

实验报告

课程名称: _____ DSP 的原理与应用

实验名称: _____ 实验 5.1 外中断

专业-班级: _____ 电气 1 班 学号: _____ 220330124 姓名: _____ 舒晟超

实验日期: _____ 2024 年 5 月 30 日

试验台号: _____

报告总分数: _____

教师评语:

助教签字: _____

教师签字: _____

日 期: _____

一、实验目的

1. 通过实验熟悉 TMS320F28335 DSP 的中断响应过程。
2. 学会 C 语言中断程序设计，以及运用中断程序控制程序流程。

二、实验过程

1. 阅读代码
2. 连接线路，编译并下载程序，进行硬件仿真
3. 观察实验结果

三、实验结果

1. 阅读代码

```
1. void Xint_configure(void)
2. {
3.     // 指定中断向量表中中断服务函数地址，指定 XINT3 线产生的中断对应的中断服务函数为 XINT3_isr
4.     EALLOW;
5.     PieVectTable.XINT3 = &XINT3_isr;
6.     EDIS;
7.     // 使能 CPU 中断 12
8.     IER |= M_INT12;
9.     // 使能外设对应的 PIE 中断（PIE 组 12 的 INT1）
10.    PieCtrlRegs.PIEIER12.bit.INTx1= 1;
11.    // PIEACK 位置 1 以等待中断
12.    PieCtrlRegs.PIEACK.all = 0xFFFF;
13.
14.    EALLOW;
15.    GpioCtrlRegs.GPBMUX2.bit.GPIO61=0;    // 使能 GPIO 引脚功能为通用 GPIO
16.    GpioCtrlRegs.GPBDIR.bit.GPIO61=0;    // 设置 GPIO 为输入方向
17.    GpioCtrlRegs.GPBQSEL2.bit.GPIO61=0;    // GPIO61 Synch to SYSCLKOUT
18.    GpioIntRegs.GPIOXINT3SEL.bit.GPIOSEL=61; // XINT3 是 GPIO61
19.    XIntruptRegs.XINT3CR.bit.POLARITY=0;    // 下降沿触发中断
20.    XIntruptRegs.XINT3CR.bit.ENABLE=1;    // 使能外设中断
21.    EDIS;
22.
23.    EINT;    // 开全局中断
24.    ERTM;    // 当使用仿真器调试时可开启 DEBUG 中断
25. }
```

2. 连接线路，编译并下载程序，进行硬件仿真

观察到：在键盘上任意按键按下时，南方 D3 交通灯亮起并在 0.1s 后熄灭，且亮起期间重复按键无效，长按按键不会使得 D3 交通灯常亮。实验现象在附件。

加入断点：

```
TI Resource Explorer  Lab310-Xint.c
34
35
36 void InitICETEK28335Ae(void)
37 {
38     InitSysCtrl();
39     InitXintf16Gpio();
40     DINT;
41     InitPieCtrl();
42     IER = 0x0000;
43     IFR = 0x0000;
44     InitPieVectTable();
45 }
46
47 interrupt void XINT3_isr(void)
48 {
49     PieCtrlRegs.PIEACK.all = PIEACK_GROUP12;
50     uKeyCode=1;
51 }
52
53 void Xint_configure(void)
54 {
55     EALLOW; // This is needed to write to EALLOW protected registers
56     PieVectTable.XINT3 = &XINT3_isr;
57     EDIS; // This is needed to disable write to EALLOW protected registers
58
59     IER |= M_INT12; // Enable CPU INT12:
60
61     PieCtrlRegs.PIEIER12.bit.INTx1= 1; //Enable TINT12 in the PIE: Group 12 interrupt 1
62     PieCtrlRegs.PIEACK.all = 0xFFFF; // Enables PIE to drive a pulse into the CPU
63
64     EALLOW;
65     GpioCtrlRegs.GPBMUX2.bit.GPIO61=0; // GPIO functionality GPIO61
66     GpioCtrlRegs.GPBDIR.bit.GPIO61=0; //GPIO61 are inputs
67     GpioCtrlRegs.GPBQSEL2.bit.GPIO61=0; // GPIO61 Synch to SYSCLKOUT
68     GpioIntRegs.GPIOXINT3SEL.bit.GPIOSEL=61; //Select GPIO61 interrupt input source
69     XIntruptRegs.XINT3CR.bit.POLARITY=0; //Positive edge triggered
70     XIntruptRegs.XINT3CR.bit.ENABLE=1; //enable
71     EDIS;
72
73     EINT; // Enable Global interrupt INTM
74     ERTM; // Enable Global realtime interrupt DBGM
75 }
76
```

在按键按下一瞬间，程序进入中断将 uKeycode 置为 1.实际上，在调试过程中，如果删去 ERTM 语句，则不会进入中断。

四、问题与思考

1、请修改程序完成按键中断控制的指示灯依次逐一点亮功能。（1 分）

程序如下，使用 led_flag 进行标记状态。

```
1. #include "DSP2833x_Device.h"
2. #include "DSP2833x_Examples.h"
3. #include "ICETEK-CTR.h"
4.
5. void InitICETEK28335Ae();
6. void Xint_configure(void);
7. interrupt void XINT3_isr(void);
8.
9. unsigned int led_flag=0;
10.
11. void main(void)
12. {
13.     int bSuccess;
14.
```

```
15. InitICETEKF28335Ae();//初始化: DSP 主频、GPIO、中断向量、中断使能
16. Xint_configure();    //外部中断配置
17.
18. bSuccess=ICETEKCTR_InitCTR(ICETEKCTRModeTeachingResearch); //初始化 ICETEK-CTR: 教
    研模式
19. while ( bSuccess ); // 如果初始化 ICETEK-CTR 错误, 停止运行, 可观察 bSuccess 取值查找初
    始化失败原因
20. ICETEKCTRGR=0x10;    //打开中断
21. ICETEKCTR_SetTrafficLEDStatus( 0 ,ICETEKCTRLEDSet);//熄灭交通灯
22.
23. for(;;){
24.     switch(led_flag)
25.     {
26.         case 0:
27.             ICETEKCTR_SetTrafficLEDStatus(0 ,ICETEKCTRLEDSet);
28.             break;
29.         case 1:
30.             ICETEKCTR_SetTrafficLEDStatus(ICETEKCTRLEDSouthRed,ICETEKCTRLEDSet);
31.             break;
32.         case 2:
33.             ICETEKCTR_SetTrafficLEDStatus(ICETEKCTRLEDSouthYellow,ICETEKCTRLEDSet);
34.             break;
35.         case 3:
36.             ICETEKCTR_SetTrafficLEDStatus(ICETEKCTRLEDSouthGreen,ICETEKCTRLEDSet);
37.             break;
38.         case 4:
39.             ICETEKCTR_SetTrafficLEDStatus(ICETEKCTRLEDWestRed,ICETEKCTRLEDSet);
40.             break;
41.         case 5:
42.             ICETEKCTR_SetTrafficLEDStatus(ICETEKCTRLEDWestYellow,ICETEKCTRLEDSet);
43.             break;
44.         case 6:
45.             ICETEKCTR_SetTrafficLEDStatus(ICETEKCTRLEDWestGreen,ICETEKCTRLEDSet);
46.             break;
47.         case 7:
48.             ICETEKCTR_SetTrafficLEDStatus(ICETEKCTRLEDNorthRed,ICETEKCTRLEDSet);
49.             break;
50.         case 8:
51.             ICETEKCTR_SetTrafficLEDStatus(ICETEKCTRLEDNorthYellow,ICETEKCTRLEDSet);
52.             break;
53.         case 9:
54.             ICETEKCTR_SetTrafficLEDStatus(ICETEKCTRLEDNorthGreen,ICETEKCTRLEDSet);
55.             break;
56.         case 10:
57.             ICETEKCTR_SetTrafficLEDStatus(ICETEKCTRLEDEastRed,ICETEKCTRLEDSet);
58.             break;
59.         case 11:
60.             ICETEKCTR_SetTrafficLEDStatus(ICETEKCTRLEDEastYellow,ICETEKCTRLEDSet);
```

```

61.     break;
62.     case 12:
63.         ICETEKCTR_SetTrafficLEDStatus(ICETEKCTRLEDEastGreen, ICETEKCTRLEDSet);
64.         break;
65.     default:
66.         ICETEKCTR_SetTrafficLEDStatus(0 , ICETEKCTRLEDSet);
67.         break;
68.     }
69.     ICETEKCTR_Delaysms(100);
70. }
71. }
72.
73.
74. void InitICETEK28335Ae(void)
75. {
76.     InitSysCtrl();
77.     InitXintf16Gpio();
78.     DINT;
79.     InitPieCtrl();
80.     IER = 0x0000;
81.     IFR = 0x0000;
82.     InitPieVectTable();
83. }
84.
85. interrupt void XINT3_isr(void)
86. {
87.     PieCtrlRegs.PIEACK.all = PIEACK_GROUP12;
88.     led_flag++;
89.     if(led_flag == 13)
90.     {
91.         led_flag = 0;
92.     }
93. }
94.
95. void Xint_configure(void)
96. {
97.     EALLOW; // This is needed to write to EALLOW protected registers
98.     PieVectTable.XINT3 = &XINT3_isr;
99.     EDIS; // This is needed to disable write to EALLOW protected registers
100.
101.     IER |= M_INT12; // Enable CPU INT12:
102.
103.     PieCtrlRegs.PIEIER12.bit.INTx1= 1; //Enable TINT12 in the PIE: Group 12 interrupt 1
104.     PieCtrlRegs.PIEACK.all = 0xFFFF; // Enables PIE to drive a pulse into the CPU

```

```
105.
106.  EALLOW;
107.  GpioCtrlRegs.GPBMUX2.bit.GPIO61=0;    // GPIO functionality GPIO61
108.  GpioCtrlRegs.GPBDIR.bit.GPIO61=0;    //GPIO61 are inputs
109.  GpioCtrlRegs.GPBQSEL2.bit.GPIO61=0;    // GPIO61 Synch to SYSCLKOUT
110.  GpioIntRegs.GPIOXINT3SEL.bit.GPIOSEL=61; //Select GPIO61 interrupt input source
111.  XIntruptRegs.XINT3CR.bit.POLARITY=0;    //Positive edge triggered
112.  XIntruptRegs.XINT3CR.bit.ENABLE=1;    //enable
113.  EDIS;
114.
115.  EINT;    // Enable Global interrupt INTM
116.  ERTM;    // Enable Global realtime interrupt DBGM
117. }
```

观察到当键盘按键，LED 灯逐一亮起并熄灭，以 12 次按键为循环。实验现象在附件。

实验报告

课程名称: DSP 的原理与应用

实验名称: 实验 5.2 通用输入输出管脚应用

专业-班级: 电气 1 班 学号: 220330124 姓名: 舒晟超

实验日期: 2024 年 5 月 30 日

试验台号: _____

报告总分数: _____

教师评语:

助教签字: _____

教师签字: _____

日 期: _____

一、实验目的

通过实验学习使用 F28335 DSP 的通用输入/输出管脚直接控制外围设备的方法，了解发光二极管的控制编程方法。

二、实验过程

1. 阅读代码
2. 连接线路，编译并下载程序，进行硬件仿真
3. 观察实验结果

三、实验结果

1. 阅读代码

```
1. void InitGPIO53() {
2.     EALLOW;
3.     GpioCtrlRegs.GPBPUD.bit.GPIO53 = 0; // 使能 GPIO53 上拉状态
4.     GpioCtrlRegs.GPBQSEL2.bit.GPIO53 = 0; // 设置与 SYSCLKOUT 同步刷新
5.     GpioCtrlRegs.GPBMUX2.bit.GPIO53 = 0; // 使能 GPIO53 的通用输入输出功能
6.     GpioCtrlRegs.GPBDIR.bit.GPIO53 = 1; // 设置 GPIO53 的方向为输出
7.     EDIS;
8. }

1. for(;;) {
2.     GpioDataRegs.GPBDAT.bit.GPIO53=0; // GPIO50 输出低电平，点亮指示灯
3.     Delaysms(100); //延时 500 毫秒
4.     GpioDataRegs.GPBDAT.bit.GPIO53=1; // GPIO50 输出高电平，熄灭指示灯，完成一次闪
   烁
5.     Delaysms(100); //延时 500 毫秒
6. }
```

2. 连接线路，编译并下载程序，进行硬件仿真，观察实验结果

观察到南侧 D3 LED 进行以 1s 为周期的闪烁。观察现象在附件中。

四、问题与思考

- 1、例程中直接对 GPIO 数据寄存器赋值，请通过置位寄存器和清零寄存器实现同样功能。（1 分）
代码修改如下，使用 GPBCLEAR 清零寄存器和 GPBSET 置位寄存器。

```
1. for(;;) {
2.     GpioDataRegs.GPBCLEAR.bit.GPIO53=1; // GPIO50 输出低电平，点亮指示灯
3.     Delaysms(100); //延时 500 毫秒
4.     GpioDataRegs.GPBSET.bit.GPIO53=1; // GPIO50 输出高电平，熄灭指示灯，完成一次闪
   烁
5.     Delaysms(100); //延时 500 毫秒
6. }
```

实验现象见附件。

实验报告

课程名称: _____ DSP 的原理与应用

实验名称: _____ 实验 5.3 液晶显示器控制显示

专业-班级: _____ 电气 1 班 学号: _____ 220330124 姓名: _____ 舒晟超

实验日期: _____ 2024 年 5 月 30 日

试验台号: _____

报告总分数: _____

教师评语:

助教签字: _____

教师签字: _____

日 期: _____

一、实验目的

通过实验学习使用 F28335 DSP 的扩展 I/O 端口控制外围设备的方法，了解液晶显示器的显示控制原理及编程方法。

二、实验过程

1. 阅读代码
2. 连接线路，编译并下载程序，进行硬件仿真
3. 观察实验结果

三、实验结果

1. 阅读代码

```
1. void main() { int bSuccess,h,m,s; char buffer[10]={ "00:00:00" };
2. InitICETEK_F28335Ae();
3. bSuccess=ICETEKCTR_InitCTR(ICETEKCTRModeTeachingResearch); //初始化 ICETEK-CTR: 教
   研模式
4. while ( bSuccess ); // 如果初始化 ICETEK-CTR 错误, 停止运行, 可观察 bSuccess 取值查找初
   始化失败原因
5. ICETEKCTR_LCDPutString("ICETEK-F28335-AF",0,LCDLINE0); // 在液晶显示屏的第一行显示
   字符串"ICETEK-F28335-AF"
6. ICETEKCTR_LCDPutString("液晶显示",2,LCDLINE1); // 在液晶显示屏的第二行显示字符串"液
   晶显示"
7. h=m=s=0; // 将小时、分钟和秒钟的值都设置为0
8. for(;;){
9.     s++; if ( s>59 ) { s=0; m++; if ( m>59 ) { m=0; h++; h%=24; } }
10.    buffer[0]=h/10+'0'; buffer[1]=h%10+'0'; // 将小时转换为字符并存入 buffer 数组
11.    buffer[3]=m/10+'0'; buffer[4]=m%10+'0'; // 将分钟转换为字符并存入 buffer 数组
12.    buffer[6]=s/10+'0'; buffer[7]=s%10+'0'; // 将秒数转换为字符并存入 buffer 数组
13.    ICETEKCTR_LCDPutString(buffer,2,LCDLINE3); // 在液晶显示屏的第三行显示当前开机时间
14.    ICETEKCTR_Delays(1000);
15. }
16. }
```

2. 连接线路，编译并下载程序，进行硬件仿真，观察实验结果

实验现象在附件，观察到 LCD 第一行显示 ICETEK-F28335-AF，左对齐；第二行显示字符串"液晶显示"，右移两个坐标；第三行显示当前开机时间。

四、问题与思考

- 1、通过 LCD 显示其他的文字效果。（1分）

代码修改如下：使得屏幕显示 HelloWorld 和 LCD Show Runtime 以及运行时间。

```
1. void main() { int bSuccess,h,m,s; char buffer[10]={ "00:00:00" };
2. InitICETEK_F28335Ae();
3. bSuccess=ICETEKCTR_InitCTR(ICETEKCTRModeTeachingResearch); //初始化 ICETEK-CTR: 教
   研模式
4. while ( bSuccess ); // 如果初始化 ICETEK-CTR 错误, 停止运行, 可观察 bSuccess 取值查找初
   始化失败原因
5. ICETEKCTR_LCDPutString("HelloWorld",2,LCDLINE0);
6. ICETEKCTR_LCDPutString("LCD Show Runtime",0,LCDLINE1);
```

```
7.  h=m=s=0;
8.  for(;;){
9.      s++; if ( s>59 ) { s=0; m++; if ( m>59 ) { m=0; h++; h%=24; } }
10.   buffer[0]=h/10+'0'; buffer[1]=h%10+'0';
11.   buffer[3]=m/10+'0'; buffer[4]=m%10+'0';
12.   buffer[6]=s/10+'0'; buffer[7]=s%10+'0';
13.   ICETEKCTR_LCDPutString(buffer,2,LCDLINE3);
14.   ICETEKCTR_Delays(1000);
15. }
16. }
```

实验现象见附件。

实验报告

课程名称: _____ DSP 的原理与应用

实验名称: _____ 实验 5.4: 键盘输入

专业-班级: _____ 电气 1 班 学号: _____ 220330124 姓名: _____ 舒晟超

实验日期: _____ 2024 年 5 月 30 日

试验台号: _____

报告总分数: _____

教师评语:

助教签字: _____

教师签字: _____

日 期: _____

一、实验目的

通过实验学习使用 F28335 DSP 的扩展端口接收外围设备信息的方法，了解键盘的使用原理及编程方法。

二、实验过程

1. 阅读代码
2. 连接线路，编译并下载程序，进行硬件仿真
3. 观察实验结果

三、实验结果

1. 阅读代码

```
1. void main() {
2.     int bSuccess; unsigned int uKeyCode,uDelayCount;
3.     InitICETEK28335Ae();
4.     bSuccess=ICETEKCTR_InitCTR(ICETEKCTRModeTeachingResearch); //初始化 ICETEK-CTR: 教
        研模式
5.     while ( bSuccess ); // 如果初始化 ICETEK-CTR 错误, 停止运行, 可观察 bSuccess 取值查找初
        始化失败原因
6.     for ( uDelayCount=2;uDelayCount<10;uDelayCount++ ) uLed[uDelayCount]=uLed[uDelay
        Count-1]; // 通过按位或的方式使得高位 LED 亮起时, 低位 LED 同时亮起, 以显示按键数值
7.     uDelayCount=0; ICETEKCTR_SetTrafficLEDStatus(uLed[0],ICETEKCTRLEDSet);
8.     for(;;) {
9.         // 通过 ICETEKCTR_GetKey() 函数获取按键码, 如果按键被按下 (uKeyCode 非零), 则设置对
            应的 LED 灯状态, 并设置 uDeLayCount 为1
10.        uKeyCode=ICETEKCTR_GetKey();
11.        if ( uKeyCode ) {
12.            ICETEKCTR_SetTrafficLEDStatus(uLed[uKeyCode],ICETEKCTRLEDSet);
13.            uDelayCount=1;
14.        }
15.        if ( uDelayCount ) {
16.            uDelayCount++;
17.            if ( uDelayCount>10 ) {
18.                ICETEKCTR_SetTrafficLEDStatus(uLed[0],ICETEKCTRLEDSet);
19.                uDelayCount=0; // 延时 1s 后将 LED 熄灭
20.            }
21.        }
22.        ICETEKCTR_Delayms(100);
23.    }
24. }
```

2. 连接线路，编译并下载程序，进行硬件仿真，观察实验结果

实验现象见附件，观察到按键按下时，和按键数值对应数量的 LED 灯亮起，1s 后熄灭。

四、问题与思考

- 1、试将键盘的按键通过 LCD 显示出来。（1 分）

代码修改如下：第一行显示 Key，第二行显示按键数值，如果没有按下则显示 NULL。

```
1. void InitICETEK28335Ae(); // 初始化: DSP 主频、GPIO、中断向量、中断使能
2. void main() {
```

```

3.  int bSuccess;
4.  unsigned int uKeyCode,uDelayCount;
5.  char buffer[1];
6.  InitICETEK28335Ae();
7.  bSuccess=ICETEKCTR_InitCTR(ICETEKCTRModeTeachingResearch); // 初始化 ICETEK-CTR: 教
   研模式
8.  while ( bSuccess ); // 如果初始化 ICETEK-CTR 错误, 停止运行, 可观察 bSuccess 取值查找初
   始化失败原因
9.  for ( uDelayCount=2;uDelayCount<10;uDelayCount++ )
10.   uLed[uDelayCount]=uLed[uDelayCount-1];
11.  uDelayCount=0;
12.  ICETEKCTR_SetTrafficLEDStatus(uLed[0],ICETEKCTRLEDSet);
13.  for(;;) {
14.   ICETEKCTR_LCDPutString("Key",2,LCDLINE0);
15.   uKeyCode=ICETEKCTR_GetKey();
16.   if ( uKeyCode ) {
17.    buffer[0] = uKeyCode + '0';
18.    ICETEKCTR_LCDCLS();
19.    ICETEKCTR_LCDPutString("Key",2,LCDLINE0);
20.    ICETEKCTR_LCDPutString(buffer,0,LCDLINE1);
21.    ICETEKCTR_SetTrafficLEDStatus(uLed[uKeyCode],ICETEKCTRLEDSet);
22.    uDelayCount=1;
23.   }
24.   if ( uDelayCount ) {
25.    uDelayCount++;
26.    if ( uDelayCount>10 ) {
27.     ICETEKCTR_LCDCLS();
28.     ICETEKCTR_LCDPutString("Key",2,LCDLINE0);
29.     ICETEKCTR_LCDPutString("Null",0,LCDLINE1);
30.     ICETEKCTR_SetTrafficLEDStatus(uLed[0],ICETEKCTRLEDSet);
31.     uDelayCount=0;
32.    }
33.   }
34.   ICETEKCTR_Delaysms(100);
35.  }
36. }

```

实验现象见附件。