EIE3810 Microprocessor System Design Laboratory

Laboratory Report #6

Name: 关嘉俊

Student ID: 116010060

Date: May 5th

The Chinese University of Hong Kong, Shenzhen

- Experiment A: Simulate the example program and finish the mini game
- Experiment B: Improve the function.

I. Experiment A

(a) Basic procedure: 1.Initialize the CLOCK_TREE, TIM3, TIM4, JOYPAD, USART, TFTLCD, KEY, LED. 2. Draw the each showing page at first. 3. Continue after the player's selection of the difficulty. 4.Receive the number sent from the computer. Here we should measure the unique baud rate. 5. Finish the ball's movement function. 6.Complete the ball's bounce function. 7. Change the position of the board in the interrupt 8. Finally finish the counting of the round and time.

(b) Raw Data:

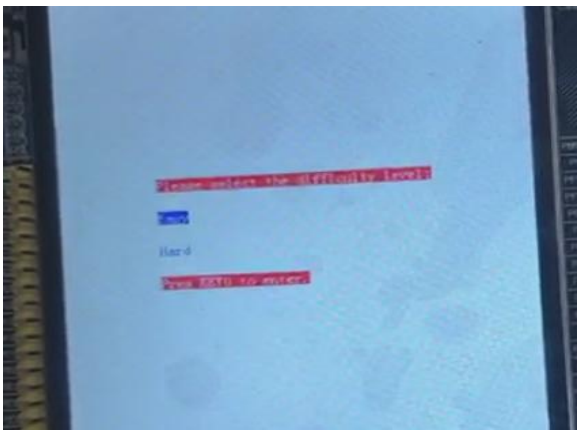

Figure 1 Difficult choice

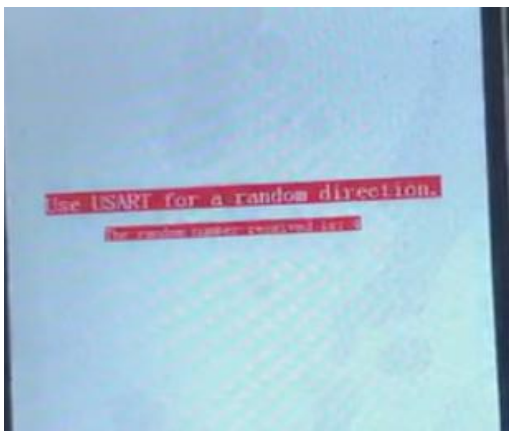

Figure 2 Receive the random number

Figure 3 Ball's movement

Figure 4 Ball's bounce

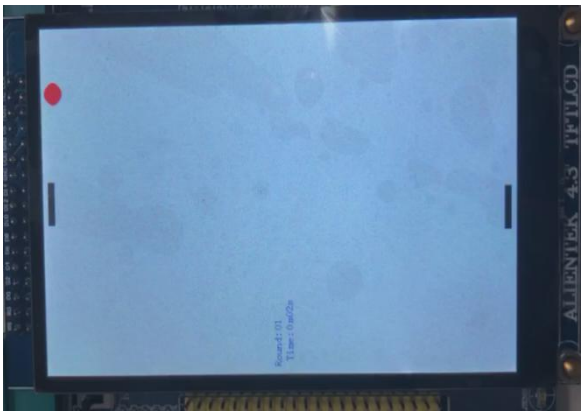

Figure 5 Heating the bound without catching

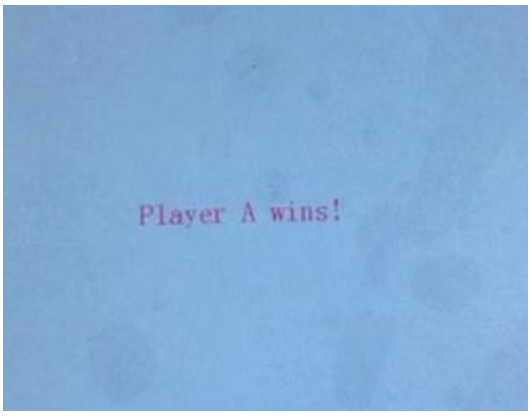

Figure 6 GameOver Showing

II. Experiment B

(a) Basic procedure:

1. To make the game more competitive, two functions are added. 2. The first function is raising power. In this function, when the joypad player keep pressing the down key or the board player pressing the Key1, they would generate certain level acceleration. There are total 5 levels, corresponding to five seconds of the pressing time 3.The second function is to change the ball's direction. In this function, when the joypad player pressing the up key or the board player pressing the Keyup, when they catch the ball at the next, they would change the balls direction and make it harder to be caught by the opponent. To realize these two functions, we just need to initialize the global variable, and in the Timer3 interrupt the pressing of the key would change the corresponding value.

(b) Raw Material:

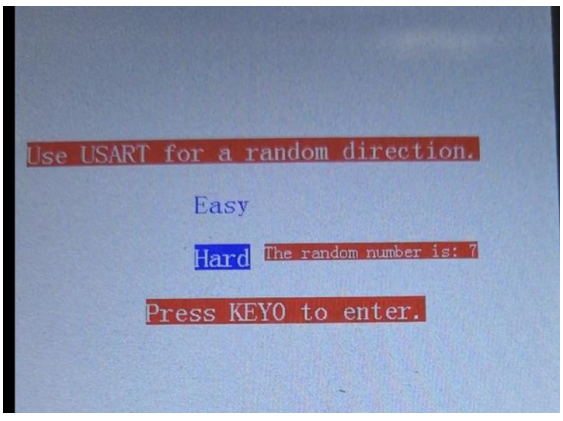

Figure 7 Select the mode and receive the number

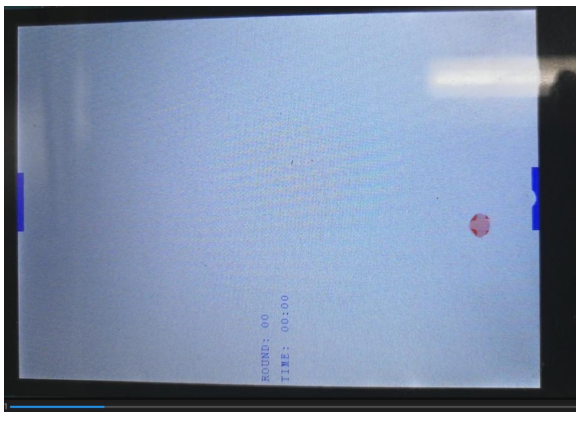

Figure 8 Game start

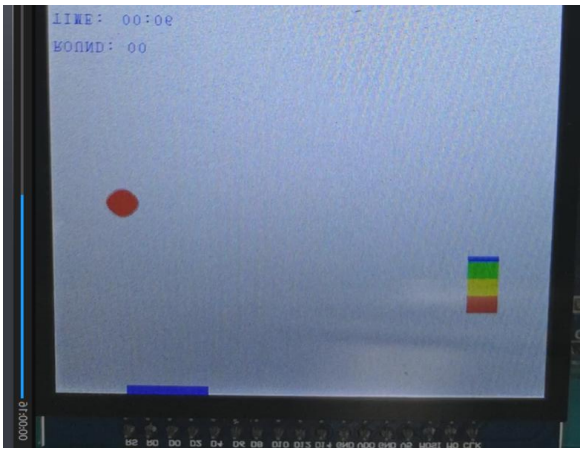

Figure 9 Pressing the down key to increase power

Figure 10 Maximum power will show “boom”

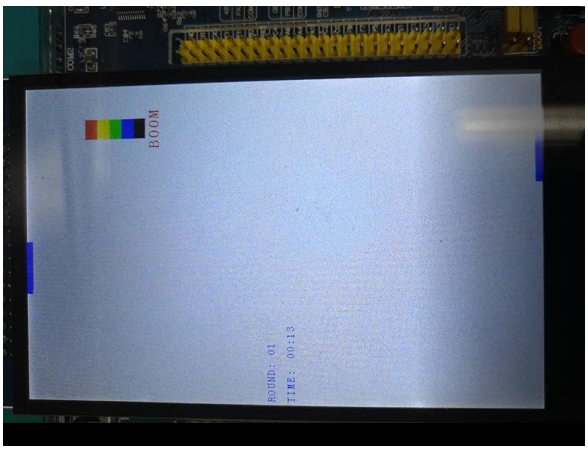

Figure 11 Press start will pause the program

Figure 12 press keyup to change direction

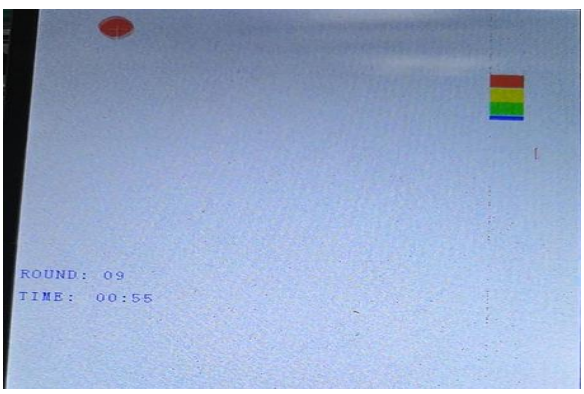

Figure 13 After rasing power, the ball just use 2s to reach from top to bottom Figure 14 Over

The end~~

C.Source Code: (Just show some key steps)

```
int main(void){
 EIE3810_clock_tree_init();
 EIE3810_LED_Init();
 EIE3810_Buzzer_Init();
 EIE3810_TFTLCD_Init();
 EIE3810_NVIC_SetPriorityGroup(5);
 JOYPAD_Init();
 EIE3810_USART1_init(72,4800);
 EIE3810_TFTLCD_DrawAll(0,0,WHITE);
 EIE3810_Key_Init();
 EIE3810_TIM3_Init(49, 7199);
 EIE3810_TIM4_Init(999, 7199);

 while(1){
 start= 0;
 bux = 200;
 bdx = 200;
 round = 0;
 uppower = 0;
 downpower = 0;
 second = 0;
 minute = 0;
 EIE3810_TFTLCD_DrawAll(0,0,WHITE);
 Show_first_page();
 Delay(20000000);
 EIE3810_TFTLCD_FillRectangle(0,480,200,400,WHITE);
 Show_second_page();
 Delay(5000000);
 int level = Choose_difficulty();
 EIE3810_TFTLCD_FillRectangle(0,480,200,500,WHITE);
 Delay(5000);
 Show_third_page();
 Delay(20000000);
  //upperright(120,700,20,7,RED,5);
 EIE3810_TFTLCD_FillRectangle(0,480,790,10,WHITE);
 EIE3810_TFTLCD_FillRectangle(0,480,0,10,WHITE);
 EIE3810_TFTLCD_FillRectangle(200,80,790,10,BLUE);
  EIE3810_TFTLCD_FillRectangle(200,80,0,10,BLUE);
 u16 dir =  ReceiveNumber();
 char t1[]={'T','h','e',' ','r','a','n','d','o','m',' ','n','u','m','b','e','r',' ','i','s',':',' ',dir+48};
 for(int i=0;i< sizeof(t1);i++)
  {
```

```
   EIE3810_TFTLCD_ShowChar(400-8*i, 300, t1[sizeof(t1)-i-1], WHITE, RED);
  }
 Delay(20000000);
 EIE3810_TFTLCD_FillRectangle(10,460,200,300,WHITE);
 Delay(5000);
 Show_fourth_page();
  start = 1;
 if (level == 1){
  upperleft(240,780,15,1,RED,4);
  }
 if (level == 0){
  upperleft(240,780,15,1,RED,4);
 }
}
}
int Choose_difficulty(void)
{
 while (1) {
 int level = 1;
 if(readKey_up()==0x0001)                  //   choose easy
   {
    char t2[]={'E','a','s','y'};
    char t3[]={'H','a','r','d'};
    for(int i=0;i< sizeof(t2);i++)
    {
     EIE3810_TFTLCD_ShowChar2412(200-12*i, 250, t2[sizeof(t2)-i-1], WHITE, BLUE);
    }
    for(int i=0;i< sizeof(t3);i++)
    {
     EIE3810_TFTLCD_ShowChar2412(200-12*i, 300, t3[sizeof(t3)-i-1], BLUE , WHITE);
    }
    //multiplier=1;
    int level = 0;
   }
   if(readKey1()==0x0000)                  //   choose hard
   {
    char t2[]={'E','a','s','y'};
    char t3[]={'H','a','r','d'};
    for(int i=0;i< sizeof(t2);i++)
    {
     EIE3810_TFTLCD_ShowChar2412(200-12*i, 250, t2[sizeof(t2)-i-1], BLUE, WHITE);
    }
    for(int i=0;i< sizeof(t3);i++)
    {
     EIE3810_TFTLCD_ShowChar2412(200-12*i, 300, t3[sizeof(t3)-i-1], WHITE , BLUE);
    }
```

```
    int level = 1;
   }
   if(readKey0()==0x0000)
   {
    if (level == 0){
       return(0);
    }
    if (level == 1){
      return(1);
    }
    break;
   }
  }
 }
u16 ReceiveNumber(void)
{
 u32 buffer;
 while(1){
 if (USART1->SR & (1<<5))
 {
    buffer = USART1->DR;
   if(buffer==0){
    return (0);
    break;
   }
   else if (buffer ==1){
    return (1);
    break;
   }

 }
}

void Show_fourth_page(void)
{
 int r1 = round/10;
 int r2 = round%10;
 EIE3810_TFTLCD_FillRectangle(80,30,380,10,WHITE);
char m[]={'R','O','U','N','D',':',' ',r1+48,r2+48};
  for(int i=0;i< sizeof(m);i++)
 {
  EIE3810_TFTLCD_ShowChar(5+10*i, 385, m[i], BLUE, WHITE);
 }
 char h[]={'T','I','M','E',':',' '};
 for(int i=0;i< sizeof(h);i++)
 {
```

```
  EIE3810_TFTLCD_ShowChar(5+10*i, 415, h[i], BLUE, WHITE);
 }
}
void upperleft(u16 x, u16 y, u8 r, u16 randnum, u16 color, u16 speed){
 int xc,yc,radius;
 xc = x;
 yc = y;
 radius = r;
 toggleBuzzer();
 Delay(800001);
 toggleBuzzer();
 while (1){

   if (xc >= radius && yc>=radius+10){
    EIE3810_TFTLCD_DrawCircle(xc,yc,r,1,color);
    Delay((10-speed)*20000);
    EIE3810_TFTLCD_DrawCircle(xc,yc,r,1,WHITE);
    if (randnum == 0){
     xc = xc - 1;
     yc = yc - 1;}
    if (randnum == 1){
     xc = xc - 1;
     yc = yc - 2;}
    if (randnum == 2){
     xc = xc - 1;
     yc = yc - 3;}
    if (randnum == 3){
     xc = xc - 2;
     yc = yc - 1;}
    if (randnum == 4){
     xc = xc - 2;
     yc = yc - 3;}
    if (randnum == 5){
     xc = xc - 4;
     yc = yc - 3;}
    if (randnum == 6){
     xc = xc - 2;
     yc = yc - 5;}
    if (randnum == 7){
     xc = xc - 3;
     yc = yc - 5;}}
   else if (xc<= radius && yc>radius+10){
     upperright(xc,yc,r,randnum,color,speed);
     break;
    }
   else if (yc<=radius+10){
```

```
      if (bux-5<=xc && bux>=xc-85){
      round +=1;
      downpower = 0;
      if (updir){
        int randir =rand()%7;
        downleft(xc,yc,r,randir,color,4+uppower/200);
        break;
        }
      else{
        downleft(xc,yc,r,randnum,color,4+uppower/200);
        break;}
      }
      else
      {
      EIE3810_TFTLCD_DrawAll(0,0,WHITE);
      char s1[]={'K','E','Y','B','O','A','R','D',' ','W','I','N','!'};
      for(int i=0;i< sizeof(s1);i++)
      {
       EIE3810_TFTLCD_ShowChar2412(400-12*i, 400, s1[sizeof(s1)-i-1], WHITE, BLUE);
      }
      Delay(20000000);
      EIE3810_TFTLCD_DrawAll(0,0,WHITE);
      Delay(500);
      break;
      }
    }
  }
}
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 TIM3_IRQHandler(void)
{
 u8 temp = 0;
 if (TIM3->SR & 1<<0){//if update interrupt pending

  temp = JOYPAD_Read();
  if ((temp>>2)&0x01){
   EIE3810_TFTLCD_ShowChar(200,200,'S',BLUE,WHITE);
  }
  if ((temp>>3)&0x01){
   EIE3810_TFTLCD_ShowChar(200,200,'S',BLUE,WHITE);
  }

  if ((temp>>4)&0x01){
  updir = 1;
  }
  if ((temp>>6)&0x01){
```

```
   EIE3810_TFTLCD_FillRectangle(bux+10,80,0,10,WHITE);
   if (bux >=0){
   bux-=1;}
   EIE3810_TFTLCD_FillRectangle(bux,80,0,10,BLUE);
  }
  if ((temp>>7)&0x01){
   EIE3810_TFTLCD_FillRectangle(bux-10,80,0,10,WHITE);
   if (bux <=400){
    bux+=1;}
    EIE3810_TFTLCD_FillRectangle(bux,80,0,10,BLUE);
  }
  if (readKey0() == 0x0000){
    downdir = 0;
    EIE3810_TFTLCD_FillRectangle(bdx-10,80,790,10,WHITE);
     if (bdx <=400){
        bdx+=1;}
     EIE3810_TFTLCD_FillRectangle(bdx,80,790,10,BLUE);
 }
  if (readKey2() == 0x0000){
    downdir = 0;
    EIE3810_TFTLCD_FillRectangle(bdx+10,80,790,10,WHITE);
     if (bdx >=0){
        bdx-=1;}
     EIE3810_TFTLCD_FillRectangle(bdx,80,790,10,BLUE);
 }
 }

 TIM3->SR &= ~(1<<0);//get rid of the pending
}
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]=0x35;//set the priority of TIM3 interrupt to 0100
NVIC->ISER[0]=(1<<30);//enable interrupt #29
}
void TIM4_IRQHandler(void)
{
 u8 temp = 0;
 if (start ==1){
  count+=1;
  Show_fourth_page();
```

```
 if (count == 10){
   EIE3810_TFTLCD_FillRectangle(70,30,380,60,WHITE);
  second +=1;
  count = 0;}
   if (second==60){
  minute+=1;
  second = 0;
  }
  int m1 = minute/10;
  int m2 = minute%10;
  int s1 = second/10;
  int s2 = second%10;
  EIE3810_TFTLCD_ShowChar(70,415,m1+48,BLUE,WHITE);
  EIE3810_TFTLCD_ShowChar(80,415,m2+48,BLUE,WHITE);
  EIE3810_TFTLCD_ShowChar(90,415,':',BLUE,WHITE);
  EIE3810_TFTLCD_ShowChar(100,415,s1+48,BLUE,WHITE);
  EIE3810_TFTLCD_ShowChar(110,415,s2+48,BLUE,WHITE);

  temp = JOYPAD_Read();
  if ((temp>>1)&0x01){
   EIE3810_TFTLCD_FillRectangle(200,100,390,410,WHITE);
   EIE3810_TFTLCD_ShowChar2412(200,400,'P',BLUE,WHITE);
   EIE3810_TFTLCD_ShowChar2412(215,400,'A',BLUE,WHITE);
   EIE3810_TFTLCD_ShowChar2412(230,400,'U',BLUE,WHITE);
   EIE3810_TFTLCD_ShowChar2412(250,400,'S',BLUE,WHITE);
   EIE3810_TFTLCD_ShowChar2412(270,400,'E',BLUE,WHITE);
    while(1)
     {
        temp = JOYPAD_Read();
       if ((temp)&0x01){
         EIE3810_TFTLCD_FillRectangle(200,100,390,410,WHITE);
        break;
           }}
  if (start==1){
    if (downpower==0){
      EIE3810_TFTLCD_FillRectangle(380,60,600,160,WHITE);
    }
    if (uppower==0){
      EIE3810_TFTLCD_FillRectangle(380,60,100,160,WHITE);
    }
  }
 }
TIM4->SR &= ~(1<<0);//get rid of the pending
}
```