BLUE,WHITE);

EIE3810\_TFTLCD\_ShowChar(210,200,'E',BLUE,WHITE);

EIE3810\_TFTLCD\_ShowChar(220,200,'F',BLUE,WHITE);

EIE3810\_TFTLCD\_ShowChar(230,200,'T',BLUE,WHITE);

}

if ((temp>>7)&0x01){

EIE3810\_TFTLCD\_DrawAll(0,0,WHITE);

EIE3810\_TFTLCD\_ShowChar(200,200,'R',BLUE,WHITE);

EIE3810\_TFTLCD\_ShowChar(210,200,'I',BLUE,WHITE);

EIE3810\_TFTLCD\_ShowChar(220,200,'G',BLUE,WHITE);

EIE3810\_TFTLCD\_ShowChar(230,200,'H',BLUE,WHITE);

EIE3810\_TFTLCD\_ShowChar(240,200,'T',BLUE,WHITE);

}

}

TIM3->SR &= ~(1<<0);//get rid of the pending}