

ESP32使用指南

使用说明

该文档配套使用资料如下，若没有拿到配套资料，请联系邮箱：1224909360@qq.com。

名称

- 1教程
- 2相关软件
- 3示例代码
- 4ESP32硬件资料

资料目录说明

“1教程”内有该文档对应的pdf版，还有Arduino相关的入门学习资料。

“2相关软件”包含Arduino软件以及其他驱动程序或工具，在下文使用过程中会用到，请注意。

“3示例代码”包含教程中使用的示例程序。

“4ESP32硬件资料”包含用到的ESP32开发板的电路图管脚说明等，需要结合里面的说明进行开发。

本指南旨在讲解ESP32在Arduino环境下的使用，有助于快速入门这类单片机的开发流程，部分概念上的细节会省略，网上开源资料已经非常丰富，想要深入了解的可以自行搜索。

阅读基础：有过编程经历，写过一点C语言即可，同时可能会用到简单的电路知识。

- 学习网站推荐：
 - [CSDN – 专业开发者社区](#)：各种丰富的教程，涉及电路、编程等各方面，但内容杂乱，需自行甄别。
 - [Arduino中文社区](#)：面向Arduino平台的中文论坛，资料丰富。（关于Arduino后面会介绍）。
 - [立创EDA开源硬件平台](#)：开源的电路设计论坛，大量DIY作品，主要面向电路设计，适合边复现边学习。

Tips：单片机的开发涉及内容多且杂，建议结合自己需求，选择性地学习，遇到不清楚的地方，善用搜索引擎，绝大部分问题都是前人遇到过，并且有现成答案的。

一、ESP32开发板介绍

首先放几个展示视频：

[ESP32-自平衡莱洛三角形](#)

[esp32音乐频谱小电视](#)

[ESP32微型麦克纳姆轮小车](#)

以上制作都是使用ESP32芯片+Arduino平台编写程序，从而处理数据和控制各种模块工作，功能非常强大，能够实现很多有意思的功能，接下来分别进行介绍。

1. ESP32芯片简介

ESP32是由我国的乐鑫公司推出的一款集成 WiFi 功能和蓝牙功能的微控制器芯片。ESP32 系列芯片规格较多，具体规格可在乐鑫官网查看：[模组概览 | 乐鑫科技 \(espressif.com\)](#)。

Tips: 微控制单元(Microcontroller Unit; MCU)，又称单片微型计算机(Single Chip Microcomputer)或者单片机。一般称单片机（缩写MCU）。--来自百度百科

2. 开发板电路说明

详细电路原理图见硬件资料，实物图及说明如下：

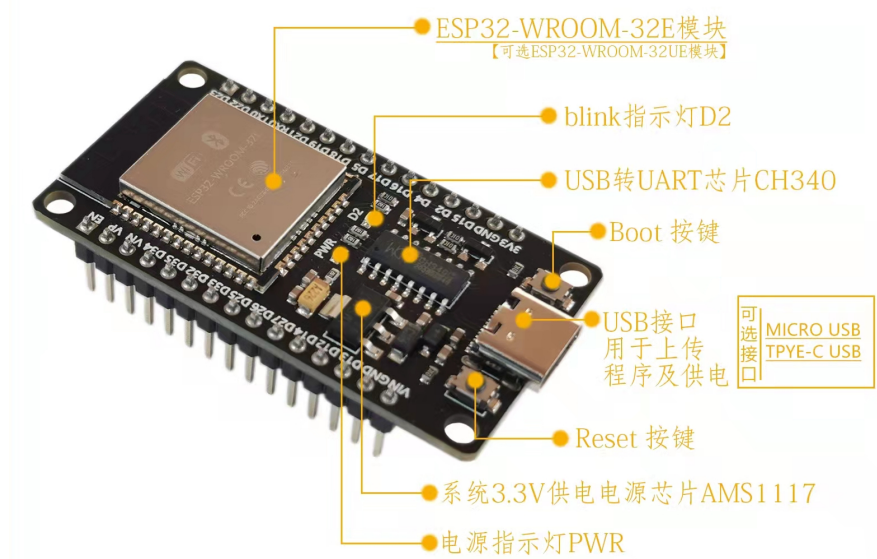


图1-开发板电路示意

型号参数：开发板型号为：ESP32-DevKitC，使用芯片型号为ESP32-WROOM-32E，采用金属罩封装用于屏蔽干扰。WROOM-32系列共有多种规格，但使用方法基本一致，后面将不加区分。

GPIO：（General-purpose Input/Output [gpio_百度百科](#)）指开发板两侧排针引出的IO端口，通过烧写程序控制IO口，能够驱动各种模块工作或者读写数据，对IO口的控制便是我们实现所需功能的主要途径。

除IO口外，排针还引出了GND（接地线）、3V3（即3.3V对外供电接口）、VIN（开发板外部供电接口）、EN（芯片复位管脚），这些接口可用于和其他模块连接和满足供电需要，各管脚功能示意图如下：

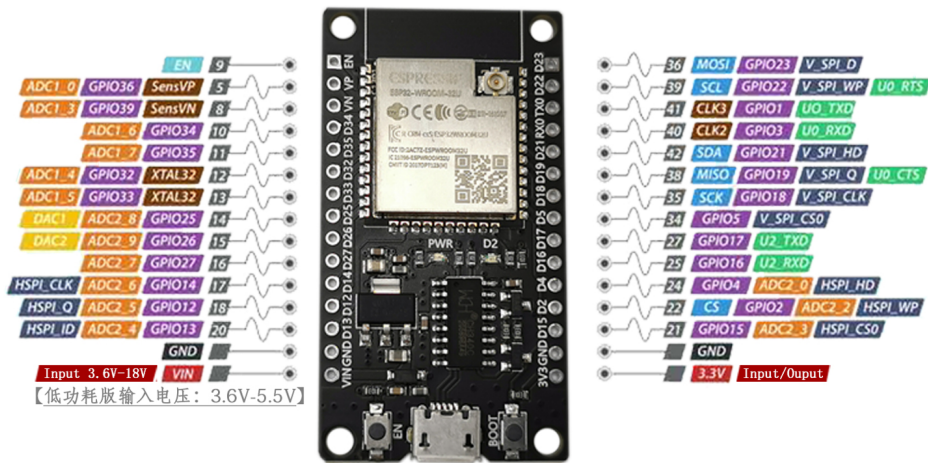


图2-GPIO示意

模块供电：

①通过Type-C或MicroUSB接口连接电脑供电，根据具体模块所带接口选择。USB接口电压为5V，芯片工作电压为3.3V，因此开发板使用了ASM1117（5V转3.3V，LDO降压芯片，见

图1) 使电压稳定在3.3V, 此连接方式还能通过USB接口直接下载程序到ESP32中。

②通过VIN接口供电, 比如使用3.7V锂电池, 正负极分别接到VIN和GND便可以给开发板供电, 这样单片机在工作时就不用拖着数据线了。

通常使用单个锂电池供电时, 需要外加稳压模块升压到5V供电, 否则电池低于3.6V时供电将不稳定。

二、搭建开发环境–Arduino

1. Arduino平台简介

Arduino是一个能够用来感应和控制现实物理世界的一套工具。 它由一个基于单片机并且开放源码的硬件平台(即Arduino开发板), 和一套为Arduino板编写程序的集成开发环境(IDE)组成。官网地址: [Arduino – Home](https://www.arduino.cc/)

由于Arduino编程方式简单, 开源资料丰富, 广受DIY爱好者欢迎。目前许多第三方芯片也提供了Arduino平台开发支持, 推出了相应的软件开发工具包(SDK)。如上述介绍的ESP32, 在Arduino IDE中下载好SDK即可进行开发。

2. Arduino安装与配置

(1) IDE软件获取方式: ①点进官网下载安装; ②在资料文件夹里的”2相关软件”中直接获取, 即arduino-nightly-windows.zip文件。

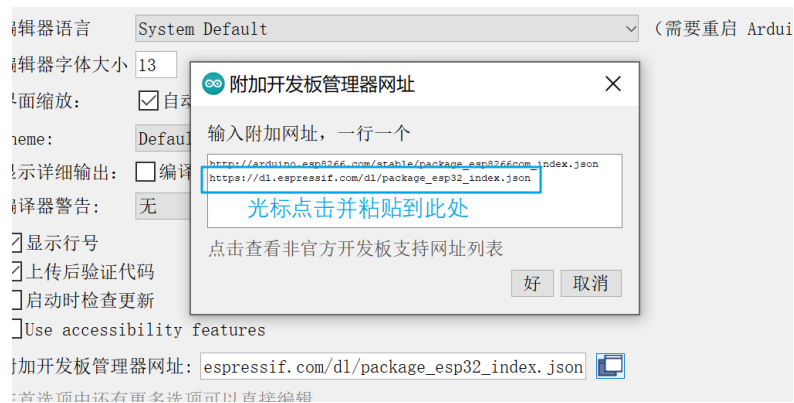
(2) 软件安装: 解压文件, 点开Arduino.exe即可直接启动。建议将其添加到桌面, 方便后续使用。(路径问题未解决, 待测试)

(3) ESP32软件开发包下载:

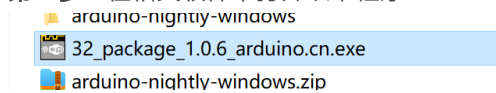
第一步: IDE内: 点击文件->首选项, 添加开发板管理器网址, 复制以下链接:

1 `https://dl.espressif.com/dl/package_esp32_index.json`

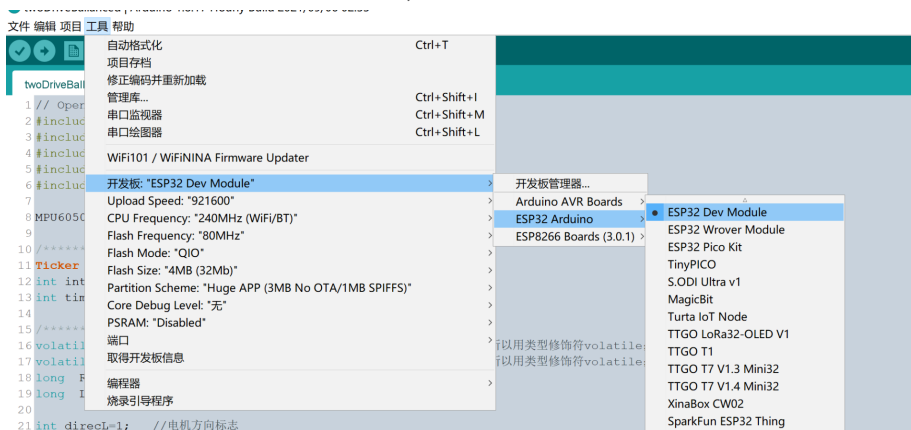




第二步：在相关软件中打开以下程序：



再重启Arduino，即可在**工具->开发板**找到ESP32系列，如果找不到的话，点击**工具->开发板->开发板管理**，加载一遍后退出即可看到开发板，开发板直接选择**ESP32 Dev Module**即可，选好之后将能看到开发板的各个参数，如下图：



3. Arduino IDE使用方法

3.1 快捷键说明



验证：即编译代码，会有报错信息

上传：验证并下载程序到开发板（每次下载均会先验证一遍，比较慢）

新建：新建文件，Arduino文件格式为.ino，每创建一个文件均保存在其同名文件夹内

打开：打开文件

保存：保存文件

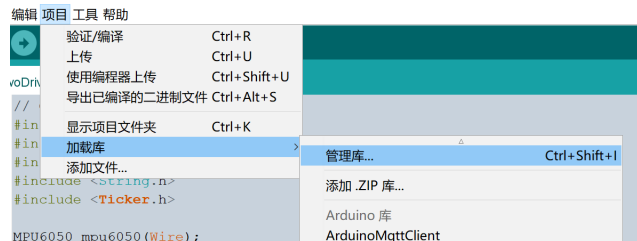
3.2 更改预设路径

在**文件->首选项**中进行使用配置，最好自定义项目文件夹，后续下载的库文件将保存在该文件夹内，方便查看，如下图所示，其余配置可自行修改。



3.3 库文件下载和使用

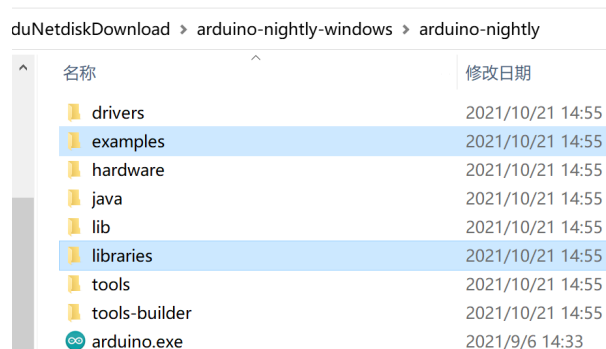
库文件下载：如下图，点击**项目**→**加载库**→**管理库**，可进入下载库文件界面。



Arduino库包括大量开源的官方库和第三方贡献库，资源非常丰富，均可直接下载调用，极大降低了个人开发难度。

库文件的使用：

①官方库文件：在下载软件时便同时下载下来了，如下图，标出了两个非常有参考价值的文件夹：

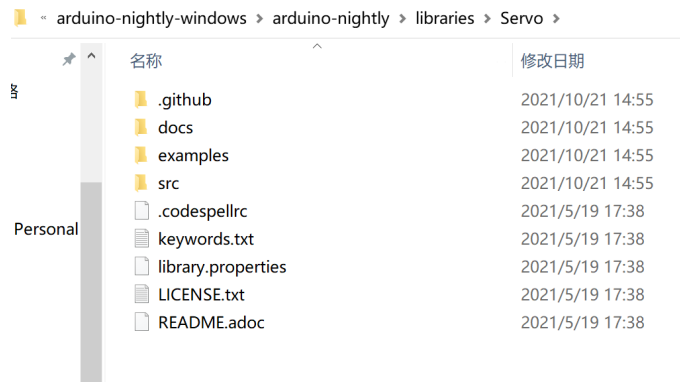


examples内是Arduino基础外设使用的示例代码，开发时可用来参考。

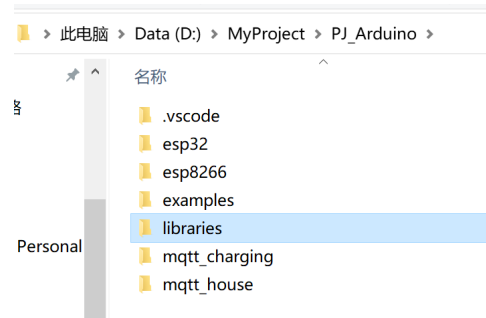
在libraries文件夹内：每个库名字和相关功能相对应。比如下图的servo即用于舵机（伺服电机驱动），其中src内是库函数源码，examples内同样也是对应的参考例程，除此之外还会有README使用说明，可参考学习，但需要一定的英语基础。

说明：

- ①Arduino库文件均以C++的形式进行封装，但接口函数使用起来难度不大，结合示例程序使用即可。
- ②Arduino官方库专门为Arduino开发板所提供，用在ESP32时大部分使用方法一致，但硬件上有差别，IO口编号有区别，同时要注意少部分功能不能在ESP32上使用，等遇到之后再了解即可。



②自行下载的第三方库：在Arduino的管理库界面搜索下载，下载之后，即可在3.2配置的路径中找到libraries，使用方法与官方库类似：



第三方库非常多，也有许多专用于ESP32和各种模块的库，需要在项目文件中include对应的.h头文件后使用。include库的方式有两种：①在项目->加载库直接点击库名即可导入；②手动输入库名称，比如：

```
#include <BluetoothSerial.h> //ESP32蓝牙功能库函数
```

三、使用GPIO点亮LED灯

IO口完整使用指南：[使用Arduino开发ESP32（11）：IO口与相关外设说明与记录](#)

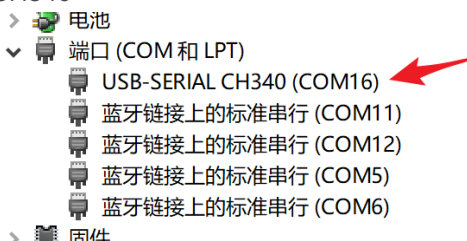
1. 连接ESP32开发板

连接后在工具栏的端口选项中选好端口和自己的开发板对应的型号

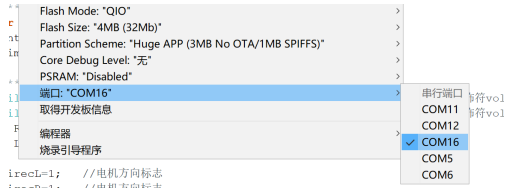


端口选择

首先保证开发板通过数据线连到电脑，然后打开电脑的设备管理器，在端口（COM）能看到CH340：

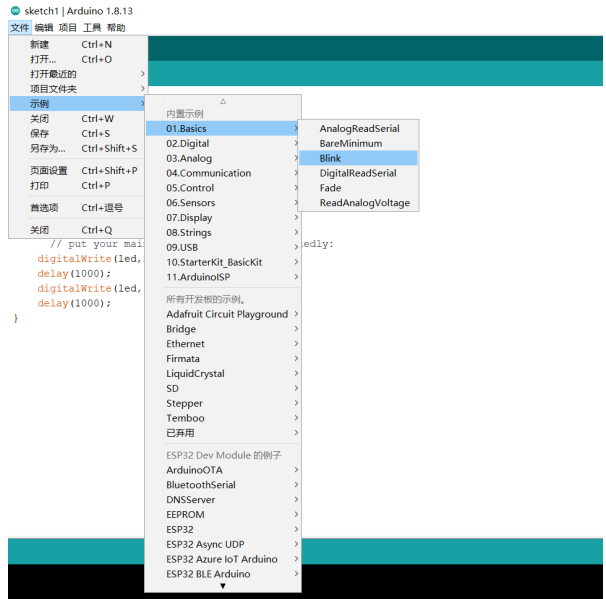


如果是第一次使用，会出现未识别设备，需要安装CH340驱动，在资料文件夹“3相关文件”中解压CH340G并点击运行即可安装。再次查看设备管理器，出现CH340，即可在Arduino的工具-端口栏选择对应端口：



端口一定要和设备管理器对应，才能保证后续下载程序和串口打印正确运行。

2. 导入例程代码



3. 程序结构说明

(如图打开的是blink闪灯程序)

```
/*
  Blink

  Turns an LED on for one second, then off for one second, repeatedly.

  Most Arduinos have an on-board LED you can control. On the UNO, MEGA and ZERO
  it is attached to digital pin 13, on MKR1000 on pin 6. LED_BUILTIN is set to
  the correct LED pin independent of which board is used.
  If you want to know what pin the on-board LED is connected to on your Arduino
  model, check the Technical Specs of your board at:
  https://www.arduino.cc/en/Main/Products

  modified 8 May 2014
  by Scott Fitzgerald
  modified 2 Sep 2016
  by Arturo Guadalupi
  modified 8 Sep 2016
  by Colby Newman

  This example code is in the public domain.

  http://www.arduino.cc/en/Tutorial/Blink
*/

// the setup function runs once when you reset or power up the board
void setup() {
  // initialize digital pin LED_BUILTIN as an output.
  pinMode(LED_BUILTIN, OUTPUT);
}

// the loop function runs over and over again forever
void loop() {
  digitalWrite(LED_BUILTIN, HIGH); // turn the LED on (HIGH is the voltage level)
  delay(1000);                     // wait for a second
  digitalWrite(LED_BUILTIN, LOW);  // turn the LED off by making the voltage LOW
  delay(1000);                     // wait for a second
}
```

注释部分

set up :程序开始的时候运行的程序，只执行一次

loop : 相当于 while (1)
这个程序会不断反复执行

4. 下载程序到开发板



四、使用串口进行通信