
实验基础知识

1、Arduino 介绍

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，和一套为Arduino 板编写程序的开发环境组成。

Arduino，是一个开放源代码的单芯片微控制器，它使用了Atmel AVR单片机，采用了开放源代码的软硬件平台，建构于简易输出/输入（simple I/O）界面板，并且具有使用类似Java、C语言的Processing/Wiring开发环境。

Arduino, 源自意大利语, 译为“强壮的朋友”，该平台是由意大利米兰互动设计学院的师生设计的。

Arudino的优点：

- 跨平台

Arduino IDE可以在Windows、Macintosh OSX、Linux三大主流操作系统上运行，而其他的大多数控制器只能在Windows上开发。

- 简单清晰的开发

Arduino IDE基于processing IDE开发。对于初学者来说，极易掌握，同时有着足够的灵活性。Arduino语言基于wiring语言开发，是对 AVRGCC库的二次封装，不需要太多的单片机基础、编程基础，简单学习后，你也可以快速的进行开发。

- 开放性

Arduino的硬件原理图、电路图、IDE软件及核心库文件都是开源的，在开源协议范围内里可以任意修改原始设计及相应代码。

- 社区与第三方支持

Arduino有着众多的开发者和用户，你可以找到他们提供的众多开源的示例代码、硬件设计。例如，可以在Github.com、Arduino.cc、Openjumper.com等网站找到Arduino第三方硬件、外设、类库等支持，更快更简单的扩展你的Arduino项目。

- 硬件开发的趋势

Arduino不仅仅是全球最流行的开源硬件，也是一个优秀的硬件开发平台，更是硬件开发的趋势。Arduino简单的开发方式使得开发者更关注创意与实现，更快的完成自己的项目开发，大大节约了学习的成本，缩短了开发的周期。

Arduino缺点：

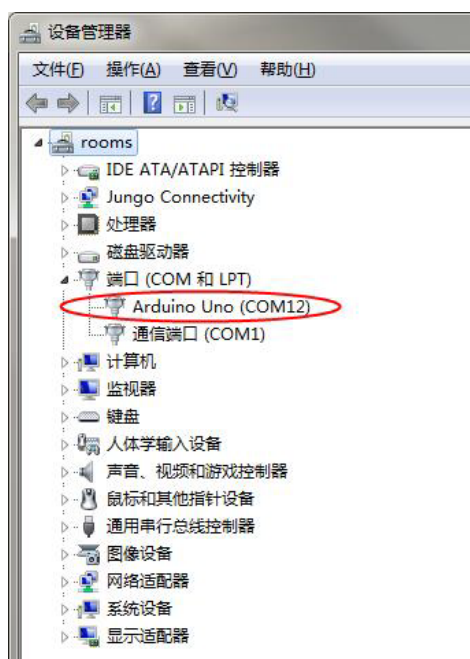
处理速度慢，实时性不高；中断2个，计数器无，时钟频率设置麻烦；PWM的频率最大是500Hz，占空比设置只能0-255，不能满足一些性能要求比较严格的控制。模拟输入的采样精度是10位，采样间隔是100us，不能满足一些采样频率高的应用；串口缓存最大的数是128，而且不能中断接收。

2、Arduino IDE 介绍

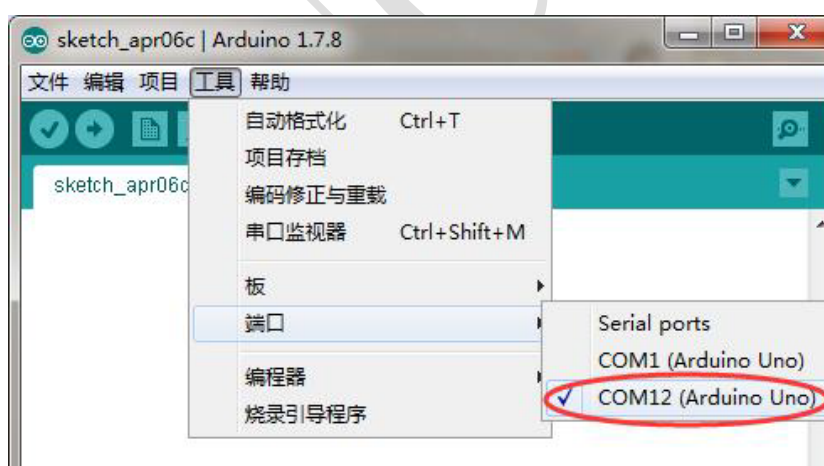
1) 双击打开安装好桌面上创建的 Arduino 软件



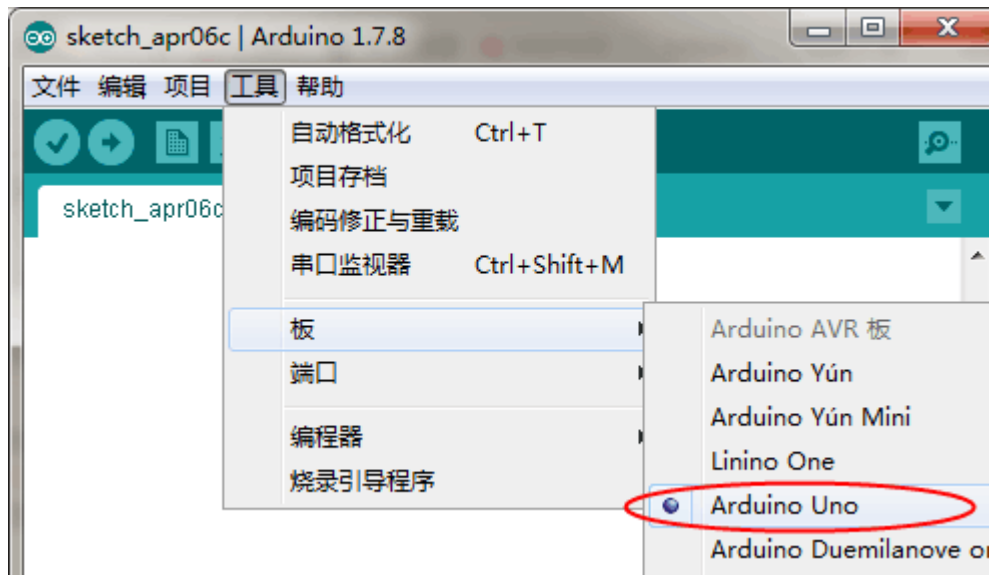
2) USB 数据线插上 UNO 板和电脑后，打开“控制面板”——“设备管理器”，查看 Arduino Uno 串口号是多少，一般就是在（）里面，图示串口号为 12



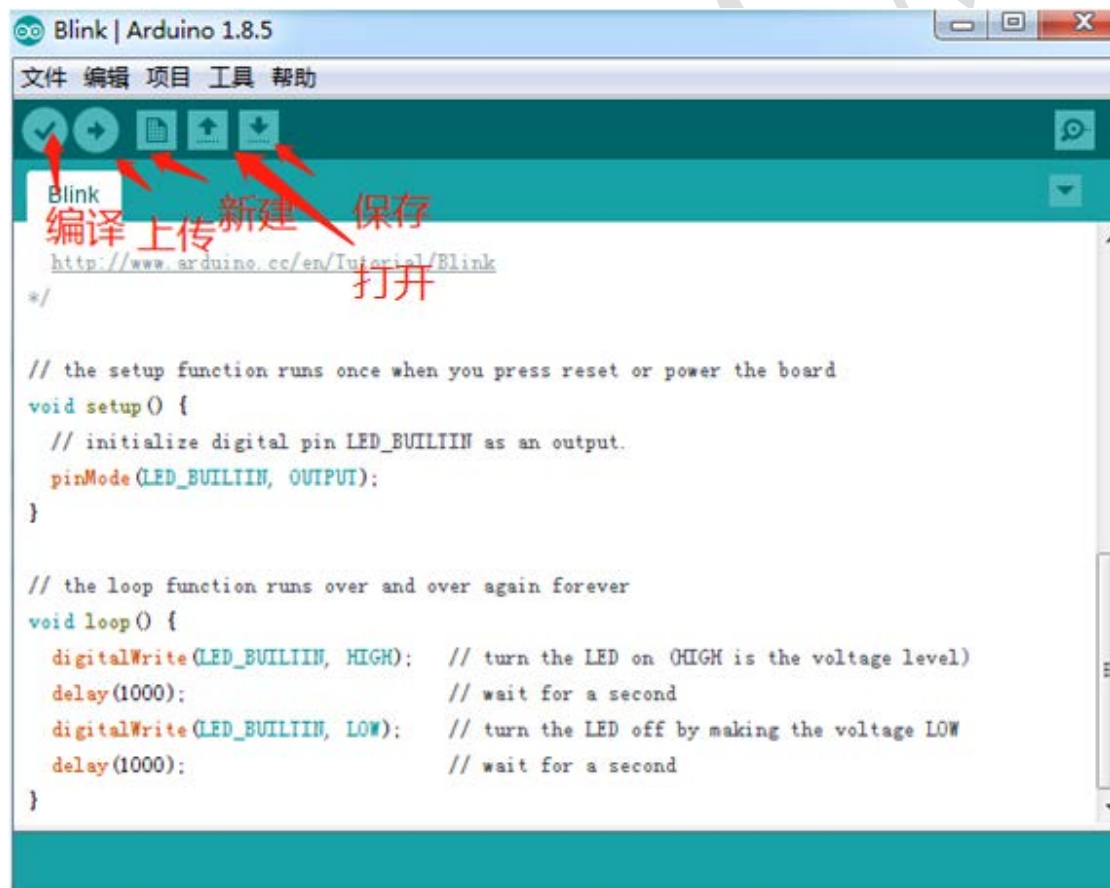
3) 点击打开 Arduino 软件的“工具”——“端口”，选择刚才设备管理器显示的串口号的那个端口



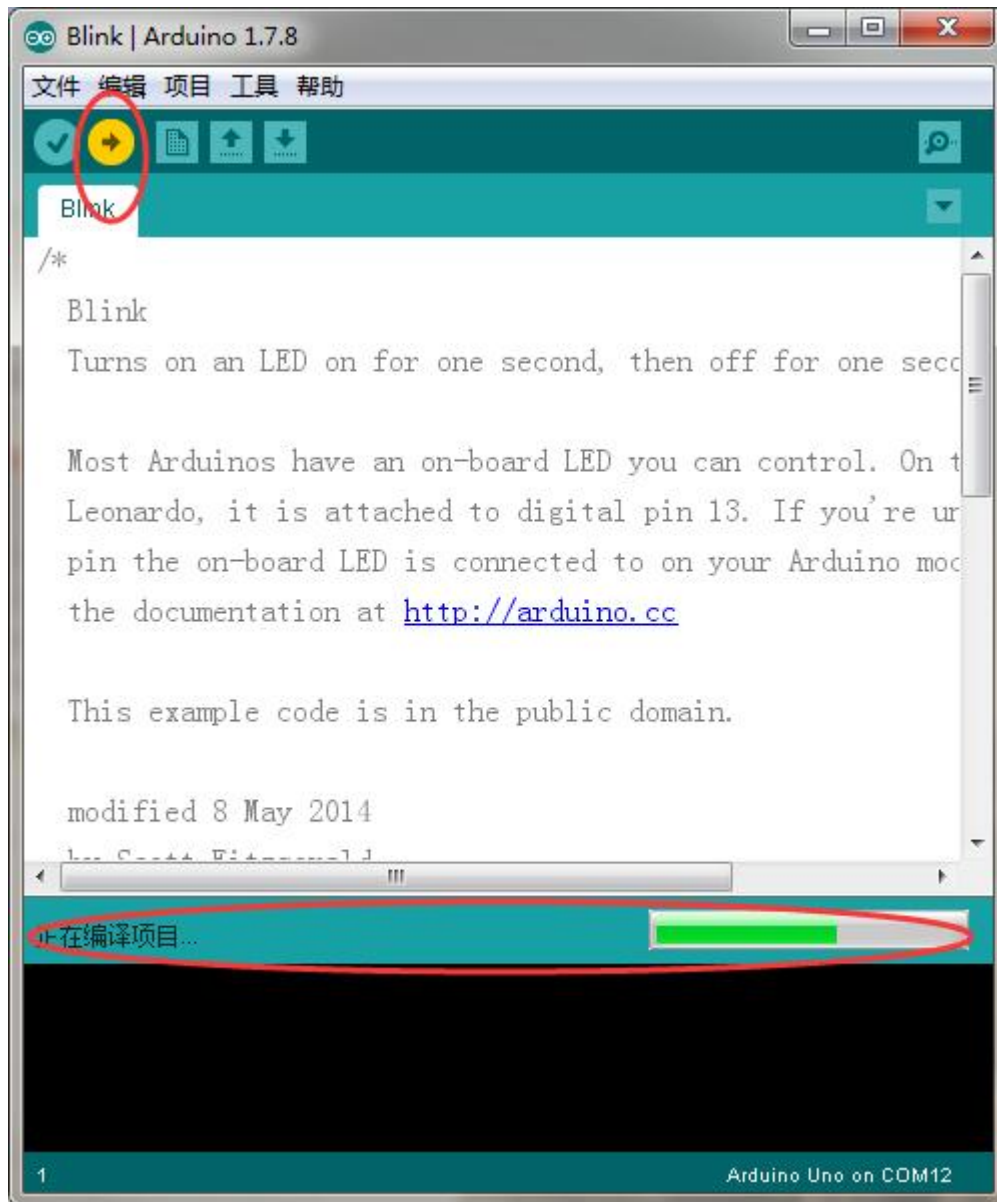
1. 点击打开 Arduino 软件的“工具”——“板”，选择对应的型号



2. 在空白区域写入程序（或直接打开.ino 后缀的文件）



3. 最后点击“上传”直至“上传成功”，则完成程序的烧录。



3、Arduino Uno 板介绍

官网: <https://store.arduino.cc/usa/arduino-uno-rev3>

Arduino Uno开发板——以**ATmega328 MCU**控制器为基础——具备14路数字输入/输出引脚(其中6路可用于PWM输出)、6路模拟输入、一个16MHz陶瓷谐振器、一个USB接口、一个电源插座、一个ICSP接头和一个复位按钮。它采用Atmega16U2芯片进行USB到串行数据的转换。

Uno PCB的最大长度和宽度分别为2.7和2.1英寸，USB连接器和电源插座超出了以前的尺寸。4个螺丝孔让电路板能够附着在表面或外壳上。请注意，数字引脚7和8之间的距离是160密耳（0.16”），不是其他引脚间距（100密耳）的偶数倍。它包含了组成微控制器的所有结构，同时，只需要一条USB数据线连接至电脑。目前，Arduino Uno已成为Arduino主推的产品。



微处理器	ATmega328P
工作电压	5V
输入电压（推荐）	7-12V
输入电压（限值）	6-20V
数字输入/输出引脚	14路（其中6路可用于PWM输出）
PWM数字I/O引脚	6
模拟输入引脚	6
每路输入/输出引脚的直流电流	20 mA
3.3V引脚的直流电流	50 mA
闪存存储器	32KB，其中引导程序占用0.5KB
SRAM	2 KB (ATmega328P)
EEPROM	1 KB (ATmega328P)
时钟频率	16 MHz
长度	68.6 mm
宽度	53.4 mm
重量	25 g

Arduino 板详细介绍

1) 电源（Power）

Arduino UNO 有三种供电方式：

- ✓ 通过 USB 接口供电，电压为 5 V；
- ✓ 通过 DC 电源输入接口供电，电压要求 7~12 V；
- ✓ 通过电源接口处 5 V 或者 VIN 端口供电，5 V 端口处供电必须为 5 V，VIN 端口处供电为 7~12 V。

2) 指示灯（LED）

Arduino UNO 带有 4 个 LED 指示灯，作用分别是：

- ✓ ON，电源指示灯。当 Arduino 通电时，ON 灯会点亮。

-
- ✓ TX，串口发送指示灯。当使用 USB 连接到计算机且 Arduino 向计算机传输数据时，TX 灯会点亮。
 - ✓ RX，串口接收指示灯。当使用 USB 连接到计算机且 Arduino 接收到计算机传来的数据时，RX 灯会点亮。
 - ✓ L13，可编程控制指示灯。该 LED 通过特殊电路连接到 Arduino 的 13 号引脚，当 13 号引脚为高电平或高阻态时，该 LED 会点亮；当为低电平时，不会点亮。因此可以通过程序或者外部输入信号来控制该 LED 的亮灭。

3) 复位按键 (Reset Button)

- ✓ 按下该按键可以使 Arduino 重新启动，从头开始运行程序。

4) 存储空间 (Memory)

Arduino 的存储空间即是其主控芯片所集成的存储空间。也可以通过使用外设芯片的方式来扩展 Arduino 的存储空间。

Arduino UNO 的存储空间分三种：

- ✓ Flash，容量为 32 KB。其中 0.5 KB 作为 BOOT 区用于储存引导程序，实现通过串口下载程序的功能；另外的 31.5 KB 作为用户储存程序的空间。相对于现在动辄几百 GB 的硬盘，可能觉得 32 KB 太小了，但是在单片机上，32 KB 已经可以存储很大的程序了。
- ✓ SRAM，容量为 2 KB。SRAM 相当于计算机的内存，当 CPU 进行运算时，需要在其中开辟一定的存储空间。当 Arduino 断电或复位后，其中的数据都会丢失。 EEPROM，容量为 1 KB。

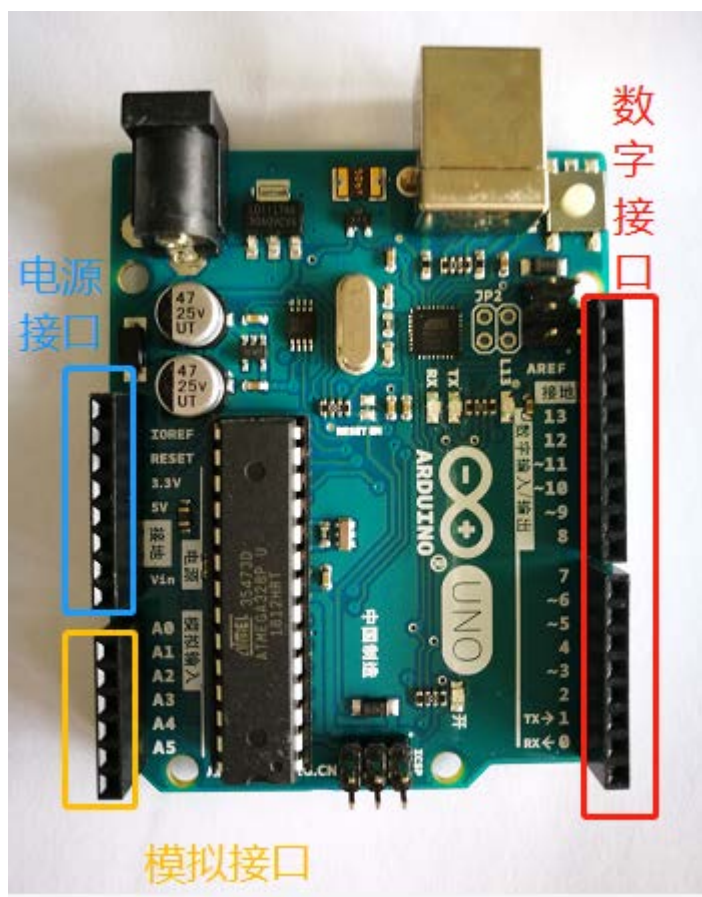
-
- ✓ EEPROM 的全称为电可擦写的可编程只读存储器，是一种用户可更改的只读存储器，其特点是在 Arduino 断电或复位后，其中的数据不会丢失。

5) 输入/ 输出端口 (Input/Output Port)

Arduino UNO 有 14 个数字输入/输出端口，6 个模拟输入端口。

其中一些带有特殊功能，这些端口如下：

- ✓ UART 通信，为 0 (RX) 和 1 (TX) 引脚，被用于接收和发送串口数据。这两个引脚通过连接到 ATmega16 U2 来与计算机进行串口通信。外部中断，为 2 和 3 引脚，可以输入外部中断信号。
- ✓ PWM 输出，为 3、5、6、9、10 和 11 引脚，可用于输出 PWM 波。SPI 通信，为 10 (SS)、11 (MOSI)、12 (MISO) 和 13 (SCK) 引脚，可用于 SPI 通信。
- ✓ TWI 通信，为 A4 (SDA)、A5 (SCL) 引脚和 TWI 接口，可用于 TWI 通信，兼容 IIC 通信。
- ✓ AREF，模拟输入参考电压的输入端口。
- ✓ IOREF，用于使盾板适配主板提供不同电压。因为有些主板提供 3.3V 电压而有些提供 5V 电压；
- ✓ Reset，复位端口。接低电平会使 Arduino 复位。当复位键被按下时，会使该端口接到低电平，从而使 Arduino 复位。



1) 数字 I/O 口

数字 I/O 口可以输入和输出数字信号。数字信号只有两种形态，高电平和低电平。高低电平是通过一个参考电压（AREF）确定的，高于 AREF 的电平即被认为是高电平，低于 AREF 的电平即被认为是低电平。Arduino 默认的参考电压大约是 1.1V，可以通过 AREF 端口设置外部参考电压。

接口 0 和 1 还被复用为 RX 和 TX 接口，它们可以用来传输数据，例如两个 Arduino 之间通信。

每个数字端口可以提供最高 40mA 电流和 5V 电压，这足够用来点亮一个 LED，但是不足以驱动电动机。因此，在使用过程中一定要注意它们的极限电压和电流。

在数字 I/O 口中，有一部分（针脚编号带有~）具有 PWM 输出能力。PWM 的中文译名是脉冲宽度调制，它是利用微处理器的数字输出来控制模拟电路的一种技术。使用 PWM 的最简单的例子就是控制 LED 的亮度，这在随后就会展示给大家。

2) 模拟 I/O 口

模拟 I/O 口可以输入模拟信号和数字信号，但是不能输出模拟信号。它可以测量连接在它上面的电压以供程序使用。模拟信号就是像每天的温度变化这样连续变化的信号，它随时可以变到任意的值。因此通过温度传感器和模拟 I/O 口，就可以检测温度。

3) 电源接口

电源端口部分有多个不同名字的接口，它们的功能介绍如下：

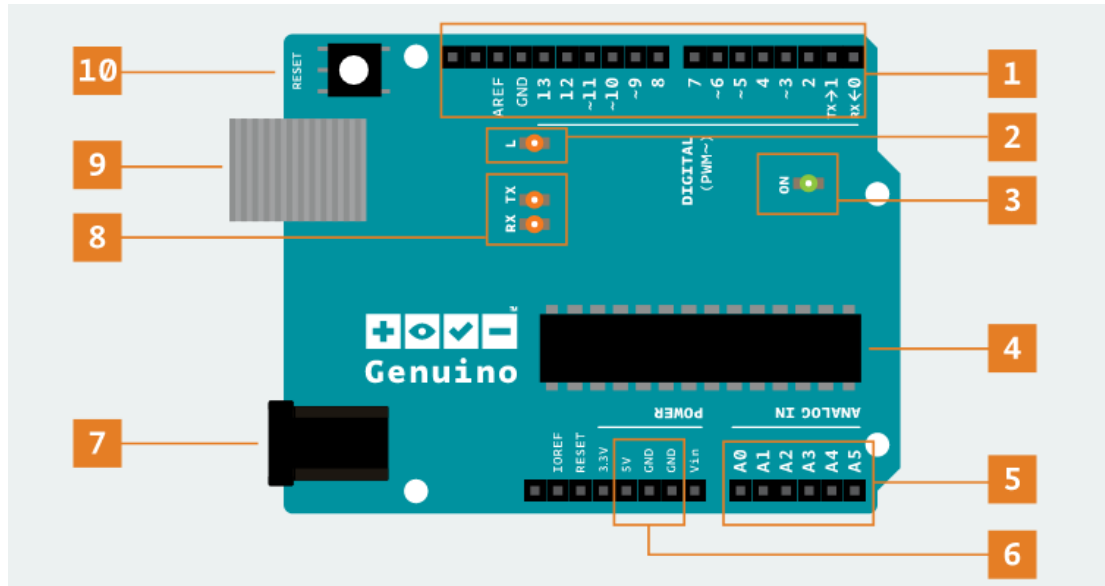
IOREF: 用于使盾板适配主板提供不同电压。因为有些主板提供 3.3V 电压而有些提供 5V 电压；

RESET: 复位端口，用来复位主板，功能与复位按钮相同；

3.3V 和 5V: 两种规格的电压输出；

两个 GND: 输入和输出电压的接地；

Vin: 外部电压输入端口，连接到这个端口的电源需要稳压，否则非常容易损坏板子。



- 1. Digital pins Use these pins with `digitalRead()`, `digitalWrite()`, and `analogWrite()`. `analogWrite()` works only on the pins with the PWM symbol.
- 2. Pin 13 LED The only actuator built-in to your board. Besides being a handy target for your first blink sketch, this LED is very useful for debugging.
- 3. Power LED Indicates that your Genuino is receiving power. Useful for debugging.
- 4. ATmega microcontroller The heart of your board.
- 5. Analog in Use these pins with `analogRead()`.
- 6. GND and 5V pins Use these pins to provide +5V power and ground to your circuits.
- 7. Power connector This is how you power your Genuino when it's not plugged into a USB port for power. Can accept voltages between 7-12V.
- 8. TX and RX LEDs These LEDs indicate communication between your Genuino and your computer. Expect them to flicker rapidly during sketch upload as well as during serial communication. Useful for debugging.
- 9. USB port Used for powering your Genuino Uno, uploading your sketches to your Genuino, and for communicating with your Genuino sketch (via `Serial.println()` etc.).
- 10. Reset button Resets the ATmega microcontroller.

4、Arduino 语言及函数介绍

常量：

- HIGH | LOW 表示数字 IO 口的电平 ,HIGH 表示高电平(1) ,
LOW 表示低电平 (0) 。

-
- INPUT | OUTPUT 表示数字 IO 口的方向，INPUT 表示输入（高阻态），OUTPUT 表示输出（AVR 能提供 5V 电压 40mA 电流）。
 - true | false true 表示真（1），false 表示假（0）。

结构

- void setup() 初始化变量，管脚模式，调用库函数等
- void loop() 连续执行函数内的语句

数字 I/O

- pinMode(pin, mode) 数字 IO 口输入输出模式定义函数，pin 表示为 0 ~ 13，mode 表示为 INPUT 或 OUTPUT。
- digitalWrite(pin, value) 数字 IO 口输出电平定义函数，pin 表示为 0 ~ 13，value 表示为 HIGH 或 LOW。比如定义 HIGH 可以驱动 LED。
- int digitalRead(pin) 数字 IO 口读输入电平函数，pin 表示为 0 ~ 13，value 表示为 HIGH 或 LOW。比如可以读数字传感器。

模拟 I/O

- int analogRead(pin) 模拟 IO 口读函数，pin 表示为 0 ~ 5（Arduino Diecimila 为 0 ~ 5，Arduino nano 为 0 ~ 7）。比如可以读模拟传感器（10 位 AD，0 ~ 5V 表示为 0 ~ 1023）。
- analogWrite(pin, value) - *PWM* 数字 IO 口 PWM 输出函数，Arduino 数字 IO 口标注了 PWM 的 IO 口可使用该函数，

pin 表示 3, 5, 6, 9, 10, 11 , value 表示为 0 ~ 255。比如可用于电机 PWM 调速或音乐播放。

扩展 I/O

- shiftOut(dataPin, clockPin, bitOrder, value) SPI 外部 IO 扩展函数 ,通常使用带 SPI 接口的 74HC595 做 8 个 IO 扩展 , dataPin 为数据口 ,clockPin 为时钟口 ,bitOrder 为数据传输方向 (**MSBFIRST** 高位在前 , **LSBFIRST** 低位在前) , value 表示所要传送的数据 (0 ~ 255) , 另外还需要一个 IO 口做 74HC595 的使能控制。
- unsigned long pulseIn(pin, value) 脉冲长度记录函数 , 返回时间参数 (us) , pin 表示为 0 ~ 13 , value 为 HIGH 或 LOW。比如 value 为 HIGH 那么当 pin 输入为高电平时 , 开始计时 , 当 pin 输入为低电平时 , 停止计时 , 然后返回该时间。

时间函数

- unsigned long millis() 返回时间函数 (单位 ms) , 该函数是指 , 当程序运行就开始计时并返回记录的参数 , 该参数溢出大概需要 50 天时间。
- delay(ms) 延时函数 (单位 ms) 。
- delayMicroseconds(us) 延时函数 (单位 us) 。

数学函数

- min(x, y) 求最小值
- max(x, y) 求最大值

-
- `abs(x)` 计算绝对值
 - `constrain(x, a, b)` 约束函数，下限 `a`，上限 `b`，`x` 必须在 `ab` 之间才能返回。
 - `map(value, fromLow, fromHigh, toLow, toHigh)` 约束函数，`value` 必须在 `fromLow` 与 `toLow` 之间和 `fromHigh` 与 `toHigh` 之间。
 - `pow(base, exponent)` 开方函数，`base` 的 `exponent` 次方。
 - `sq(x)` 平方
 - `sqrt(x)` 开根号

三角函数

- `sin(rad)`
- `cos(rad)`
- `tan(rad)`

随机数函数

- `randomSeed(seed)` 随机数种子定义函数，`seed` 表示读模拟口 `analogRead(pin)` 函数。
- `long random(max)` 随机数函数，返回数据大于等于 0，小于 `max`。
- `long random(min, max)` 随机数函数，返回数据大于等于 `min`，小于 `max`。

外部中断函数

-
- `attachInterrupt(interrupt, , mode)` 外部中断只能用到数字 IO 口 2 和 3, `interrupt` 表示中断口初始 0 或 1, 表示一个功能函数, `mode`: **LOW** 低电平中断, **CHANGE** 有变化就中断, **RISING** 上升沿中断, **FALLING** 下降沿中断。
 - `detachInterrupt(interrupt)` 中断开关, `interrupt=1` 开, `interrupt=0` 关。

中断使能函数

- `interrupts()` 使能中断
- `noInterrupts()` 禁止中断

串口收发函数

- `Serial.begin(speed)` 串口定义波特率函数, `speed` 表示波特率, 如 9600, 19200 等。
- `int Serial.available()` 判断缓冲器状态。
- `int Serial.read()` 读串口并返回收到参数。
- `Serial.flush()` 清空缓冲器。
- `Serial.print(data)` 串口输出数据。
- `Serial.println(data)` 串口输出数据并带回车符。

/*****Arduino 语言库文件*****/

官方库文件

- **EEPROM** - EEPROM 读写程序库
- **Ethernet** - 以太网控制器程序库

-
- LiquidCrystal - LCD 控制程序库
 - Servo - 舵机控制程序库
 - SoftwareSerial - 任何数字 IO 口模拟串口程序库
 - Stepper - 步进电机控制程序库
 - Wire - TWI/I2C 总线程序库
 - Matrix - LED 矩阵控制程序库
 - Sprite - LED 矩阵图象处理控制程序库

非官方库文件

- DateTime - a library for keeping track of the current date and time in software.
- Debounce - for reading noisy digital inputs (e.g. from buttons)
- Firmata - for communicating with applications on the computer using a standard serial protocol.
- GLCD - graphics routines for LCD based on the KS0108 or equivalent chipset.
- LCD - control LCDs (using 8 data lines)
- LCD 4 Bit - control LCDs (using 4 data lines)
- LedControl - for controlling LED matrices or seven-segment displays with a MAX7221 or MAX7219.
- LedControl - an alternative to the Matrix library for driving multiple LEDs with Maxim chips.

-
- Messenger - for processing text-based messages from the computer
 - Metro - help you time actions at regular intervals
 - MsTimer2 - uses the timer 2 interrupt to trigger an action every N milliseconds.
 - OneWire - control devices (from Dallas Semiconductor) that use the One Wire protocol.
 - PS2Keyboard - read characters from a PS2 keyboard.
 - Servo - provides software support for Servo motors on any pins.
 - Servotimer1 - provides hardware support for Servo motors on pins 9 and 10
 - Simple Message System - send messages between Arduino and the computer
 - SSerial2Mobile - send text messages or emails using a cell phone (via AT commands over software serial)
 - TextString - handle strings
 - TLC5940 - 16 channel 12 bit PWM controller.
 - X10 - Sending X10 signals over AC power lines

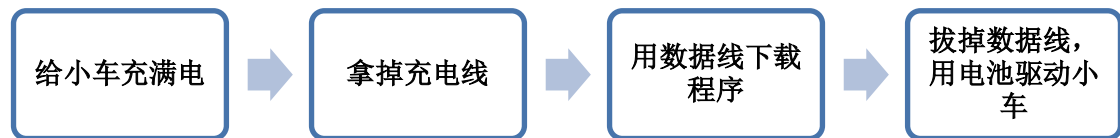
5、Arduinio 智能四驱车介绍

使用注意事项:

1) 不要用手触摸芯片！

2) 不要带电操作！在断电的情况下，拔插接插件。

3) 小车正确使用流程：



4) 电池电压低于 9.4V 就要充电，充电 4 小时即可，不要过充！

5) 电池满电的情况下，小车可以运行 40 分钟左右。

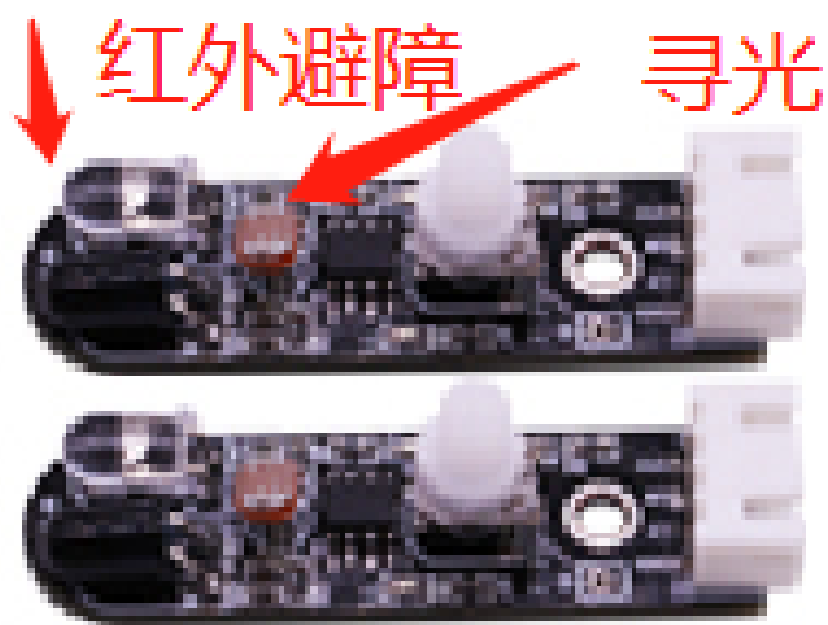


直流减速电机



舵机





四路巡线模块



灰度模块



超声波模块





ZUCC CS