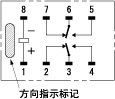
继电器

1继电器的工作原理

实验使用的是G6K-2F继电器。它的端子配置/内部连接图如下图：



1. 内部链接图

当1、8电路不通时，2、3为常闭，3、4为常开；6、7为常闭，6、5为常开。

当1、8电路导通时，2、3为常闭断开，3、4为常开闭合；6、7为常闭断开，6、5为常开闭合。

2继电器的原理图

实验是利用ZigBee模块，控制继电器的开关转换，继而利用继电器来工作的。下面是继电器的原理图，如图：



1. 原理图

ZigBee模块通过P0\_7输入高电平，使得三极管发射极和集电极处导通，这样，R两端1、8导通，继电器内部连接开关状态改变，然后继电器开始工作。

3代码实现

代码及注释如下：

#include <ioCC2530.h> //加载头文件

#define uint unsigned int //把uint定义为int

#define uchar unsigned char //把uchar定义为char

/\*定义控制灯的端口\*/

#define RLED P0\_5 //定义LED1为P0\_5口控制

#define YLED P2\_0 //定义LED2为P2\_0口控制

#define relay P0\_7 //定义延时为P0\_7口控制

#define K1 P1\_1 //定义物理K1按键为P1\_1控制,初始化为低电平1，当物理按键K1按下，由低电平1转为高电平0，同时K2由高电平0转为低电平1

#define K2 P1\_0 //定义物理K2按键为P1\_0控制，初始化为高电平0，当物理按键K2按下，有高电平0转为低电平1，同时K1由低电平1转为高电平0

#define ON 0 //定义ON状态为0

#define OFF 1 //定义OFF状态为1

//函数声明

void Delay(uint); //延时函数

void Initial(void); //初始化P口

void InitKey(void); //初始化按键键值

uchar KeyScan(void); //声明键值扫描函数

uchar Keyvalue = 0 ; //定义全局变量Keyvalue，初始化为0

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

//延时

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

void Delay(uint n) //延时函数

{

uint tt; //定义tt

for(tt = 0;tt < n;tt++); //采用多次的for循环起到延时的作用，延时时间长短有循环次数和机器的周期有关。

for(tt = 0;tt < n;tt++);

for(tt = 0;tt < n;tt++);

for(tt = 0;tt < n;tt++);

for(tt = 0;tt < n;tt++);

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

//按键初始化

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

void InitKey(void)

{

P1SEL &= ~0X03;//P1SEL只是一个变量名称，是一个4位的端口，~0X03就是将0011按位取反，为1100b，将四位中的高两位置为1，低两位清零.

P1DIR &= ~0X03; //按键在P12 P13

P1INP |= 0x03; //上、下拉

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

//初始化程序

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

void Initial(void)

{

P0DIR |= 0x20; //P0\_5定义为输出

P2DIR |= 0x01; //P2\_0定义为输出

RLED = OFF; //红色LED

YLED = OFF; //黄色LED

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

//读键值

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

uchar KeyScan(void)

{

if(K1 == 0) //当物理K1按键按下时，K1由低电平转为高电平，K1=0,K2=1，

{

Delay(100); //去抖操作，防止因为抖动造成电位改变

if(K1 == 0) //循环

{

while(!K1); //K1为1的时候跳出循环

return(1);

}

};

if(K2 == 0) //当物理K2按键按下时，K2由低电平转为高电平，K1=1，K2=0

{

Delay(100); //去抖操作，防止因为抖动造成电位改变

if(K2 == 0) //循环

{

while(!K2);//K2为1的时候跳出循环

return(2);

}

};

return(0);

}

void write\_relay\_1(void)//继电器控制管脚 输出1

{

relay = 1; //继电器控制管脚为1

}

void write\_relay\_0(void)//继电器控制管脚 输出0

{

relay = 0; //继电器控制管脚为0

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

//主函数

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

void main(void)

{

Initial(); //调用初始化函数

InitKey(); //初始化按键K1，K2

Delay(4000); //延时操作

Delay(4000); //延时操作

P0DIR |= 0x80; //P0赋值

YLED = ON; //初始化状态为黄灯亮

relay=0; //初始化状态继电器输出0，断开

while(1) //继电器通电状态下不断重复以下操作

{

Keyvalue = KeyScan(); //扫键

if(Keyvalue>0) //扫描键存在按键输入时，

{

if(Keyvalue == 1) //k1按键继电器开红灯亮

{

RLED = ON; //红亮灯

YLED = OFF; //黄灯灭

write\_relay\_1();//控制继电器输出1

}

if(Keyvalue == 2)//K2按键 继电器开黄灯亮

{

YLED = ON; //黄亮灯

RLED = OFF; //红灯灭

write\_relay\_0();//控制继电器输出0s

}

}

}

}