

ESP32流水灯和按键中断

2021年6月3日 0:06

ESP32的编程跟Arduino相差不是很大，所以写起程序来很容易，就是基本的C语言函数调用的过程，可以去[Arduino 常用函数参考](#)了解一下Arduino基本函数，然后调用就可以了。

`pinMode(pin, mode)`

函数功能：设置GPIO端口的状态

参数pin指定需要设置的GPIO端口

参数mode指定GPIO端口的模式，有以下三个参数可选

INPUT：输入模式

OUTPUT：输出模式

INPUT_PULLUP：输入上拉模式

`digitalWrite(pin, value)`

函数功能：可以使用这个函数让GPIO口输出高电平或者低电平

参数pin指定需要设置电平的GPIO端口

参数value指定端口输出的电平状态，有以下两个参数可选

HIGH：高电平

LOW：低电平

默认的工程下面有两个建立工程就有的函数 `void setup()` 和 `void loop()`

`void setup()`为程序初始化函数，在上电启动的时候会执行一次，执行完一次之后程序将会进入到`loop()`函数不再执行`setup()`，直到程序再次重新启动。

`void loop()`为循环执行的函数，在程序执行完`setup()`初始化函数之后执行，`loop()`中的程序代码执行完一次之后将会从`loop()`中最开始的代码重新执行，也就是说我们这个函数中的代码会一直循环的执行下去

代码在下面：

```
int led = 13 ; //对应ESP32板子上的G13引脚
```

```
void setup() {
```

```
    // 把你的初始化代码放在这里，只运行一次:
```

```
    pinMode(led,OUTPUT); //将GPIO13配置为输出模式
```

```
}
```

```
void loop() {
```

```
    // 把你的功能性（主）代码放在这里，重复运行:
```

```
    digitalWrite(led,HIGH); //GPIO2输出高电平 灯亮
```

```
    delay(500); //保持GPIO2高电平500毫秒，即灯亮500ms
```

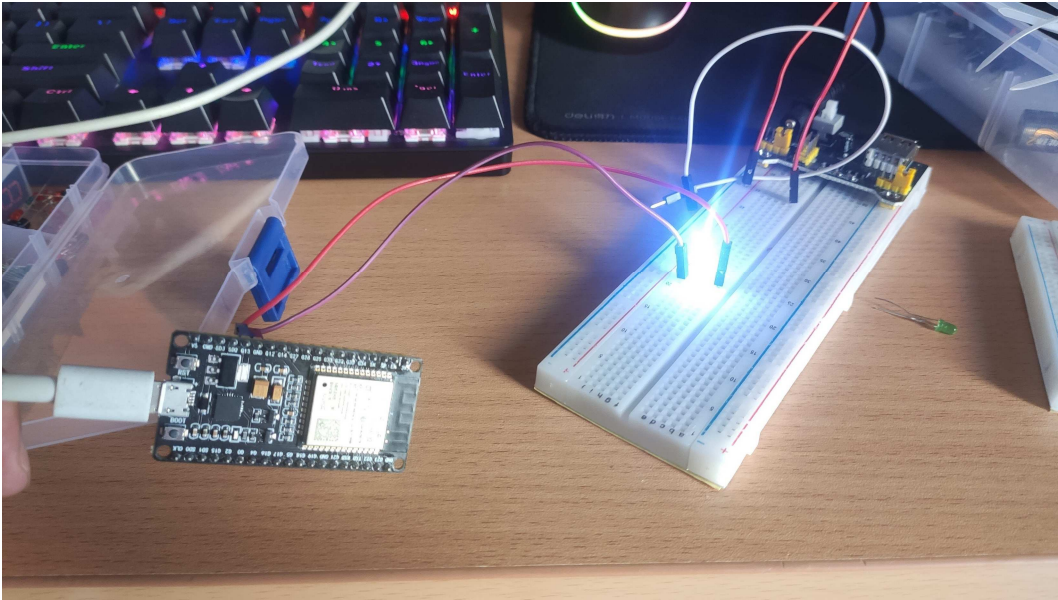
```
    digitalWrite(led,LOW); //GPIO2输出低电平 灯灭
```

```
    delay(500); //保持GPIO低电平500毫秒，即灯灭500ms
```

```
}
```

编译下载（注意事项），在第一讲讲过了，我也不再BB了！

效果图



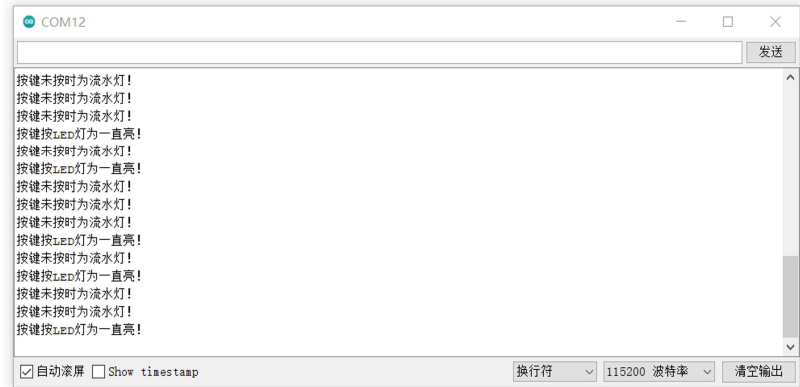
接下来是简单的软件按键中断实验，我连接的硬件电路是单按键一边连在单片机3.3V，一边连在G12引脚上，这样当按键按下的时候3.3V和G12引脚会形成回路，可以说是G12引脚上读到的是高电平；没按下的时候是短路状态，读到的当然是低电平；
代码在下面：

```
int led = 13; //对应ESP32板子上的G13引脚
int key = 12; //对应ESP32板子上的G12引脚
void setup() {
    // 把你的初始化代码放在这里，只运行一次：
    Serial.begin(115200); //初始化串口
    pinMode(led,OUTPUT); //将GPIO13配置为输出模式
    pinMode(key,INPUT);
}

void loop() {
    // 把你的功能性（主）代码放在这里，重复运行：
    if(digitalRead(key) == HIGH) //当开关按时，12号引脚是高点平
    {
        delay(10); //按键消抖
        digitalWrite(led,HIGH); //LED一直亮状态
        Serial.println("按键按LED灯为一直亮！"); //并把这句话打印在串口
        while(digitalRead(key) == HIGH); //一直到按键松开为止，12号引脚低电平
    }
    if(digitalRead(key) == LOW) //当开关松开时时，12号引脚是低点平
    {
        Serial.println("按键未按时为流水灯！"); //并把这句话打印在串口，再执行下面流水灯程序
        digitalWrite(led,HIGH); //GPIO2输出高电平 灯亮
        delay(500); //保持GPIO2高电平500毫秒，即灯亮500ms
        digitalWrite(led,LOW); //GPIO2输出低电平 灯灭
        delay(500); //保持GPIO低电平500毫秒，即灯灭500ms
    }
}
```

编译完以后，下载到开发板，并且打开右上角的串口，可以验证实验结果！
效果在下面：

```
__LED
1 int led = 13; //对应ESP32板子上的G13引脚
2 int key = 12; //对应ESP32板子上的G12引脚
3 void setup() {
4   // 把你的初始化代码放在这里，只运行一次：
5   Serial.begin(115200);
6   pinMode(led,OUTPUT); //将GPIO13配置为输出模式
7   pinMode(key,INPUT);
8 }
9
10 void loop() {
11   // 把你的功能性（主）代码放在这里，重复运行：
12   if(digitalRead(key) == HIGH)
13   {
14     delay(10);
15     digitalWrite(led,HIGH);
16     Serial.println("按键按LED灯为一直亮！");
17     while(digitalRead(key) == HIGH);
18   }
19   if(digitalRead(key) == LOW)
20   {
21     Serial.println("按键未按时为流水灯！");
22     digitalWrite(led,HIGH); //GPIO2输出高电平 灯亮
23     delay(500); //保持GPIO2高电平500毫秒，即灯亮500ms
24     digitalWrite(led,LOW); //GPIO2输出低电平 灯灭
25     delay(500); //保持GPIO低电平500毫秒，即灯灭500ms
26   }
27 }
```



https://git.oschina.net/dfrobot/FireBeetle-ESP32/raw/master/package_esp32_index.json

