Lesson 2 - Upload Your First Sketch To The Nano

Table

1、Nano Board Overview: 2
2、Nano Board Specification： 2
3、Nano Board Introduce： 2
4、Upload your first sketch 4

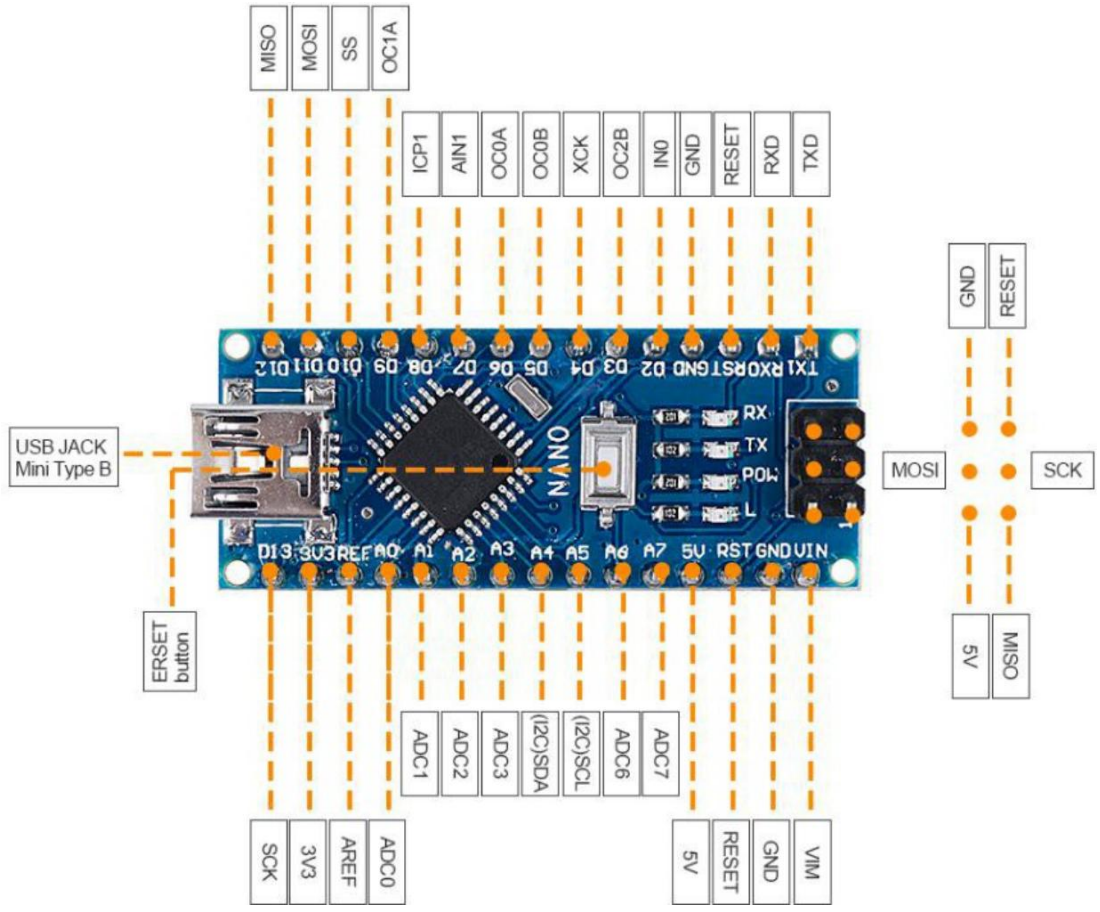
1、Nano Board Overview：

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、Nano Board 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
Bootloader：nano

3、Nano Board Introduce：



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

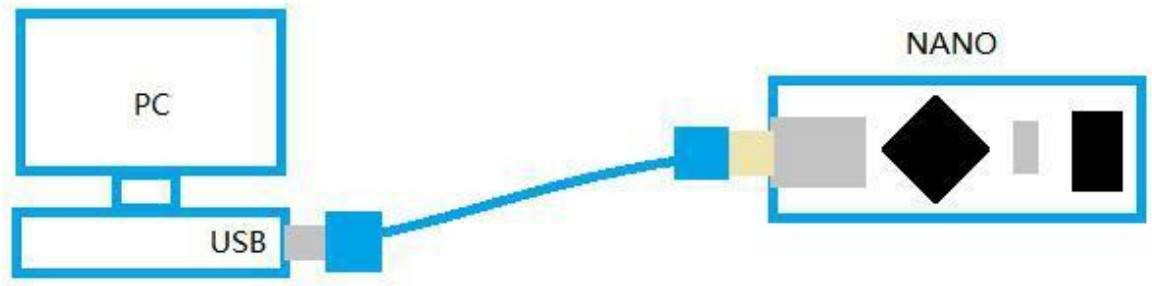
Rather than requiring a physical press of the reset button before an upload, the Arduino Nano is designed in a way that allows it to be reset by software running on a connected computer. One of the hardware flow control lines (DTR) of the FT232RL is connected to the reset line of the ATmega328 via a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

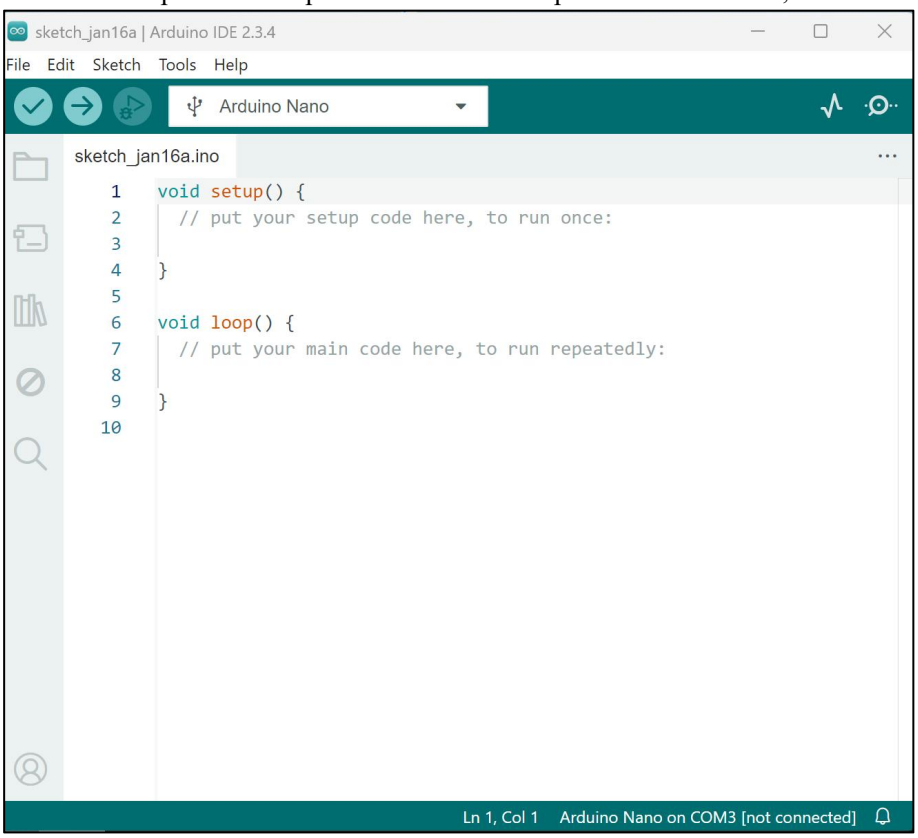
4、Upload your first sketch

Connect the Arduino Nano board to the computer via the micro USB cable

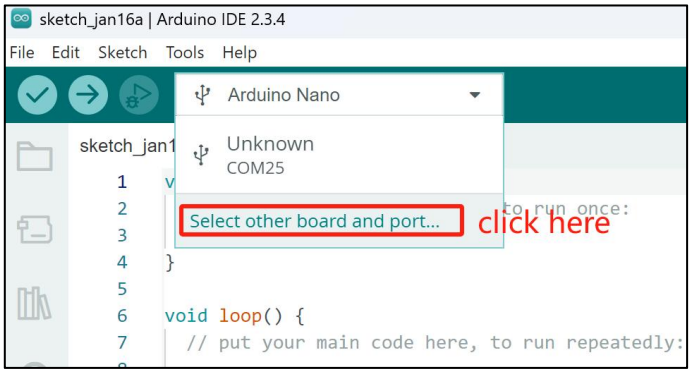


Double-click the computer desktop icon 

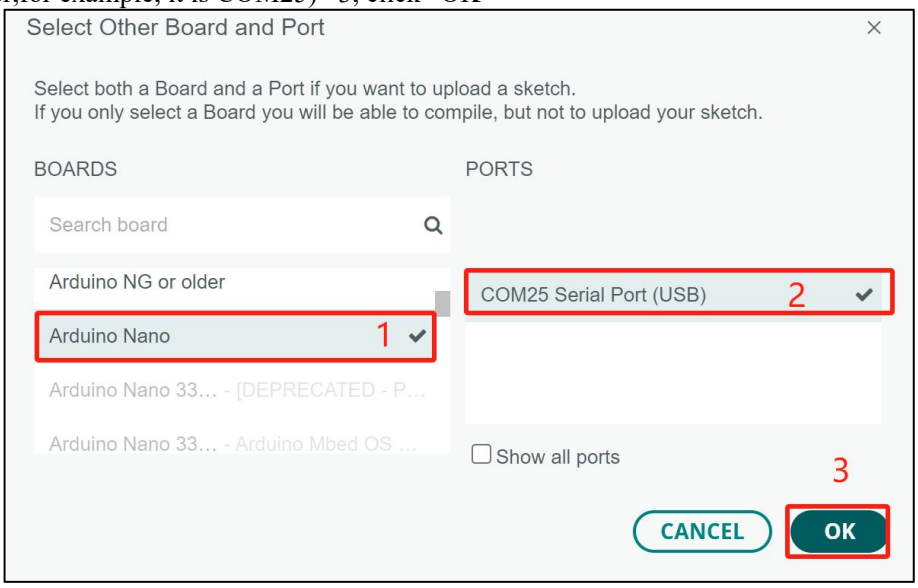
to open the arduino IDE, as shown below:



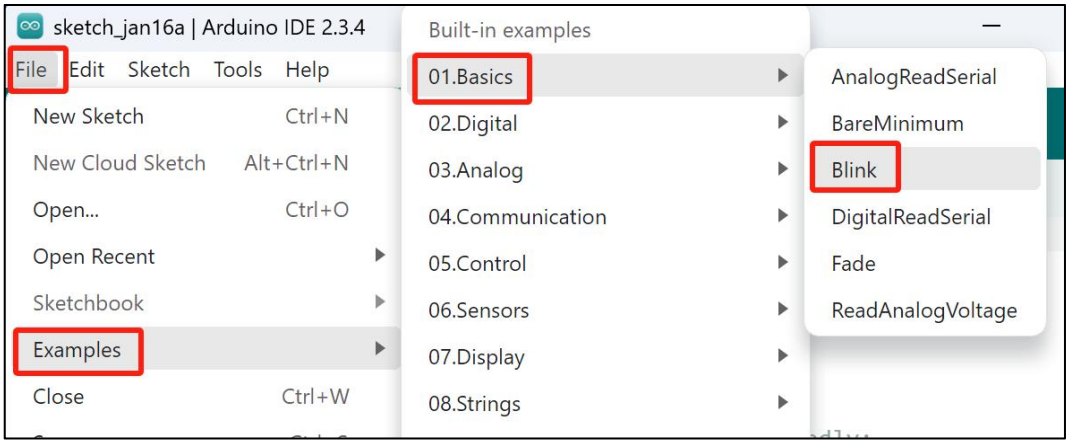
Click here to select Nano board and it's COM port



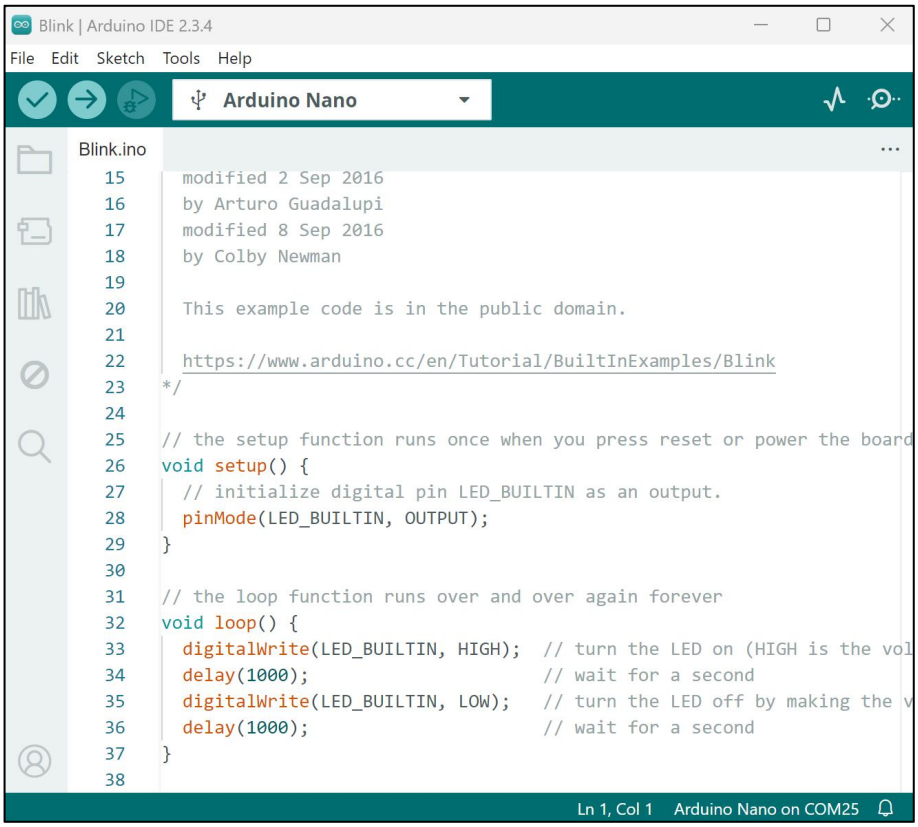
1,select the Arduino Nano 2,select the COMx(the COMx is the Nano board's port in the computer,for example, it is COM25) 3, click "OK"



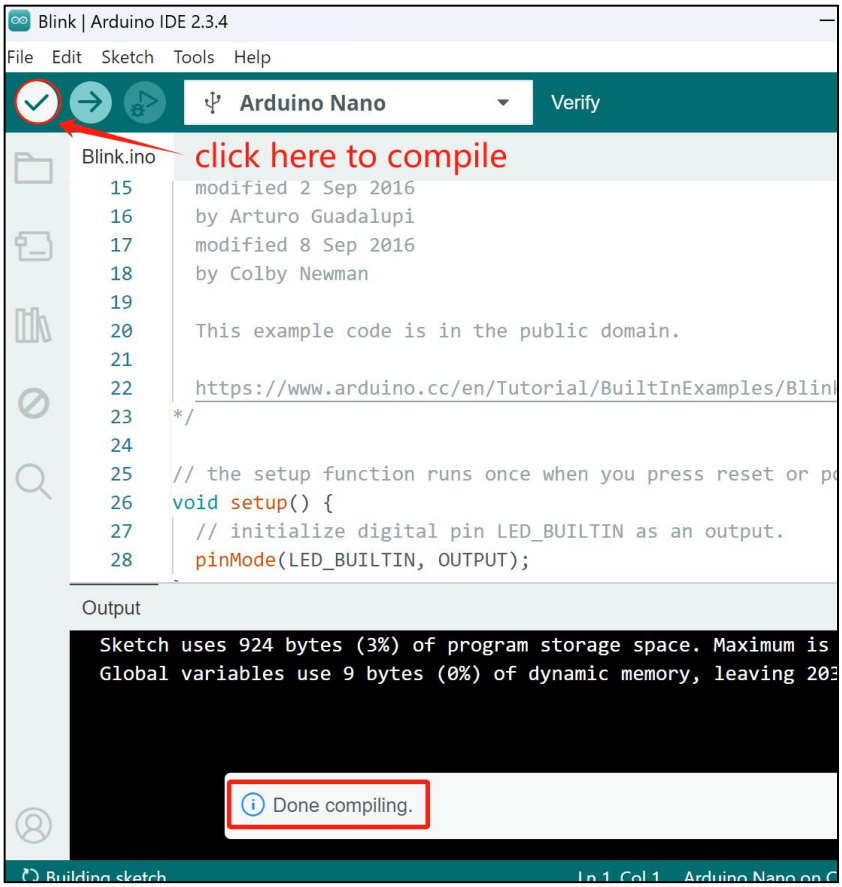
Open the LED blink example sketch: File > Examples >01.Basics > Blink.



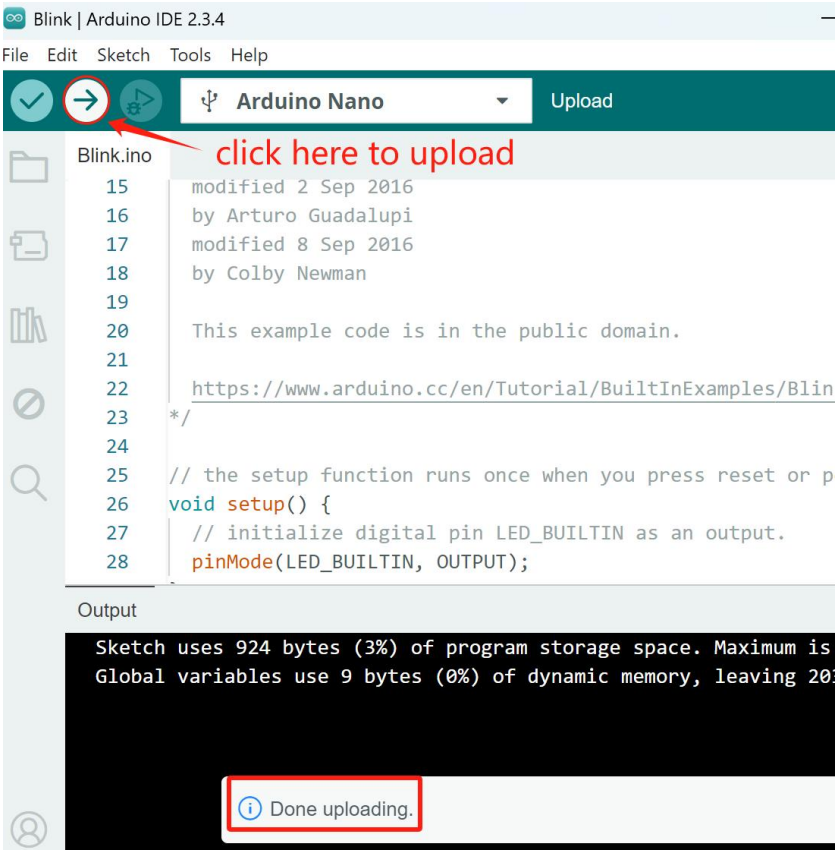
The blink example sketch like below:



Click compile button  , successfully compiled the code will display "Done compiling"



Click upload button  to upload, successfully uploading the code will display "Done uploading".



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