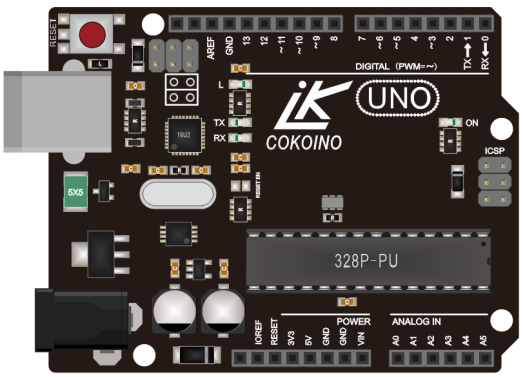
**ARDUINO UNO R3 开发板**



1. **概述**

这是一款非常常用的Arduino板，也是最受欢迎的一款Arduino板。在学习了解这块控制板前，我们需要先了解什么是arduino？

Arduino是一个基于易用硬件和软件的原型平台（开源），它由可编程的电路板（称为微控制器）和称为Arduino IDE（集成开发环境）的现成软件组成。

主要特点是：

①硬件开源，且硬件技术成熟，可通过arduino官网获取相关技术支持。

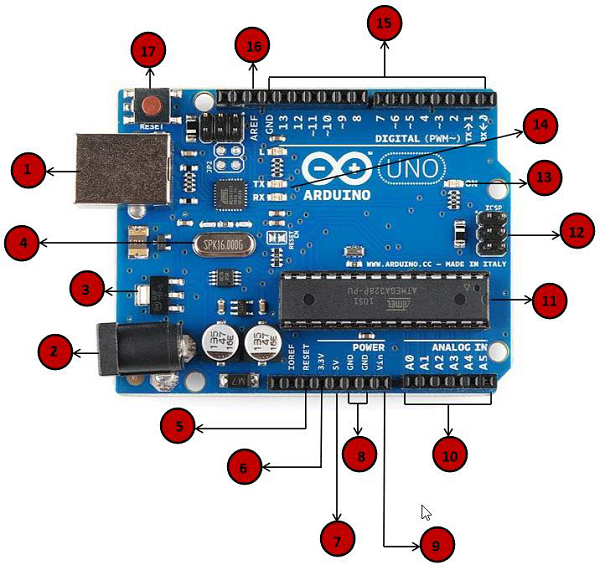
②软件开源，使用封装式C语言，不用理会底层操作，可通过库函数形式实现众多功能，简单易懂，资源众多。

③爱好者多，形成生态式覆盖，网络资源多，趣味且简单，非常合适想学习硬件、软件的电子爱好者。

**2、Arduino板的说明**

Arduino板不是单指一种板子，它是指可以在arduino平台上使用的一系列控制板，有很多种类，其中有3款是我们用得最多的，分别是Arduino UNO R3、Arduino mega 2560和Arduino Nano板。

首先我们来看下Arduino UNO R3板，以下为Arduino UNO R3板各个引脚及主要元器件说明。



|  |  |  |
| --- | --- | --- |
| 1 | 编程USB口 | 通过此USB口可以将PC上的IDE代码烧录到UNO R3板上 |
| 2 | 电源  （DC插座） | 此电源接口可外接DC7-12V电源给UNO R3主板供电。 |
| 3 | 稳压器 | 稳压器的功能是给UNO R3主板提供稳定的+5V电源，芯片型号为AMS1117-5.0。 |
| 4 | 晶体振荡器 | 16M晶体振荡器，给MEGA328P主芯片提供振荡频率。 |
| 5、17 | Arduino  复位 | 可以通过两种方式重置UNO板。1、通过使用板上的复位按钮（17）。2、也可以通过将标有RESET（5）的引脚接到GND，从而达到复位。 |
| 6、7、8、9 | 引脚  （3.3，5，GND，Vin） | 3.3V（6） - 提供3.3输出电压，最大输出电流为50mA，  5V（7） - 提供5输出电压，  GND（8）- 接地，  Vin（9） - 此引脚也可用于外接DC7-12V电源 |
| 10 | 模拟引脚 | UNO R3主板有六个模拟输入引脚，A0到A5。这些引脚可以读取模拟值（也可当作数字口用），A4\A5也可作为IIC接口。 |
| 11 | 微控制器 | 每个Arduino板都有自己的微控制器（11），你可以假设它作为板的大脑。微控制器通常是ATMEL公司的，基本信息位于IC顶部。有关IC结构和功能的更多详细信息，请参阅数据表。 |
| 12 | ICSP引脚 | 此接口由MOSI，MISO，SCK，RESET，VCC和GND组成的Arduino的微型编程头，它通常被称为SPI（串行外设接口）。 |
| 13 | 电源LED指示灯 | 当你将Arduino插入电源时，此LED指示灯应亮起，表明你的电路板已正确通电。如果这个指示灯不亮，表示板不能正常工作。 |
| 14 | TX和RX LED | TX和RX LED，发送串行数据时，TX LED以不同的速度闪烁，RX在接收过程中闪烁，闪烁速度取决于板所使用的波特率。 |
| 15 | 数字I/O | Arduino UNO板有14个数字I/O引脚。有些引脚也可作特殊功能接口，其中标有“〜”的6个引脚可用于产生PWM；0\1为串口；10\11\12\13为SPI接口。 |
| 16 | AREF | AREF代表模拟参考。用于设置外部参考电压（0至5伏之间），使用analogReference() 函数设置。 |

**3、规格参数**

Microcontroller：ATmega328P

Operating Voltage ：5V

Input Voltage (recommended)：7-12V

Input Voltage (limit)：6-20V（Is not recommended to use）

Digital I/O Pins：14 (of which 6 provide PWM output)

PWM Digital I/O Pins：6（D3, D5, D6, D9, D10, D11）

Analog Input Pins：6（A0~A5）

DC Current per I/O Pin ：20 mA

DC Current for 3.3V Pin：50 mA

Flash Memory：32 KB (ATmega328P) of which 0.5 KB used by bootloader

SRAM：2 KB (ATmega328P)

EEPROM：1 KB (ATmega328P)

Clock Speed：16 MHz

LED\_BUILTIN：13（IO）

size：68.6mm\*53.4mm

Bootloader：UNO REV3

**Power**

The Arduino Uno board can be powered via the USB connection or with an external power supply. The power source is selected automatically.

External (non-USB) power can come either from an AC-to-DC adapter (wall-wart) or battery. The adapter can be connected by plugging a 2.1mm center-positive plug into the board's power jack. Leads from a battery can be inserted in the GND and Vin pin headers of the POWER connector.

The board can operate on an external supply from 6 to 20 volts. If supplied with less than 7V, however, the 5V pin may supply less than five volts and the board may become unstable. If using more than 12V, the voltage regulator may overheat and damage the board. The recommended range is 7 to 12 volts.

The power pins are as follows:

\*Vin. The input voltage to the Arduino/Genuino board when it's using an external power source

(as opposed to 5 volts from the USB connection or other regulated power source). You can

supply voltage through this pin, or, if supplying voltage via the power jack, access it through

this pin.

\*5V.This pin outputs a regulated 5V from the regulator on the board. The board can be supplied

with power either from the DC power jack (7 - 12V), the USB connector (5V), or the VIN pin

of the board (7-12V). Supplying voltage via the 5V or 3.3V pins bypasses the regulator, and can

damage your board. We don't advise it.

\*3V3. A 3.3 volt supply generated by the on-board regulator. Maximum current draw is 50 mA.

\*GND. Ground pins.

\*IOREF. This pin on the Arduino/Genuino board provides the voltage reference with which the

microcontroller operates. A properly configured shield can read the IOREF pin voltage and

select the appropriate power source or enable voltage translators on the outputs to work with the

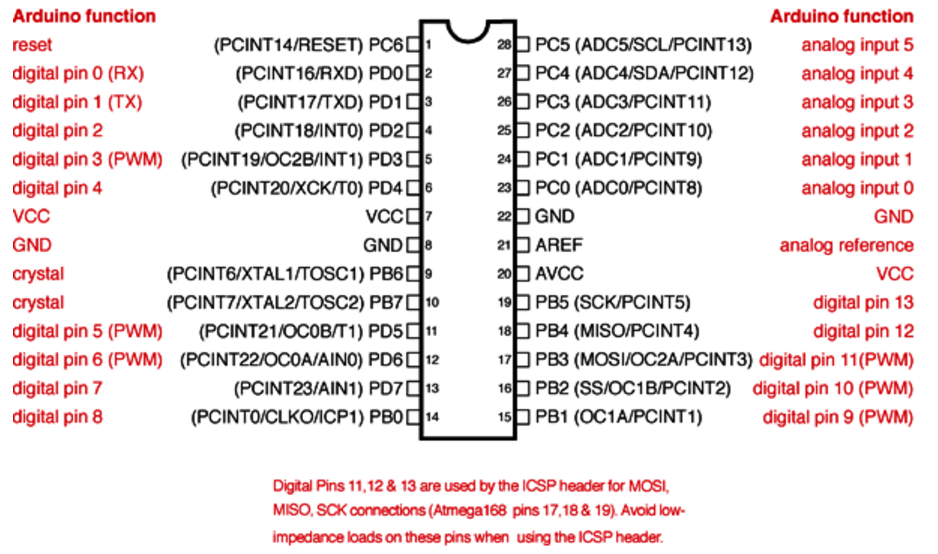
5V or 3.3V.

**Memory**

The ATmega328 has 32 KB (with 0.5 KB occupied by the bootloader). It also has 2 KB of SRAM and 1 KB of EEPROM (which can be read and written with the EEPROM library).

**Input and Output**

See the mapping between Arduino pins and ATmega328P ports. The mapping for the Atmega8, 168, and 328 is identical.



Each of the 14 digital pins on the Uno can be used as an input or output, using pinMode(),digitalWrite(), and digitalRead() functions. They operate at 5 volts. Each pin can provide or receive 20 mA as recommended operating condition and has an internal pull-up resistor (disconnected by default) of 20-50k ohm. A maximum of 40mA is the value that must not be exceeded on any I/O pin to avoid permanent damage to the microcontroller.

In addition, some pins have specialized functions:

Serial: 0 (RX) and 1 (TX). Used to receive (RX) and transmit (TX) TTL serial data. These pins are connected to the corresponding pins of the ATmega8U2 USB-to-TTL Serial chip.

External Interrupts: 2 and 3. These pins can be configured to trigger an interrupt on a low value, a rising or falling edge, or a change in value. See the attachInterrupt() function for details.

PWM: 3, 5, 6, 9, 10, and 11. Provide 8-bit PWM output with the analogWrite() function.

SPI: 10 (SS), 11 (MOSI), 12 (MISO), 13 (SCK). These pins support SPI communication using the SPI library.

LED: 13. There is a built-in LED driven by digital pin 13. When the pin is HIGH value, the LED is on, when the pin is LOW, it's off.

TWI: A4 or SDA pin and A5 or SCL pin. Support TWI communication using the Wire library.

The Uno has 6 analog inputs, labeled A0 through A5, each of which provide 10 bits of resolution (i.e. 1024 different values). By default they measure from ground to 5 volts, though is it possible to change the upper end of their range using the AREF pin and the analogReference() function. There are a couple of other pins on the board:

AREF. Reference voltage for the analog inputs. Used with analogReference().

Reset. Bring this line LOW to reset the microcontroller. Typically used to add a reset button to shields which block the one on the board.

**Communication**

Arduino/Genuino Uno has a number of facilities for communicating with a computer, another Arduino/Genuino board, or other microcontrollers. The ATmega328 provides UART TTL (5V) serial communication, which is available on digital pins 0 (RX) and 1 (TX). An ATmega16U2 on the board channels this serial communication over USB and appears as a virtual com port to software on the computer. The 16U2 firmware uses the standard USB COM drivers, and no external driver is needed. However, on Windows, a .inf file is required. The Arduino Software (IDE) includes a serial monitor which allows simple textual data to be sent to and from the board. The RX and TX LEDs on the board will flash when data is being transmitted via the USB-to-serial chip and USB connection to the computer (but not for serial communication on pins 0 and 1).

A SoftwareSerial library allows serial communication on any of the Uno's digital pins.

The ATmega328 also supports I2C (TWI) and SPI communication. The Arduino Software (IDE) includes a Wire library to simplify use of the I2C bus; see the documentation for details. For SPI communication, use the SPI library.

Automatic (Software) Reset

Rather than requiring a physical press of the reset button before an upload, the Arduino/Genuino Uno board is designed in a way that allows it to be reset by software running on a connected computer. One of the hardware flow control lines (DTR) of the ATmega8U2/16U2 is connected to the reset line of the ATmega328 via a 100 nanofarad capacitor. When this line is asserted (taken low), the reset line drops long enough to reset the chip. The Arduino Software (IDE) uses this capability to allow you to upload code by simply pressing the upload button in the interface toolbar. This means that the bootloader can have a shorter timeout, as the lowering of DTR can be well-coordinated with the start of the upload.

This setup has other implications. When the Uno is connected to either a computer running Mac OS X or Linux, it resets each time a connection is made to it from software (via USB). For the following half-second or so, the bootloader is running on the Uno. While it is programmed to ignore malformed data (i.e. anything besides an upload of new code), it will intercept the first few bytes of data sent to the board after a connection is opened. If a sketch running on the board receives one-time configuration or other data when it first starts, make sure that the software with which it communicates waits a second after opening the connection and before sending this data.

The Uno board contains a trace that can be cut to disable the auto-reset. The pads on either side of the trace can be soldered together to re-enable it. It's labeled "RESET-EN". You may also be able to disable the auto-reset by connecting a 110 ohm resistor from 5V to the reset line.

**Revisions**

Revision 3 of the board has the following new features:

1.0 pinout: added SDA and SCL pins that are near to the AREF pin and two other new pins placed near to the RESET pin, the IOREF that allow the shields to adapt to the voltage provided from the board. In future, shields will be compatible with both the board that uses the AVR, which operates with 5V and with the Arduino Due that operates with 3.3V. The second one is a not connected pin, that is reserved for future purposes.

Stronger RESET circuit.

Atmega 16U2 replace the 8U2

**4、Windows arduino IDE(安装版)**

（备注：IDE有需要安装版本和免安装版本，选择其中一种即可，如需旧版本IED请到此链接下载：https://www.arduino.cc/en/Main/OldSoftwareReleases#1.0.x）

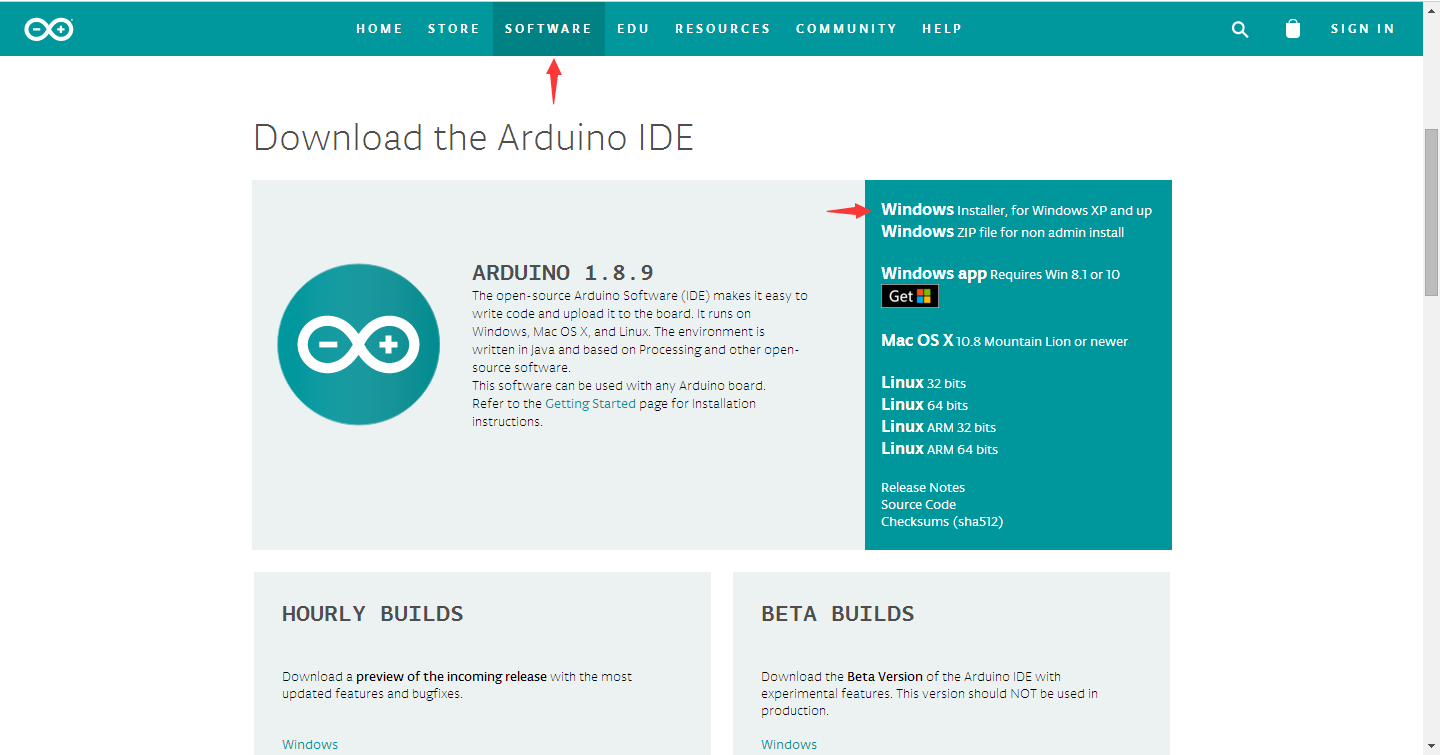
4.1、最新IDE安装包下载链接（https://www.arduino.cc/en/Main/Software）：

图1



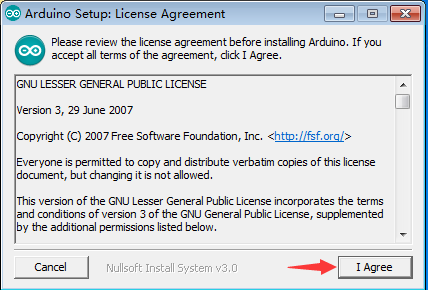
图2

4.2、安装

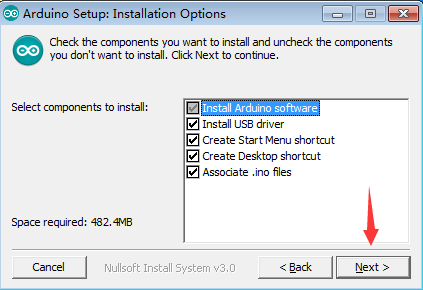
4.2.1、双击刚才下载名为“arduino-1.8.9-windows.exe”的安装包开始安装IDE，如下图：



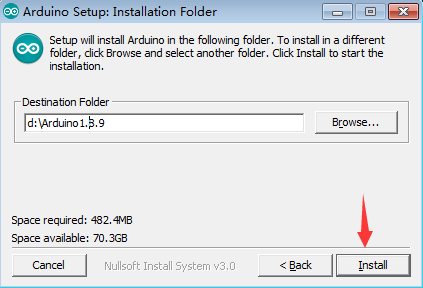
4.2.2、 选择“I Agree”进行安装：



4.2.3、接着选择“Next”，如下图：



4.2.4、安装目录可以根据自己需求改到对应系统盘里面，接着点击“Install”，如下图：



4.2.5、安装进行时与安装完成，如下图：

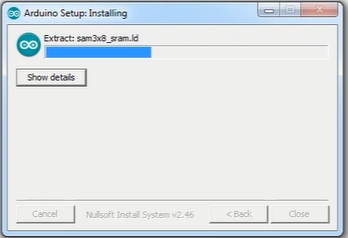


图1，正在安装

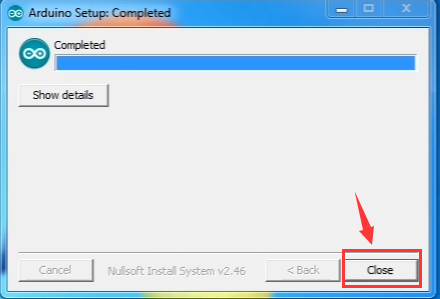
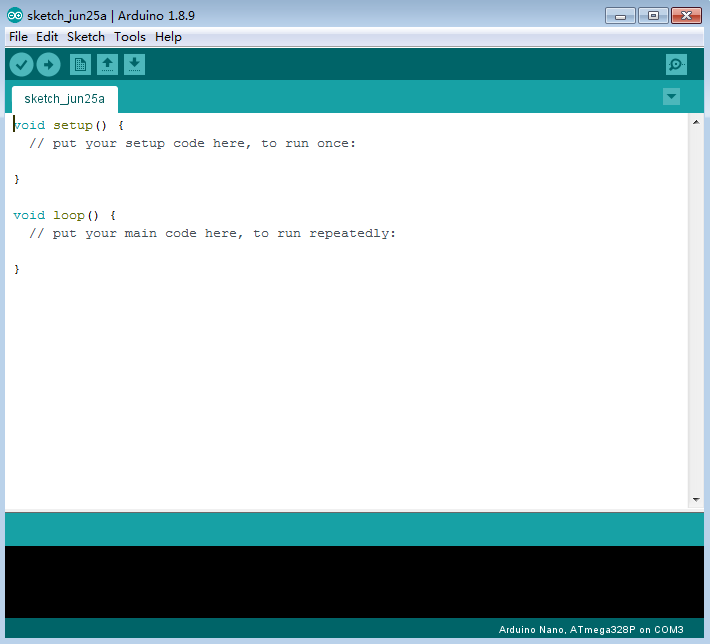


图2，完成安装

4.2.5、双击安装完成后PC桌面上生成的快捷打开方式图标，即可打arduino IDE，如下图：



1. **Windows arduino IDE(免安装版)**

（备注：IDE有需要安装版本和免安装版本，选择其中一种即可，如需旧版本IED请到此链接下载：https://www.arduino.cc/en/Main/OldSoftwareReleases#1.0.x）

5.1、最新IDE安装包下载链接（https://www.arduino.cc/en/Main/Software）：

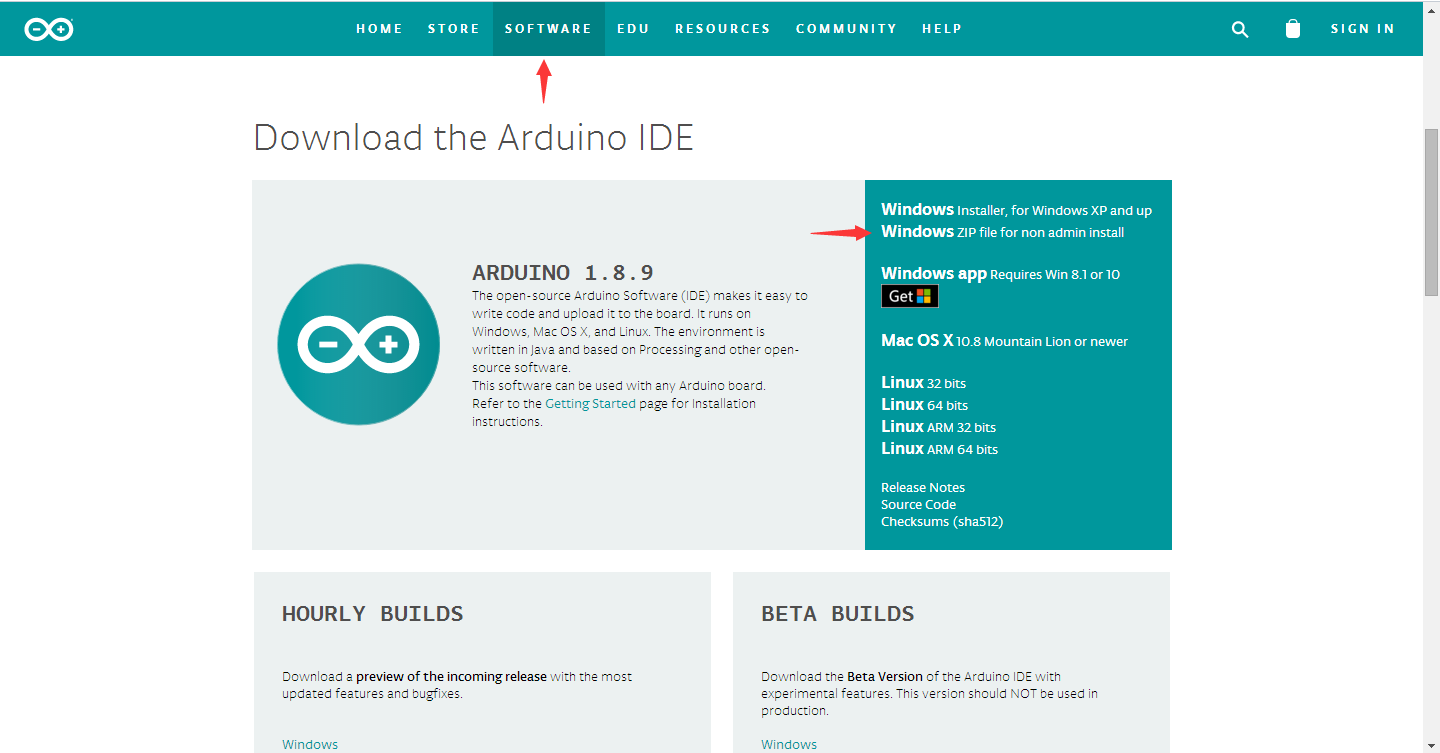


图1



图2

5.2、下载的为免安装的IDE，解压即可用，无需安装，如下图：

解压刚才下载的压缩包，将解压好的压缩包复制到需要放置的系统盘里，双击进入文件夹，然后点击如arduino图标打开IDE，如下图：

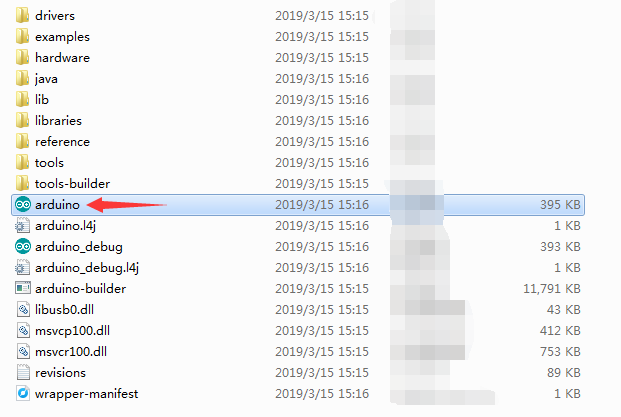


图1

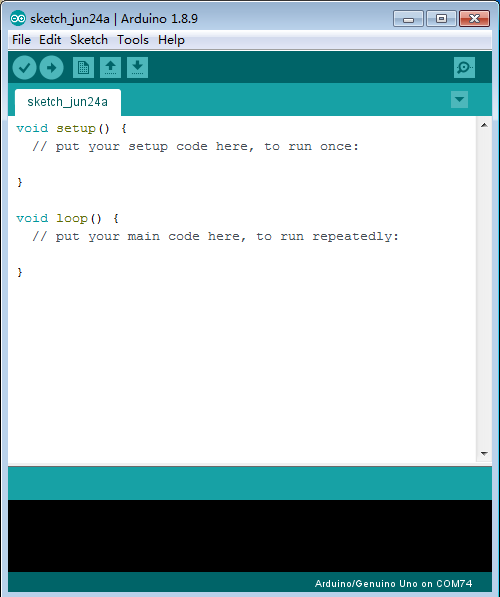


图2

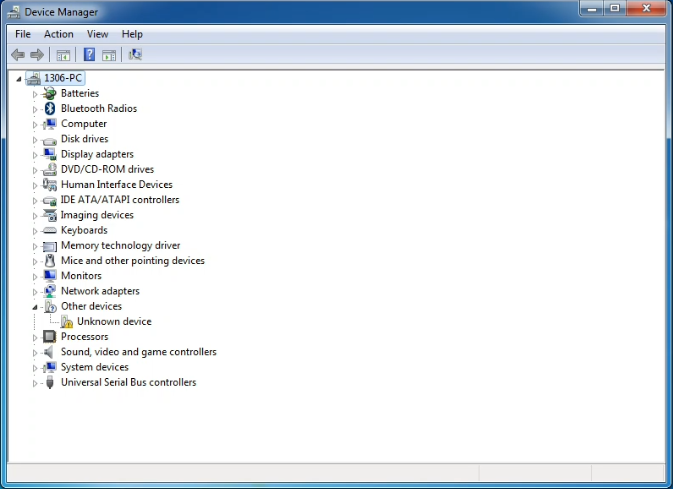
**6、arduino 官网主板USB驱动安装**

官网版本的主控板USB驱动的安装方法，如：UNO R3、MEGA2560等，下面以UNO R3为例：

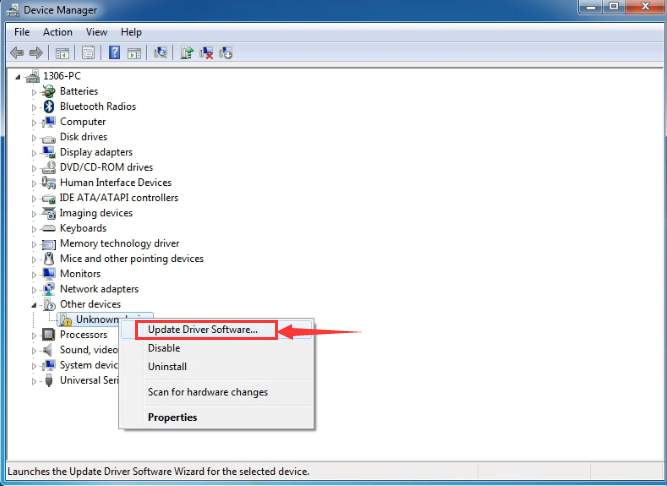
6.1、电脑第一次安装arduino IDE时，正常安装完成IDE 后，直接连接UNO R3主板是无法识别到主板的端口的，需要安装相应的驱动。

首先电脑USB口通过USB线连接UNO R3的USB口，然后右击电脑桌面图标“Computer”—>“Properties”

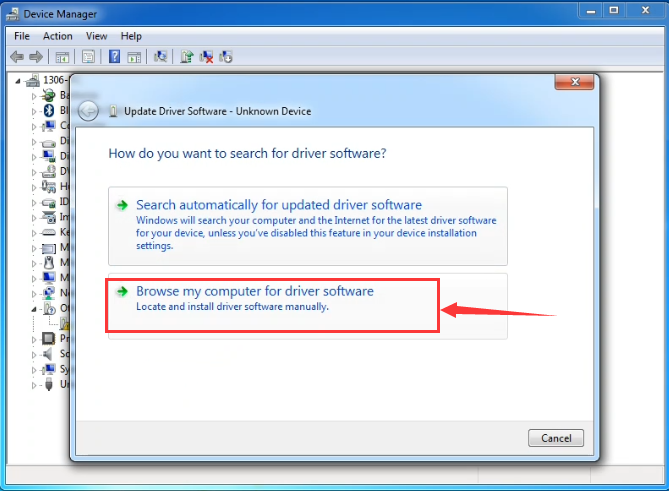
—> “Device manager”，会看到无法识别的USB口“Unknown devices”，如下图：



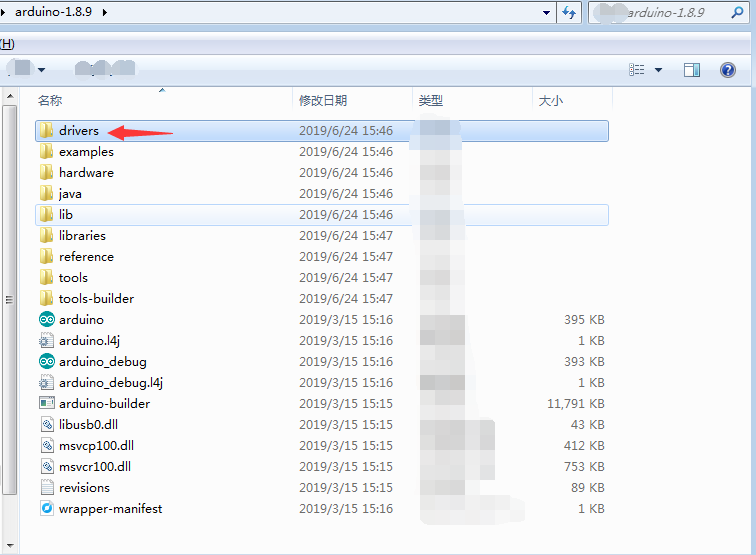
6.2、右键点击“Unknown devices”, 选择“Update Driver software”，如下图：



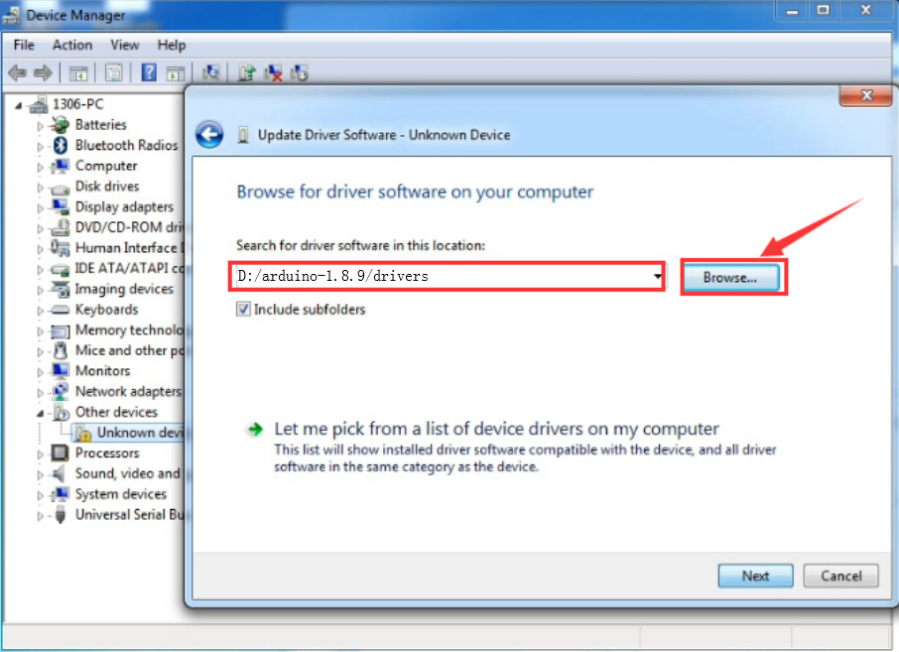
6.3、点击“Browse my computer for driver software”，如下图：



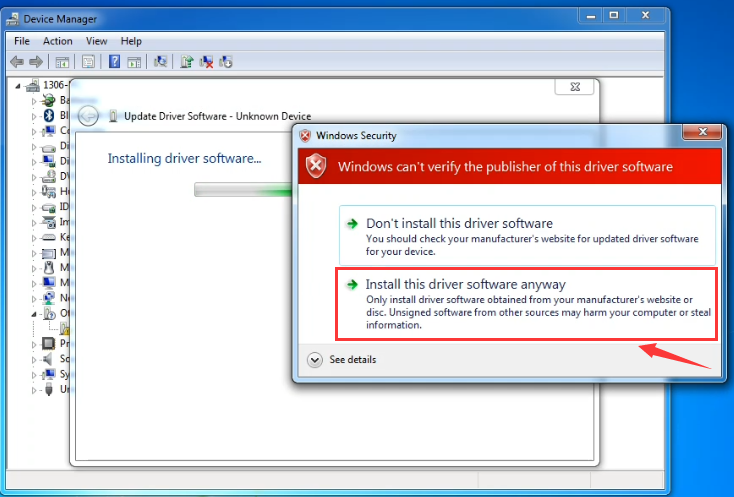
6.4、arduino IDE自带有官网版本主板的USB驱动，在IDE安装目录下的“drivers”文件夹下，如下图：



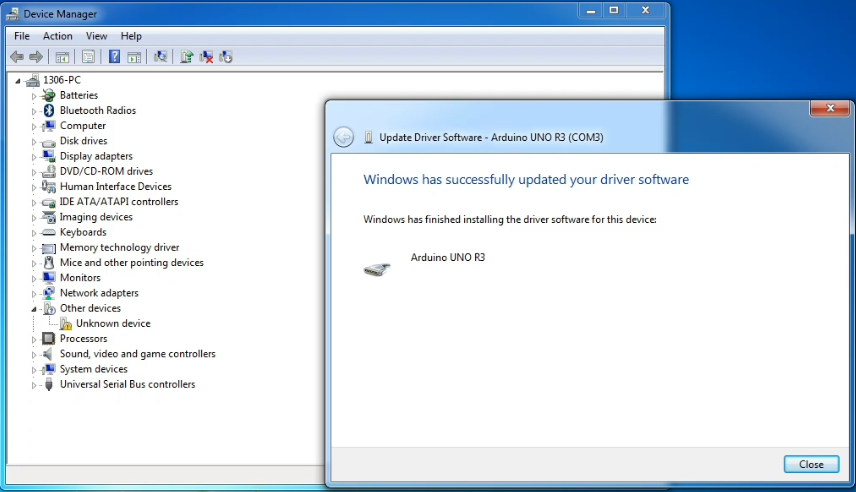
6.5、找到驱动所在的文件夹“drivers”，然后点击“Next”，如下图：



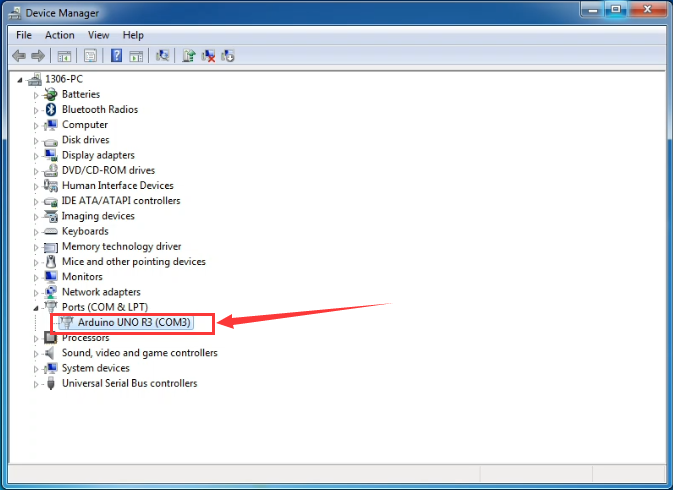
6.6、然后选择 “Install this driver software anyway” 进行安装，如下图：



6.7、安装完成后点击 “Close”，如下图：

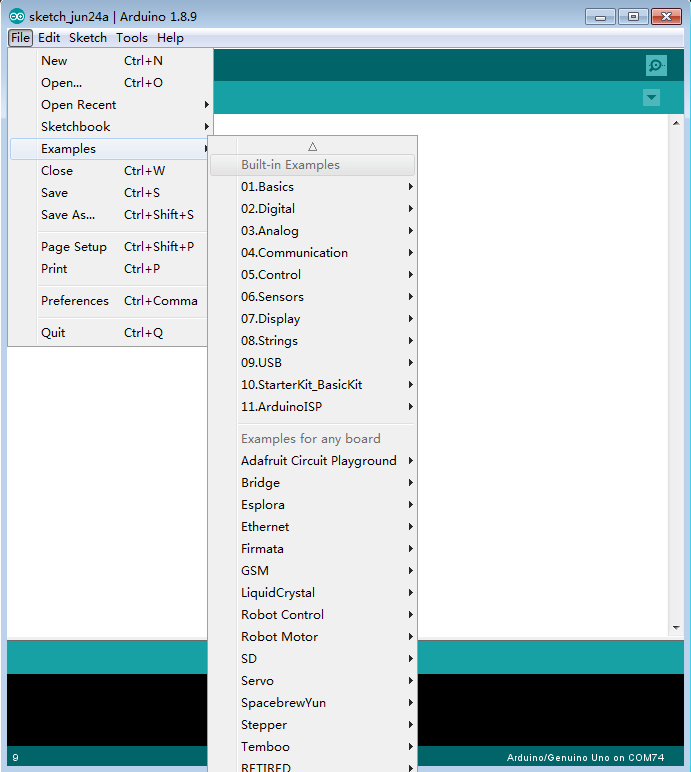


6.8、安装完成后，重新查看 “Device manager”。 右击电脑桌面图标“Computer”—>“Properties”—> “Device manager”，此时可以找到UNO R3的USB端口，如下图：

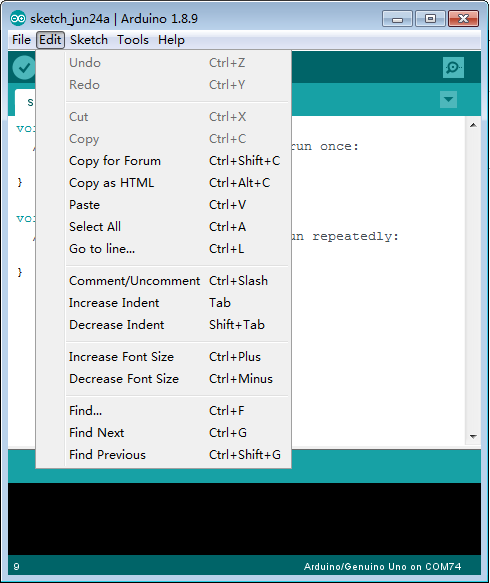


**7、arduino IDE 简介**

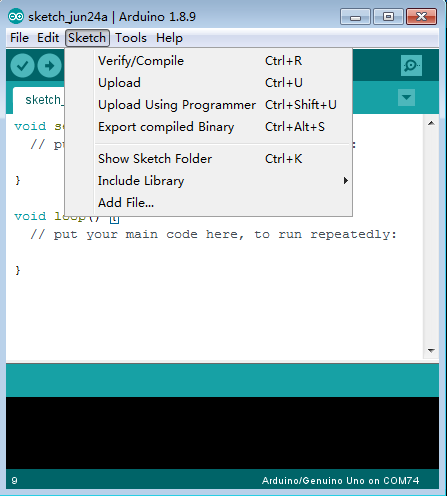
7.1、：新建工程、打开已有工程等，Examples下还自带有很多现成的示例程序，如下图：



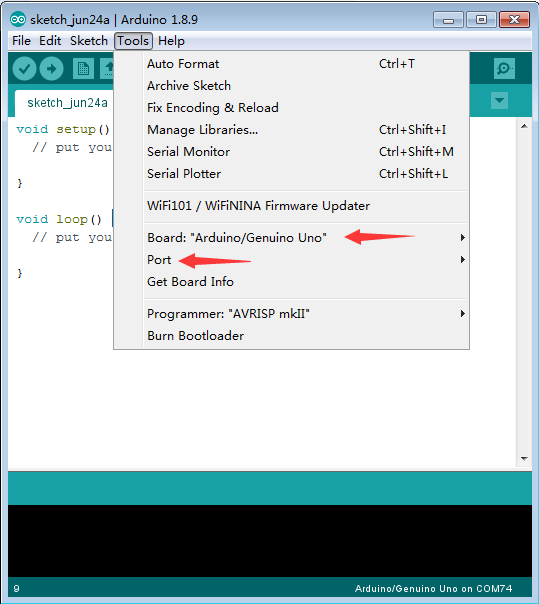
7.2、：撤销、复原等，当写错程序时，想恢复原来的程序，此功能经常用到。



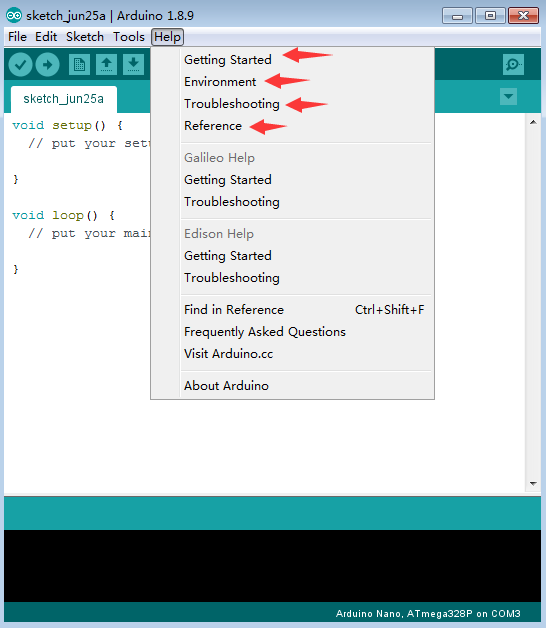
7.3、：编译、下载程序等。



7.4、：主要是板型与端口等信息，当烧录程序时需选择对应的板型与端口号，否则程序烧录不进主板中，如下图：

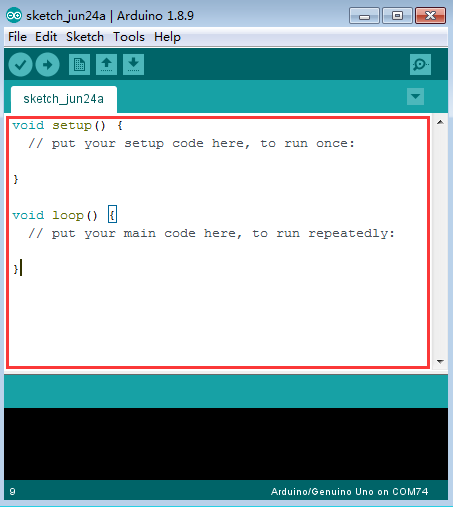


7.5、：主要是官网入门教程、IDE详细简介、常见错误解决、编程语言学习等，建议初学者务必去看，绝对受益匪浅。

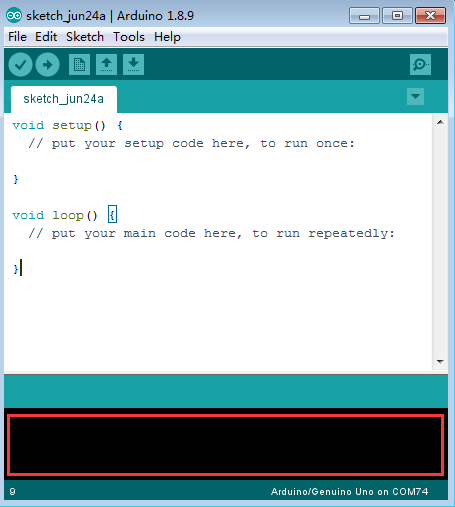


7.6、代码编写区，用于编写所需程序，如下图：

（编程学习网址：https://www.arduino.cc/reference/en/）



7.7、信息输出区，主要输出编译与上传代码的信息。



7.9、串口监视器，要在“tools”--->“Port”选择到对应主板的端口时才能打得开（板型可以不选择），如下图是接上了UNO主板，Port选择COM3端口，设置波特率为9600。

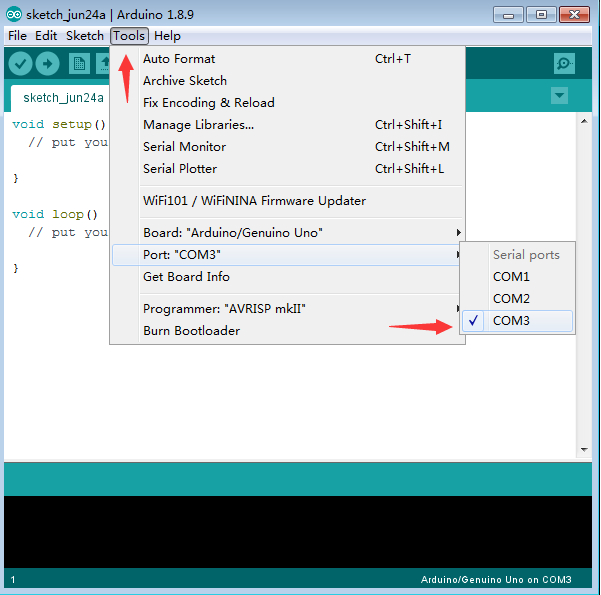


图1

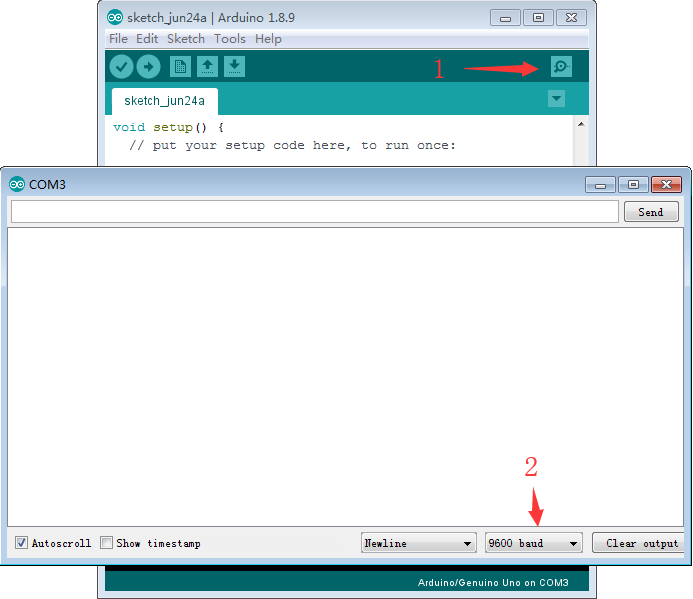


图2

7.10、：快速编译按钮，检验代码的是否正确。

7.11、：快速上传代码按钮，将代码上传到主板。

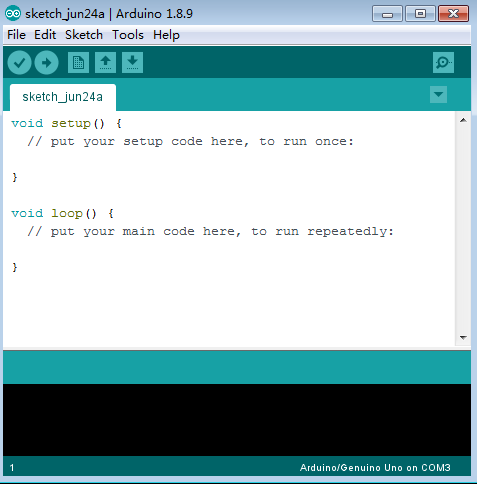
7.12、：快速新建工程按钮。

7.13、：快速打开已有工程与IDE自带示例程序按钮。

7.14、：快速保存正在编写程序的按钮。

**8、示例**

8.1、双击电脑桌面图标，打开arduino IDE，如下图：



8.2、打开示例闪灯程序，点击“File”-->“Examples”-->“Basics”-->“Blink”，如下图：

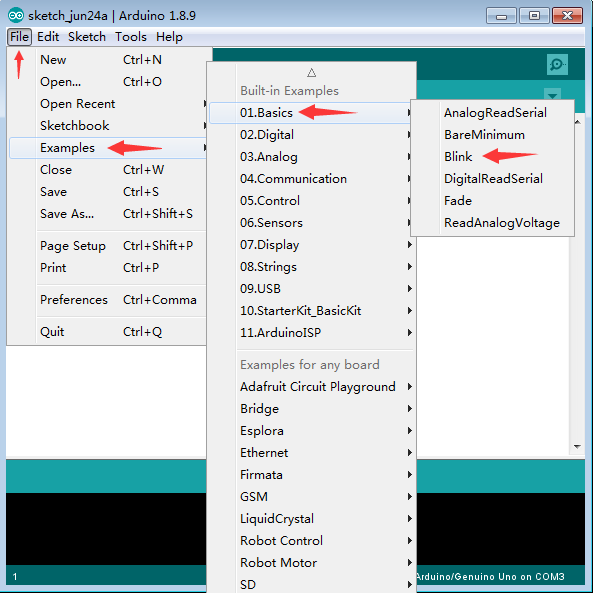


图1

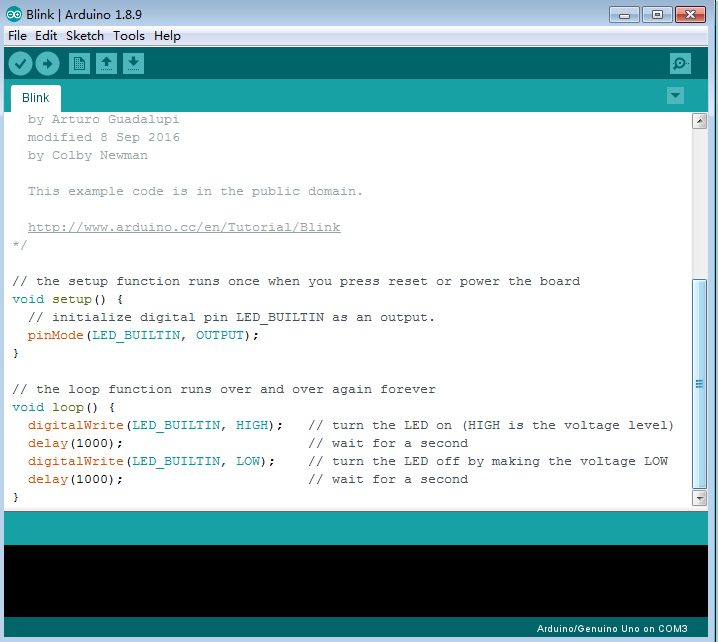
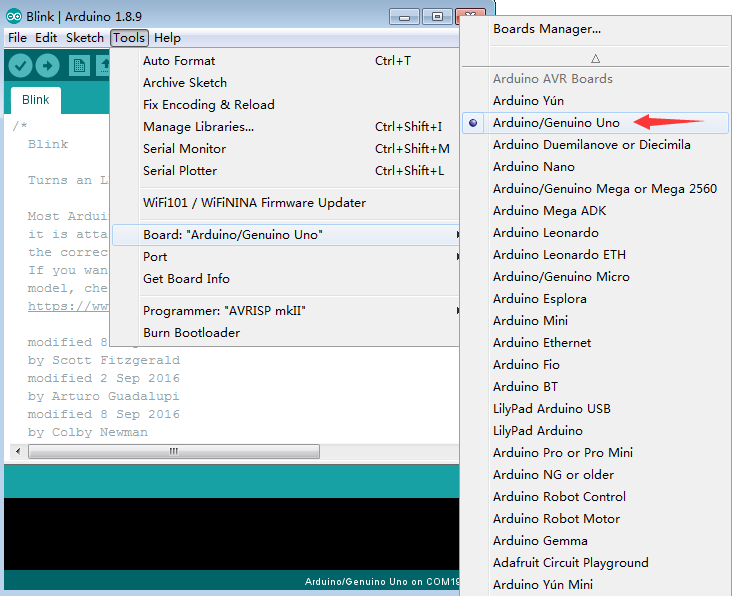
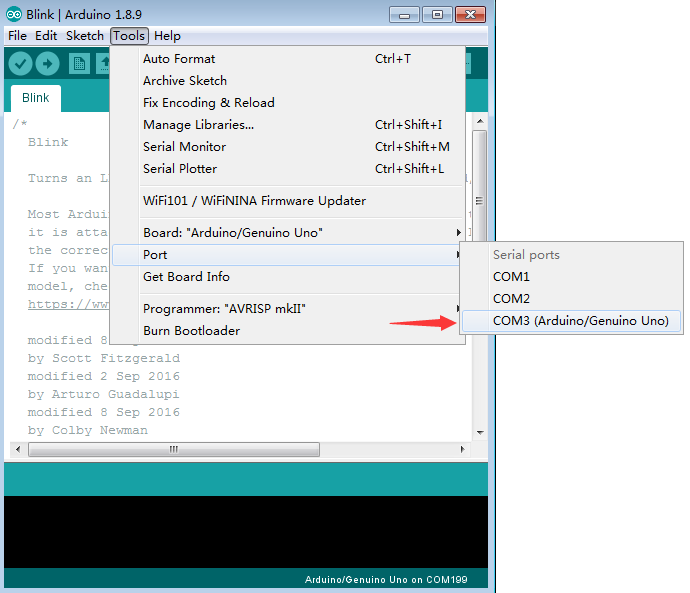


图2

8.3、选择板型，点击“tools”--->“Board”--->“Arduino/Genuino UNO”,如下图：



8.5、选择端口，“tools”--->“Port”--->“arduino主板的COM口”，如下图：



8.6、编译代码，检验代码的正确性，如下图：



图一，正在编译检验代码

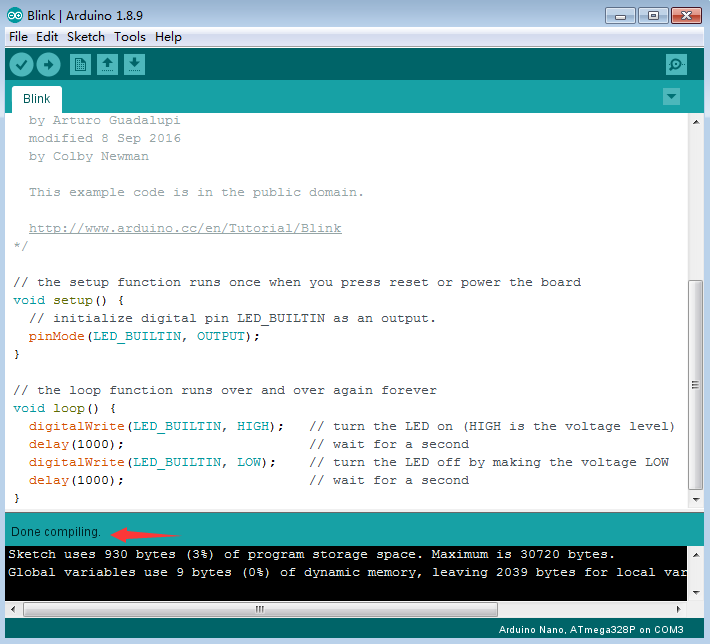


图2，正确完成编译

8.7、上传代码，PC的USB口通过USB线连接arduino主板的USB口，然后上传代码到arduino主板上，如下图：

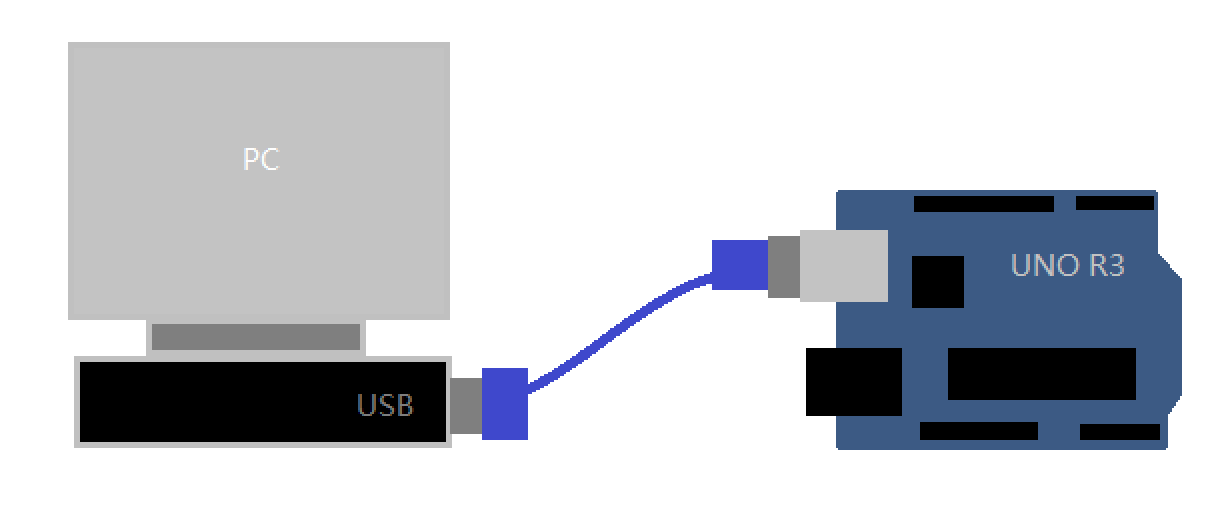


图1，PC与UNO R3接线方式



图2，正在上传代码

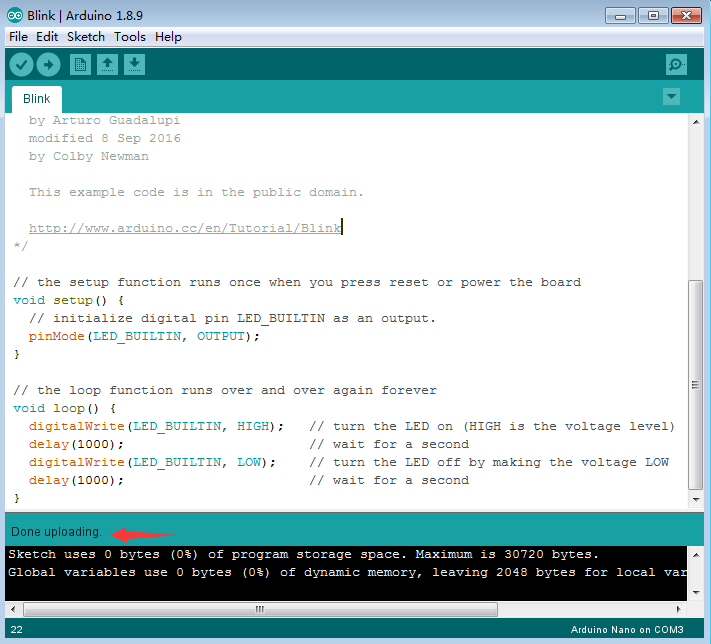
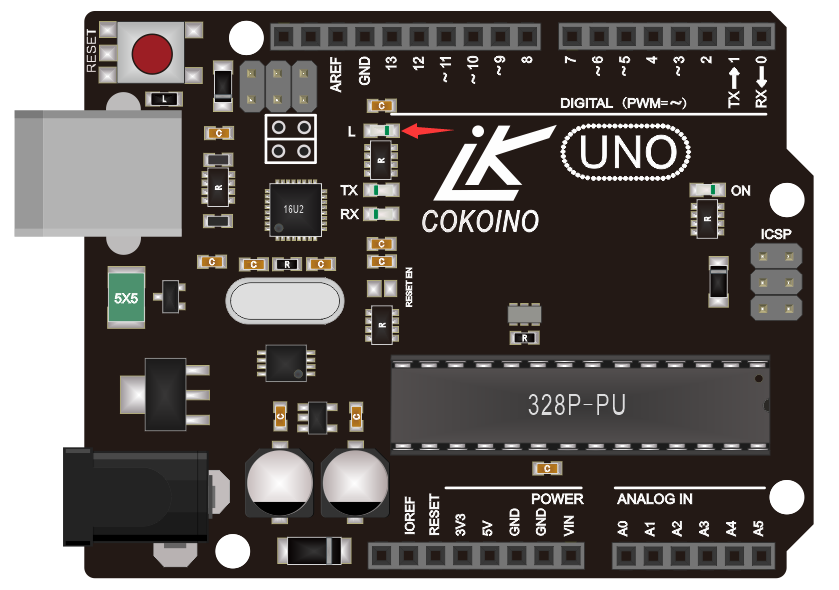


图3，完成上传代码

8.8、示例效果，nano主板L丝印位置的LED灯每间隔1秒钟闪亮一次，如下图：



**9、关于arduino库文件讲解**

Arduino之所以说容易上手，是因为它把相对于初学者很难的底层操作封装成库的形式，把应用接口函数预留出来给使用者操作，使用者只要使用对应的接口函数就可以实现需要的功能，这样大大降低了写程序的门槛。

**9.1、何为arduino库文件**

库文件在程序源文件中显示“#include<xxx.h>”，如下图为伺服电机的库函数：

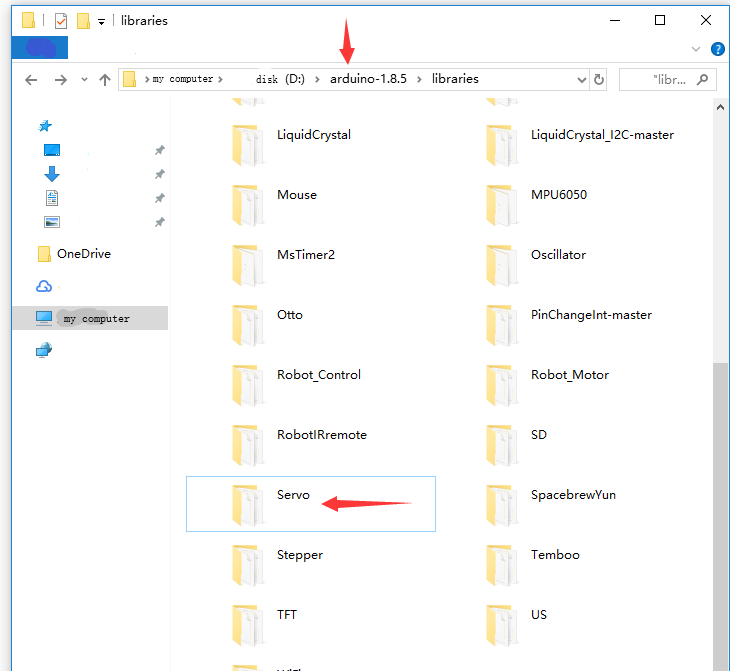


**9.2、库文件放在哪里？**

库文件就是把一些以“xxx.h”结尾的头文件和以“xxx.c”结尾的源文件放在一个文件夹里面，就组合一个库文件。

关于如何写arduino库文件，请自行参考相关资料学习。

我们只需要把网上下载或者自己写的库文件放在arduino IDE安装文件夹下的libraries文件下即可在arduino IDE中正常使用，否则检验程序时会报错，如下图为我电脑arduinoIDE库文件放置目录：



备注：更多arduino 官方参考资料请查阅如下网站

https://www.arduino.cc/en/Reference/Libraries