

# Getting Started with Arduino and Nano

## I 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



## II Setting up a programming environment

### 1. Install the Arduino IDE

#### 1.1 Install in Windows System

See: <https://www.arduino.cc/en/Guide/Windows>

## 1.2 Install in macOS System

See: <https://www.arduino.cc/en/Guide/macOS>

## 1.3 Install in Linux System

See: <https://www.arduino.cc/en/Guide/Linux>

## 1.4 Portable IDE (Windows and Linux)

See: <https://www.arduino.cc/en/Guide/PortableIDE>

## 1.5 ChromeOS (Arduino Create App) in the Chrome Web Store

See: <https://chrome.google.com/webstore/detail/arduino-create/dcgicpihgkmccjigalccipmjlnjopdfe>

Note that the free download package:



## 2. Use Arduino IED

Please refer to the Arduino IDE operation interface and tutorial:  
<https://www.arduino.cc/en/Guide/Environment>

## 3. PC install CH340 USB to serial port driver

A USB to serial port chip made in China is used in our Nano motherboard. You can select

the installation package you need according to this link and install:

[http://www.wch-ic.com/downloads/CH341SER\\_EXE.html](http://www.wch-ic.com/downloads/CH341SER_EXE.html)

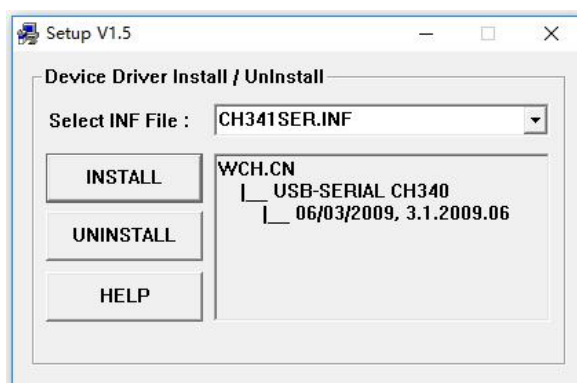
relation files	
file name	file content
CH341SER.ZIP	CH340/CH341 USB to serial port Windows driver, includes DLL dynamic library and non-standard baud rate settings and other instructions. Supports 32/64-bit Windows 10/8.1/8/7/VISTA/XP, Server 2016/2012/2008/2003, 2000/ME/98, Microsoft WHQL Certified, supports USB to 3 and 9 wire serial ports.
CH341SER_LINUX.ZIP	CH340/CH341 USB to serial port LINUX driver, supports 32/64-bit operation system.
CH341SER_MAC.ZIP	CH340/CH341 USB to serial port MAC OS driver, supports 32/64-bit operation system, contains instructions for use.
CH341SER_ANDROID.ZIP	CH340/CH341 USB to serial port Android free drive application library, for Android OS 3.1 and above version which supports USB Host mode already, no need to load Android kernel driver, no root privileges. Contains apk, lib library file (Java Driver), App Demo Example (USB to UART Demo SDK).

## Windows system installation example:

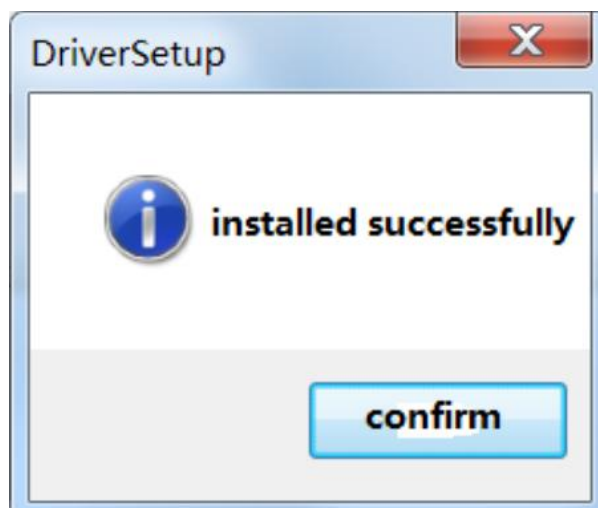
3.1 After unzipping, double-click to open the installer. As shown below:



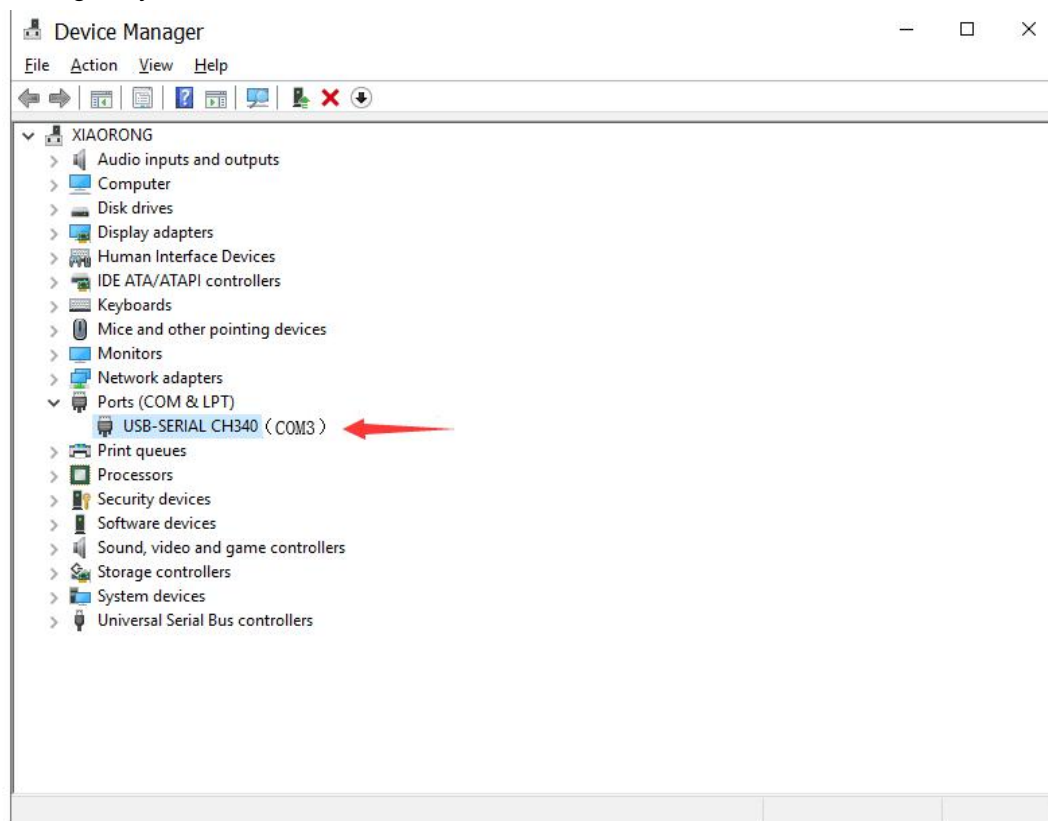
3.2 Pop up a install interface, click the top button to install the driver



3.3 Successful installation



3.4 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.



## 4. More information on how to use Arduino IDE

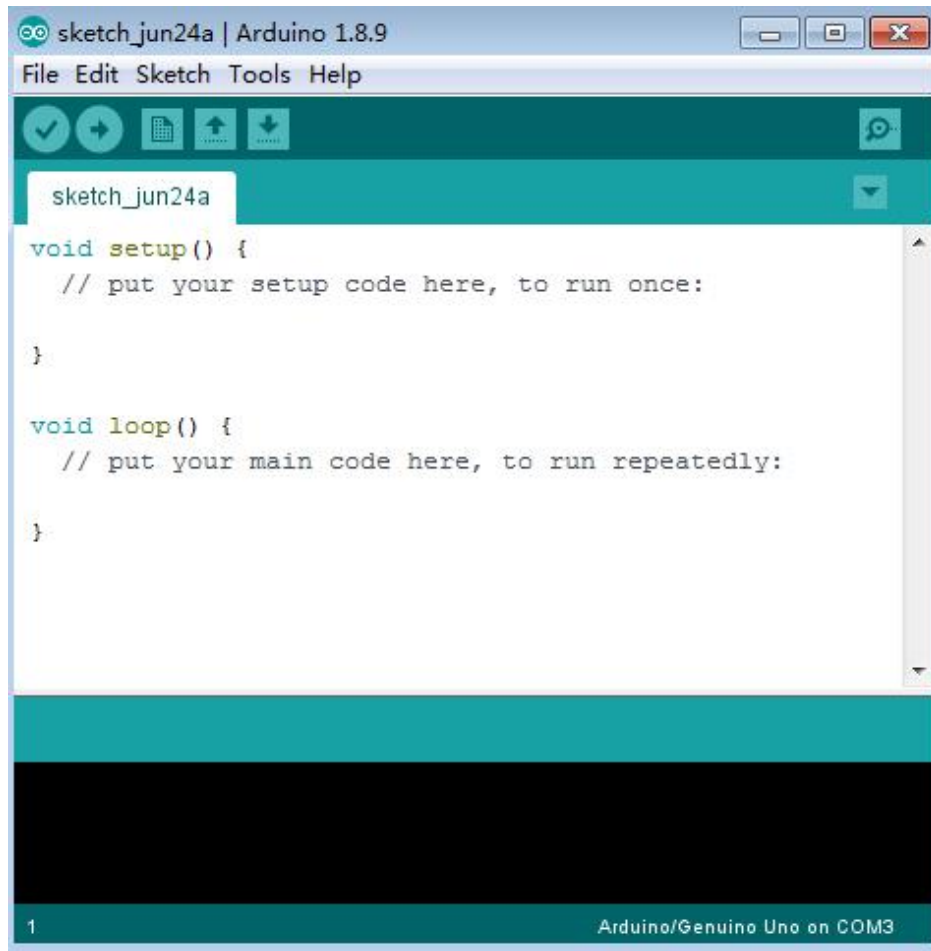
Arduino official tutorials: <https://www.arduino.cc/en/Tutorial/HomePage>

Language Reference: <https://www.arduino.cc/reference/en/>

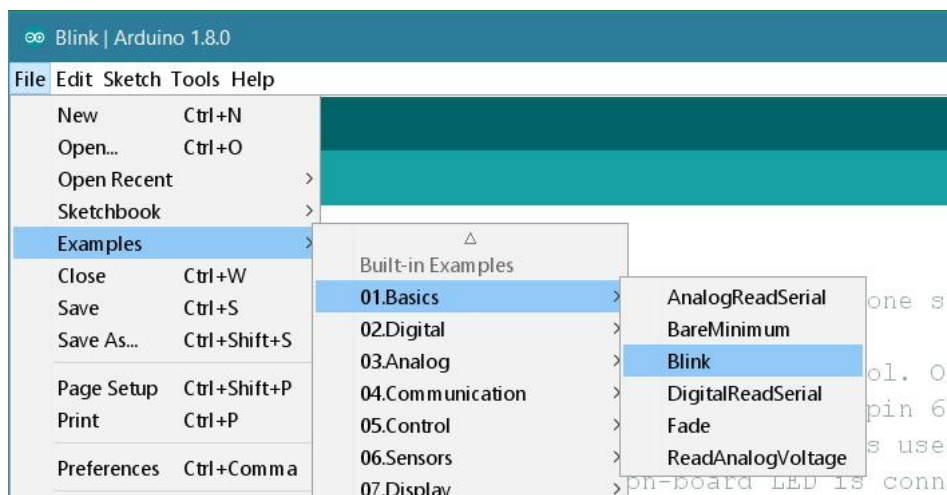
## 5. The first example program

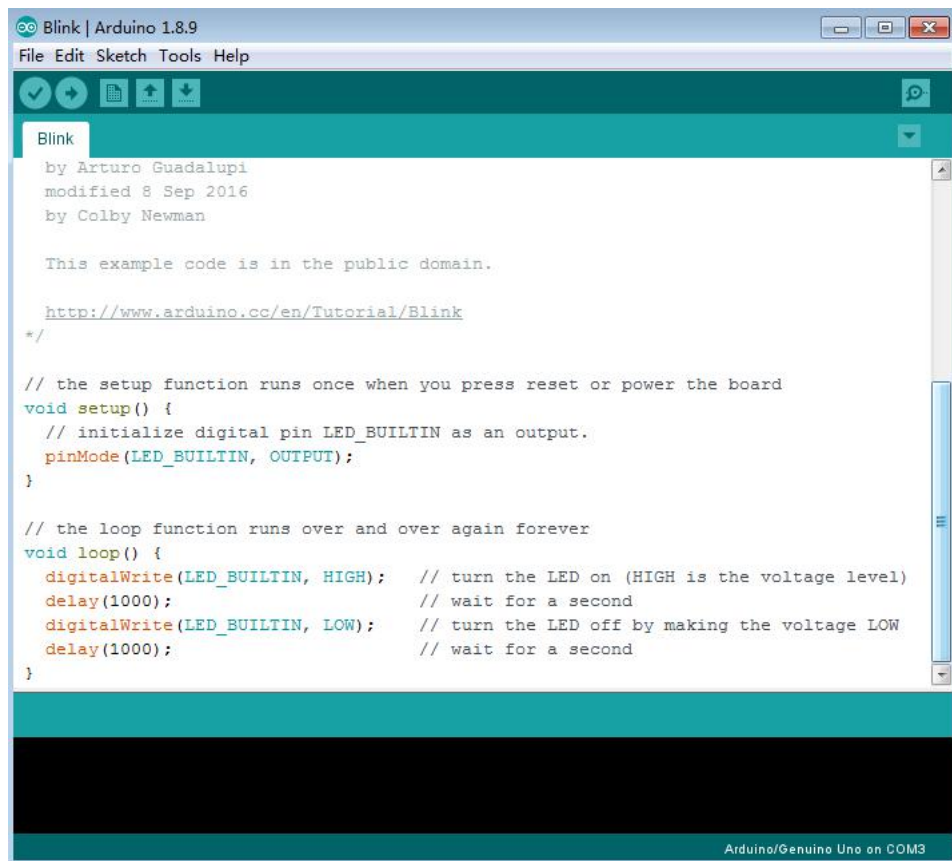


5.1 Double-click the computer desktop icon to open the arduino IDE, as shown below:

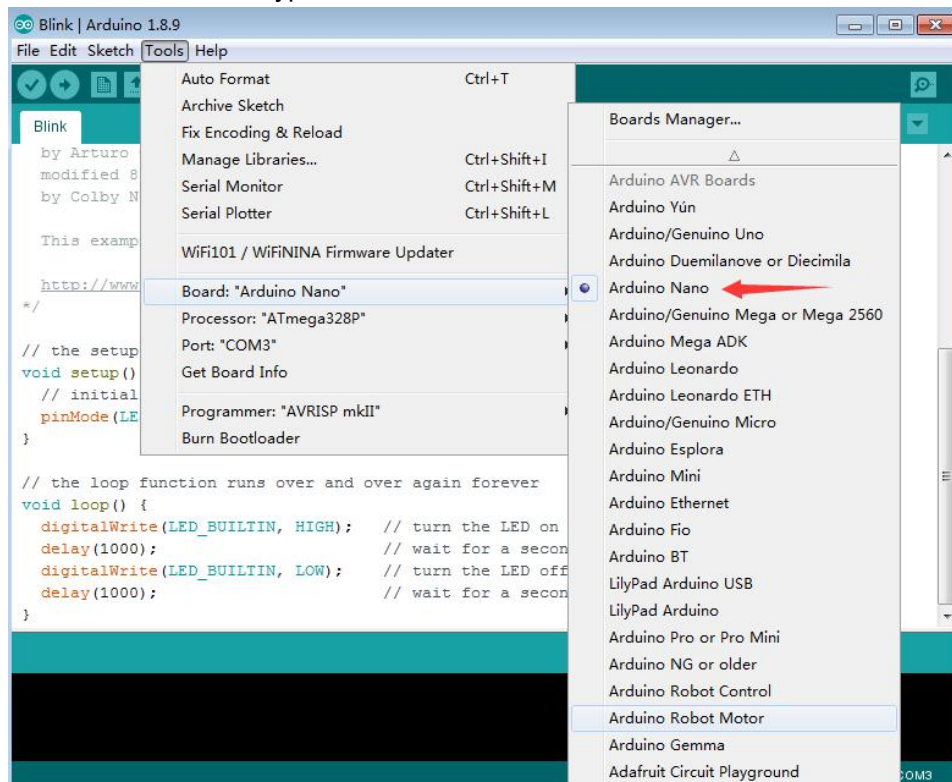


5.2 Open the LED blink example sketch: File > Examples > 01.Basics > Blink.





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



5.4 Nano processor selection:

You will find this option in the Arduino IDE above version 1.8.5.

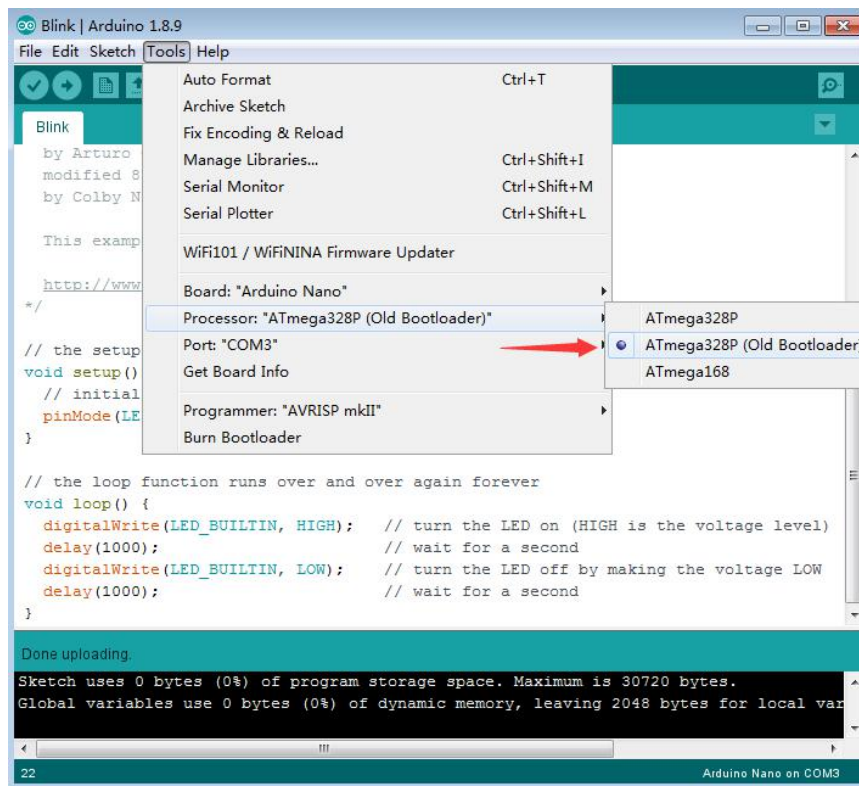
If the nano manufacturer burns the old version of the bootloader to the nano motherboard,



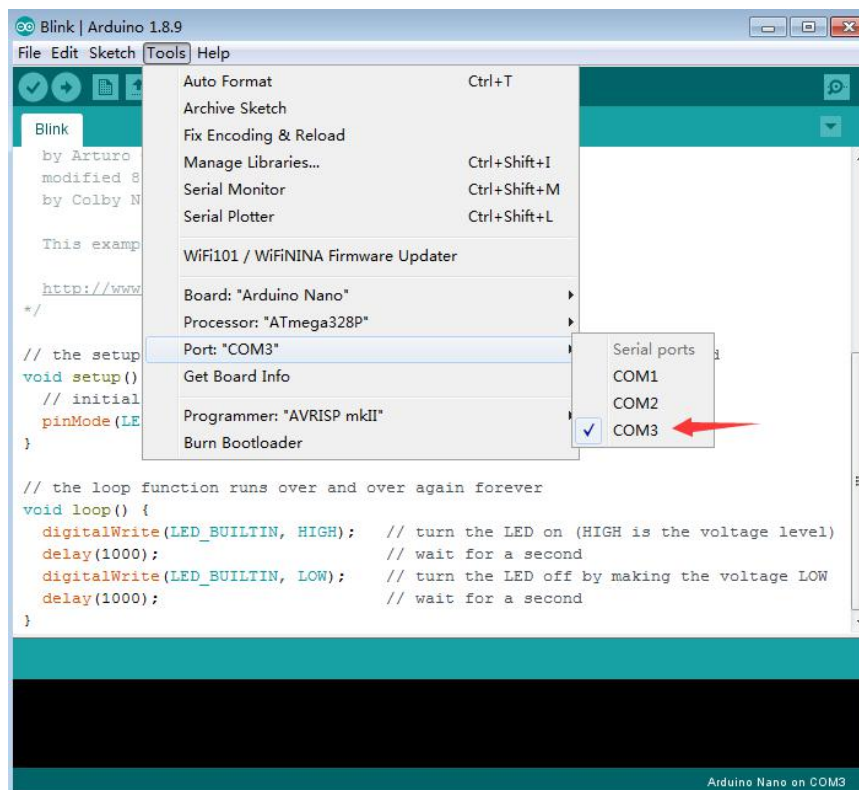
you must select "ATmega328P (Old Bootloader)";

If the nano board is burned with a new version of the bootloader, you must select "ATmega328P".

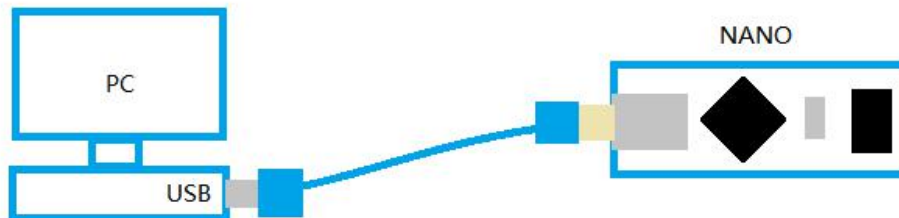
Here we choose the old bootloader, as shown below :



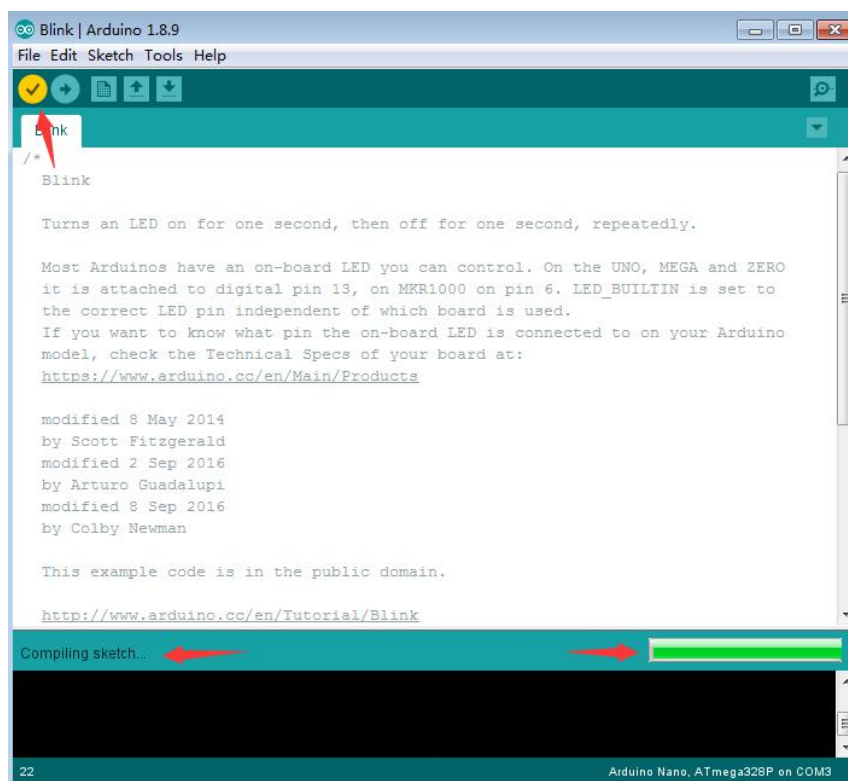
5.5 Select the port, "tools"----> "Port"----> "COM port of nano motherboard", as shown below:



## 5.6 Connect the PC to the nano motherboard via the USB cable

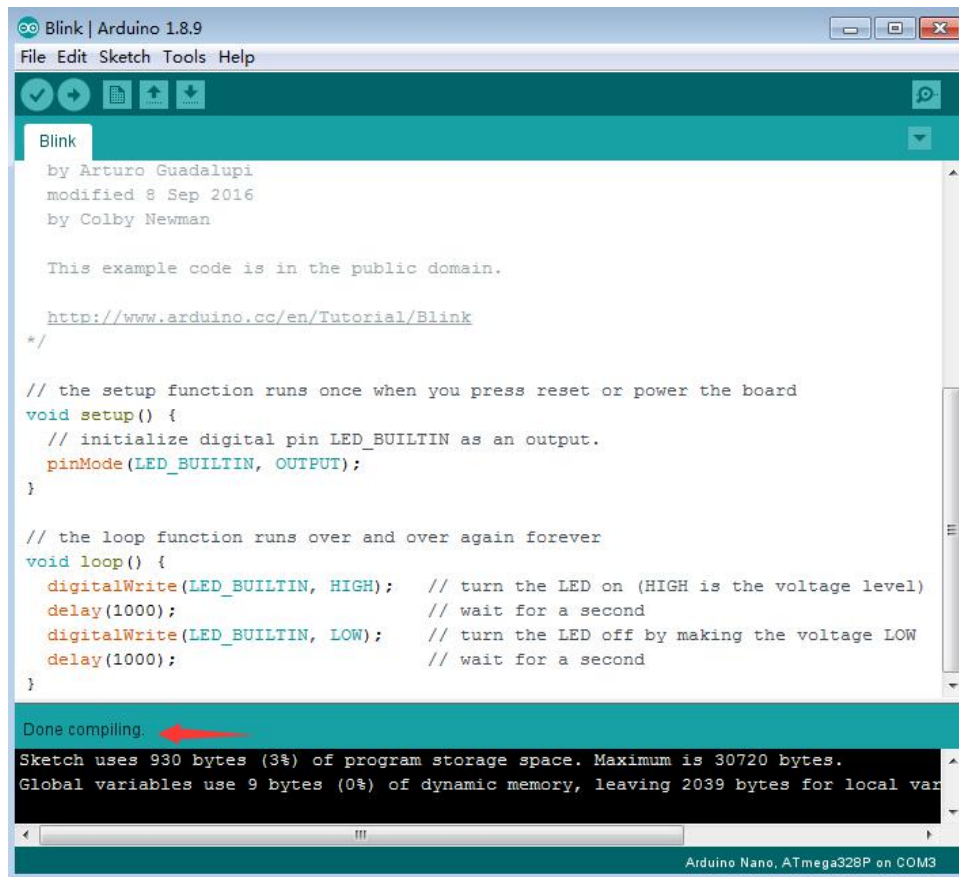


## 5.7 Compile the code and verify the correctness of the code, as shown below:





## 5.8 Upload the code



```
Blink
by Arturo Guadalupi
modified 8 Sep 2016
by Colby Newman

This example code is in the public domain.

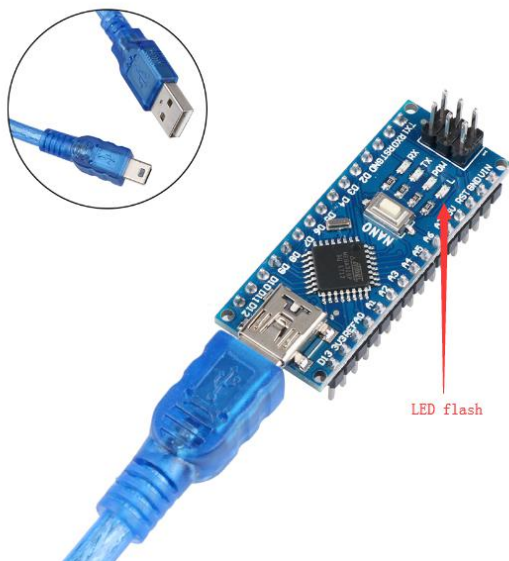
http://www.arduino.cc/en/Tutorial/Blink
*/

// the setup function runs once when you press reset or power the board
void setup() {
  // initialize digital pin LED_BUILTIN as an output.
  pinMode(LED_BUILTIN, OUTPUT);
}

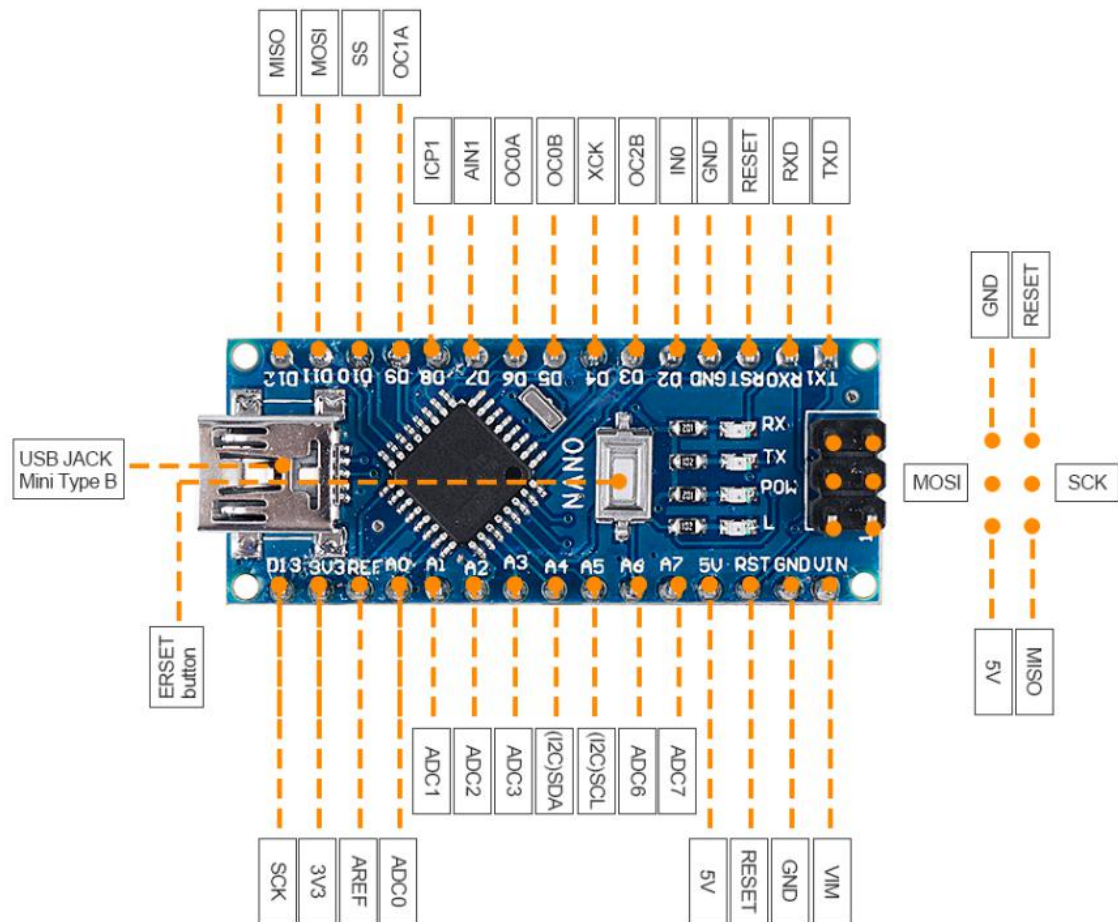
// the loop function runs over and over again forever
void loop() {
  digitalWrite(LED_BUILTIN, HIGH);  // turn the LED on (HIGH is the voltage level)
  delay(1000);                      // wait for a second
  digitalWrite(LED_BUILTIN, LOW);   // turn the LED off by making the voltage LOW
  delay(1000);                      // wait for a second
}

Done compiling.
Sketch uses 930 bytes (3%) of program storage space. Maximum is 30720 bytes.
Global variables use 9 bytes (0%) of dynamic memory, leaving 2039 bytes for local variables.
Arduino Nano, ATmega328P on COM3
```

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



### III 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.

Company website address: <http://cokoino.com/>

Github : <https://github.com/Cokoino/>