1. 控制小灯，按下亮，弹起灭

#define LED D5

#define INPUT\_KEY D2

int val = 0;

void setup() {

// put your setup code here, to run once:

pinMode(LED, OUTPUT);

//pinMode(INPUT\_KEY, INPUT);

pinMode(INPUT\_KEY, INPUT\_PULLUP);

}

void loop() {

// put your main code here, to run repeatedly:

if (digitalRead(INPUT\_KEY) == LOW)

{

digitalWrite(LED, LOW);

}

else

{

digitalWrite(LED, HIGH);

}

}

1. 控制小灯，按一下灭，按一下亮

#define LED D5

#define INPUT\_KEY D2

int val = 0;

void setup() {

// put your setup code here, to run once:

pinMode(LED, OUTPUT);

pinMode(INPUT\_KEY, INPUT\_PULLUP);

digitalWrite(LED, LOW);

}

void loop() {

// put your main code here, to run repeatedly:

ScanKey();

if (val == 1)

{

digitalWrite(LED, !digitalRead(LED));

}

}

//消除抖动,确认按键的处理

void ScanKey()

{

val = 0;

if (digitalRead(INPUT\_KEY) == LOW)

{

delay(20);

if (digitalRead(INPUT\_KEY) == LOW)

{

val = 1;

while (digitalRead(INPUT\_KEY) == LOW)

{

delay(20);

}

}

}

}

1. 制作抢答器

//LED灯的引脚

int greenled = D4;

int redled = D5;

int yellowled = D6;

//开关的引脚，通过开关控制LED灯

int greenpin = D1;

int redpin = D2;

int yellowpin = D3;

//读取开关的信，来控制LED灯

int red;

int yellow;

int green;

//初始化

void setup()

{

//设置引脚模式为输出

pinMode(redled, OUTPUT);

pinMode(yellowled, OUTPUT);

pinMode(greenled, OUTPUT);

//设置开关的引脚为输入

pinMode(redpin, INPUT);

pinMode(yellowpin, INPUT);

pinMode(greenpin, INPUT);

digitalWrite(greenled, LOW);

digitalWrite(redled, LOW);

digitalWrite(yellowled, LOW);

}

//循环

void loop()

{

//设置为低电平

digitalWrite(greenled, LOW);

digitalWrite(redled, LOW);

digitalWrite(yellowled, LOW);

//读取控制红灯的开关的信号

red = digitalRead(redpin);

//根据控制开关的信，给LED引脚写入电平

if (red == LOW)

{

//给LED引脚写入低电平

digitalWrite(redled, LOW);

}

else

{

//给LED引脚写入高电平

digitalWrite(redled, HIGH);

}

//读取控制黄灯的开关的信号

yellow = digitalRead(yellowpin);

//根据控制开关的信，给LED引脚写入电平

if (yellow == LOW)

{

//给LED引脚写入低电平

digitalWrite(yellowled, LOW);

}

else

{

//给LED引脚写入高电平

digitalWrite(yellowled, HIGH);

}

//读取控制绿灯的开关的信号

green = digitalRead(greenpin);

//根据控制开关的信，给LED引脚写入电平

if (green == LOW)

{

//给LED引脚写入低电平

digitalWrite(greenled, LOW);

}

else

{

//给LED引脚写入高电平

digitalWrite(greenled, HIGH);

}

}

1. 控制发声模块发声

int tonepin=D3;//设置控制蜂鸣器的数字D3脚

void setup()

{

pinMode(tonepin,OUTPUT);//设置数字IO脚模式，OUTPUT为输出

}

void loop()

{

unsigned char i,j;

while(1)

{

for(i=0;i<80;i++)//输出一个频率的声音

{

digitalWrite(tonepin,HIGH);//发声音

delay(1);//延时1ms

digitalWrite(tonepin,LOW);//不发声音

delay(1);//延时ms

}

for(i=0;i<100;i++)//输出另一个频率的声音,这里的100与前面的80一样，用来控制频率，可以自己调节

{

digitalWrite(tonepin,HIGH);

delay(2);

digitalWrite(tonepin,LOW);

delay(2);

}

}

}

1. 播放乐谱（欢乐颂）

#define NTD0 -1

#define NTD1 294

#define NTD2 330

#define NTD3 350

#define NTD4 393

#define NTD5 441

#define NTD6 495

#define NTD7 556

#define NTDL1 147

#define NTDL2 165

#define NTDL3 175

#define NTDL4 196

#define NTDL5 221

#define NTDL6 248

#define NTDL7 278

#define NTDH1 589

#define NTDH2 661

#define NTDH3 700

#define NTDH4 786

#define NTDH5 882

#define NTDH6 990

#define NTDH7 112

//列出全部D调的频率

#define WHOLE 1

#define HALF 0.5

#define QUARTER 0.25

#define EIGHTH 0.25

#define SIXTEENTH 0.625

//列出所有节拍

int tune[]= //根据简谱列出各频率

{

NTD3,NTD3,NTD4,NTD5,

NTD5,NTD4,NTD3,NTD2,

NTD1,NTD1,NTD2,NTD3,

NTD3,NTD2,NTD2,

NTD3,NTD3,NTD4,NTD5,

NTD5,NTD4,NTD3,NTD2,

NTD1,NTD1,NTD2,NTD3,

NTD2,NTD1,NTD1,

NTD2,NTD2,NTD3,NTD1,

NTD2,NTD3,NTD4,NTD3,NTD1,

NTD2,NTD3,NTD4,NTD3,NTD2,

NTD1,NTD2,NTDL5,NTD0,

NTD3,NTD3,NTD4,NTD5,

NTD5,NTD4,NTD3,NTD4,NTD2,

NTD1,NTD1,NTD2,NTD3,

NTD2,NTD1,NTD1

};

float durt[]= //根据简谱列出各节拍

{

1,1,1,1,

1,1,1,1,

1,1,1,1,

1+0.5,0.5,1+1,

1,1,1,1,

1,1,1,1,

1,1,1,1,

1+0.5,0.5,1+1,

1,1,1,1,

1,0.5,0.5,1,1,

1,0.5,0.5,1,1,

1,1,1,1,

1,1,1,1,

1,1,1,0.5,0.5,

1,1,1,1,

1+0.5,0.5,1+1,

};

int length;

int tonepin=D3; //得用D3号接口

void setup()

{

pinMode(tonepin,OUTPUT);

length=sizeof(tune)/sizeof(tune[0]); //计算长度

}

void loop()

{

for(int x=0;x<length;x++)

{

tone(tonepin,tune[x]);

delay(500\*durt[x]); //这里用来根据节拍调节延时，500这个指数可以自己调整，在该音乐中，我发现用500比较合适。

noTone(tonepin);

}

delay(2000);

}