Lesson 1 - 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

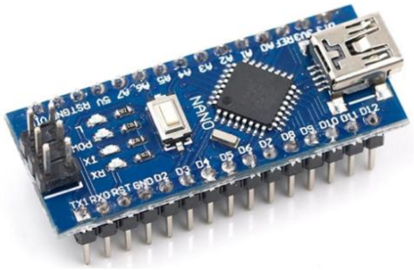


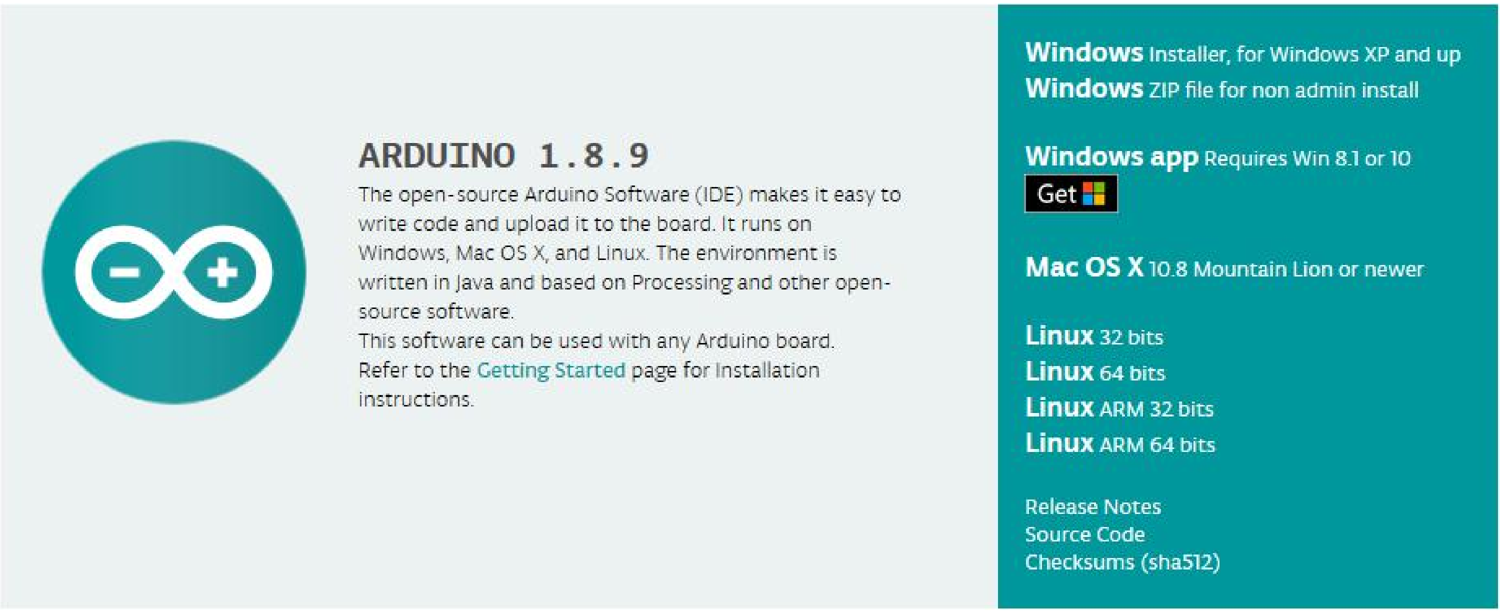
Table of Contents

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.
This software 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

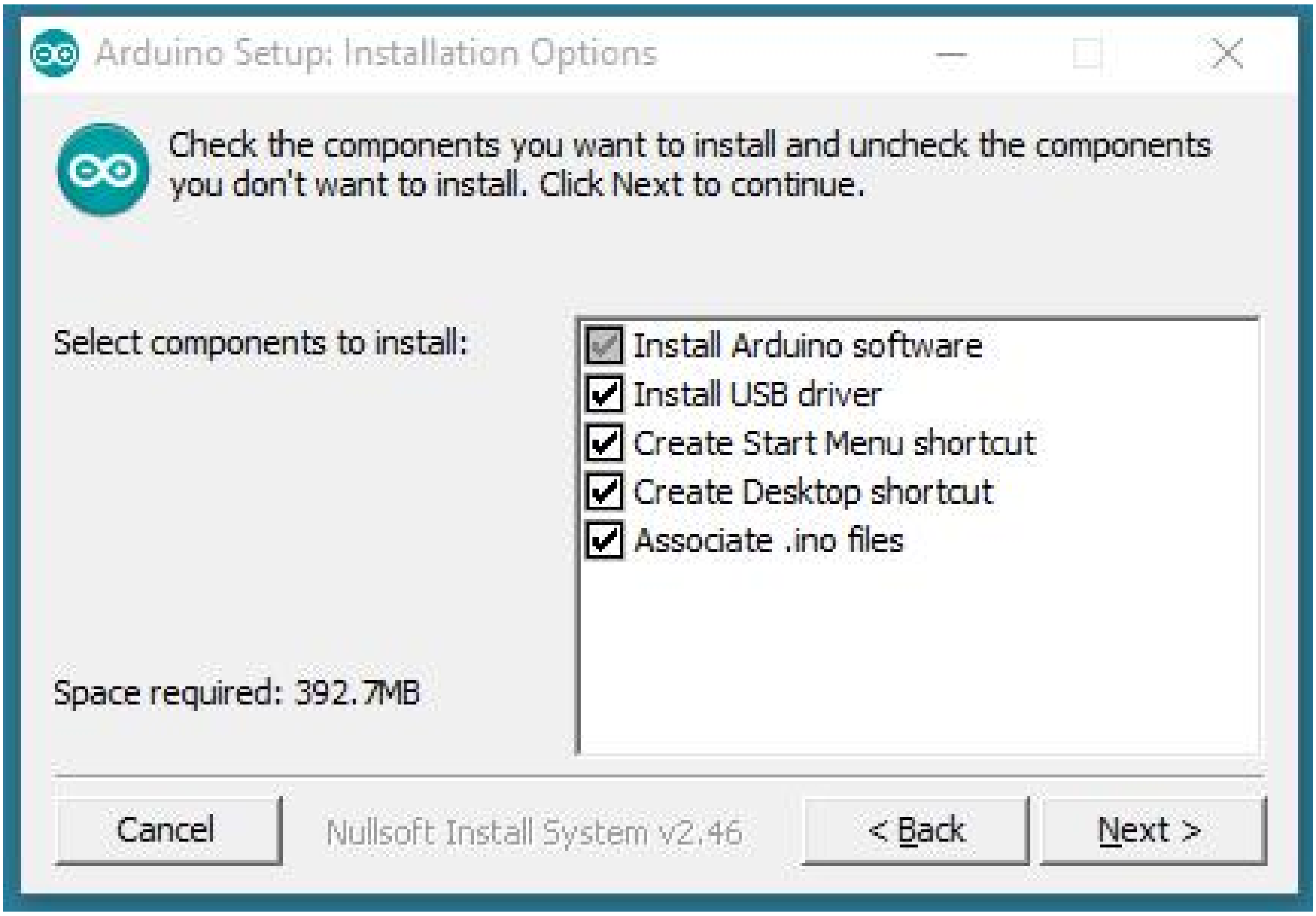
Release Notes
Source Code
Checksums (sha512)

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.


Arduino Setup: Installation Options

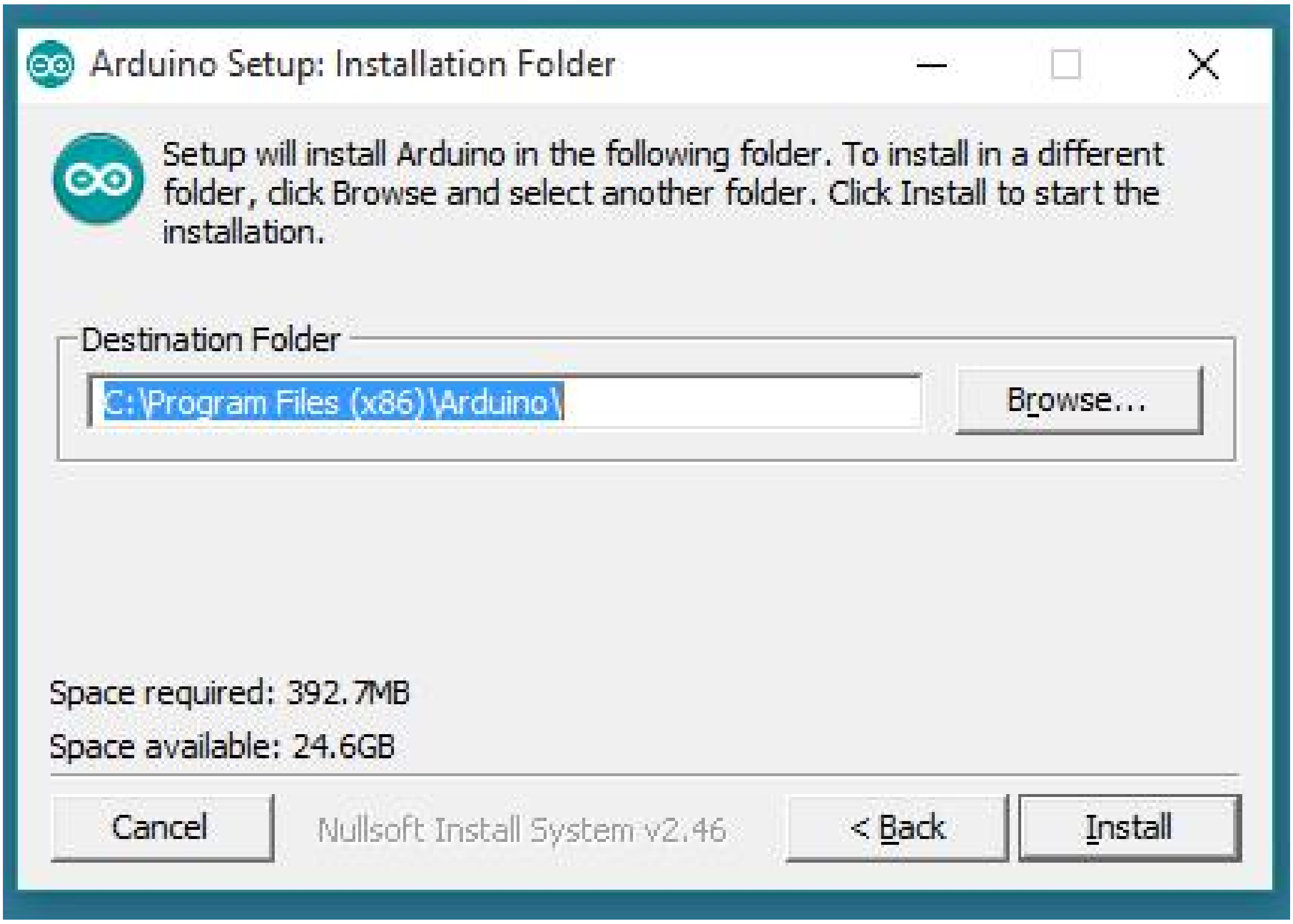
Check the components you want to install and uncheck the components you don't want to install. Click Next to continue.

Select components to install:

- ☑ Install Arduino software
- ☑ Install USB driver
- ☑ Create Start Menu shortcut
- ☑ Create Desktop shortcut
- ☑ Associate .ino files

Space required: 392.7MB

Choose the components to install


Arduino Setup: Installation Folder

Setup will install Arduino in the following folder. To install in a different folder, click Browse and select another folder. Click Install to start the installation.

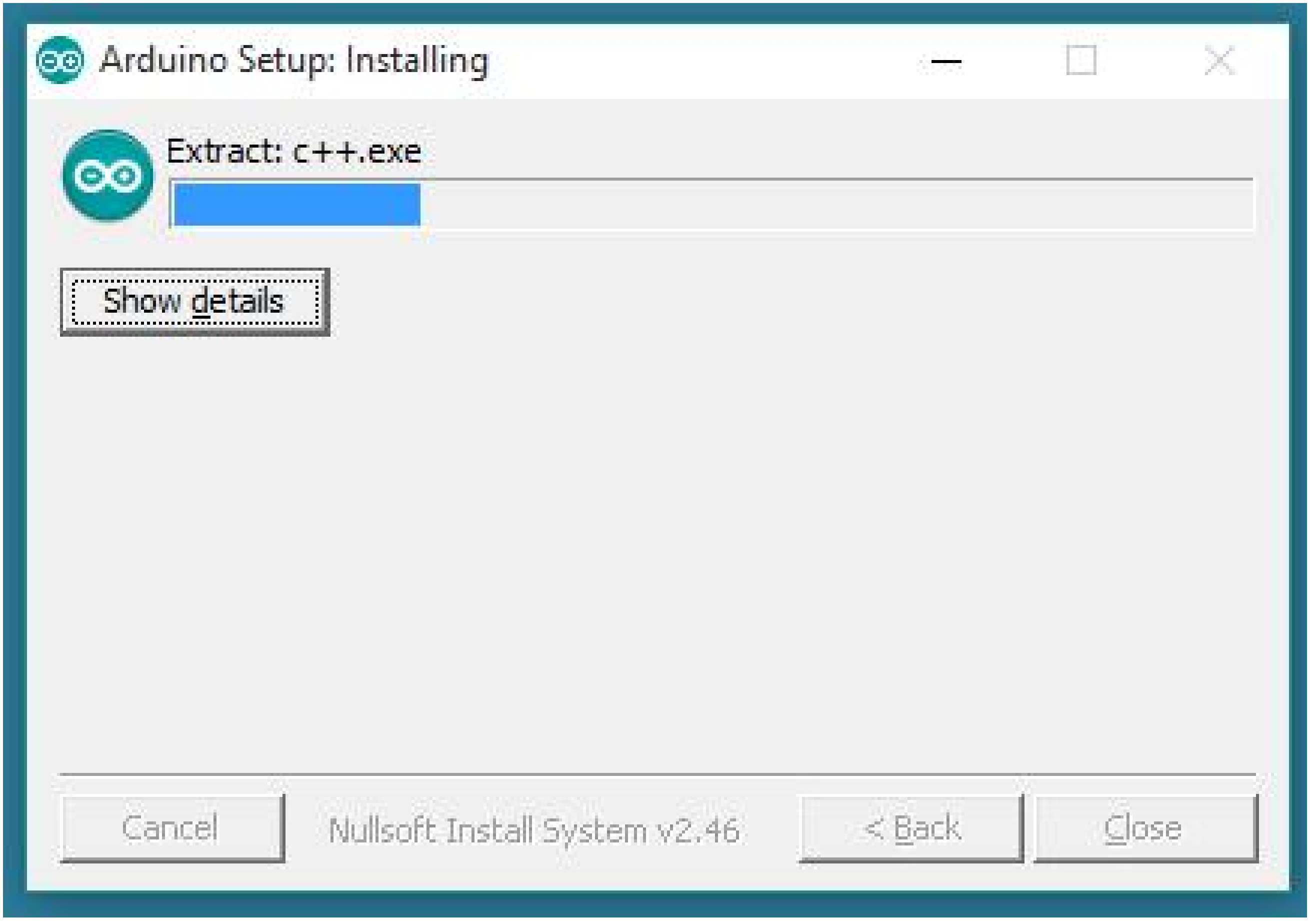
Destination Folder

C:\Program Files (x86)\Arduino\

Space required: 392.7MB

Space available: 24.6GB

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

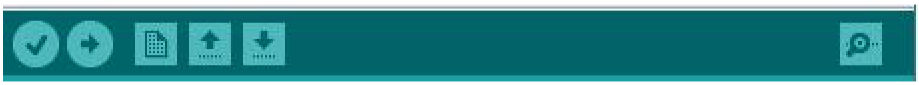


Arduino Setup: Installing

Extract: c++.exe

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

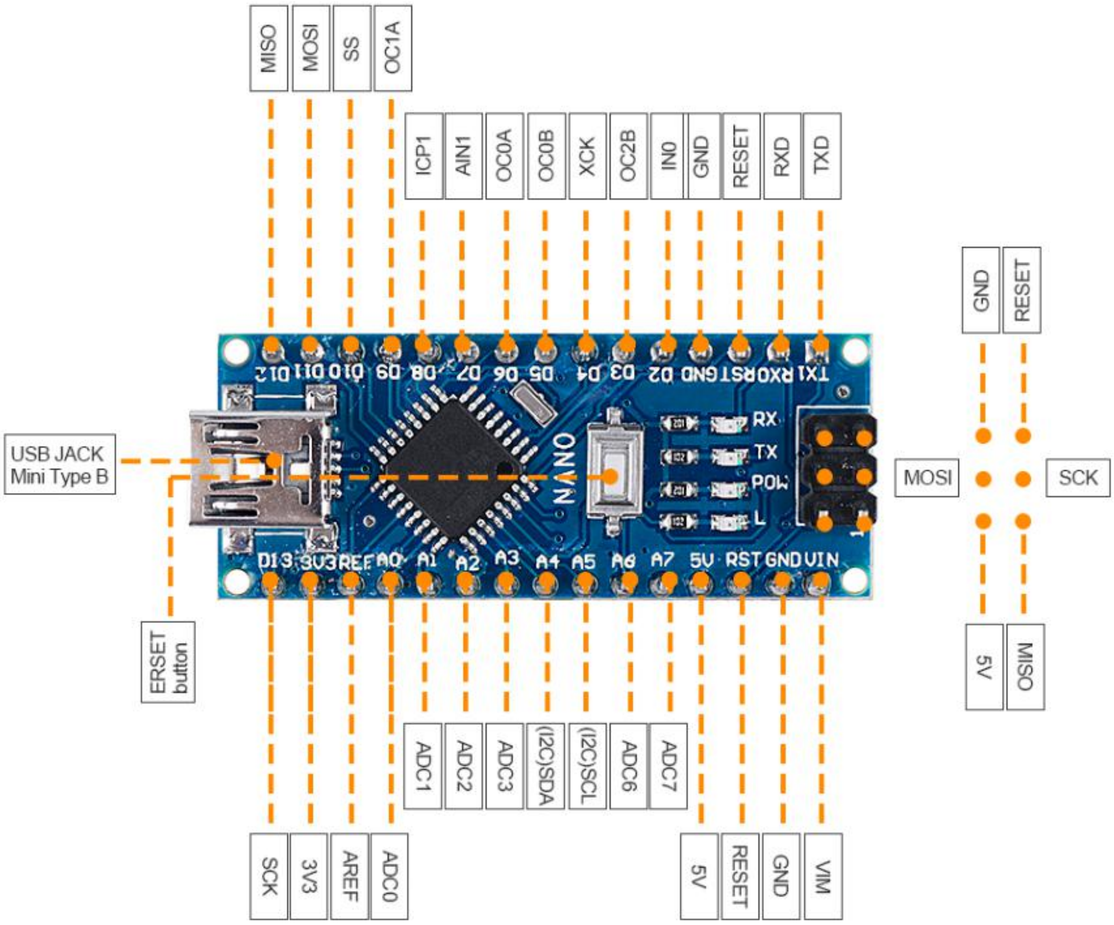
The functions of each button on the Toolbar are listed below:



✓ **Verify/Compile**	Check the code for errors
→ **Upload**	Upload the current Sketch to the Arduino
🗎 **New**	Create a new blank Sketch

Open	Show a list of Sketches
Save	Save the current Sketch
Serial Monitor	Display the serial data being sent from the Arduino

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

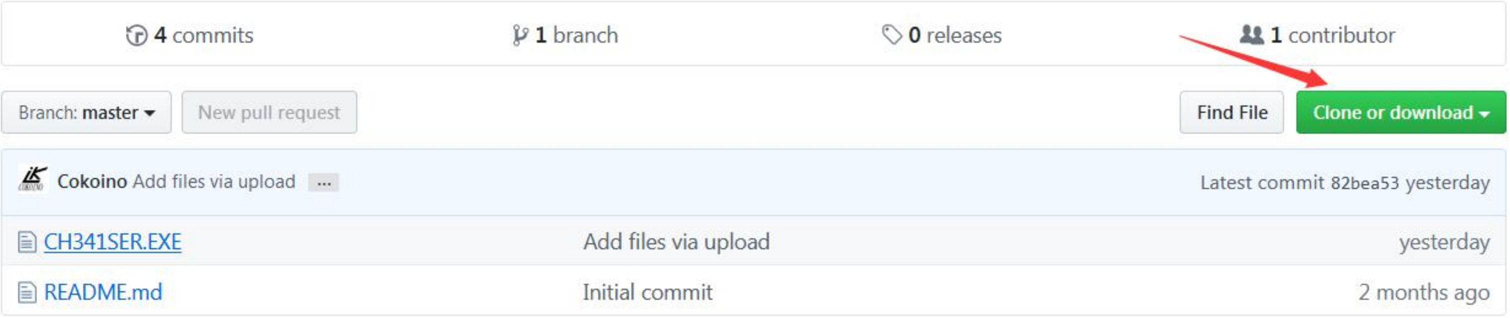
Rather then requiring a physical press of the reset button before an upload, the Arduino Nano is designed in a way that allows it to be reset by software running on a connected computer. One of the hardware flow control lines (DTR) of the FT232RL is connected to the reset line of the ATmega328 via a 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.

CH340 driver can be obtained through browser search.
Or you can also download it from our github webpage: https://github.com/Cokoino/ch340-drive.

1. Open this link in your browser: https://github.com/Cokoino/ch340-drive, and click "Open or download" icon to download the drive.

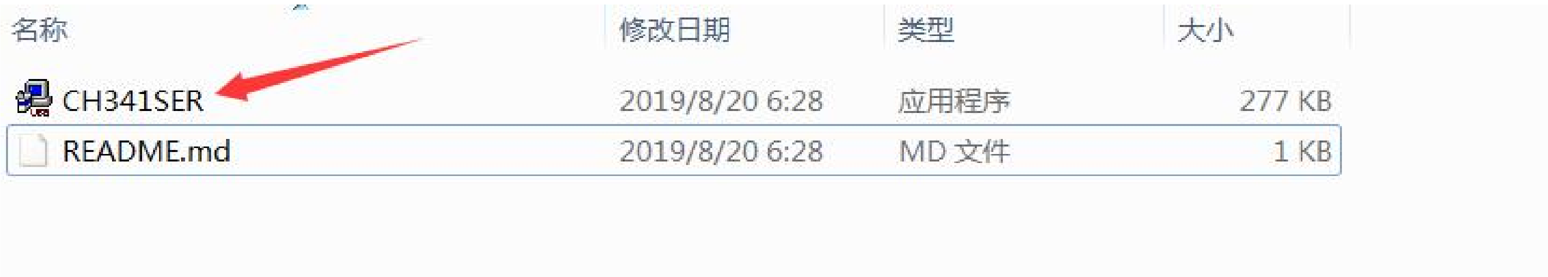


4 commits | 1 branch | 0 releases | 1 contributor

Cokoino Add files via upload ... Latest commit 82bea53 yesterday

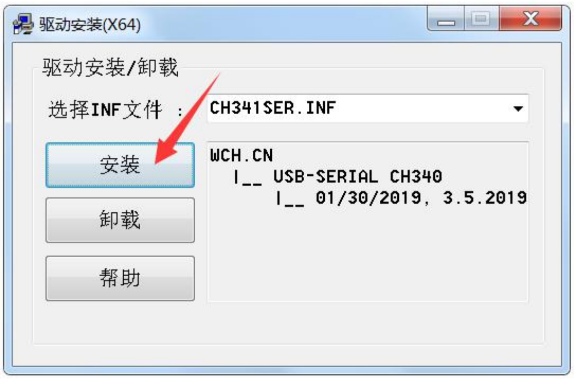
CH341SER.EXE	Add files via upload	yesterday
README.md	Initial commit	2 months ago

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



名称	修改日期	类型	大小
CH341SER	2019/8/20 6:28	应用程序	277 KB
README.md	2019/8/20 6:28	MD 文件	1 KB

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



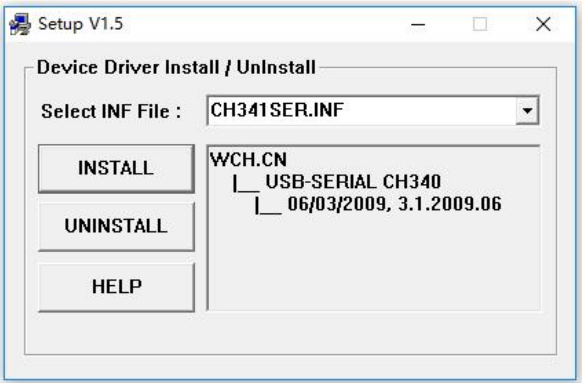
驱动安装(X64)

驱动安装/卸载

选择INF文件 : CH341SER.INF

```
WCH.CN
  |__ USB-SERIAL CH340
      |__ 01/30/2019, 3.5.2019
```

translate



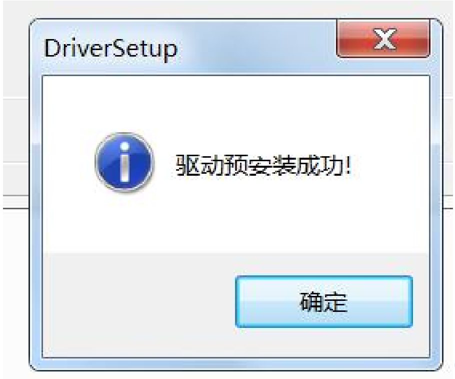
Setup V1.5

Device Driver Install / UnInstall

Select INF File : CH341SER.INF

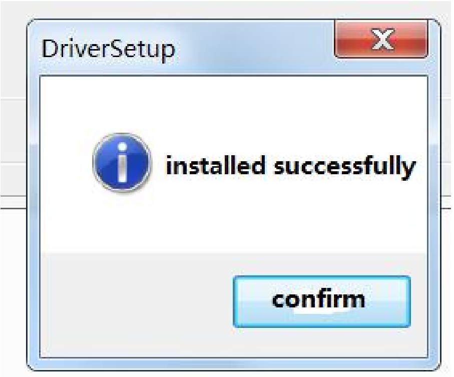
```
WCH.CN
 |__ USB-SERIAL CH340
     |__ 06/03/2009, 3.1.2009.06
```

4. Successful installation


DriverSetup

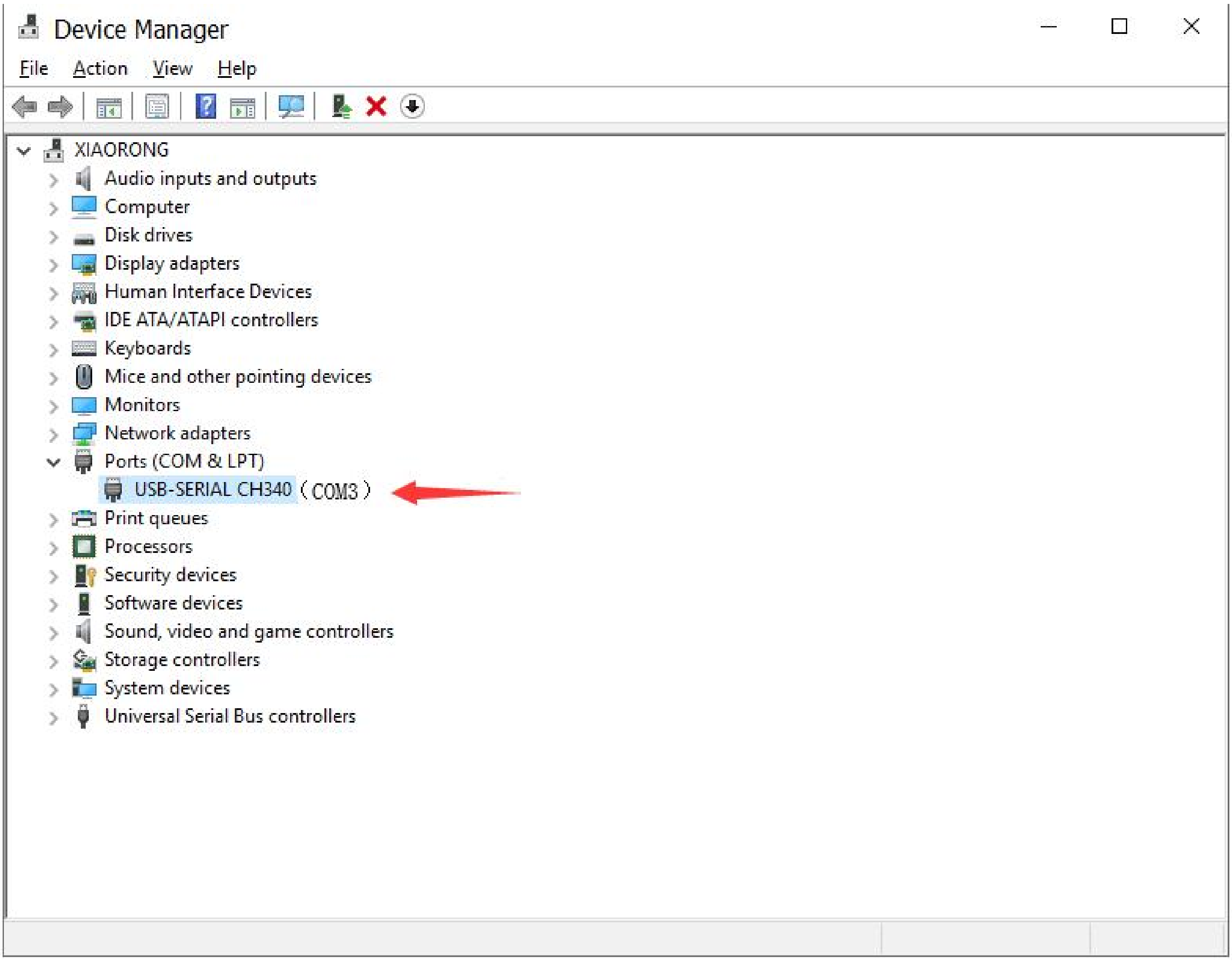
驱动预安装成功!

translate


DriverSetup

installed successfully

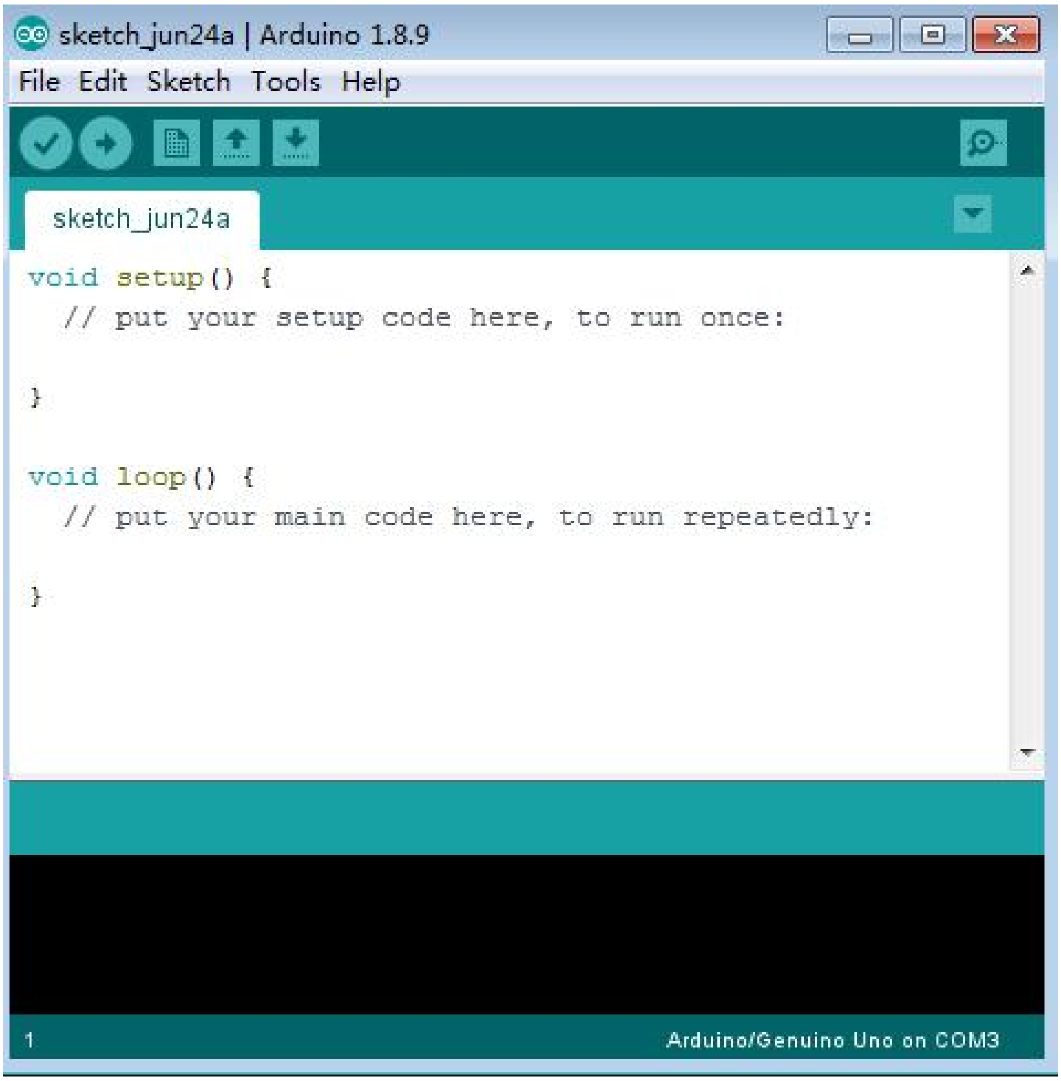
5. If the driver is successfully installed, when the nano motherboard is connected to the computer through the USB cable, under the path "Computer" -> "Properties" -> "Device manager", you can find CH340 COM3, as shown below.


Device Manager

XIAORONG
- Audio inputs and outputs
- Computer
- Disk drives
- Display adapters
- Human Interface Devices
- IDE ATA/ATAPI controllers
- Keyboards
- Mice and other pointing devices
- Monitors
- Network adapters
- Ports (COM & LPT)
  - USB-SERIAL CH340（COM3）
- Print queues
- Processors
- Security devices
- Software devices
- Sound, video and game controllers
- Storage controllers
- System devices
- Universal Serial Bus controllers

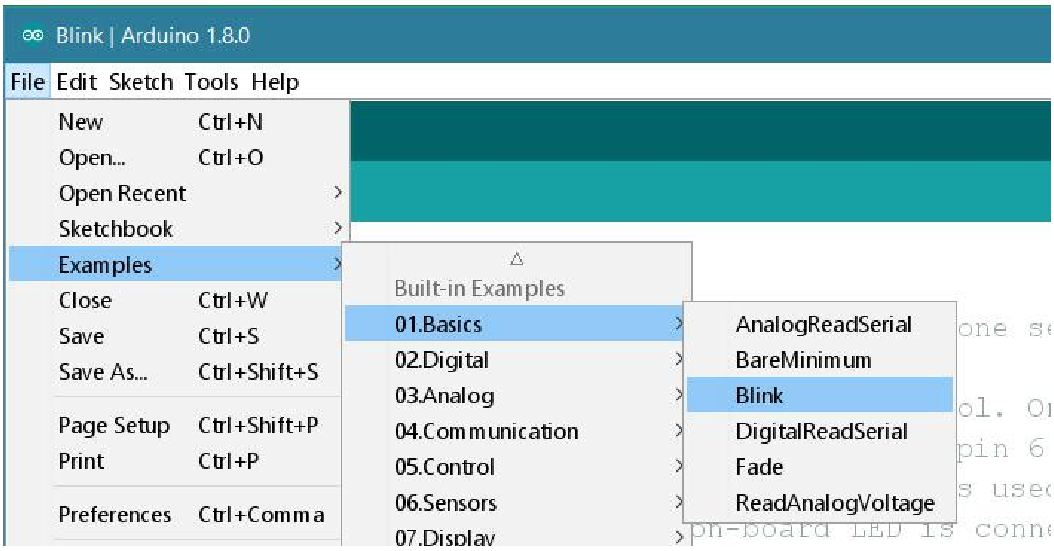
Upload your first sketch

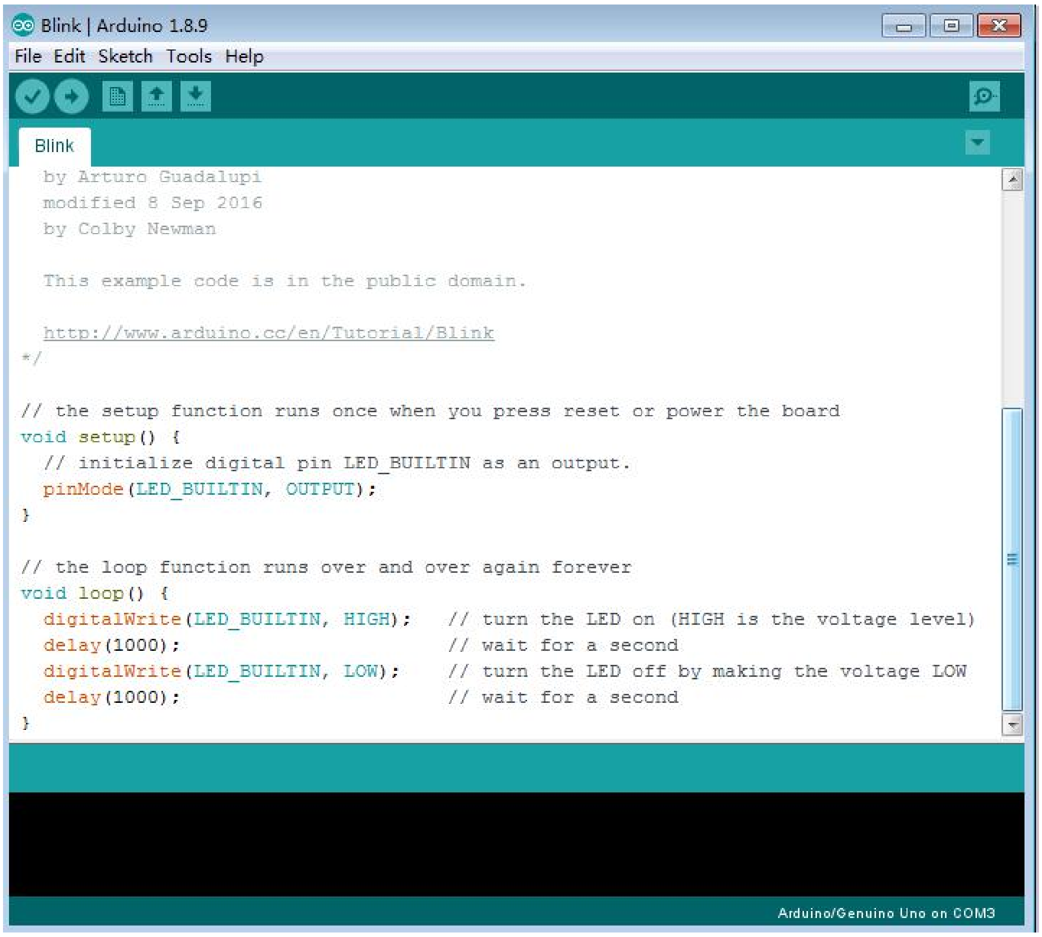
Double-click the computer desktop icon to open the arduino IDE, as shown below:


sketch_jun24a | Arduino 1.8.9
File Edit Sketch Tools Help
sketch_jun24a
1
Arduino/Genuino Uno on COM3

```
void setup() {
  // put your setup code here, to run once:

}

void loop() {
  // put your main code here, to run repeatedly:

}
```

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

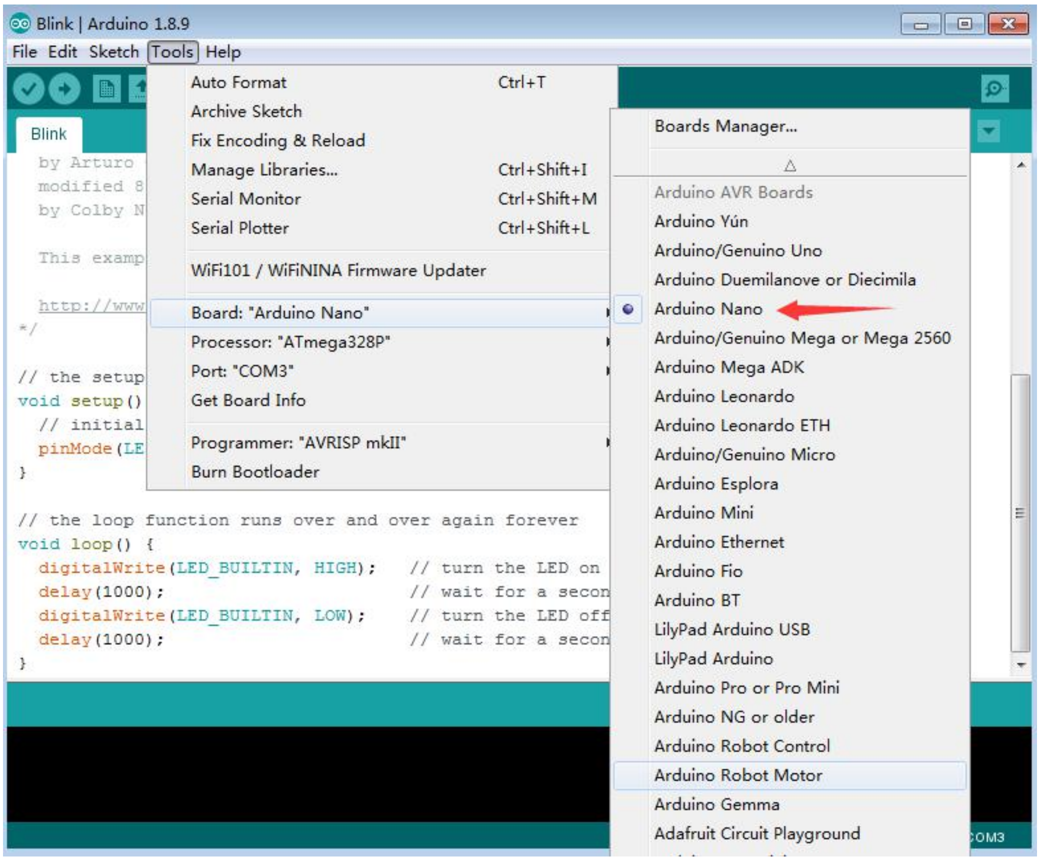




```
  by Arturo Guadalupi
  modified 8 Sep 2016
  by Colby Newman

  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
}
```

Select your board type and port

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



```
  by Arturo
  modified 8
  by Colby N

  This examp

  http://www
*/

// the setup
void setup()
  // initial
  pinMode(LE
}

// the loop function runs over and over again forever
void loop() {
  digitalWrite(LED_BUILTIN, HIGH);   // turn the LED on
  delay(1000);                       // wait for a secon
  digitalWrite(LED_BUILTIN, LOW);    // turn the LED off
  delay(1000);                       // wait for a secon
}
```

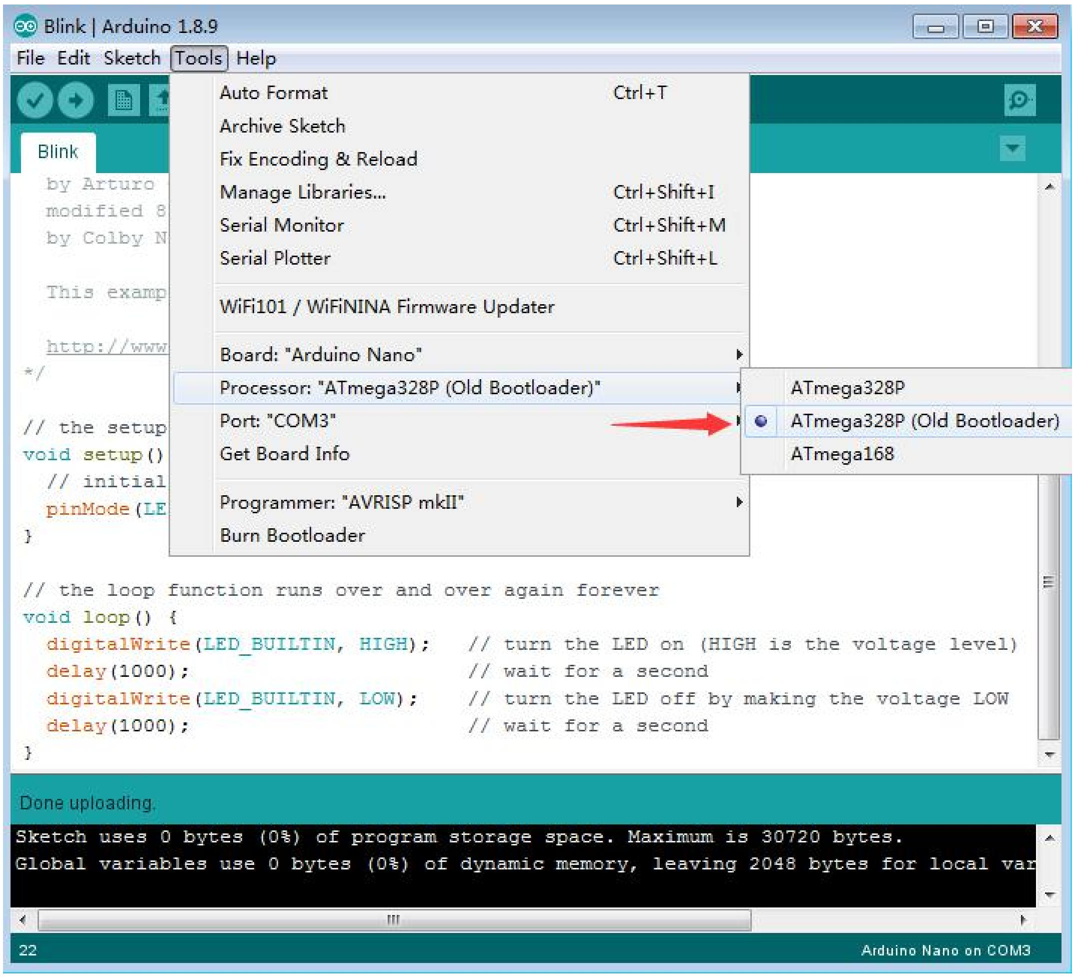
Nano processor selection:

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 Figure 1:

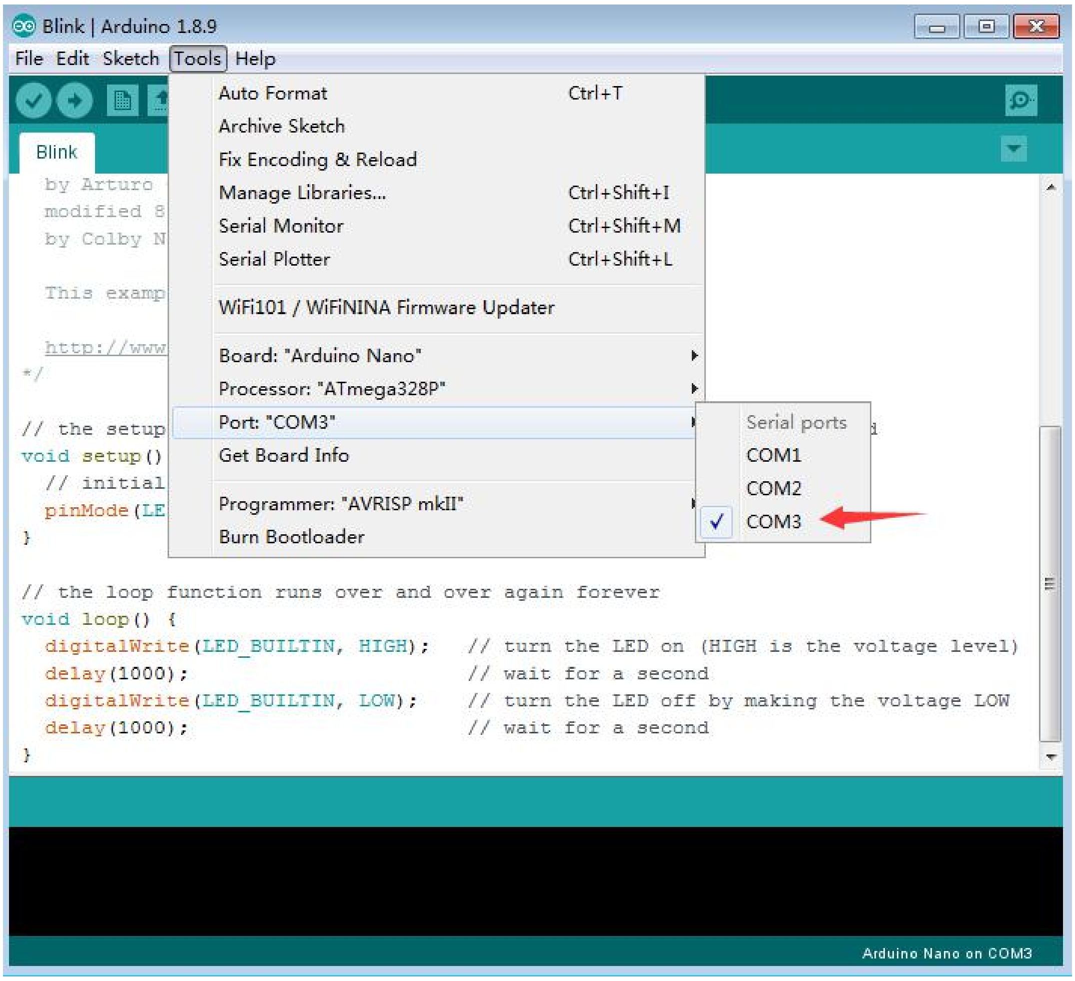

Blink | Arduino 1.8.9
File Edit Sketch Tools Help
Blink
Auto Format Ctrl+T
Archive Sketch
Fix Encoding & Reload
Manage Libraries... Ctrl+Shift+I
Serial Monitor Ctrl+Shift+M
Serial Plotter Ctrl+Shift+L
WiFi101 / WiFiNINA Firmware Updater
Board: "Arduino Nano"
Processor: "ATmega328P (Old Bootloader)"
Port: "COM3"
Get Board Info
Programmer: "AVRISP mkII"
Burn Bootloader
ATmega328P
● ATmega328P (Old Bootloader)
ATmega168
22
Arduino Nano on COM3

```
  by Arturo
  modified 8
  by Colby N

  This examp

  http://www
*/

// the setup
void setup()
  // initial
  pinMode(LE
}

// 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 uploading.

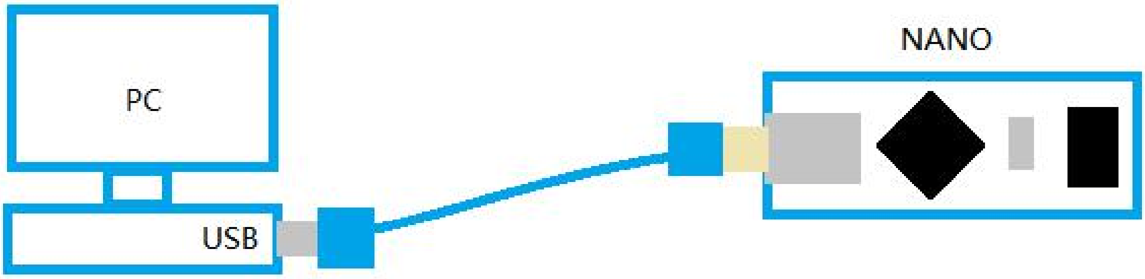
```
Sketch uses 0 bytes (0%) of program storage space. Maximum is 30720 bytes.
Global variables use 0 bytes (0%) of dynamic memory, leaving 2048 bytes for local var
```

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

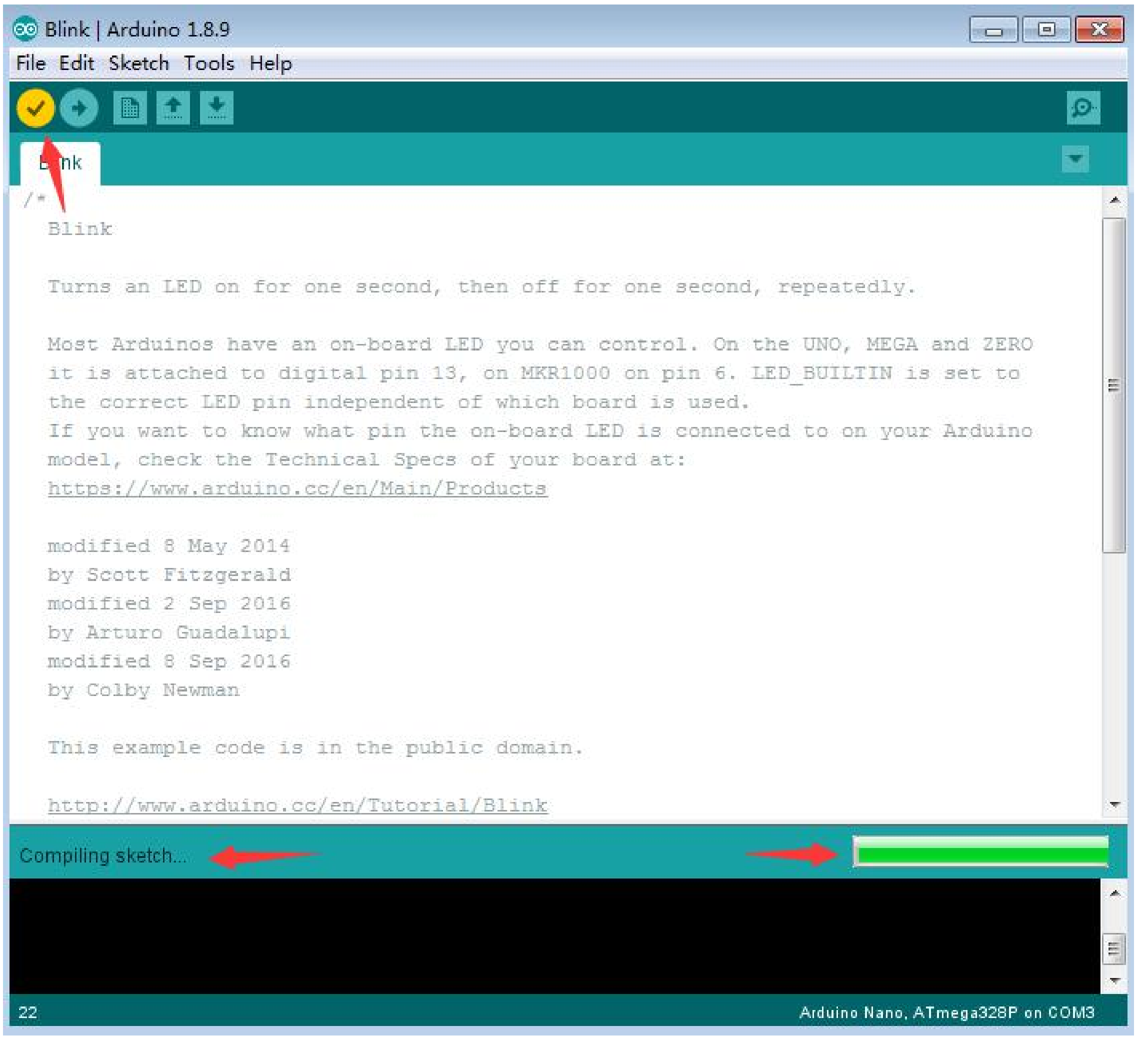


```
  by Arturo
  modified 8
  by Colby N

  This examp

  http://www
*/

// the setup
void setup()
  // initial
  pinMode(LE
}

// 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
}
```

Connect the PC to the nano motherboard via the USB cable



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



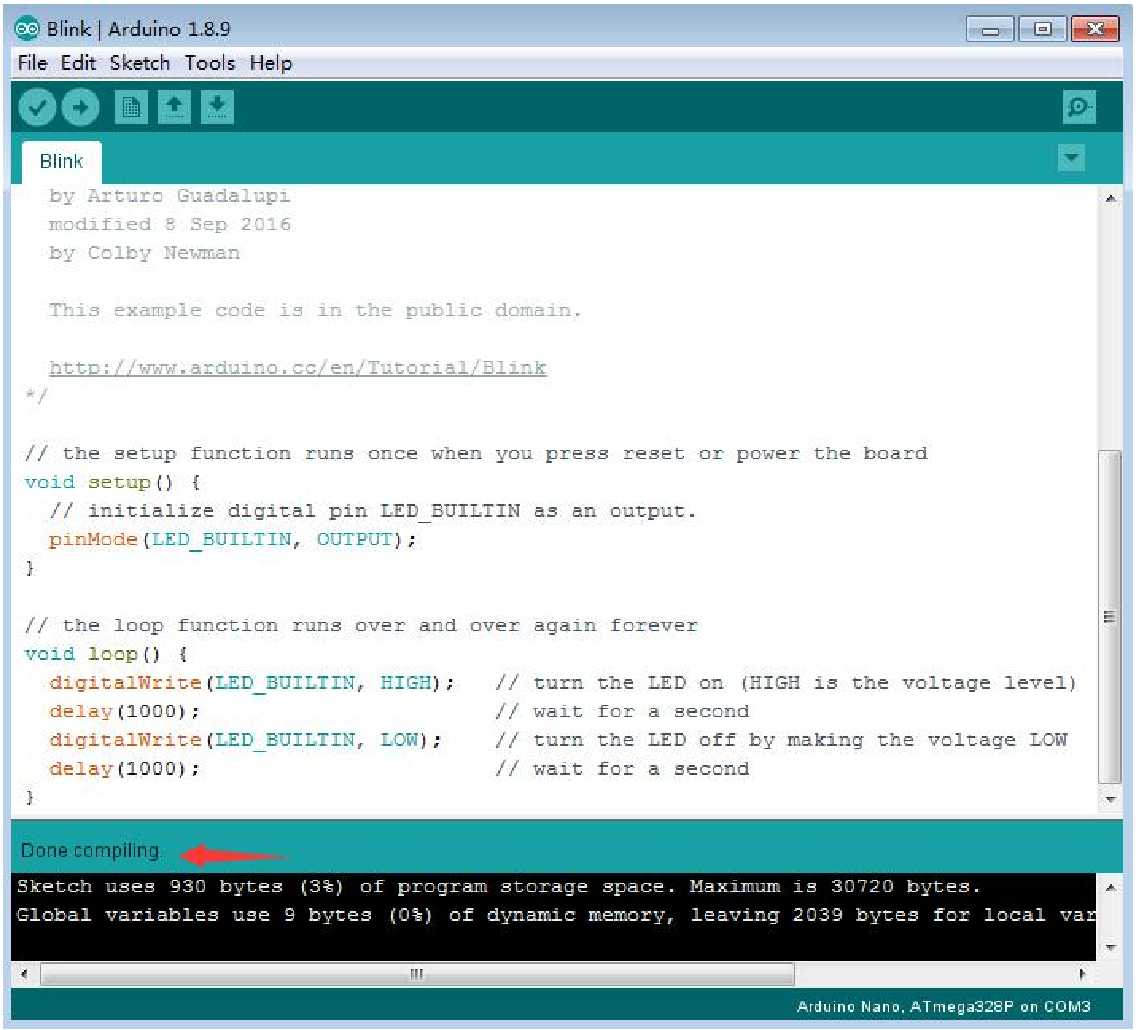
```
/*
  Blink

  Turns an LED on for one second, then off for one second, repeatedly.

  Most Arduinos have an on-board LED you can control. On the UNO, MEGA and ZERO
  it is attached to digital pin 13, on MKR1000 on pin 6. LED_BUILTIN is set to
  the correct LED pin independent of which board is used.
  If you want to know what pin the on-board LED is connected to on your Arduino
  model, check the Technical Specs of your board at:
  https://www.arduino.cc/en/Main/Products

  modified 8 May 2014
  by Scott Fitzgerald
  modified 2 Sep 2016
  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
```

Compiling sketch...

22

Arduino Nano, ATmega328P on COM3

Upload the code



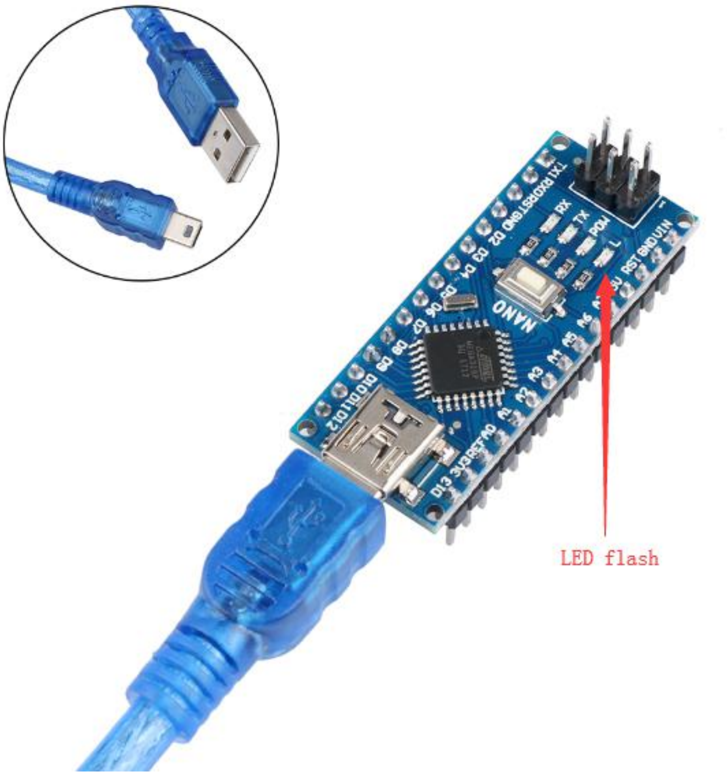
```
  by Arturo Guadalupi
  modified 8 Sep 2016
  by Colby Newman

  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 var
```

Arduino Nano, ATmega328P on COM3

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



LED flash