

---

# Getting Started with Arduino and Nano

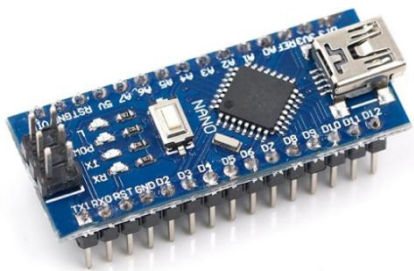
## Points of this section

**You will learn:**

- ◆ How to install the Arduino Software (IDE) on Windows PCs
- ◆ About Nano Board
- ◆ Use your nano on the arduino desktop ide and upload your first sketch

**Need to prepare:**

- ◆ A computer with a reliable Internet connection
- ◆ A LK COKOINO nano board
- ◆ A USB cable



## Catalogue

The Arduino IDE.....	2
Install the Arduino Software (IDE) on Windows PCs.....	2
About the Nano Board.....	5
1、 Overview: .....	5
2、 Specification: .....	5
3、 Description: .....	6
Install The Board Driver.....	7
Upload your first sketch.....	8

---

# The Arduino IDE

The open-source Arduino Software (IDE) makes it easy to write code and upload it to the board. It runs on Windows, Mac OS X, and Linux. The environment is written in Java and based on Processing and other open-source software. It can be used with any Arduino board.

## Install the Arduino Software (IDE) on Windows PCs

Get the latest version from this link (<https://www.arduino.cc/en/Main/Software>). You can choose between the Installer (.exe) and the Zip packages. We suggest you use the first one that installs directly everything you need to use the Arduino Software (IDE), including the drivers. With the Zip package you need to install the drivers manually. The Zip file is also useful if you want to create a portable installation.




### ARDUINO 1.8.9

The open-source Arduino Software (IDE) makes it easy to write code and upload it to the board. It runs on Windows, Mac OS X, and Linux. The environment is written in Java and based on Processing and other open-source software.

This software can be used with any Arduino board. Refer to the [Getting Started](#) page for Installation instructions.

**Windows** Installer, for Windows XP and up  
**Windows** ZIP file for non admin install

**Windows app** Requires Win 8.1 or 10  
[Get](#) 

**Mac OS X** 10.8 Mountain Lion or newer

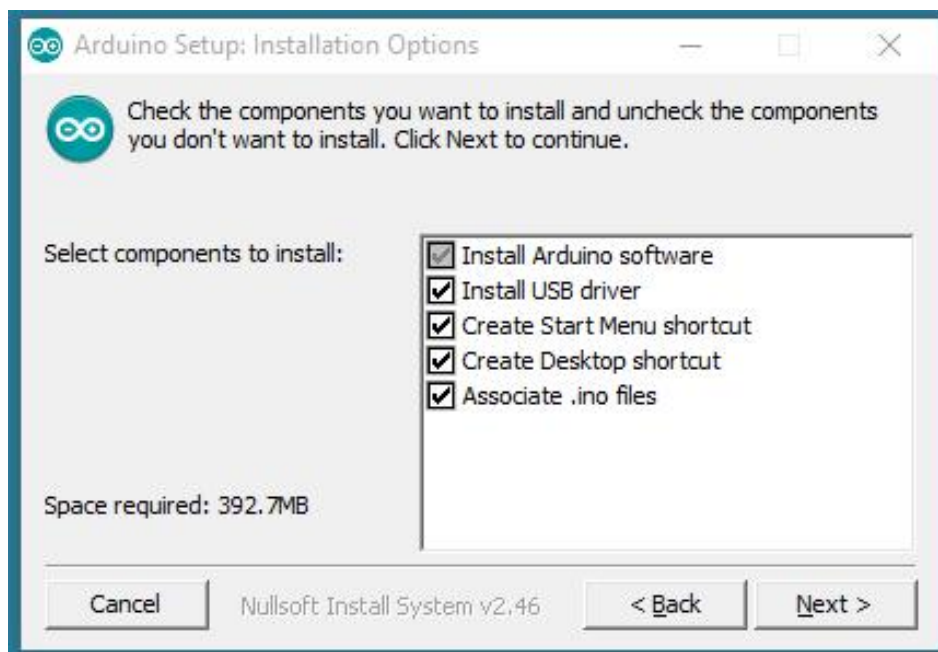
**Linux** 32 bits  
**Linux** 64 bits  
**Linux** ARM 32 bits  
**Linux** ARM 64 bits

[Release Notes](#)  
[Source Code](#)  
[Checksums \(sha512\)](#)

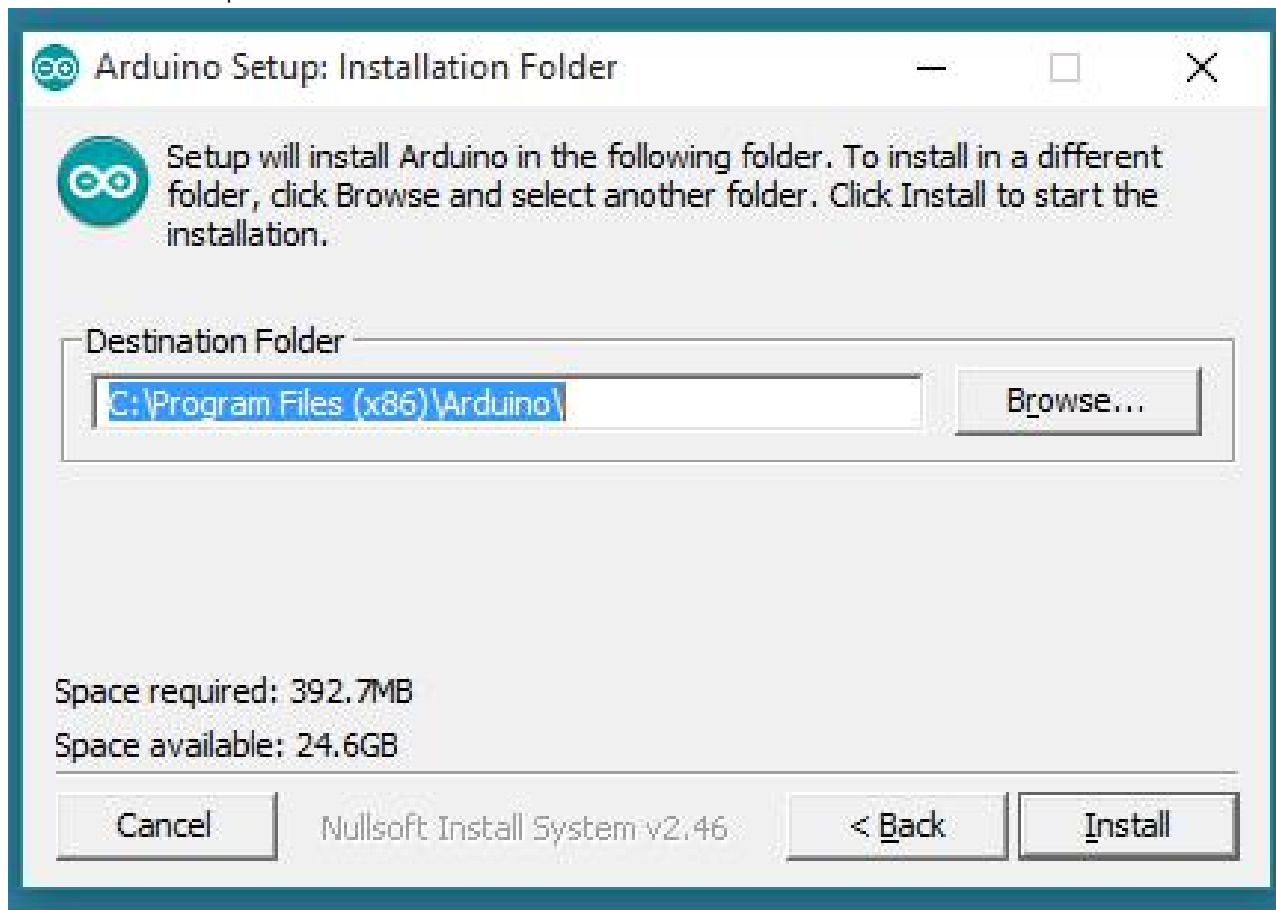
## Previous Releases

Download the [previous version](#) of the current release the classic Arduino 1.0.x, or the Arduino 1.5.x Beta version. <https://www.arduino.cc/en/Main/OldSoftwareReleases#previous>

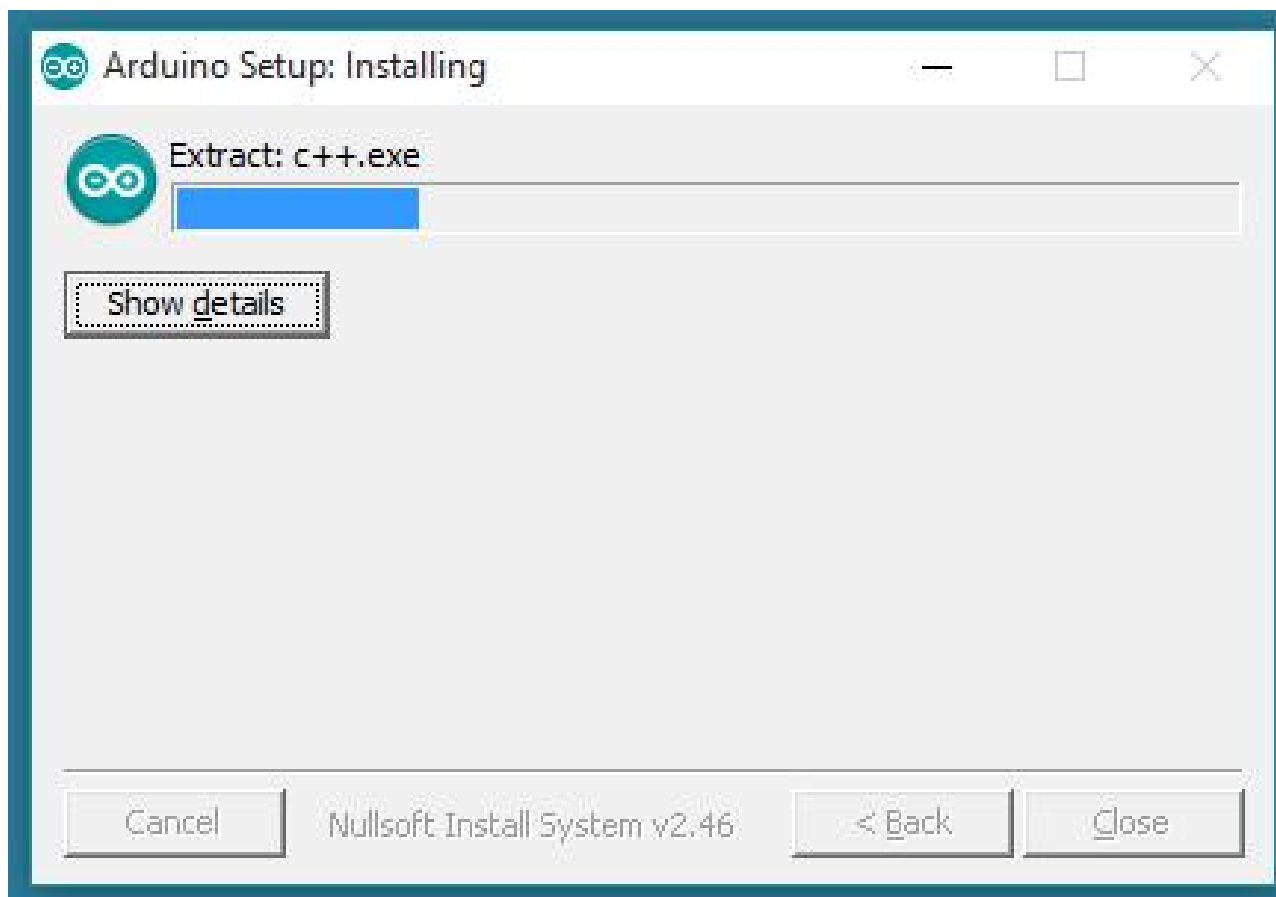
When the download finishes, proceed with the installation and please allow the driver installation process when you get a warning from the operating system.



Choose the components to install

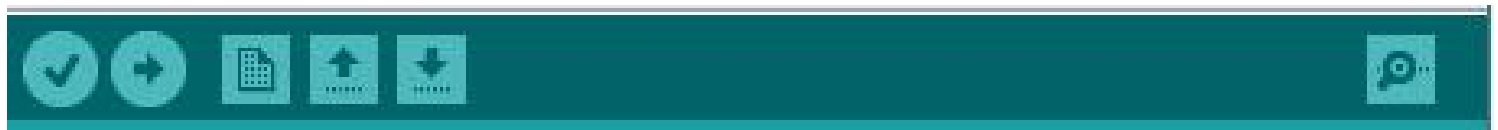








Choose the installation directory (we suggest to keep the default one)



The process will extract and install all the required files to execute properly the Arduino Software (IDE)

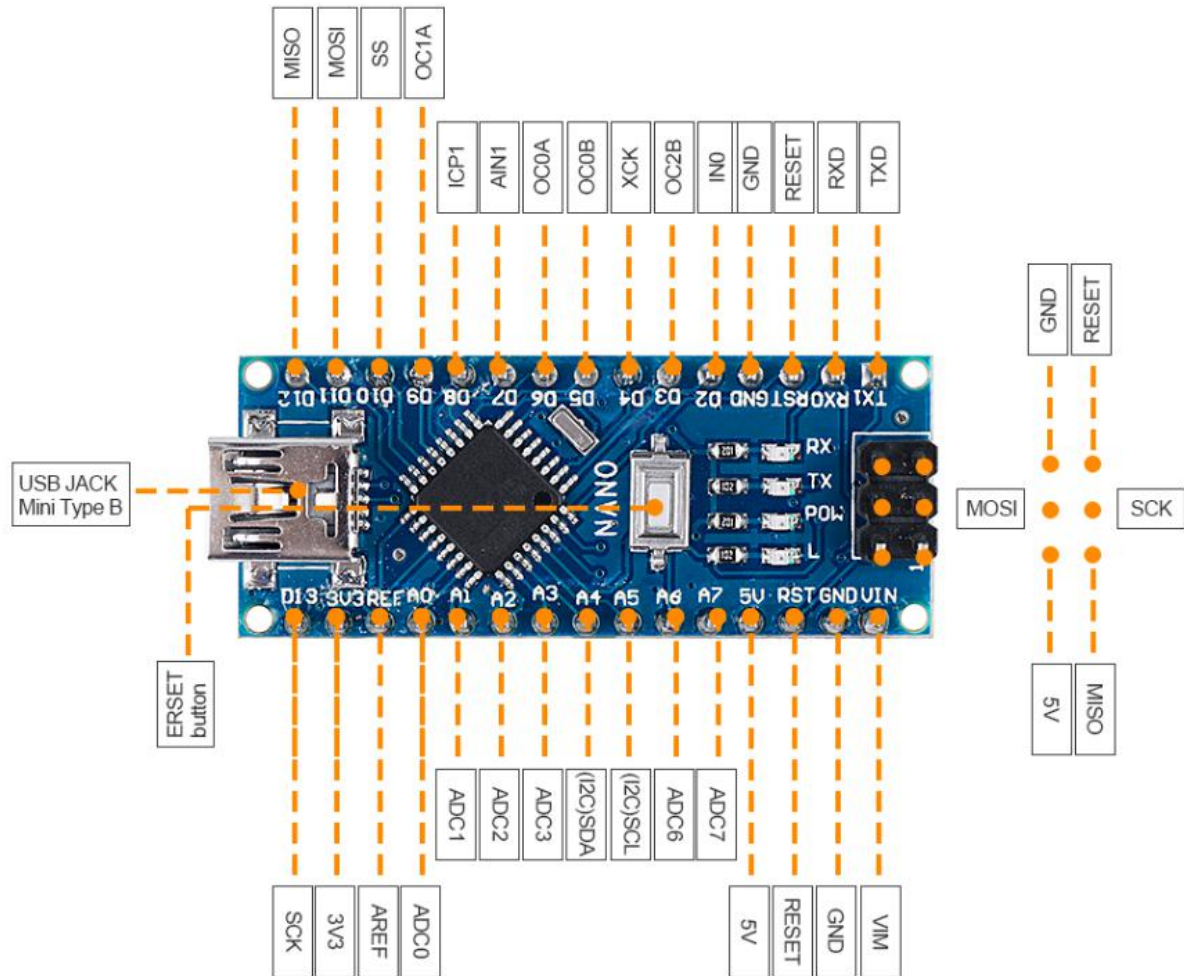
The functions of each button on the Toolbar are listed below:



 <b>Verify/Compile</b>	Check the code for errors
 <b>Upload</b>	Upload the current Sketch to the Arduino
 <b>New</b>	Create a new blank Sketch
 <b>Open</b>	Show a list of Sketches
 <b>Save</b>	Save the current Sketch
 <b>Serial Monitor</b>	Display the serial data being sent from the Arduino

See this page for more arduino programming knowledge: <https://www.arduino.cc/reference/en>

# About the Nano Board



## 1、 Overview:

COKOINO NANO Board is an open source software and hardware controller board, and there is no much change compared to the official Arduino Nano. The main control chip of COKOINO NANO Board is still ATmega328P, and the serial communication chip is CH340.

## 2、 Specification:

Microcontroller: ATmega328P  
Architecture: AVR  
Operating Voltage: 5 V  
Flash Memory: 32 KB of which 2 KB used by bootloader  
SRAM: 2 KB  
Clock Speed: 16 MHz  
Analog IN Pins: 8 (A0~A7)  
EEPROM: 1 KB  
DC Current per I/O Pins: 40 mA (I/O Pins)  
Input Voltage: DC6-12 V (recommended DC7-9V)  
Digital I/O Pins: 22 (6 of which are PWM)  
PWM Output: 6 (D3, D5, D6, D9, D10, D11)  
Power Consumption: 19 mA  
Bootloader: nano

---

## 3、Description:

### Power

The Arduino Nano can be powered via the Mini-B USB connection, 6-12V unregulated external power supply (pin 30), or 5V regulated external power supply (pin 27). The power source is automatically selected to the highest voltage source.

### Memory

The ATmega328 has 32 KB, (also with 2 KB used for the bootloader. The ATmega328 has 2 KB of SRAM and 1 KB of EEPROM.

### Input and Output

Each of the 14 digital pins on the Nano 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 a maximum of 40 mA and has an internal pull-up resistor (disconnected by default) of 20-50 kOhms. 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 FTDI 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, which, although provided by the underlying hardware, is not currently included in the Arduino language.

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

The Nano has 8 analog inputs, 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 `analogReference()` function. Analog pins 6 and 7 cannot be used as digital pins. Additionally, some pins have specialized functionality:

**I2C:** A4 (SDA) and A5 (SCL). Support I2C (TWI) communication using the Wire library (documentation on the Wiring website).

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

The Arduino Nano has a number of facilities for communicating with a computer, another Arduino, or other microcontrollers. The ATmega328 provide UART TTL (5V) serial communication, which is available on digital pins 0 (RX) and 1 (TX). An FTDI FT232RL on the board channels this serial communication over USB and the FTDI drivers (included with the Arduino software) provide a virtual com port to software on the computer. The Arduino software includes a serial monitor which allows simple textual data to be sent to and from the Arduino board. The RX and TX LEDs on the board will flash when data is being transmitted via the FTDI chip and USB connection to the computer (but not for serial communication on pins 0 and 1). A SoftwareSerial library allows for serial communication on any of the Nano's digital pins. The ATmega328 also support I2C (TWI) and SPI communication. The Arduino software includes a Wire library to simplify use of the I2C bus. To use the SPI communication, please see ATmega328 datasheet.

### Programming

The Arduino Nano can be programmed with the Arduino software (<https://www.arduino.cc/en/Main/Software>). Select "Arduino Duemilanove or Nano w/ ATmega328" from the Tools > Board menu (according to the microcontroller on your board). The ATmega328 on the Arduino Nano comes preburned with a bootloader that allows you to upload new code to it without the use of an external hardware programmer. It communicates

using the original STK500 protocol. You can also bypass the bootloader and program the microcontroller through the ICSP (In-Circuit Serial Programming) header using Arduino ISP or similar.

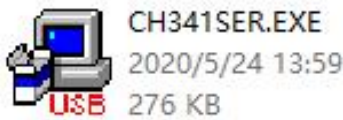
### Automatic (Software) Reset

Rather than requiring a physical press of the reset button before an upload, the Arduino Nano 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 FT232RL 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 uses this capability to allow you to upload code by simply pressing the upload button in the Arduino environment. 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 Nano 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 Nano. 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.

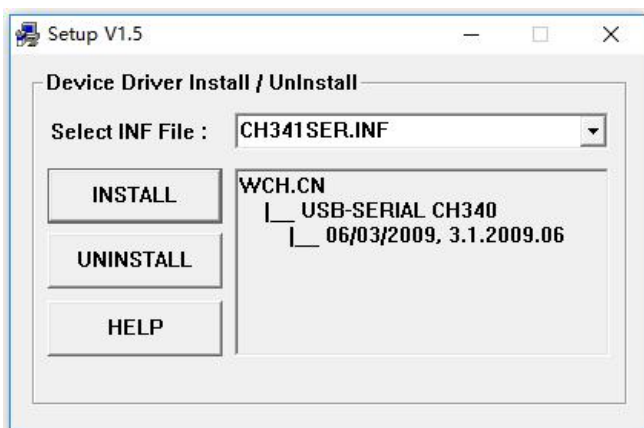
## Install The Board Driver

The main control chip of LK COKOINO NANO Board is ATmega328P, and the serial communication chip is CH340. You can also download CH340 driver from webpage: [http://www.wch-ic.com/downloads/CH341SER\\_EXE.html](http://www.wch-ic.com/downloads/CH341SER_EXE.html)

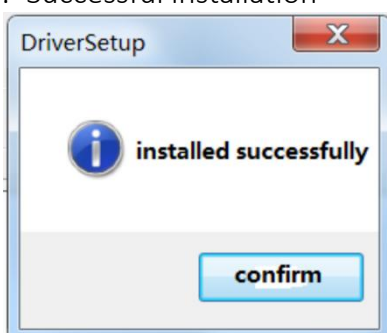
1. Double-click to open the installer. As shown below:



2. Pop up a install interface, click the top button to install the driver

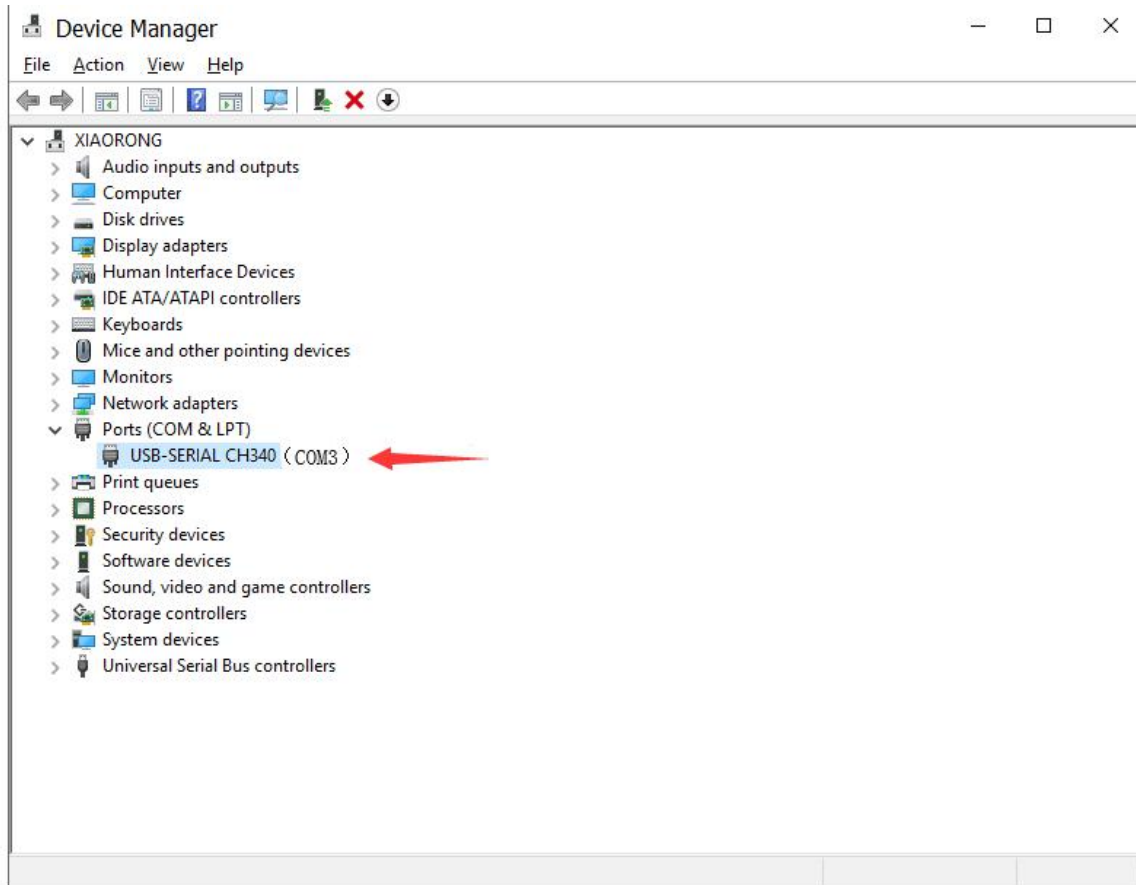


3. Successful installation



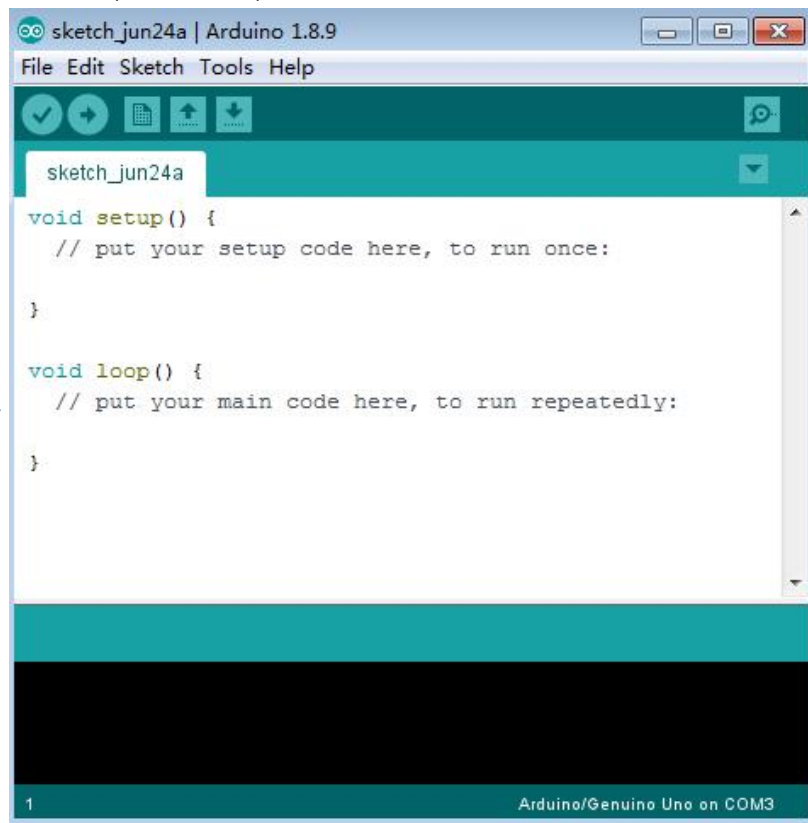


4. If the driver is successfully installed, when the nano motherboard is connected to the computer through the USB cable, under the path "Computer" -> "Properties" -> "Device manager", you can find CH340 COM3, as shown below.



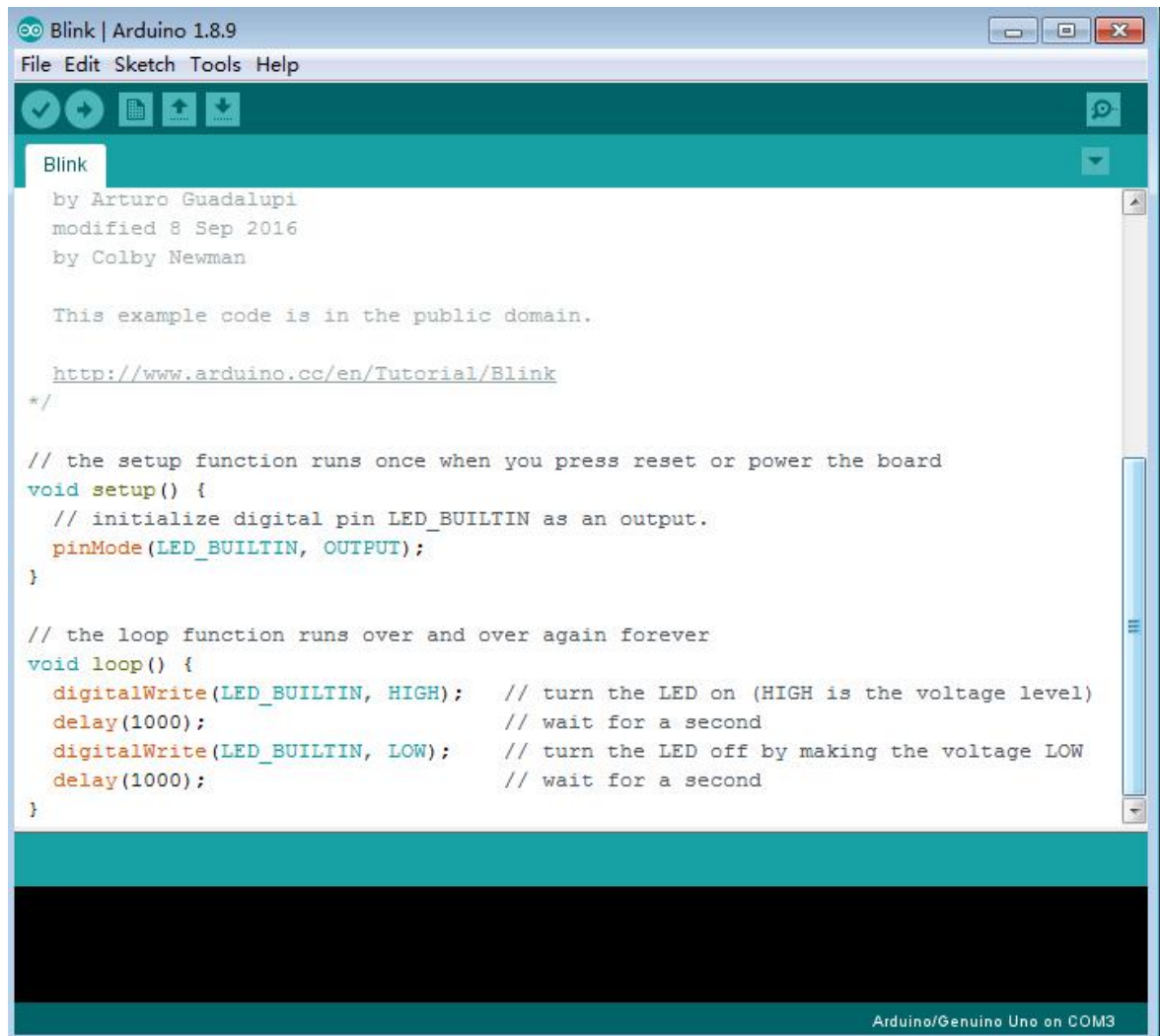
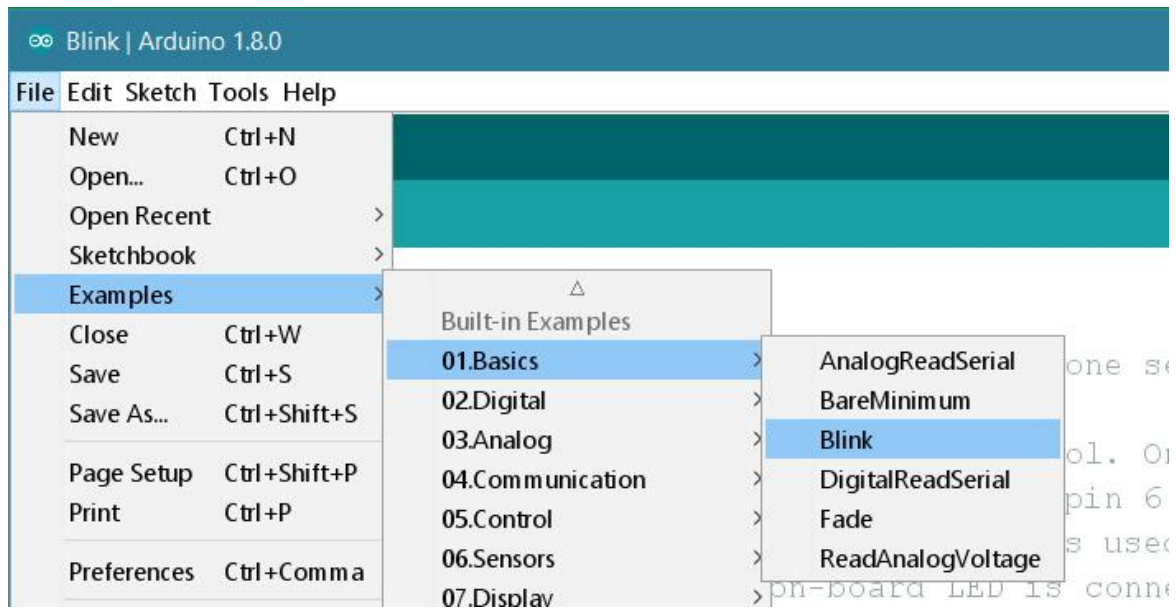
## Upload your first sketch

Double-click the computer desktop icon to open the arduino IDE, as shown below:



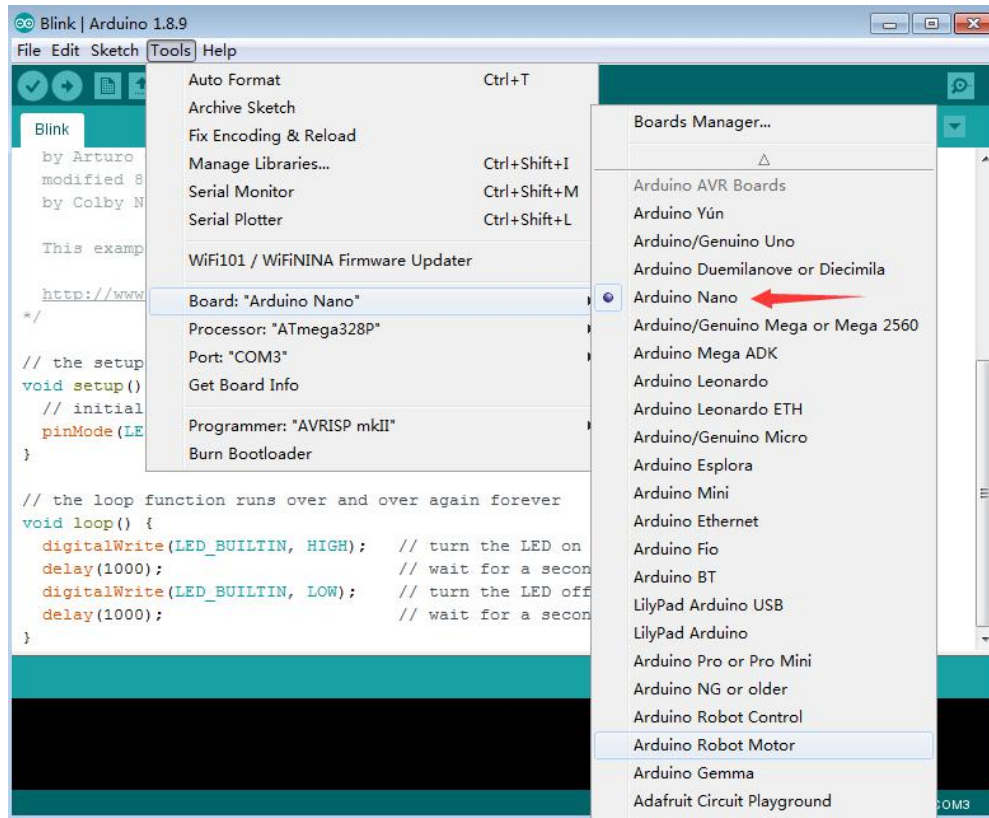


Open the LED blink example sketch: File > Examples > 01.Basics > Blink.



## Select your board type and port

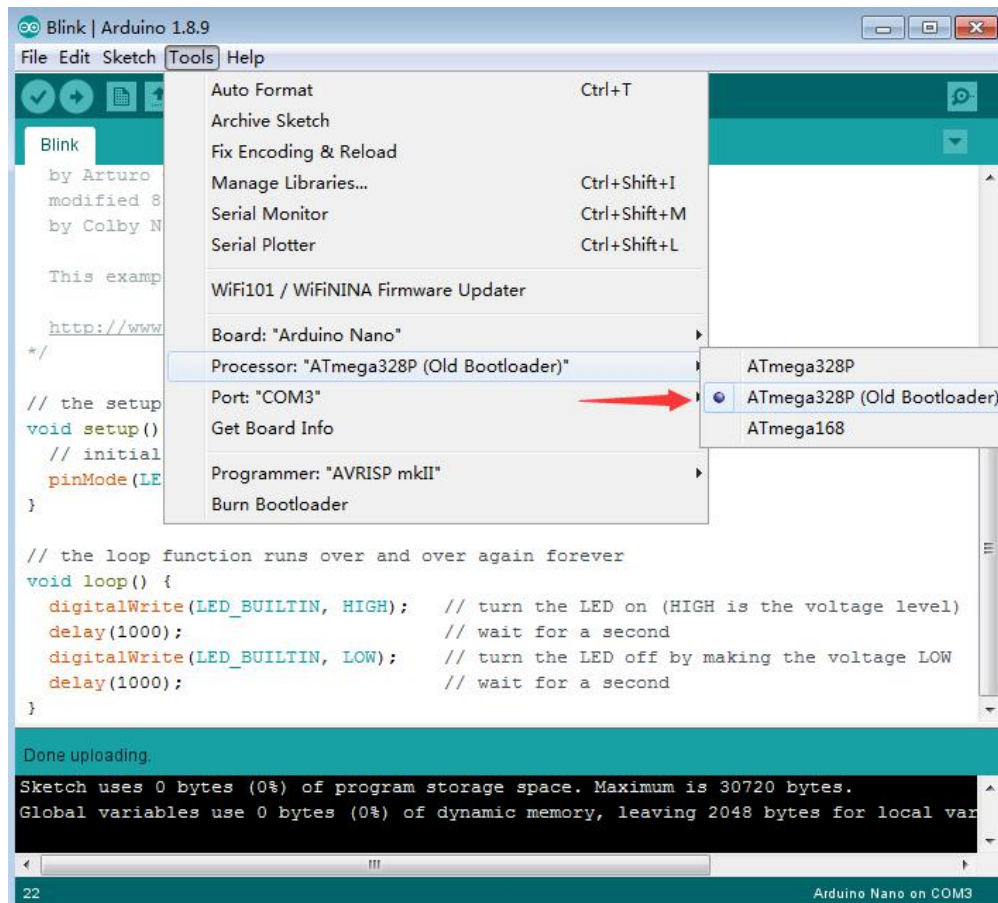
Select the board type, click "tools"--->"Board"--->"Arduino Nano", as shown below:



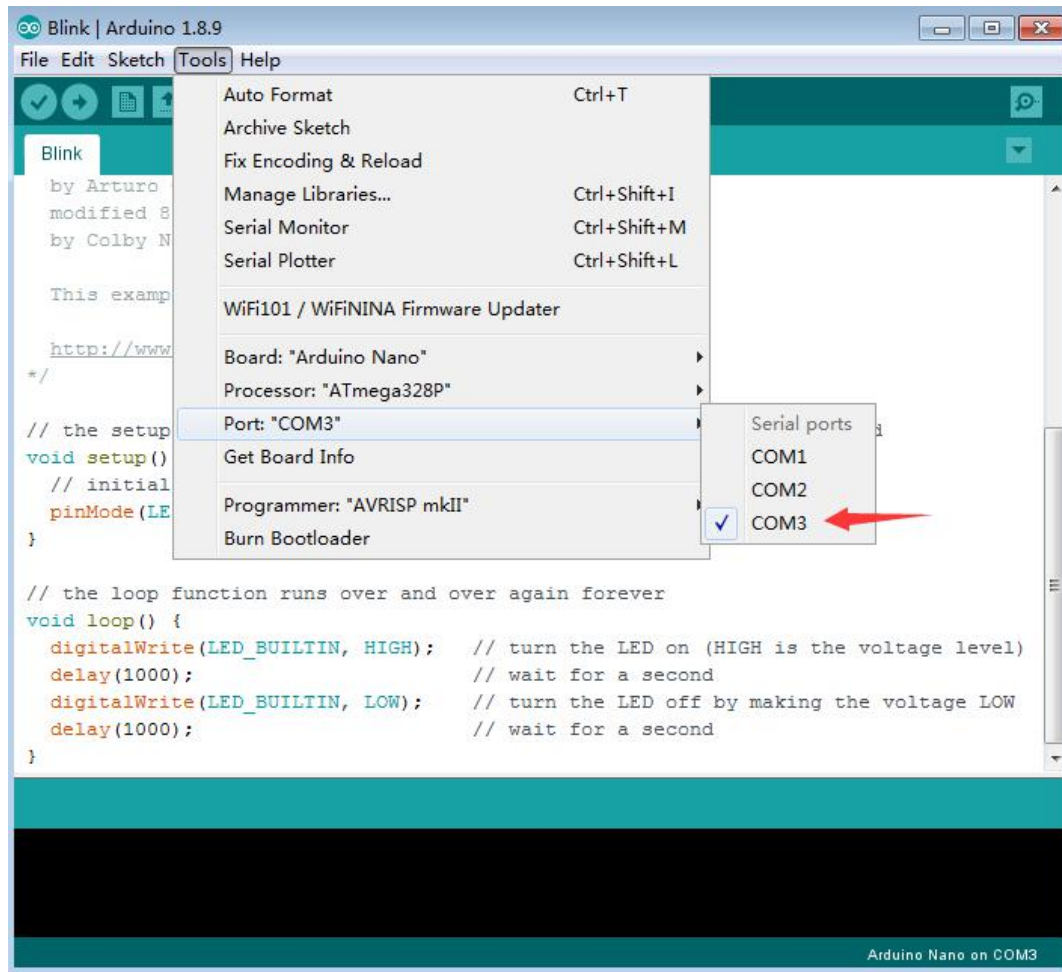
## Nano processor selection:

You will find this option in the Arduino IDE above version 1.8.5. If the nano manufacturer burns the old version of the bootloader to the nano motherboard, you must select "ATmega328P (Old Bootloader)". If the nano board is burned with a new version of the bootloader, you must select "ATmega328P".

Here we choose the old bootloader, as shown in below:



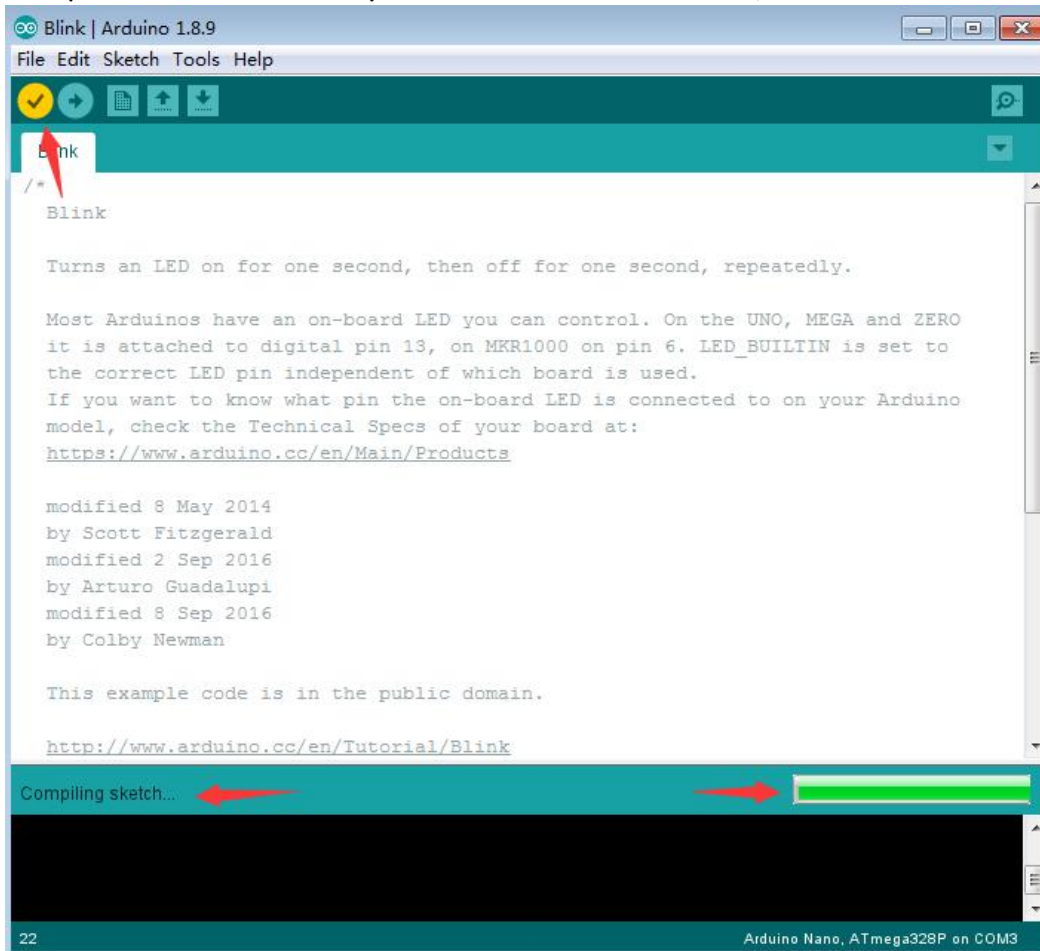
Select the port, "tools"----> "Port"----> "COM port of nano motherboard", as shown below:



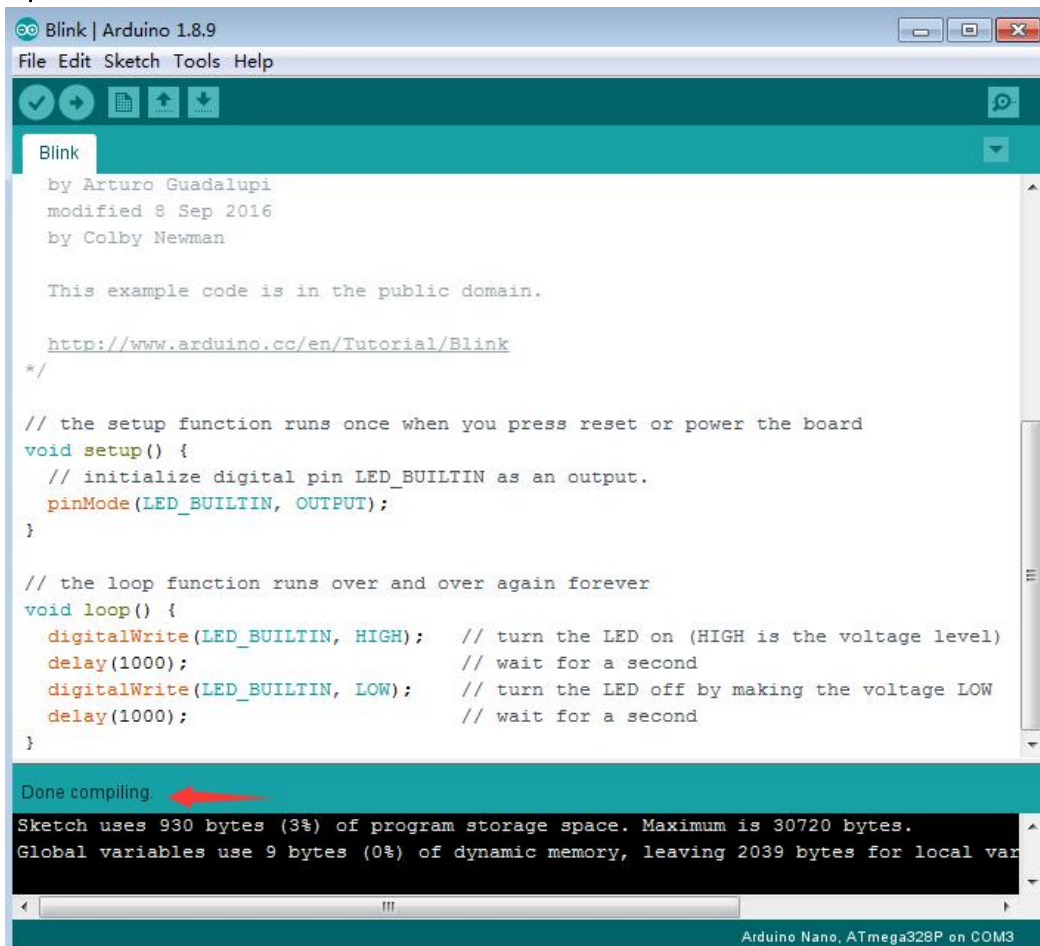
Connect the PC to the nano motherboard via the USB cable



Compile the code and verify the correctness of the code, as shown below:



Upload the code



As a result, the L-label LED of the nano motherboard flashes once every 1 second, as shown below:

