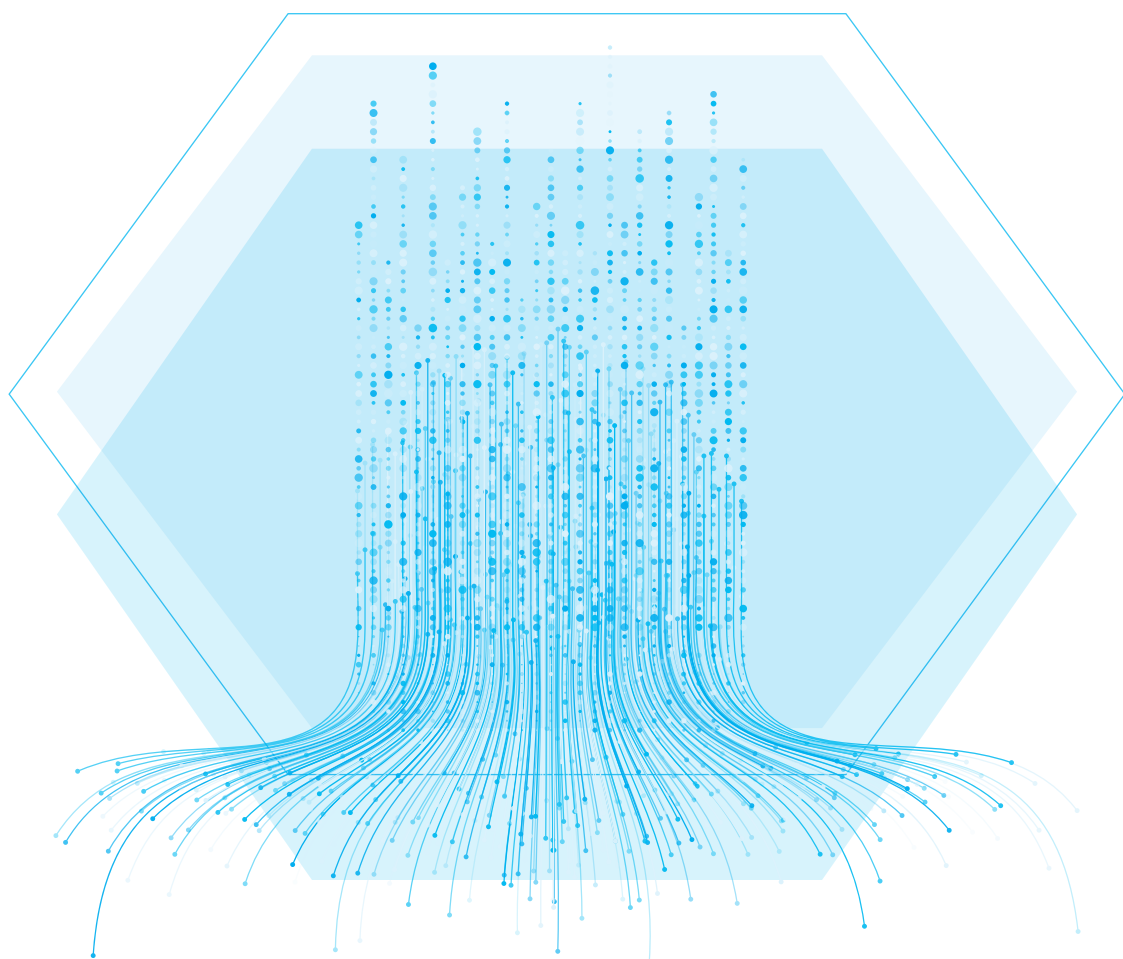


高等院校电子信息系列精品教材

物联网导论： Arduino实例

主 编 胡耀东 王双岭 尹章轩

副主编 何 栋



上海交通大学出版社
SHANGHAI JIAO TONG UNIVERSITY PRESS

内容提要

Arduino 是一款便捷灵活、方便使用的开源电子原型平台,ESP32 是国产 MCU 集成 WiFi 和蓝牙的佼佼者。Arduino+ESP32 的组合能很好地帮助读者进入物联网的殿堂。全书以此为基础展开介绍,共分 10 个章节,包括物联网与工业物联网概述、搭建物联网开发环境、GPIO 接口、模拟量采集、常用接口总线分析、联网、物联网应用综合案例、EDA、Arduino 基础语法和辅助资料。

本书既可作为高等院校计算机类、电子信息类、自动化类、通信类等专业的教学用书,也可作为使用 C/C++ 语言进行物联网开发的工程师及其他对物联网感兴趣的人员的参考书。

图书在版编目 (CIP) 数据

物联网导论: Arduino 实例 / 胡耀东, 王双岭, 尹章轩主编. -- 上海: 上海交通大学出版社, 2025. 7.
ISBN 978-7-313-32819-9

I. TP393.4; TP18

中国国家版本馆 CIP 数据核字第 2025HJ7520 号

物联网导论: Arduino 实例

WULIANWANG DAOLUN: Arduino SHILI

主 编: 胡耀东 王双岭 尹章轩

出版发行: 上海交通大学出版社

邮政编码: 200030

印 制: 大厂回族自治县聚鑫印刷有限责任公司

开 本: 787 mm×1 092 mm 1/16

字 数: 364 千字

版 次: 2025 年 7 月第 1 版

书 号: ISBN 978-7-313-32819-9

定 价: 48.00 元

地 址: 上海市番禺路 951 号

电 话: 021-64071208

经 销: 全国新华书店

印 张: 14.25

印 次: 2025 年 7 月第 1 次印刷

版权所有 侵权必究

告读者: 如您发现本书有印装质量问题请与印刷厂质量科联系

联系电话: 0316-8836866

前言

Preface

21 世纪,人类进入了信息化时代,信息成为新经济时代具有极大开发价值的资源。在通信技术、互联网、传感器等新技术的推动下,人与人、人与物、物与物之间沟通的网络架构——物联网逐步形成。2014 年 5 月,乐鑫发布第一款超低功耗、高度集成的物联网芯片 ESP8266,其被视为物联网、智能家电的开端。编者对 ESP8266 最早的印象是它只是一个 WiFi 模块,仔细搜索其相关资料才发现这个模块支持 Lua、Python、JavaScript、Forth、LISP 等开发语言,还发现它竟然带有 Arduino IDE 的插件,直接支持 WiFi,有不错的性能、丰富的外围接口。这不就是物联网的体现吗?

ESP32 是 ESP8266 的更新版本,其具有更强的性能、更多的资源,支持蓝牙,价格相较 ESP8266 略有提高。

Arduino 是一款便捷灵活、方便上手的开源平台,包含硬件(各种型号的 Arduino 开发板)和软件(Arduino IDE)。Arduino 的硬件原理图、IDE 软件及核心库文件都是开源的,在开源协议范围内用户可以任意修改原始设计及相应代码。

编者为了降低物联网的学习门槛,激发学生学习物联网的兴趣,为物联网的实际应用或毕业设计提供基础资料,让更多学生能轻松学习和掌握物联网开发、嵌入式开发技术而编写了本书。

本书共 10 章。

第 1、2 章介绍了物联网的整体架构、开发环境的搭建。

第 3 章介绍了微控制器的 GPIO 操作,除了融入对硬件的操作,还融入了高级语言的内容,使常量、变量、数组、函数、对象、循环结构等高级语言的语法得以在硬件上体现出来。

第 4、5 章介绍了模拟量的采集和常用接口,使初学者对常用接口有初步了解,为后续课程的学习奠定基础。

第 6 章介绍了联网操作,提供了两种网络的组网模式——PAN 和 LAN,

并没有强调网络的层次结构等内容,而是将其以自然的方式融入物联网应用中。随着后续课程的学习,学生对这部分内容会有更深刻的认识。

第7章提供了两个比较经典的物联网应用案例,能够激发学生的学习兴趣,让学生对物联网有一个整体的印象。

第8章介绍了一些与物联网相关的 EDA 软件。正确使用 EDA 软件,可以简化设计流程,提高效率,达到事半功倍的效果。特别是一些 EDA 软件具有强大的仿真功能,且提供了高端的测试仪表,极大降低了实验的门槛,更加方便教学。

第9、10章讲解了 Arduino 基础语法以及和物联网相关的辅助资料,供查阅。

本书建议配套硬件是以 ESP32-WROOM-32 模组为基础的 ESP32 DEVKIT V1 开发板,这是一块集成了 WiFi 和蓝牙的双核 CPU 物联网开发板,没有说明时,默认使用该开发板。

本书由河南工程学院胡耀东、郑州经贸学院王双岭、郑州工商学院尹章轩任主编,河南工程学院何栎任副主编。郑州木云电子科技有限公司赵庆军对本书要解决的问题,从企业的真实需求提出了宝贵的意见。编者在编写本书的过程中得到了出版社编辑的支持和帮助,在此一并表示衷心的感谢。

由于编者水平有限,书中难免会存在一些不足之处,恳请广大读者指正。

编 者

目 录 Contents

第 1 章 物联网与工业物联网概述	1
1.1 物联网的概念	1
1.2 物联网的体系框架	2
1.3 物联网的应用领域	4
1.4 物联网的发展趋势	6
1.5 工业物联网简介	8
第 2 章 搭建物联网开发环境	11
2.1 物联网开发的整体架构	11
2.2 物联网开发环境的类型	11
2.3 Arduino 的特点	13
2.4 在 Arduino IDE 中搭建 ESP32 开发环境	14
2.5 用 Python IDE 搭建 ESP32 开发环境	22
2.6 用 VSCode+PlatformIO 搭建 ESP32 开发环境	28
第 3 章 GPIO 接口	32
3.1 输出	32
3.2 输入	36
3.3 GPIO 接口的应用	44
3.4 GPIO 接口输出速度的比较	66
第 4 章 模拟量采集	70
4.1 ADC	70
4.2 DHT11 数字温湿度传感器	71

4.3 HC-SR04 超声波测距传感器	76
第5章 常用接口总线分析	81
5.1 USB 接口	82
5.2 RS-232 接口	91
5.3 IIC 总线	97
5.4 SPI 总线	104
5.5 I2S 总线	109
第6章 联网.....	116
6.1 蓝牙	116
6.2 用 WiFi 组建 LAN	125
6.3 物联网协议	136
第7章 物联网应用综合案例.....	150
7.1 ESP32 开发板+阿里云案例	150
7.2 ESP32 开发板+ThingsCloud 案例	159
第8章 EDA	186
8.1 Fritzing	186
8.2 Multisim	189
8.3 Proteus	192
8.4 立创 EDA	198
8.5 其他 EDA 软件	198
第9章 Arduino 基础语法	200
9.1 程序结构	200
9.2 特殊标点及符号	200
9.3 关键字	201
9.4 变量	201
9.5 控制指令	202
9.6 运算符	204
9.7 输入输出函数	205
9.8 时间函数	206
9.9 数学函数	207

9.10 随机数函数	208
9.11 串行通信函数	208
第 10 章 辅助资料	211
10.1 相关器材	211
10.2 一些相关链接	211
10.3 常见开发板外形及引脚功能图	212
参考文献	220

物联网与工业物联网概述

1.1 物联网的概念

宇宙中存在物质、能量、信息三大要素,物质和能量在一定条件下可以互相转化,并遵循爱因斯坦质能方程 $E=mc^2$ 。其中, E 表示物体的静止能量, m 表示物体的静止质量, c 表示真空中的光速。在物质和能量的运动转化过程中,信息不停地演变,形成丰富多彩的现实世界。进入 21 世纪,信息在引导物质和能量的运动转化中发挥的作用越来越强大,信息变成具有极大开发价值的资源,成为新经济时代的优势资源。在通信、互联网、传感器等新技术的推动下,人与人、人与物、物与物之间沟通的网络架构——物联网逐步形成。互联网让人与人之间的距离变成零或者接近零,只存在逻辑关系;而物联网让人与物、物与物之间的距离变得可以忽略不计,只是纯逻辑位置关系。互联网颠覆了人类传统的信息体系架构;而物联网融合了物质、能量,将会强烈冲击和改变世界现有信息体系架构。

物联网掀起了继计算机、互联网和移动通信之后的第三次信息产业革命浪潮,它融合了云计算、云存储等多种技术。可以预测,物与物之间的信息传递规模和人与人之间的通信规模相比将变得越来越大,物联网的应用规模及市场前景极为广阔。

1995 年,比尔·盖茨在《未来之路》一书中对未来的描述为:你不会忘记带走你遗留在办公室或教室里的网络连接用品,它将不仅仅是你随身携带的一个小物件,或是你购买的一个用具,还会是你进入一种新的媒介生活的通行证。这个设想在当时的技术条件下仅是一个梦想。但是,比尔·盖茨对未来的描述为未来信息技术的发展指出了—一个鲜明的方向。

1998 年,美国麻省理工学院(MIT)创造性地提出了电子产品编码(electronic product code, EPC)系统的物联网架构构想。1999 年,在物品编码、射频识别(radio frequency identification, RFID)技术和互联网的基础上,美国 Auto-ID Center 首先提出了物联网的概念。

1999 年,麻省理工学院 Auto-ID Center 提出关于物联网的定义:“物联网把所有物品通过射频识别(RFID)和条码等信息传感设备与互联网连接起来,实现智能化识别和管理。其实质就是将 RFID 技术与互联网相结合并加以应用。”

2005 年,国际电信联盟(ITU)在《物联网技术》报告中对物联网的概念进行了阐述:在任何时刻、任何地点,任何人或物可以连接到任意物体上。为实现无所不在的网络和无所不在的计算,物联网广泛采用 RFID 技术、传感器技术、纳米技术和智能终端技术等。

欧洲智能系统集成技术平台对物联网的定义为:“物联网是由具有标识、虚拟个性的物体/对象所组成的网络,这些标识和个性等信息在智能空间使用智慧的接口与用户、社会和环境进行通信。”

欧盟第 7 框架下 RFID 和物联网研究项目组对物联网的定义为:“物联网是未来互联网的一个组成部分,可以定义为基于标准的和可互操作的通信协议,且具有自配置能力的、动态的全球网络基础架构。物联网中的‘物’都具有标识、物理属性和实质上的个性,使用智能接口实现与信息网络的无缝整合。”

从上述几种定义可以看出,物联网的概念最初起源于利用 RFID 技术对物体进行标识并利用网络进行信息传递的应用,是在不断扩充、完善的过程中而逐步形成的一个概念。最初,物联网架构主要由 RFID 标签、阅读器、信息处理系统和互联网等组成,通过全球唯一编码的标识实现对物品的跟踪、溯源、防伪、定位、监控以及自动化管理等功能。

综合以上观点,本书对物联网的定义如下:采用传感技术获取物体、环境的各种参数信息,通过各种无线通信技术将信息汇总到信息通信网络上,并传输到后端进行数据分析挖掘处理,提取有价值的信息给决策层,然后通过一定的机制、措施实现对现实世界的智慧控制。物联网是一个综合性的技术系统,还可以融合区块链、人工智能、可穿戴设备、增强现实(AR)、人体增强、机器人、自动驾驶、无人机等技术,实现“物联网+”。

我们也可以从数字计算机体系架构来理解物联网。数字计算机体系架构主要分为两种基本类型:冯·诺依曼体系架构(von Neumann architecture)和哈佛架构(Harvard architecture)。这两种架构的区别在于指令和数据的存取方式,冯·诺依曼体系架构的指令和数据存储在同一个存储器中,通过同一条地址总线 and 数据总线传输;哈佛架构是一种分离式存储器体系架构,它将指令和数据存储在不同的内存中,CPU 通过不同的总线从指令内存和数据内存中读取指令和数据。两种架构均由运算器、控制器、存储器、输入设备和输出设备五部分构成。它们最早采用打孔纸带输入 0、1,后来发展为用键盘输入。但键盘的输入速度依然满足不了需求,因此人们研制了声、光、电等各类自动输入设备,有时甚至采用各种传感器网络来满足更快的输入要求。各种输入方式的成熟也使互联网发展出了一个单独的学科——物联网,物联网研究的主要内容就是感知层,也就是各类传感器和传感器网络。

1.2 物联网的体系框架

物联网是一个集成创新技术的综合系统。按照信息生成、传输、处理和应用的原则,可将物联网划分为感知层、网络层和应用层。图 1-1 所示为物联网的三层结构。

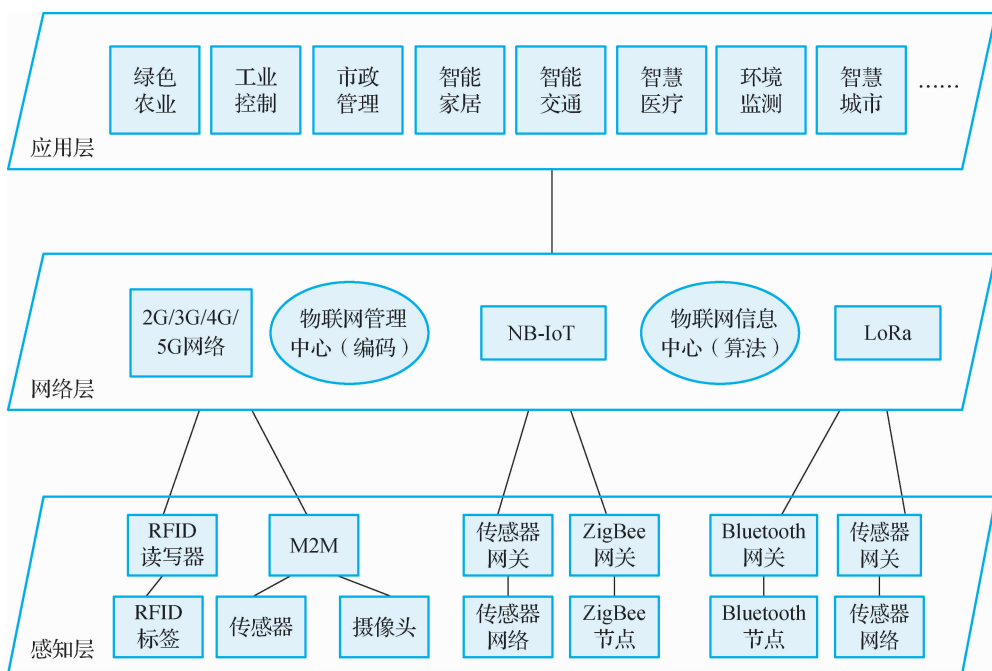


图 1-1 物联网的三层结构

1. 感知层

感知层是物联网的底层结构,是联系物理世界和信息世界的纽带,包括各种感知技术。感知技术本质上是赋予物体“感知与表达”能力的技术,如 RFID 标签中存储的信息,通过无线网络被自动传输到中央系统,实现物体参数的识别和管理。传感器网络主要通过各种类型的传感器获取物质性质、环境状态、行为模式等参数信息。近年来,各类互联网电子产品层出不穷,如智能手机、平板电脑、可穿戴设备等迅速普及,人们可以随时随地接入互联网分享信息。信息生成、传输方式多样化是物联网区别于其他网络的重要特征。

2. 网络层

网络层的主要作用是把感知层数据接入互联网,为应用层提供服务,其功能包括数据传输、交换与信息整合。物联网的核心是互联网。网络层使用各种无线网络提供被感知物体参数信息的网络接入服务。2G、3G、4G 及 5G,虽然网络覆盖较为完善,但存在成本较大、功耗较高的劣势,在难以持续供电的应用场景中使用受限;WiMax 技术(IEEE 802. 16 标准)提供城域范围的高速数据传输服务;WiFi(IEEE 802. 11 系列标准)、Bluetooth(IEEE 802. 15. 1 标准)、ZigBee(IEEE 802. 15. 4 标准)等通信方式的特点是低功耗、低传输速率、短距离,广泛应用于个人电子产品互联、工业设备控制等领域;LoRa、NB-IoT 的特点是低功耗、低传输速率、长距离,适用于智慧城市、数字乡村等应用场景。可根据应用场景采用不同的无线网络技术及其组合,这是实现物联网高效无线传输的关键。

3. 应用层

应用层利用经过分析、处理、挖掘的感知信息数据为用户提供丰富的服务,实现智能化

感知与识别,精准定位与溯源,实时监控和管理。例如,市政管理系统通过一种传感器感应到某个井盖被移动的信息,然后利用网络进行监控并及时采取应对措施。

应用层主要包含应用支撑平台子层和应用服务子层。应用支撑平台子层支持跨行业、跨应用、跨系统之间的信息协调、共享、互通,包括公共中间件、信息开放平台、云计算平台和服务支撑平台。应用支撑平台子层可以将大规模数据高效、可靠地组织起来,为行业应用提供智能的支撑平台。应用服务子层包括智慧城市、智慧医疗、智能交通、智能家居、工业控制等行业应用。

1.3 物联网的应用领域

物联网的应用就是把感应器嵌入和装配到电网、铁路、桥梁、隧道、大坝、油气管道等各种物体中,然后再将其与互联网整合起来,实现人类社会与物理系统的整合。在这个整合的网络中,存在能力超级强大的中心计算机群,能够对整合网络内的人员、机器、设备和基础设施实施实时的管理和控制,在此基础上,人类可以采用更加精细和动态的方式管理生产和生活,使其达到“智慧”状态,提高资源利用率和生产力水平,改善人与自然间的关系。因此,物联网用途非常广泛,遍及智能交通、环境保护、政府工作、公共安全、平安家居、智能消防、工业监测、环境监测、老人护理、个人健康、花卉栽培、水系监测和食品溯源等多个领域。《中华人民共和国国民经济和社会发展第十四个五年规划和 2035 年远景目标纲要》(以下简称《规划》)曾五次提到“物联网”,划定了七大数字经济产业,包括人工智能、大数据、区块链、云计算、物联网、工业互联网、虚拟现实和增强现实。《规划》中提到物联网的内容主要包括:“分级分类推进新型智慧城市建设,将物联网感知设施、通信系统等纳入公共基础设施统一规划建设,推进市政公用设施、建筑等物联网应用和智能化改造……推动物联网全面发展,打造支持固移融合、宽窄结合的物联接入能力。”

1.3.1 智能交通

智能交通是现代交通体系的发展方向,它是将先进的信息技术、数据通信传输技术、电子传感技术、控制技术及计算机技术等有效地集成运用于整个地面交通管理系统而建立的一种在大范围内,全方位发挥作用的,实时、准确、高效的综合交通运输管理系统。

(1)在公交方面,利用物联网技术构建的智能公交系统通过综合运用网络通信、GIS 地理信息、GPS 定位及电子控制等手段,集智能运营调度、电子站牌发布、IC 卡收费、ERP(快速公交系统)管理等于一体。通过该系统,客运公司可以详细掌握每辆公交车的运营状况。另外,在公交候车站台上的乘客通过定位系统可以准确获得等候下一趟公交车需要的时间,还可以通过公交查询系统查询最佳的公交换乘方案。

(2)在智能停车场方面,物联网技术可以帮助人们更快地找到车位。智能化的停车场通过采用超声波传感器、摄像感应、地感传感器、太阳能供电等技术,第一时间感应到车辆停入,然后将车位信息立即反馈到公共停车智能管理平台,显示当前的停车位数量,并将周边地段的停车场信息整合在一起,为车主提供停车向导,能够大大缩短车主找车位的时间。

1.3.2 智能家居

智能家居利用先进的计算机技术、网络通信技术、ZigBee 无线技术,融合个性需求,将与家居生活有关的各个子系统,如安防、灯光控制、窗帘控制、燃气阀控制、信息家电、场景联动、地板采暖等有机地结合在一起,通过网络化综合智能控制和管理,实现“以人为本”的全新家居生活体验。

智能家居可以让用户轻松享受生活。出门在外,用户可以通过手机、计算机远程遥控家中的各智能系统。例如,在回家的路上,可以提前打开家中的空调和热水器;到家时,借助门磁或红外传感器,系统会自动打开过道灯,同时打开电子门锁,撤销安防,开启家中的照明灯具和窗帘;回到家中,用户使用遥控器可以选择预设的灯光场景,如在书房里营造舒适的阅读环境,在卧室里营造浪漫的灯光氛围……这一切,用户都可以安坐在沙发上从容操作。一个控制器可以遥控家里的一切,如关闭窗帘,给浴缸放水并自动加热调节水温,调整窗帘、灯光、音响的状态。如果在厨房配上可视电话机,用户可以一边做饭一边接打电话或查看门口的来访者;在公司上班时,还可以用手机或平板电脑随时查看家里的情况。门口机具有拍照留影功能,家中无人时如果有来访者,系统会拍下照片供用户查看。

1.3.3 智能工业

智能工业是指将信息技术、网络技术和智能技术应用于工业领域,给工业注入“智慧”的综合技术。它突出了采用计算机技术模拟人在制造过程中和产品使用过程中的智力活动,以进行分析、推理、判断、构思和决策,从而扩大延伸和部分替代人类专家的脑力劳动,实现知识密集型生产和决策自动化。例如,中国科学院计算技术研究所将一系列便携式、低成本、无线传感器节点配备在矿工身上,以便在有线系统达不到的地方形成无线感知网络,由此实现井上与井下语音信号的传输,让地面上的人随时了解矿工的工作位置、环境状况及工作进度等。

再如,在工业现场,图形和声音也是极具潜在优势的感知手段,多类型传感器数据从不同角度描述物理世界,对同一场景的多类型数据进行融合,可以得到对环境更为全面且有效的感知。工业仪表无线识读装置是一种新型的工业无线装置,它安装在工业仪表的表盘外,通过内置的图像传感器拍摄仪表盘上的图形,并对其进行处理和识别,得到仪表读数,然后将这些读数通过工业无线网络发送到工厂监控中心,从而实现对传统本地显示仪表的远程监测。

1.3.4 智能物流

智能物流简单地说就是物联网在物流领域的应用。智能物流在物联网广泛应用的基础上利用先进的信息管理、信息处理、信息采集、信息流通等技术完成将货物从供应者向需求者移动的整个过程。其中包括流通加工、仓储、包装、装卸搬运、运输、信息处理等多项基本活动。智能物流是为需方提供最佳的服务,为供方提供最大化利润,同时消耗最少的社会和

自然资源,以最少的投入获得最大效益的整体智能社会物流管理体系。智能物流是物流信息化的发展目标及现代物流业的发展方向。

例如,上海浦菱储运有限公司,在上海总部设立了信息中心,基于 Internet 开发了运输/仓库管理系统通过 Internet 对各分支机构进行统一管理。公司在所属的 50 辆长途厢式货车上安装了 GPS(全球定位系统)定位装置,每隔一定时段(如 1 h)总部就会对车辆的当前位置进行确认。通过 GSM(全球移动通信系统)也可与驾驶员进行联系,下达指令或了解发生的问题,客户可以用公司提供的用户密码登录该管理系统,查询有关信息。

再如,在物流商品中植入传感芯片(节点),使供应链上的采购、生产制造、包装/装卸、堆栈、运输、配送/分销、销售及售后、服务所有环节都能被精准感知和监控。这些信息与后台的 GIS/GPS 数据库无缝结合,构成高效的物流信息网络。

1.3.5 智慧医疗

智慧医疗是指通过构建健康档案区域医疗信息平台,依托先进的物联网技术,实现患者、医务人员、医疗机构及医疗设备之间的互动,逐步达到信息化。未来,医疗行业将融入更多人工智能、传感技术等前沿科技,使医疗服务迈向更高水平的智能化,推动医疗事业繁荣发展。

物联网方案在智慧医疗领域的应用有医院的耗材管理、药品的追踪溯源和血液安全管理等。

例如,为了确保血液存储安全稳定,可将 RFID 芯片与温度感应器集成应用于血液质量监测系统。采血完成后,工作人员即将集成芯片与感应器的标签贴附于血袋上;在运送、仓储过程中,感应器所测得的温度都会被记录在芯片中,在使用前,工作人员可通过读取器查看血液的所有温度记录,借此判断血液质量。

此外,RFID 技术还被应用于特殊药品的智能管控。例如,对于某些药品,在药盒上安装芯片读取器,只有其读取到认可的标签时才会将药盒打开。又如,也可在护士巡房用的药品推车中植入 RFID 系统,医院可以据此管控、记录护士的用药情形和病人的用药时间,以作为日后医疗依据或产生医疗纠纷时的评判证据。

智慧医疗结合无线网、条码、RFID、移动计算、数据融合等技术,可进一步提升医疗诊疗流程的服务效率和质量,提升医院综合管理水平,实现监护工作无线化,全面改变和解决现代化数字医疗模式、智慧医疗及健康管理、医院信息系统等领域的问题。该技术可大幅提高医疗资源共享效率,降低公众医疗成本。

1.4 物联网的发展趋势

随着科技的快速发展,物联网(IoT)已成为推动社会发展的重要驱动力之一。物联网通过连接各种智能设备,实现了信息的互通与智能控制,为人们的生活带来了前所未有的便利。下面将从技术、应用和面临的挑战三个方面深入探讨物联网的未来发展趋势。

1.4.1 技术发展趋势

1. 边缘计算的崛起

随着物联网设备数量的不断增加,数据处理和分析的需求也日益增长。传统的中心化计算架构已经难以有效处理物联网设备产生的海量数据。因此,边缘计算技术应运而生,它将数据处理和分析的任务转移到设备边缘,大大降低了数据传输的延迟率,提高了处理效率。未来,边缘计算将与云计算相互配合,形成云边协同的计算架构,为物联网应用提供更高效、灵活的计算资源支持。

2. 5G 与物联网的深度融合

5G 技术的广泛应用为物联网提供了更高速、更可靠的数据传输能力。5G 网络的高带宽、低延迟特性使得物联网设备能够实时传输大量数据,为各种应用场景提供更多的可能性。未来,5G 将与物联网深度融合,推动物联网技术在各个领域应用和发展。

3. AI 与物联网的结合

AI 技术的快速发展为物联网带来了新的机遇。通过引入 AI 技术,物联网设备可以具备更强的自主学习和决策能力,从而实现更高级别的智能化。例如,智能家居系统可以通过学习用户的生活习惯自动调整室内温度、光照等环境参数,为用户提供更加舒适的生活环境。未来,AI 与物联网的结合将更加紧密,它将推动物联网技术向更高层次发展。

1.4.2 应用发展趋势

1. 智能家居与智慧城市的快速发展

随着消费者对智能生活的需求不断增长,智能家居和智慧城市领域将迎来快速发展期。智能家居系统将更加普及,人们可以通过多种先进物联网技术远程控制家中的各种设备,实现家居生活的智能化。同时,智慧城市的建设也将加速推进,利用物联网技术可以实现城市基础设施的智能化管理和服务。

2. 工业物联网的广泛应用

工业物联网将成为未来工业发展的重要方向。通过引入物联网技术,工业企业可以实现生产过程的智能化改造和自动化升级,提升生产效率和降低成本。例如,可以将工厂中的设备和传感器互联互通,使其实时监测生产数据并进行智能分析,从而提升生产效率和产品质量。未来,工业物联网将在制造业、能源、农业等领域得到广泛应用。

3. 医疗健康领域的创新应用

物联网在医疗健康领域的应用前景广阔。通过物联网技术,医疗机构可以实现对患者身体状况的远程监控和诊断,提高医疗服务的及时性和准确性。同时,物联网还可以帮助医疗机构实现药品追溯、医疗设备管理等功能,提升医疗管理的效率和安全性。未来,随着技术的不断进步和应用场景的拓展,物联网将在医疗健康领域发挥更大的作用。

1.4.3 物联网面临的挑战

物联网的未来发展虽然前景广阔,但也面临着一些挑战。

1. 安全性问题

随着物联网设备的普及,网络安全问题日益凸显。如何确保数据传输和存储的安全性,防止数据泄露和被恶意攻击,是物联网发展需要解决的重要问题。

2. 互操作性

由于物联网设备来自不同的制造商,如何实现设备之间的无缝连接和互操作是一个技术挑战。需要制定统一的标准和协议,以确保设备之间具有兼容性。

3. 数据隐私保护

在物联网环境中,大量个人数据的收集和处理是不可避免的。如何在保护用户隐私的同时充分利用这些数据为用户提供更好的服务,是物联网发展需要平衡的一个问题。

1.5 工业物联网简介

工业物联网(industrial internet of things,IIoT)是当今科技领域的一个热门话题,它代表了工业与物联网技术的深度融合。通过物联网技术,工业制造过程中的各种设备和系统可以实现智能化、互联互通,进而提高生产效率,优化资源配置,降低运营成本,为企业创造更大的价值。

1.5.1 工业物联网的定义

工业物联网是指将物联网技术应用于工业领域,实现工业生产过程中设备与系统的智能化、互联互通的方法。通过传感器、控制器、通信网络以及数据分析等技术手段,工业物联网可以实时监控生产设备的运行状态、收集生产数据,并通过智能分析为企业提供有价值的信息和决策支持。

1.5.2 工业物联网的发展历程

工业物联网的发展可以追溯到 20 世纪 90 年代,当时其用于实现设备之间的简单数据交换和远程控制。随着技术的不断进步和应用需求的提高,工业物联网逐渐发展成为一个涵盖感知、传输、处理和应用等多个环节的复杂系统。

进入 21 世纪后,云计算、大数据、人工智能等技术的快速发展为工业物联网的广泛应用提供了有力支持。这些技术使得工业物联网能够处理和分析海量的数据,提供更加精准和智能的决策支持。同时,随着 5G、边缘计算等新一代信息技术的不断涌现,工业物联网的实时性、可靠性和安全性也得到了显著提升。

1.5.3 工业物联网的应用场景

工业物联网的应用领域非常广泛,几乎涵盖了所有工业制造领域。以下是工业物联网的一些主要应用场景。

(1)智能制造。工业物联网可以实现生产设备的智能化监控和管理,提升生产效率和产品质量。通过实时收集和分析生产数据,企业可以及时发现并处理生产过程中的问题,确保

生产线的稳定运行。

(2)供应链管理。工业物联网可以实时监控物流过程中的货物状态、位置和运输情况,提高供应链的透明度和可追溯性。这有助于企业优化库存管理、减少库存积压和降低运营成本。

(3)能源管理。在能源领域,工业物联网可以实时监测能源消耗情况,为企业提供能源使用报告和节能建议。这有助于企业降低能源消耗、减少碳排放并提高能源利用效率。

(4)远程监控与维护。工业物联网可以实现设备的远程监控和维护,降低企业的运维成本。通过实时监测设备的运行状态和性能数据,企业可以及时发现并解决潜在问题,确保设备的正常运行并延长其使用寿命。

(5)安全与环保。工业物联网还可以应用于工业安全和环保领域。通过实时监测生产过程中的安全风险和环境污染情况,企业可以及时采取措施保障员工的人身安全和企业的生产环境。

1.5.4 工业物联网的技术特点

工业物联网的技术特点主要体现在以下几个方面。

(1)大规模连接与互联互通。工业物联网能够实现海量设备的连接和互联互通,确保数据在设备之间顺畅传输和共享。这为企业提供了全面的生产数据视图,有助于企业做出更加明智的决策。

(2)实时性与可靠性。工业物联网对数据传输的实时性和可靠性要求非常高。通过采用先进的通信技术和数据传输协议,工业物联网可以确保数据在传输过程中的准确性和及时性。

(3)数据分析与智能决策。工业物联网收集了大量的生产数据,需要通过数据分析技术来提取有价值的信息。这些数据可以为企业在生产过程中做出优化建议、故障预测及市场需求预测等智能决策提供支持。

(4)安全性与隐私保护。随着工业物联网的普及和应用范围的扩大,数据安全和隐私保护问题也日益凸显。工业物联网需要采用先进的加密技术和身份验证机制来确保数据传输和存储的安全性,并保护用户的隐私信息不被泄露。

(5)可扩展性与灵活性。工业物联网需要具有良好的可扩展性和灵活性,以适应不断变化的生产需求和技术环境。通过采用模块化设计和开放式架构,工业物联网可以方便地与其他系统和平台进行集成和扩展,满足企业不断增长的业务需求。

工业物联网作为当今科技和工业领域的重要发展方向之一,正推动着工业生产方式的深刻变革。通过实现设备之间的互联互通和智能化管理,工业物联网将为企业提供更加高效、精准和智能的生产方式和管理模式。工业物联网在实际应用过程中尽管还面临着一些挑战和问题,但随着技术的不断进步和应用场景的拓展,相信它将迎来更加广阔的发展前景,并为人类创造更多的价值。

本章小结

1. 随着技术的发展,物联网经历了从概念提出到技术研究、试点实验,再到实际应用和产业化的逐步发展过程。物联网的关键技术包括传感器技术、无线通信技术、数据处理与分析技术、云计算和边缘计算、网络安全技术等。

2. 物联网的架构通常分为感知层、网络层和应用层。感知层负责收集信息,网络层负责传输信息,应用层则是提供服务和应用。物联网的应用领域非常广泛,包括智能家居、智慧城市、工业控制、农业监测、智慧医疗、环境监测、智能交通等。物联网的未来发展趋势可能包括技术标准的统一、安全性的加强、隐私保护的提升、人工智能与物联网的融合,以及物联网在更多领域的深入应用等。物联网在发展过程中也面临着一些挑战,如技术标准的不统一、数据安全和隐私保护问题、设备兼容性问题等。同时,物联网的发展也为各行各业带来了巨大的机遇,推动了产业升级和社会进步。

搭建物联网开发环境

2.1 物联网开发的整体架构

在物联网的三个层面中,网络层和应用层使用了较成熟的网络环境和高层协议,所以物联网的开发重点在感知层,包括信息的采集和动作的执行,并采用单片机加互联网及其延伸出的方式来实现。

在感知层,获取信息的方式主要有两种,即集中式和分布式。集中式主要适用于采集信息地理范围小、采集信息量比较大的场景,由一个比较强的处理器负责采集,并通过网络层将数据上传至应用层。分布式适用于采集信息地理范围较大、采集信息较为分散的场景,常用一个功能较为强大的中心点或者网关负责收集分散节点的信息,然后将信息汇总后通过网络层上传至应用层,分散的节点则负责采集一个或几个信息并将采集的信息上传至中心点或网关。

2.2 物联网开发环境的类型

根据单片机开发方式的不同,可将物联网的开发环境分为裸机开发和嵌入式系统开发两种。

1. 裸机开发

裸机开发早期以汇编语言为主,由设计人员将汇编程序编译后导入单片机中进行调试、验证。目前,裸机开发已经过渡到以 C 语言等高级语言为主,不用直接操作寄存器地址,而是利用现成的类库进行,极大地降低了单片机的开发难度,开发效率也得到了很大的提高。裸机开发使用高级语言进行开发,目前主要有以下两种方式。

(1) Arduino 集成开发环境(Arduino IDE)。它包含一个用于写代码的文本编辑器、一个消息区、一个文本控制台及一个带有常用功能按钮和文本菜单的工具栏。其默认开发环境支持以 AVR 微控制器为主的 Uno、Nano、Mega2560 等开发板,可通过第三方工具扩展,现在可以支持以 STM32 芯片为基础的开发板,也可以支持 ESP8266、ESP32 等开发板。Arduino IDE 可用来轻松创建微控制器项目。开发人员只需要借助 USB 将开发板连接至该

平台,便可以通过简单易用的程序进行开发。由于 Arduino IDE 提供了很多库,所以复杂的部件(如显示器、传感器)及软件平台(如 Adafruit IO)都可以使用。

(2)Keil 集成开发环境。Keil 公司是一家业界领先的微控制器开发工具独立供应商。Keil 公司制造和销售种类繁多的开发工具,包括 ANSI C 编译器、宏汇编程序编译器、调试器、连接器、库管理器和固件等。Keil C51 编译器自 1988 年被引入市场,成为事实上的行业标准,并支持超过 500 种 8051 系列微控制器。Keil 公司在 2005 年被 ARM(安谋科技)公司收购。被收购后,Keil 公司向高速发展的 32 位微控制器市场提供完整的解决方案,同时继续在 μ Vision 环境下支持 8051 微控制器和 C16x 编译器。而后,Keil 公司推出基于 μ Vision 的仿真界面,用于调试 ARM7、ARM9、Cortex-M 内核的 MDK-ARM 开发工具及微控制器领域的其他开发工具,成为业界公认的开发组合。

2. 嵌入式系统开发

嵌入式系统是一种以应用为中心,以计算机技术为基础,并且具有软硬件可裁剪性,适用于对功能、可靠性、成本、体积、功耗有严格要求的专用系统。嵌入式系统一般指非 PC 系统,它包括硬件和软件两部分。硬件包括处理器/微处理器、存储器及外设器件和 I/O 端口、图形控制器等。软件部分包括操作系统(OS,要求实时和多任务操作)和应用程序,或者两者的组合。应用程序控制着操作系统的运作和行为,而操作系统控制着应用程序的编程及其与硬件的交互作用。嵌入式系统的节点功能较为强大,一般在物联网的感知层作为节点或网关,能够保证在对任务实时性要求较高的情况下执行多个任务的时效性。

国外的嵌入式操作系统有 μ Clinux、 μ C/OS-II、eCos、FreeRTOS、Mbed OS、RTX、VxWorks、QNX、NuttX,而国内的嵌入式操作系统包括都江堰操作系统(DJYOS)、AliOS Things、Huawei LiteOS、RT-Thread、SylixOS。

其中开源嵌入式操作系统——RT-Thread是一个集实时操作系统(RTOS)内核、中间件组件和开发者社区于一体的技术平台,也是一个组件完整丰富、高度可伸缩、开发简易、功耗超低、安全性极高的物联网操作系统。RT-Thread 具备一个物联网操作系统所需要的所有关键组件,如图形用户界面(GUI)、网络协议栈、低功耗组件等。RT-Thread 由上海睿赛德电子科技有限公司负责开发维护和运营。通过长时间的沉淀积累,专业化的运营推广,RT-Thread 凭借可靠性高、安全、高度可伸缩性和中间组件丰富易用等特点极大地满足了市场需求,已经成为市面上装机量最大(超 20 亿台)、开发者数量最多(超 20 万)、软硬件生态最好的操作系统之一,被广泛应用于航空、电力、轨道交通、车载、工业自动化、消费电子等众多行业领域。

RT-Thread 拥有良好的软件生态,支持市面上主流的编译工具,如 GCC、Keil、IAR 等,工具链完善、友好;支持各类标准接口,如 POSIX、CMSIS、C++ 应用环境,JavaScript 执行环境等,方便开发者移植各类应用程序。商用 RT-Thread 不仅支持所有主流 MCU 架构,如 ARM Cortex-M/R/A、MIPS、X86、Xtensa、C-Sky、RISC-V,还支持市场上大部分主流的 WiFi 芯片。

RT-Thread 遵循 GPLv2 + 许可证,其内核及所有开源组件在商业产品中都可以免费使用,并且不需要公布应用程序源码,没有潜在商业风险。

对于初次接触物联网的开发人员,建议使用 Arduino IDE 开发环境。该开发环境入门门槛较低,本身含有大量的例程,对于有一定 C 语言基础的开发人员来说更容易上手。当有了一定的基础,对硬件体系有了一定的了解以及开发对存储要求严格的物联网时,建议开发人员使用 Keil 公司的 μ Vision 集成开发环境,该开发环境可以更有效地开发较大的物联网应用。如果开发对实时性有要求的大型物联网应用,那么建议开发人员根据应用要求裁剪合适的嵌入式系统以完成开发设计。

2.3 Arduino 的特点

Arduino 是一款便捷灵活、方便上手的开源平台,包含硬件(各种型号的 Arduino 开发板)和软件(Arduino IDE)。硬件部分是可以用来做电路连接的 Arduino 开发板;Arduino IDE 提供程序开发环境,开发人员将在 IDE 中编写的程序编译上传到 Arduino 开发板后,Arduino 开发板便可执行相应的操作。

Arduino 能通过各种各样的传感器来感知环境,通过控制灯光、电机和其他装置来反馈、影响环境。对 Arduino 电路的编程是通过 Arduino 编程语言和 Arduino IDE 来实现的。基于 Arduino 的项目可以只包含 Arduino,也可以包含 Arduino 和其他一些在 PC 等设备上运行的软件,它们之间通过通信方式来实现信息交换,也就是通常所说的上位机/下位机方式。

Arduino 的特点如下。

(1)跨平台。Arduino 可以在 Windows、Macintosh OS(Mac OS)、Linux 三大主流操作系统上运行,而大多数控制器只能在 Windows 上开发。

(2)简单清晰。Arduino IDE 基于 Processing,对于初学者来说,极易掌握,同时有着足够的灵活性。Arduino 编程语言基于 Wiring,是对 AVR-GCC 库的二次封装,不需要太多的单片机基础和编程基础。

(3)开放性。Arduino 的硬件原理图、IDE 软件及核心库文件都是开源的,在开源协议范围内用户可以任意修改原始设计及相应代码。

(4)发展迅速。Arduino 不仅仅是全球最流行的开源平台,还是一个优秀的硬件开发平台,更是硬件开发的发展趋势。Arduino 的开发方式简单,所以开发者可以更多关注创意与实现,更快完成自己的项目开发,大大节约学习的成本,缩短开发的周期。

因为 Arduino 的种种优势,越来越多的专业开发者使用 Arduino 来开发他们的项目和产品,许多大学专业如自动化、软件,甚至艺术专业,都纷纷开设了 Arduino 相关课程。

2.4 在 Arduino IDE 中搭建 ESP32 开发环境

ESP32 系列产品的设计公司是乐鑫信息科技(上海)股份有限公司,其官方网站提供的软件开发工具包(SDK)有以下三种。

(1) ESP-IDF, 是乐鑫官方网站提供的物联网开发框架,适用于 ESP32、ESP32-S、ESP32-C 和 ESP32-H 系列单片系统(SoC)。它基于 C/C++ 语言提供了一个自给自足的 SDK,方便用户在这些平台上开发通用应用程序。目前,ESP-IDF 已服务支持数以亿计的物联网设备,被用于开发构建了多种物联网产品,如照明、消费电子、大小家电、支付终端、工控等各类物联网设备。

(2) ESP-Arduino, 为 ESP32 开发板提供 Arduino IDE 和开发支持,以开源的方式托管在 <https://github.com/espressif/arduino-esp32>。

(3) ESP-AT, 它将 ESP SoC 集成至 AT 指令(应用于终端设备与 PC 之间的连接与通信指令),以开源的方式托管在 <https://github.com/espressif/esp-at>。

作为初学者,建议首选 ESP-Arduino,也就是利用 Arduino IDE 搭建 ESP32 开发环境,其学习门槛低,方便入门;ESP-AT 通常需要使用 AT 指令,ESP32 开发板作为 WiFi 或蓝牙模块使用,方便其他单片机接入网络;ESP-IDF 作为乐鑫官方主推的物联网开发框架,功能最为强大,使用了嵌入式系统,学习门槛很高,不易入门,不建议初学者使用。

除了乐鑫官方提供的开发方式外,ESP32 开发板作为流行的高性价比的 SoC,众多的第三方也为其提供了开发支持,如 Python IDE 的支持,VSCode+PlatformIO 的支持等。

2.4.1 在 Windows 环境下配置 ESP32 开发环境的方法

在 Windows 环境下,使 Arduino IDE 支持第三方开发板的方法有两种:一种是经典法,即安装第三方开发板的支持软件或驱动;另一种是实用方法,也就是在没有网络的环境下安装第三方开发板的支持软件和驱动。

1. 经典法

(1) 确认 ESP32 开发板能否被 Windows 系统识别。在 Windows 环境下,单击桌面上的此电脑图标,执行“更多选项”→“管理”→“设备管理器”→“端口(COM 和 LPT)”命令,重新插拔连接 ESP32 开发板的 USB 线缆,查看是否有 USB to UART Bridge(COM6)字样,如图 2-1 所示。如果有惊叹号或 X 等新硬件提示,代表 Windows 系统不能识别此 USB 转串口硬件,需要安装新的驱动程序。如果没有任何反应,说明使用的 USB 线缆可能是充电线而不是数据线,需要将 USB 线缆更换为数据线。

(2) 访问 <https://www.arduino.cc/>, 切换到“SOFTWARE”选项卡,选择右侧的“Windows ZIP file”选项,如图 2-2 所示。下载绿色版本 arduino-ide_2.3.3_Windows_64bit.zip。

(3) 解压缩下载的文件,将其放到非中文且不含空格的文件夹下。双击运行文件夹下的 Arduino IDE.exe,需要下载一些文件,安装驱动程序及设置访问网络的请求,界面如图 2-3 所示。



图 2-1 检查串口驱动

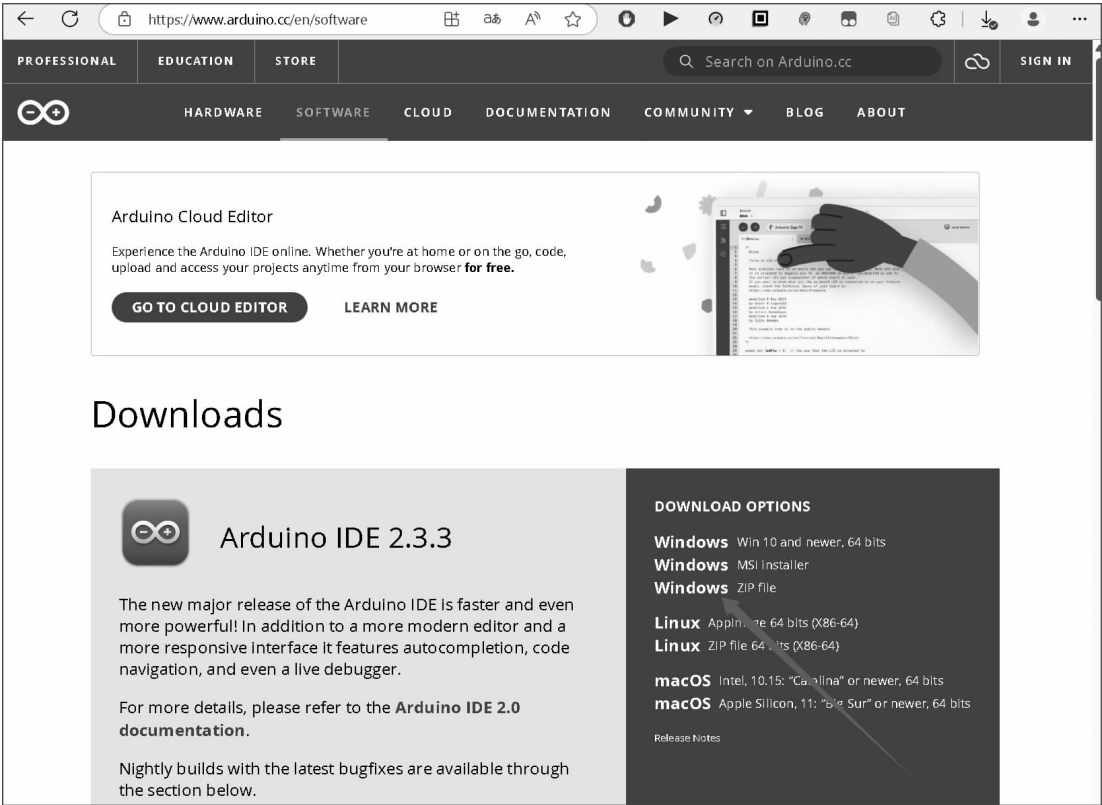


图 2-2 下载 Arduino IDE

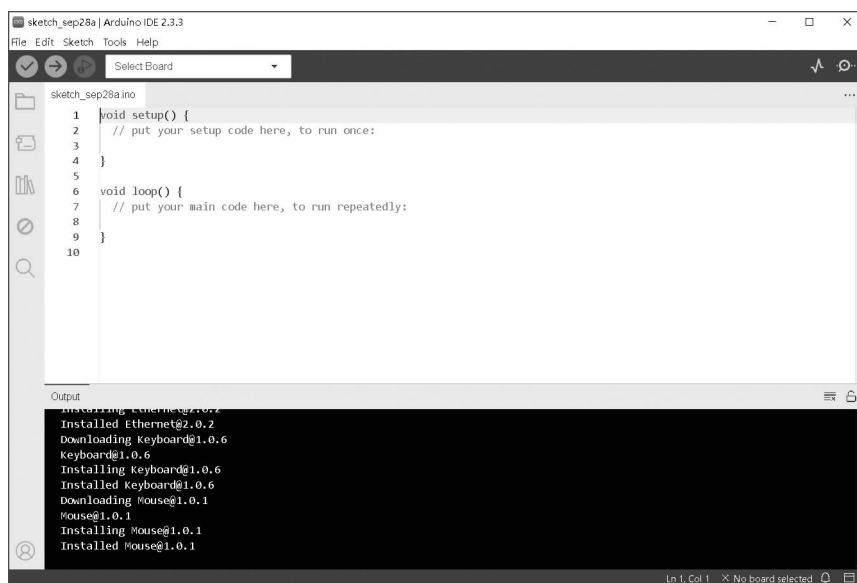


图 2-3 Arduino IDE 的初始界面

(4)执行“File”→“Preferences”命令,单击“Language”下拉列表框箭头,选择“中文(简体)”,单击“OK”按钮,可以将部分菜单转换为中文,如图 2-4 所示。

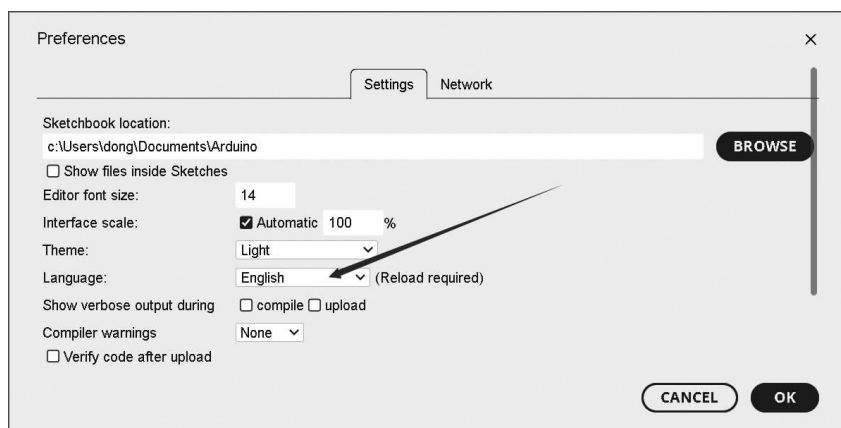


图 2-4 界面语言选择

(5)在 Arduino IDE 界面中,可以直接在主界面执行“工具”→“开发板”→“Arduino AVR Boards”命令,选择 Arduino 官方支持的 Uno、Nano、Mega or Mega2560 等开发板和合适的串行端口,然后就可以连接开发板进行编程、上传程序了。

ESP32 开发板对 Arduino IDE 来说属于第三方开发板,Arduino IDE 默认是不支持的,因此,需要下载合适的插件或者驱动使 Arduino IDE 支持 ESP32 开发板。可以在主界面执行“工具”→“开发板”→“开发板管理器...”命令,或者直接单击主界面左侧的开发板管理器快捷方式,在搜索框中输入“esp32”,在列表中找到“esp32 by Espressif Systems”,如图 2-5 所示。

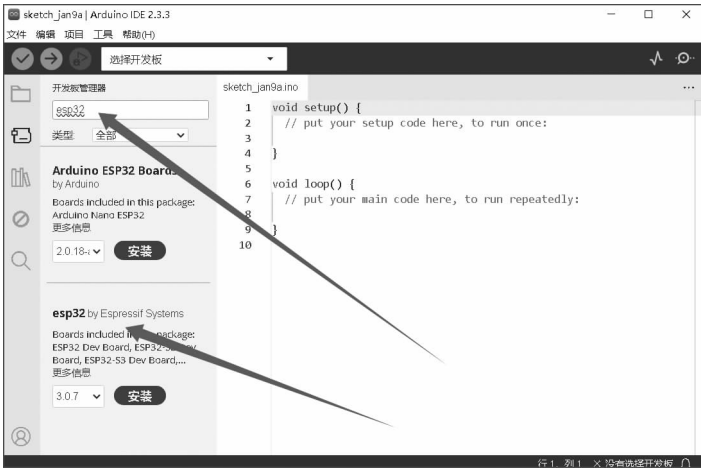


图 2-5 安装 ESP32 开发板支持软件

(6)单击“安装”按钮,安装 ESP32 的支持软件。根据网络状况的不同,这个过程通常需要持续较长时间,或者会出现中断,因此可以选择网速较快时段进行安装。如果安装完成,就在主界面中执行“工具”→“开发板:”ESP32 Dev Module”→“esp32”命令,展开“esp32”列表,选择“ESP32 Dev Module”选项,如图 2-6 所示。这样就完成了 ESP32 开发板支持软件的安装。

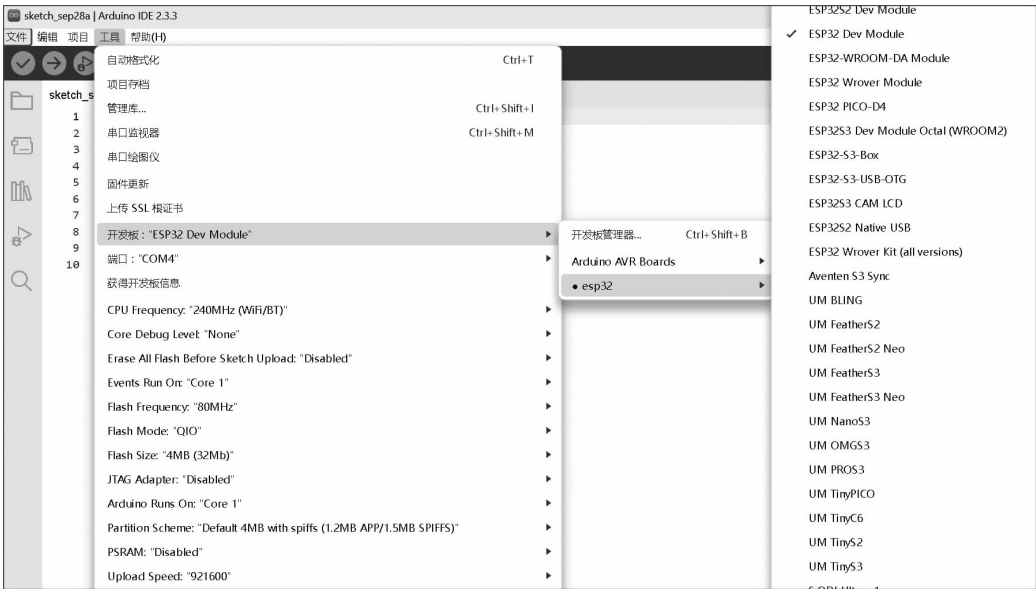


图 2-6 选择 ESP32 开发板型号

2. 实用方法

采用经典法实现 Arduino IDE 对第三方开发板的支持的弊端是对网络要求较高,在有些情况下 3~5 小时也不一定能完成安装。下面介绍一个比较方便,可以在不连接网络的情况下配置 Arduino IDE 支持第三方开发板的方法,这个方法称为实用方法。

(1)下载两个压缩文件,一个是官方的 Arduino IDE 绿色版本,如 arduino-ide_2.3.3_Windows_64bit.zip;另一个是 Arduino15_ESP32.zip,这个文件是由安装在已经支持 ESP32 开发板的 PC 上的压缩“用户目录\AppData\Local\Arduino15”文件夹下的内容组成的。可将这两个文件放到用户自己的 Web 服务器或 FTP 服务器或 U 盘等可移动磁盘中。

(2)在需要安装支持 ESP32 开发板的 Arduino IDE 程序的 PC 中下载或拷贝这两个压缩文件,解压 arduino-ide_2.3.3_Windows_64bit.zip 到非中文、不含空格的路径下,如 d:\arduino (这个是 Arduino IDE 的程序,直接支持官方硬件);解压压缩文件 Arduino15_ESP32.zip,展开为 Arduino15 文件夹,将其复制到“用户目录\AppData\Local”下,如 C:\Users\xxx\AppData\Local(这个是 ESP32 开发板的补丁,这样做就不需要在线下载了,xxx 是登录 Windows 的用户名)。这里需要强调两点:一是拷贝 Arduino15 文件夹的时候,不可以运行 Arduino IDE 程序,否则会找不到该补丁;二是 AppData 文件夹是隐藏的,最方便的方法是在资源管理器中直接输入该路径,或者在“文件夹选项”中选中“显示隐藏的文件、文件夹和驱动器”单选按钮,如图 2-7 所示。

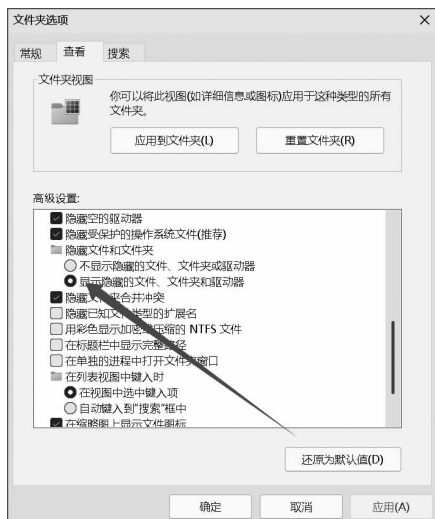


图 2-7 显示隐藏的文件、文件夹和驱动器

(3)双击运行 arduino-ide_2.3.3_Windows_64bit \arduino.exe,可以创建快捷方式,方便以后使用。在 Arduino IDE 主界面中执行“工具”→“开发板”→“ESP32 Arduino”→“ESP32 Dev Module”命令,选择合适的端口,然后就可以运行示例程序了。

2.4.2 在 Linux 环境下配置 ESP32 开发环境的方法

在 Linux 系统环境下,以 Debian12:12.7 released 为例来介绍 EPS32 开发板在 Arduino IDE 中的搭建过程。

(1)确认 Linux 系统能否识别 USB 转串口硬件,也就是确认 Linux 系统是否有合适的 EPS32 开发板驱动程序。在 Linux 环境下,硬件设备是通过设备文件方式来访问的。通常情况下,设备文件的路径在/dev 目录下,USB 转串口硬件的名字一般是/dev/ttyUSBx 或者/dev/ACMx 等,其中,x 表示数字 0、1 等。可以通过直接插拔连接 ESP32 开发板的 USB

线缆的方式来观察/dev 目录的变化,以确认 Linux 系统能否识别 USB 转串口硬件。在 Linux 系统环境下,USB 转串口硬件的设备文件属主是 root,组是 dialout,因此如果能确保当前用户属于 dialout 组,就可以以 dialout 组权限来访问串口设备了。可以执行“: \$ sudo usermod -a -G dialout your_username”命令将用户 your_username 添加到 dialout 组中。

在 Linux 系统环境下,CH34x 芯片的 USB 转串口硬件设备名字通常是 ttyUSBx, CH9102 芯片的 USB 转串口硬件设备名字通常是 ACMx。本例中插入两个 ESP32 开发板,显示的名字是 ttyACM0 和 ttyUSB0,如图 2-8 所示。

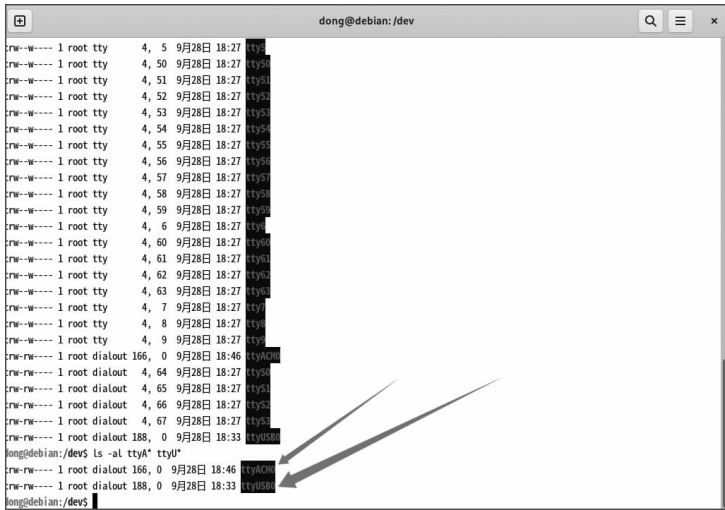


图 2-8 USB 转串口硬件设备文件名

(2)确认 Linux 系统能够识别 ESP32 开发板,下载 Linux 环境下的 Arduino IDE 文件 arduino-ide_2.3.3_Linux_64bit.zip,解压缩文件后直接运行 arduino-ide,此后的界面和配置步骤基本和 Windows 环境下的一致,如图 2-9 所示。

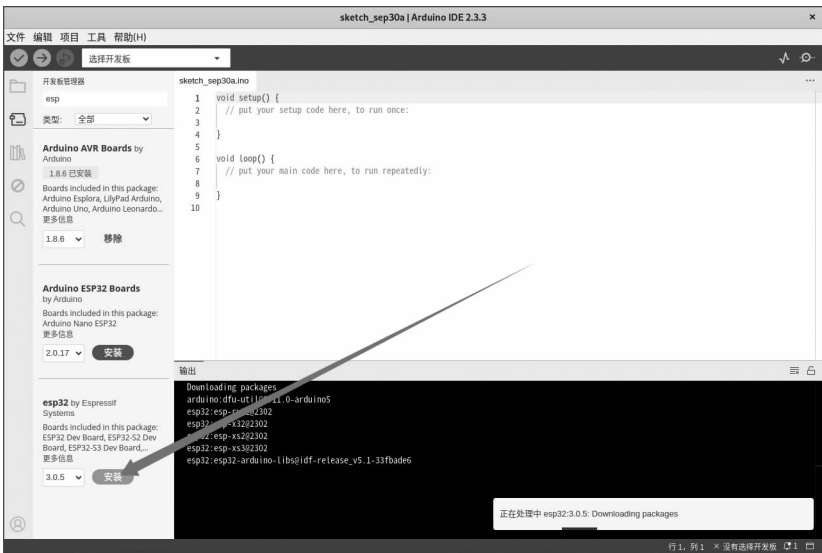


图 2-9 在 Linux 环境下安装 ESP32 开发板的支持软件

(3)安装完 ESP32 开发板支持软件后,选择端口,编译时系统提示找不到 Serial 模块,这时就需要安装 pyserial 模块,可以通过 pip3 install pyserial 命令安装。系统会给出一些安装提示,可以采用系统建议的参数 break-system-packages 来安装,完整地输入命令“pip3 install pyserial —break-system-packages”,如图 2-10 所示。

```
root@debian:~# pip3 install pyserial
error: externally-managed-environment

× This environment is externally managed
╰─> To install Python packages system-wide, try apt install
python3-xyz, where xyz is the package you are trying to
install.

If you wish to install a non-Debian-packaged Python package,
create a virtual environment using python3 -m venv path/to/venv.
Then use path/to/venv/bin/python and path/to/venv/bin/pip. Make
sure you have python3-full installed.

If you wish to install a non-Debian packaged Python application,
it may be easiest to use pipx install xyz, which will manage a
virtual environment for you. Make sure you have pipx installed.

See /usr/share/doc/python3.11/README.venv for more information.

note: If you believe this is a mistake, please contact your Python installation or OS distribution provider. You can
override this, at the risk of breaking your Python installation or OS, by passing --break-system-packages.
hint: See PEP 668 for the detailed specification.
root@debian:~# pip3 install pyserial --break-system-packages
Collecting pyserial
  Downloading pyserial-3.5-py2.py3-none-any.whl (90 kB)
    |-----90.6/90.6 kB 910.3 kB/s eta 0:00:00
Installing collected packages: pyserial
Successfully installed pyserial-3.5
WARNING: Running pip as the 'root' user can result in broken permissions and conflicting behaviour with the system package manager. It is recommended to use a virtual environment instead: https://pip.pypa.io/warnings/venv
root@debian:~#
```

图 2-10 安装 pyserial 模块

(4)安装完 pyserial 模块后,可以顺利运行 Blink,如图 2-11 所示。



图 2-11 运行 Blink

2.4.3 在 macOS 环境下配置 ESP32 开发环境的方法

下面以 macOS15.0 版本,也就是 Sequoia 版本为例,介绍在 macOS 环境下配置 ESP32

开发环境的过程。

(1) 确认 macOS 能否识别 USB 转串口硬件, 和 Linux 一样, 可通过在终端直接查看 /dev 目录下的设备文件名的方式来判断。插拔连接 ESP32 开发板的 USB 线缆, 如果 /dev 目录里出现新的类似 tty. usbserial 字样的设备文件名, 如本例中是 tty. usbserial-52BF0283191, 表示系统可以识别 ESP32 开发板, 如图 2-12 所示。

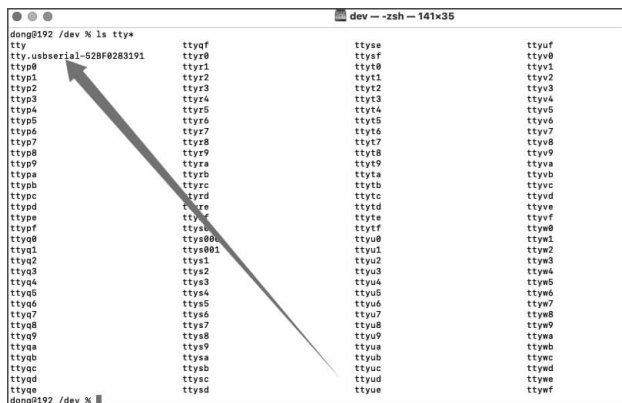


图 2-12 查看 macOS 环境串口硬件

(2) 访问 <https://www.arduino.cc/> 下载 Mac 版本的 arduino-ide_2.3.3_macOS_64bit.dmg, 双击直接运行, 系统会弹出权限请求界面, 单击“允许”按钮, 显示 Arduino IDE 界面, 选择开发板, 在搜索框中输入“esp”, 在列表中选择要安装的支持软件, 单击下面的“安装”按钮, 如图 2-13 所示。

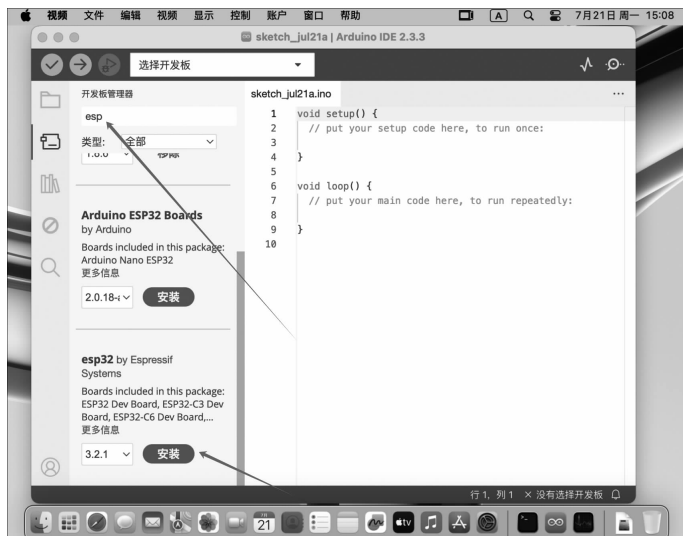


图 2-13 在 macOS 环境下安装 ESP32 开发板的支持软件

(3) 安装完 ESP32 开发板的支持软件后, 打开示例程序 Blink, 添加 #define LED_BUILTIN 2 语句(见图 2-14), 编译、上传并运行程序, 板载蓝色 LED 开始闪烁。



图 2-14 在 macOS 环境下编译 Blink

除了对 ESP32 开发板提供支持外,Arduino IDE 也支持众多的第三方开发板,如常见的 STM32C0、STM32F0、STM32F1、STM32F2、STM32F3、STM32F4、STM32F7 和 Raspberry Pi RP2040 Boards,甚至对部分常见的 FPGA 开发板也提供支持。GitHub 上面提供了一个较为全面的 Arduino IDE 的第三方插件支持项目,可以访问 <https://github.com/arduino/Arduino/wiki/Unofficial-list-of-3rd-party-boards-support-urls> 获取 Arduino IDE 对第三方插件的支持情况。

经过验证,Arduino IDE 对 STM32F1 系列、STM32F4 系列开发板的支持也不错,并且经过配置后,用户可以直接通过开发板自带的 USB 接口上传程序,不需要另外使用 JTAG 或 SWD 等价格较高的仿真工具,这无疑降低了购买硬件的成本。

GitHub 网站在国内并不稳定,经常出现无法访问或访问中断的情况。如果不能访问,可以采用科学上网方法或寻找其国内镜像下载合适的插件。

2.5 用 Python IDE 搭建 ESP32 开发环境

除了官方支持的开发环境外,一些较新的嵌入式开发系统如 Python 也支持 ESP32 开发板,乐鑫官方没有提供 Python 的支持信息,需要自己寻找解决方案,这也是该开发板功能强大的体现,即有众多的第三方支持环境。

Thonny 是一款面向初学者的小型 Python IDE,由塔尔图大学开发,在 Windows、Mac、Linux 环境下均可使用。Thonny 提供了清晰明了的用户界面和导航栏,支持语法着色、代码自动补全、debug 等 IDE 常见功能,并自带 Python 解释器,可在不需要安装其他软件的情况下进行编程和测试。其简单易用的特点能大幅节省开发者下载配置 Python 解释器的时间,而且 Thonny 还集成了大量的 Python 教学资源,如教程、示例代码、在线文档,对于开发者快速入门 Python 很有帮助。

MicroPython 是 Thonny 中的一款 Python 解释器,一般分为三个部分,如图 2-15 所示。

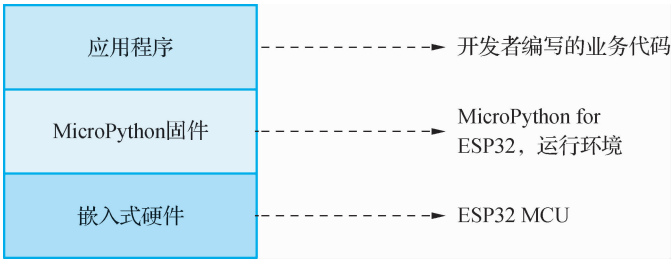


图 2-15 MicroPython 的结构

(1)嵌入式硬件。没有硬件支撑的代码就是一堆看不懂的文字,此部分为固定不变的部分。

(2)MicroPython 固件。没有软件支持的硬件就好比无源之水。要想 ESP32 开发板能运行 Python 程序,就必须给它建立一套可运行环境,也就是在 ESP32 开发板中烧录专用的 MicroPython 固件,这样等运行环境启动后就可以执行应用程序了。此部分也为固定不变的部分。

(3)应用程序。这里特指开发者编写的业务代码,此部分属于多变的。

想要搭建 EPS32 开发环境,就需要下载适用于嵌入式硬件的 MicroPython 固件并将其烧录到硬件中,然后通过 IDE 连接开发板进行代码编译及上传。目前,高版本的 Thonny 已经可以直接支持自动更新 MicroPython 固件,开发者可以直接编写代码、编译运行代码及上传代码。下面介绍用 Python IDE 搭建 EPS32 开发环境的步骤。

(1)直接访问 <https://thonny.org/>, 官方网站提供适用于 Windows、Mac、Linux 系统的 Python 软件,在 Windows 环境下提供有 .exe 格式的安装文件和绿色版本,如图 2-16 所示。

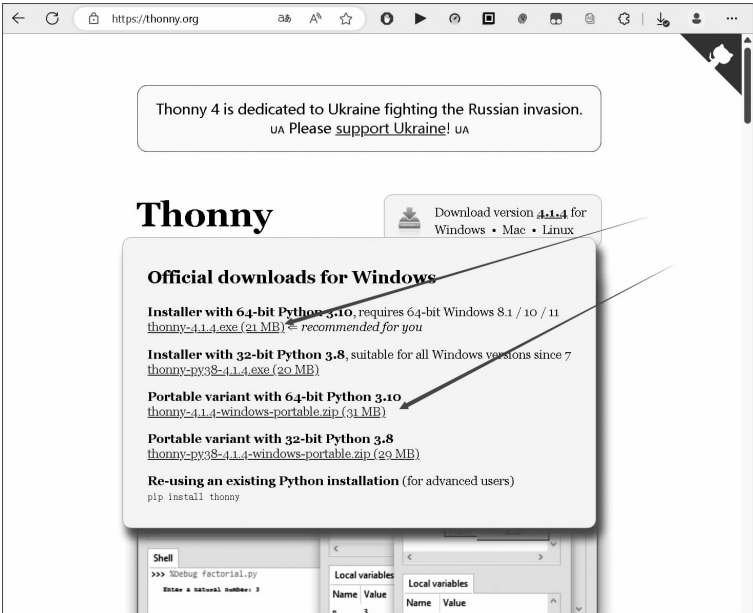


图 2-16 下载 Thonny

■ 物联网导论:Arduino 实例

(2)以绿色版本为例,下载 thonny-4. 1. 4-windows-portable. zip,并将其解压在非中文、不含空格的路径下,直接双击运行 thonny. exe,如图 2-17 所示。

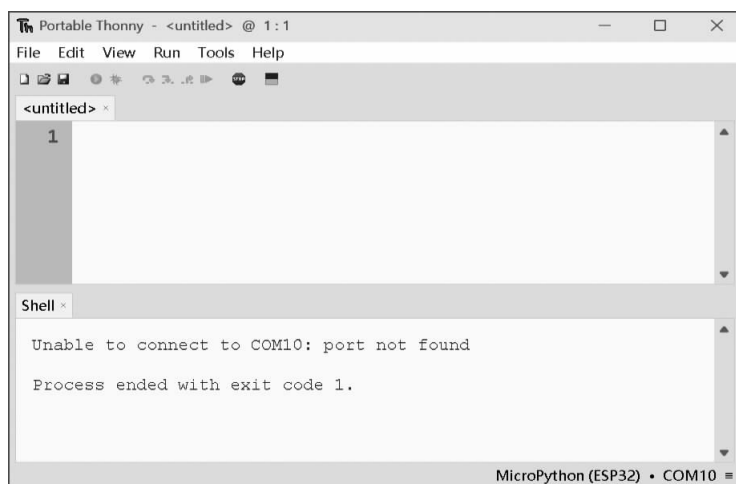


图 2-17 运行 thonny. exe

(3)执行“Tools”→“Options”命令,在“Language”的下拉列表中选择“简体中文”选项,可以将界面切换到中文界面,如图 2-18 所示。

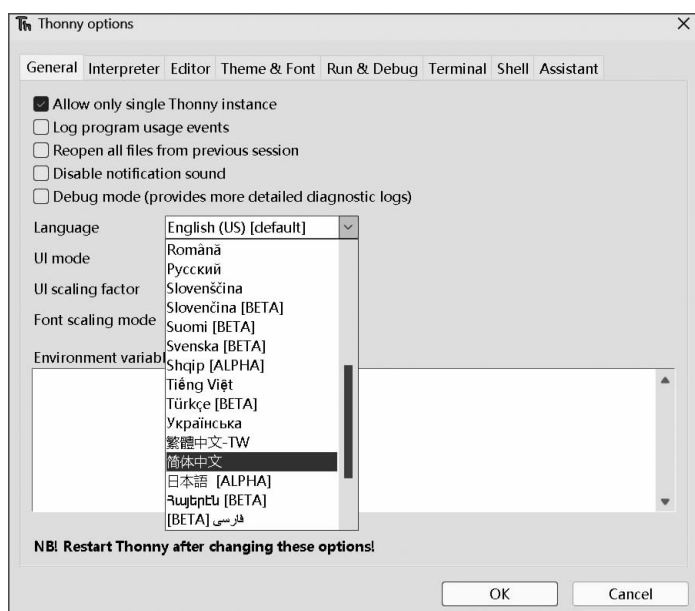


图 2-18 选择中文界面

(4)关闭 Thonny,重新打开后界面切换为中文界面。执行“工具”→“选项”命令,打开“Thonny 选项”窗口,切换到“解释器”选项卡。在“Thonny 应该使用哪种解释器来运行您的代码?”选项区中选择“MicroPython(ESP32)”选项,如图 2-19 所示。

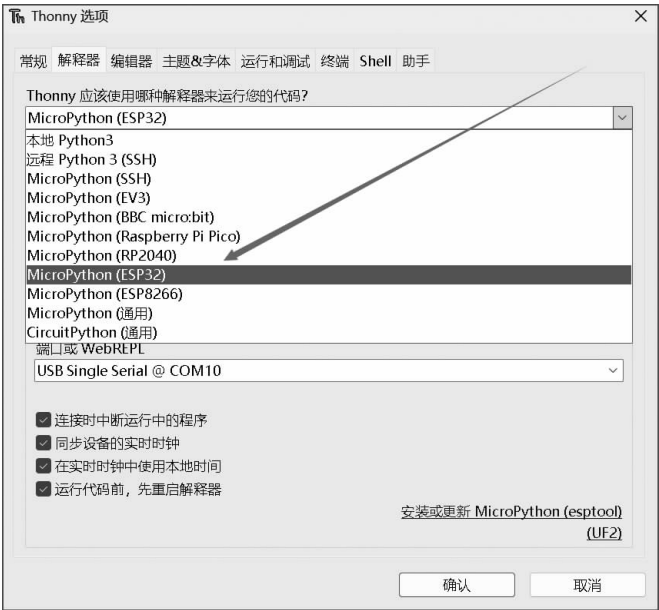


图 2-19 选择合适的解释器

(5)连接 ESP32 开发板到计算机。在图 2-19 下方的“端口或 WebREPL”选项区单击右下方的“安装或更新 MicroPython(esptool)”，弹出“Install MicroPython(esptool)”对话框，如图 2-20 所示。

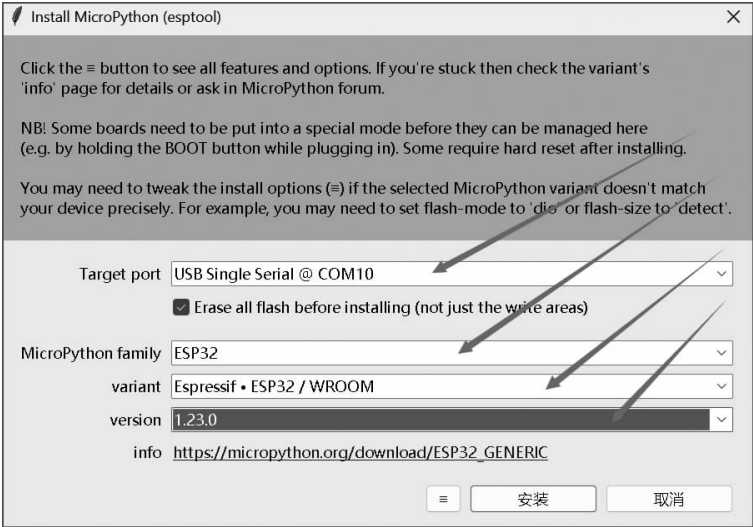


图 2-20 安装或更新 MicroPython

(6)此处以安装 MicroPython(esptool)为例,选择合适的 Target port、MicroPython family、variant、version,单击“安装”按钮,即可在线安装或更新 MicroPython。高版本的 Thonny 省略了另外下载合适的 MicroPython 的过程,比低版本的 Thonny 方便多了。安装

■ 物联网导论:Arduino 实例

或更新完成后,单击“关闭”按钮,返回主界面,主界面如图 2-21 所示,这表明 MicroPython 已经安装成功。

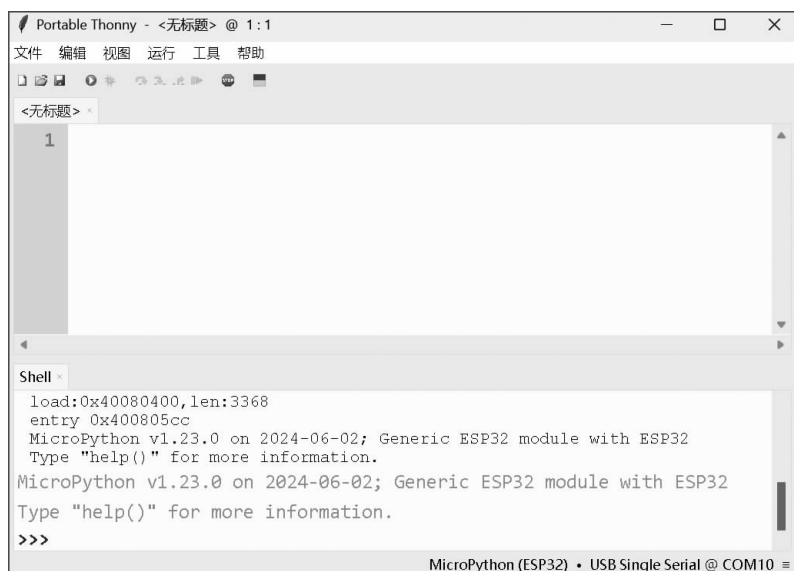


图 2-21 MicroPython 安装完成后的主界面

在“Shell”窗口中按提示输入“help()”,就可以显示帮助信息。在帮助信息里,使用 help('modules')可以显示已经安装的模块。在“Shell”窗口中输入以下内容。

```
from machine import Pin
p2 = Pin(2,Pin.OUT)
while True:
    p2.value(1)
```

直接运行代码(见图 2-22),发现 ESP32 开发板的板载 LED 已经点亮。

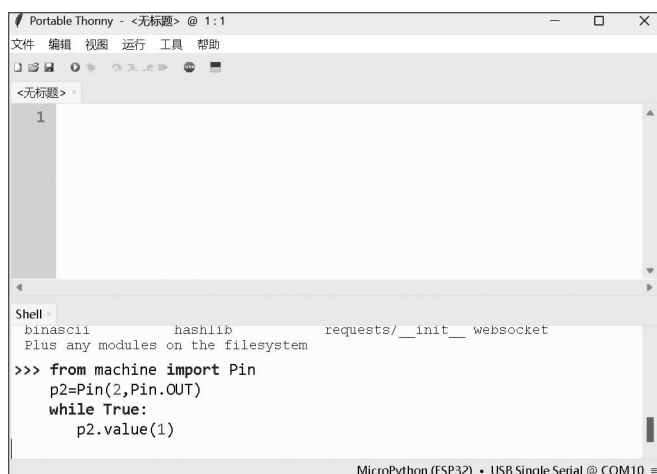


图 2-22 在“Shell”窗口中直接运行代码

在代码输入框中输入以下内容。

```
from machine import Pin
import time
p2 = Pin(2, Pin.OUT)
while True:
    p2.value(0)
    time.sleep_ms(500)
    p2.value(1)
    time.sleep_ms(500)
```

在主界面中执行“文件”→“保存”命令,弹出保存地址提示框,如图 2-23 所示。

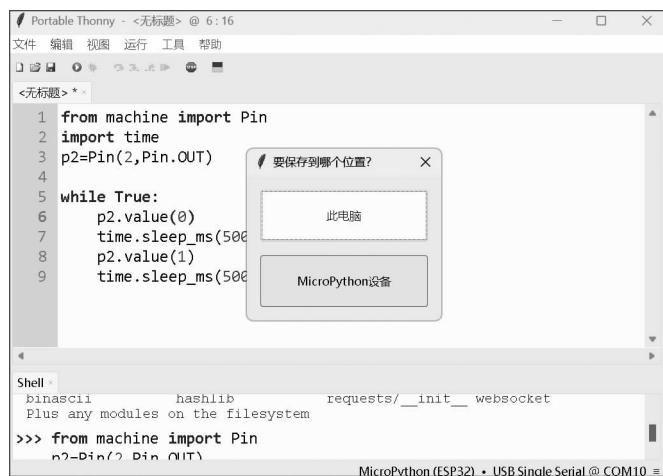


图 2-23 保存文件

选择“MicroPython 设备”,系统可能会提示设备忙,这时可以重启一下开发板,重新保存文件,如图 2-24 所示。

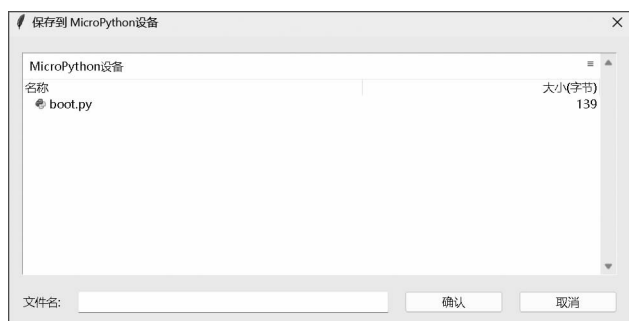


图 2-24 将代码保存到 MicroPython 设备

如果在代码输入框里输入 main.py,代码便保存在 ESP32 开发板里。按一下开发板上的重启按钮后,发现板载的 LED 已经开始闪烁了。此外,开发板重新通电后可自动运行上

面的代码。

要想 ESP32 开发板工作,可以在“Shell”窗口中输入代码,或者运行保存在本地的代码。在“Shell”窗口里有代码的情况下,单击工具栏中的运行图标,则 ESP32 开发板直接执行代码。但注意在这种情况下,编译后的代码并没有保存在开发板中,重启开发板后,开发板恢复原来的状态。

2.6 用 VSCode+PlatformIO 搭建 ESP32 开发环境

Visual Studio Code (VSCode) 是一款免费、开源的代码编辑器,由 Microsoft 开发,是一个可运行于 macOS、Windows 和 Linux 系统中的,针对编写现代 Web 和云应用的跨平台源代码编辑器。它也可在桌面上运行。它支持多种编程语言的代码高亮、智能提示、代码调试等功能,非常适用于嵌入式开发。VSCode 支持众多的插件,深受开发者喜爱。通过安装相应的插件及配置,VSCode 可变成功能强大的 IDE 工具,如成为常见的 C、C++、JAVA、Python、PHP 等语言的开发工具。经实验验证,不配置 Python 环境并不影响 ESP32 开发板的应用。

PlatformIO 是一个用于物联网开发的开源生态系统,它提供跨平台的开发环境和统一的调试器,还支持远程单元测试和固件更新,曾经获得过 IoT 最佳开发软件和工具奖,支持 1 000 多种开发板以及 50 多个开发平台,还有 12 000 个库,使开发者不必浪费时间在环境搭建上,而是拥有更多时间去编写自己的程序。安装 VSCode 是开始使用 PlatformIO 的第一步。下面简要介绍利用 VSCode 和 PlatformIO 搭建 ESP32 开发环境的步骤。

(1)访问 VSCode 官网,选择相应平台的绿色版本,解压缩后直接运行。在应用商店中搜索中文插件并安装,可以将界面转为中文,如图 2-25 所示。

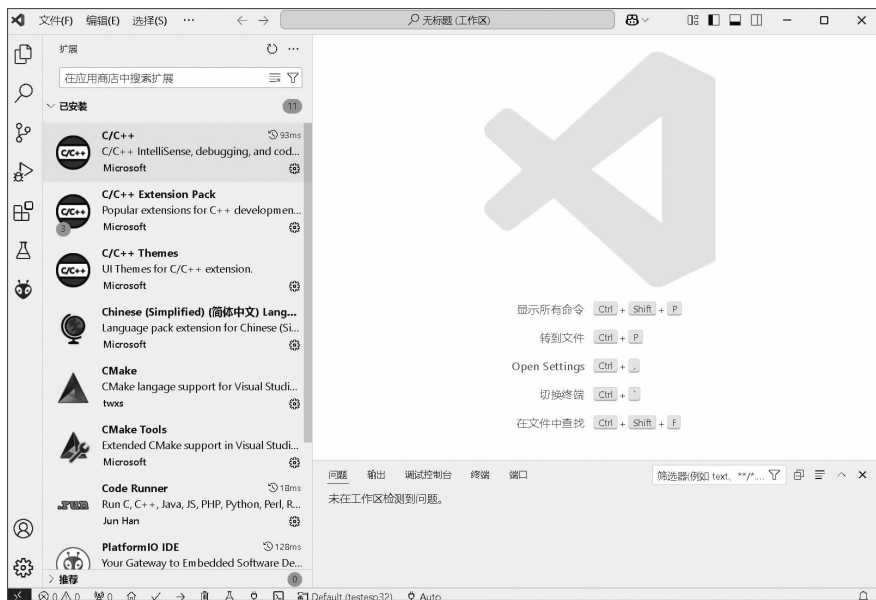


图 2-25 VSCode 的中文界面

(2) 启动 VSCode, 在左边的扩展搜索框里直接输入“PlatformIO”, 在弹出的列表中选择“PlatformIO IDE”, 单击安装即可。安装过程中如果需要安装其他扩展插件, 系统会默认选择安装。安装过程不会很久, 等系统提示安装完成后, 重启 VScode。

(3) 重启 VSCode 后, 发现界面最左边一侧出现了  图标, 单击  图标, 展开“PIO Home”选项, 选择“Open”, 然后单击“+New Project”, 弹出“Project Wizard”对话框, 填写项目名称等信息, 这里以新建一个 ESP32TEST 工程为例。在 Board 文本框中输入“esp32”, 在弹出的下拉列表中选择“Espressif ESP32 Dev Module”选项; 在“Framework”下拉列表中选择“Arduino”选项, 单击“Finish”按钮, 如图 2-26 所示。

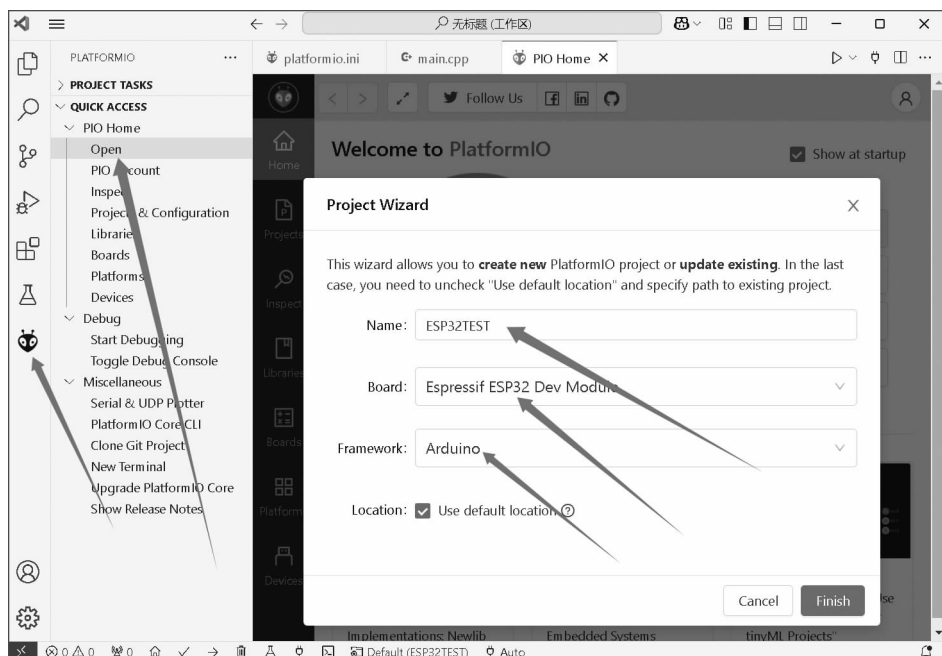


图 2-26 新建 PIO 项目

这时系统会下载开发环境所需要的所有文件、编译工具等, 下载时间会比较长, 因此应找一个网络好的环境下下载。下载完成后, 界面如图 2-27 所示, 就自动打开了 platformio. ini 文件。

(4) 将 ESP32 开发板连接到计算机。在设备管理器中查看使用的端口, 然后在 platformio. ini 文件最下端添加如下代码。

```
[env:esp32dev]
platform = espressif32
board = esp32dev
framework = arduino
upload_port = COM4
monitor_speed = 9600
```

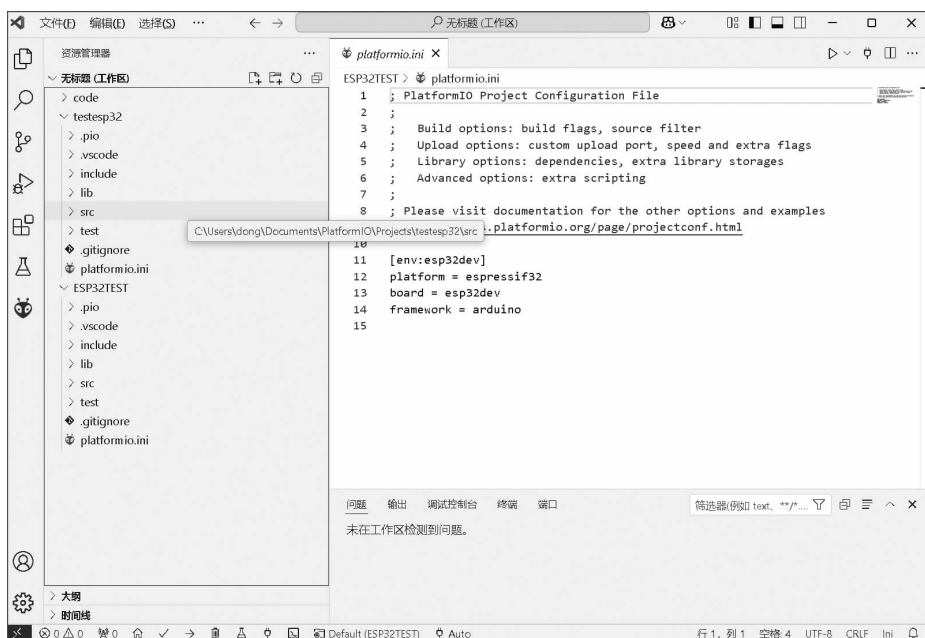


图 2-27 打开 platformio.ini 文件

双击打开 main.cpp 函数,将原来的内容删除,增加使板载 LED 闪烁的语句,增加串口输出语句,如图 2-28 所示。

(5)单击图 2-28 左下角的右向箭头,可以直接编译上传程序。上传程序后发现板载 LED 已经闪烁,在“终端”窗口也有相应的信息输出。

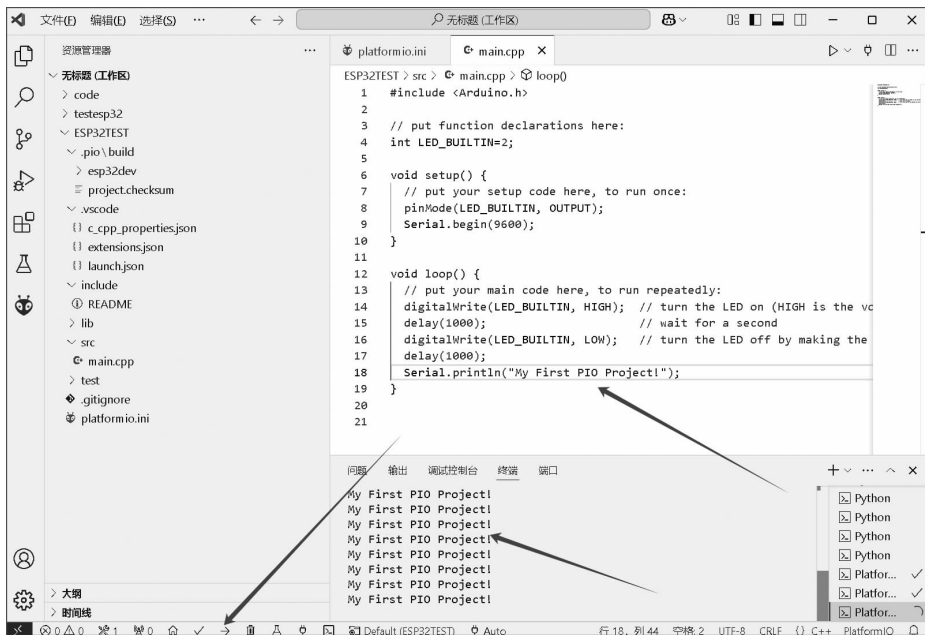


图 2-28 使 LED 闪烁程序