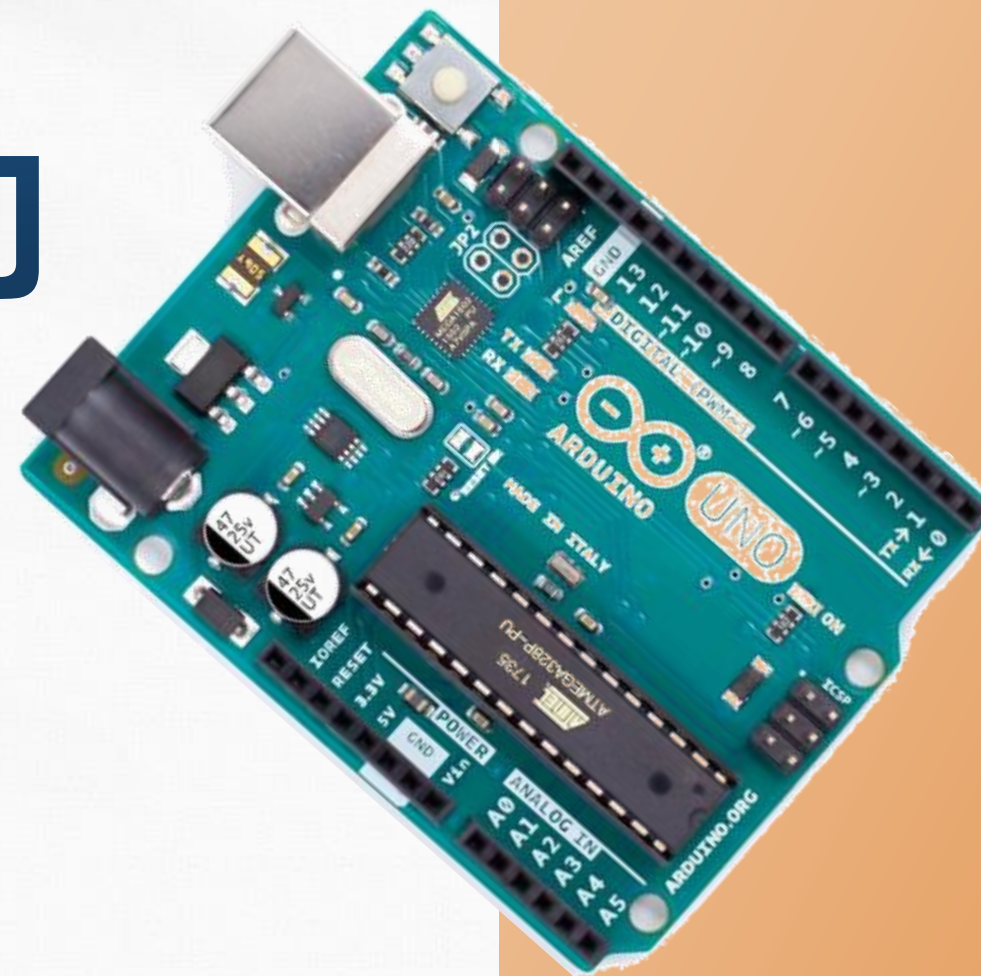


Arduino系统入门



讲师：Tim Zhao A1

目录

CONTENTS

01. Arduino简介

02. Arduino硬件系统

03. Arduino软件系统

01 PART ONE

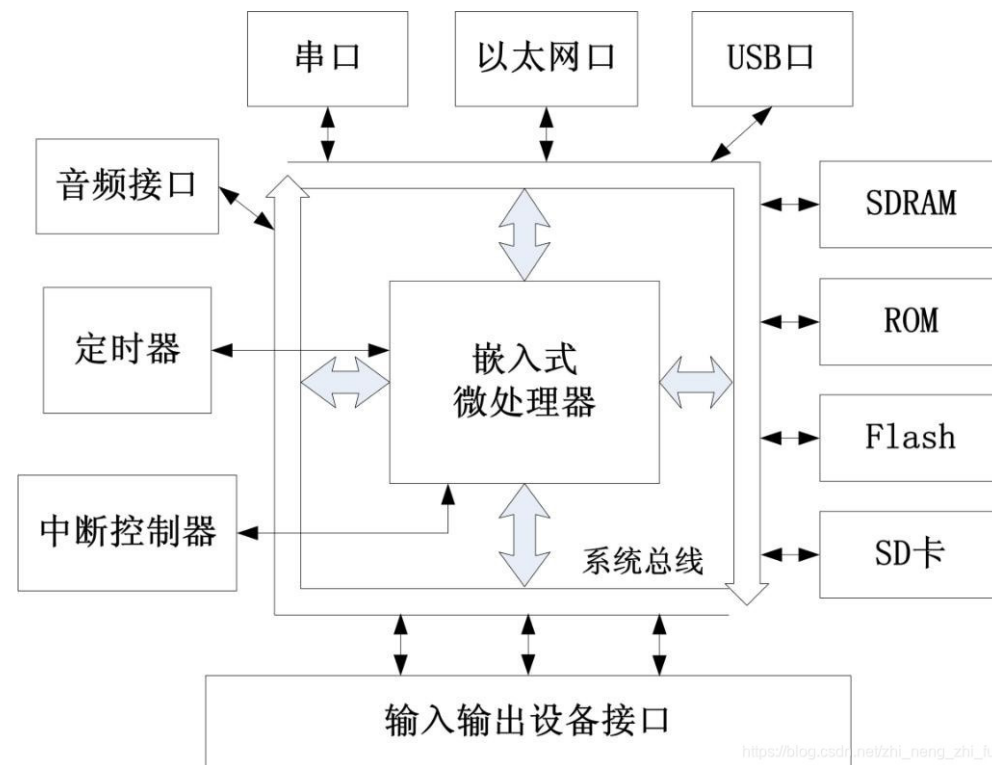
Arduino简介

什么是Arduino? --- 一种嵌入式系统



嵌入式系统:

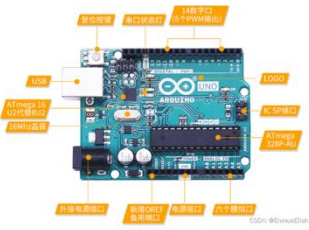
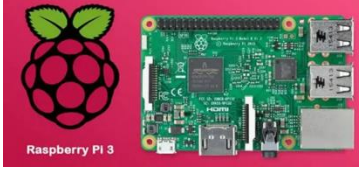

- 相比个人电脑，嵌入式系统是**专用的计算机系统**，它的硬件组成应包含计算机的五大组成部分，但大多数嵌入式系统是**面向某一种特殊应用的**，每一种应用都有独特的要求，因此不同的嵌入式系统的硬件构成在具体应用中千变万化，但从整体来看，都大致可分为如下几部分：**微处理器、存储器、输入输出设备和通讯与扩展接口**
- **嵌入式微处理器是嵌入式系统的核心部件**，它担负着控制、协调系统工作的重要任务，其功能的强弱直接决定了嵌入式应用的适用范围和开发复杂度。嵌入式微处理器通过数据线、地址线和控制线与存储器和各种外设相连
- 嵌入式系统的应用十分广泛，涉及**工业生产、日常生活、工业控制、航空航天**等多个领域



Arduino是一种简单、易用的嵌入式系统和开发平台

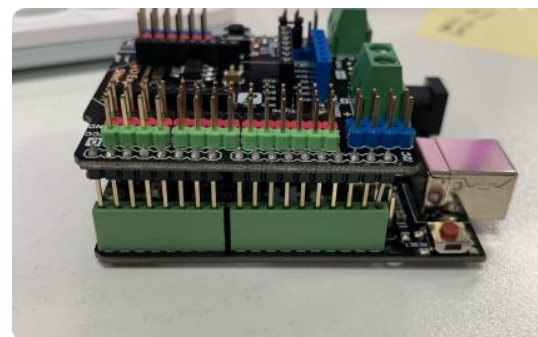
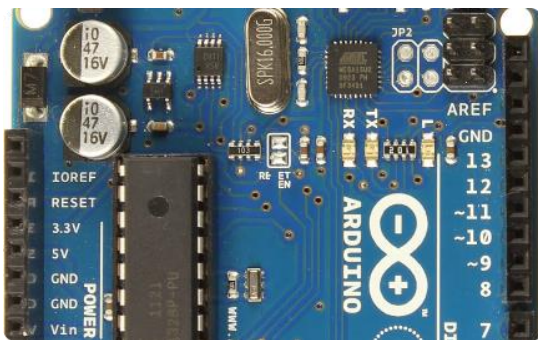


三种流行嵌入式系统和开发平台，各有优缺点

	Arduino	树莓派 Raspberry Pi	STM32
嵌入式系统			
微处理器及开发厂商	AVR架构单片机 -- Atmel公司	ARM架构单片机 -- Raspberry Pi	ARM架构单片机 --- ST（意法半导体）
特点	<ul style="list-style-type: none">• Arduino侧重于实现简单的控制和感应功能，更低功耗、更小体积和更便宜的价格• 更易于入门和学习，面向教育、工业控制等领域• 在处理能力和软件扩展方面存在局限性，也难以适应较大或复杂的系统	<ul style="list-style-type: none">• 树莓派具有高性能的CPU、更多的RAM和存储选项，适合执行更复杂的计算任务• 莓派能替代日常桌面计算机的多种用途包括文字处理、电子表格、媒体中心甚至是游戏	<ul style="list-style-type: none">• 主要用于工业自动化、机器人、智能家居等领域，具有更高调试性和实时控制性能• 在较小的系统中实现复杂的控制算法，则STM32可能更适合

Arduino的系统 and 开发平台：包括四大部分

Arduino是一款便捷灵活、方便上手的开源电子原型平台，包含硬件（各种型号的Arduino板）和软件（Arduino IDE）；其芯片基于atmel公司mega系列单片机，具有使用类似C/C++语言的IDE(集成开发环境)。可以很容易实现与Flash, processing 的互动。



硬件开发板

Arduino**系列开发板**、扩展模块、传感器等组成了Arduino的硬件部分

软件

Arduino开发环境是一款强大的**IDE**，供用户编写和上传代码到Arduino板

扩展模块

通过连接各种扩展模块，提供更多接口，可实现更多功能和应用。

驱动器+传感器

能通过各种各样的传感器来感知周围环境，开发板可以轻松获取各类传感器的

Arduino的特点与功能介绍：最流行的嵌入式学习和开发平台

开源

Arduino的硬件和软件都是开源的，可以自由获取和修改。**Arduino是全球最流行的开源硬件平台**

易于使用

Arduino提供了简单易懂的编程语言和开发环境，适合初学者入门。

丰富的扩展性

Arduino支持各种扩展模块和传感器，可以实现多样化的项目应用。

广泛的应用领域

Arduino可应用于物联网、机器人、自动化控制等众多领域。

电子产品原型设计

- 快速搭建原型，验证设计方案
- 缩短产品开发周期

物联网应用

- 传感器数据采集与处理
- 设备间通信与控制

教育领域

- 学习电子原理
- 实践创新项目

Arduino的历史



Arduino的起源

- Arduino起源于2005年，意大利的一个设计学院的学生项目，由学生导师和学生和微处理器专家共同开发
- 设计者最初为了寻求一个廉价好用的微控制器开发板从而决定自己动手制作开发板，Arduino降低了电子开发的门槛，即使是从零开始的入门者也能迅速上手，制作有趣的东西，这便是开源Arduino的魅力
- 因其**开源**、**廉价**、**简单易懂**的特性迅速受到了广大电子迷的喜爱和推崇



Arduino的发展历程

- 2005年：Arduino原型诞生
- 2006年：Arduino 1.0发布
- 2007年：Arduino 2.0发布，引入USB接口
- 2009年：Arduino 3.0发布，支持SD卡
- 2010年：Arduino 4.0发布，支持以太网
- 2011年：Arduino 5.0发布，支持蓝牙和Wi-Fi
- 功能不断丰富，开发板不断丰富

PART TWO

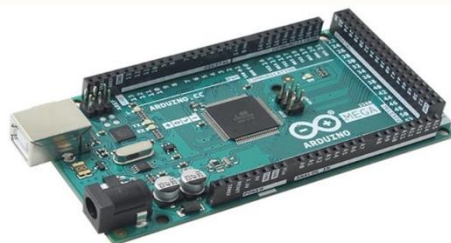
Arduino硬件系统

Arduino 四大主流硬件平台（开发板）介绍



Arduino UNO

- 最受欢迎的版本，功能齐全
- 12个数字输入/输出端口
- 6个模拟输入/输出端口
- USB接口



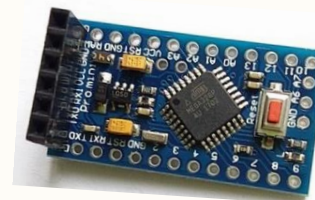
Arduino Mega

- 功能强大，适合大型项目
- 25个数字输入/输出端口
- 16个模拟输入/输出端口
- USB接口



Arduino Nano

- 体积小巧，适合空间受限的项目
- 14个数字输入/输出端口
- 6个模拟输入/输出端口
- USB接口



Arduino Pro Mini

- 低价版，适合初学者
- 8个数字输入/输出端口
- 6个模拟输入/输出端口
- 2个SPI端口
- 1个UART端口

- **Arduino UNO：最受欢迎的版本，功能齐全，价格亲民**

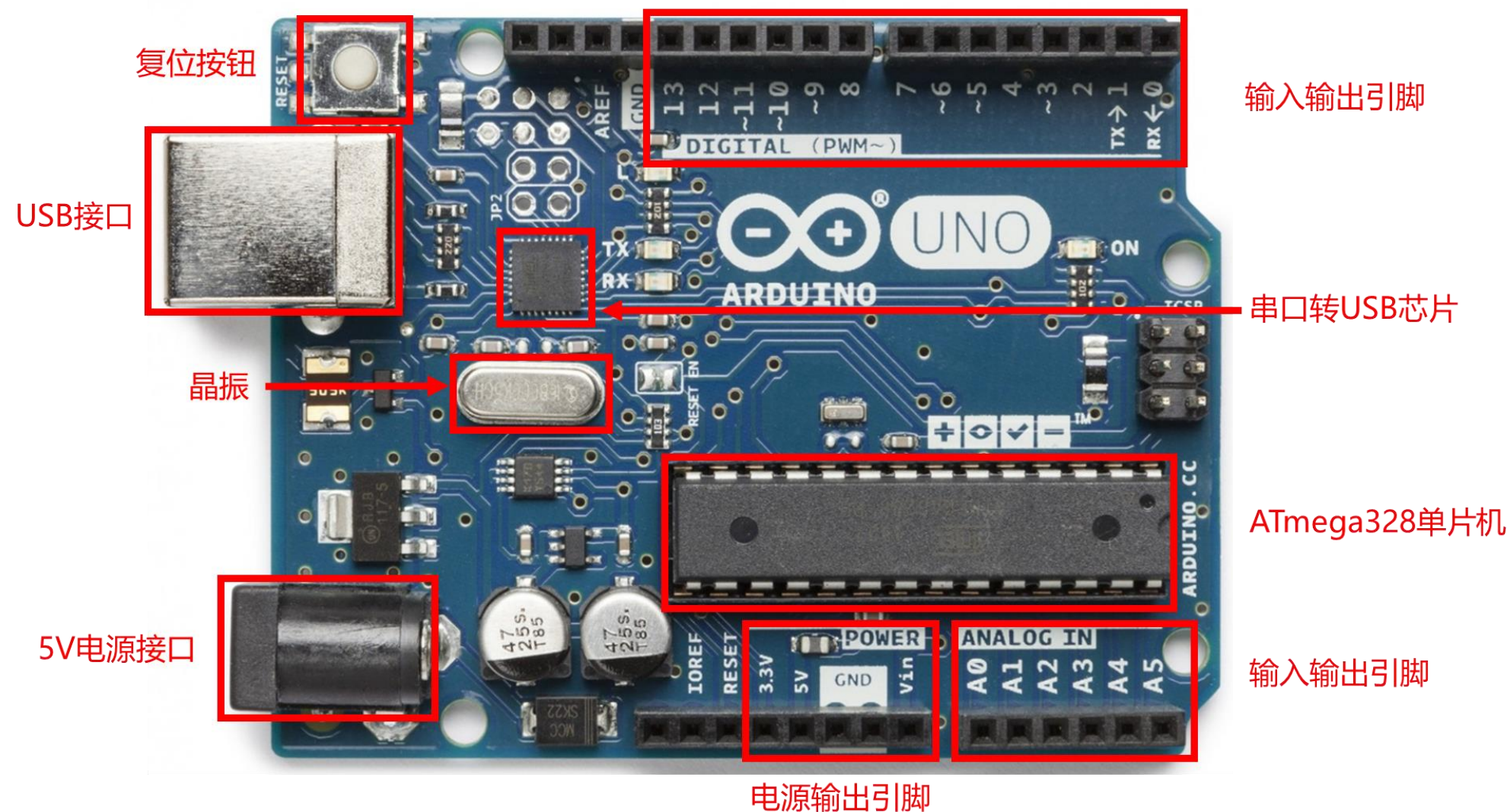
- Arduino Mega：功能强大，适合大型项目
- Arduino Nano：体积小巧，适合空间受限的项目
- Arduino Pro Mini：低价版，适合初学者

- Arduino UNO：16MHz ATmega328P微控制器
- Arduino Mega：20MHz ATmega2560P微控制器
- Arduino Nano：16MHz ATmega328P微控制器
- Arduino Pro Mini：16MHz ATmega328P微控制器

Arduino 四大主流硬件平台（开发板） 关键参数比较

Name	Processor	Operating/Input Voltage	CPU Speed	Analog In/Out	Digital IO/PWM	EEPROM [kB]	SRAM [kB]	Flash [kB]	USB	UART
Uno	ATmega328P	5 V / 7-12 V	16 MHz	6/0	14/6	1	2	32	Regular	1
Nano	ATmega168	5 V / 7-9 V	16 MHz	8/0	14/6	0.512	1	16	Mini	1
	ATmega328P					1	2	32		
Mini	ATmega328P	5 V / 7-9 V	16 MHz	8/0	14/6	1	2	32	-	-
Micro	ATmega32U4	5 V / 7-12 V	16 MHz	12/0	20/7	1	2.5	32	Micro	1
Mega 2560	ATmega2560	5 V / 7-12 V	16 MHz	16/0	54/15	4	8	256	Regular	4

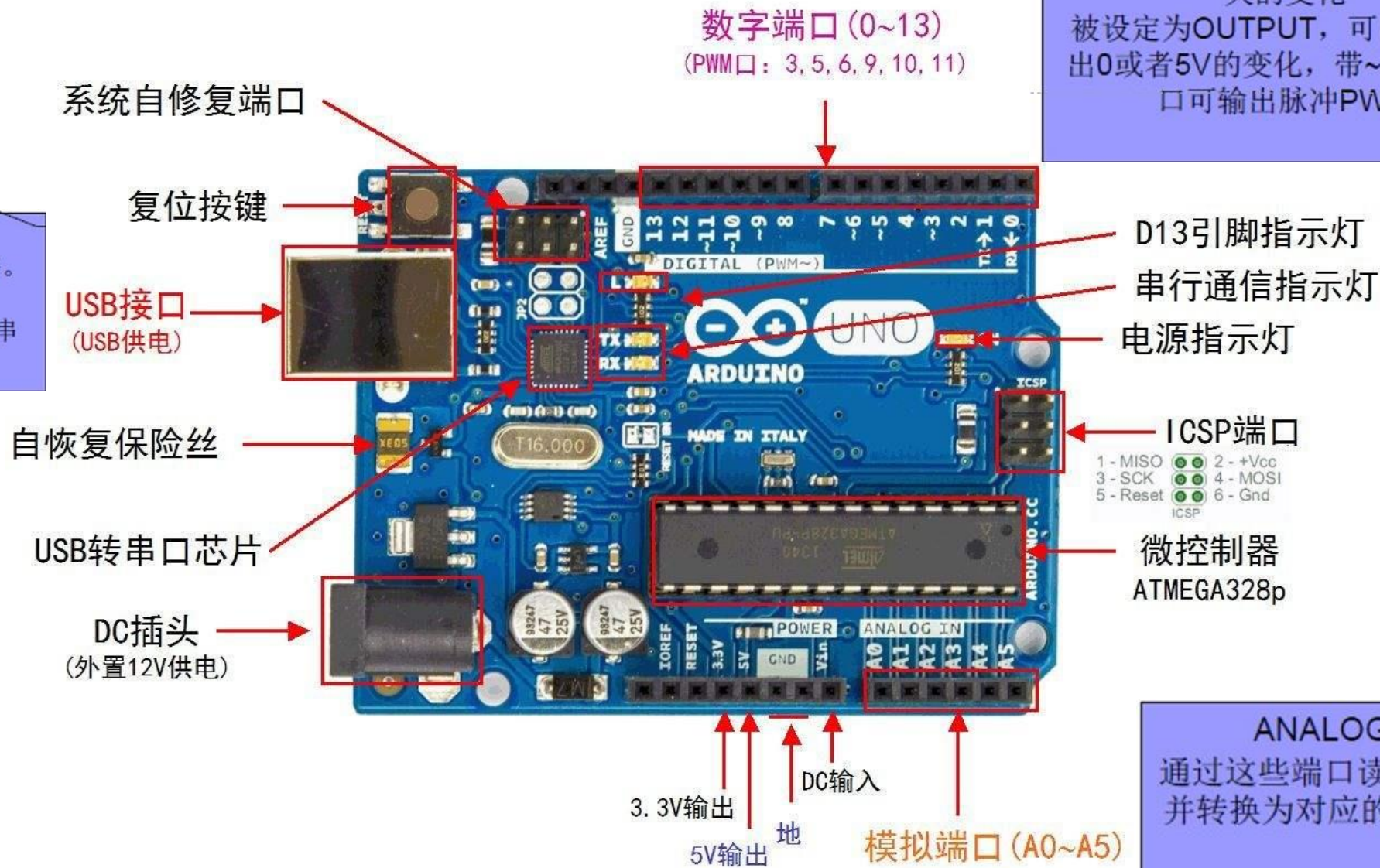
Arduino UNO 开发板的主要部件



Arduino UNO 开发板的详细部件和接口

USB
1、往arduino上写程序。
2、提供5V电源
3、提供与计算机进行串口通信的硬件连接

DIGITAL 14个端口
被设定为INPUT，可以读入外界开关的变化
被设定为OUTPUT，可以向外输出0或者5V的变化，带~标志的端口可输出脉冲PWM



1 - MISO 2 - +Vcc
3 - SCK 4 - MOSI
5 - Reset 6 - Gnd
ICSP

ANALOG IN 6个端口
通过这些端口读取外界的模拟量，
并转换为对应的0—1024的某个整数

Arduino UNO 开发板的引脚说明

Arduino Uno开发板可以使用三种方式供电：

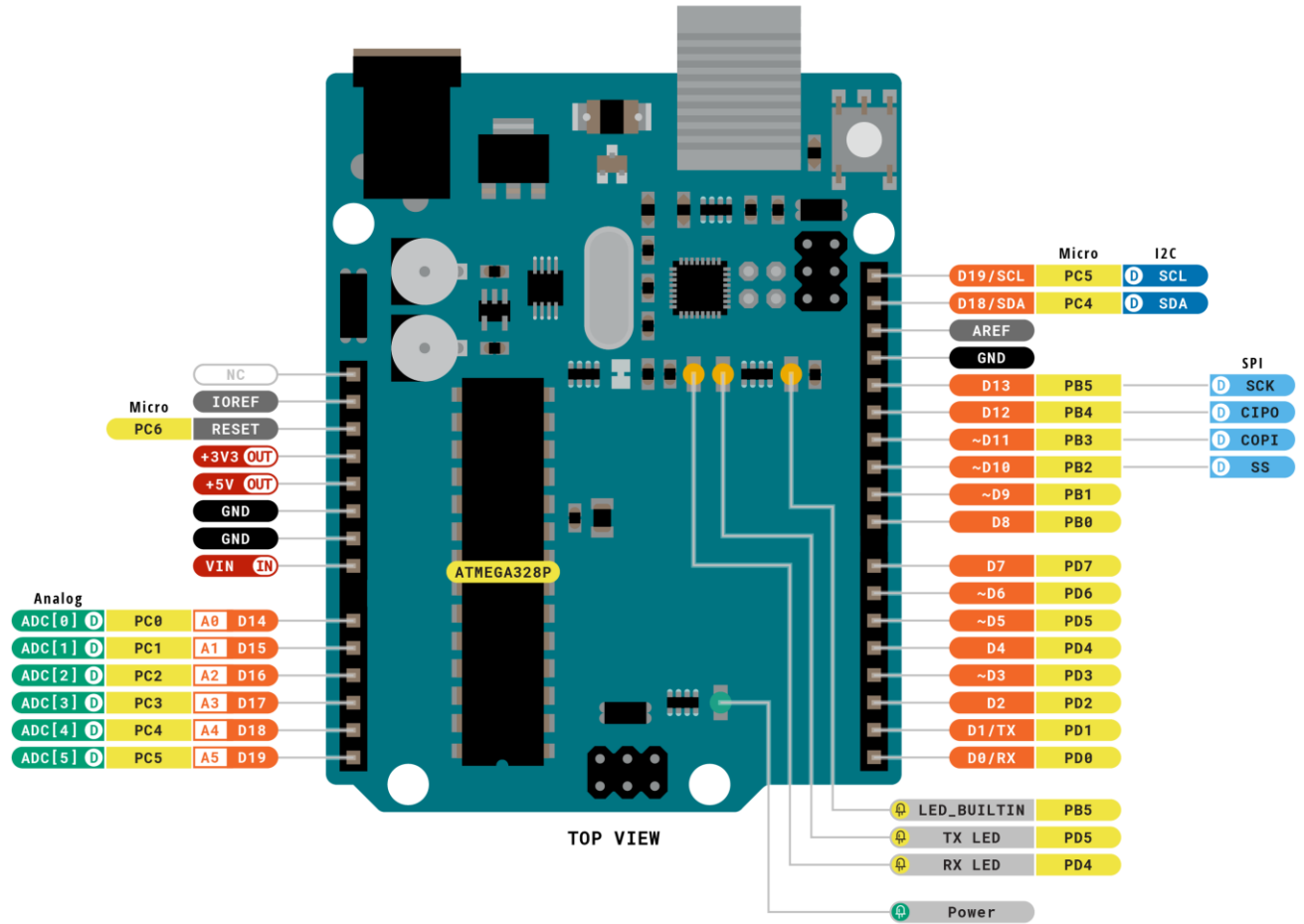
- 直流电源插孔 -可以使用电源插孔为Arduino开发板供电。电源插孔通常连接到一个适配器。开发板的供电范围可以是5-20V，但制造商建议将其保持在7-12V之间。高于12V时，稳压芯片可能会过热，低于7V可能会供电不足
- VIN引脚 - 该引脚用于使用外部电源为Arduino Uno开发板供电。电压应控制在上述提到的范围内
- USB电缆 - 连接到计算机时，提供500mA/5V电压。

Arduino Uno引脚分配 - 数字引脚

- Arduino Uno的引脚0-13用作数字输入/输出引脚。其中，引脚13连接到板载的LED指示灯；引脚3、5、6、9、10、11具有PWM功能

Arduino Uno引脚分配 - 模拟输入

- Arduino Uno有6个模拟引脚A0-A5，它们作为ADC（模数转换器）使用，这些引脚用作模拟输入，能够读取模拟电压
- ADC表示模拟到数字转换器。ADC是用于将模拟信号转换为数字信号的电子电路。模拟信号的这种数字表示允许处理器（其是数字设备）测量模拟信号并在其操作中使用它

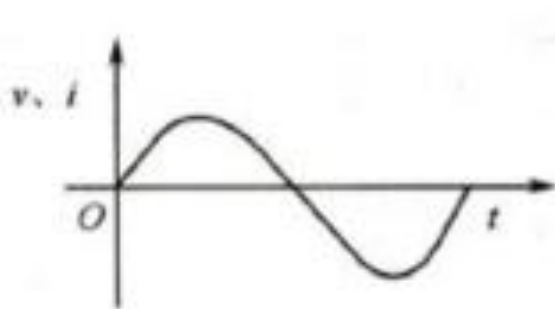


Legend:	Digital	I2C
Power	Analog	SPI
Ground	Main Part	Analog

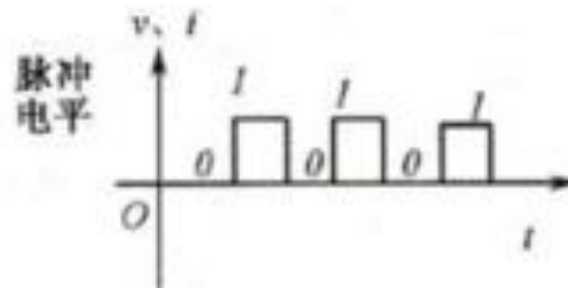
ARDUINO

ARDUINO UNO REV3
SKU code: A000066
Pinout
Last update: 6 Oct, 2022

模拟信号和数字信号



(a) 模拟信号



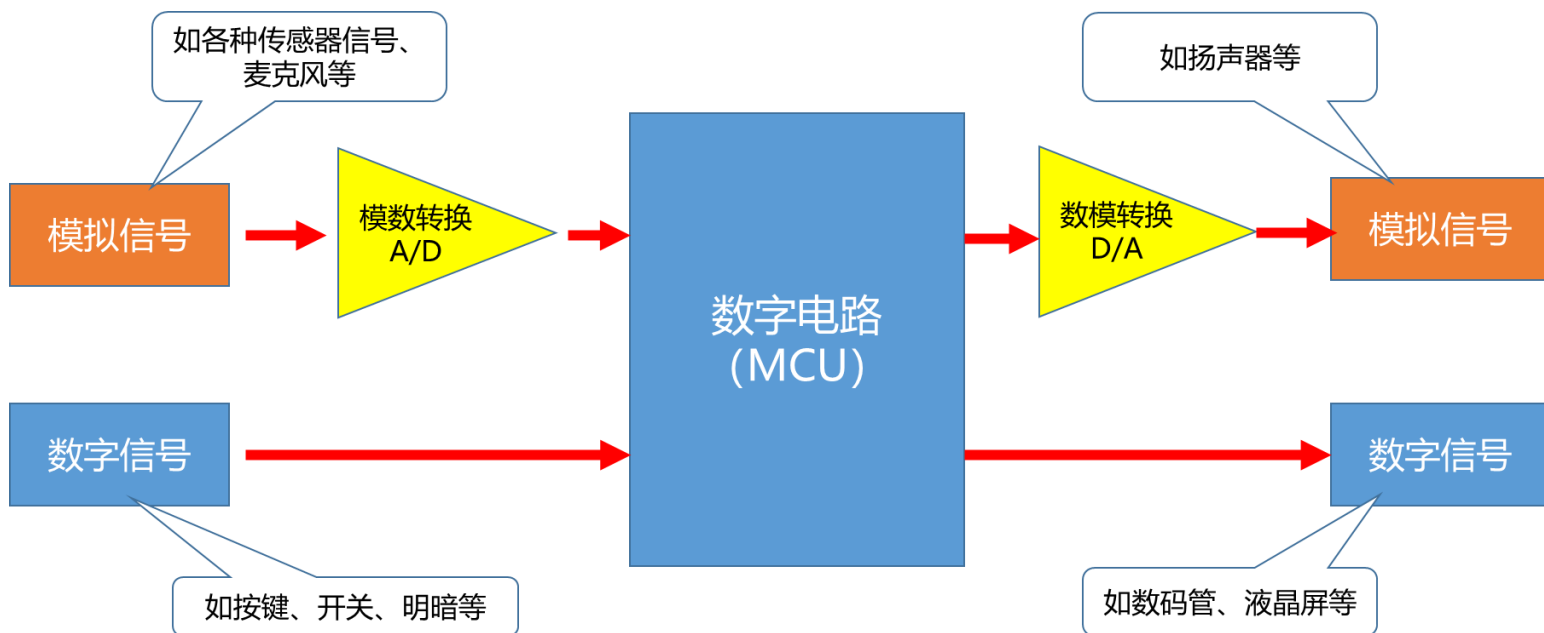
(b) 数字信号



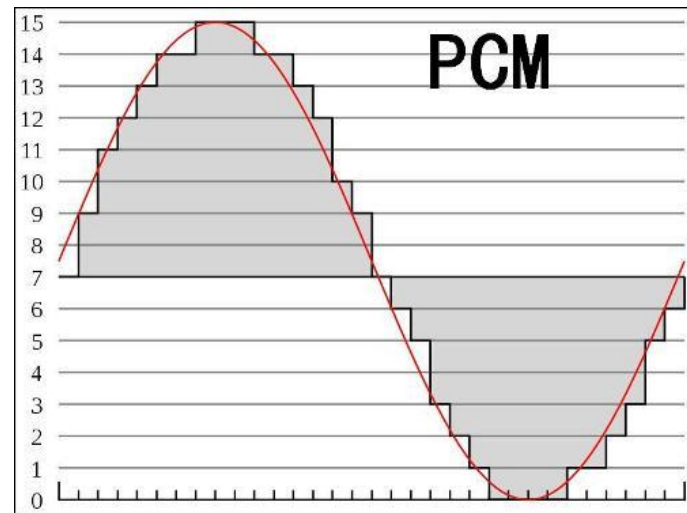
- 模拟信号：物理量的变化在时间和幅度上都是连续的信号，反映在电路上就是不用电压数值状态。它的变化是连续的，不会总从一个高电压跳崖式的陡降低电压，当然也不会陡然爬升
比如：温度、声音、速度都是模拟量
- 数字信号：物理量的变化在时间和幅值上都是离散的（不连续）的信号，反映在电路上就是高电平低电平两种状态（即只有0和1两个逻辑值）。数字信号一般通过模拟信号转换而来。
比如：灯有亮和不亮两种状态，门有开和不开两种状态。

Arduino 模拟输入：可以将外部模拟信号转换为数字信号

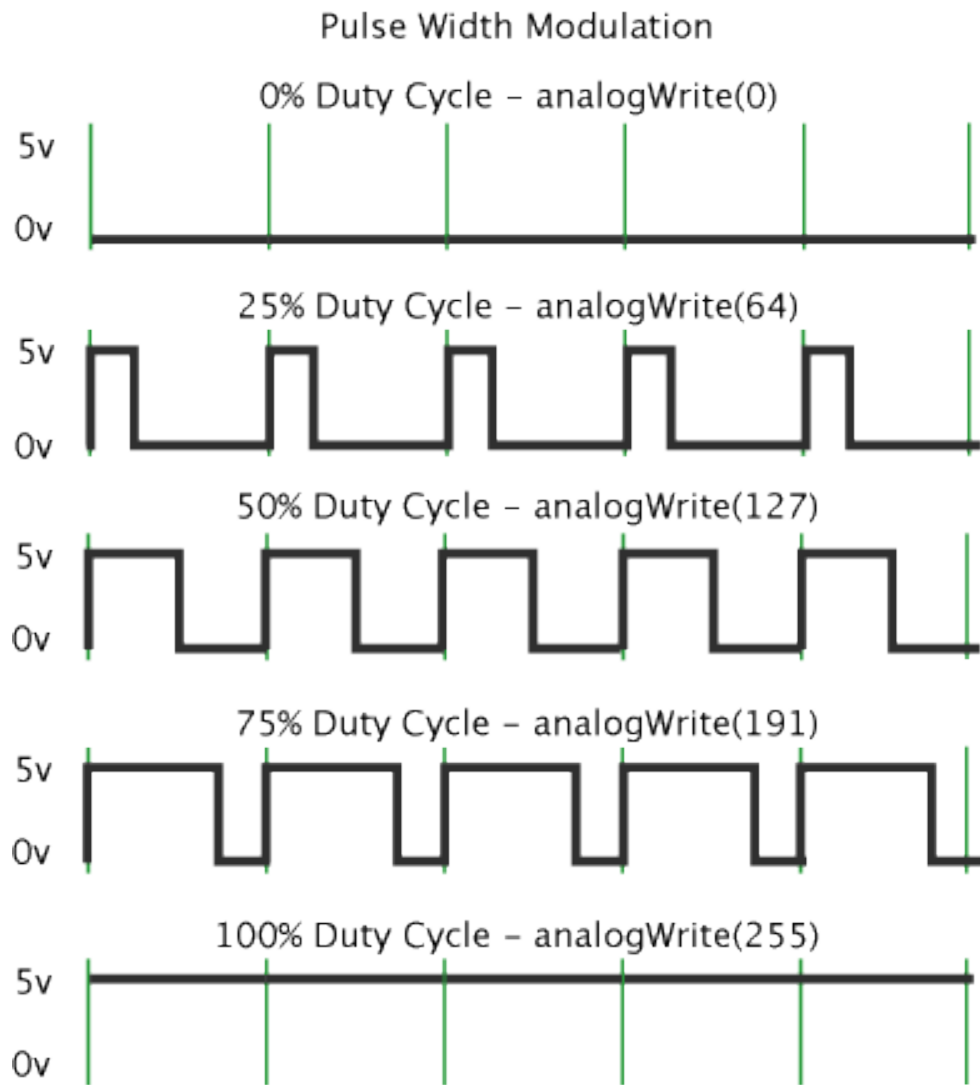
- 生活中，接触到的大多数信号都是模拟信号，如声音、温度的变化等。在模拟输入引脚是指带有ADC（Analog-to-Digital Converter，模数转换器）功能的引脚。非常常见的外设，用于将模拟信号（如电压）转换成数字信号。以便可以读取和处理
- ADC表示模拟到数字转换器。ADC是用于将模拟信号转换为数字信号的电子电路。模拟信号的这种数字表示允许处理器（其是数字设备）测量模拟信号并在其操作中使用
- 很多的传感器都是接在模拟口的，因为传感器得到的信号来自大自然，如果传感器上没有进行模数转换（所谓模数转换那就是将模拟信号转换为数字信号），我们就必须接到模拟信号口上作为输入
- 一个常见的ADC例子是IP语音（VoIP）。每部智能手机都有一个麦克风，可将声波（语音）转换为模拟电压。这通过设备的ADC，转换成数字数据，通过互联网传输到接收端



采样与量化：模数转换的核心



Arduino PWM引脚：“模拟输出”引脚，用数字信号产生输出模拟信号



面临的挑战：

- 单片机输出引脚上接一个LED，如何改变其亮度？单片机只能输出高低电平（数字信号），高电平约5V，低电平接近0V。无法输出连续变化的电压（模拟信号电压），因此只能采用变通的办法

PWM的原理：

- PWM (Pulse Width Modulation) 是一种方波控制信号。采用不同的占空比来模拟“模拟输出”。PWM由两个关键部分组成：频率和占空比。PWM频率决定了完成单个周期（周期）所需的时间以及信号从高到低的波动速度。占空比决定信号在总时间段内保持高电平的时间。占空比以百分比表示

PWM引脚的用途：

- PWM允许你通过改变信号的占空比（即高电平和低电平的时间比例）来模拟模拟信号，从而控制例如LED的亮度、舵机的位置、电机调速、风扇速度等外部设备
- 只有特定引脚支持PWM，通过指定引脚和0-255的占空比，可以实现不同强度的PWM信号，模拟电压将介于0%至100%之间
- 能够输出PWM信号的引脚叫做“模拟输出”引脚，PWM的实质并非真正的模拟信号，而是数字信号。PWM信号通常用于实现对各种模拟量（如亮度、转速）等的控制，直观上看，就像是输出了模拟信号一样

Arduino系统传感器



RGB LED传感器模块



7彩LED灯模块



继电器模块



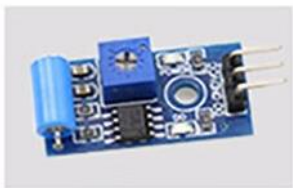
激光传感器模块



轻触开关按键模块



倾斜开关传感器



震动传感器



IR接收模块



IIC LCD 1602液晶显示模块



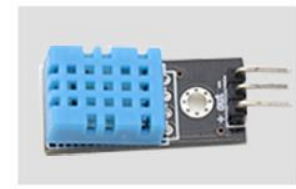
超声波传感器



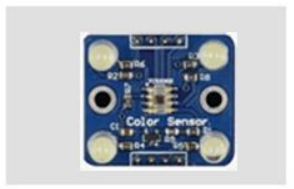
DS18B20温度传感器模块



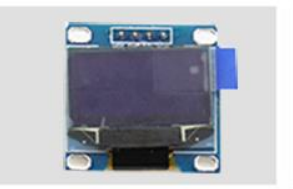
AD/DA转换PCF8591传感器



DHT11湿度传感器模块



颜色传感器模块



OLED液晶显示模块



BMP180气压传感器模块



RFID射频读卡器模块



DS1302时钟传感器模块



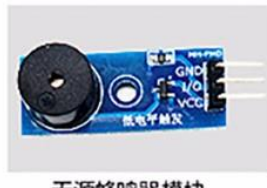
舵机模块



步进电机+电机驱动模块



有源蜂鸣器模块



无源蜂鸣器模块



干簧管传感器模块



U型光电传感器模块



360旋转传感器模块



雨滴检查传感器模块



PS操作杆传感器模块



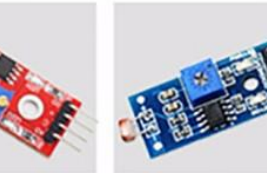
电位器传感器模块



霍尔开关传感器模块



模拟温度传感器模块



声音传感器模块



光敏电阻传感器模块



火焰传感器模块



烟雾传感器模块



红外遥控器



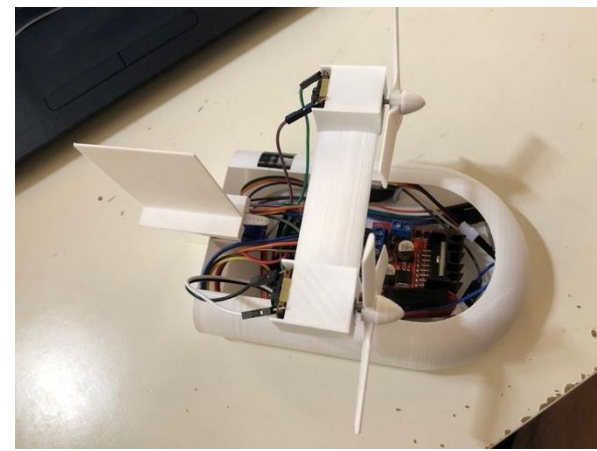
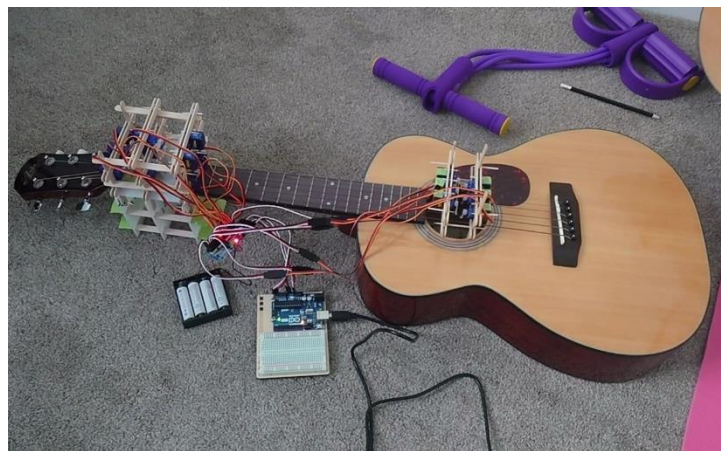
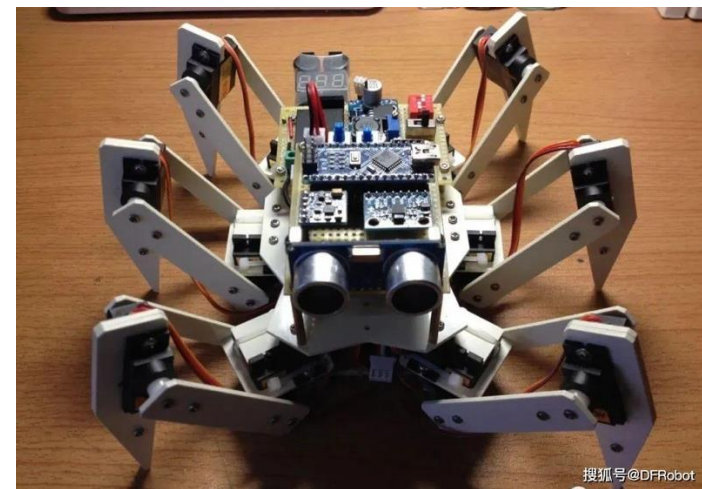
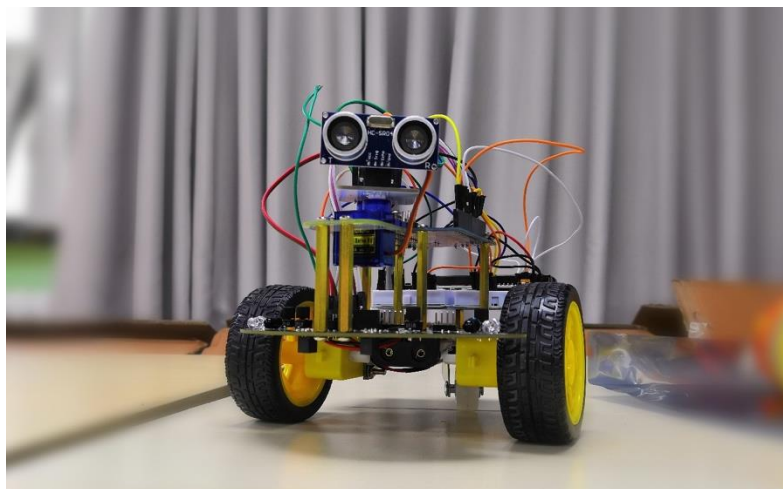
触摸传感器



红外循迹模块

Arduino生态系统中有大量的传感器和模块可供选择，比如温湿度传感器、距离传感器、光线传感器、电机驱动模块等。这些传感器和模块可以方便地连接到Arduino主板的引脚上，为项目提供丰富的功能扩展支持。

基于Arduino开发的各种有趣的小设备



A stylized black number '3' composed of four thick, curved segments arranged in a circular pattern.

PART THREE

Arduino软件系统

Arduino软件开发的两个关键

Arduino编程语言



**Arduino软件集成开发工具
IDE**

Arduino编程语言是基于C/C++的编程语言的扩展

Arduino语言

对AVR单片机和各类传感器
设置和控制的函数化和库化

C/C++

基于C/C++的编程语言

- 语法简单，易于学习
- 支持大量的库函数
- 适用于嵌入式系统开发
- Arduino编程是建立在C/C++语言基础上，利用基础的C/C++语言把AVR单片机及周边传感器相关的一些参数设置等进行函数化，不用去了解它的底层硬件，以便于开发者更加快速地使用。其实就是使用C/C++语言编写程序，只是有了更丰富的硬件相关的函数库和类库
- 学习Arduino 单片机可以完全不需要了解其内部硬件结构和寄存器设置，仅仅知道它的端口作用即可；可以不懂硬件知识，只要会简单的C语言，就可用Arduino 单片机编写程序。针对周边I/O设备的Arduino 编程，很多常用的I/O 设备都已经带有库文件或者样例程序，在此基础上进行简单的修改，即可编写出比较复杂的程序，完成功能多样化的作品。

Arduino编程语言的基本语法完全Follow C/C++

变量、数据类型与运算符与C/C++一致

变量

- 用于存储数据
- 需要声明数据类型
- 可以在程序中随时赋值

数据类型

- 整数 (int)
- 浮点数 (float)
- 字符串 (String)
- 布尔值 (boolean)

运算符

- 算术运算符 (+、-、*、/)
- 比较运算符 (==、!=、>、<)
- 逻辑运算符 (&&、||、!)

控制结构及循环语句与C/C++一致

控制结构

- 条件判断 (if-else)
- 循环结构 (while、for)

循环语句

- while循环
- for循环
- do-while循环

增加大量Arduino系统特有的硬件相关的库函数

- 数字输入/输出 (Serial、DigitalWrite、DigitalRead)
- 模拟输入/输出 (AnalogWrite、AnalogRead)
- 时间处理 (delay、millis)
- I2C通信 (Wire)，可以使用I2C接口连接数字传感器
- SPI通信 (SPI)

Arduino编程语言的程序结构（最重要的两个系统函数）

setup()

在Arduino中程序运行时将**首先调用 setup() 函数**。用于**初始化** 变量、设置针脚的输出\输入类型、配置串口、引入类库文件等 等。每次 Arduino 上电或重启后，setup 函数**只运行一次**

```
int ledPin = 13;

void setup()
{
    pinMode(LedPin, OUTPUT);
}
```

loop()

loop函数在程序运行过程中**不断的循环**。通过该函数动态控制 Arduino主控板。

Arduino所有其他代码的运行和函数调用都在Loop进行

```
void loop()
{
    digitalWrite(ledPin, HIGH);
    delay(1000);
    digitalWrite(ledPin, LOW);
    delay(1000);
}
```

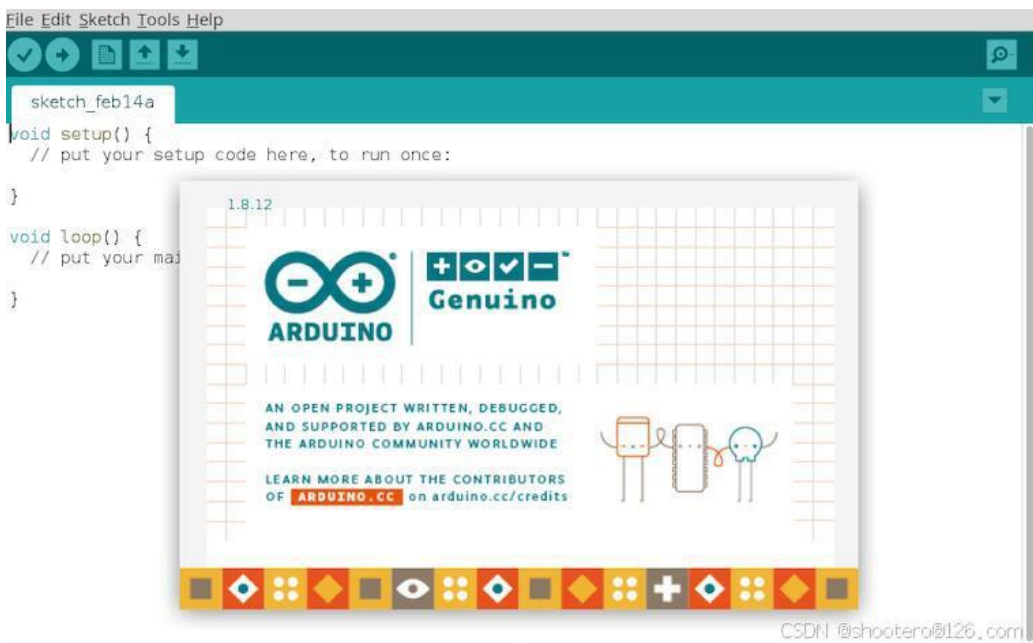
典型的Arduino Uno工作流程图，展示了从电源接通到执行用户代码的过程



Arduino IDE是专门为Arduino开发板量身定做的集成开发环境

Arduino IDE（集成开发环境）是为Arduino硬件开发而设计的一款软件，它提供了一个易于使用的图形界面，允许用户编写、编辑、编译和上传代码到Arduino开发板。Arduino IDE的目标是简化编程过程，使初学者和专业人士都能轻松地开发微控制器的应用

IDE兼容于Windows、Mac OS X和Linux等平台



从官方网站 (www.arduino.cc) 下载页面可下载最新版本

Arduino IDE的主要功能

编写和编译代码

- 打开Arduino IDE，创建新的项目
- 使用Arduino语言编写代码
- 保存项目文件
- 编译代码

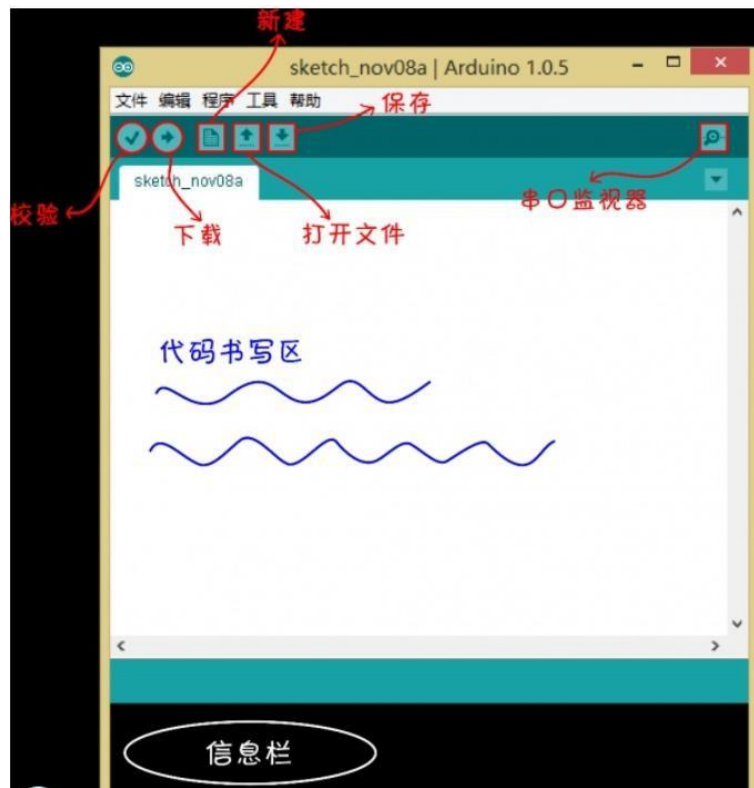
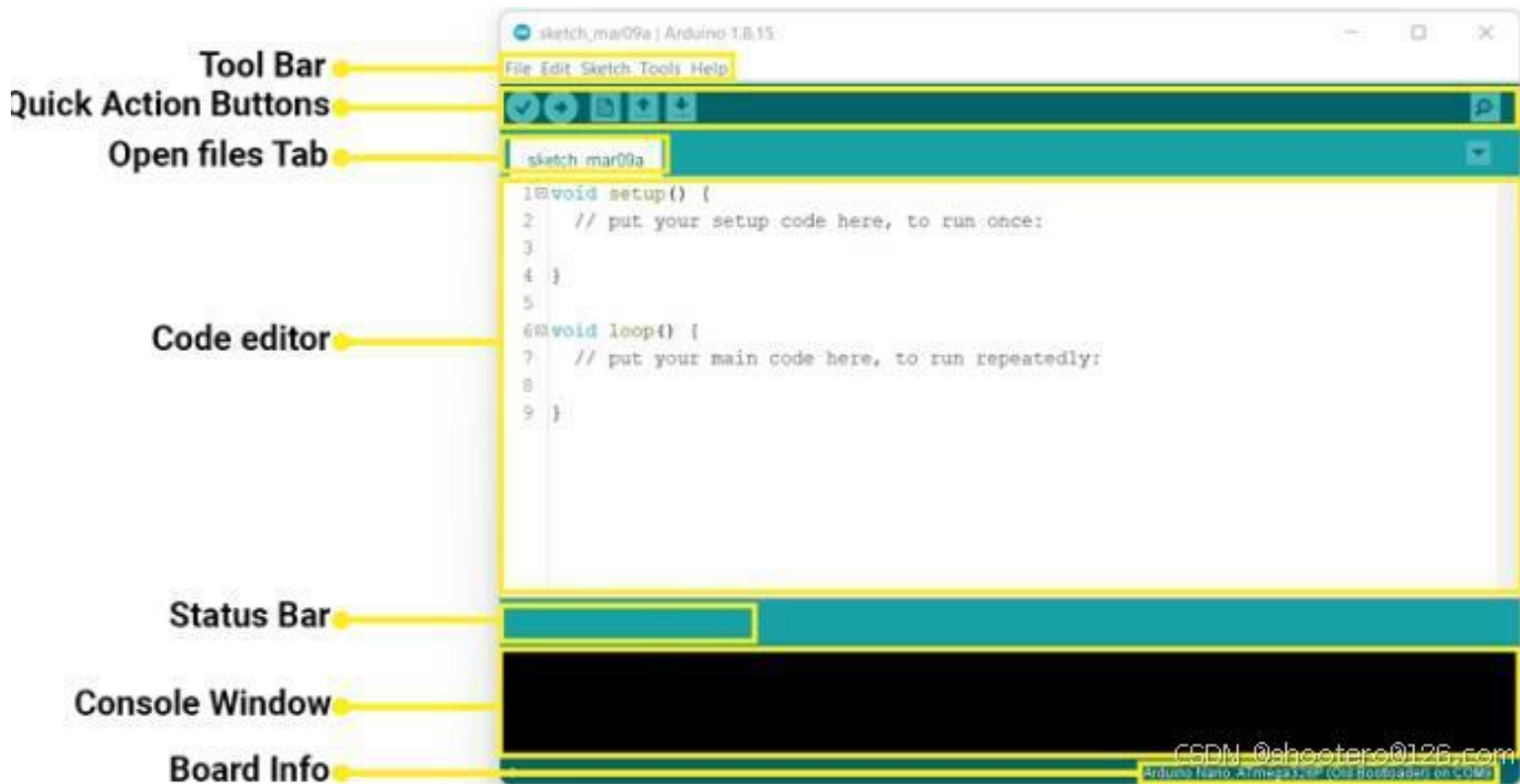
上传代码

- 将Arduino硬件连接到计算机
- 选择正确的硬件型号和串口
- 点击“上传”按钮，将代码上传到Arduino硬件

调试代码

- 使用Arduino IDE的调试功能
- 观察模拟量和数字量的变化
- 调整代码，实现预期功能

Arduino IDE的软件界面



- **菜单栏**: 包含文件菜单、编辑菜单、程序菜单、工具菜单和帮助菜单
- **工具栏**: 包含编译、上传、新建程序(sketch)、打开程序(sketch)、保存程序(sketch)和串口监视器(Serial Monitor)
- **编辑区**: 编写程序代码区域
- **状态区**: 显示程序编译和上传等信息,如果程序出现错误会有错误提示

- **代码编辑器**: 占据IDE大部分空间, 是一个文本编辑器, 用于编写和编辑代码。编辑器支持语法高亮和自动完成功能, 使代码更易阅读和编写
- **消息面板**: 位于底部, 显示编译时的输出信息, 包括错误和警告。当上传代码时, 也会显示上传进度和结果
- **串口监视器**: 可通过工具栏中的串口监视器按钮打开, 它提供了一个界面, 用于发送和接收串行数据, 常用于调试和监控程序运行时的状态

授课结束，谢谢观看

A1 Tim

