

物联网应用技术专业创新教材

# 物联网技术应用 项目实训

主编 韩卫宏 周仕林



SE 东南大学出版社  
SOUTHEAST UNIVERSITY PRESS

# 物联网技术应用 项目实训

主 编 韩卫宏 周仕林

副主编 李文亮 艾芹芹 赵孟真

参 编 郭丽君 张明雄

 东南大学出版社  
SOUTHEAST UNIVERSITY PRESS

· 南京 ·

## 内容提要

本教材共分为四个项目。项目一介绍了如何利用物联网技术实现对LED灯的云控制；项目二详细讲解了液晶屏的显示原理等，并配以相应的实验案例和实训项目，帮助学生深入理解和掌握如何利用物联网技术实现对液晶屏的云控制；项目三介绍了温度监控和风扇控制的原理，并利用实际的项目帮助学生体验用物联网技术实现对温度的监控和风扇的控制；项目四介绍了触摸按键和舵机的控制原理，配以智能家居中的典型应用，让学生理解并掌握如何实现云控制。

本教材可作为中职学校电子信息类相关专业的教材，也可作为相关领域工作人员的参考用书。

## 图书在版编目(CIP)数据

物联网技术应用项目实训 / 韩卫宏, 周仕林主编.  
南京: 东南大学出版社, 2024. 7. — ISBN 978-7-5766-1524-1  
I. TP393.4; TP18  
中国国家版本馆 CIP 数据核字第 2024R6P224 号

责任编辑: 子雪莲 责任校对: 李成思 封面设计: 易 帅 责任印制: 周荣虎

## 物联网技术应用项目实训

WULIANWANG JISHU YINGYONG XIANGMU SHIXUN

---

主 编: 韩卫宏 周仕林  
出版发行: 东南大学出版社  
出 版 人: 白云飞  
社 址: 南京市四牌楼2号 邮编 210096  
印 刷: 天津市蓟县宏图印务有限公司  
开 本: 787mm×1092mm 1/16  
印 张: 10  
字 数: 213 千  
版 次: 2024 年 7 月第 1 版  
印 次: 2024 年 7 月第 1 次印刷  
书 号: ISBN 978-7-5766-1524-1  
定 价: 42.00 元

---

(凡因印装质量问题, 请直接与营销中心调换, 电话: 025-83791830)

# PREFACE 前言

随着信息技术的迅猛发展，物联网(Internet of Things, IoT)作为新一代信息技术的核心组成部分，已经渗透到了我们生活的方方面面。从智能家居到智慧城市，从工业自动化到农业精准管理，物联网技术的应用正逐步改变着世界。

为了满足中等职业学校相关专业学生对物联网技术的学习需求，我们编写了本教材。本教材旨在帮助学生掌握物联网技术的基本概念、原理和应用方法，并通过实际项目的实训，提高学生的动手能力和解决问题的能力。本教材注重培养学生的创新思维和实践能力，以适应未来社会对物联网技术人才的需求。

在编写过程中，我们充分考虑了中等职业学校学生的特点和需求，力求使教材内容深入浅出、通俗易懂。同时，我们也注重与时俱进，紧跟物联网技术的最新发展动态，确保教材内容的先进性和实用性。

本教材在内容编排上，采用了项目驱动的教学方式，通过一系列精心设计的实训项目，引导学生从基础理论入手，逐步深入到物联网技术的核心应用领域。每个项目都包含了详细的任务要求、任务准备、任务实施和任务评价模块，旨在帮助学生系统地掌握物联网技术的开发流程和应用方法。

在实训项目的选择上，我们力求贴近实际应用场景，结合当前物联网技术的热点和难点问题，设计了一系列具有挑战性和实用性的实训项目。这些项目既涵盖物联网技术的各个环节，又注重培养学生的团队协作和沟通能力，让学生在实践中学习和成长。

本教材共分为四个项目。项目一介绍了如何利用物联网技术实现对 LED 灯的云控制；项目二详细讲解了液晶屏的显示原理等，并配以相应的实验案例和实训项目，帮助学生深入理解和掌握如何利用物联网技术实现对液晶屏的云控制；项目三介绍了温度监控和风扇控制的原理，并利用实际的项目帮助学生体验用物联网技术实现对温度的监控和风扇的控制；项目四介绍了触摸按键和舵机的控制原理，配以智能家居中的典型应用，让学生理解并掌握如何实现云控制。





本教材由武汉市仪表电子学校韩卫宏和周仕林担任主编。韩卫宏编写了项目一，并负责全书统稿；艾芹芹编写了项目二中的任务一、任务二；郭丽君编写了项目二中的任务三；周仕林编写了项目三中的任务一、任务二、任务三，并负责全部代码的编写与调试；张明雄编写了项目三中的任务四；北京新大陆时代教育科技有限公司总工程师李文亮编写了项目四中的任务一、任务二；湖北铁道运输职业学院赵孟真编写了项目四中的任务三、任务四。

由于编者水平有限，书中难免存在疏漏之处，我们期待您的反馈和建议，以便我们不断改进和完善教材内容。

编 者



# CONTENTS 目录

<b>项目一 智能灯光控制系统</b> .....	( 1 )
任务一 点亮一个灯 .....	( 2 )
任务二 流水灯 .....	( 15 )
任务三 基于新大陆云平台控制 LED .....	( 20 )
<b>项目二 智能显示系统</b> .....	( 36 )
任务一 LCD1602 显示字符 .....	( 37 )
任务二 LCD1602 显示数字钟 .....	( 49 )
任务三 基于新大陆云平台控制 LCD1602 显示 .....	( 62 )
<b>项目三 智能温控系统</b> .....	( 72 )
任务一 实时监控室内温度 .....	( 73 )
任务二 实现风扇调速 .....	( 85 )
任务三 实现智能温控风扇 .....	( 90 )
任务四 基于云平台控制风扇智能控制与手动控制的切换 .....	( 95 )
<b>项目四 智能门禁系统</b> .....	( 110 )
任务一 键盘按键识别 .....	( 112 )
任务二 舵机控制门锁 .....	( 124 )
任务三 电子密码锁 .....	( 135 )
任务四 基于云平台的电子密码锁 .....	( 145 )

# 项目一

## 智能灯光控制系统



### 项目导入

物联网在生活中的应用越来越广泛，智能家居就是典型的应用之一。灯光的控制智能家居使用中必不可少，不少家庭会定制一些有特殊效果的灯光控制系统。本项目主要介绍如何实现智能灯光控制。



### 项目目标

#### ◎ 知识目标：

- 能够理解 LED 的控制原理。

#### ◎ 技能目标：

- 掌握 Keil 编译软件的使用方法。
- 掌握下载软件 STC-ISP 的使用方法。
- 掌握 LED 的编程方法。
- 掌握在新大陆云平台中创建设计项目的方法。
- 掌握利用新大陆云平台控制 LED 的编程方法。

#### ◎ 素质目标：

- 培养自主学习、总结反思的能力，在学习过程中培养团队协作意识。
- 在完成任务的过程中养成规范操作的职业习惯，形成良好的安全意识、环保意识、节能意识。

## 任务一 点亮一个灯

### 任务要求

在智能家居系统中，灯光的控制十分常见。本次任务是搭建灯光控制电路，并编写程序实现对一个灯的控制。

### 任务准备

#### 1. 电路准备

单片机主控板、LED 模块，如图 1-1 和图 1-2 所示。

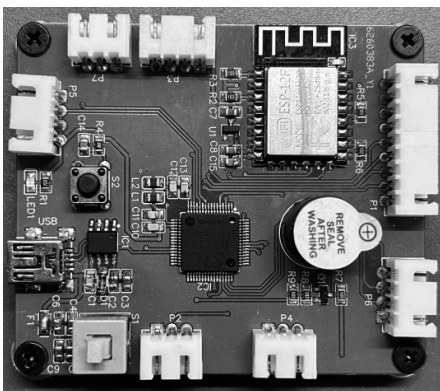


图 1-1 单片机主控板

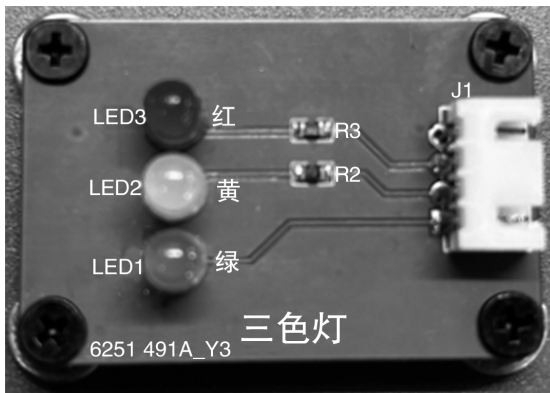


图 1-2 LED 模块

#### 2. 软件准备

Keil C51、STC-ISP。

#### 3. 任务所需相关知识

##### (1) 单片机最小系统。

STC8A8K 单片机具有高性能、低功耗、低成本等特点，被广泛应用于工业控制、家用电器、汽车电子等领域。它采用 8 位单片机的结构，拥有 256 个字节的内部存储器，每个字拥有 8 位的宽度，支持多种编程语言，可以轻松实现复杂的控制任务。

STC8A8K 有 40 个引脚，32 个外部双向输入/输出(I/O)端口，同时内含 2 个外中断口，3 个 16 位可编程定时计数器，2 个全双工串行通信口，2 个读写口线。





(2)LED 控制原理(如图 1-3 所示)。

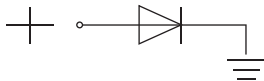


图 1-3 LED 控制原理

LED 亮的条件是 LED 阳极接高电平、阴极接低电平。由于 LED 模块中，LED 阴极已经接地，只需将 STC 单片机的 I/O 口接到 LED 的阳极就可以实现对其的控制。

(3)Keil C51 指令控制单片机 I/O 口。

在 Keil C51 编程中，利用“P03=1;”语句，可以使单片机的 P0 口第 3 位输出高电平。同理，利用“P03=0;”语句，可以使单片机的 P0 口第 3 位输出低电平。

## 任务实施

### 一、电路设计

主机模块使用的主控芯片为 STC8A8K64D4，芯片的 I/O 口都设计为特定的控制脚和输入脚。芯片的 P0.3、P0.4、P0.5 用于控制 LED 模块的红、黄、绿灯；LED 模块的红、黄、绿灯采用了共阴极的设计。主控板电路原理图和 LED 模块电路图如图 1-4 所示。

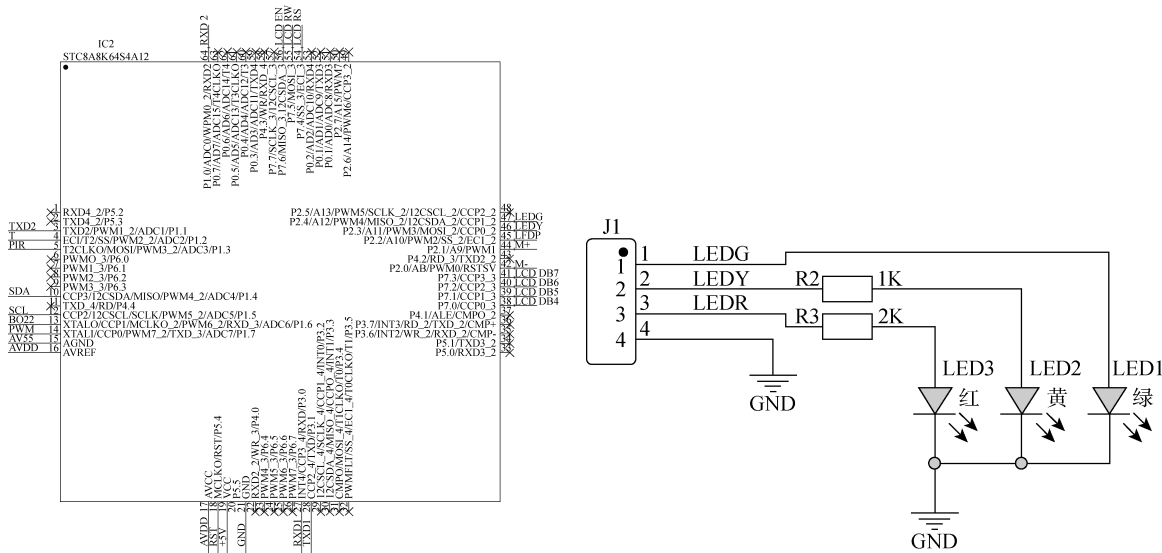


图 1-4 主控板电路原理图和 LED 模块电路图

### 二、搭建电路

本次任务需要使用两个模块：STC 主机模块、LED 模块。模块接线框图如图 1-5 所示。



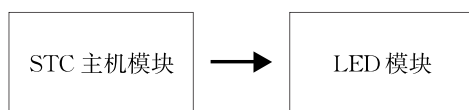


图 1-5 模块接线框图

实物只需利用排线将 I/O 口扩展模块中的 P6 端子与 LED 模块端子对接即可。具体的实物搭接图如图 1-6 所示。

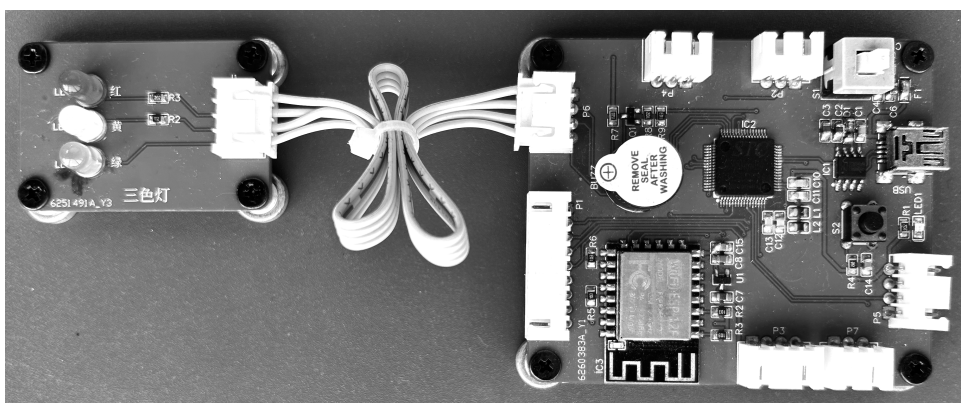


图 1-6 控制电路实物搭接图

### 三、打开工程

本书配套有设计好的工程文件，文件夹名称为“项目工程”。程序设计只需在提供的空工程中完成即可。打开软件 Keil  $\mu$ Vision5，显示窗口界面如图 1-7 所示。

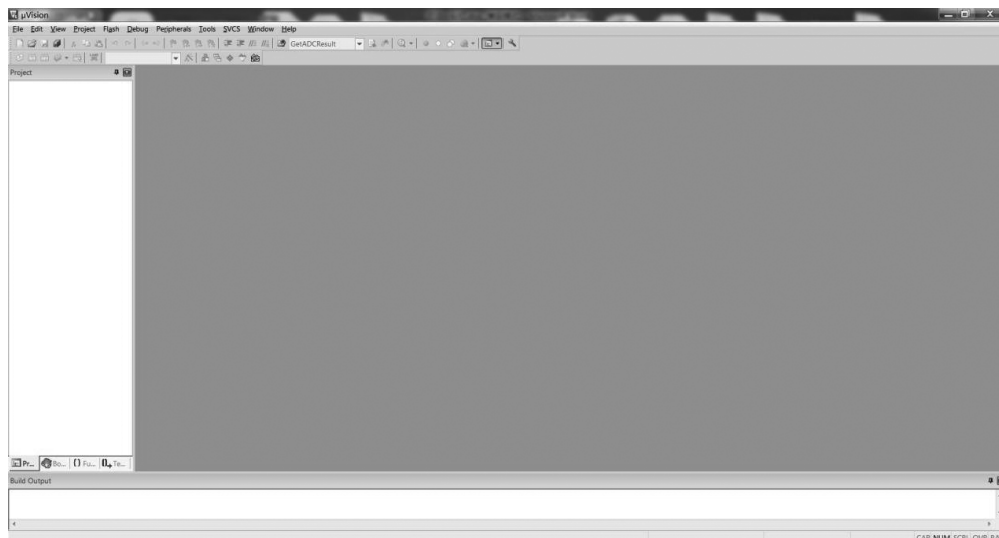


图 1-7 Keil  $\mu$ Vision5 的界面

左键单击菜单栏“Project”，出现下拉菜单，选择“Open Project”，操作如图 1-8 所示。

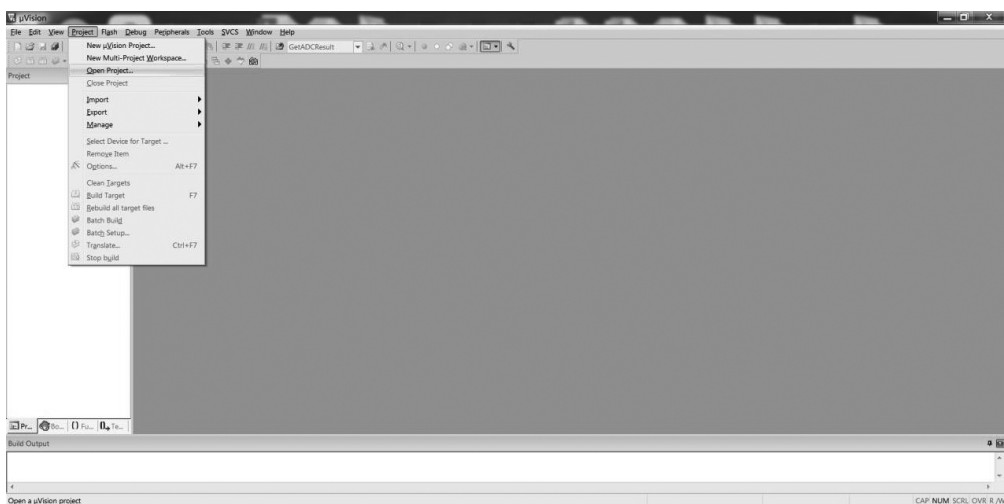


图 1-8 打开工程

找到本任务对应的工程文件，点击“打开”。具体路径为：项目工程→1. 智能灯光控制系统→1. 点亮一个灯→led.uvproj，如图 1-9 所示。



图 1-9 选择要打开的工程文件

打开工程文件后，在工程管理菜单中双击 user 文件夹下的 main.c 文件进入代码编辑界面，如图 1-10 所示。根据具体任务要求，在主函数 main() 里添加所需代码即可。

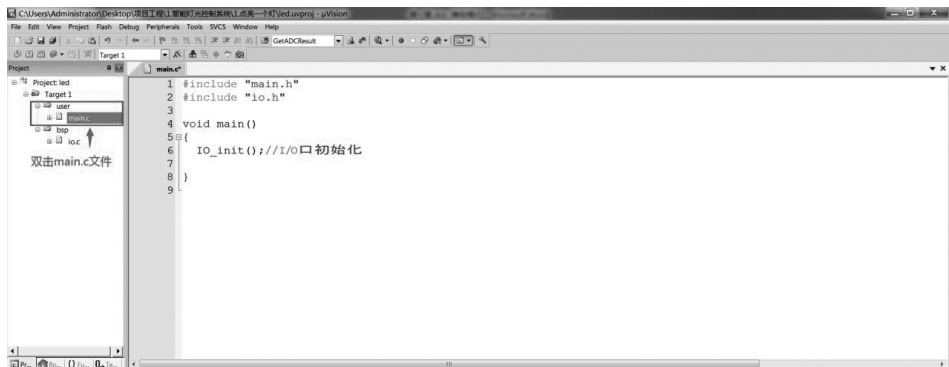


图 1-10 代码编辑界面



## 四、输入程序

根据图 1-4 电路图可知，LED 模块中的三个灯采用的是共阴极的接法。红、黄、绿灯的阳极分别接在了单片机的 P0.3、P0.4、P0.5 引脚上。我们编写代码控制对应的 I/O 口输出高、低电平即可点亮或者熄灭对应的灯。例如，红灯亮，黄灯、绿灯熄灭，只需 P0.3 输出高电平，P0.4、P0.5 输出低电平，即可实现。主函数代码如下：

```
void main ( )  
{  
    IO_init(); //I/O 初始化  
    P03=1;    //P0.3 输出高电平，点亮红灯  
    P04=0;    //P0.4 输出低电平，熄灭黄灯  
    P05=0;    //P0.5 输出低电平，熄灭绿灯  
    while(1); //循环执行空指令  
}
```

具体操作如图 1-11 所示。

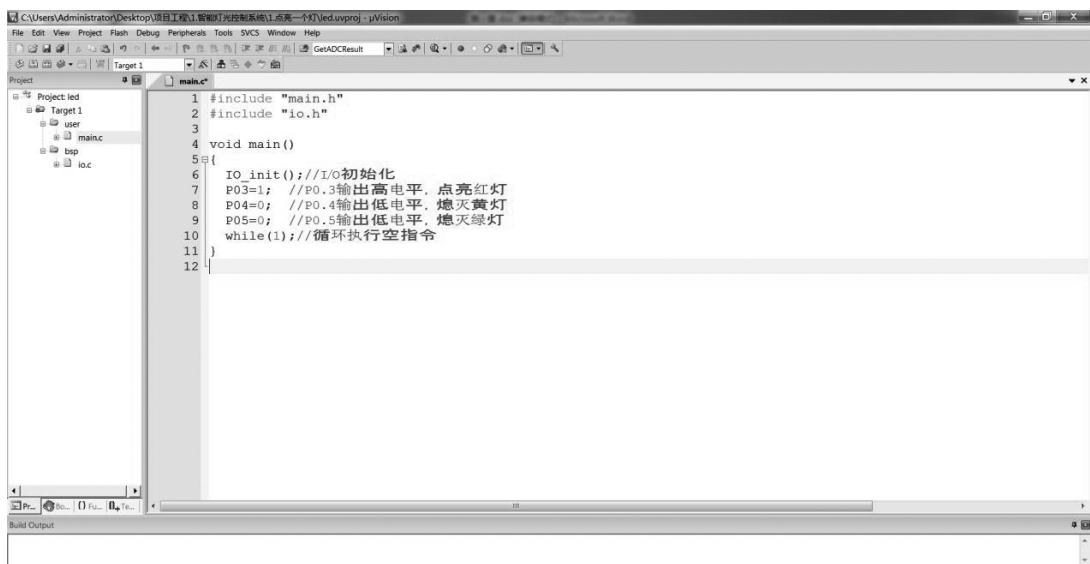


图 1-11 输入代码

计算机只识别二进制数据 0 和 1，为了将编写的代码转换成单片机可识别的数据格式，需要对程序进行编译。编译按钮位于软件界面的左上角，如图 1-12 所示。

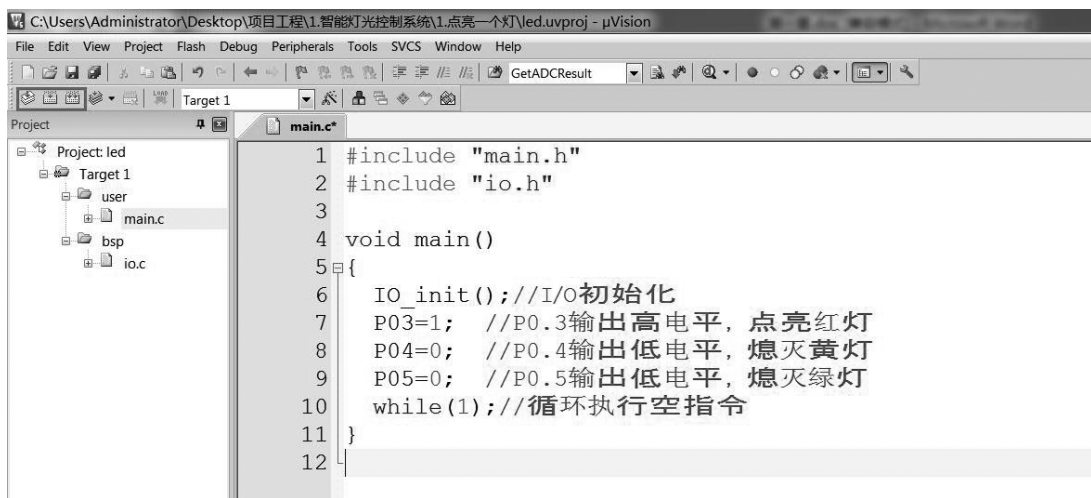


图 1-12 软件的编译按钮位置图

图 1-12 方框中的三个按钮都是编译按钮，作用各有不同。第一个：编译当前界面所在的一个 C 文件。第二个：联合编译整个工程，发生修改的文件重新编译，并生成可执行文件。第三个：联合编译整个工程，所有文件都重新编译，并生成可执行文件。一般情况下，点击第二个编译按钮进行代码编译。

点击编译按钮后，软件会自动对代码进行语法检查，并将编译结果显示在软件最下方的 Build Output(编译输出窗口)里。编译输出窗口里有很多的提示信息，其中最重要的是编译提示信息，在这条信息里可以看到代码是否有问题。示例如图 1-13 所示。

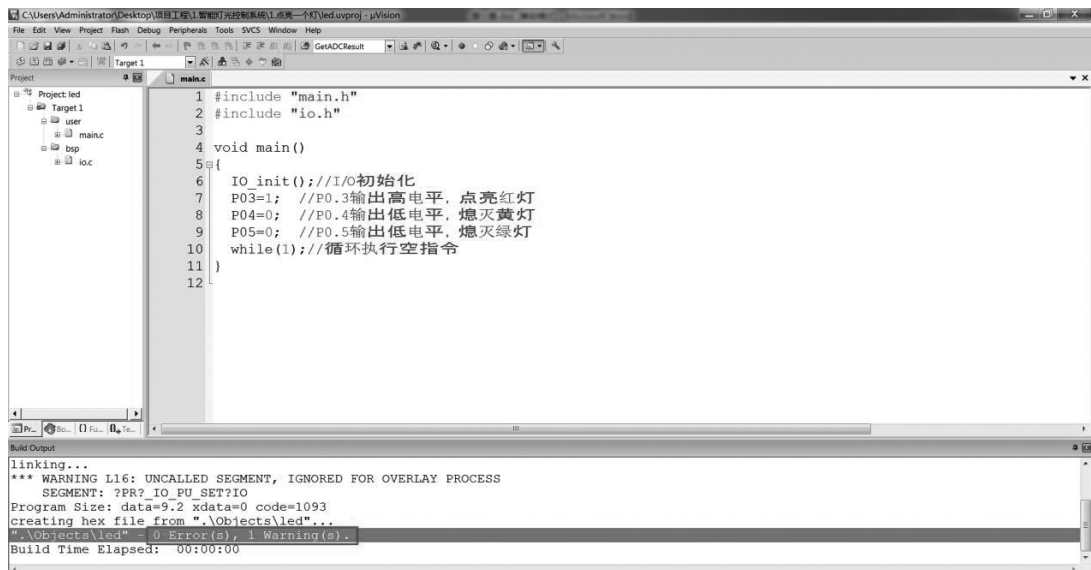


图 1-13 编译输出窗口示例

程序可能存在的问题有两种：一种是 Error，另一种是 Warning。Error 是指代码有语法错误，此时代码无法完成编译。Warning 是警告，它不影响代码的编译和运行。图 1-13



中的编译输出窗口显示“0 Error(s), 1Warning(s)”，这说明当前代码没有语法错误。有一个警告是因为工程中其他 C 文件里有许多定义过的函数未被调用，因此出现了警告，但不影响代码的编译和运行。

## 五、生成 HEX 文件

代码编译完毕后，需要生成 HEX 文件(文件的后缀是“.hex”)，再通过 STC-ISP 软件将 HEX 文件烧录到单片机里，单片机才能运行代码，实现相应功能。

第一步，左键单击配置目标输出按钮，如图 1-14 所示。



图 1-14 单击配置目标输出按钮

第二步，选择“Output”选项，如图 1-15 所示。

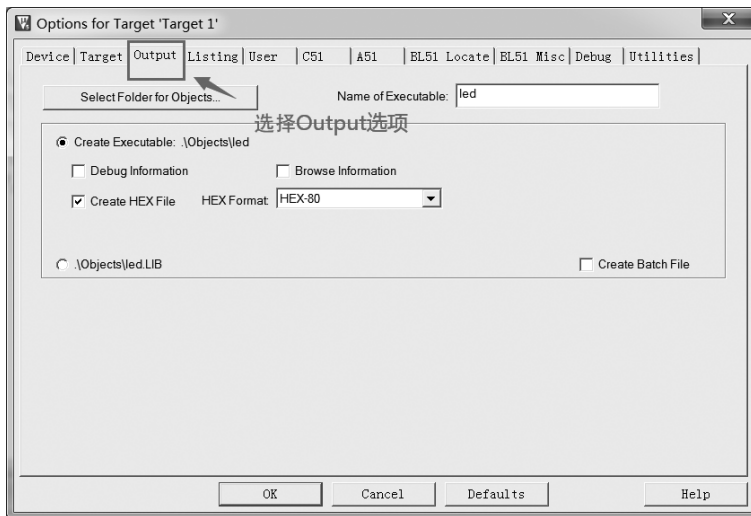


图 1-15 选择“Output”选项

第三步，点击“Select Folder for Objects”设置输出文件保存路径。输出的文件一般会存放在工程文件夹下的“Objects”文件夹内。在“Name of Executable”后填写输出文件的名称。勾选“Create HEX File”选项，工程编译后才会生成 HEX 文件。具体操作如图 1-16 所示。

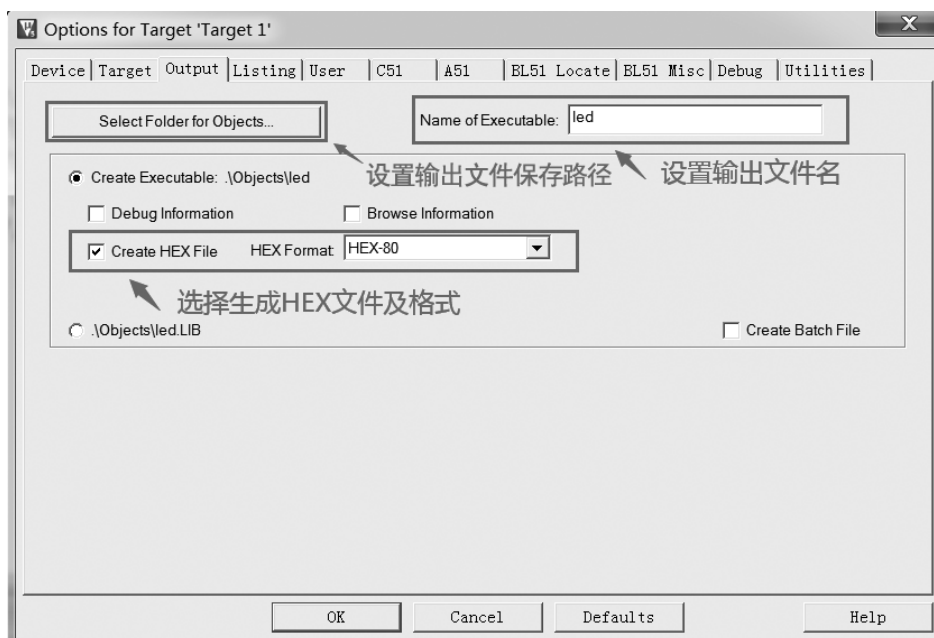


图 1-16 生成 HEX 文件的设置

第四步，点击编译按钮对工程进行编译，编译成功后，在指定目录下生成 HEX 文件。编译输出窗口也会提示 HEX 文件生成信息，如图 1-17 所示。

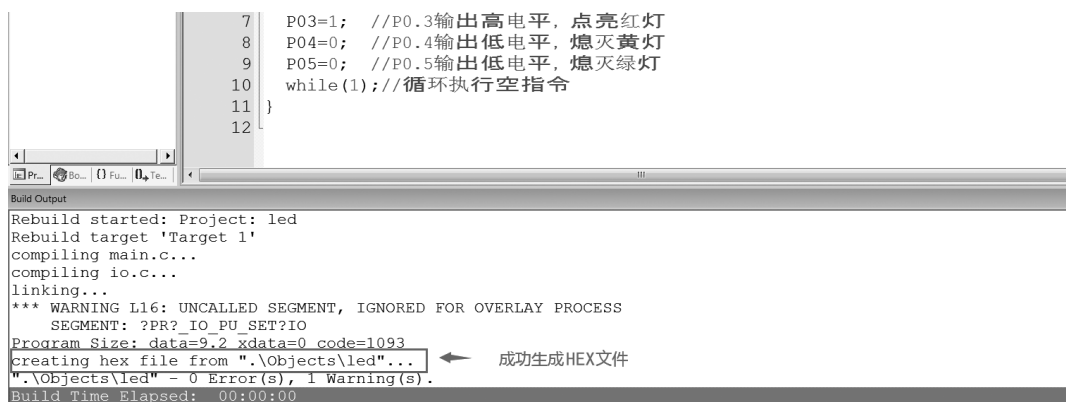


图 1-17 HEX 文件生成信息

需要注意的是，每次更改代码后需要重新编译，才能生成新的 HEX 文件。

## 六、HEX 文件的烧录

HEX 文件生成后，利用 STC-ISP 软件将 HEX 文件烧录至单片机里，以实现代码功能。双击打开 STC-ISP 软件后：

第一步，设置芯片型号，选择 STC8A8K64D4，如图 1-18 所示。





图 1-18 选择芯片型号

第二步，打开串口，选择“USB—SERIAL CH340 (COM11)”，如图 1-19 所示。

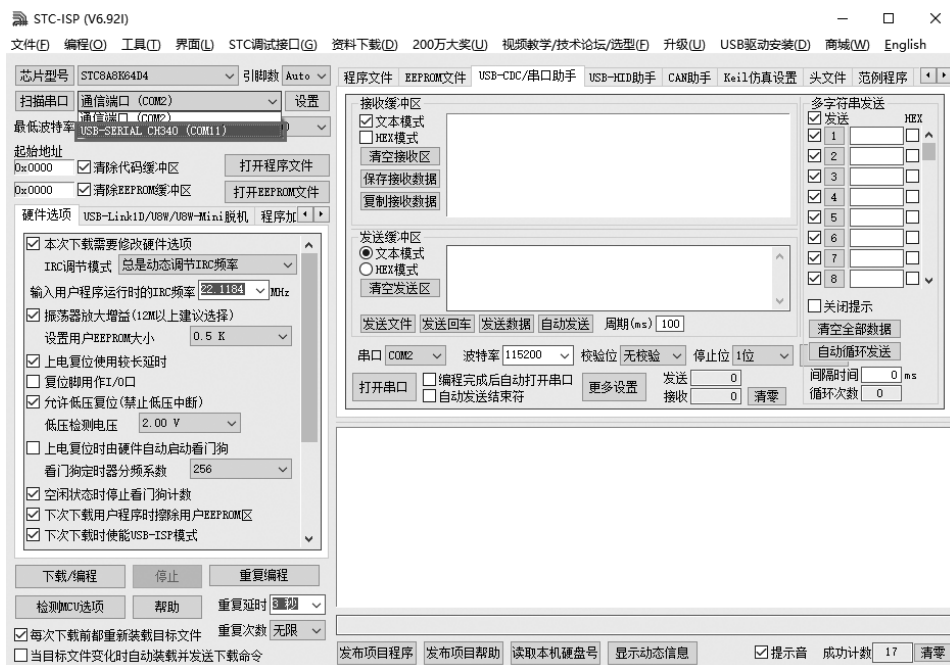


图 1-19 打开串口

第三步，打开程序文件，具体路径为：项目工程→1. 智能灯光控制系统→1. 点亮一个灯→Objects→led. hex。如图 1-20 所示。



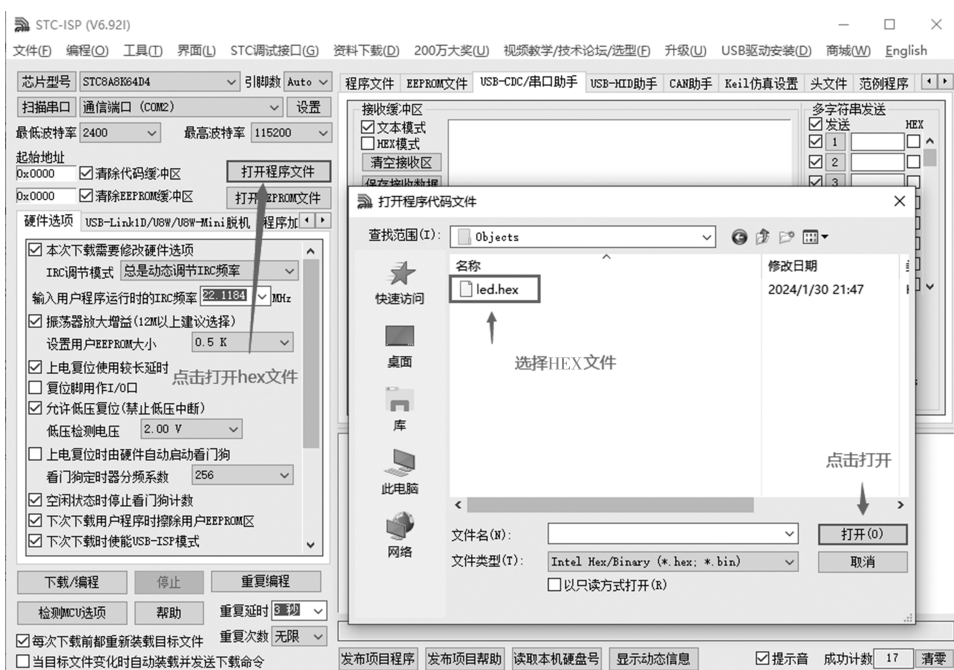


图 1-20 打开 HEX 文件

第四步，设置芯片相关运行参数。本项目单片机没有使用外部晶振，因此勾选“本次下载需要修改硬件选项”设置单片机使用内部晶振。在“输入用户程序运行时的 IRC 频率”处选择频率为 22.1184 MHz。如图 1-21 所示。



图 1-21 设置芯片相关运行参数

第五步，下载编程。先将单片机主控板断电，然后点击“下载编程”。软件右下角会提示“正在检测目标单片机...”。如图 1-22 所示。



图 1-22 检测目标单片机

此时，给单片机主控板供电即可完成下载。如图 1-23 所示。

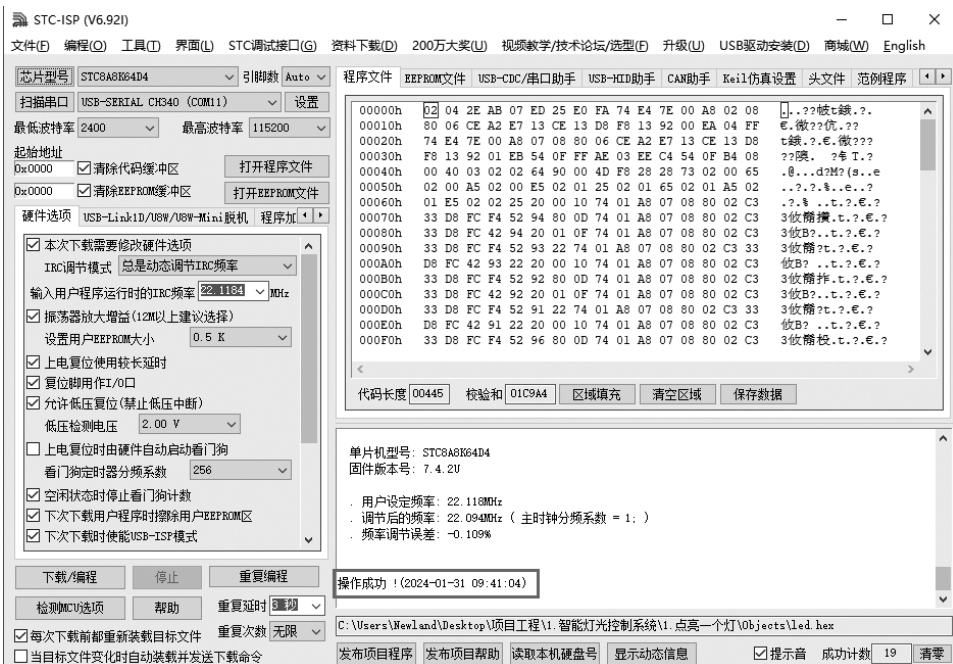


图 1-23 下载成功



## 七、观察实验结果

代码烧录成功后，给系统上电观察现象。通过上面的操作，最终红色的 LED 会被点亮，黄色和绿色的 LED 处于熄灭状态。如图 1-24 所示。

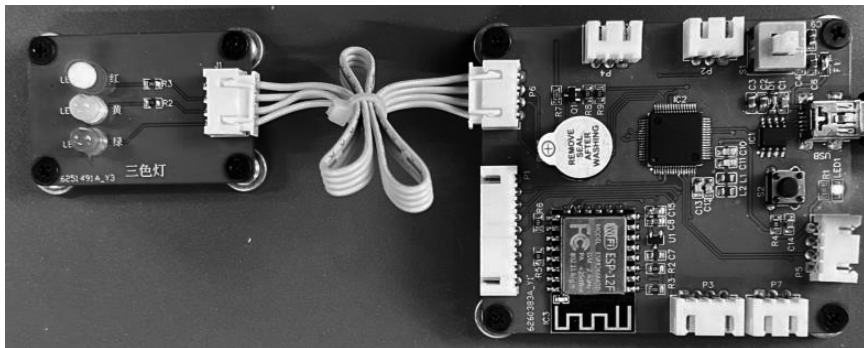


图 1-24 实验结果

## 任务评价

点亮一个灯任务评价表如表 1-1 所示。

表 1-1 点亮一个灯任务评价表

评价点	分值	评价标准	评价要求	评价结果	
点亮一个灯	搭建电路	1	能够正确搭接控制电路	无操作失误	
	打开配套的工程文件	1	打开配套的工程文件，并能打开 main.c 文件进行编辑	无操作失误	
	编写程序	1	能根据前面的分析，写出对应的代码	编辑代码，并知道代码的含义	
	编译程序且无错误	2	编译程序，参考编译输出窗口，检查程序是否有错误，有错误需修改至无错误	编译结果无错误	
	生成 HEX 文件	1	设置使下次编译生成 HEX 文件	生成 HEX 文件	
	正确安装串口驱动	1	安装 CH340 驱动	烧录软件识别到串口	
	下载烧录 HEX 文件	2	正确设置并烧录 HEX 文件至单片机	烧录 HEX 文件成功	
	检查实验结果	1	检查对应的 LED 是否点亮	与要求不符不给分	
总 评					



## 任务小结

点亮一个灯任务小结如图 1-25 所示。

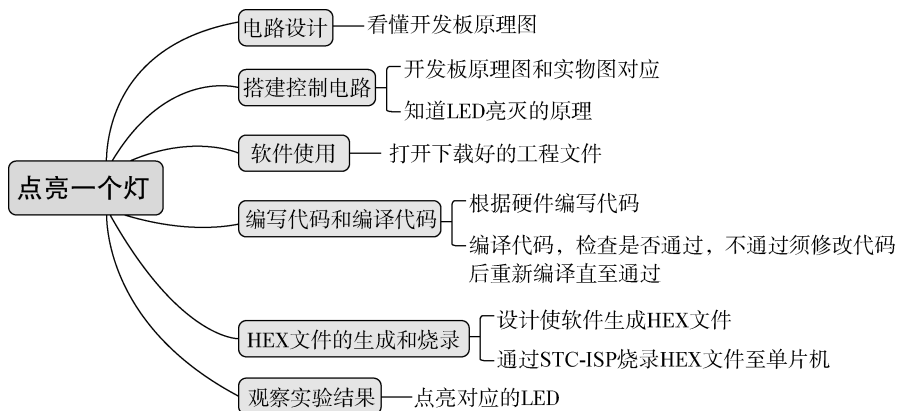


图 1-25 点亮一个灯任务小结

## 任务拓展

1. 思考：程序中不添加“while(1);”，实验现象是否一样？
2. 尝试修改程序，点亮自己喜欢颜色的 LED。
3. 还能用其他方式编写点亮 LED 的代码吗？



## 任务二 流水灯

### 任务要求

在智能家居系统中，灯光有不同的效果，如灯光闪烁、呼吸灯、流水灯等。本任务是搭建流水灯控制电路，通过编程让单片机控制 LED 达到流水灯效果。

### 任务准备

#### 1. 电路准备

单片机主控板、LED 模块。

#### 2. 软件准备

Keil C51、STC-ISP。

#### 3. 任务所需相关知识

STC8A8K 单片机的信号时钟。

为了对 STC8A8K 单片机时序进行分析，需了解三个概念：振荡周期、机器周期、指令周期。

**振荡周期( $T$ )**：又称为时钟周期，由单片机的振荡源产生，常定义为时钟脉冲频率  $f$  的倒数，是时序中最小的时间单位。如果时钟源为 22.1184 MHz，则振荡周期为  $1/22.1184$  MHz，即  $0.0452 \mu\text{s}$ 。时钟脉冲是单片机的基本工作脉冲，它控制着单片机的工作节奏。

**机器周期**：为实现特定功能所需的时间，通常由若干个振荡周期  $T$  构成。STC8A8K 单片机的机器周期常定义为 12 个振荡周期，它是单片机执行一种基本操作的时间单位。

**指令周期**：执行一条指令所需的时间。1 个指令周期由 1~4 个机器周期组成。通常将包含 1 个机器周期的指令称为单周期指令，包含 2 个机器周期的指令称为双周期指令。

### 任务实施

#### 一、实现流水灯的程序流程图

通过仔细观察我们会发现，流水灯是三个 LED 依次点亮形成的效果。流水灯程序流



程图如图 1-26 所示。

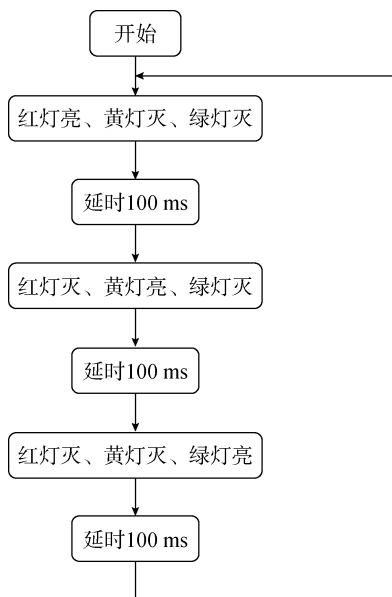


图 1-26 流水灯程序流程图

## 二、延时函数的设计

利用 STC-ISP 软件自带的功能实现延时。STC-ISP 除了具有烧录功能外，还提供了其他功能，如串口助手、Keil 仿真、波特率计算器等功能，软件延时计算器就是其中之一。

使用此功能生成代码的操作步骤如下：

步骤一：调整界面，找到“软件延时计算器”功能。

步骤二：调整频率为 22.1184 MHz。

步骤三：设置延时时间，注意单位有两个，根据需求设置为毫秒级的延时或者微秒级的延时。

步骤四：指令集调整为 STC-Y6。

步骤五：复制代码。默认生成 C 代码，可直接复制使用。

STC-ISP 生成延时函数的具体操作如图 1-27 所示。

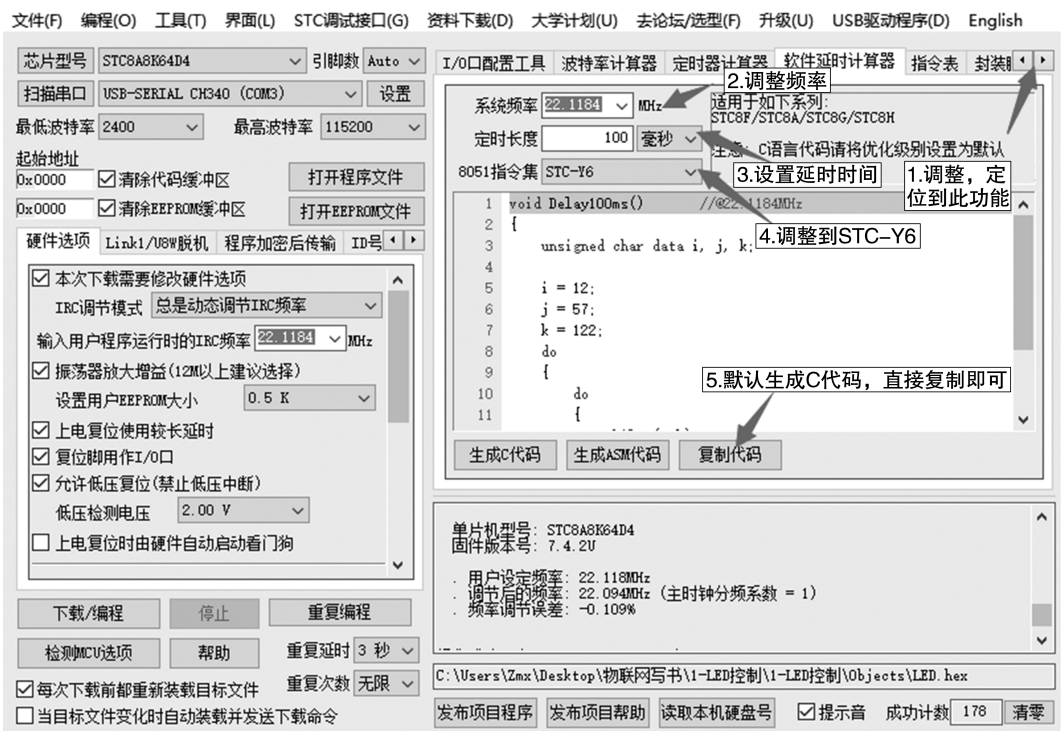


图 1-27 STC-ISP 生成延时函数

生成的延时函数代码如下：

```
void Delay100ms()//@22.1184 MHz
{
    unsigned char data i, j, k;
    i = 12;
    j = 57;
    k = 122;
    do
    {
        do
        {
            while (--k);
        } while (--j);
    } while (--i);
}
```

### 三、函数清单

```
#include "main.h"
void Delay100ms()//@22.1184 MHz
{
```

```

    unsigned char data i, j, k;
    i = 12;
    j = 57;
    k = 122;
    do
    {
        do
        {
            while (--k);
        } while (--j);
    } while (--i);
}
void main()
{
    IO_init(); //I/O 初始化
    P03=0;    //熄灭红灯
    P04=0;    //熄灭黄灯
    P05=0;    //熄灭绿灯
    while(1)
    {
        P03=1; //点亮红灯
        P04=0; //熄灭黄灯
        P05=0; //熄灭绿灯
        Delay100ms();
        P03=0; //熄灭红灯
        P04=1; //点亮黄灯
        P05=0; //熄灭绿灯
        Delay100ms();
        P03=0; //熄灭红灯
        P04=0; //熄灭黄灯
        P05=1; //点亮绿灯
        Delay100ms();
    }
}

```

程序编写完毕后，编译并生成 HEX 文件，通过软件烧录进单片机。通过上面的操作，可发现红灯、黄灯、绿灯依次点亮，实现了流水灯的控制。

## 任务评价

流水灯任务评价表如表 1-2 所示。





表 1-2 流水灯任务评价表

评价点	分值	评价标准	评价要求	评价结果	
流水灯的控制	知道流水灯的原理	1	能够说出流水灯的原理	讲出的原理无差错	
	记下本教材提供的延时函数	2	在 Keil 中编辑代码	函数编写无误	
	使用软件生成延时代码	2	操作软件生成延时的代码	操作无误	
	编写程序并调用延时函数	2	编写流水灯程序，在编写过程中正确调用延时函数	编写程序无误，编译通过	
	生成并烧录 HEX 文件	2	正确操作软件，生成 HEX 文件并烧录进单片机	操作无误	
	检查实验结果	1	是否实现了流水灯效果	与要求不符不给分	
总 评					

## 任务小结

流水灯任务小结如图 1-28 所示。

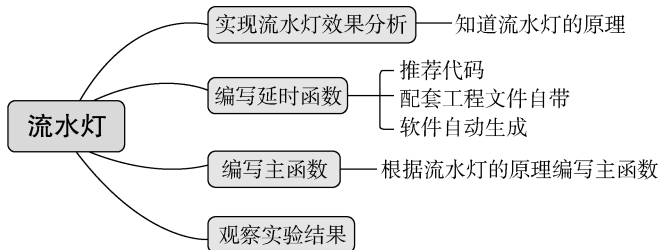


图 1-28 流水灯任务小结

## 任务拓展

1. 尝试让流水灯换一个方向流动。
2. 缩短流水灯的延时时间，观察会出现什么现象。
3. 根据自己的喜好编写呼吸灯的程序代码。



## 任务三 基于新大陆云平台控制 LED

### 任务要求

在智能家居系统中，很多时候我们需要远程控制一些设备。这就要求设备能与云平台进行通信，云平台将控制命令下发给主控板，主控板再控制外部设备。本任务是基于新大陆云平台远程控制 LED，且在云平台中显示 LED 亮灭的状态。

### 任务准备

#### 1. 电路准备

单片机主控板、LED 模块。

#### 2. 软件准备

Keil C51、STC-ISP。

#### 3. 云平台

新大陆云平台。

#### 4. 任务所需相关知识

##### (1) MQTT 协议。

MQTT 是一种机器对机器 (M2M) 的物联网连接协议。它被设计为一个极其轻量级的发布和订阅消息传输。它对于与需要少量代码和/或网络带宽非常宝贵的远程位置的连接非常有用。每个 MQTT 客户端订阅某些主题并在发布者开始推送这些主题的消息时接收消息。实现 MQTT 协议需要客户端和服务端通信完成，在通信过程中，MQTT 协议中有三种身份：发布者 (Publish)、代理 (Broker) (服务器)、订阅者 (Subscribe)。其中，消息的发布者和订阅者都是客户端，消息代理是服务器，消息发布者可以同时是订阅者。

MQTT 传输的消息分为主题 (Topic) 和负载 (Payload) 两部分。

① Topic，可以理解为消息的类型，订阅者订阅 (Subscribe) 后，就会收到该主题的消息内容 (Payload)。

② Payload，可以理解为消息的内容，是指订阅者具体要使用的内容。

##### (2) ESP8266 模块介绍。

ESP8266 是一种强大的 Wi-Fi 模块，它由 ESPRESSIF SYSTEMS 公司开发，可以实现与网络通信的功能。ESP8266 在物联网中被广泛使用，可以通过串行通信接口连接到单



片机制器，以实现智能家居、智能车辆等应用场景。

ESP8266 的主要特点是：

①高度集成：ESP8266 集成了 TCP/IP 协议栈，可以直接连接到 Wi-Fi 网络，无需额外的外部芯片。

②低功耗：ESP8266 的功耗非常低，可以在待机模式下耗电仅为  $20\ \mu\text{A}$ ，适合用于电池供电的应用场景。

③易于编程：ESP8266 可以通过 AT 指令和 Lua 脚本进行编程，非常容易上手。

④低成本：ESP8266 的成本非常低，价格在几美元左右，可以大规模应用于物联网产品中。

ESP8266 的应用场景非常广泛，例如：

①智能家居：ESP8266 可以用于控制智能灯光、智能窗帘、智能门锁等家居设备。

②智能车辆：ESP8266 可以用于智能小车、智能飞行器等控制设备。

③工业自动化：ESP8266 可以用于远程监控、数据采集等工业自动化应用。

④智能农业：ESP8266 可以用于智能灌溉、环境监测等农业应用。

ESP8266 的使用非常简单，只需要几个步骤：

第一步，连接 ESP8266：将 ESP8266 连接到微控制器的串行通信接口。

第二步，设置 Wi-Fi：通过 AT 指令设置 ESP8266 的 Wi-Fi 连接信息。

第三步，连接服务器：通过 AT 指令连接到服务器，实现与服务器的通信。

第四步，发送数据：通过 AT 指令发送数据到服务器。

第五步，接收数据：通过 AT 指令从服务器接收数据。

ESP8266 是一种非常强大的 Wi-Fi 模块，可以实现与网络通信的功能，广泛应用于物联网领域。它的高度集成、低功耗、易编程和低成本等特点，使得它成为物联网开发者的首选模块之一。与本书配套的主控板上就集成了 ESP8266 模块，如果没有集成这个模块，这块开发板就只是一个本地设备，不能够联网，也就称不上是物联网开发板了。主控板上的 ESP8266 模块如图 1-29 所示，开发板原理图上的 ESP8266 模块如图 1-30 所示。

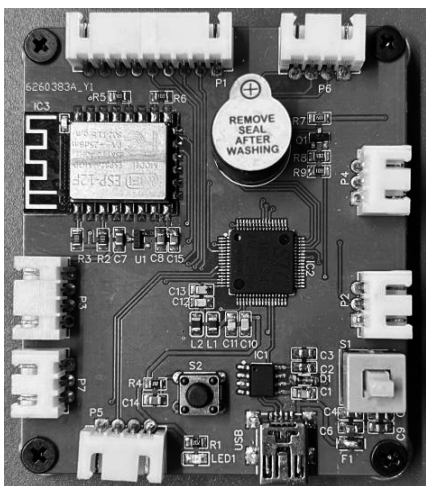


图 1-29 主控板上的 ESP8266 模块



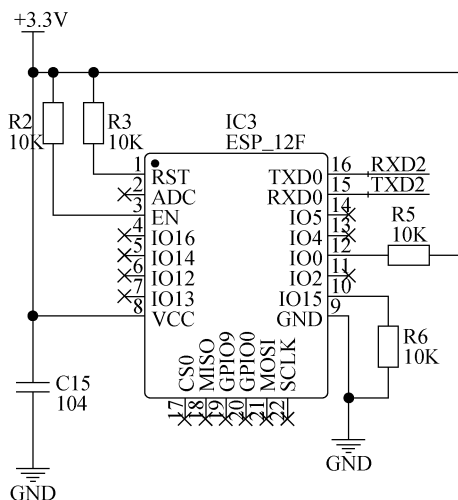


图 1-30 开发板原理图上的 ESP8266 模块

## 任务实施

### 一、控制流程

本任务将会用到三大模块：本地物联网设备、云平台、Web 端，如图 1-31 所示。

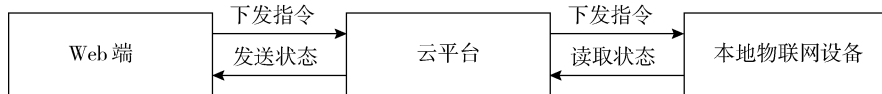


图 1-31 控制流程图

物联网开发板接入了互联网，手机终端接入了互联网，这两个接入互联网的设备之间如何建立联系呢？二者之间的联系建立在物联网云平台上。物联网云平台使互不相干的两个设备之间建立关系，能够让手机显示物联网设备上的信息，也能够通过手机给物联网设备发送指令，从而操控物联网设备。由此可知，在物联网领域，云平台充当着很重要的角色。

### 二、云平台的操作

物联网云平台很多，选择物联网云平台的重要依据是稳定、安全。新大陆云平台安全、稳定、界面简洁，适合新手入门开发。

新大陆云平台网址为 <http://www.nlecloud.com/>，右上角有新用户注册选项，注册完成后登录。

#### (一) 登录云平台

进入登录界面后，填写相应信息进行登录，操作如图 1-32 所示。



图 1-32 登录新大陆云平台

## (二) 生成开发平台 API 密钥

登录成功后，点击右上角开发设置，操作如图 1-33 所示。



图 1-33 开发设置

点击“生成”后出现一串 API 密钥，然后填写 API 密钥的期限。日期尽量延长，后续开发需要此密钥。最后点击“确认提交”。操作如图 1-34 所示。

图 1-34 生成 API 密钥



### (三) 创建项目

返回开发者中心界面，点击“新增项目”或“马上添加一个项目”皆可创建新项目。如图 1-35 所示。



图 1-35 创建新项目

填写项目名称(任意输入)，选择行业类别(任意选择)，联网方式选择 Wi-Fi 模式，然后点击“下一步”，如图 1-36 所示。

The screenshot shows a modal window titled '添加项目' (Add Project). It contains the following fields and options:

- \*项目名称 (Project Name): A text input field with the value '物联网开发' and a note '支持输入最多30个字符' (Supports up to 30 characters).
- \*行业类别 (Industry Category): A dropdown menu with '智能家居' (Smart Home) selected.
- \*联网方案 (Networking Solution): Radio buttons for 'Wi-Fi' (selected), '以太网' (Ethernet), '蜂窝网络(2G/3G/4G)' (Cellular Network), '蓝牙' (Bluetooth), and 'NB-IoT'.
- 项目简介 (Project Introduction): A large text area for a description.
- Buttons: '下一步' (Next Step) and '关闭' (Close).

图 1-36 添加项目

### (四) 添加设备

进入该页面后，填写设备名称(任意输入)，选择通信协议(TCP)，填写设备标识(要求 6~30 个字符)，点击“确定添加设备”即可完成，如图 1-37 所示。



添加设备
✕

**\*设备名称**

支持输入最多30个字符

**\*通信协议**

TCP
  MQTT
  CoAP
  HTTP
  LWM2M
  ModbusTCP
  TCP透传
 ?

**\*设备标识**

! 英文、数字或其组合6到30个字符 解绑被占用的设备

**数据保密性**

公开(访客可在浏览中浏览设备的传感器数据)
  私有

**数据上报状态**

马上启用(禁用会使设备无法上报传感数据)
  稍后启用

确定添加设备

关闭

图 1-37 添加设备

添加设备成功后，进入设备管理页面。将设备标识和传输密钥记录下来，后续编写代码需要使用。点击相应图标(传感器管理)进入设备传感器管理界面，如图 1-38 所示。



图 1-38 设备管理界面

### (五)创建执行器

进入设备传感器管理界面后，点击图后边蓝色“马上创建一个执行器”按钮。进入设备传感器页面后，进行执行器创建，如图 1-39 所示。





图 1-39 执行器创建界面

点击“马上创建一个执行器”后，在添加执行器界面中，完成“三色灯\_红灯”的添加。选择自定义模式，填写传感器名称和标识名。标识名很重要，云平台会依据标识名进行上报和下发报文。传输类型选择“上报和下发”，数据类型选择“布尔型”，操作类型选择“开关型”。具体操作如图 1-40 所示。

### 添加执行器

自定义 NEWLab Modbus 模拟量 数字量 ZigBee 农业器件 家居器件

**\*传感名称**   
用于描述该设备或设备功能的名称（如：灯泡开关、灯泡亮度调节），以中英文、数字或下划线，最多30个字符便于阅读

**\*标识名**   
数据上报及API调用的变量名（如：LightSwitch, LighBrightness），支持英文、数字与下划线，须以英文字母开头

**\*传输类型**  上报和下发  报警  故障

**\*数据类型**

**\*操作类型**  开关型  开关停  按钮型  刻度型  输入框型

图 1-40 添加“三色灯\_红灯”执行器

红灯所有参数填写完毕后，点击“确定并继续添加”按键，实现“三色灯\_绿灯”执行器的添加操作，具体操作如图 1-41 所示。





### 添加执行器

×

自定义
NEWLab
Modbus
模拟量
数字量
ZigBee
农业器件
家居器件

**\*传感名称**

用于描述该设备或设备功能的名称（如：灯泡开关、灯泡亮度调节），以中英文、数字或下划线，最多30个字符便于阅读

**\*标识名**

数据上报及API调用的变量名（如：LightSwitch, LighBrightness），支持英文、数字与下划线，须以英文字母开头

**\*传输类型**  上报和下发  报警  故障

**\*数据类型**

**\*操作类型**  开关型  开关停  按钮型  刻度型  输入框型

确定
确定并继续添加
关闭

图 1-41 添加“三色灯\_绿灯”执行器

重复以上操作，继续添加“三色灯\_黄灯”执行器，如图 1-42 所示。

### 添加执行器

×

自定义
NEWLab
Modbus
模拟量
数字量
ZigBee
农业器件
家居器件

**\*传感名称**

用于描述该设备或设备功能的名称（如：灯泡开关、灯泡亮度调节），以中英文、数字或下划线，最多30个字符便于阅读

**\*标识名**

数据上报及API调用的变量名（如：LightSwitch, LighBrightness），支持英文、数字与下划线，须以英文字母开头

**\*传输类型**  上报和下发  报警  故障

**\*数据类型**

**\*操作类型**  开关型  开关停  按钮型  刻度型  输入框型

确定
确定并继续添加
关闭

图 1-42 添加“三色灯\_黄灯”执行器



所有执行器添加成功后，在执行器界面中就可显示出所添加的内容，如图 1-43 所示。



图 1-43 查看执行器

### (六) 生成应用

回到开发者中心页面，点击“生成应用”创建应用，操作如图 1-44 所示。



图 1-44 生成应用

填写应用相关信息。所属项目系统默认填写，应用名称和应用标识由用户自己定义，操作如图 1-45 所示。



所属项目

应用名称  最多支持输入15个字符!

应用标识  只能是英文组合唯一标识,会以“http://app.nlecloud.com/xxxxxx.shtml”来访问应用

应用模板  自行设计  基础案例  智能家居  项目生成器

分享设置  公开(任意游客可在浏览器中访问以上网址)

应用简介

应用徽标 


图 1-45 填写应用相关信息

应用信息填写完毕后,点击“确定”,生成新的应用,在应用管理界面可查看,如图 1-46 所示。



图 1-46 应用管理界面

### (七) 设计应用

在新增应用中,点击设计图标,对应用进行设计。在设计界面组件库中找到所要设计的项目,用鼠标选中拖动到设计页面中,点击“保存”后再发布,显示发布成功即可,如图 1-47 所示。



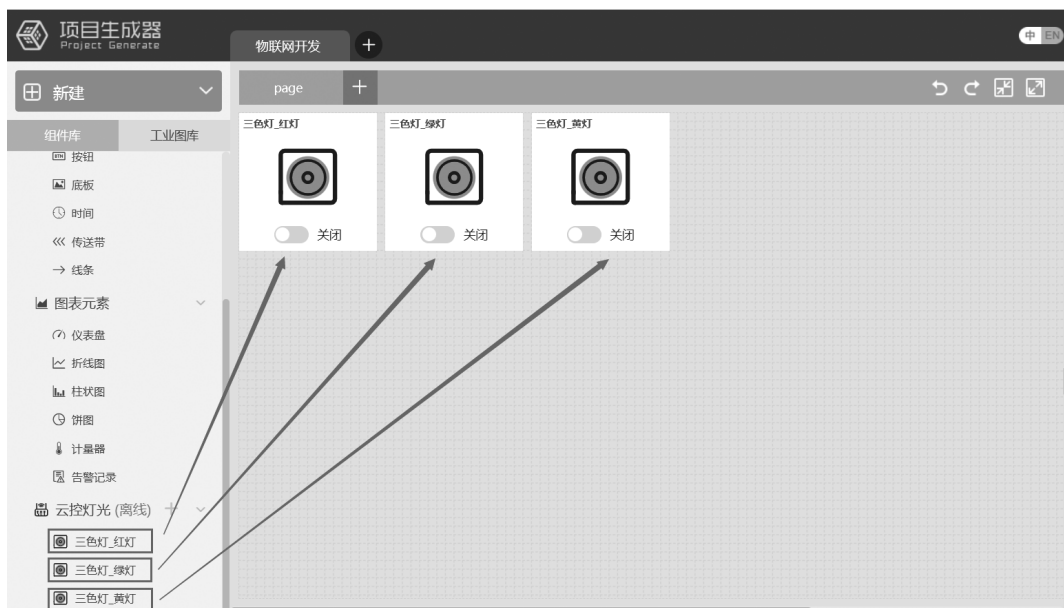


图 1-47 设计页面

发布成功后，返回应用管理页面，可以看到浏览图标由原来的灰色变成蓝色，说明应用已成功发布，如图 1-48 所示。



图 1-48 显示应用发布成功

#### (八) 实现云平台控制 LED

应用发布成功后，点击“浏览”进入应用操作界面。在应用页面中，点击相应图标实现云平台对三色灯的开、关控制，如图 1-49 所示。

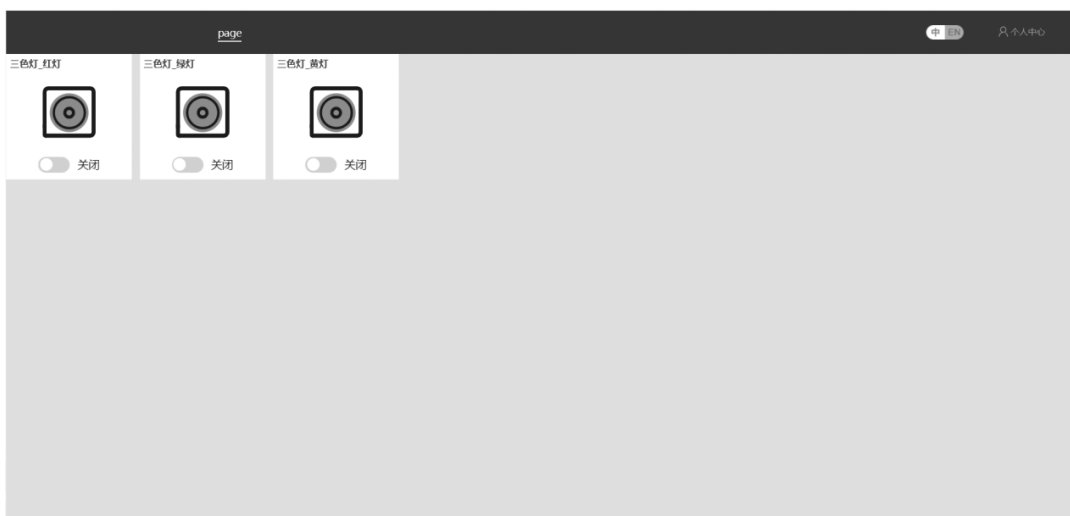


图 1-49 应用操作界面

### 三、程序清单

#### 1. 连接云平台程序。

```
void iot_init()
{
    esp8266_ssid_ste("vivo");           //设置连接 Wi-Fi 名称
    esp8266_key_ste("123456789");      //设置 Wi-Fi 密码
    esp8266_wifi_connect();           //连接 Wi-Fi

    suer_set("IOT_LED");               //设置云平台连接用户名(即设备标识)
    password_set("ad1c6dfd7f1d41b2b86bebeb09fa39f5"); //设置登录密码(即传输
    密钥)
    log_in(); //登录
    Delay_ms(500); //等待 500 ms
    iop_uploading_num("LED_R", 0); //上报红灯熄灭
    iop_uploading_num("LED_G", 0); //上报绿灯熄灭
    iop_uploading_num("LED_Y", 0); //上报黄灯熄灭
}
```

#### 2. MQTT 协议中发送心跳包和解析接收报文程序。

```
void iot_loop()
{
    iot_heartbeat(); //发送心跳包，保持与云平台连接
    if(uart2_r_finish==1) //报文接收完毕
```



```

    {
        iot_analyse();          //解析云平台发送的报文
        uart2_r_count=0;
        uart2_r_finish=0;
    }
}

```

### 3. 解析云平台下发报文的程序。

```

void iot_analyse()
{
    u8 * n;
    n=char_inquire(uart2_rx_buf, "\t", uart2_r_count, 3);
    if((n!=NULL)&&(n[4]=='5'))
    {
        n=char_inquire(uart2_rx_buf, "apitag", uart2_r_count, 8);
        if(char_judgment(&n[9], "LED_R", 7)==1)
        {
            printf("接收 R \r\n");
            led_data[0]=iot_read_data_num();
        }
        if(char_judgment(&n[9], "LED_G", 7)==1)
        {
            printf("接收 G \r\n");
            led_data[1]=iot_read_data_num();
        }
        if(char_judgment(&n[9], "LED_Y", 7)==1)
        {
            printf("接收 Y \r\n");
            led_data[2]=iot_read_data_num();
        }
        printf("R%bd G%bd Y%bd \r\n", led_data[0], led_data[1], led_data[2]);
    }
}

```

将红、绿、黄灯的状态分别存放在 led\_data[0], led\_data[1], led\_data[2]数组中。



#### 4. 完成主要任务 main 函数。

```
#include "main. h"
#include "uart1. h"
#include "io. h"
#include "uart2. h"
#include "esp8266. h"
#include "iot. h"

void iot _ init()
{
    esp8266 _ ssid _ ste("vivo");           //设置连接 Wi-Fi 名称
    esp8266 _ key _ ste("123456789");       //设置 Wi-Fi 密码
    esp8266 _ wifi _ connect();           //连接 Wi-Fi

    suer _ set("IOT _ LED");               //设置云平台连接用户名(即设备标识)
    password _ set("ad1c6dfd7f1d41b2b86bebeb09fa39f5"); //设置登录密码(即传输
    密钥)
    log _ in(); //登录
    Delay _ ms(500); //等待 500 ms
    iop _ uploading _ num("LED _ R", 0); //上报红灯熄灭
    iop _ uploading _ num("LED _ G", 0); // 上报绿灯熄灭
    iop _ uploading _ num("LED _ Y", 0); // 上报黄灯熄灭
}
u32 time _ 200ms=0;
void loop _ 200ms()//200 ms 循环
{
    if(xzsj>=time _ 200ms)
    {
        time _ 200ms=xzsj+200;
        LED _ R=iot _ read _ num("LED _ R"); //读取红灯平台要求状态并赋值
        到 I/O 口
        LED _ G=iot _ read _ num("LED _ G"); //读取绿灯平台要求状态并赋值
        到 I/O 口
        LED _ Y=iot _ read _ num("LED _ Y"); //读取黄灯平台要求状态并赋值
        到 I/O 口
    }
}
void main()
{
```



```

tick_Init(); //系统时钟初始化
IO_init(); //I/O 初始化
LED_R=0; //关闭所有灯
LED_Y=0;
LED_G=0;
BUZZ=0;
uart1_init(); //串口 1 初始化
uart2_init(); //串口 2 初始化
Delay_ms(500);
iot_init(); //云平台初始化
while(1)
{
    loop_200ms(); //设 200 ms 执行函数
    iot_loop(); //IoT 执行函数
}
}

```

5. 编译单片机程序，编译后烧录 HEX 文件至单片机。打开手机热点，将 Wi-Fi 名设置为“vivo”，密码设置为“123456789”。登录新大陆云平台，观察设备是否在线，打开发布的应用，点击图标开关，观察 LED 状态是否有变化。

## 任务评价

基于新大陆云平台控制 LED 任务评价表，如表 1-3 所示。

表 1-3 基于新大陆云平台控制 LED 任务评价表

评价点	分值	评价标准	评价要求	评价结果	
流水灯的控制	新建产品	1	能够新建产品	操作无误	
	新建设备	1	能够新建设备	操作无误	
	使用工具生成签名	1	正确使用签名工具	操作无误	
	创建功能	2	创建出功能	操作无误	
	开发移动应用	3	云平台中正确控制生成项目	正确开发并发布	
	编写程序并烧录	1	编写程序并烧录进单片机	操作无误	
	检查实验结果	1	是否实现远程控制 LED	与要求不符不给分	
总 评					





## 任务小结

基于新大陆云平台控制 LED 任务小结如图 1-50 所示。

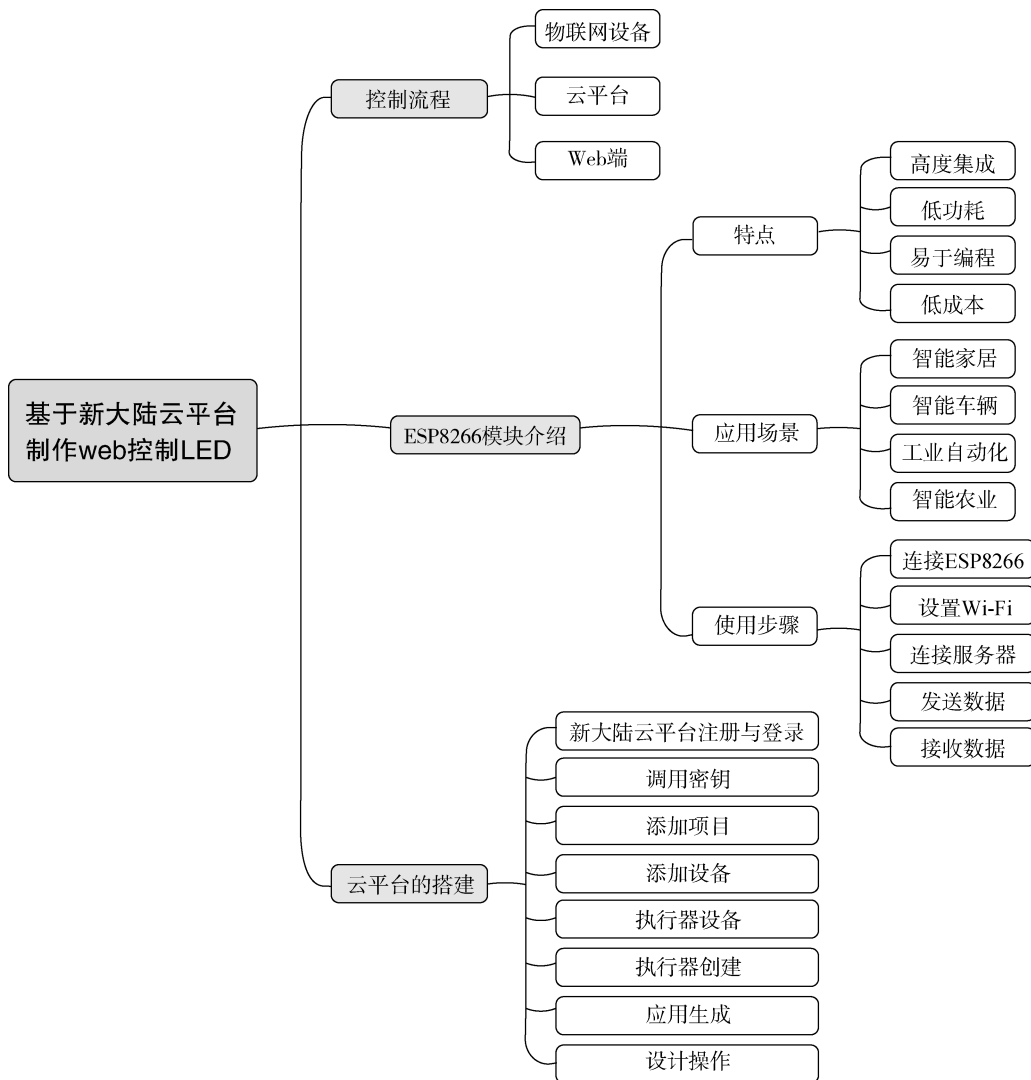


图 1-50 基于新大陆云平台控制 LED 任务小结

## 任务拓展

1. 编写代码实现云平台控制 LED 流动。
2. 编写代码实现不同效果的灯光秀。
3. 根据自己的喜好编写控制灯的程序代码。

