一、74HC138 3-8译码器

//全局变量

int segCount = 3;

int segPins[] = { D2, D3, D4 }; // A0,A1,A2

//初始化

void setup()

{

//循环设置，把对应的端口都设置成输出

for (int thisSeg = 0; thisSeg < segCount; thisSeg++)

{

//设置引脚为输出模式

pinMode(segPins[thisSeg], OUTPUT);

}

}

//数据处理，把需要处理的 byte 数据写到对应的引脚端口。

void deal(unsigned char value)

{

for (int i = 0; i<3; i++)

{

digitalWrite(segPins[i], bitRead(value, i));//使用了 bitRead函数，非常简单

}

}

//循环执行

void loop()

{

//循环输出0-7,输出端是 Y0-Y7 依次为0,任何时刻只有1个脚为低电平。

//只要A0,A1,A2输入，38译码器会转换成对应的数字,然后译码器会让

//Y0-Y7中对应下标的引脚输出低电平,从而电路导通,灯亮起

for (int i = 0; i<8; i++)

{

deal(i);//数据处理

delay(1000);//延时1秒

}

}

1. RGB三基色LED

//模拟红绿灯

#define LED\_R D7

#define LED\_G D6

#define LED\_B D5

enum{ Color\_R, Color\_RG, Color\_G }; //红,黄,绿

void setup()

{

pinMode(LED\_R, OUTPUT);

pinMode(LED\_G, OUTPUT);

pinMode(LED\_B, OUTPUT);

}

void loop()

{

unsigned int i;

for (i = 0; i < 3; i++)

{

Change\_Color(i);

delay(1000);

};

}

void Change\_Color(unsigned char data\_color) //选择颜色函数

{

int j;

switch (data\_color)

{

case Color\_R: //红色

digitalWrite(LED\_R, HIGH);

digitalWrite(LED\_G, LOW);

digitalWrite(LED\_B, LOW);

delay(3000);

break;

case Color\_G: //绿色

digitalWrite(LED\_R, LOW);

digitalWrite(LED\_G, HIGH);

digitalWrite(LED\_B, LOW);

delay(8000);

break;

case Color\_RG:

for (j = 0; j < 5; j++){ //黄色

digitalWrite(LED\_R, HIGH);

digitalWrite(LED\_G, HIGH);

digitalWrite(LED\_B, LOW);

delay(1000);

digitalWrite(LED\_R, LOW);

digitalWrite(LED\_G, LOW);

digitalWrite(LED\_B, LOW);

delay(1000);

}

break;

default:

break;

};

}

1. 舵机使用

180°转舵

**接线：红线接VCC正极，橙色线接D6引脚，棕色或黑色的线接GND负极**

#include <Servo.h>

Servo myservo; //定义舵机对象，最多可以控制8路舵机

int pos = 0; //用于存储舵机位置的变量

//初始化

void setup()

{

myservo.attach(D6); //舵机控制信号引脚,连接舵机

Serial.begin(115200);

}

//主循环

void loop()

{

for(pos = 0; pos < 180; pos += 1) //从0度-180度

{ //步进角度1度

myservo.write(pos); //输入对应的角度值，舵机会转到此位置

delay(15); //15ms后进入下一个位置

Serial.println(pos);

}

for(pos = 180; pos>=1; pos-=1) //从180度-0度

{

myservo.write(pos); //输入对应的角度值，舵机会转到此位置

delay(15); //15ms后进入下一个位置

Serial.println(pos);

}

}

1. DS1302时钟

#include <Servo.h>

//创建一个舵机控制对象

Servo myservo;

int potpin = A0;//该变量用于存储用电位器的模拟值

int val;//该变量用于存储舵机角度位置

void setup()

{

myservo.attach(D6);//该舵机由ESP8266MCU,D6引脚控制

Serial.begin(115200);

}

void loop()

{

val = analogRead(potpin); //读取电位器控制的模拟值 (范围在0-1023)

val = map(val, 0, 1023, 0, 179); //scale it to use it with the servo (value between 0 and 180)

myservo.write(val); //指定舵机转向的角度

Serial.println(val);

delay(15); //等待15ms让舵机到达指定位置

}