

---

# Board for Arduino

**keyestudio**

**Jun 20, 2022**

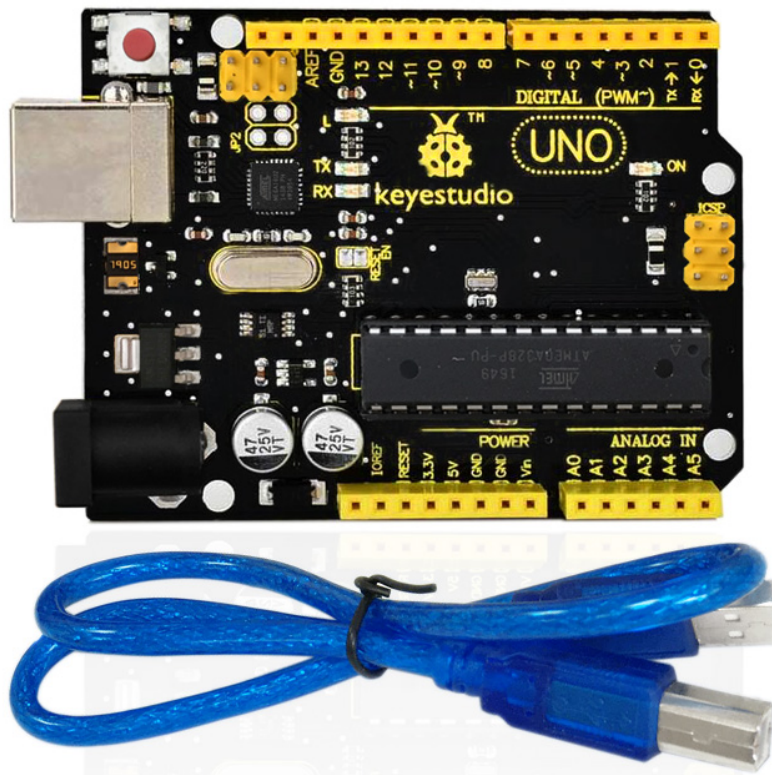




## MCU DEVELOPMENT BOARD FOR ARDUINO

1	KS0002 Keyestudio MEGA 2560 R3 development board	27
2	KS0172 Keyestudio UNO with pin header development board	55
3	KS0173 Keyestudio CH340 NANO development board	83
4	KS0247 Keyestudio Pro Mini development board	113
5	KS0248 Keyestudio Leonardo R3 development board	143
6	KS0249 Keyestudio PRO MICRO development board	173
7	KS0341 Keyestudio UNO Compatible advanced development board	203
8	KS0342 Keyestudio MEGA Compatible advanced development board	233
9	KS0486 Keyestudio PLUS development board	263
10	KS0497 Keyestudio V4.0 development board	287
11	KS0498 Keyestudio MEGA 2560 development board	311
12	KS0502 Keyestudio MEGA PRO development board	441
13	KS0546 Keyestudio 328 PLUS development board	465
14	KS0547 Keyestudio NANO PLUS development board	493





### Introduction:

Keyestudio UNO R3 development board is a microcontroller board based on the ATmega328P ([datasheet](#)), fully compatible with ARDUINO UNO REV3. It has 14 digital input/output pins (of which 6 can be used as PWM outputs), 6 analog inputs, a 16 MHz quartz crystal, a USB connection, a power jack, 2 ICSP headers and a reset button.

It contains everything needed to support the microcontroller; simply connect it to a computer with a USB cable or power it with a AC-to-DC adapter or battery to get started.

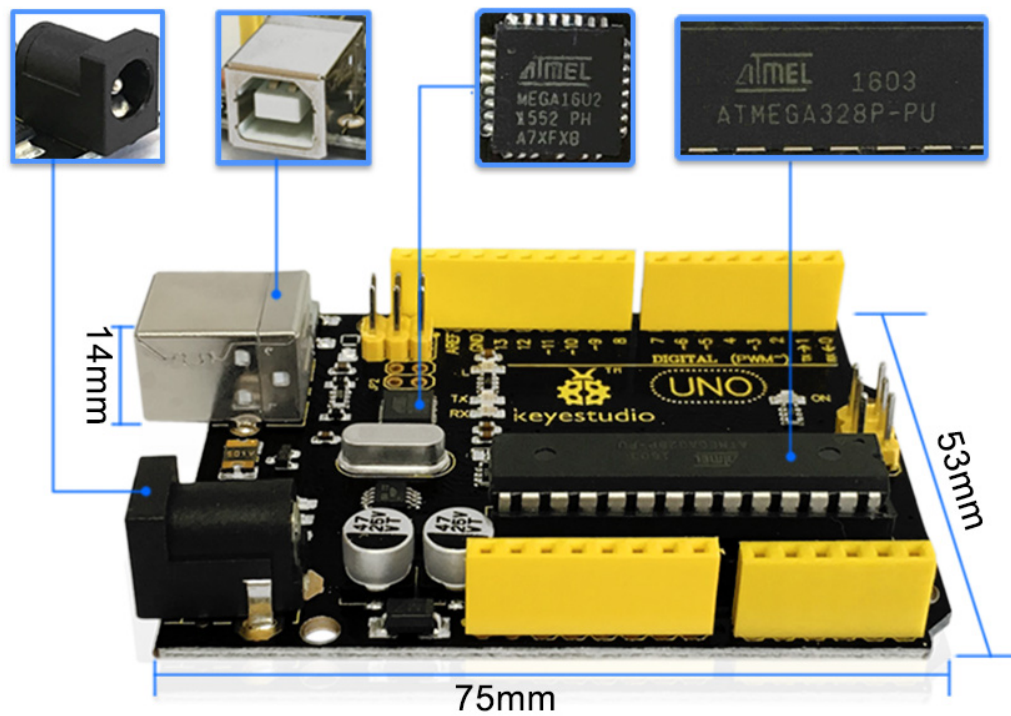
Note that the two ICSP headers separately used to program the firmware to ATMEGA16U2-MU and ATMEGA328P-PU, but we have programmed the two chips, so generally no need to use them.

The Uno R3 differs from all preceding boards in that it does not use the FTDI USB-to-serial driver chip. Instead, it features the Atmega16U2 programmed as a USB-to-serial converter.

The UNO is the best board to get started with electronics and coding. If this is your first experience tinkering with the platform, the UNO is the most robust board you can start playing with.

tech specs:

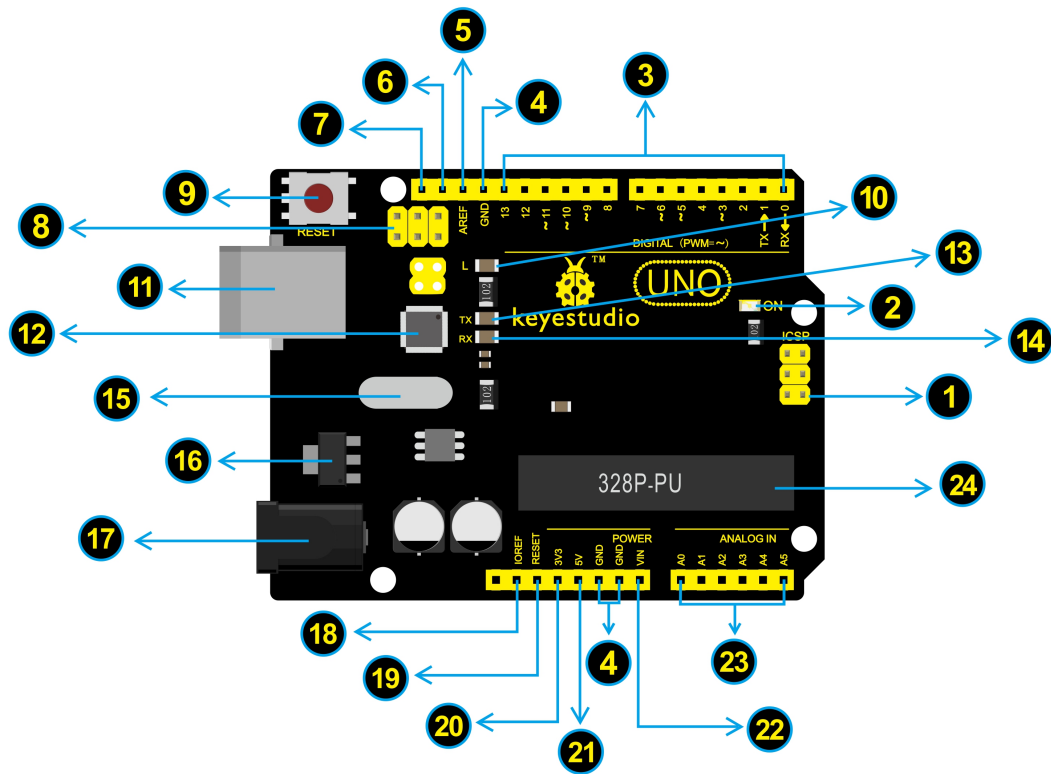
<b>Microcontroller</b>	ATmega328P-PU
<b>Operating Voltage</b>	5V
<b>Input Voltage (recommended)</b>	7-12V
<b>Digital I/O Pins</b>	14 (of which 6 provide PWM output)
<b>PWM Digital I/O Pins</b>	6 (D3, D5, D6, D9, D10, D11)
<b>Analog Input Pins</b>	6 (A0-A5)
<b>DC Current per I/O Pin</b>	20 mA
<b>DC Current for 3.3V Pin</b>	50 mA
<b>Flash Memory</b>	32 KB (ATmega328P-PU) of which 0.5 KB used by bootloader
<b>SRAM</b>	2 KB (ATmega328P-PU)
<b>EEPROM</b>	1 KB (ATmega328P-PU)
<b>Clock Speed</b>	16 MHz
<b>LED_BUILTIN</b>	D13



**Dimensions:**

**Element and Interfaces:**

Here is an explanation of what every element and interface of the board does:



1	<b>ICSP (In-Circuit Serial Programming) Header</b> In most case, ICSP is the AVRAn Arduino micro-program header consisting of MOSI, MISO, SCK, RESET, VCC, and GND. It is often called the SPI (serial peripheral interface) and can be considered an “extension” of the output. In fact, slave the output devices under the SPI bus host. When connecting to PC, program the firmware to ATMEGA328P-PU.
2	<b>Power LED Indicator</b> Powering the Arduino, LED on means that your circuit board is correctly powered on. If LED is off, connection is wrong.
3	<b>Digital I/O</b> Arduino UNO has 14 digital input/output pins (of which 6 can be used as PWM outputs). These pins can be configured as digital input pin to read the logic value (0 or 1). Or used as digital output pin to drive different modules like LED, relay, etc. The pin labeled “” can be used to generate PWM.
4	<b>GND ( Ground pin headers)</b> Used for circuit ground
5	<b>AREF</b> Reference voltage (0-5V) for analog inputs. Used with <code>analogReference()</code> .
6	<b>SDA</b> IIC communication pin
7	<b>SCL</b> IIC communication pin
8	<b>ICSP (In-Circuit Serial Programming) Header</b> In most case, ICSP is the AVR, an Arduino micro-program header consisting of MOSI, MISO, SCK, RESET, VCC, and GND. Connected to ATMEGA 16U2-MU. When connecting to PC, program the firmware to ATMEGA 16U2-MU.
9	<b>RESET Button</b> You can reset your Arduino board, for example, start the program from the initial status. You can use the RESET button.
10	<b>D13 LED</b> 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.
11	<b>USB Connection</b> Arduino board can be powered via USB connector. All you needed to do is connecting the USB port to PC using a USB cable.
12	<b>ATMEGA 16U2-MU</b> USB to serial chip, can convert the USB signal into serial port signal.
13	<b>TX LED</b> Onboard you can find the label: TX (transmit) When Arduino board communicates via serial port, send the message, TX led flashes.
14	<b>RX LED</b> Onboard you can find the label: RX(receive ) When Arduino board communicates via serial port, receive the message, RX led flashes.
15	<b>Crystal Oscillator</b> Helping Arduino deal with time problems. How does Arduino calculate time? by using a crystal oscillator. The number printed on the top of the Arduino crystal is 16.000H9H. It tells us that the frequency is 16,000,000 Hertz or 16MHz.
16	<b>Voltage Regulator</b> To control the voltage provided to the Arduino board, as well as to stabilize the DC voltage used by the processor and other components. Convert an external input DC7-12V voltage into DC 5V, then switch DC 5V to the processor and other components.
17	<b>DC Power Jack</b> Arduino board can be supplied with an external power DC7-12V from the DC power jack.
18	<b>IOREF</b> Used to configure the operating voltage of microcontrollers. Use it less.
19	<b>RESET Header</b> Connect an external button to reset the board. The function is the same as reset button (labeled 9)
20	<b>Power Pin 3V3</b> A 3.3 volt supply generated by the on-board regulator. Maximum current draw is 50 mA.
21	<b>Power Pin 5V</b> Provides 5V output voltage
22	<b>Vin</b> You can supply an external power input DC7-12V through this pin to Arduino board.
23	<b>Analog Pins</b> Arduino UNO board has 6 analog inputs, labeled A0 through A5. These pins can read the signal from analog sensors (such as humidity sensor or temperature sensor), and convert it into the digital value that can read by microcontrollers) Can also used as digital pins, A0=D14, A1=D15, A2=D16, A3=D17, A4=D18, A5=D19.
24	

### Specialized Functions of Some Pins:

- **Serial communication:** Digital pins 0 (RX) and 1 (TX).
- **PWM Interfaces (Pulse-Width Modulation):** D3, D5, D6, D9, D10, D11
- **External Interrupts:** D2 (interrupt 0) and D3 (interrupt 1). These pins can be configured to trigger an interrupt on a low value, a rising or falling edge, or a change in value.
- **SPI communication:** D10 (SS), D11 (MOSI), D12 (MISO), D13 (SCK). These pins support SPI communication using the SPI library.
- **IIC communication:** A4 (SDA); A5(SCL)

### Warnings:

- The Arduino Uno has a resettable polyfuse that protects your computer's USB ports from shorts and overcurrent. If more than 500 mA is applied to the USB port, the fuse will automatically break the connection until the short or overload is removed.
- Automatic (Software) Reset:  
Rather than requiring a physical press of the reset button before an upload, the Arduino Uno board is designed in a way that allows it to be reset by software running on a connected computer.
- 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; see [this forum thread](#) for details.

### Detailed Use with ARDUINO Software as follows:

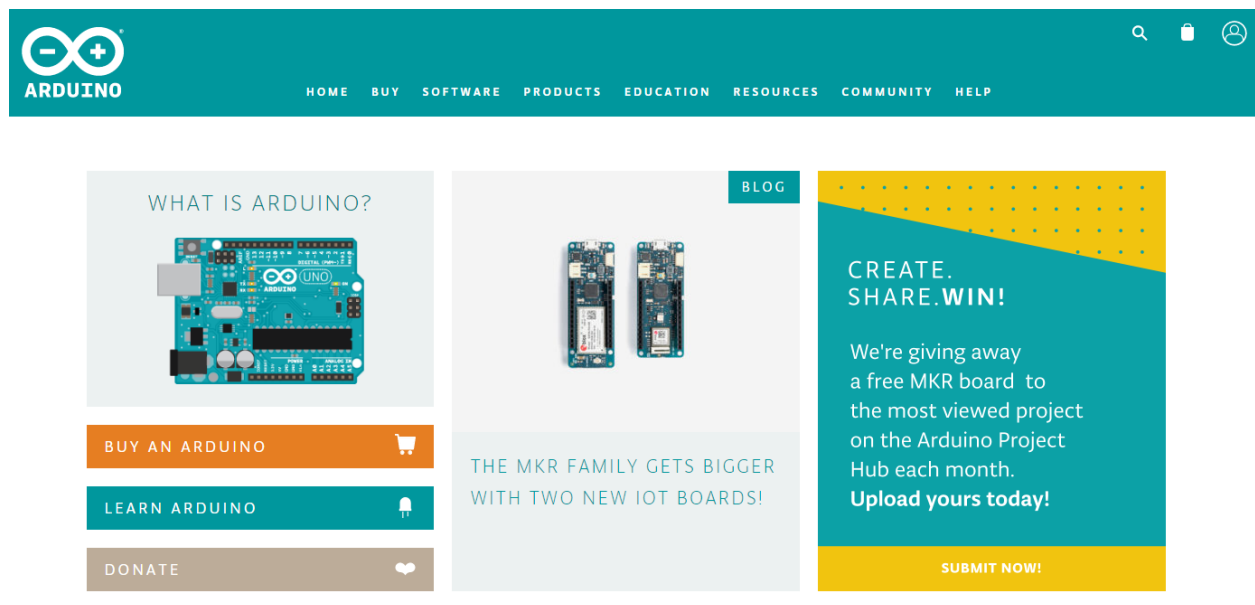
#### Step1 | Download the Arduino environment (IDE)

When you get the UNO development board, first you should install the Arduino software and driver.

We usually use the Windows software Arduino 1.5.6 version. You can download it from the link below:

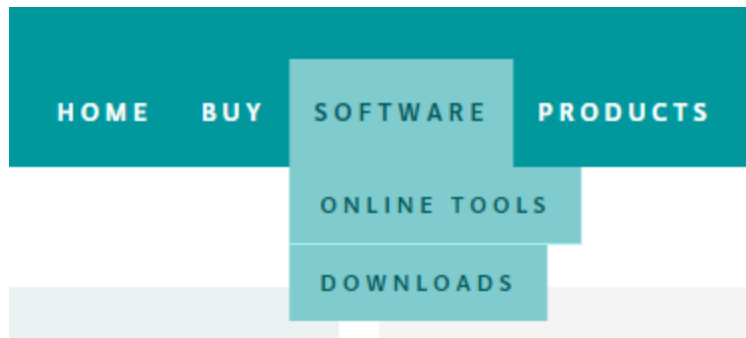
<https://www.arduino.cc/en/Main/OldSoftwareReleases#1.5.x>

Or you can browse the ARDUINO website to download the latest version from this link, <https://www.arduino.cc>, pop up the following interface.



Then click the **SOFTWARE** on the browse bar, you will have two options ONLINE TOOLS and DOWNLOADS.





Click **DOWNLOADS**, it will appear the latest software version of ARDUINO 1.8.5 shown as below.

## Download the Arduino IDE

A screenshot of the Arduino 1.8.5 download page. On the left, there is a large teal circle containing a white infinity symbol with a minus sign on the left and a plus sign on the right. To the right of this icon, the text reads: 'ARDUINO 1.8.5', '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.', and 'This software can be used with any Arduino board. Refer to the [Getting Started](#) page for installation instructions.' On the right side of the page, there is a teal sidebar with white text. It lists: 'Windows Installer, for Windows XP and up', 'Windows ZIP file for non admin install', 'Windows app Requires Win 8.1 or 10' with a 'Get' button, 'Mac OS X 10.7 Lion or newer', 'Linux 32 bits', 'Linux 64 bits', 'Linux ARM', 'Release Notes', 'Source Code', and 'Checksums (sha512)'.

In this software page, on the right side you can see the version of development software for different operating systems. ARDUINO has a powerful compatibility. You should download the software that is compatible with the operating system of your computer.

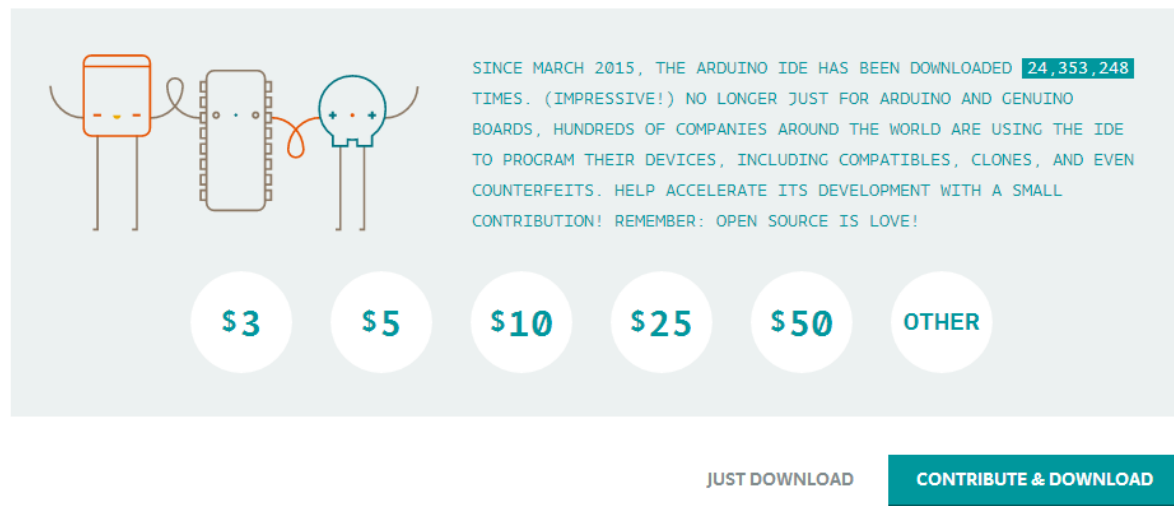
We will take **WINDOWS system** as an example here. There are also two options under Windows system, one is installed version, the other is non-installed version.

For simple installed version, first click **Windows Installer**, you will get the following page.



## Contribute to the Arduino Software

Consider supporting the Arduino Software by contributing to its development. (US tax payers, please note this contribution is not tax deductible). [Learn more on how your contribution will be used.](#)



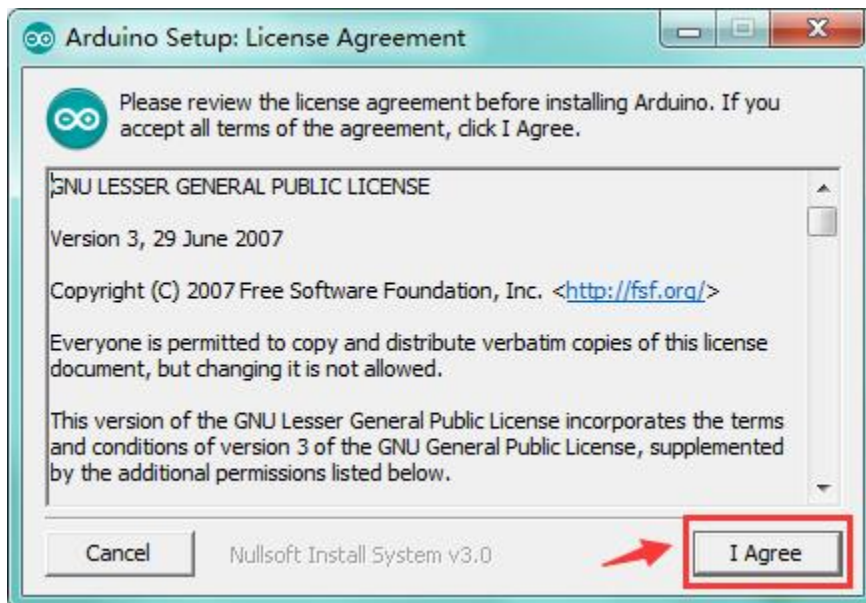
This way you just need to click JUST DOWNLOAD, then click the downloaded file to install it.

For non-installed version, first click Windows ZIP file, you will also get the pop-up interface as the above figure.

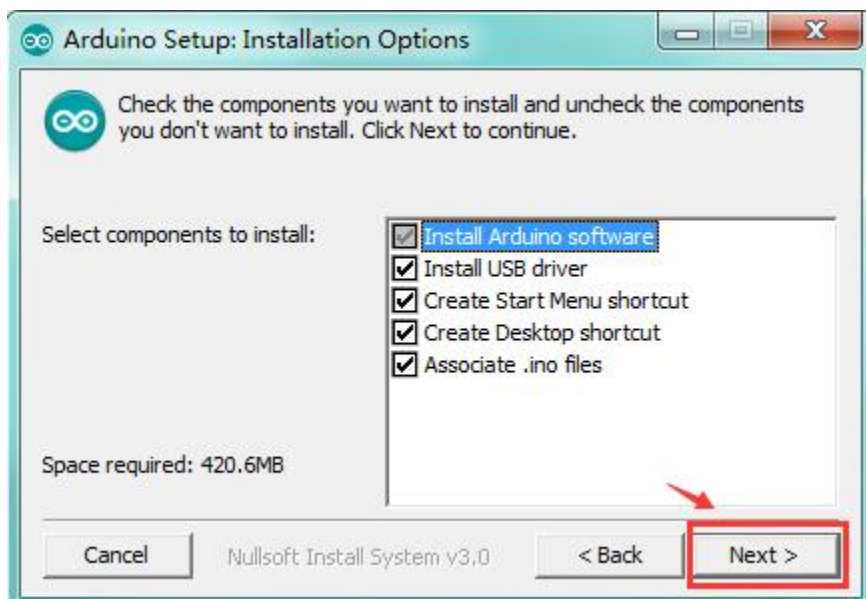
Click JUST DOWNLOAD, and when the ZIP file is downloaded well to your computer, you can directly unzip the file and click the icon of ARDUINO software to start it.

### Installing Arduino (Windows):

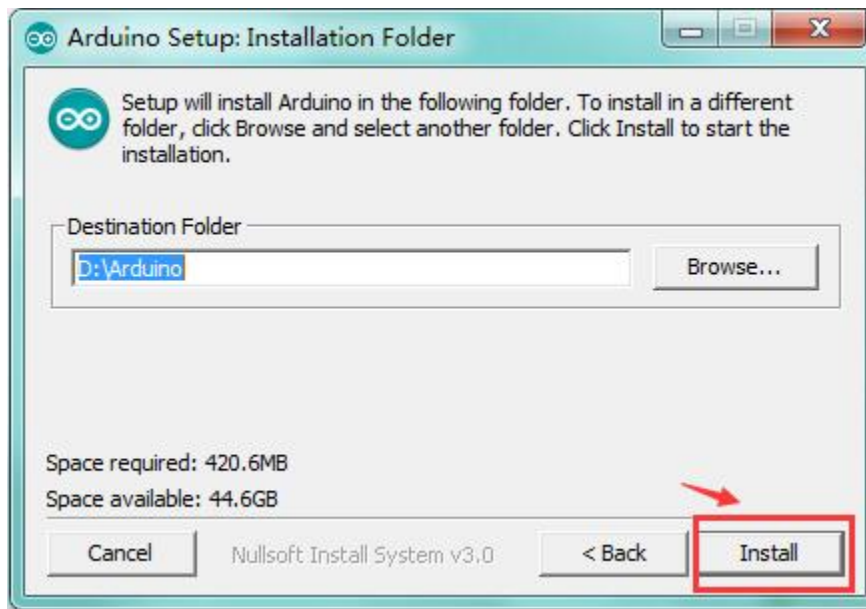
Install Arduino with the exe. Installation package downloaded well.



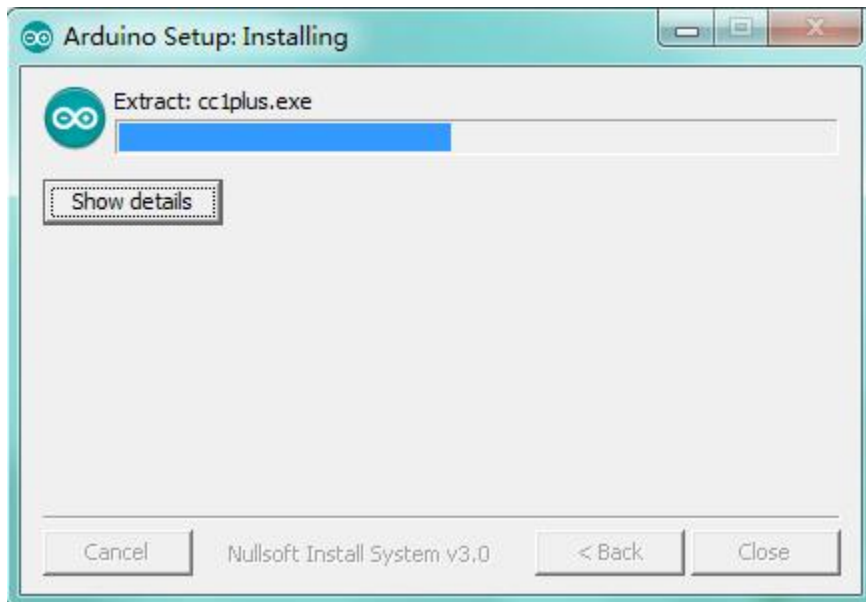
Click “I Agree” to see the following interface.



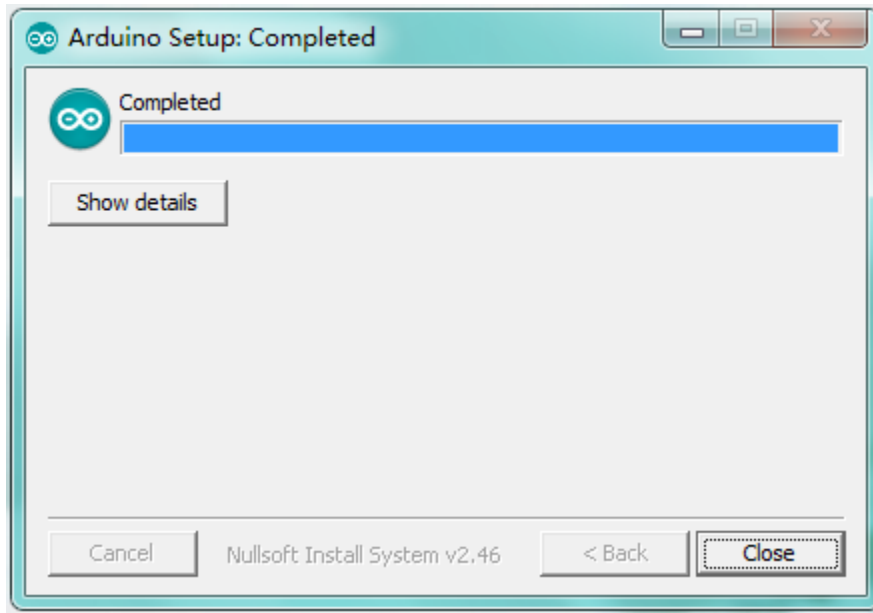
Click “Next”. Pop up the interface below.



You can press Browse... to choose an installation path or directly type in the directory you want. Then click “Install” to initiate installation.



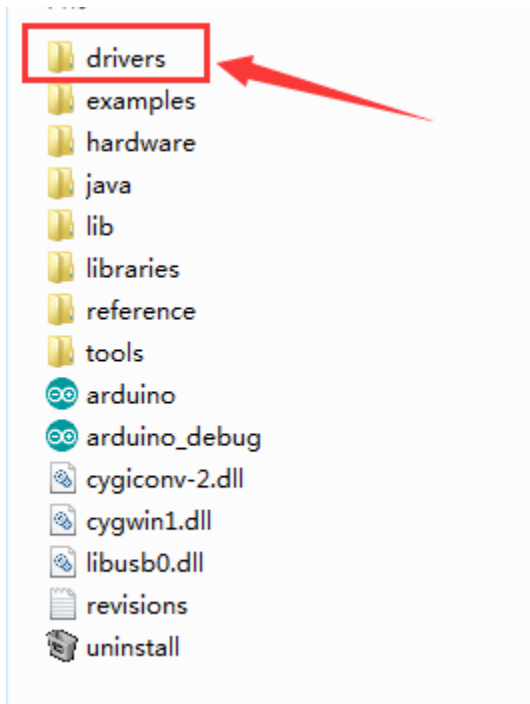
Wait for the installing process, if appear the interface of Window Security, just continue to click Install to finish the installation.



### Installing Driver:

Next, we will introduce the driver installation of UNO R3 development board. The driver installation may have slight differences in different computer systems. So in the following let's move on to the driver installation in the WIN 7 system.

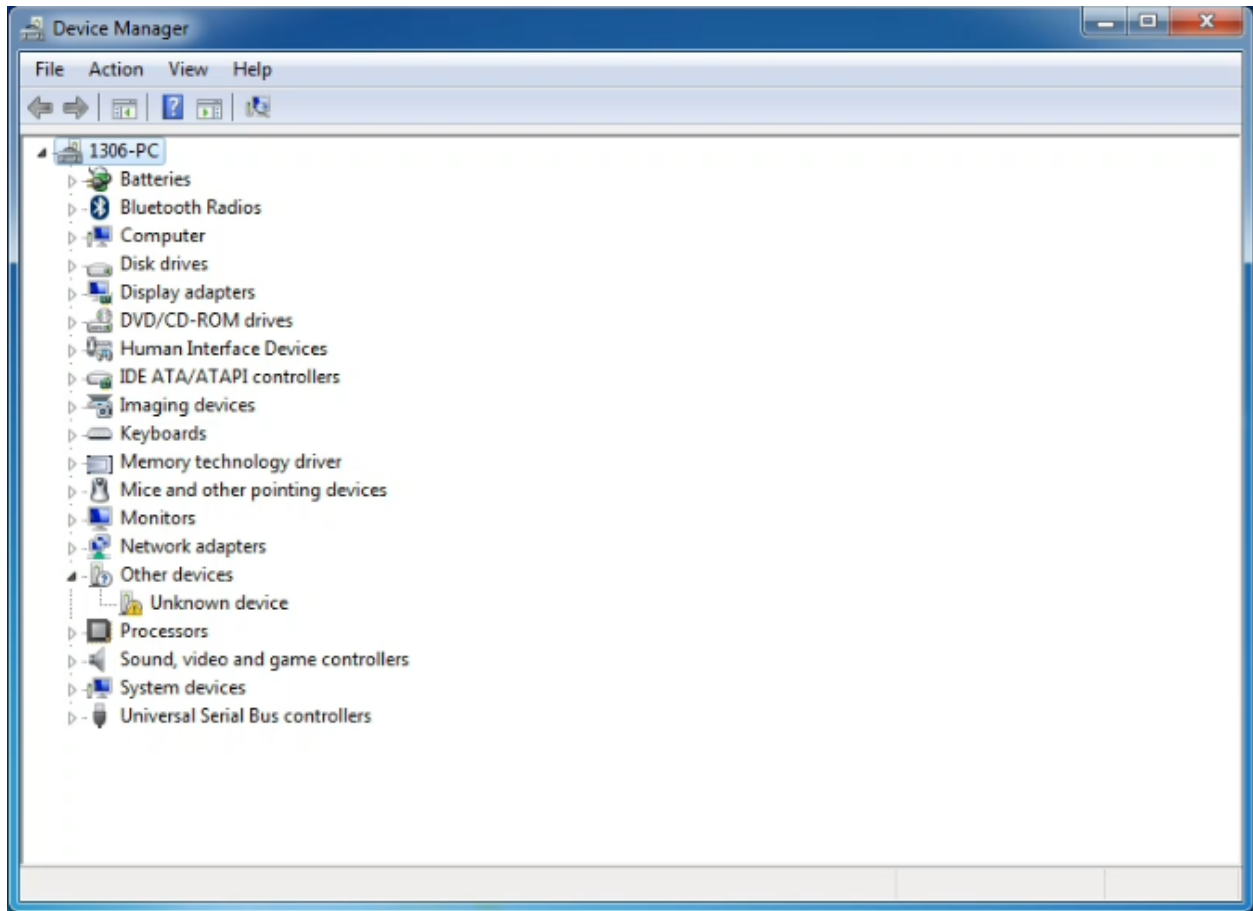
The Arduino folder contains both the Arduino program itself and the drivers that allow the Arduino to be connected to your computer by a USB cable. Before we launch the Arduino software, you are going to install the USB drivers.



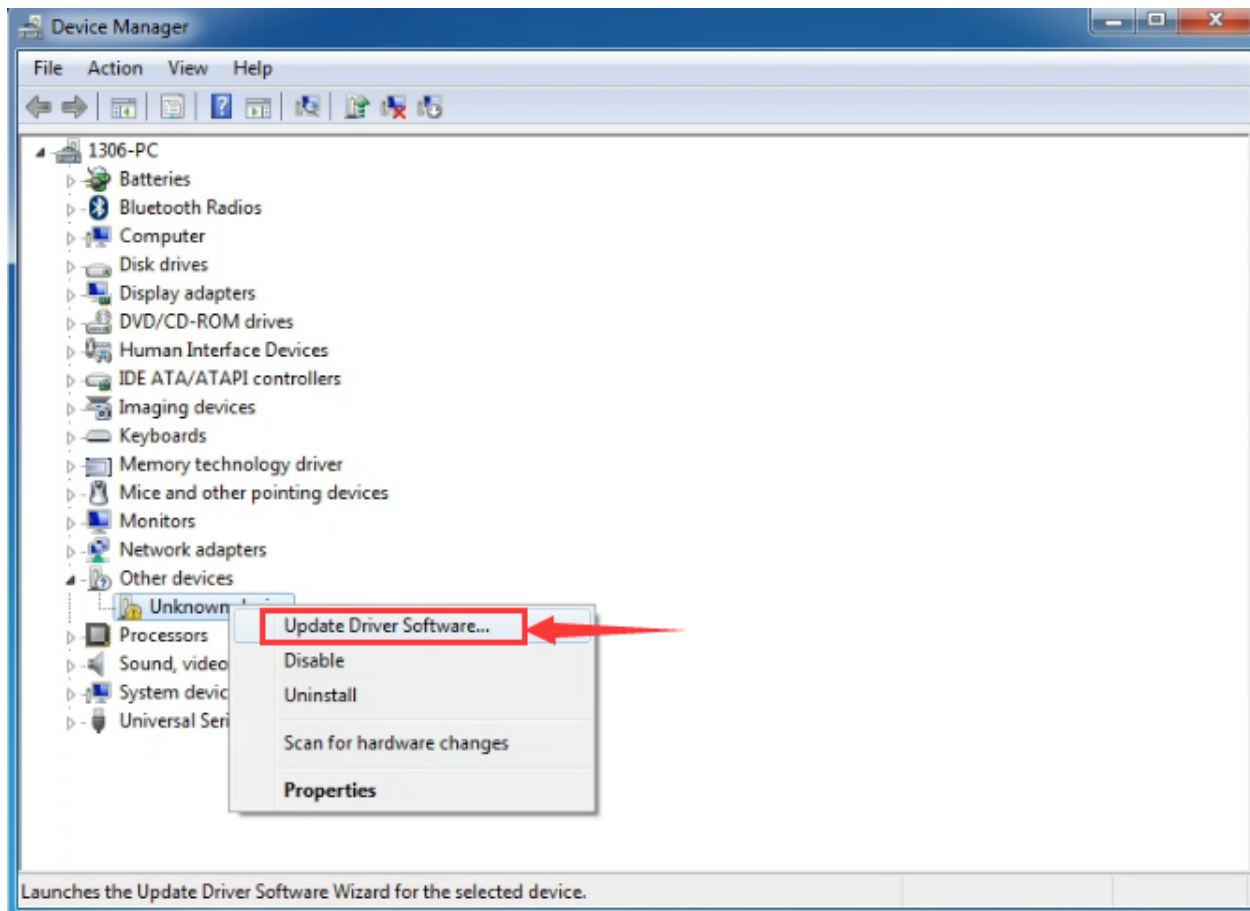
Plug one end of your USB cable into the Arduino and the other into a USB socket on your computer.

When you connect UNO board to your computer at the first time, right click the icon of your "Computer" —>for "Properties"—> click the "Device manager", under "Other Devices", you should see an icon for "Unknown device"

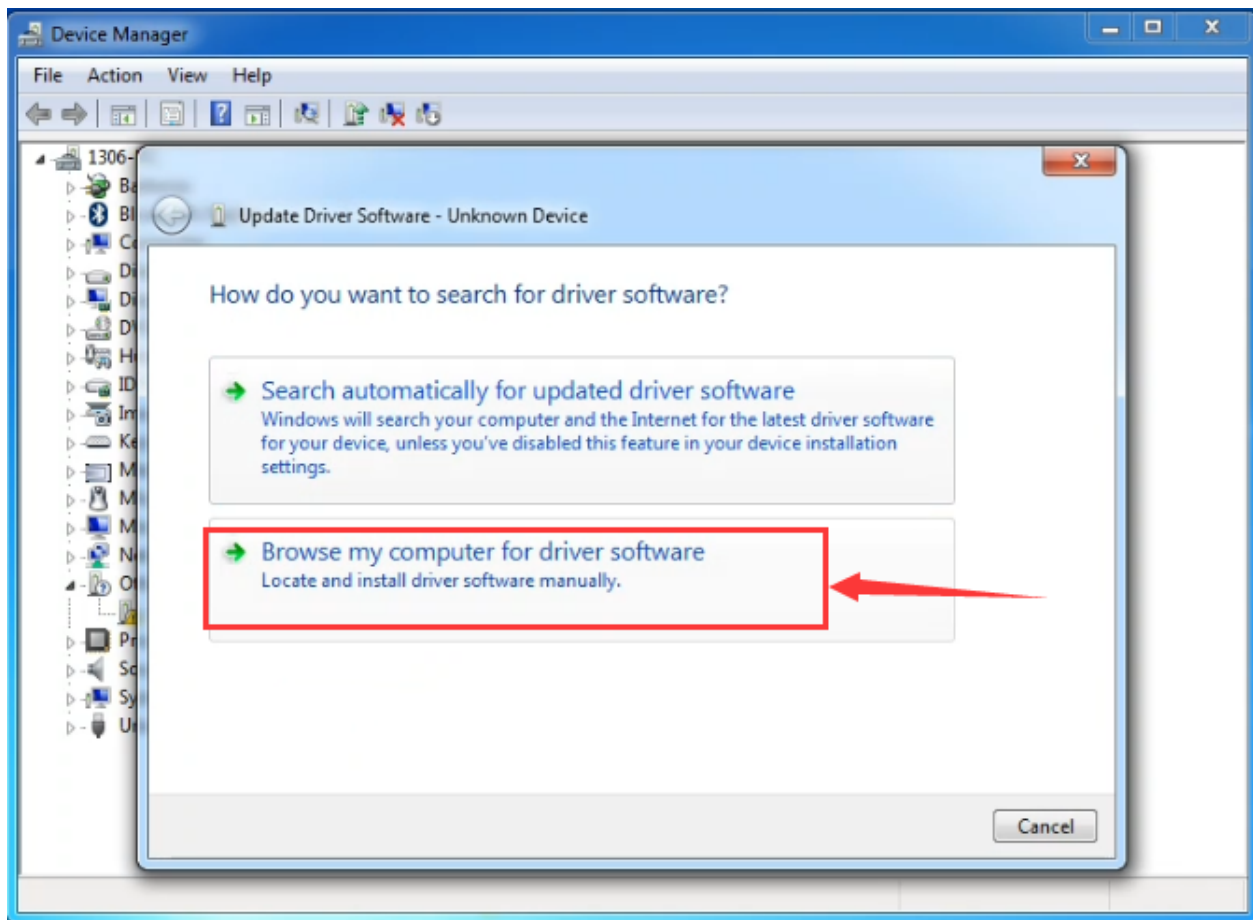
with a little yellow warning triangle next to it. This is your Arduino.



Then right-click on the device and select the top menu option (Update Driver Software...) shown as the figure below.

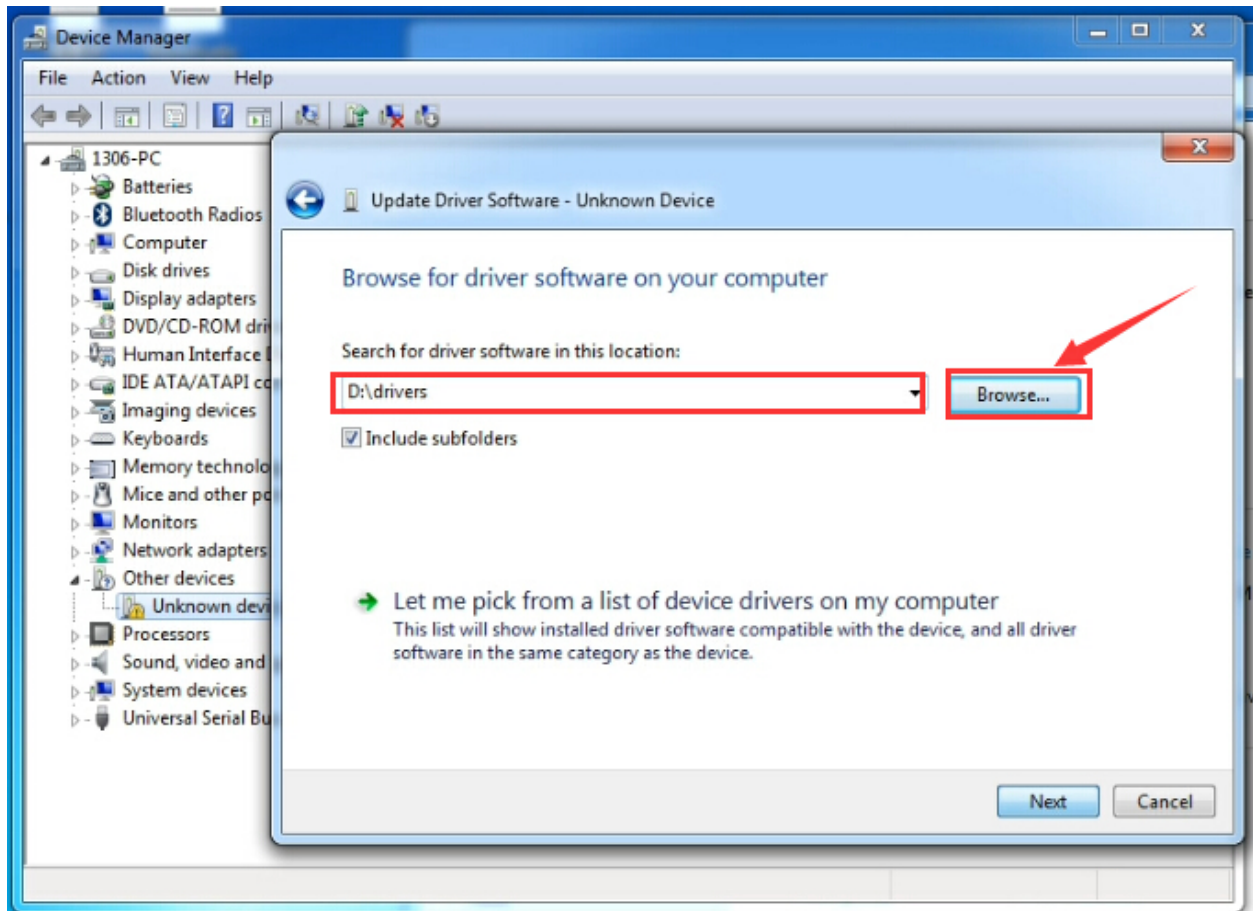


It will then be prompted to either “Search Automatically for updated driver software” or “Browse my computer for driver software”. Shown as below. In this page, select “Browse my computer for driver software”.

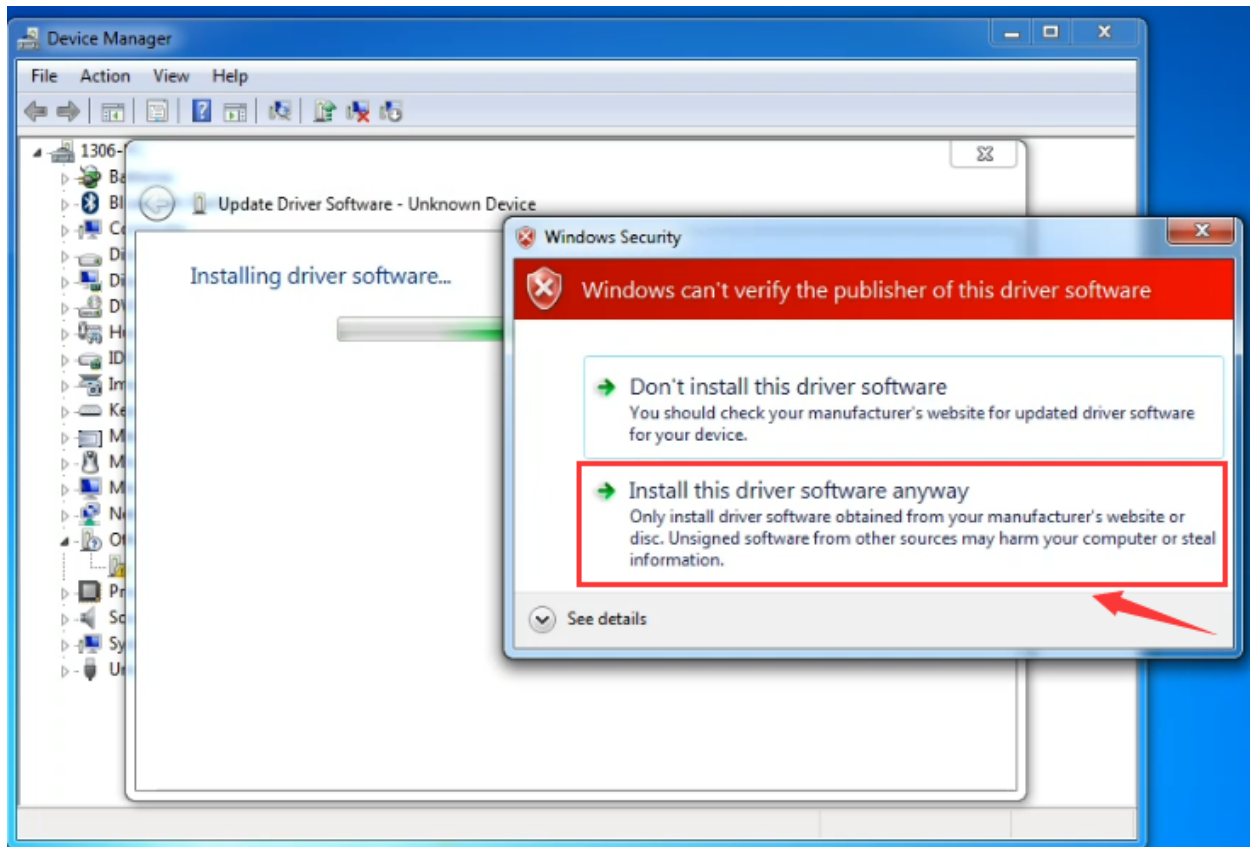


After that, select the option to browse and navigate to the “drivers” folder of Arduino installation.

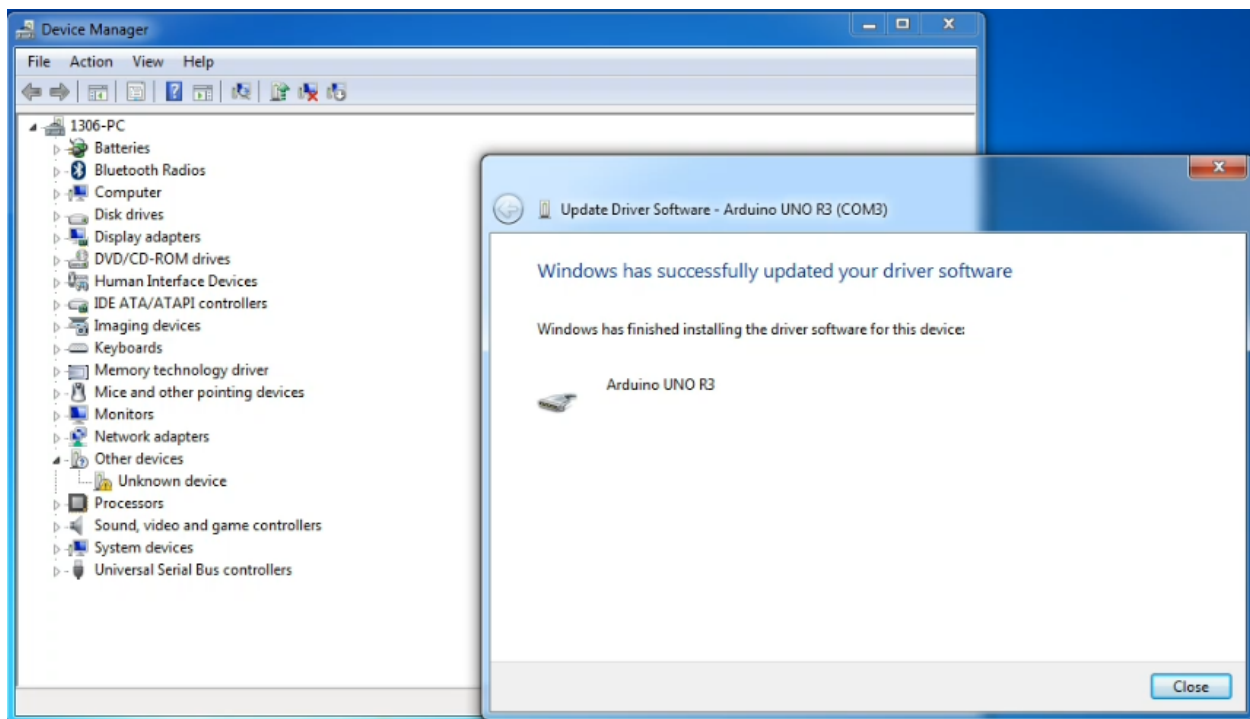




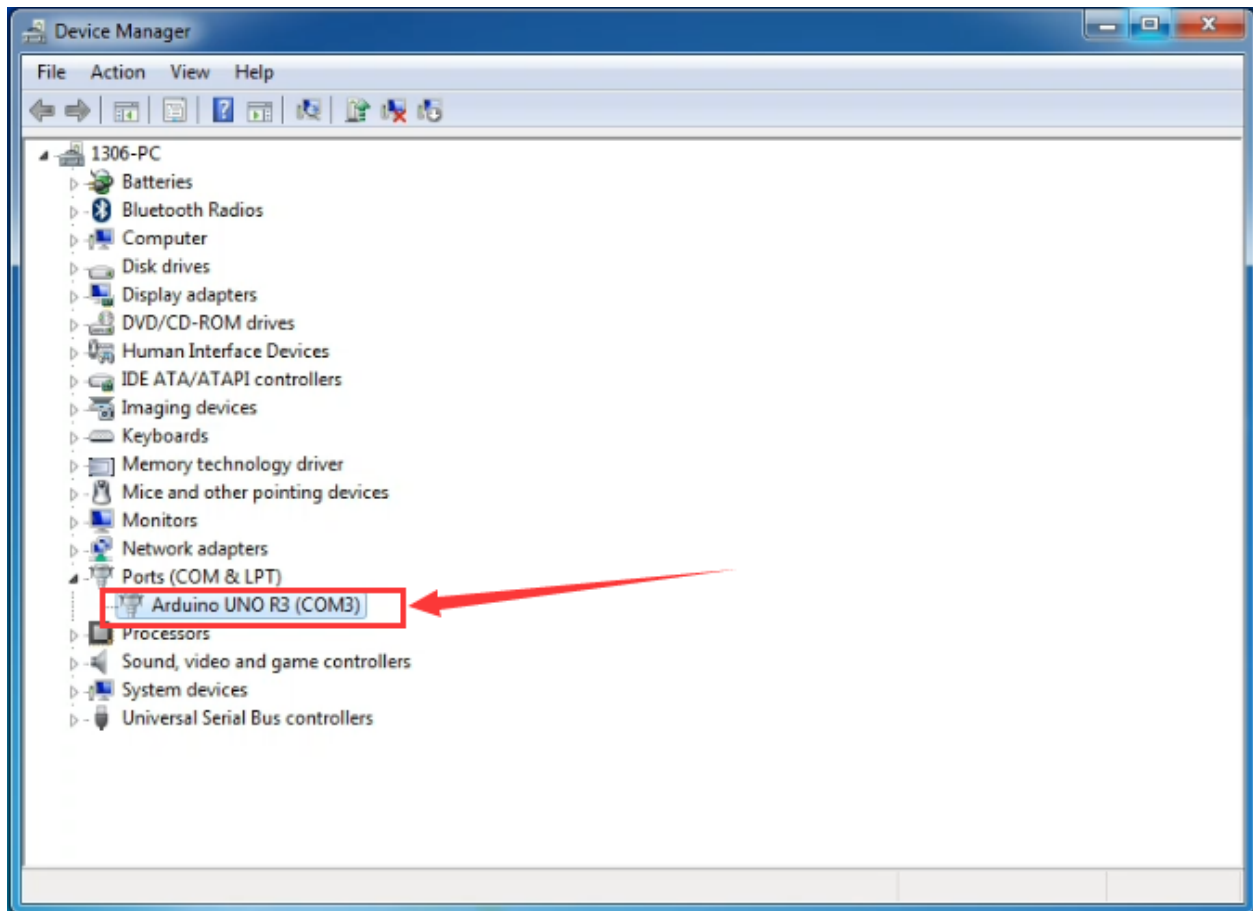
Click “Next” and you may get a security warning, if so, allow the software to be installed. Shown as below.



Installation completed, click “Close”.



Up to now, the driver is installed well. Then you can right click “Computer” —> “Properties”—> “Device manager”, you should see the device shown below.

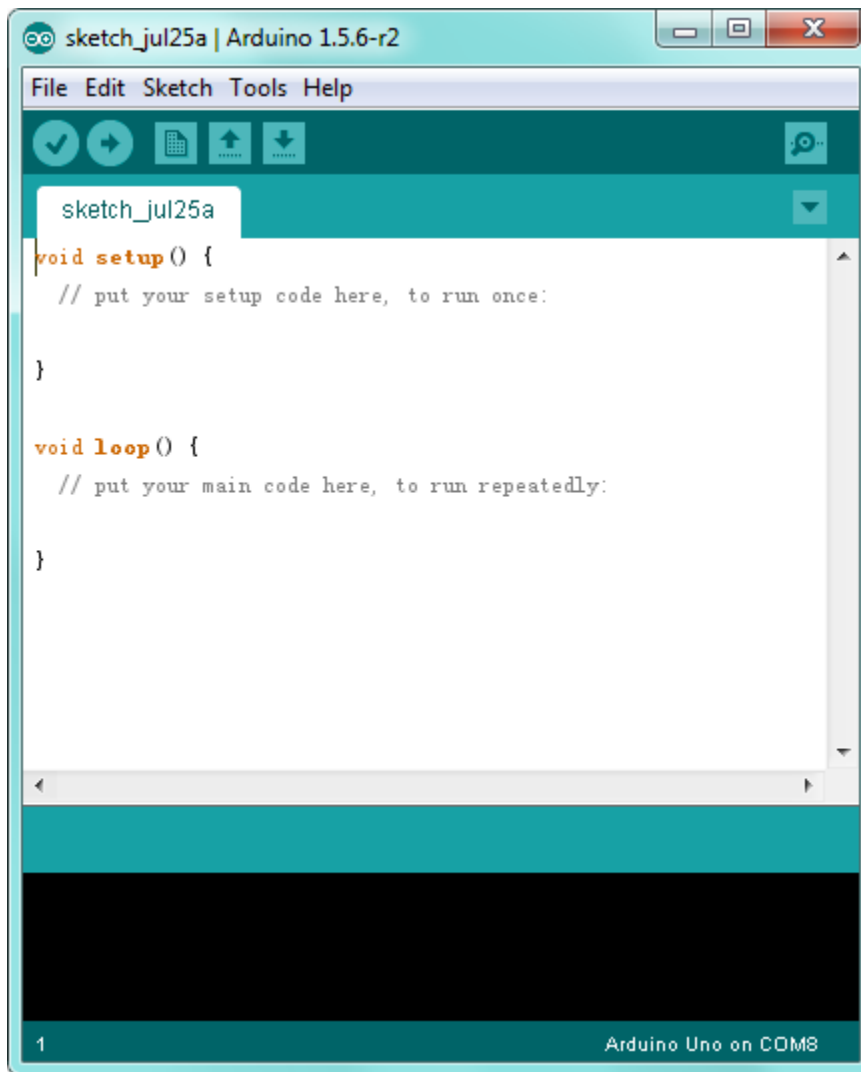


### Introduction for Arduino IDE Toolbar:

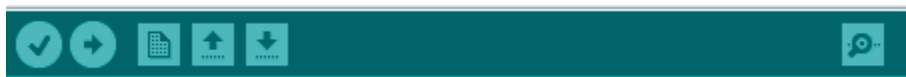
Double-click the icon of Arduino software downloaded well, you will get the interface shown below.









(Note: if the Arduino software loads in the wrong language, you can change it in the preferences dialog. See [the environment page](#) for details.)



The functions of each button on the Toolbar are listed below: <http://wiki.keyestudio.com/index.php/File:IDE.png>



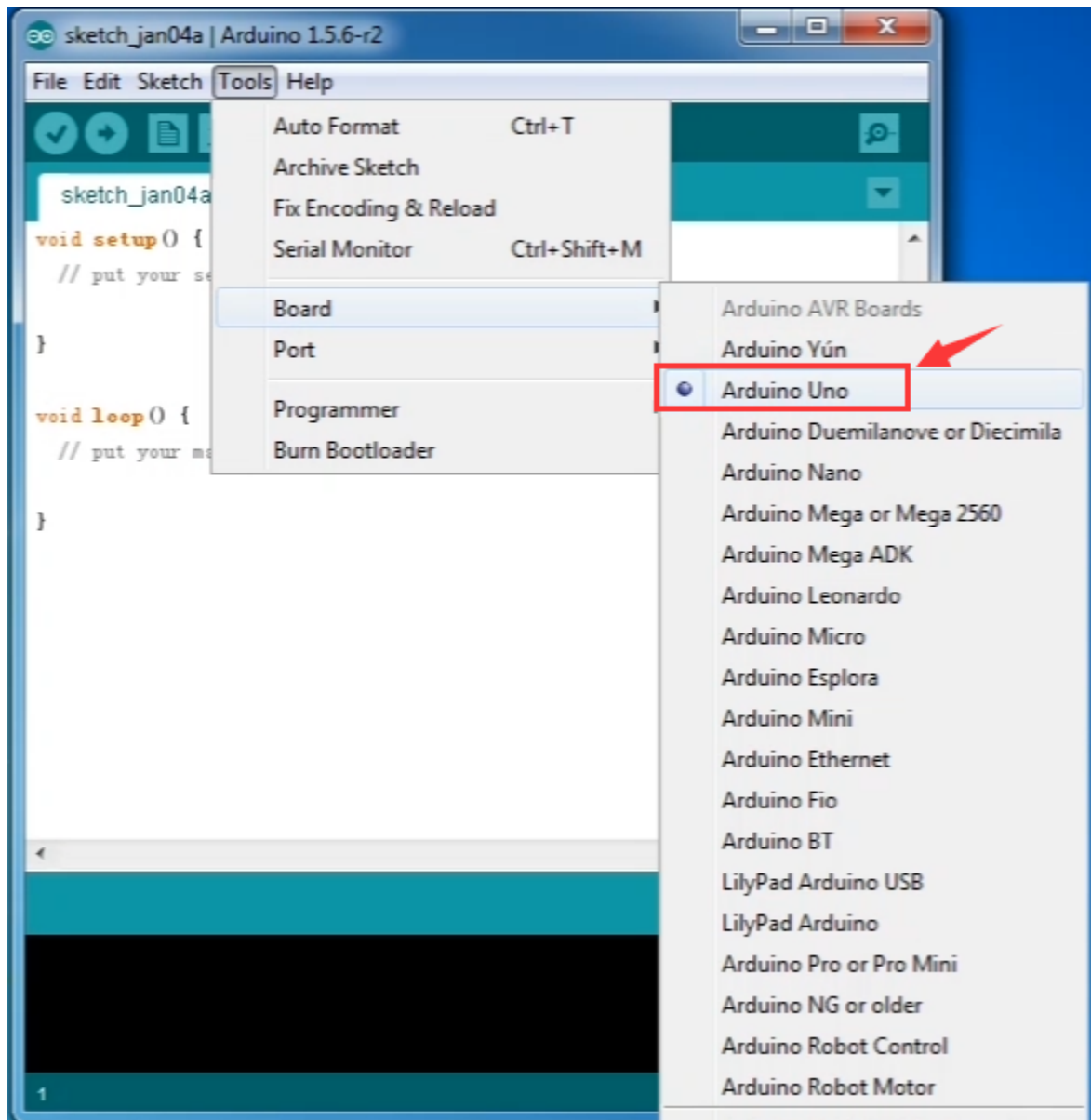
	<b>Verify/Compile</b>	Check the code for errors
	<b>Upload</b>	Upload the current Sketch to the Arduino
	<b>New</b>	Create a new blank Sketch
	<b>Open</b>	Show a list of Sketches
	<b>Save</b>	Save the current Sketch
	<b>Serial Monitor</b>	Display the serial data being sent from the Arduino

**Step2| Connect the board**

Connect the UNO board to your computer using the USB cable. The green power LED should go on.

**Step3| Select the Arduino Board**

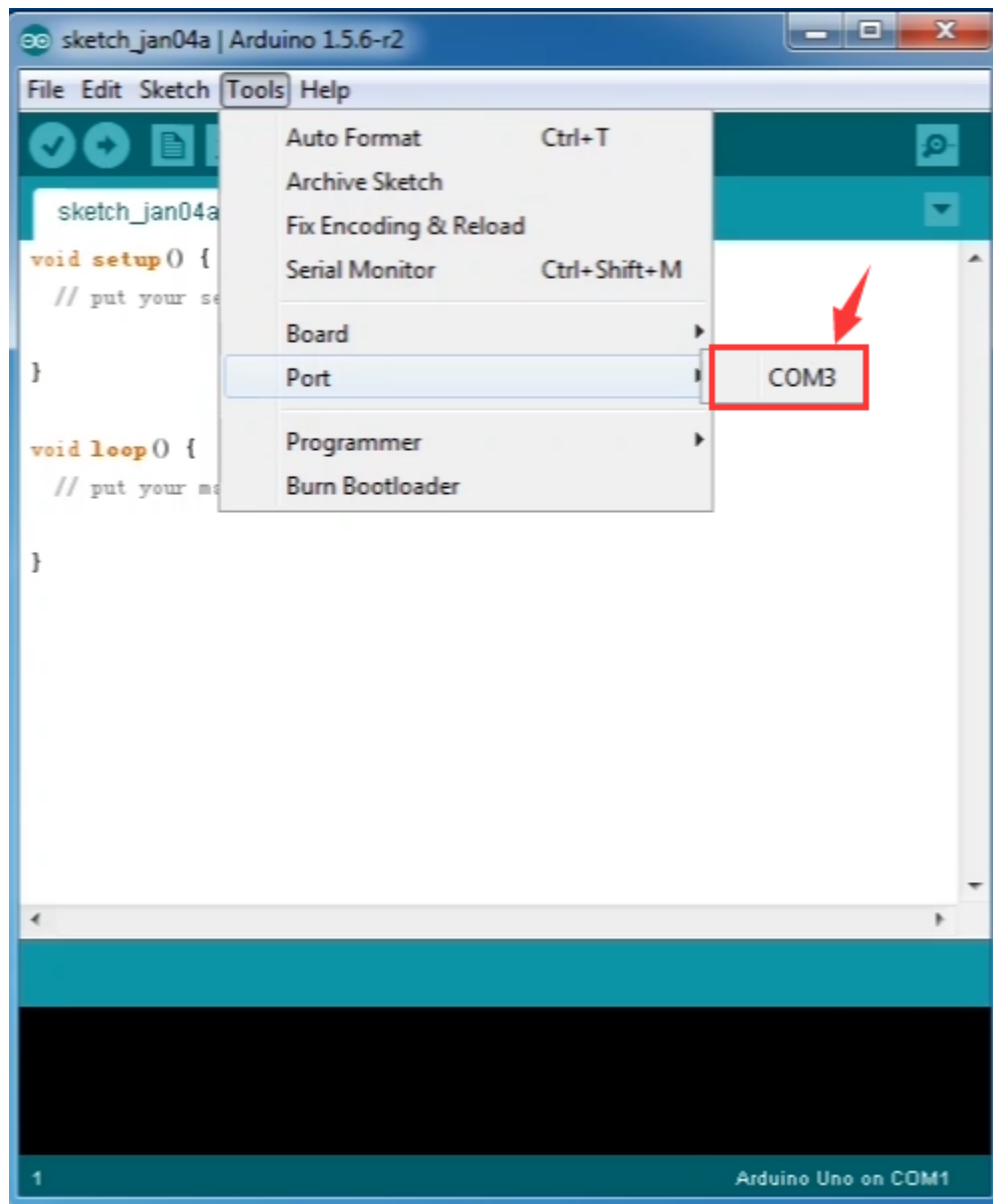
Open the Arduino IDE, you'll need to click the "Tools", then select the Board that corresponds to your Arduino.



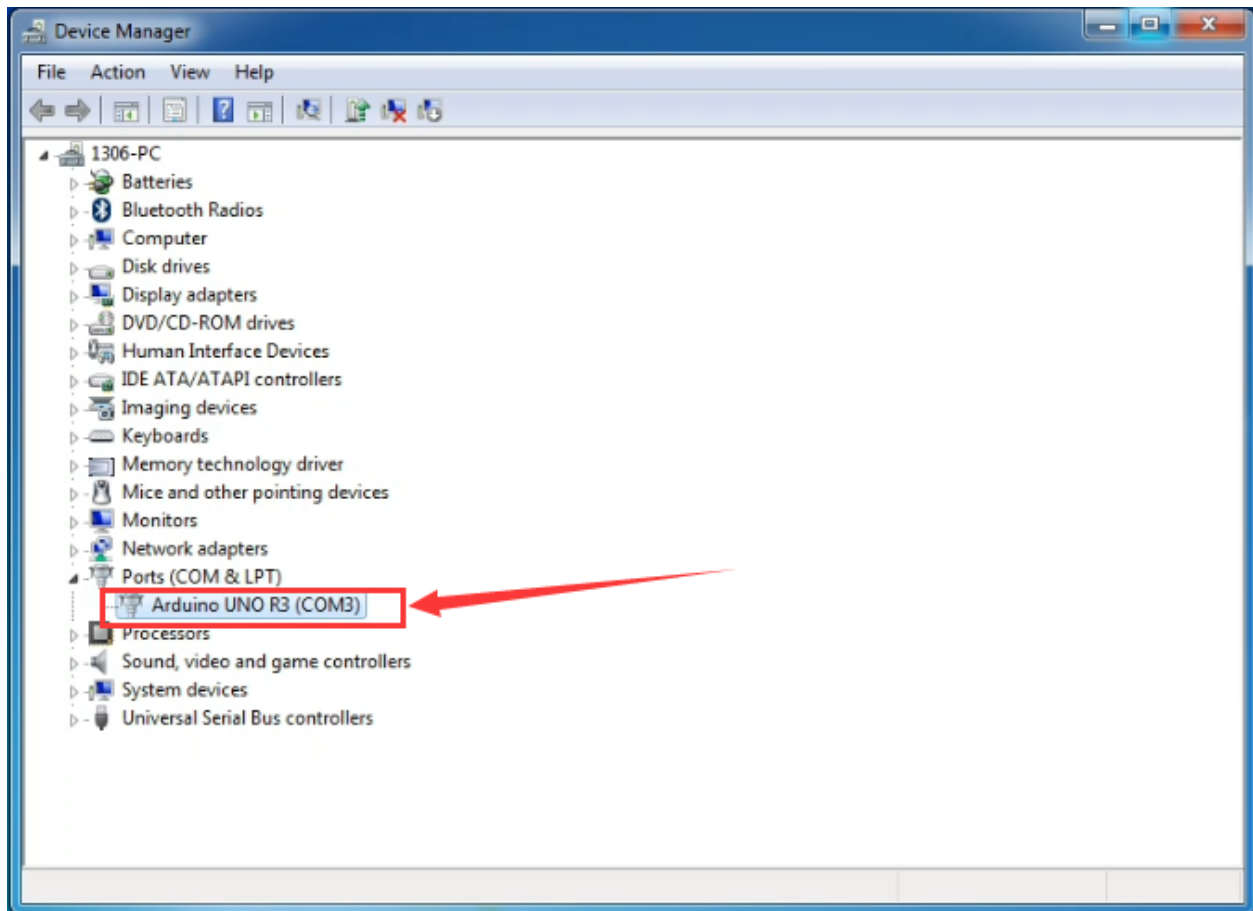
### Step4| Select your serial port

Select the serial device of the Arduino board from the **Tools | Serial Port menu**. This is likely to be COM3 or higher (COM1 and COM2 are usually reserved for hardware serial ports). To find out, you can disconnect your Arduino board and re-open the menu; the entry that disappears should be the Arduino board. Reconnect the board and select that serial port.

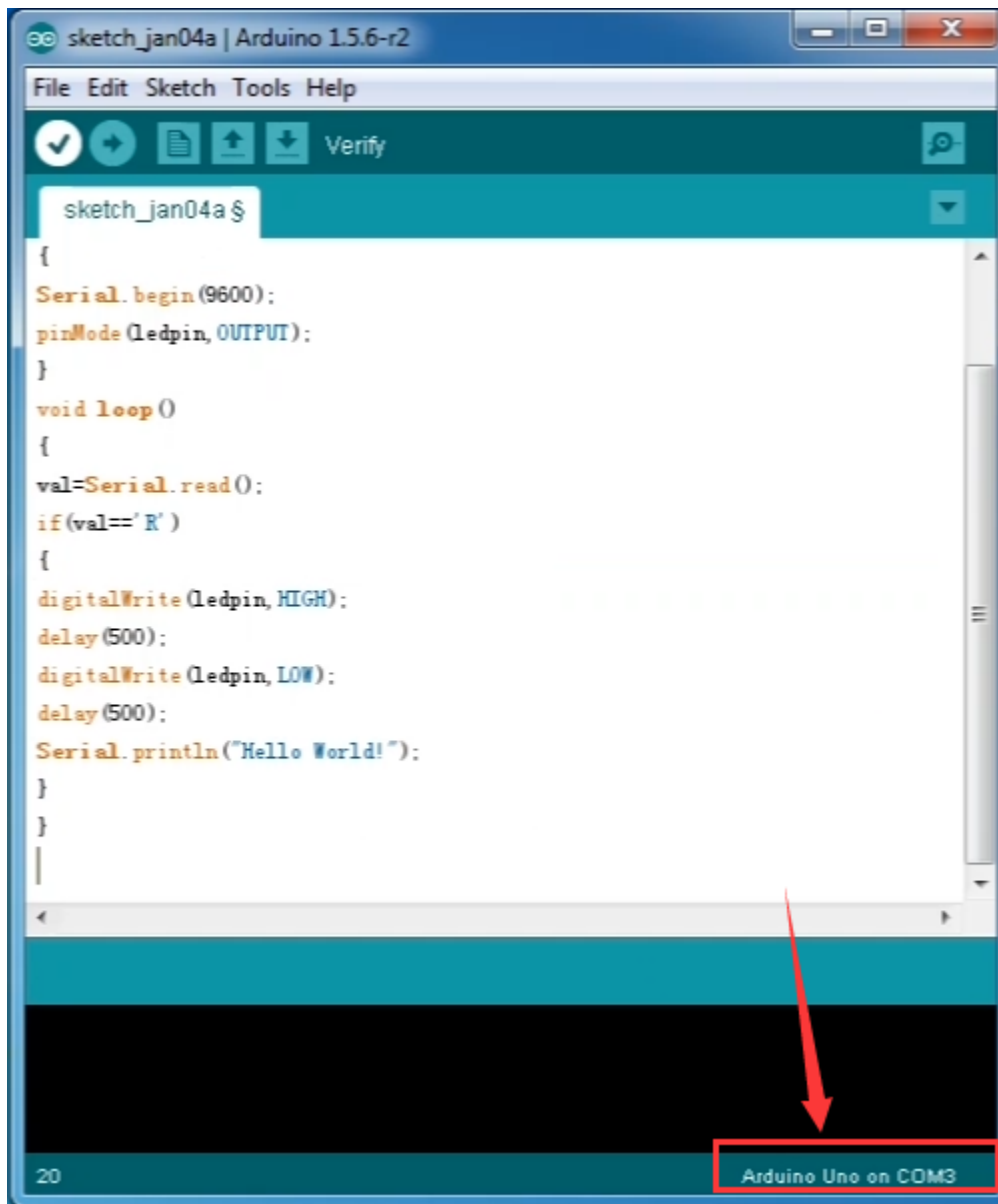
Here you should select COM 3 as below.



**Note:** to avoid errors, the COM Port should keep the same as the Ports shown on Device Manager.







### Step5|Upload the Program

Below is an example program for displaying the Hello World!

Copy and paste the code to the Arduino environment IDE.

////////////////////////////////////////////////////////////////

```
int val;

int ledpin=13;

void setup()

{
```

(continues on next page)

(continued from previous page)

```
Serial.begin(9600);  
  
pinMode(ledpin,OUTPUT);  
  
}  
  
void loop()  
{  
  val=Serial.read();  
  if(val=='R')  
  {  
    digitalWrite(ledpin,HIGH);  
    delay(500);  
    digitalWrite(ledpin,LOW);  
    delay(500);  
    Serial.println("Hello World!");  
  }  
}
```

////////////////////////////////////

Then click verify button to check the errors. If compiling successfully, the message “Done compiling.” will appear in the status bar.



After that, click the “Upload” button to upload the code. Wait a few seconds - you should see the RX and TX leds on the board flashing. If the upload is successful, the message “Done uploading.” will appear in the status bar. (Note: If you have an Arduino Mini, NG, or other board, you’ll need to physically present the reset button on the board immediately before pressing the upload button.)

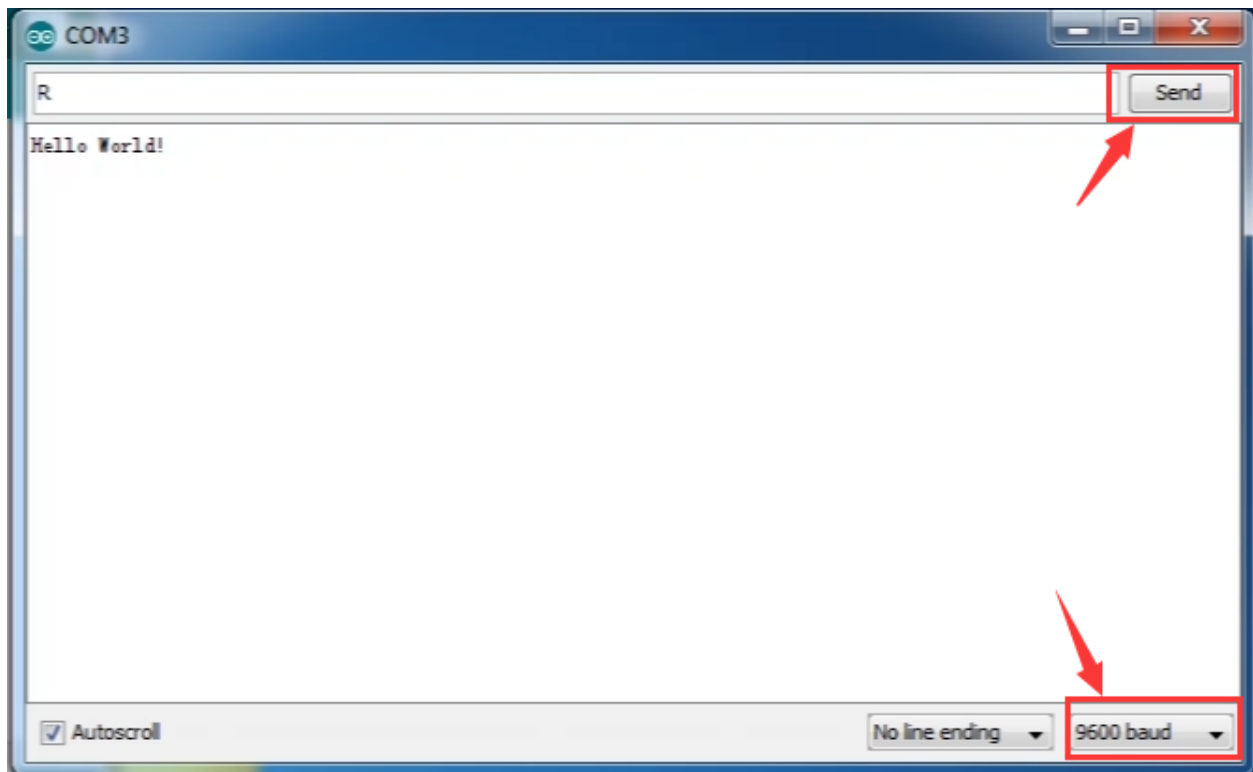


### **Step6|Open the Serial Monitor**

After that, click the serial monitor button to open the serial monitor.



Then set the baud rate as 9600, enter an “R” and click Send, you should see the RX led on the board blink once, and then D13 led blink once, finally “Hello World!” is showed on the monitor, the TX led blink once.



### Other Links:

You might also want to look at:

[the examples](#) for using various sensors and actuators;

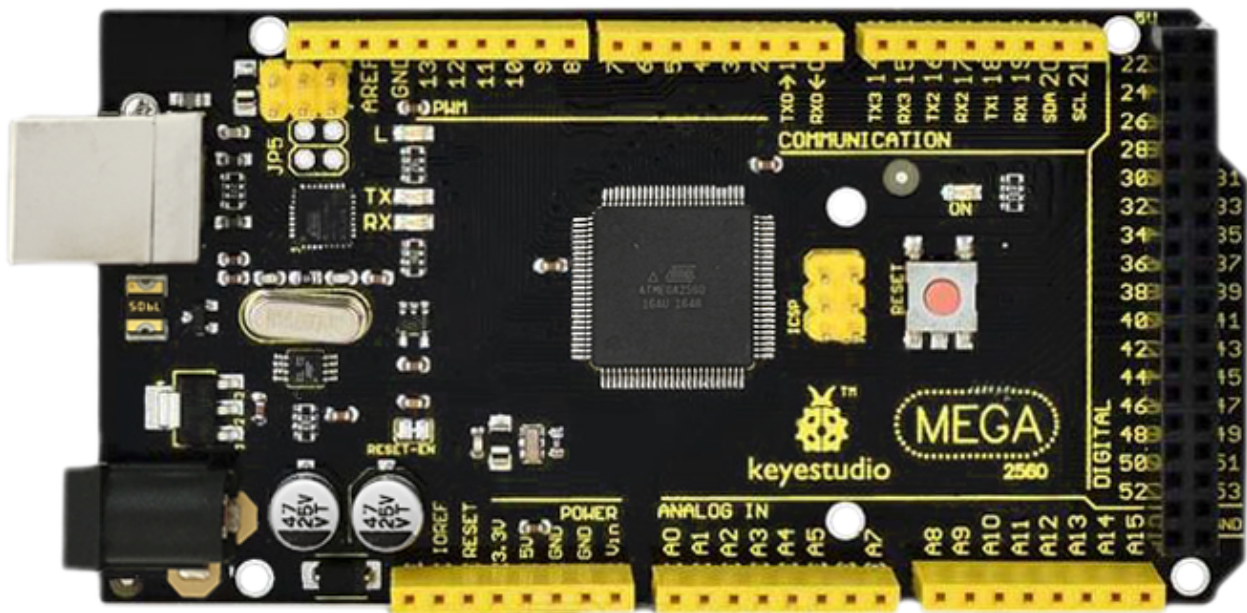
[the reference](#) for the Arduino language;

The text of the Arduino getting started guide is licensed under a [Creative Commons Attribution-ShareAlike 3.0 License](#). Code samples in the guide are released into the public domain.

### Troubleshooting:

If you have problems, please see the [troubleshooting suggestions](#).

## KS0002 KEYESTUDIO MEGA 2560 R3 DEVELOPMENT BOARD



### Introduction:

Keyestudio Mega 2560 R3 is a microcontroller board based on the [ATMEGA2560-16AU](#) , fully compatible with ARDUINO MEGA 2560 REV3.

It has 54 digital input/output pins (of which 15 can be used as PWM outputs), 16 analog inputs, 4 UARTs (hardware serial ports), a 16 MHz crystal oscillator, a USB connection, a power jack, 2 ICSP headers, and a reset button.

It contains everything needed to support the microcontroller. With its bootloader, program can be downloaded directly with USB and you don't need to use other external programmer.

Just simply connect it to a computer with a USB cable or power it with a AC-to-DC adapter or battery to get started.

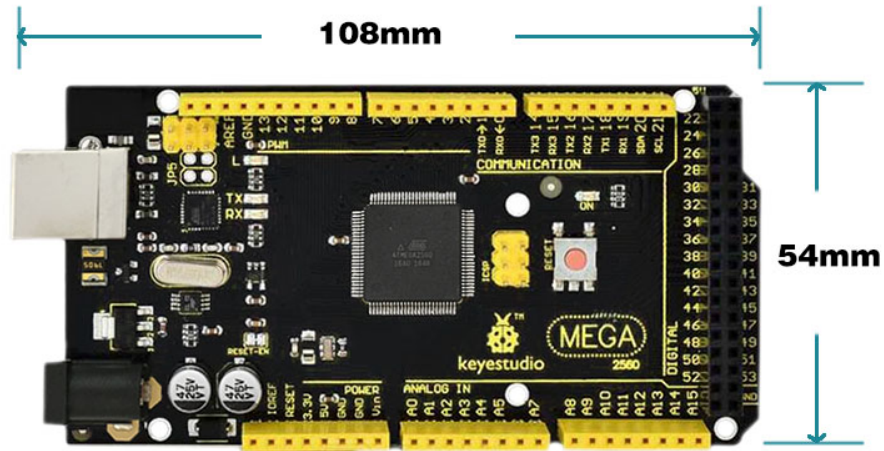
The Mega 2560 board is compatible with most shields designed for the Uno.

The 2560 R3 differs from all preceding boards in that it does not use the FTDI USB-to-serial driver chip. Instead, it features the Atmega16U2 programmed as a USB-to-serial converter.

The MEGA 2560 is designed for more complex projects. With 54 digital I/O pins, 16 analog inputs and a larger space for your sketch, it is the recommended board for 3D printers and robotics projects. This gives your projects plenty of room and opportunities.

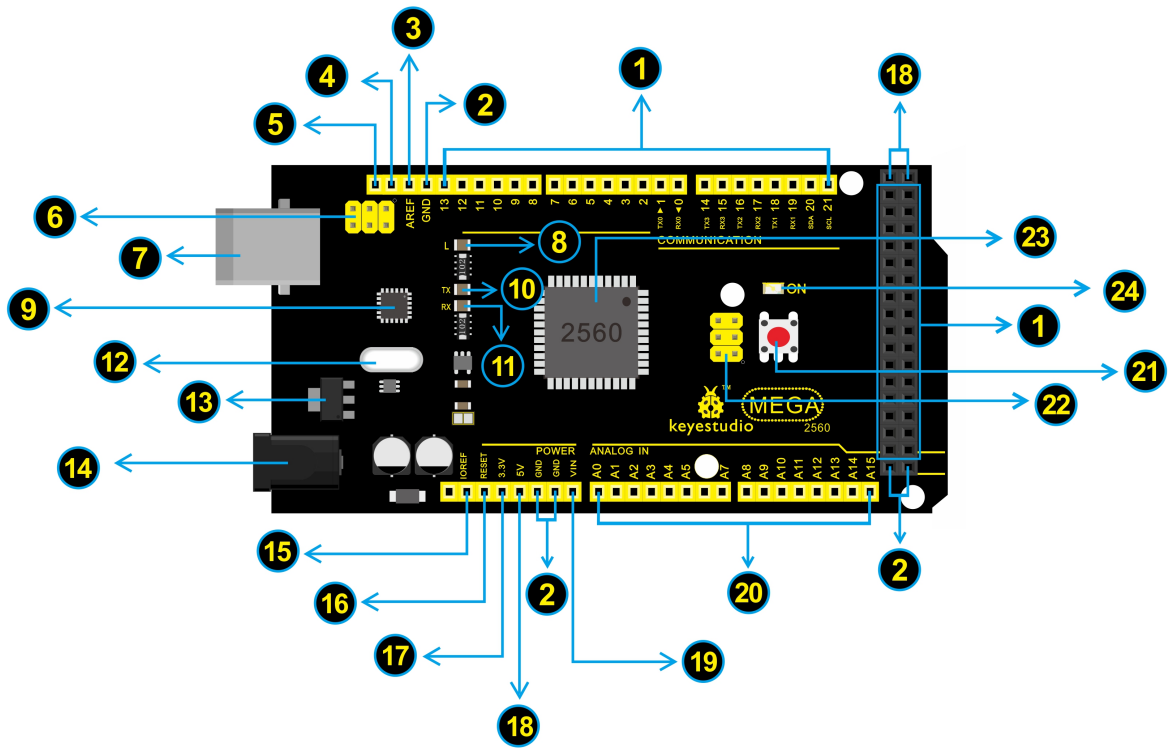
### TECH SPECS:

<b>Microcontroller</b>	ATMEGA2560-16AU
<b>Operating Voltage</b>	5V
<b>Input Voltage (recommended)</b>	DC 7-12V
<b>Digital I/O Pins</b>	54 (D0-D53)
<b>PWM Digital I/O Pins</b>	15 (D2-D13; D44-D46)
<b>Analog Input Pins</b>	16 (A0-A15)
<b>DC Current per I/O Pin</b>	20 mA
<b>DC Current for 3.3V Pin</b>	50 mA
<b>Flash Memory</b>	256 KB of which 8 KB used by bootloader
<b>SRAM</b>	8 KB
<b>EEPROM</b>	4 KB
<b>Clock Speed</b>	16 MHz
<b>LED_BUILTIN</b>	D13



Element and Pin Interfaces:

Dimensions:



is an explanation of what every element and interface of the board does:

Here



1	<b>Digital I/O</b> Arduino MEGA has 54 digital input/output pins (of which 15 can be used as PWM outputs). These pins can be configured as digital input pin to read the logic value (0 or 1). Or used as digital output pin to drive different modules like LED, relay, etc. Using <code>pinMode()</code> , <code>digitalWrite()</code> , and <code>digitalRead()</code> functions.
2	<b>GND</b> Ground pins
3	<b>AREF</b> Reference voltage (0-5V) for analog inputs. Used with <code>analogReference()</code> . Configures the reference voltage used for analog input (i.e. the value used as the top of the input range).
4	<b>SDA</b> IIC communication pin
5	<b>SCL</b> IIC communication pin
6	<b>ICSP (In-Circuit Serial Programming) Header</b> the AVR, an Arduino micro-program header consisting of MOSI, MISO, SCK, RESET, VCC, and GND. Connected to the ATMEGA16U2-MU. When connecting to PC, program the firmware to ATMEGA16U2-MU.
7	<b>USB Connection</b> Arduino board can be powered via USB connector. All you needed to do is connecting the USB port to PC using a USB cable.
8	<b>D13 LED</b> 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.
9	<b>ATMEGA 16U2-MU</b> USB to serial chip, can convert the USB signal into serial port signal.
10	<b>TX LED</b> Onboard you can find the label: TX (transmit) When Arduino board communicates via serial port, send the message, TX led flashes.
11	<b>RX LED</b> Onboard you can find the label: RX(receive ) When Arduino board communicates via serial port, receive the message, RX led flashes.
12	<b>Crystal Oscillator</b> How does Arduino calculate time? by using a crystal oscillator. The number printed on the top of the Arduino crystal is 16.000H9H. It tells us that the frequency is 16,000,000 Hertz or 16MHz.
13	<b>Voltage Regulator</b> To control the voltage provided to the Arduino board, as well as to stabilize the DC voltage used by the processor and other components. Convert an external input DC7-12V voltage into DC 5V, then switch DC 5V to the processor and other components.
14	<b>DC Power Jack</b> Arduino board can be supplied with an external power DC7-12V from the DC power jack.
15	<b>IOREF</b> This pin on the 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 for working with the 5V or 3.3V.
16	<b>RESET Header</b> Connect an external button to reset the board. The function is the same as reset button.
17	<b>Power Pin 3V3</b> A 3.3 volt supply generated by the on-board regulator. Maximum current draw is 50 mA.
18	<b>Power Pin 5V</b> Provides 5V output voltage
19	<b>Vin</b> You can supply an external power input DC7-12V through this pin to Arduino board.
20	<b>Analog Pins</b> Onboard has 16 analog inputs, labeled A0 to A15.
21	<b>RESET Button</b> You can reset your Arduino board, for example, start the program from the initial status. You can use the RESET button.
22	<b>ICSP (In-Circuit Serial Programming) Header</b> the AVR, an Arduino micro-program header consisting of MOSI, MISO, SCK, RESET, VCC, and GND. It is often called the SPI (serial peripheral interface) and can be considered an “extension” of the output. In fact, slave the output devices to the SPI bus host. When connecting to PC, program the firmware to ATMEGA2560-16AU.
23	<b>Microcontroller</b> Each Arduino board has its own microcontroller. You can regard it as the brain of your board. The main IC (integrated circuit) on the Arduino is slightly different from the panel pair. Microcontrollers are usually from ATMEL. Before you load a new program on the Arduino IDE, you must know what IC is on your board. This information can be checked at the top of IC.
24	

### Specialized Functions of Some Pins:

- **Serial Communication:** D0 (RX0) and D1 (TX1); Serial 1: D19 (RX1) and D18 (TX1); Serial 2: D17 (RX2) and D16 (TX2); Serial 3: D15 (RX3) and D14 (TX3).

Used to receive (RX) and transmit (TX) TTL serial data. Pins 0 and 1 are also connected to the corresponding pins of the ATmega16U2 USB-to-TTL Serial chip.

- **PWM Pins (Pulse-Width Modulation):** D2 to D13, and D44 to D46.

Provide 8-bit PWM output with the `analogWrite()` function.

- **External Interrupts:** D2 (interrupt 0), D3 (interrupt 1), D18 (interrupt 5), D19 (interrupt 4), D20 (interrupt 3), and D21 (interrupt 2).

These pins can be configured to trigger an interrupt on a low level, a rising or falling edge, or a change in level. See the `attachInterrupt()` function for details.

- **SPI communication:** D53 (SS), D52 (SCK), D51 (MOSI), D50 (MISO).

These pins support SPI communication using the [SPI library](#). The SPI pins are also broken out on the ICSP header, which is physically compatible with the Arduino Uno.

- **IIC communication:** D20 (SDA); D21 (SCL). Support TWI communication using the [Wire library](#).

### Warnings:

- The Mega 2560 has a resettable polyfuse that protects your computer's USB ports from shorts and overcurrent. If more than 500 mA is applied to the USB port, the fuse will automatically break the connection until the short or overload is removed.

- Automatic (Software) Reset:

Rather than requiring a physical press of the reset button before an upload, the Mega 2560 board is designed in a way that allows it to be reset by software running on a connected computer.

- The Mega 2560 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; see [this forum thread](#) for details.

### Detailed Use with ARDUINO Software as follows:

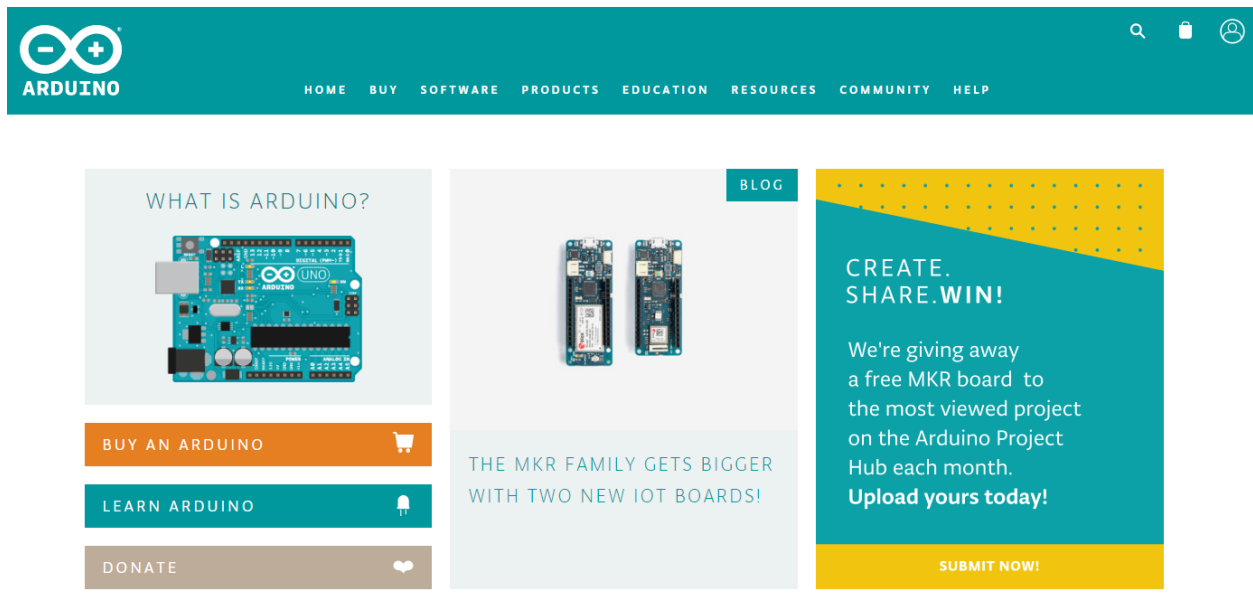
#### Step1 | Download the Arduino environment (IDE)

When you get the Mega 2560 development board, first you should install the Arduino software and driver.

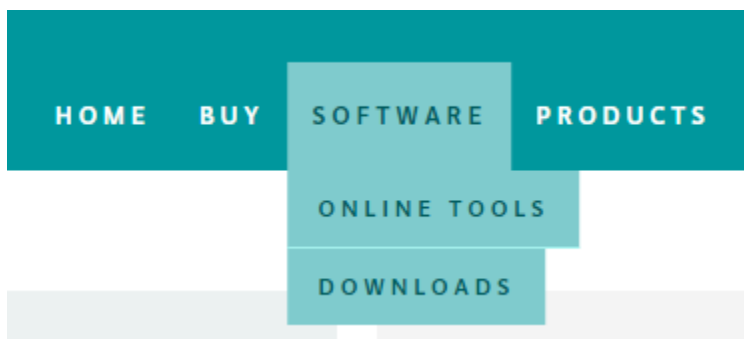
We usually use the Windows software Arduino 1.5.6 version. You can download it from the link below:

<https://www.arduino.cc/en/Main/OldSoftwareReleases#1.5.x>

Or you can browse the ARDUINO website to download the latest version from this link, <https://www.arduino.cc>, pop up the following interface.




Then click the **SOFTWARE** on the browse bar, you will have two options **ONLINE TOOLS** and **DOWNLOADS**.



Click **DOWNLOADS**, it will appear the latest software version of ARDUINO 1.8.5 shown as below.

## Download the Arduino IDE



**ARDUINO 1.8.5**

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** 10.7 Lion or newer

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

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

In this software page, on the right side you can see the version of development software for different operating systems. ARDUINO has a powerful compatibility. You should download the software that is compatible with the operating

system of your computer.

We will take **WINDOWS system** as an example here. There are also two options under Windows system, one is installed version, the other is non-installed version.

For simple installed version, first click **Windows Installer**, you will get the following page.



## Contribute to the Arduino Software

Consider supporting the Arduino Software by contributing to its development. (US tax payers, please note this contribution is not tax deductible). [Learn more on how your contribution will be used.](#)



This way you just need to click JUST DOWNLOAD, then click the downloaded file to install it.

For non-installed version, first click Windows ZIP file, you will also get the pop-up interface as the above figure.

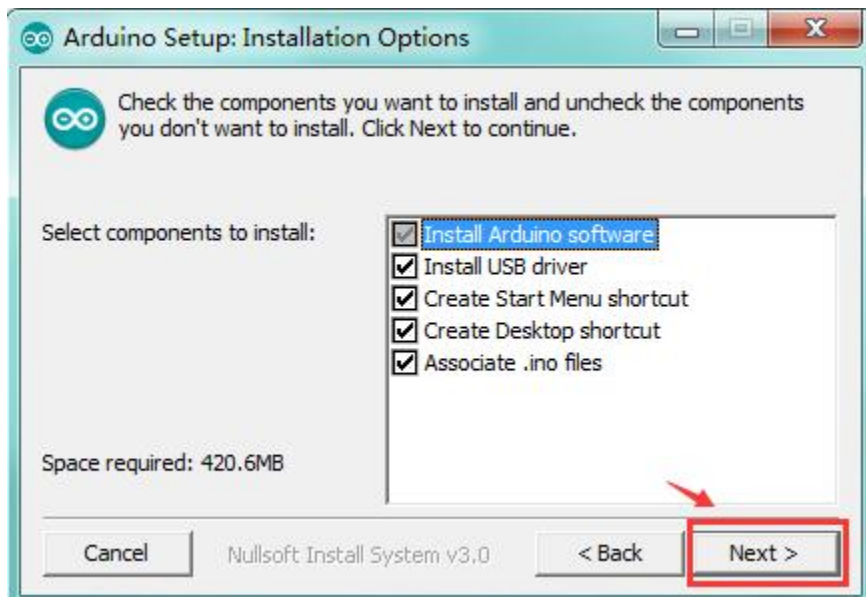
Click JUST DOWNLOAD, and when the ZIP file is downloaded well to your computer, you can directly unzip the file and click the icon of ARDUINO software to start it.

### Installing Arduino (Windows):

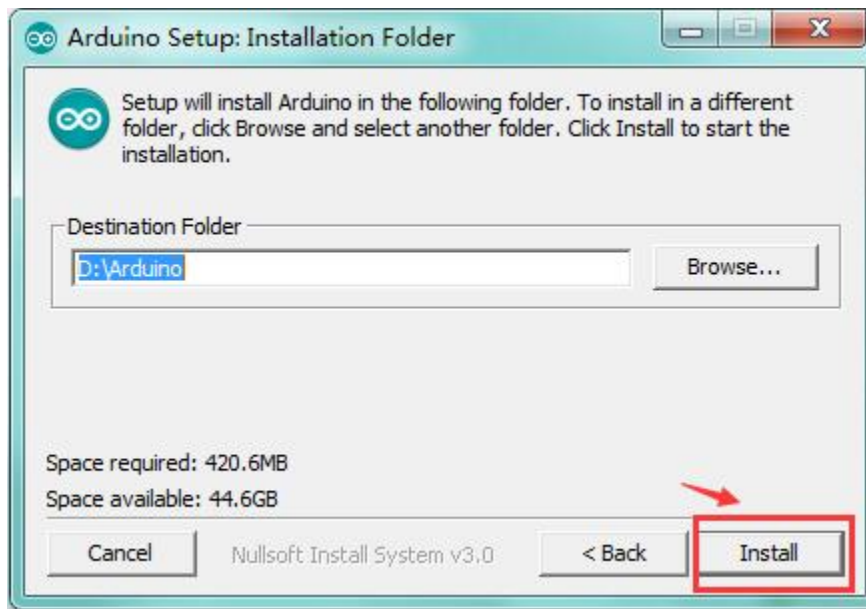
Install Arduino with the exe. Installation package downloaded well.



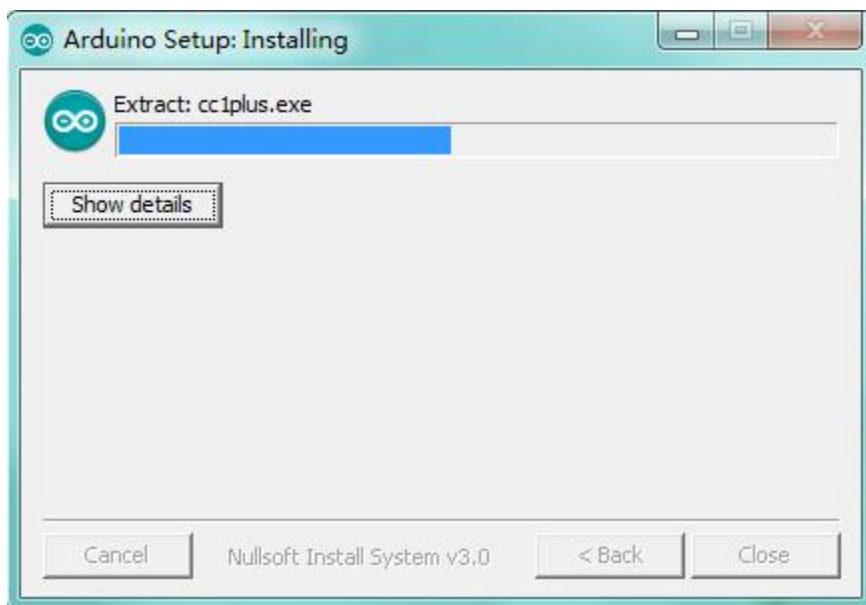
Click “I Agree” to see the following interface.



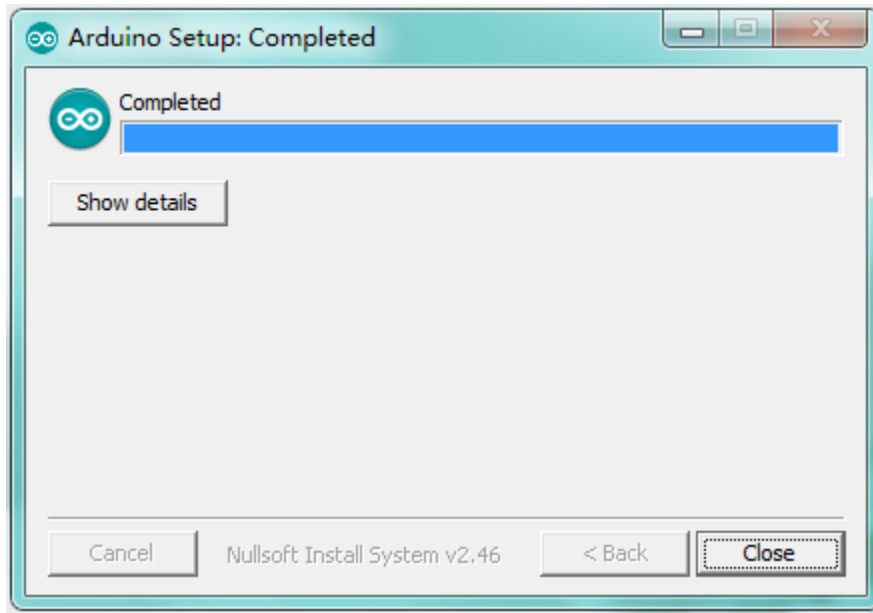
Click “Next”. Pop up the interface below.



You can press Browse... to choose an installation path or directly type in the directory you want. Then click “Install” to initiate installation.



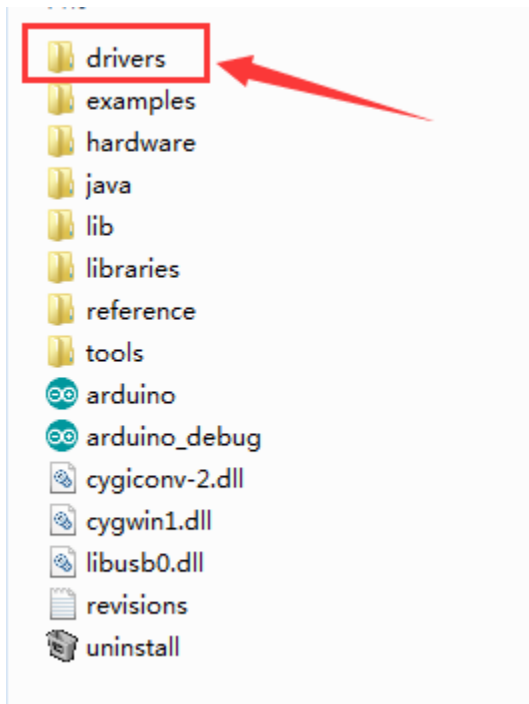
Wait for the installing process, if appear the interface of Window Security, just continue to click Install to finish the installation.



### Installing Driver:

Next, we will introduce the driver installation of Mega 2560 development board. The driver installation may have slight differences in different computer systems. So in the following let's move on to the driver installation in the WIN 7 system.

The Arduino folder contains both the Arduino program itself and the drivers that allow the Arduino to be connected to your computer by a USB cable. Before we launch the Arduino software, you are going to install the USB drivers.

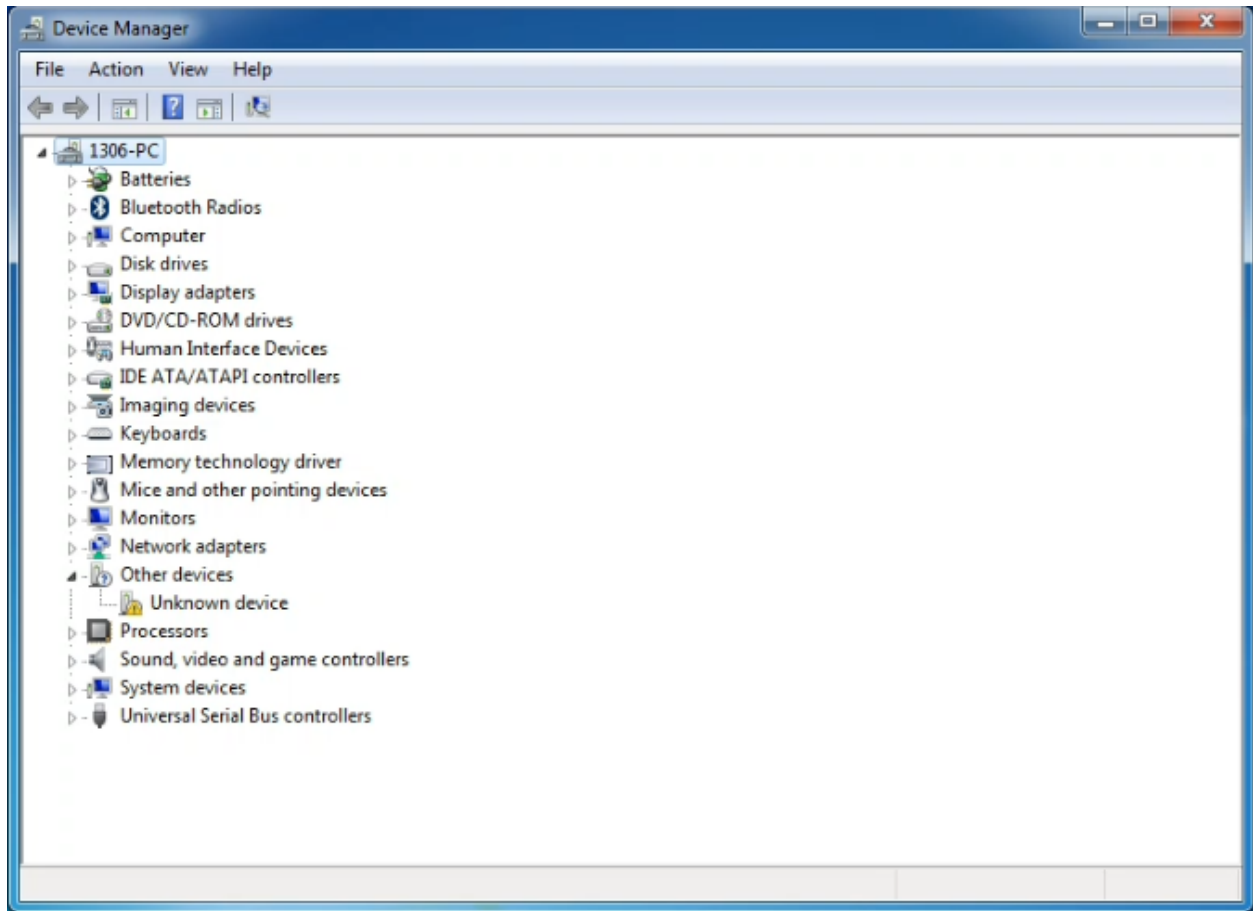


Plug one end of your USB cable into the Arduino and the other into a USB socket on your computer.

When you connect Mega 2560 board to your computer at the first time, right click the icon of your "Computer" —>for "Properties"—> click the "Device manager", under "Other Devices", you should see an icon for "Unknown device"

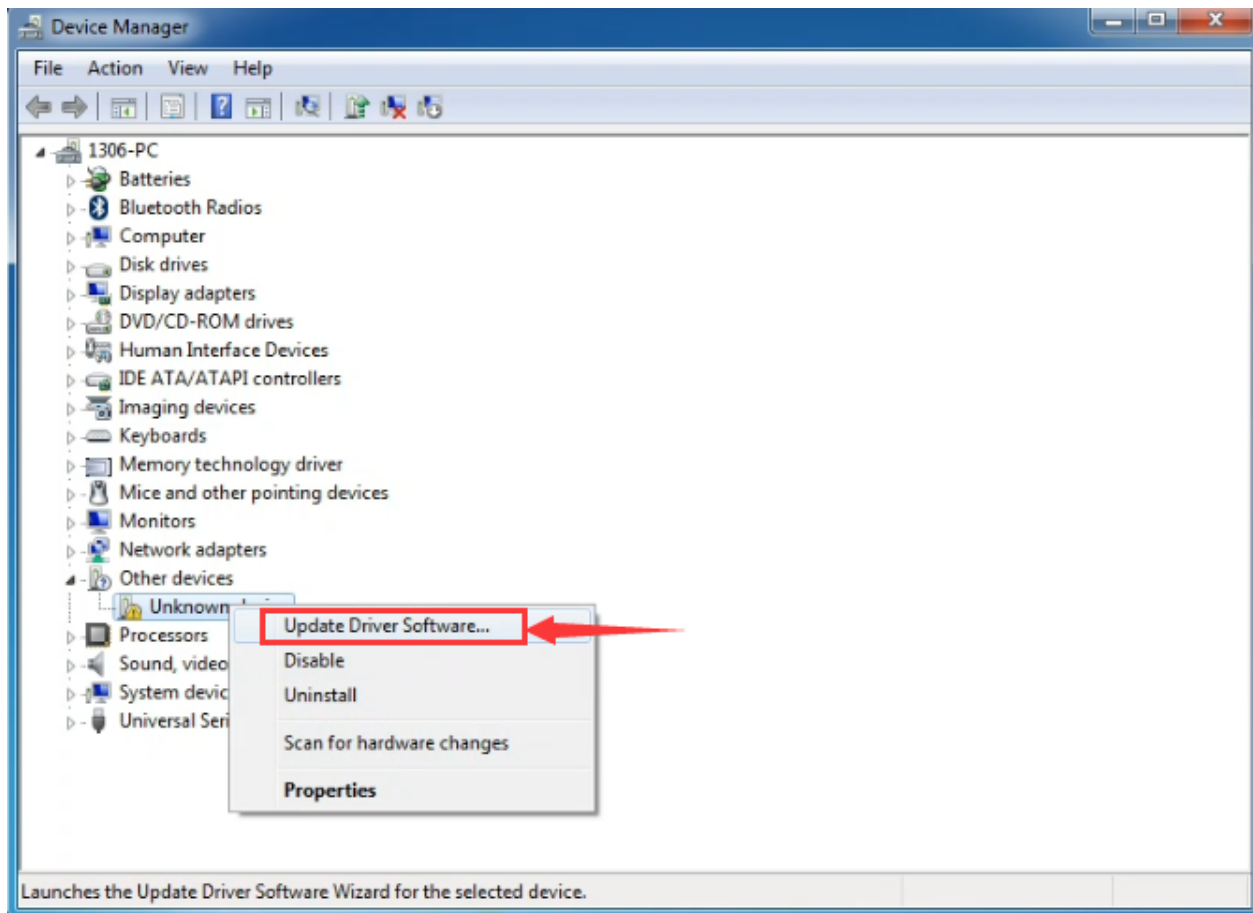


with a little yellow warning triangle next to it. This is your Arduino.

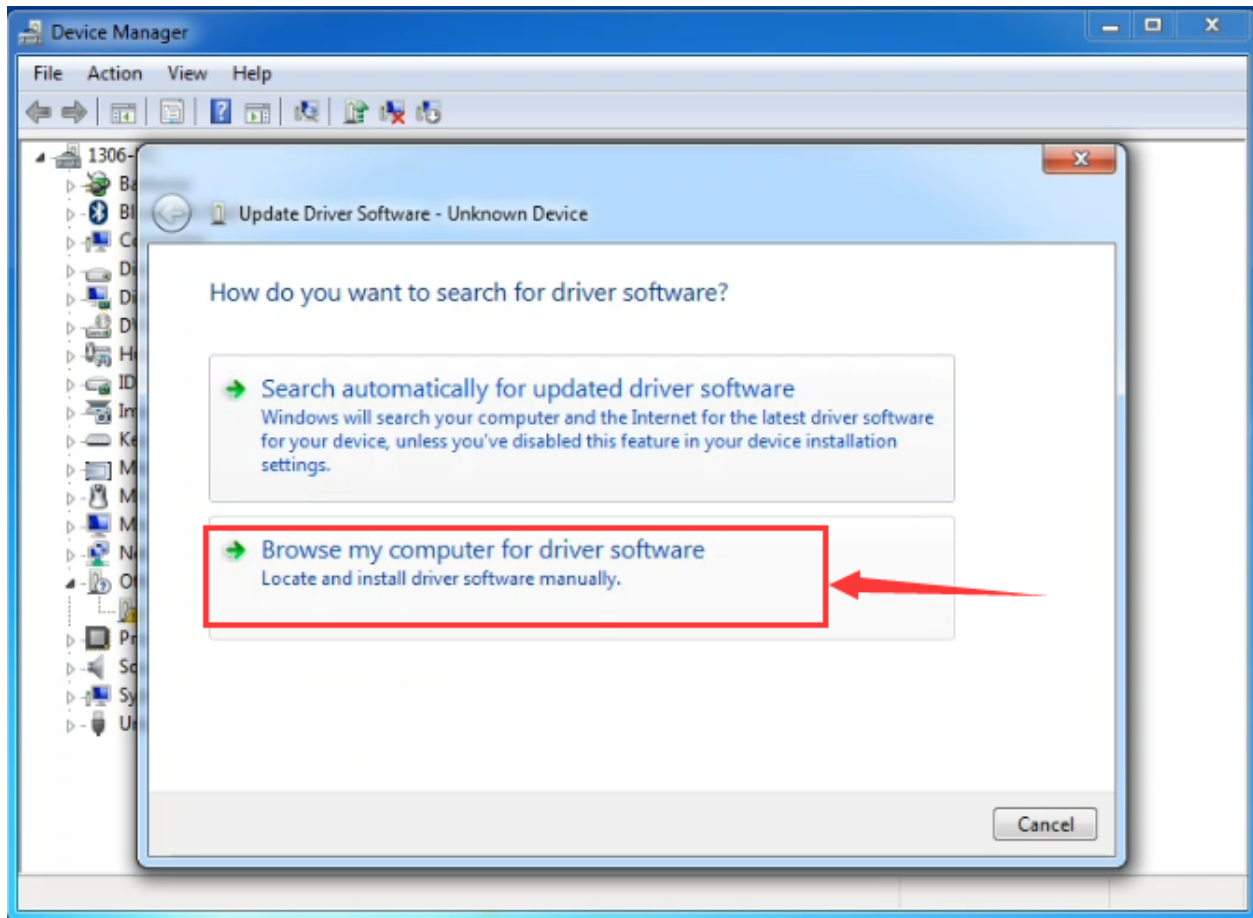


Then right-click on the device and select the top menu option (Update Driver Software...) shown as the figure below.

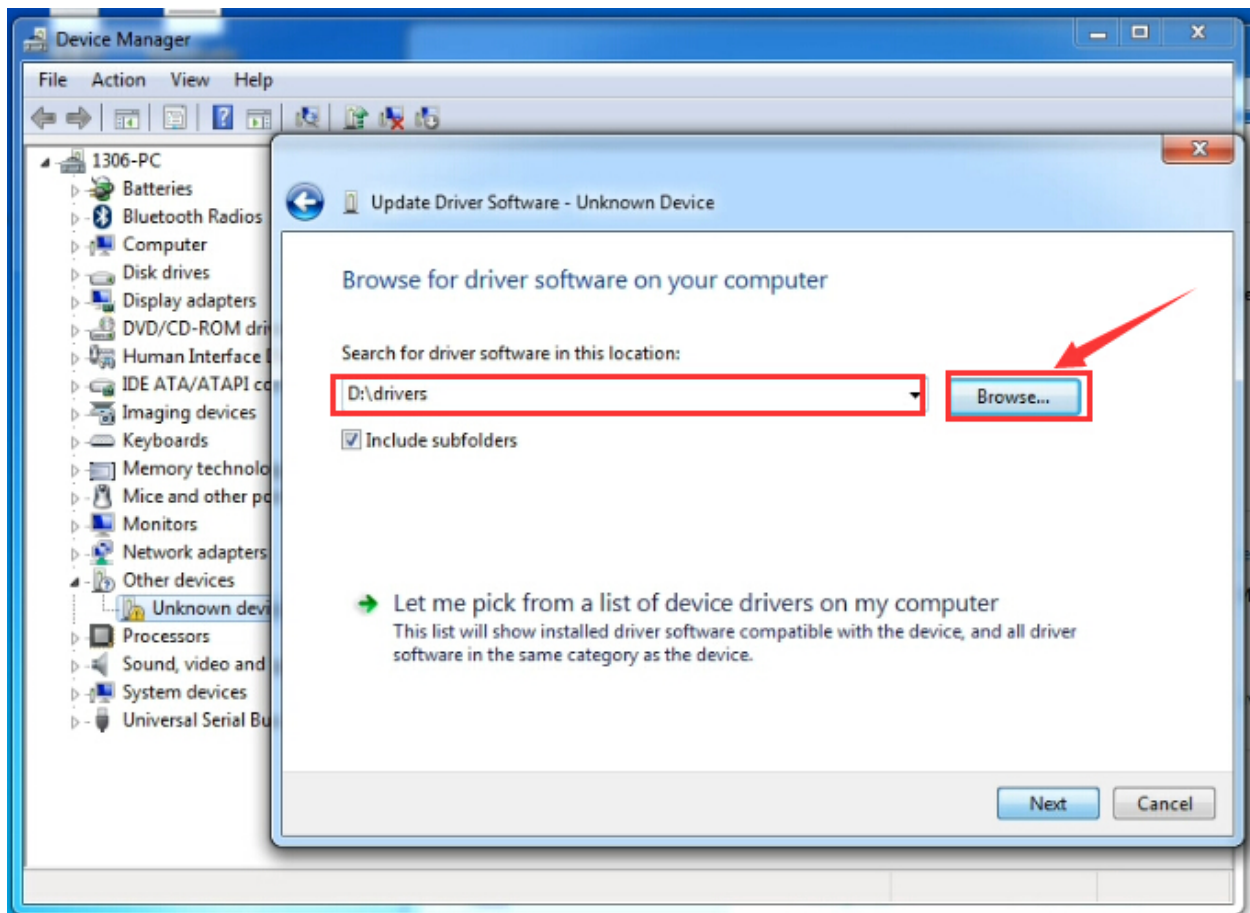




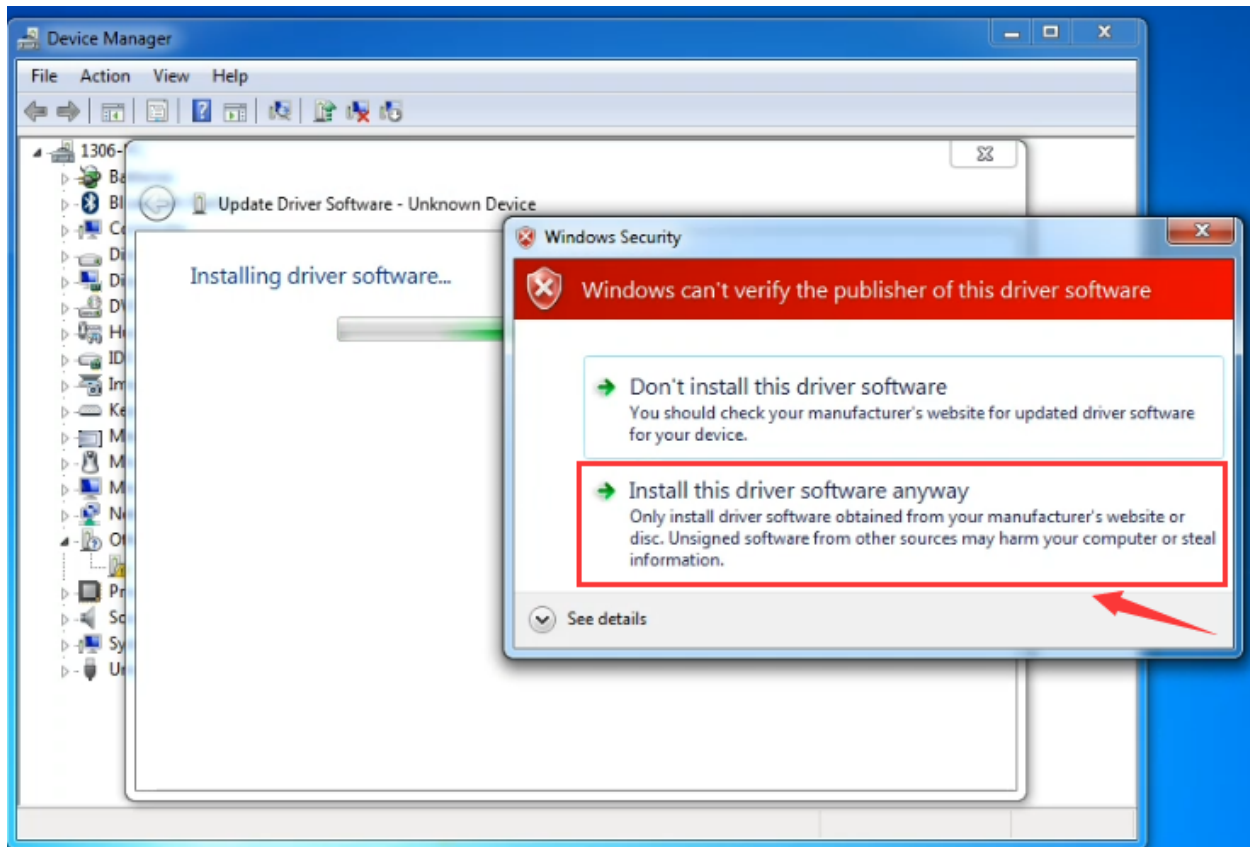
It will then be prompted to either “Search Automatically for updated driver software” or “Browse my computer for driver software”. Shown as below. In this page, select “Browse my computer for driver software”.



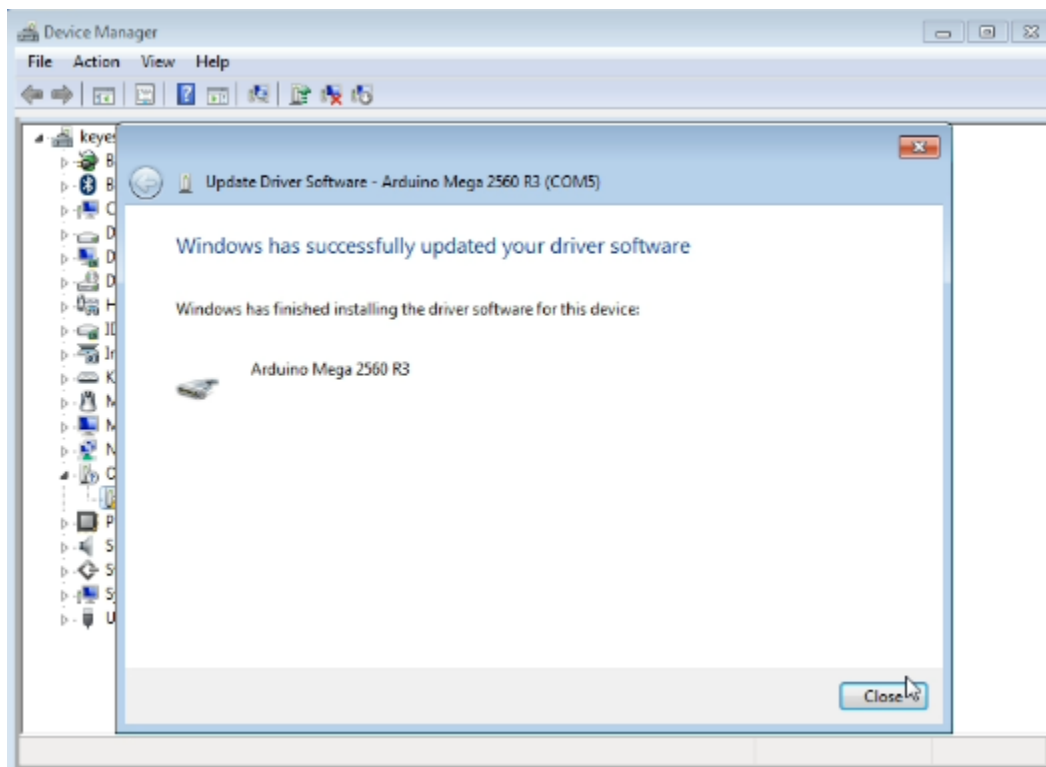
After that, select the option to browse and navigate to the “drivers” folder of Arduino installation.



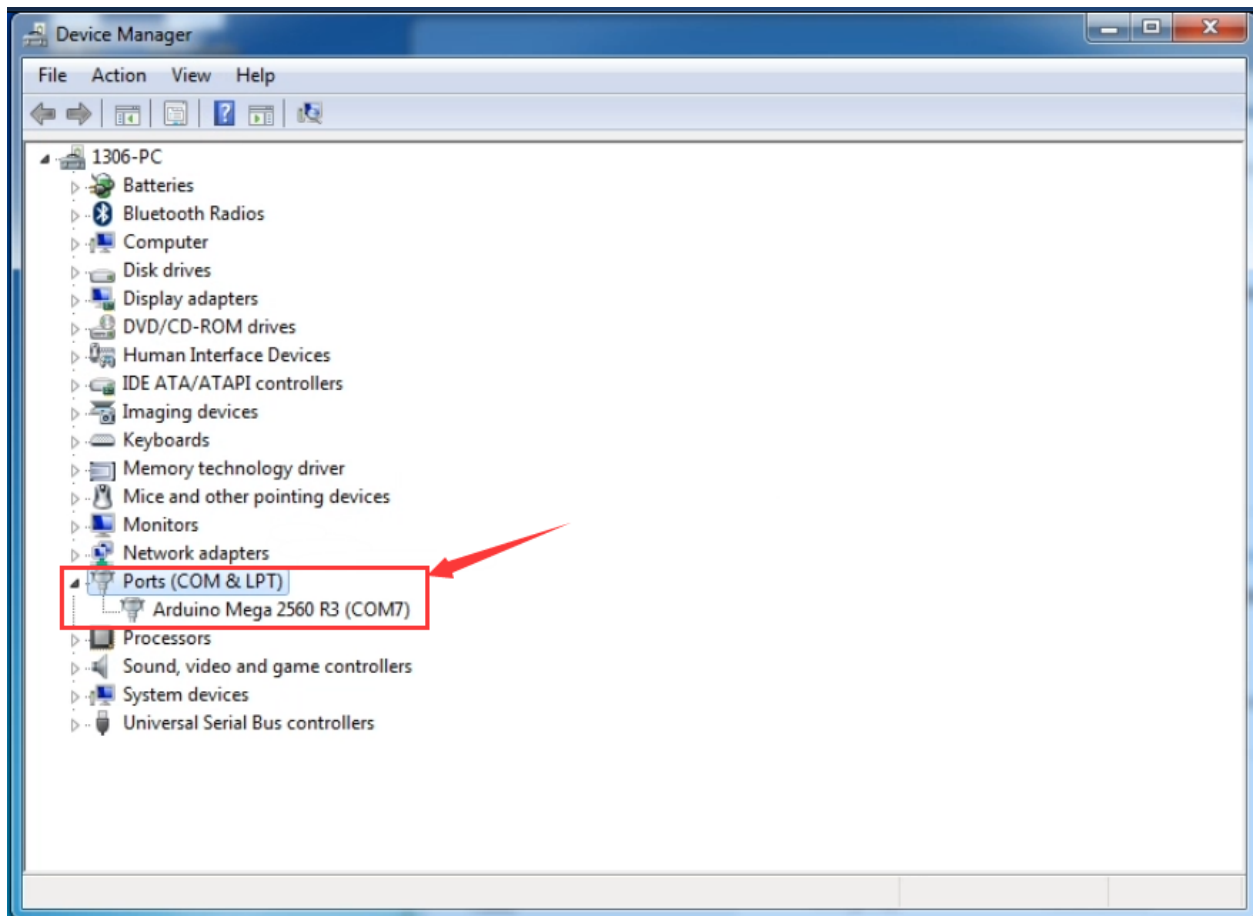
Click “Next” and you may get a security warning, if so, allow the software to be installed. Shown as below.



Installation completed, click “Close”.



Up to now, the driver is installed well. Then you can right click “Computer” —> “Properties”—> “Device manager”, you should see the device shown below.

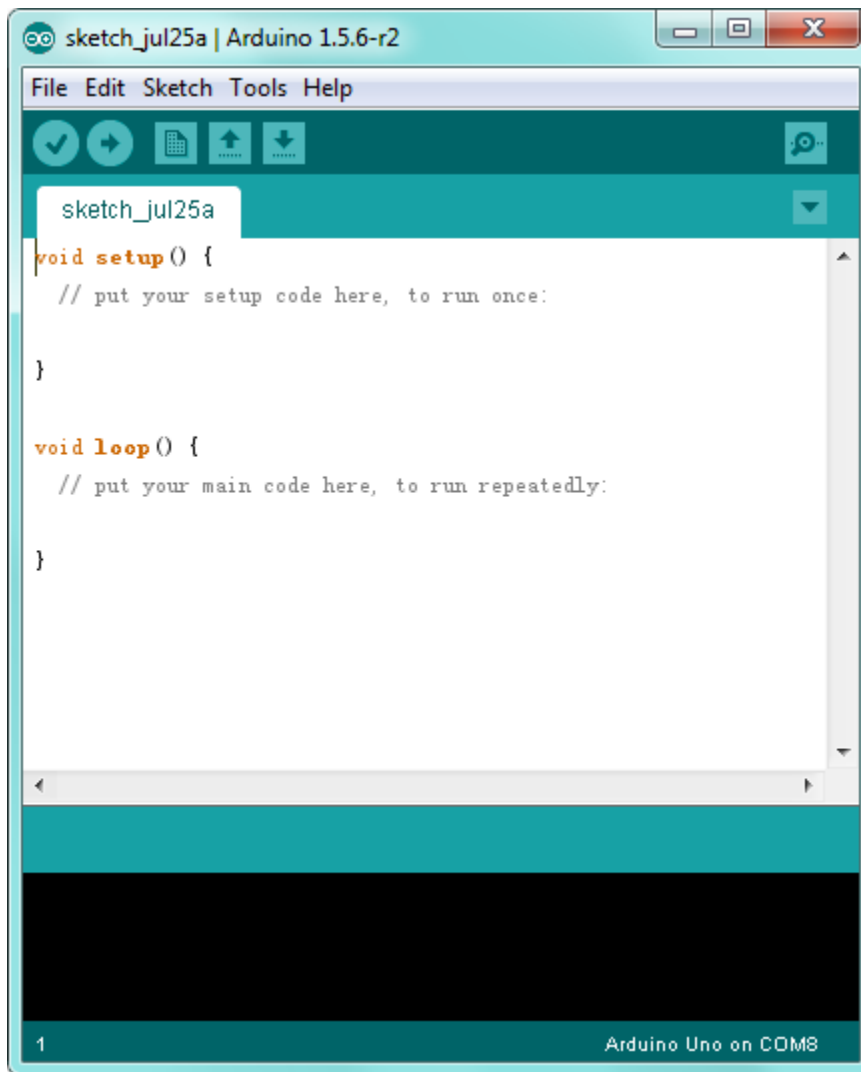


### Introduction for Arduino IDE Toolbar:

Double-click the icon of Arduino software downloaded well, you will get the interface shown below.









(Note: if the Arduino software loads in the wrong language, you can change it in the preferences dialog. See the [environment](#) page for details.)



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



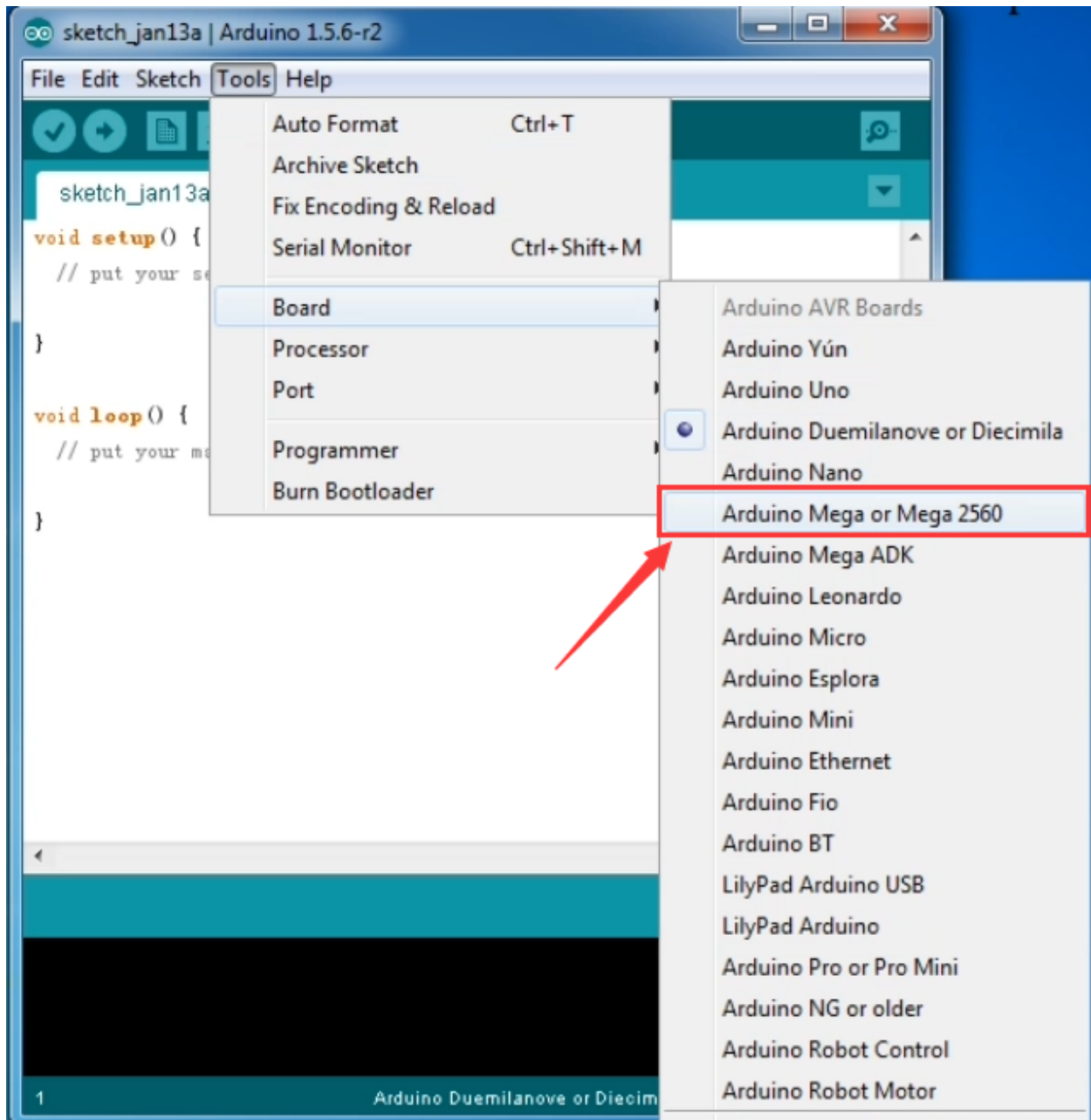
	<b>Verify/Compile</b>	Check the code for errors
	<b>Upload</b>	Upload the current Sketch to the Arduino
	<b>New</b>	Create a new blank Sketch
	<b>Open</b>	Show a list of Sketches
	<b>Save</b>	Save the current Sketch
	<b>Serial Monitor</b>	Display the serial data being sent from the Arduino

**Step2| Connect the board**

Connect the Mega 2560 board to your computer using the USB cable. The green power LED should go on.

**Step3| Select the Arduino Board**

Open the Arduino IDE, you'll need to click the "Tools", then select the Board that corresponds to your Arduino.

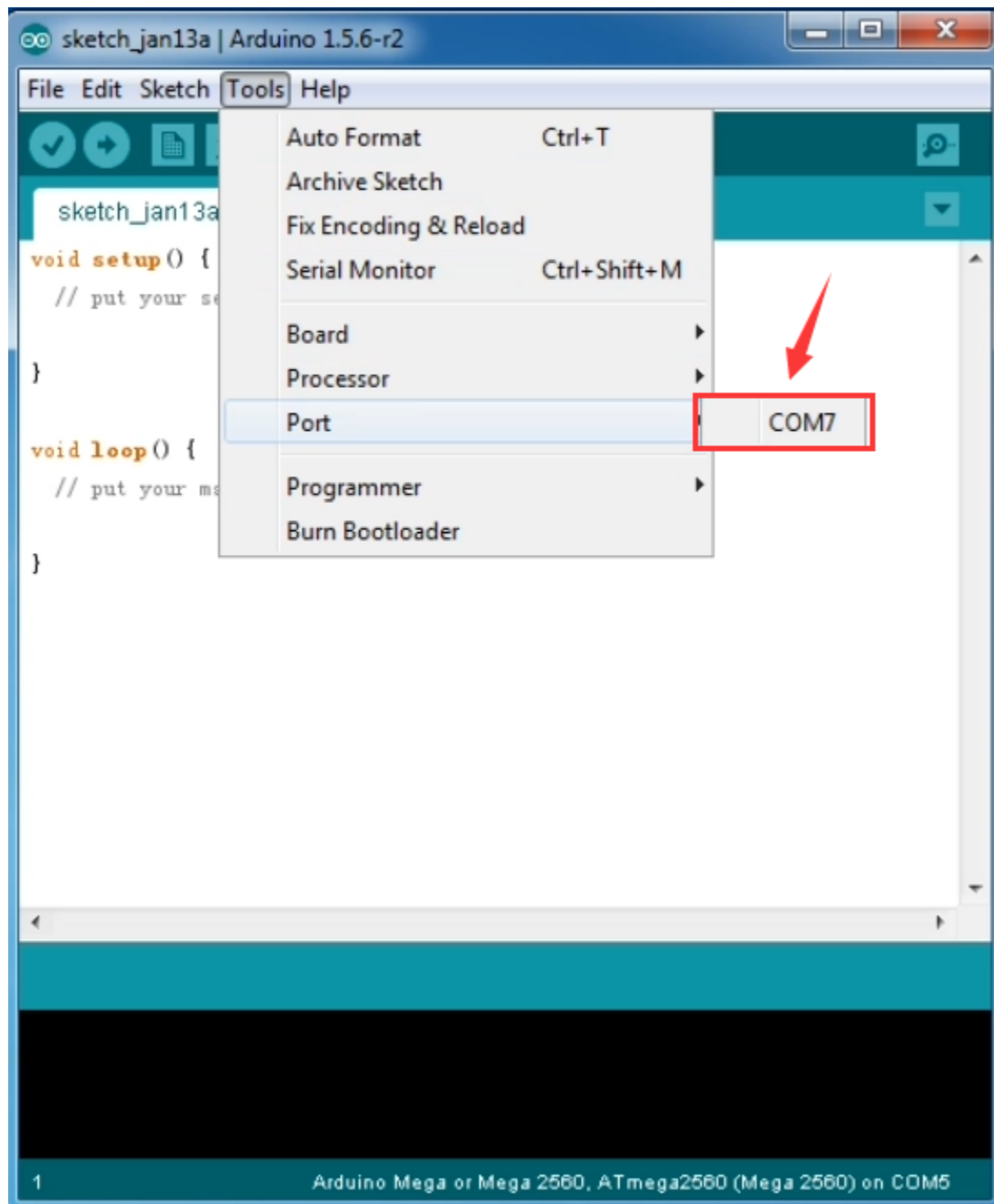


### Step4| Select your serial port

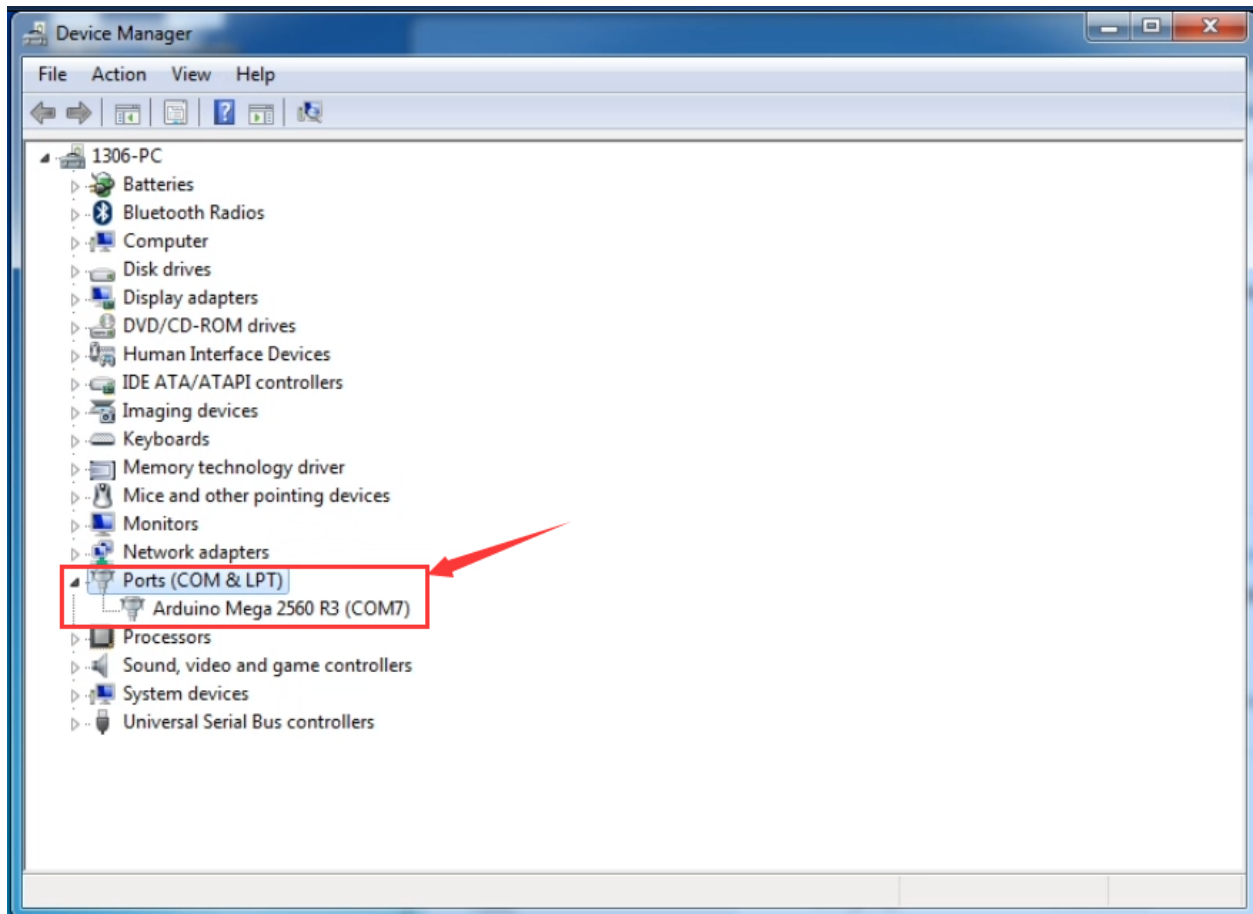
Select the serial device of the Arduino board from the **Tools | Serial Port** menu. This is likely to be COM3 or higher (COM1 and COM2 are usually reserved for hardware serial ports). To find out, you can disconnect your Arduino board and re-open the menu; the entry that disappears should be the Arduino board. Reconnect the board and select that serial port.

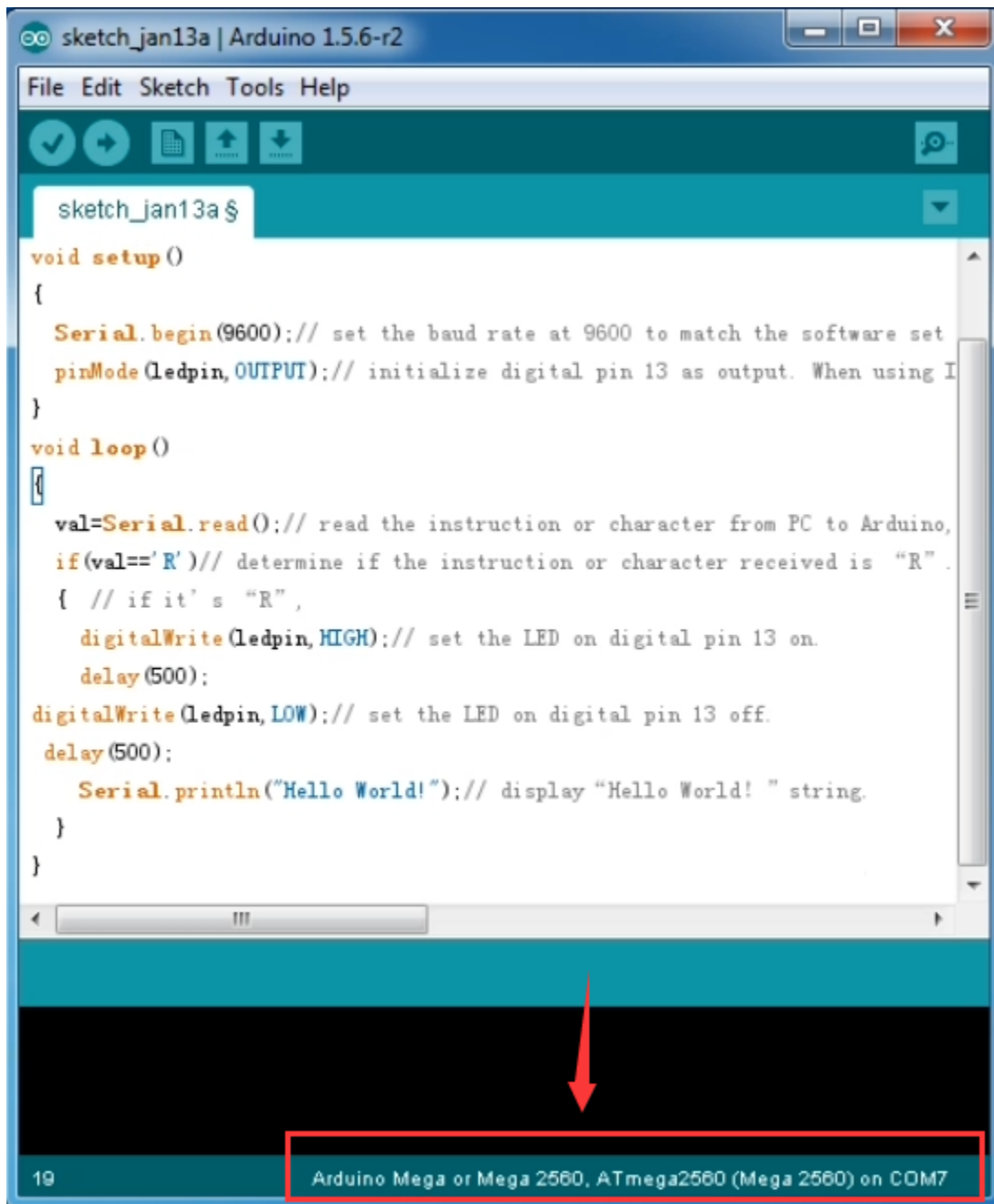
Here you should select COM 7 as below.





**Note:** to avoid errors, the COM Port should keep the same as the Ports shown on Device Manager.





### Step5|Upload the Program

Below is an example program for displaying the Hello World!

Copy and paste the code to the Arduino environment IDE.

////////////////////////////////////

```
int val;//define variable val
```

```
int ledpin=13;// define digital pin13
```

```
void setup()
```

```
{
```

Serial.begin(9600);// set the baud rate at 9600. When connected to a specific device, (e.g. bluetooth), the baud rate needs to be the same with the device.

pinMode(ledpin,OUTPUT);// initialize digital pin 13 as output. When using I/O ports on an Arduino, this kind of set up is always needed.

```
}
```

```
void loop()
```

```
{
```

```
val=Serial.read();// read the instruction or character from PC to Arduino, and assign them to Val.
```

```
if(val=='R')// determine if the instruction or character received is an "R".
```

```
{ // if it's "R",
```

```
digitalWrite(ledpin,HIGH);// set the LED on digital pin 13 on.
```

```
delay(500);
```

```
digitalWrite(ledpin,LOW);// set the LED on digital pin 13 off.
```

```
delay(500);
```

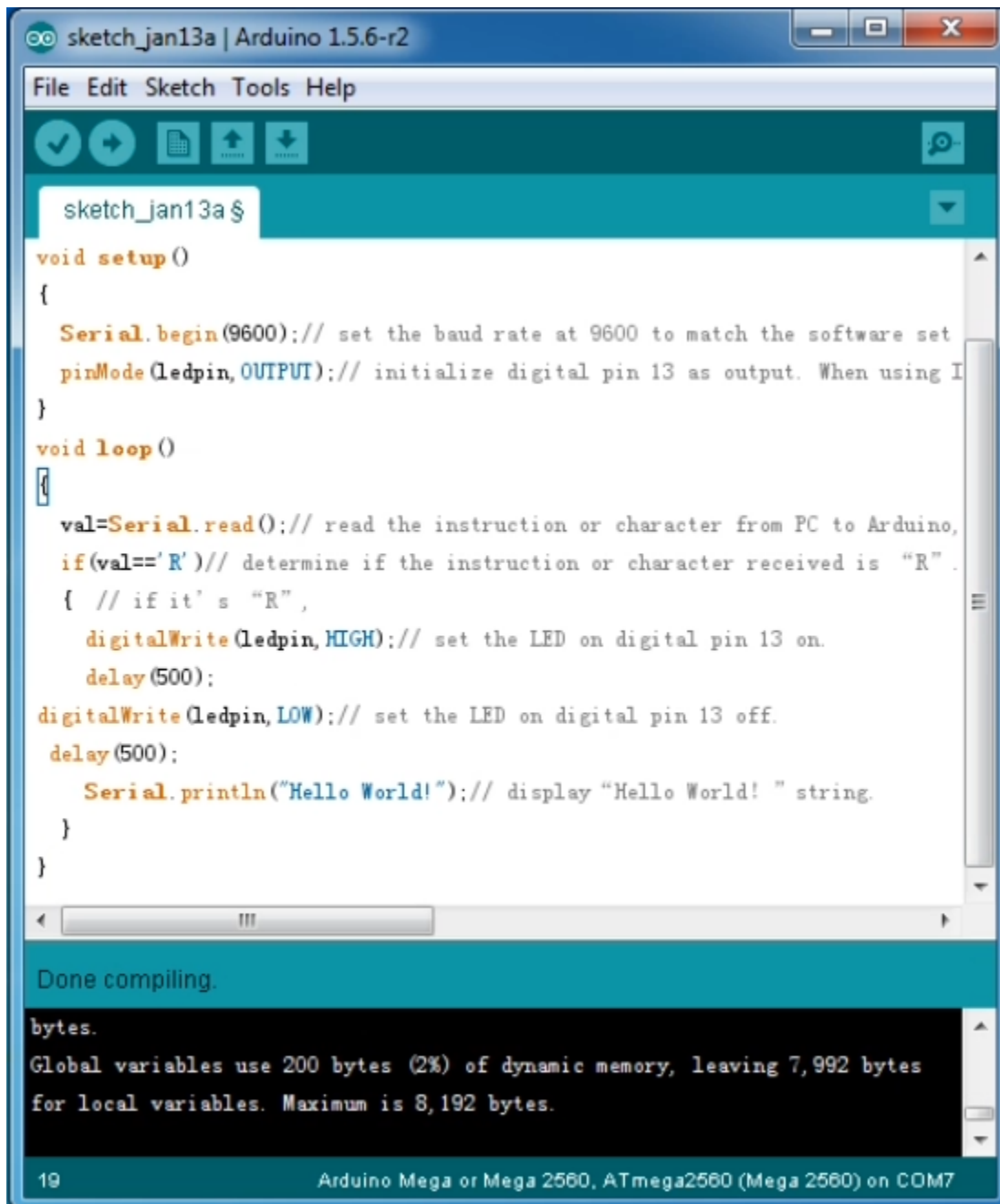
```
Serial.println("Hello World!");// display"Hello World".
```

```
}
```

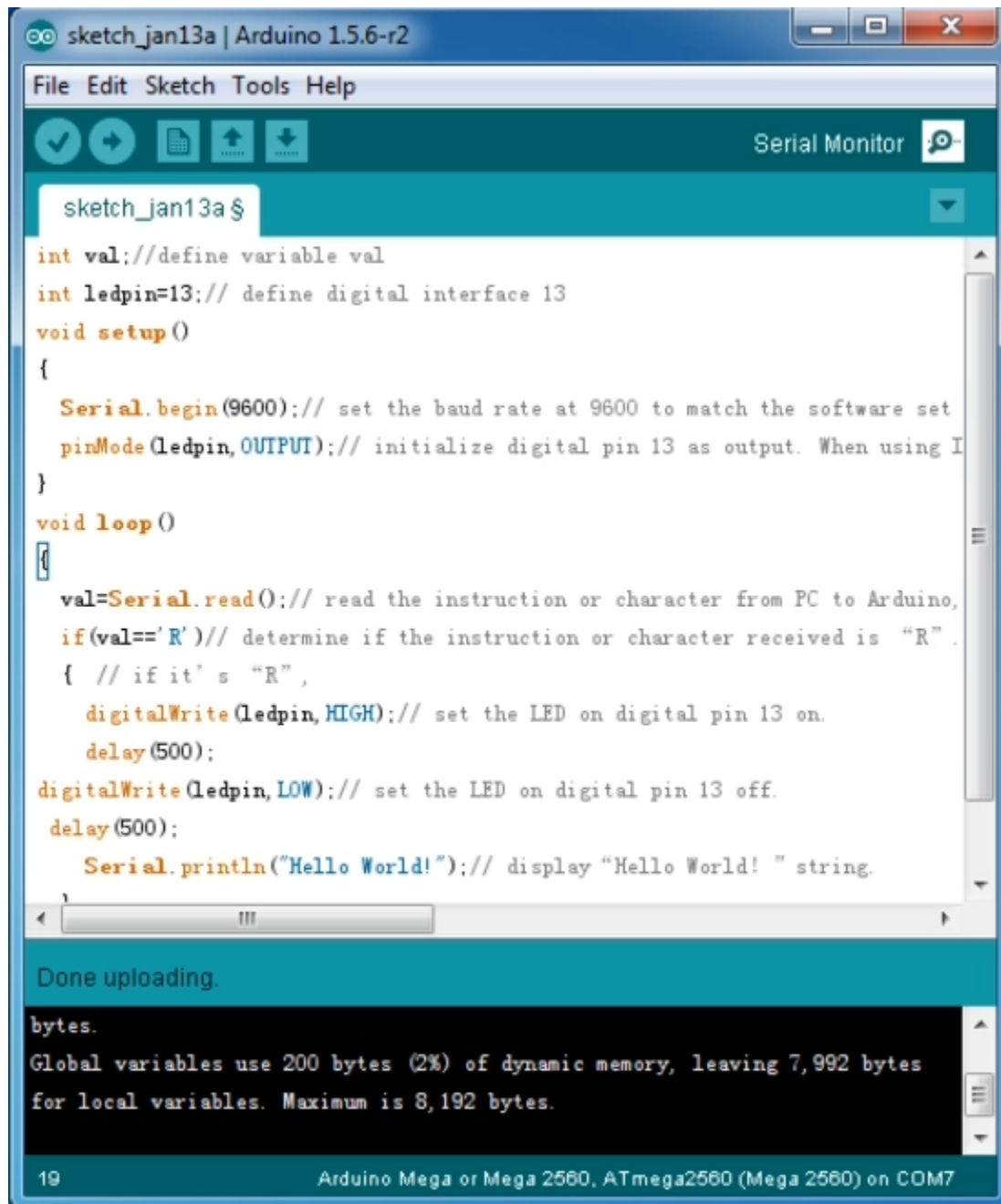
```
}
```

```
////////////////////////////////////
```

Then click verify button to check the errors. If compiling successfully, the message "Done compiling." will appear in the status bar.



After that, click the “Upload” button to upload the code. Wait a few seconds - you should see the RX and TX leds on the board flashing. If the upload is successful, the message “Done uploading.” will appear in the status bar. (Note: If you have an Arduino Mini, NG, or other board, you’ll need to physically present the reset button on the board immediately before pressing the upload button.)

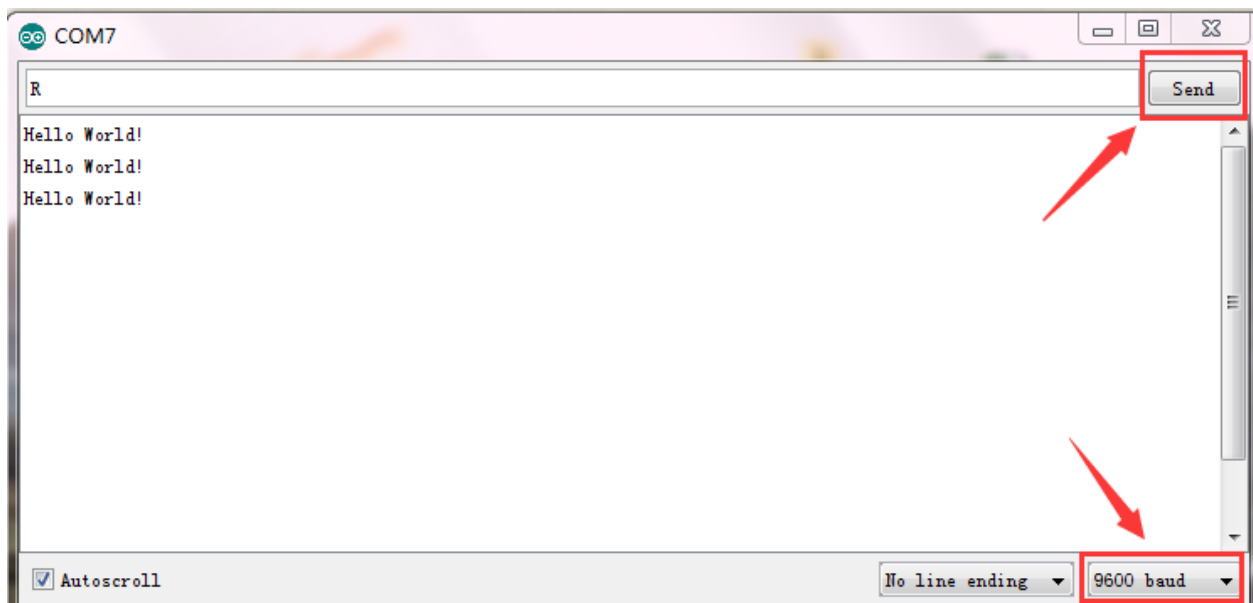


### Step6|Open the Serial Monitor

After that, click the serial monitor button to open the serial monitor.



Then set the baud rate as 9600, enter an "R" and click Send, you should see the TX led on the board blink once, and then D13 led blink once, finally "Hello World!" is showed on the monitor, the RX led blink once.



#### Other Links:

You might also want to look at:

[the examples](#) for using various sensors and actuators;

the [reference](#) for the Arduino language;

The text of the Arduino getting started guide is licensed under a [Creative Commons Attribution-ShareAlike 3.0 License](#). Code samples in the guide are released into the public domain.

You can download the datasheet from the link:

[https://drive.google.com/open?id=1eKufH3g5lYrfX7\\_Ew6CS9b-jQ1\\_RGtxV](https://drive.google.com/open?id=1eKufH3g5lYrfX7_Ew6CS9b-jQ1_RGtxV)

Software Download:

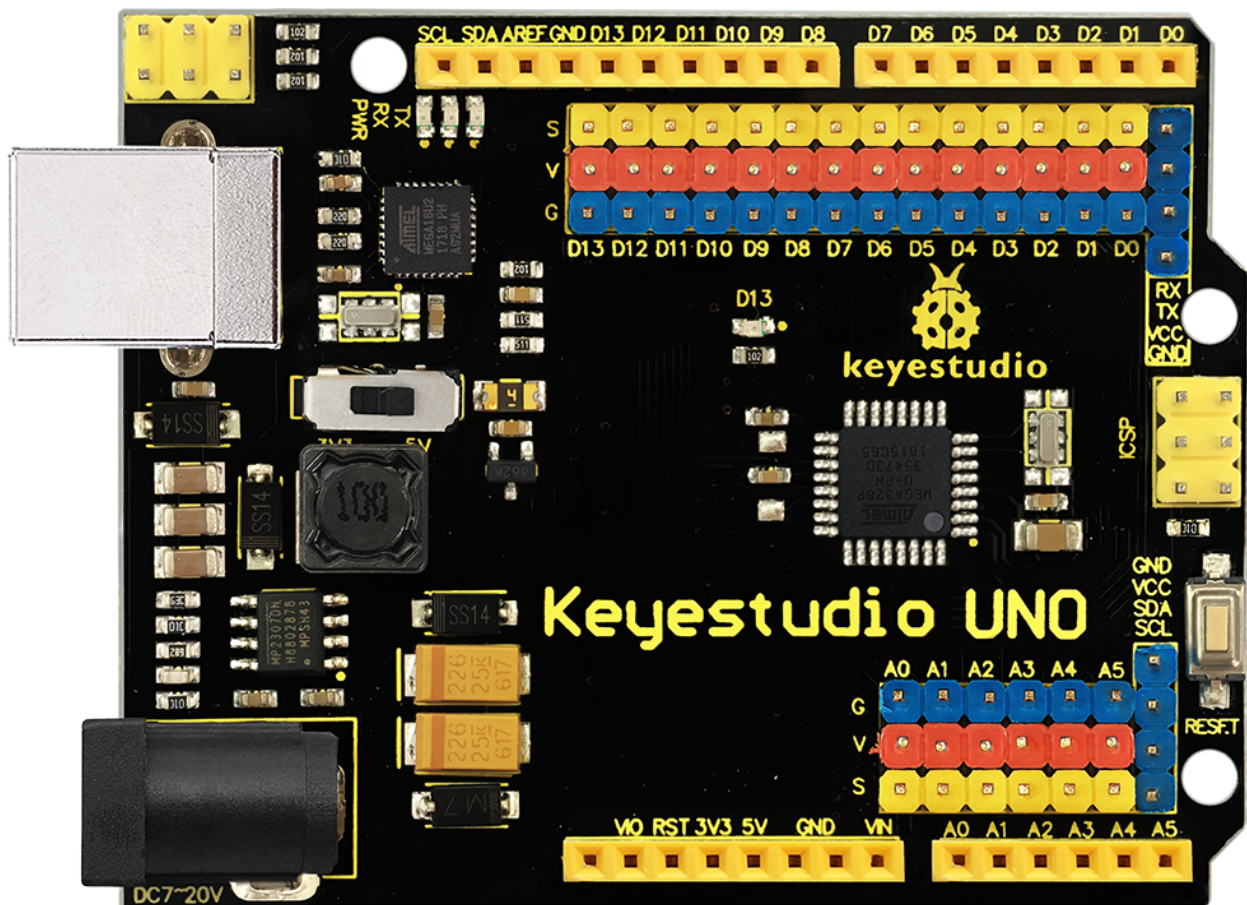
<https://drive.google.com/open?id=12D-JkXdNm03Qt4dlPQr3RP6OmgXqpvHc>

### **Troubleshooting:**

If you have problems, please see the [troubleshooting](#) suggestions.



## KS0172 KEYESTUDIO UNO WITH PIN HEADER DEVELOPMENT BOARD



### Introduction:

keyestudio UNO with pin headers has the same basic functions as [keyestudio UNO R3 BOARD](#). It is a microcontroller board based on the [ATMEGA328P-AU](#), which has the same function as ATMEGA328P(-PU), fully compatible with ARDUINO UNO REV3.

It has 14 digital input/output pins (of which 6 can be used as PWM outputs), 6 analog inputs, a 16 MHz crystal oscillator, a USB connection, a power jack, 2 ICSP headers, and a reset button.

This UNO with pin headers makes the improvement based on keyestudio UNO R3 board.

It breaks out all the digital and analog pins in the form of 3PIN headers (G, V, S).

S pins correspond to all 14 digital pins, 6 analog pins. G pins for ground. V pins for VCC. You can control the voltage of VCC via a slide switch for 5V or 3.3V.

When switched to 5V, level on serial communication port is 5V, voltage of pins is 5V. When switched to 3.3V, level on serial communication port is 3.3V, voltage of pins is 3.3V.

It also breaks out two 4PIN headers for serial communication and IIC communication. So it is more easier to connect external sensors and modules.

As for keyestudio UNO R3 board, its voltage-regulator chip is NSP1117. When connect external power, output 5V, drive current is 1A.

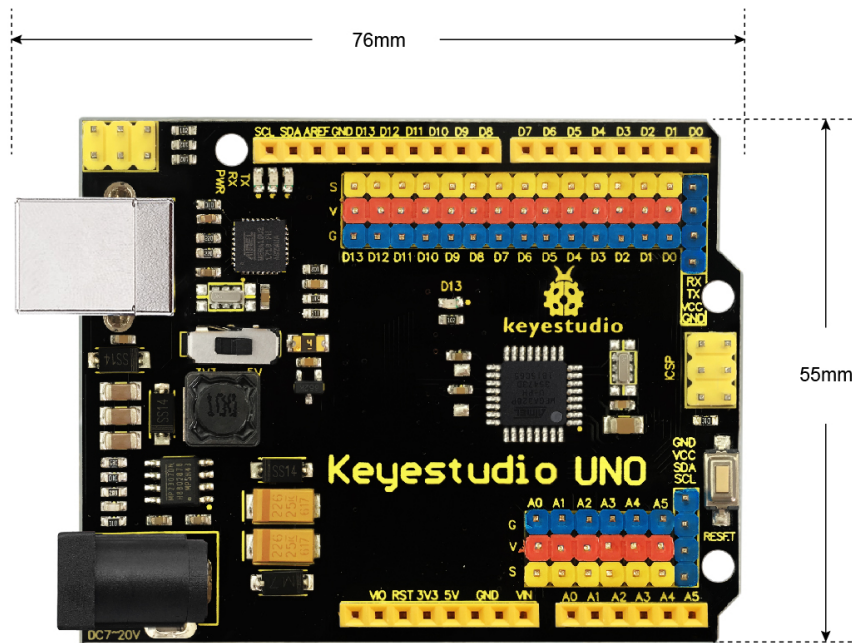
But for this keyestudio UNO with headers, its voltage-regulator chip is MP2307DN. When connect external power, output 5V, drive current is 2A.

It contains everything needed to support the microcontroller; simply connect it to a computer with a USB cable or power it with a AC-to-DC adapter or battery to get started.

### TECH SPECS:

<b>Microcontroller</b>	ATMEGA328P-AU
<b>Operating Voltage</b>	5V
<b>Input Voltage (recommended)</b>	DC 7-12V
<b>Digital I/O Pins</b>	14 (D0-D13)
<b>PWM Digital I/O Pins</b>	6 (D3 D5 D6 D9 D10 D11)
<b>Analog Input Pins</b>	6 (A0-A5)
<b>DC Current per I/O Pin</b>	20 mA
<b>DC Current for 3.3V Pin</b>	50 mA
<b>Flash Memory</b>	32 KB (ATmega328) of which 0.5 KB used by bootloader
<b>SRAM</b>	2 KB
<b>EEPROM</b>	1 KB
<b>Clock Speed</b>	16 MHz
<b>LED_BUILTIN</b>	D13

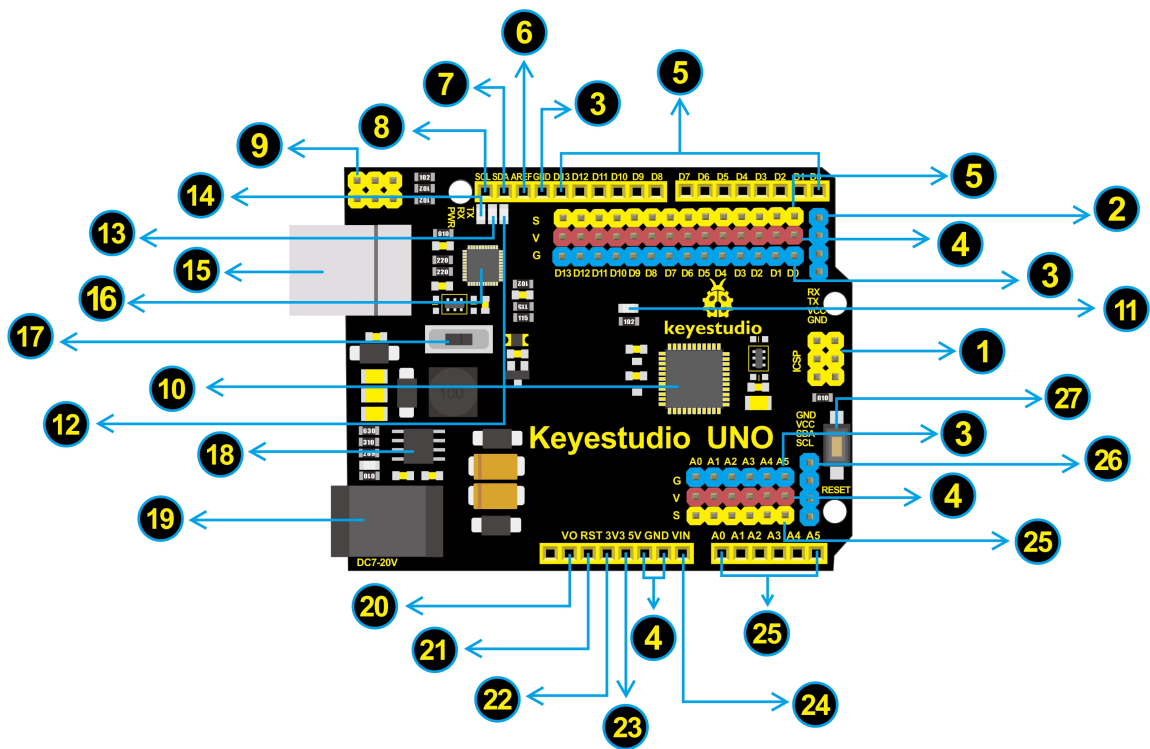
### Dimensions:



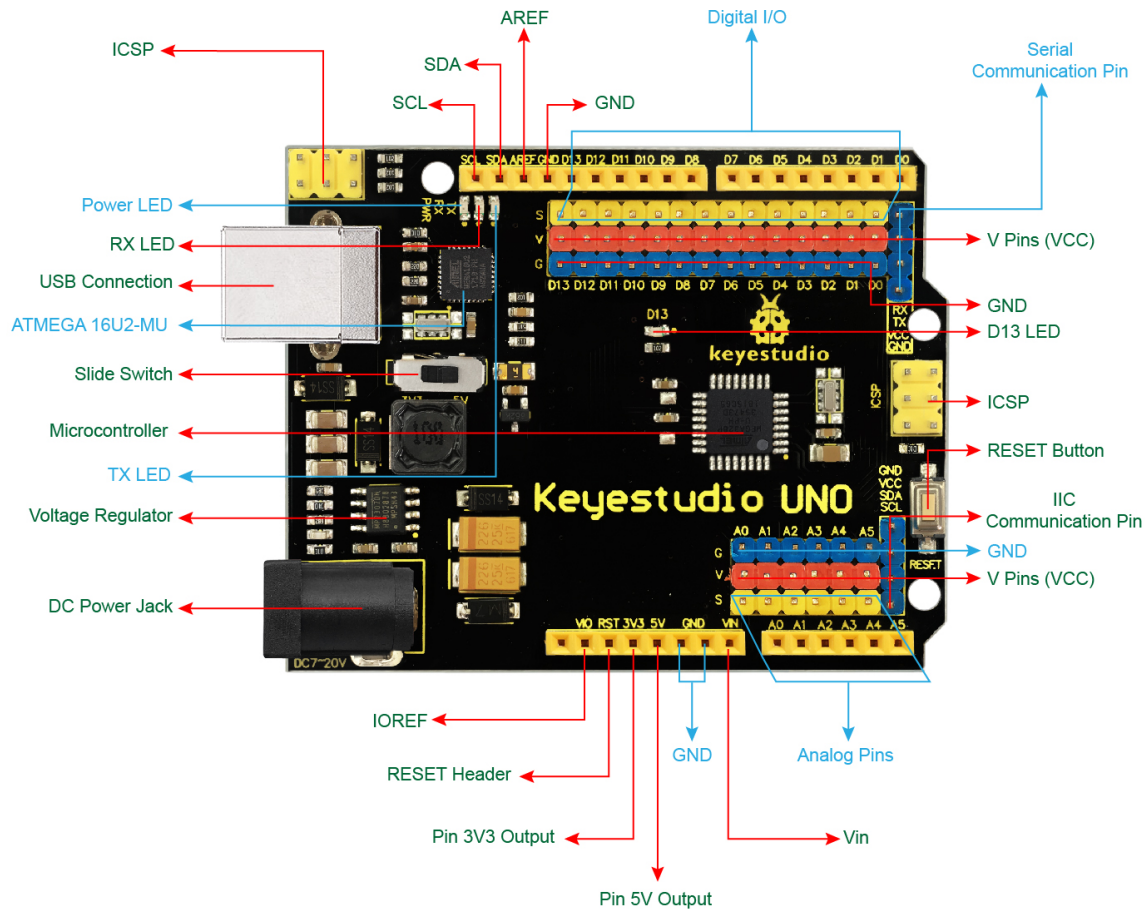
H:15mm

**Element and Interfaces:**

Here is an explanation of what every element and interface of the board does:



1	<b>ICSP (In-Circuit Serial Programming) Header</b> ICSP is the AVR, an Arduino micro-program header consisting of MOSI, MISO, SCK, RESET, VCC, and GND. It is often called the SPI (serial peripheral interface) and can be considered an “extension” of the output. In fact, slave the output devices under the SPI bus host. When connecting to PC, program the firmware to ATMEGA328P-AU.
2	<b>Serial Communication Pin</b> Connect to serial communication. 4Pins (GND, VCC (3.3V or 5V controlled by slide switch), RX, TX)
3	<b>GND</b> Ground pins
4	<b>V Pins (VCC)</b> Power the external sensors and modules. Select the voltage of 3.3V or 5V via a slide switch.
5	<b>Digital I/O</b> It has 14 digital input/output pins, labeled D0 to D13 (of which 6 can be used as PWM outputs). These pins can be configured as digital input pin to read the logic value (0 or 1). Or used as digital output pin to drive different modules like LED, relay, etc. The pin D3, D5, D6, D9, D10, and D11 can be used to generate PWM. For digital port, you can connect through female headers, or through pin headers (labeled S) of 2.54mm pitch.
6	<b>AREF</b> For Analog reference. Sometimes used to set an external reference voltage (0-5V) as the upper limit of analog input pins.
7	<b>SDA</b> IIC communication pin
8	<b>SCL</b> IIC communication pin
9	<b>ICSP (In-Circuit Serial Programming) Header</b> ICSP is an AVR, an Arduino micro-program header consisting of MOSI, MISO, SCK, RESET, VCC, and GND. Connected to ATMEGA 16U2-MU. When connecting to PC, program the firmware to ATMEGA 16U2-MU.
10	<b>Microcontroller</b> Each control board has its own microcontroller. You can regard it as the brain of your board. Microcontrollers are usually from ATMEL. Before you load a new program on the Arduino IDE, you must know what IC is on your board. This information can be checked at the top of IC. The microcontroller used in this board is <a href="#">ATMEGA328P-AU</a> .
11	<b>D13 LED</b> 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.
12	<b>TX LED</b> Onboard you can find the label: TX (transmit) When UNO board communicates via serial port, send the message, TX led flashes.
13	<b>RX LED</b> Onboard you can find the label: RX(receive ) When UNO board communicates via serial port, receive the message, RX led flashes.
14	<b>Power LED</b> LED on means that your circuit board is correctly powered on. Otherwise LED is off.
15	<b>USB Connection</b> You can power the board via USB connection. Or can upload the program to the board via USB port. Connect the board to PC using a USB cable via USB port.
16	<b>ATMEGA 16U2-MU</b> USB to serial chip, can convert the USB signal into serial port signal.
17	<b>Slide Switch</b> You can slide the switch to control the voltage of pin V (VCC), 3.3V or 5V.
18	<b>Voltage Regulator</b> To control the voltage provided to the UNO board, as well as to stabilize the DC voltage used by the processor and other components. Convert an external input DC7-12V voltage into DC 5V, then switch DC 5V to the processor and other components, output DC 5V, drive current is 2A.
19	<b>DC Power Jack</b> The board can be supplied with an external power DC7-12V from the DC power jack.
20	<b>IOREF</b> Used to configure the operating voltage of microcontrollers. Use it less.
21	<b>RESET Header</b> Connect an external button to reset the board. The function is the same as reset button.
22	<b>Pin 3V3 Output</b> Provides 3.3V voltage output
23	<b>Pin 5V Output</b> Provides 5V voltage output
24	<b>Vin</b> You can supply an external voltage input DC7-12V through this pin to the board.



Specialized Functions of Some Pins:

- **Serial communication:** Digital pins 0 (RX) and 1 (TX).
- **PWM Interfaces (Pulse-Width Modulation):** D3, D5, D6, D9, D10, D11
- **External Interrupts:** D2 (interrupt 0) and D3 (interrupt 1). These pins can be configured to trigger an interrupt on a low value, a rising or falling edge, or a change in value.
- **SPI communication:** D10 (SS), D11 (MOSI), D12 (MISO), D13 (SCK). These pins support SPI communication using the SPI library.
- **IIC communication:** A4 (SDA); A5(SCL)

\*\* Detailed Test with Arduino Software:\*\*

Step1 | Download the Arduino IDE

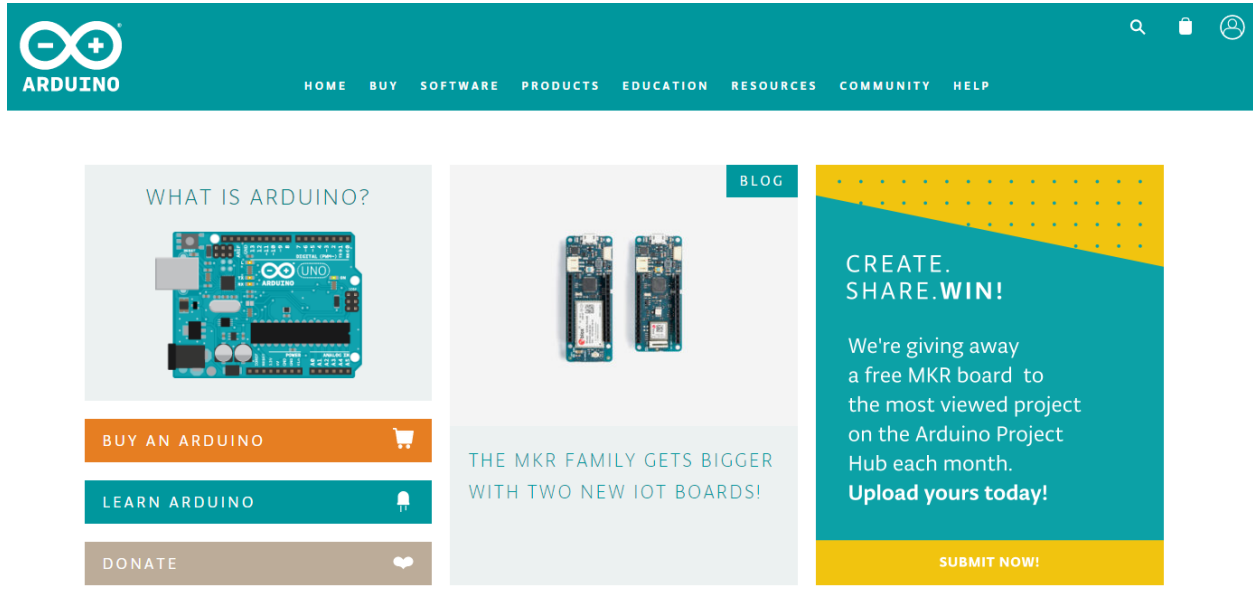


When you get the UNO development board, first you should install the Arduino software and driver.

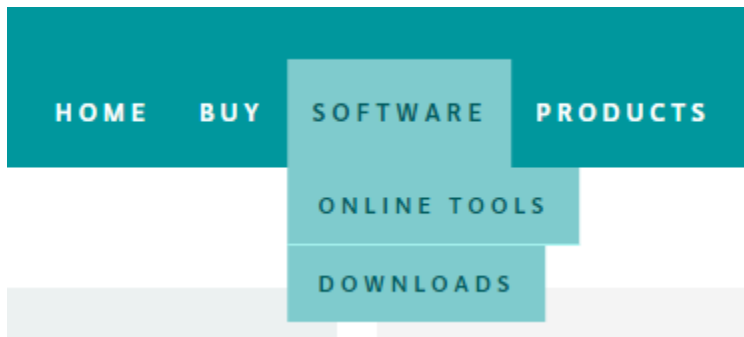
We usually use the Windows software Arduino 1.5.6 version. You can download it from the link below:

<https://www.arduino.cc/en/Main/OldSoftwareReleases#1.5.x>

Or you can browse the ARDUINO website to download the latest version from this link, <https://www.arduino.cc>, pop up the following interface.




Then click the **SOFTWARE** on the browse bar, you will have two options **ONLINE TOOLS** and **DOWNLOADS**.



Click **DOWNLOADS**, it will appear the latest software version of ARDUINO 1.8.5 shown as below.

## Download the Arduino IDE



**ARDUINO 1.8.5**

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** 10.7 Lion or newer

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

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

In this software page, on the right side you can see the version of development software for different operating systems. ARDUINO has a powerful compatibility. You should download the software that is compatible with the operating system of your computer.

We will take **WINDOWS system** as an example here. There are also two options under Windows system, one is installed version, the other is non-installed version.

For simple installed version, first click **Windows Installer**, you will get the following page.

**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** 10.7 Lion or newer


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

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



## Contribute to the Arduino Software

Consider supporting the Arduino Software by contributing to its development. (US tax payers, please note this contribution is not tax deductible). [Learn more on how your contribution will be used.](#)



Since March 2015, the Arduino IDE has been downloaded **24,353,248** times. (IMPRESSIVE!) NO LONGER JUST FOR ARDUINO AND GENUINO BOARDS, HUNDREDS OF COMPANIES AROUND THE WORLD ARE USING THE IDE TO PROGRAM THEIR DEVICES, INCLUDING COMPATIBLES, CLONES, AND EVEN COUNTERFEITS. HELP ACCELERATE ITS DEVELOPMENT WITH A SMALL CONTRIBUTION! REMEMBER: OPEN SOURCE IS LOVE!

Donation options: \$3, \$5, \$10, \$25, \$50, OTHER

Buttons: JUST DOWNLOAD, CONTRIBUTE & DOWNLOAD

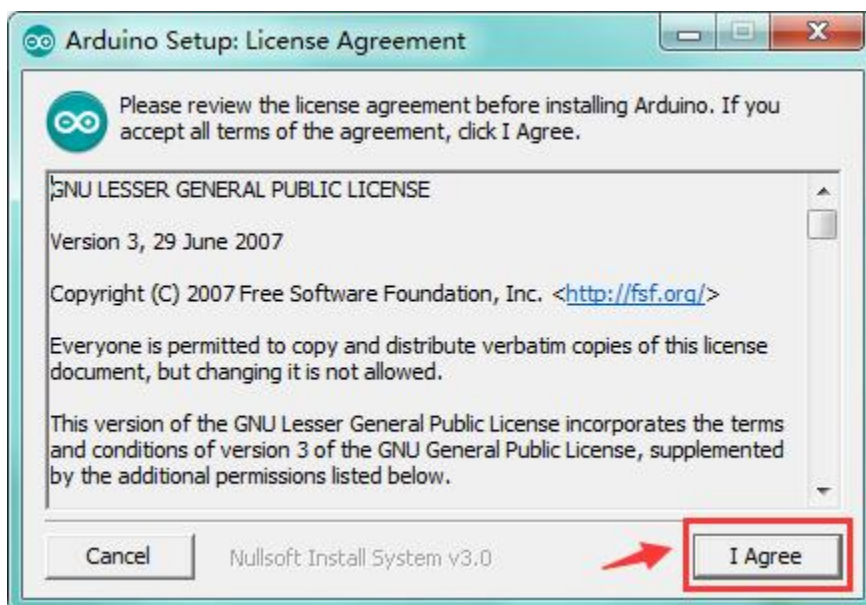
This way you just need to click JUST DOWNLOAD, then click the downloaded file to install it.

For non-installed version, first click Windows ZIP file, you will also get the pop-up interface as the above figure.

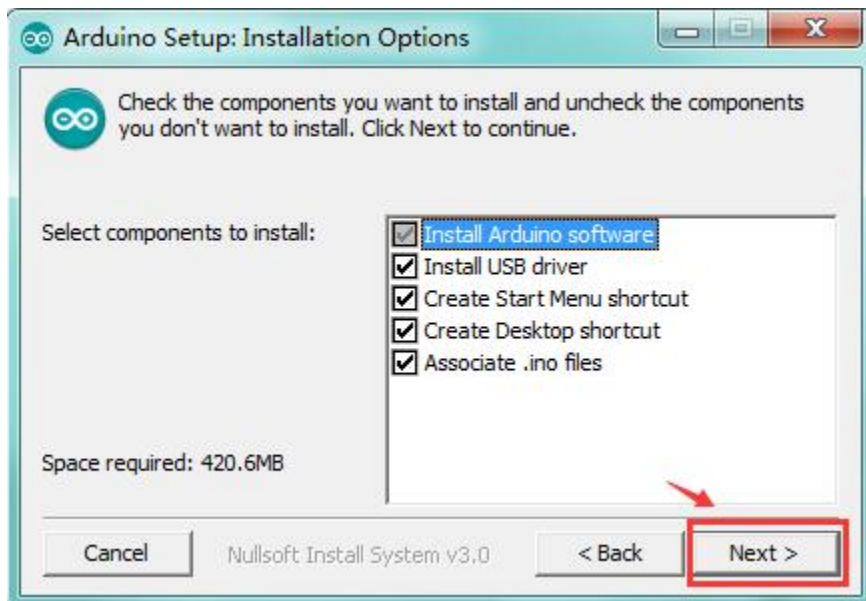
Click JUST DOWNLOAD, and when the ZIP file is downloaded well to your computer, you can directly unzip the file and click the icon of ARDUINO software to start it.

Step2 [Installing Arduino (Windows):

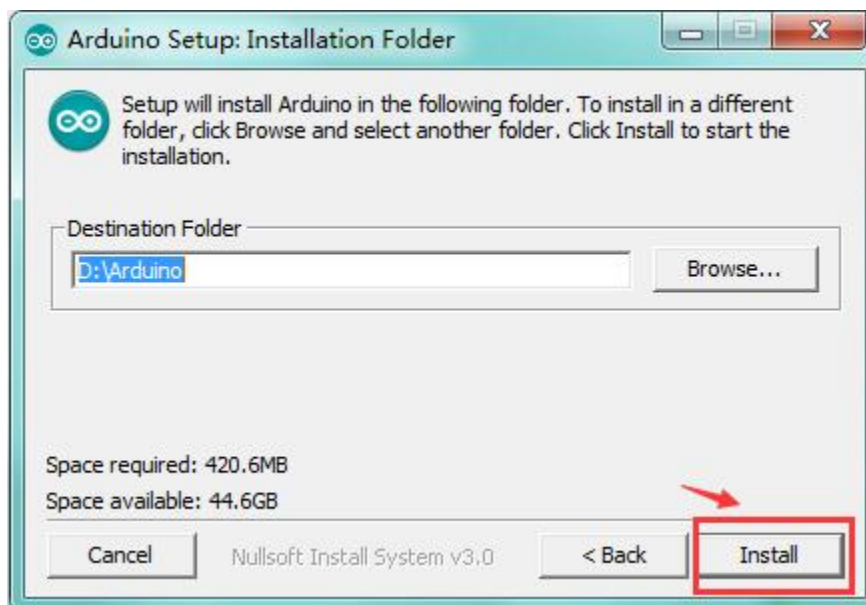
Install Arduino with the exe. Installation package downloaded well.



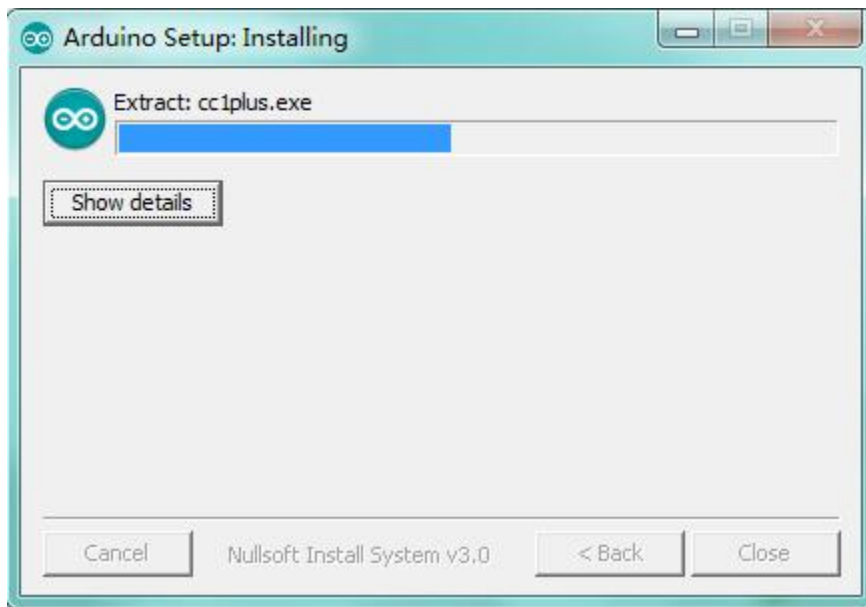
Click "I Agree" to see the following interface.



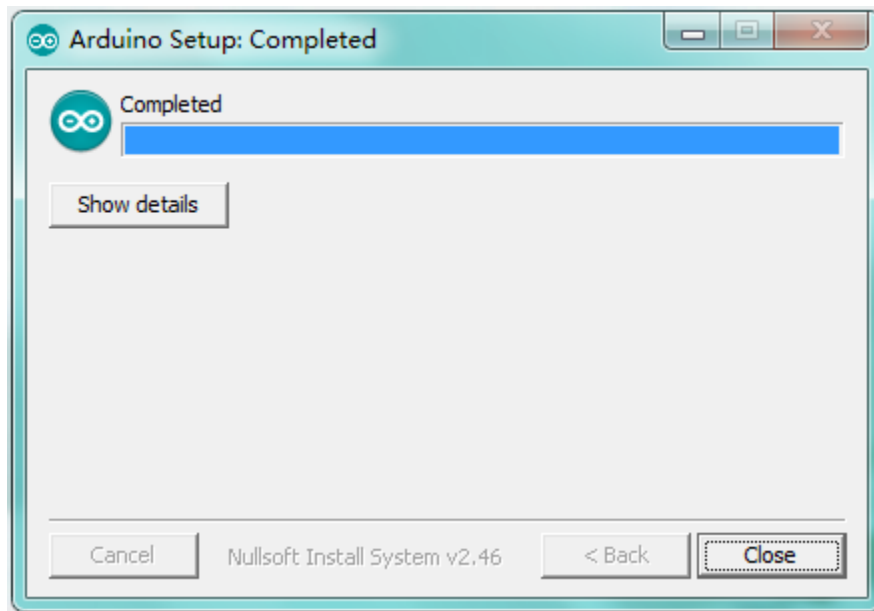
Click “Next”. Pop up the interface below.



You can press Browse... to choose an installation path or directly type in the directory you want. Then click “Install” to initiate installation.



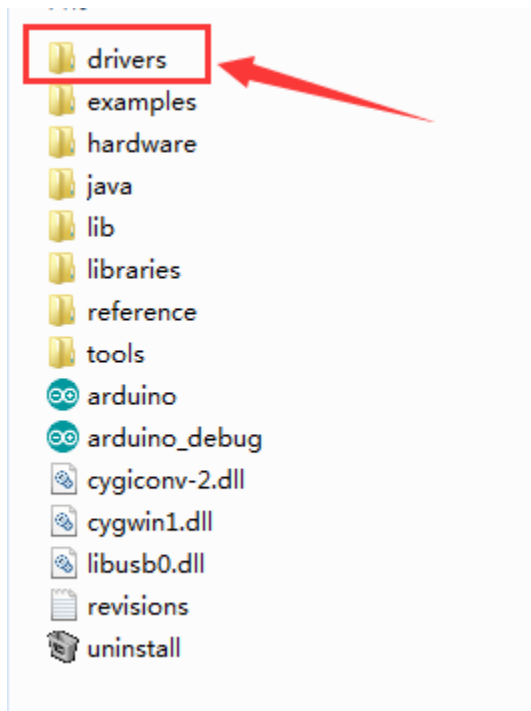
Wait for the installing process, if appear the interface of Window Security, just continue to click Install to finish the installation.



### Step3 |Installing Driver:

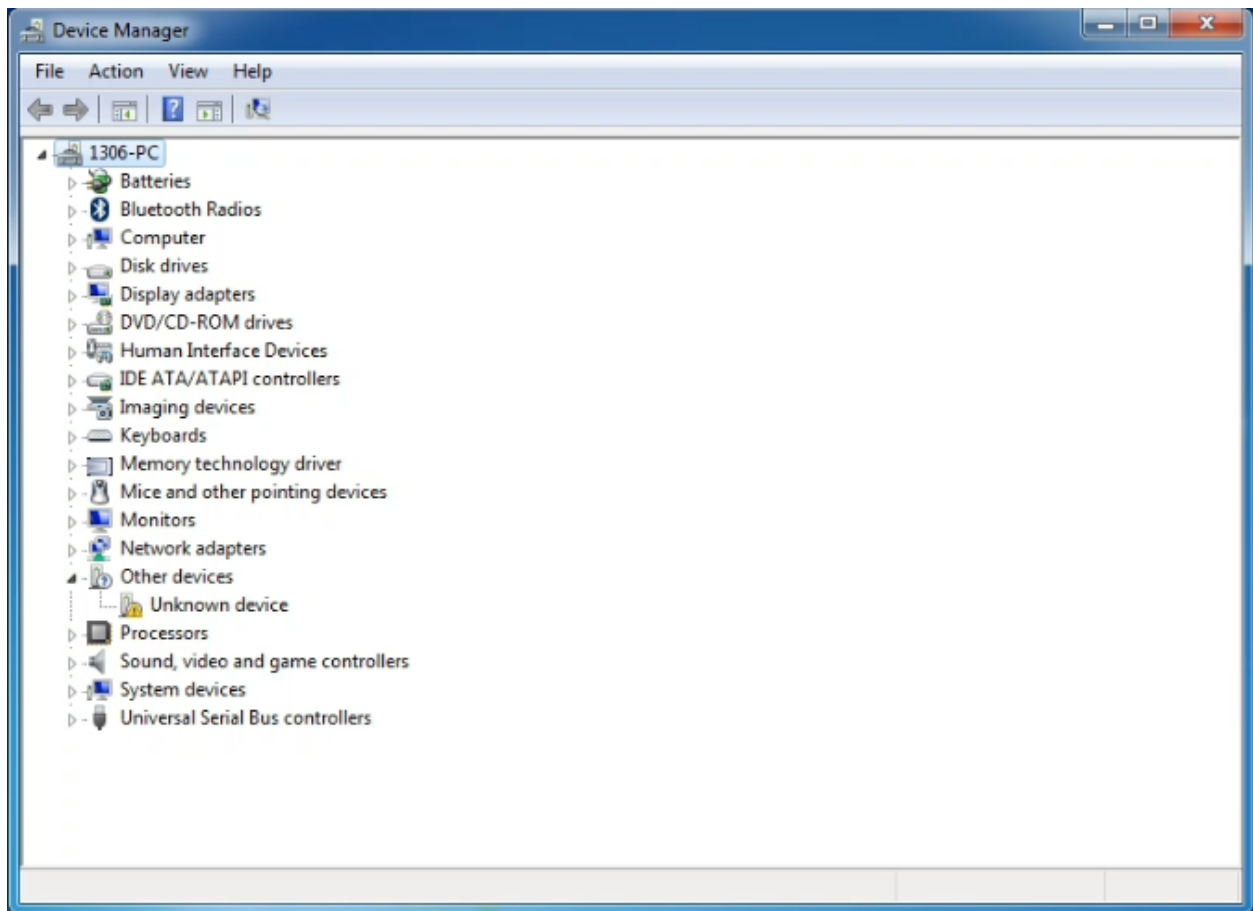
Next, we will introduce the driver installation of UNO development board. The driver installation may have slight differences in different computer systems. So in the following let's move on to the driver installation in the WIN 7 system.

The Arduino folder contains both the Arduino program itself and the drivers that allow the Arduino to be connected to your computer by a USB cable. Before we launch the Arduino software, you are going to install the USB drivers.

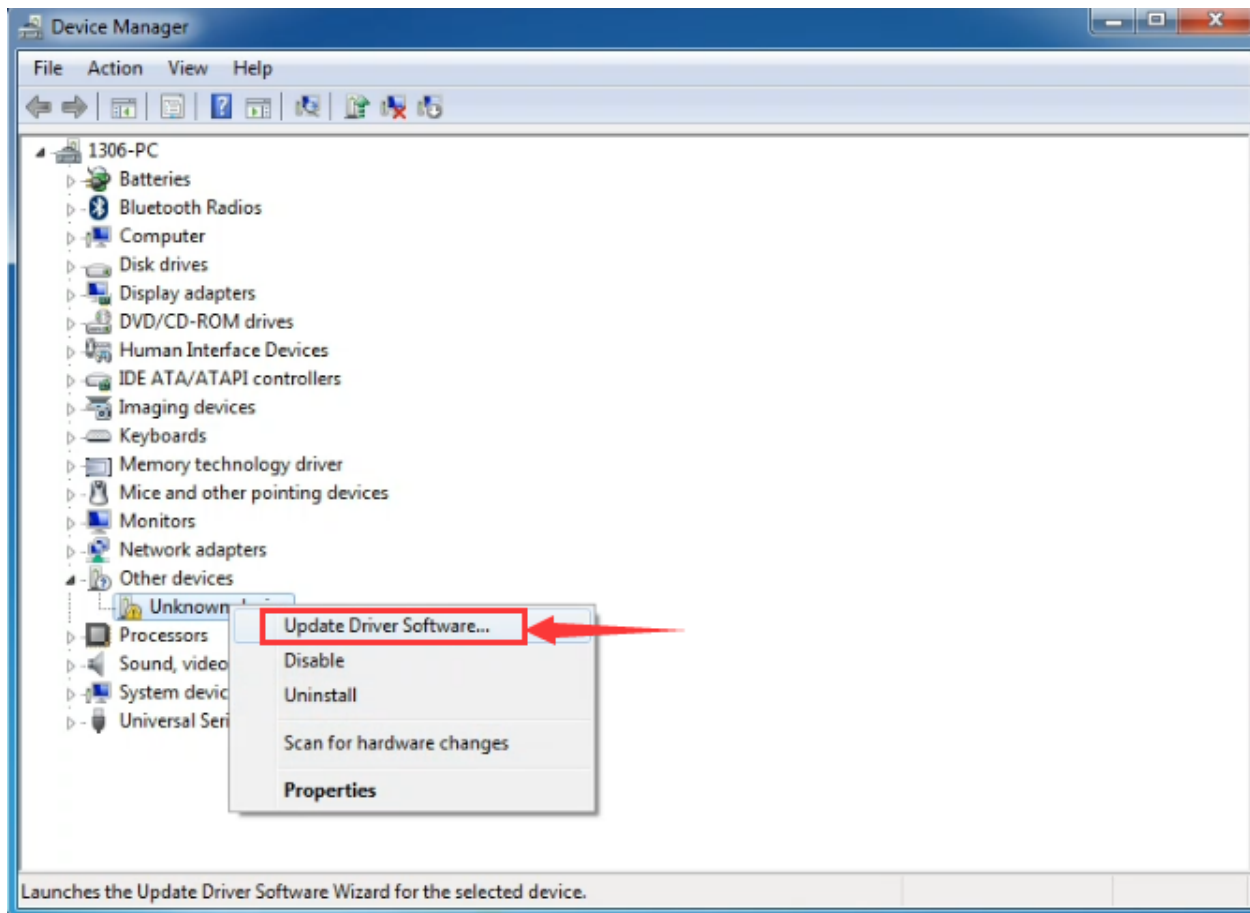


Plug one end of your USB cable into the Arduino and the other into a USB socket on your computer.

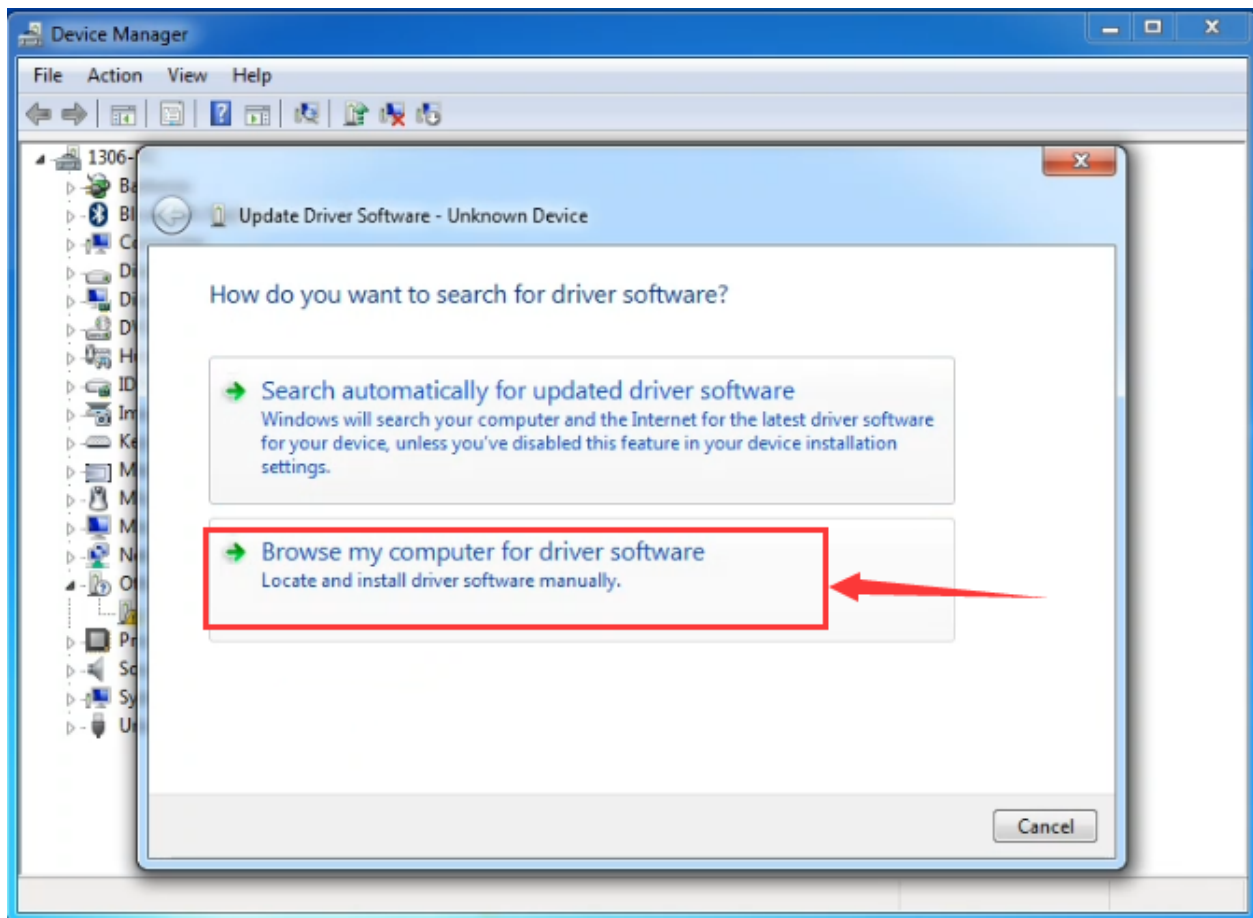
When you connect UNO board to your computer at the first time, right click the icon of your “Computer” —>for “Properties”—> click the “Device manager”, under “Other Devices”, you should see an icon for “Unknown device” with a little yellow warning triangle next to it. This is your Arduino.



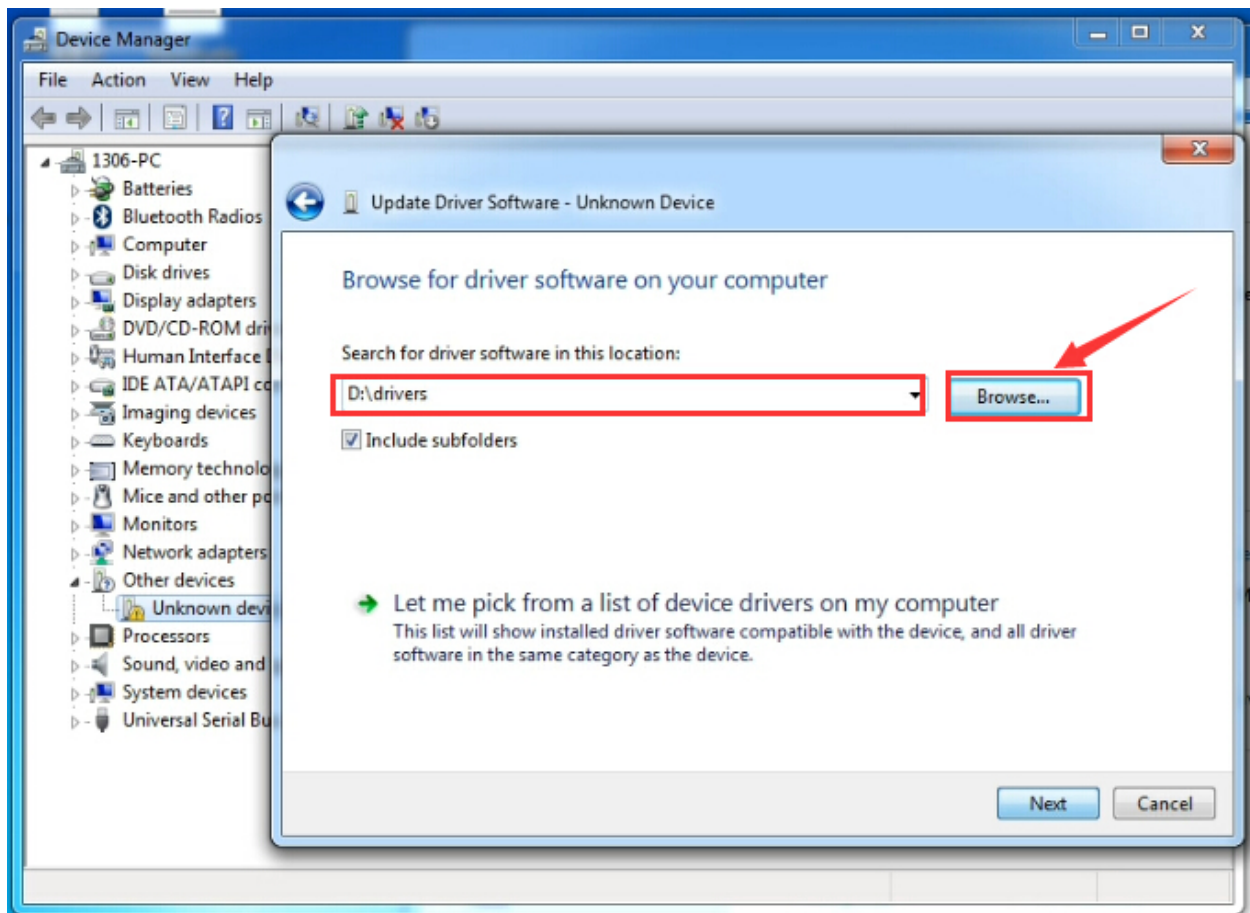
Then right-click on the device and select the top menu option (Update Driver Software...) shown as the figure below.



It will then be prompted to either “Search Automatically for updated driver software” or “Browse my computer for driver software”. Shown as below. In this page, select “Browse my computer for driver software”.

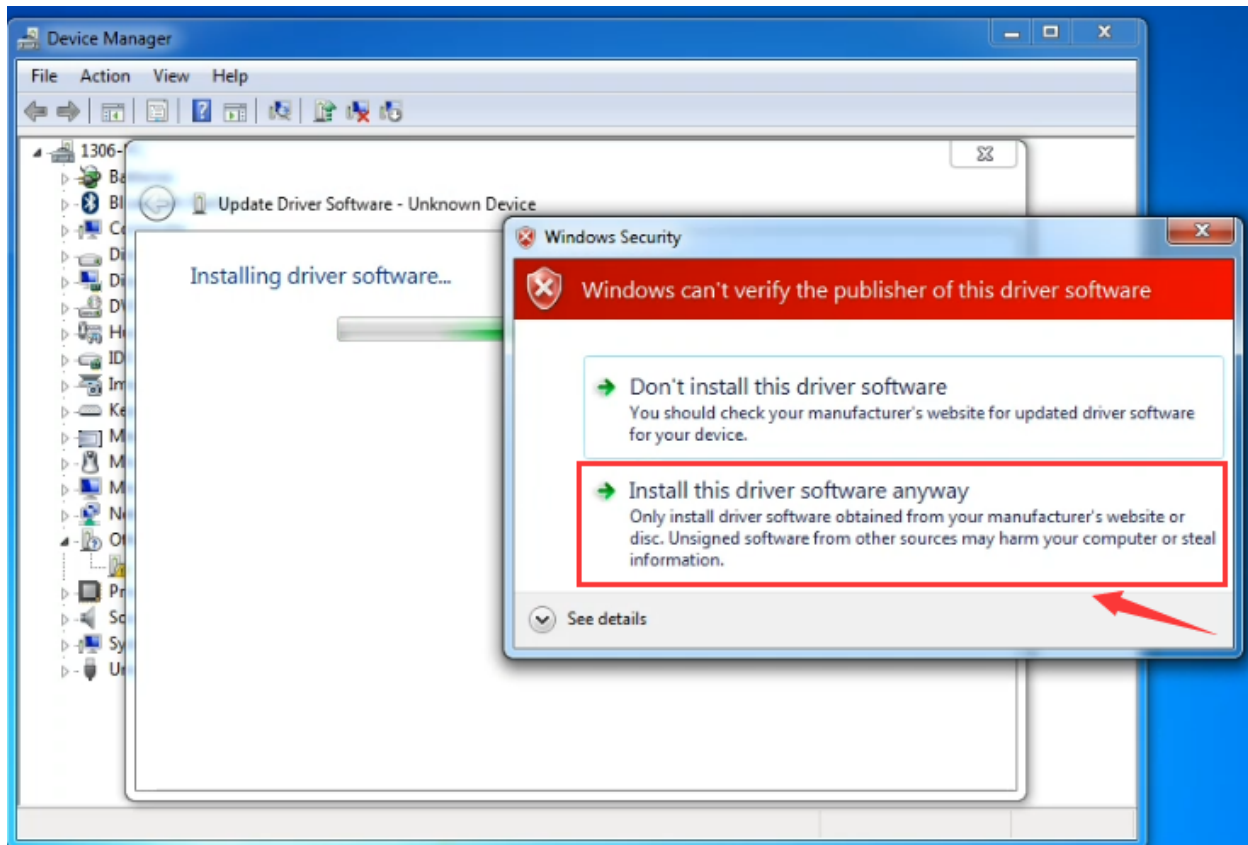


After that, select the option to browse and navigate to the “drivers” folder of Arduino installation.

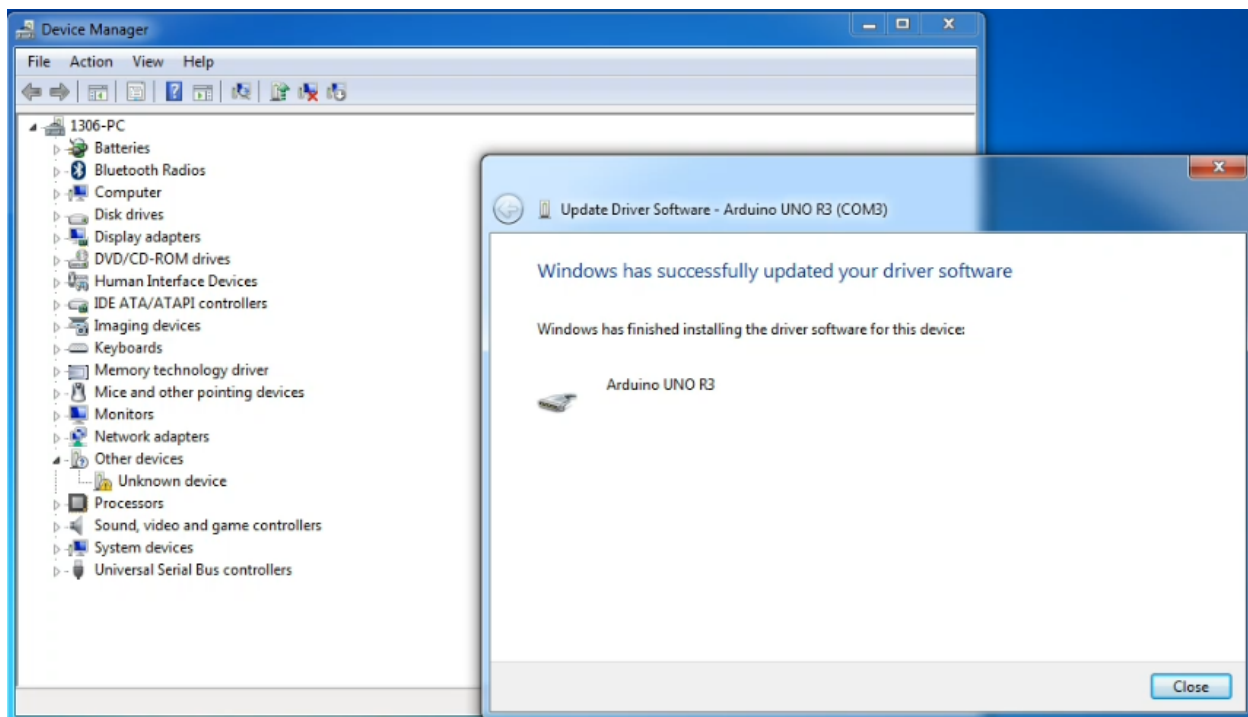


Click “Next” and you may get a security warning, if so, allow the software to be installed. Shown as below.

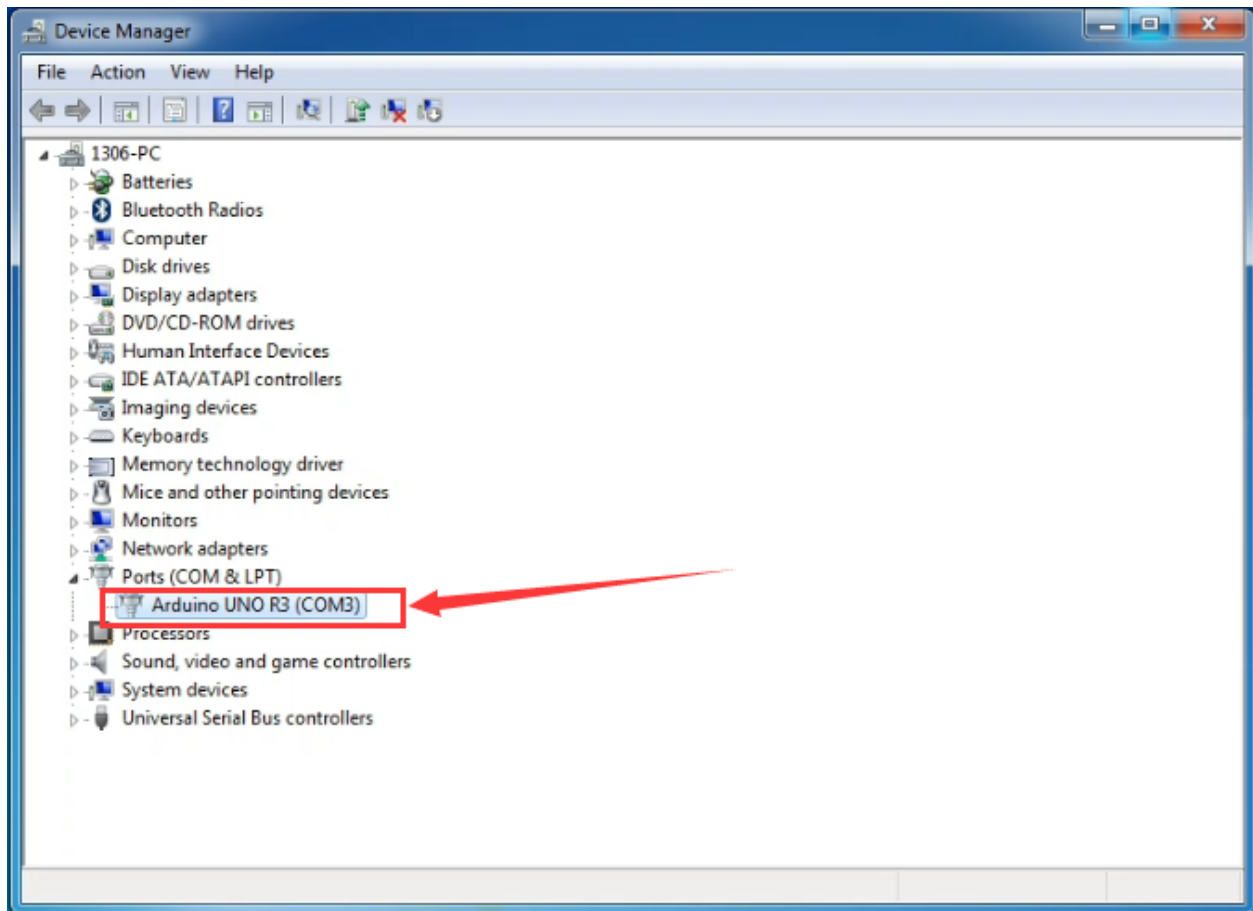




Installation completed, click “Close”.



Up to now, the driver is installed well. Then you can right click “Computer” —> “Properties”—> “Device manager”, you should see the device shown below.

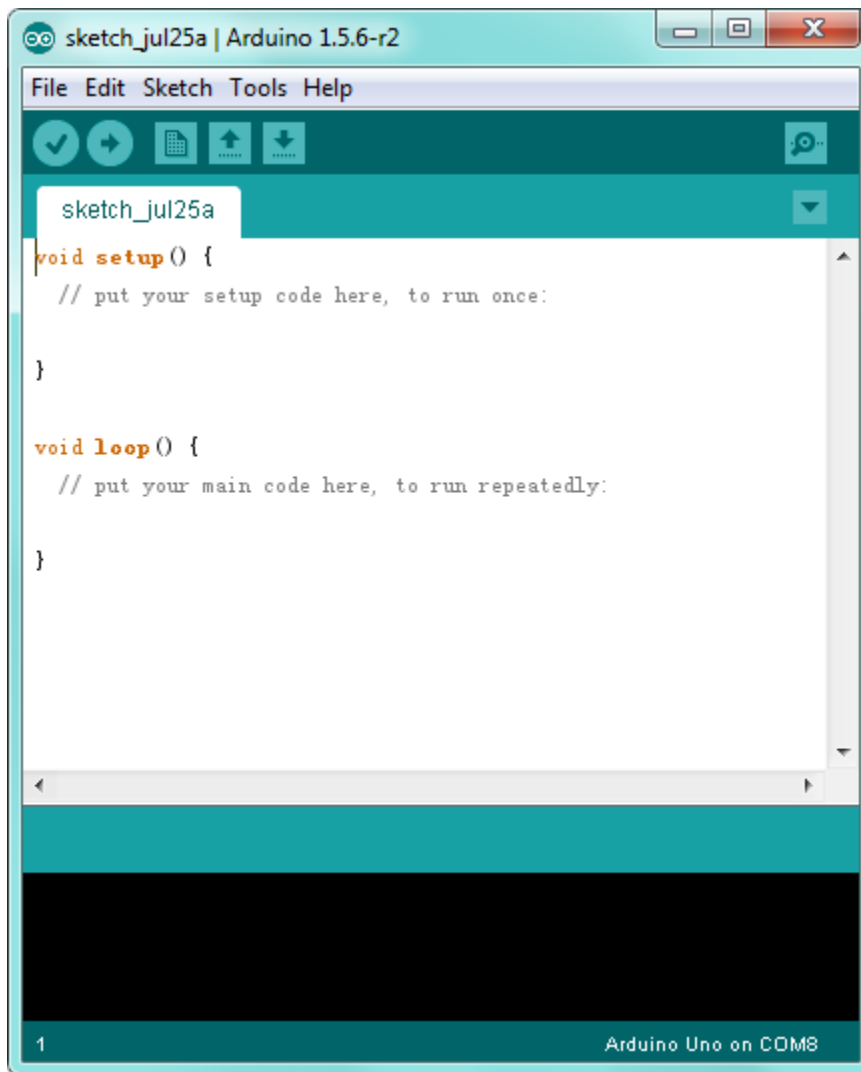


\*\* Introduction for Arduino IDE Toolbar:\*\*

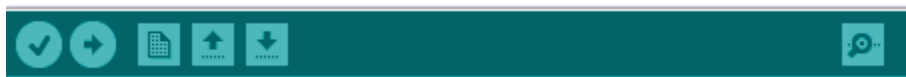
Double-click the icon of Arduino software downloaded well, you will get the interface shown below.









(Note: if the Arduino software loads in the wrong language, you can change it in the preferences dialog. See [the environment page](#) for details.)



The functions of each button on the Toolbar are listed below: <http://wiki.keyestudio.com/index.php/File:IDE.png>



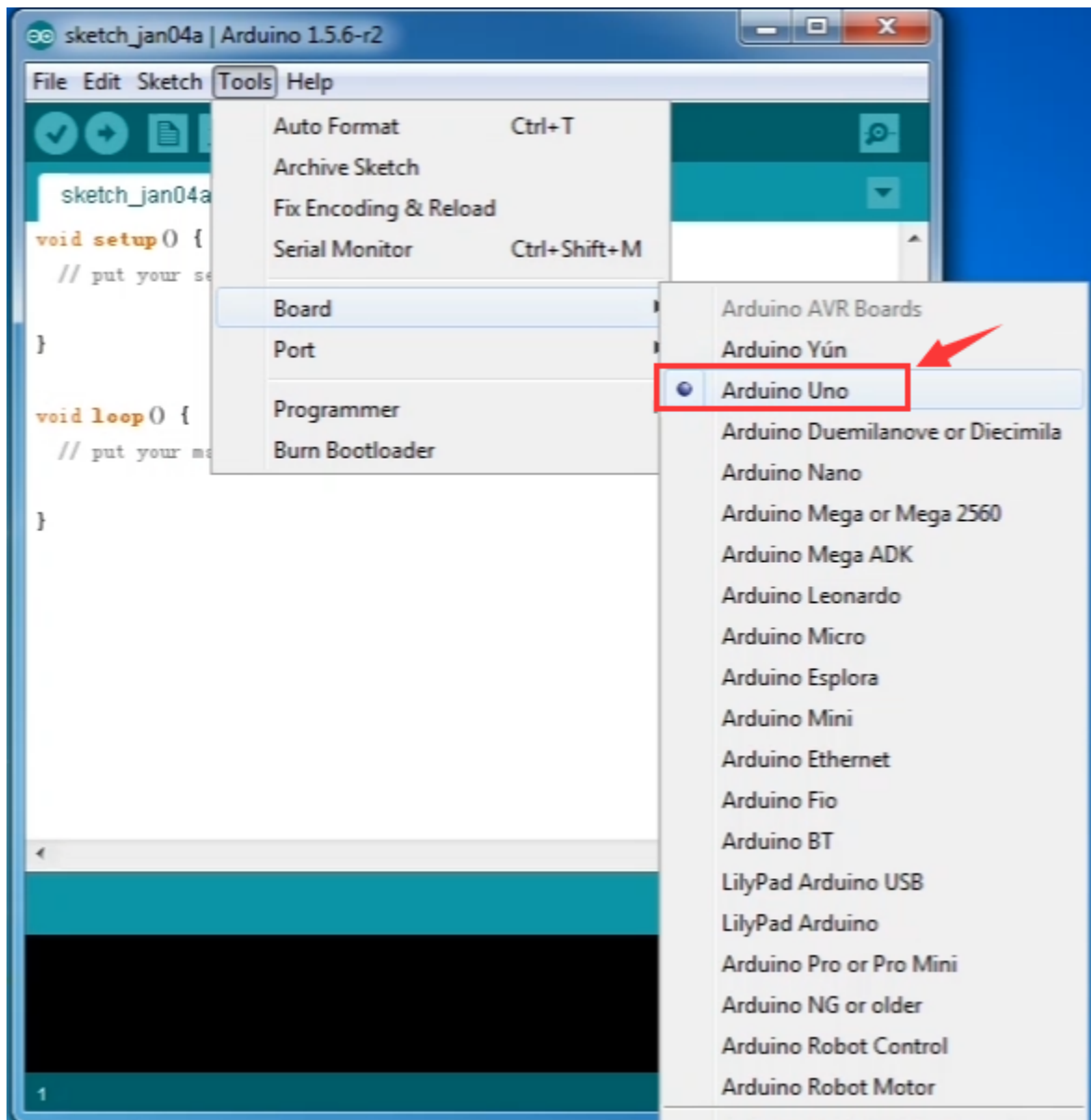
	<b>Verify/Compile</b>	Check the code for errors
	<b>Upload</b>	Upload the current Sketch to the Arduino
	<b>New</b>	Create a new blank Sketch
	<b>Open</b>	Show a list of Sketches
	<b>Save</b>	Save the current Sketch
	<b>Serial Monitor</b>	Display the serial data being sent from the Arduino

Step2| Connect the board

the board to your computer using the USB cable. The red power LED should go on.

Step3| Select the Arduino Board

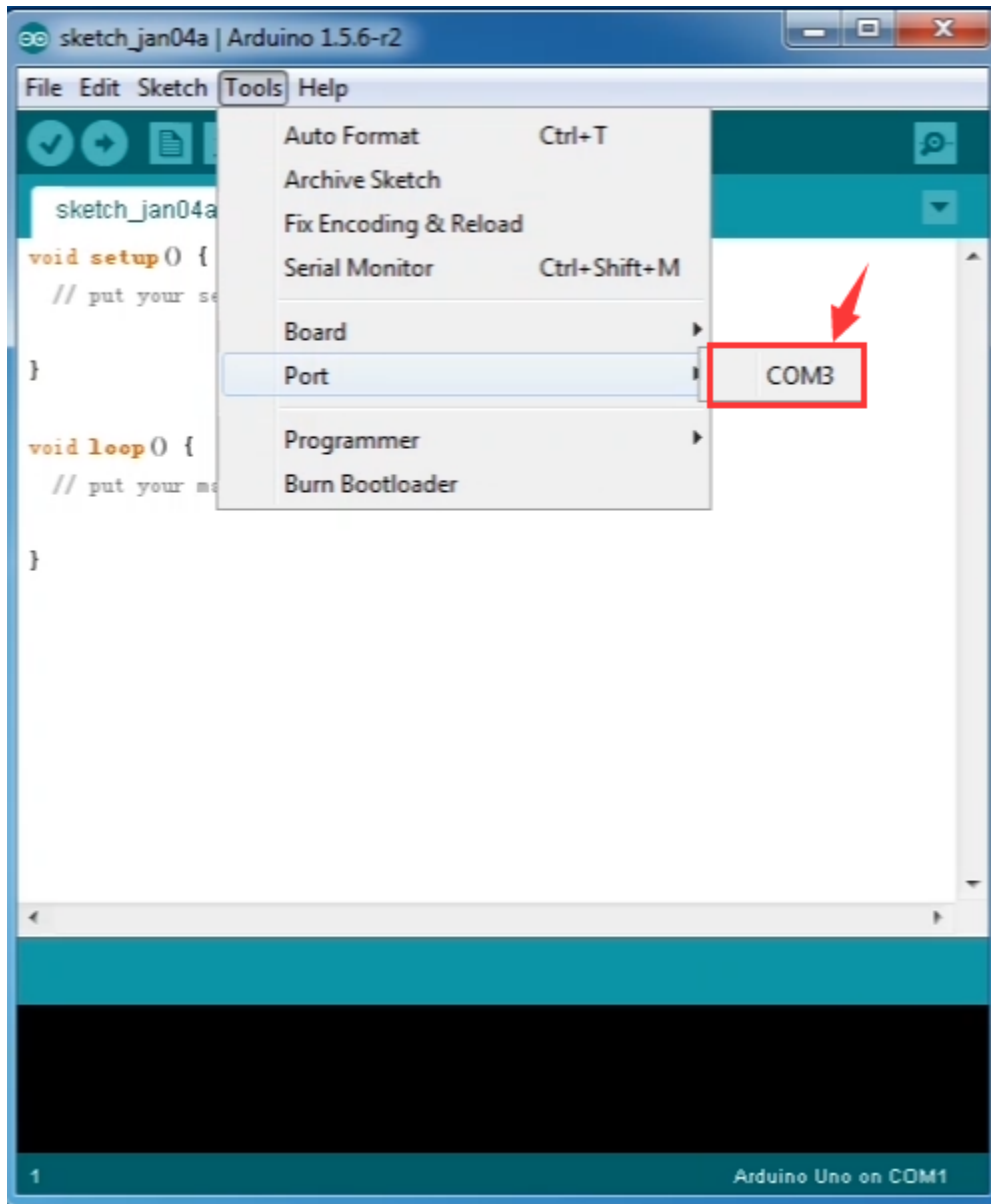
Open the Arduino IDE, you'll need to click the "Tools", then select the Board that corresponds to your Arduino.



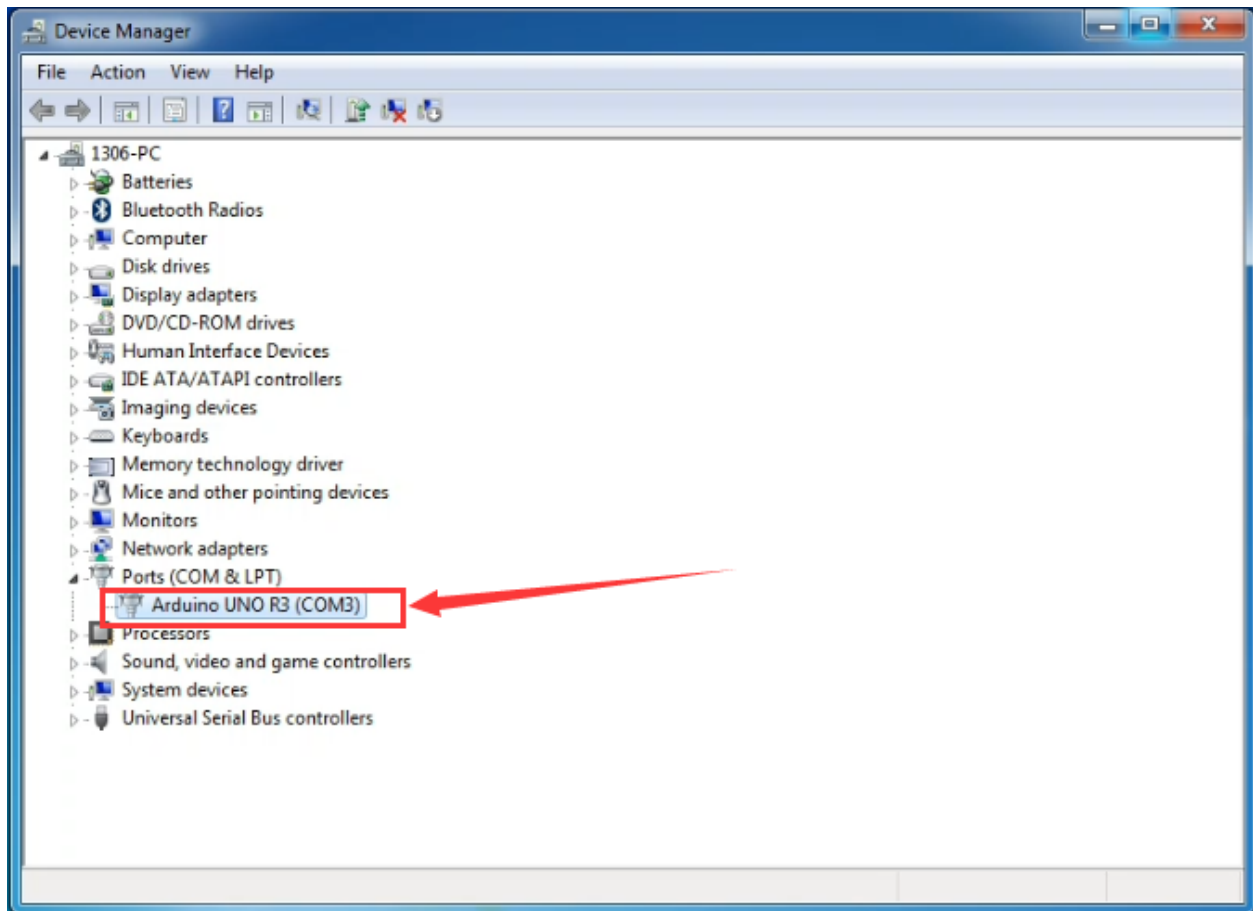
Step4| Select your serial port

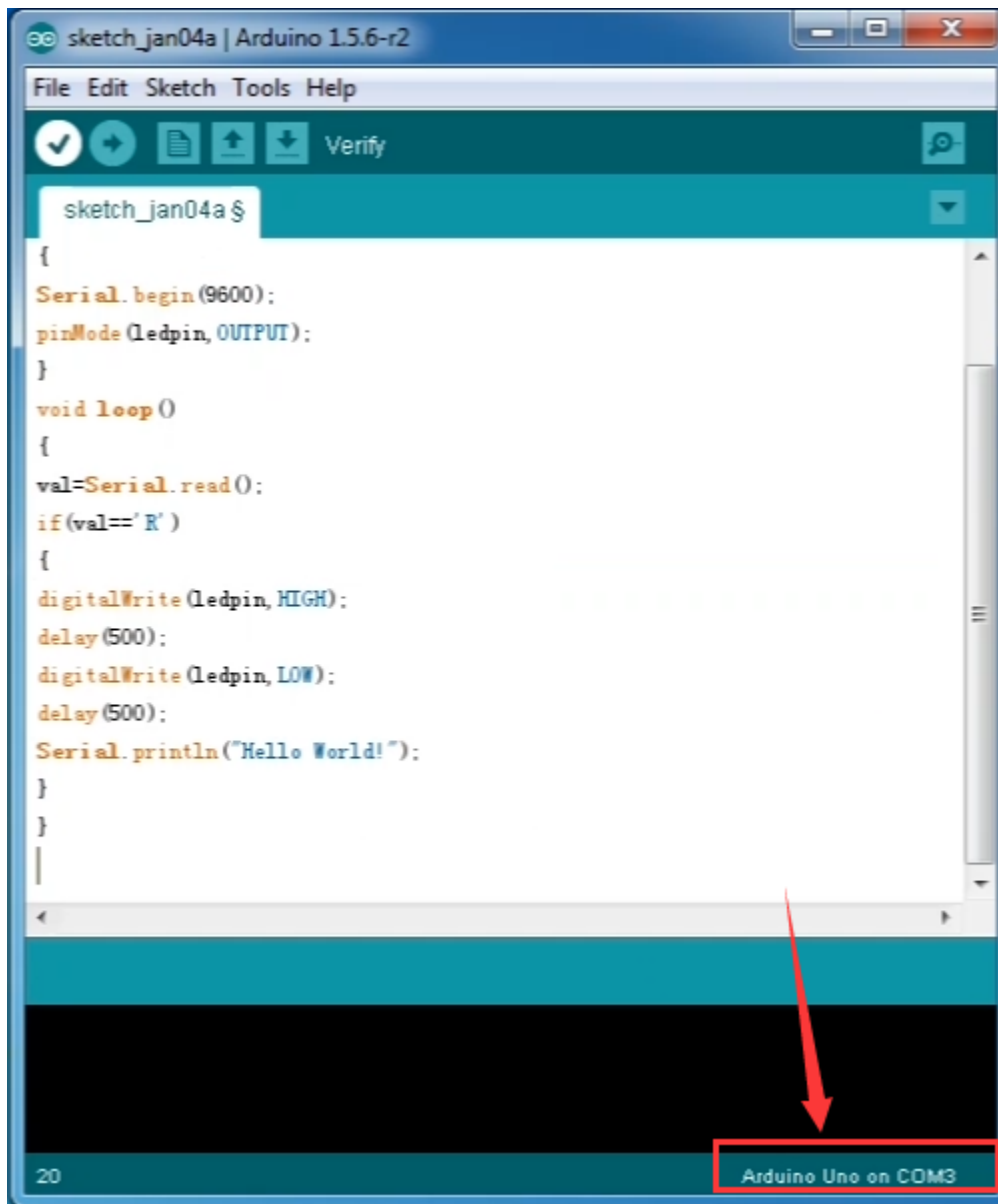
Select the serial device of the Arduino board from the **Tools | Serial Port menu**. This is likely to be COM3 or higher (COM1 and COM2 are usually reserved for hardware serial ports). To find out, you can disconnect your Arduino board and re-open the menu; the entry that disappears should be the Arduino board. Reconnect the board and select that serial port.

Here you should select COM 3 as below.



**Note:** to avoid errors, the COM Port should keep the same as the Ports shown on Device Manager.





Step5| Upload the Program

Below is an example program for displaying the Hello World!

Copy and paste the code to the Arduino environment IDE.

////////////////////////////////////////////////////////////////

```
int val;  
int ledpin=13;  
void setup()  
{
```

(continues on next page)



(continued from previous page)

```

Serial.begin(9600);

pinMode(ledpin,OUTPUT);

}

void loop()

{

val=Serial.read();

if(val=='R')

{

digitalWrite(ledpin,HIGH);

delay(500);

digitalWrite(ledpin,LOW);

delay(500);

Serial.println("Hello World!");

}

}

```

////////////////////////////////////

Then click verify button to check the errors. If compiling successfully, the message “Done compiling.” will appear in the status bar.



After that, click the “Upload” button to upload the code. Wait a few seconds - you should see the RX and TX leds on the board flashing. If the upload is successful, the message “Done uploading.” will appear in the status bar.

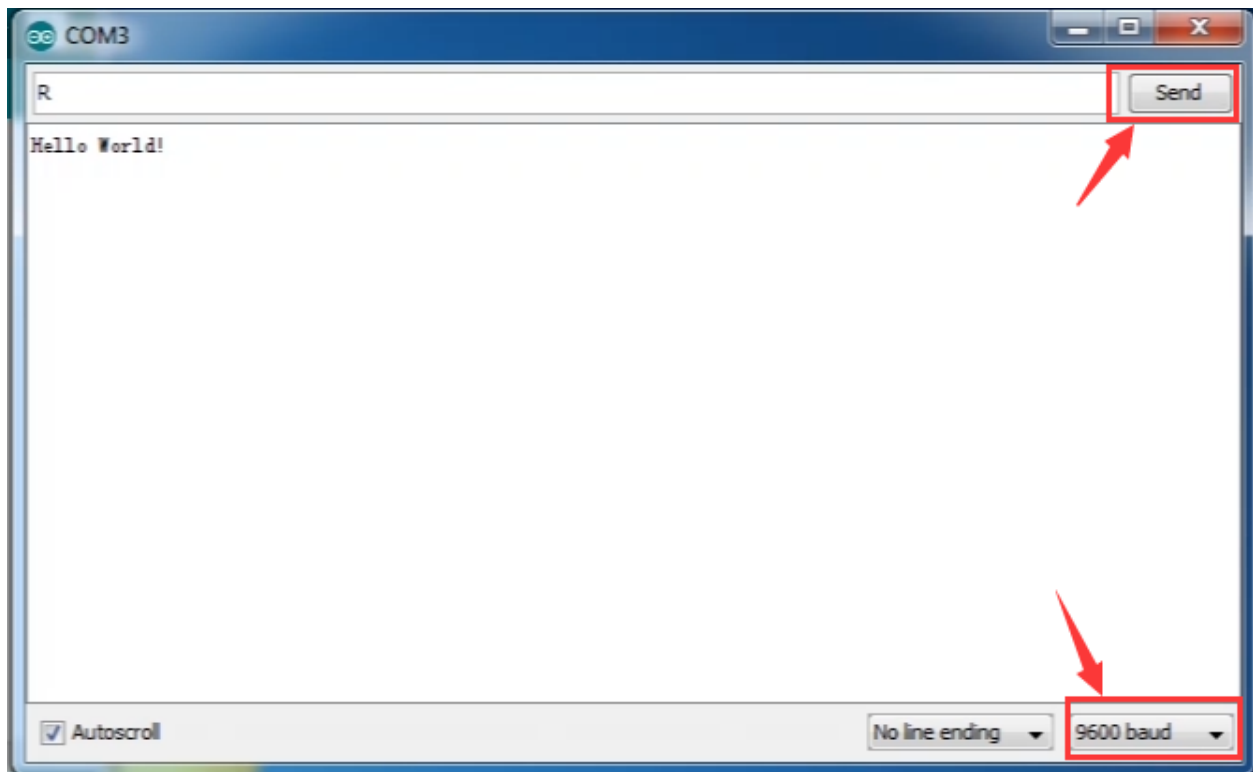


Step6| Open the Serial Monitor

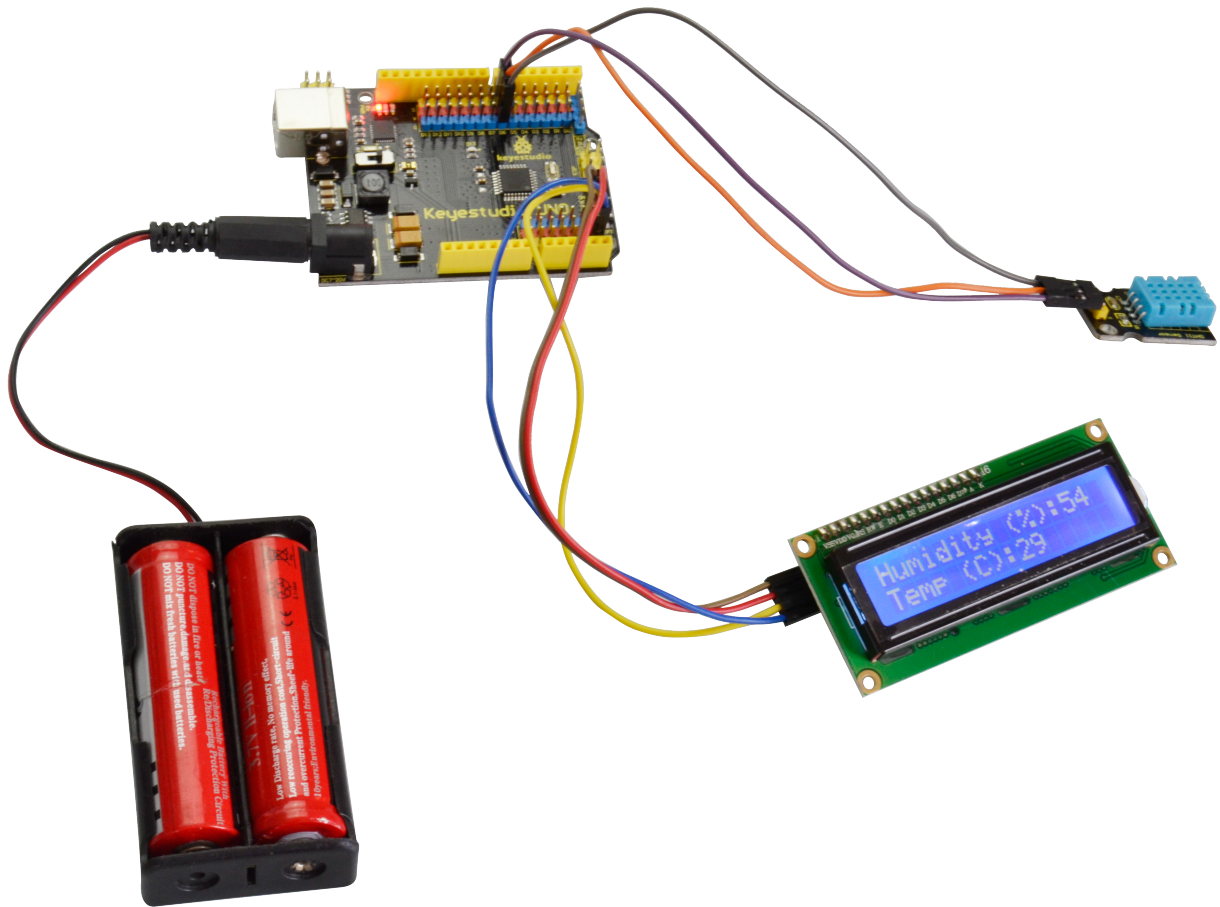
After that, click the serial monitor button to open the serial monitor.



Then set the baud rate as 9600, enter an “R” and click Send, you should see the RX led on the board blink once, and then D13 led blink once, finally “Hello World!” is showed on the monitor, the TX led blink once.



For Extension:



Other Links:

You can download the datasheet from the link:

<https://drive.google.com/open?id=1otyBI7as3S5qBd9GZYyVwYdRMLF7qDnr>

Software Download:

<https://drive.google.com/open?id=12D-JkXdNm03Qt4dIPQr3RP6OmgXqpVHc>

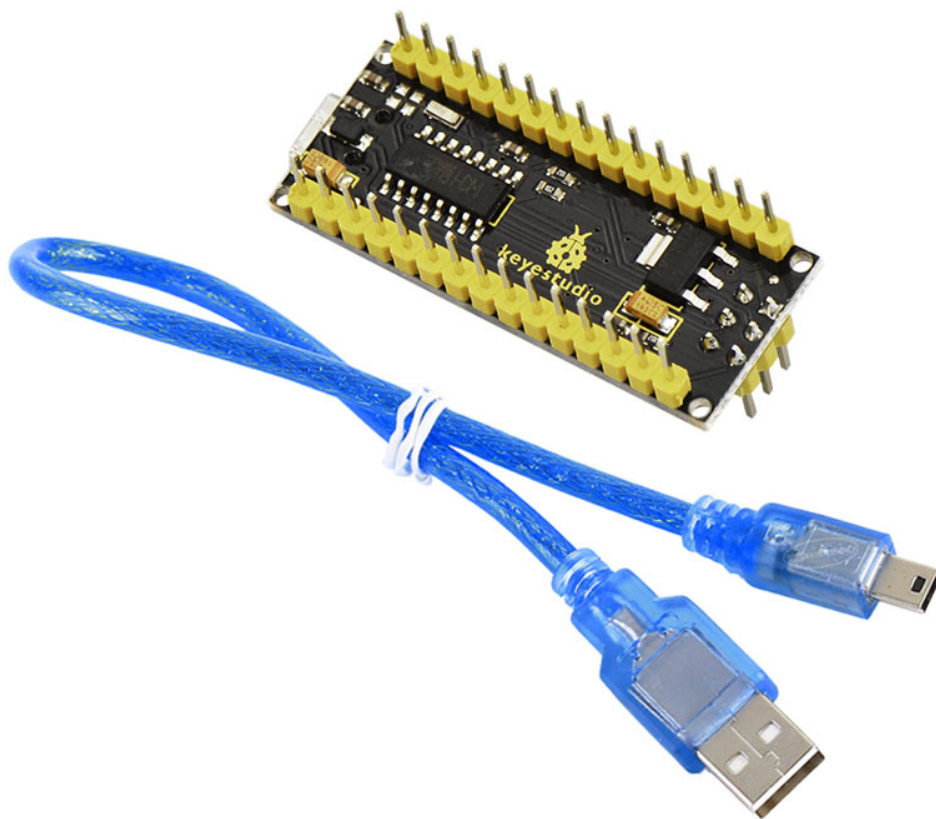
You might also want to look at:

[the reference](#) for the Arduino language;

Troubleshooting:

If you have problems, please see the [troubleshooting suggestions](#).

## KS0173 KEYESTUDIO CH340 NANO DEVELOPMENT BOARD



Description

The processor core of Keyestudio Nano CH340 is ATMEGA328P-AU. It is as same as the official Arduino Nano in addition to driver file and USB to serial chip (CH340G).

It also has 14 digital input / output interfaces (6 of which can be used as PWM output), 8 analog input interfaces, 1 16MHz crystal oscillator, 1 mini USB port, 1 ICSP interface, and a reset button.

The ICSP interface is used to program the Atmega328P-Au. We can supply power with a USB cable, the port VIN GND (DC 7-12V) and GND

### Specification

Microcontroller: ATMEGA328P-AU

Operating Voltage: 5V

Input Voltage (recommended): DC 7-12V

Digital I/O Pins: 14 (D0-D13)

PWM Digital I/O Pins6 (D3 D5 D6 D9 D10 D11)

Analog Input Pins: 8(A0-A7)

DC Current per I/O Pin: 40 mA

Flash Memory: 32 KB of which 2 KB used by bootloader

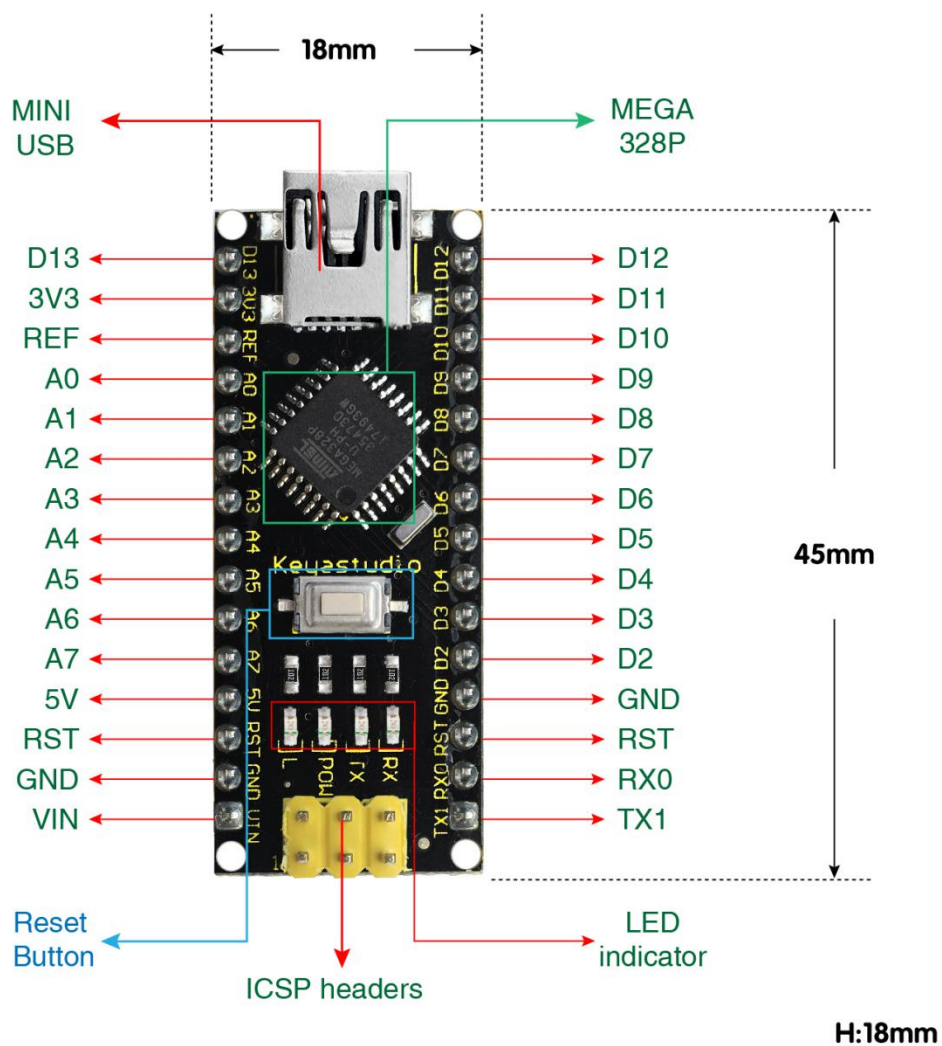
SRAM:2 KB

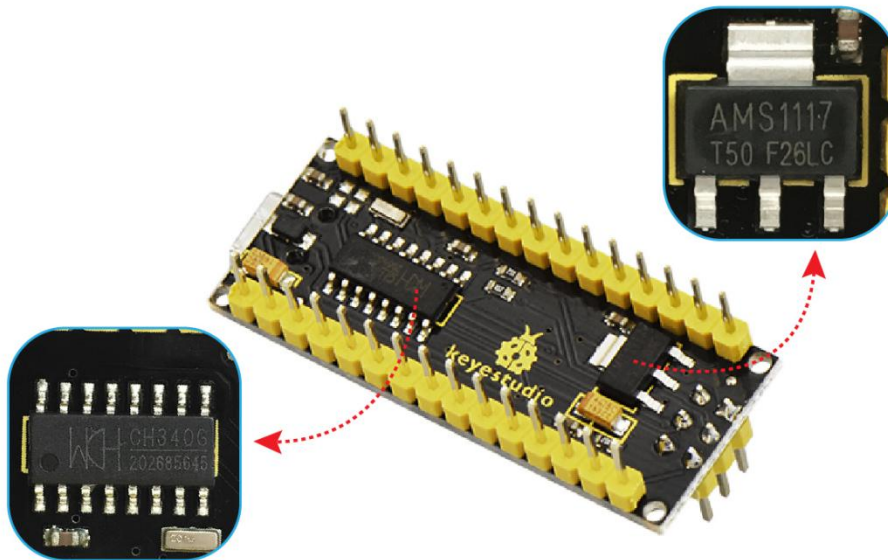
EEPROM: 1 KB

Clock Speed:16 MHz

LED\_BUILTIN:D13

### Pins







1	<b>ICSP Header</b>	<b>ICSP (In-Circuit Serial Programming) Header</b> ICSP is the AVR, an micro-program header consisting of MOSI, MISO, SCK, RESET, VCC, and GND. It is often called the SPI (serial peripheral interface) and can be considered an “extension” of output. In fact, slave the output devices under the SPI bus host. When connecting to PC, program the firmware to ATMEGA328P-AU.
2	<b>LED indicator RX</b>	Onboard you can find the label: RX(receive ) When control board communicates via serial port, receive the message, RX led flashes.
3	<b>LED indicator TX</b>	Onboard you can find the label: TX (transmit) When control board communicates via serial port, send the message, TX led flashes.
4	<b>LED indicator POW</b>	Power up the control board, LED on, otherwise LED off.
5	<b>LED indicator L</b>	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.
6	<b>RX0D0 TX1D1 D2-D13</b>	It has 14 digital input/output pins D0-D13 (of which 6 can be used as PWM outputs). These pins can be configured as digital input pin to read the logic value (0 or 1). Or used as digital output pin to drive different modules like LED, relay, etc.
7	<b>RST</b>	Reset pin: connect external button. The function is the same as RESET button.
8	<b>MEGA 328P</b>	Each board has its own microcontroller. You can regard it as the brain of your board. Microcontrollers are usually from ATMEL. Before you load a new program on the Arduino IDE, you must know what IC is on your board. This information can be checked at the top surface of IC. The board's microcontroller is ATMEGA328P-AU. More info. see the <a href="#">datasheet</a>
9	<b>MINI USB</b>	The board can be powered via Mini-B USB connection. Also upload the program to the board via USB port.
10	<b>3V3 pin</b>	Provides 3.3V voltage output
11	<b>REF</b>	Reference external voltage (0-5 volts) for the analog input pins. Used with <a href="#">analogReference()</a> .
12	<b>A0-A7</b>	The Nano has 8 Analog Pins, labeled A0 through A7.
13	<b>5V pin</b>	Provides 5V voltage output
14	<b>GND</b>	Ground pin
15	<b>VIN</b>	Input an external voltage DC7-12V to power the board.
16	<b>Reset Button</b>	Used to reset the control board
17	<b>CH340G</b>	USB-to-serial port chip, converting the USB signal into Serial port signal.
18	<b>AMS1117</b>	Convert the external voltage input DC7-12V into DC5V, then transfer it to the processor and other elements.

### 3. Specialized Functions of Some Pins:

- **Serial communication:** 0 (RX) and 1 (TX). Used to receive (RX) and transmit (TX) TTL serial data.
- **PWM (Pulse-Width Modulation):** D3, D5, D6, D9, D10, D11

- **External Interrupts:** D2 (interrupt 0) and D3 (interrupt 1). 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.
- **SPI communication:** D10 (SS), D11 (MOSI), D12 (MISO), D13 (SCK).
- **IIC communication:** A4 (SDA); A5(SCL)

Windows System

### 4.1 Download the Arduino IDE

When getting this control board, we need to install Arduino IDE

Enter the website <https://www.arduino.cc/> and click



You can select the latest version—1.8.13. Alternatively, the previous release is your another choice.

In this project, we use 1.8.12 version.

## Previous Releases

Download the previous version of the current release, the classic 1.0.x, or old beta releases.

### DOWNLOAD OPTIONS

**Previous Release (1.8.12)**

**Arduino 1.0.x**

**Arduino 1.5.x beta**

**Arduino 1.9.x beta**

Click **Previous Release (1.8.12)** to enter the new page. As shown below;

The **Windows installer** needs installing manually. Yet , the **Windows zip file for non admin install** a zip file of Arduino 1.8.12 version, can be directly downloaded and installed.

## Previous IDE Releases

### ARDUINO 1.8.12

Arduino IDE that can be used with any Arduino board, including the Arduino Yún and Arduino DUE. Refer to the [Getting Started](#) page for Installation instructions.

[See the release notes.](#)

**Windows** Installer

**Windows** ZIP file for non admin install

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

**Linux** 32 bits

**Linux** 64 bits

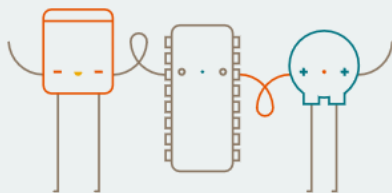
**Linux** ARM 32

**Linux** ARM 64

**Source**

## Contribute to the Arduino Software

Consider supporting the Arduino Software by contributing to its development. (US tax payers, please note this contribution is not tax deductible). [Learn more on how your contribution will be used.](#)



SINCE MARCH 2015, THE ARDUINO IDE HAS BEEN DOWNLOADED **45,290,706** TIMES. (IMPRESSIVE!) NO LONGER JUST FOR ARDUINO AND GENUINO BOARDS, HUNDREDS OF COMPANIES AROUND THE WORLD ARE USING THE IDE TO PROGRAM THEIR DEVICES, INCLUDING COMPATIBLES, CLONES, AND EVEN COUNTERFEITS. HELP ACCELERATE ITS DEVELOPMENT WITH A SMALL CONTRIBUTION! REMEMBER: OPEN SOURCE IS LOVE!

\$3

\$5

\$10

\$25

\$50

OTHER

JUST DOWNLOAD

CONTRIBUTE & DOWNLOAD

 JUST DOWNLOAD

CONTRIBUTE &amp; DOWNLOAD

Click icon

to download Arduino IDE.

#### 4.2 Download Driver of CH340

<https://fs.keyestudio.com/CH340-WIN>

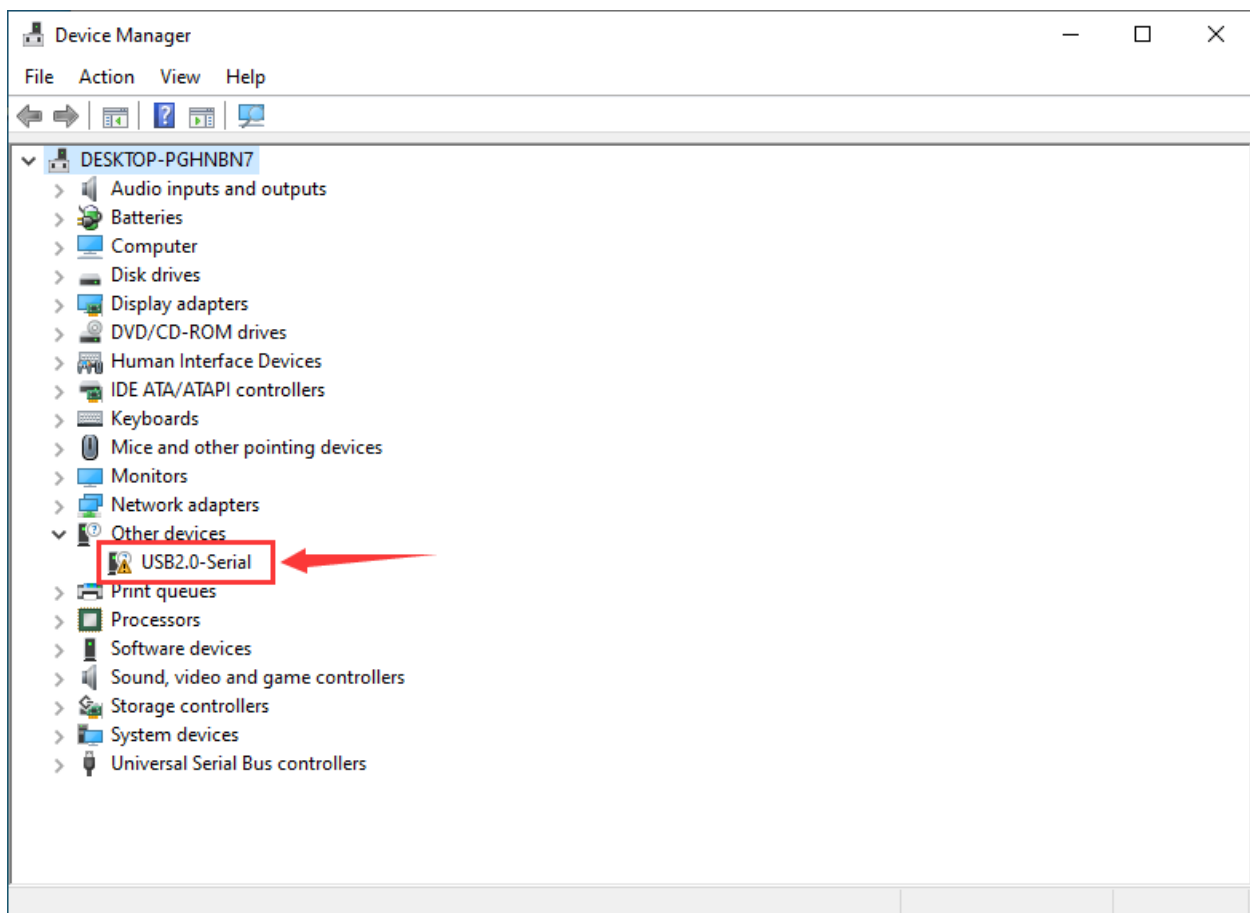
#### 4.3 Install the Driver


For this part, we need to install the driver of Arduino IDE

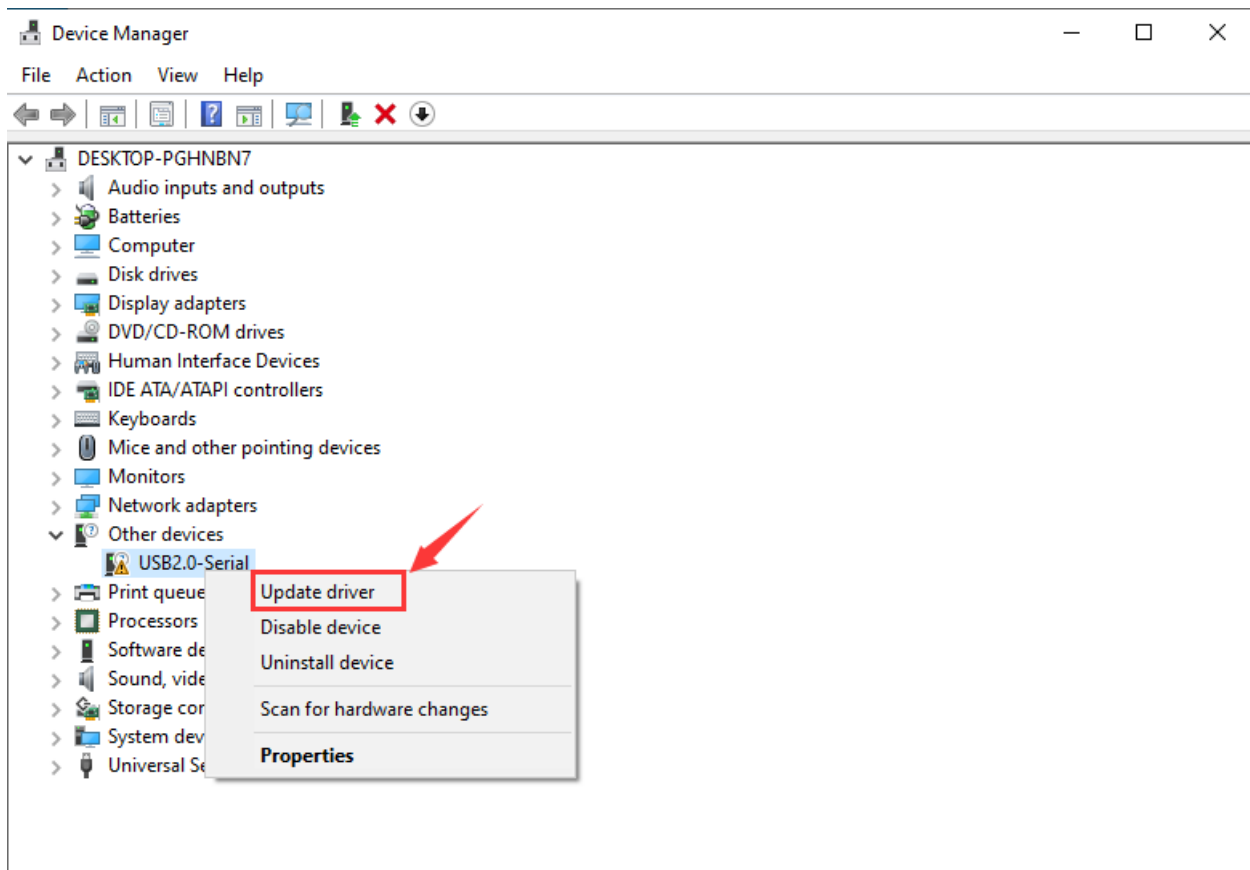
First, let's attach USB cable to computer. The driver can be installed automatically if the PC system is Windows 10, however, you need to install the driver manually if the PC system is other version.

The USB to serial chip of control board is CH340G, therefore, we will install its driver(usb\_ch341\_3.1.2009.06).

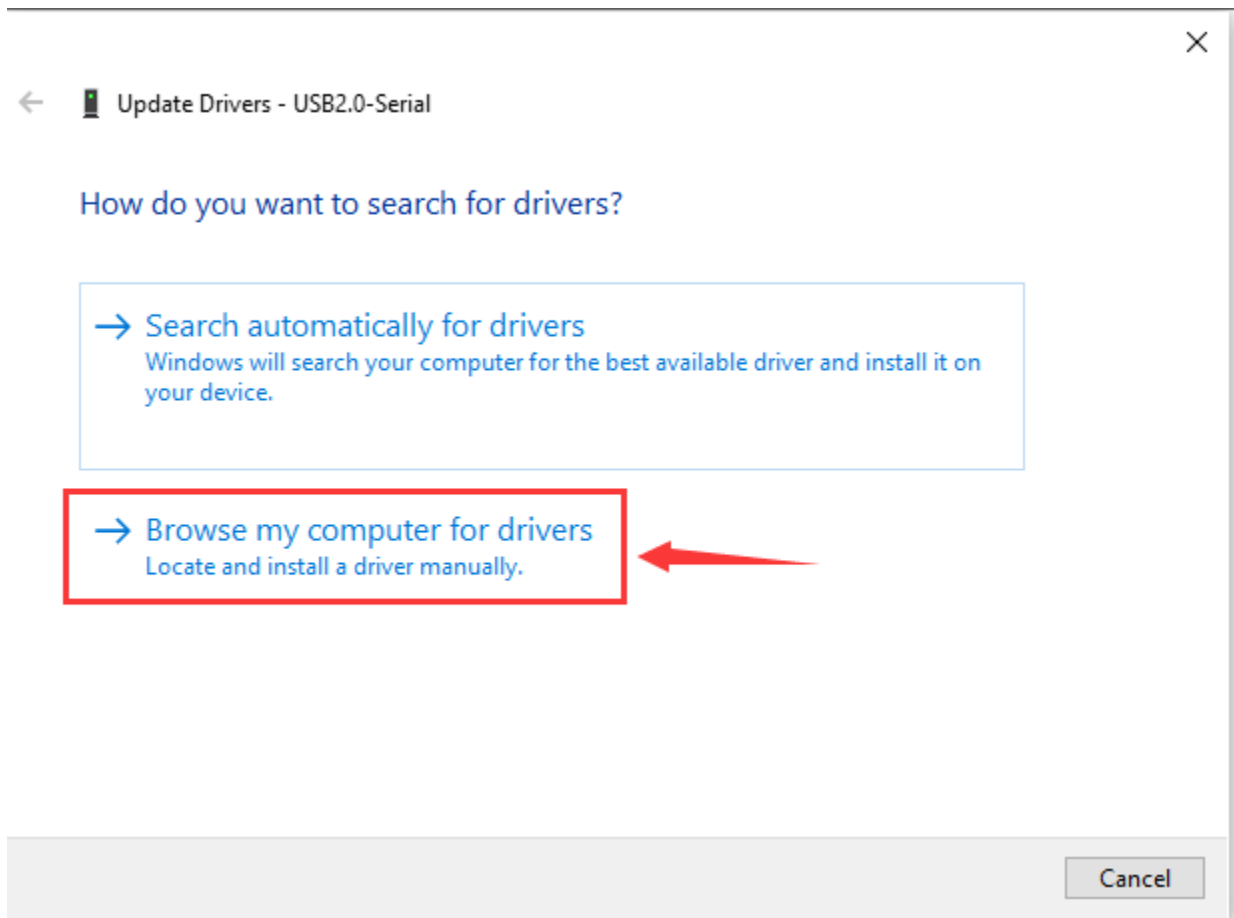
Click Computer— Properties— Device Manager, as shown below:



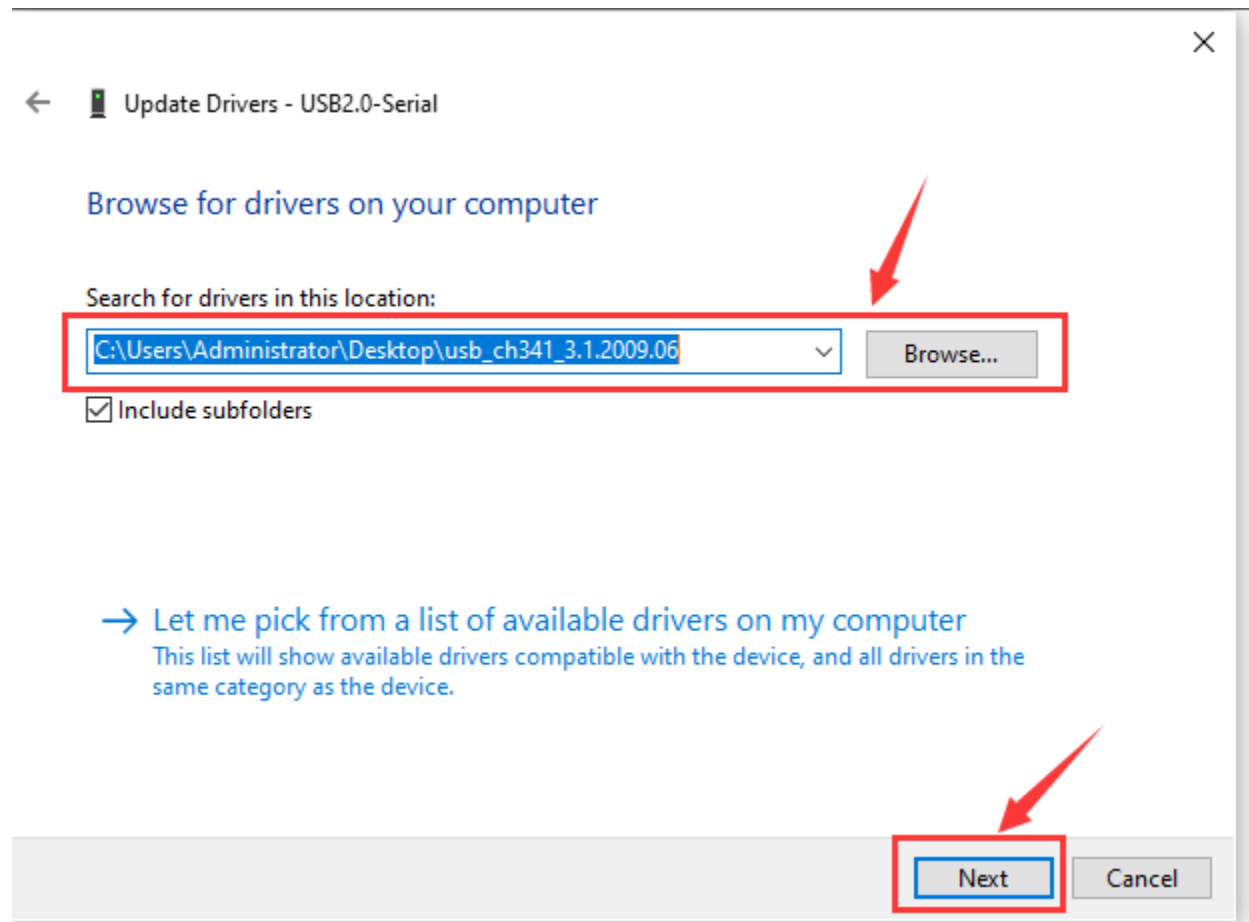
Click  USB Serial and “Update Driver Program”



Jump into the following page and select “Browse my computer for driver software”.



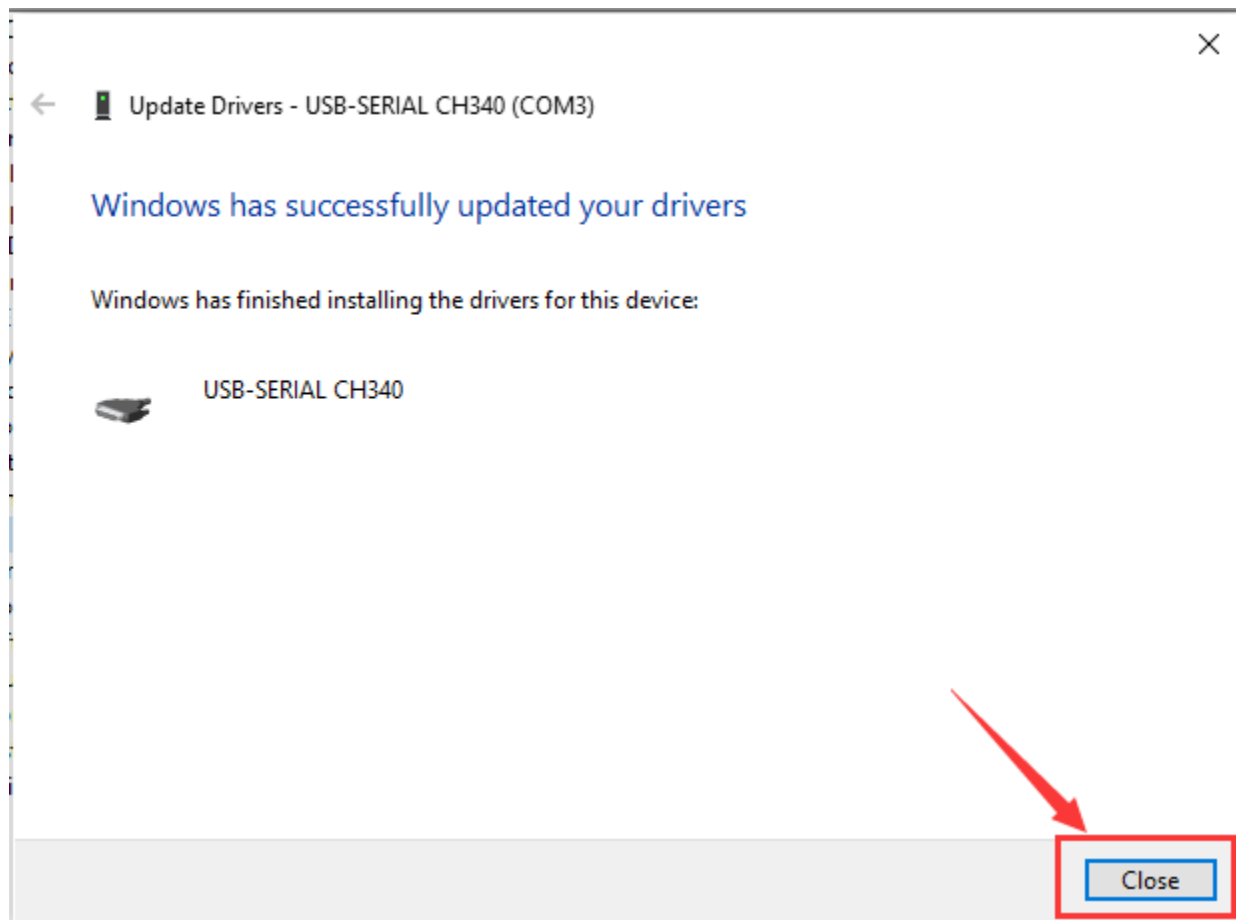
Search the **usb\_ch341\_3.1.2009.06** folder



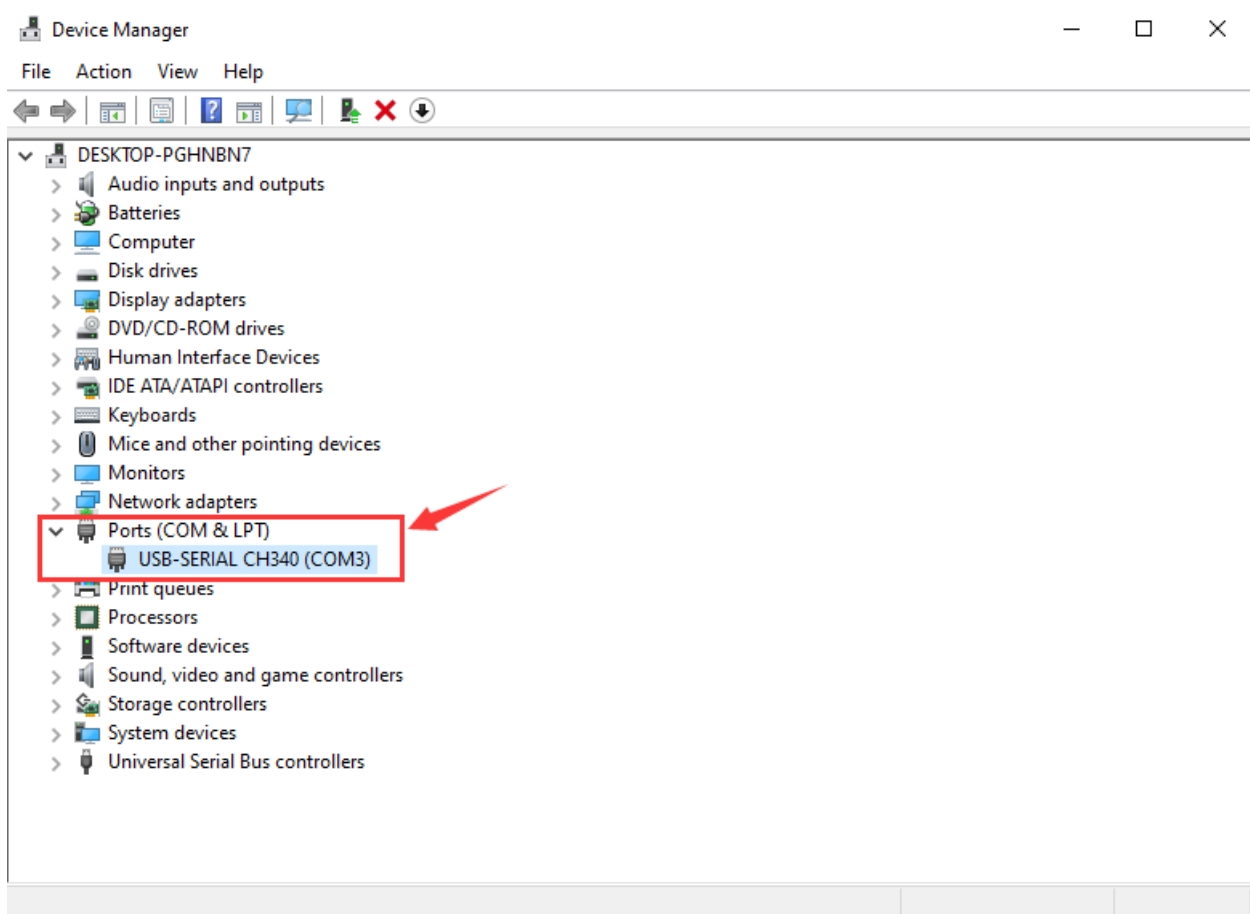
After the driver is installed, you need to click **Close**.

[http://wiki.keyestudio.com/index.php/File:Driver\\_6.png](http://wiki.keyestudio.com/index.php/File:Driver_6.png)

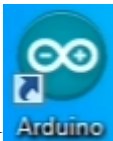





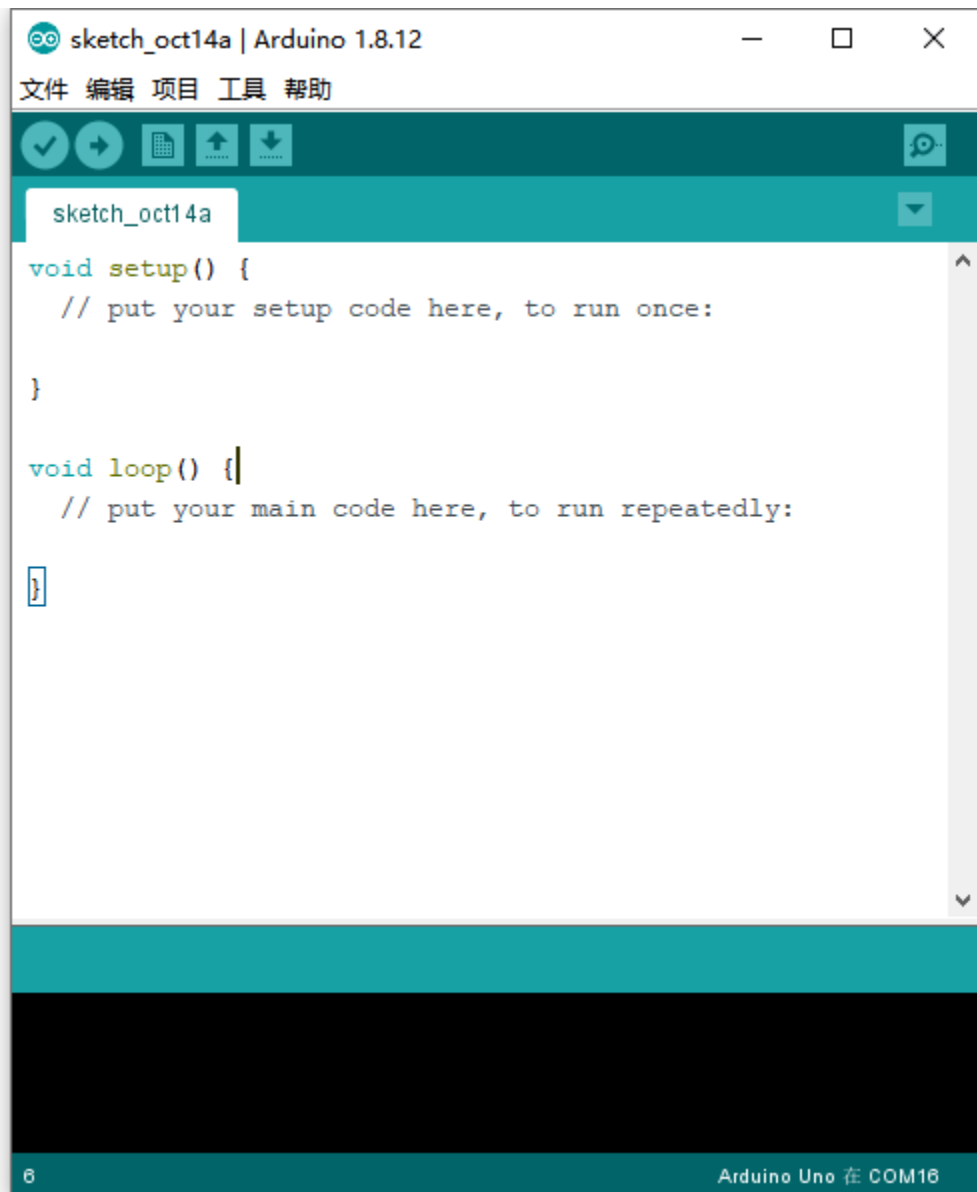
Click Computer— Properties— Device Manager, as shown below.



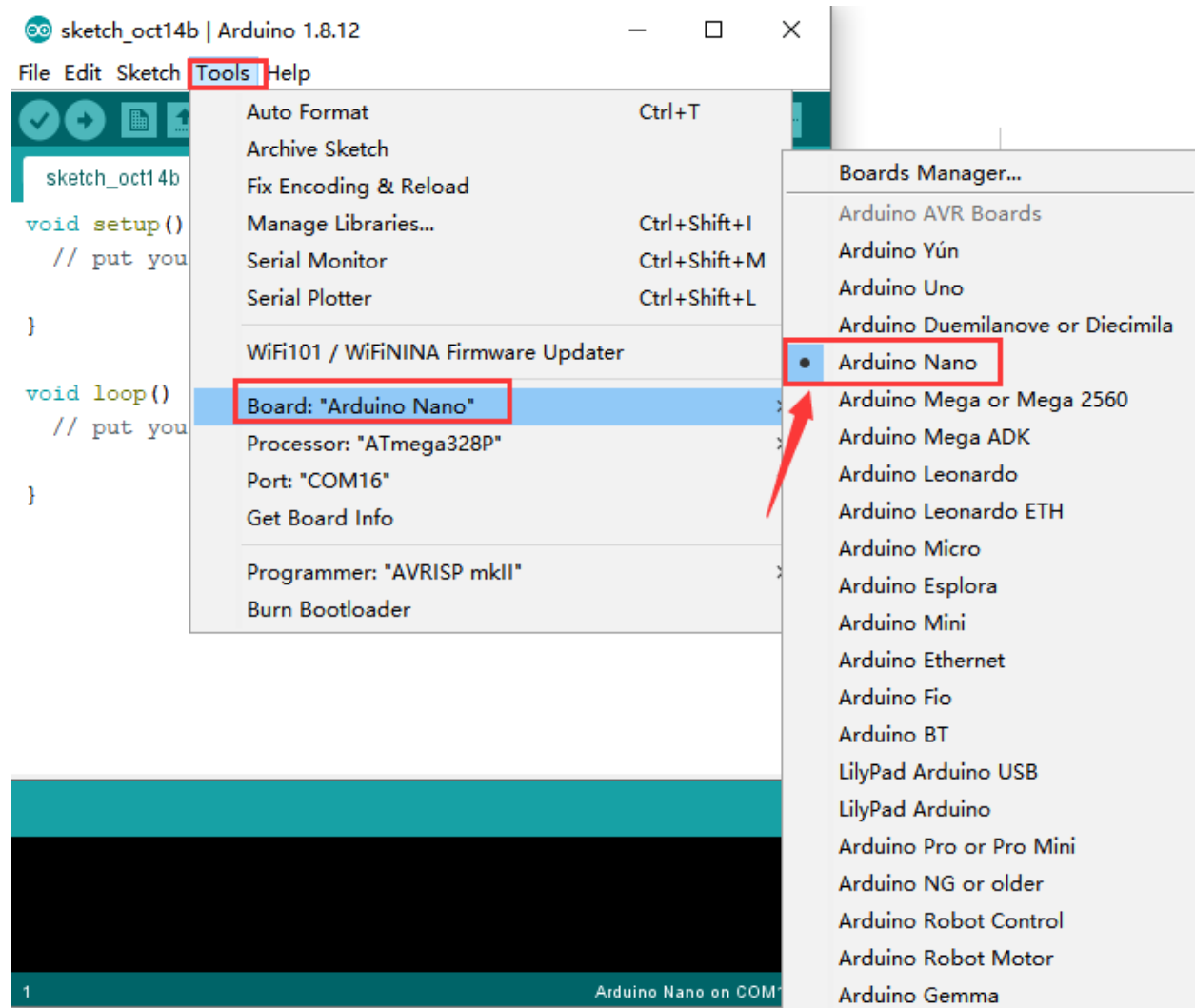
### 4.4 Arduino IDE Setting

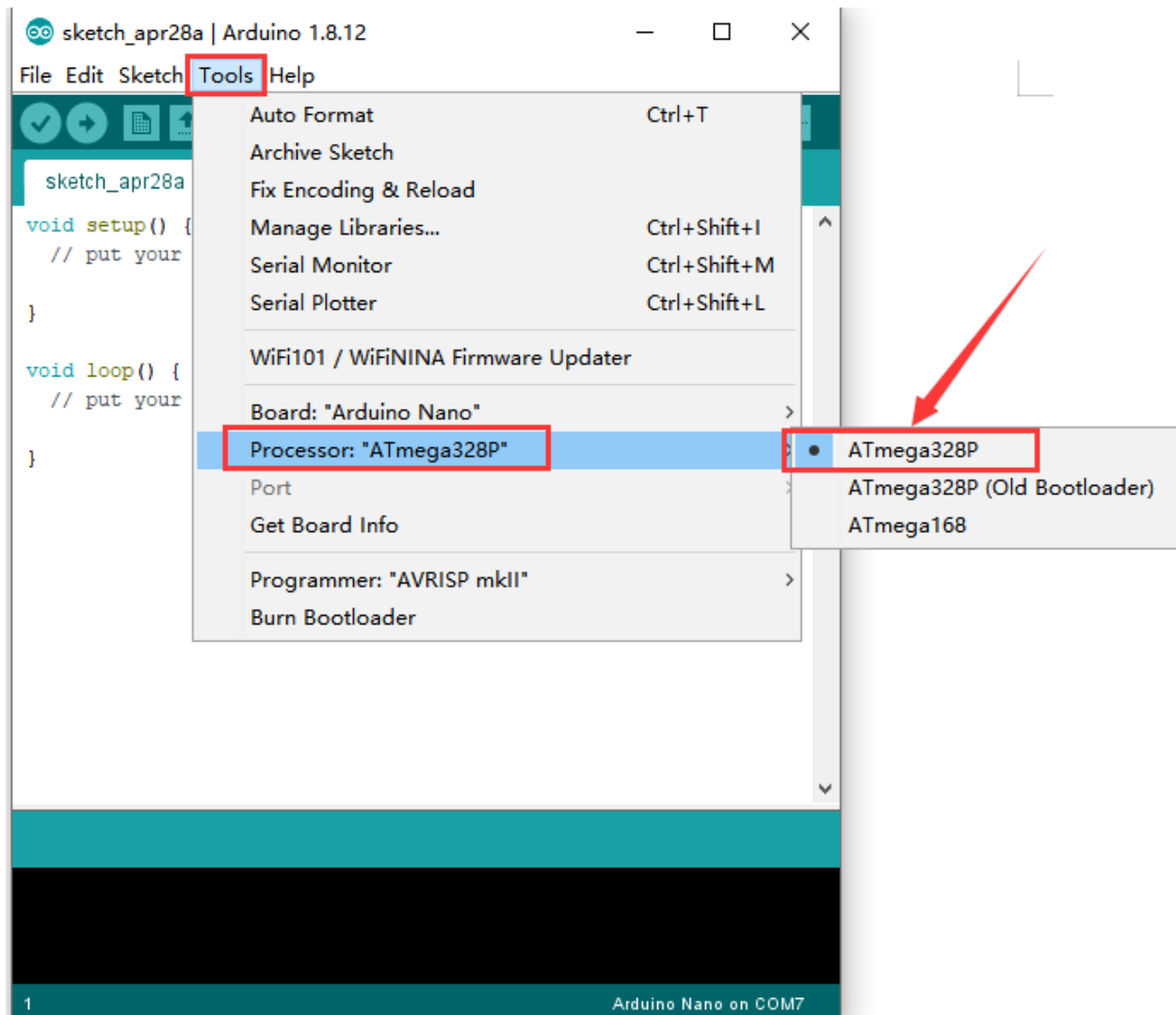


Click  icon to pen Arduino IDE.

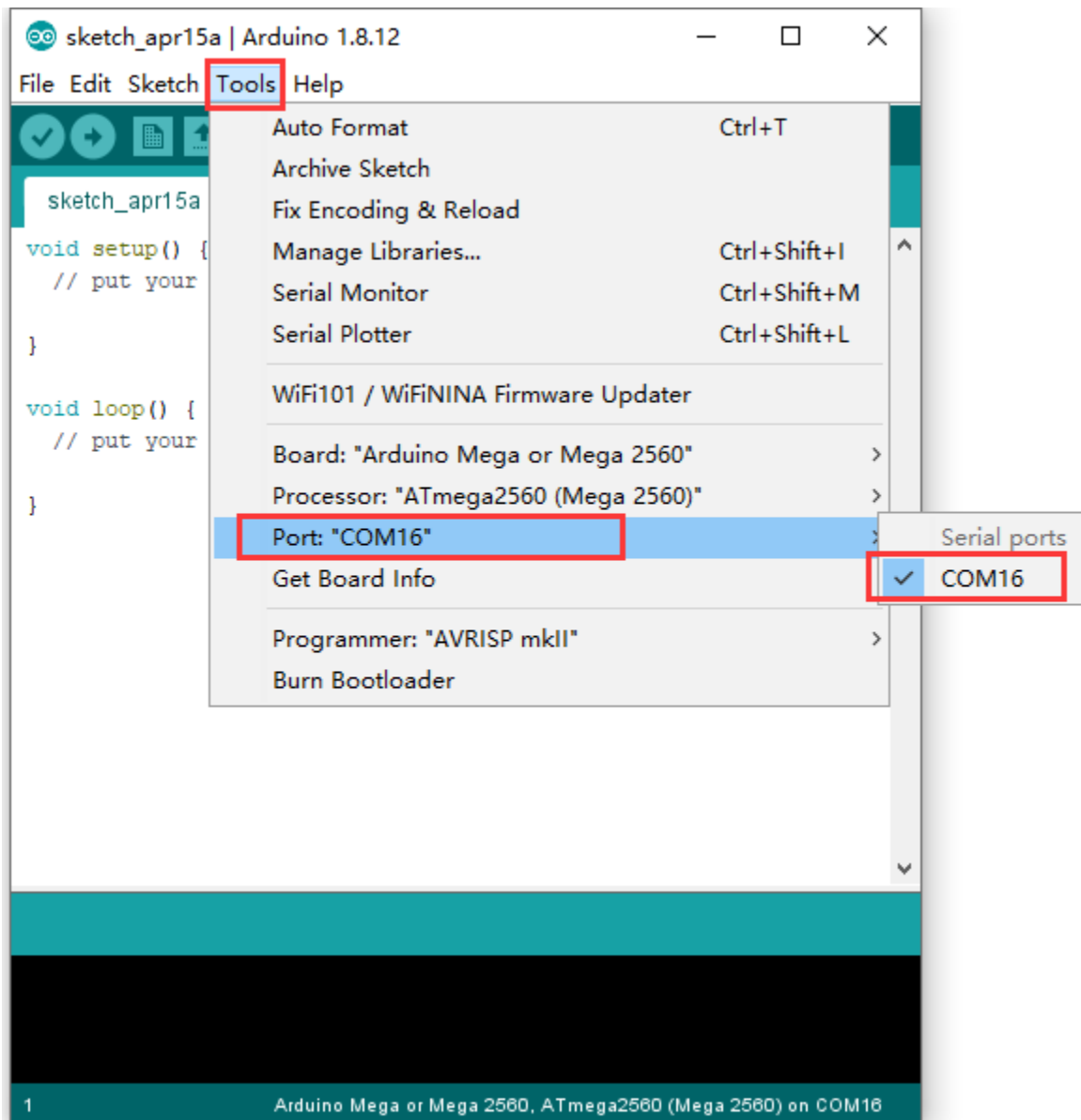


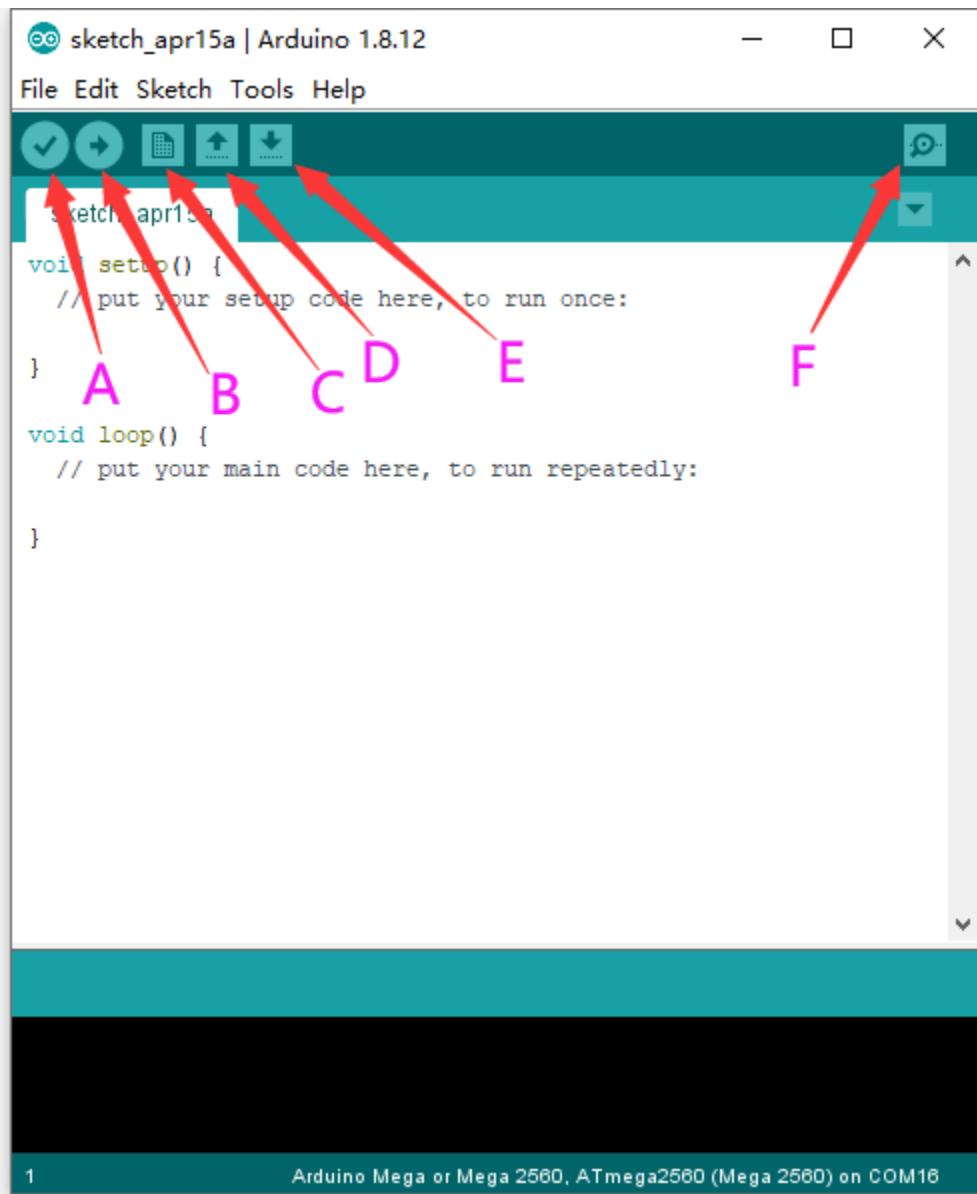
When downloading the sketch to the board, you must select the correct name of Arduino board that matches the board connected to your computer. As shown below;





Then select the correct COM port (you can see the corresponding COM port after the driver is successfully installed).





A- Used to verify whether there is any compiling mistakes or not.

B- Used to upload the sketch to your Arduino board.

C- Used to create shortcut window of a new sketch.

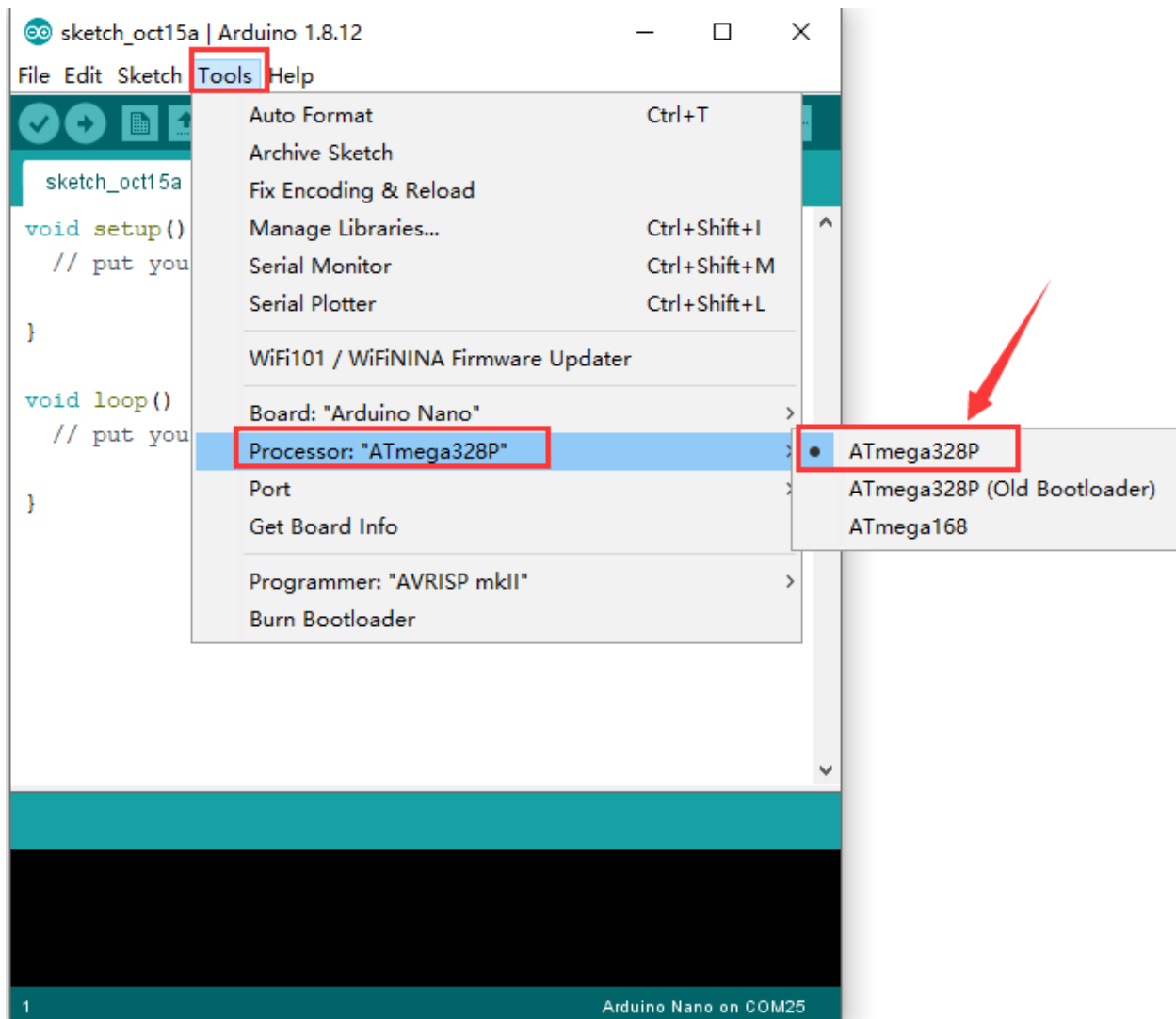
D- Used to directly open an example sketch.

E- Used to save the sketch.

F- Used to send the serial data received from board to the serial monitor.

#### 4.5 Setting IDE for New Bootloader

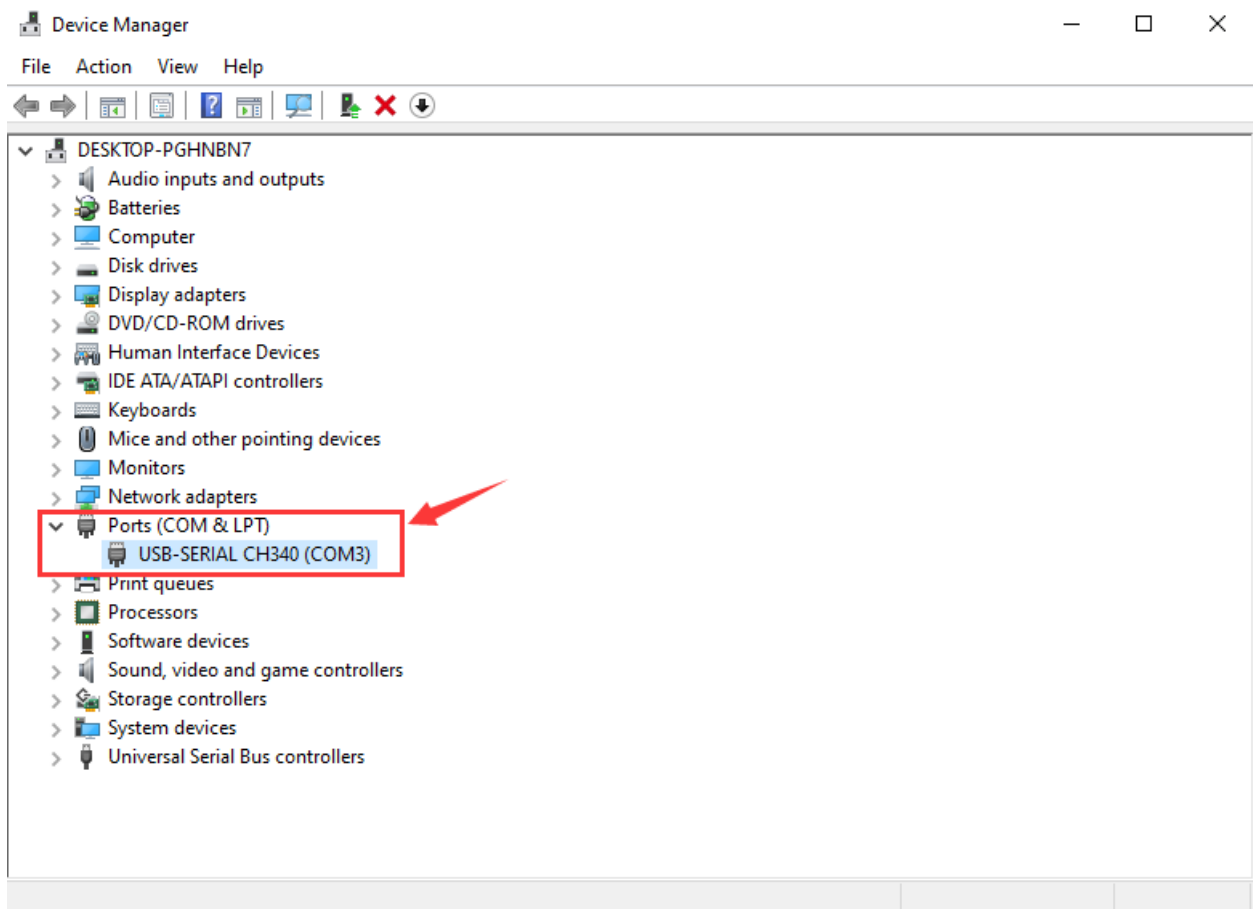
Arduino Nano board could burn new and old bootloader. New bootloader is only compatible with 1.8.9 IDE and above. Yet the old one is compatible with all versions.

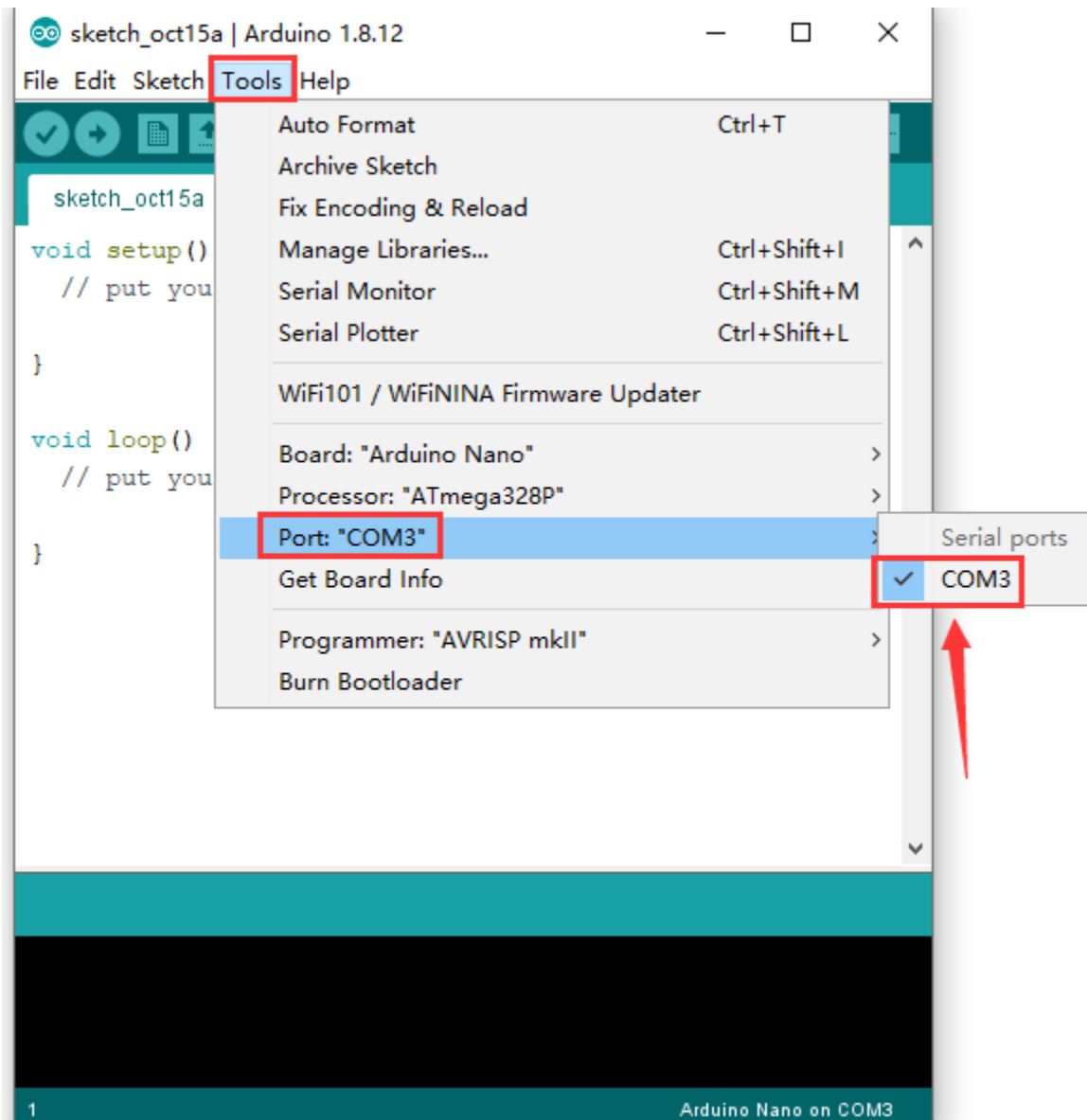


### 4.6 Select Your Serial Port

Select the serial device of the Arduino board from the **Tools | Serial Port** menu. **Note:** to avoid errors, the COM Port should keep the same as the Ports shown on Device Manager.





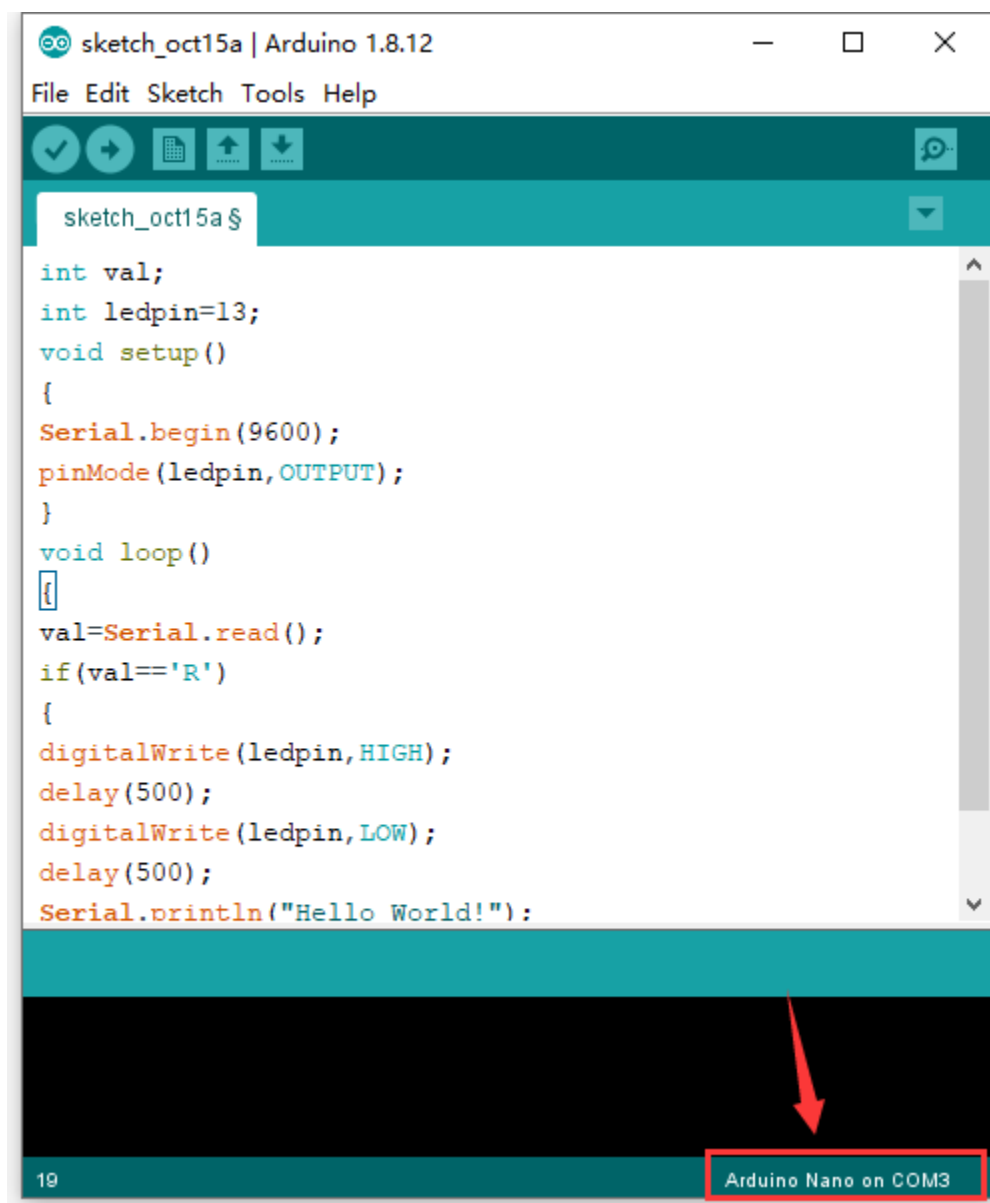


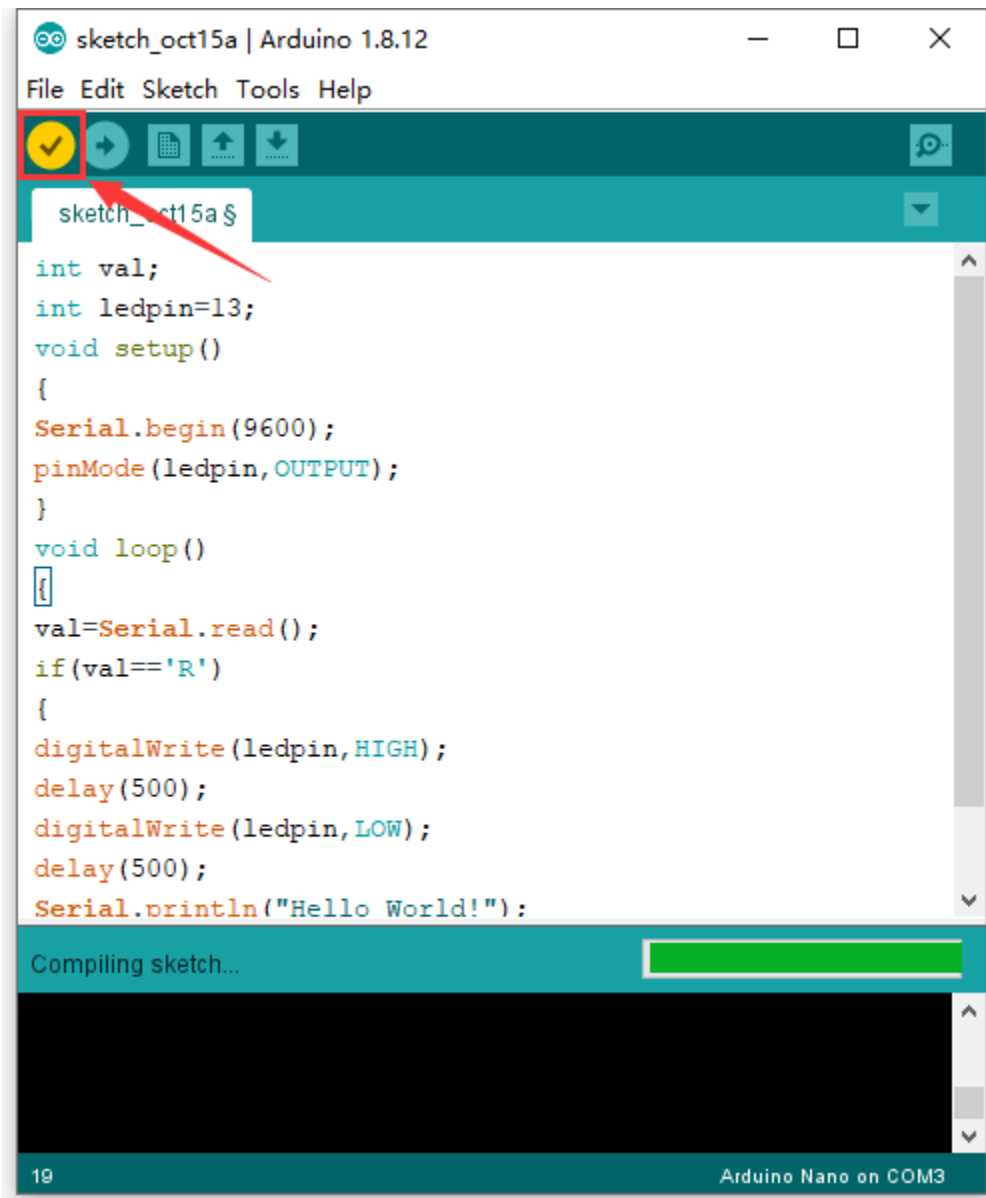
#### 4.7 Hello World!

Copy the following code in the Arduino IDE.

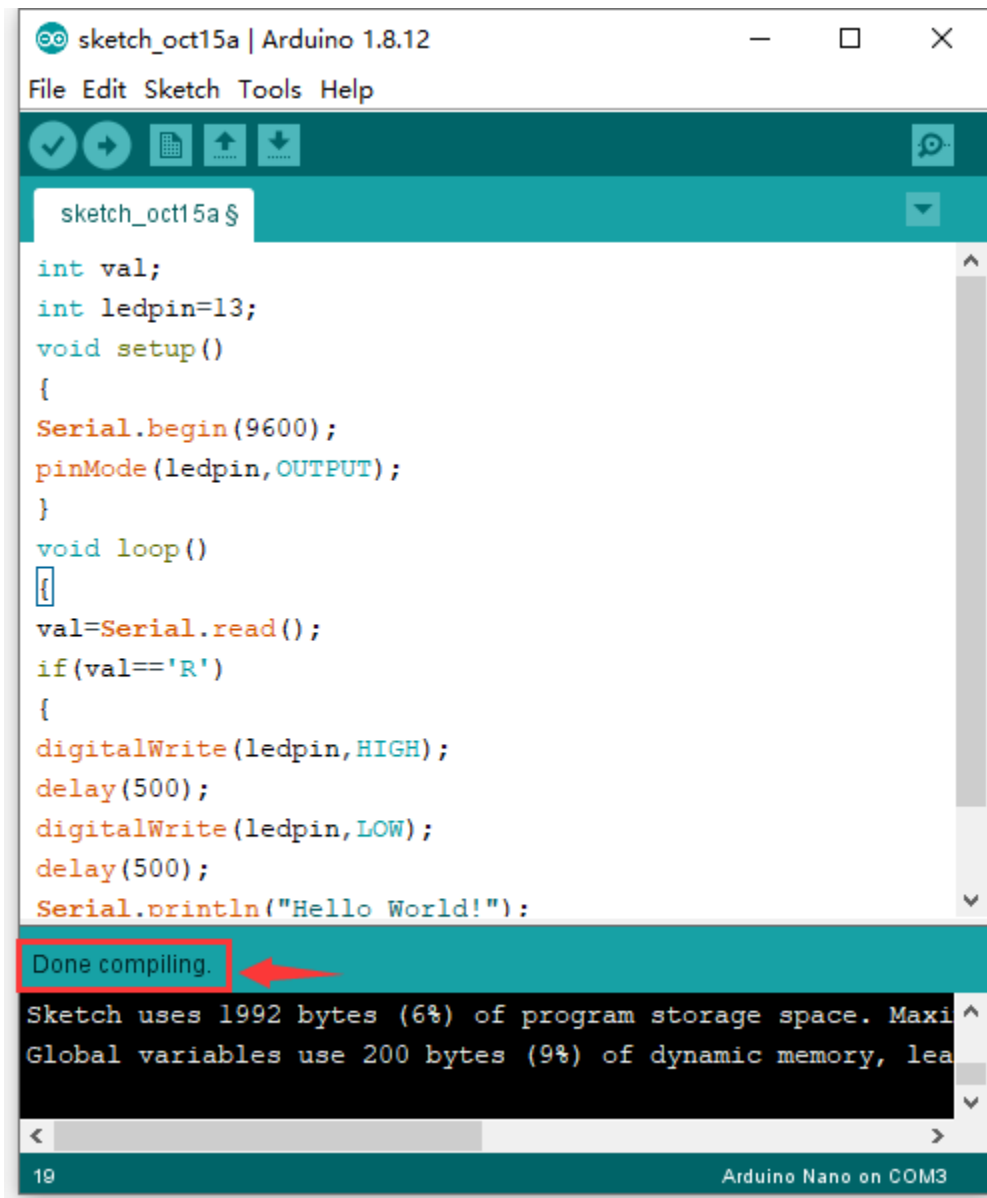
```
int val;  
int ledpin=13;  
void setup()  
{  
  Serial.begin(9600);  
  pinMode(ledpin,OUTPUT);  
}  
void loop()  
{
```

```
val=Serial.read();  
if(val=='R')  
{  
digitalWrite(ledpin,HIGH);  
delay(500);  
digitalWrite(ledpin,LOW);  
delay(500);  
Serial.println("Hello World!");  
}  
}
```

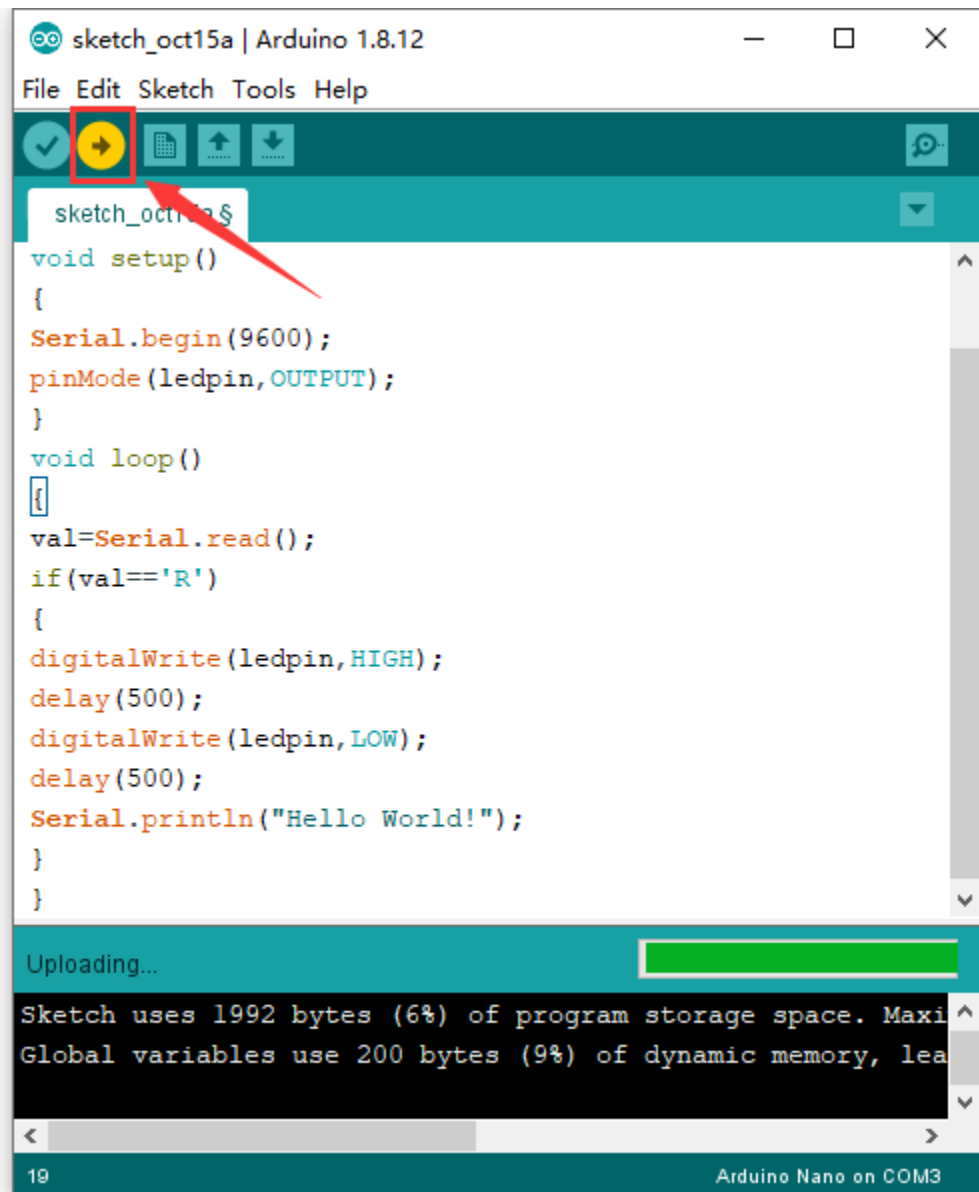


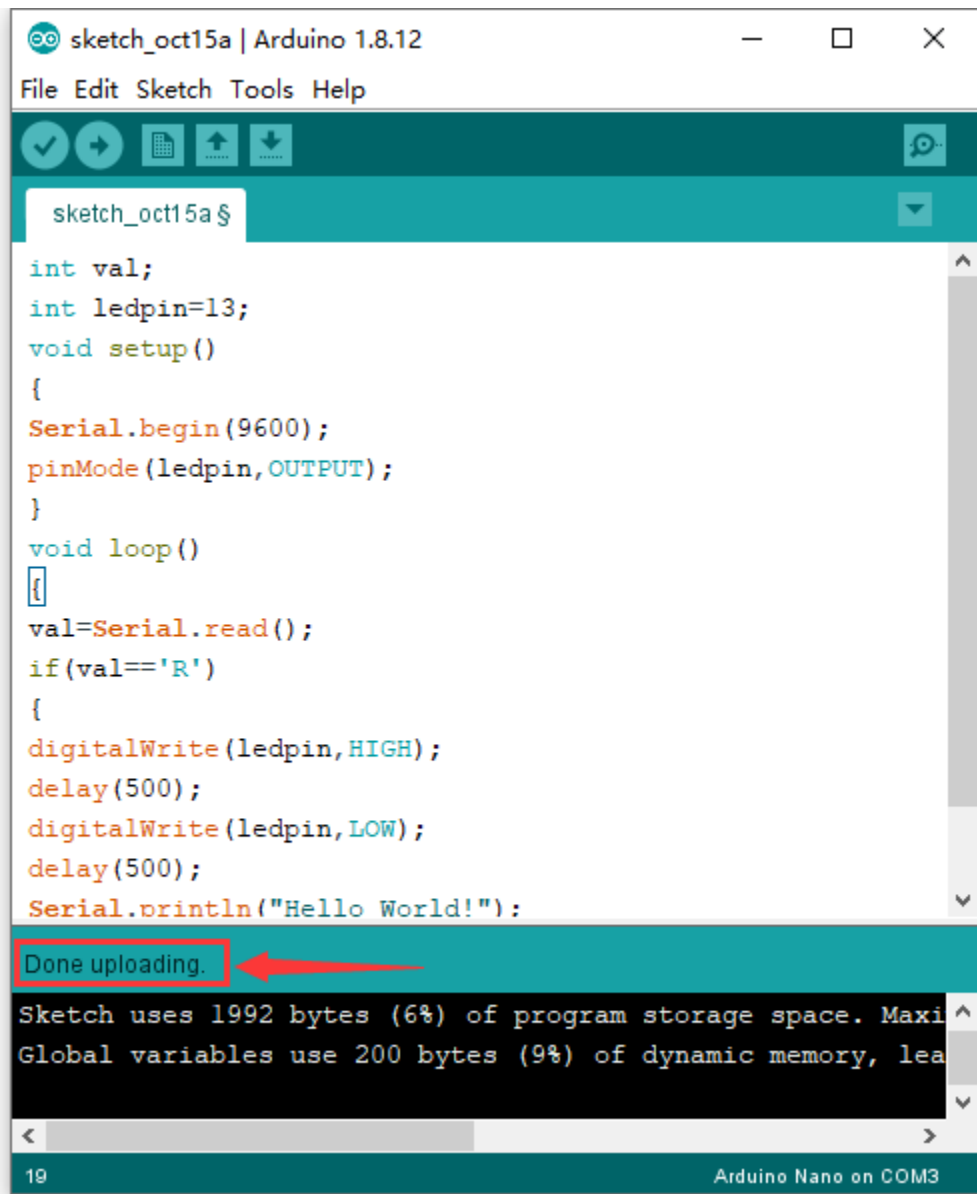



Then click verify button to check the errors. If compiling successfully, the message “Done compiling.” will appear in the status bar.

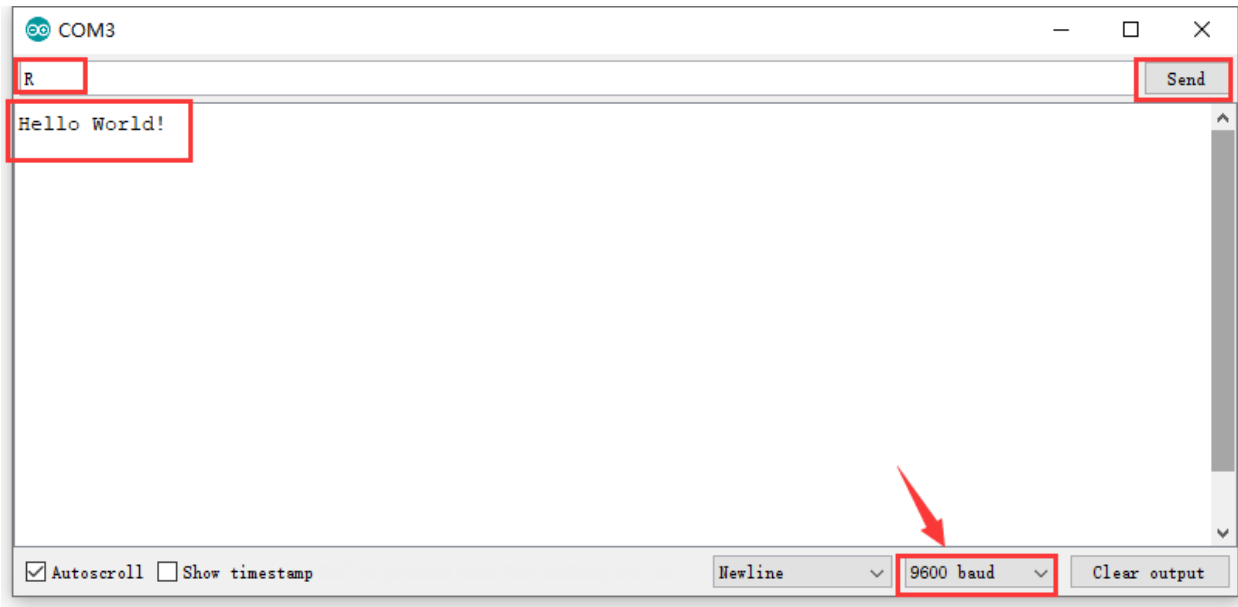


After that, click the “Upload” button to upload the code. If the upload is successful, the message “Done uploading.” will appear in the status bar.





Then click  to open serial monitor and set the baud rate to 9600, enter an “R” and click Send, that is, the computer will send the character R. When NANO board receives it, you should see the RX led on the board flash once, and then D13 led flash once; when keyestudio NANO board sends “Hello World!” to the computer, finally you should see the “Hello World!” is showed on the monitor, and TX led on the board flash once.



MAC System

### 5.1 Install Arduino IDE on MAC System

The installation instruction is as same as the chapter 4.1, as shown below:



### 5.2 Download the CH340 driver

<https://fs.keyestudio.com/CH340-MAC>

### 5.3 How to Install the CH340 driver

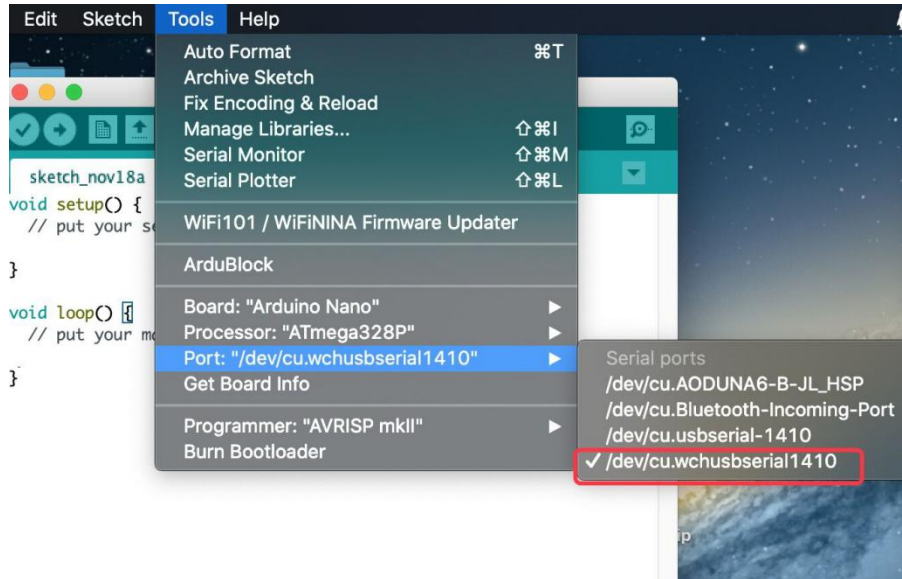


Please refer to the following link:

[https://wiki.keyestudio.com/Download\\_CH340\\_Driver\\_on\\_MAC\\_System](https://wiki.keyestudio.com/Download_CH340_Driver_on_MAC_System)

#### 5.4 Setting Arduino IDE

The setting method is as same as the chapter 4.4 except from COM port, as shown below:



#### Shipping List

Keyestudio NANO ch340\*1pcs

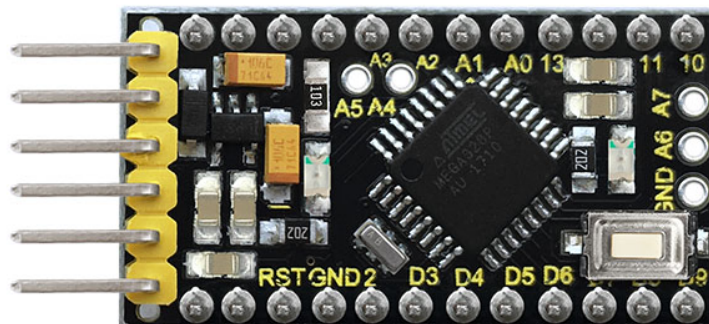
30cm Blue mini USB\*1pcs

Resource:

<https://fs.keyestudio.com/KS0173>



## KS0247 KEYESTUDIO PRO MINI DEVELOPMENT BOARD



## Introduction:

keyestudio Pro Mini is a small microcontroller board based on the ATmega328. ([datasheet](#)), intended for use on breadboards and when space is at a premium.

It has 14 digital input/output pins (of which 6 can be used as PWM outputs), 8 analog inputs, a 16 MHz crystal oscillator, and a reset button.

Note that the board is without an on-board USB to Serial connection. To program the Pro Mini, you will need an external USB to Serial module.

The keyestudio pro mini can be powered via the pins 5V GND (DC 5V), or female headers Vin GND (DC 7-9V).

Features:

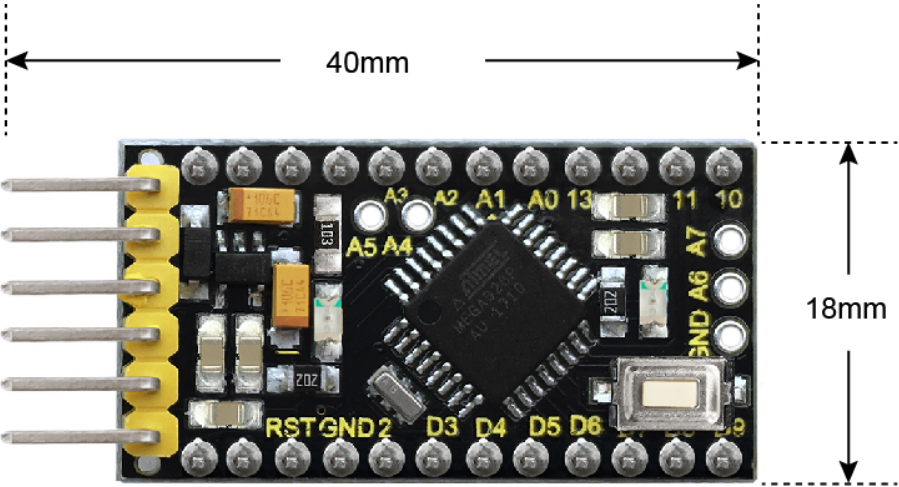
- Using ATmega328P-AU from Atmel
- 14 digital input/output pins
- 8 analog input pins labeled A0 ~ A7
- TTL level serial pins RX/TX;
- 6 PWM pins (D3, D5, D6, D9, D10, D11)
- Support external DC 7-9V power supply

TECH SPECS:

<b>Microcontroller</b>	ATmega328P-AU
<b>Operating Voltage</b>	5V
<b>Input Voltage (recommended)</b>	DC7-9V
<b>Digital I/O Pins</b>	14 (D0-D13) (of which 6 provide PWM output)
<b>PWM Digital I/O Pins</b>	6 (D3, D5, D6, D9, D10, D11)
<b>Analog Input Pins</b>	8 (A0-A7)
<b>DC Current per I/O Pin</b>	40 mA
<b>Flash Memory</b>	32 KB of which 2 KB used by bootloader
<b>SRAM</b>	2 KB
<b>EEPROM</b>	1 KB
<b>Clock Speed</b>	16 MHz
<b>LED_BUILTIN</b>	D13

Technical Details:

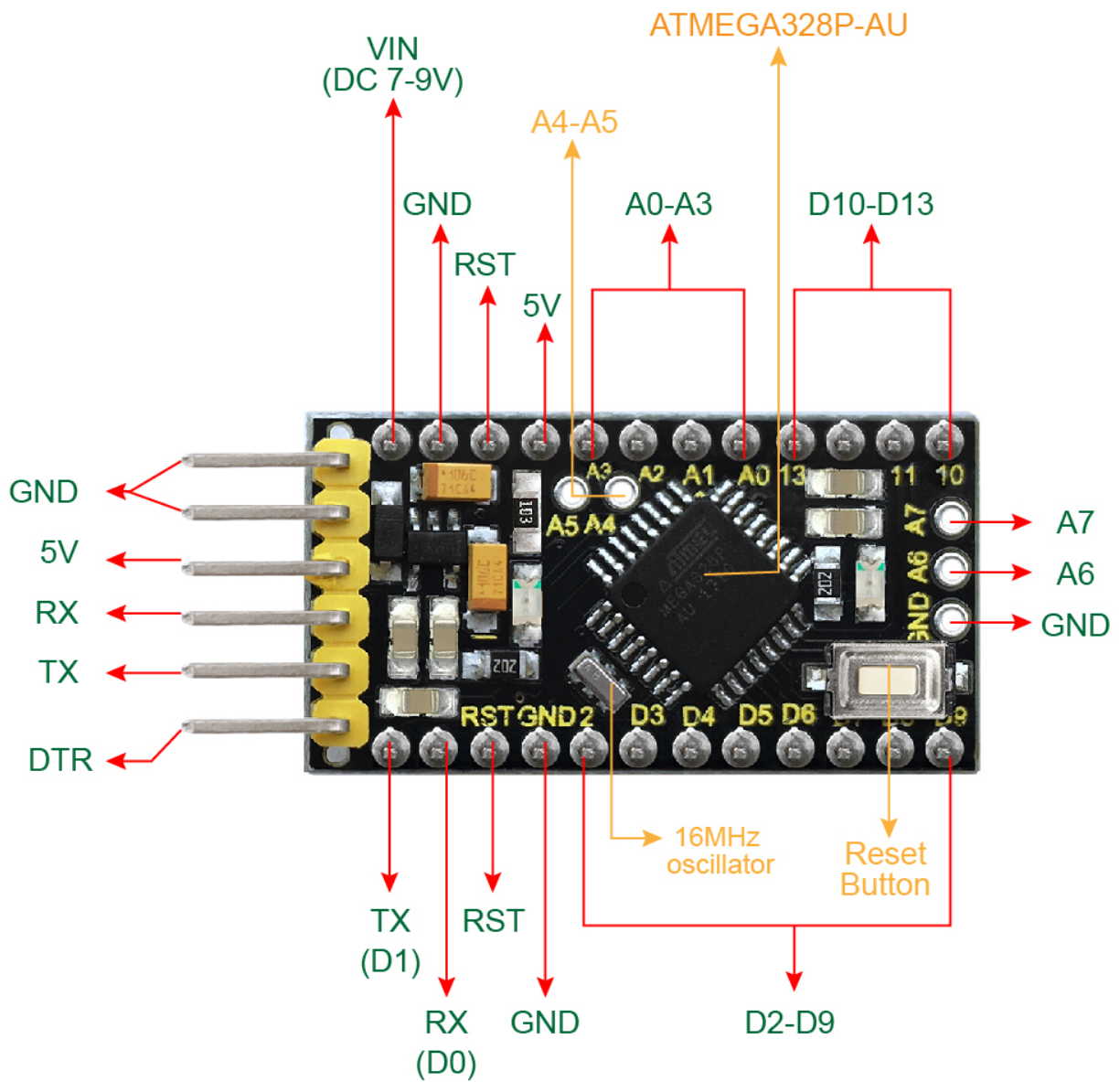
- Dimensions: 40mm x 18mm x 14mm
- Weight: 4.1g

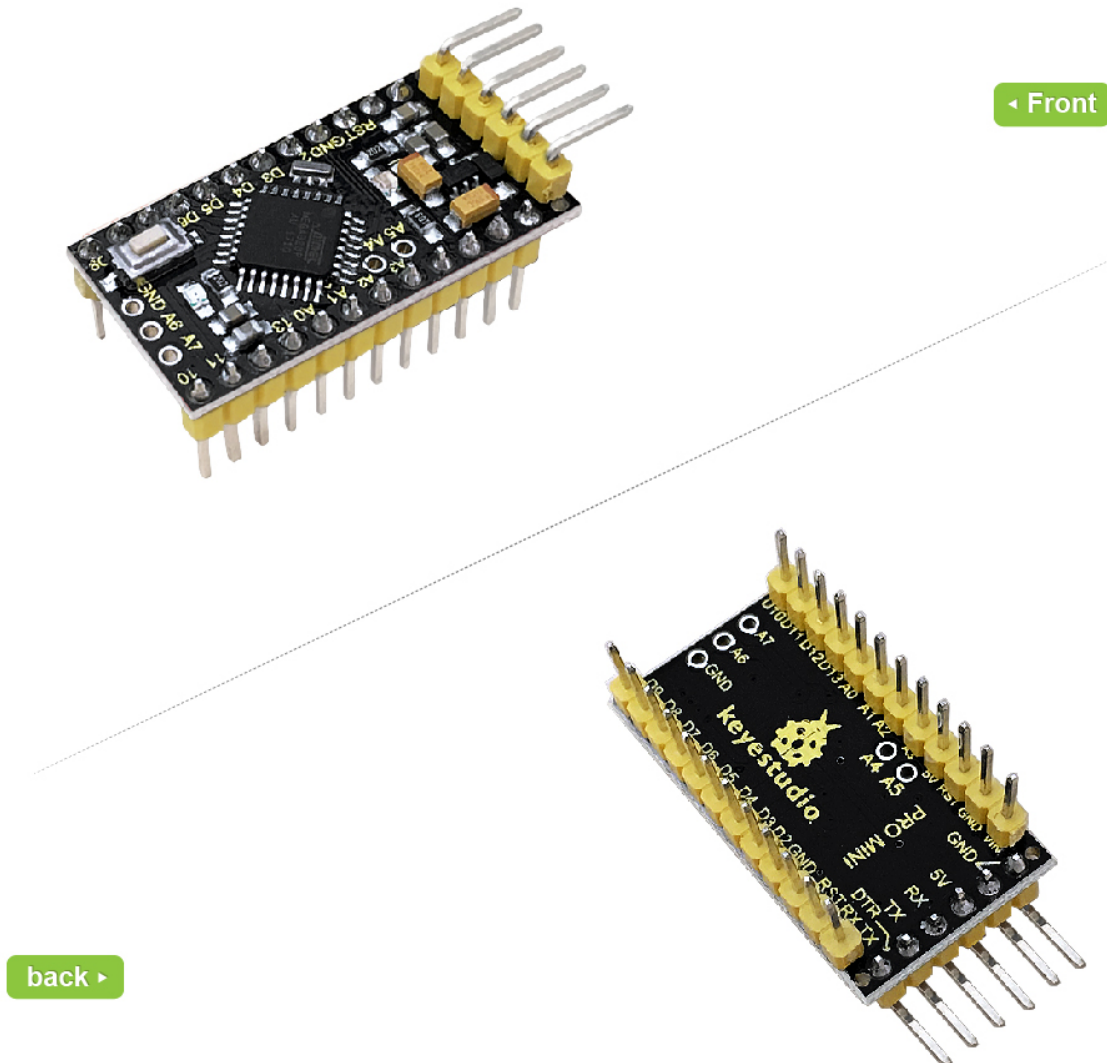


H:14mm  
only board weight :4.1g

### Element and Interfaces:

Here is an explanation of what every element and interface of the board does:





Specialized Functions of Some Pins:

- **Serial communication:** 0 (RX) and 1 (TX). Used to receive (RX) and transmit (TX) TTL serial data.
- **PWM (Pulse-Width Modulation):** D3, D5, D6, D9, D10, D11
- **External Interrupts:** D2 (interrupt 0) and D3 (interrupt 1).
- **SPI communication:** D10 (SS), D11 (MOSI), D12 (MISO), D13 (SCK).
- **IIC communication:** A4 (SDA); A5(SCL)

Detailed Using Methods are as follows:

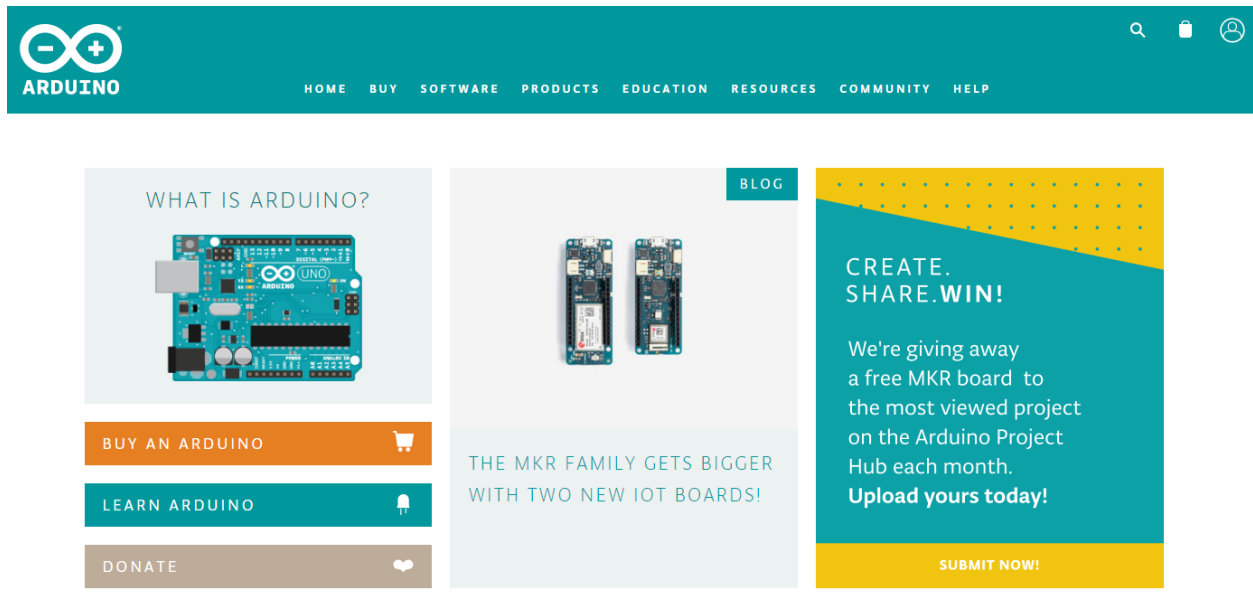
Step1| Download the Arduino IDE

When you get the board, first you should install the Arduino software and driver.

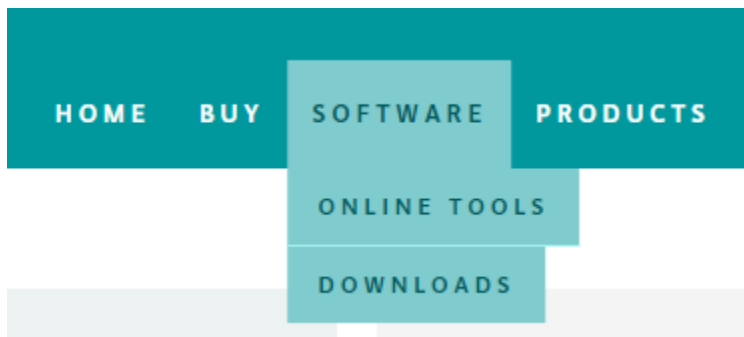
We usually use the Windows software Arduino 1.5.6 version. You can download it from the link below:

<https://www.arduino.cc/en/Main/OldSoftwareReleases#1.5.x>

Or you can browse the ARDUINO website to download the latest version from this link, <https://www.arduino.cc>, pop up the following interface.




Then click the **SOFTWARE** on the browse bar, you will have two options **ONLINE TOOLS** and **DOWNLOADS**.



Click **DOWNLOADS**, it will appear the latest software version of ARDUINO 1.8.5 shown as below.



## Download the Arduino IDE




**ARDUINO 1.8.5**

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** 10.7 Lion or newer

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

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

In this software page, on the right side you can see the version of development software for different operating systems. ARDUINO has a powerful compatibility. You should download the software that is compatible with the operating system of your computer.

We will take **WINDOWS system** as an example here. There are also two options under Windows system, one is installed version, the other is non-installed version.

For simple installed version, first click **Windows Installer**, you will get the following page.

**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** 10.7 Lion or newer

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

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

## Contribute to the Arduino Software

Consider supporting the Arduino Software by contributing to its development. (US tax payers, please note this contribution is not tax deductible). [Learn more on how your contribution will be used.](#)



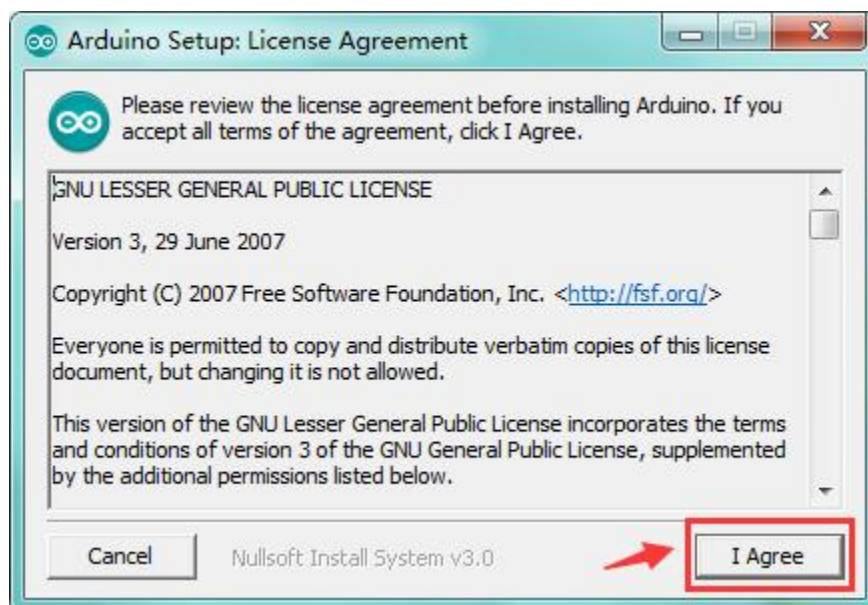
This way you just need to click JUST DOWNLOAD, then click the downloaded file to install it.

For non-installed version, first click Windows ZIP file, you will also get the pop-up interface as the above figure.

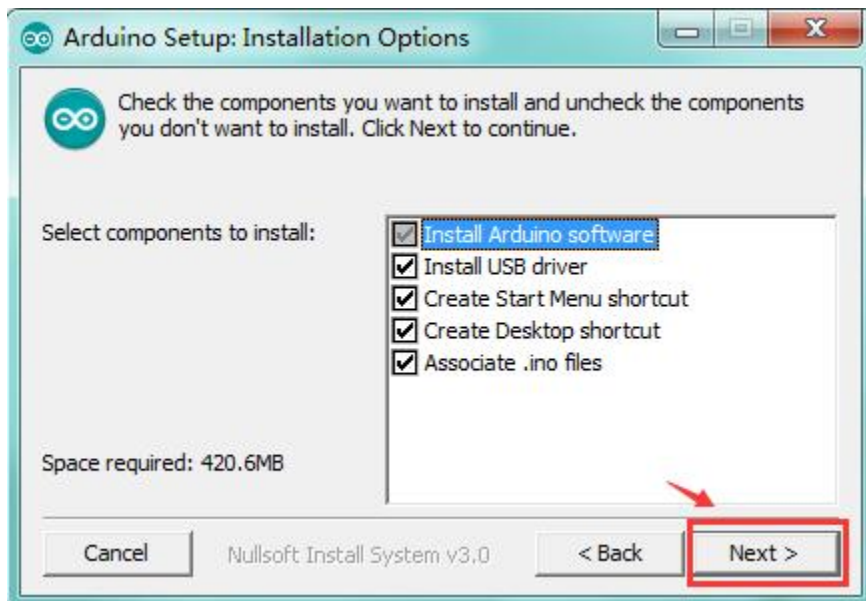
Click JUST DOWNLOAD, and when the ZIP file is downloaded well to your computer, you can directly unzip the file and click the icon of ARDUINO software to start it.

Installing Arduino (Windows):

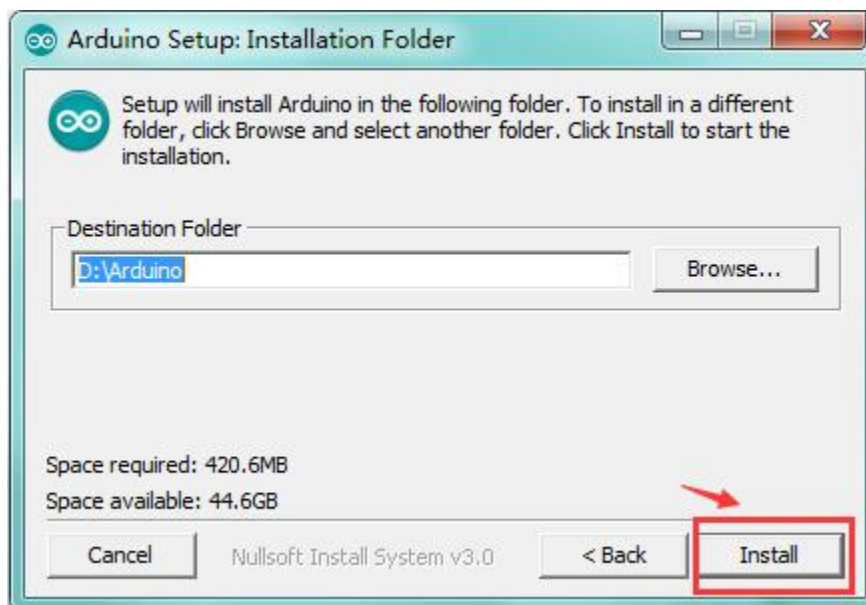
Install Arduino with the exe. Installation package downloaded well.



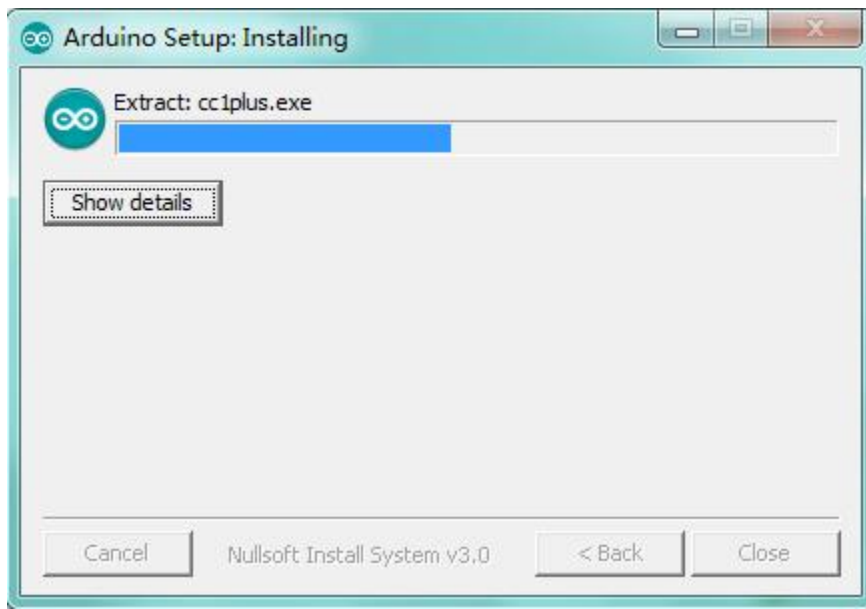
Click "I Agree" to see the following interface.



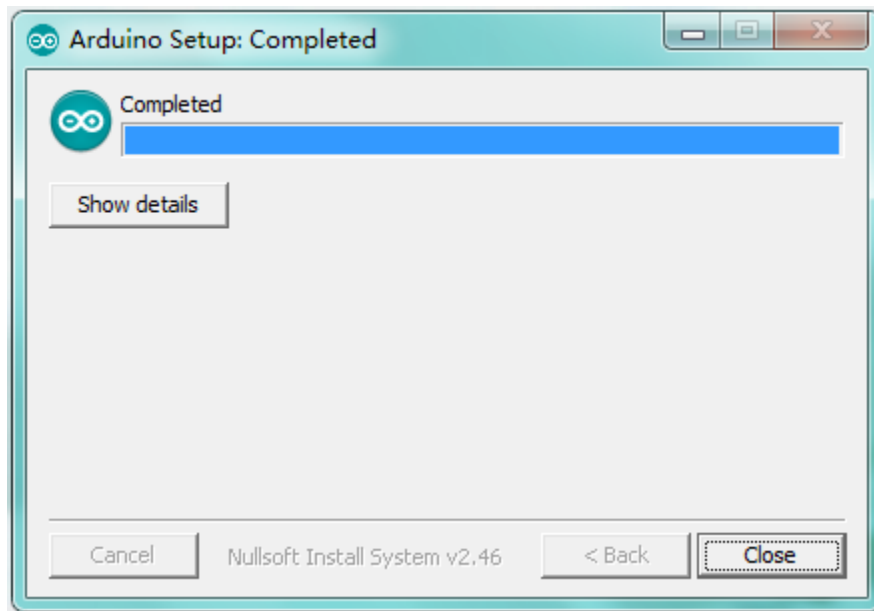
Click “Next”. Pop up the interface below.



You can press Browse... to choose an installation path or directly type in the directory you want. Then click “Install” to initiate installation.



Wait for the installing process, if appear the interface of Window Security, just continue to click Install to finish the installation.

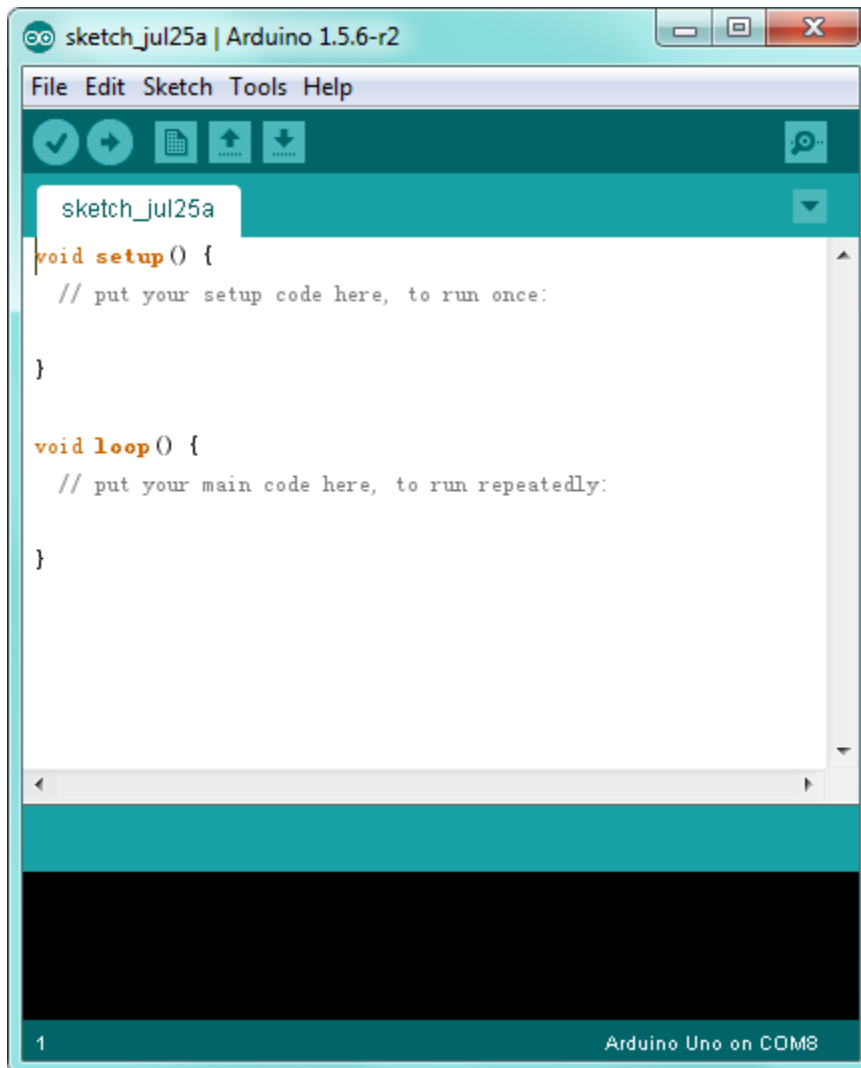


Introduction for Arduino IDE Toolbar:

Double-click the icon of Arduino software downloaded, you will get the interface shown below.

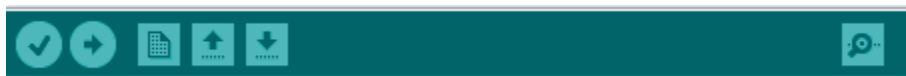








(Note: if the Arduino software loads in the wrong language, you can change it in the preferences dialog. See [the environment page](#) for details.)



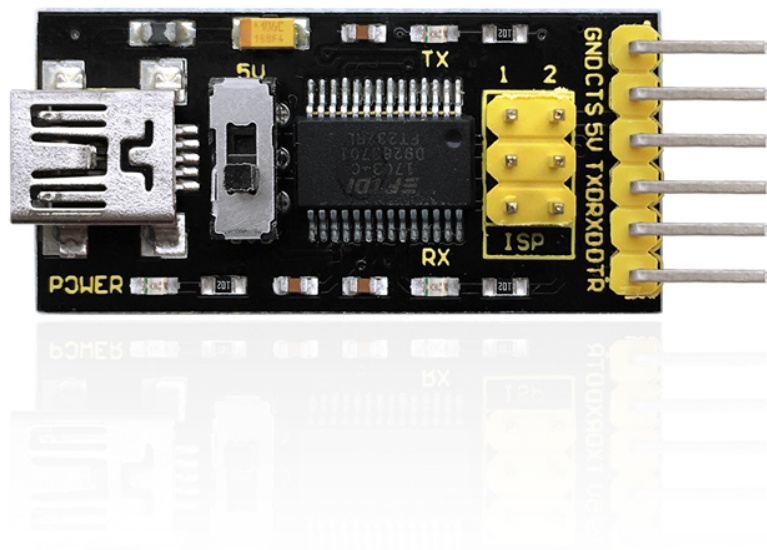
The functions of each button on the Toolbar are listed below:

<http://wiki.keyestudio.com/index.php/File:IDE.png>



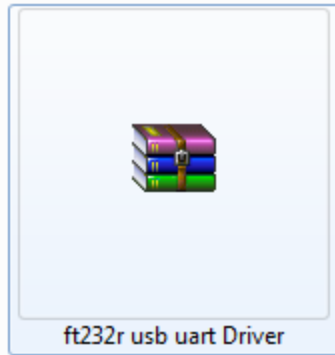
	<b>Verify/Compile</b>	Check the code for errors
	<b>Upload</b>	Upload the current Sketch to the Arduino
	<b>New</b>	Create a new blank Sketch
	<b>Open</b>	Show a list of Sketches
	<b>Save</b>	Save the current Sketch
	<b>Serial Monitor</b>	Display the serial data being sent from the Arduino

Step2| Connect a USB to Serial Module



To program the Pro Mini board, you will need to connect an external USB to Serial module. Different USB to Serial modules might have different driver installation. In the following, we will introduce the use of pro mini with our [keyestudio FTDI Basic Program Downloader](#). See more details [here](#).

Download the Driver Files



You can download the driver file from the link below:

<https://drive.google.com/open?id=1vj4Is2LokXf3GDGcnumnAPmaI39uPhQk>

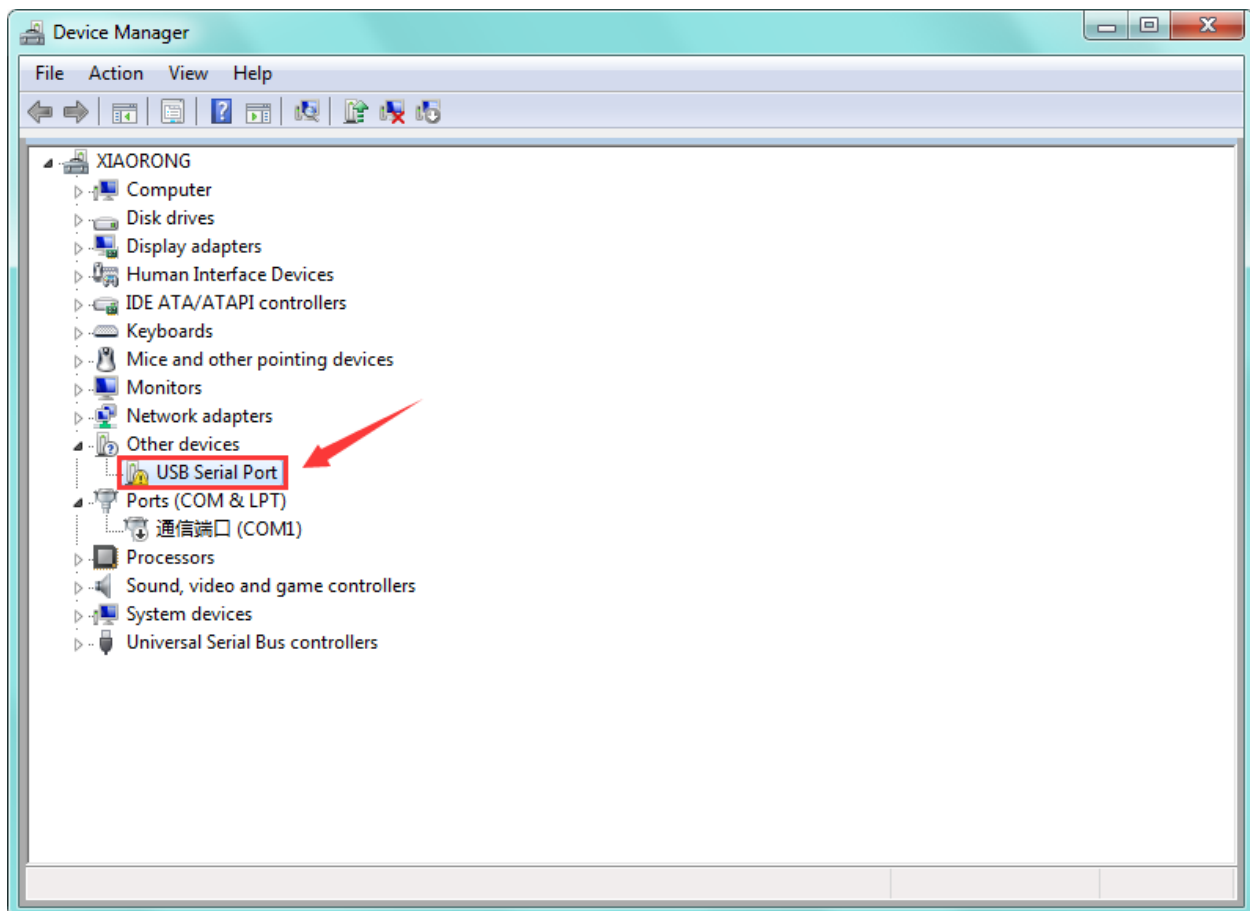
### Installing the Driver

The USB to serial port chip of this module is FT232RL. So you need to install the [ft232r usb uart driver](#). You could click the driver file above to download it.

In different systems, the driver installation might be similar. Here we start to install the driver on the Win7 system.

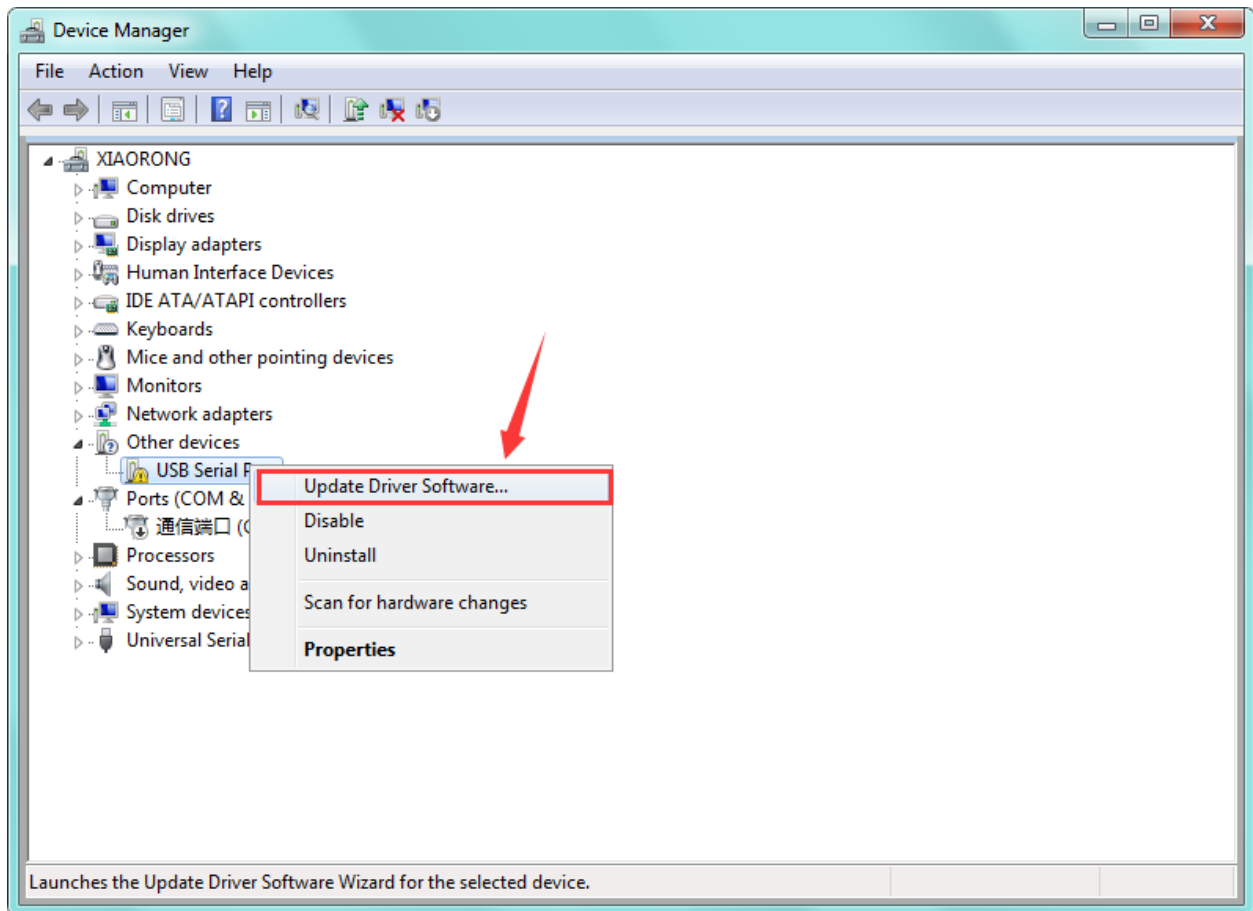
Plug one end of your USB cable into the module and the other into a USB socket on your computer.

Then right click your “Computer” —>for “Properties”—> click “Device manager”, under Other devices, you should see the “USB Serial Port”.



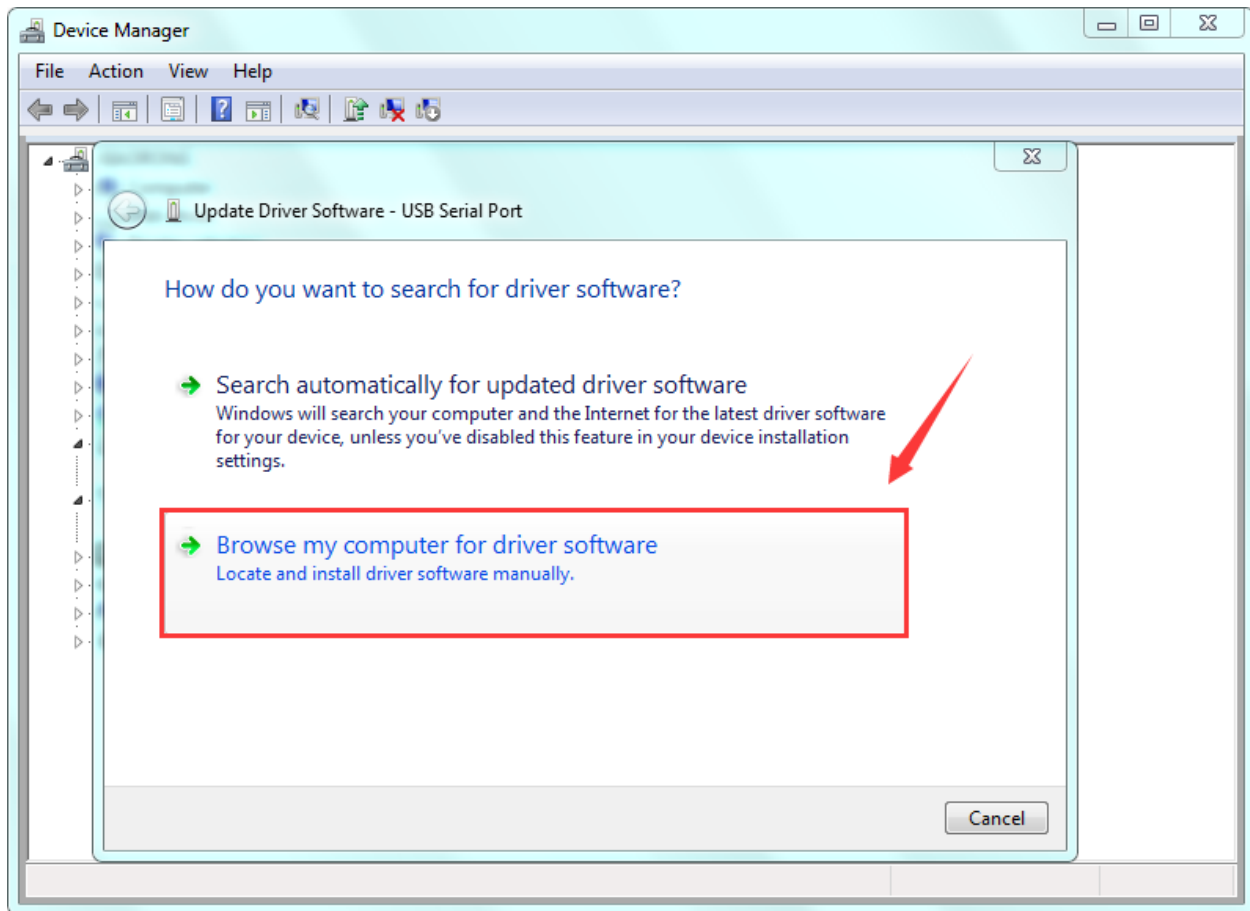


Then right-click on the USB Serial Port and select the top menu option (Update Driver Software...) shown as the figure below.

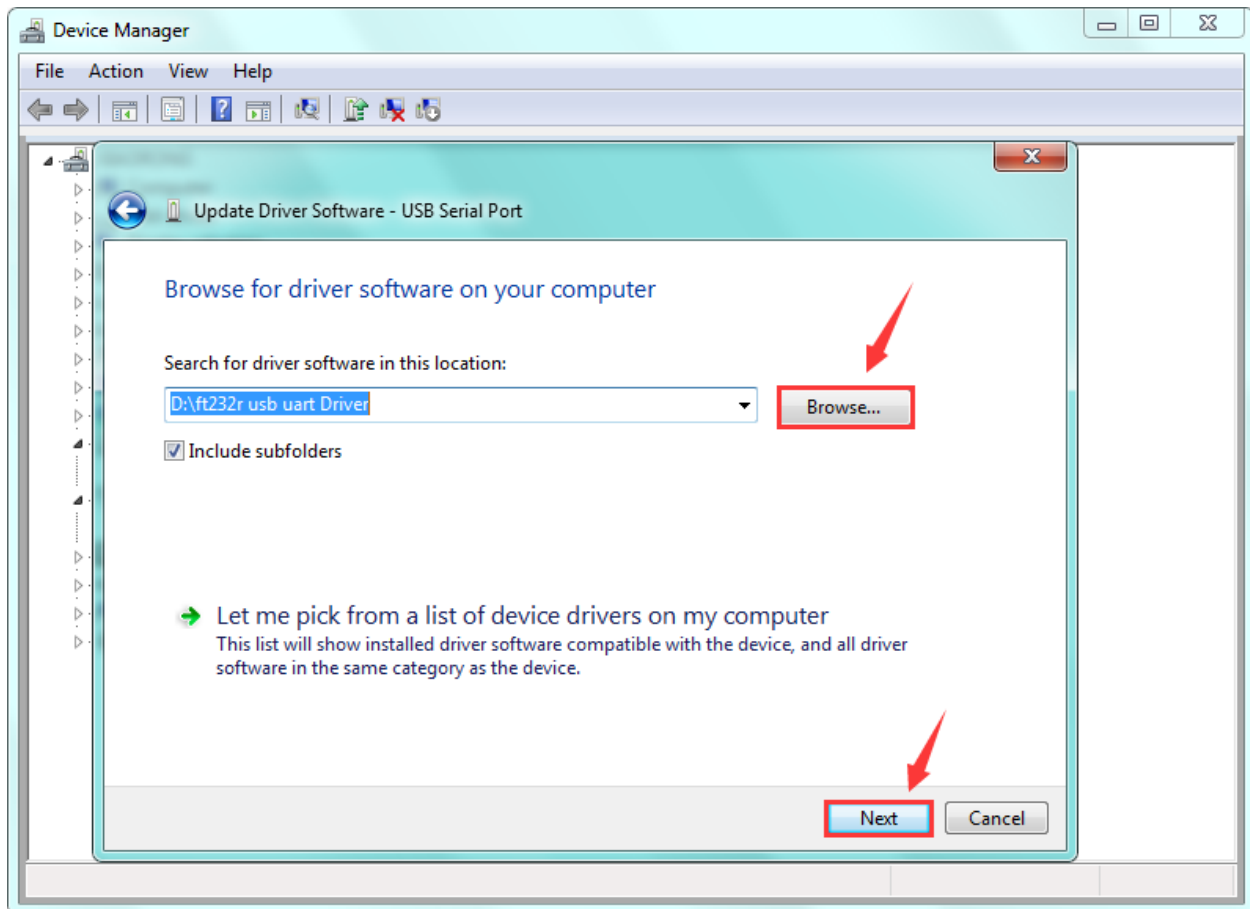


[http://wiki.keyestudio.com/index.php/File:Driver\\_2.png](http://wiki.keyestudio.com/index.php/File:Driver_2.png)

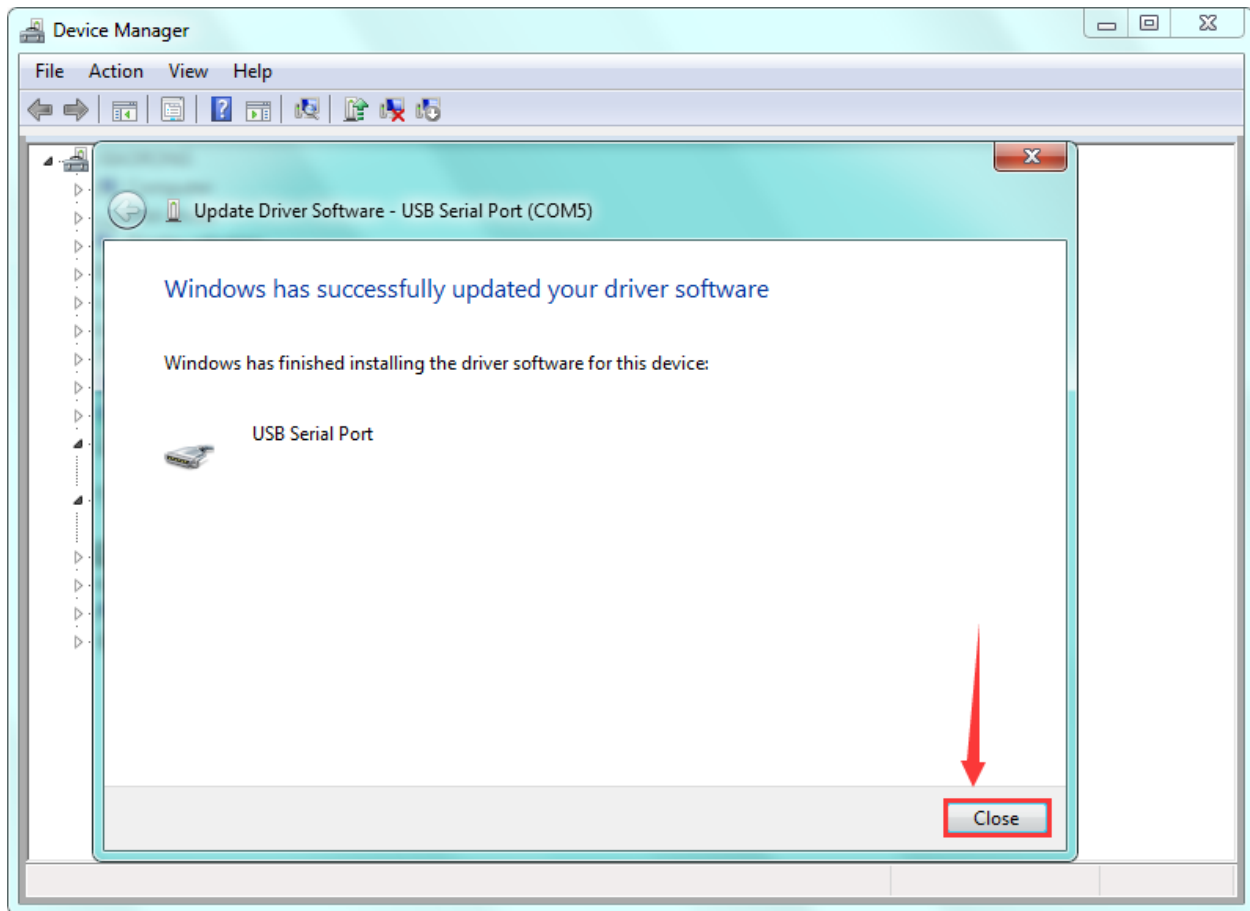
Then it will be prompted to either "Search Automatically for updated driver software" or "Browse my computer for driver software". Shown as below. In this page, select "Browse my computer for driver software".



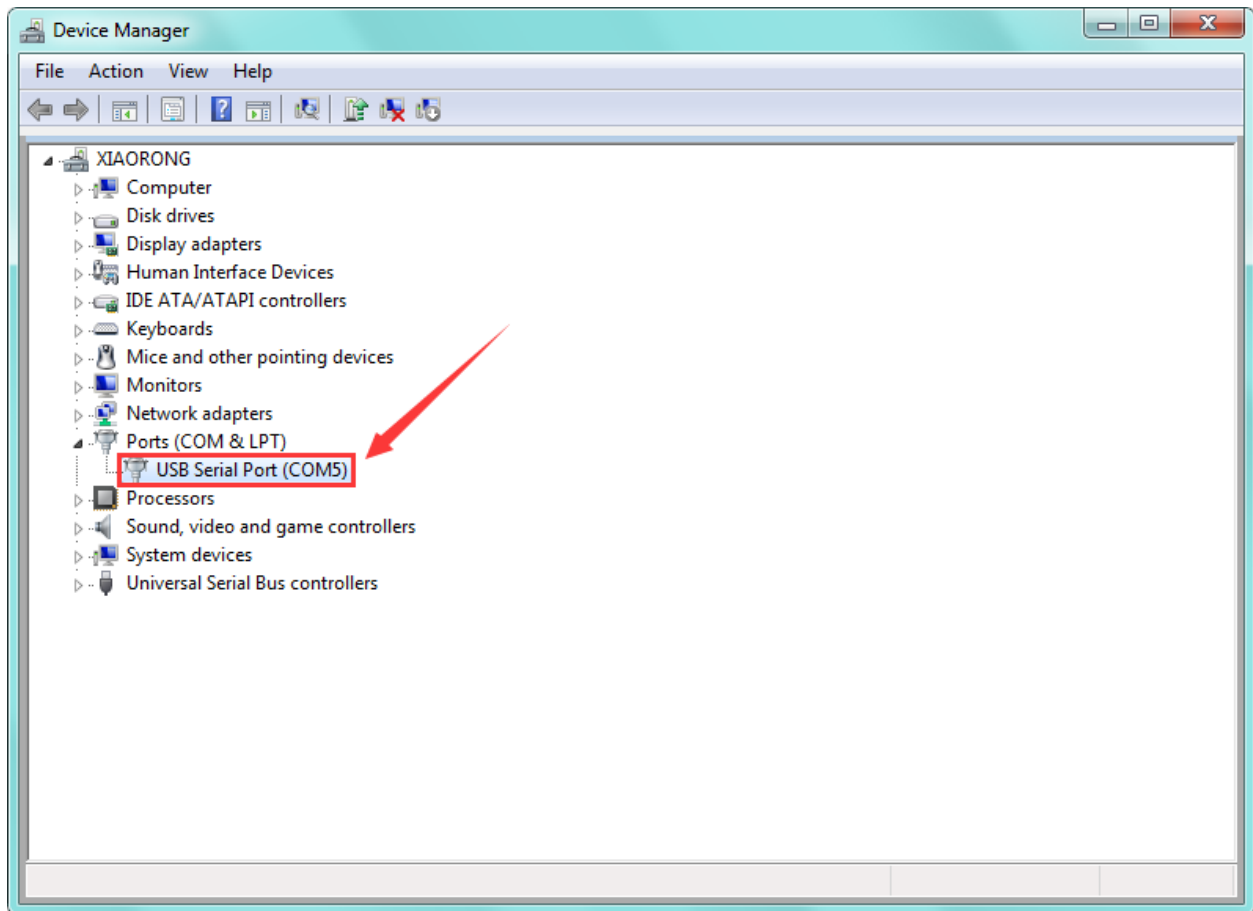
After that, select the option to browse and navigate to the “drivers” folder.



Once the driver has been installed, you will get a confirmation message. Installation completed, click “Close”. [http://wiki.keyestudio.com/index.php/File:Driver\\_6.png](http://wiki.keyestudio.com/index.php/File:Driver_6.png)

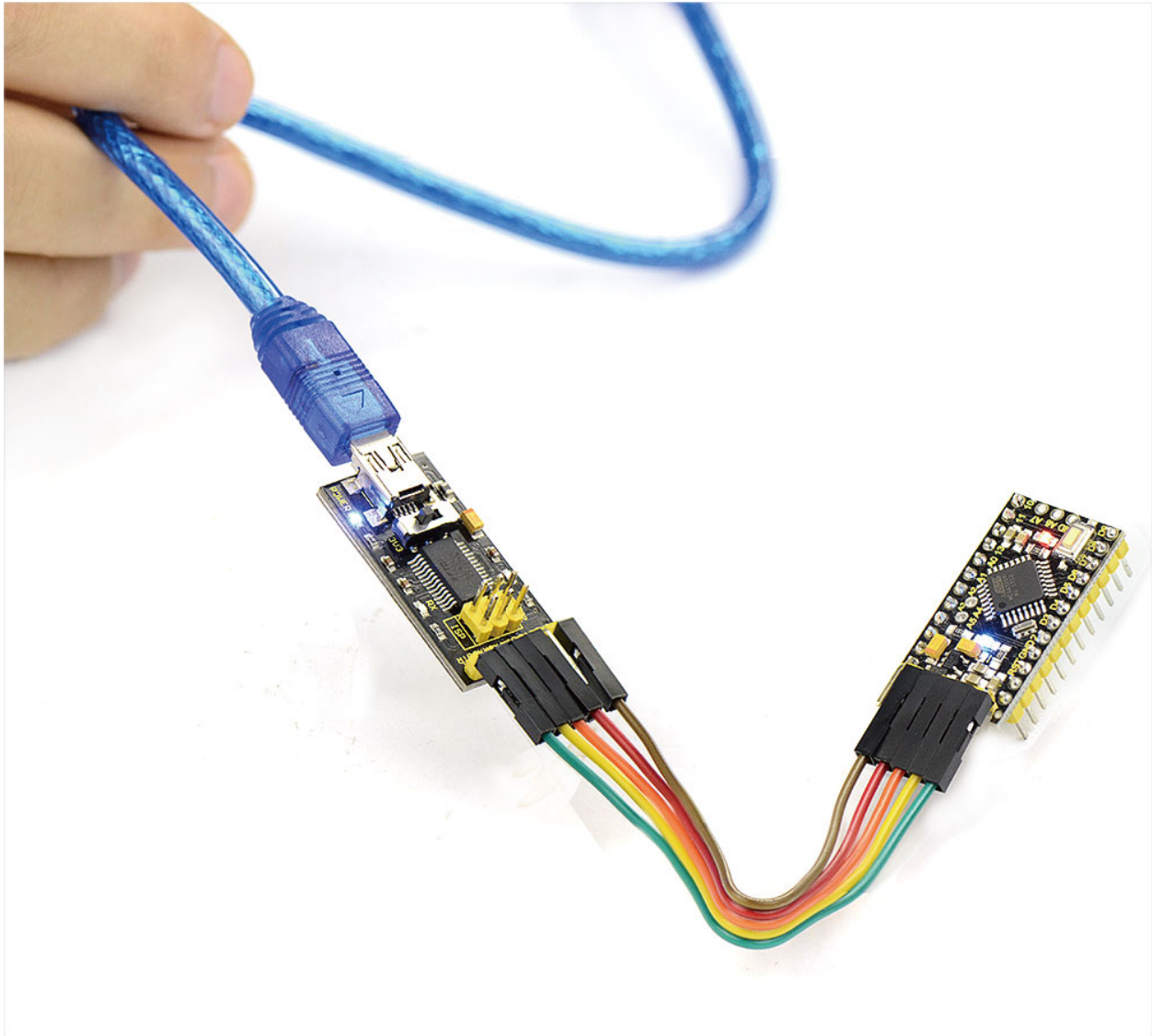


Up to now, the driver is installed well. Then you can right click "Computer" —>"Properties"—>"Device manager", you should see the device as the figure shown below.



### Step3| Hookup Guide

Driver installation completed, disconnect the module to computer, then connect it to pro mini board using F-F jumper wires. Shown below. The slide switch on the module is used to control the voltage for serial communication. Whether it is set to 5V or 3.3V, it can be programmed with the Pro Mini.



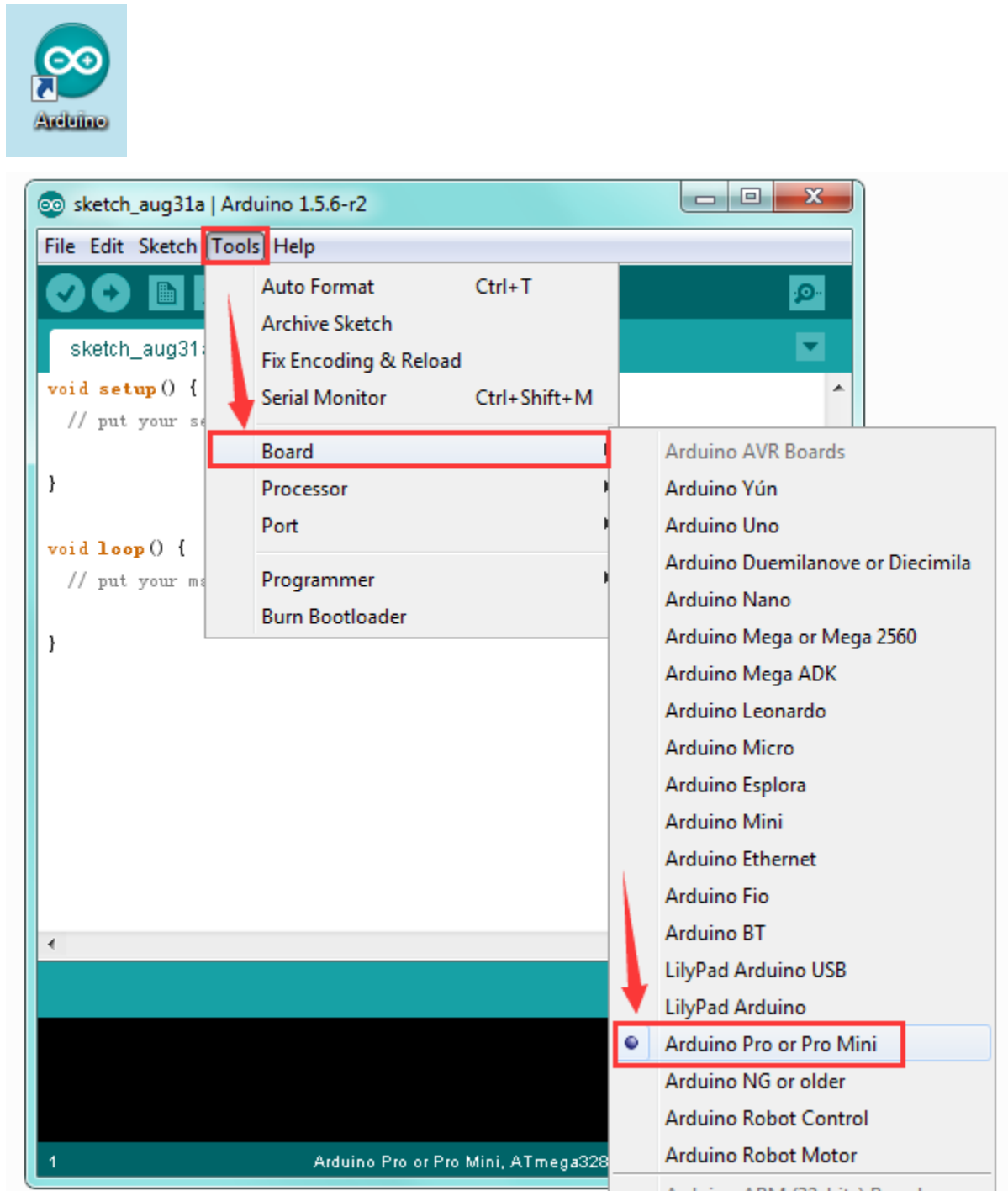
programming for KEYES PRO MINI control board by using FT232 module.

FTDI Basic Program Downloader	keyestudio pro mini
GND	GND
5V	5V
TXD	RX
RXD	TX
DIR	DIR

Connect well the module to pro mini board, then connect the pro mini board to your computer with a mini USB cable.

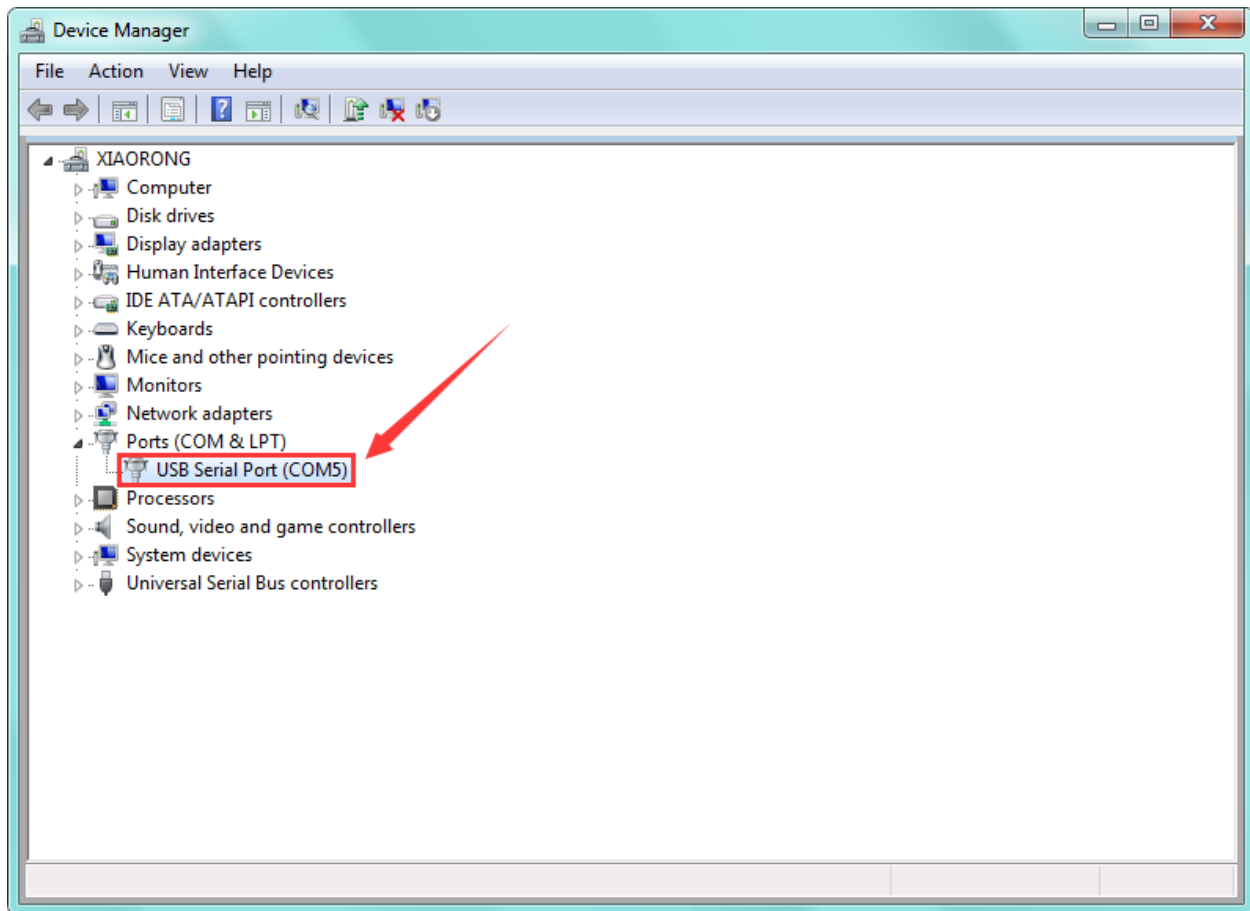
Step4| Select the Arduino Board

Open the Arduino IDE, you'll need to click the "Tools", then select the Board that corresponds to your Arduino.

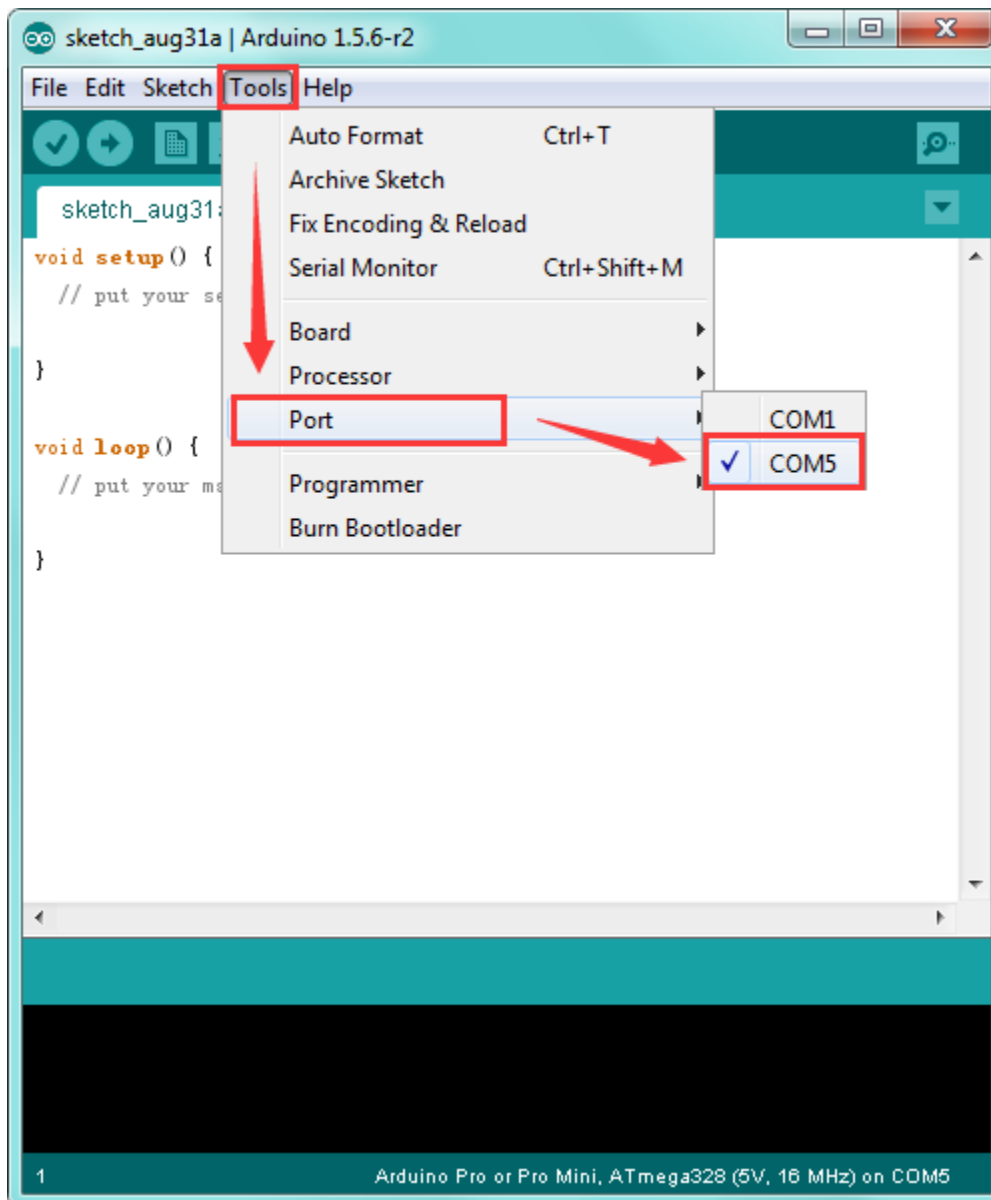


Step5| Select your serial port

Select the serial device of the Arduino board from the **Tools | Serial Port** menu. **Note:** to avoid errors, the COM Port should keep the same as the Ports shown on Device Manager.







Step6| Upload the Code

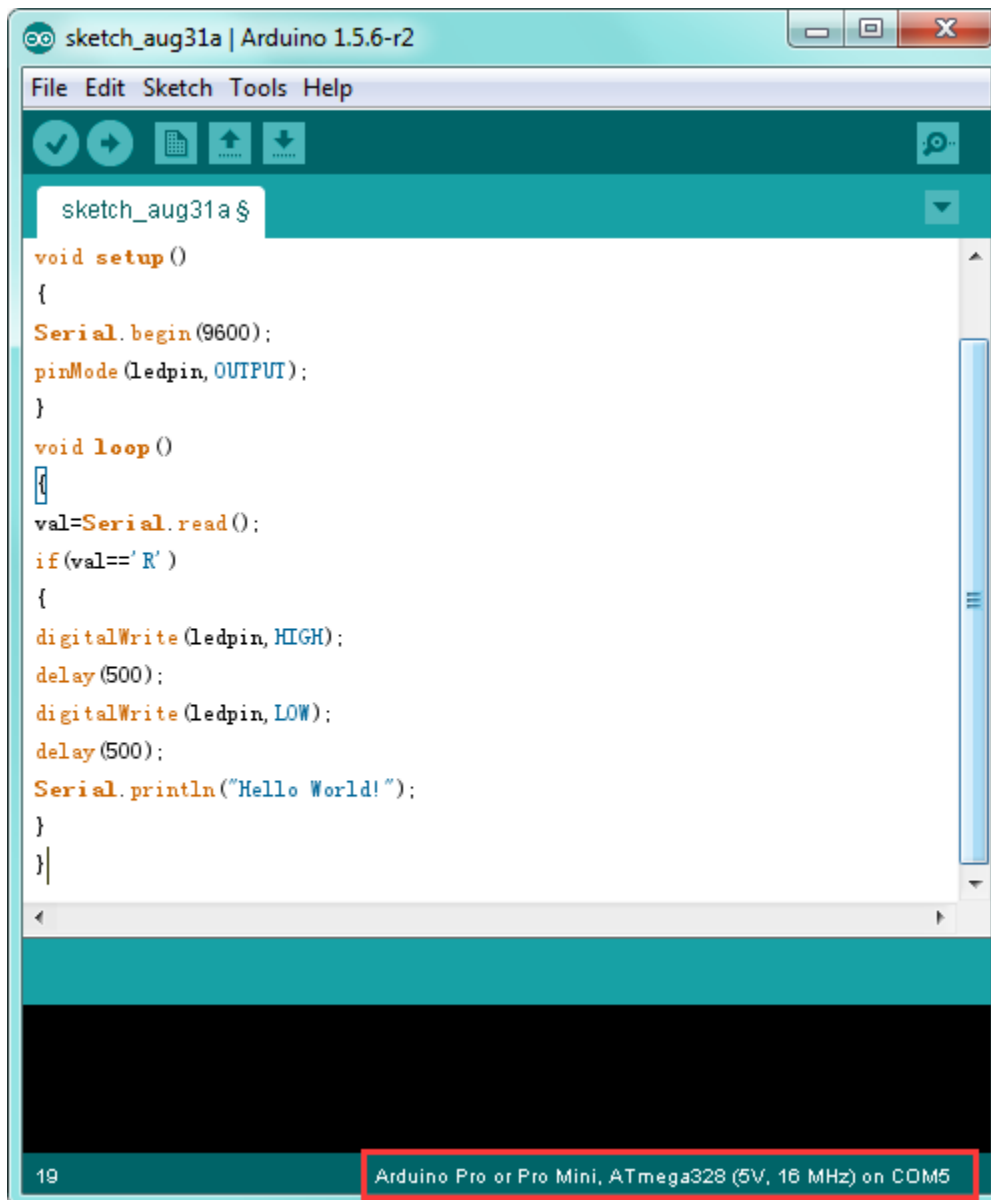
Below is an example code for displaying the Hello World!

Copy and paste the code to the Arduino environment IDE.

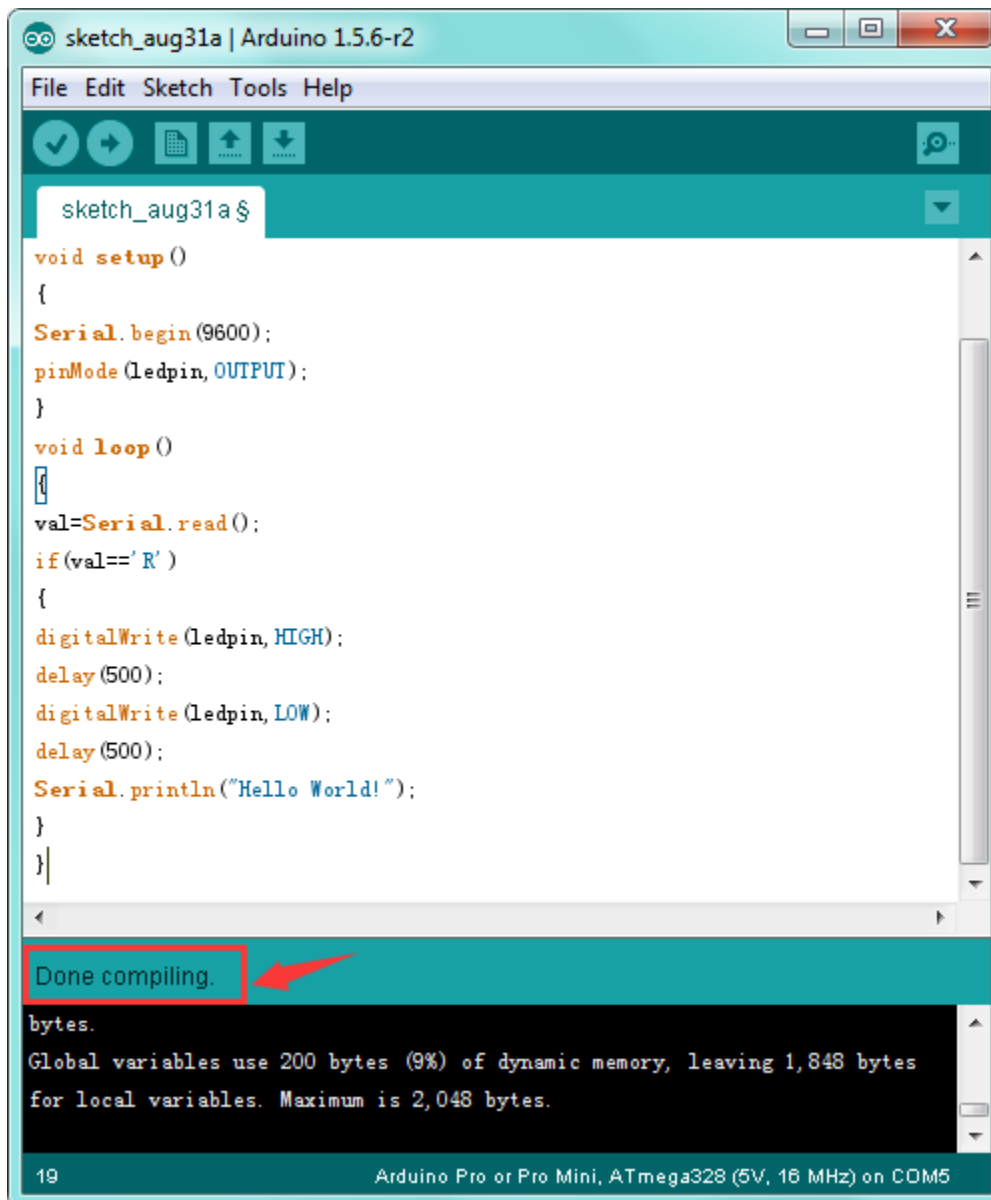
////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////

```
int val;  
int ledpin=13;  
void setup()  
{  
  Serial.begin(9600);  
  pinMode(ledpin,OUTPUT);  
}
```

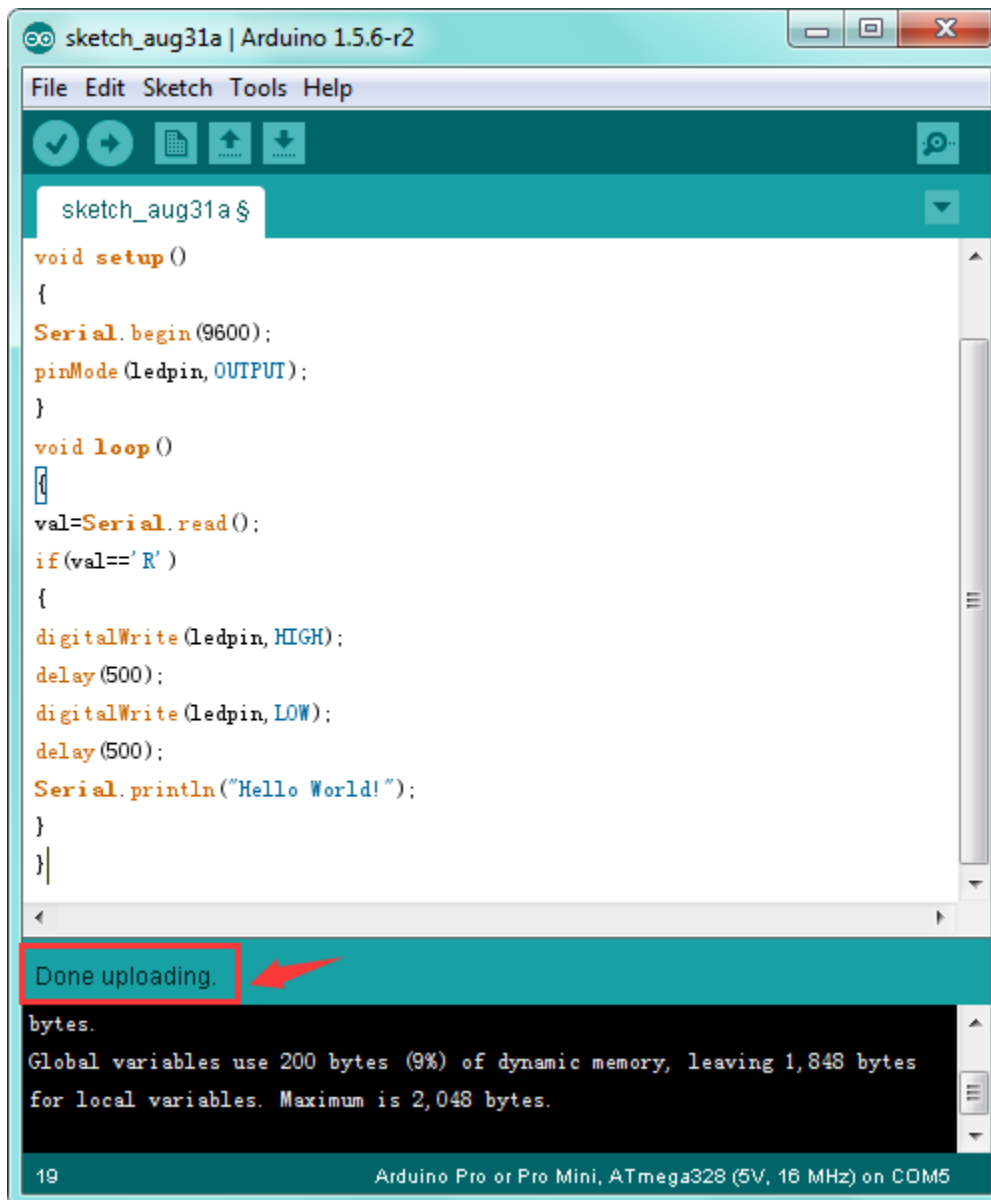
```
void loop()
{
  val=Serial.read();
  if(val=='R')
  {
    digitalWrite(ledpin,HIGH);
    delay(500);
    digitalWrite(ledpin,LOW);
    delay(500);
    Serial.println("Hello World!");
  }
}
////////////////////////////////////////////////////////////////
```



Then click verify button to check the errors. If compiling successfully, the message “Done compiling.” will appear in the status bar.

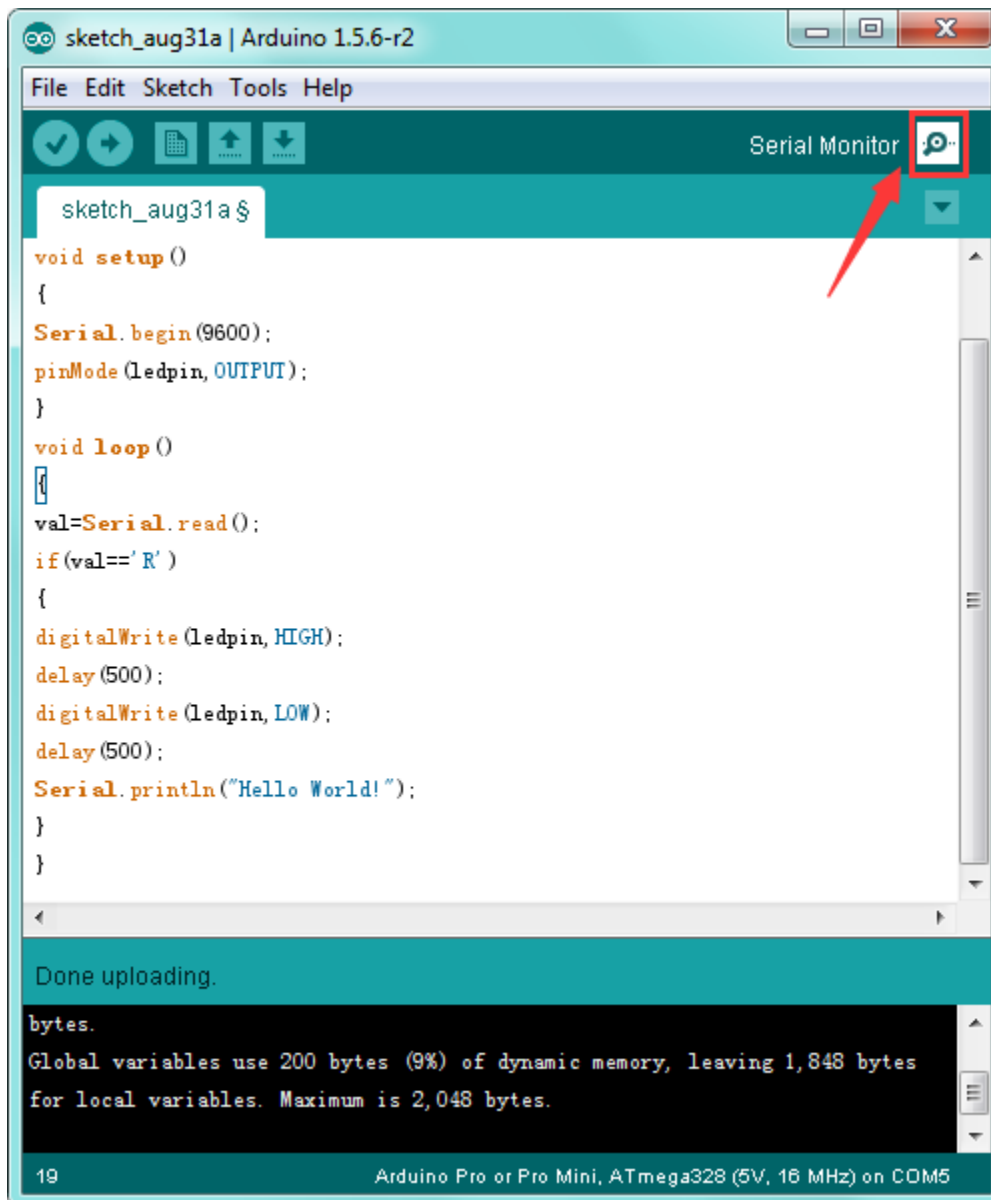


After that, click the “Upload” button to upload the code. If the upload is successful, the message “Done uploading.” will appear in the status bar.

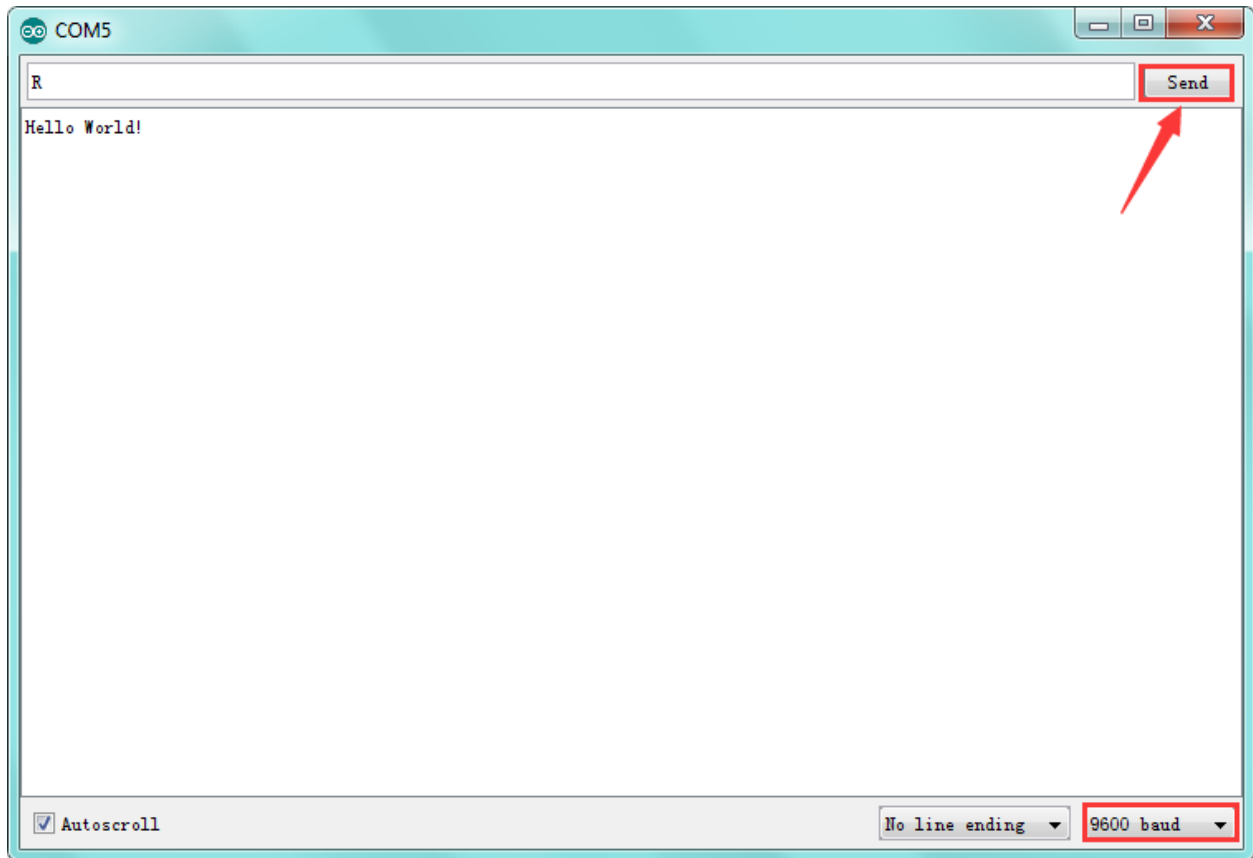


Step7| Open the Serial Monitor

After that, click the button to open the serial monitor.



Then set the baud rate to 9600, enter an “R” and click Send, that is, the computer will send the character R. When pro mini board receives it, you should see the D13 led flash once. “Hello World!” is sent to the computer, so that you should see the “Hello World!” is showed on the monitor.



Resource Links:

Download code and libraries:

<https://fs.keyestudio.com/KS0247>

You might also want to look at:

[the reference](#) for the Arduino language;

Download ARDUINO Software:

<https://www.arduino.cc/en/Main/OldSoftwareReleases#1.5.x>

Troubleshooting:

If you have problems, please see the [troubleshooting suggestions](#).





## **KS0248 KEYESTUDIO LEONARDO R3 DEVELOPMENT BOARD**

### Introduction:

The keyestudio Leonardo is a microcontroller board based on the ATmega32u4 ([datasheet](#)). It is an easy-to-use open source hardware.

It has 20 digital input/output pins (of which 7 can be used as PWM outputs), 12 analog inputs, a 16 MHz crystal oscillator, a micro USB connection, a power jack, an ICSP header, and a reset button.

It contains everything needed to support the microcontroller; simply connect it to a computer with a USB cable or power it with a AC-to-DC adapter or battery to get started.

Note that ICSP (In-Circuit Serial Programming) header can not only program the firmware to Atmega32u4, but also be used as SPI communication interface.

The keyestudio Leonardo can be powered via the micro USB connection, or via an external power supply jack (DC 7-12V) or even with female headers Vin /GND (DC 7-12V).

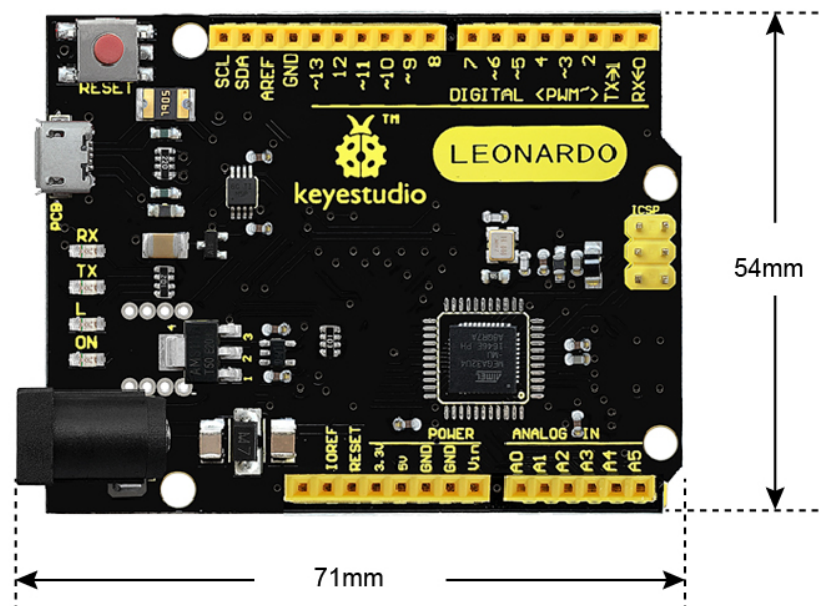
The Leonardo differs from other Arduino boards using separate USB-Serial chip in that the ATmega32u4 has built-in USB communication, eliminating the need for a secondary processor. This allows the Leonardo to appear to a connected computer as a mouse and keyboard.

### TECH SPECS:

<b>Microcontroller</b>	Atmega32u4
<b>Operating Voltage</b>	5V
<b>Input Voltage (recommended)</b>	DC7-12V
<b>Digital I/O Pins</b>	20 (of which 7 provide PWM output)
<b>PWM Digital I/O Pins</b>	7
<b>Analog Input Pins</b>	12
<b>DC Current per I/O Pin</b>	40 mA
<b>DC Current for 3.3V Pin</b>	50 mA
<b>Flash Memory</b>	32 KB (Atmega32u4) of which 4 KB used by bootloader
<b>SRAM</b>	2.5 KB (ATmega32u4)
<b>EEPROM</b>	1 KB (Atmega32u4)
<b>Clock Speed</b>	16 MHz
<b>LED_BUILTIN</b>	D13
<b>Dimensions</b>	71mm*54mm*15mm
<b>Weight</b>	18.4g

### Details:

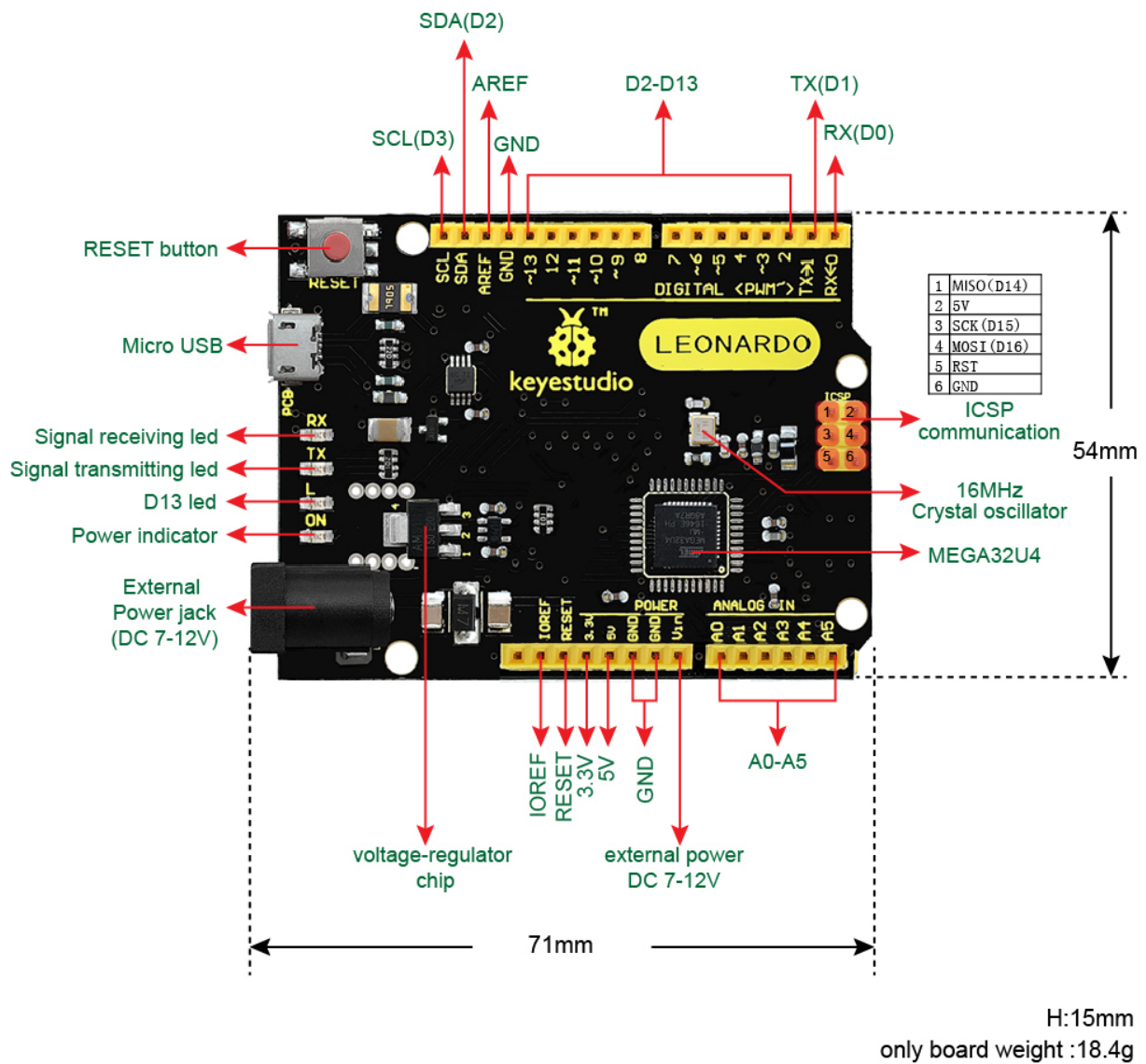
- PCB Dimensions: 71mm\*54mm\*15mm
- Weight: 18.4g



H:15mm  
only board weight :18.4g

Element and Interfaces:

Here is an explanation of what every element and interface of the board does:



Specialized Functions of Some Pins:

<b>Digital I/O pins</b>	D0-D13 and A0-A5 (D18-D23); Note that if the digital pins are not enough, the ICSP pins can be used as digital pins. MISO (D14); SCK(D15); MOSI (D16).
<b>Analog Inputs</b>	A0-A5, A6-A11 (on digital pins 4, 6, 8, 9, 10, and 12). That is, D4 (A6)D6 (A7)D8 (A8)D9 (A9)D10 (A10) and D12 (A11). Pins A0-A5 appear in the same locations as on the Uno; inputs A6-A11 are on digital i/o pins 4, 6, 8, 9, 10, and 12 respectively. Each analog input provide 10 bits of resolution (i.e. 1024 different values). By default the analog inputs measure from ground to 5 volts, though is it possible to change the upper end of their range using the AREF pin and the <a href="#">analogReference()</a> function.
<b>PWM (Pulse-Width Modulation)</b>	D3, D5, D6, D9, D10, D11 and D13. Provide 8-bit PWM output with the <a href="#">analogWrite()</a> function.
<b>External Interrupts</b>	D3 (interrupt 0); D2 (interrupt 1); D0 (interrupt 2), D1 (interrupt 3) and D7 (interrupt 4). 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 <a href="#">attachInterrupt()</a> function for details.
<b>Serial communication</b>	D0 (RX) and D1 (TX).
<b>SPI communication</b>	On the ICSP header. These pins support SPI communication using the <a href="#">SPI library</a> . Note: the SPI pins are not connected to any of the digital I/O pins as they are on the Uno. They are only available on the ICSP connector. This means that if you have a shield that uses SPI, but does NOT have a 6-pin ICSP connector that connects to the Leonardo's 6-pin ICSP header, the shield will not work.
<b>AREF</b>	Reference voltage for the analog inputs. Used with <a href="#">analogReference()</a> . Sometimes used to set the external reference voltage (0-5 volts) as the upper end of analog input pins.
<b>IOREF</b>	The voltage at which the i/o pins of the board are operating (i.e. VCC for the board). This is 5V on the Leonardo. Used to configure the operating voltage of microcontroller.

Tips:

- **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.

- **USB Overcurrent Protection:**

The Leonardo has a resettable polyfuse that protects your computer's USB ports from shorts and overcurrent. Although most computers provide their own internal protection, the fuse provides an extra layer of protection. If more than 500 mA is applied to the USB port, the fuse will automatically break the connection until the short or overload is removed.

Detailed Use with ARDUINO Software as follows:

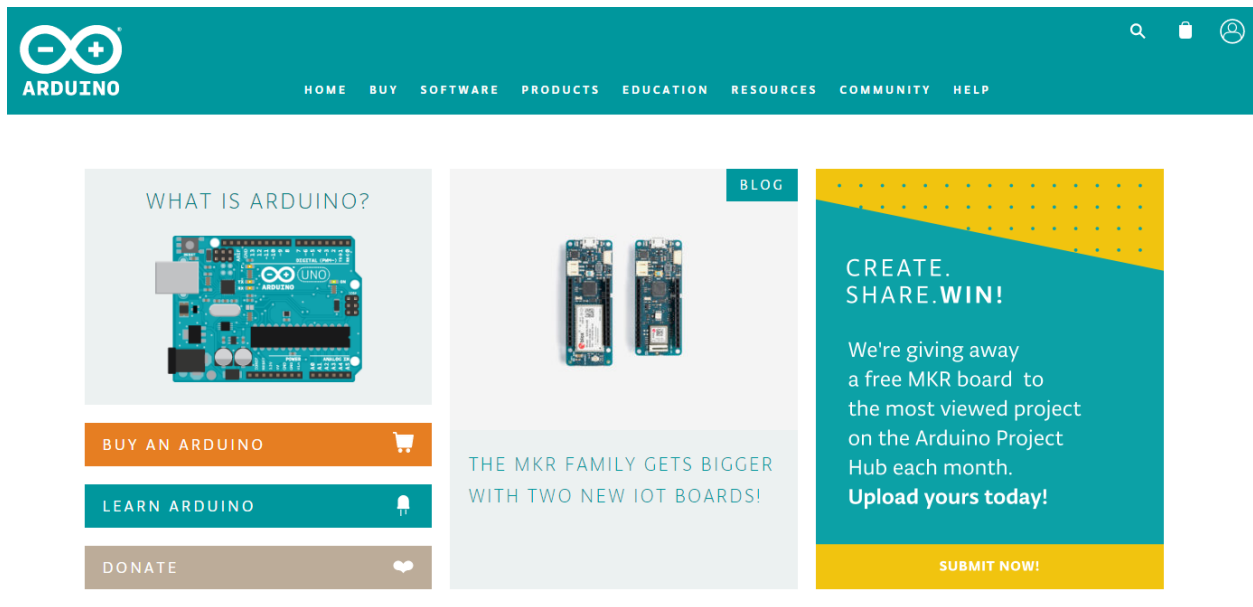
Step1| Download the Arduino IDE

When you get the board, first you should install the Arduino software and driver.

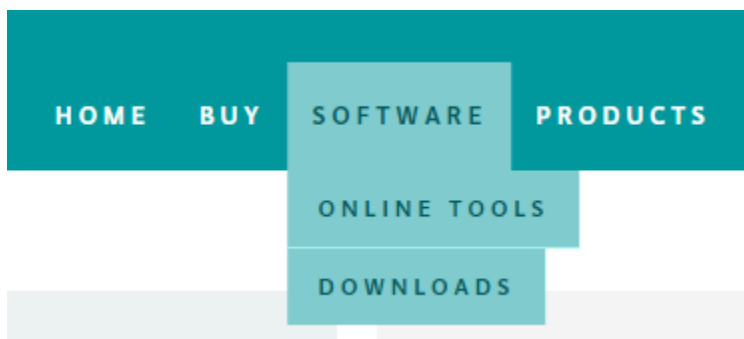
We usually use the Windows software Arduino 1.5.6 version. You can download it from the link below:

<https://www.arduino.cc/en/Main/OldSoftwareReleases\1.5.x>

Or you can browse the ARDUINO website to download the latest version from this link, <https://www.arduino.cc>, pop up the following interface.




Then click the **SOFTWARE** on the browse bar, you will have two options **ONLINE TOOLS** and **DOWNLOADS**.



Click **DOWNLOADS**, it will appear the latest software version of ARDUINO 1.8.5 shown as below.

## Download the Arduino IDE



**ARDUINO 1.8.5**

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** 10.7 Lion or newer

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

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

In this software page, on the right side you can see the version of development software for different operating systems. ARDUINO has a powerful compatibility. You should download the software that is compatible with the operating

system of your computer.

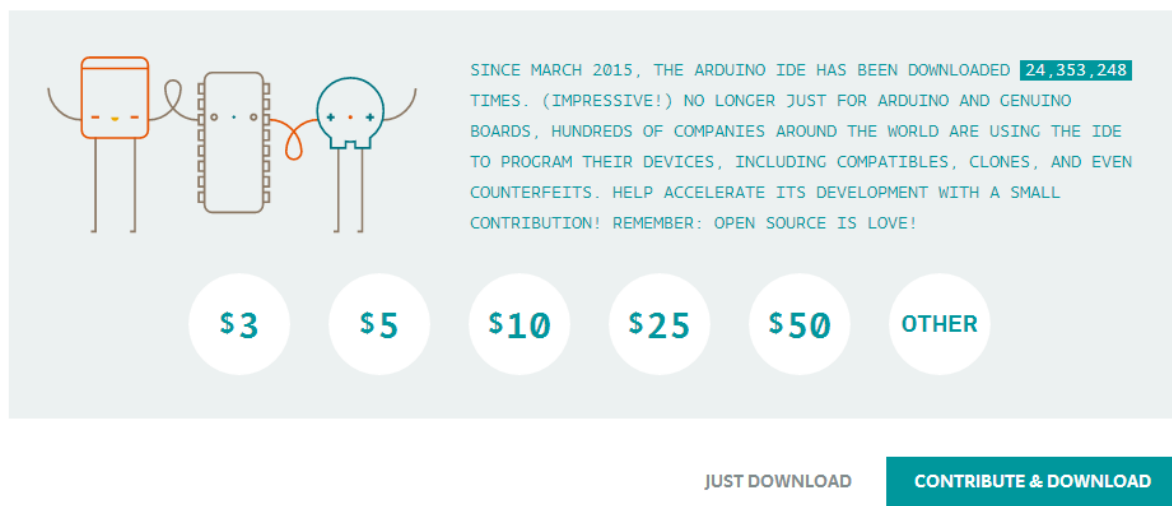
We will take **WINDOWS system** as an example here. There are also two options under Windows system, one is installed version, the other is non-installed version.

For simple installed version, first click **Windows Installer**, you will get the following page.



## Contribute to the Arduino Software

Consider supporting the Arduino Software by contributing to its development. (US tax payers, please note this contribution is not tax deductible). [Learn more on how your contribution will be used.](#)



This way you just need to click JUST DOWNLOAD, then click the downloaded file to install it.

For non-installed version, first click Windows ZIP file, you will also get the pop-up interface as the above figure.

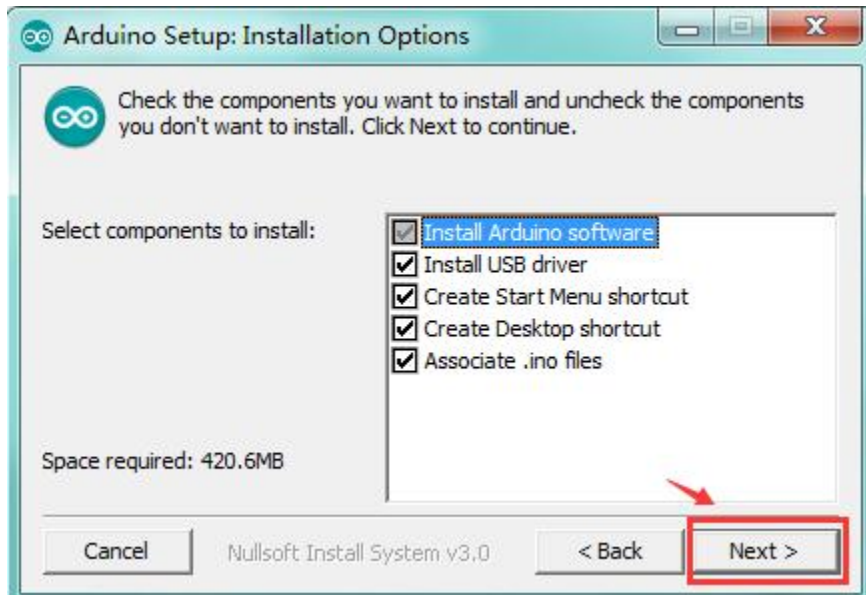
Click JUST DOWNLOAD, and when the ZIP file is downloaded well to your computer, you can directly unzip the file and click the icon of ARDUINO software to start it.

### Installing Arduino (Windows):

Install Arduino with the exe. Installation package downloaded well.

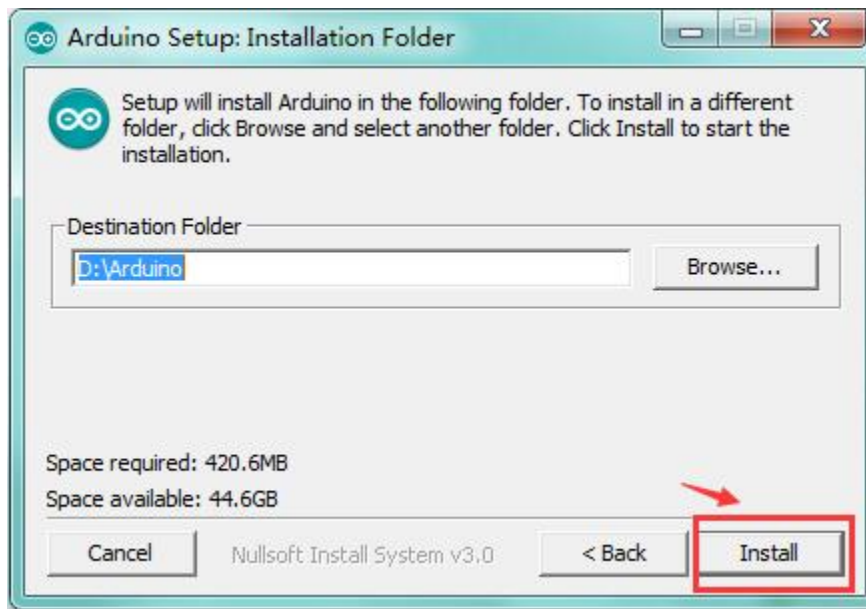


Click “I Agree” to see the following interface.

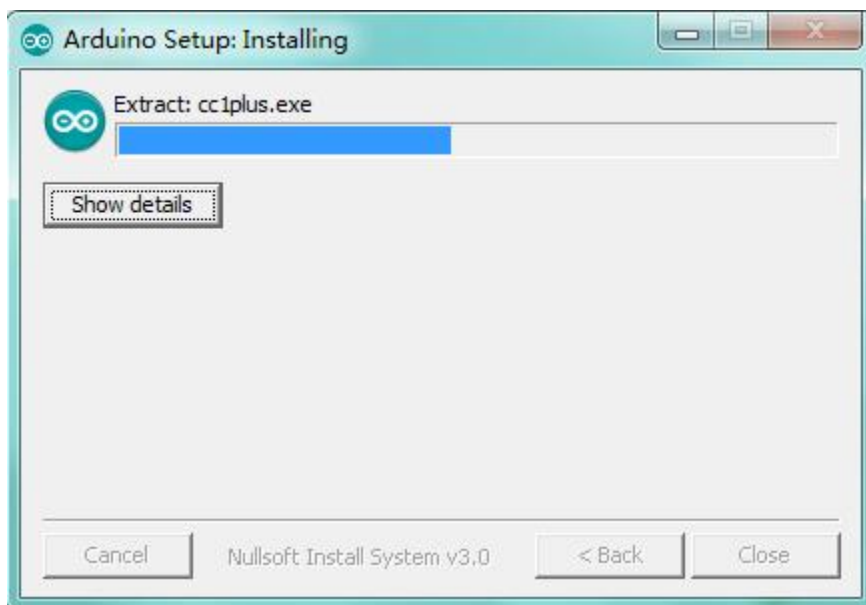


Click “Next”. Pop up the interface below.



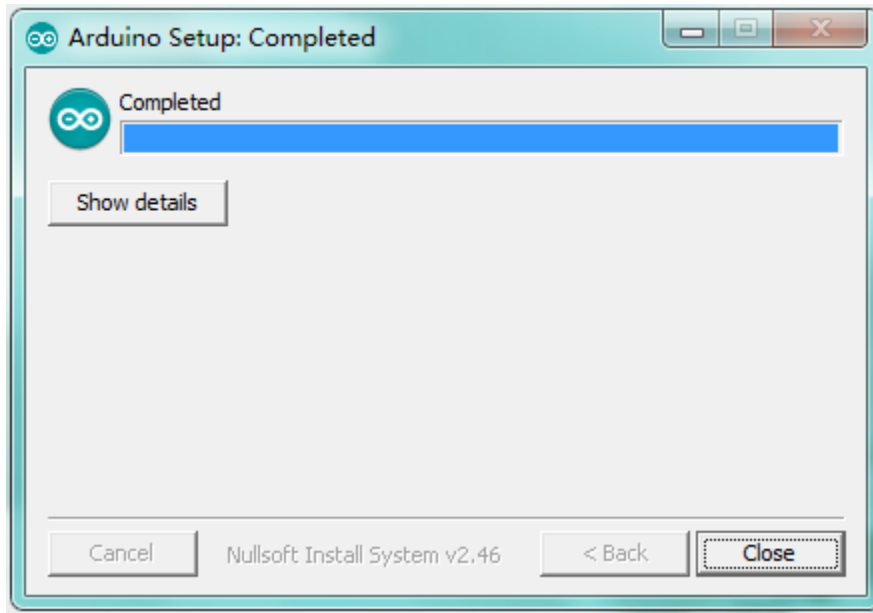


You can press Browse... to choose an installation path or directly type in the directory you want. Then click “Install” to initiate installation.



Wait for the installing process, if appear the interface of Window Security, just continue to click Install to finish the installation.

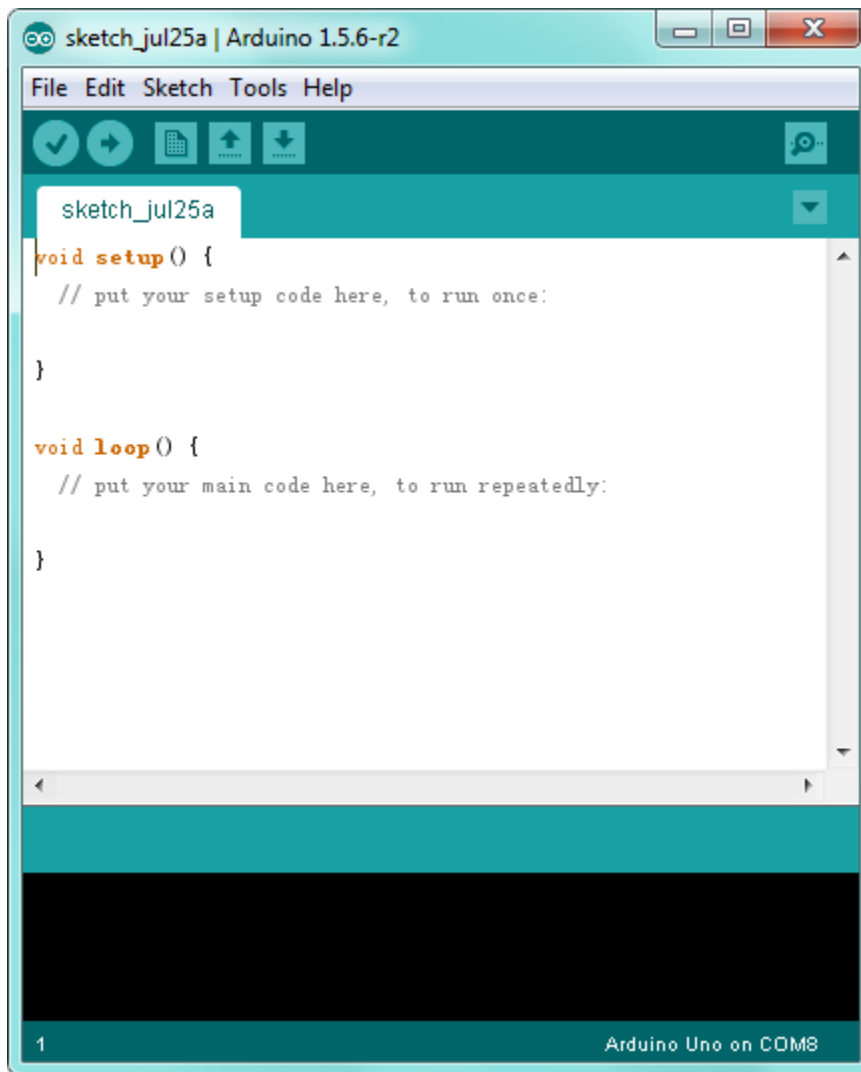


**Introduction for Arduino IDE Toolbar:**

Double-click the icon of Arduino software downloaded, you will get the interface shown below.

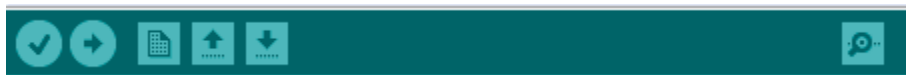








(Note: if the Arduino software loads in the wrong language, you can change it in the preferences dialog. See [the environment page](#) for details.)



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

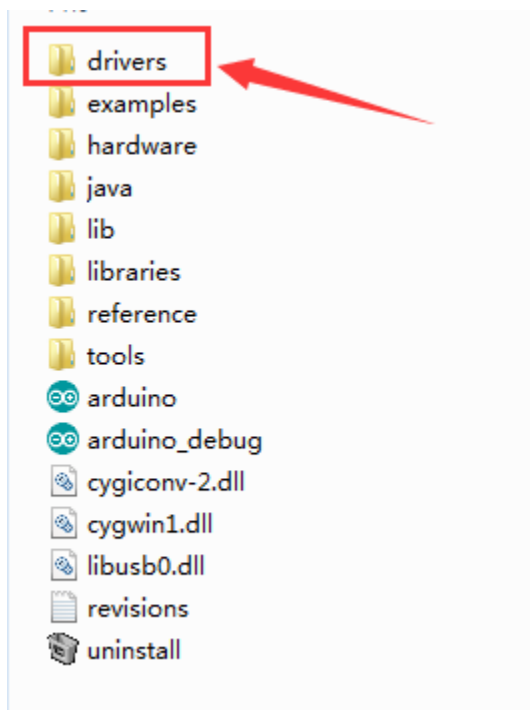
<http://wiki.keyestudio.com/index.php/File:IDE.png>



	<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

### Step2| Installing the Driver

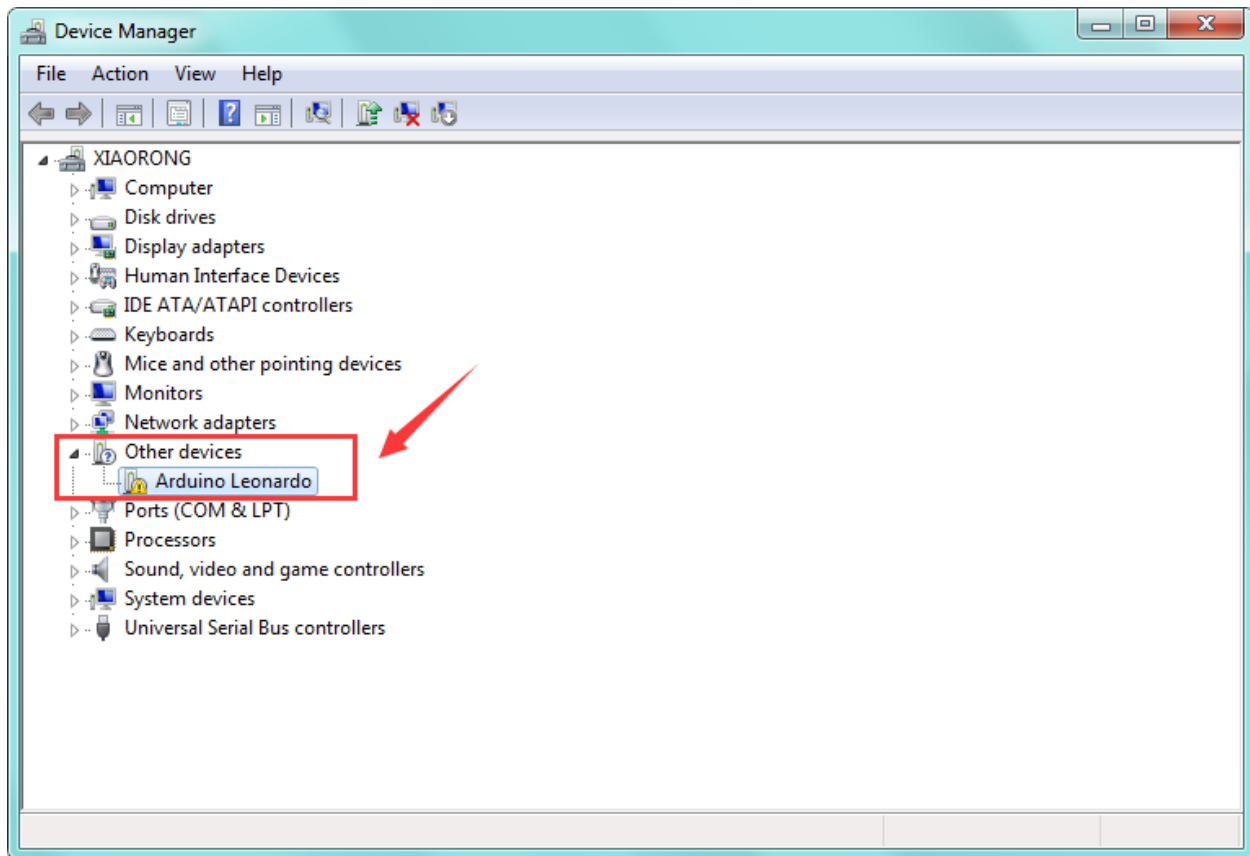
Installed well the Arduino, the next step is to install the driver. The Arduino folder contains both the Arduino program itself and the drivers that allow the Arduino to be connected to your computer with a USB cable.



In different systems, the driver installation is similar. Here we start to install the driver on the Win7 system.

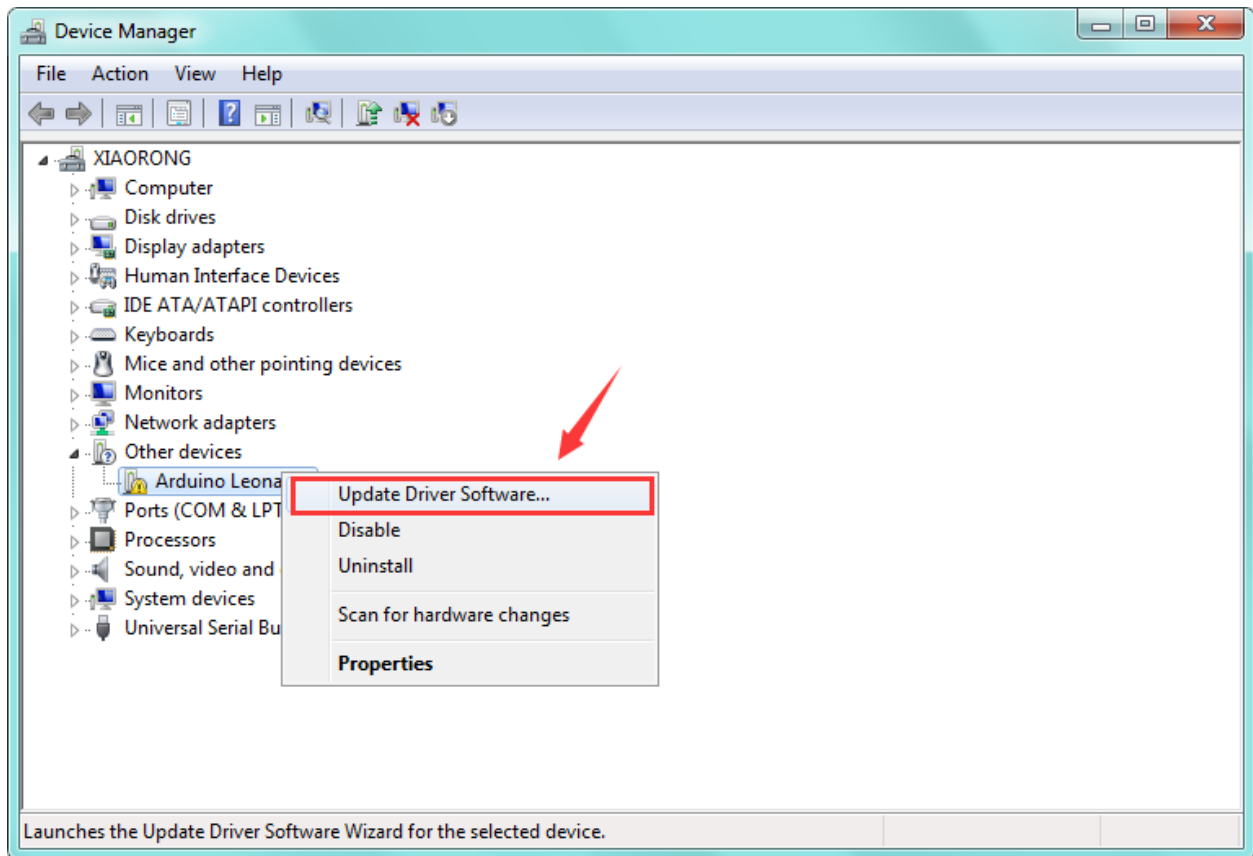
Plug one end of your USB cable into the keystudio Leonardo and the other into a USB socket on your computer.

When you connect the board to your computer at the first time, right click your “Computer” —>for “Properties”—> click the “Device manager”, under Other devices, you should see the “Arduino Leonardo”.

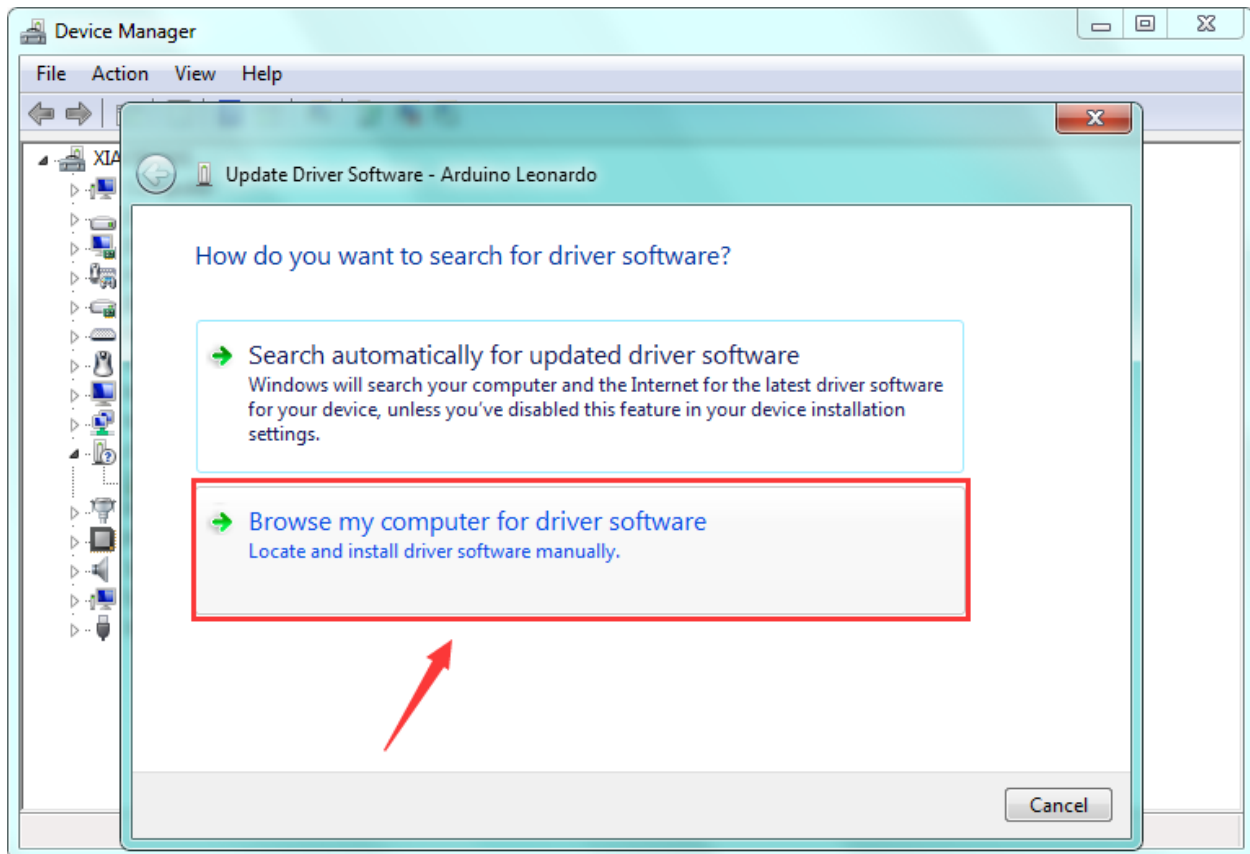


[http://wiki.keyestudio.com/index.php/File:Driver\\_2.png](http://wiki.keyestudio.com/index.php/File:Driver_2.png)

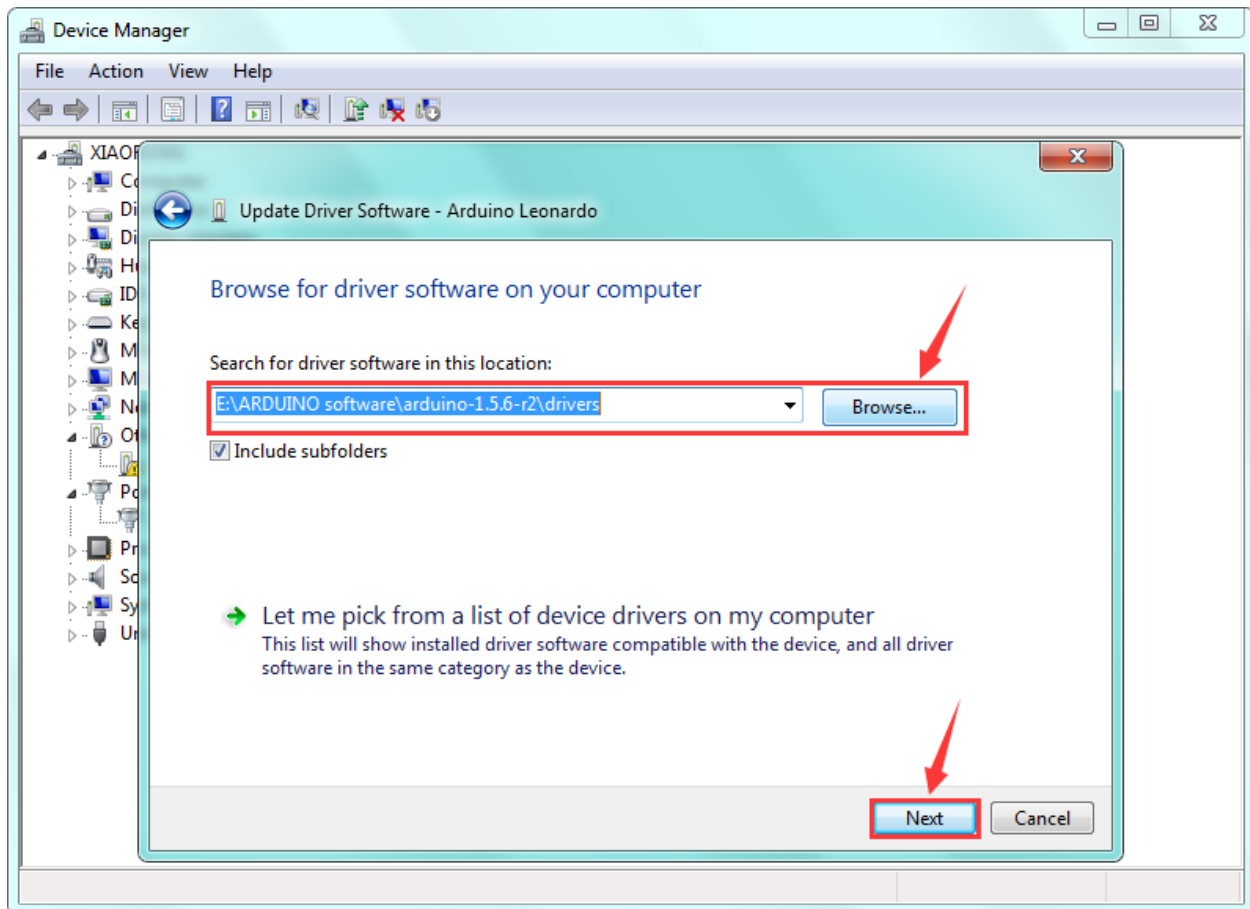
Then right-click on the Arduino Leonardo and select the top menu option (Update Driver Software...) shown as the figure below.



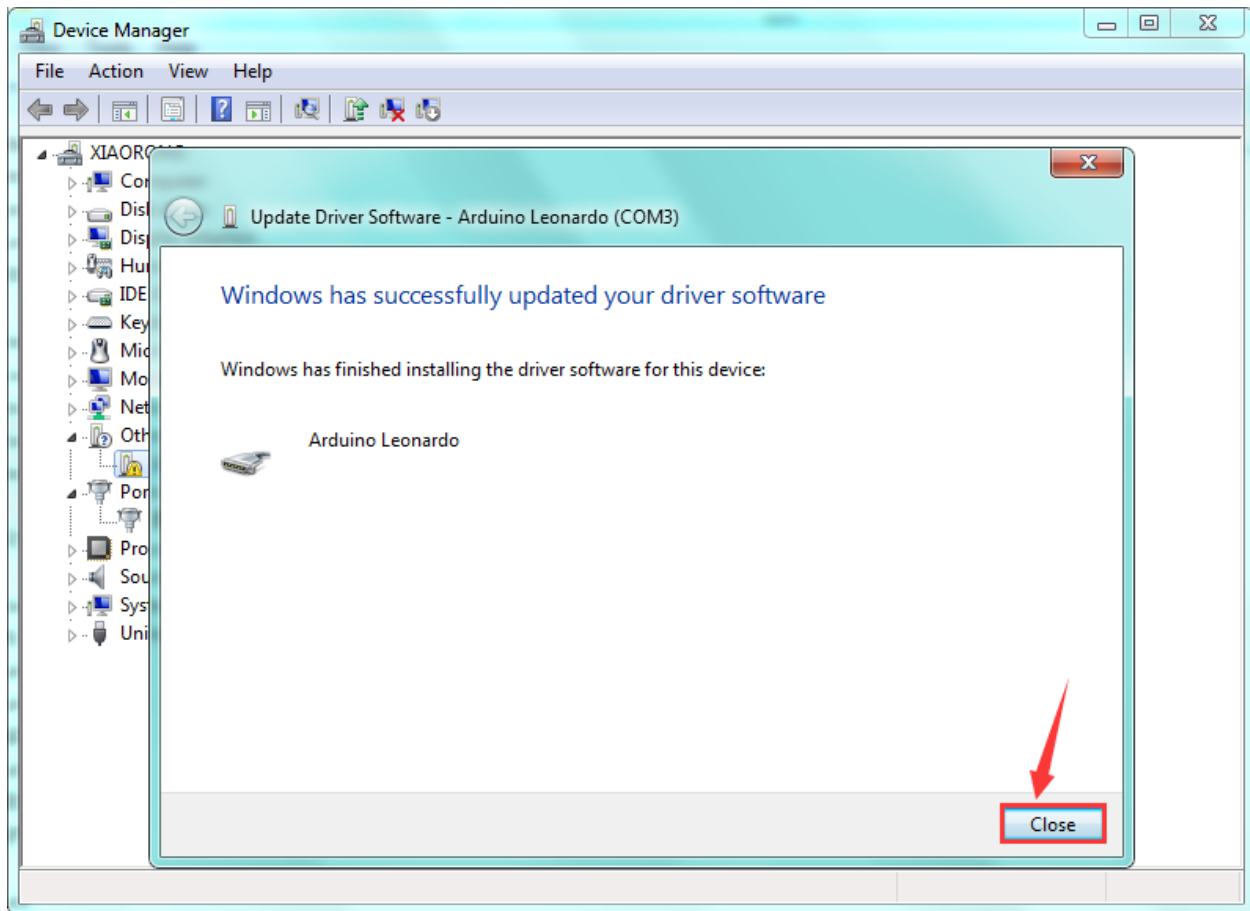
Then it will be prompted to either “Search Automatically for updated driver software” or “Browse my computer for driver software”. Shown as below. In this page, select “Browse my computer for driver software”.



After that, select the option to browse and navigate to the “drivers” folder.

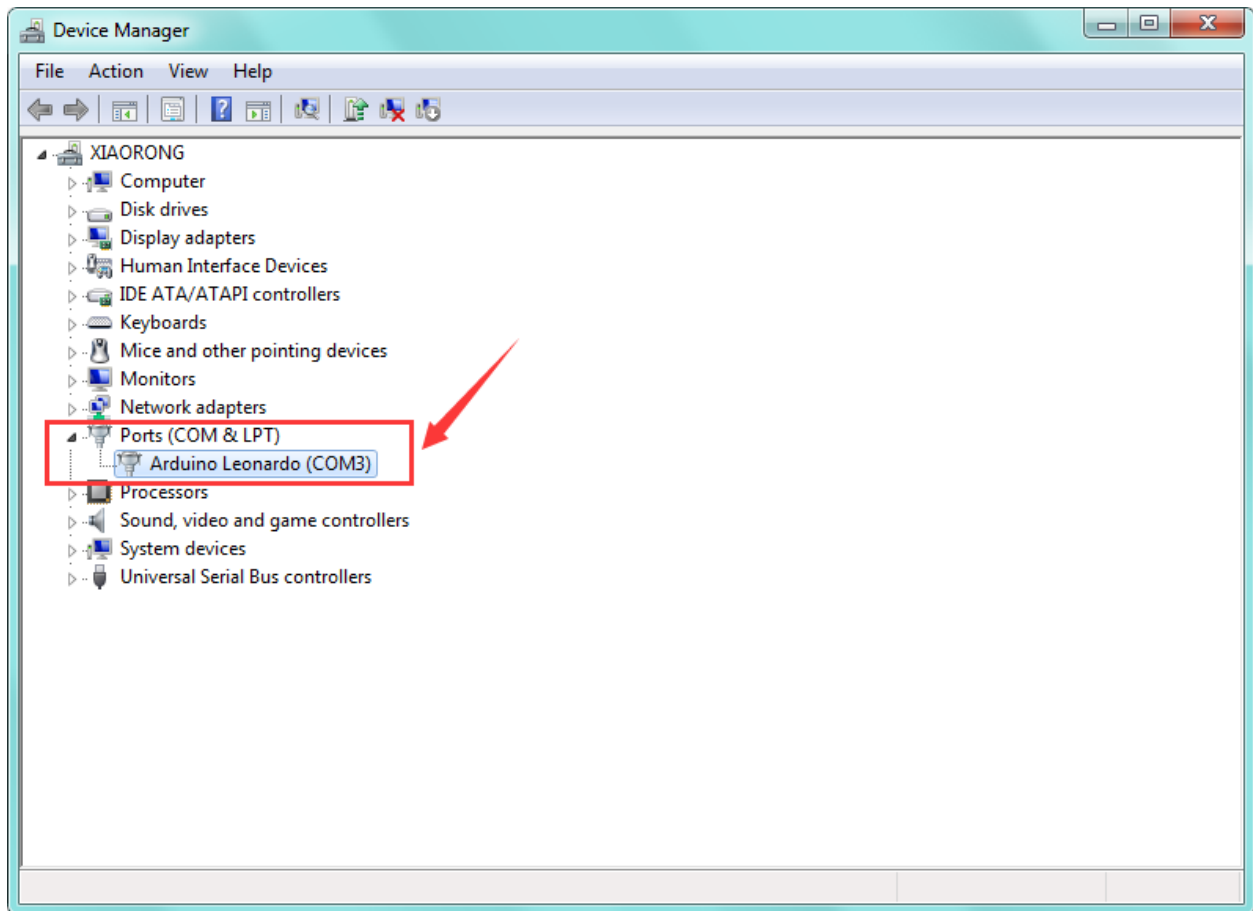


Once the software has been installed, you will get a confirmation message. Installation completed, click "Close".  
[http://wiki.keyestudio.com/index.php/File:Driver\\_6.png](http://wiki.keyestudio.com/index.php/File:Driver_6.png)



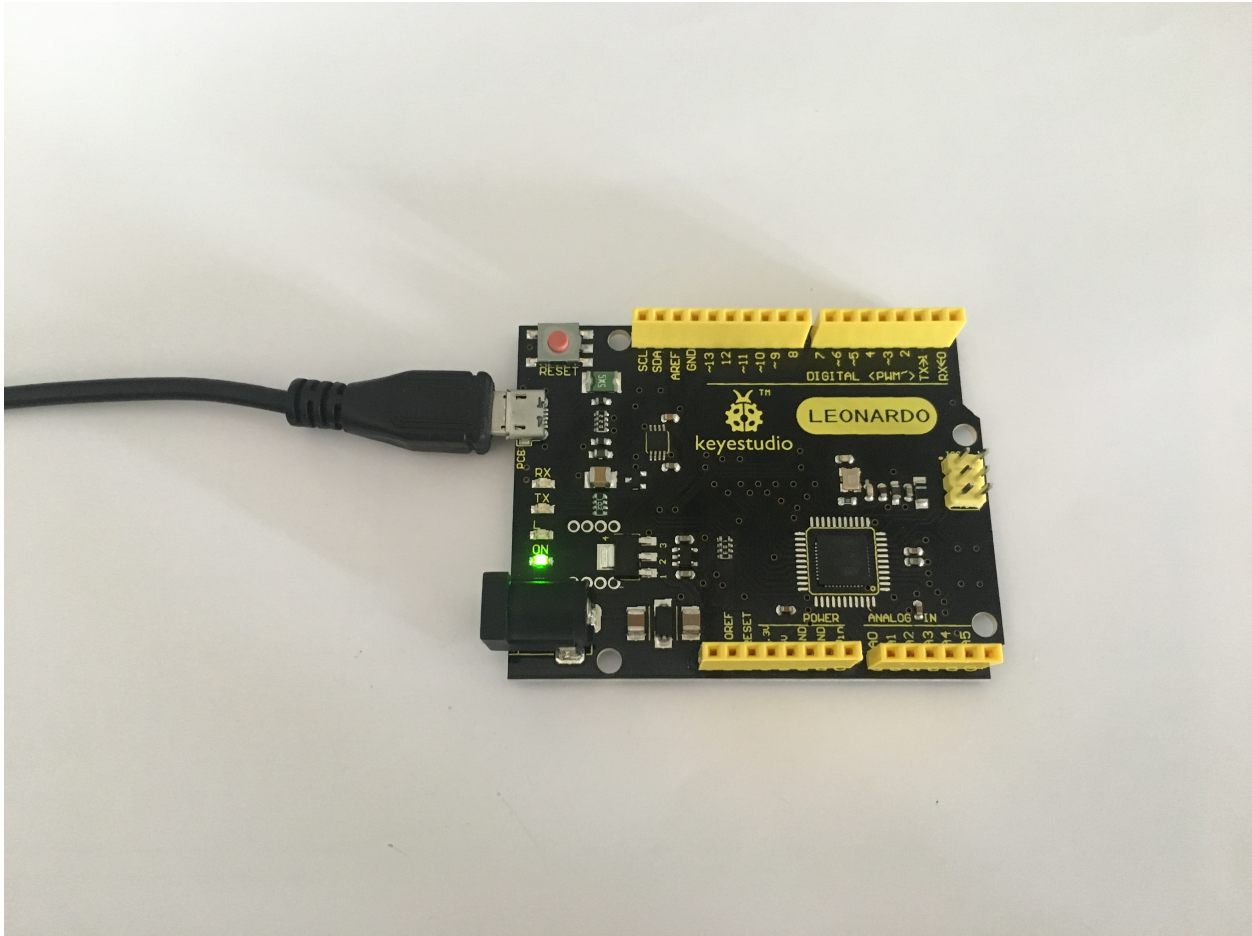
Up to now, the driver is installed well. Then you can right click “Computer” —>“Properties”—>“Device manager”, you should see the device as the figure shown below.





Step3| Connect the board

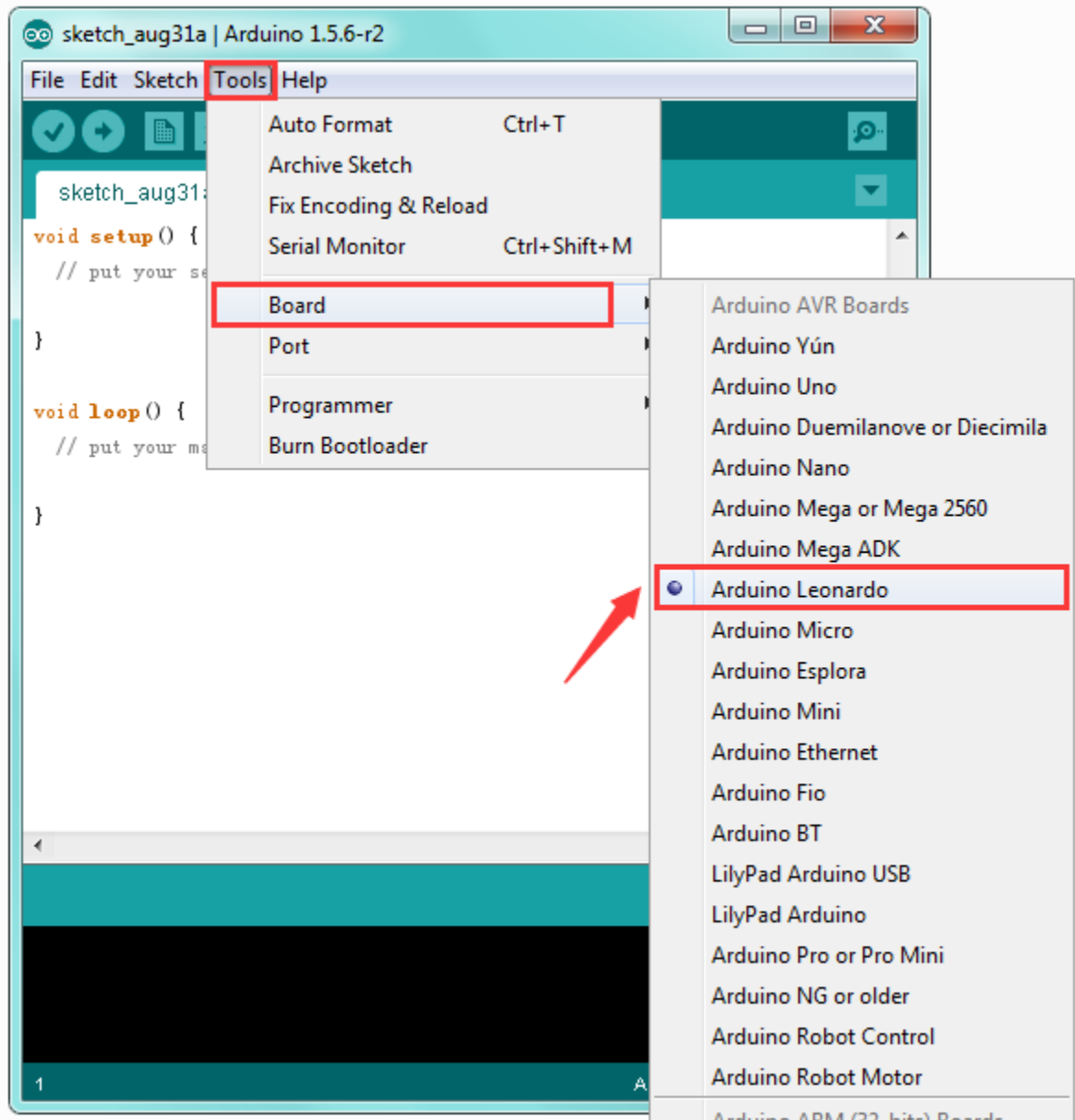
Connect the Leonardo board to your computer using the USB cable. The power LED should go on.



Step4| Select the Arduino Board

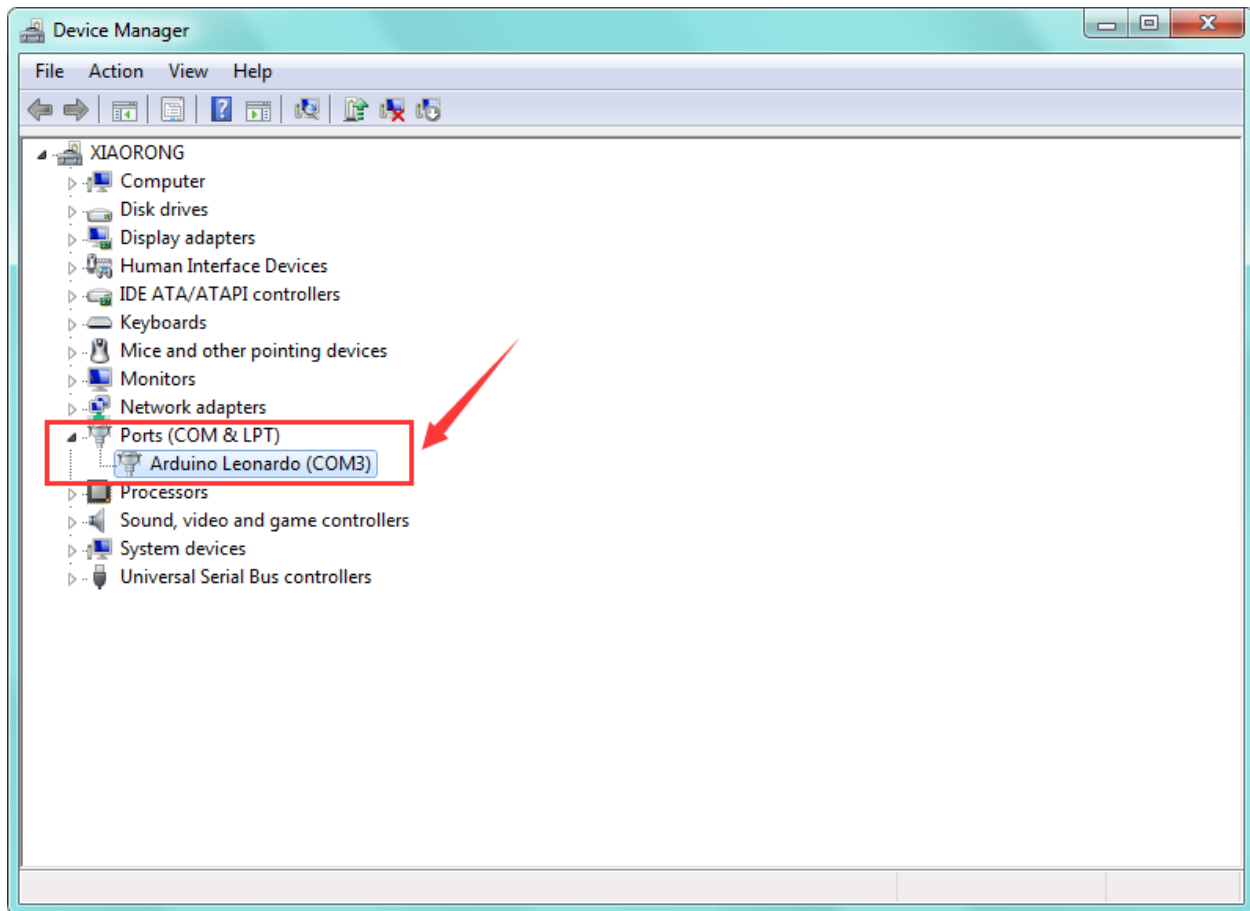
Open the Arduino IDE, you'll need to click the "Tools", then select the Board that corresponds to your Arduino.

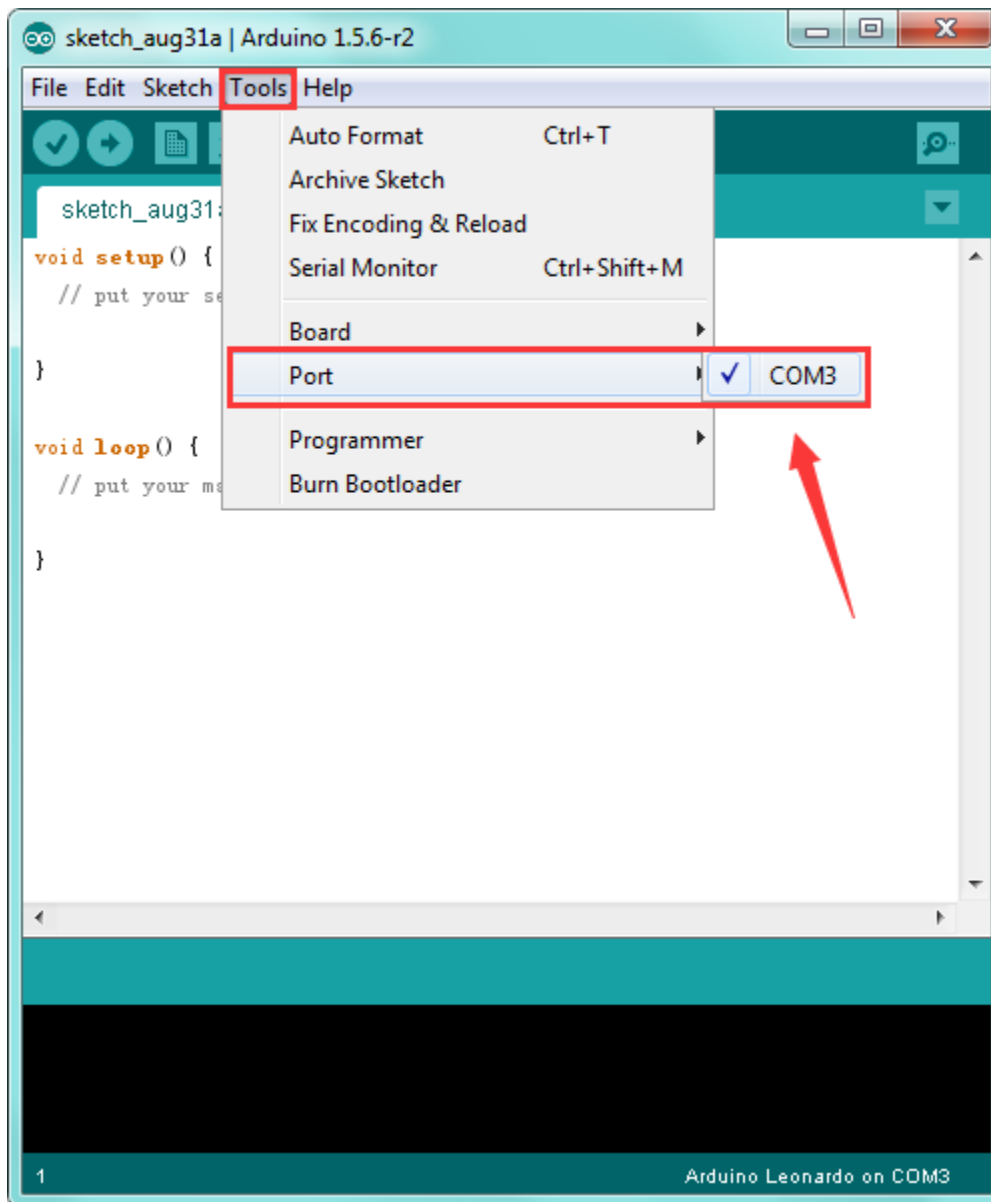




Step5| Select your serial port

Select the serial device of the Arduino board from the **Tools | Serial Port menu**. **Note:** to avoid errors, the COM Port should keep the same as the Ports shown on Device Manager.





Step6| Upload the Code

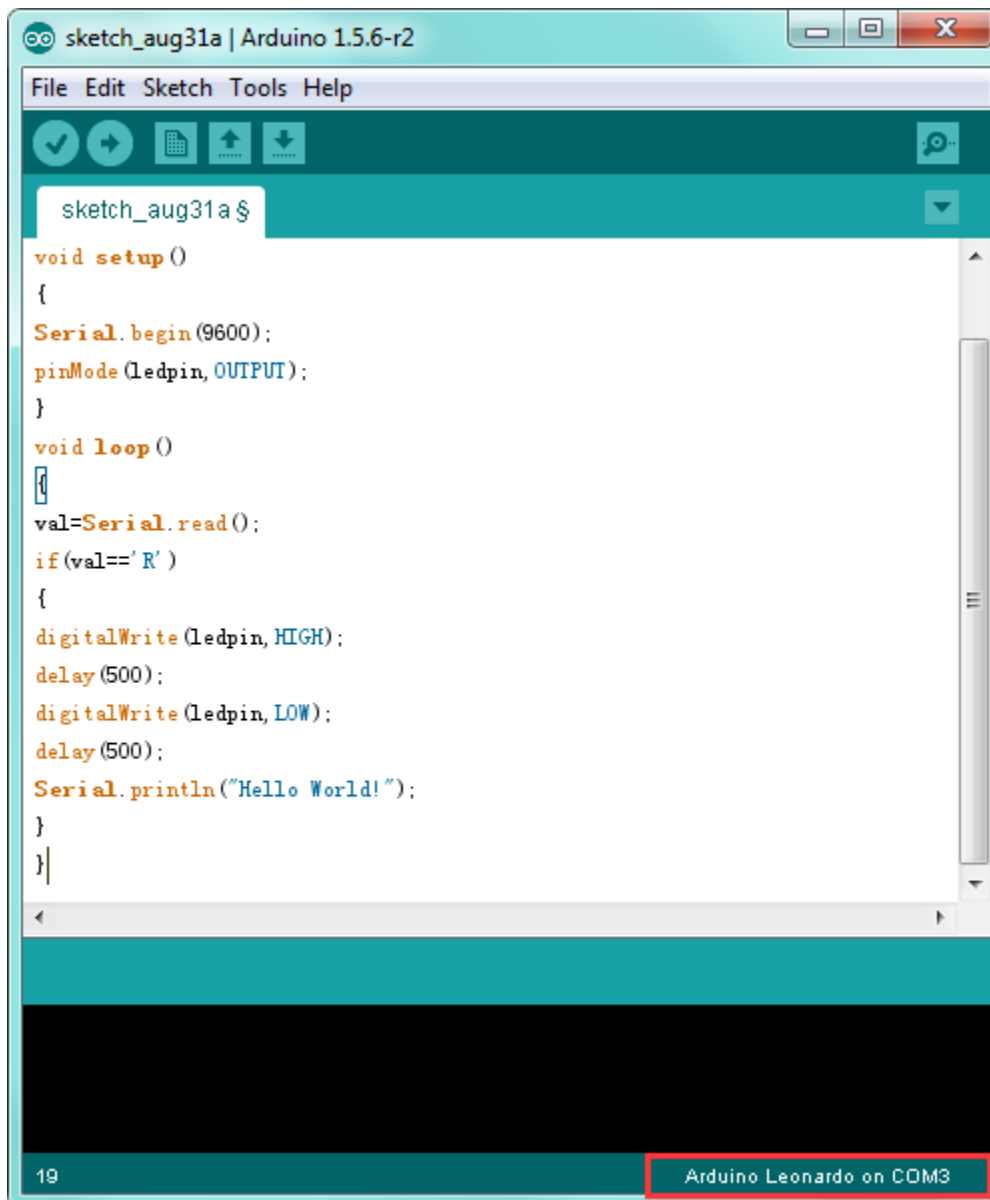
Below is an example code for displaying the Hello World!

Copy and paste the code to the Arduino environment IDE.

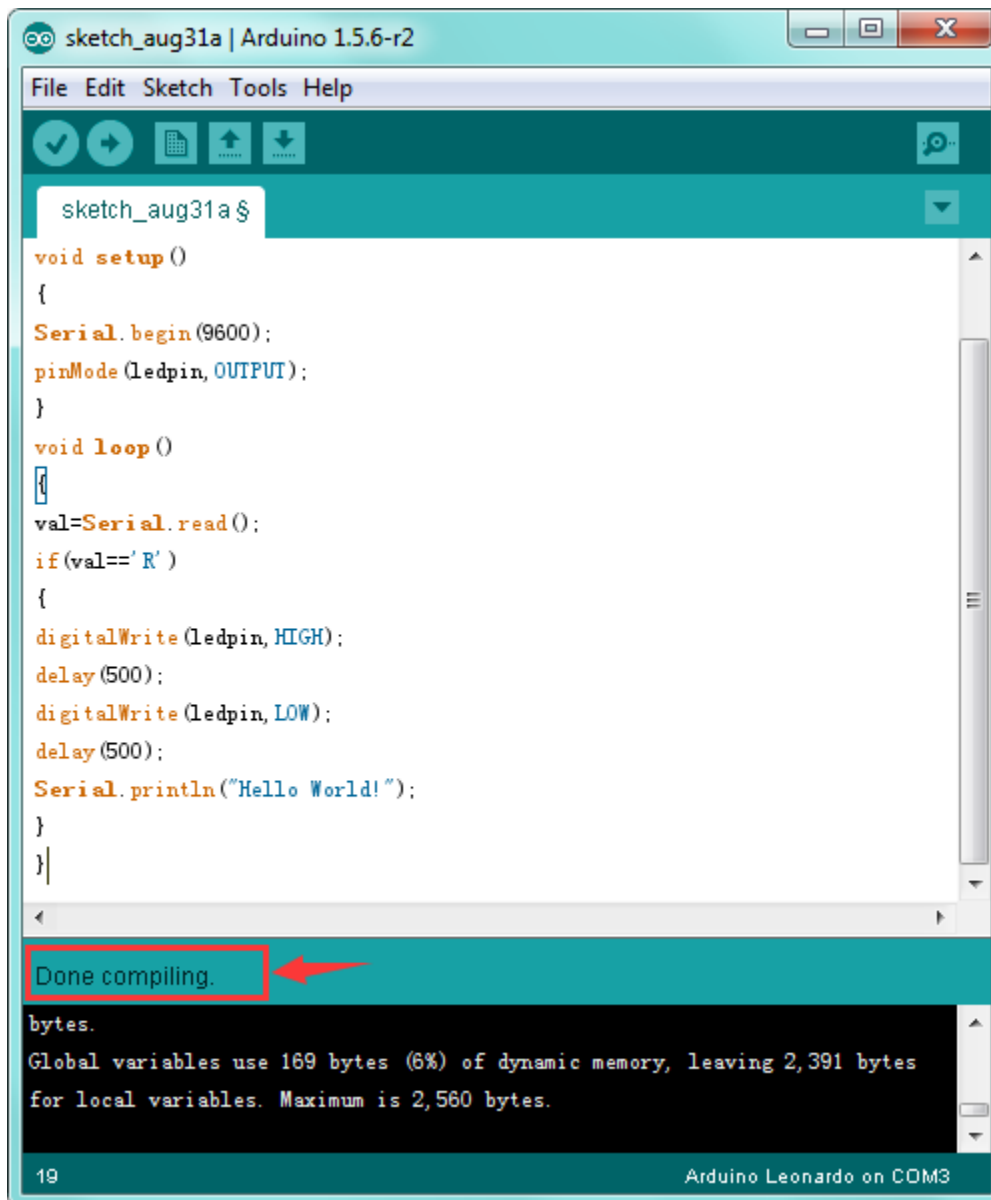
////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////

```
int val;  
int ledpin=13;  
void setup()  
{  
  Serial.begin(9600);  
  pinMode(ledpin,OUTPUT);  
}
```

```
void loop()
{
  val=Serial.read();
  if(val=='R')
  {
    digitalWrite(ledpin,HIGH);
    delay(500);
    digitalWrite(ledpin,LOW);
    delay(500);
    Serial.println("Hello World!");
  }
}
////////////////////////////////////////////////////////////////
```

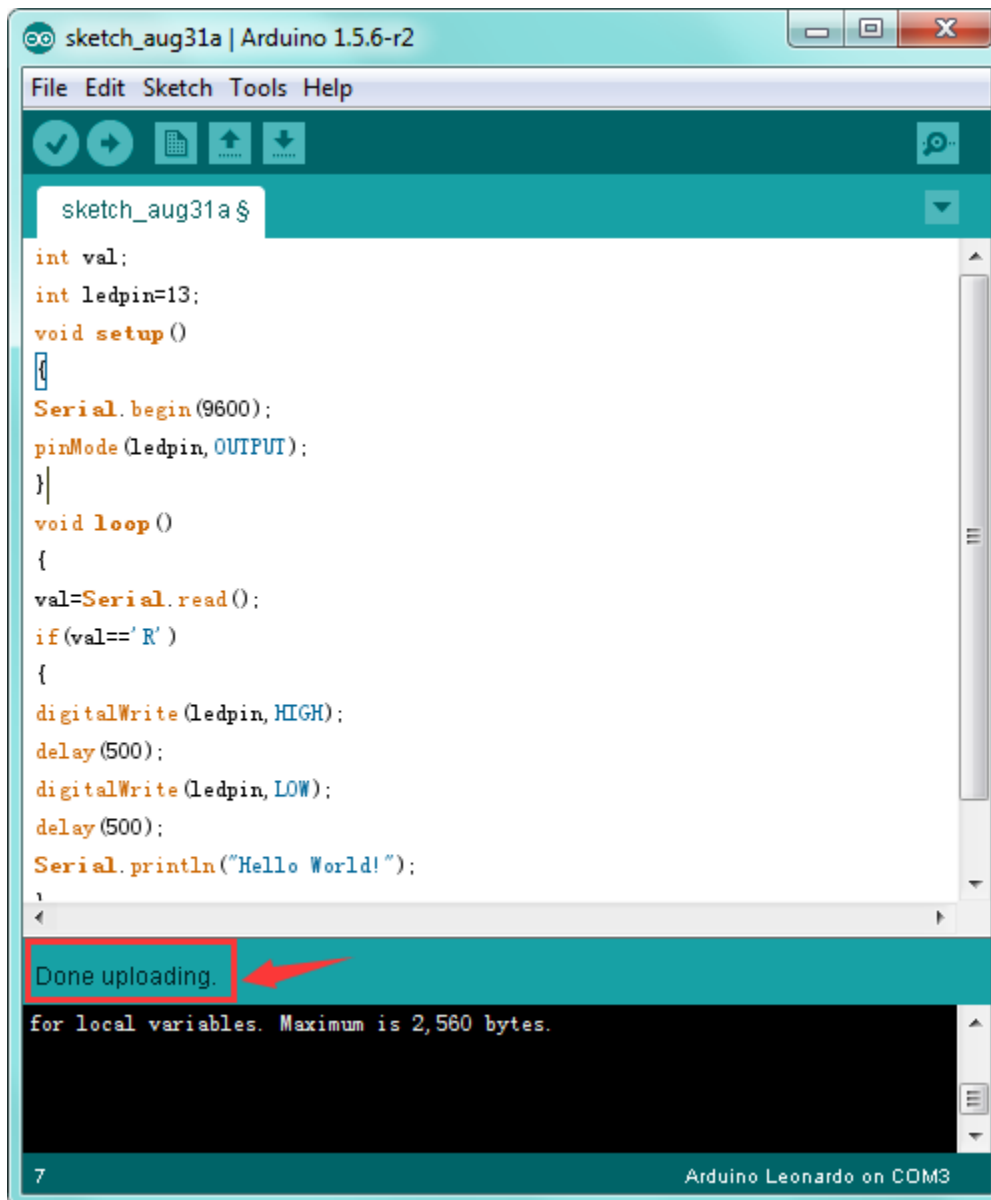


Then click verify button to check the errors. If compiling successfully, the message “Done compiling.” will appear in the status bar.



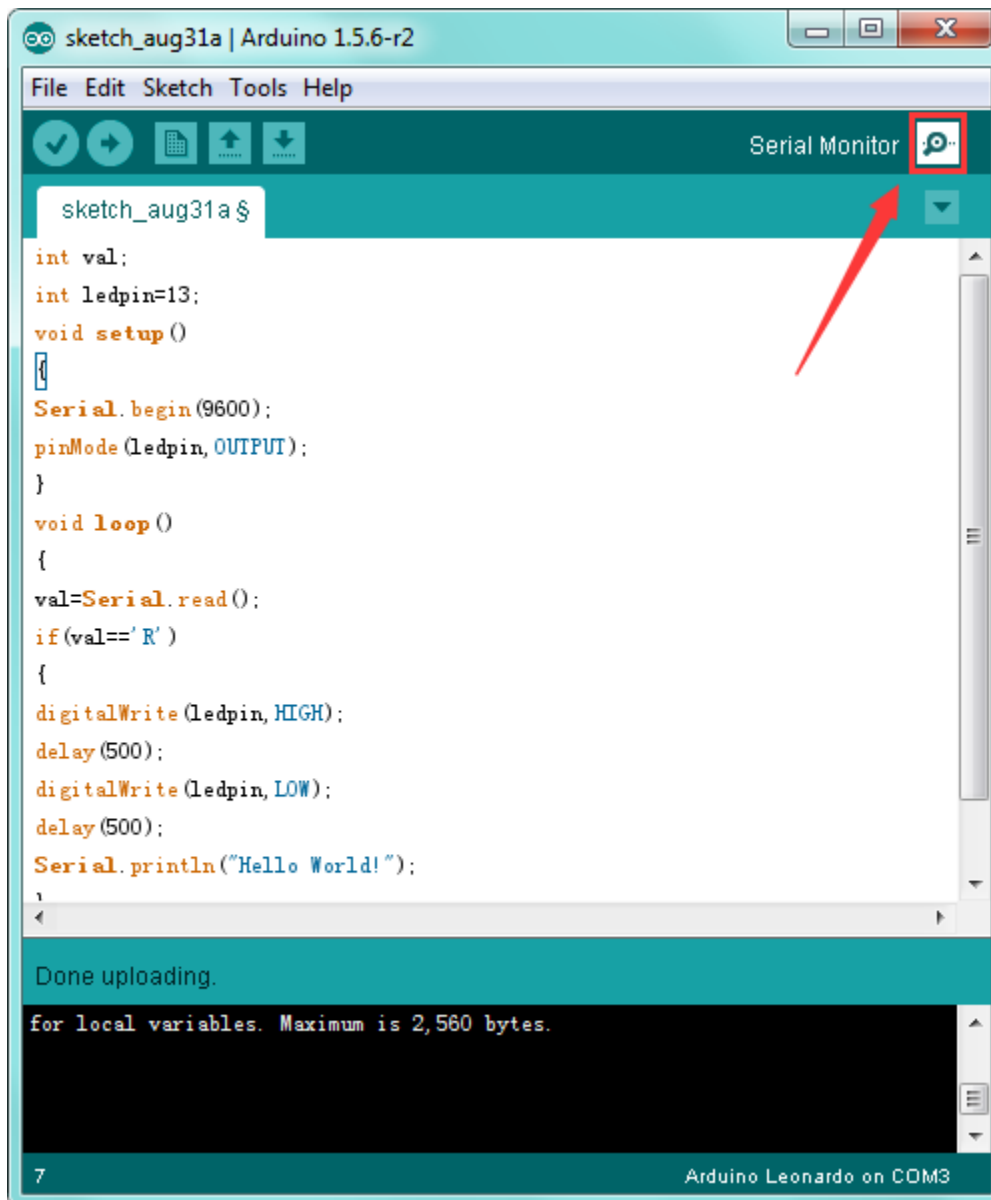
After that, click the “Upload” button to upload the code. If the upload is successful, the message “Done uploading.” will appear in the status bar.



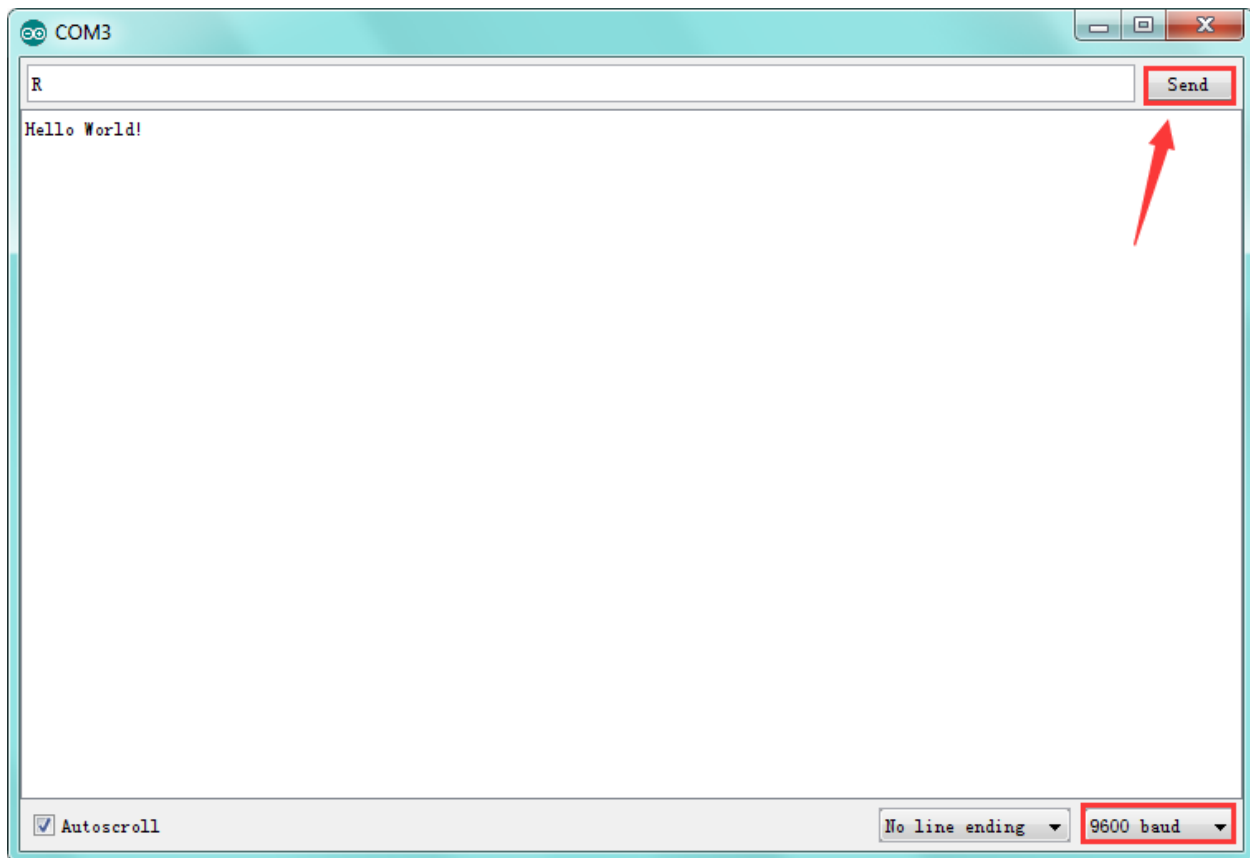


Step7| Open the Serial Monitor

After that, click the button to open the serial monitor.



Then set the baud rate to 9600, enter an “R” and click Send, that is, the computer will send the character R. When the board receives it, you should see the RX led on the board flash once, and then D13 led flash once; when keystudio Leonardo R3 board successfully sends “Hello World!” to the computer, you should see the “Hello World!” is showed on the monitor, and TX led on the board flash once.



Package Included:

- keyestudio Leonardo R3 board \* 1pcs
- Black micro USB cable 1m \* 1pcs



Resource Links:

You can download the datasheet from the link:

<https://drive.google.com/open?id=1fikRPqsnIBHzVaAgCX0QBcCTpWg9BcQP>

You might also want to look at:

[the reference](#) for the Arduino language;

Download ARDUINO Software:

<https://www.arduino.cc/en/Main/OldSoftwareReleases\1.5.x>

Or download the software and driver from the link below:

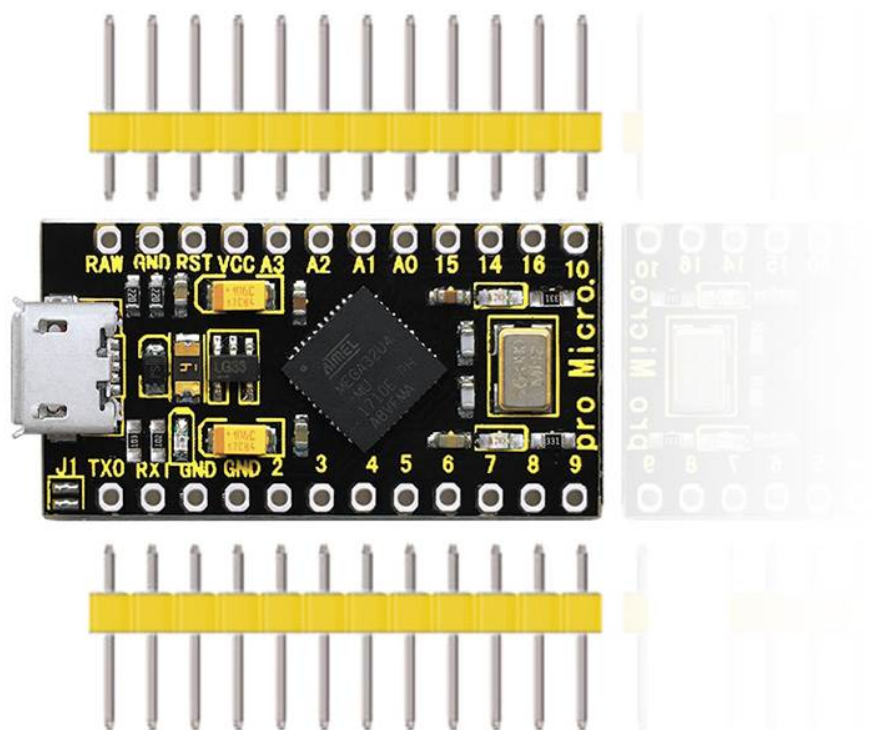
<https://drive.google.com/open?id=12D-JkXdNm03Qt4dIPQr3RP6OmgXqpVHc>

Troubleshooting:

If you have problems, please see the [troubleshooting suggestions](#).



## KS0249 KEYESTUDIO PRO MICRO DEVELOPMENT BOARD



Introduction:

keystudio PRO MICRO is a microcontroller board based on the ATMEGA32U4-MU. It is an easy-to-use open source hardware.

It has 18 digital input/output pins (of which 5 can be used as PWM outputs), 9 analog inputs, a 16 MHz crystal oscillator, and a micro USB connection.

It contains everything needed to support the microcontroller; simply connect it to a computer with a USB cable to get started.

The keystudio PRO MICRO can be powered via the micro USB connection, or via the interface RAW GND (DC 7-9V).

Note that the operating voltage of PRO MICRO is 3.3V.

The PRO MICRO differs from other Arduino boards using separate USB-Serial chip in that the ATMEGA32U4-MU has built-in USB communication, eliminating the need for a secondary processor.

It is easy to integrate this Micro in everyday objects to make them interactive.

To facilitate the physical design, the board is not welded with pin headers, so you can solder the pin headers by yourself. And the package includes 2pcs of yellow 1\*12 2.54 straight pins and 1m black micro USB cable.

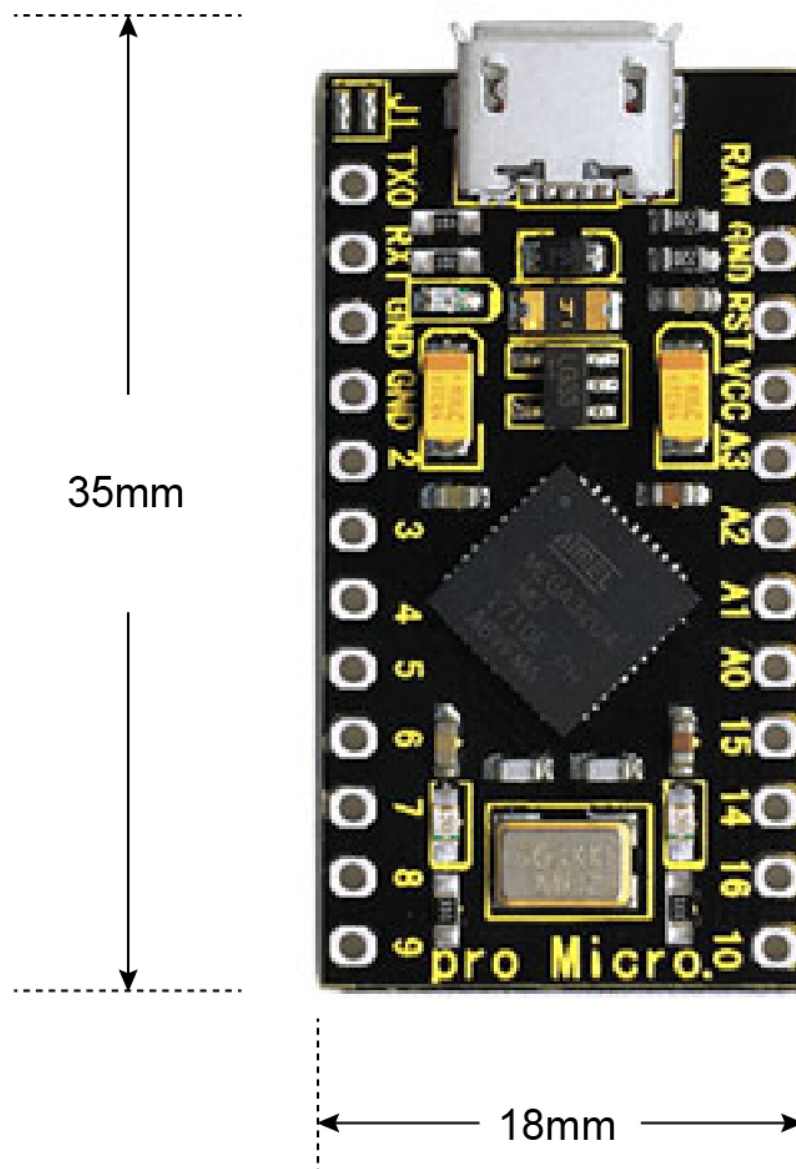
### TECH SPECS:

<b>Microcontroller</b>	ATMEGA32U4-MU
<b>RAW</b>	DC 7-9V
<b>VCC</b>	3.3V at 500mA
<b>Operating Voltage</b>	3.3V
<b>Digital I/O Pins</b>	18 (of which 5 provide PWM output)
<b>Analog Input Pins</b>	9
<b>Maximum current for chip</b>	200mA
<b>Maximum current per pin</b>	40mA
<b>Recommended current per pin</b>	20mA
<b>Atmel AVR</b>	8-bit
<b>Flash Memory</b>	32 KB
<b>SRAM</b>	2.5 KB
<b>EEPROM</b>	1 KB
<b>ADC</b>	10-bit
<b>PWM</b>	8-bit

### Details:

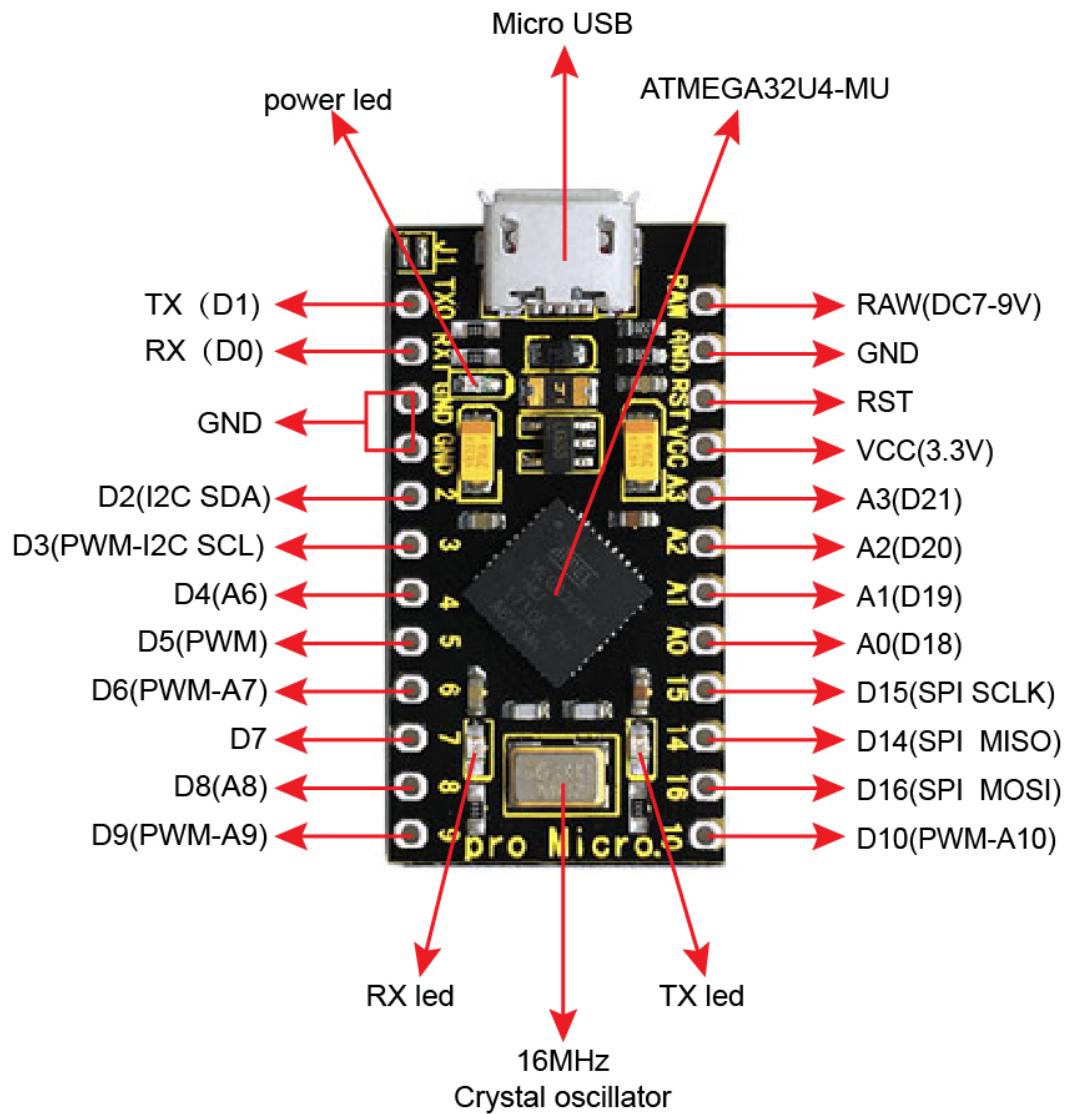
- PCB Dimensions: 35mm\*18mm\*2mm
- Weight: 2.6g





Element and Interfaces:

Here is an explanation of what every element and interface of the board does:



Specialized Functions of Some Pins:

<b>Digital pins</b>	RX (D0)TX (D1)D2-D10D14-D16A0-A3(D18-D21)
<b>Analog Inputs</b>	A0-A3, A6-A10 (on digital pins 4, 6, 8, 9, and 10). That is, D4 (A6)D6 (A7)D8 (A8)D9 (A9)D10 (A10).
<b>PWM (Pulse-Width Modulation)</b>	D3, D5, D6, D9 and D10. Provide 8-bit PWM output with the <code>analogWrite()</code> function.
<b>External Interrupts</b>	D3 (interrupt 0); D2 (interrupt 1); D0 (interrupt 2), D1 (interrupt 3) and D7 (interrupt 4). 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 <code>attachInterrupt()</code> function for details.
<b>Serial communication</b>	D0 (RX) and D1 (TX).
<b>SPI communication</b>	D14 (MISO); D15 (SCLK); D16 (MOSI)
<b>I2C communication</b>	D2 (SDA) and D3 (SCL)
<b>RAM</b>	Connect the external power DC 7-9V

Detailed Use with ARDUINO Software as follows:

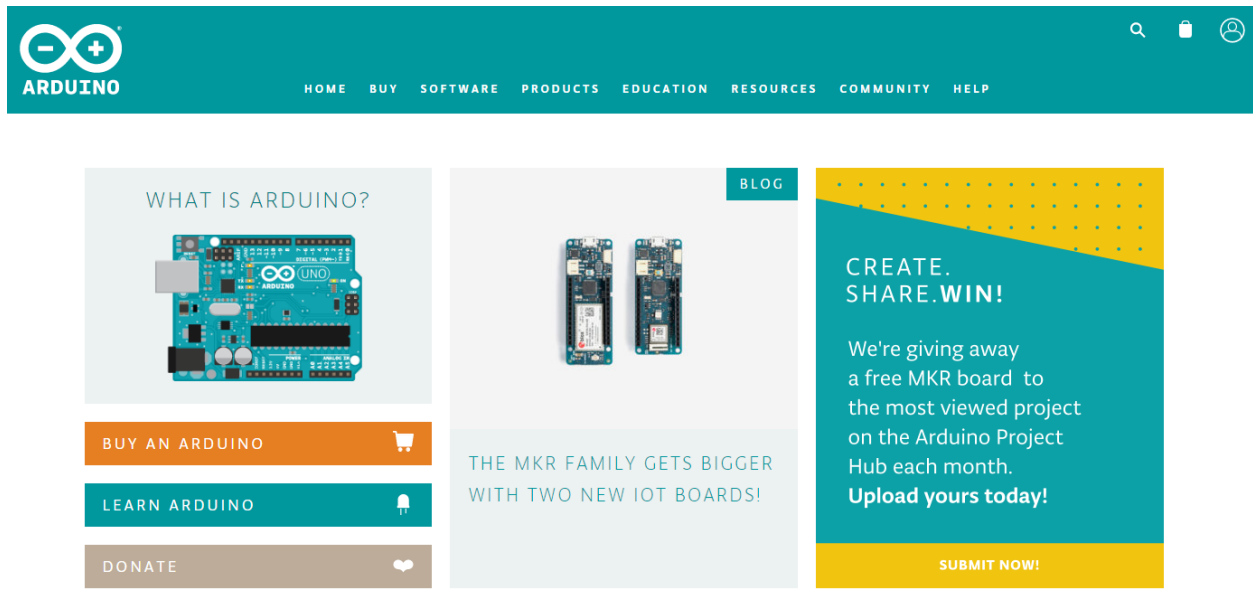
Step1| Download the Arduino IDE

When you get the board, first you should install the Arduino software and driver.

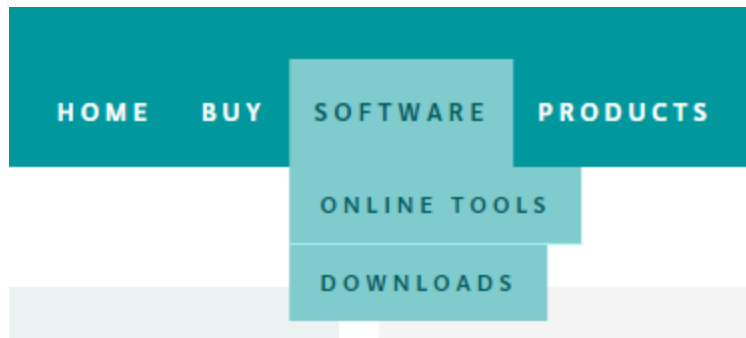
We usually use the Windows software Arduino 1.5.6 version. You can download it from the link below:

<https://www.arduino.cc/en/Main/OldSoftwareReleases\1.5.x>

Or you can browse the ARDUINO website to download the latest version from this link, <https://www.arduino.cc>, pop up the following interface.



Then click the **SOFTWARE** on the browse bar, you will have two options ONLINE TOOLS and DOWNLOADS.



Click **DOWNLOADS**, it will appear the latest software version of ARDUINO 1.8.5 shown as below.

## Download the Arduino IDE

A screenshot of the Arduino 1.8.5 download page. On the left, there is a large teal circle containing the Arduino logo (an infinity symbol with a minus sign on the left and a plus sign on the right). To the right of the logo, the text reads: 'ARDUINO 1.8.5', '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.', and 'This software can be used with any Arduino board. Refer to the [Getting Started](#) page for installation instructions.' On the right side of the page, there is a teal sidebar with the following links: 'Windows Installer, for Windows XP and up', 'Windows ZIP file for non admin install', 'Windows app Requires Win 8.1 or 10' (with a 'Get' button and the Windows logo), 'Mac OS X 10.7 Lion or newer', 'Linux 32 bits', 'Linux 64 bits', 'Linux ARM', 'Release Notes', 'Source Code', and 'Checksums (sha512)'.

In this software page, on the right side you can see the version of development software for different operating systems. ARDUINO has a powerful compatibility. You should download the software that is compatible with the operating system of your computer.

We will take **WINDOWS system** as an example here. There are also two options under Windows system, one is installed version, the other is non-installed version.

For simple installed version, first click **Windows Installer**, you will get the following page.

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

**Windows app** Requires Win 8.1 or 10  
 Get 

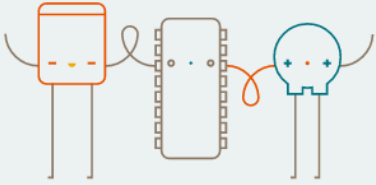
**Mac OS X** 10.7 Lion or newer

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

Release Notes  
 Source Code  
 Checksums (sha512)

## Contribute to the Arduino Software

Consider supporting the Arduino Software by contributing to its development. (US tax payers, please note this contribution is not tax deductible). [Learn more on how your contribution will be used.](#)



SINCE MARCH 2015, THE ARDUINO IDE HAS BEEN DOWNLOADED **24,353,248** TIMES. (IMPRESSIVE!) NO LONGER JUST FOR ARDUINO AND GENUINO BOARDS, HUNDREDS OF COMPANIES AROUND THE WORLD ARE USING THE IDE TO PROGRAM THEIR DEVICES, INCLUDING COMPATIBLES, CLONES, AND EVEN COUNTERFEITS. HELP ACCELERATE ITS DEVELOPMENT WITH A SMALL CONTRIBUTION! REMEMBER: OPEN SOURCE IS LOVE!

**\$3**   **\$5**   **\$10**   **\$25**   **\$50**   **OTHER**

JUST DOWNLOAD   **CONTRIBUTE & DOWNLOAD**

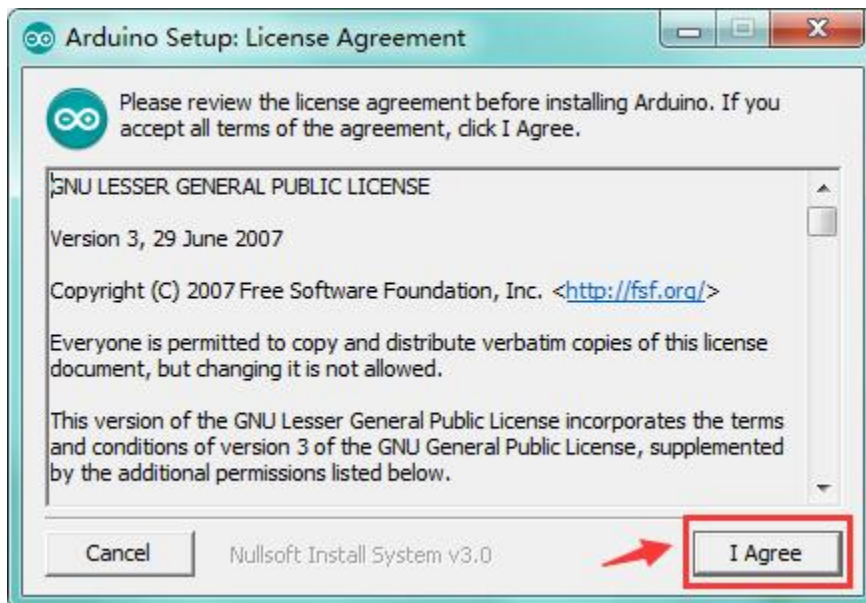
This way you just need to click JUST DOWNLOAD, then click the downloaded file to install it.

For non-installed version, first click Windows ZIP file, you will also get the pop-up interface as the above figure.

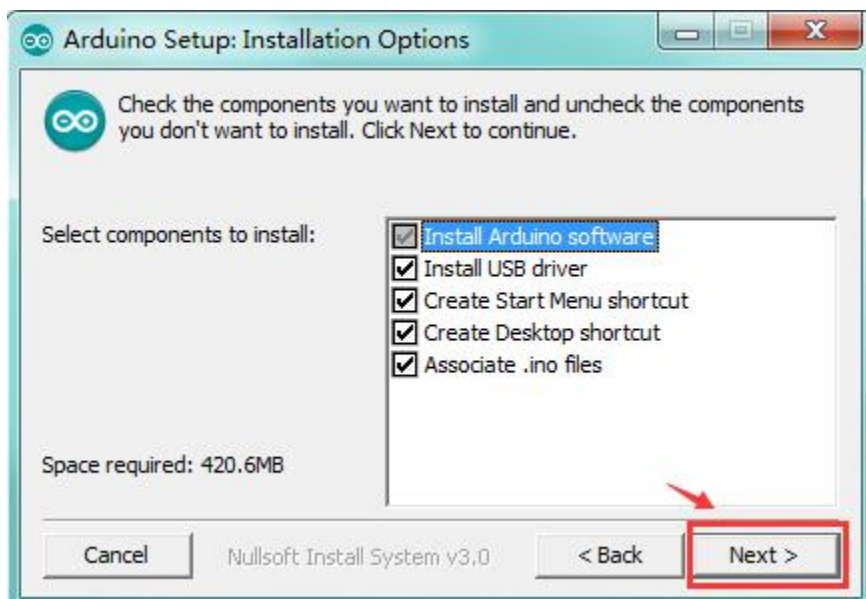
Click JUST DOWNLOAD, and when the ZIP file is downloaded well to your computer, you can directly unzip the file and click the icon of ARDUINO software to start it.

### Installing Arduino (Windows):

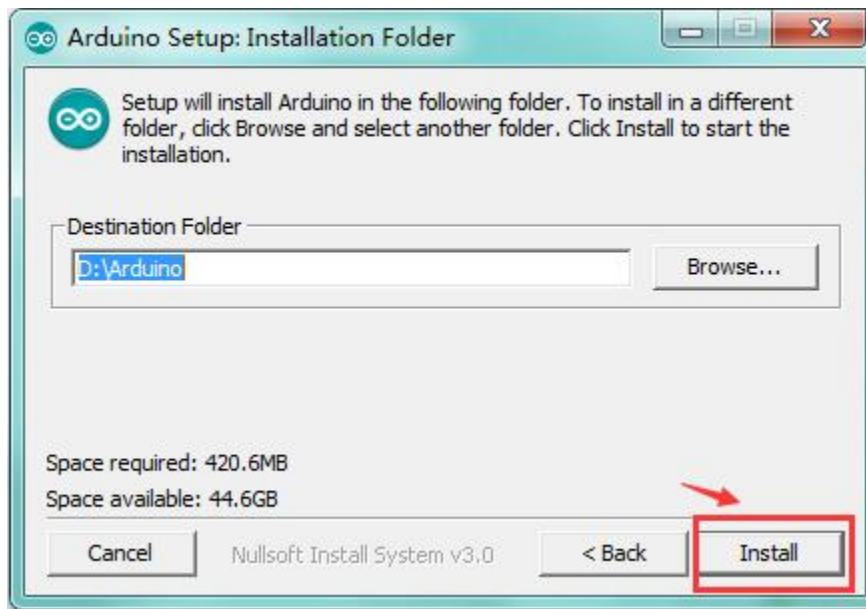
Install Arduino with the exe. Installation package downloaded well.



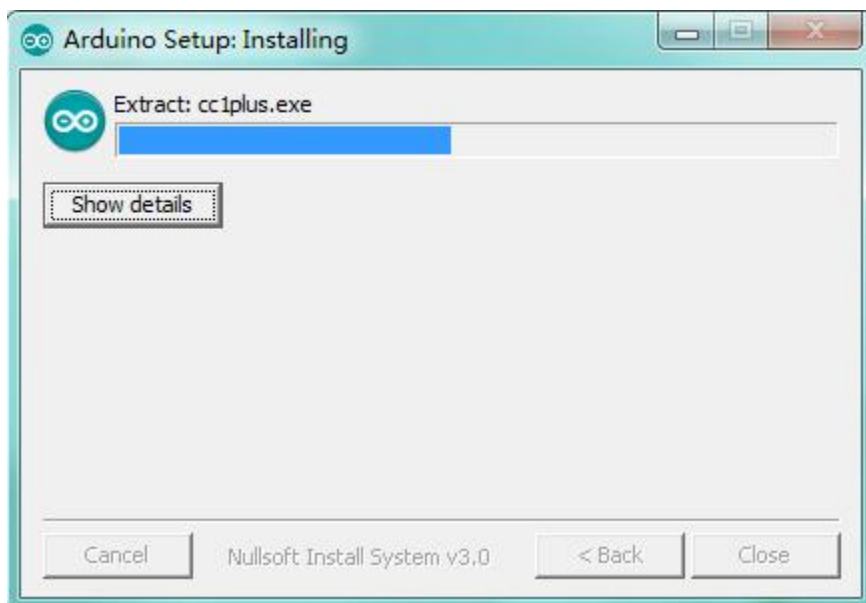
Click “I Agree” to see the following interface.



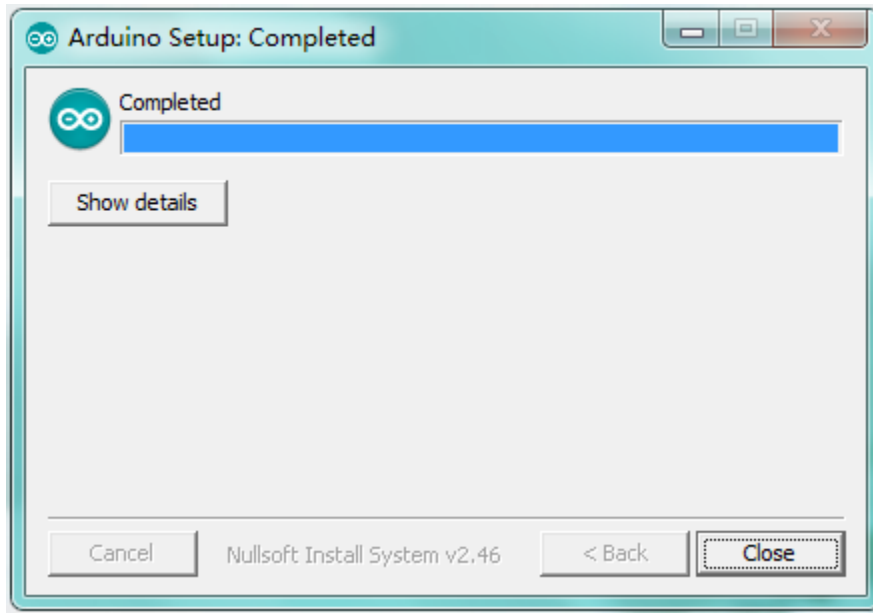
Click “Next”. Pop up the interface below.



You can press Browse... to choose an installation path or directly type in the directory you want. Then click “Install” to initiate installation.



Wait for the installing process, if appear the interface of Window Security, just continue to click Install to finish the installation.



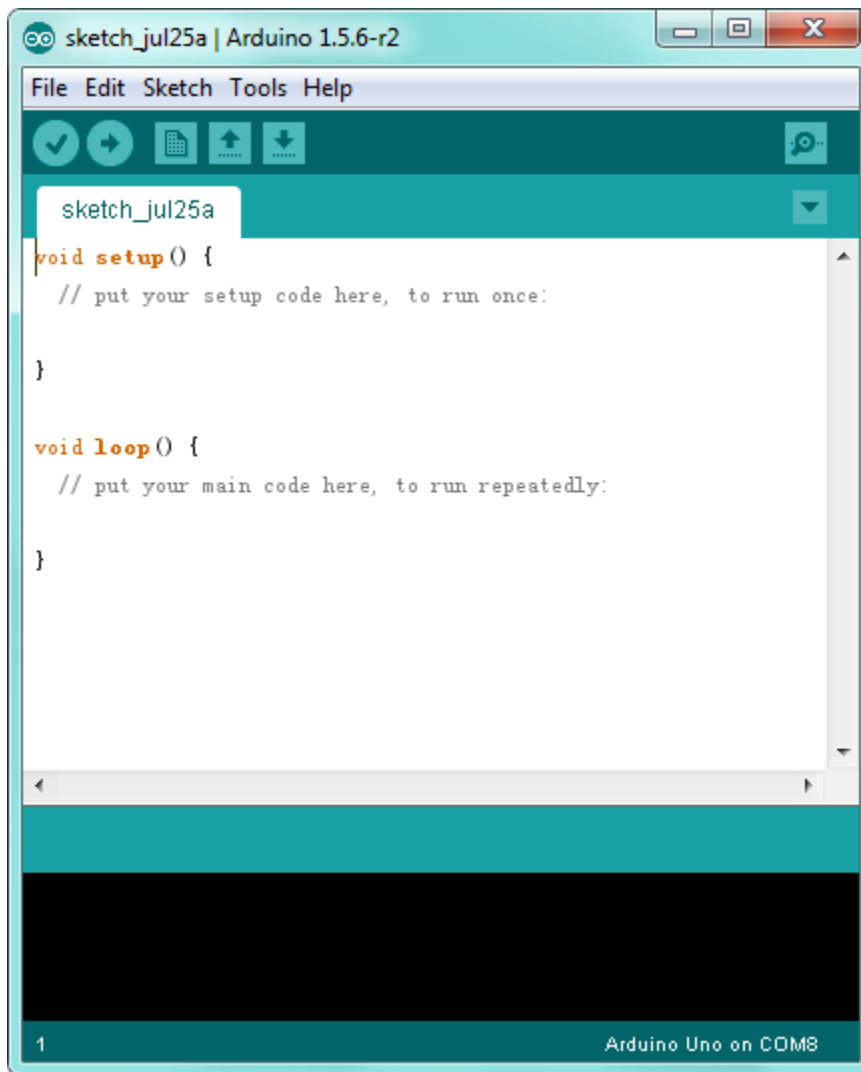
### Introduction for Arduino IDE Toolbar:

Double-click the icon of Arduino software downloaded, you will get the interface shown below.



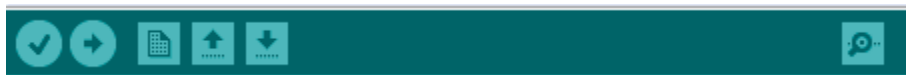
(Note: if the Arduino software loads in the wrong language, you can change it in the preferences dialog. See [the environment page](#) for details.)











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

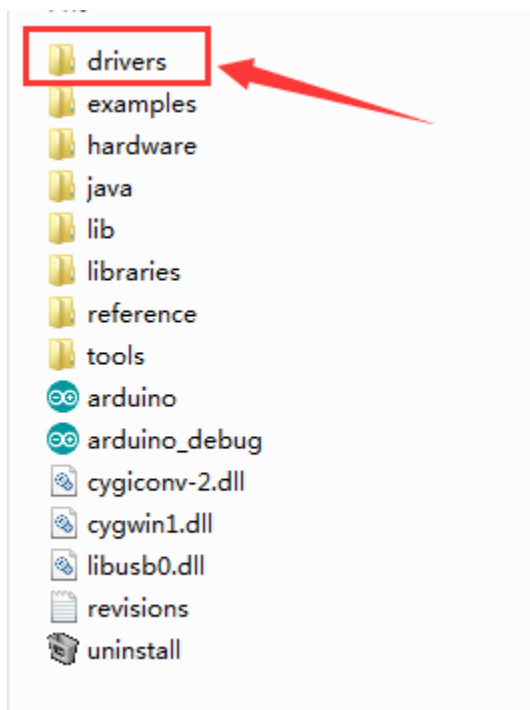
<http://wiki.keyestudio.com/index.php/File:IDE.png>



	<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

### Step2| Installing the Driver

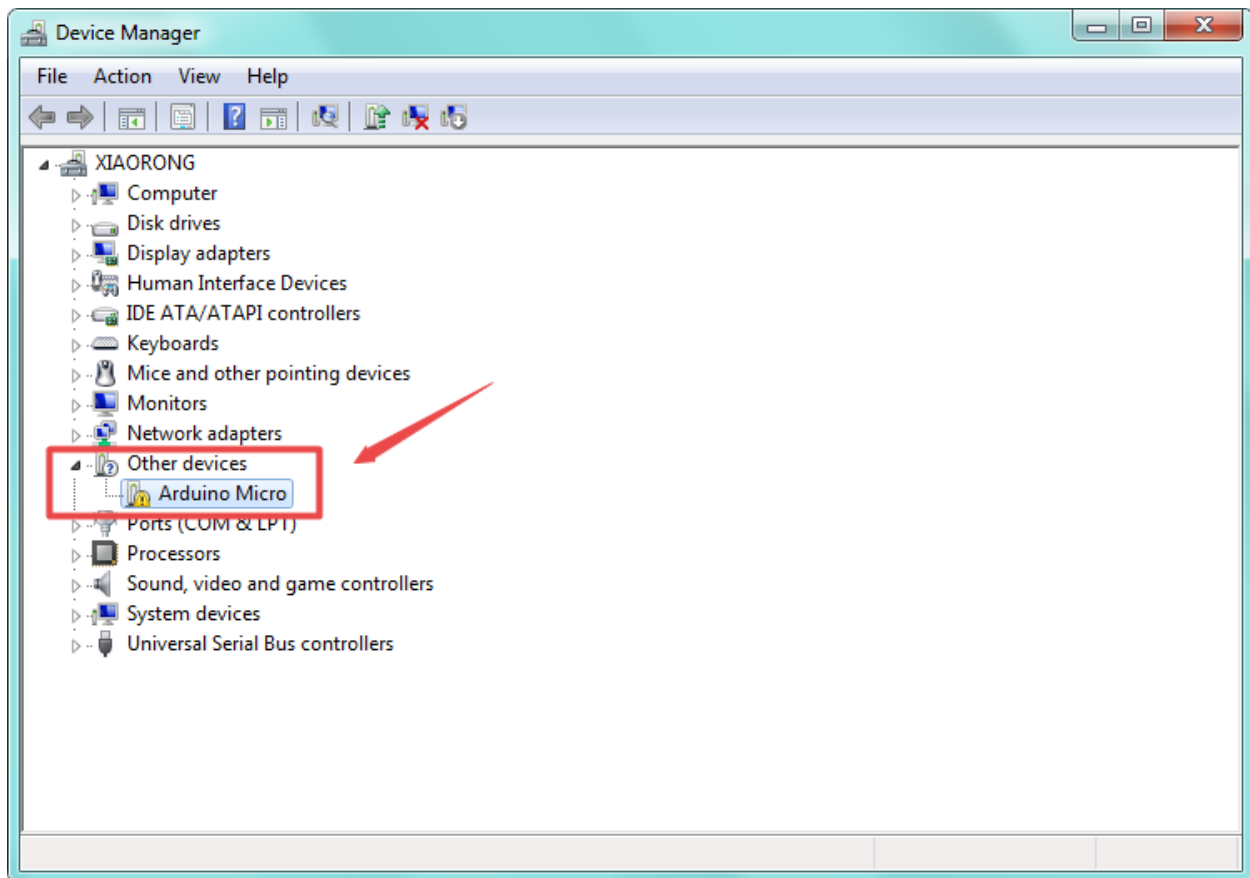
Installed well the Arduino, the next step is to install the driver. The Arduino folder contains both the Arduino program itself and the drivers that allow the Arduino to be connected to your computer with a USB cable.



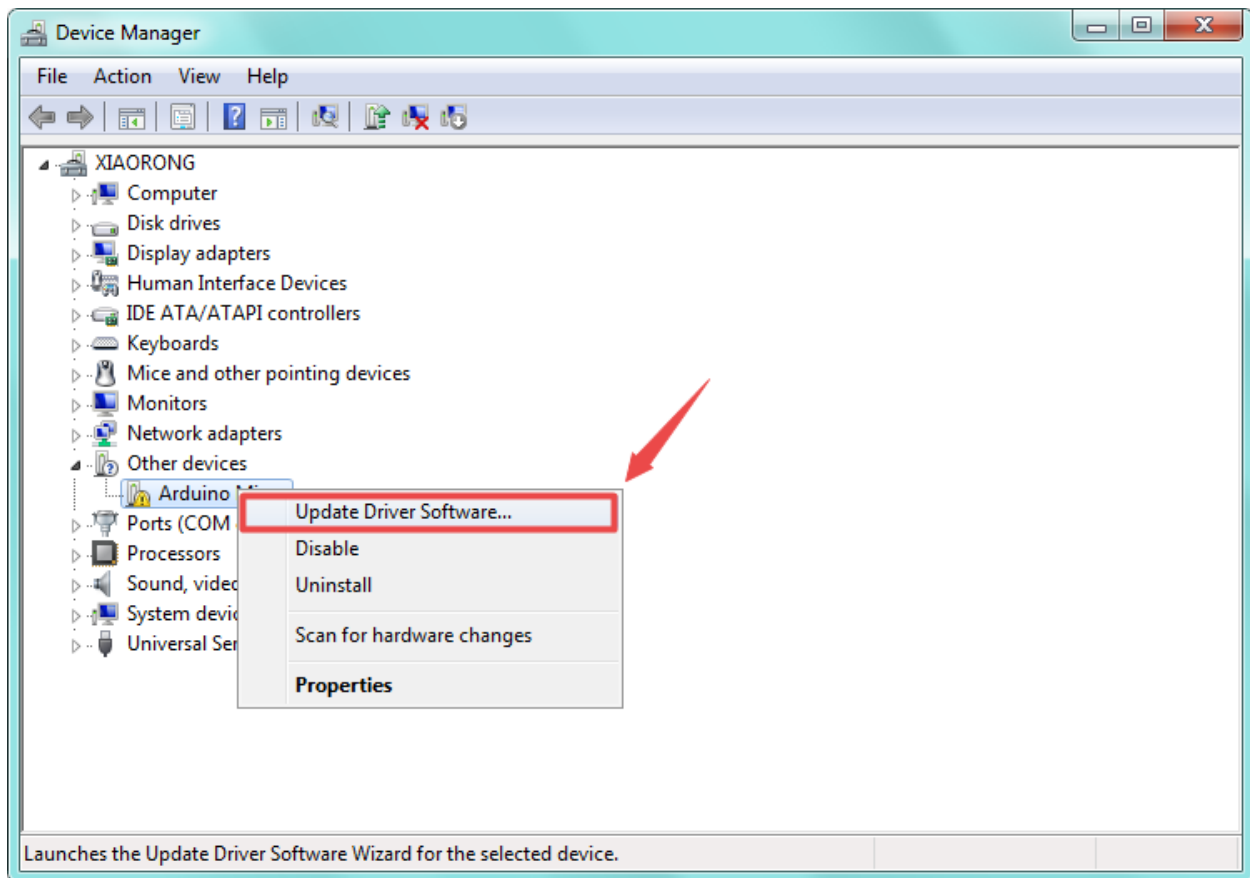
In different systems, the driver installation is similar. Here we start to install the driver on the Win7 system.

Plug one end of your USB cable into the keystudio PRO MICRO and the other into a USB socket on your computer.

When you connect the board to your computer at the first time, right click your “Computer” —>for “Properties”—> click the “Device manager”, under Other devices, you should see the “Arduino Micro”.



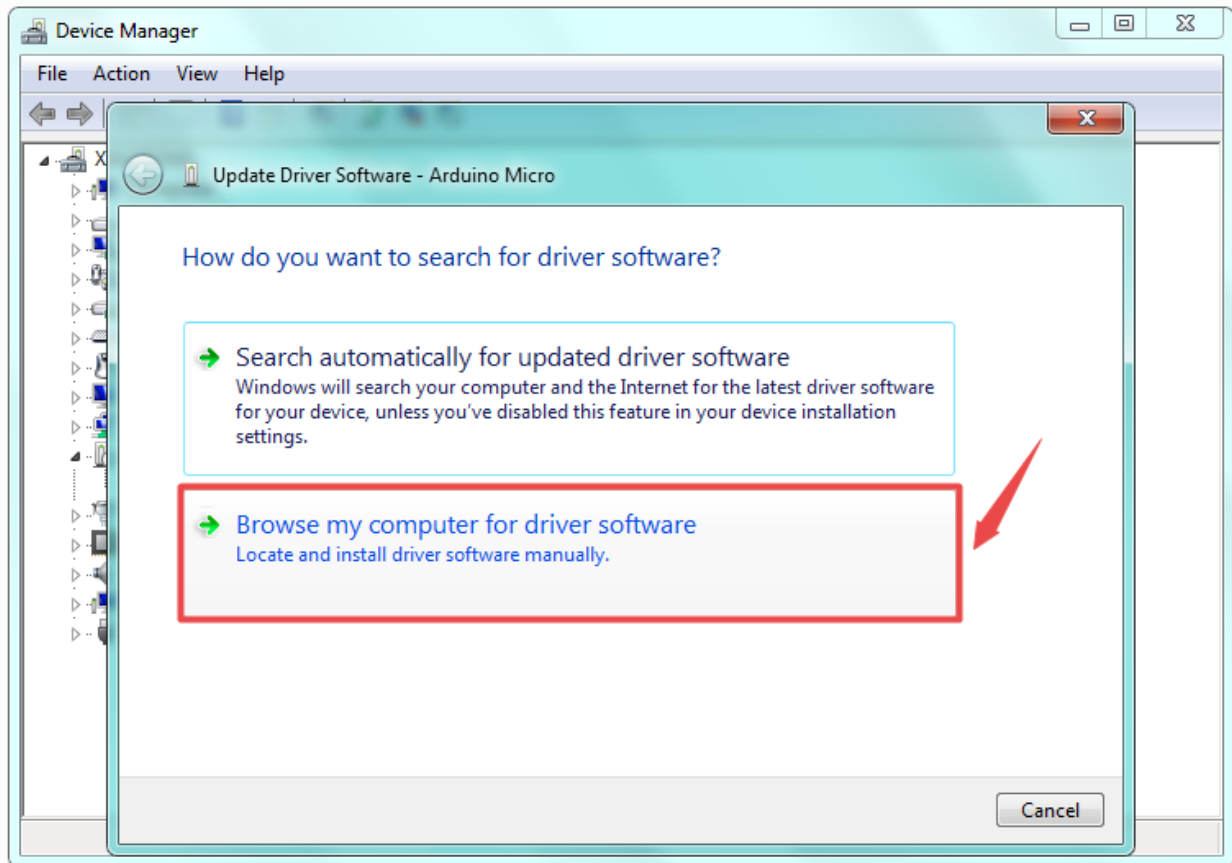
[http://wiki.keyestudio.com/index.php/File:Driver\\_2.png](http://wiki.keyestudio.com/index.php/File:Driver_2.png)



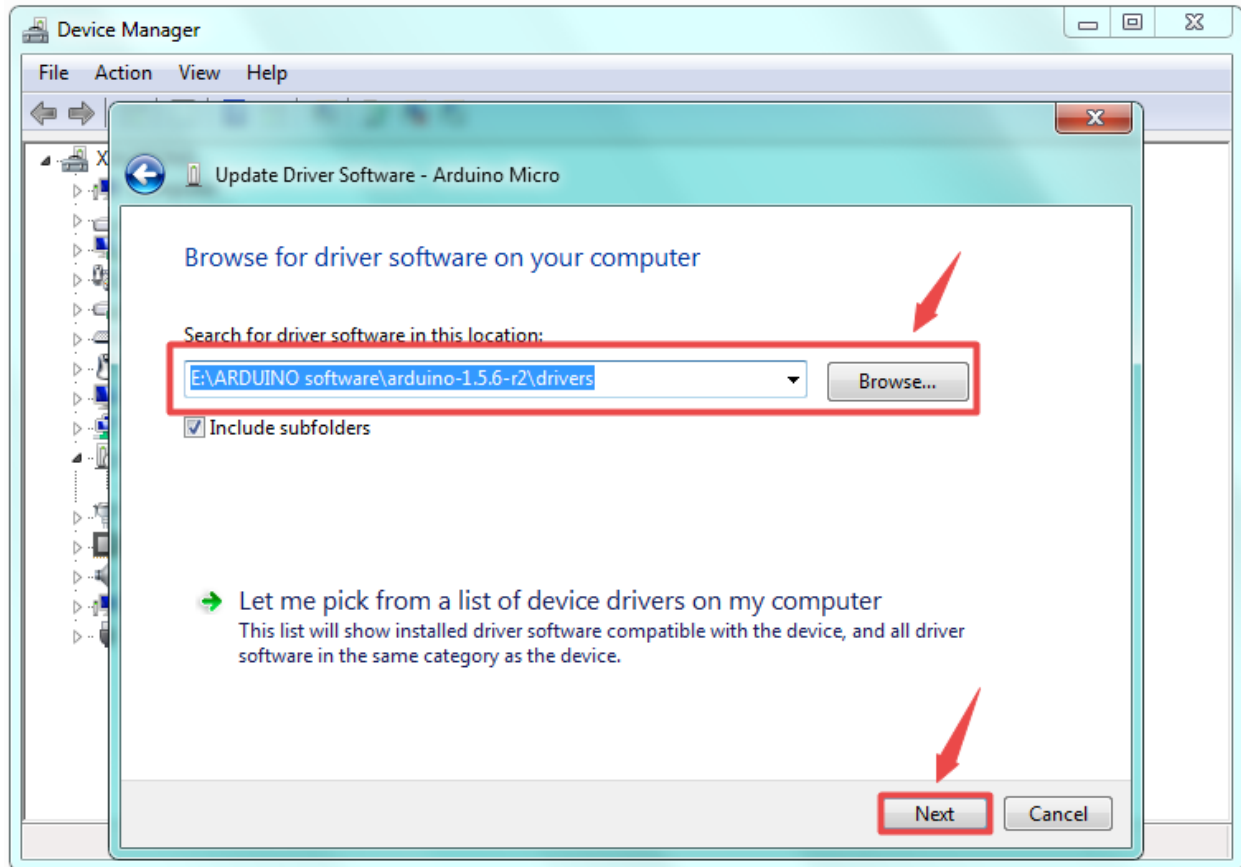
Then

right-click on the Arduino Micro and select the top menu option (Update Driver Software...) shown as the figure below.

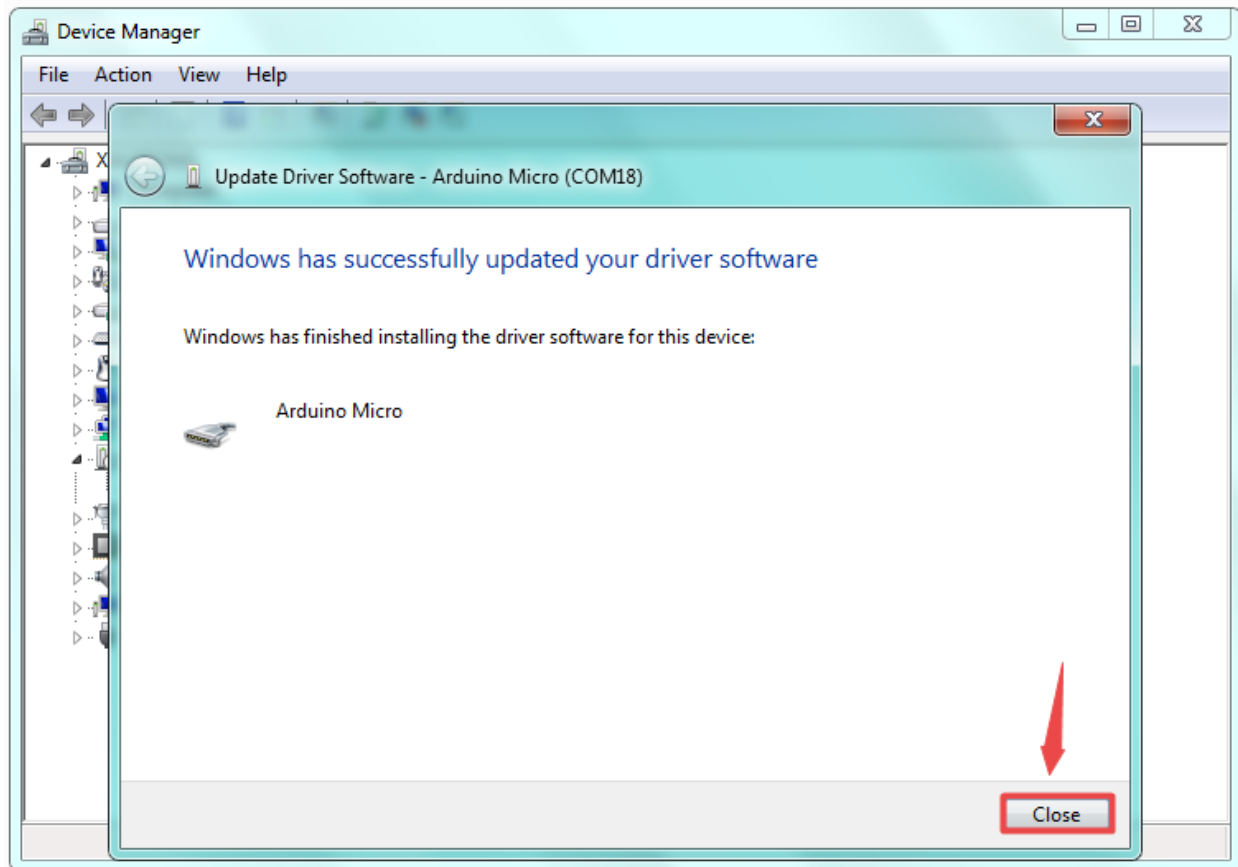
Then it will be prompted to either "Search Automatically for updated driver software" or "Browse my computer for driver software". Shown as below. In this page, select "Browse my computer for driver software".



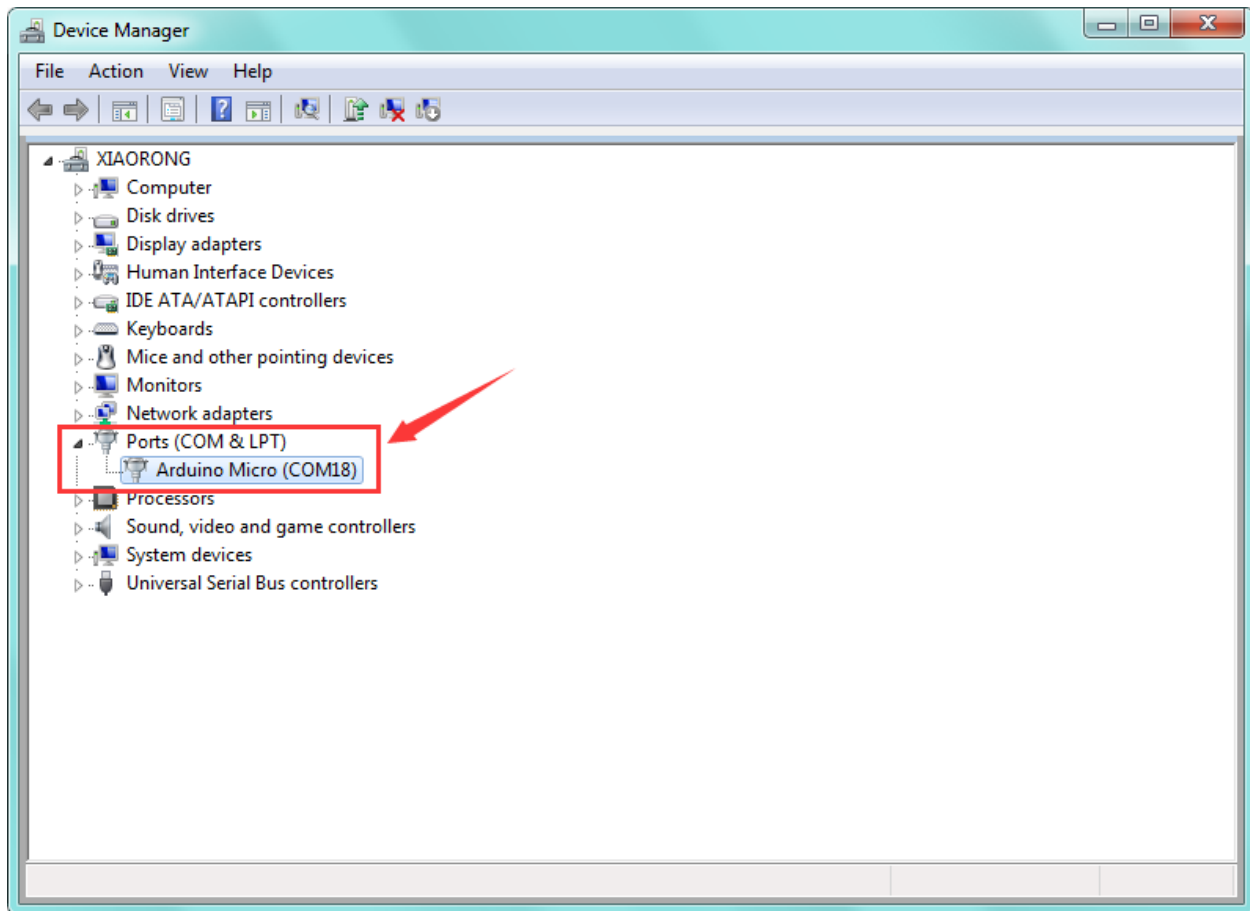
After that, select the option to browse and navigate to the “drivers” folder.



Once the software has been installed, you will get a confirmation message. Installation completed, click “Close”.  
[http://wiki.keyestudio.com/index.php/File:Driver\\_6.png](http://wiki.keyestudio.com/index.php/File:Driver_6.png)



Up to now, the driver is installed well. Then you can right click "Computer" —>"Properties"—>"Device manager", you should see the device as the figure shown below.



Step3| Connect the board

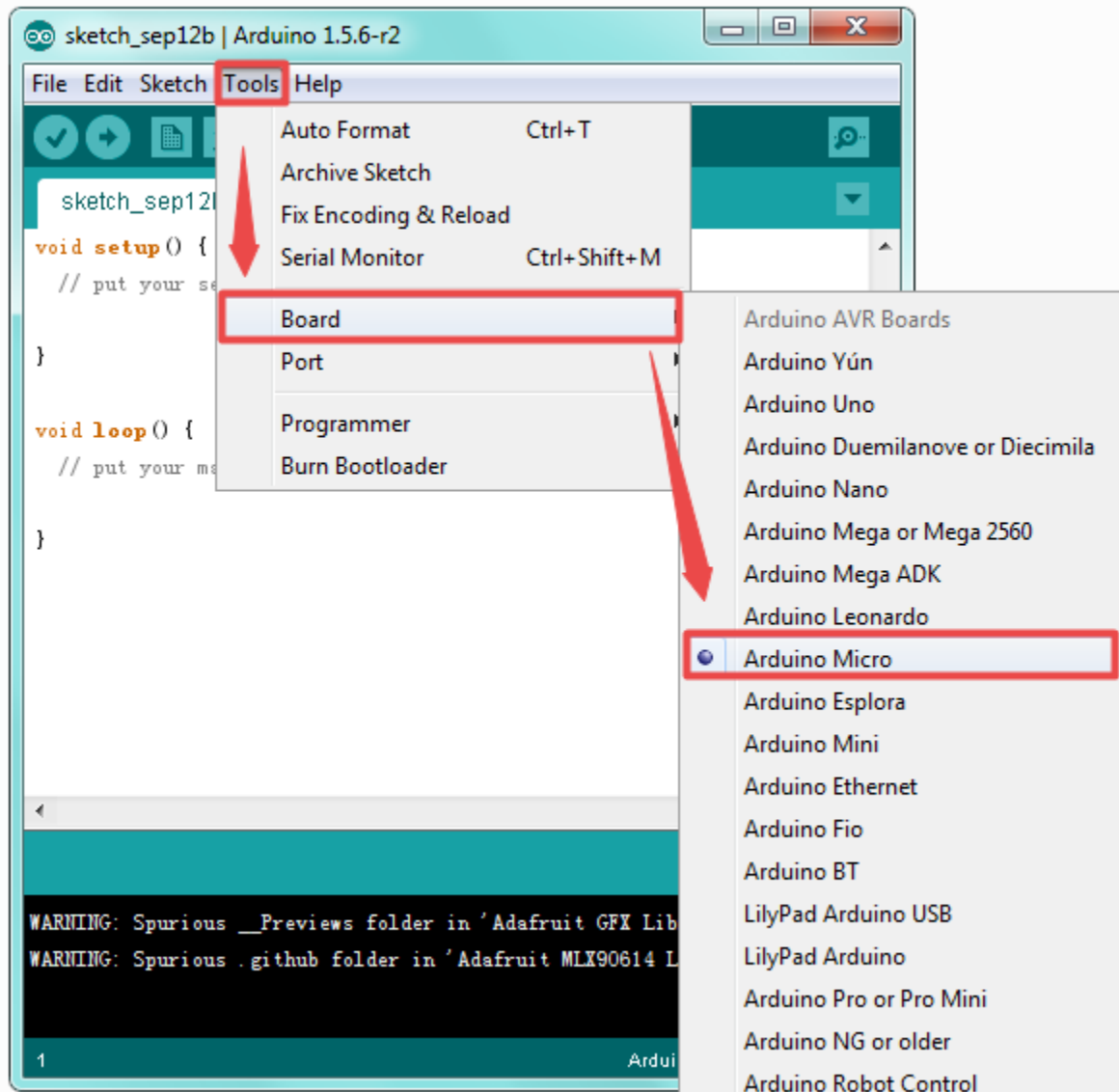
Connect the Pro Micro board to your computer using the USB cable. The power LED should go on.

Step4| Select the Arduino Board

Open the Arduino IDE, you'll need to click the "Tools", then select the Board that corresponds to your Arduino.

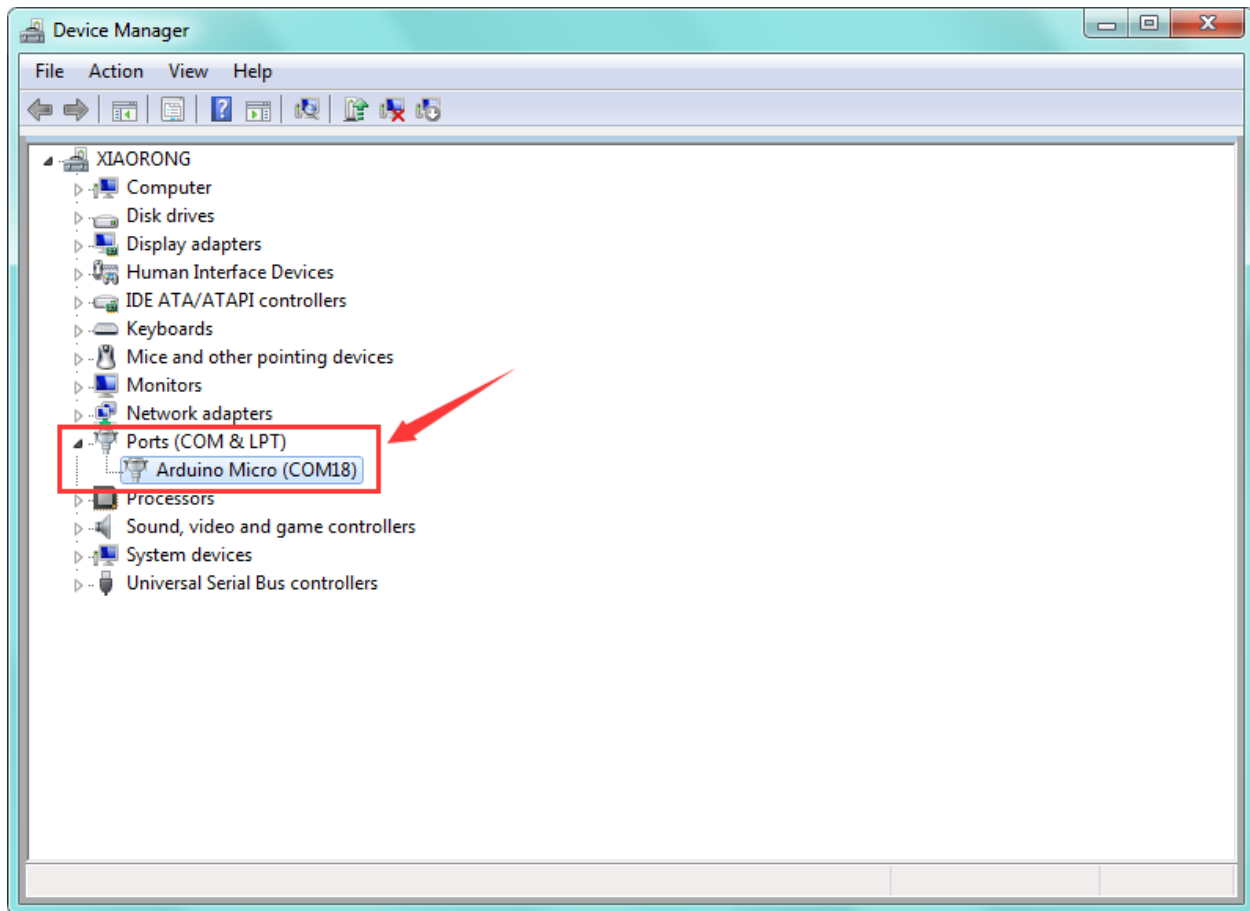


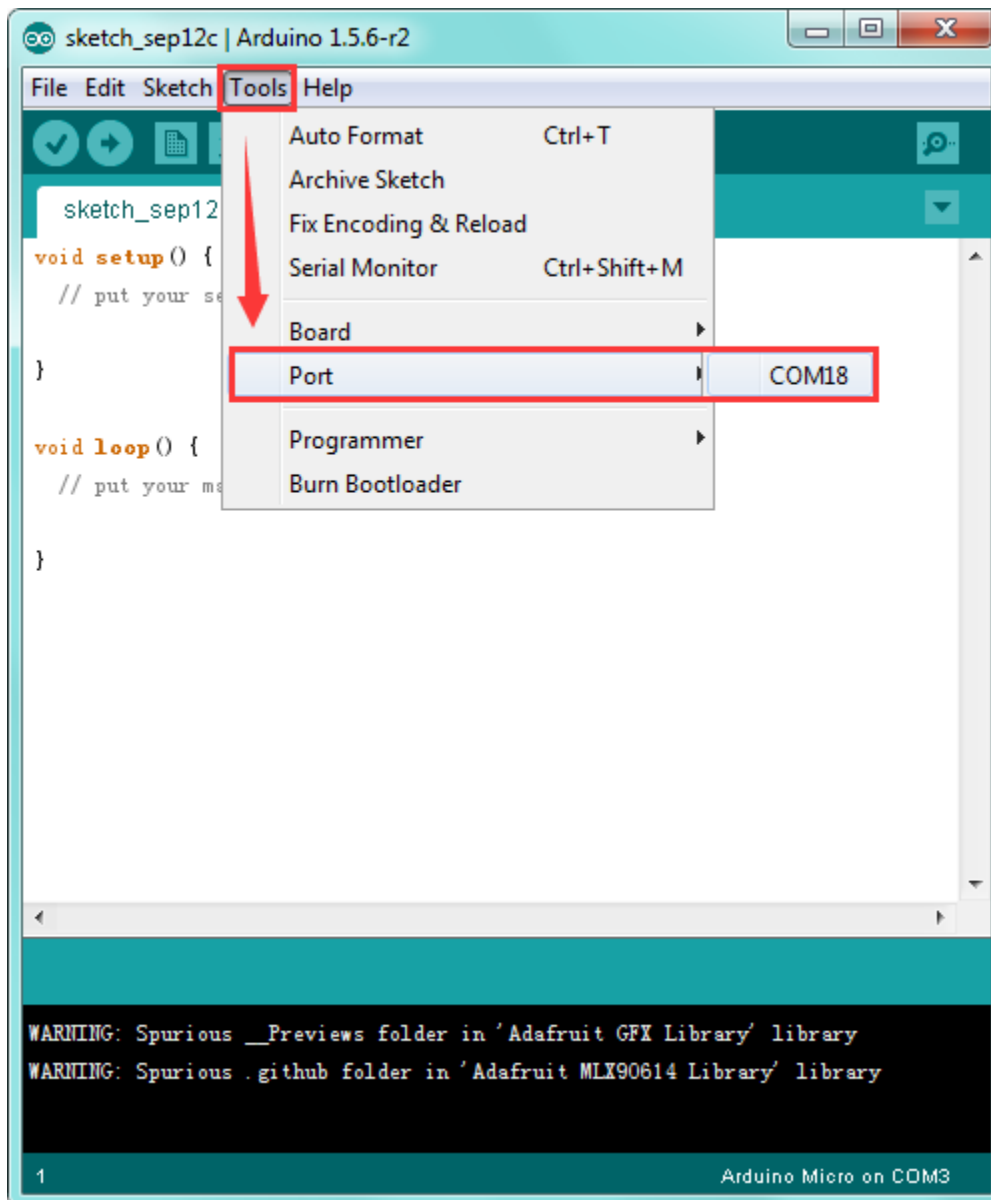




Step5| Select your serial port

Select the serial device of the Arduino board from the **Tools | Serial Port menu**. **Note:** to avoid errors, the COM Port should keep the same as the Ports shown on Device Manager.





Step6| Upload the Code

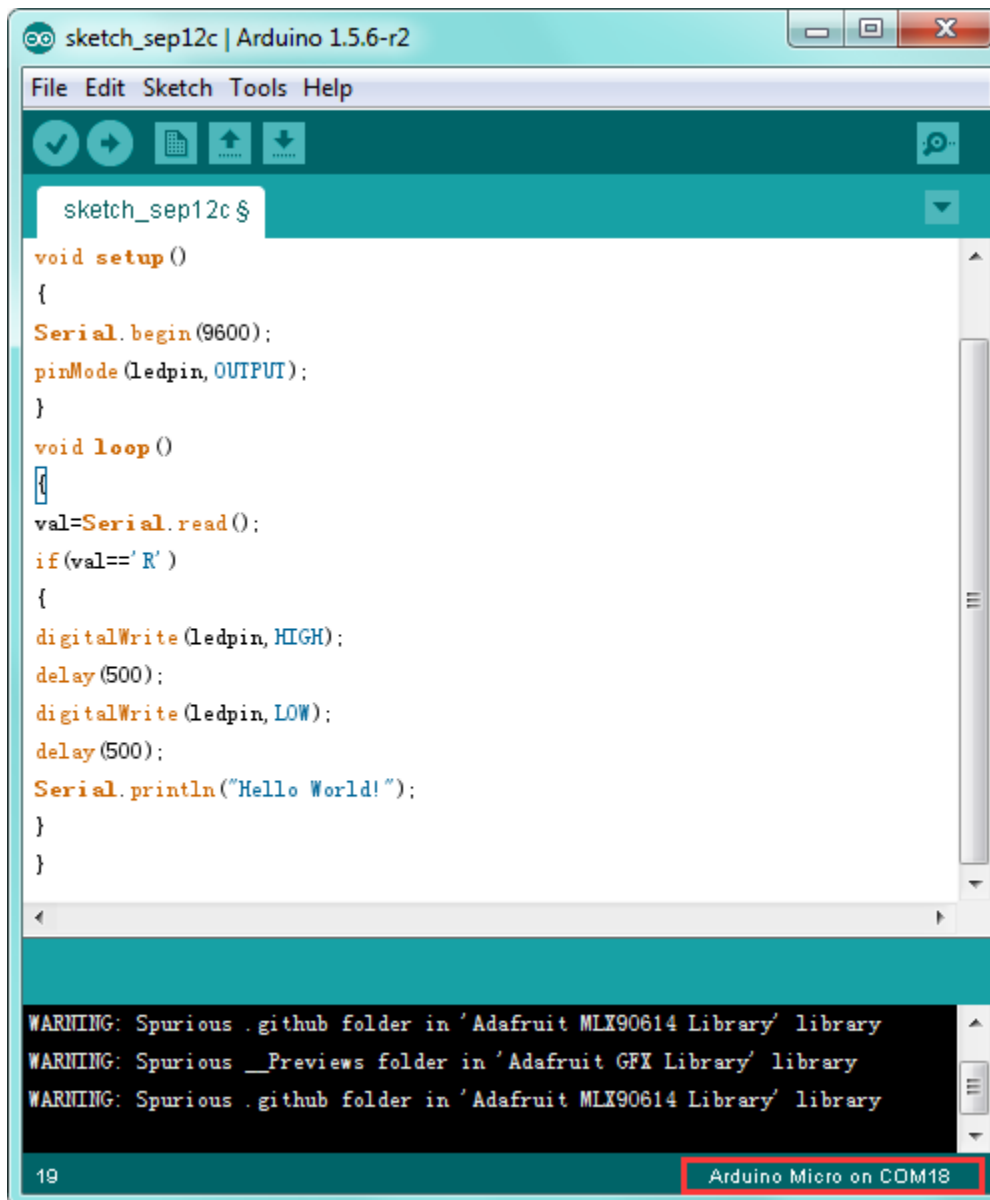
Below is an example code for displaying the Hello World!

Copy and paste the code to the Arduino environment IDE.

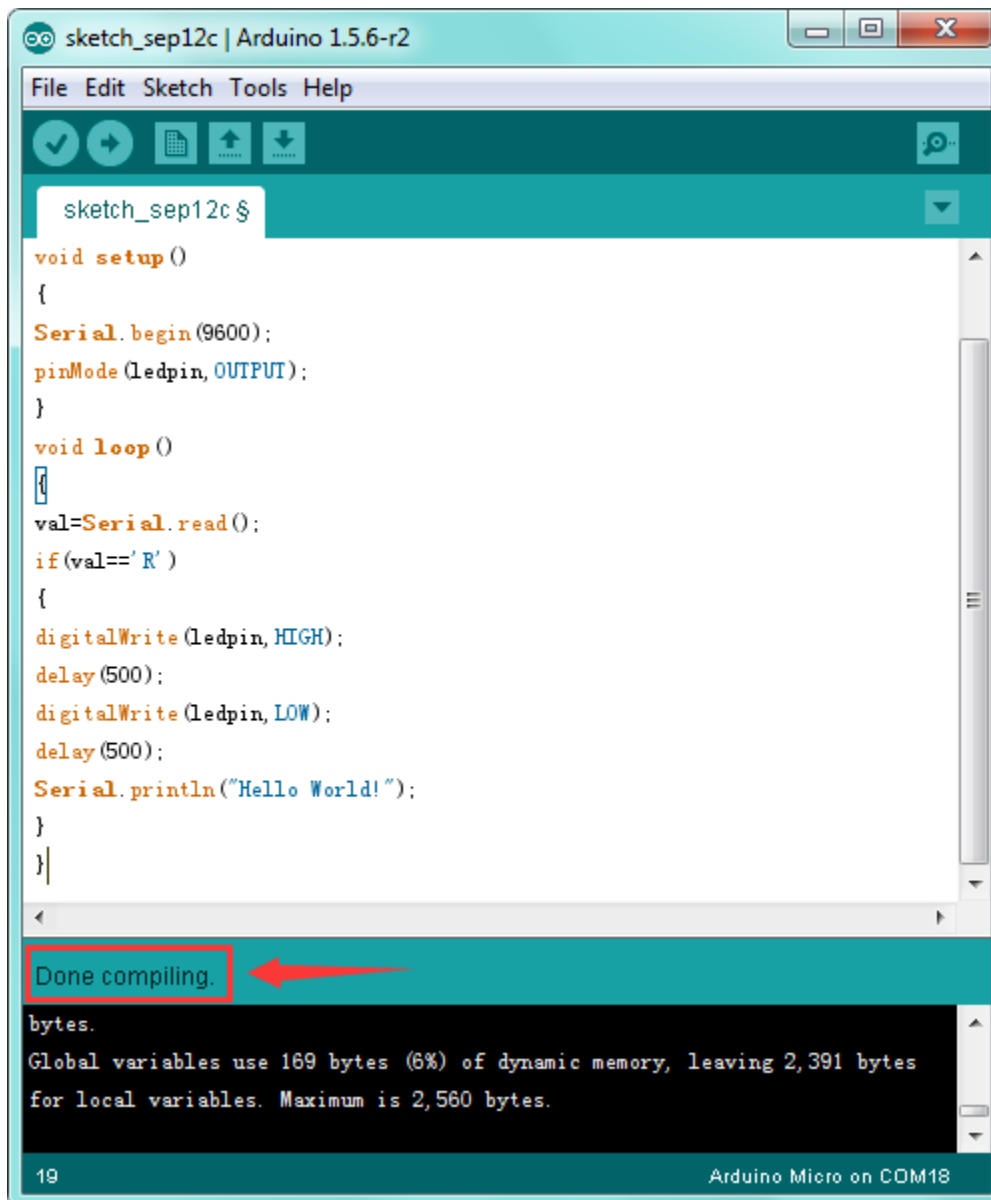
////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////

```
int val;
int ledpin=13;
void setup()
{
  Serial.begin(9600);
  pinMode(ledpin,OUTPUT);
}
```

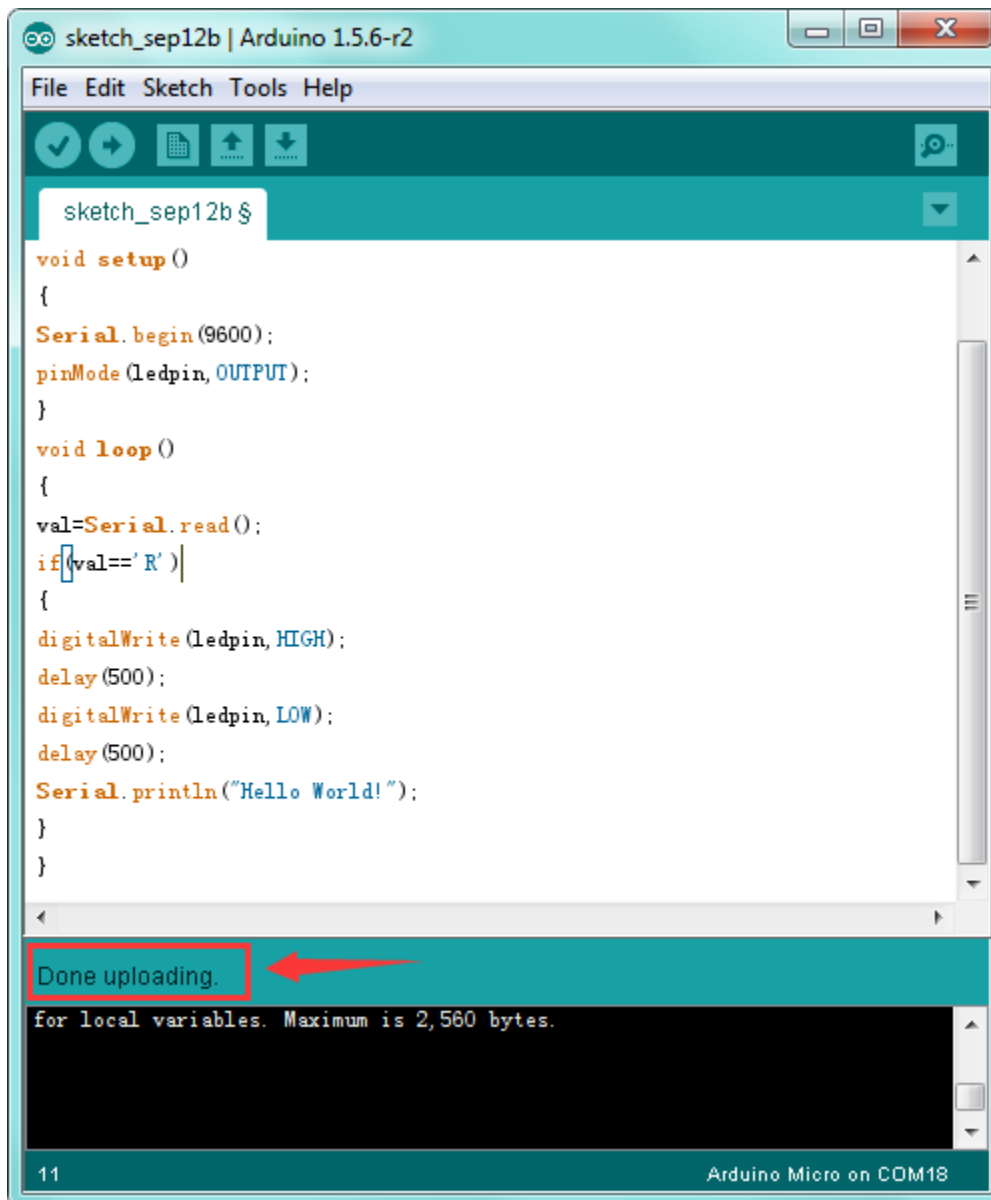
```
void loop()
{
  val=Serial.read();
  if(val=='R')
  {
    digitalWrite(ledpin,HIGH);
    delay(500);
    digitalWrite(ledpin,LOW);
    delay(500);
    Serial.println("Hello World!");
  }
}
////////////////////////////////////////////////////////////////
```



Then click verify button to check the errors. If compiling successfully, the message “Done compiling.” will appear in the status bar.

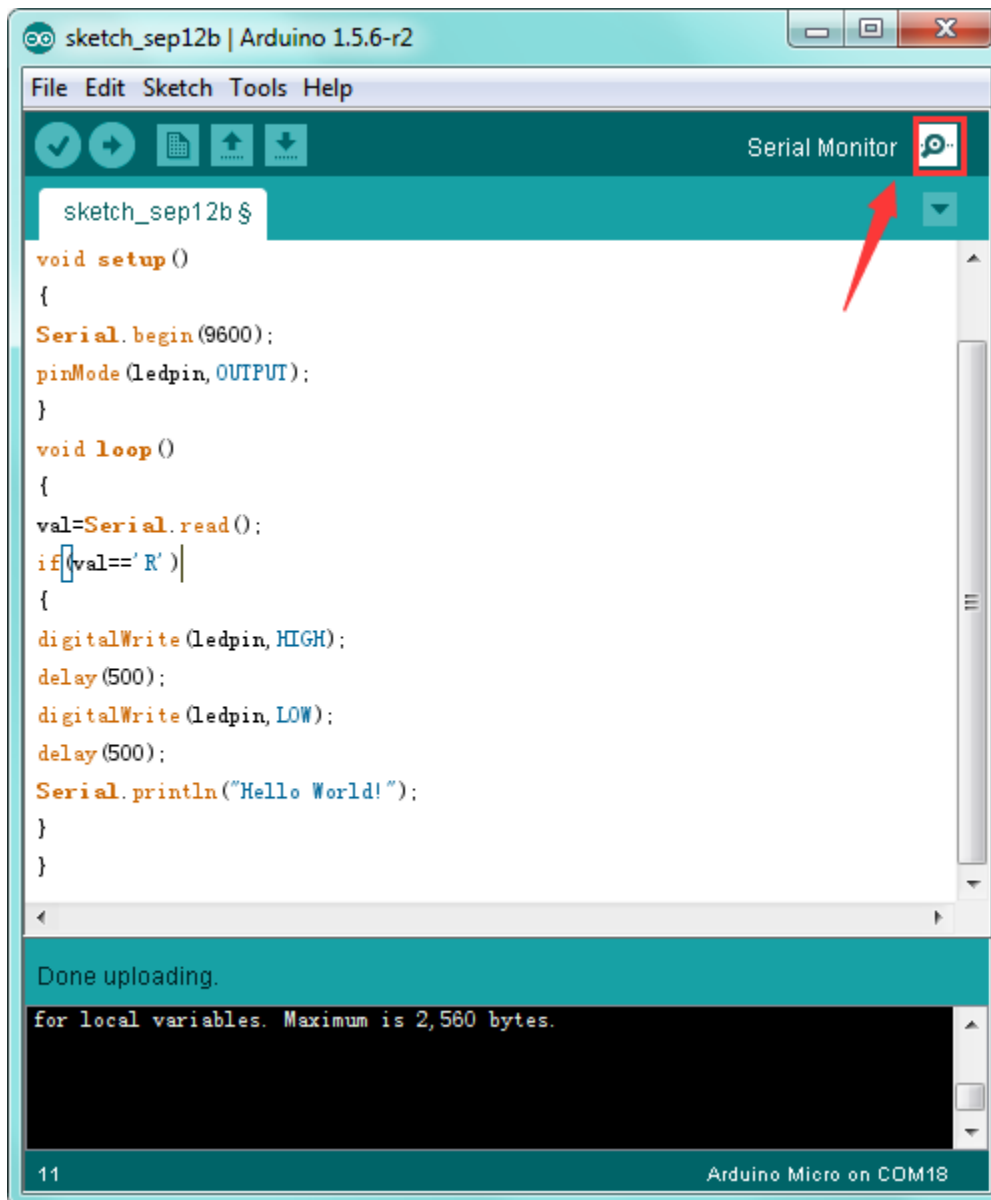


After that, click the “Upload” button to upload the code. If the upload is successful, the message “Done uploading.” will appear in the status bar.



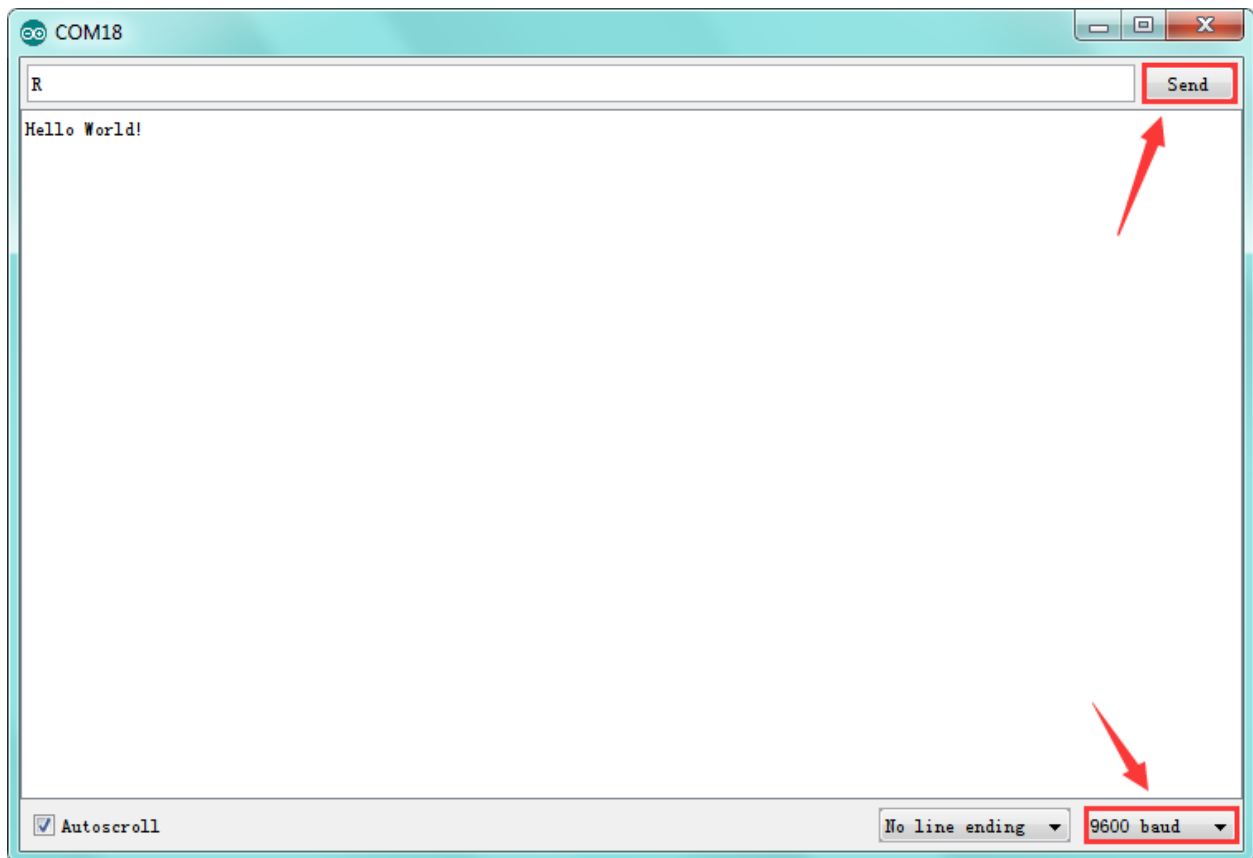
Step7| Open the Serial Monitor

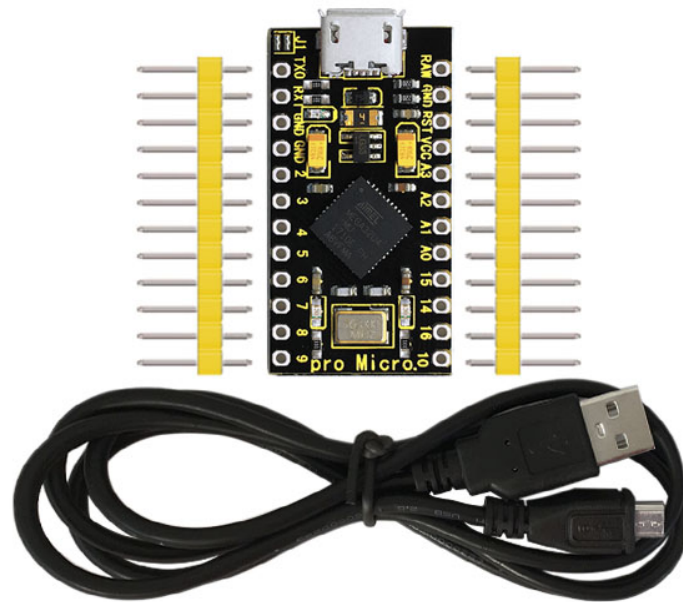
After that, click the button to open the serial monitor.



Then set the baud rate to 9600, enter an “R” and click Send, that is, the computer will send the character R. When the board receives it, you should see the RX led on the board flash once, and then D13 led flash once; when keystudio PRO MICRO successfully sends “Hello World!” to the computer, you should see the “Hello World!” is showed on the monitor, and TX led on the PRO MICRO board flash once.







Package

Includes:

- keyestudio PRO MICRO\* 1pcs
- Black micro USB cable 1m \* 1pcs
- Yellow Pin headers \* 2pcs

Resource Links:

Download test code and driver:

<https://fs.keyestudio.com/KS0249>

You might also want to look at:

[the reference](#) for the Arduino language;

Download ARDUINO Software:

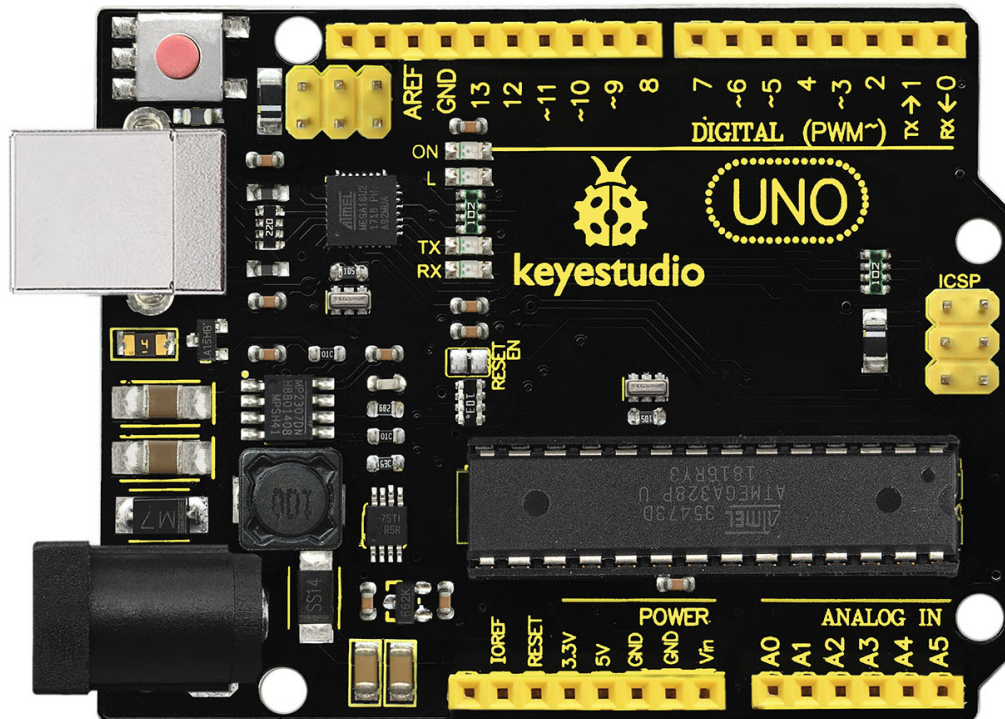
<https://www.arduino.cc/en/Main/OldSoftwareReleases\1.5.x>

Troubleshooting:

If you have problems, please see the [troubleshooting suggestions](#).



## KS0341 KEYESTUDIO UNO COMPATIBLE ADVANCED DEVELOPMENT BOARD



Introduction:

The keystudio UNO compatible board Advanced is a microcontroller board based on the ATmega328P ([datasheet](#)), fully compatible with [keystudio UNO R3 board](#) and ARDUINO UNO REV3.

It has 14 digital input/output pins (of which 6 can be used as PWM outputs), 6 analog inputs, a 16 MHz quartz crystal, a USB connection, a power jack, 2 ICSP headers and a reset button.

Note that the two ICSP headers are separately used to program the firmware to ATMEGA16U2-MU and ATMEGA328P-PU, but generally the two chips have been programmed well.

It contains everything needed to support the microcontroller; simply connect it to a computer with a USB cable or power it via an external DC power jack (DC 7-12V) or via female headers Vin/ GND(DC 7-12V) to get started.

**Note:** the most important difference between this keystudio UNO compatible board Advanced and keystudio UNO R3 board is the voltage regulator chip used in the board.

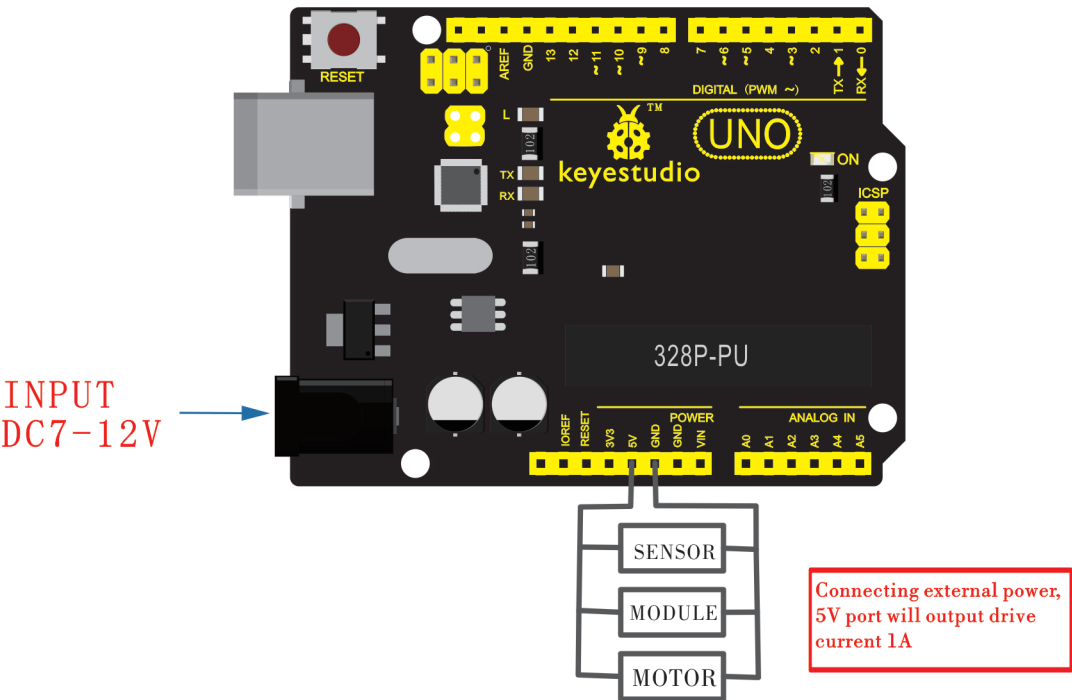
The voltage regulator chip used in keystudio UNO R3 board is NSP1117. When connect an external power, output 5V, the drive current is 1A.

However, keystudio UNO compatible board Advanced features the voltage-regulator chip MP2307DN. When connect the external power, output 5V, the drive current can be 2A.

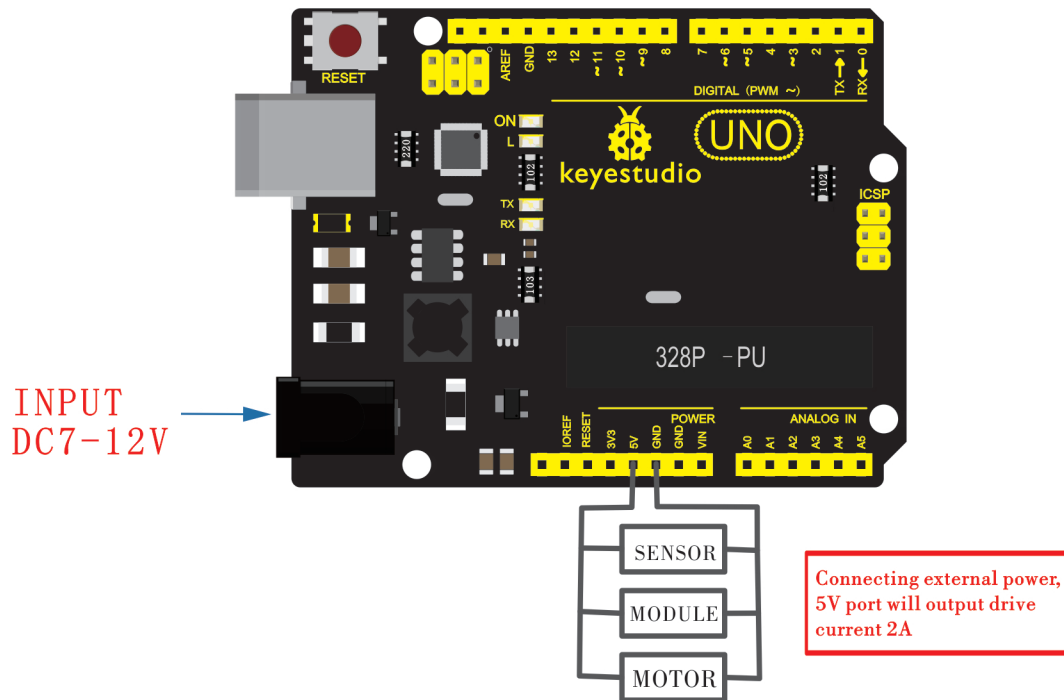
The UNO is the best board to get started with electronics and coding. If this is your first experience tinkering with the platform, the UNO is the most robust board you can start playing with.

	keystudio UNO compatible board Advanced	keystudio UNO R3 board
<b>voltage-regulator chip</b>	MP2307DN	NSP1117
<b>Power Output</b>	5V	5V
<b>Drive Current (external power)</b>	2A	1A
<b>Drive Current (USB power)</b>	0.5A	0.5A

keystudio UNO R3 board:



keystudio UNO compatible board Advanced:

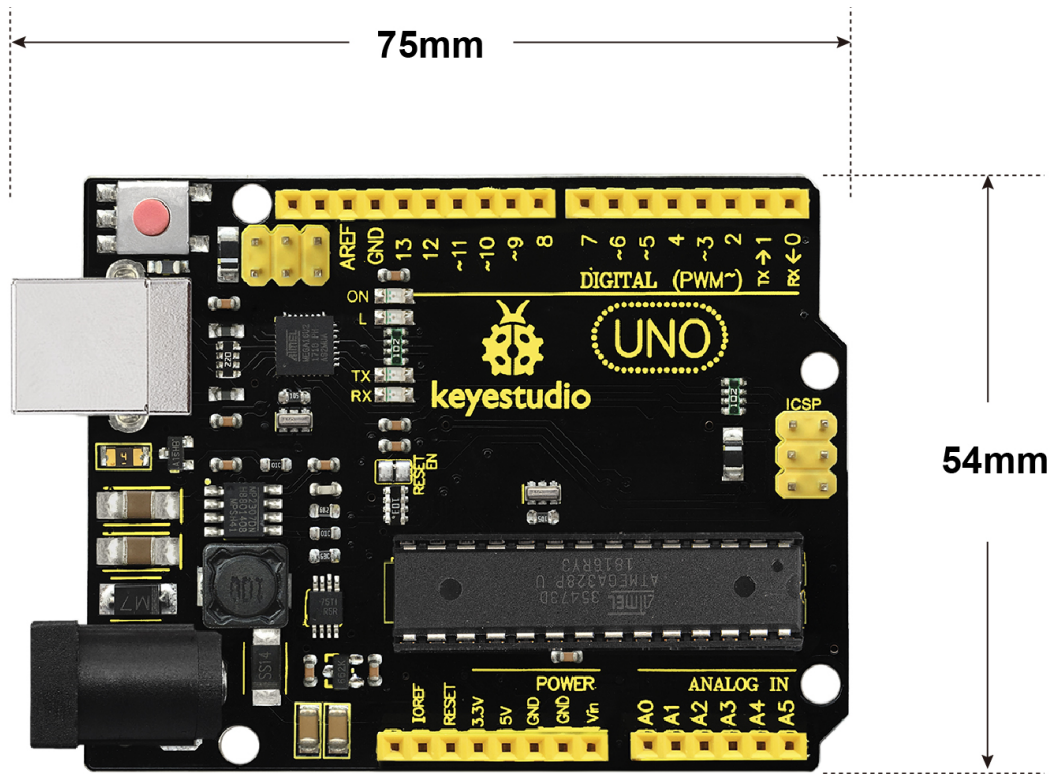


TECH SPECS:

<b>Microcontroller</b>	ATmega328P-PU
<b>Operating Voltage</b>	5V
<b>Input Voltage (recommended)</b>	DC7-12V
<b>Digital I/O Pins</b>	14 (D0-D13) (of which 6 provide PWM output)
<b>PWM Digital I/O Pins</b>	6 (D3, D5, D6, D9, D10, D11)
<b>Analog Input Pins</b>	6 (A0-A5)
<b>DC Current per I/O Pin</b>	20 mA
<b>DC Current for 3.3V Pin</b>	50 mA
<b>Flash Memory</b>	32 KB (ATmega328P-PU) of which 0.5 KB used by bootloader
<b>SRAM</b>	2 KB (ATmega328P-PU)
<b>EEPROM</b>	1 KB (ATmega328P-PU)
<b>Clock Speed</b>	16 MHz
<b>LED_BUILTIN</b>	D13

Details:

- Dimensions: 75mm x 54mm x 15mm



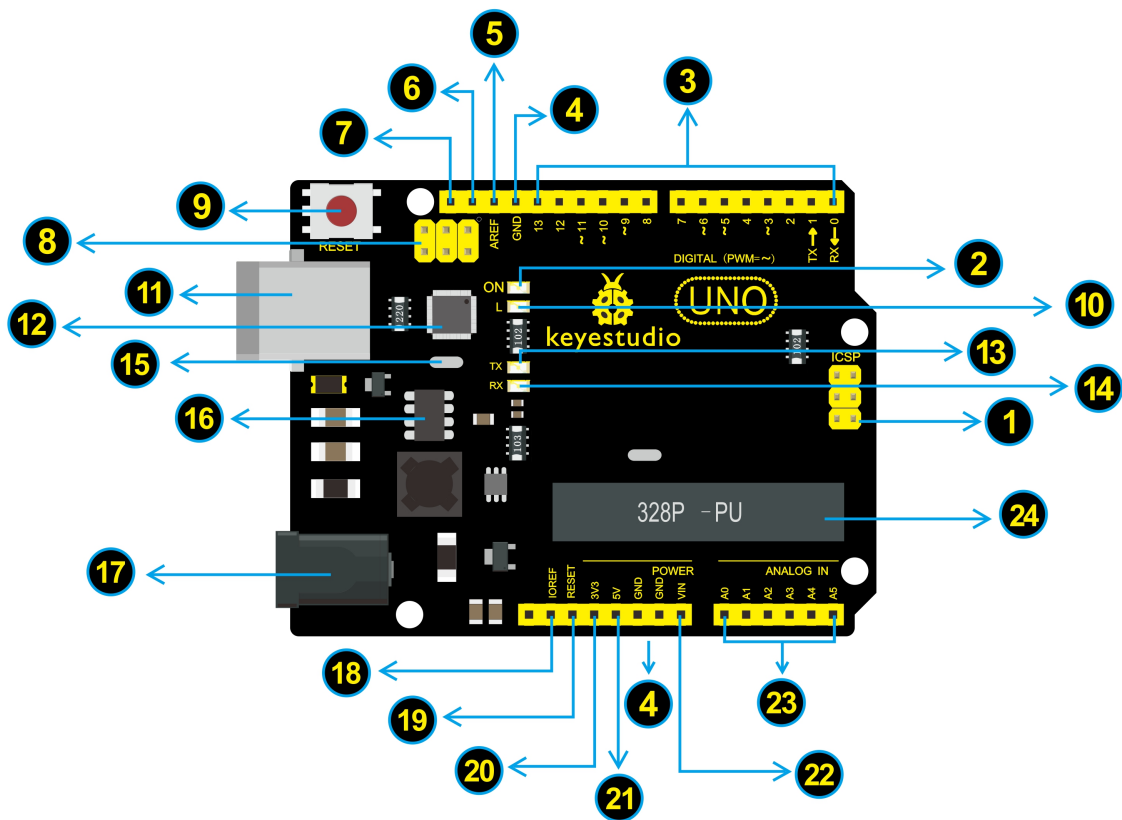
•  
27g

H:15mm      N.W.:27g      Weight:

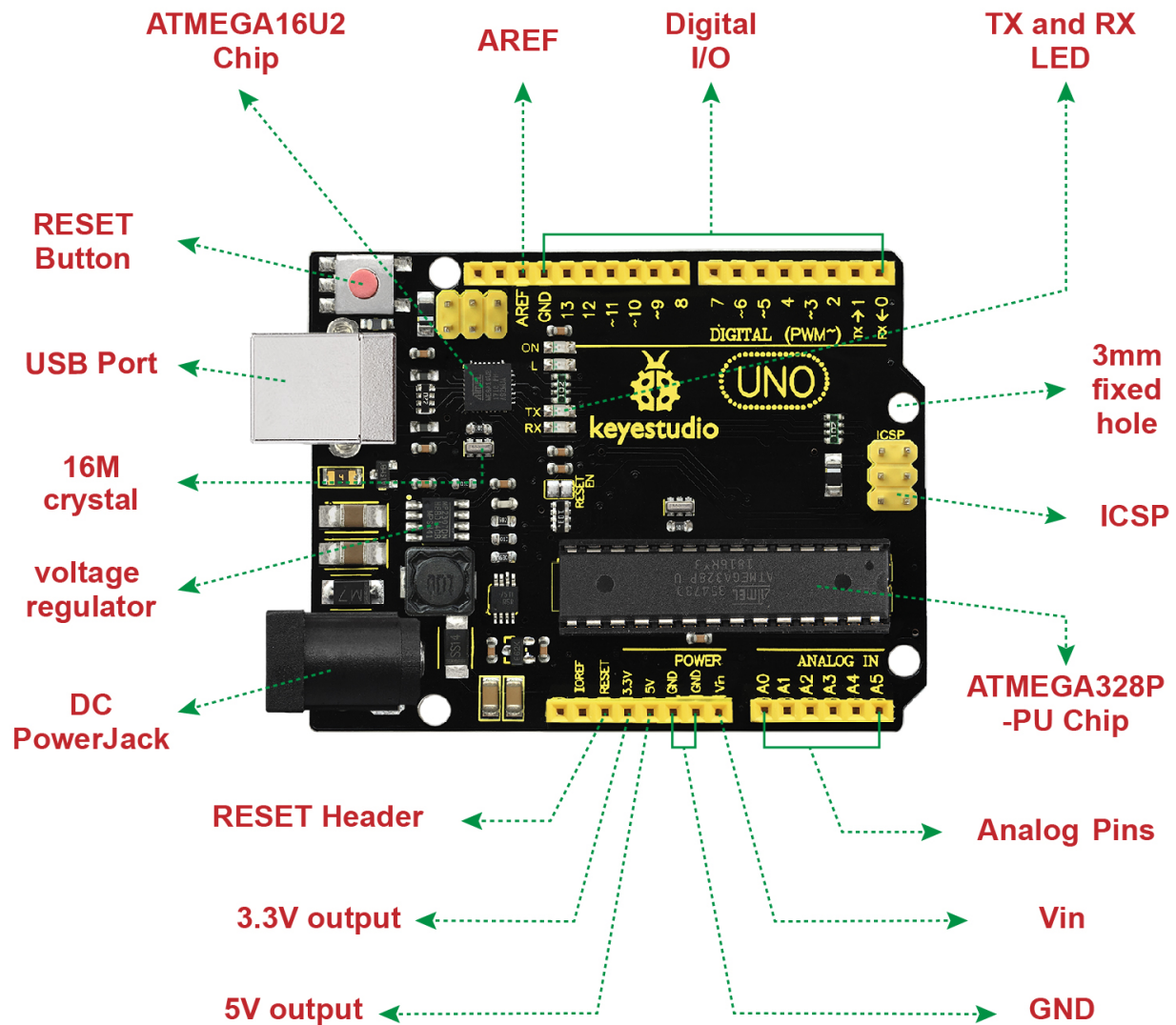
Element and Interfaces:

Here is an explanation of what every element and interface of the board does:





1	<b>ICSP (In-Circuit Serial Programming) Header</b> In most case, ICSP is the AVR, an Arduino micro-program header consisting of MOSI, MISO, SCK, RESET, VCC, and GND. It is often called the SPI (serial peripheral interface) and can be considered an “extension” of the output. In fact, slave the output devices under the SPI bus host. When connecting to PC, program the firmware to ATMEGA328P-PU.
2	<b>Power LED Indicator</b> Powering the Arduino, LED on means that your circuit board is correctly powered on. If LED is off, connection is wrong.
3	<b>Digital I/O</b> Arduino UNO has 14 digital input/output pins (of which 6 can be used as PWM outputs). These pins can be configured as digital input pin to read the logic value (0 or 1). Or used as digital output pin to drive different modules like LED, relay, etc. The pin labeled “” can be used to generate PWM.
4	<b>GND ( Ground pin headers)</b> Used for circuit ground
5	<b>AREF</b> Reference voltage (0-5V) for analog inputs. Used with <code>analogReference()</code> .
6	<b>SDA</b> IIC communication pin
7	<b>SCL</b> IIC communication pin
8	<b>ICSP (In-Circuit Serial Programming) Header</b> In most case, ICSP is the AVR, an Arduino micro-program header consisting of MOSI, MISO, SCK, RESET, VCC, and GND. Connected to ATMEGA 16U2-MU. When connecting to PC, program the firmware to ATMEGA 16U2-MU.
9	<b>RESET Button</b> You can reset your Arduino board, for example, start the program from the initial status. You can use the RESET button.
10	<b>D13 LED</b> 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.
11	<b>USB Connection</b> Arduino board can be powered via USB connector. All you needed to do is connecting the USB port to PC using a USB cable.
12	<b>ATMEGA 16U2-MU</b> USB to serial chip, can convert the USB signal into serial port signal.
13	<b>TX LED</b> Onboard you can find the label: TX (transmit) When Arduino board communicates via serial port, send the message, TX led flashes.
14	<b>RX LED</b> Onboard you can find the label: RX(receive ) When Arduino board communicates via serial port, receive the message, RX led flashes.
15	<b>Crystal Oscillator</b> How does Arduino calculate time? by using a crystal oscillator. The number printed on the top of the Arduino crystal is 16.000H9H. It tells us that the frequency is 16,000,000 Hertz or 16MHz.
16	<b>Voltage Regulator</b> Convert an external input DC7-12V voltage into DC 5V, then switch DC 5V to the processor and other components. Output DC 5V, the drive current is 2A.
17	<b>DC Power Jack</b> Arduino board can be supplied with an external power DC7-12V from the DC power jack.
18	<b>IOREF</b> Used to configure the operating voltage of microcontrollers. Use it less.
19	<b>RESET Header</b> Connect an external button to reset the board. The function is the same as reset button (labeled 9)
20	<b>Power Pin 3V3</b> A 3.3 volt supply generated by the on-board regulator. Maximum current draw is 50 mA.
21	<b>Power Pin 5V</b> Provides 5V output voltage
22	<b>Vin</b> You can supply an external power input DC7-12V through this pin to Arduino board.
23	<b>Analog Pins</b> Arduino UNO board has 6 analog inputs, labeled A0 through A5. These pins can read the signal from analog sensors (such as humidity sensor or temperature sensor), and convert it into the digital value that can read by microcontrollers.) Can also used as digital pins. A0=D14, A1=D15, A2=D16, A3=D17, A4=D18, A5=D19.
208	<b>Chapter 7: KS0341 Keystudio UNO Compatible advanced development board</b>
24	<b>Microcontroller</b> Each Arduino board has its own microcontroller. You can regard it as the brain of your board. The main IC (integrated circuit) on the Arduino is slightly different from the panel pair. Microcon-



Specialized Functions of Some Pins:

- **Serial communication:** Digital pins 0 (RX) and 1 (TX).
- **PWM Interfaces (Pulse-Width Modulation):** D3, D5, D6, D9, D10, D11
- **External Interrupts:** D2 (interrupt 0) and D3 (interrupt 1). These pins can be configured to trigger an interrupt on a low value, a rising or falling edge, or a change in value.
- **SPI communication:** D10 (SS), D11 (MOSI), D12 (MISO), D13 (SCK). These pins support SPI communication using the SPI library.
- **IIC communication:** A4 (SDA); A5(SCL)

Tips:

- Automatic (Software) Reset:

Rather than requiring a physical press of the reset button before an upload, the Arduino Uno board is designed in a way that allows it to be reset by software running on a connected computer.

- 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; see [this forum thread](#) for details.

Detailed Use with ARDUINO Software as follows:

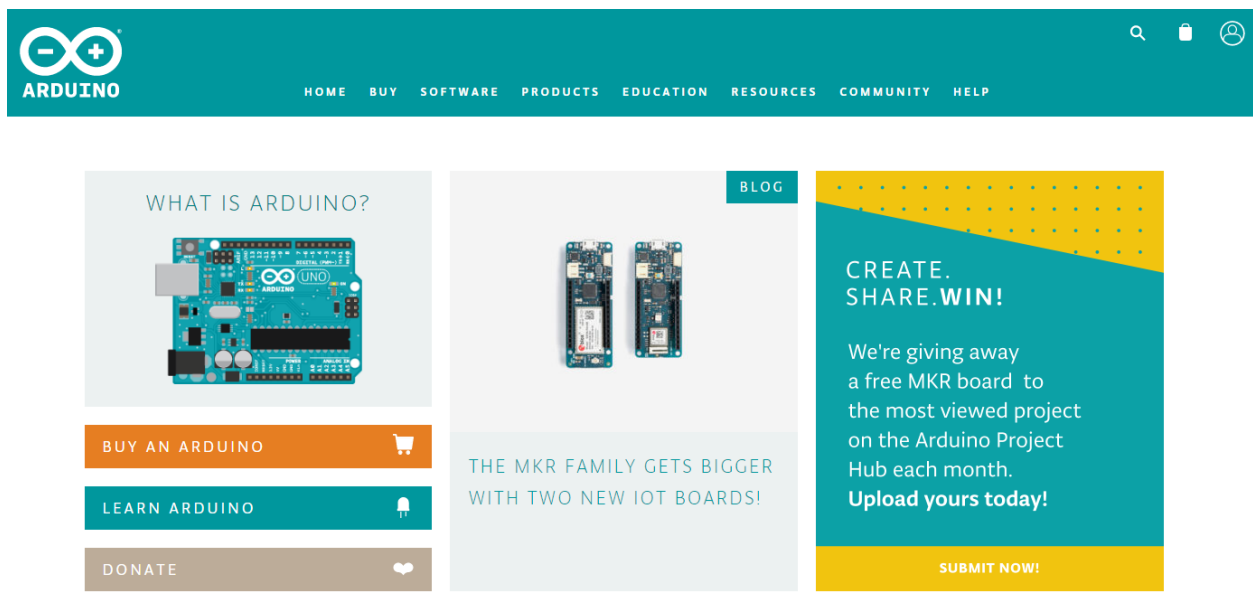
Step1| Download the Arduino environment (IDE)

When you get the UNO advanced board, first you should install the Arduino software and driver.

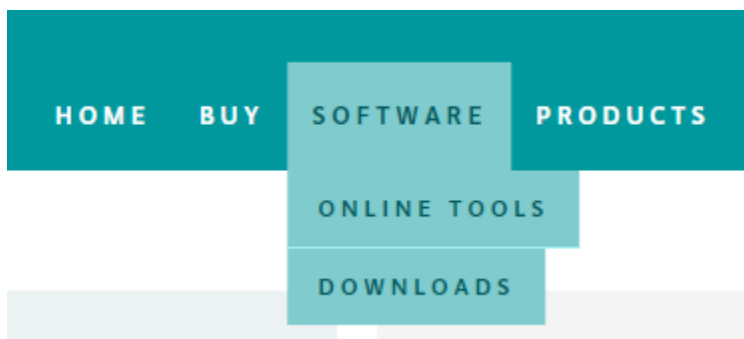
We usually use the Windows software Arduino 1.5.6 version. You can download it from the link below:

<https://www.arduino.cc/en/Main/OldSoftwareReleases\1.5.x>

Or you can browse the ARDUINO website to download the latest version from this link, <https://www.arduino.cc>, pop up the following interface.




Then click the **SOFTWARE** on the browse bar, you will have two options **ONLINE TOOLS** and **DOWNLOADS**.



Click **DOWNLOADS**, it will appear the latest software version of ARDUINO 1.8.5 shown as below.

## Download the Arduino IDE




**ARDUINO 1.8.5**

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** 10.7 Lion or newer

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

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

In this software page, on the right side you can see the version of development software for different operating systems. ARDUINO has a powerful compatibility. You should download the software that is compatible with the operating system of your computer.

We will take **WINDOWS system** as an example here. There are also two options under Windows system, one is installed version, the other is non-installed version.

For simple installed version, first click **Windows Installer**, you will get the following page.

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

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

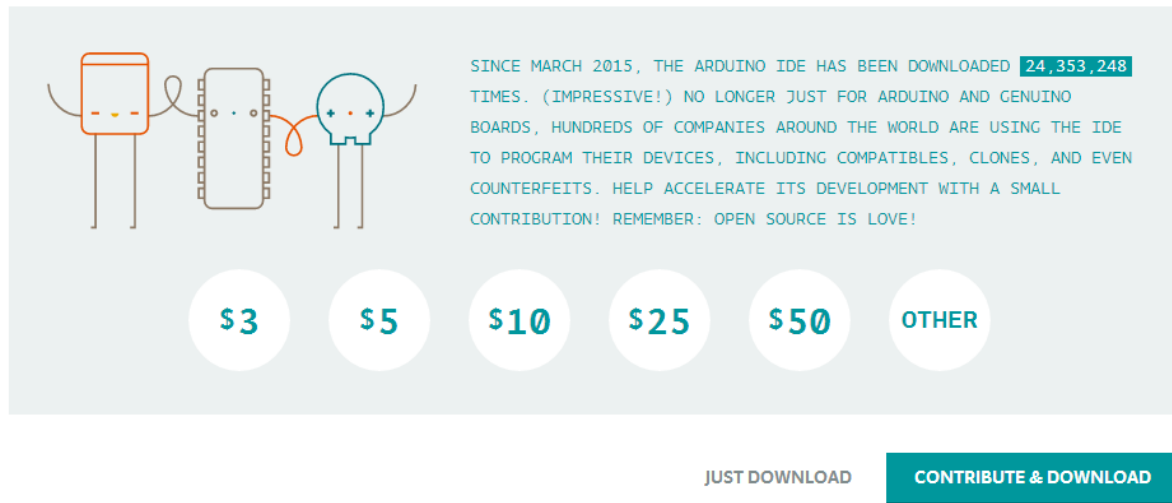
**Mac OS X** 10.7 Lion or newer

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

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

## Contribute to the Arduino Software

Consider supporting the Arduino Software by contributing to its development. (US tax payers, please note this contribution is not tax deductible). [Learn more on how your contribution will be used.](#)



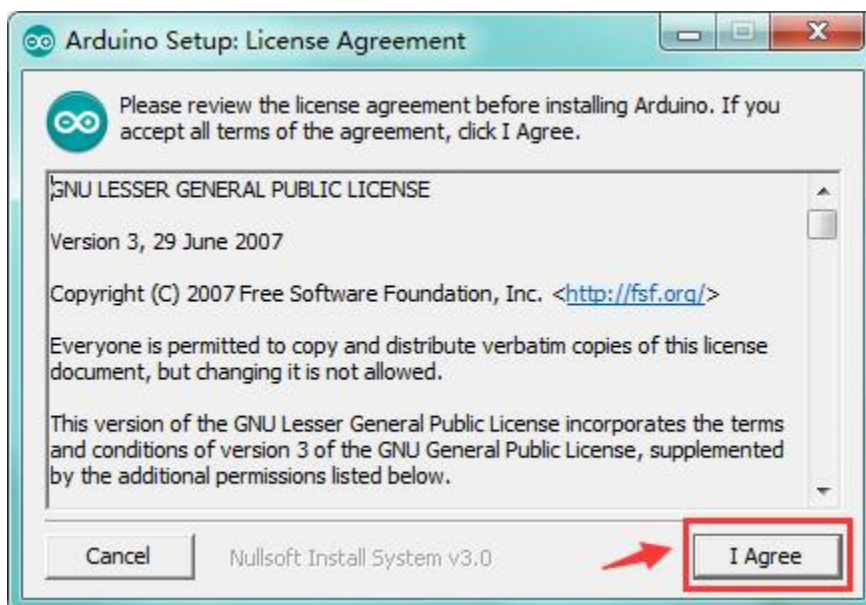
This way you just need to click JUST DOWNLOAD, then click the downloaded file to install it.

For non-installed version, first click Windows ZIP file, you will also get the pop-up interface as the above figure.

Click JUST DOWNLOAD, and when the ZIP file is downloaded well to your computer, you can directly unzip the file and click the icon of ARDUINO software to start it.

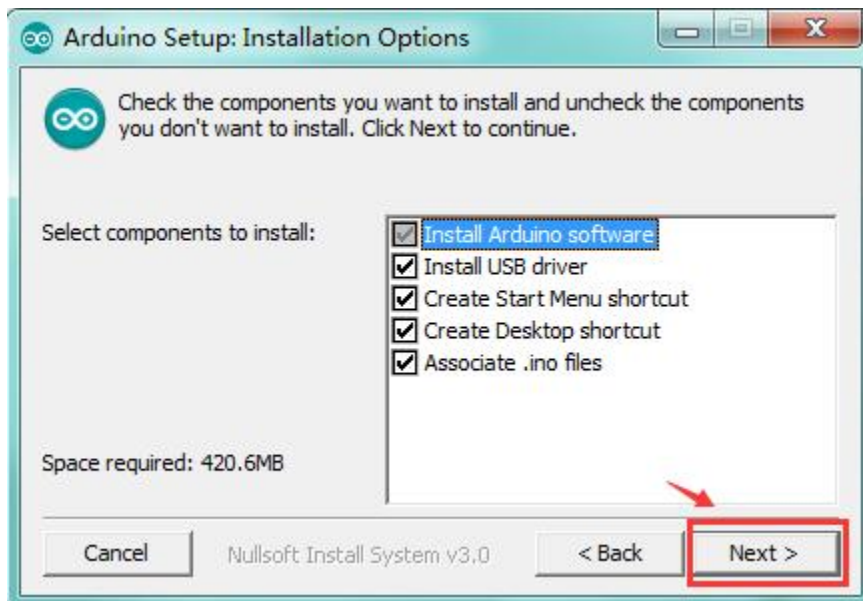
### Installing Arduino (Windows):

Install Arduino with the exe. Installation package downloaded well.

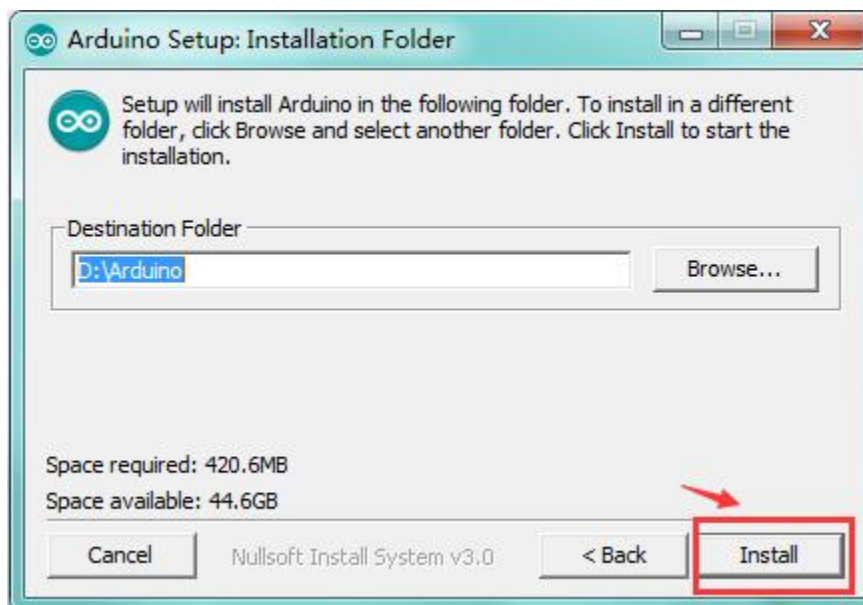


Click "I Agree" to see the following interface.

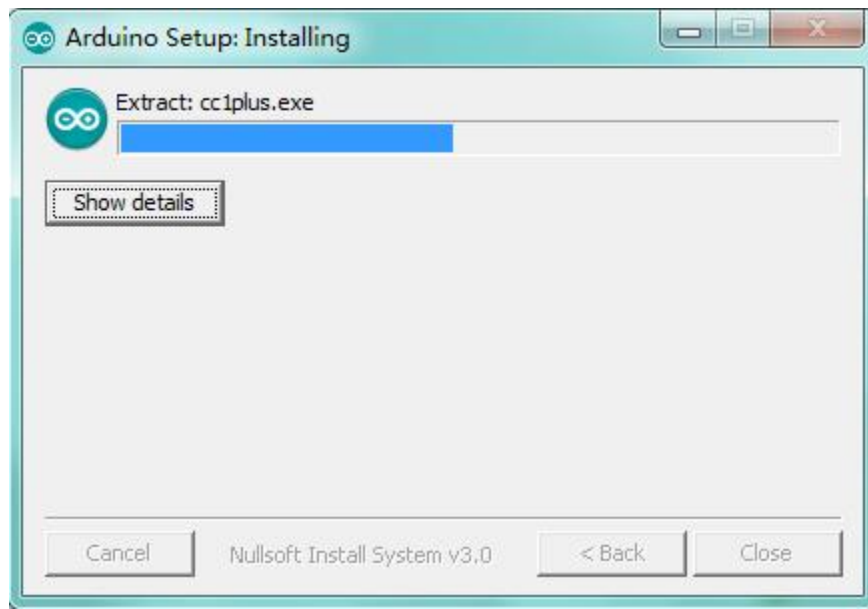




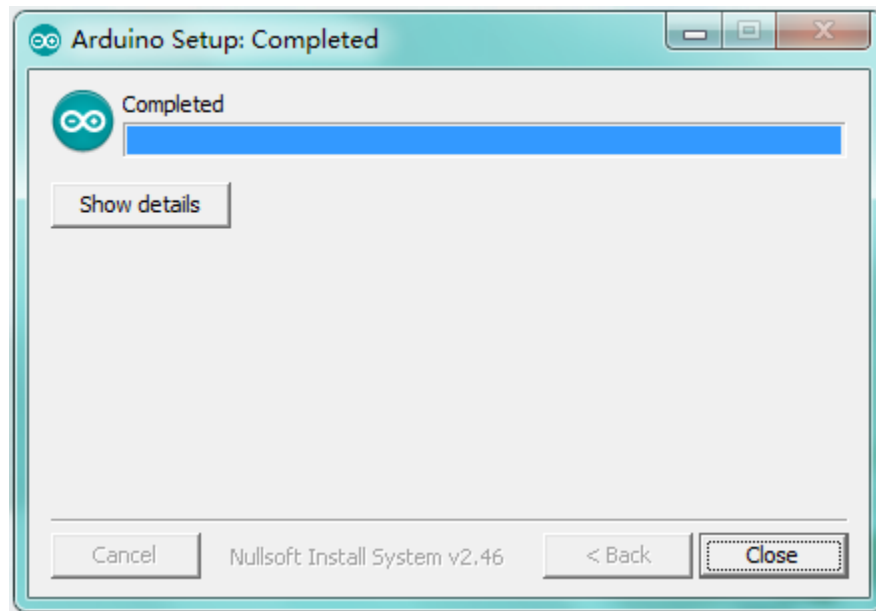
Click “Next”. Pop up the interface below.



You can press Browse... to choose an installation path or directly type in the directory you want. Then click “Install” to initiate installation.



Wait for the installing process, if appear the interface of Window Security, just continue to click Install to finish the installation.

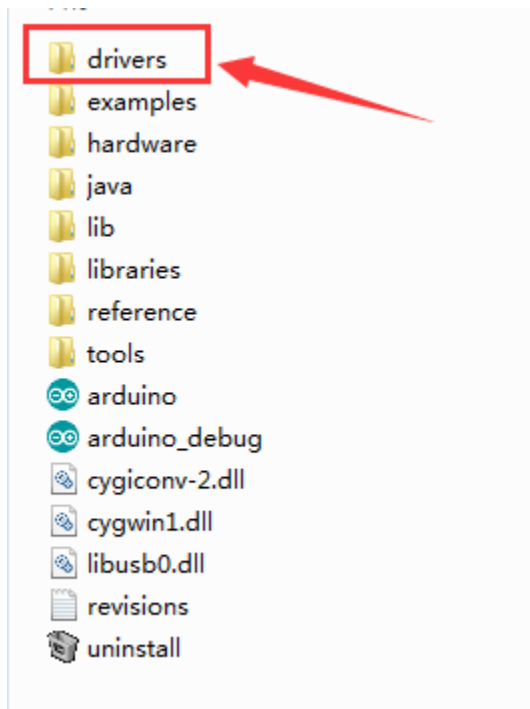


### Installing Driver:

Next, we will introduce the driver installation of UNO R3 development board. The driver installation may have slight differences in different computer systems. So in the following let's move on to the driver installation in the WIN 7 system.

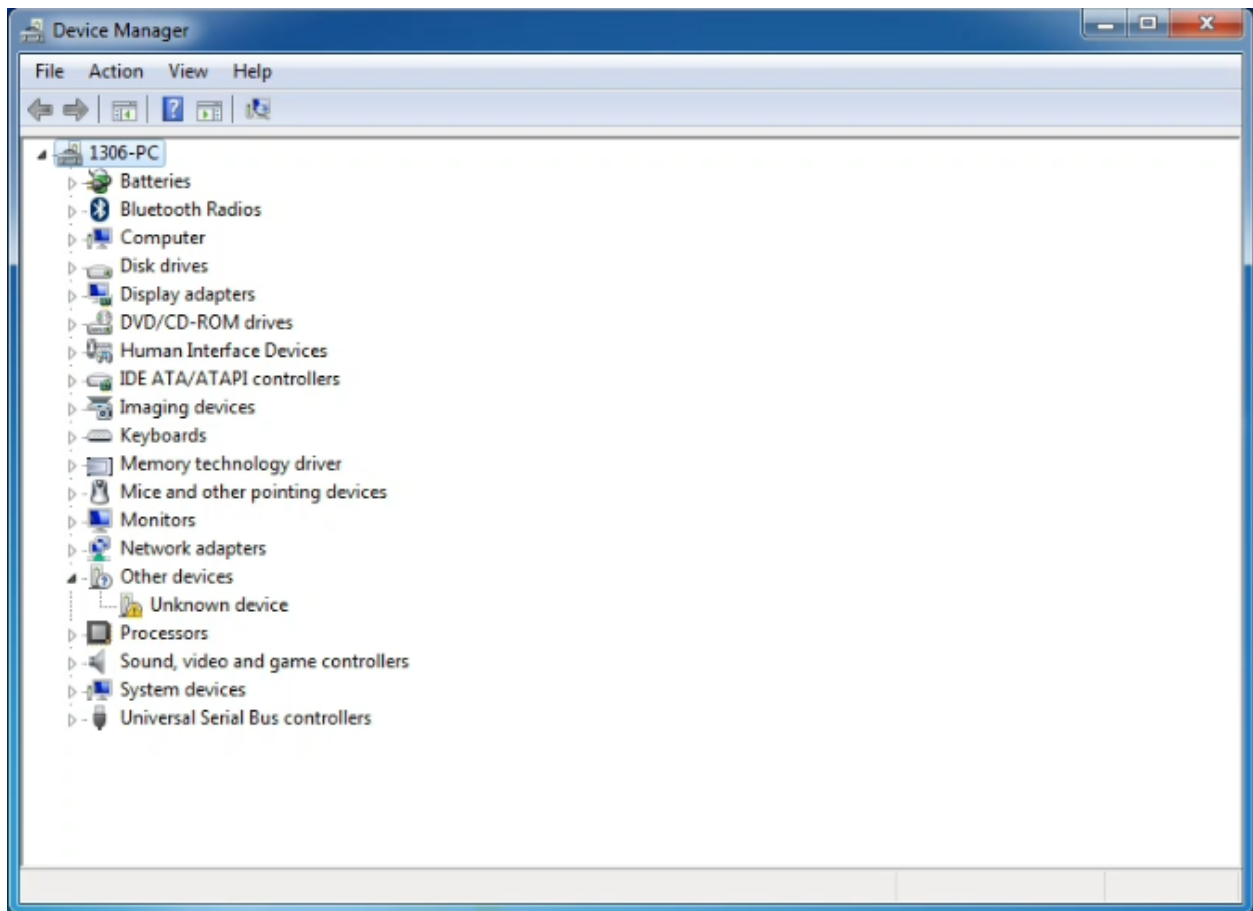
The Arduino folder contains both the Arduino program itself and the drivers that allow the Arduino to be connected to your computer by a USB cable. Before we launch the Arduino software, you are going to install the USB drivers.



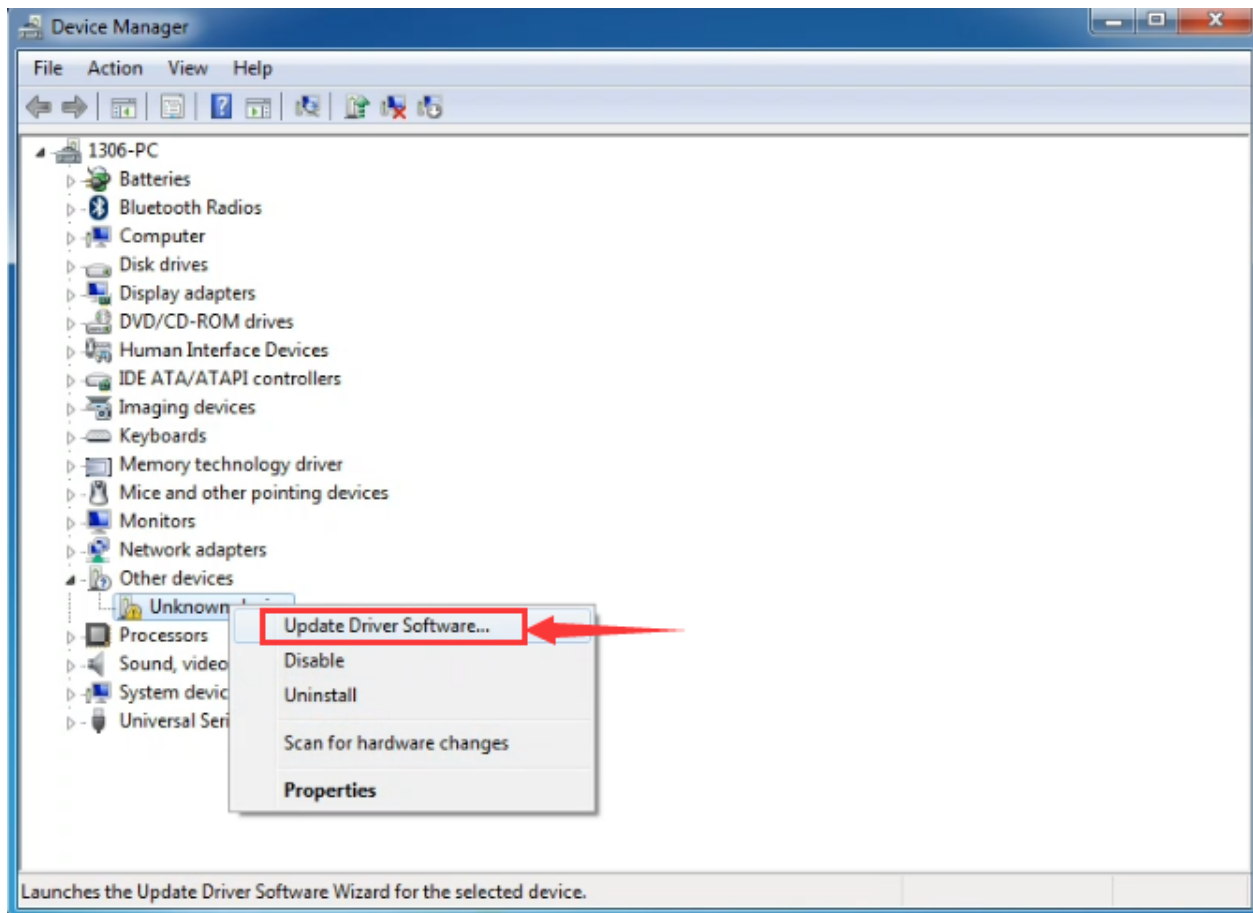


Plug one end of your USB cable into the Arduino and the other into a USB socket on your computer.

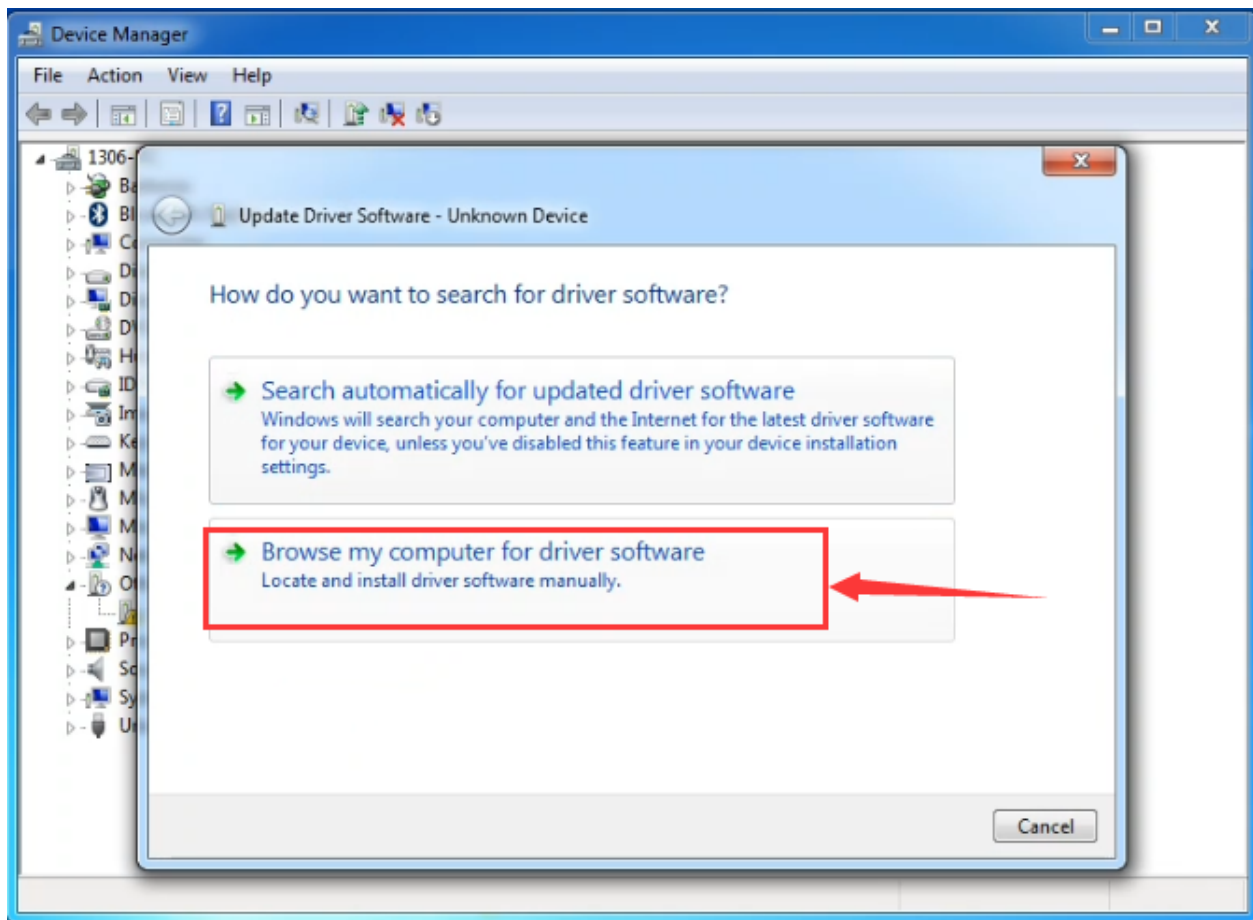
When you connect UNO board to your computer at the first time, right click the icon of your “Computer” —>for “Properties”—> click the “Device manager”, under “Other Devices”, you should see an icon for “Unknown device” with a little yellow warning triangle next to it. This is your Arduino.



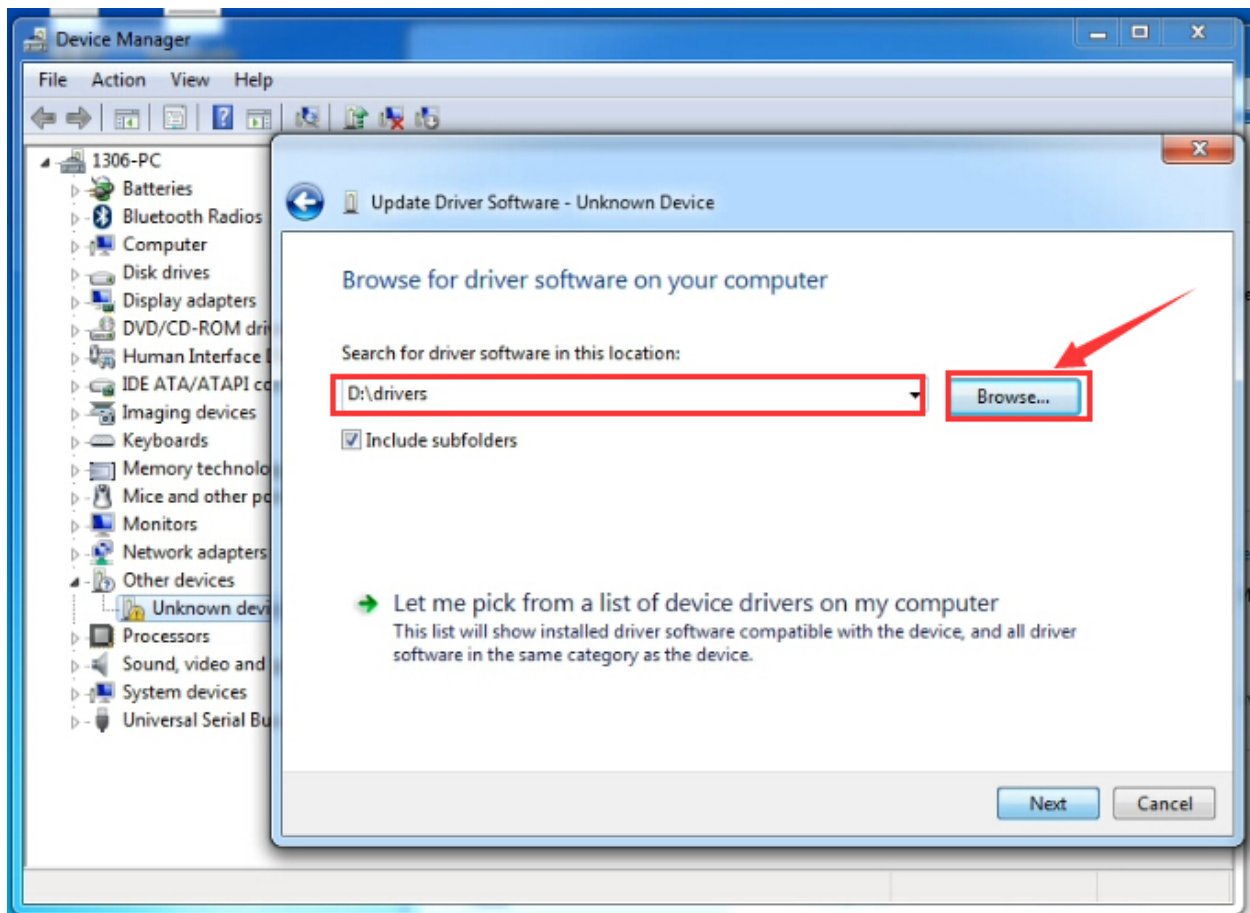
Then right-click on the device and select the top menu option (Update Driver Software...) shown as the figure below.



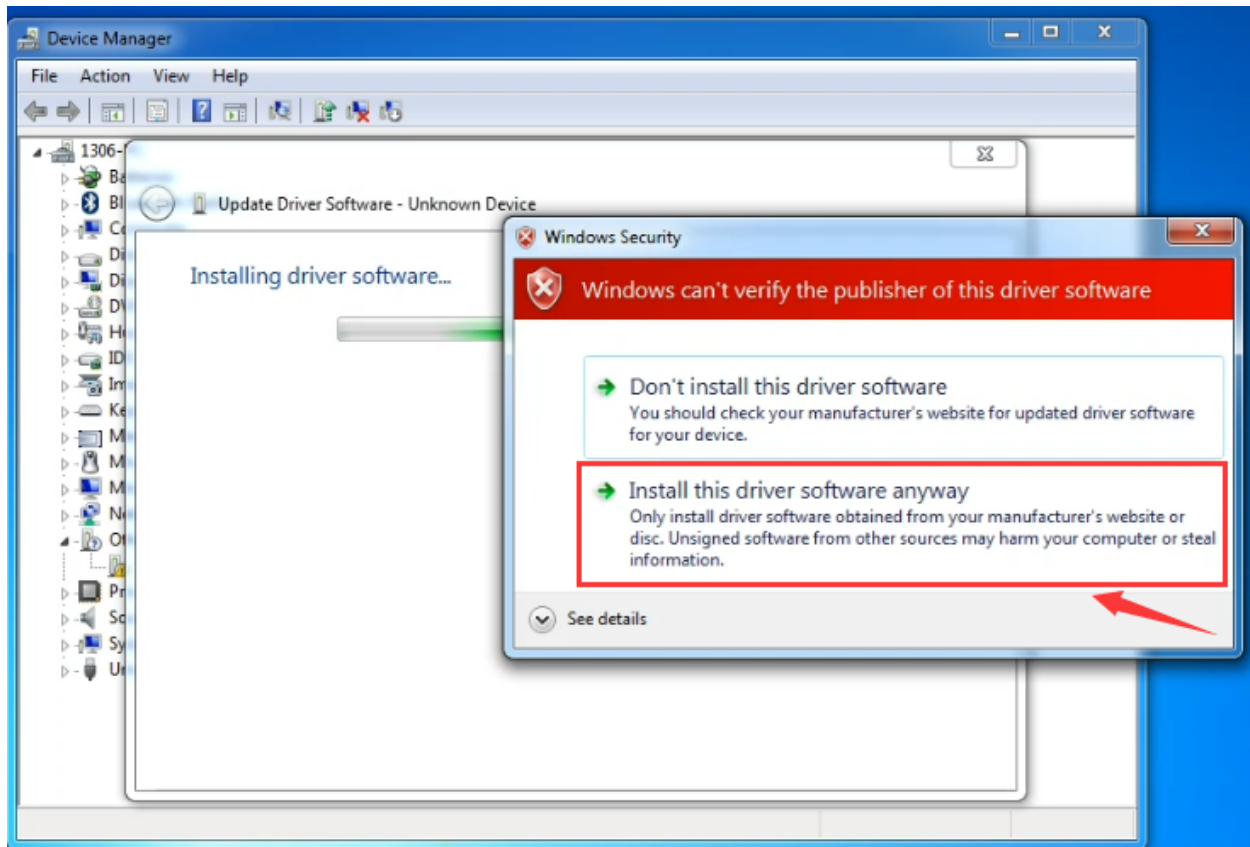
It will then be prompted to either “Search Automatically for updated driver software” or “Browse my computer for driver software”. Shown as below. In this page, select “Browse my computer for driver software”.



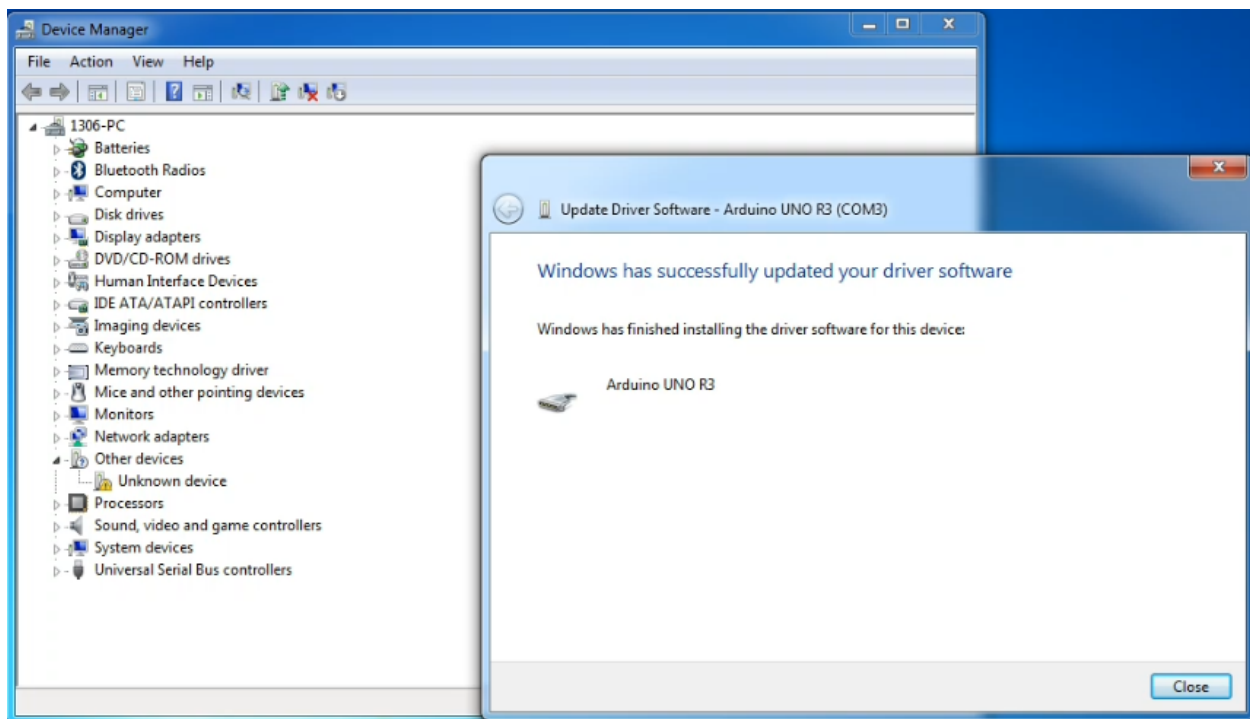
After that, select the option to browse and navigate to the “drivers” folder of Arduino installation.



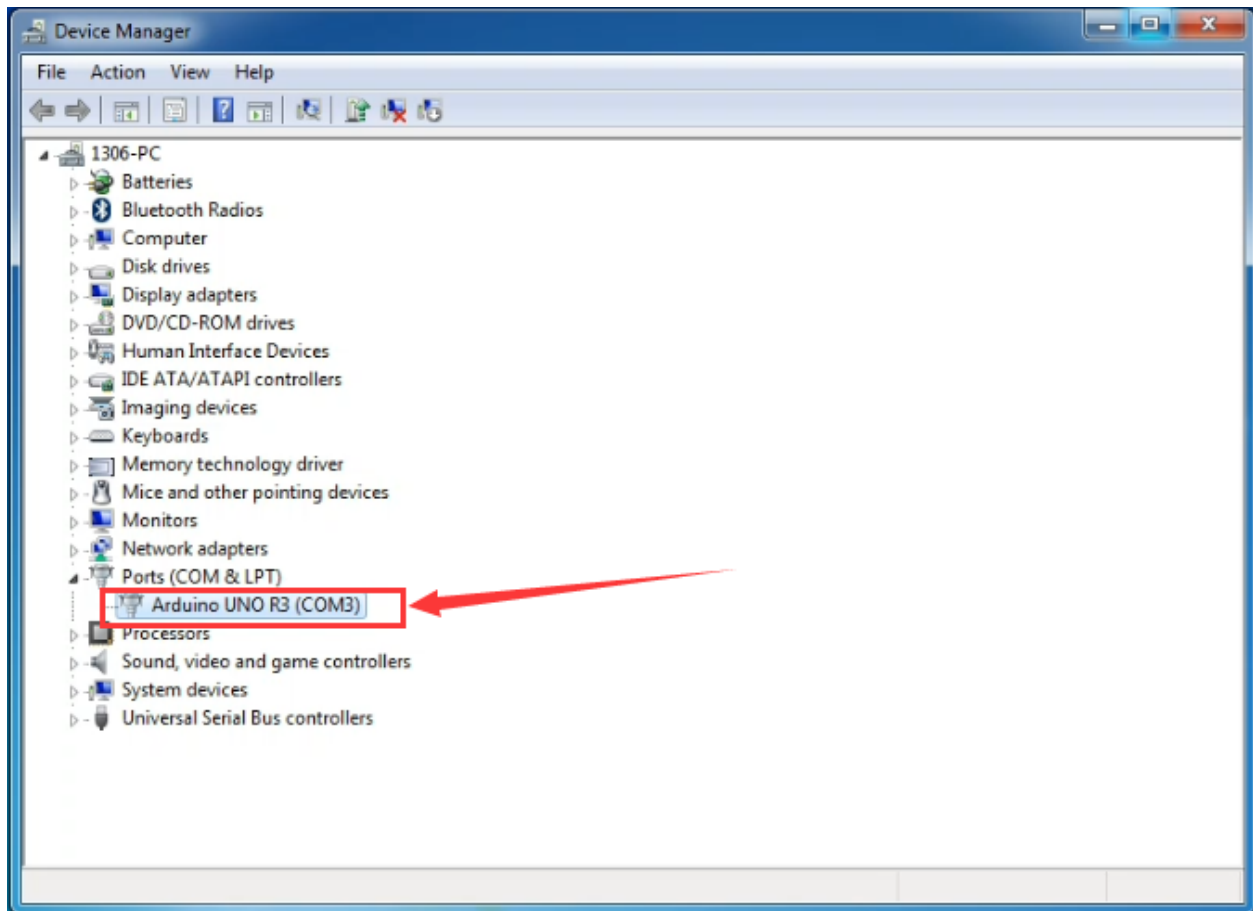
Click “Next” and you may get a security warning, if so, allow the software to be installed. Shown as below.



Installation completed, click “Close”.



Up to now, the driver is installed well. Then you can right click “Computer” —> “Properties”—> “Device manager”, you should see the device shown below.

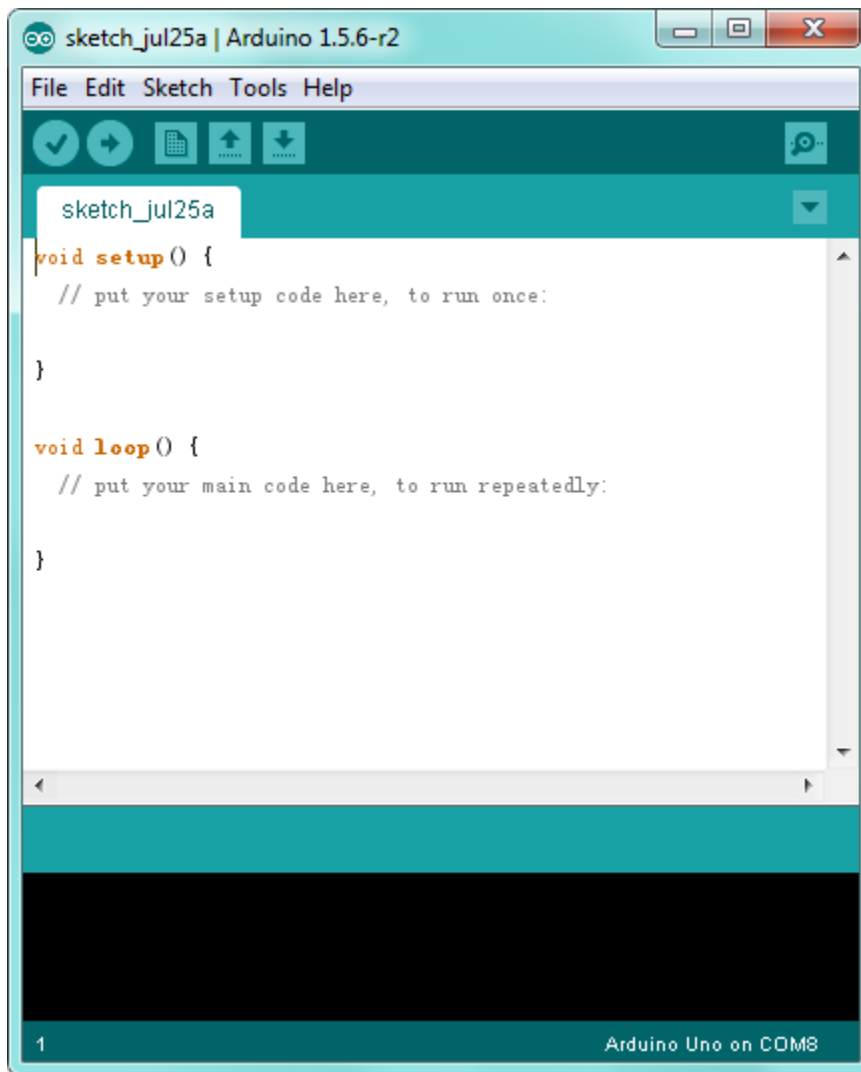


### Introduction for Arduino IDE Toolbar:

Double-click the icon of Arduino software downloaded well, you will get the interface shown below.









(Note: if the Arduino software loads in the wrong language, you can change it in the preferences dialog. See [the environment page](#) for details.)



The functions of each button on the Toolbar are listed below: <http://wiki.keyestudio.com/index.php/File:IDE.png>





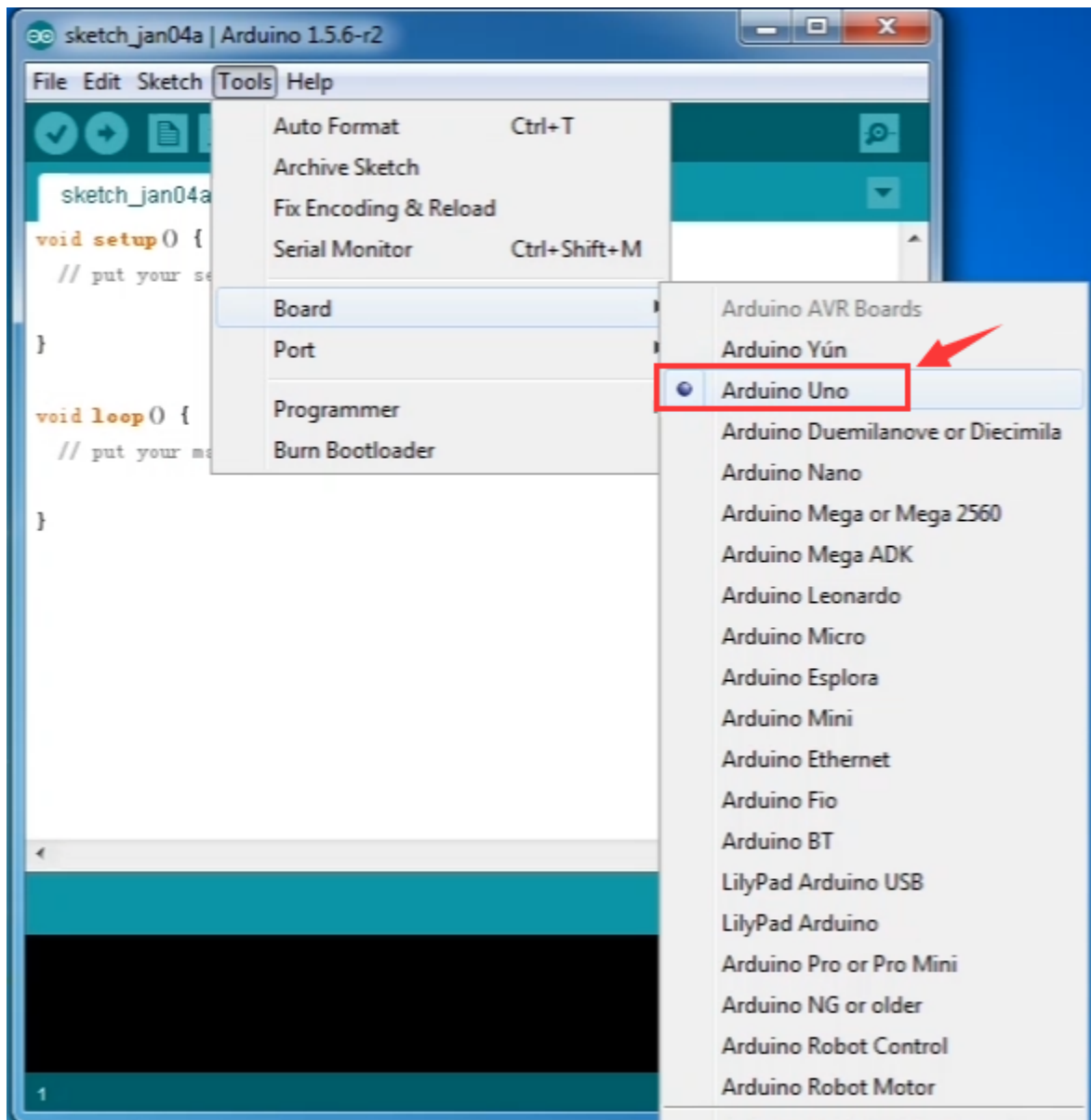
	<b>Verify/Compile</b>	Check the code for errors
	<b>Upload</b>	Upload the current Sketch to the Arduino
	<b>New</b>	Create a new blank Sketch
	<b>Open</b>	Show a list of Sketches
	<b>Save</b>	Save the current Sketch
	<b>Serial Monitor</b>	Display the serial data being sent from the Arduino

Step2| Connect the board

Connect the UNO board to your computer using the USB cable. The green power LED should go on.

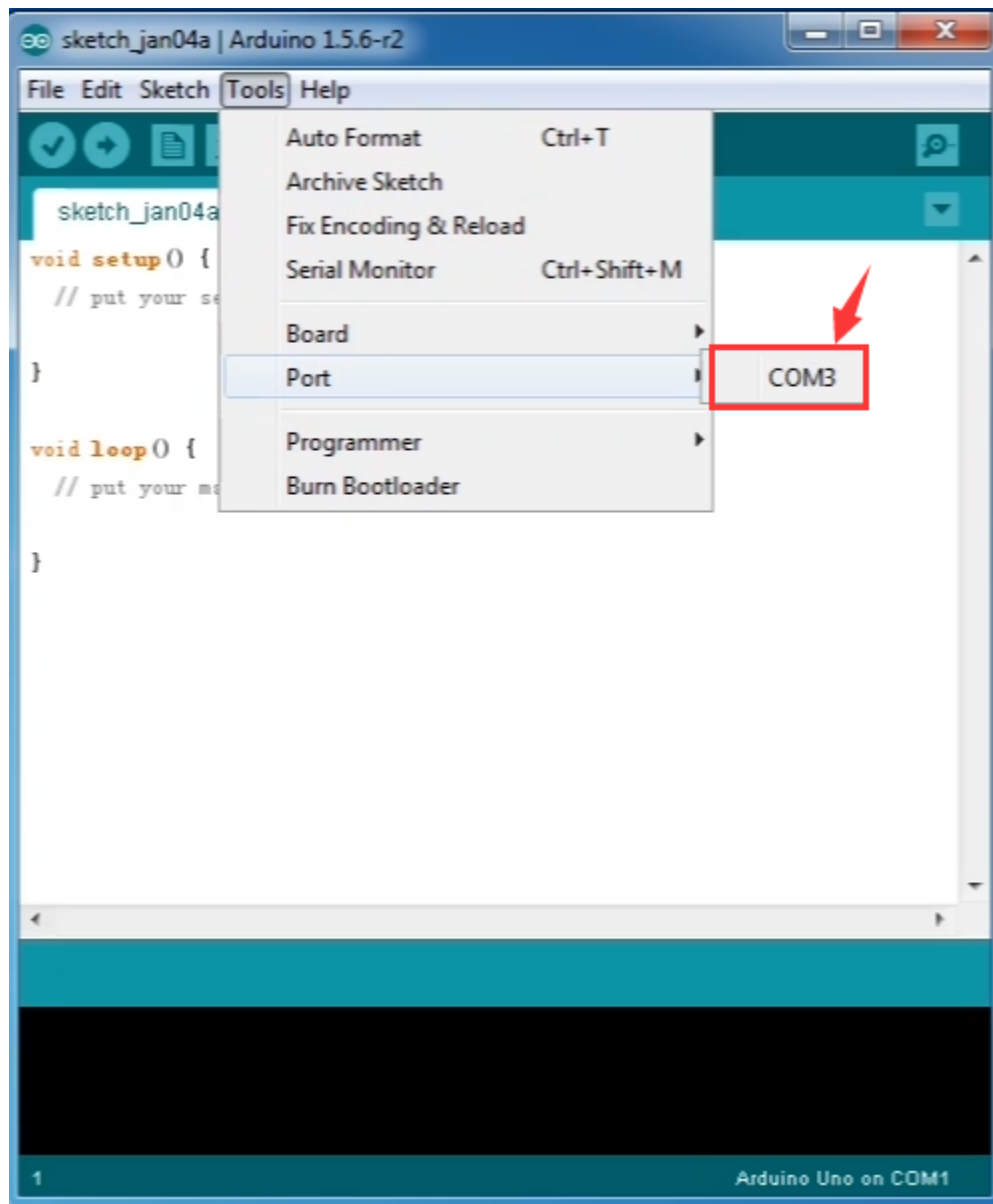
Step3| Select the Arduino Board

Open the Arduino IDE, you'll need to click the "Tools", then select the Board that corresponds to your Arduino.

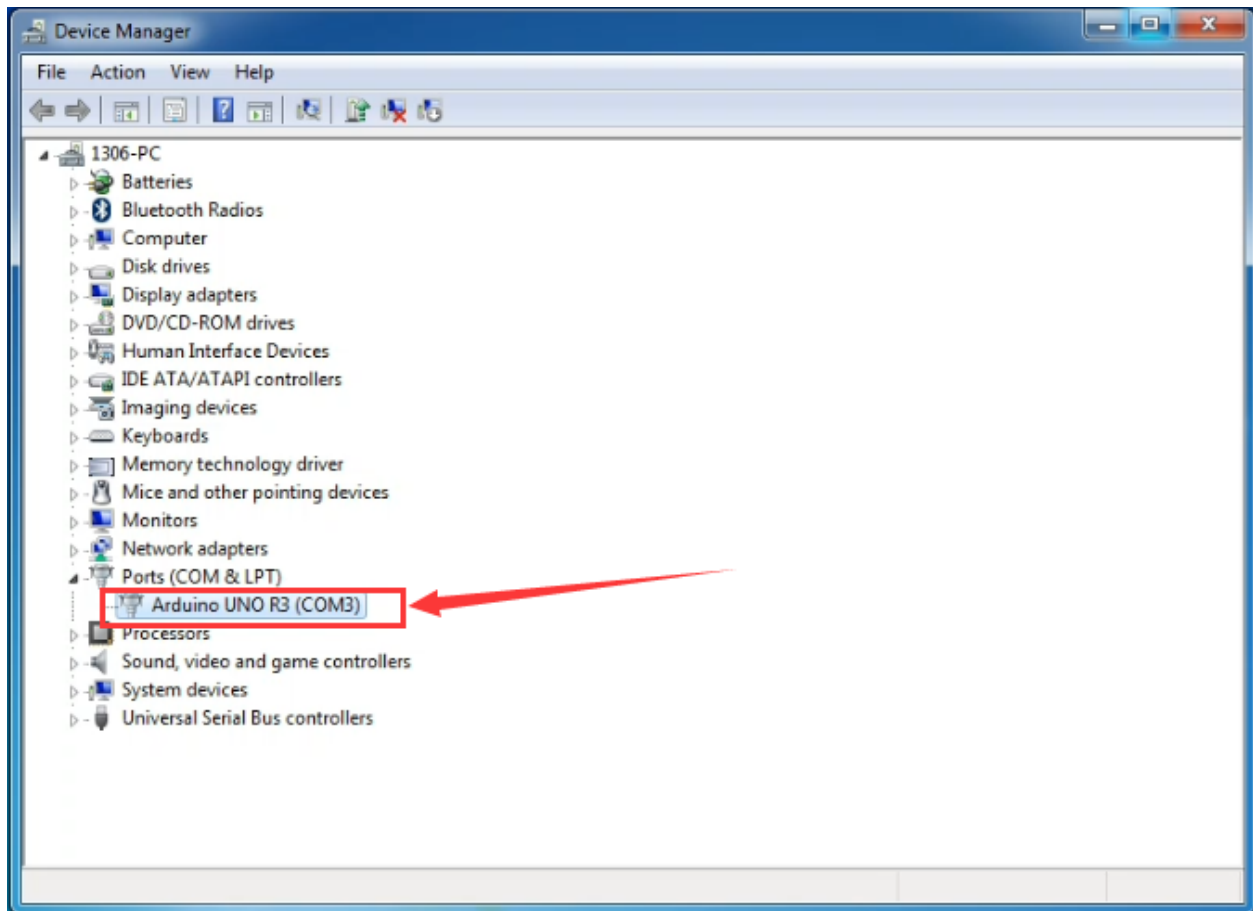


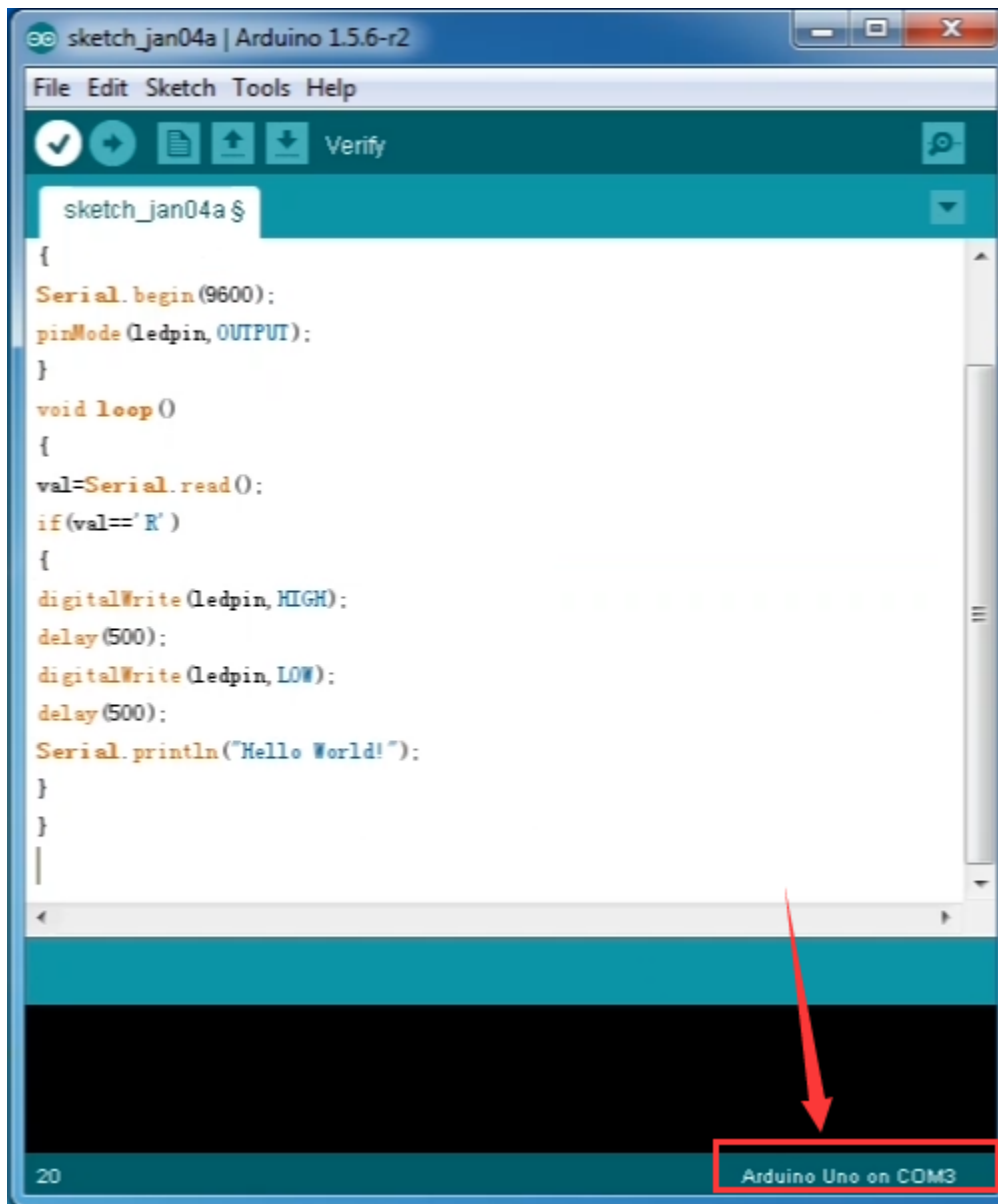
Step4| Select your serial port

Select the serial device of the Arduino board from the **Tools | Serial Port menu**. This is likely to be COM3 or higher (COM1 and COM2 are usually reserved for hardware serial ports). To find out, you can disconnect your Arduino board and re-open the menu; the entry that disappears should be the Arduino board. Reconnect the board and select that serial port.



**Note:** to avoid errors, the COM Port should keep the same as the Ports shown on Device Manager.





Step5| Upload the Program

Below is an example program for displaying the Hello World!

Copy and paste the code to the Arduino environment IDE.

////////////////////////////////////

```
int val;
```

```
int ledpin=13;
```

```
void setup()
```

```
{
```

```
  Serial.begin(9600);
```

```
  pinMode(ledpin,OUTPUT);
```

```
}  
void loop()  
{  
  val=Serial.read();  
  if(val=='R')  
  {  
    digitalWrite(ledpin,HIGH);  
    delay(500);  
    digitalWrite(ledpin,LOW);  
    delay(500);  
    Serial.println("Hello World!");  
  }  
}  
////////////////////////////////////
```

Then click verify button to check the errors. If compiling successfully, the message “Done compiling.” will appear in the status bar.



After that, click the “Upload” button to upload the code. Wait a few seconds - you should see the RX and TX leds on the board flashing. If the upload is successful, the message “Done uploading.” will appear in the status bar.

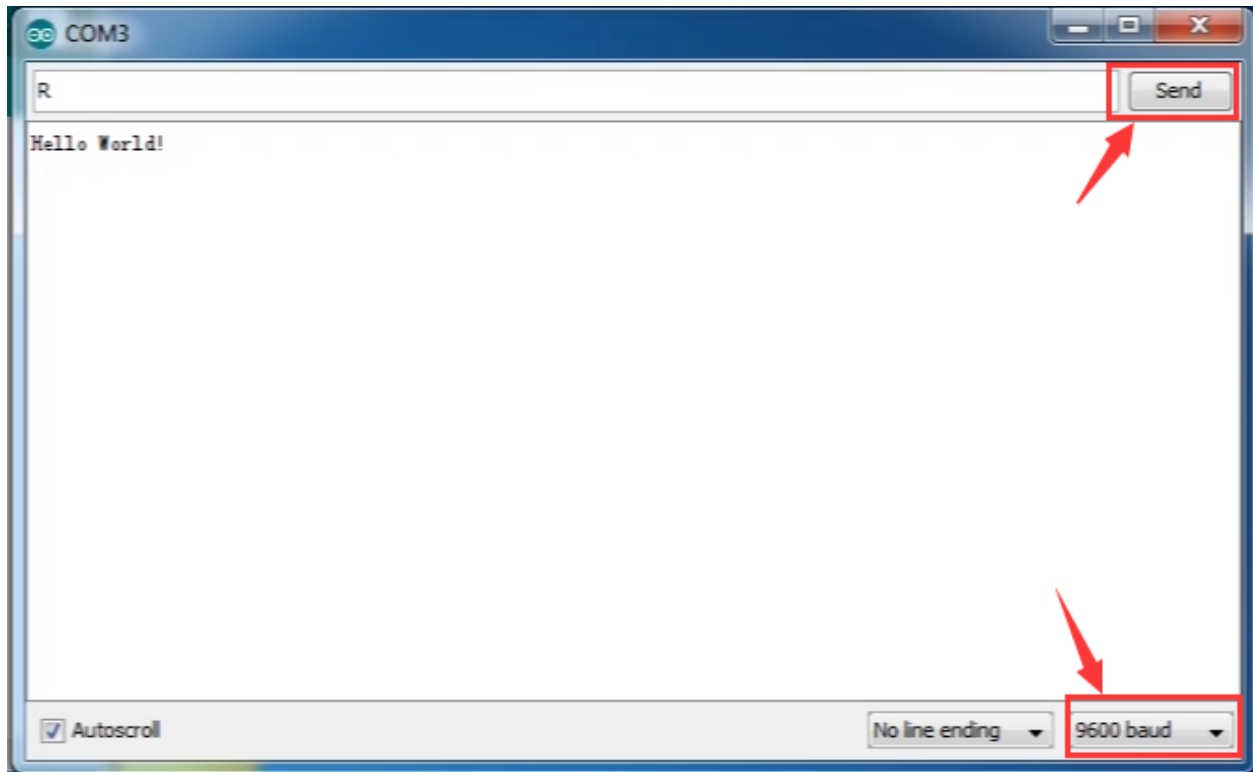


Step6| Open the Serial Monitor

After that, click the serial monitor button to open the serial monitor.



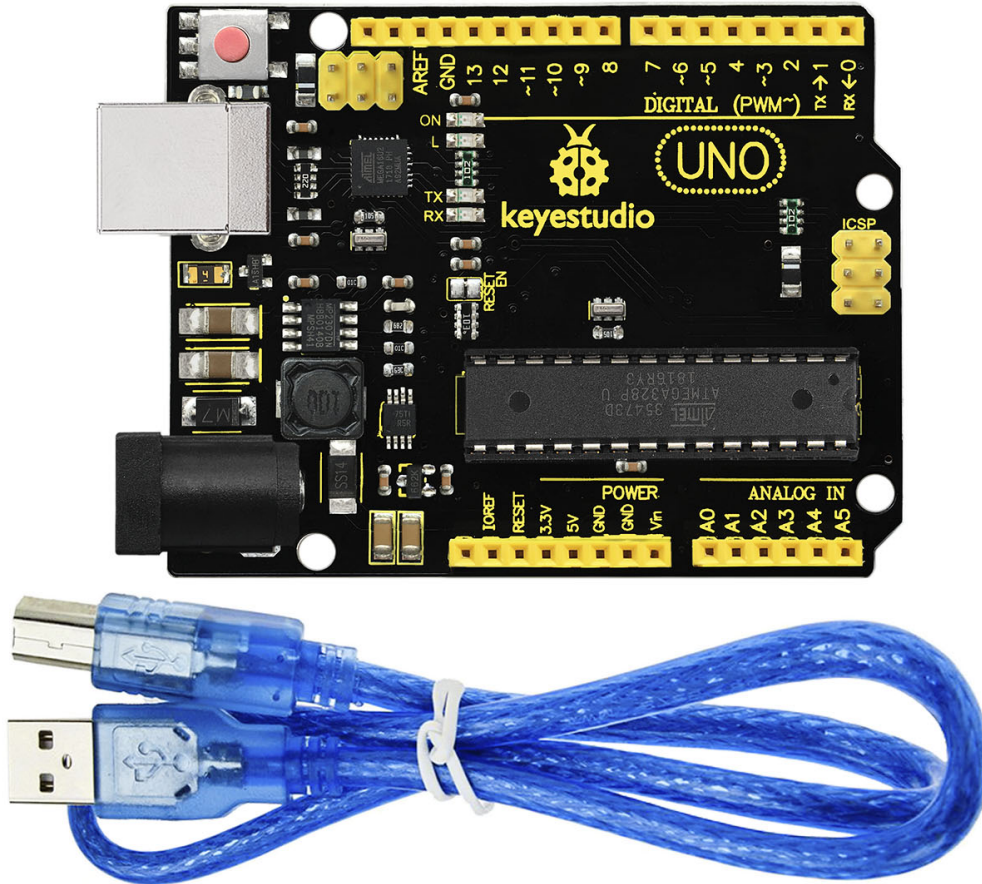
Then set the baud rate as 9600, enter an “R” and click Send, you should see the RX led on the board blink once, and then D13 led blink once, finally “Hello World!” is showed on the monitor, the TX led blink once.



Package Included:

- Keystudio UNO Advanced board \* 1pcs
- USB cable \* 1pcs





Other Links:

You might also want to look at:

[the examples](#) for using various sensors and actuators;

[the reference](#) for the Arduino language;

You can download the datasheet from the link:

<https://drive.google.com/open?id=1t6mqdd4EUA4kvwIJrRG-VHU2H9iZEJHF>

Software Download:

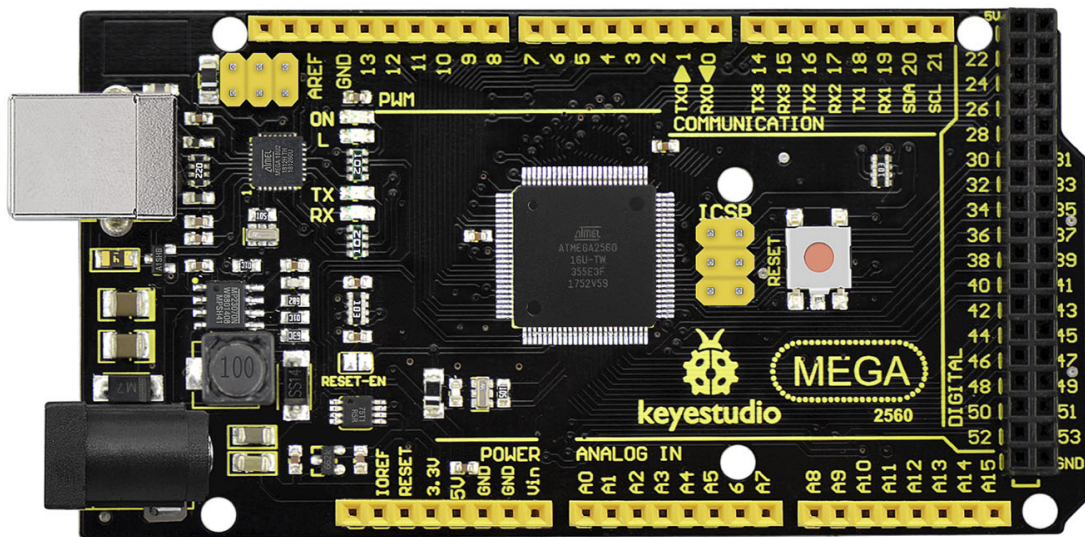
<https://drive.google.com/open?id=12D-JkXdNm03Qt4dIPQr3RP6OmgXqpVHc>

Troubleshooting:

If you have problems, please see the [troubleshooting](#) suggestions.



## KS0342 KEYESTUDIO MEGA COMPATIBLE ADVANCED DEVELOPMENT BOARD



Introduction:

Keystudio MEGA compatible board Advanced is a microcontroller board based on the [ATMEGA2560-16AU](#) , fully compatible with [Keystudio MEGA R3 board](#) and ARDUINO MEGA REV3.

It has 54 digital input/output pins (of which 15 can be used as PWM outputs), 16 analog inputs, 4 UARTs (hardware serial ports), a 16 MHz crystal oscillator, a USB connection, a power jack, 2 ICSP headers, and a reset button.

It contains everything needed to support the microcontroller. With its bootloader, program can be downloaded directly with USB and you don't need to use other external programmer.

Just simply connect it to a computer with a USB cable or power it via an external DC power jack (DC 7-12V) or via female headers Vin /GND (DC 7-12V) to get started.

**Note:** the most important difference between keystudio MEGA compatible board Advanced and keystudio MEGA R3 board is the voltage regulator chip used in the board.

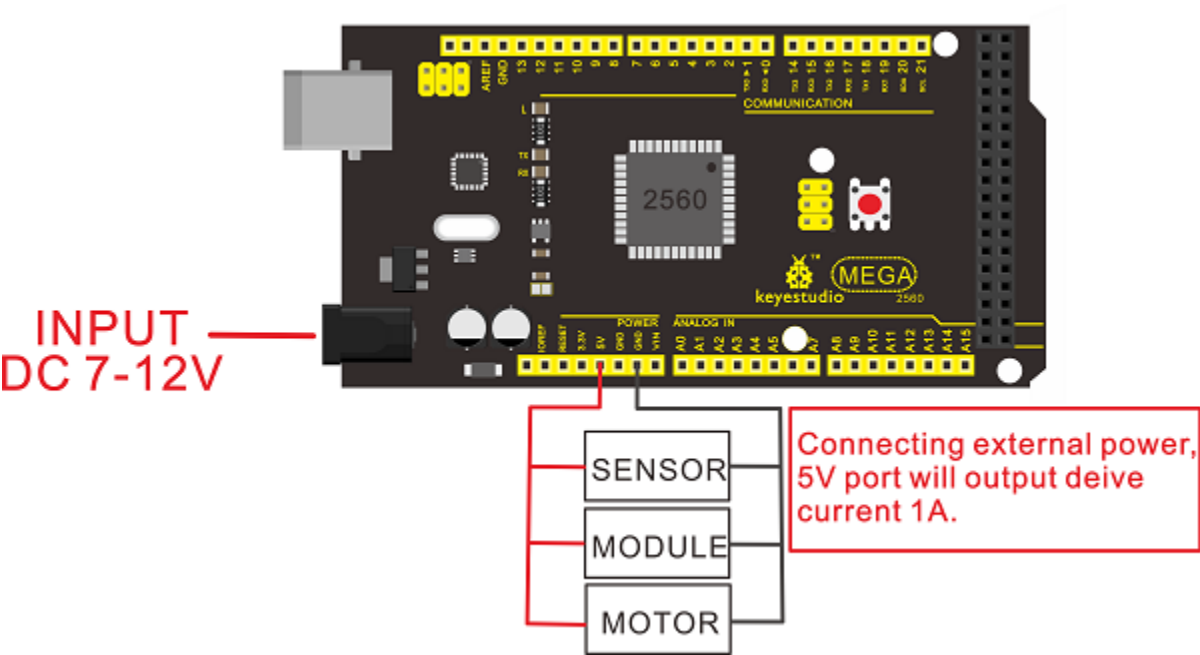
The voltage regulator chip used in keystudio MEGA R3 board is NSP1117. When connect the external power, output 5V, the drive current is 1A.

However, keystudio MEGA compatible board Advanced features the voltage-regulator chip MP2307DN. When connect the external power, output 5V, the drive current can be 2A.

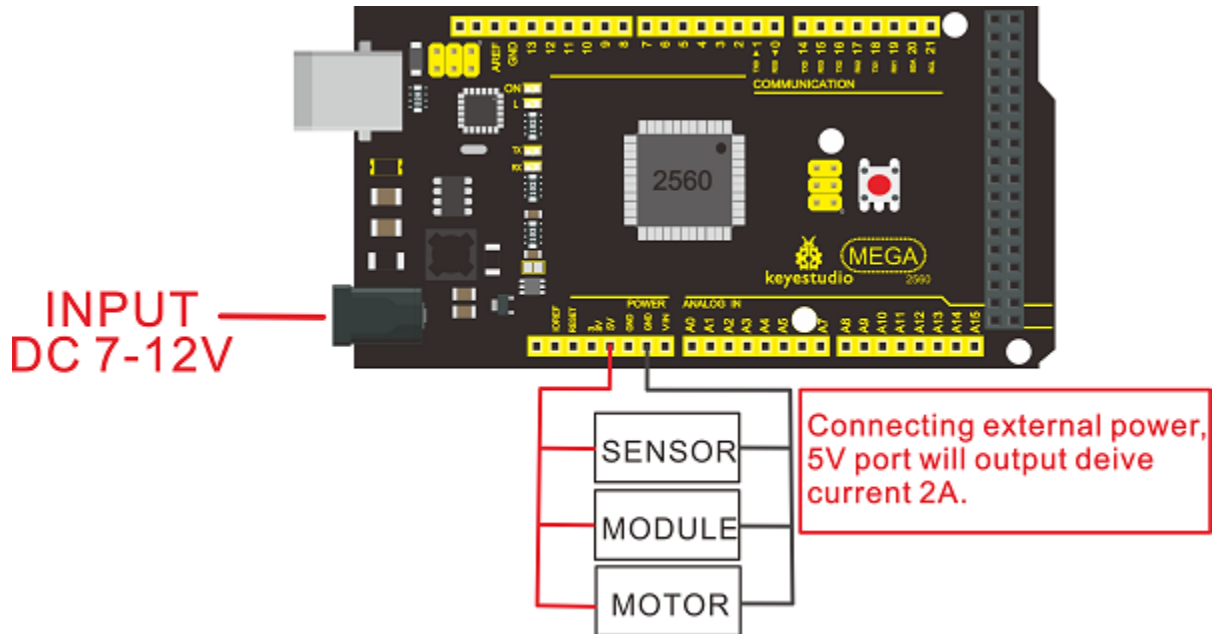
The MEGA is designed for more complex projects. With 54 digital I/O pins, 16 analog inputs and a larger space for your sketch, it is the recommended board for 3D printers and robotics projects. This gives your projects plenty of room and opportunities.

	keystudio MEGA compatible board Ad- vanced	keystudio MEGA R3 board
<b>voltage-regulator chip</b>	MP2307DN	NSP1117
<b>Power Output</b>	5V	5V
<b>Drive Current (external power)</b>	2A	1A
<b>Drive Current (USB power)</b>	0.5A	0.5A

keystudio MEGA R3 board:



keyestudio MEGA compatible board Advanced:

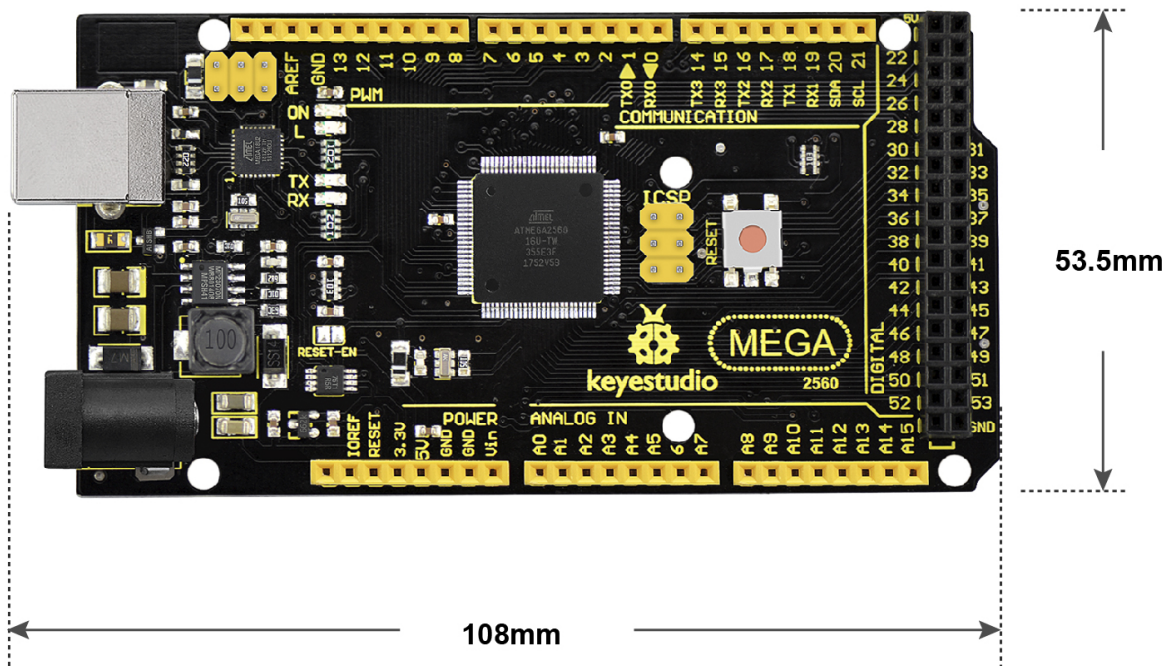


TECH SPECS:

<b>Microcontroller</b>	ATMEGA2560-16AU
<b>Operating Voltage</b>	5V
<b>Input Voltage (recommended)</b>	DC 7-12V
<b>Digital I/O Pins</b>	54 (D0-D53)
<b>PWM Digital I/O Pins</b>	15 (D2-D13; D44-D46)
<b>Analog Input Pins</b>	16 (A0-A15)
<b>DC Current per I/O Pin</b>	20 mA
<b>DC Current for 3.3V Pin</b>	50 mA
<b>Flash Memory</b>	256 KB of which 8 KB used by bootloader
<b>SRAM</b>	8 KB
<b>EEPROM</b>	4 KB
<b>Clock Speed</b>	16 MHz
<b>LED_BUILTIN</b>	D13

Details:

- Dimensions: 108mm x 53.5mm x 15mm
- Weight: 33.6g

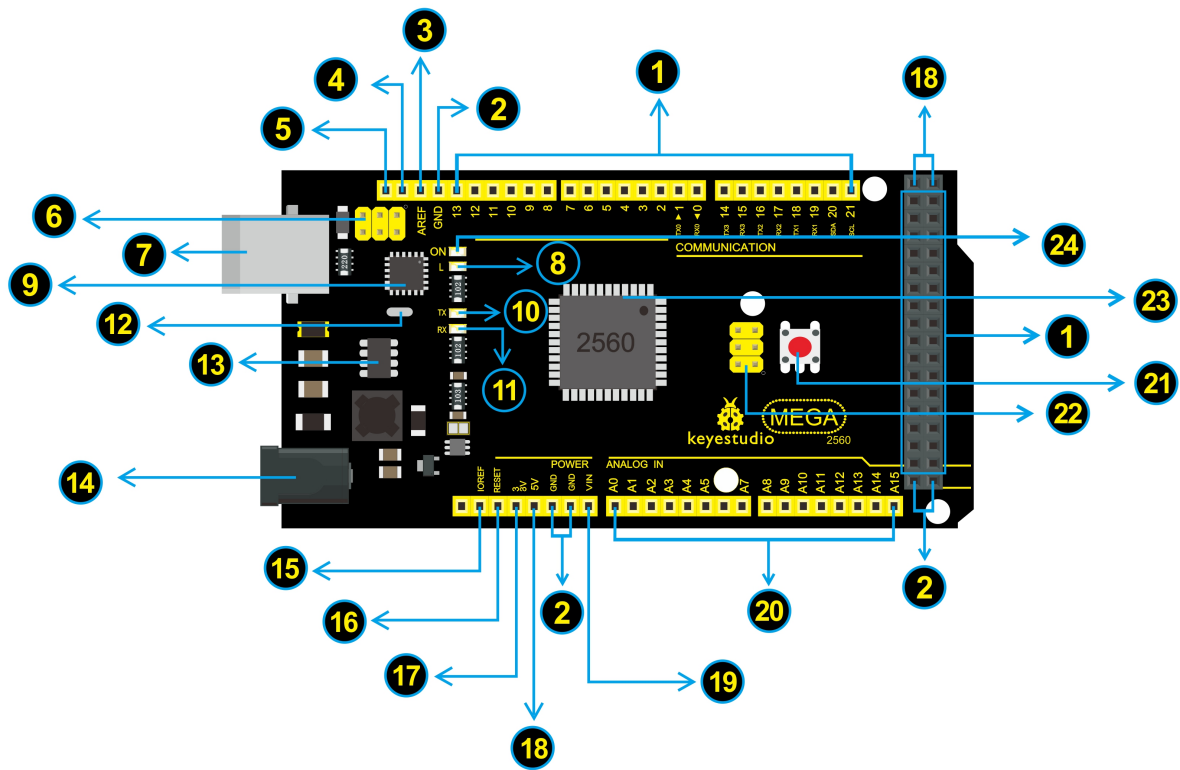


H:15mm  
only board weight :33.6g

#### Element and Pin Interfaces:

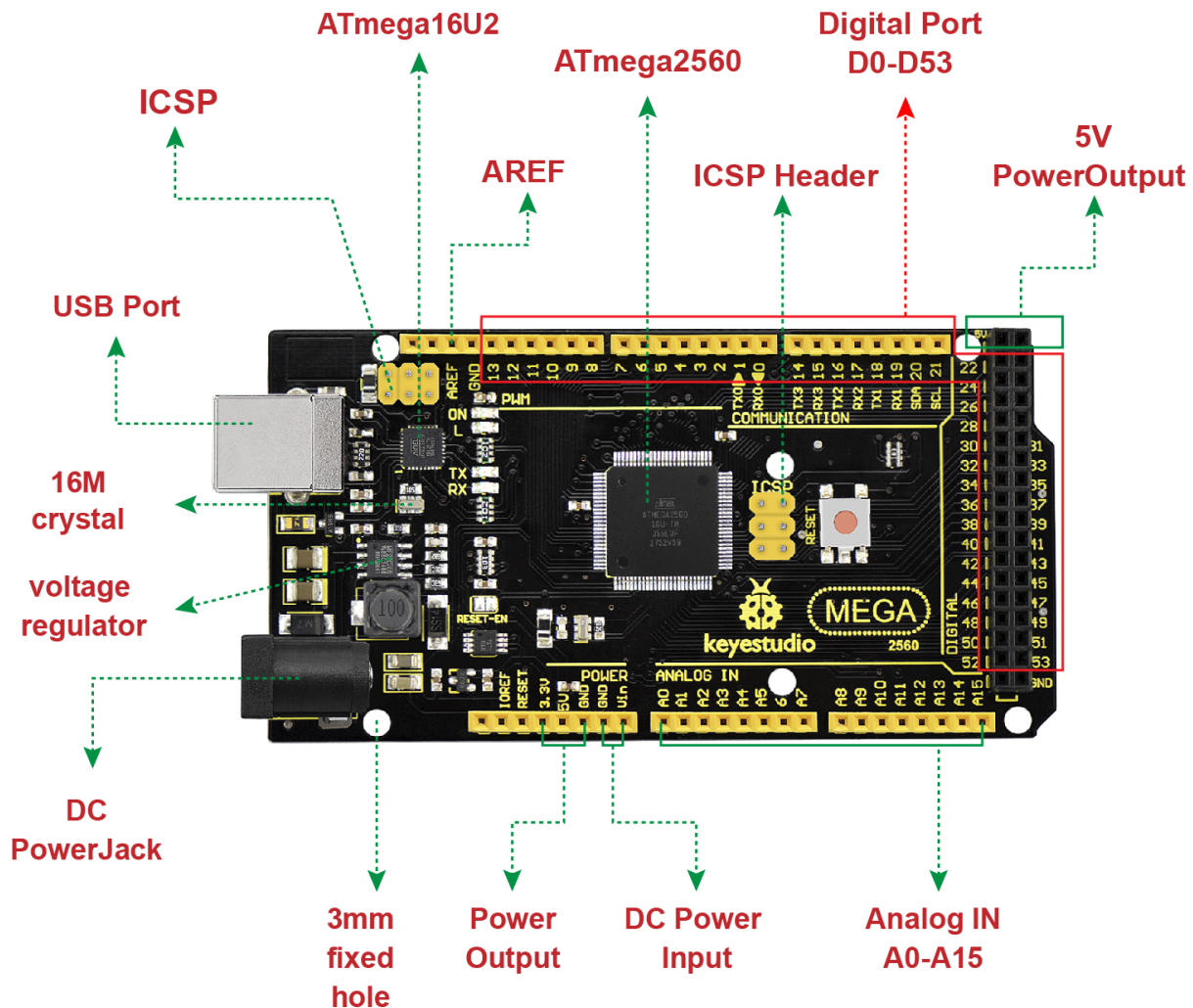
Here is an explanation of what every element and interface of the board does:





1	<b>Digital I/O</b> Arduino MEGA has 54 digital input/output pins (of which 15 can be used as PWM outputs). These pins can be configured as digital input pin to read the logic value (0 or 1). Or used as digital output pin to drive different modules like LED, relay, etc. Using <code>pinMode()</code> , <code>digitalWrite()</code> , and <code>digitalRead()</code> functions.
2	<b>GND</b> Ground pins
3	<b>AREF</b> Reference voltage (0-5V) for analog inputs. Used with <code>analogReference()</code> . Configures the reference voltage used for analog input (i.e. the value used as the top of the input range).
4	<b>SDA</b> IIC communication pin
5	<b>SCL</b> IIC communication pin
6	<b>ICSP (In-Circuit Serial Programming) Header</b> the AVR, an Arduino micro-program header consisting of MOSI, MISO, SCK, RESET, VCC, and GND. Connected to the ATMEGA16U2-MU. When connecting to PC, program the firmware to ATMEGA16U2-MU.
7	<b>USB Connection</b> Arduino board can be powered via USB connector. All you needed to do is connecting the USB port to PC using a USB cable.
8	<b>D13 LED</b> 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.
9	<b>ATMEGA 16U2-MU</b> USB to serial chip, can convert the USB signal into serial port signal.
10	<b>TX LED</b> Onboard you can find the label: TX (transmit) When Arduino board communicates via serial port, send the message, TX led flashes.
11	<b>RX LED</b> Onboard you can find the label: RX(receive ) When Arduino board communicates via serial port, receive the message, RX led flashes.
12	<b>Crystal Oscillator</b> How does Arduino calculate time? by using a crystal oscillator. The number printed on the top of the Arduino crystal is 16.000H9H. It tells us that the frequency is 16,000,000 Hertz or 16MHz.
13	<b>Voltage Regulator</b> Convert an external input DC7-12V voltage into DC 5V, then switch DC 5V to the processor and other components. Output DC 5V, the drive current is 2A.
14	<b>DC Power Jack</b> Arduino board can be supplied with an external power DC7-12V from the DC power jack.
15	<b>IOREF</b> This pin on the 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 for working with the 5V or 3.3V.
16	<b>RESET Header</b> Connect an external button to reset the board. The function is the same as reset button.
17	<b>Power Pin 3V3</b> Provides 3.3V voltage output
18	<b>Power Pin 5V</b> Provides 5V voltage output
19	<b>Vin</b> You can supply an external power input DC7-12V through this pin to Arduino board.
20	<b>Analog Pins</b> Onboard has 16 analog inputs, labeled A0 to A15.
21	<b>RESET Button</b> You can reset your Arduino board, for example, start the program from the initial status. You can use the RESET button.
22	<b>ICSP (In-Circuit Serial Programming) Header</b> the AVR, an Arduino micro-program header consisting of MOSI, MISO, SCK, RESET, VCC, and GND. It is often called the SPI (serial peripheral interface) and can be considered an “extension” of the output. In fact, slave the output devices to the SPI bus host. When connecting to PC, program the firmware to ATMEGA2560-16AU.
23	<b>Microcontroller</b> Each Arduino board has its own microcontroller. You can regard it as the brain of your board. The main IC (integrated circuit) on the Arduino is slightly different from the panel pair. Microcontrollers are usually from ATMEL. Before you load a new program on the Arduino IDE, you must know what IC is on your board. This information can be checked at the top of IC.
238	<b>Chapter 8: KS0342 Keystudio MEGA Compatible advanced development board</b>
24	<b>Power LED Indicator</b> Powering the Arduino, LED on means that your circuit board is correctly powered





Specialized Functions of Some Pins:

- **Serial Communication:** D0 (RX0) and D1 (TX1); Serial 1: D19 (RX1) and D18 (TX1); Serial 2: D17 (RX2) and D16 (TX2); Serial 3: D15 (RX3) and D14 (TX3).

Used to receive (RX) and transmit (TX) TTL serial data. Pins 0 and 1 are also connected to the corresponding pins of the ATmega16U2 USB-to-TTL Serial chip.

- **PWM Pins (Pulse-Width Modulation):** D2 to D13, and D44 to D46.

Provide 8-bit PWM output with the `analogWrite()` function.

- **External Interrupts:** D2 (interrupt 0), D3 (interrupt 1), D18 (interrupt 5), D19 (interrupt 4), D20 (interrupt 3), and D21 (interrupt 2).

These pins can be configured to trigger an interrupt on a low level, a rising or falling edge, or a change in level. See the `attachInterrupt()` function for details.

- **SPI communication:** D53 (SS), D52 (SCK), D51 (MOSI), D50 (MISO).

These pins support SPI communication using the [SPI library](#). The SPI pins are also broken out on the ICSP header, which is physically compatible with the Arduino Uno.

- **IIC communication:** D20 (SDA); D21 (SCL). Support TWI communication using the [Wire library](#).

Tips:

- Automatic (Software) Reset:

Rather than requiring a physical press of the reset button before an upload, the MEGA board is designed in a way that allows it to be reset by software running on a connected computer.

- The MEGA 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; see [this forum thread](#) for details.

Detailed Use with ARDUINO Software as follows:

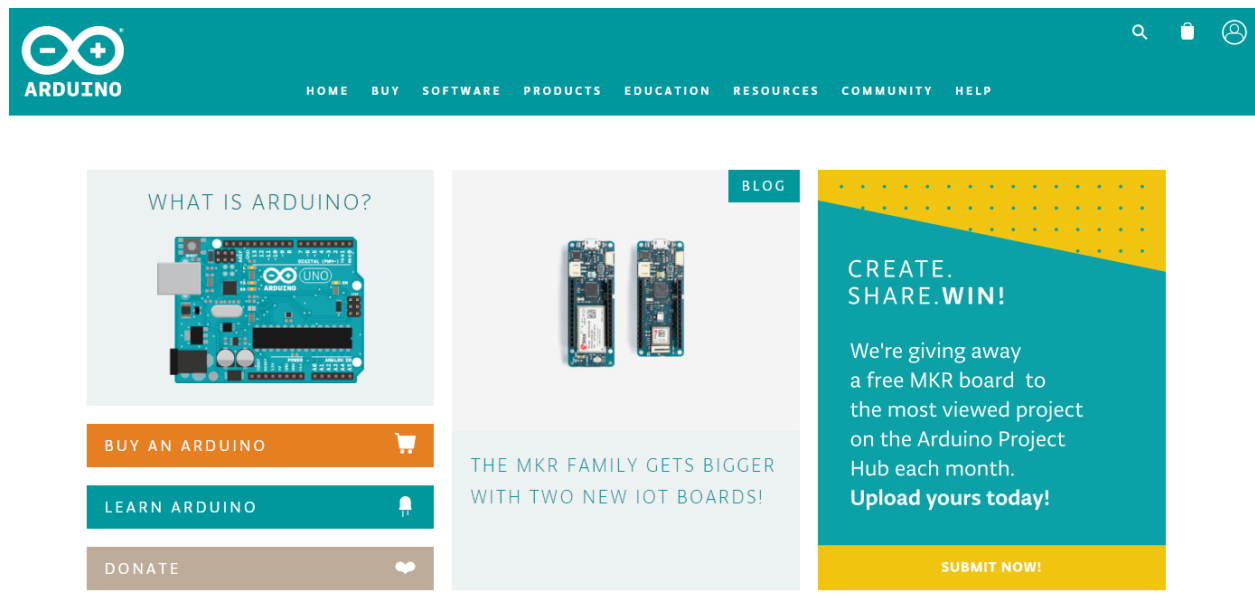
Step1| Download the Arduino environment (IDE)

When you get the MEGA development board, first you should install the Arduino software and driver.

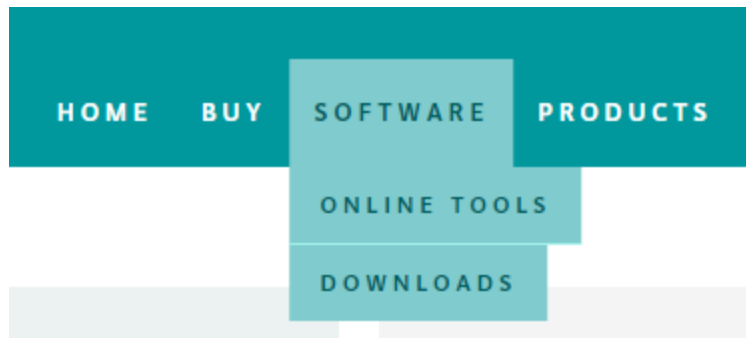
We usually use the Windows software Arduino 1.5.6 version. You can download it from the link below:

<https://www.arduino.cc/en/Main/OldSoftwareReleases\1.5.x>

Or you can browse the ARDUINO website to download the latest version from this link, <https://www.arduino.cc>, pop up the following interface.



Then click the **SOFTWARE** on the browse bar, you will have two options ONLINE TOOLS and DOWNLOADS.



Click **DOWNLOADS**, it will appear the latest software version of ARDUINO 1.8.5 shown as below.

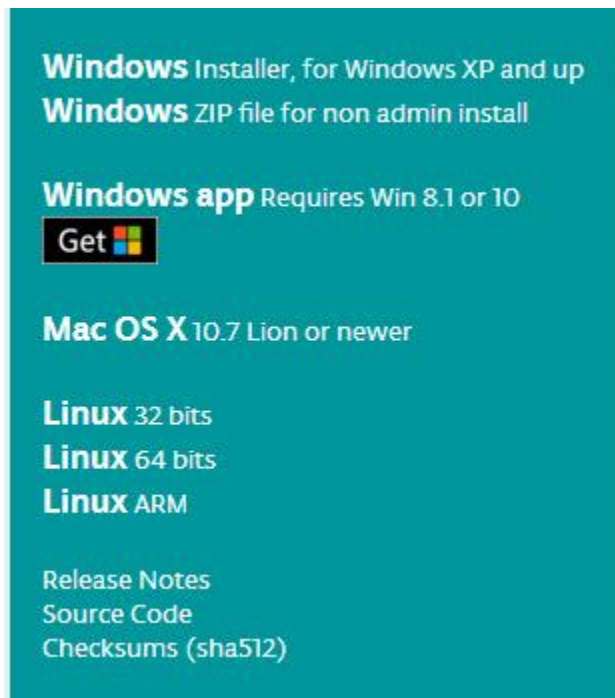
## Download the Arduino IDE

A screenshot of the Arduino 1.8.5 download page. On the left, there is a large teal circle containing a white infinity symbol with a minus sign on the left and a plus sign on the right. To the right of this icon, the text reads: 'ARDUINO 1.8.5', '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.', and 'This software can be used with any Arduino board. Refer to the [Getting Started](#) page for installation instructions.' On the right side of the page, there is a teal sidebar with white text listing download options: 'Windows Installer, for Windows XP and up', 'Windows ZIP file for non admin install', 'Windows app Requires Win 8.1 or 10' (with a 'Get' button), 'Mac OS X 10.7 Lion or newer', 'Linux 32 bits', 'Linux 64 bits', 'Linux ARM', 'Release Notes', 'Source Code', and 'Checksums (sha512)'.

In this software page, on the right side you can see the version of development software for different operating systems. ARDUINO has a powerful compatibility. You should download the software that is compatible with the operating system of your computer.

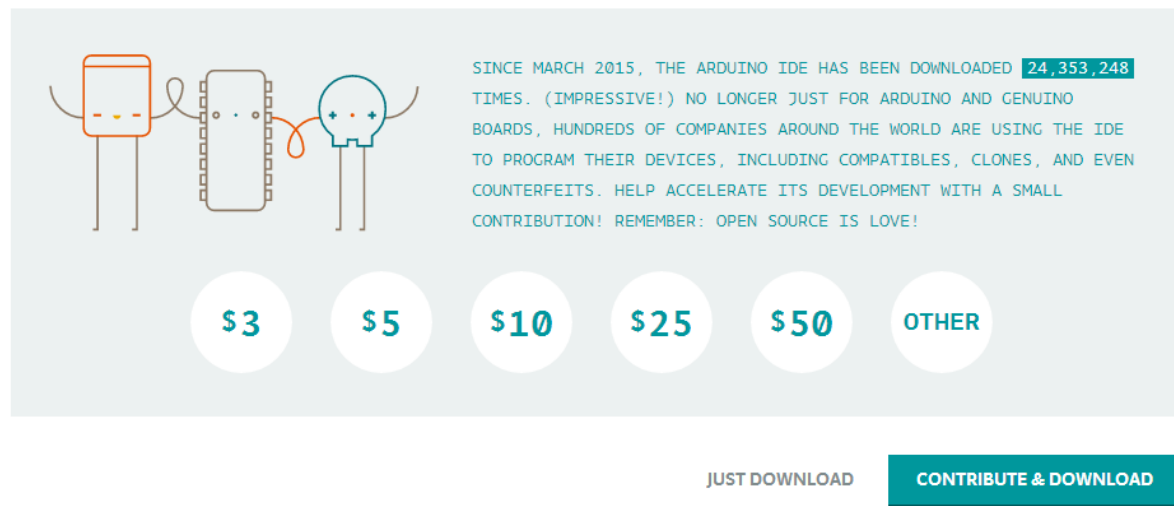
We will take **WINDOWS system** as an example here. There are also two options under Windows system, one is installed version, the other is non-installed version.

For simple installed version, first click **Windows Installer**, you will get the following page.



## Contribute to the Arduino Software

Consider supporting the Arduino Software by contributing to its development. (US tax payers, please note this contribution is not tax deductible). [Learn more on how your contribution will be used.](#)



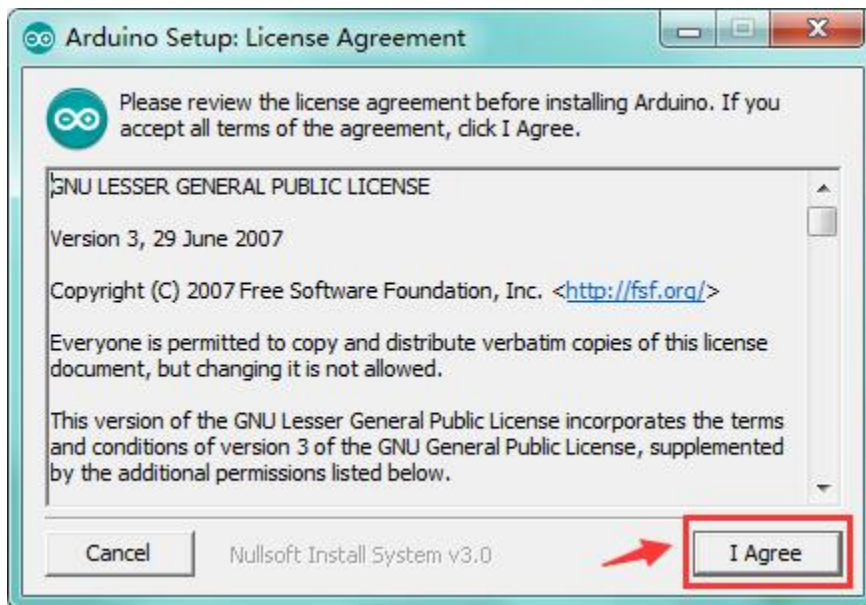
This way you just need to click JUST DOWNLOAD, then click the downloaded file to install it.

For non-installed version, first click Windows ZIP file, you will also get the pop-up interface as the above figure.

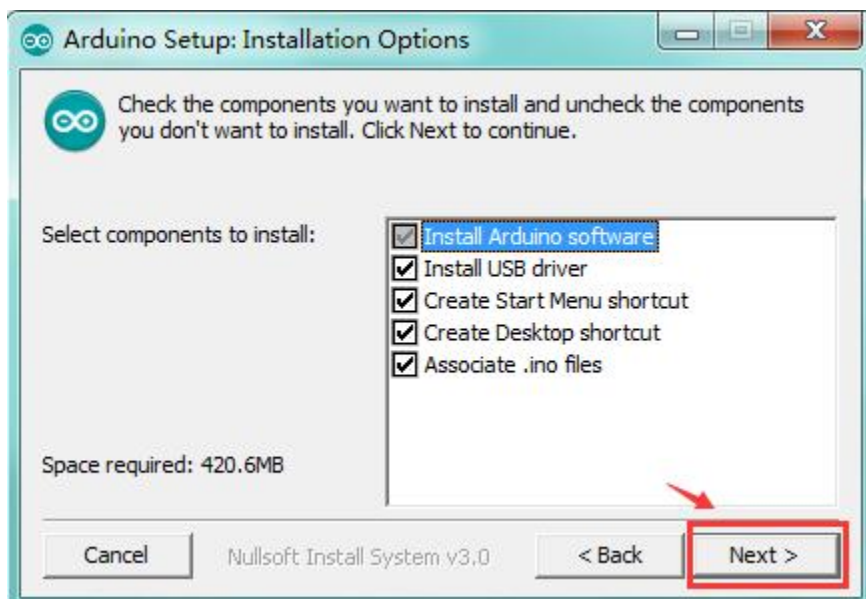
Click JUST DOWNLOAD, and when the ZIP file is downloaded well to your computer, you can directly unzip the file and click the icon of ARDUINO software to start it.

### Installing Arduino (Windows):

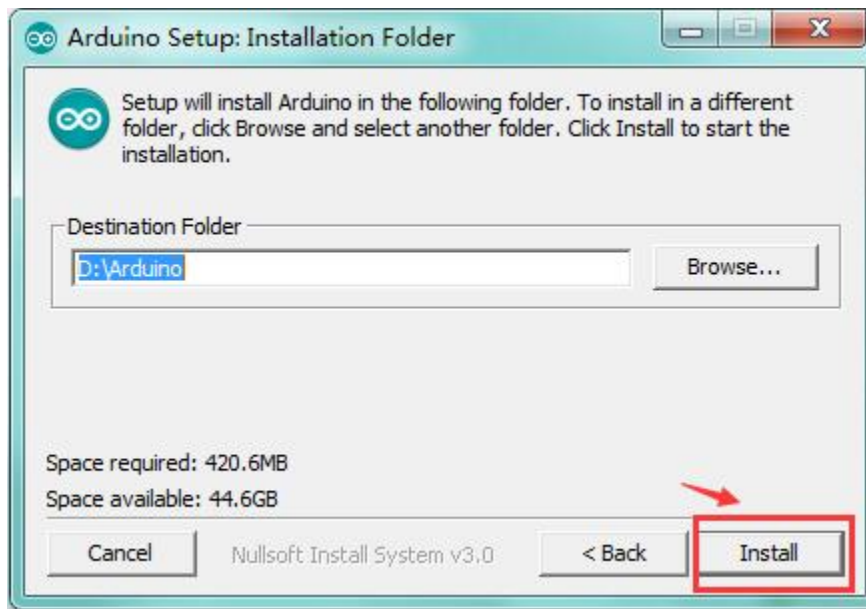
Install Arduino with the exe. Installation package downloaded well.



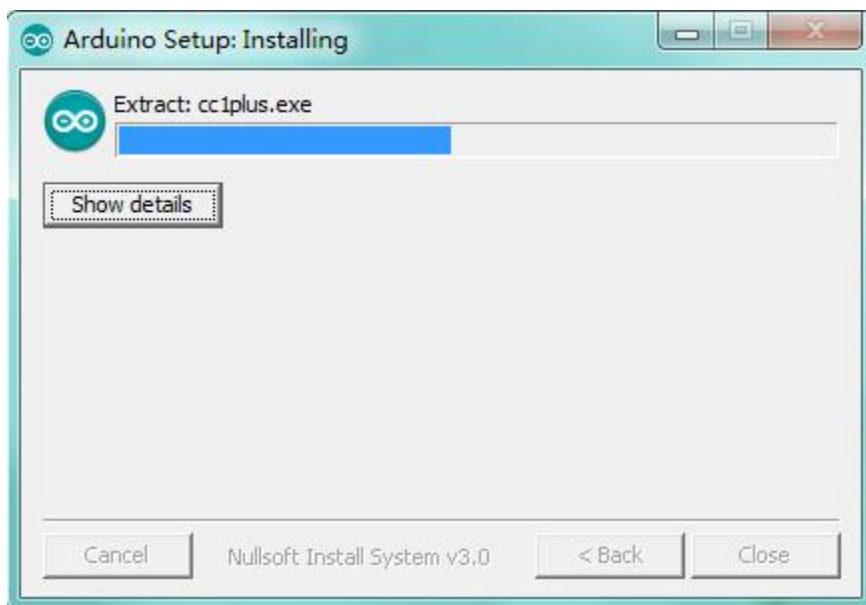
Click “I Agree” to see the following interface.



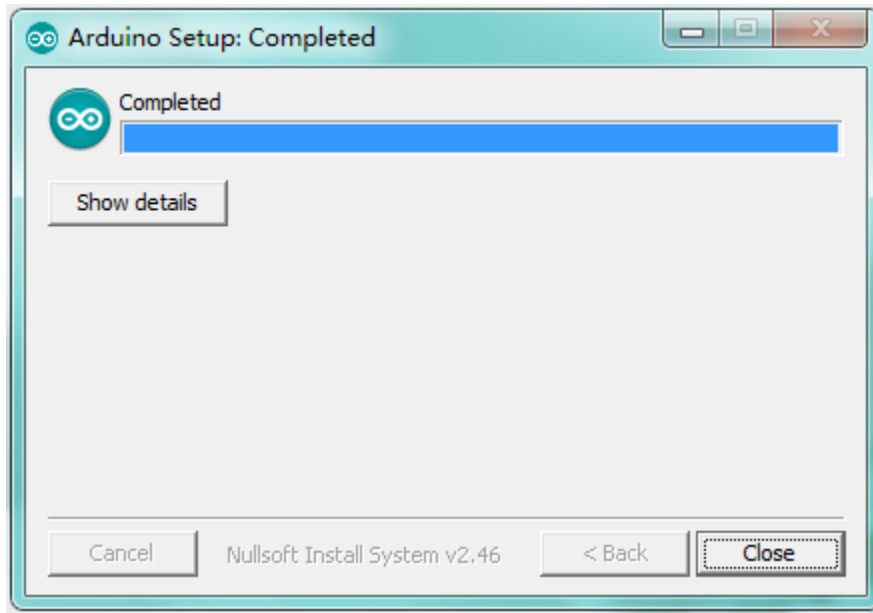
Click “Next”. Pop up the interface below.



You can press Browse... to choose an installation path or directly type in the directory you want. Then click “Install” to initiate installation.



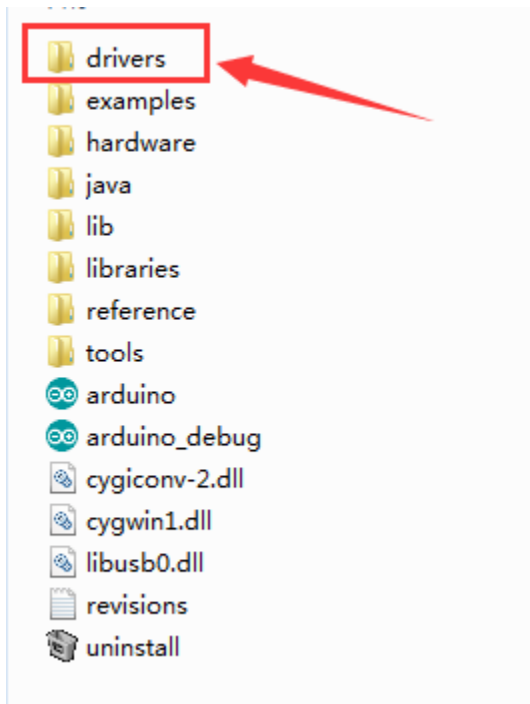
Wait for the installing process, if appear the interface of Window Security, just continue to click Install to finish the installation.



### Installing Driver:

Next, we will introduce the driver installation of MEGA development board. The driver installation may have slight differences in different computer systems. So in the following let's move on to the driver installation in the WIN 7 system.

The Arduino folder contains both the Arduino program itself and the drivers that allow the Arduino to be connected to your computer by a USB cable. Before we launch the Arduino software, you are going to install the USB drivers.

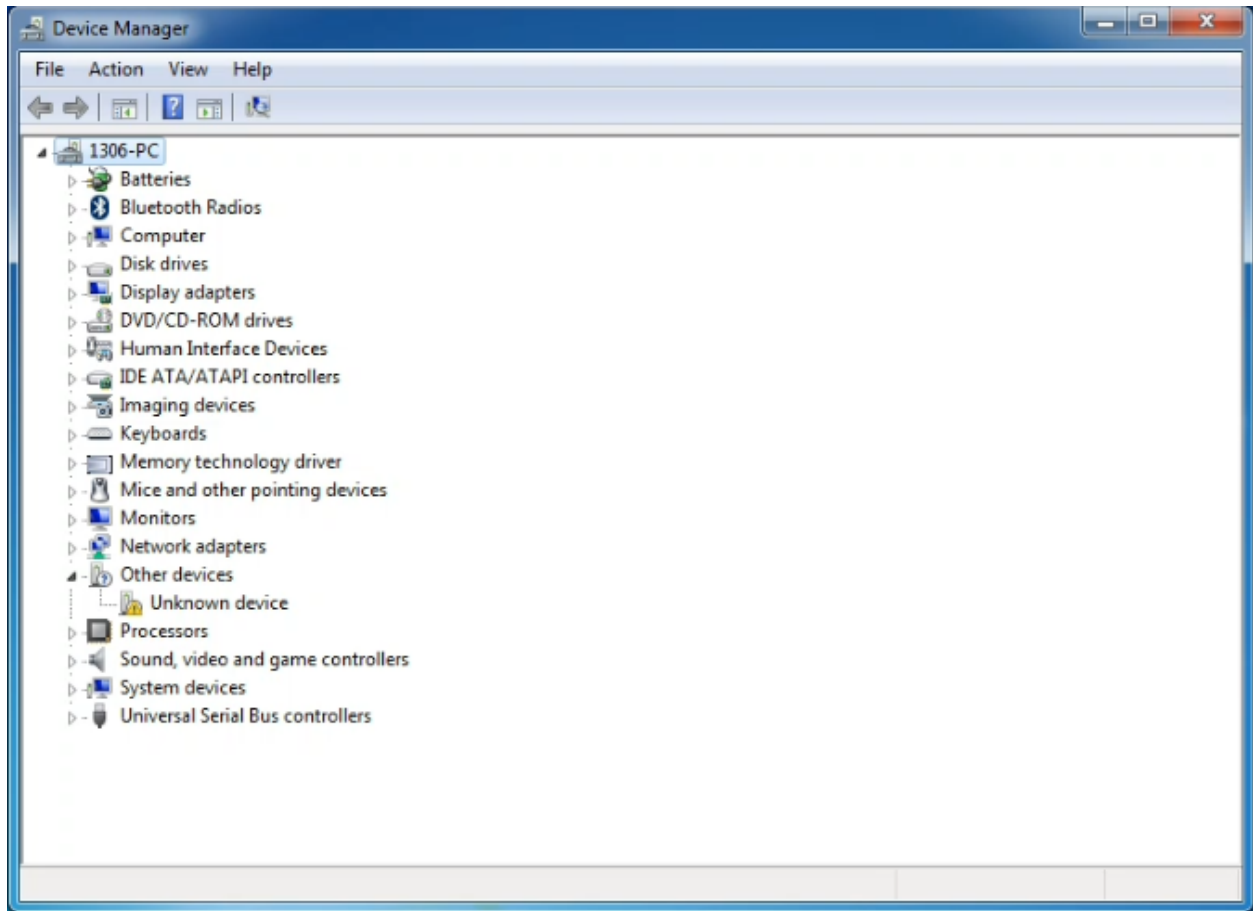


Plug one end of your USB cable into the Arduino and the other into a USB socket on your computer.

When you connect MEGA board to your computer at the first time, right click the icon of your “Computer” —>for “Properties”—> click the “Device manager”, under “Other Devices”, you should see an icon for “Unknown device”

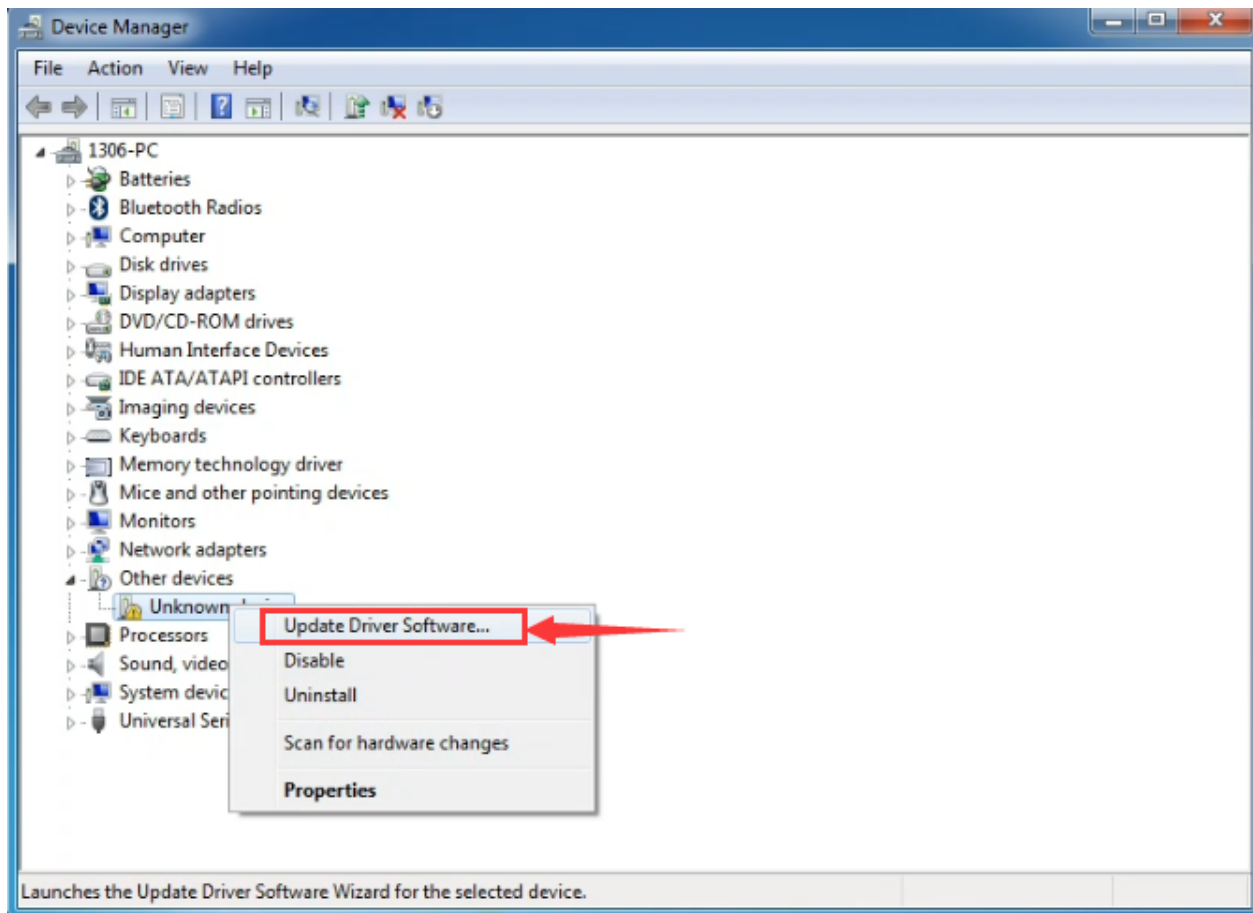


with a little yellow warning triangle next to it. This is your Arduino.

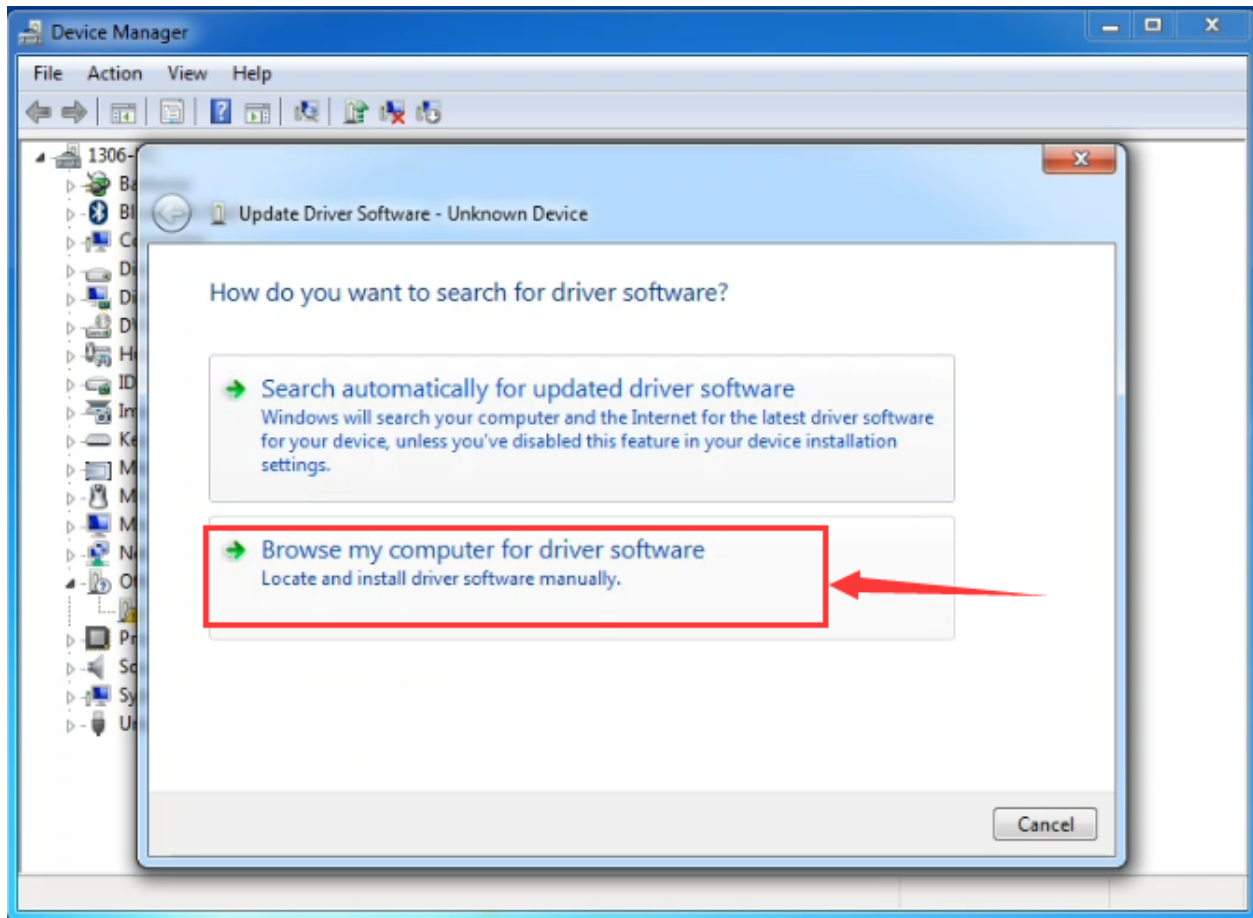


Then right-click on the device and select the top menu option (Update Driver Software...) shown as the figure below.

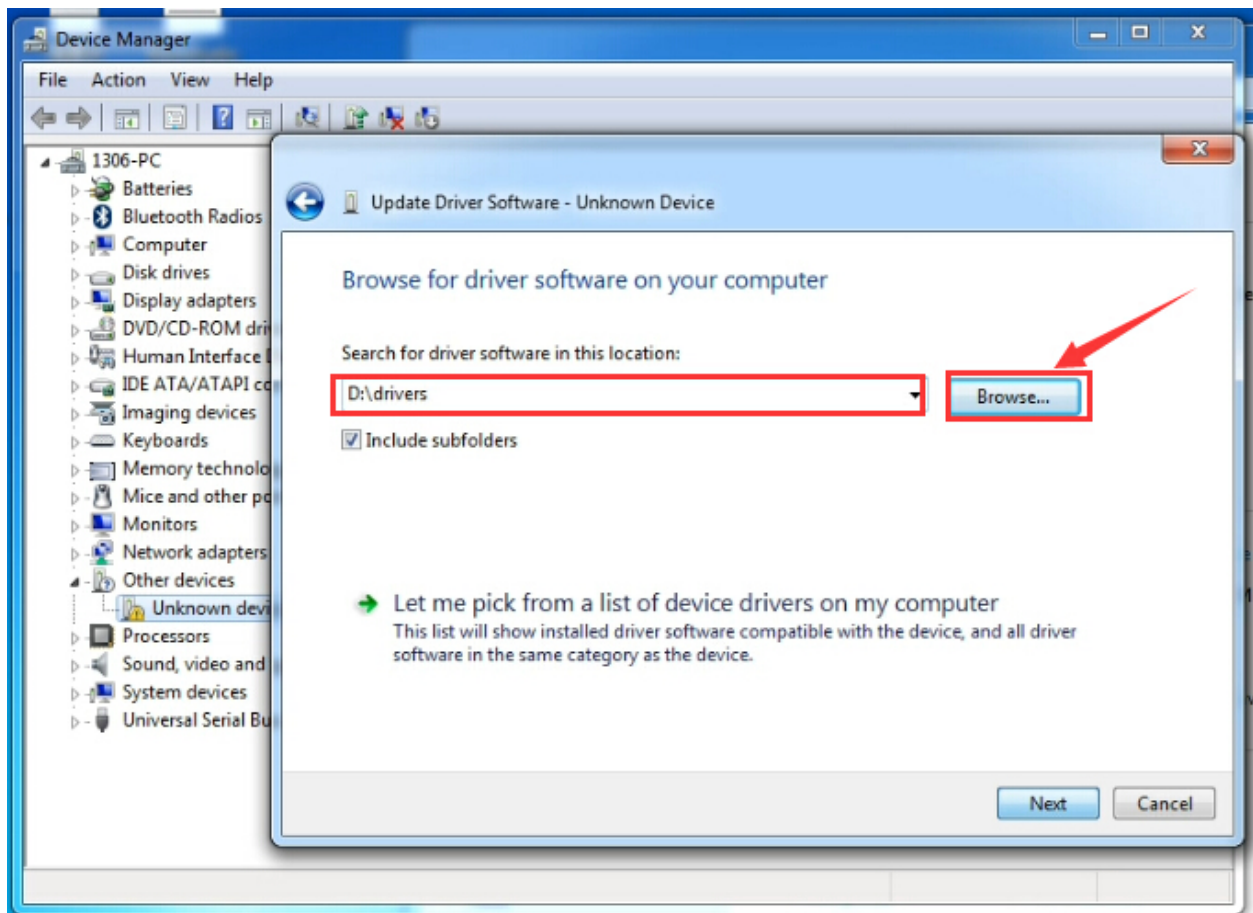




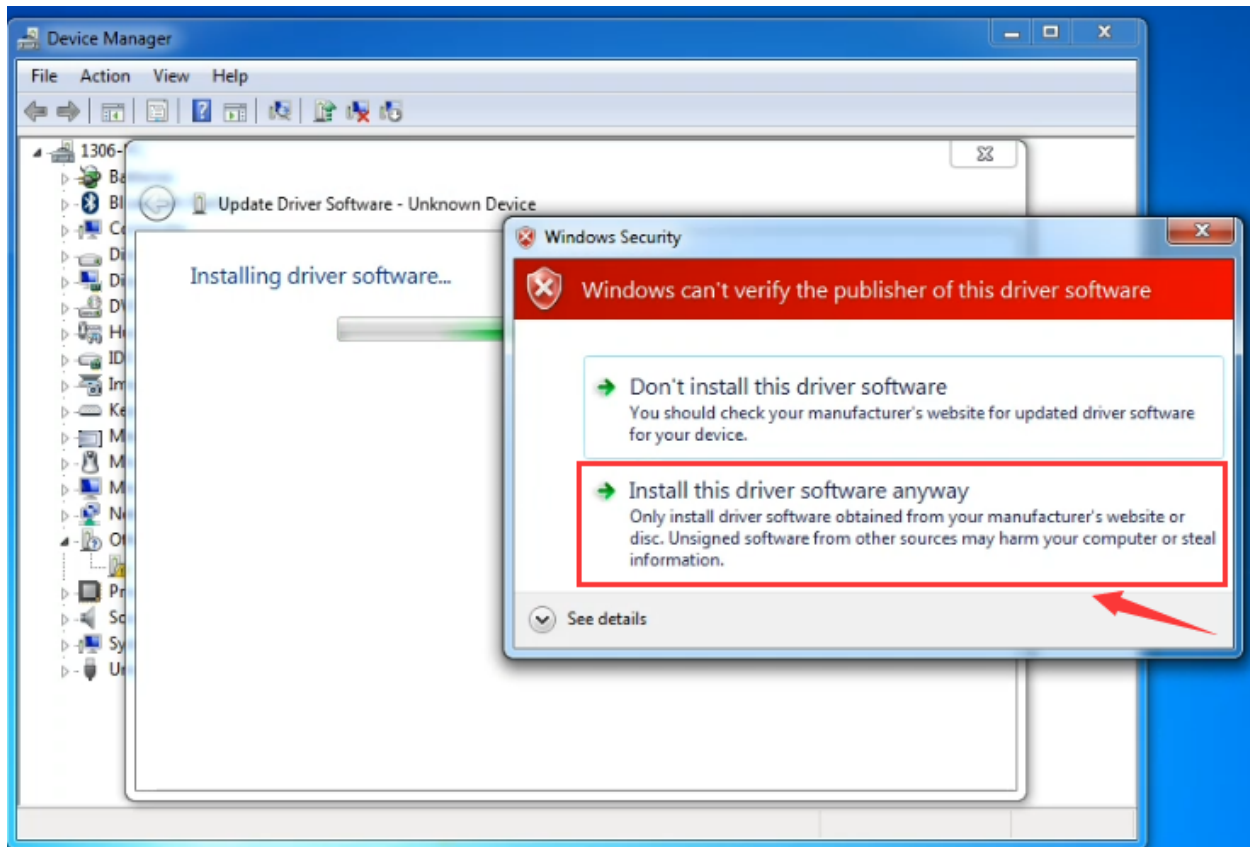
It will then be prompted to either “Search Automatically for updated driver software” or “Browse my computer for driver software”. Shown as below. In this page, select “Browse my computer for driver software”.



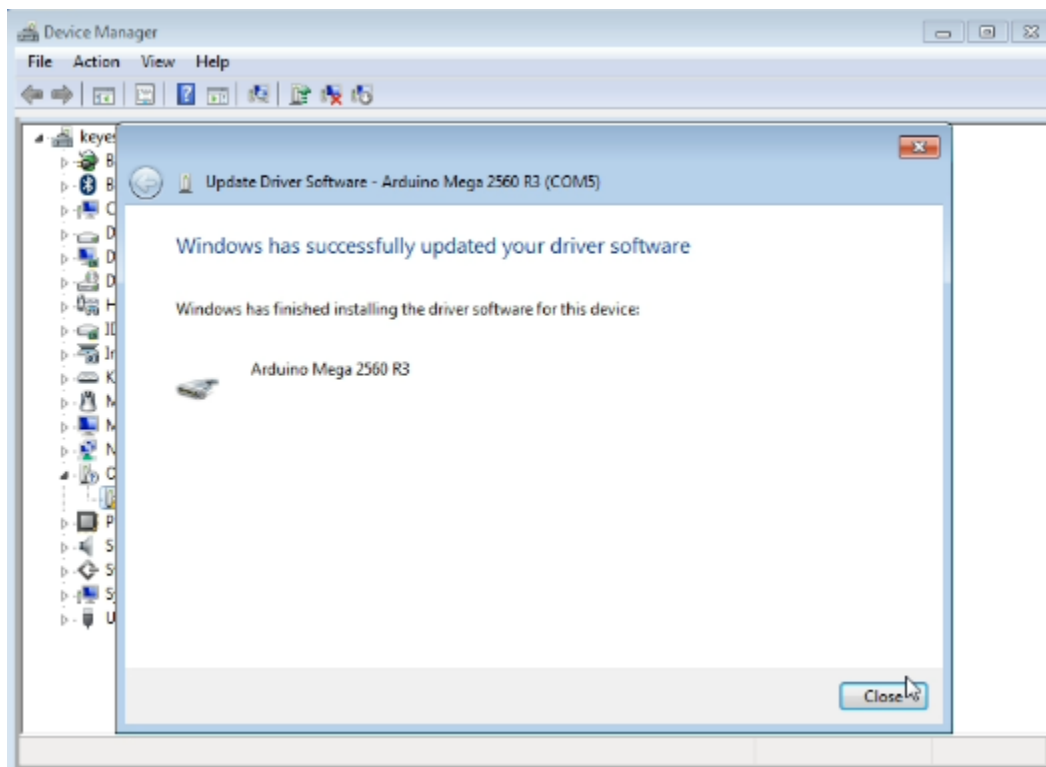
After that, select the option to browse and navigate to the “drivers” folder of Arduino installation.



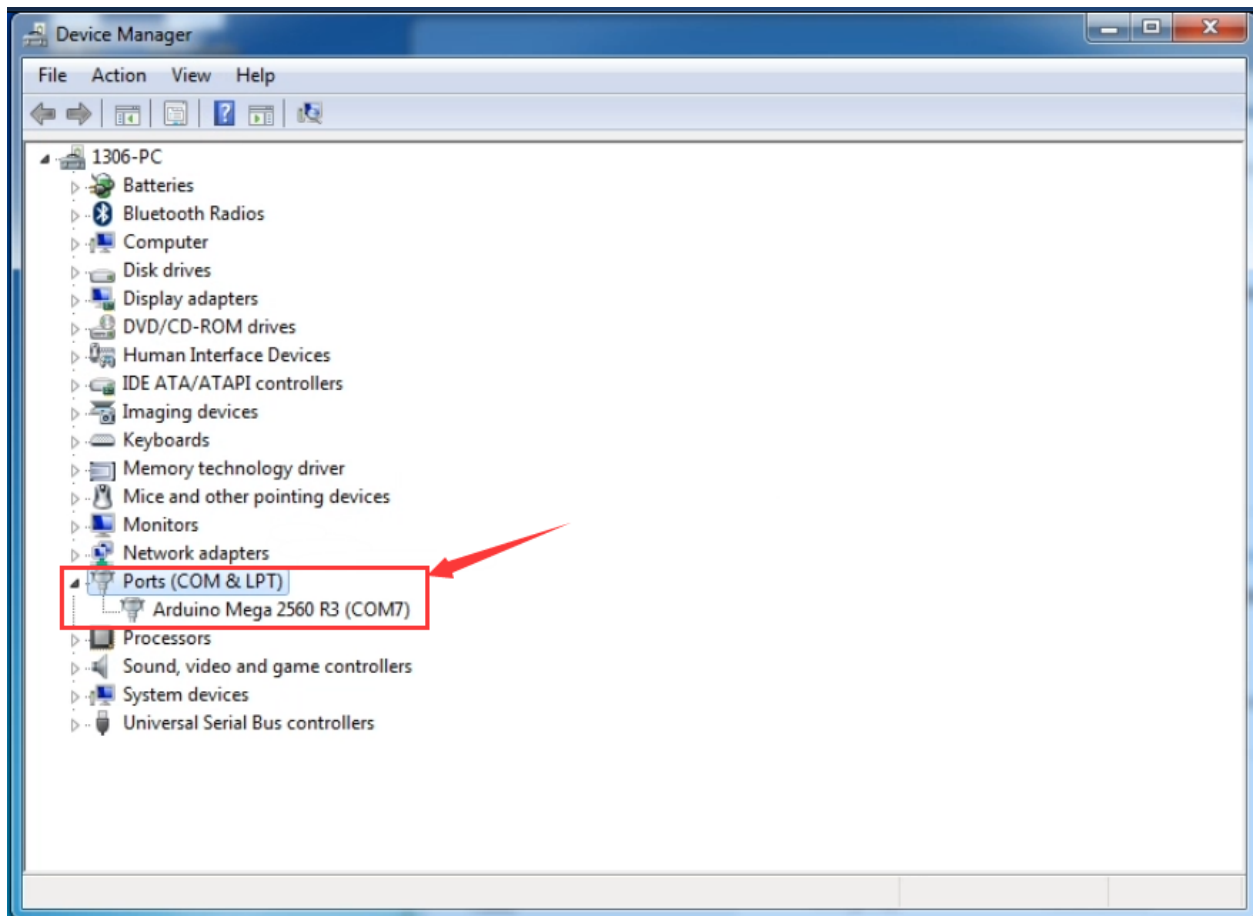
Click “Next” and you may get a security warning, if so, allow the software to be installed. Shown as below.



Installation completed, click “Close”.



Up to now, the driver is installed well. Then you can right click “Computer” —> “Properties” —> “Device manager”, you should see the device shown below.

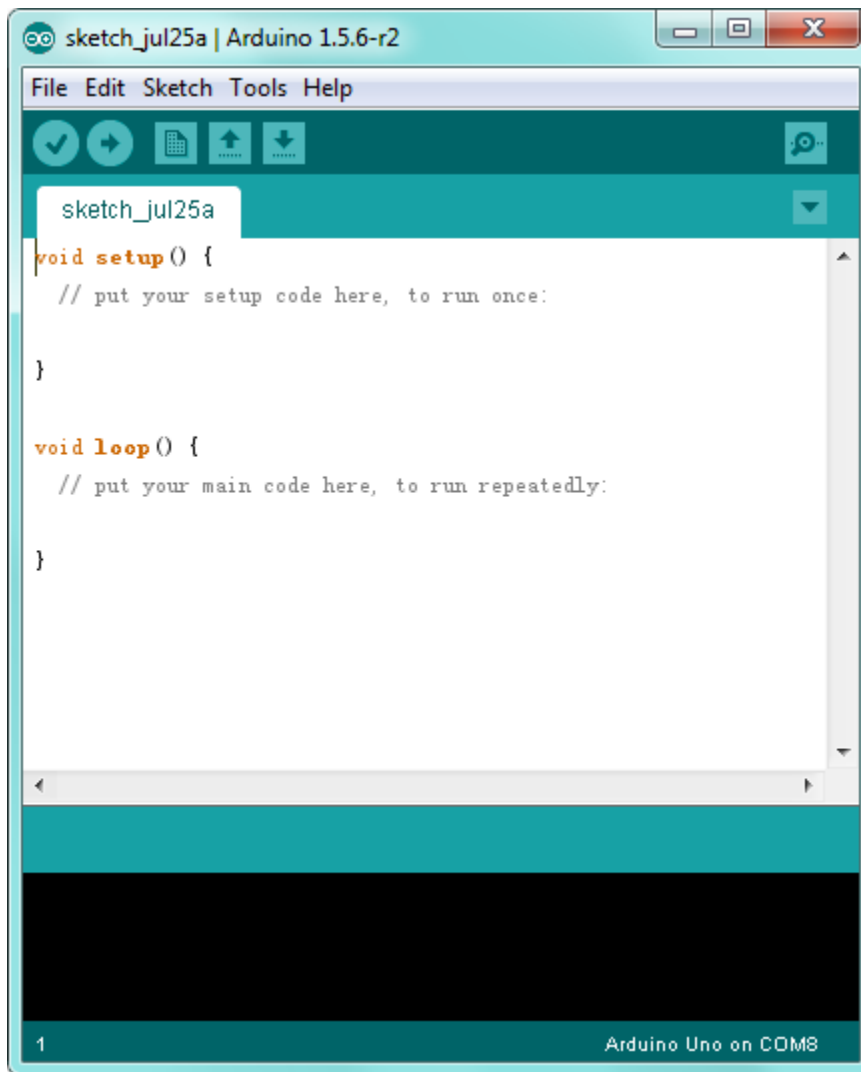


### Introduction for Arduino IDE Toolbar:

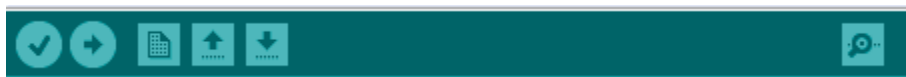
Double-click the icon of Arduino software downloaded well, you will get the interface shown below.









(Note: if the Arduino software loads in the wrong language, you can change it in the preferences dialog. See the [environment](#) page for details.)



The functions of each button on the Toolbar are listed below: <http://wiki.keyestudio.com/index.php/File:IDE.png>



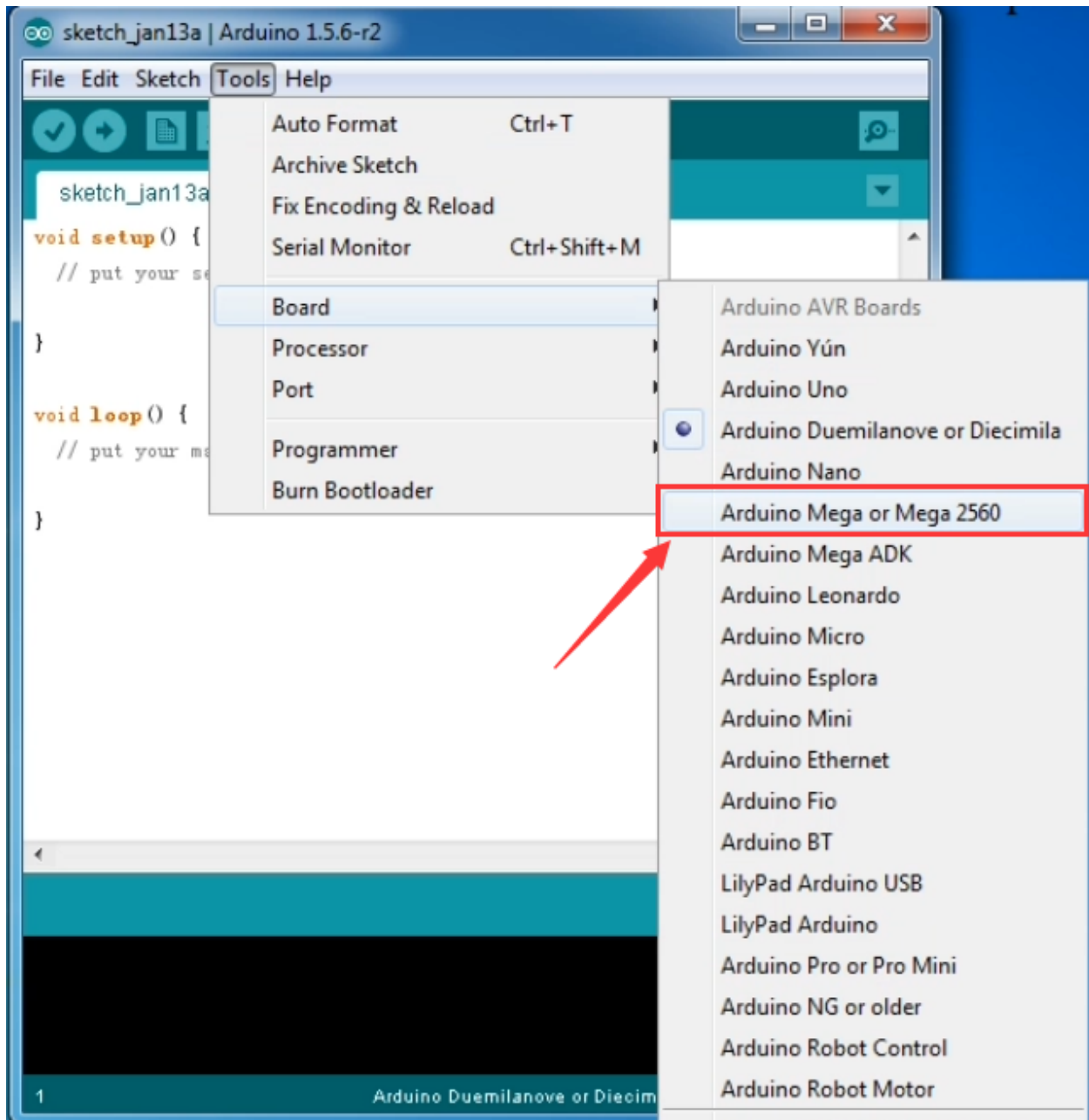
	<b>Verify/Compile</b>	Check the code for errors
	<b>Upload</b>	Upload the current Sketch to the Arduino
	<b>New</b>	Create a new blank Sketch
	<b>Open</b>	Show a list of Sketches
	<b>Save</b>	Save the current Sketch
	<b>Serial Monitor</b>	Display the serial data being sent from the Arduino

Step2| Connect the board

Connect the MEGA board to your computer using the USB cable. The power LED should go on.

Step3| Select the Arduino Board

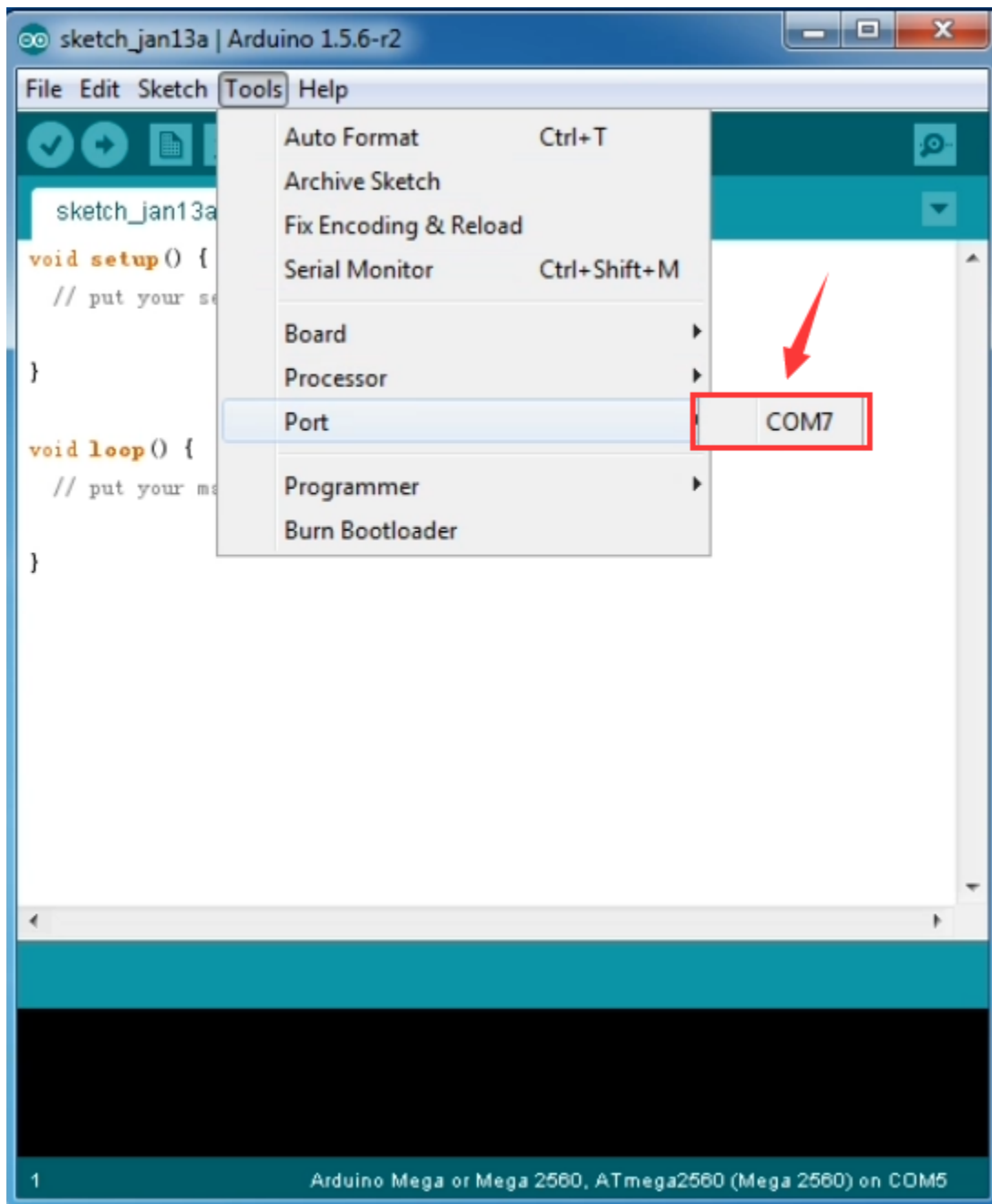
Open the Arduino IDE, you'll need to click the "Tools", then select the Board that corresponds to your Arduino.



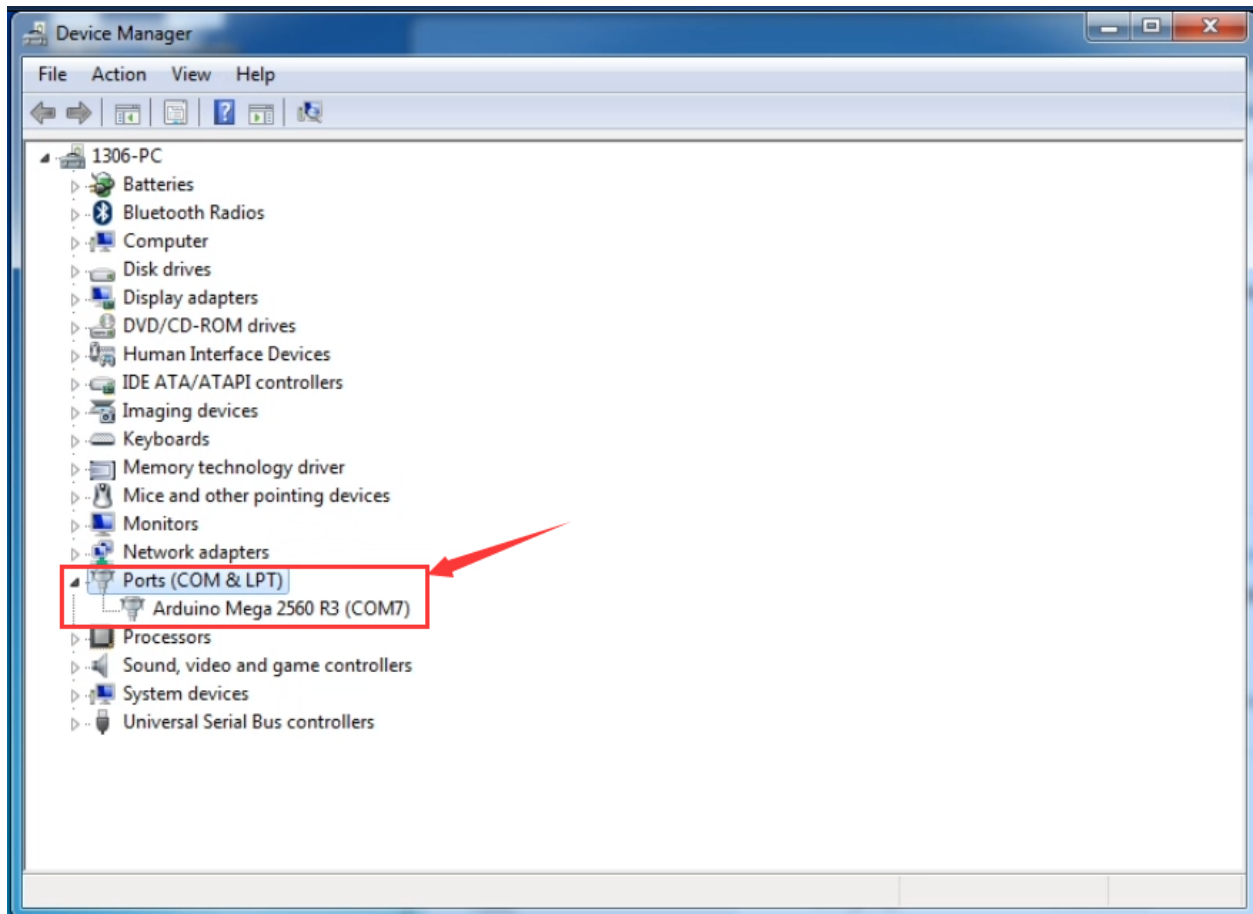
Step4| Select your serial port

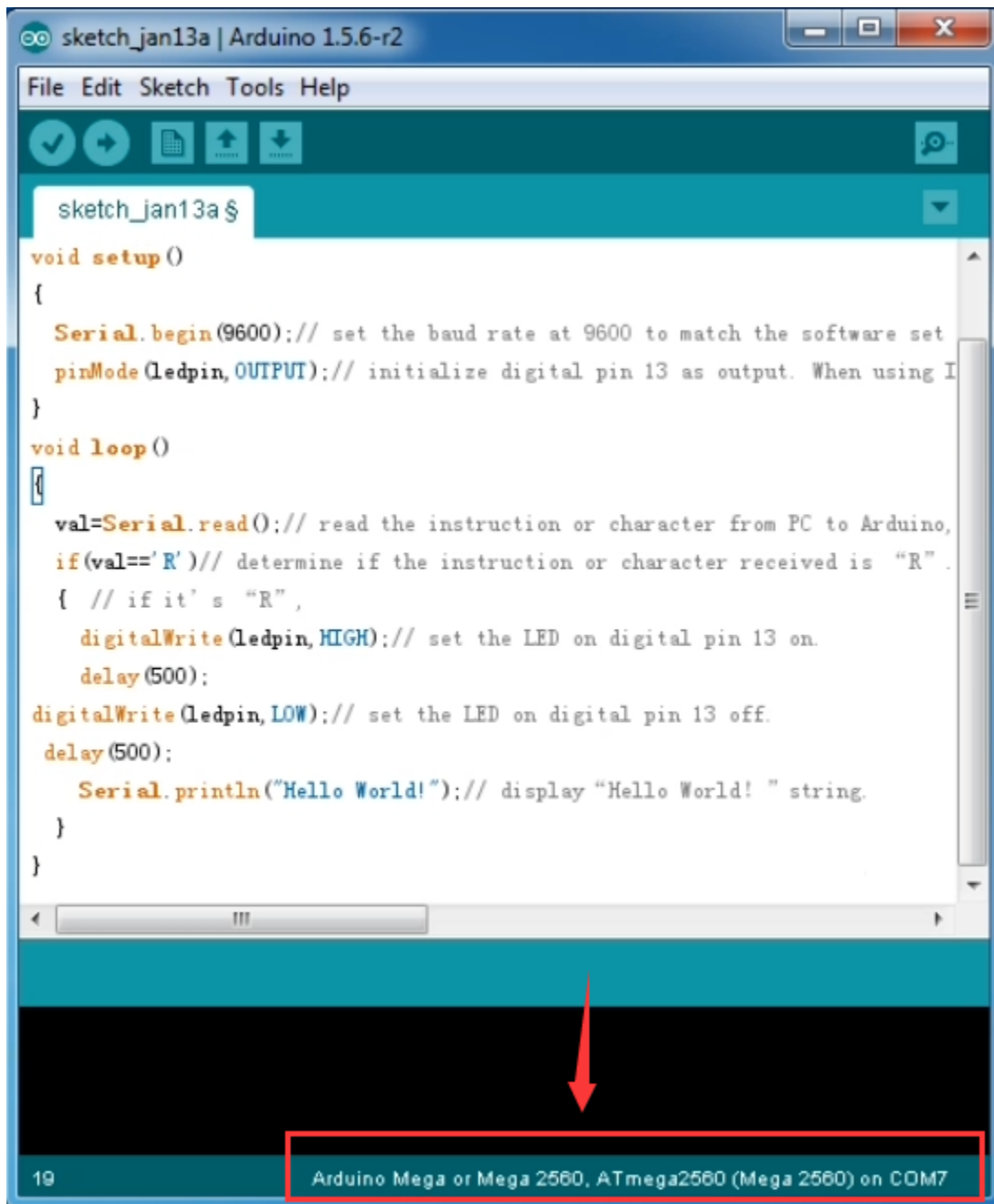
Select the serial device of the Arduino board from the **Tools | Serial Port** menu. This is likely to be COM3 or higher (COM1 and COM2 are usually reserved for hardware serial ports). To find out, you can disconnect your Arduino board and re-open the menu; the entry that disappears should be the Arduino board. Reconnect the board and select that serial port.





**Note:** to avoid errors, the COM Port should keep the same as the Ports shown on Device Manager.





Step5| Upload the Program

Below is an example program for displaying the Hello World!

Copy and paste the code to the Arduino environment IDE.

////////////////////////////////////

int val;//define variable val

int ledpin=13;// define digital pin13

void setup()

{

Serial.begin(9600);// set the baud rate at 9600. When connected to a specific device, (e.g. bluetooth), the baud rate needs to be the same with the device.

pinMode(ledpin,OUTPUT);// initialize digital pin 13 as output. When using I/O ports on an Arduino, this kind of set up is always needed.

```
}
```

```
void loop()
```

```
{
```

```
val=Serial.read();// read the instruction or character from PC to Arduino, and assign them to Val.
```

```
if(val=='R')// determine if the instruction or character received is an "R".
```

```
{ // if it's "R",
```

```
digitalWrite(ledpin,HIGH);// set the LED on digital pin 13 on.
```

```
delay(500);
```

```
digitalWrite(ledpin,LOW);// set the LED on digital pin 13 off.
```

```
delay(500);
```

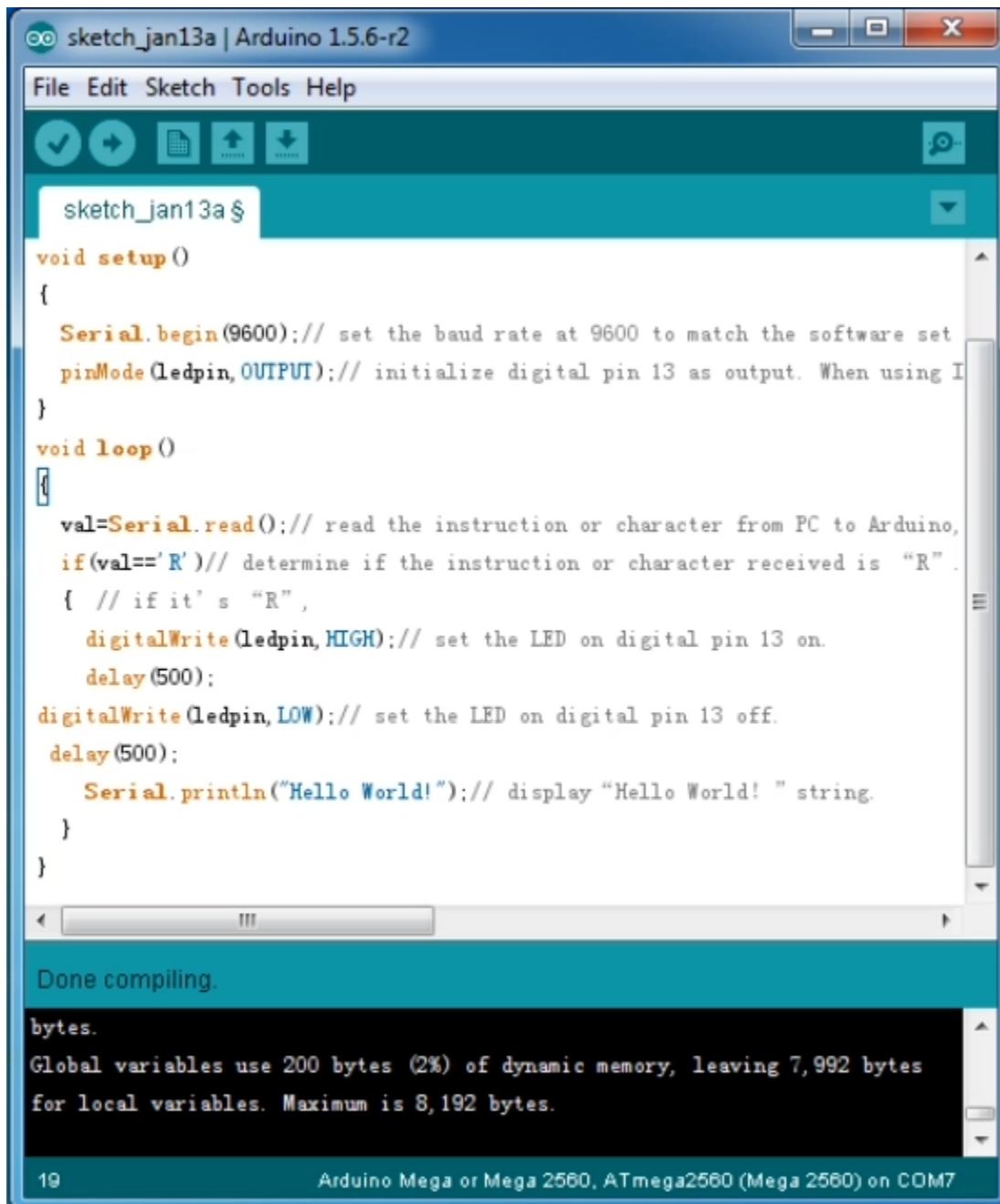
```
Serial.println("Hello World!");// display"Hello World".
```

```
}
```

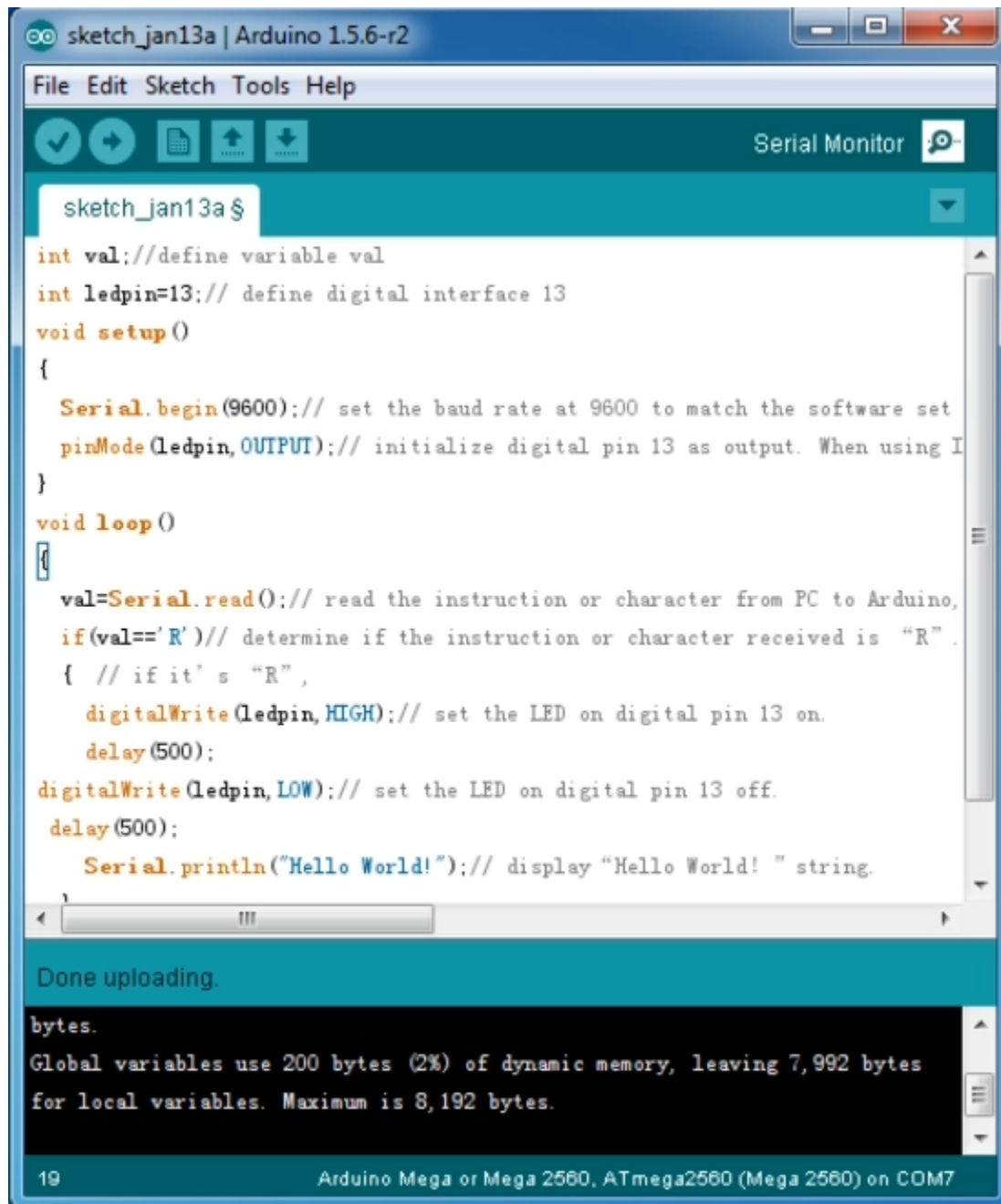
```
}
```

```
////////////////////////////////////
```

Then click verify button to check the errors. If compiling successfully, the message "Done compiling." will appear in the status bar.



After that, click the “Upload” button to upload the code. Wait a few seconds - you should see the RX and TX leds on the board flashing. If the upload is successful, the message “Done uploading.” will appear in the status bar.

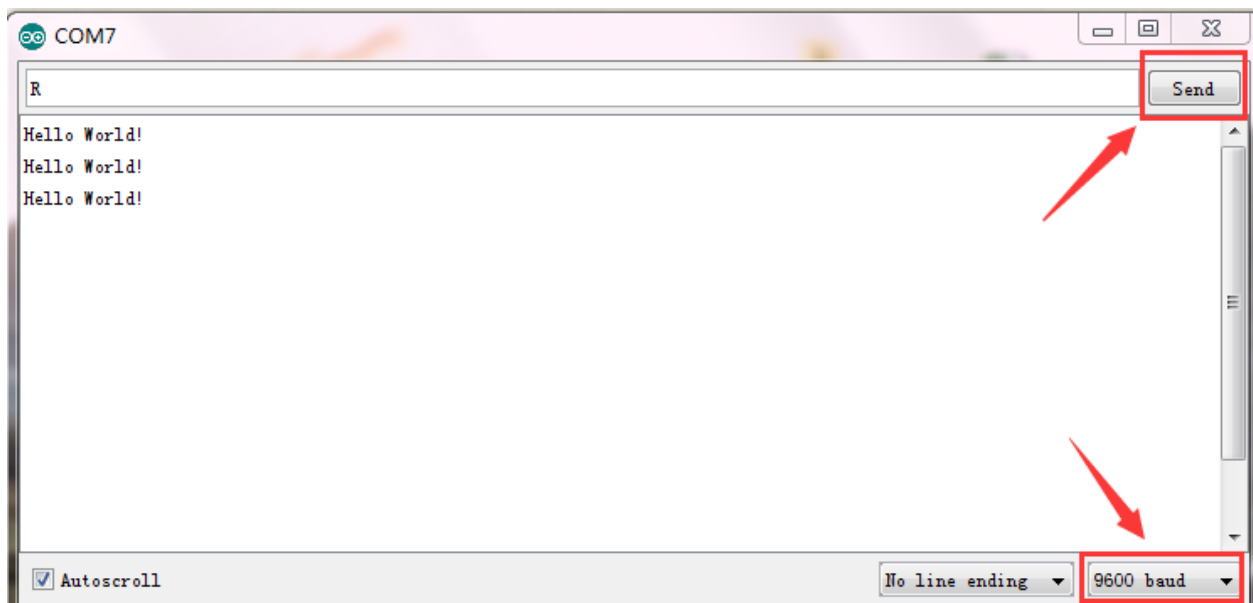


Step6| Open the Serial Monitor

After that, click the serial monitor button to open the serial monitor.

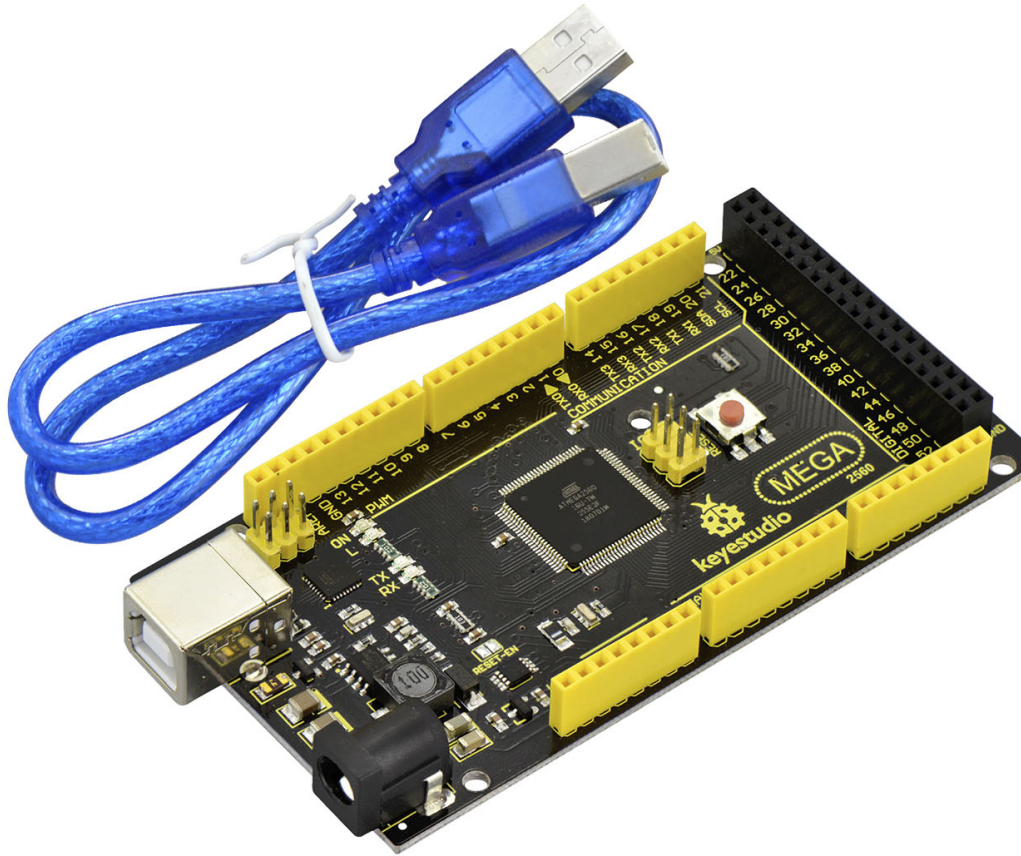


Then set the baud rate as 9600, enter an "R" and click Send, you should see the TX led on the board blink once, and then D13 led blink once, finally "Hello World!" is showed on the monitor, the RX led blink once.



Package Included:

- Keyestudio MEGA Advanced board \* 1pcs
- USB cable \* 1pcs



Other Links:

You might also want to look at:

[the examples](#) for using various sensors and actuators;

[the reference](#) for the Arduino language;

You can download the datasheet from the link:

[https://drive.google.com/open?id=1EdtG7by2aCPR55Q5uYlwcwk\\_e5cWnh2M](https://drive.google.com/open?id=1EdtG7by2aCPR55Q5uYlwcwk_e5cWnh2M)

Software Download:

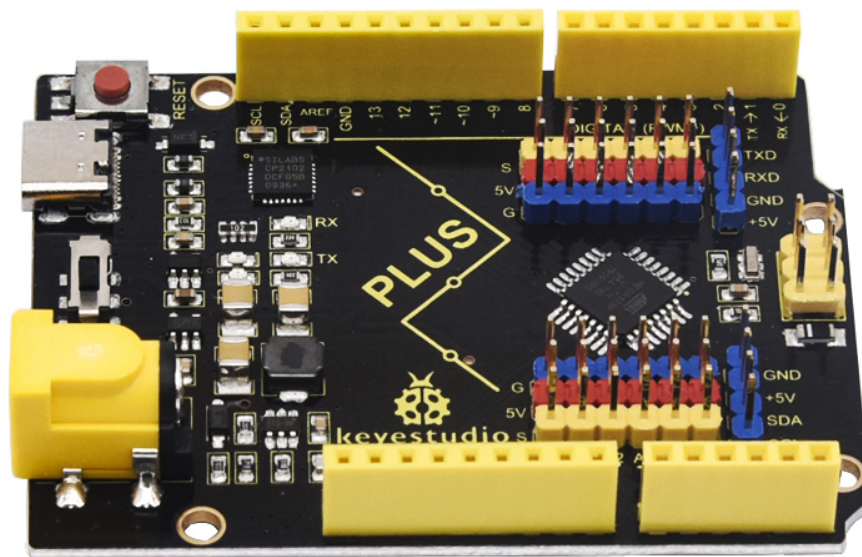
<https://drive.google.com/open?id=12D-JkXdNm03Qt4dIPQr3RP6OmgXqpVHc>

Troubleshooting:

If you have problems, please see the [troubleshooting](#) suggestions.



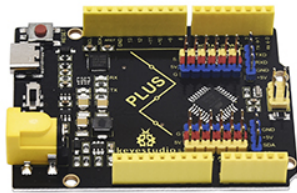
## KS0486 KEYESTUDIO PLUS DEVELOPMENT BOARD



### 1.Description

Doing experiments with electronic products, we often program on the Arduino IDE development environment with Arduino series microcontrollers.

Keystudio PLUS control board is fully compatible with Arduino IDE development environment. It is as same as the Arduino UNO R3 board. Moreover, some improvements we made highly strengthen its function(as shown below). In order to wire efficiently, we equip it with a 1m USB cable with type-c interface for you.



Keystudio PLUS

VS



Arduino UNO R3

1. USB CP2102, stability and compatibility are better for turn chip



1. USB turnstile chip is 16 U2, Some systems are not compatible

2. Working voltage can be selected 3.3 V or 5 V, can connect 3.3 V sensor



2. Only 5 V working voltage, no way to connect 3.3 V sensor

3. Two more IO mouth, A6,A7, the best



3. Design did not leave these 2 IO ports, defective

4. Extended serial communication and I2C interface, can be easily connected to similar devices



4. No expansion port, more complex connection required

5. Unique DC-DC power supply design, working voltage 5 V, the current of 2 A, can directly drive some high current load, such as steering gear and motor



5. Working voltage 5 V, the current only 1 A, can not drive the equipment with high current.

6. Extension of 6 PWM ports and 6 analog port interfaces to connect sensors directly



6. No expansion port, more complex connection required

7. Extended serial communication and I2C interface, can be easily connected to similar devices



7. No expansion port, more complex connection required

8. Input voltage 6-15 V, wider voltage range, choose more



8. Input voltage 7-12 V, optional power supply mode is not

9. Choose the current more popular type-c interface, beautiful and generous, faster transmission speed



9. Adopt traditional square USB interface, more common

## 2.Specification

Microcontroller ATMEGA328P-AU

USB to serial chip: CP2102

Working voltage: 5V or 3.3V (DIP switch control)

External power: DC 6-15V (recommend 9V)

Digital I / O pins: 14 (D0-D13)

PWM channel: 6 (D3 D5 D6 D9 D10 D11)

Analog input channel (ADC): 8 (A0-A7)

Each I / O Port of DC output capability : 20 mA

Output capability of 3.3V port: 50 mA

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

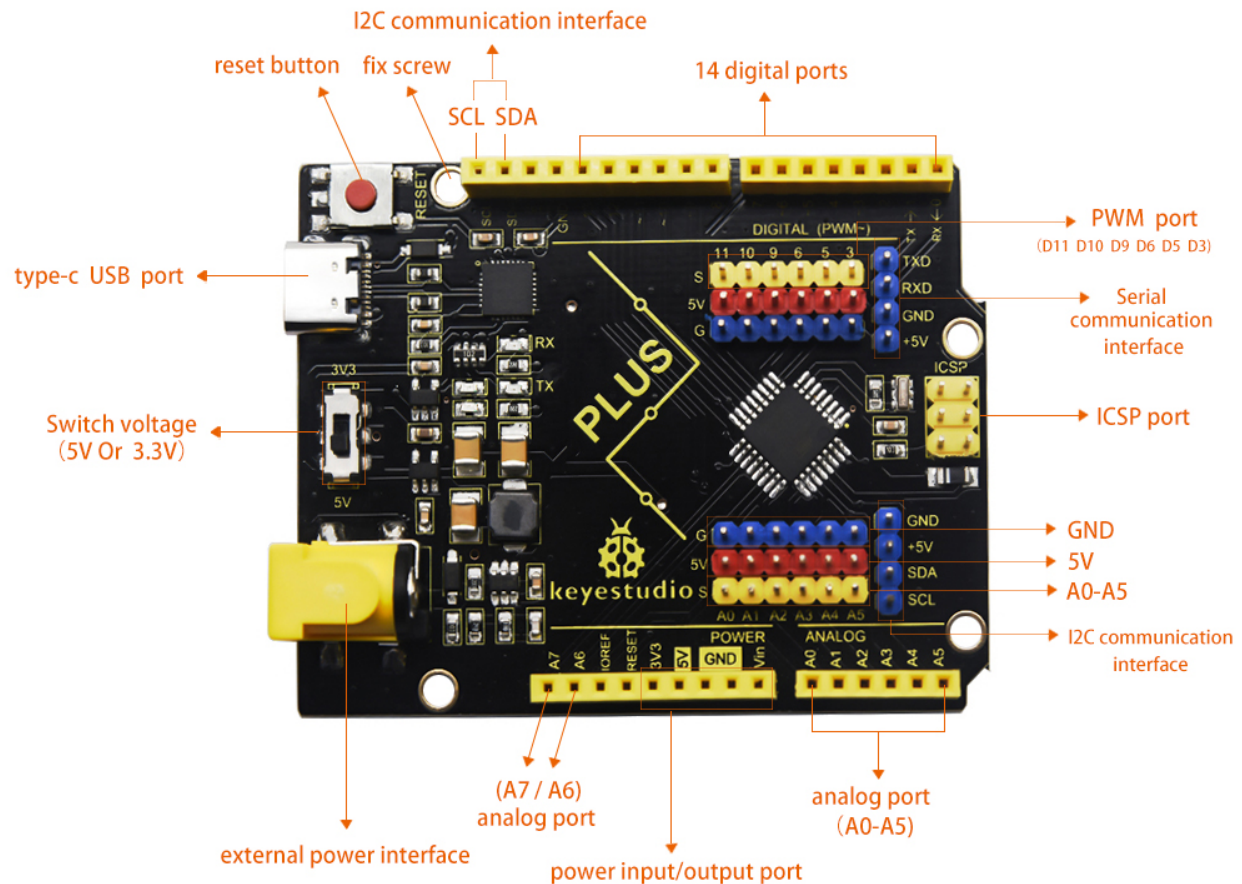
SRAM: 2 KB (ATMEGA328P-AU)

EEPROM: 1 KB (ATMEGA328P-AU)

Clock speed: 16MHz

On-board LED pin: D13

### **3.Interfaces**



#### 4.Specialized Functions of Some Pins:

Serial communication interface: D0 is RX, D1 is TX

PWM interface (pulse width modulation): D3 D5 D6 D9 D10 D11

External interrupt interface: D2 (interrupt 0) and D3 (interrupt 1)

SPI communication interface: D10 is SS, D11 is MOSI, D12 is MISO, D13 is SCK

IIC communication port: A4 is SDA, A5 is SCL

#### 5. Install Arduino IDE and Driver

##### 5.1 Download the Arduino IDE

When getting this control board, we need to install Arduino IDE.

Enter the website <https://www.arduino.cc> click



Download the version you want and the latest one is available for downloading too.

## Download the Arduino IDE



Alternatively, you could select previous release.

In this project, we use 1.8.12 version

### Previous Releases

Download the [previous version of the current release](#) the classic [Arduino 1.0.x](#), or the [Arduino 1.5.x Beta version](#).

All the [Arduino 00xx versions](#) are also available for download. The Arduino IDE can be used on Windows, Linux (both 32 and 64 bits), and Mac OS X.

Click [previous version of the current release](#) to view the below page

## ARDUINO 1.8.12

Arduino IDE that can be used with any Arduino board, including the Arduino Yún and Arduino DUE. Refer to the [Getting Started](#) page for Installation instructions.

[See the release notes.](#)

[Windows Installer](#)

[Windows ZIP file for non admin install](#)

[Mac OS X](#) 10.8 Mountain Lion or newer

[Linux](#) 32 bits

[Linux](#) 64 bits

[Linux](#) ARM 32

[Linux](#) ARM 64

[Source](#)

Click [Windows Installer](#) to download an installer of Arduino 1.8.12 version which needs to be installed manually. When you tap [Windows ZIP file for non admin install](#) a zip file of Arduino 1.8.12 version will be directly downloaded, and you only need to unzip it to finish the installation.

## Contribute to the Arduino Software

Consider supporting the Arduino Software by contributing to its development. (US tax payers, please note this contribution is not tax deductible). [Learn more on how your contribution will be used.](#)



SINCE MARCH 2015, THE ARDUINO IDE HAS BEEN DOWNLOADED **45,290,706** TIMES. (IMPRESSIVE!) NO LONGER JUST FOR ARDUINO AND GENUINO BOARDS, HUNDREDS OF COMPANIES AROUND THE WORLD ARE USING THE IDE TO PROGRAM THEIR DEVICES, INCLUDING COMPATIBLES, CLONES, AND EVEN COUNTERFEITS. HELP ACCELERATE ITS DEVELOPMENT WITH A SMALL CONTRIBUTION! REMEMBER: OPEN SOURCE IS LOVE!

**\$3** **\$5** **\$10** **\$25** **\$50** **OTHER**

JUST DOWNLOAD

CONTRIBUTE & DOWNLOAD


 JUST DOWNLOAD

CONTRIBUTE &amp; DOWNLOAD

Click icon

to download Arduino IDE.

## 5.2 Installing Driver

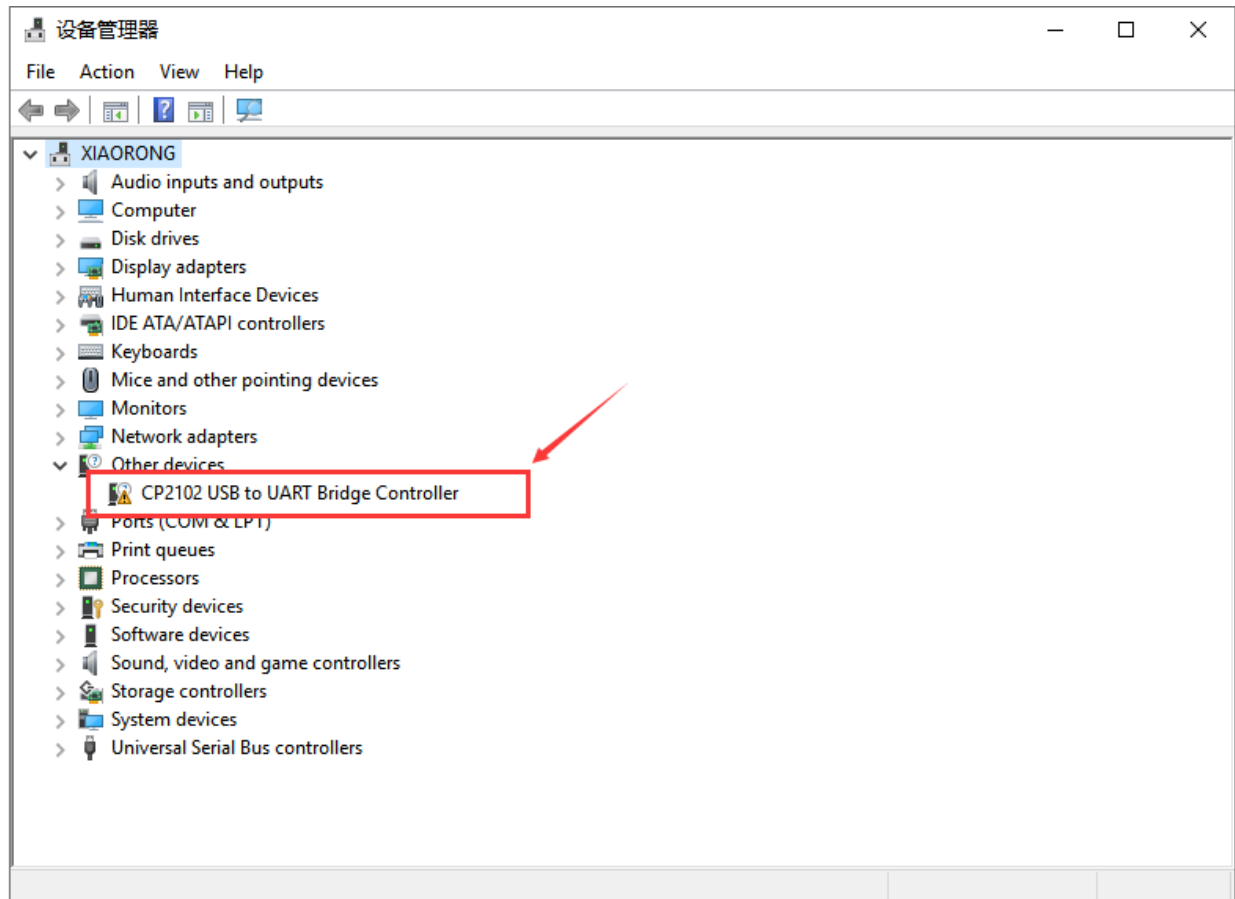
Let's install the driver of keystudio PLUS control board . The USB-TTL chip on PLUS board adopts CP2102 serial chip. The driver program of CP2102 is included in Arduino 1.8 version and above, which is convenient to operate. Plugged in USB, the computer can recognize the hardware and automatically install the driver of CP2102.

**Note: If the version of Arduino IDE you download is below 1.8, you need to download the driver of CP2102.**

**Download the driver of CP2102:**

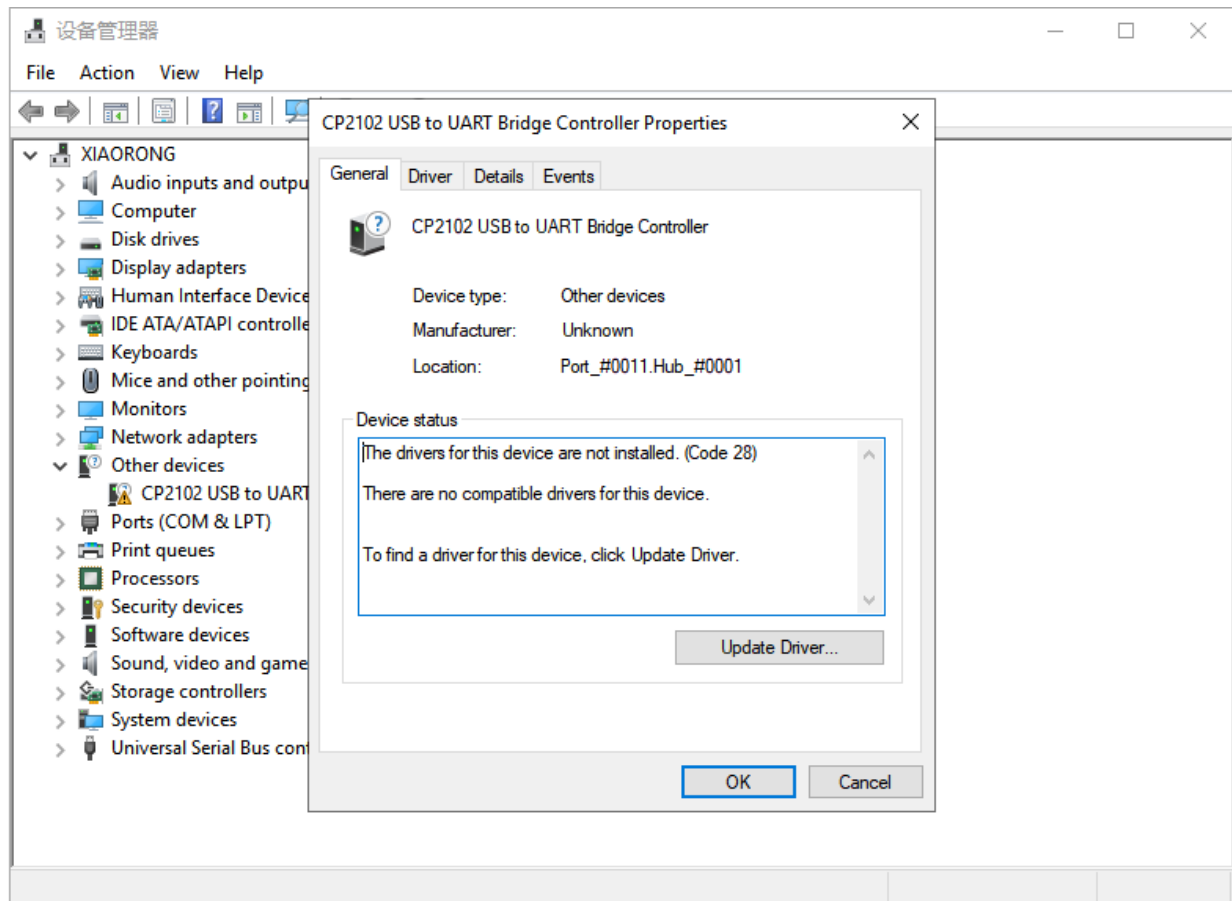
<https://fs.keyestudio.com/CP2102-WIN>

If install unsuccessfully, or you intend to install manually, open the device manager of computer. Right click Computer— Properties— Device Manager.



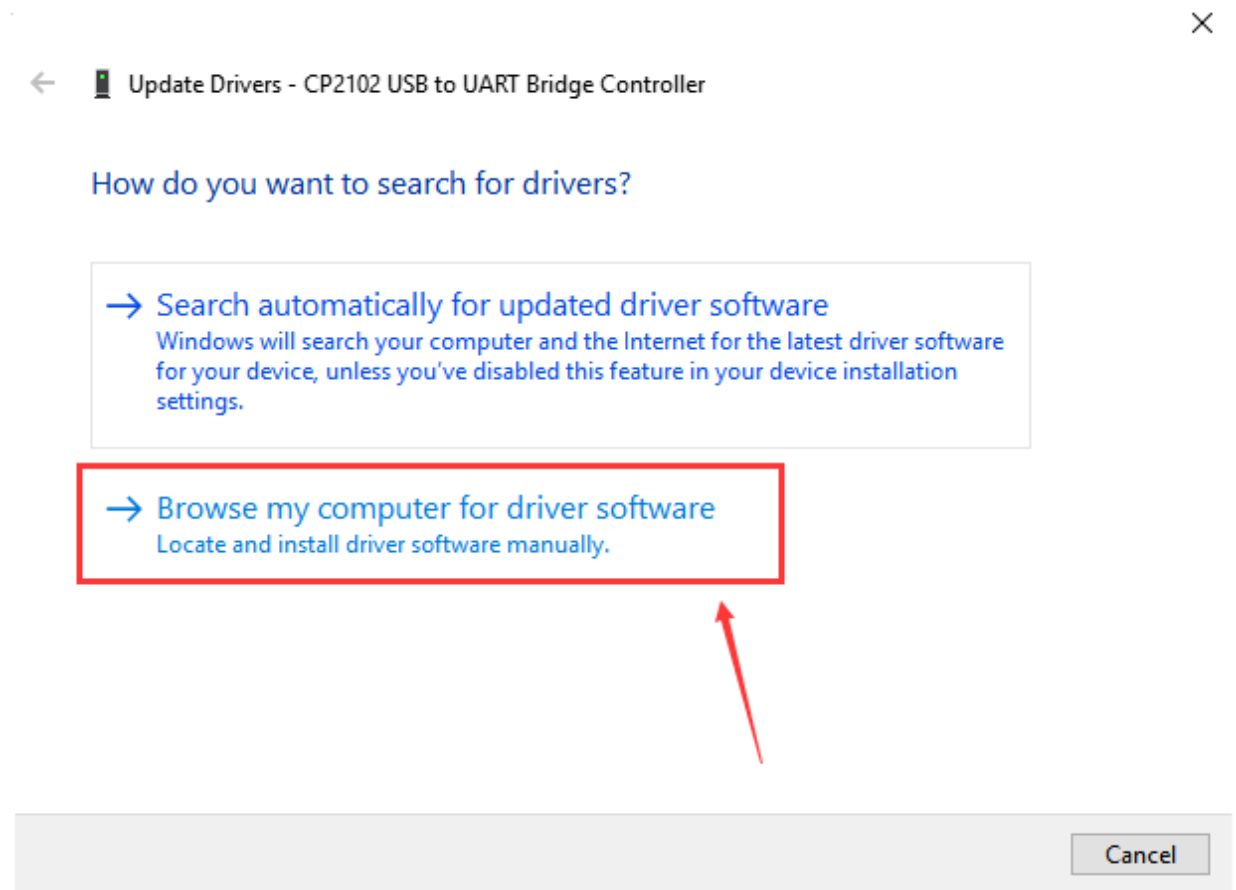
There is a yellow exclamation mark on the page, which implies an unsuccessful installation of the driver of CP2102 . Then please double click the hardware and update the driver.





Click “OK” to enter the following page, click “browse my computer for updated driver software”, and find the installed or downloaded ARDUINO software. As shown below:



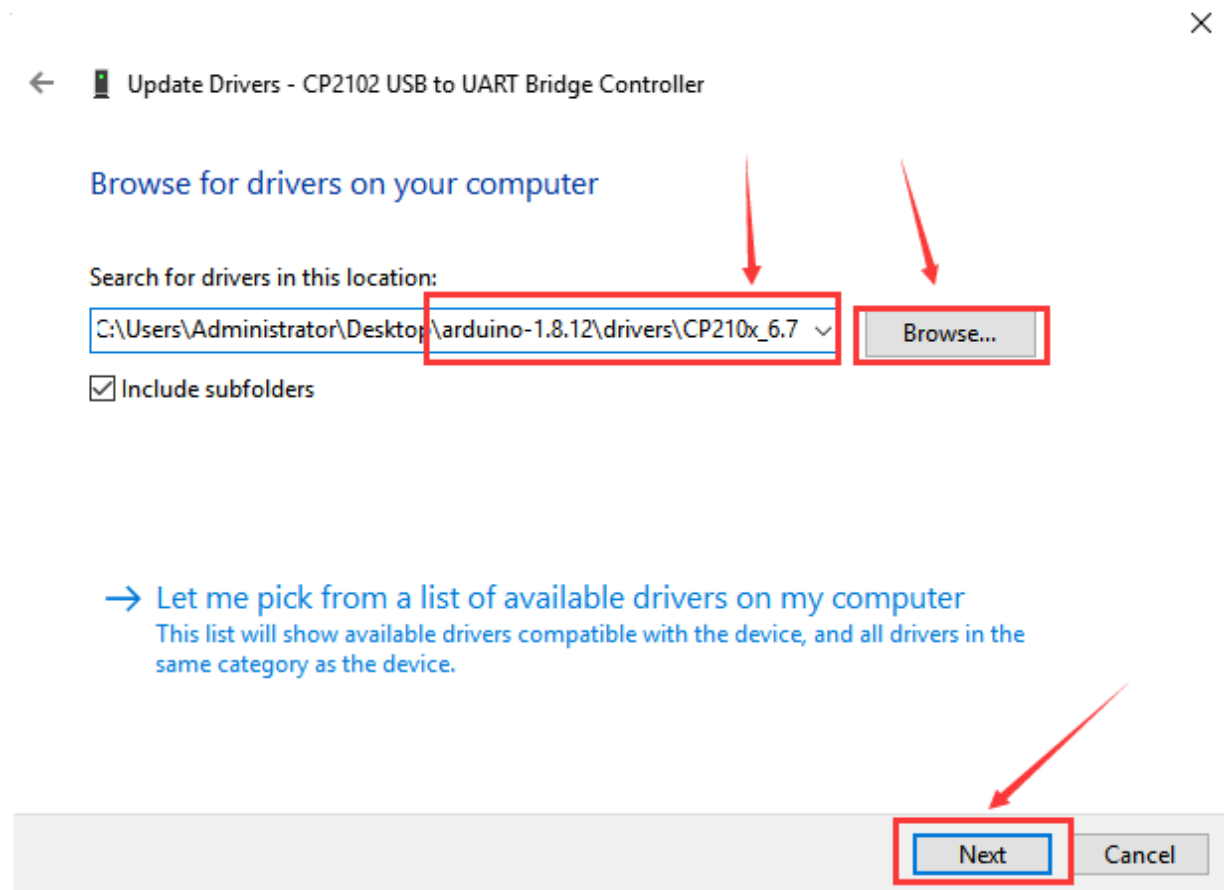


There is a DRIVERS folder in Arduino software installed package

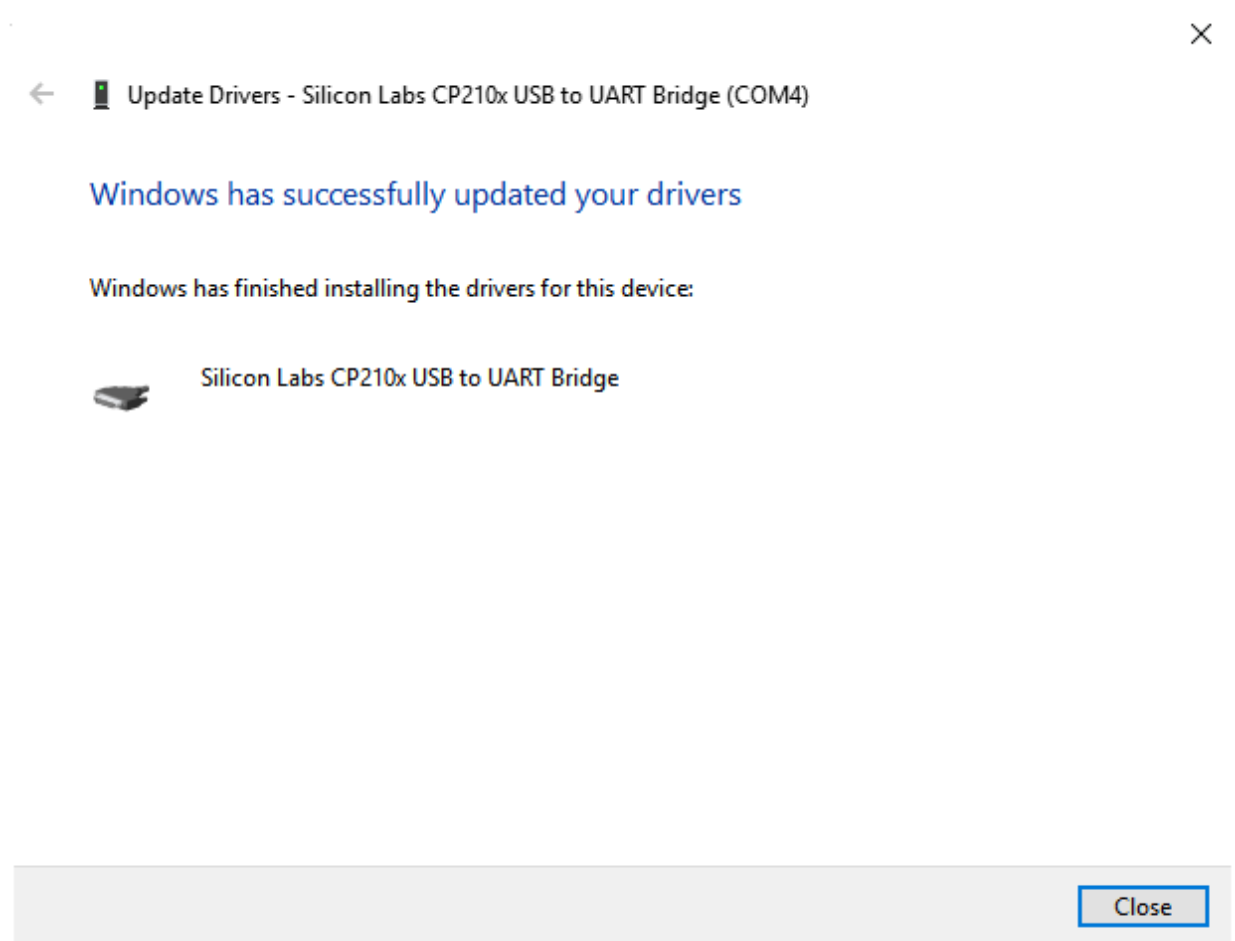


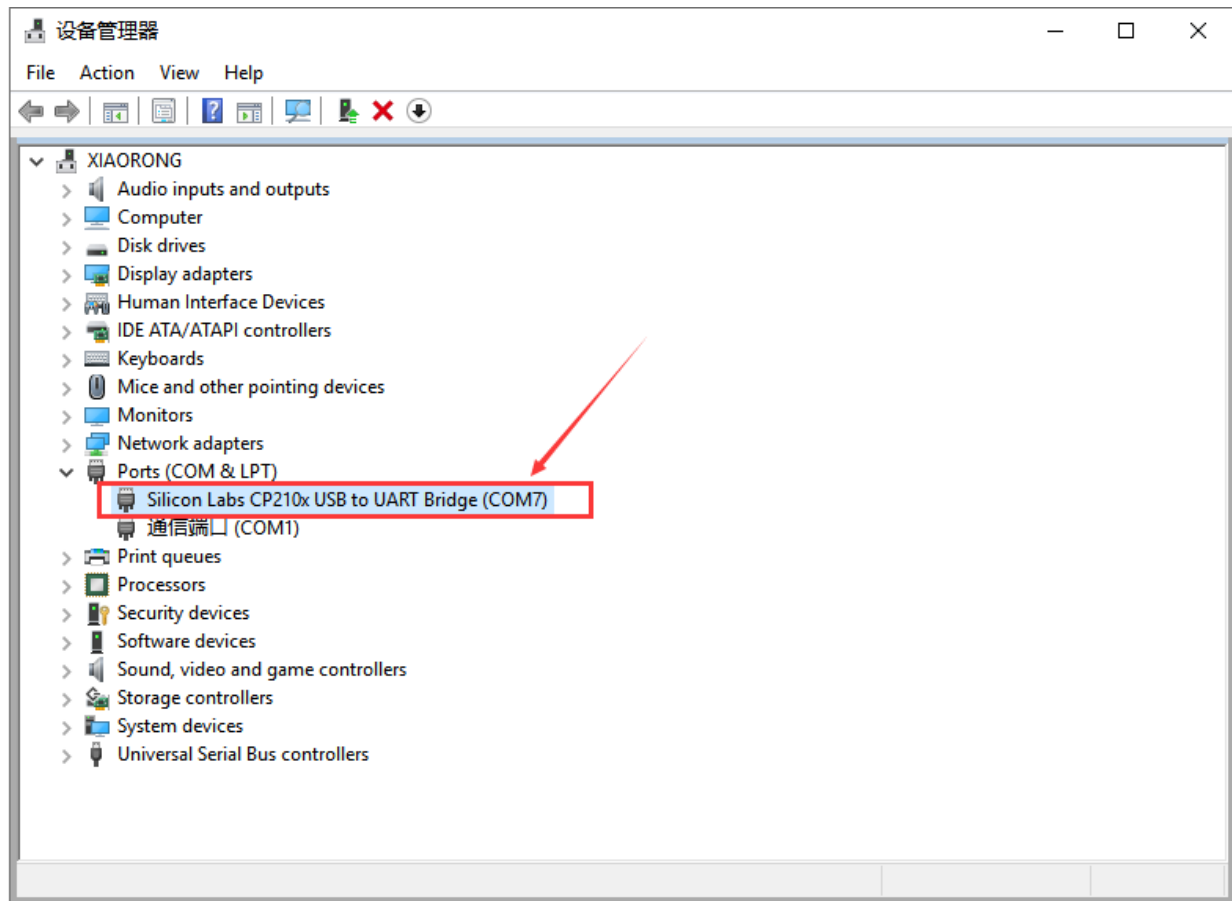
arduino-1.8.12

, open driver folder and

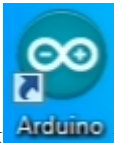


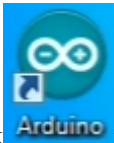
Open device manager, we will find the yellow exclamation mark disappear. The driver of CP2102 is installed successfully.

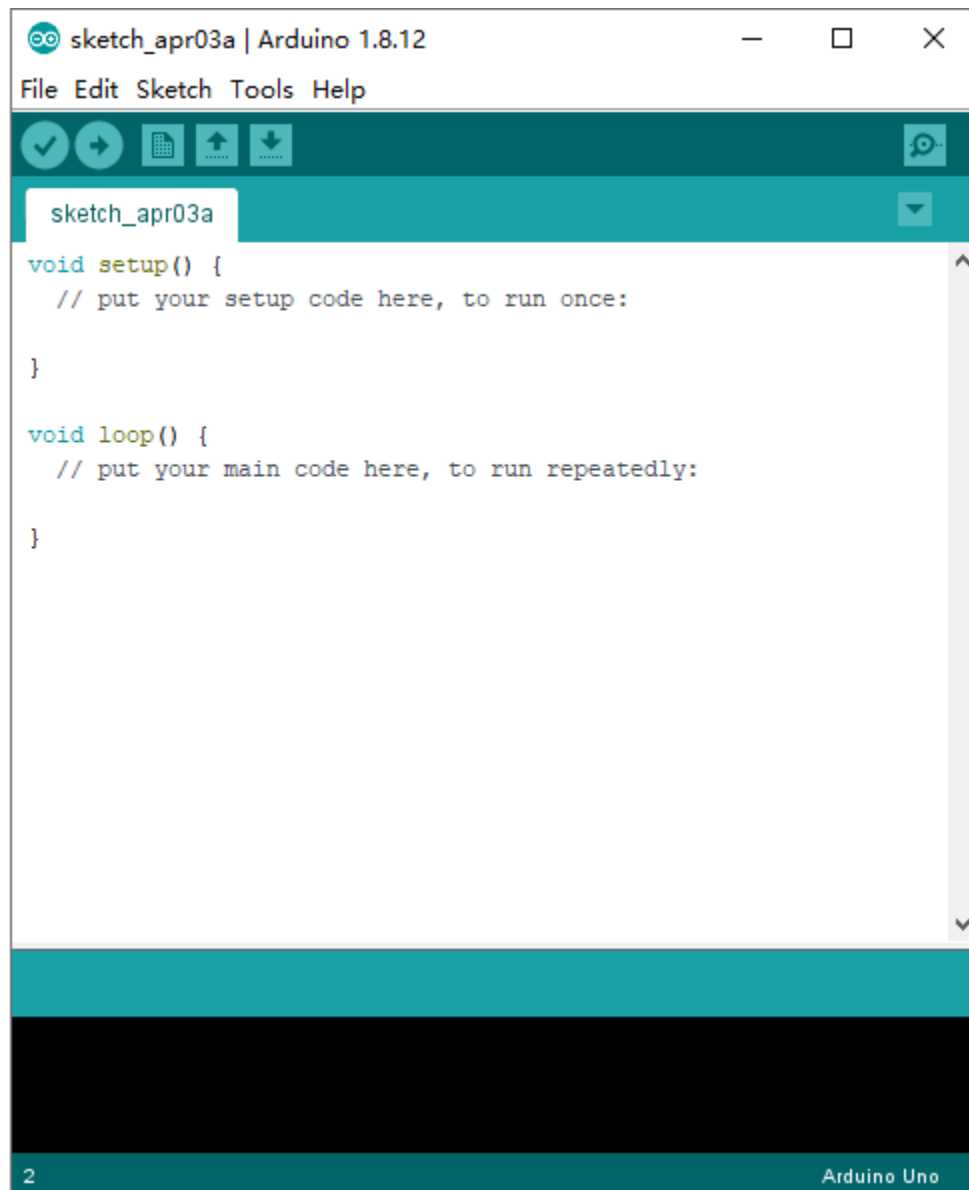




### 5.3 Arduino IDE Setting

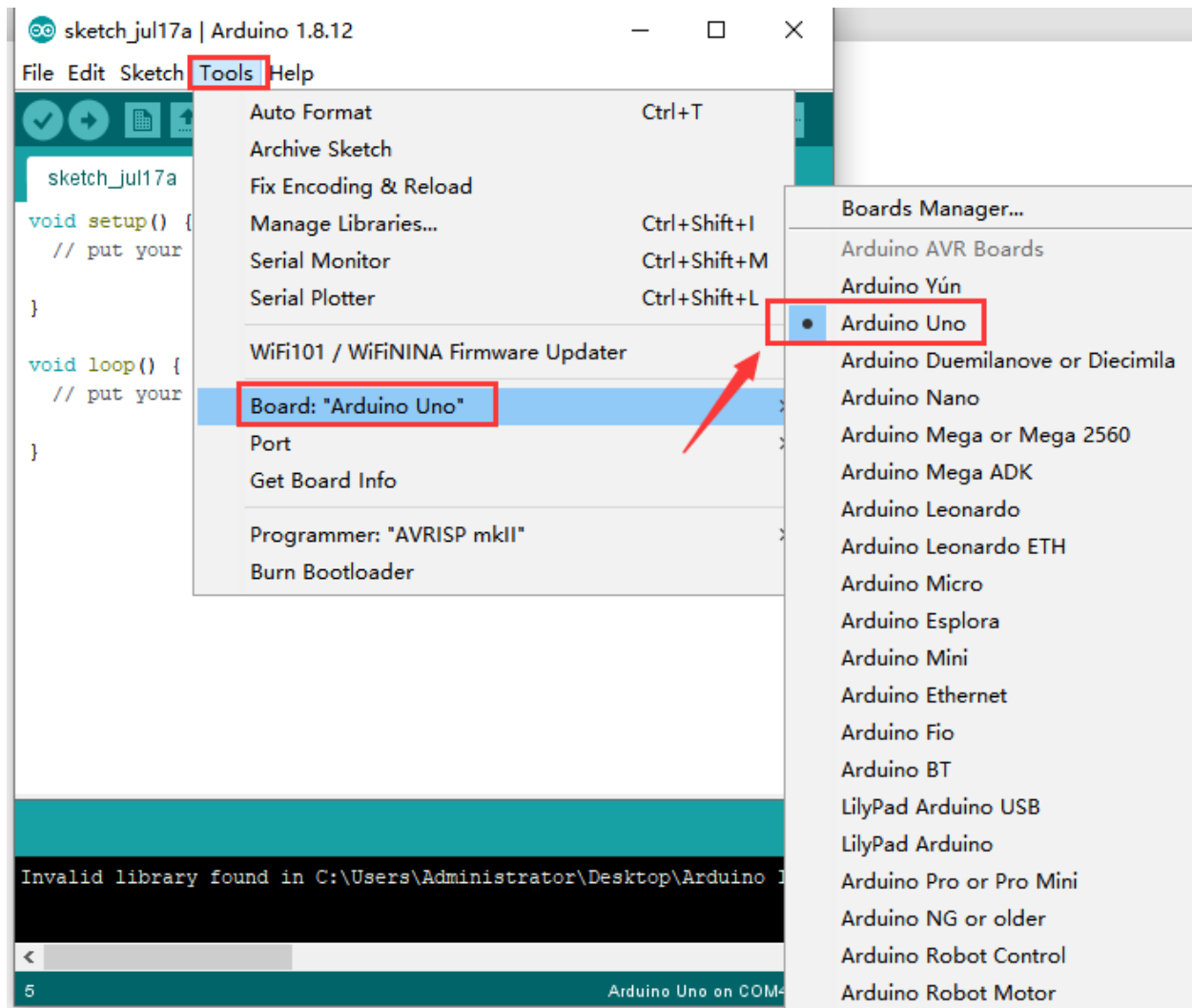


Click  icon to open Arduino IDE.

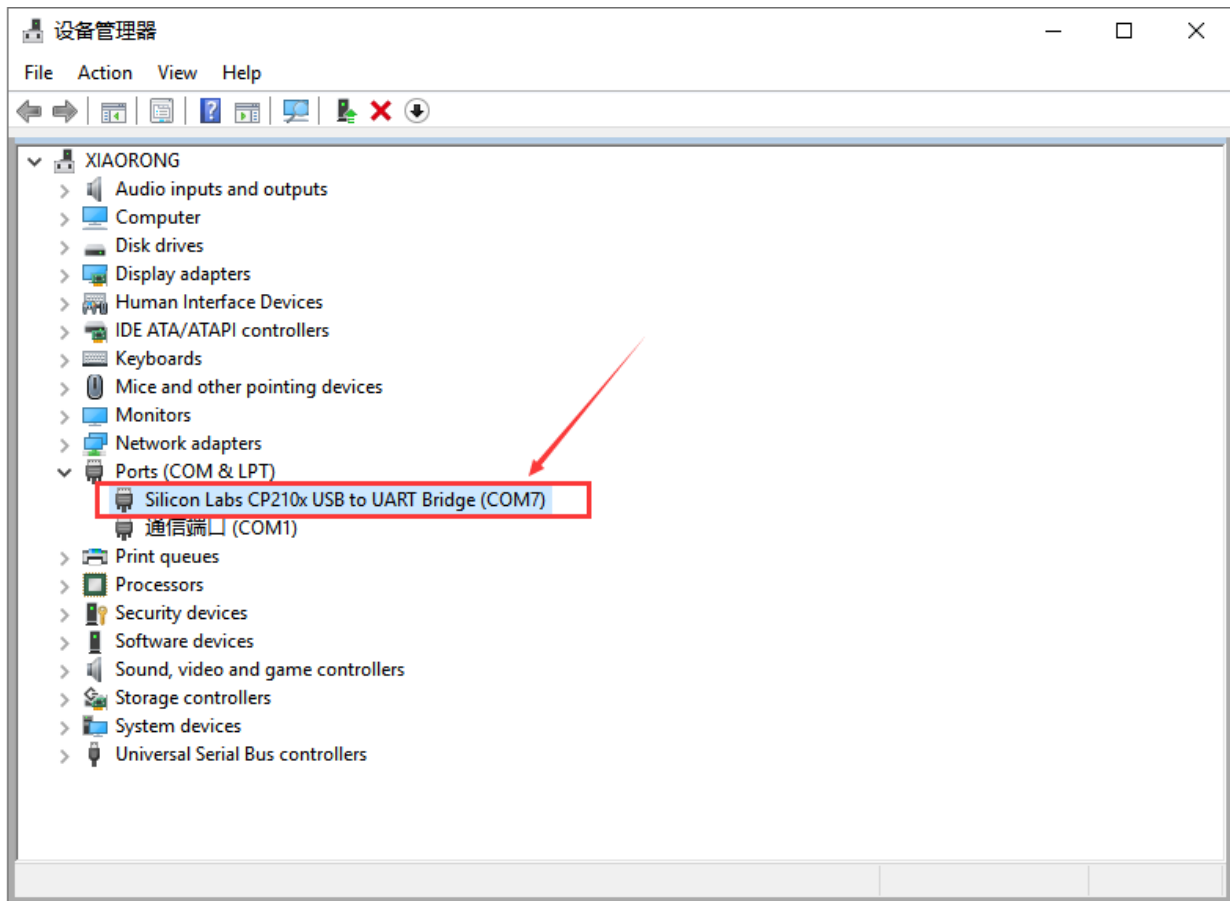


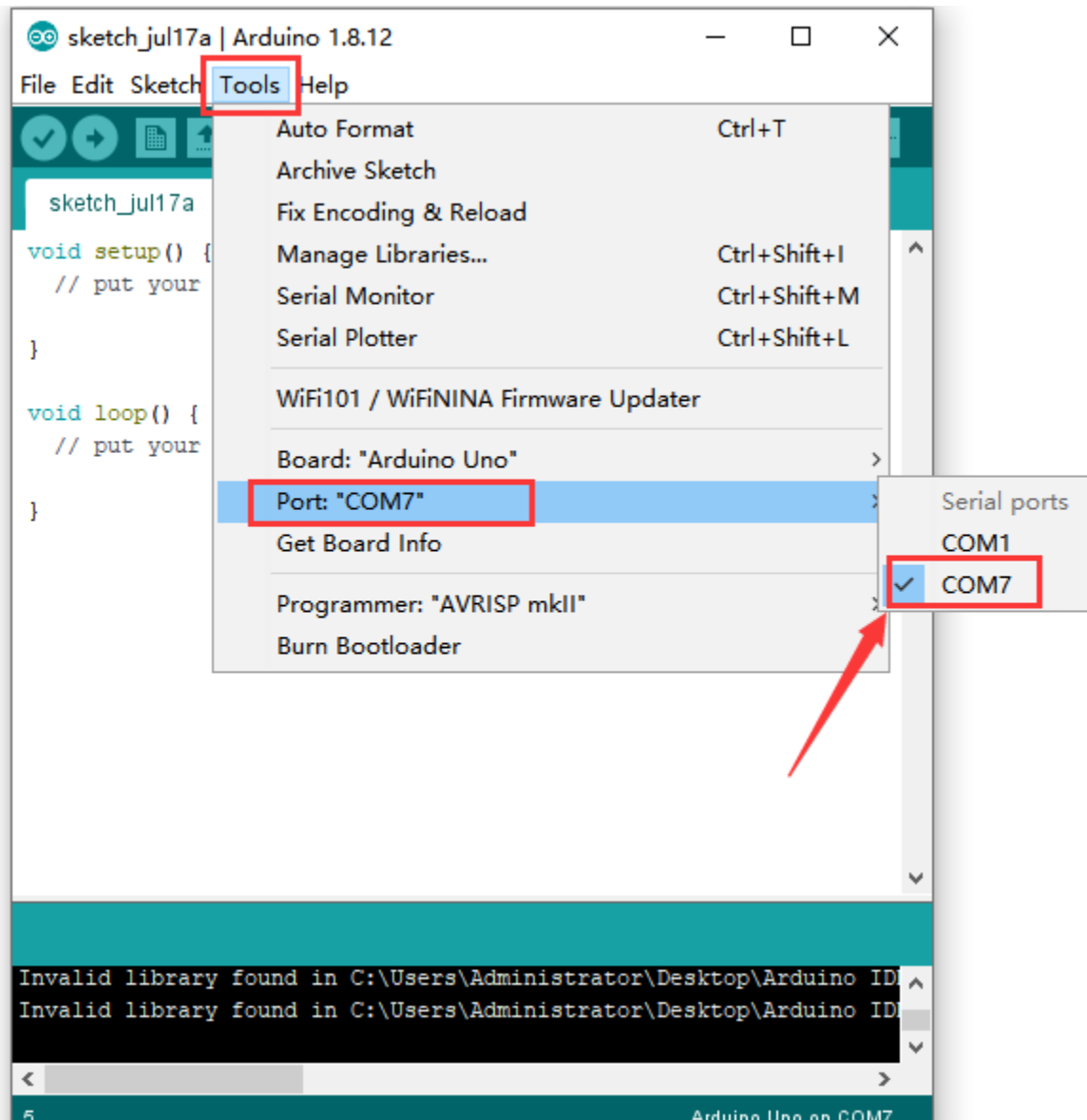
To avoid the errors when uploading the program to the board, you need to select the correct Arduino board that matches the board connected to your computer.

Then come back to the Arduino software, you should click Tools→Board, to select the board. (as shown below)



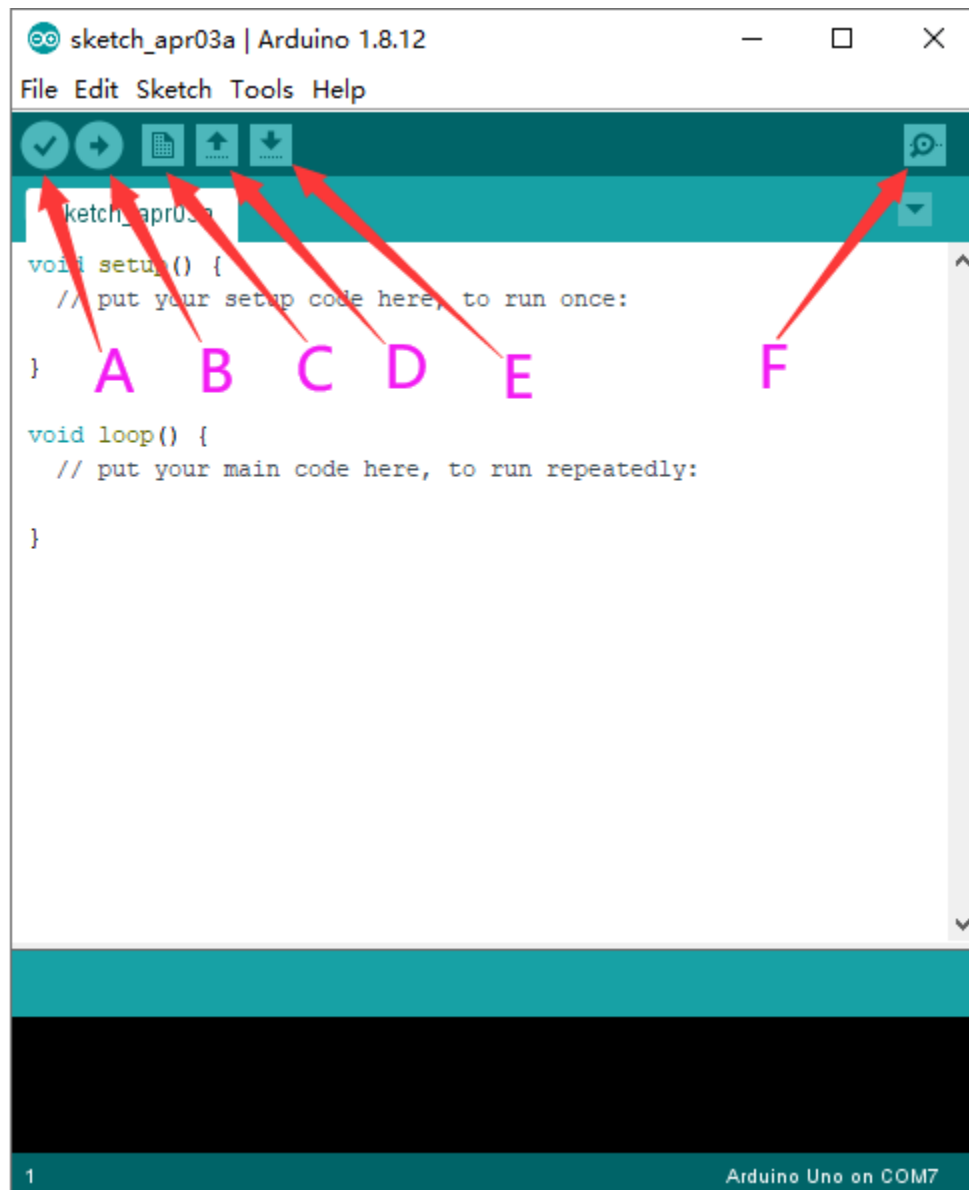
Then select the correct COM port (you can see the corresponding COM port after the driver is successfully installed)





Before uploading the program to the board, let's demonstrate the function of each symbol in the Arduino IDE toolbar.





A- Used to verify whether there is any compiling mistakes or not.

B- Used to upload the sketch to your Arduino board.

C- Used to create shortcut window of a new sketch.

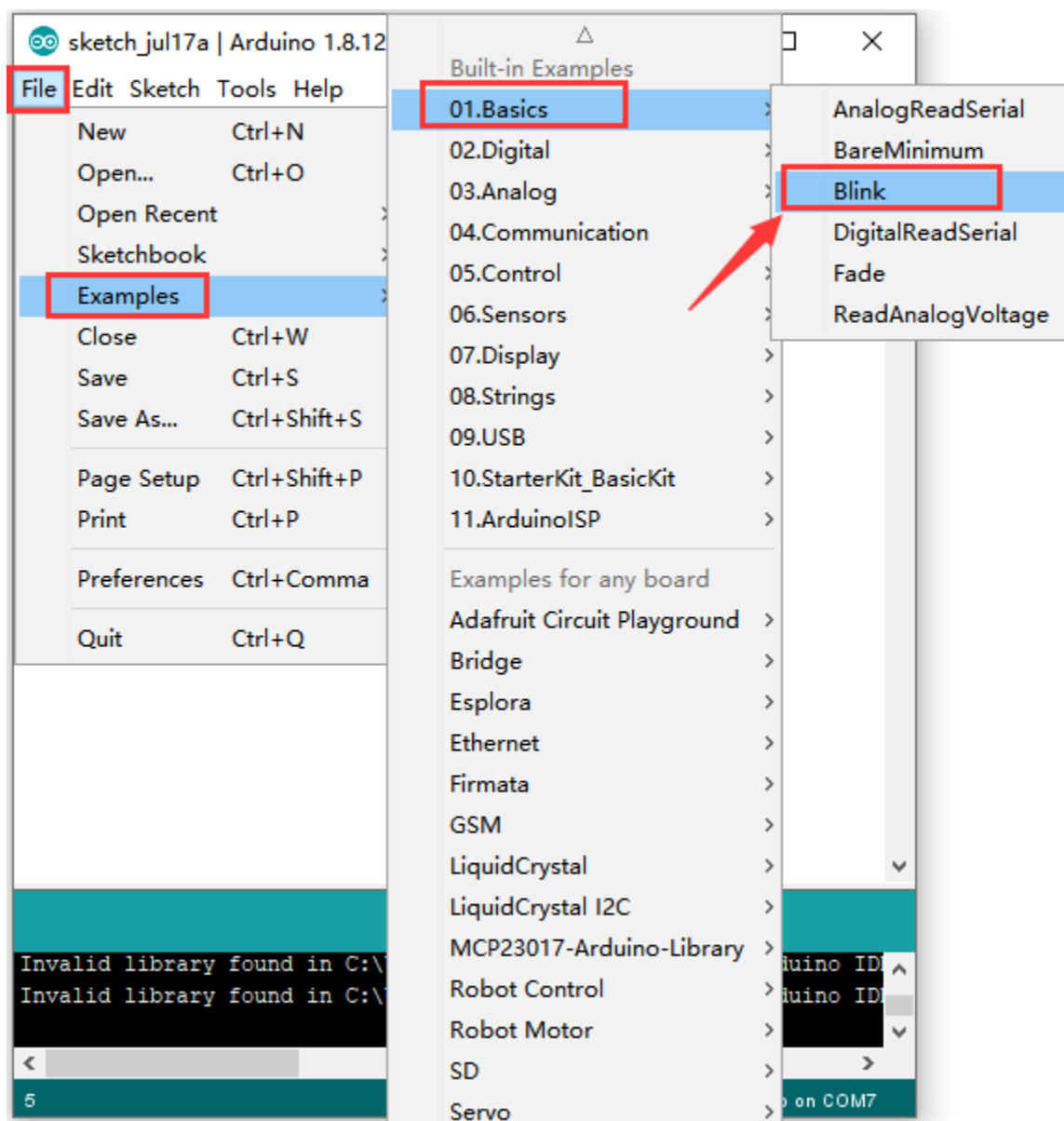
D- Used to directly open an example sketch.

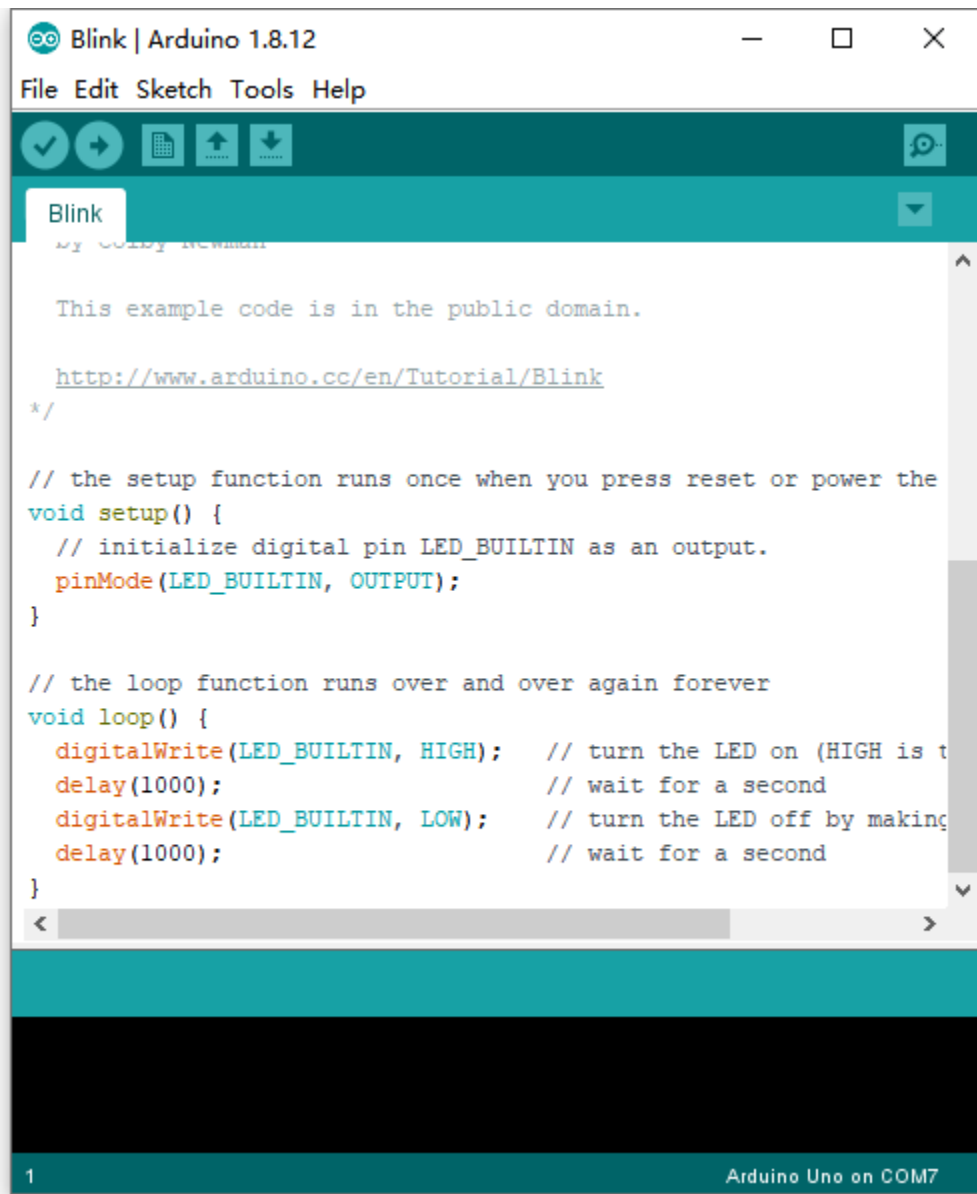
E- Used to save the sketch.

F- Used to send the serial data received from board to the serial monitor.

#### 5.4 Start First Program


Open the file to select Example, and choose BLINK from BASIC, as shown below:

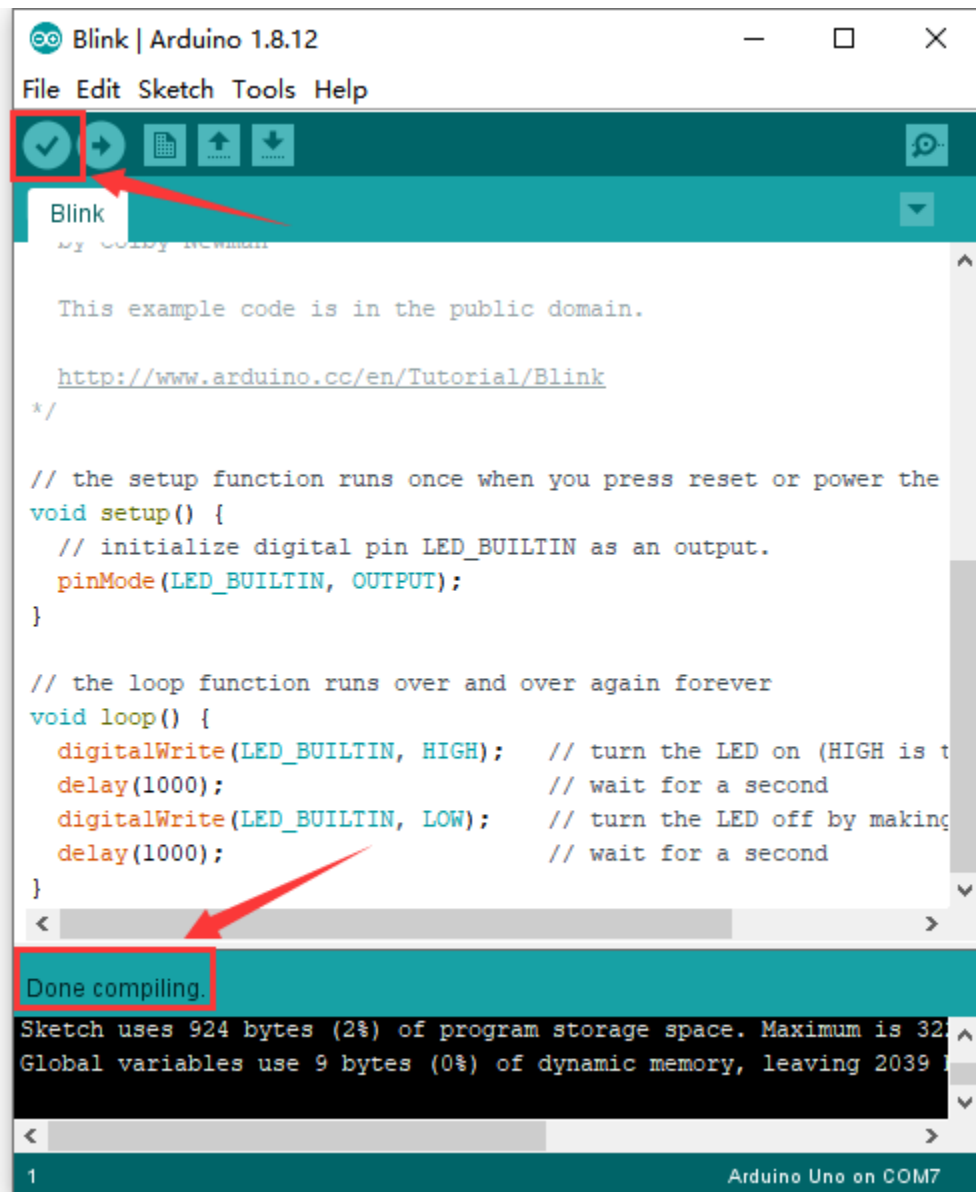





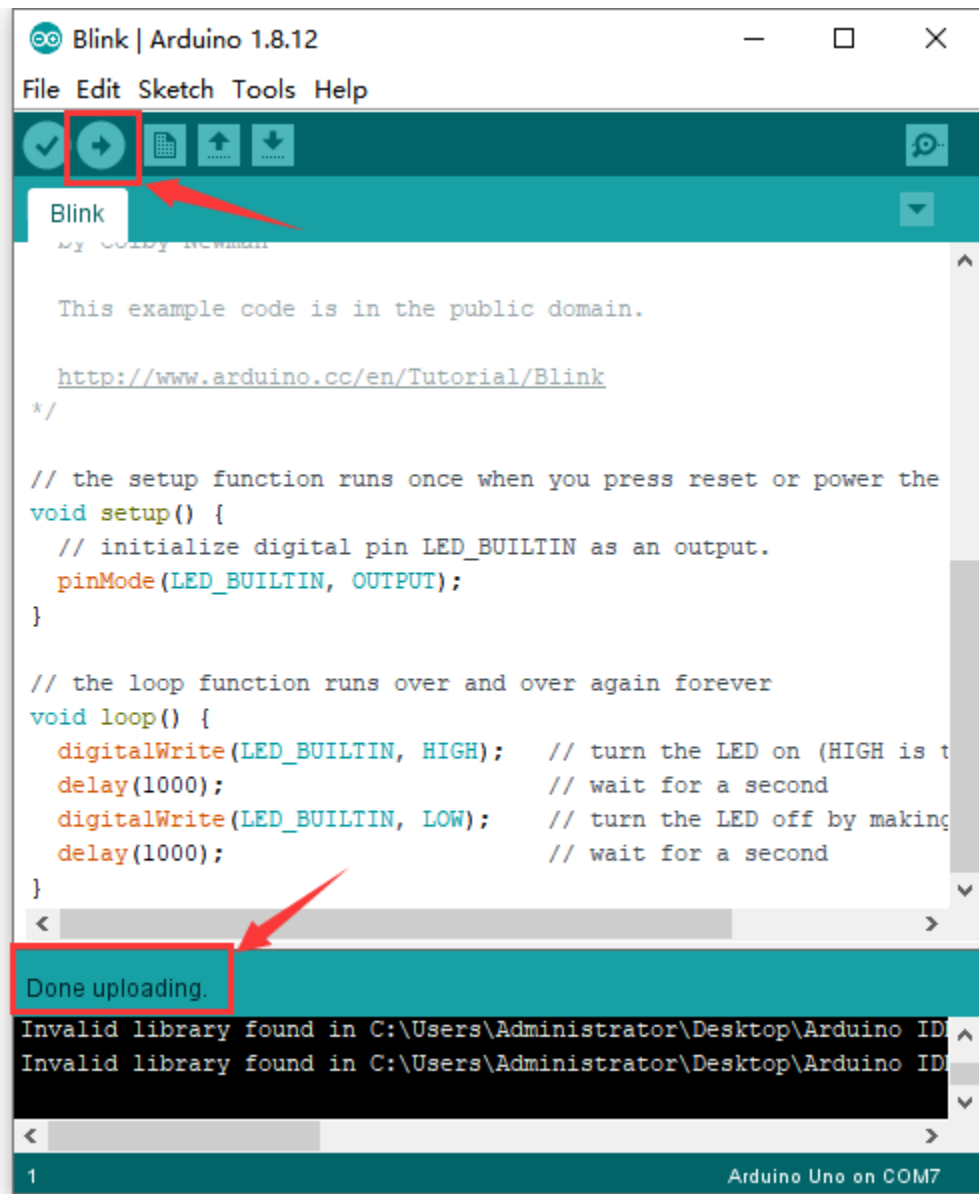
Set board and COM port, the corresponding board and COM port are shown on the lower right of IDE.



Click  to start compiling the program, check errors.



Click  to upload the program, upload successfully.



Upload the program successfully, the onboard LED lights on for 1s, lights off for 1s. Congratulation, you finish the first program.

## 6. MAC System

### 6.1 Install Arduino IDE on MAC System

The installation instruction is as same as the chapter 5.1, as shown below:

## ARDUINO 1.8.12

Arduino IDE that can be used with any Arduino board, including the Arduino Yún and Arduino DUE. Refer to the [Getting Started](#) page for Installation instructions.

[See the release notes.](#)

**Windows** Installer

**Windows** ZIP file for non admin install

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

**Linux** 32 bits

**Linux** 64 bits

**Linux** ARM 32

**Linux** ARM 64

**Source**

6.2 Download the Driver of CP2102

<https://fs.keyestudio.com/CP2102-MAC>

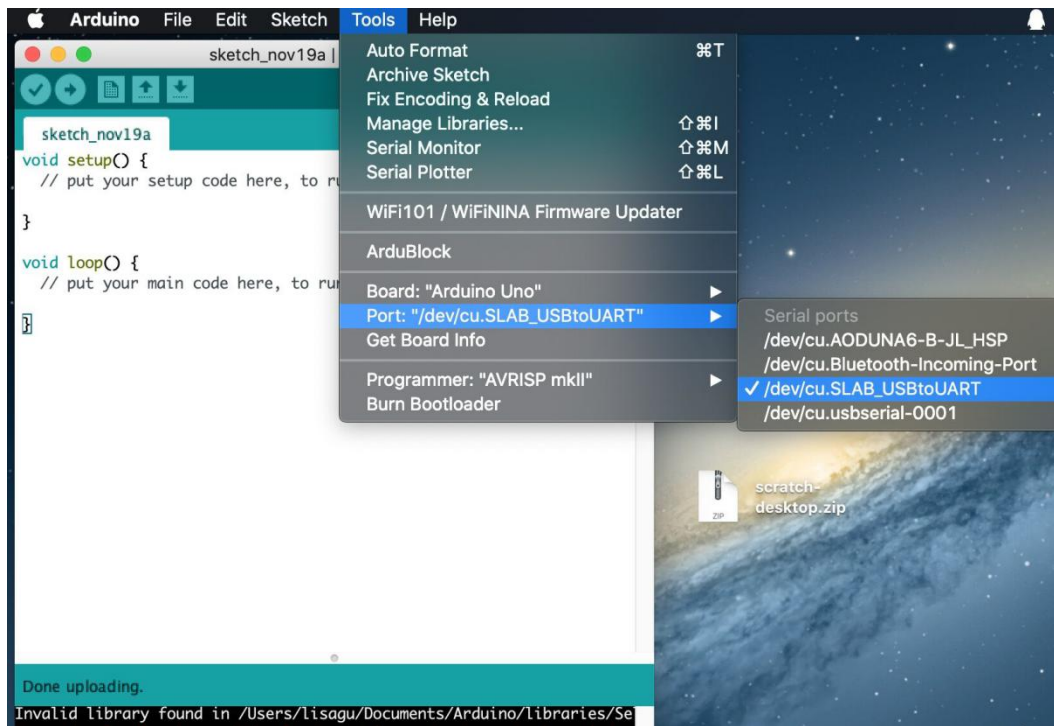
6.3 How to Install the Driver of CP2102

The following link is for your reference:

[https://wiki.keyestudio.com/How\\_to\\_Install\\_the\\_Driver\\_of\\_CP2102\\_on\\_MAC\\_System](https://wiki.keyestudio.com/How_to_Install_the_Driver_of_CP2102_on_MAC_System)

6.4 Arduino IDE Setting

The setting method is as same as that in the chapter 5.3 except from COM port, as shown below.

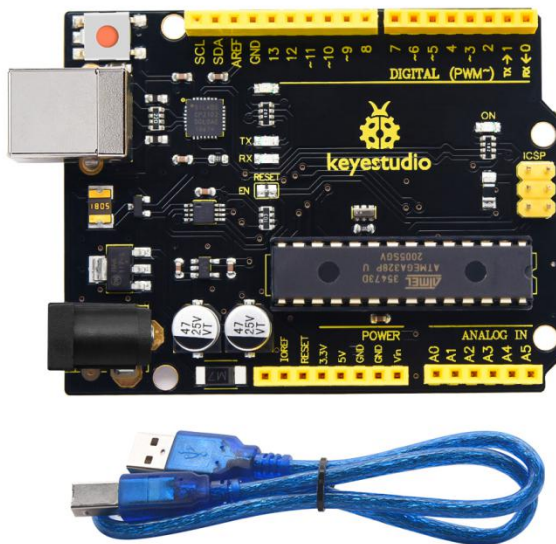


Resource:

<https://fs.keyestudio.com/KS0486>



## KS0497 KEYESTUDIO V4.0 DEVELOPMENT BOARD



### Description

The processor core of Keyestudio V4.0 development board is ATMEGA328P-PU , fully compatible with ARDUINO UNO REV3.

It has 14 digital input/output pins(of which 6 can be used as PWM outputs), 6 analog inputs, a 16 MHz crystal oscillator, a USB connection, a power jack, an ICSP header, and a reset button.

What's more, you can burn the firmware for ATMEGA328P-PU through the built-in ICSP port. The firmware of this chip is burnt well before delivery. Therefore, you don't need to burn the firmware.

The power can be supplied through USB wire, DC head and Vin GND pins.

### Specification

Microcontroller: ATMEGA328P-PU

USB to serial chipCP2102

Operating Voltage: 5V

Input Voltage (recommended):DC 7-12V

Digital I/O Pins: 14 (D0-13)

PWM Digital I/O Pins6 (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-PU) of which 0.5 KB used by bootloader

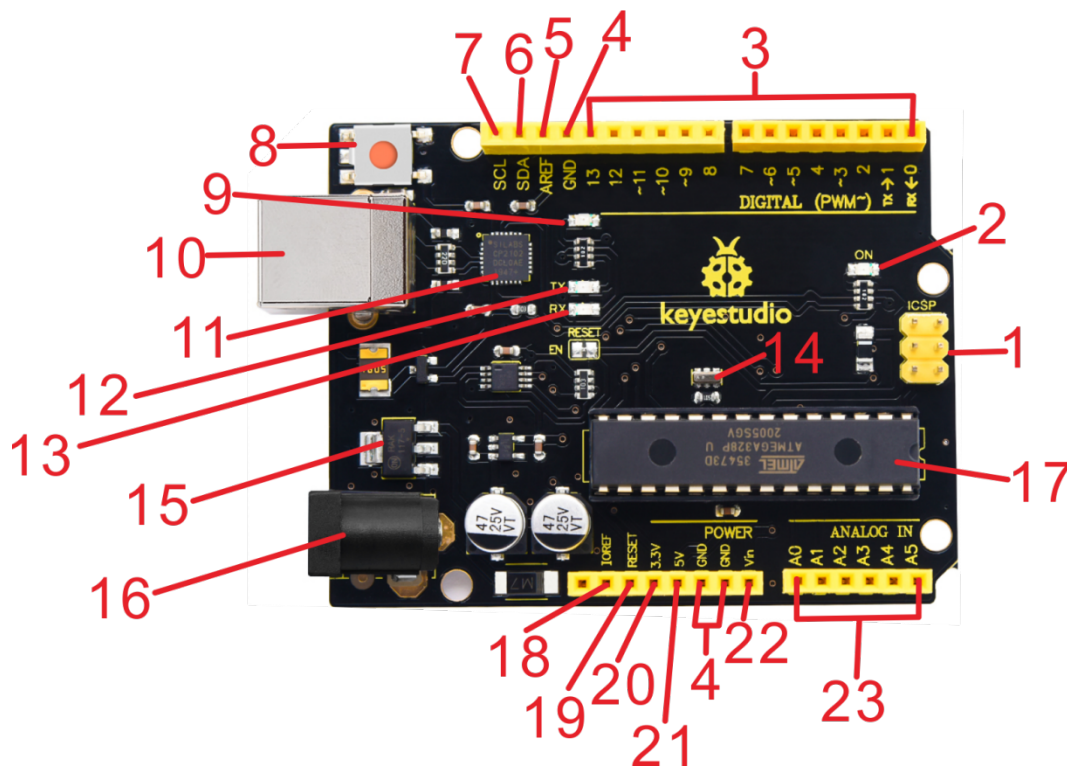
SRAM:2 KB (ATMEGA328P-PU)

EEPROM: 1 KB (ATMEGA328P-PU)

Clock Speed:16 MHz

LED\_BUILTIN:D13

### Pins and Parts Introduction



1	<b>ICSP (In-Circuit Serial Programming) Header</b>	the AVR, an Arduino micro-program header consisting of MOSI, MISO, SCK, RESET, VCC, and GND. It is often called the SPI (serial peripheral interface) and can be considered an “extension” of the output. In fact, slave the output devices to the SPI bus host. When connecting to PC, program the firmware to ATMEGA328P-PU.
2	<b>Power LED Indicator</b>	Powering the Arduino, LED on means that your circuit board is correctly powered on. If LED is off, connection is wrong.
3	<b>Digital I/O</b>	Arduino MEGA has 14 digital input/output pins (of which 6 can be used as PWM outputs). These pins can be configured as digital input pin to read the logic value (0 or 1). Or used as digital output pin to drive different modules like LED, relay, etc. Using <code>pinMode()</code> , <code>digitalWrite()</code> , and <code>digitalRead()</code> functions.
4	<b>GND</b>	GND
5	<b>AREF</b>	Reference voltage (0-5V) for analog inputs. Used with <code>analogReference()</code> . Configures the reference voltage used for analog input (i.e. the value used as the top of the input range).
6	<b>SDA</b>	IIC communication pin
7	<b>SCL</b>	IIC communication pin
8	<b>RESET Button</b>	You can reset your Arduino board,
9	<b>D13 LED</b>	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.
10	<b>USB Connection</b>	Arduino board can be powered via USB connector. All you needed to do is connecting the USB port to PC using a USB cable.
11	<b>CP2102</b>	USB serial chip, translate the USB signal of computer into serial signal
12	<b>TX LED</b>	Onboard you can find the label: TX (transmit) When Arduino board communicates via serial port, send the message, TX led flashes.
13	<b>RX LED</b>	Onboard you can find the label: RX (receive) When Arduino board communicates via serial port, receive the message, RX led flashes.
14	<b>Crystal Oscillator</b>	How does Arduino calculate time? by using a crystal oscillator. The number printed on the top of the Arduino crystal is 16.000H9H. It tells us that the frequency is 16,000,000 Hertz or 16MHz.
15	<b>Voltage Regulator</b>	To control the voltage provided to the Arduino board, as well as to stabilize the DC voltage used by the processor and other components. Convert an external input DC7-12V voltage into DC 5V, then switch DC 5V to the processor and other components.
16	<b>DC Power Jack</b>	Arduino board can be supplied with an external power DC7-12V from the DC power jack.
17	<b>Microcontroller</b>	Each Arduino board has its own microcontroller. You can regard it as the brain of your board. The main IC (integrated circuit) on the Arduino is slightly different from the panel pair. Microcontrollers are usually from ATMEL. Before you load a new program on the Arduino IDE, you must know what IC is on your board. This information can be checked at the top of IC.
18	<b>IOREF</b>	This pin on the 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 for working with the 5V or 3.3V.
19	<b>RESET Header</b>	Connect an external button to reset the board. The function is the same as reset button.
20	<b>Power Pin 3V3</b>	A 3.3 volt supply generated by the on-board regulator. Maximum current draw is 50 mA.
21	<b>Power Pin 5V</b>	Provides 5V output voltage
22	<b>Vin</b>	You can supply an external power input DC7-12V through this pin to Arduino board.
23	<b>Analog Pins</b>	Onboard has 6 analog inputs, labeled A0 to A5.

#### 4. Specialized Functions of Some Pins:

**Serial Communication:** D0 (RX) and D1 (TX)

**PWM Pins (Pulse-Width Modulation):** D3 D5 D6 D9 D10 D11

**External Interrupts:** D2 (interrupt 0), D3 (interrupt 1)

**SPI communication:** D10 (SS), D13 (SCK), D11 (MOSI), D12 (MISO).

These pins support SPI communication using the [SPI library](#). The SPI pins are also broken out on the ICSP header, which is physically compatible with the Arduino Uno.

**IIC communication:** A4 (SDA); A5 (SCL).

## 5. Windows System

### 5.1 Download the Arduino IDE

When getting this control board, we need to install Arduino IDE

Enter the website <https://www.arduino.cc/> and click **SOFTWARE** > **DOWNLOADS**



The screenshot shows the Arduino website's navigation bar with the 'SOFTWARE' menu highlighted. A dropdown menu is visible, showing 'ARDUINO CREATE' and 'DOWNLOADS'. The 'DOWNLOADS' option is highlighted with a red box and a red arrow. The main content area features a 'WHAT IS ARDUINO?' section with an image of an Arduino Uno, a 'BUY AN ARDUINO' button, a 'LEARN ARDUINO' button, and a 'DONATE' button. There is also a 'LATEST' section titled 'ARDUINO EDUVISION' with a discussion about the importance of coding, and a 'NEW BOARDS AND SHIELDS AND KITS, OH MY! CHECK THEM OUT!' section with a background of boxes.

You can select the latest version—1.8.13. Alternatively, the previous release is your another choice.

## Download the Arduino IDE



### ARDUINO 1.8.13

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 7 and up

**Windows** ZIP file for non admin install

**Windows app** Requires Win 8.1 or 10

**Get** 

**Mac OS X** 10.10 or newer

**Linux** 32 bits

**Linux** 64 bits

**Linux** ARM 32 bits

**Linux** ARM 64 bits

Release Notes

Source Code

Checksums (sha512)

In this project, we use 1.8.12 version

### Previous Releases

Download the [previous version of the current release](#) the classic [Arduino 1.0.x](#), or the [Arduino 1.5.x Beta version](#).

All the [Arduino 00xx versions](#) are also available for download. The Arduino IDE can be used on Windows, Linux (both 32 and 64 bits), and Mac OS X.

Click [previous version of the current release](#) to view the below page

## ARDUINO 1.8.12

Arduino IDE that can be used with any Arduino board, including the Arduino Yún and Arduino DUE. Refer to the [Getting Started](#) page for Installation instructions.

[See the release notes.](#)

**Windows** Installer

**Windows** ZIP file for non admin install

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

**Linux** 32 bits

**Linux** 64 bits

**Linux** ARM 32

**Linux** ARM 64

**Source**

The **Windows installer** needs installing manually. Yet , the **Windows zip file for non admin install** a zip file of Arduino 1.8.12 version, can be directly downloaded and installed.

## Contribute to the Arduino Software

Consider supporting the Arduino Software by contributing to its development. (US tax payers, please note this contribution is not tax deductible). [Learn more on how your contribution will be used.](#)



JUST DOWNLOAD

CONTRIBUTE & DOWNLOAD

JUST DOWNLOAD

CONTRIBUTE & DOWNLOAD

Click icon

to download Arduino IDE.

### 5.2 Installing Driver

Let's install the driver of Keystudio V4.0 development board. The USB-TTL chip on V4.0 development board adopts CP2102 serial chip.

Its driver files are included in the Arduino 1.8 version and above. Therefore, when the board is attached to the computer, the computer can recognize the driver of CP2102 automatically.

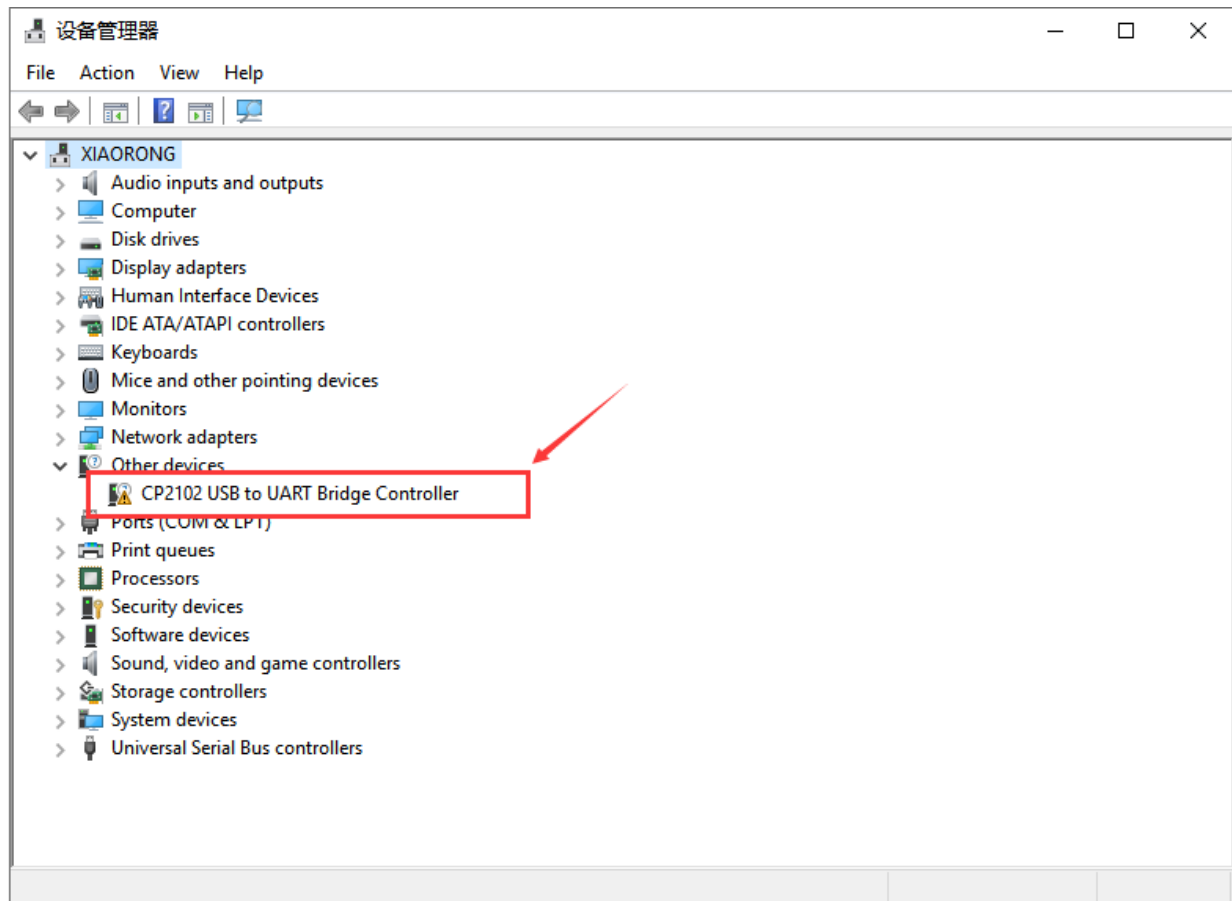
**Note: If the version of Arduino IDE you download is below 1.8, you need to download the driver of CP2102.**

**Download the driver of CP2102:**

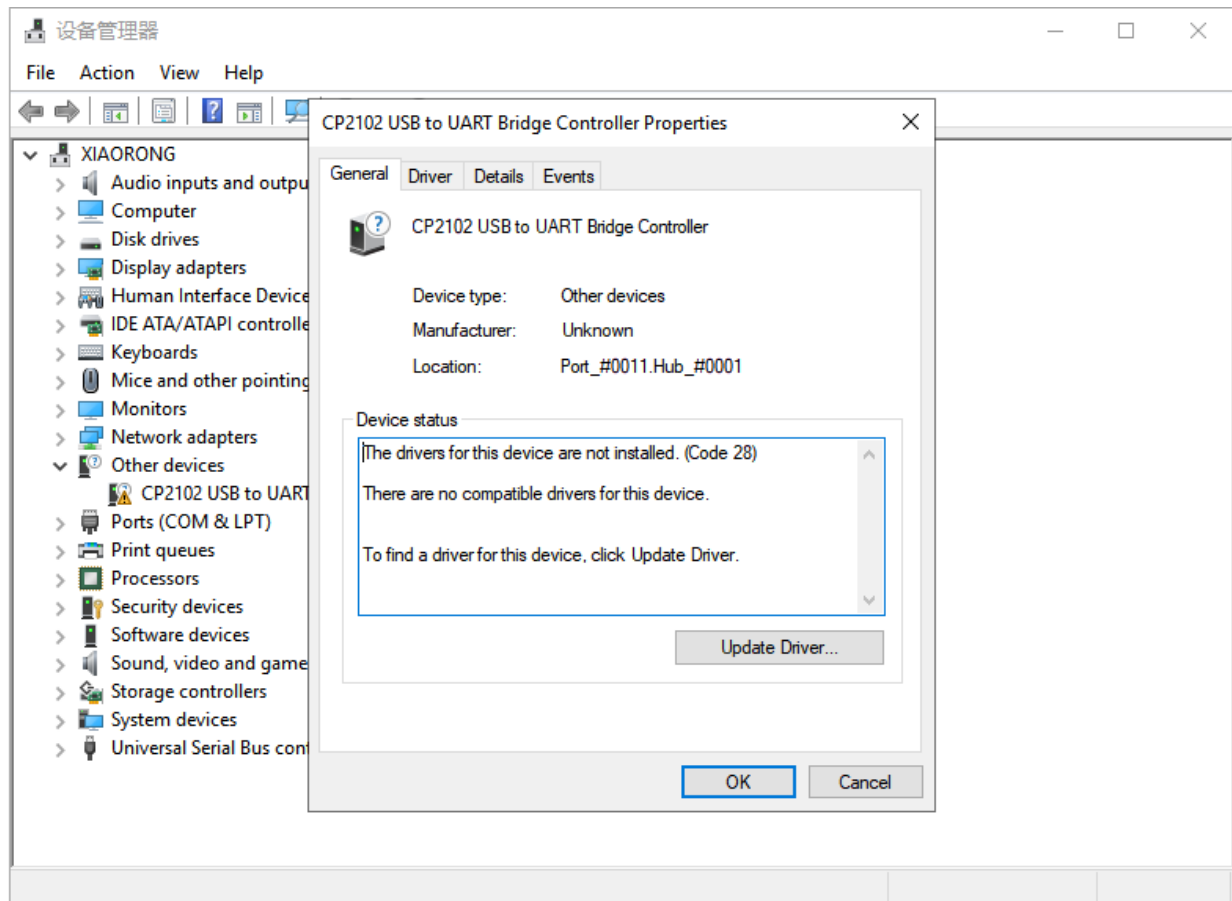
<https://fs.keyestudio.com/CP2102-WIN>

If it is installed unsuccessfully, you need to install it manually.

Click Computer— Properties— Device Manager, as shown below:

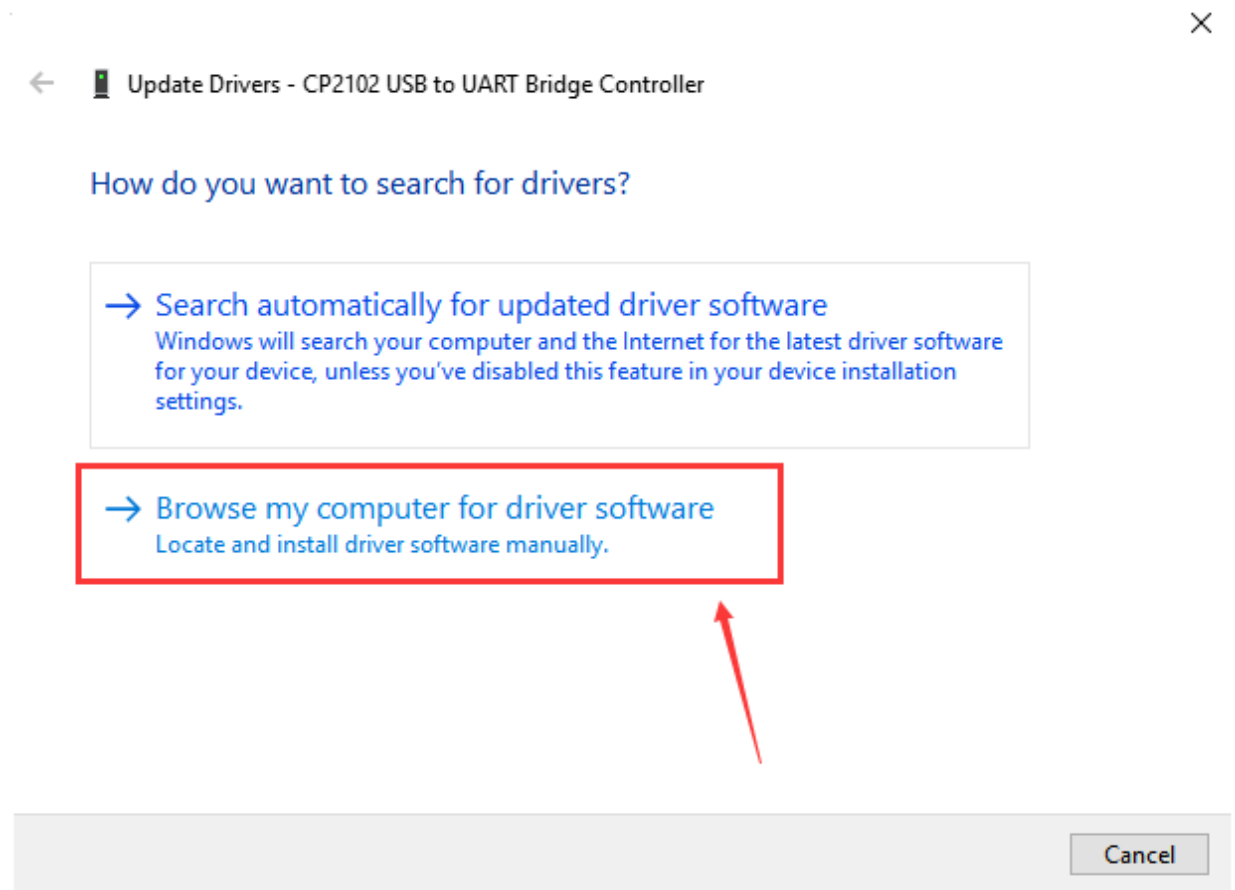


There is a yellow exclamation mark on the page, which implies the installation of the driver of CP2102 has failed.




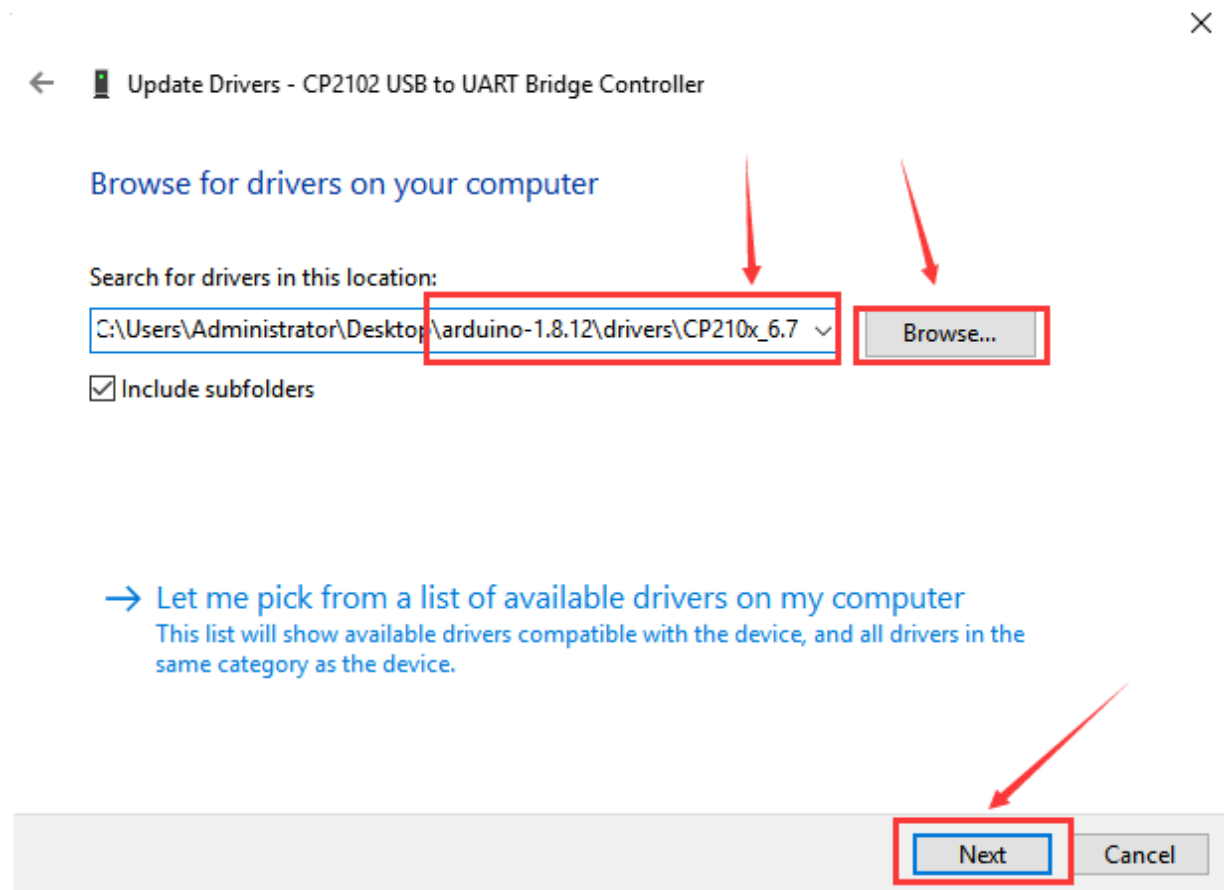
Click “OK” to enter the following page and click “browse my computer for updated driver software”.



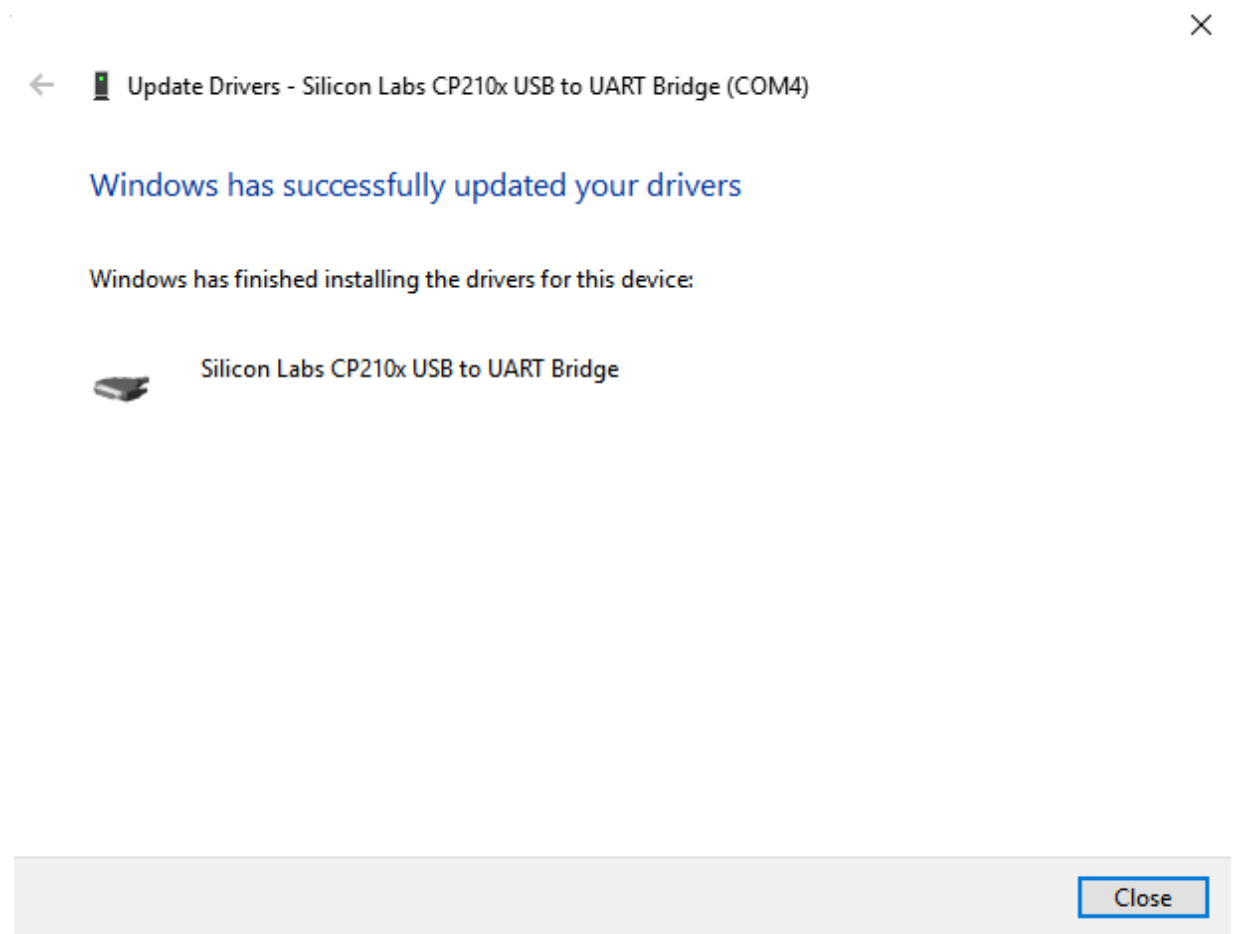


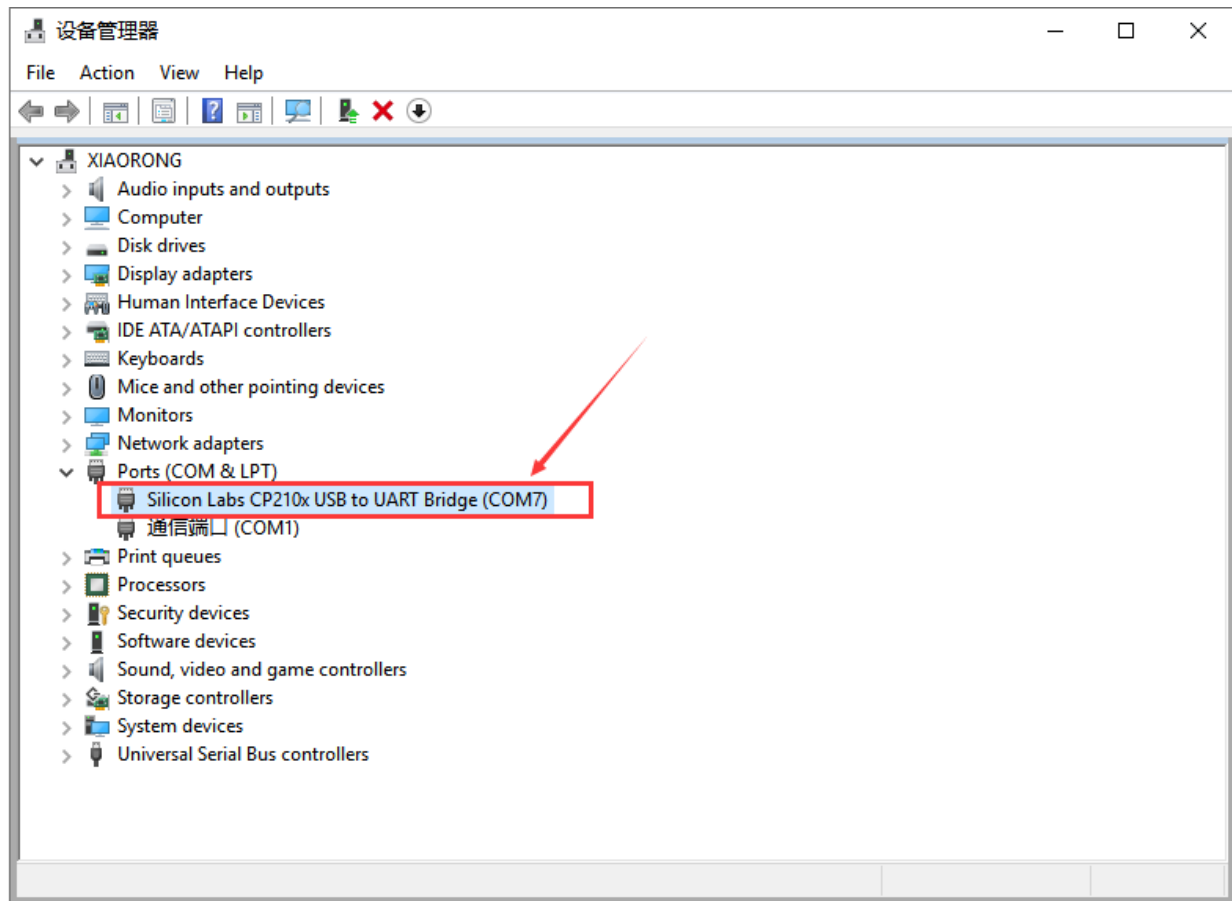
Click “Browse”, then search the driver of CP2102 and click “Next”,

There is a DRIVERS folder in Arduino software installed package  **arduino-1.8.12**, open driver folder and check the driver of CP210X series chips.

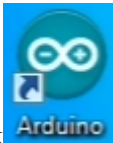


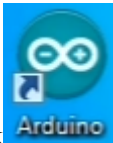
When opening the device manager, we will find the yellow exclamation mark disappear. The driver of CP2102 is installed successfully.

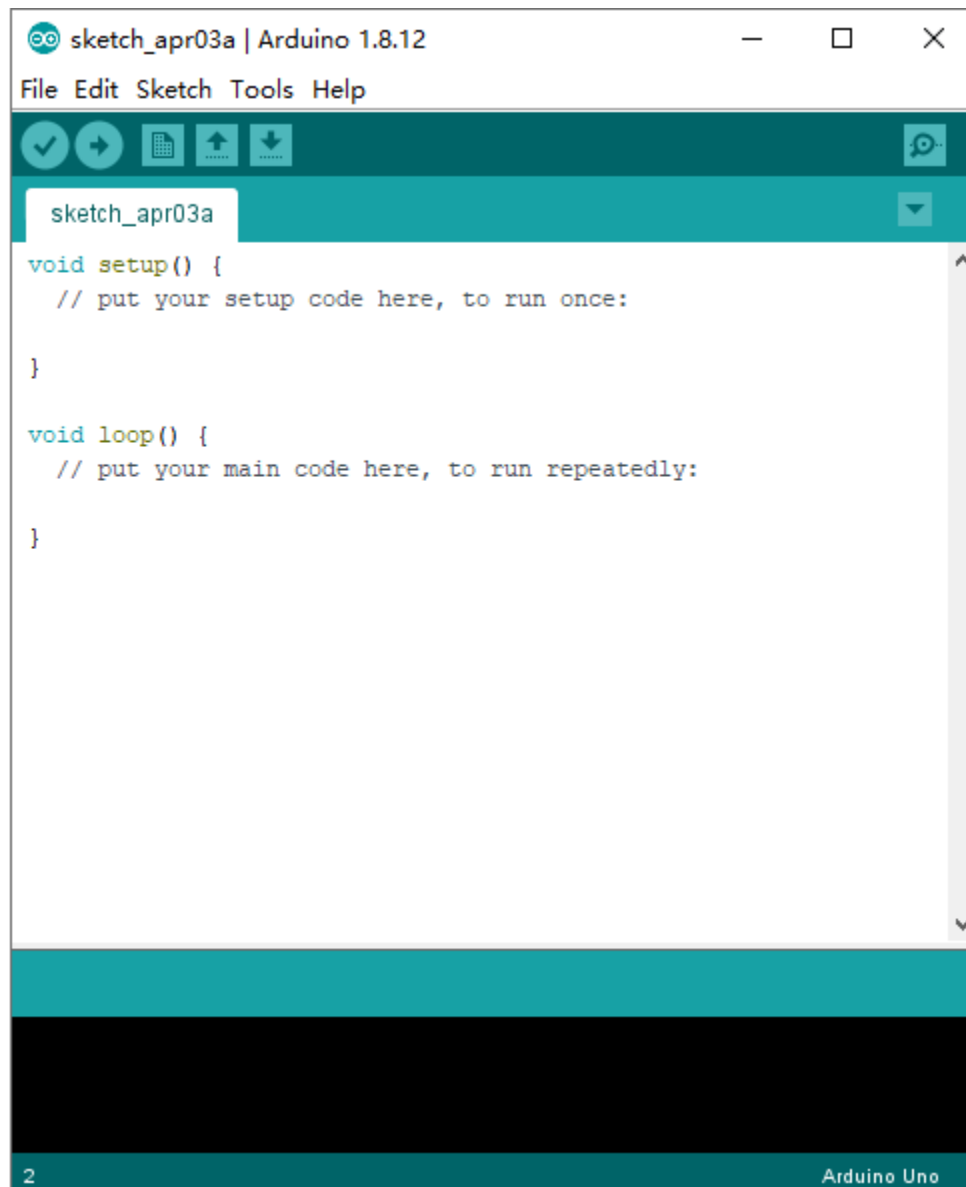




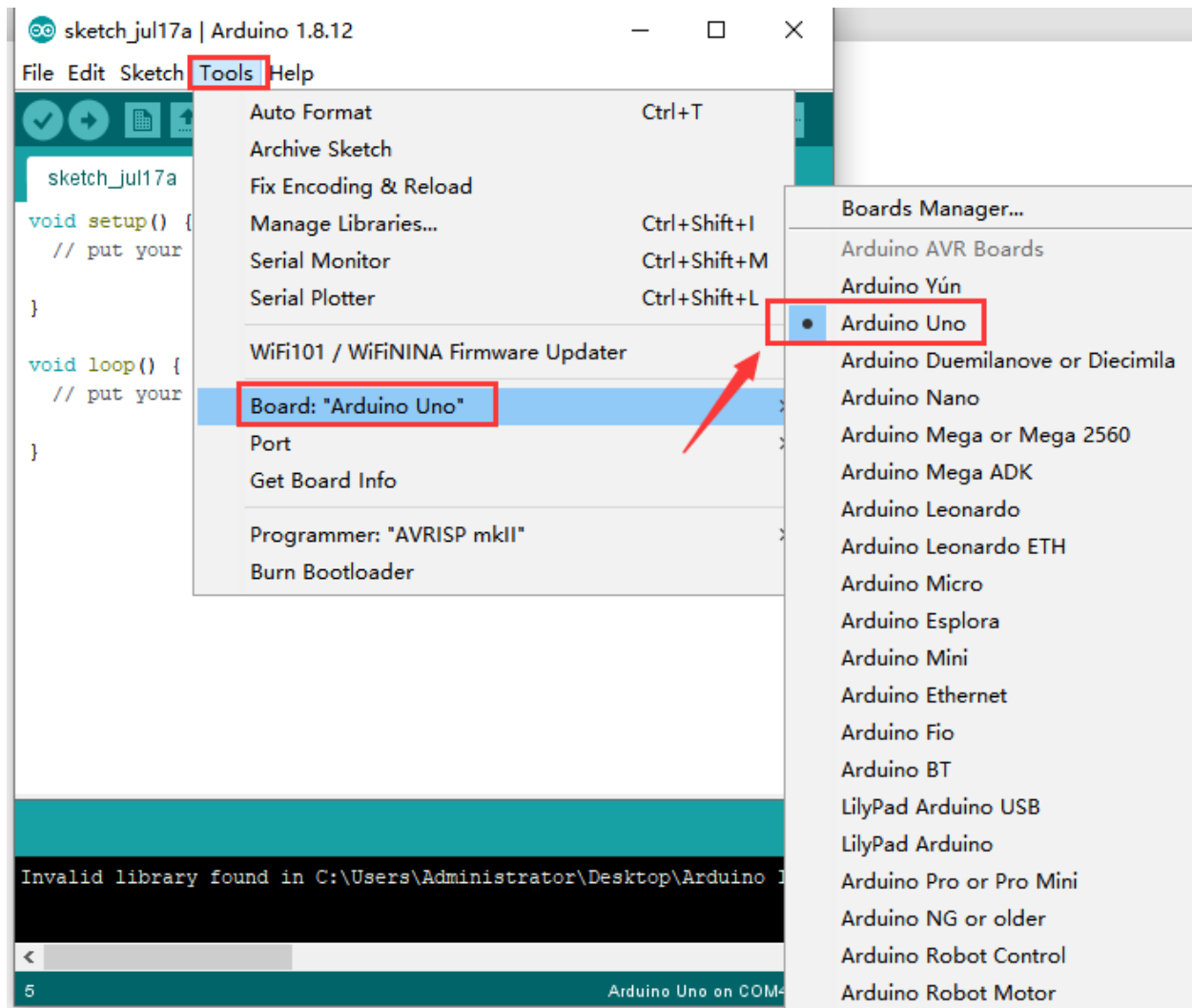
### 5.3 Arduino IDE Setting



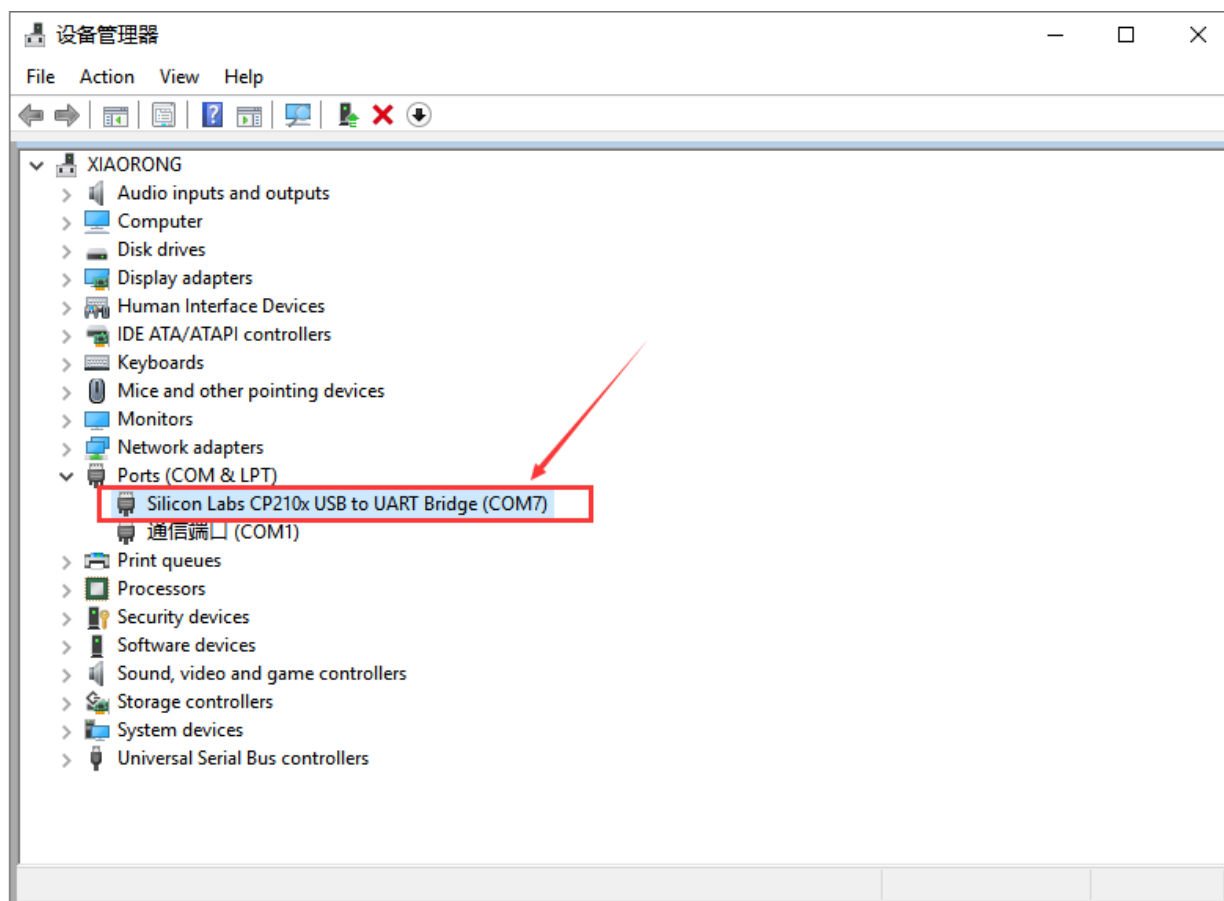
Click  icon open Arduino IDE.

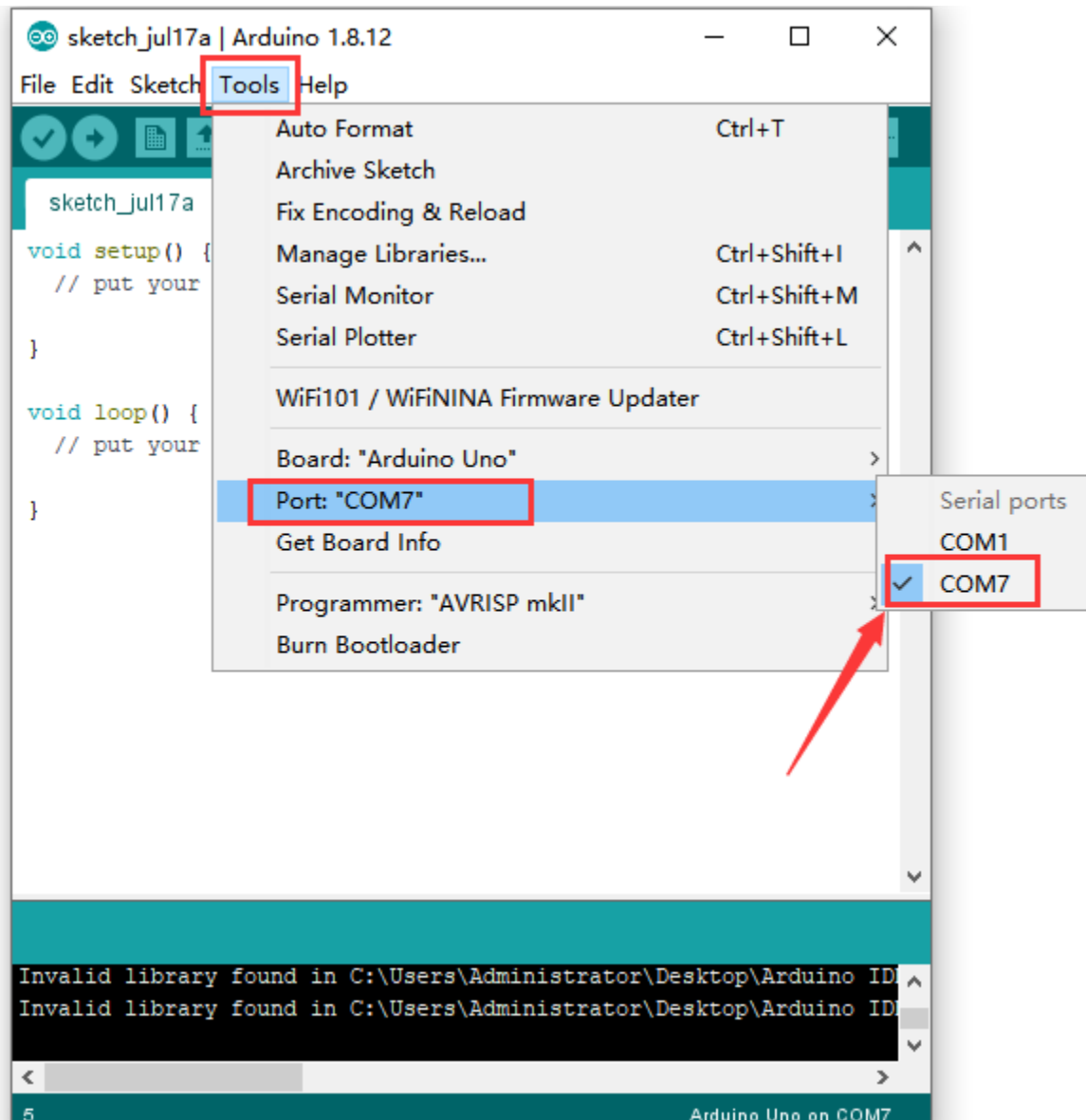


When downloading the sketch to the board, you must select the correct name of Arduino board that matches the board connected to your computer. As shown below;

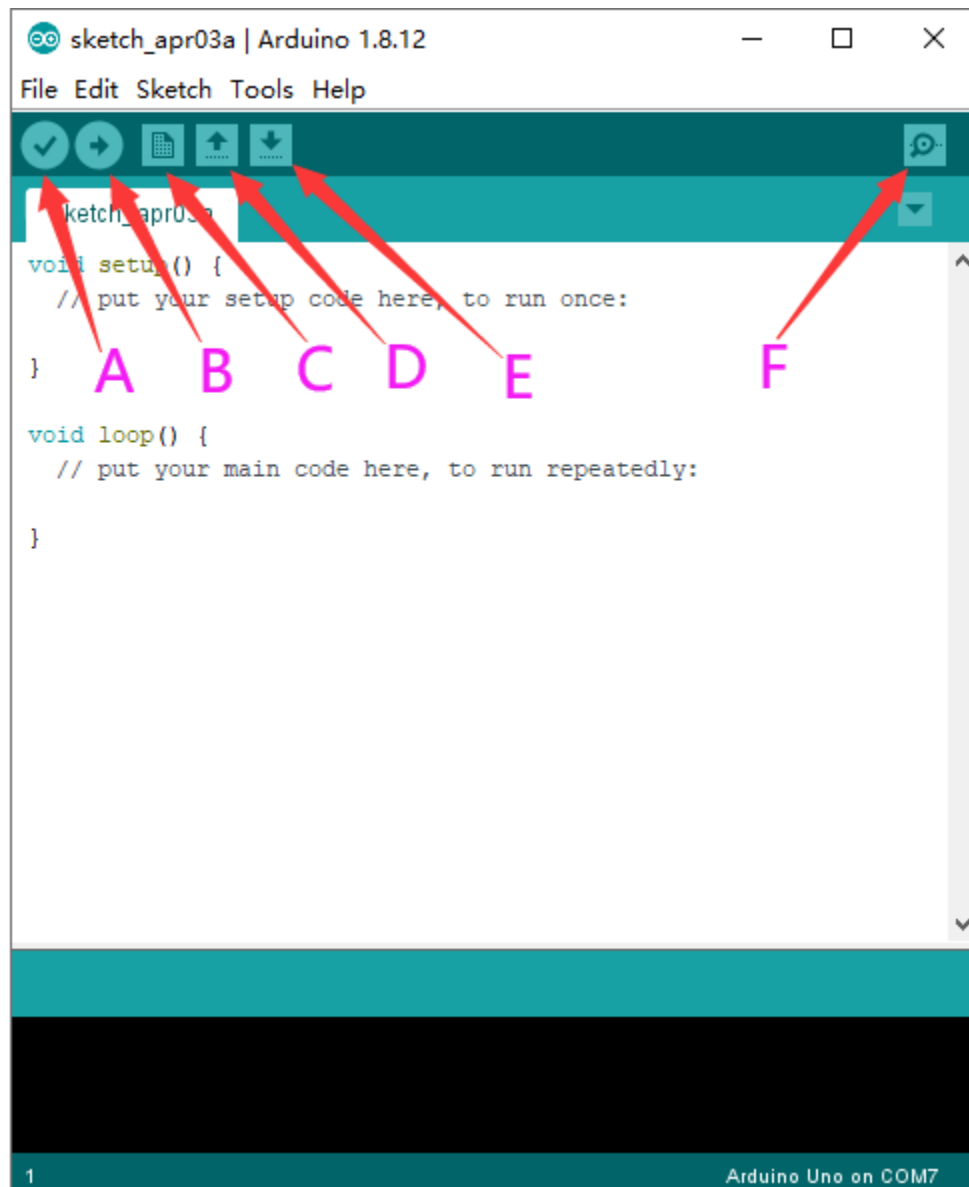


Then select the correct COM port (you can see the corresponding COM port after the driver is successfully installed)





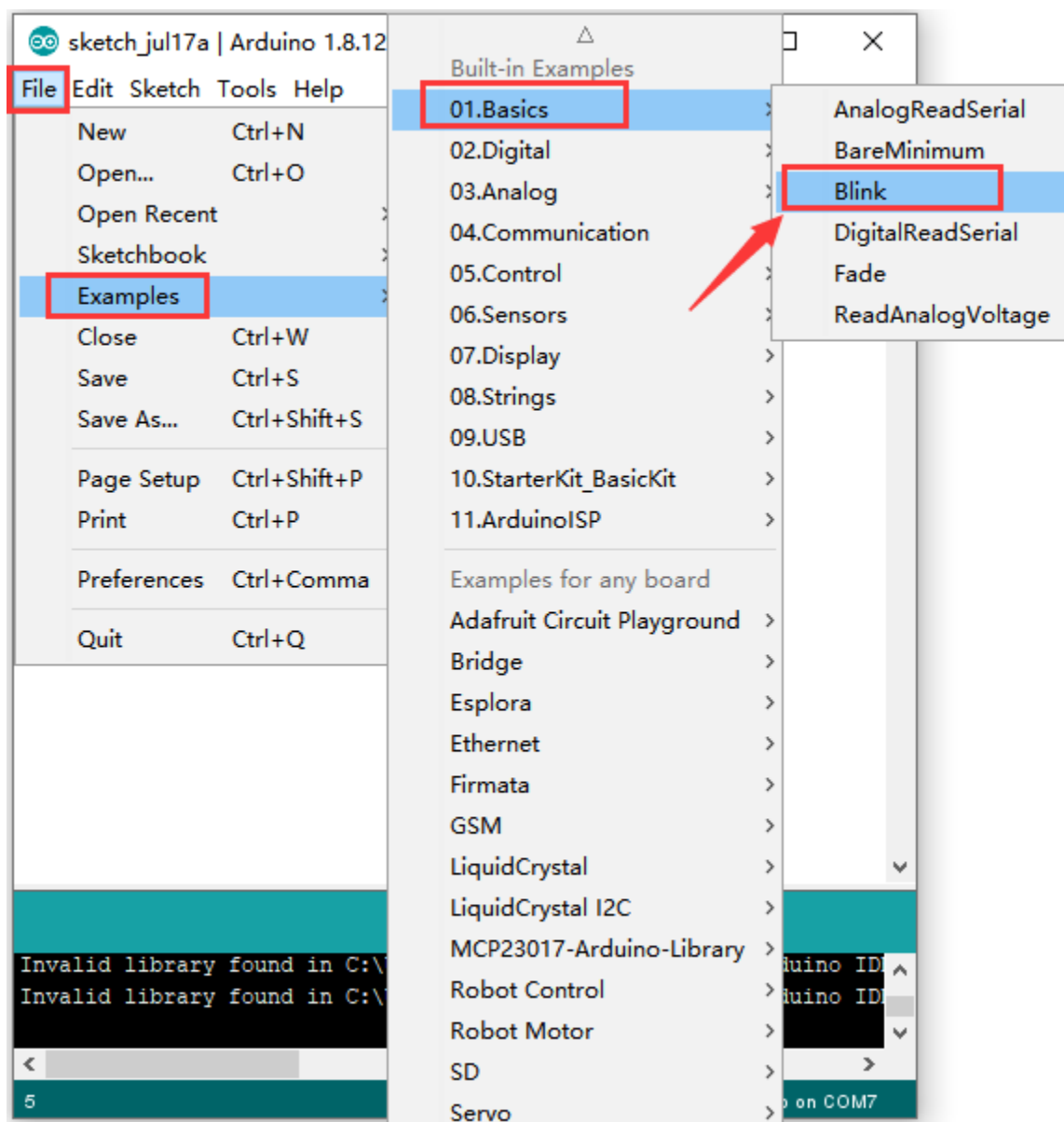


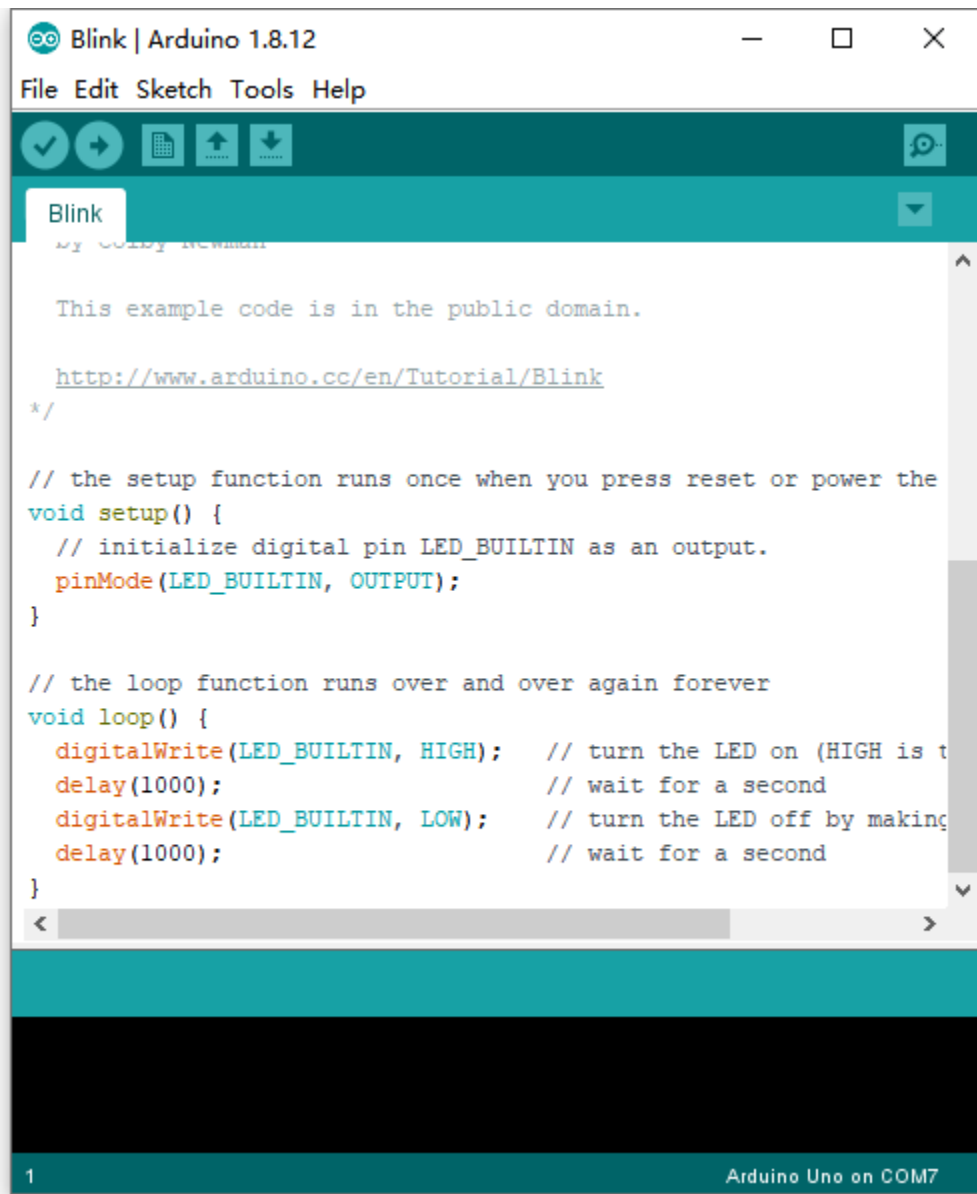


- A- Used to verify whether there is any compiling mistakes or not.
- B- Used to upload the sketch to your Arduino board.
- C- Used to create shortcut window of a new sketch.
- D- Used to directly open an example sketch.
- E- Used to save the sketch.
- F- Used to send the serial data received from board to the serial monitor.

#### 5.4 Start first program


Open the file to select **Example**, and click **BASIC>BLINK**, as shown below:

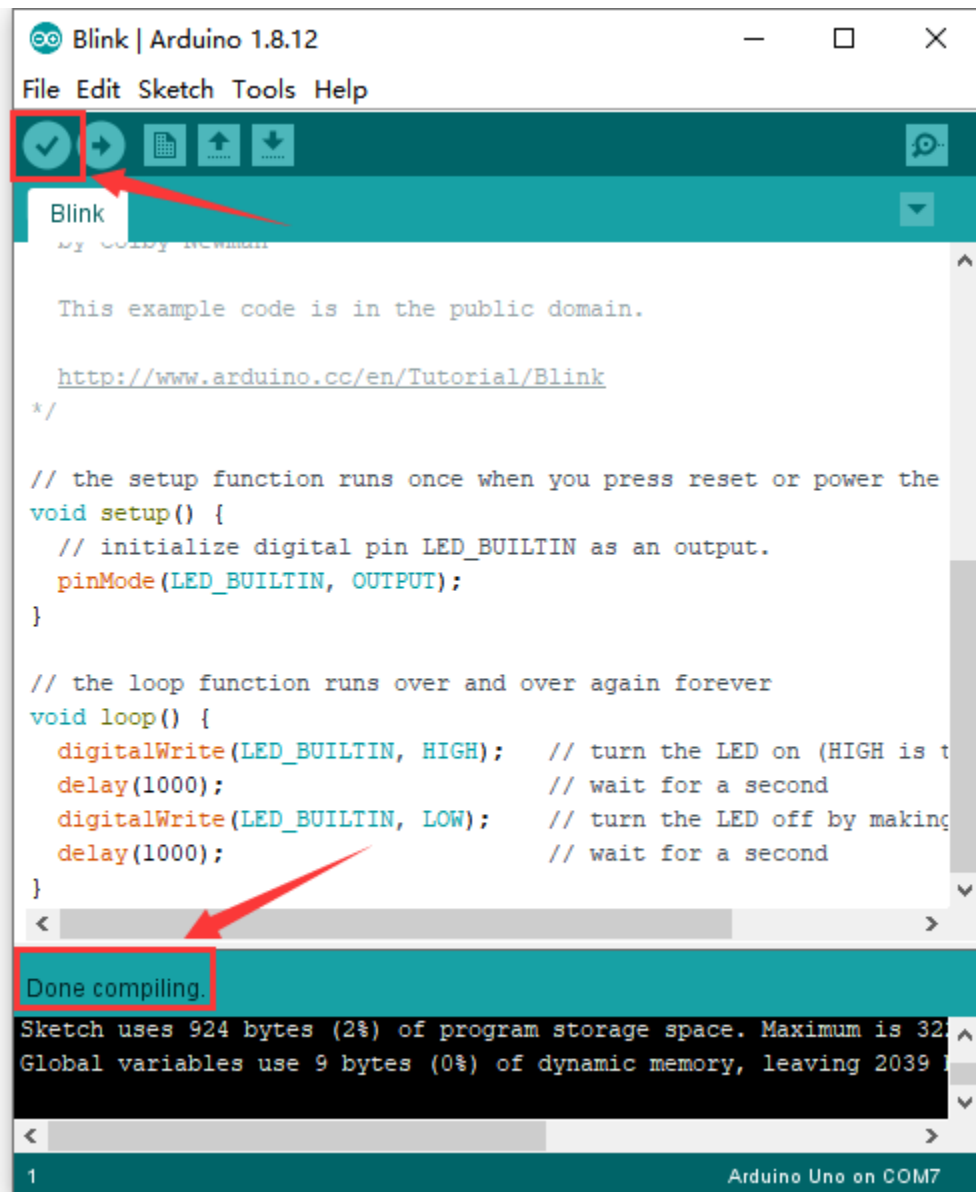




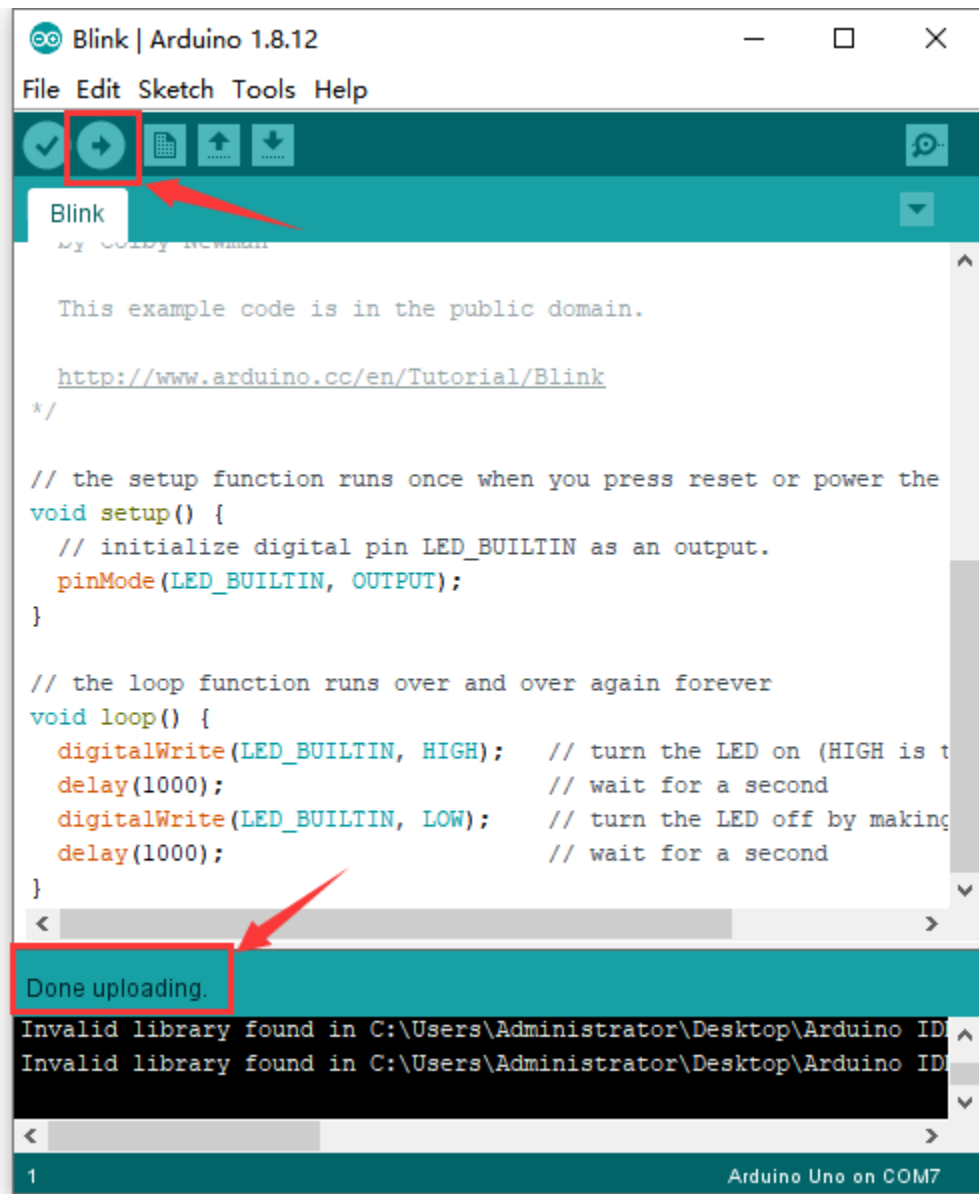
Set board and COM port, the corresponding board and COM port are shown on the lower right of IDE.



Click  to start compiling the program, check errors.



Click  to upload the program.



After the code is uploaded, the onboard LED will blink. Congratulations, you have finished the first program.

## 6. MAC System 6.1 Install Arduino IDE on MAC System

The installation instruction is the same as the chapter 5.1, as shown below:

## ARDUINO 1.8.12

Arduino IDE that can be used with any Arduino board, including the Arduino Yún and Arduino DUE. Refer to the [Getting Started](#) page for Installation instructions.

[See the release notes.](#)

**Windows** Installer

**Windows** ZIP file for non admin install

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

**Linux** 32 bits

**Linux** 64 bits

**Linux** ARM 32

**Linux** ARM 64

**Source**

6.2 Download the Driver of CP2102

<https://fs.keyestudio.com/CP2102-MAC>

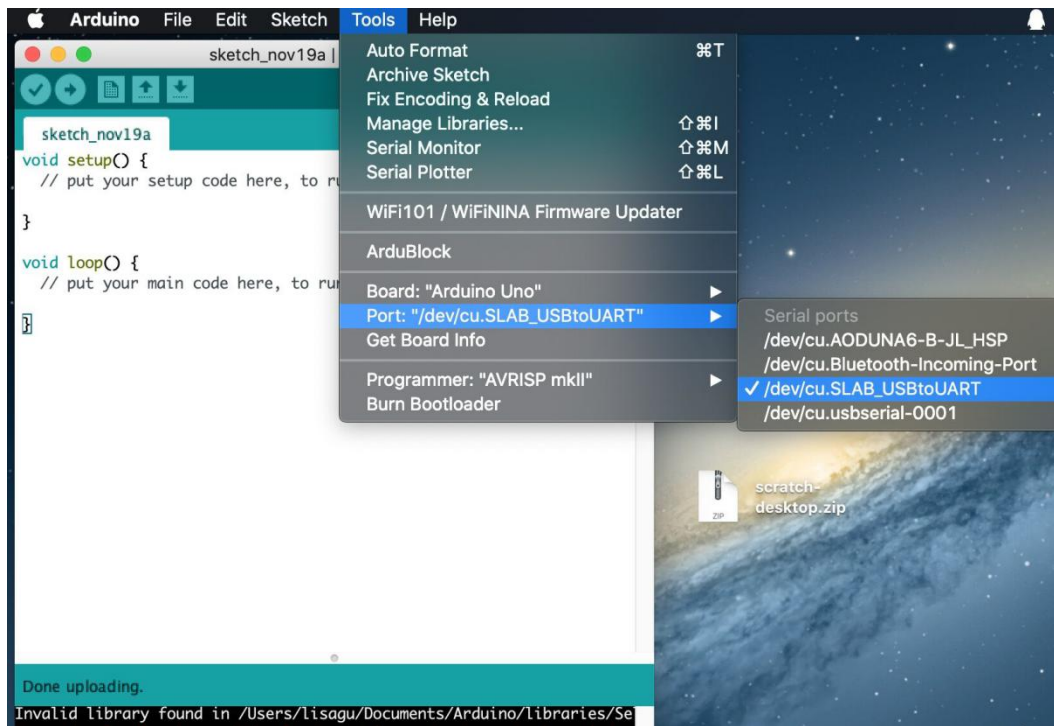
6.3 How to Install the Driver of CP2102

The following link is for your reference:

[https://wiki.keyestudio.com/How\\_to\\_Install\\_the\\_Driver\\_of\\_CP2102\\_on\\_MAC\\_System](https://wiki.keyestudio.com/How_to_Install_the_Driver_of_CP2102_on_MAC_System)

6.4 Arduino IDE Setting

The setting method is as same as the chapter 5.3 except from COM port, as shown below.

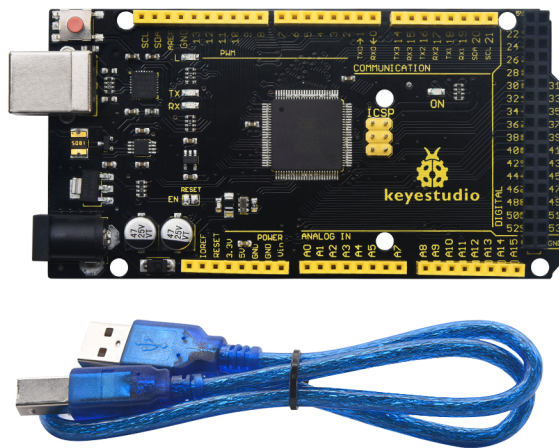


Resources:

<https://fs.keyestudio.com/KS0497>



## KS0498 KEYESTUDIO MEGA 2560 DEVELOPMENT BOARD



### 1. Description

The processor core of Keyestudio MEGA R3 development board is the [ATMEGA2560-16AU](#) , fully compatible with ARDUINO MEGA R3.

It has 54 digital input/output pins (of which 15 can be used as PWM outputs), 16 analog inputs, 4 UARTs (hardware serial ports), a 16 MHz crystal oscillator, a USB connection, a power jack, 1 ICSP header, and a reset button.

What's more, you can burn the firmware for ATMEGA2560-16AU through the built-in ICSP port. The firmware of this chip is burnt well before delivery. Therefore, you don't need to burn the firmware.

The power can be supplied through USB wire, DC head and Vin GND pins.

### 2.Specification

Microcontroller: ATMEGA2560-16AU

USB serial chipCP2102

Operating Voltage: 5V

Input Voltage (recommended):DC 7-12V

Digital I/O Pins: 54 (D0-D53)

PWM Digital I/O Pins15(D2-D13 D44-D46)

Analog Input Pins: 16(A0-A15)

DC Current per I/O Pin: 20 mA

DC Current for 3.3V Pin: 50 mA

Flash Memory: 256 KB of which 8 KB used by bootloader

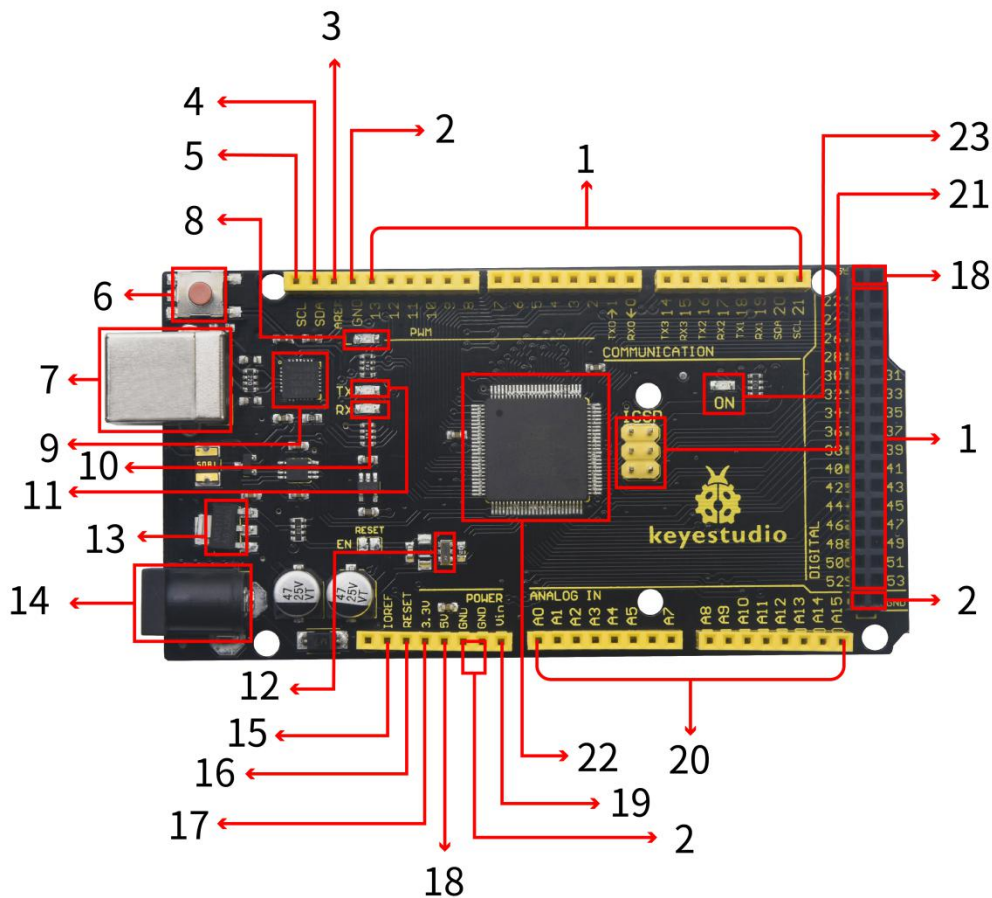
SRAM: 8 KB

EEPROM: 4 KB

Clock Speed: 16 MHz

LED\_BUILTIN:D13

### 3.Pins and Parts Introduction:



1	<b>Digital I/O</b>	Arduino MEGA has 54 digital input/output pins (of which 15 can be used as PWM outputs). These pins can be configured as digital input pin to read the logic value (0 or 1). Or used as digital output pin to drive different modules like LED, relay, etc. Using <code>pinMode()</code> , <code>digitalWrite()</code> , and <code>digitalRead()</code> functions.
2	<b>GND</b>	GND
3	<b>AREF</b>	Reference voltage (0-5V) for analog inputs. Used with <code>analogReference()</code> . Configures the reference voltage used for analog input (i.e. the value used as the top of the input range).
4	<b>SDA</b>	IIC communication pin
5	<b>SCL</b>	IIC communication pin
6	<b>RESET Button</b>	You can reset your Arduino board,
7	<b>USB Connection</b>	Arduino board can be powered via USB connector. All you needed to do is connecting the USB port to PC using a USB cable.
8	<b>D13 LED</b>	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.
9	<b>CP2102</b>	USB serial chip, translate the USB signal of computer into serial signal
10	<b>TX LED</b>	Onboard you can find the label: TX (transmit) When Arduino board communicates via serial port, send the message, TX led flashes.
11	<b>RX LED</b>	Onboard you can find the label: RX(receive ) When Arduino board communicates via serial port, receive the message, RX led flashes.
12	<b>Crystal Oscillator</b>	How does Arduino calculate time? by using a crystal oscillator. The number printed on the top of the Arduino crystal is 16.000H9H. It tells us that the frequency is 16,000,000 Hertz or 16MHz.
13	<b>Voltage Regulator</b>	To control the voltage provided to the Arduino board, as well as to stabilize the DC voltage used by the processor and other components. Convert an external input DC7-12V voltage into DC 5V, then switch DC 5V to the processor and other components.
14	<b>DC Power Jack</b>	Arduino board can be supplied with an external power DC7-12V from the DC power jack.
15	<b>IOREF</b>	This pin on the 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 for working with the 5V or 3.3V.
16	<b>RESET Header</b>	Connect an external button to reset the board. The function is the same as reset button.
17	<b>Power Pin 3V3</b>	A 3.3 volt supply generated by the on-board regulator. Maximum current draw is 50 mA.
18	<b>Power Pin 5V</b>	Provides 5V output voltage
19	<b>Vin</b>	You can supply an external power input DC7-12V through this pin to Arduino board.
20	<b>Analog Pins</b>	Onboard has 16 analog inputs, labeled A0 to A15.
21	<b>ICSP (In-Circuit Serial Programming) Header</b>	the AVR, an Arduino micro-program header consisting of MOSI, MISO, SCK, RESET, VCC, and GND. It is often called the SPI (serial peripheral interface) and can be considered an “extension” of the output. In fact, slave the output devices to the SPI bus host. When connecting to PC, program the firmware to ATMEGA2560-16AU.
22	<b>Microcontroller</b>	Each Arduino board has its own microcontroller. You can regard it as the brain of your board. The main IC (integrated circuit) on the Arduino is slightly different from the panel pair. Microcontrollers are usually from ATMEL. Before you load a new program on the Arduino IDE, you must know what IC is on your board. This information can be checked at the top of IC.
23	<b>Power LED Indicator</b>	Powering the Arduino, LED on means that your circuit board is correctly powered on. If LED is off, connection is wrong.

### 4.Specialized Functions of Some Pins:

- **Serial Communication:** D0 (RX0) and D1 (TX1); Serial 1: D19 (RX1) and D18 (TX1); Serial 2: D17 (RX2) and D16 (TX2); Serial 3: D15 (RX3) and D14 (TX3).

Used to receive (RX) and transmit (TX) TTL serial data. Pins 0 and 1 are also connected to the corresponding pins of the CP2102 USB-to-TTL Serial chip.

- **PWM Pins (Pulse-Width Modulation):** D2 to D13, and D44 to D46.

Provide 8-bit PWM output with the `analogWrite()` function.

- **External Interrupts:** D2 (interrupt 0), D3 (interrupt 1), D18 (interrupt 5), D19 (interrupt 4), D20 (interrupt 3), and D21 (interrupt 2).

These pins can be configured to trigger an interrupt on a low level, a rising or falling edge, or a change in level. See the `attachInterrupt()` function for details.

- **SPI communication:** D53 (SS), D52 (SCK), D51 (MOSI), D50 (MISO).

These pins support SPI communication using the [SPI library](#). The SPI pins are also broken out on the ICSP header, which is physically compatible with the Arduino Uno.

- **IIC communication:** D20 (SDA); D21 (SCL). Support TWI communication using the [Wire library](#).

## 4. Windows System

### 4.1 Download Arduino IDE

When getting this control board, we need to install Arduino IDE

Enter the website <https://www.arduino.cc/> and click



The screenshot shows the Arduino website's navigation bar with 'SOFTWARE' and 'DOWNLOADS' buttons highlighted. Below the navigation bar, there's a 'WHAT IS ARDUINO?' section with an image of an Arduino Uno and buttons for 'BUY AN ARDUINO', 'LEARN ARDUINO', and 'DONATE'. To the right, there's a 'LATEST' section titled 'ARDUINO EDUVISION' with a discussion about the importance of coding, dated October 15th. Further right, there's a section titled 'NEW BOARDS AND SHIELDS AND KITS, OH MY! CHECK THEM OUT!' with a background of cardboard boxes.

You can select the latest version—1.8.13. Alternatively, the previous release is your another choice.

## Download the Arduino IDE



**ARDUINO 1.8.13**

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 7 and up

**Windows** ZIP file for non admin install

**Windows app** Requires Win 8.1 or 10

**Get**

**Mac OS X** 10.10 or newer

**Linux** 32 bits

**Linux** 64 bits

**Linux** ARM 32 bits

**Linux** ARM 64 bits

Release Notes

Source Code

Checksums (sha512)

In this project, we use 1.8.12 version

### Previous Releases

Download the [previous version of the current release](#) the classic [Arduino 1.0.x](#), or the [Arduino 1.5.x Beta version](#).

All the [Arduino 00xx versions](#) are also available for download. The Arduino IDE can be used on Windows, Linux (both 32 and 64 bits), and Mac OS X.

Click [previous version of the current release](#) to view the below page

**ARDUINO 1.8.12**

Arduino IDE that can be used with any Arduino board, including the Arduino Yún and Arduino DUE. Refer to the [Getting Started](#) page for Installation instructions.

[See the release notes.](#)

**Windows** Installer

**Windows** ZIP file for non admin install

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

**Linux** 32 bits

**Linux** 64 bits

**Linux** ARM 32

**Linux** ARM 64

**Source**

The **Windows installer** needs installing manually. Yet , the **Windows zip file for non admin install** a zip file of Arduino 1.8.12 version, can be directly downloaded and installed.

## Contribute to the Arduino Software

Consider supporting the Arduino Software by contributing to its development. (US tax payers, please note this contribution is not tax deductible). [Learn more on how your contribution will be used.](#)



JUST DOWNLOAD

CONTRIBUTE & DOWNLOAD

JUST DOWNLOAD

CONTRIBUTE & DOWNLOAD

Click icon

to download Arduino IDE.

### 4.2 Installing Driver

Let's install the driver of MEGA R3 Development Board. The USB-TTL chip of 2560 R3 board adopts CP2102 serial chip.

Its driver files are included in the Arduino 1.8 version and above.

When the board is attached to the computer, the computer can recognize the driver of CP2102 automatically.

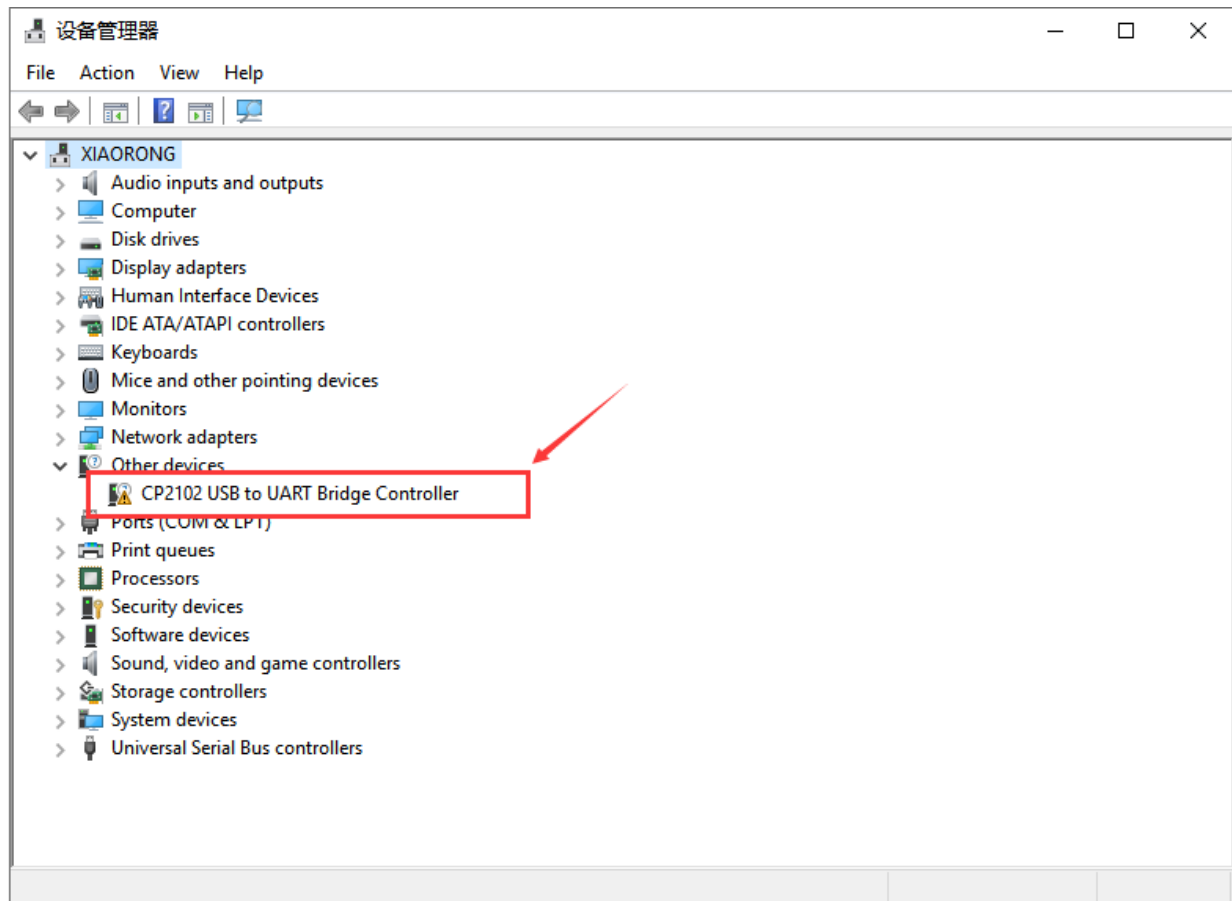
**Note: If the version of Arduino IDE you download is below 1.8, you need to download the driver of CP2102.**

**Download the driver of CP2102:**

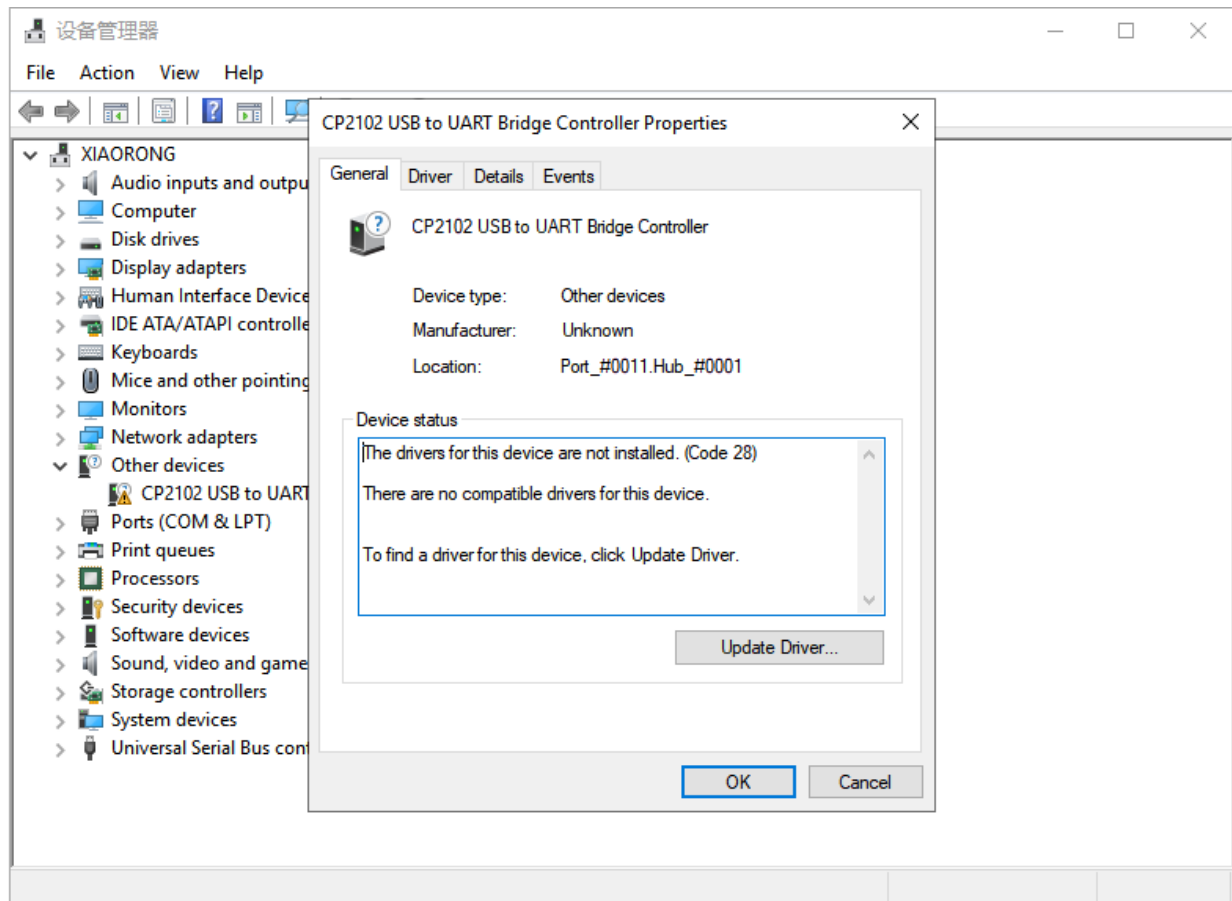
<https://fs.keyestudio.com/CP2102-WIN>

If it is installed unsuccessfully, you need to install it manually.

Click Computer— Properties— Device Manager, as shown below:

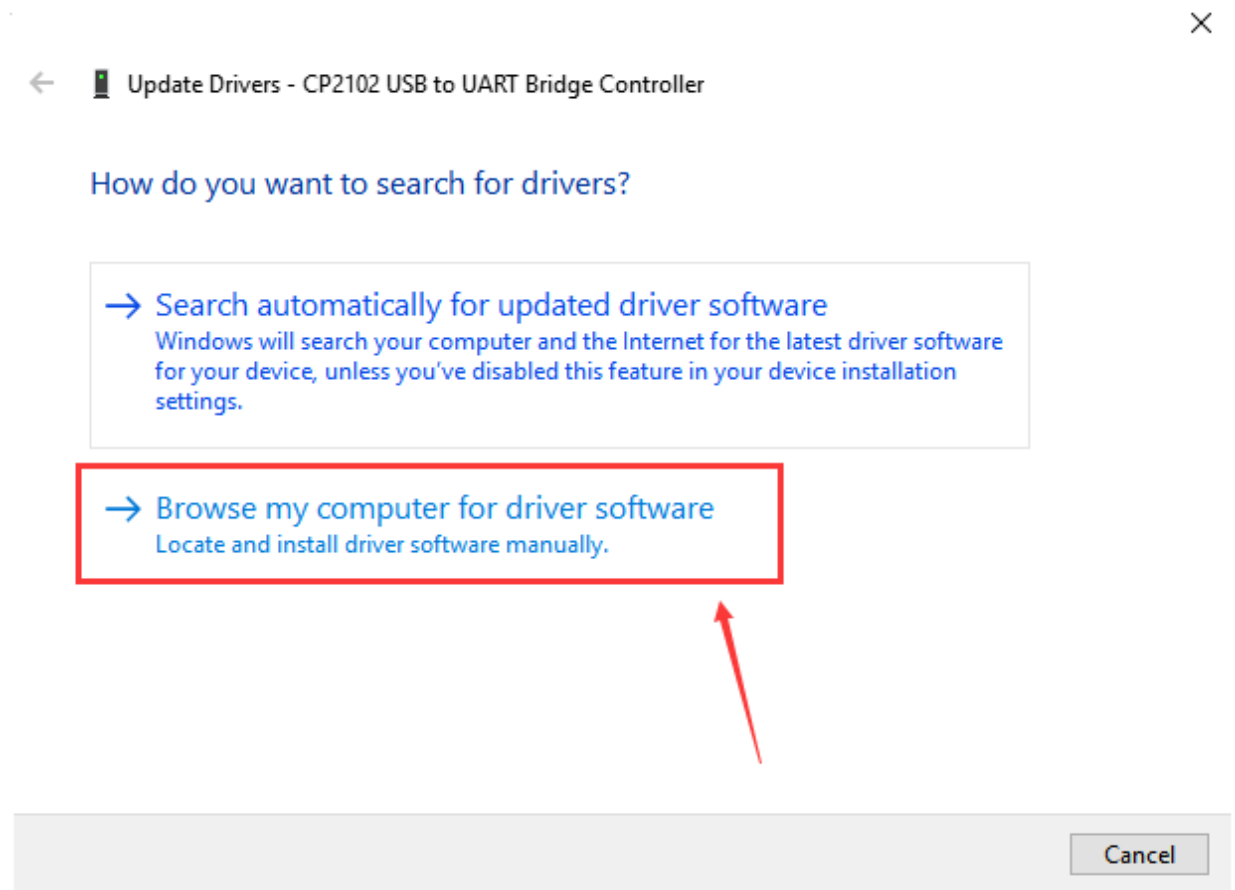


There is a yellow exclamation mark on the page, which implies the failed installation of the driver of CP2102.




Click “OK” to enter the following page and click “browse my computer for updated driver software”.





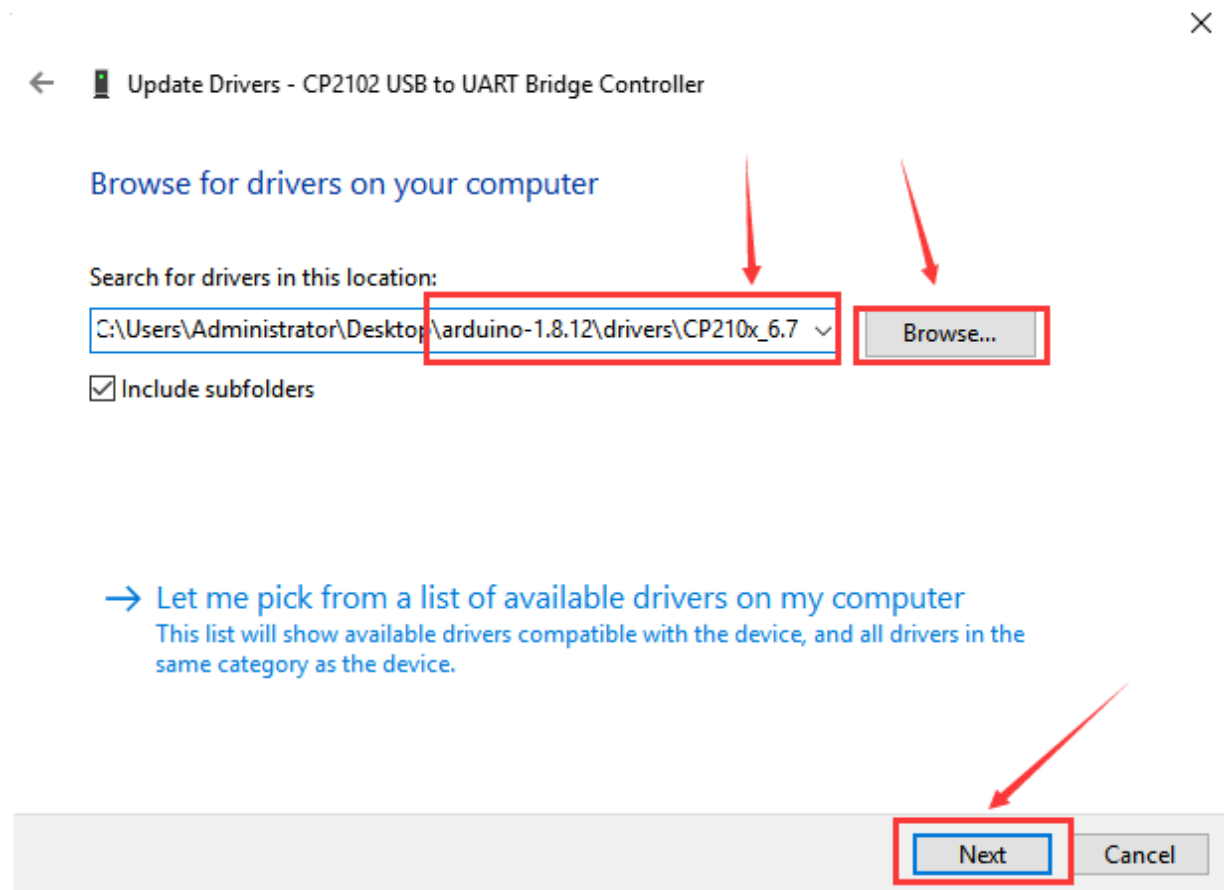
Click “Browse”, then search the driver of CP2102 and click “Next”,

There is a DRIVERS folder in Arduino software installed package

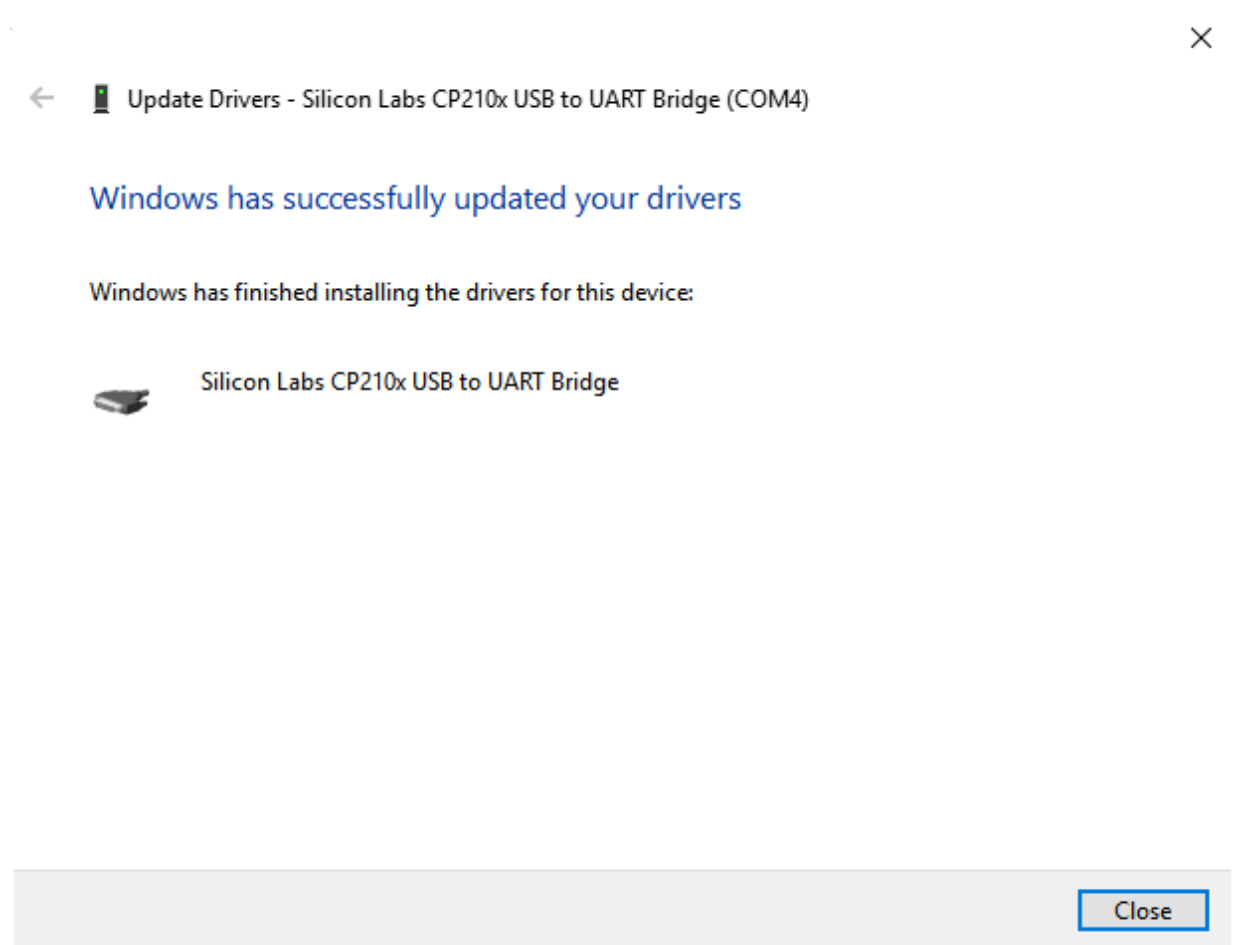
 **arduino-1.8.12**

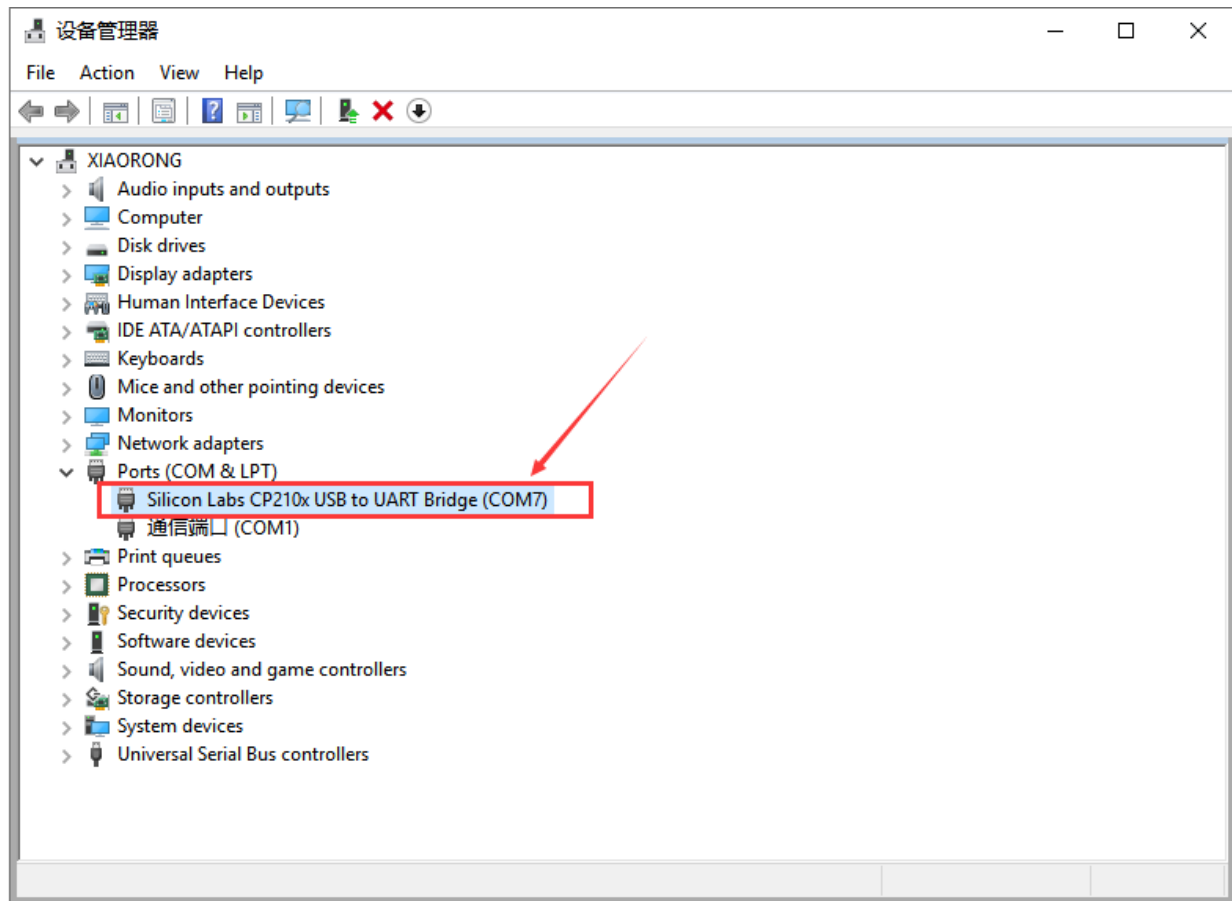
. Open driver folder and

check the driver of CP210X series chips.

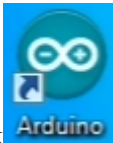


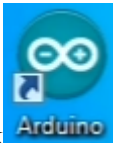
When opening the device manager, we will find the yellow exclamation mark disappear. The driver of CP2102 is installed successfully.

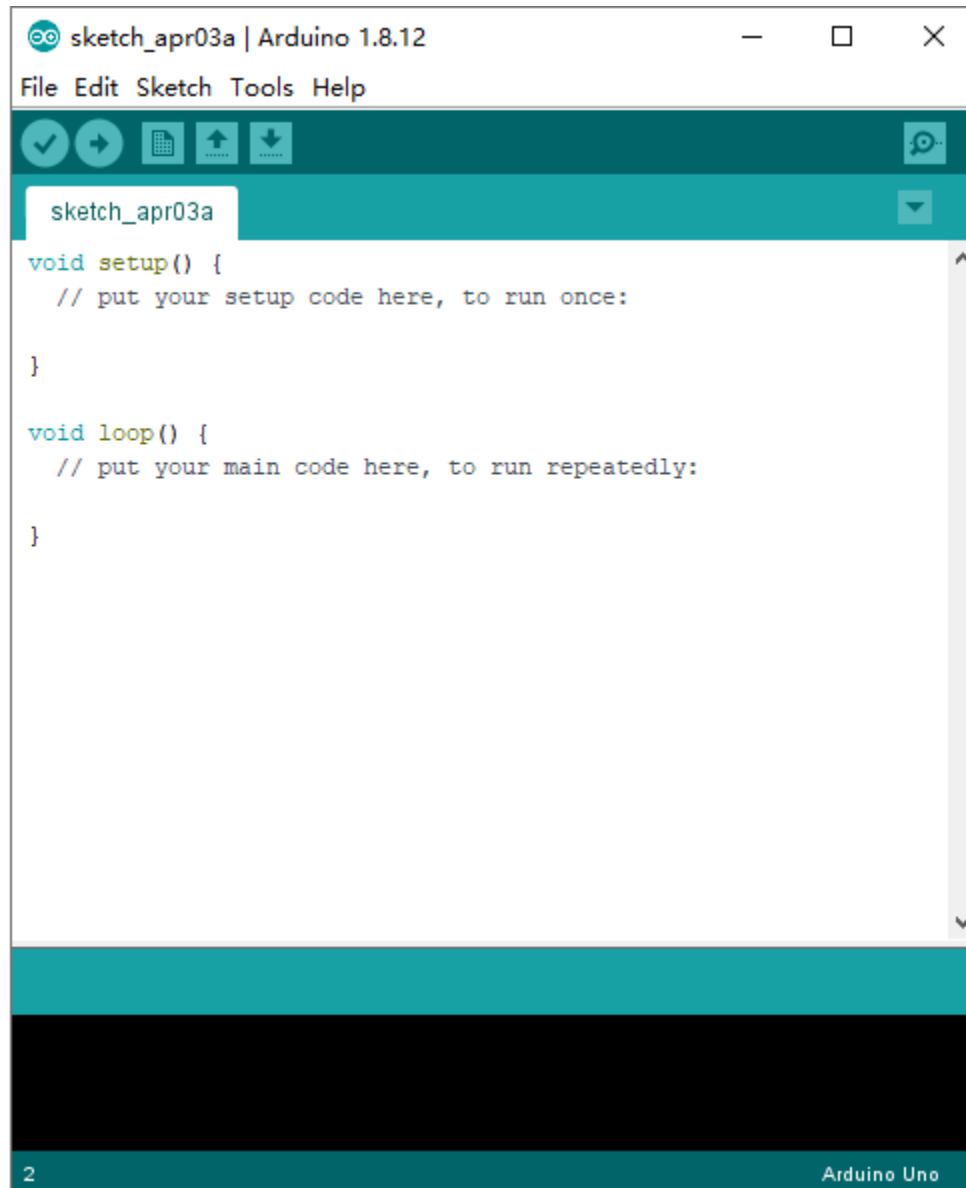




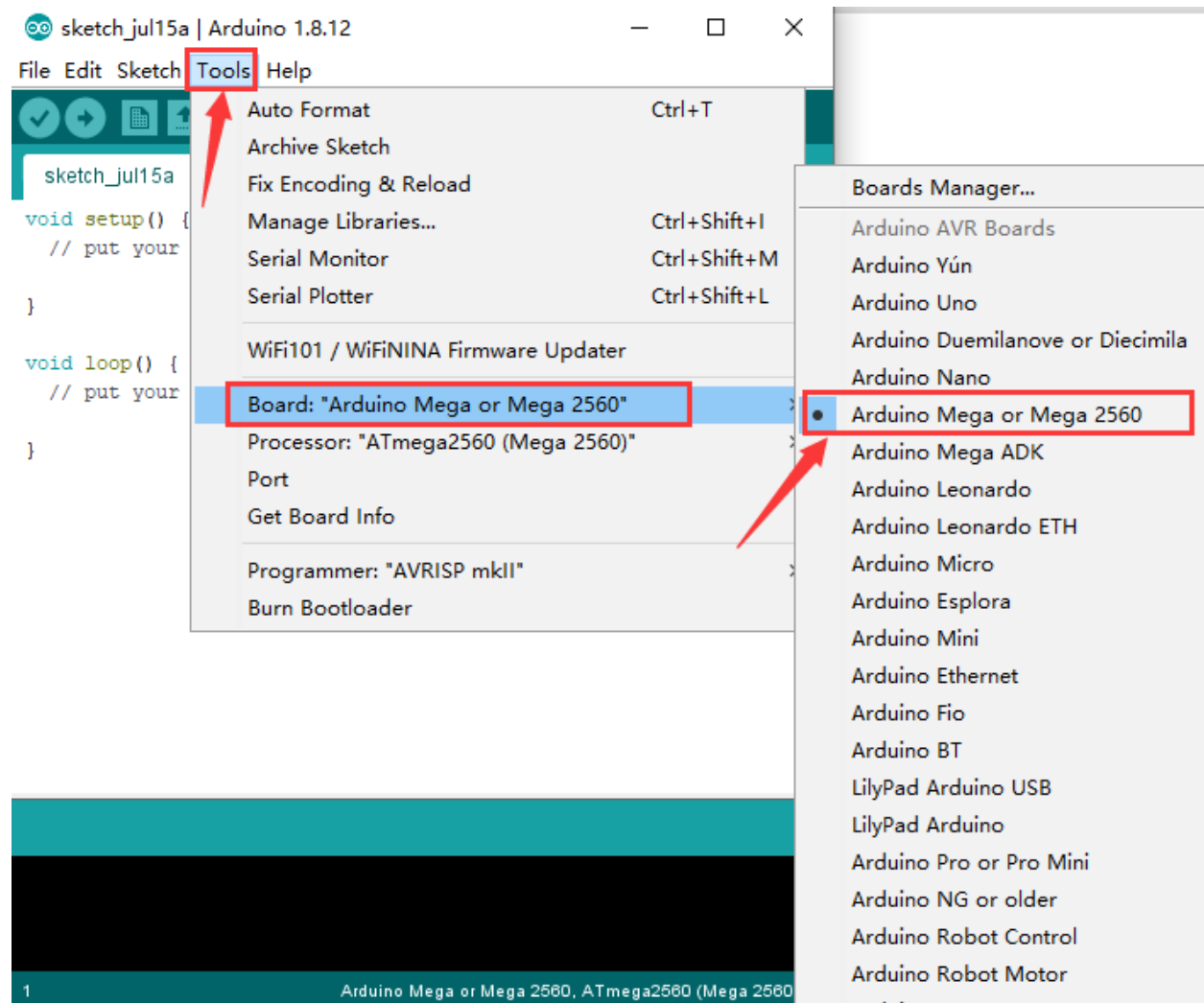
### 4.3 Arduino IDE Setting



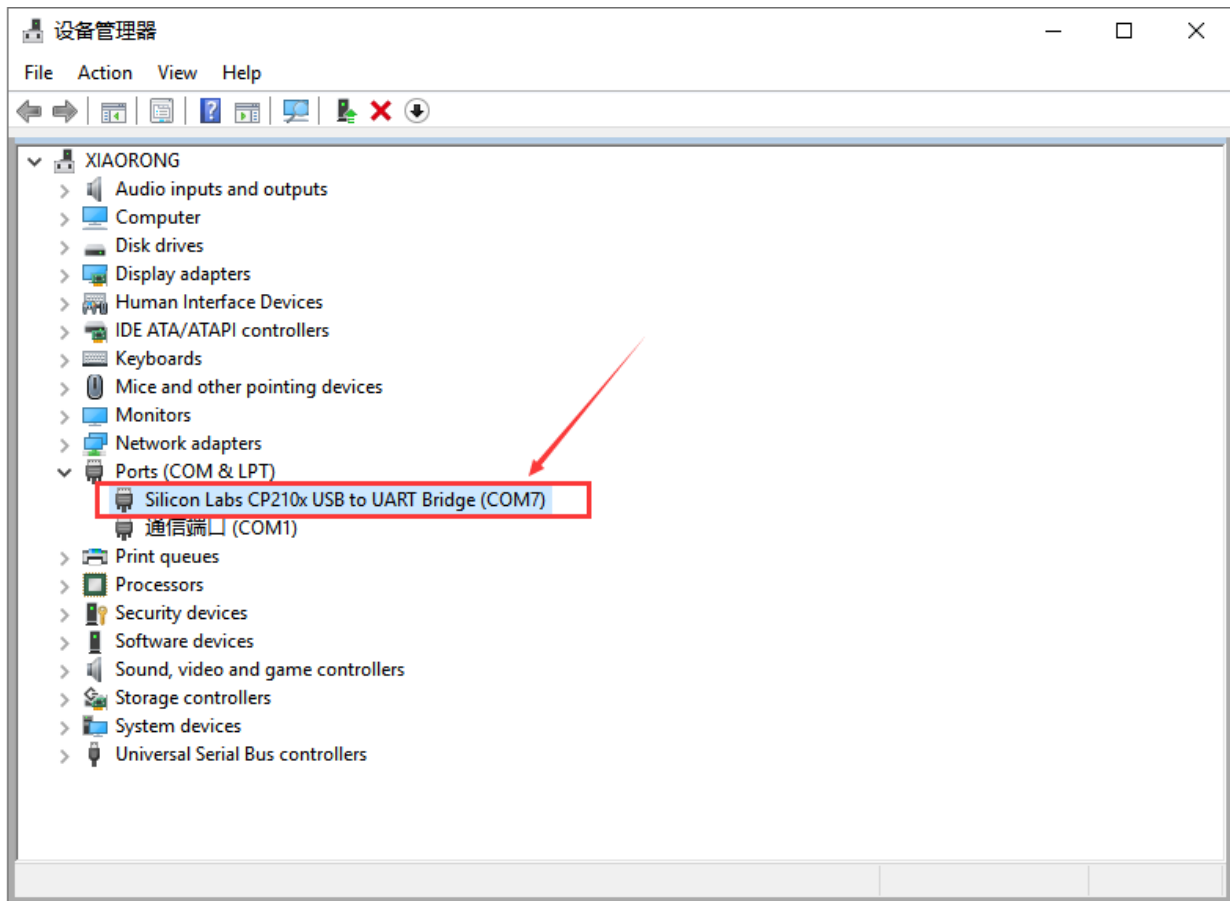
Click  icon to open Arduino IDE.

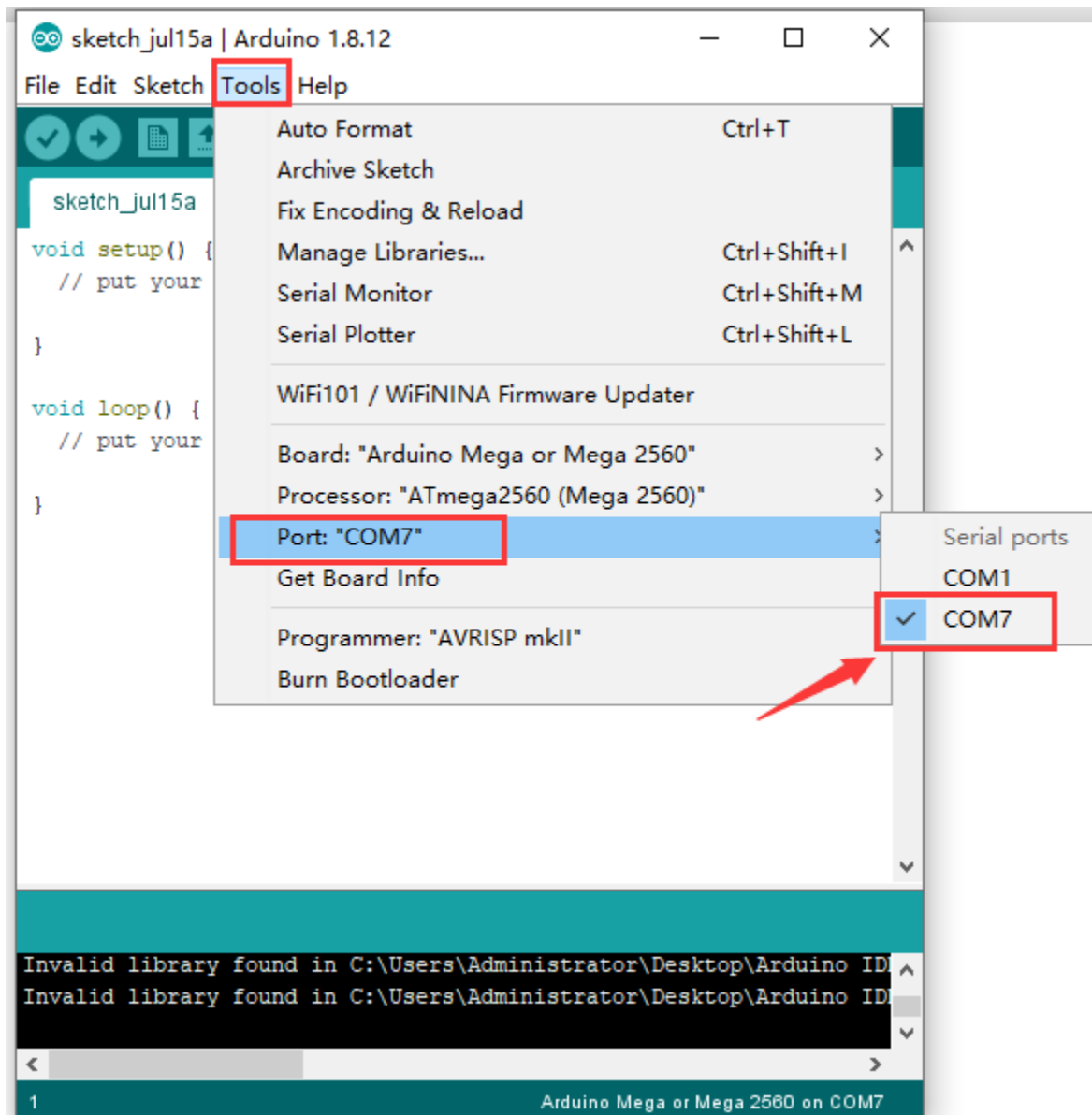


When downloading the sketch to the board, you must select the correct name of Arduino board that matches the board connected to your computer. As shown below;

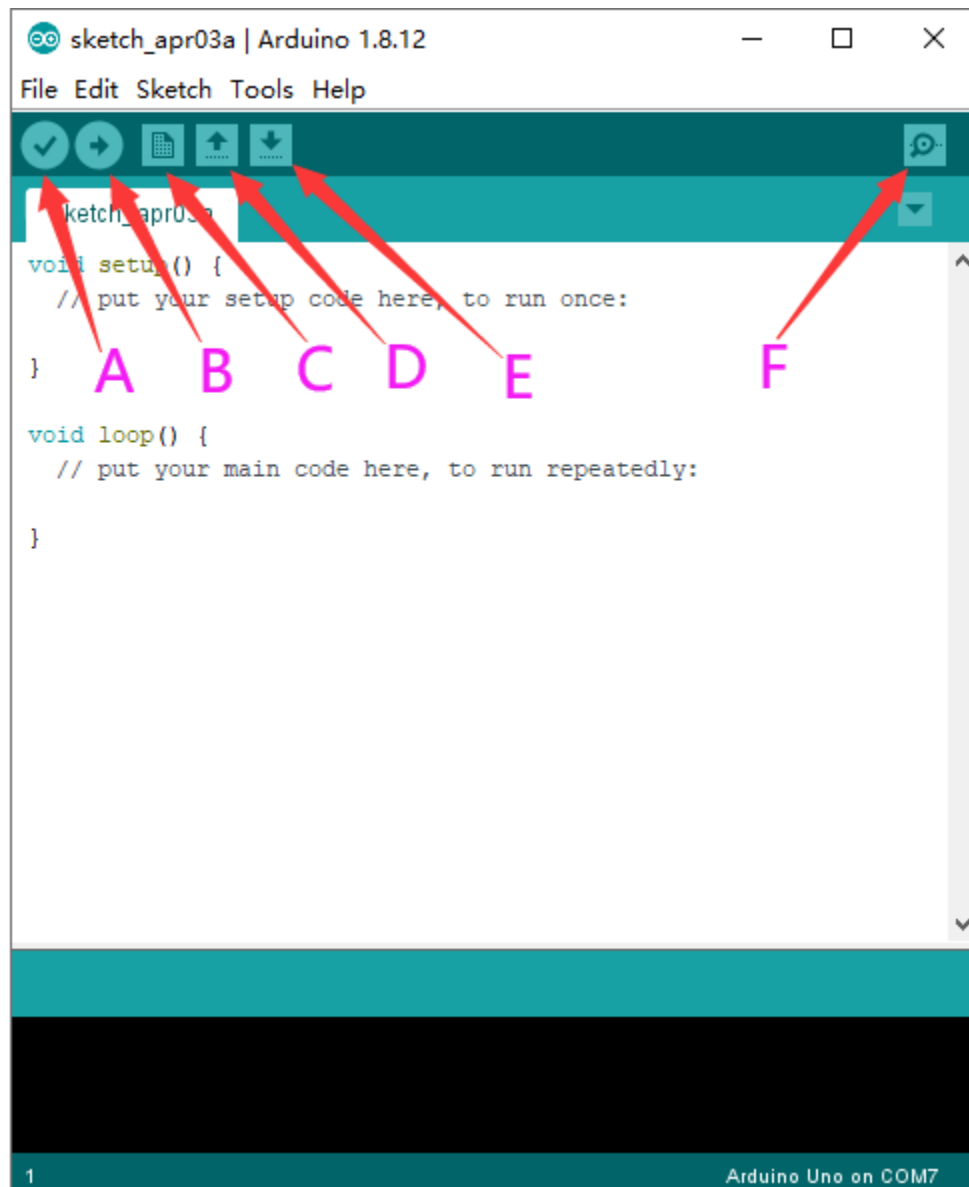


Then select the correct COM port (you can see the corresponding COM port after the driver is successfully installed)





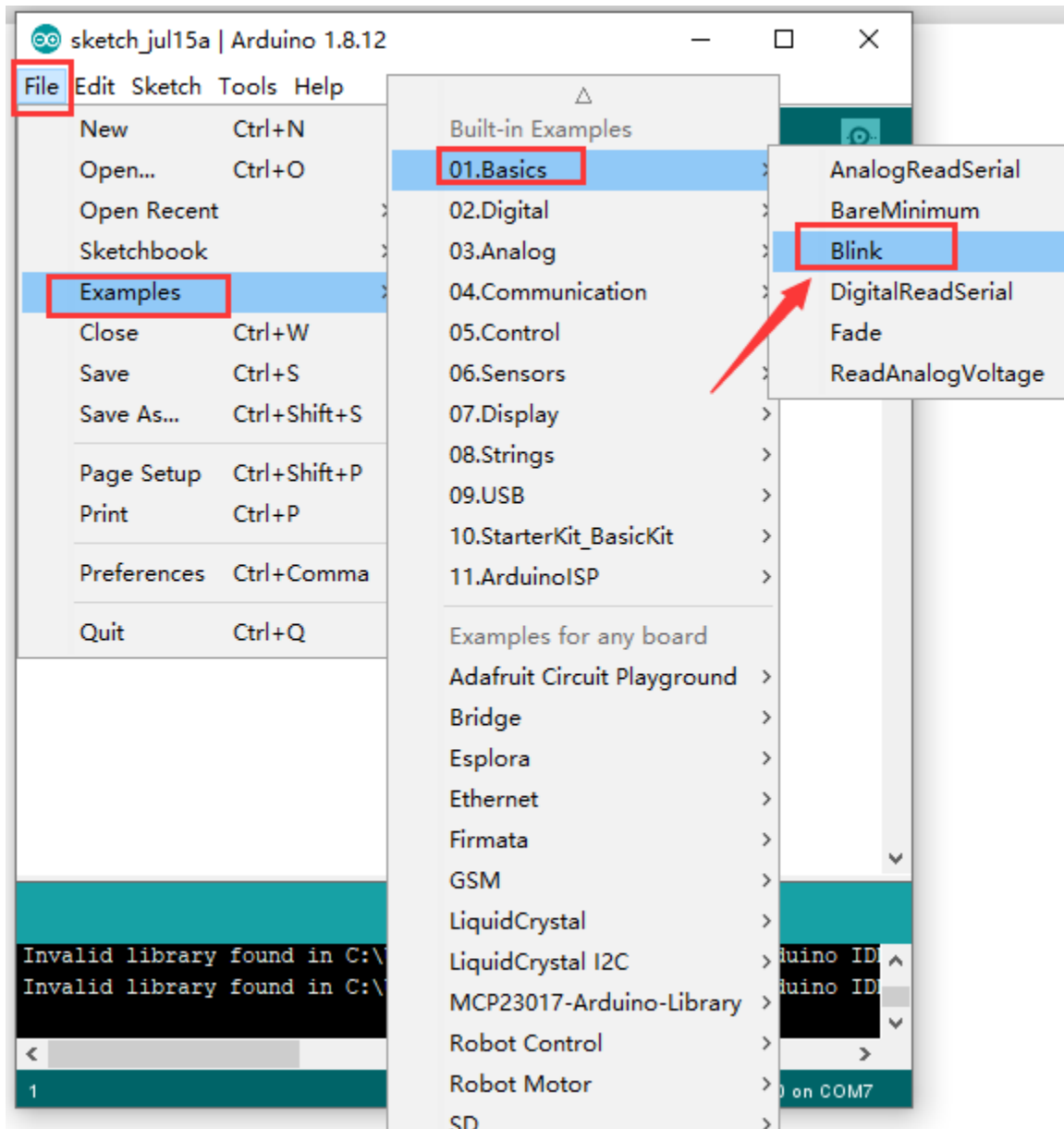


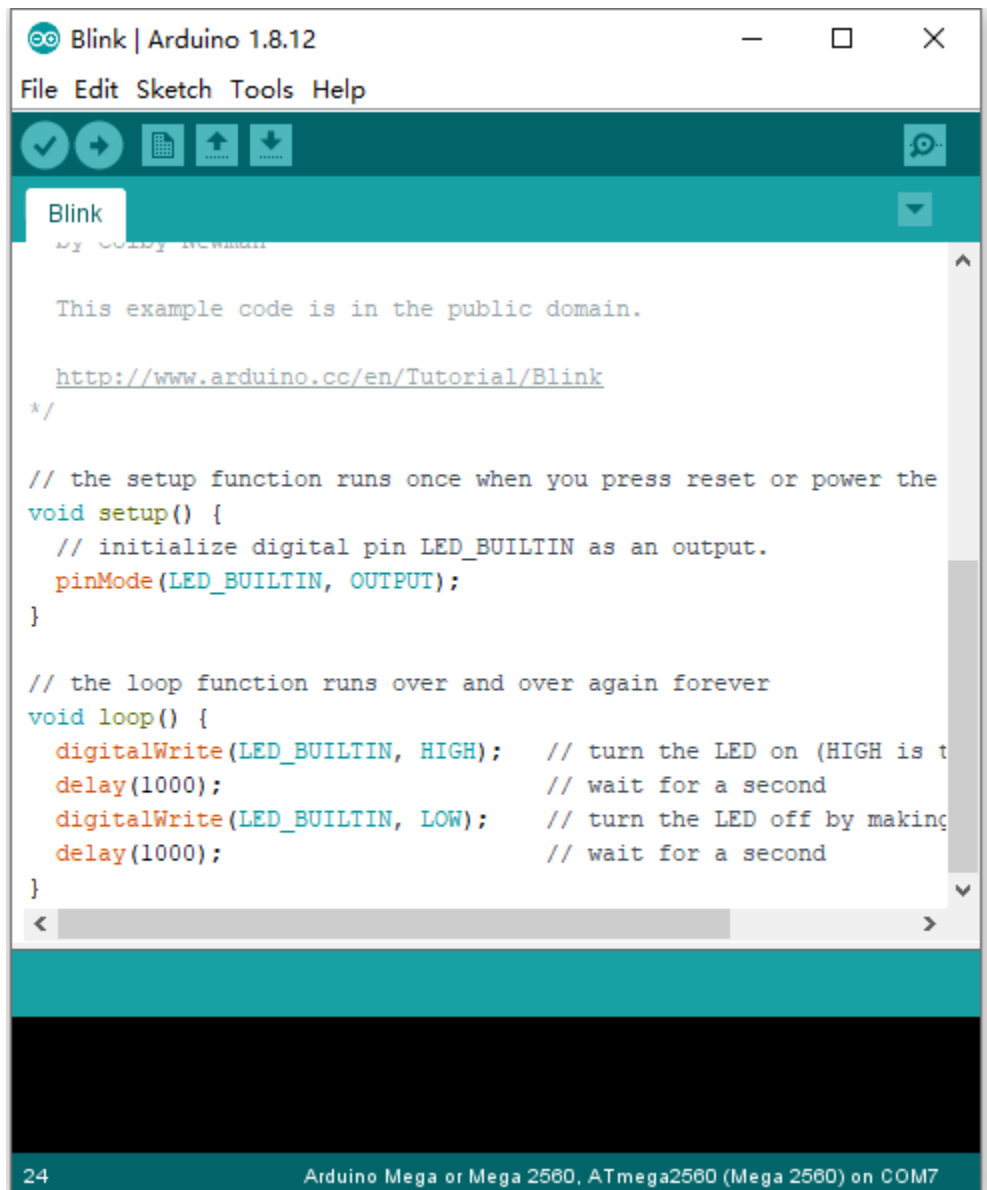


- A- Used to verify whether there is any compiling mistakes or not.
- B- Used to upload the sketch to your Arduino board.
- C- Used to create shortcut window of a new sketch.
- D- Used to directly open an example sketch.
- E- Used to save the sketch.
- F- Used to send the serial data received from board to the serial monitor.

#### 4.4 Start a Program


Open the file to select **Example**, and click **BASIC>BLINK**, as shown below:

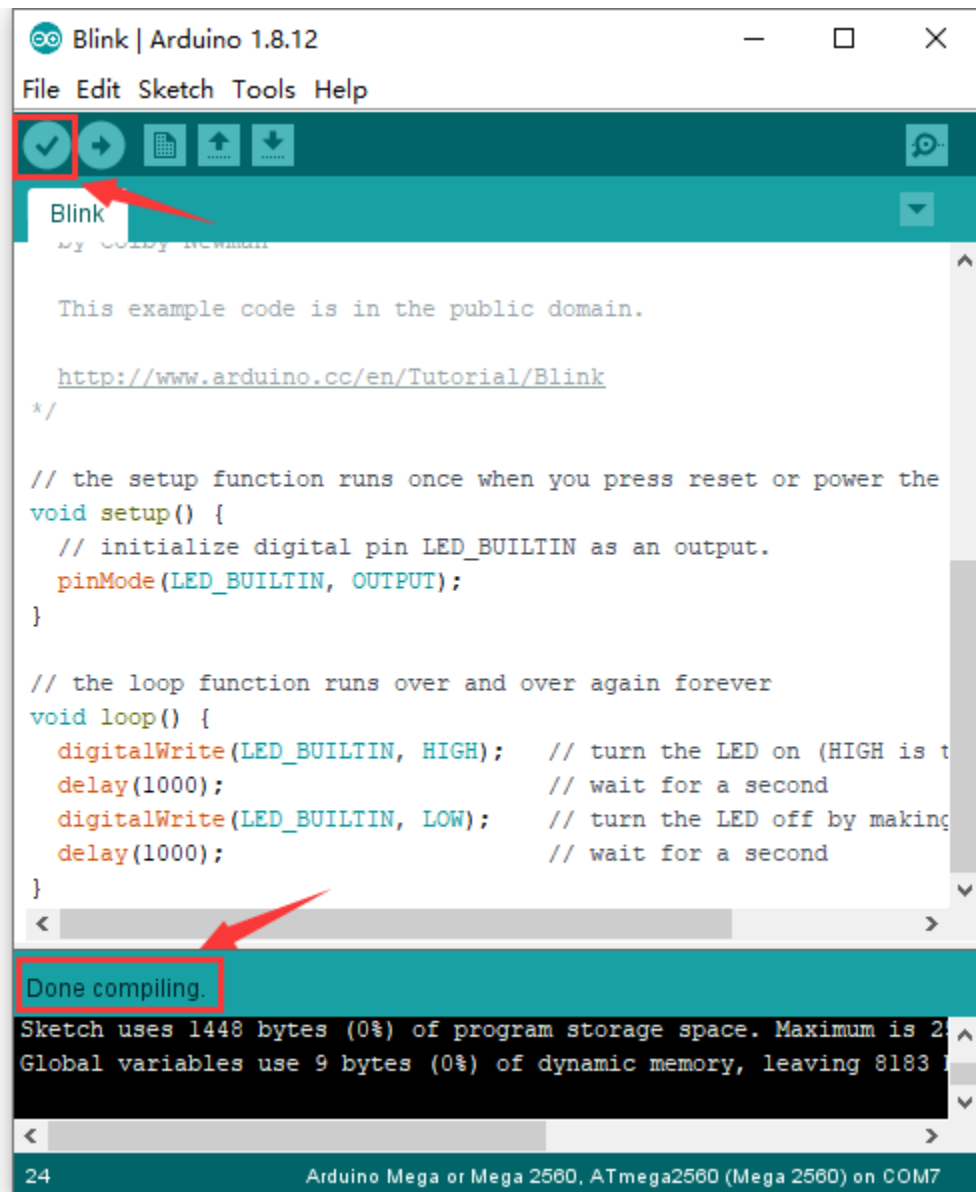




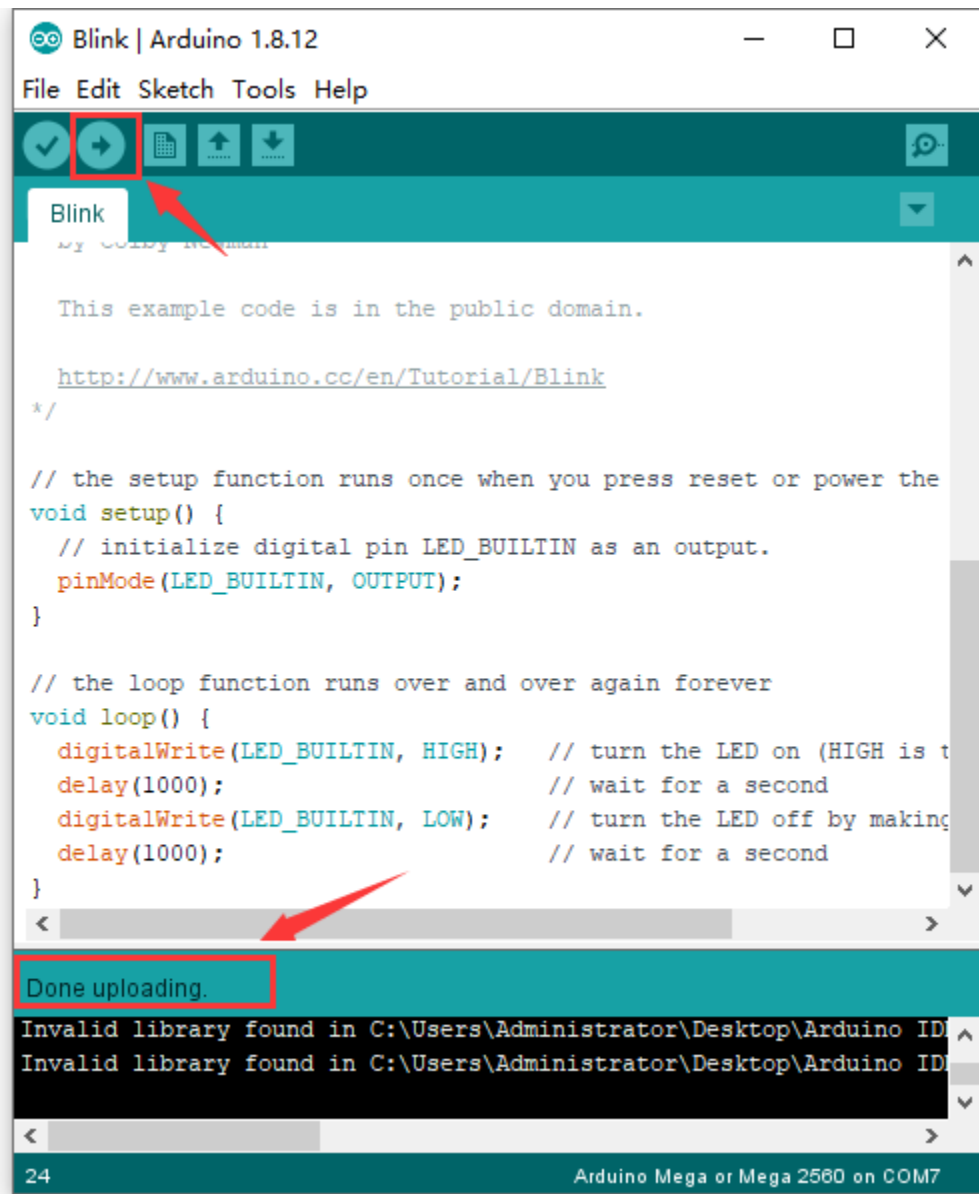
Set board and COM port, the corresponding board and COM port are shown on the lower right of IDE.



Click  to start compiling the program, and check errors.



Click  to upload the program



After the code is uploaded, the onboard LED will blink.

MAC System

### 5.1 Install Arduino IDE on MAC System

The installation instruction is as same as that in the chapter 4.1, as shown below:

## ARDUINO 1.8.12

Arduino IDE that can be used with any Arduino board, including the Arduino Yún and Arduino DUE. Refer to the [Getting Started](#) page for Installation instructions.

[See the release notes.](#)

**Windows** [Installer](#)

**Windows** [ZIP file for non admin install](#)

**Mac OS X** [10.8 Mountain Lion or newer](#)

**Linux** [32 bits](#)

**Linux** [64 bits](#)

**Linux** [ARM 32](#)

**Linux** [ARM 64](#)

**Source**

5.2 Download the Driver of CP2102

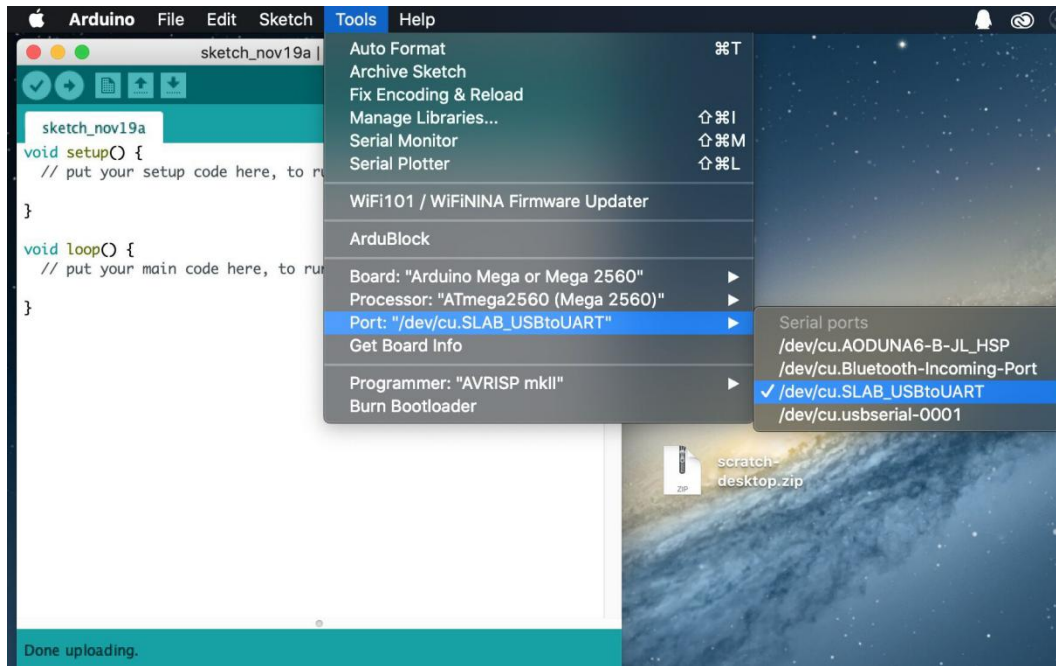
<https://fs.keyestudio.com/CP2102-MAC>

5.3 How to Install the Driver of CP2102

[https://wiki.keyestudio.com/How\\_to\\_Install\\_the\\_Driver\\_of\\_CP2102\\_on\\_MAC\\_System](https://wiki.keyestudio.com/How_to_Install_the_Driver_of_CP2102_on_MAC_System)

5.4 Arduino IDE Setting

The setting method is as same as the chapter 4.3 except from COM port, as shown below.



### Resources

<https://fs.keyestudio.com/KS0498>

### KS0499 Keyestudio Mega Plus development board

#### 1. Description

KEYESTUDIO Mega plus Board, whose processor core is ATMEGA2560-16AU, is fully compatible with ARDUINO Mega REV3.

USB to TTL chip adopts more economic and stable CP2012.

This plus board consists of 54-channel digital input and output ports, of which 15 pins are served as PWM output, 16 analog inputs, 4 serial communication ports, one 16MHz crystal oscillator, 1 USB port, 1 power socket, 1 ICSP interface and 1 reset button.

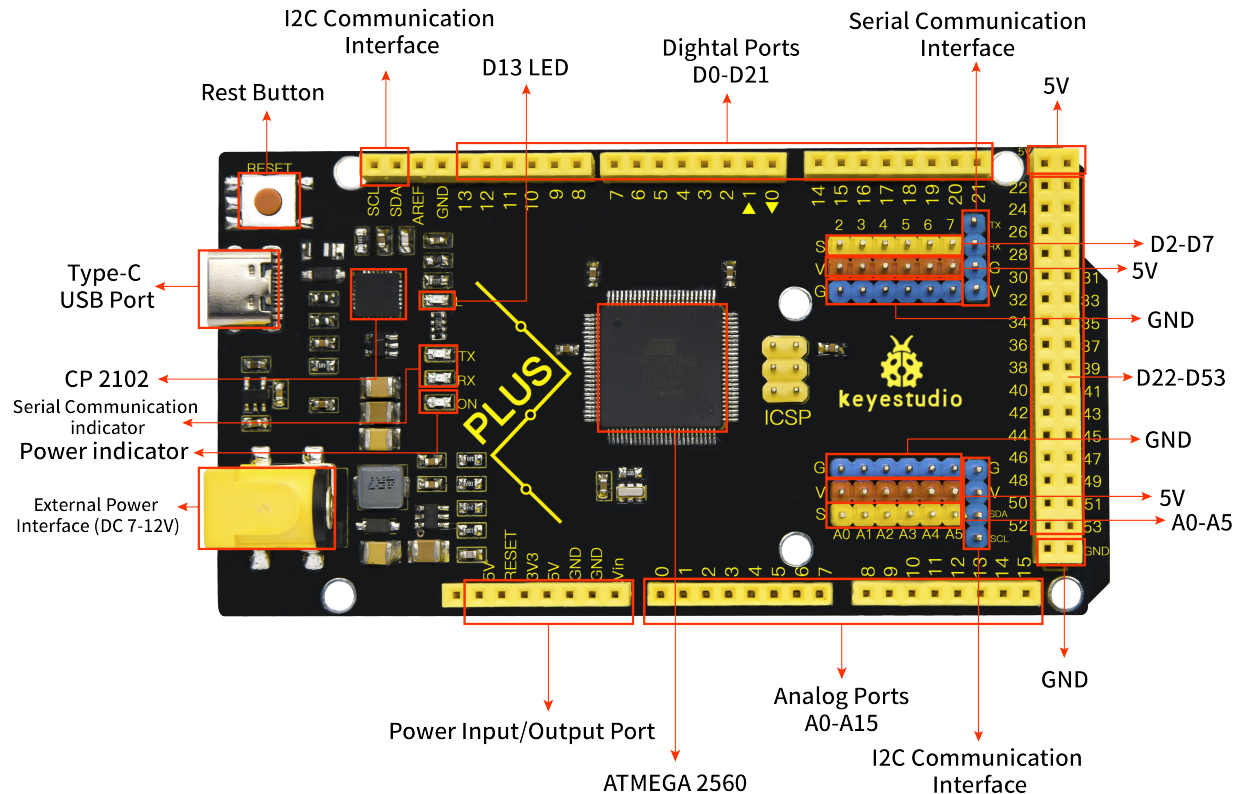
#### 1. Specification

- Microcontroller: ATMEGA2560-16AU
- USB to TTL chip CP2012
- Operating Voltage: 5V
- Input Voltage (recommended): DC 7-12V
- Digital I/O Pins: 54 (D0-D53)
- PWM Digital I/O Pins: 15 (D2-D13, D44-D46)
- Analog Input Pins: 16 (A0-A15)
- DC Current per I/O Pin: 20 mA
- DC Current for 3.3V Pin: 50 mA
- Flash Memory: 256 KB of which 8 KB used by bootloader
- SRAM: 8 KB
- EEPROM: 4 KB



- Clock Speed: 16 MHz
- LED\_BUILTIN:D13

#### Interfaces Description



#### Special Interfaces Description

Serial communication interface(4 channel): Serial0 =RX0, D1 =TX0, Serial1D19 is RX1, D18 is TX1)

Serial2 D17 is RX2, D16 equals to TX2, Serial3D15 is RX3, D14 is TX3), D0 and D1 are connected to ATMEGA16U2-MU

PWM port(Pulse width modulation): D2-D13 and D44-D46

External interrupt pinsD2interrupt 0), D3(interrupt 1), D21interrupt 2), D20 (interrupt 3), D19(interrupt 4and D18interrupt 5

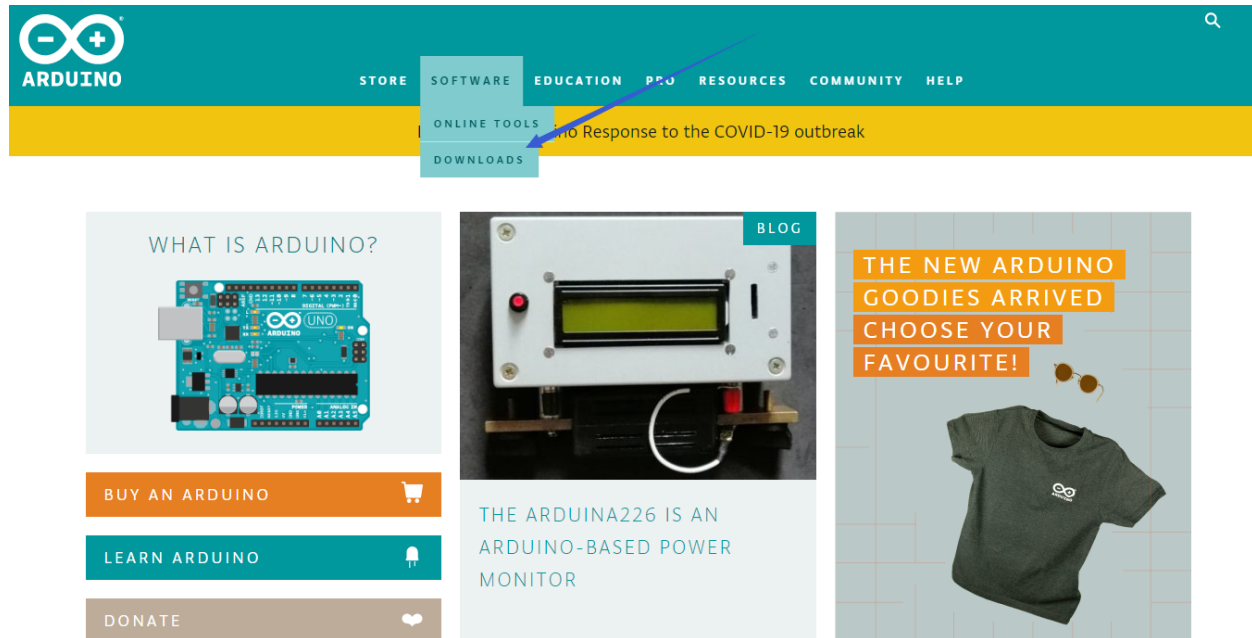
SPI communication interface D53 stands for SS, D51 is MOSI, D50 is MISO, D52 equals to SCK

IIC communication interface D20 represents SDA, D21 is SCL

Install Arduino IDE and Driver

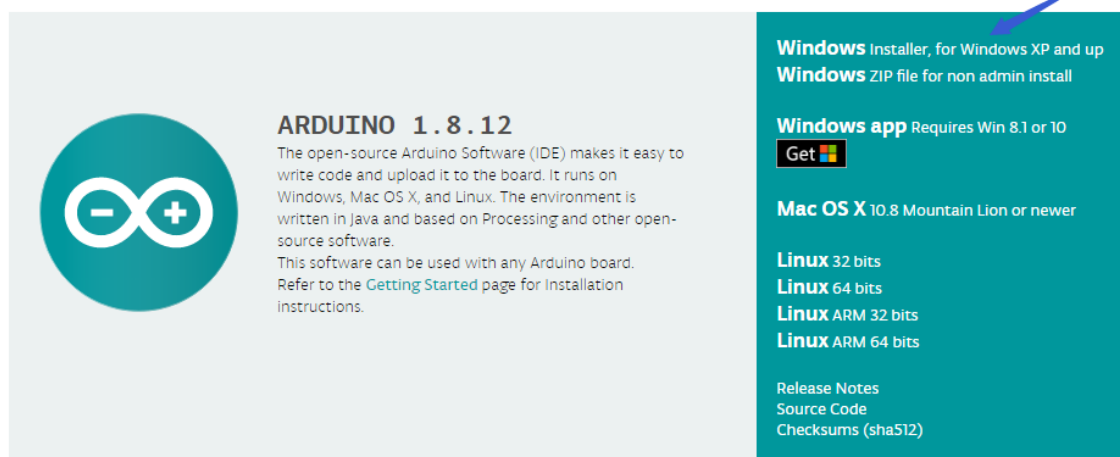
### 5.1 Installing Arduino IDE

We firstly navigate the ARDUINO official website: <https://www.arduino.cc/>, click “SOFTWARE”—“DOWNLOADS”, we’ll see the following page.



There are many versions for Arduino software but you only need to download the version suitable for your system. Here, we take WINDOWS system as an example to introduce how to download and install Arduino IDE.

## Download the Arduino IDE



There are two versions for your choice, one is an installer which can be directly downloaded in computer and the other is zip file which requires you to unzip and install it.

Consider supporting the Arduino Software by contributing to its development. (US tax payers, please note this contribution is not tax deductible). [Learn more on how your contribution will be used.](#)



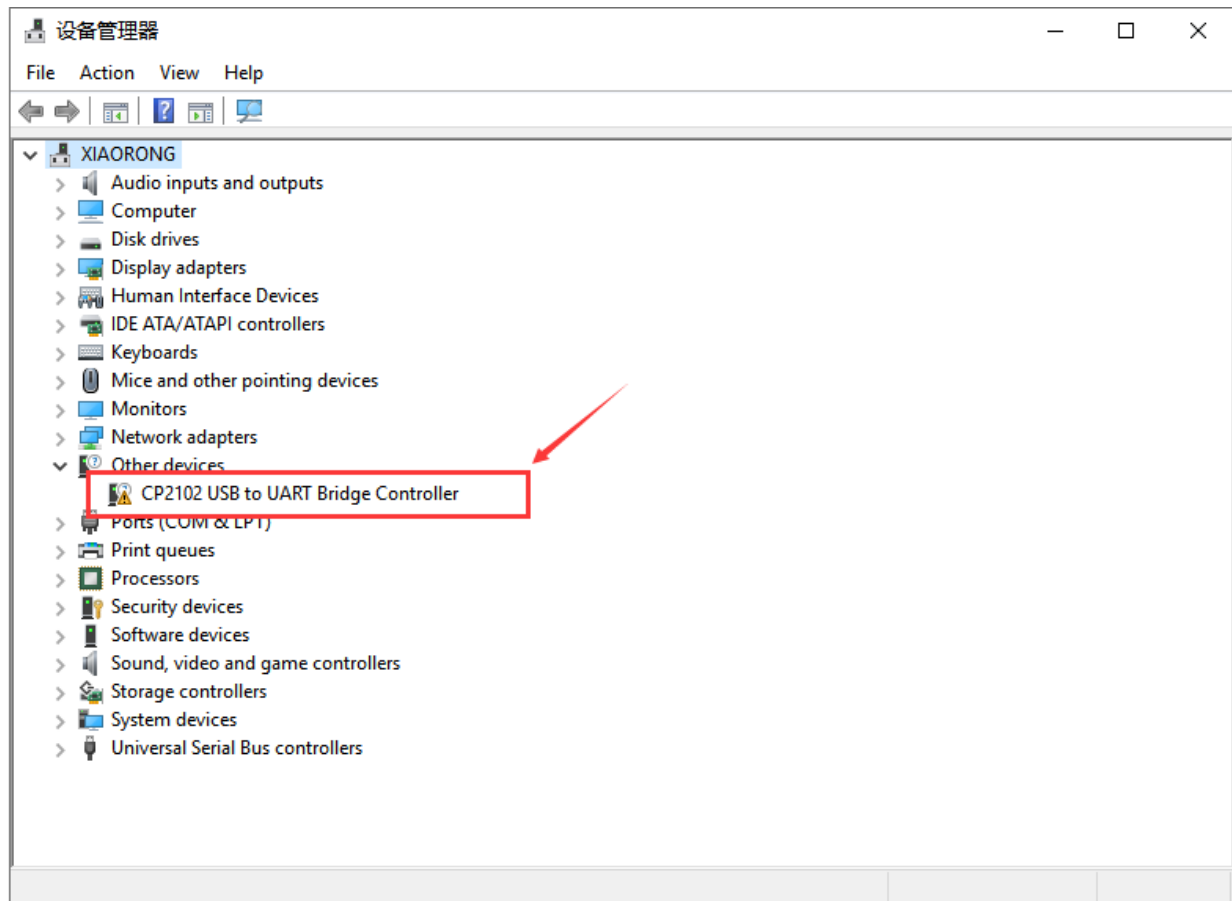
Just click“JUST DOWNLOAD”.

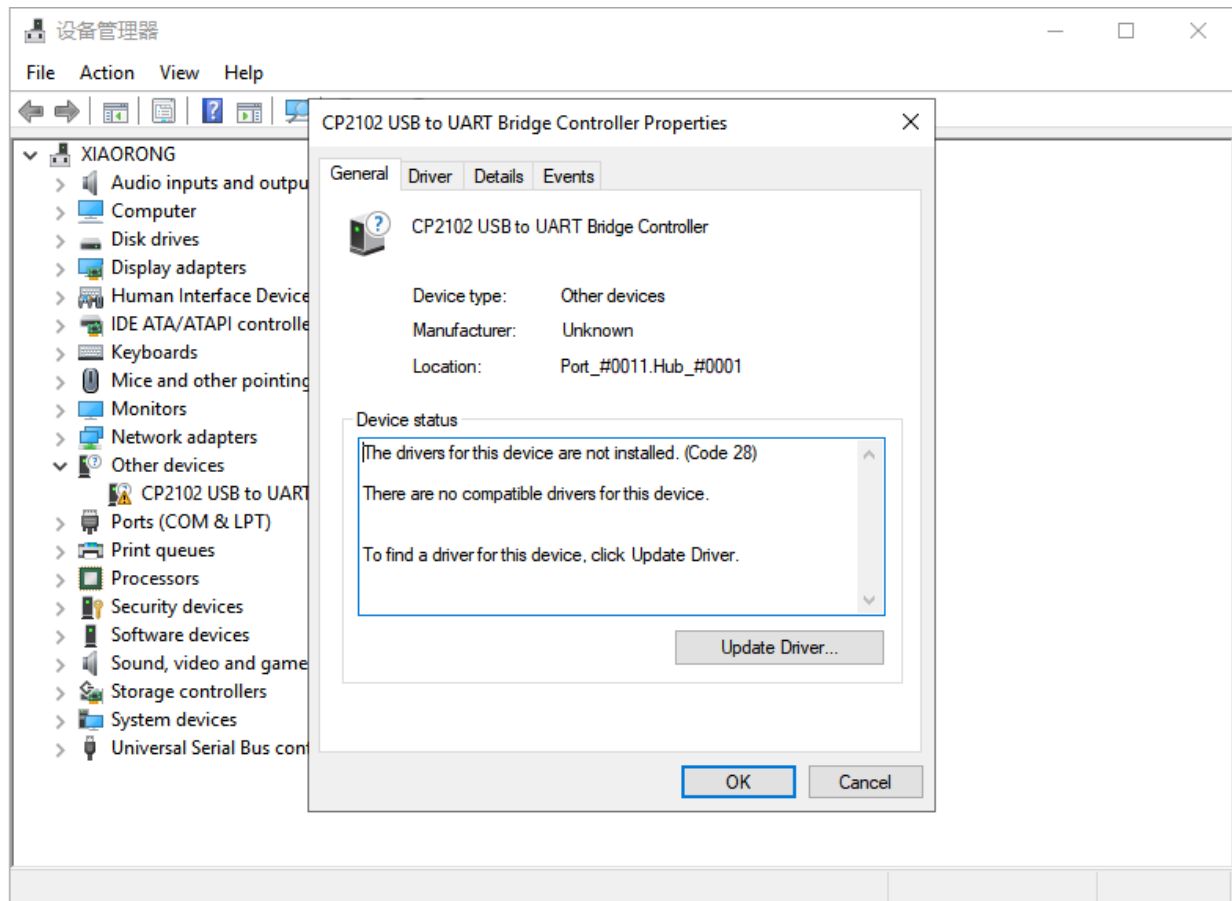
## 5.2 Installing Driver

The downloading process is finished. Now, let's install the driver of KEYESTUDIO PLUS control board. Its chip is CP2102 serial chip. In general, the driver of CP2102 serial chip is included in ARDUINO version 1.8 and above. The driver will be recognized and installed as long as you connect board to computer with USB.

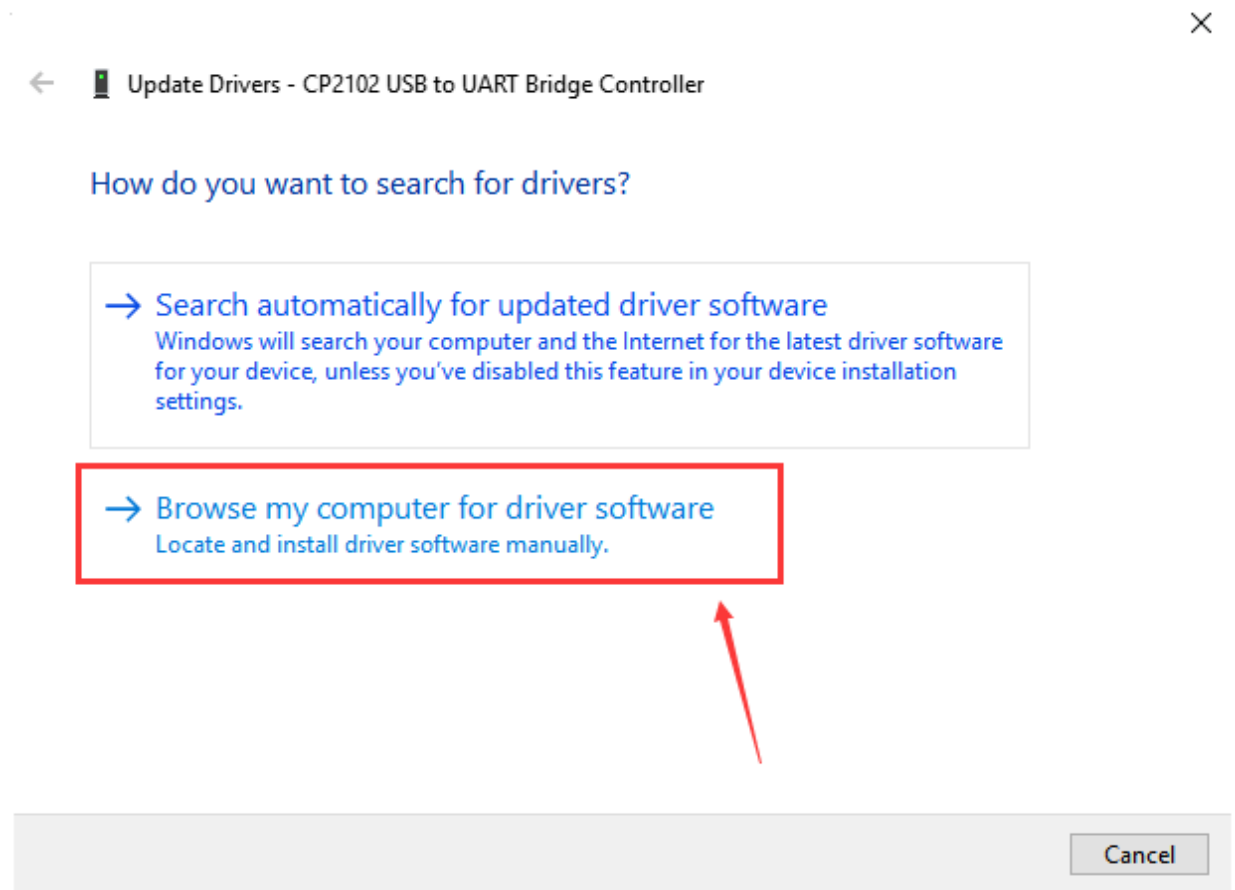
Some system can't install automatically driver but you could install it by hand. Open the device manager of computer and you will see a yellow exclamation mark which means the driver of CP2102 isn't installed successfully.

Double-click the“Cp2102 USB to UART Bridge Controller”and a window pops up, click“update driver”, as shown below:

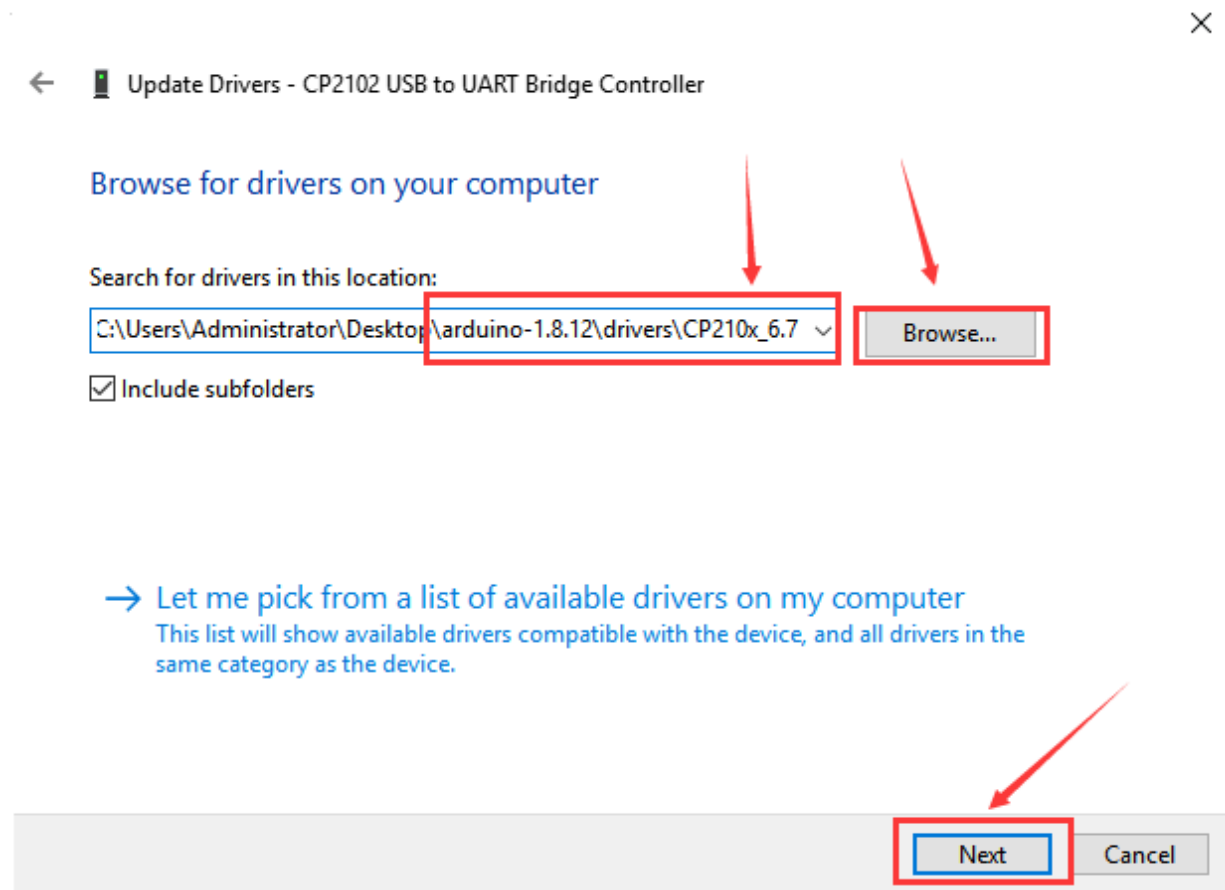




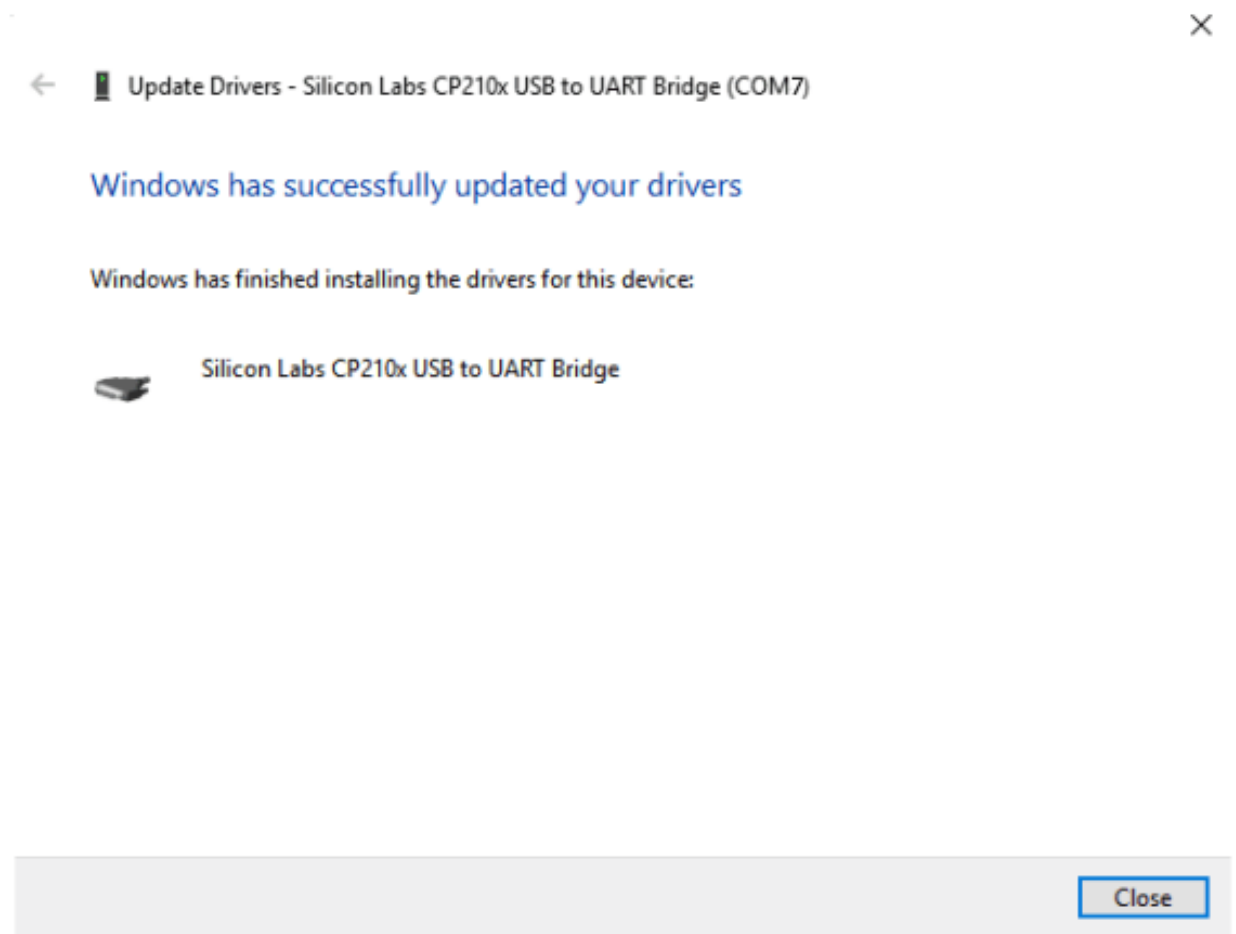
Click "Browse my computer for driver software" to find out the ARDUINO software to be installed.



There is a **DRIVERS** folder inside, enter it and the driver of CP210X series chip is shown.



Choose the folder to be installed, click "Next", the driver of CP21010 is installed successfully.

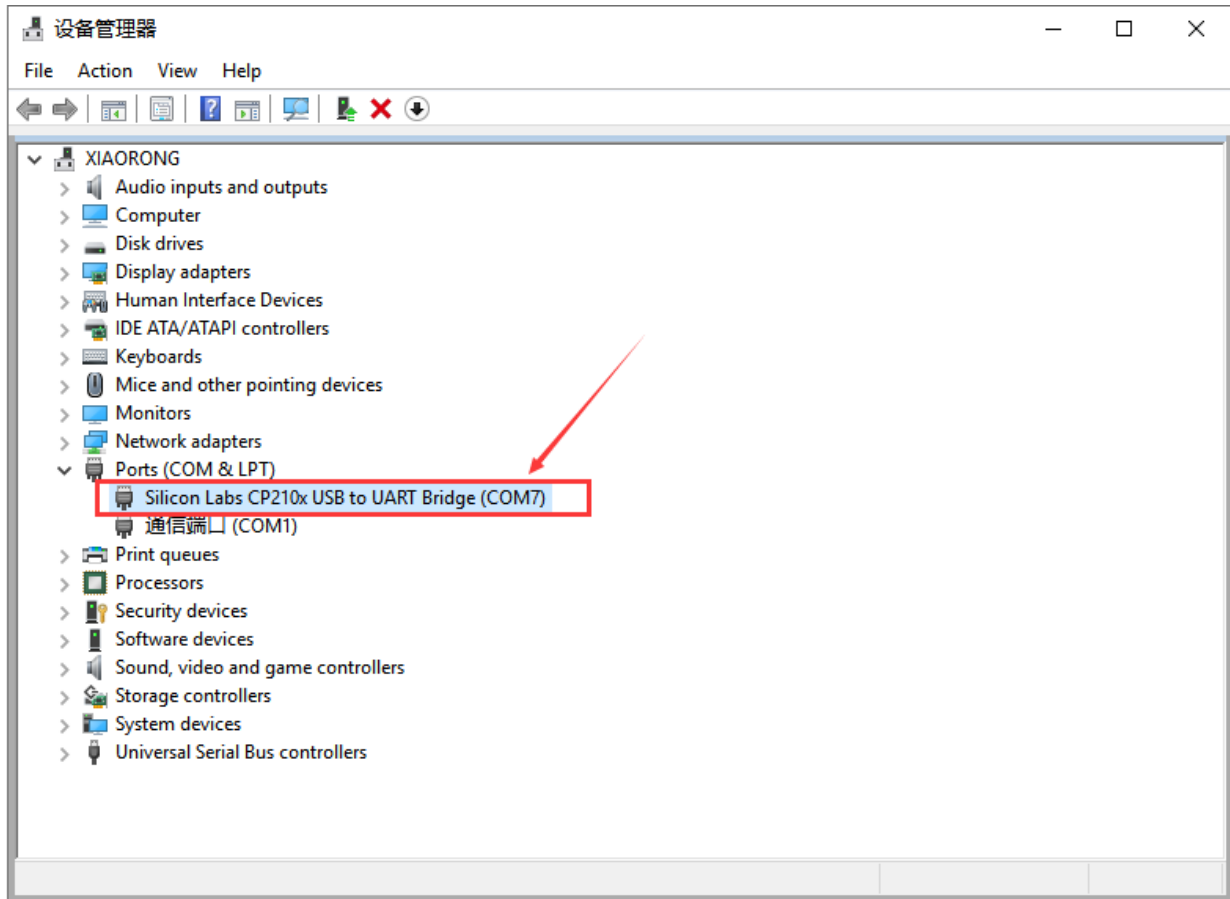


Up to now. Then you can right click “Computer” —> “Properties”

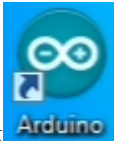
—> “Device manager”, you will see the device as the figure shown below.

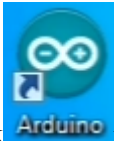
The driver is installed successfully and the yellow exclamation mark is gone.

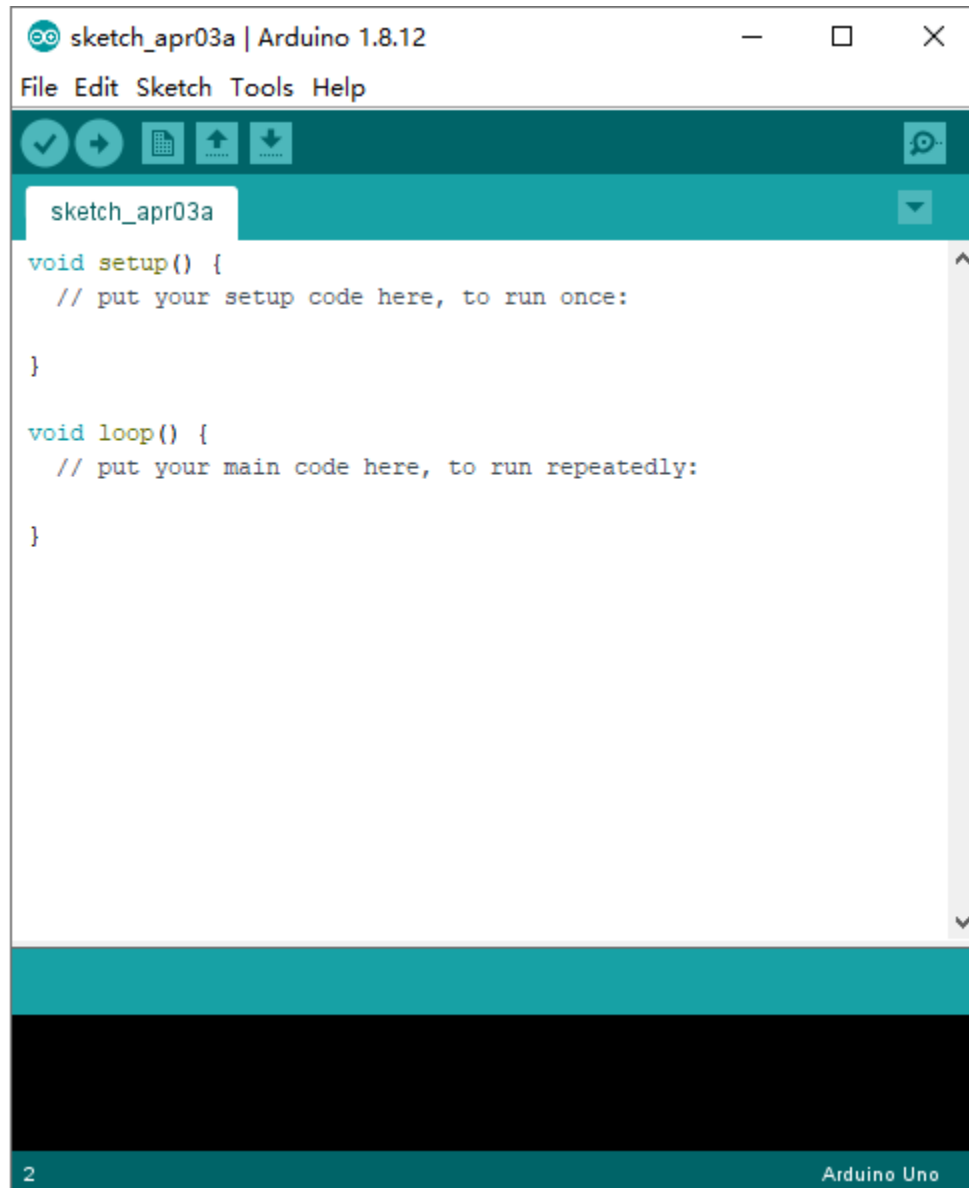




### 5.3 Arduino IDE Setting

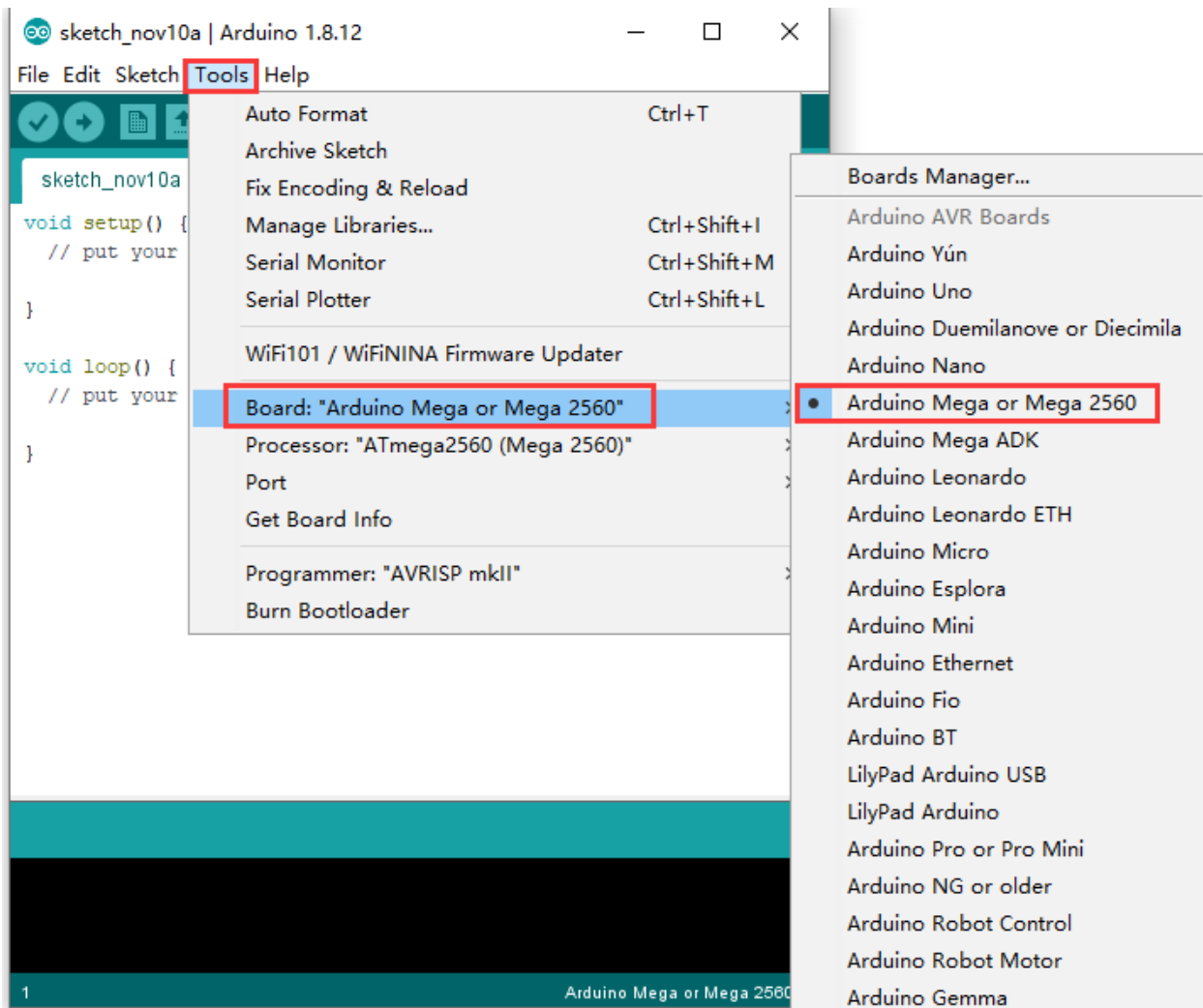


Click  icon and open Arduino IDE.

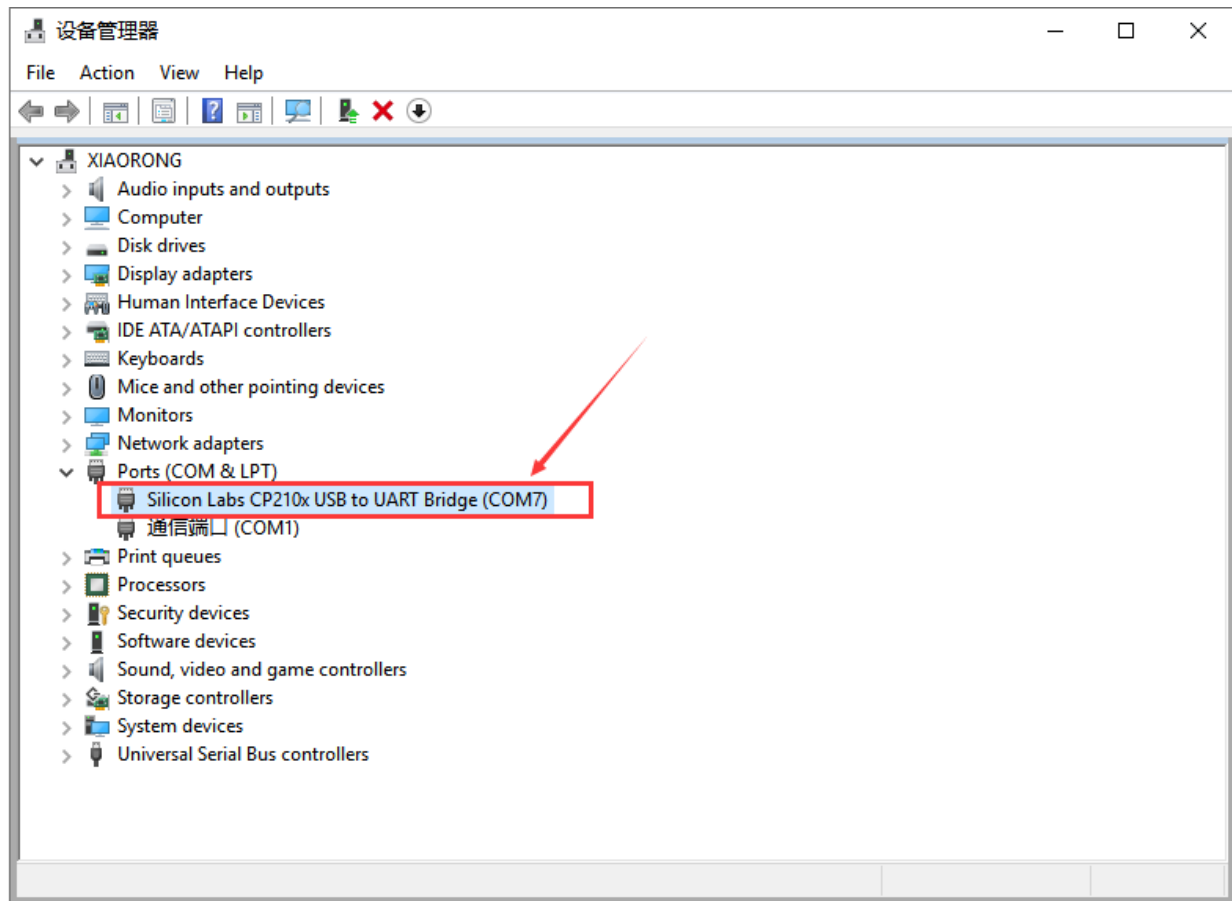


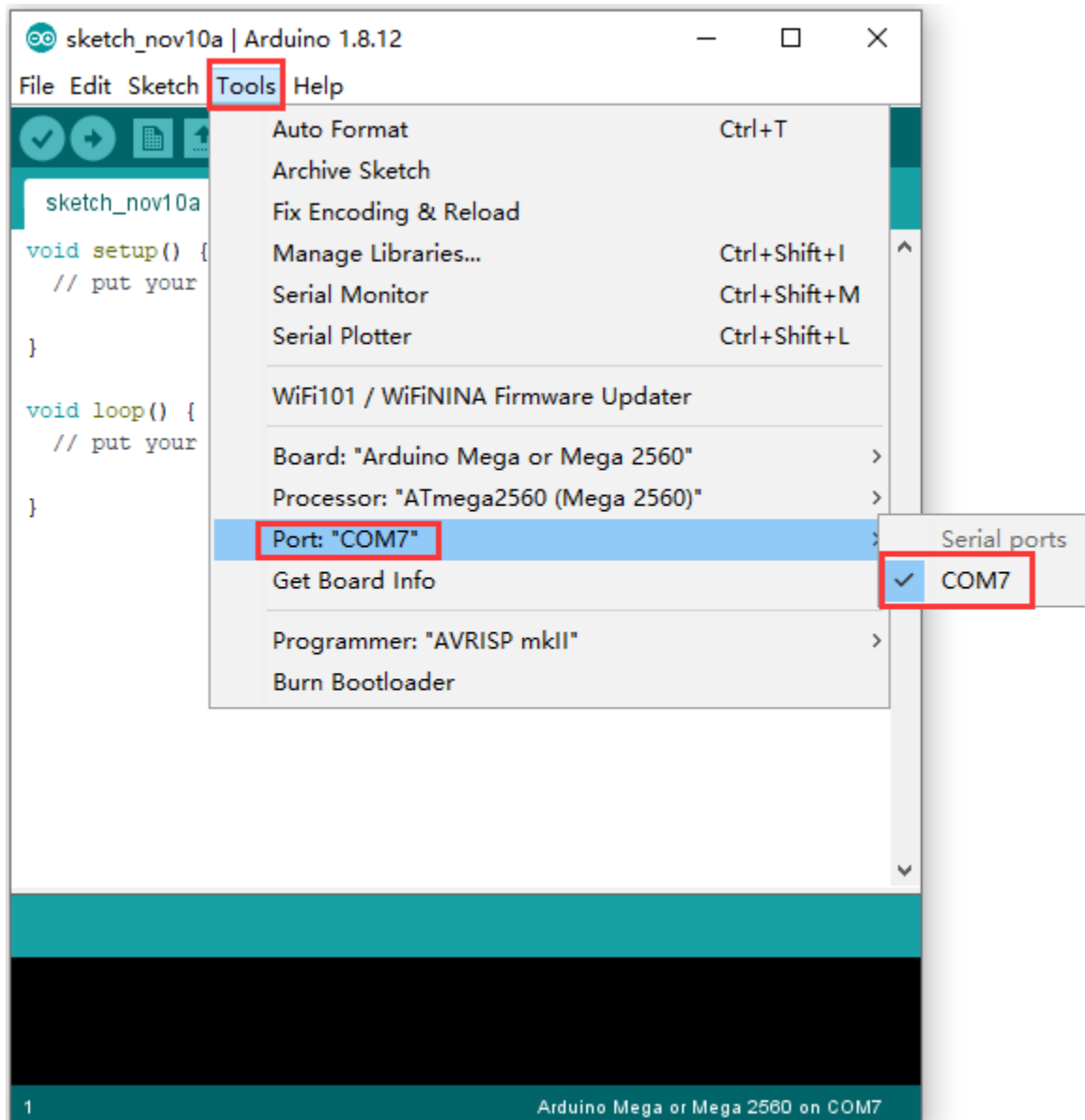
To avoid the errors when uploading the program to the board, you need to select the correct Arduino board that matches the board connected to your computer.

Then come back to the Arduino software, you should click Tools→Board, select the board. (as shown below)

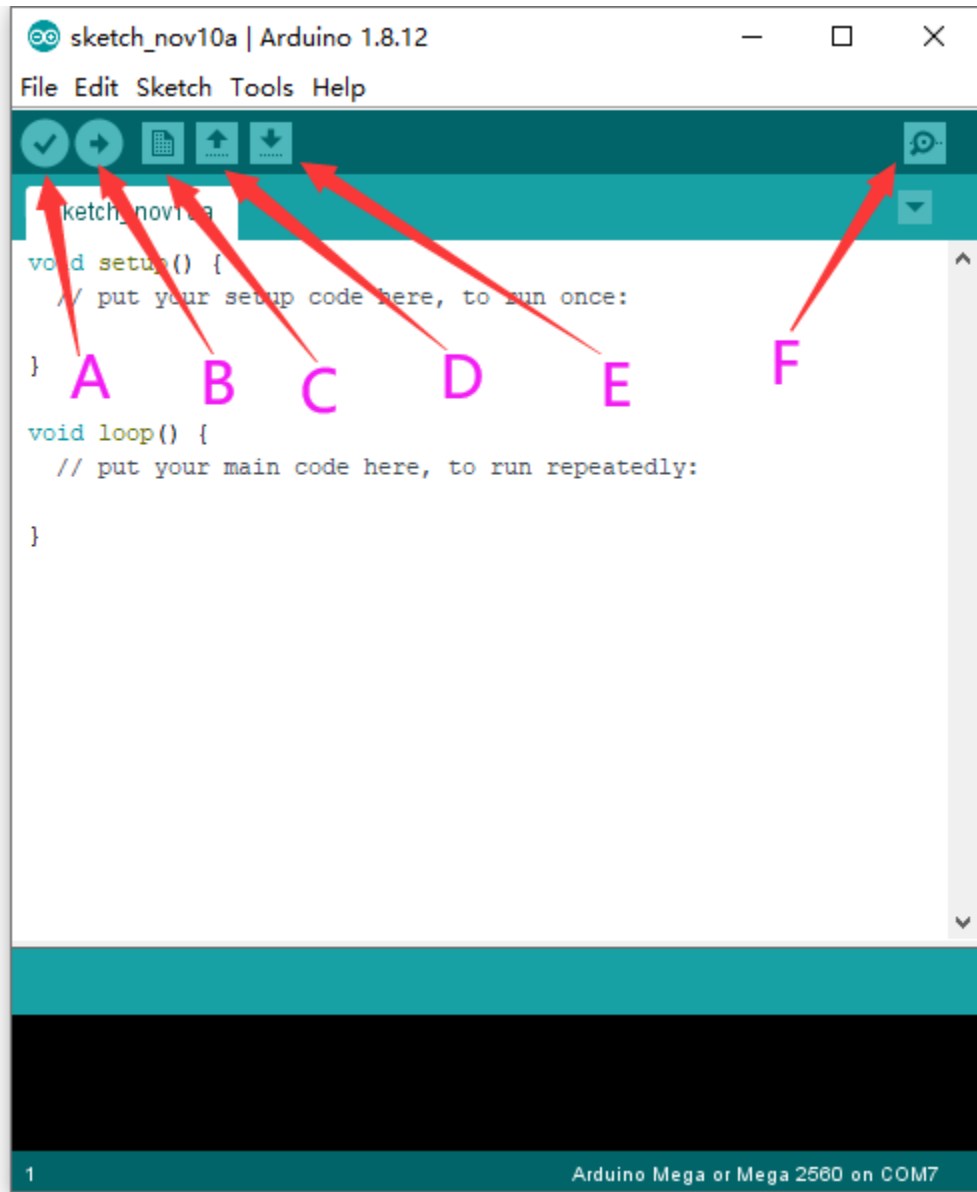


Then select the correct COM port (you can see the corresponding COM port after the driver is successfully installed)





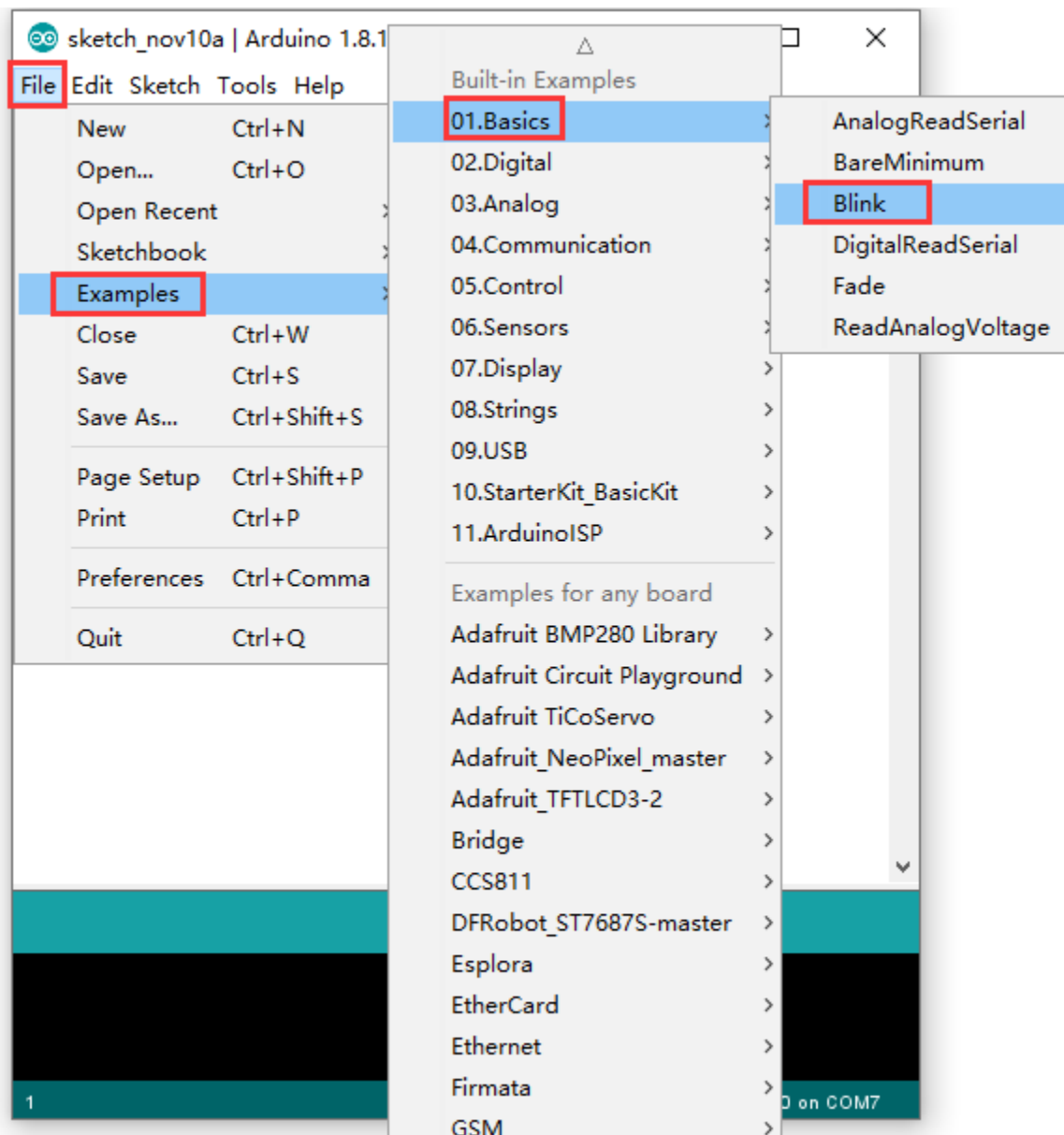
Before uploading the program to the board, let's demonstrate the function of each symbol in the Arduino IDE toolbar.

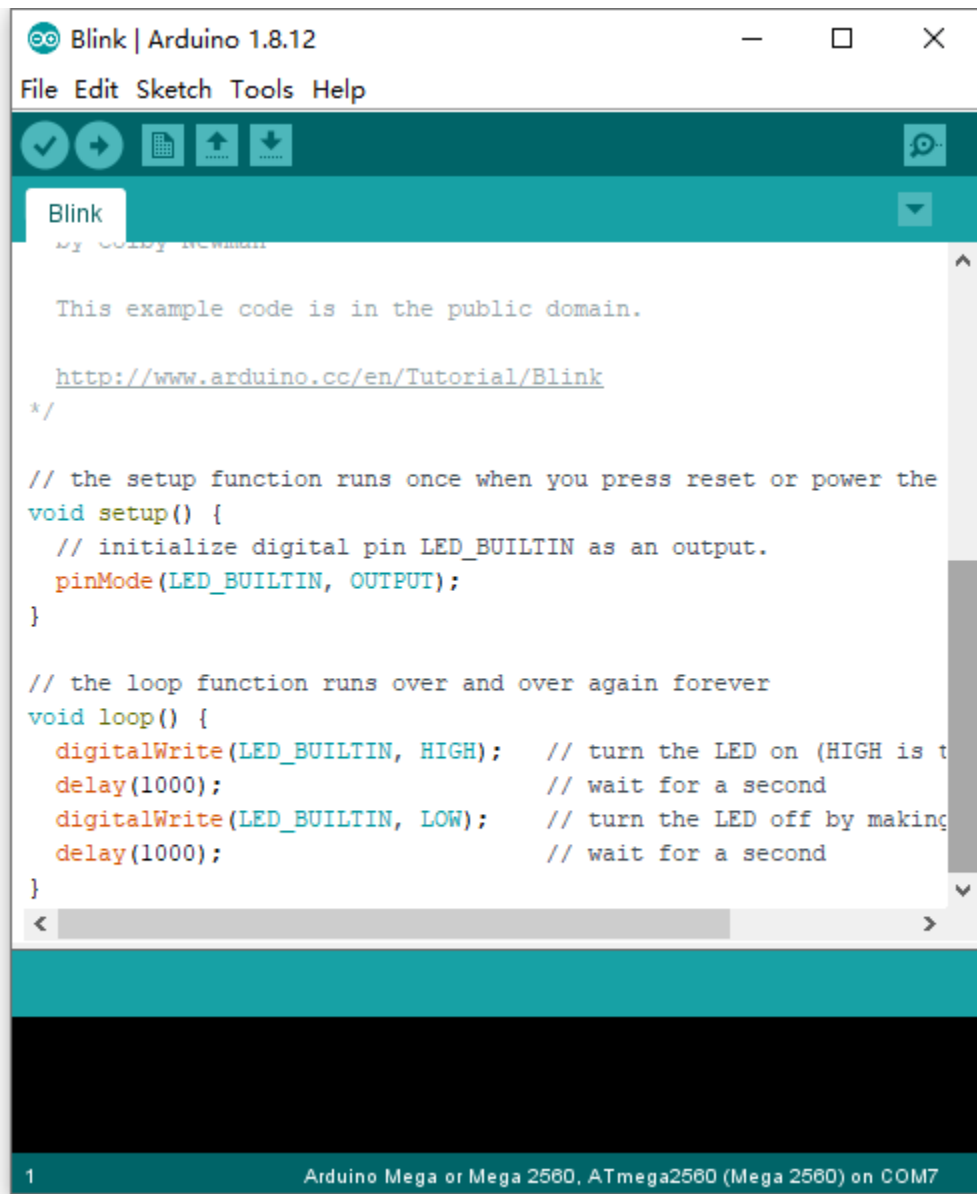


1. Used to verify whether there is any compiling mistakes or not.
- B- Used to upload the sketch to your Arduino board.
- C- Used to create shortcut window of a new sketch.
- D- Used to directly open an example sketch.
- E- Used to save the sketch.
- F- Used to send the serial data received from board to the serial monitor.

### 5.4 Start Your First Program

Open file to choose Examples—>BASIC—>BLINK




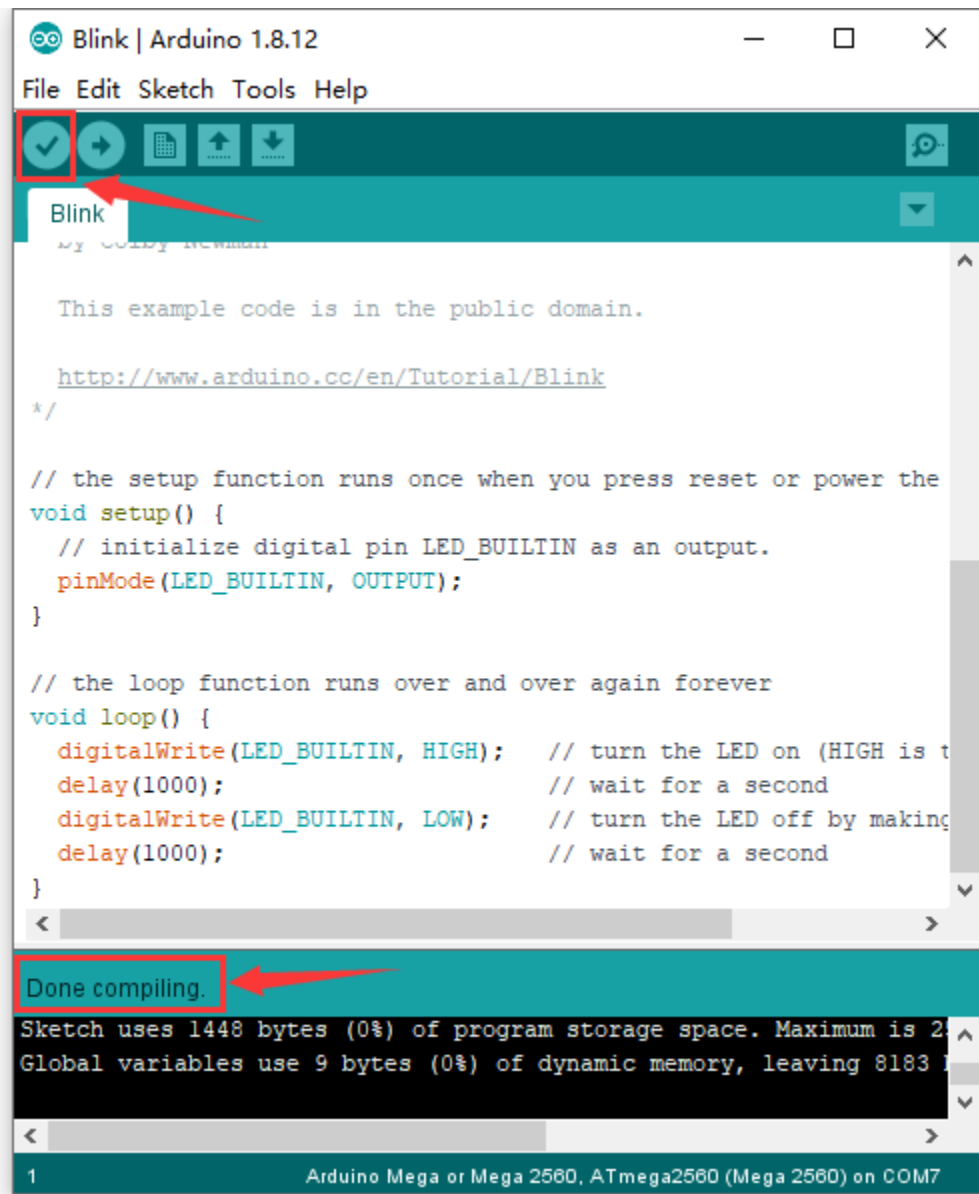



Set board and COM port, the bottom right corner of the IDE displays the corresponding board and COM port.

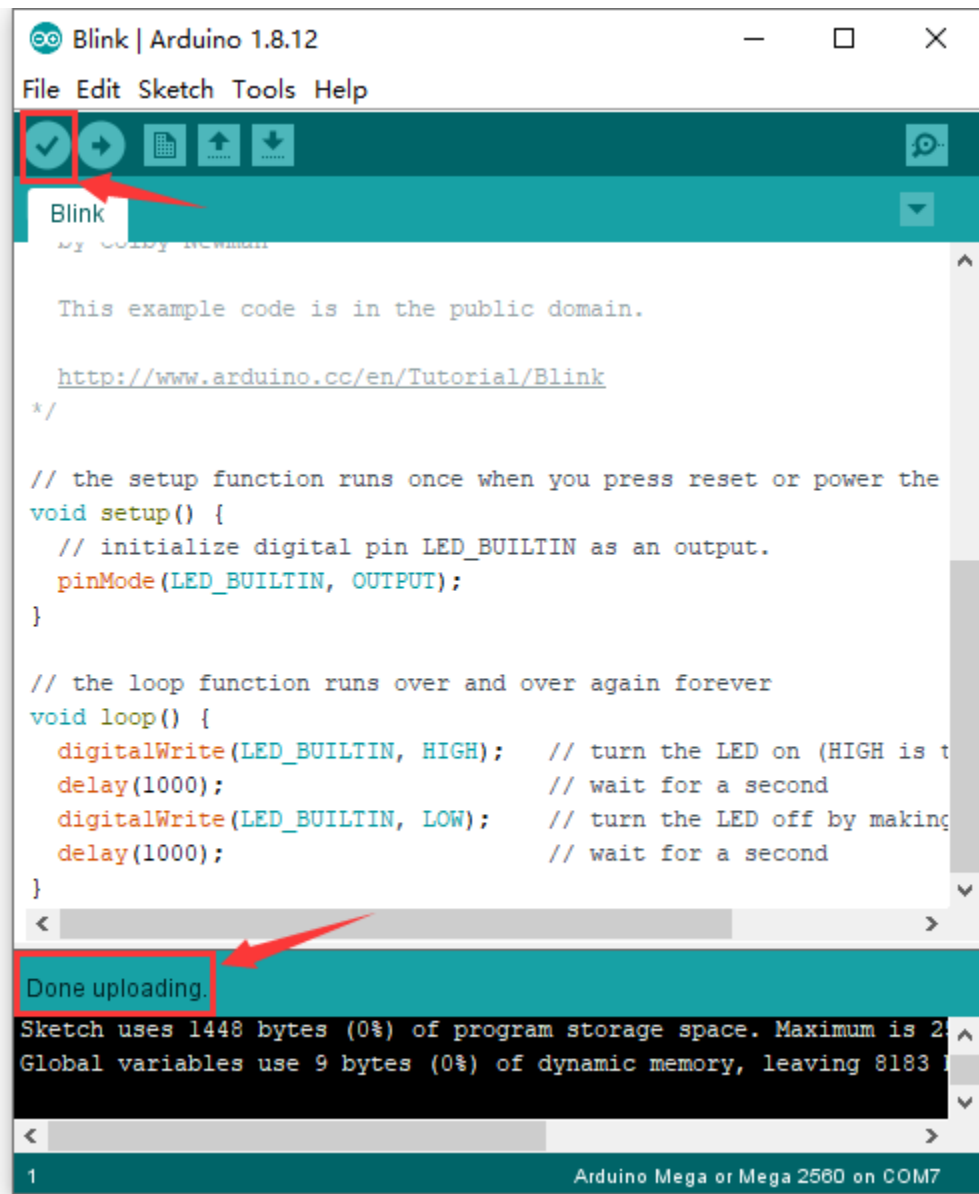




Click  icon to start compiling the program, and check errors.

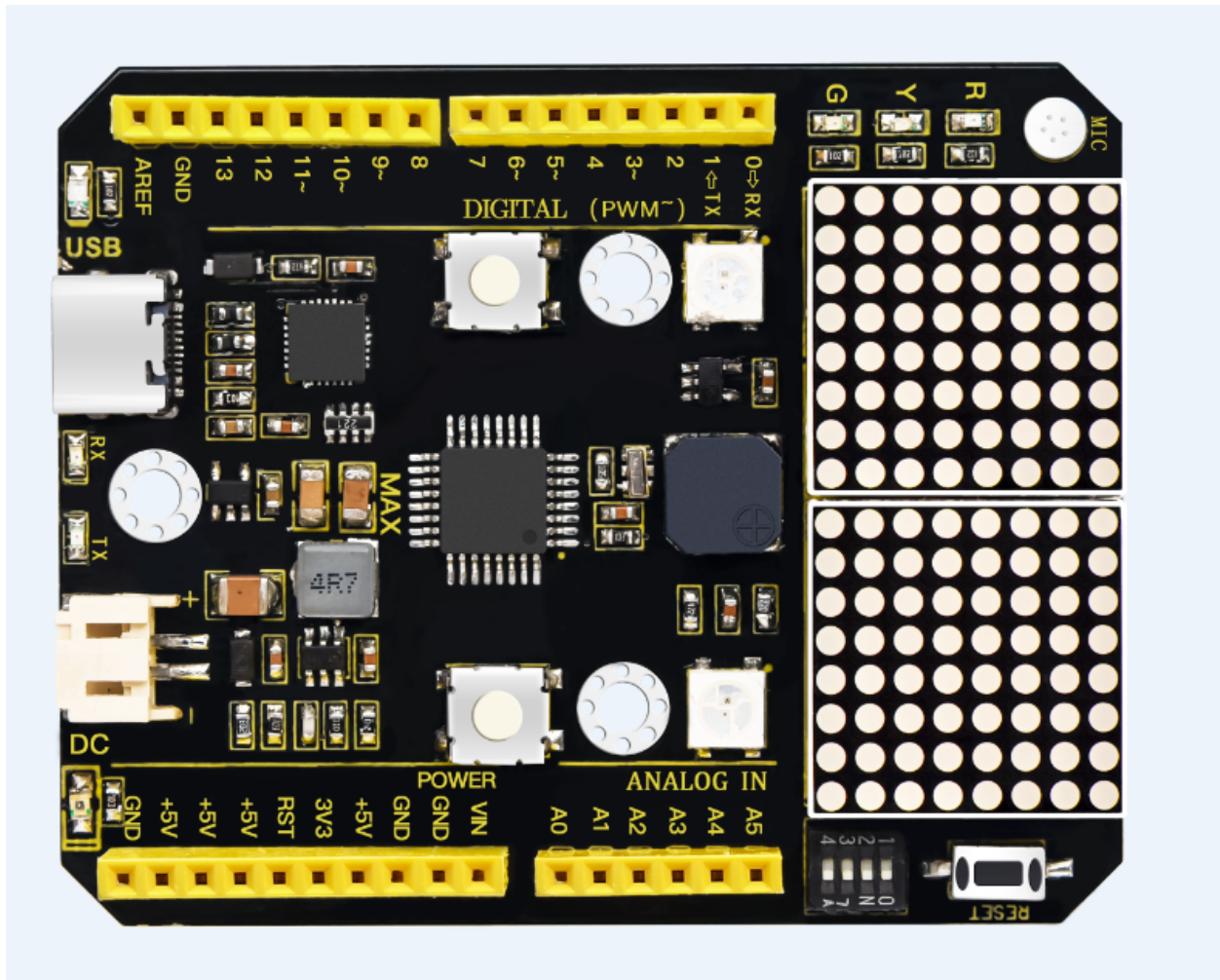


Click  icon to start upload the program, upload successfully.



The program is uploaded successfully, LED of board lights on for 1s, and lights off for 1s.

**KS0500 Keyestudio Max Development Board**



## 1.Introduction:

Keyestudio Max development board is an Arduino uno-compatible board, which is based on ATmega328P MCU, and with a cp2102 Chip as a UART-to-USB converter. MAX development board integrates three LEDs, two 6812RGB LEDs, two button switches, a buzzer, a microphone sensor, a light sensor, and an 8\*16 dot matrix.

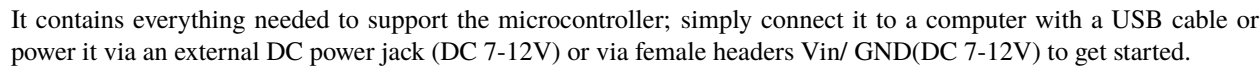
Using the onboard hardware, you can start 20 project courses. If you are a beginner, a MAX board is enough!

MAX can also be connected to all Arduino sensors and modules, and it is also compatible with all Arduino shield boards.

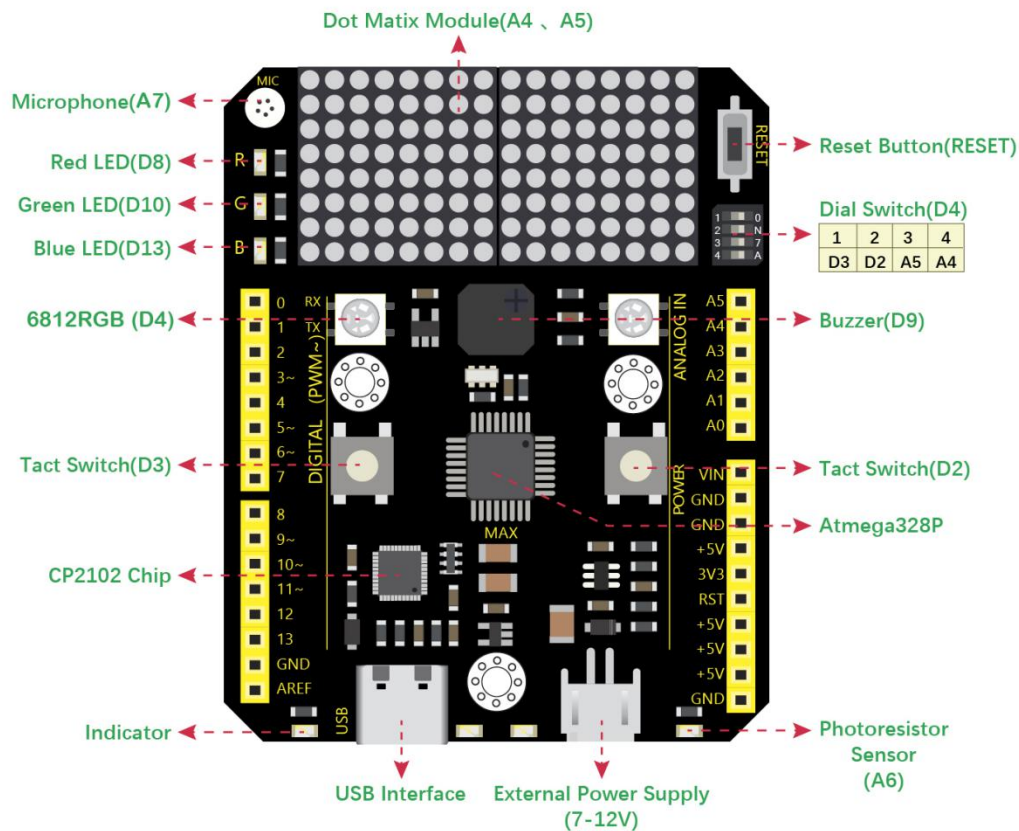
Now, use MAX to enter the magical world of ARDUINO!

## 2.Max development board Pinout

Max has 14 digital input/output pins (of which 6 can be used as PWM outputs), 6 analog inputs, a 16 MHz quartz crystal, a USB connection, a power jack, 2 ICSP headers and a reset button.



### 3.Max development board onboard interface



red LED: D8

yellow LED: D10

green LED: D13

6812 RGB LED: D4

two button switches: D2, D3

buzzer: D9

microphone sensor: A7

light sensor: A6

8\*16 dot matrix A4, A5 (I2C interface)

4-digit DIP switch: on right, D2, D3, A4, A5, these onboard components interface is connected. on left, these interface is disconnected,

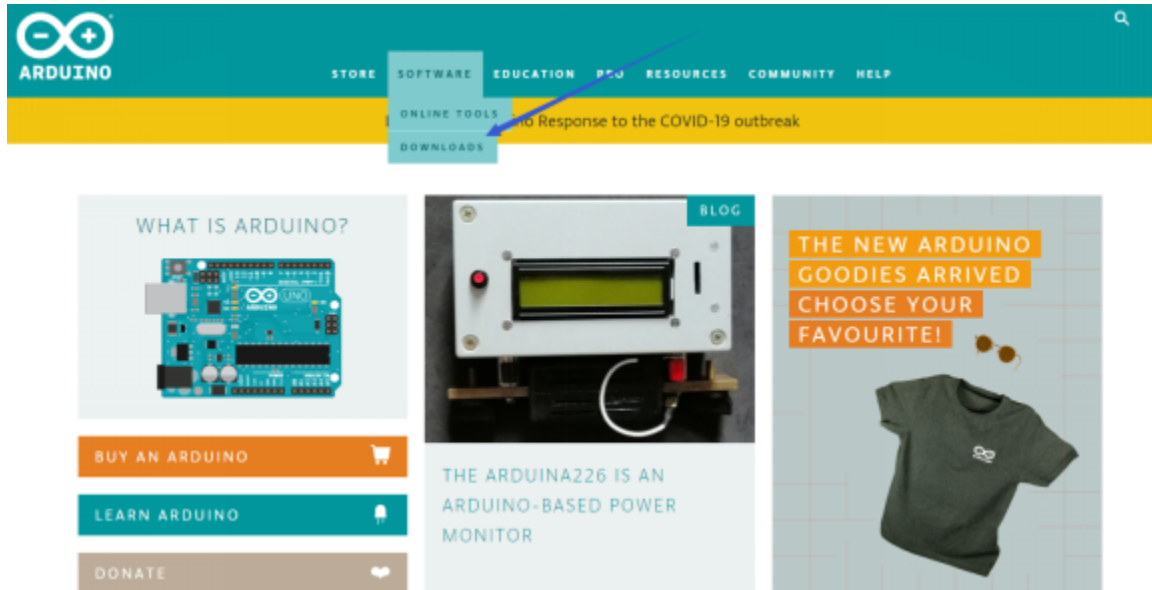
They are connected to the corresponding interface on the max board pin header.

#### 4. Install Arduino software and Driver

## 1Installing Arduino IDE

When we get max control board, we need to download Arduino IDE and driver firstly. You could download Arduino IDE from the official website:

<https://www.arduino.cc/>, click the **SOFTWARE** on the browse bar, click “**DOWNLOADS**” to enter download page, as shown below:



There are various versions Of IDE for Arduino, just download a version that compatible with your system, here we will show you how to download and install the windows version Arduino IDE.

## Download the Arduino IDE



There are two versions of IDE for WINDOWS system. 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.

Consider supporting the Arduino Software by contributing to its development. (US tax payers, please note this contribution is not tax deductible). [Learn more on how your contribution will be used.](#)



You just need to click JUST DOWNLOAD.

### (2) Installing Max board Driver

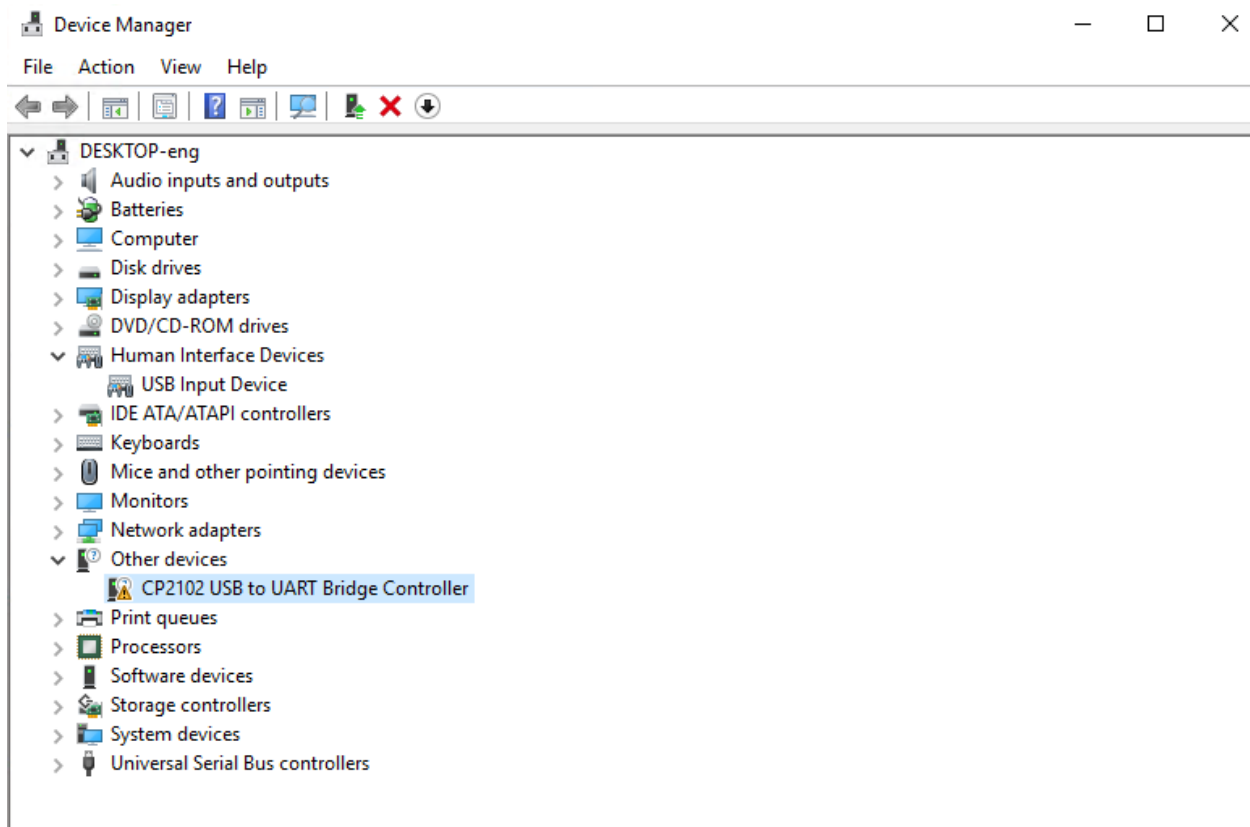
Let's install the driver of Keystudio Max board. The USB-TTL chip on Max board adopts CP2102 serial chip. The driver program of this chip is included in Arduino 1.8 version and above, which is convenient.

When you attach USB port to computer, the driver of CP2102 can be installed.

If the driver is installed unsuccessfully, you need to install it manually.

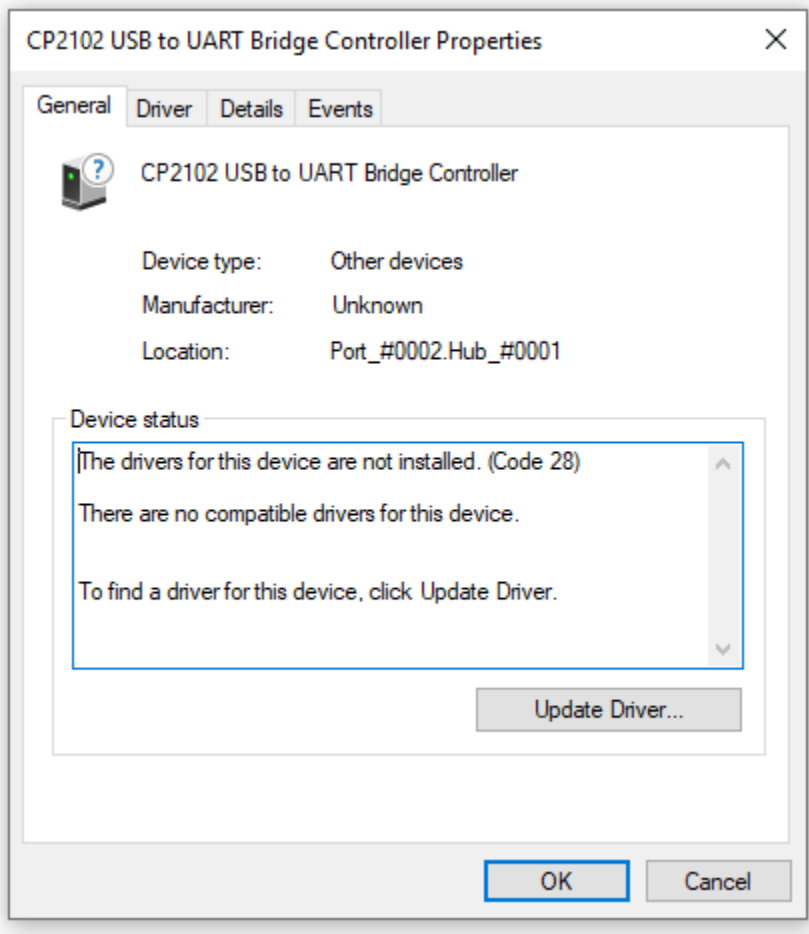
Open the device manager of computer. Right click Computer— Properties— Device Manager.



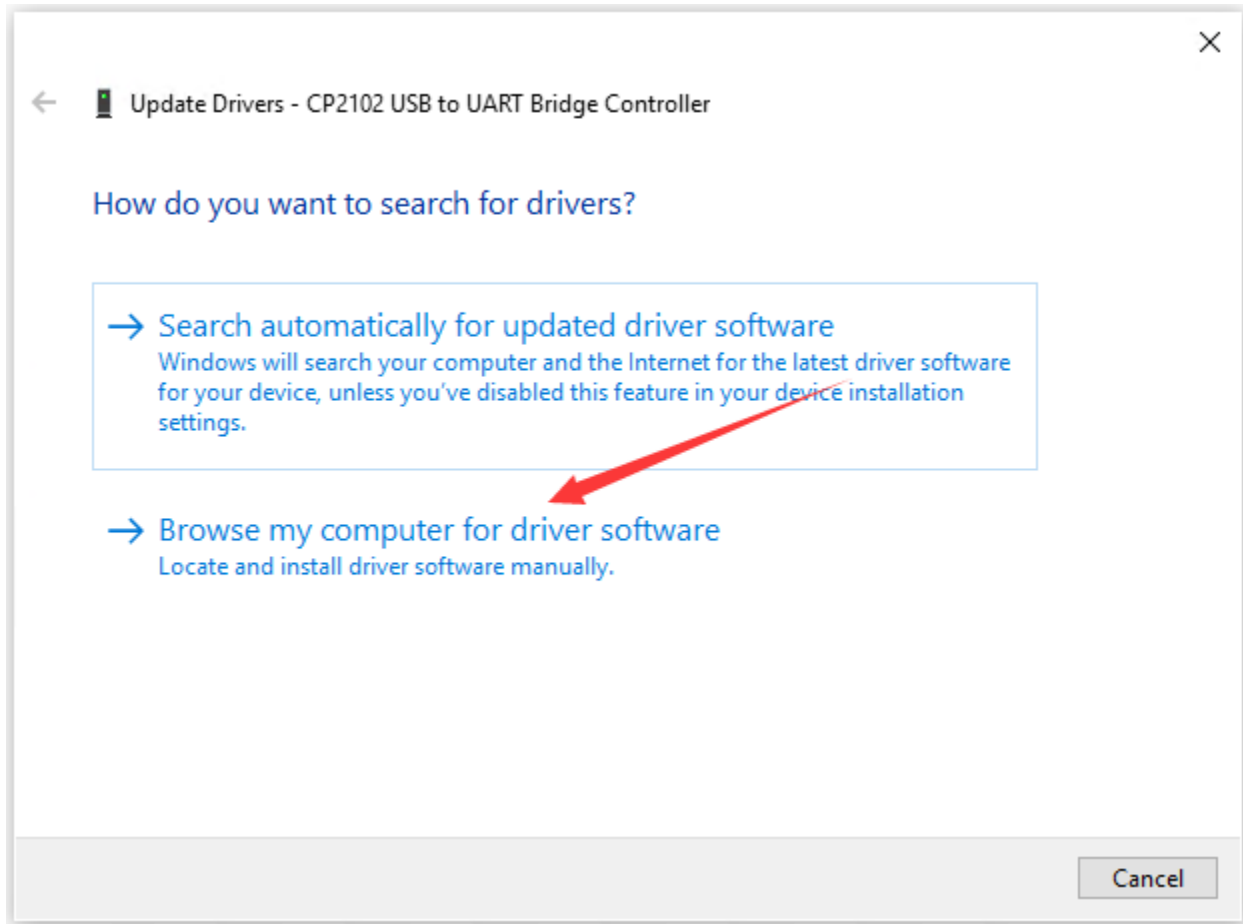


There is a yellow exclamation mark on the page, which implies the failure installation of the driver of CP2102.


Operate as follows

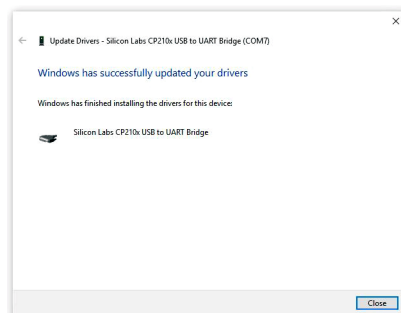
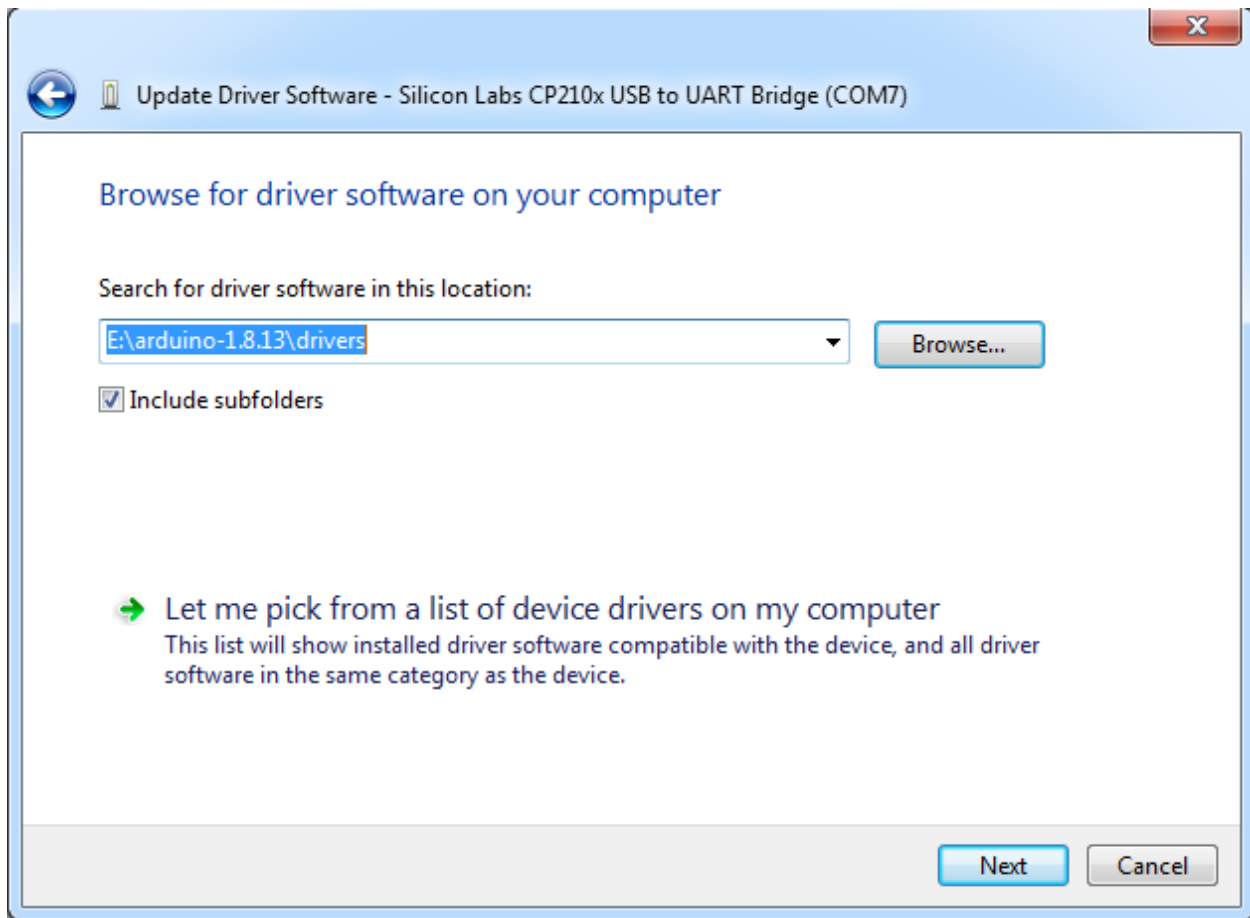


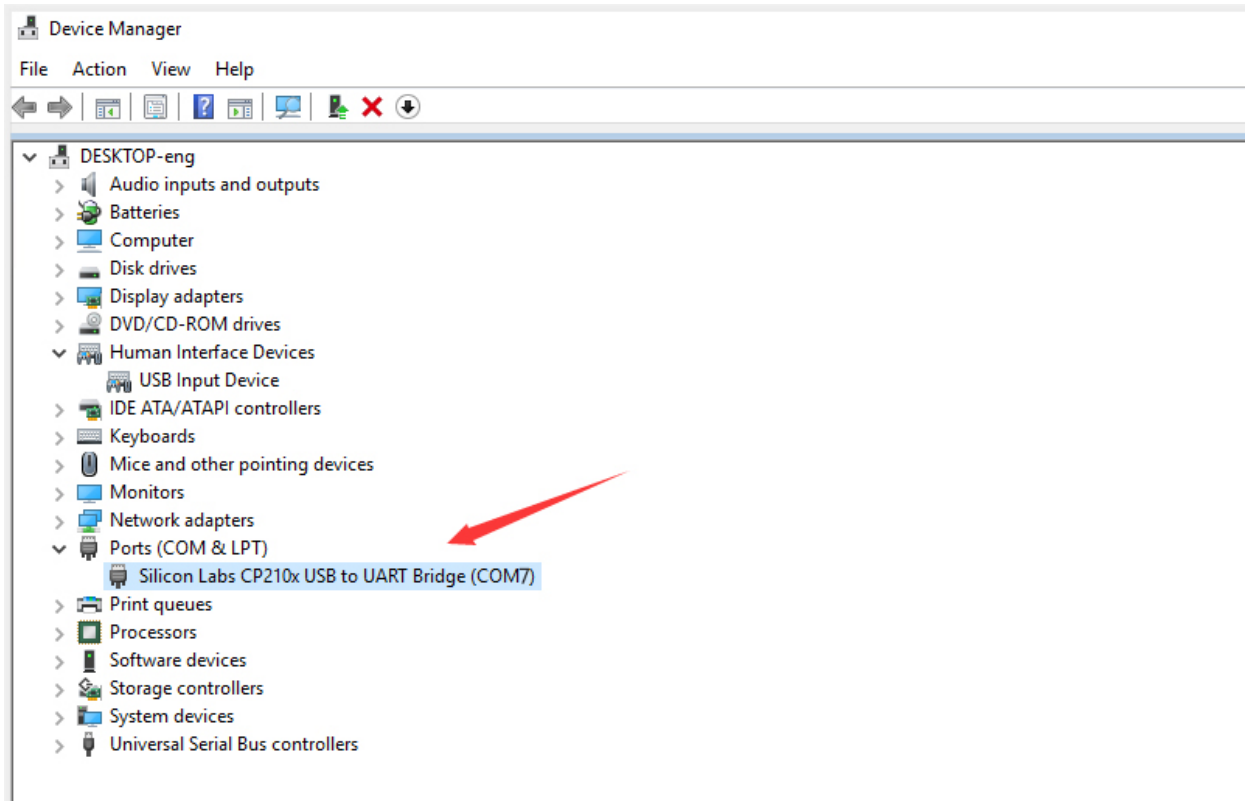
Click “OK” to enter the following page and click “browse my computer for updated driver software”.



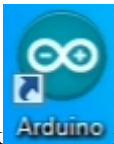
Click “Browse”, then search the driver of CP2102 and click “Next”,

There is a DRIVERS folder in Arduino software installed package  **arduino-1.8.12**, open driver folder and check the driver of CP210X series chips.

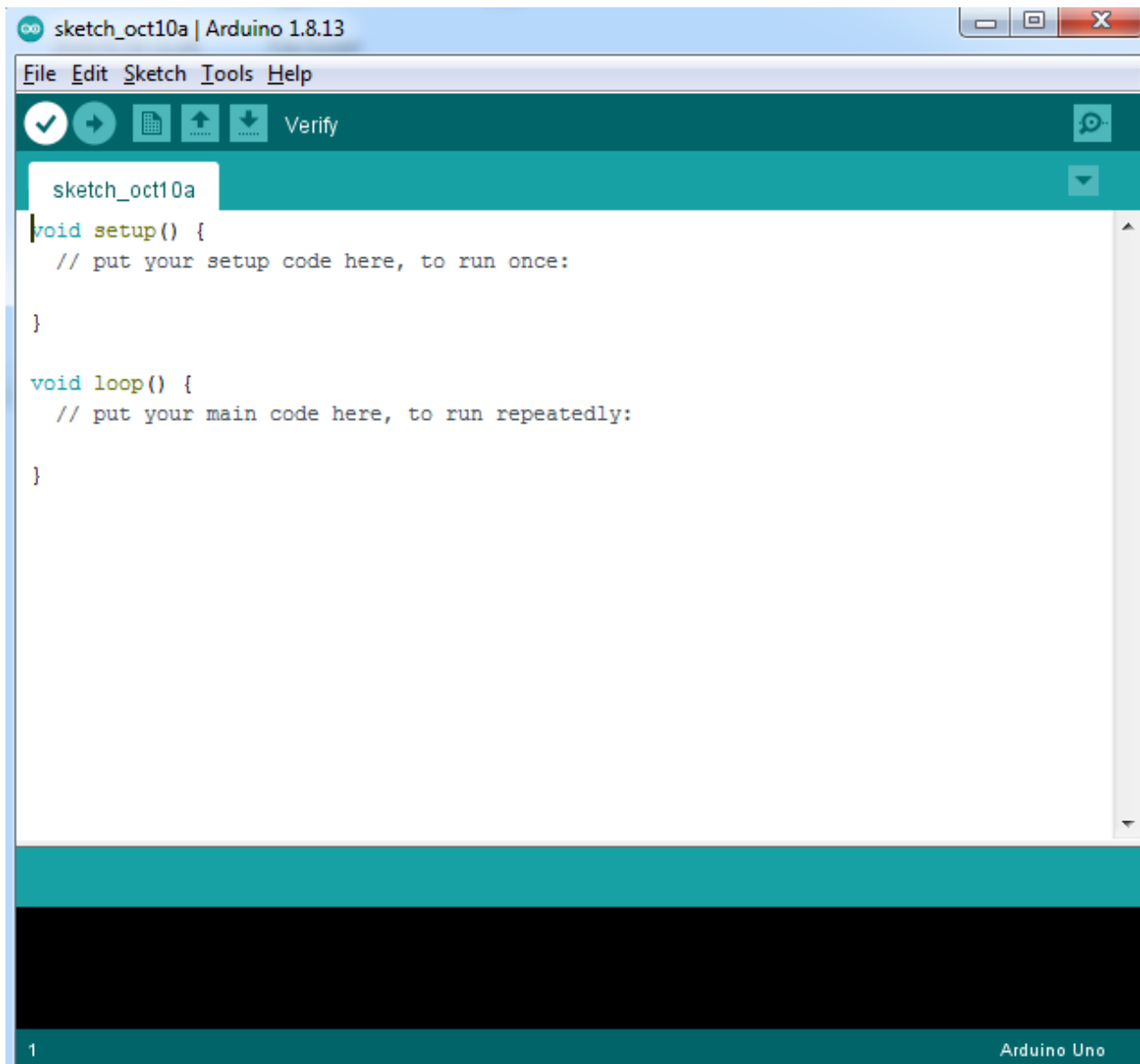




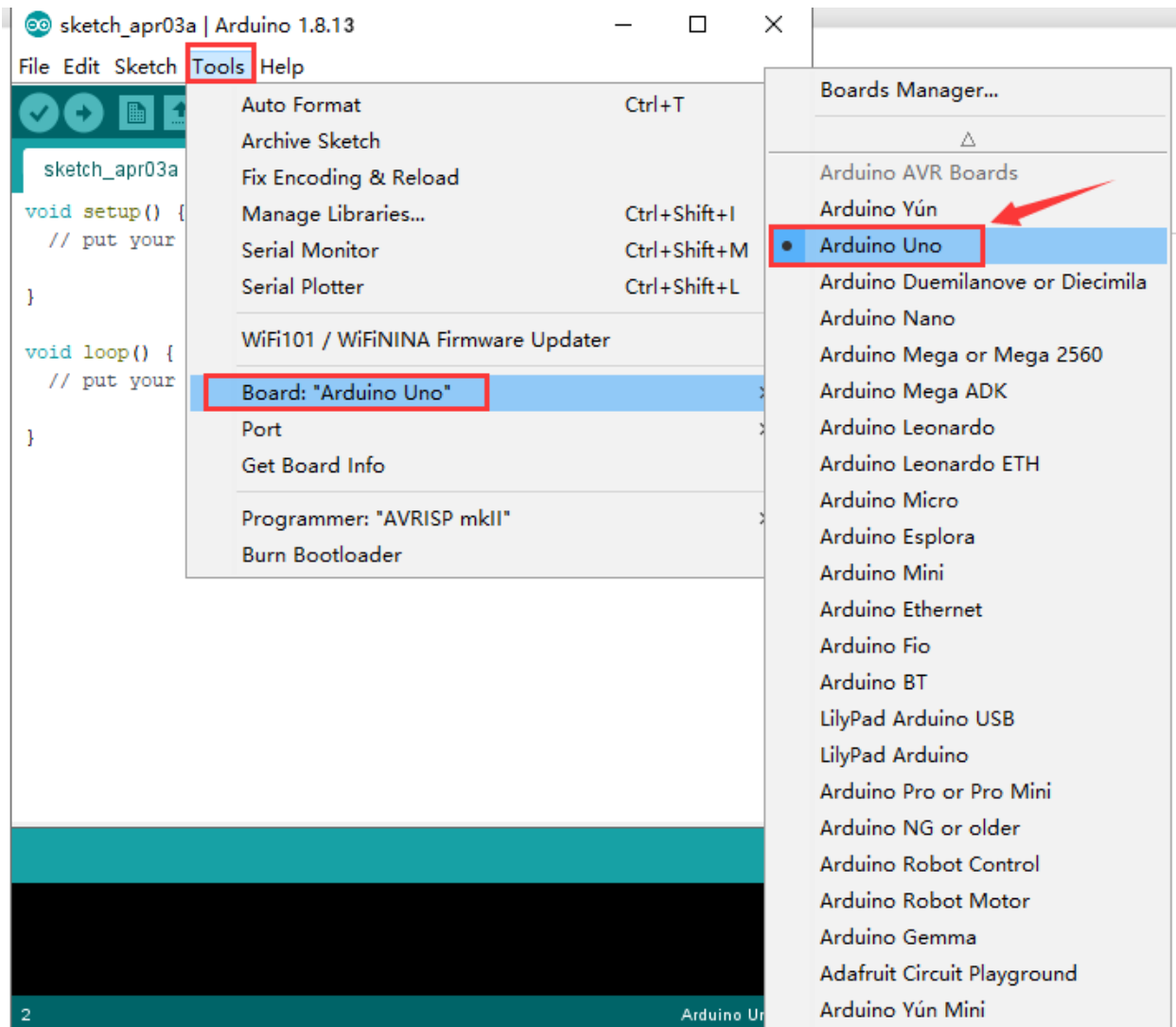
### 3 Arduino IDE Setting



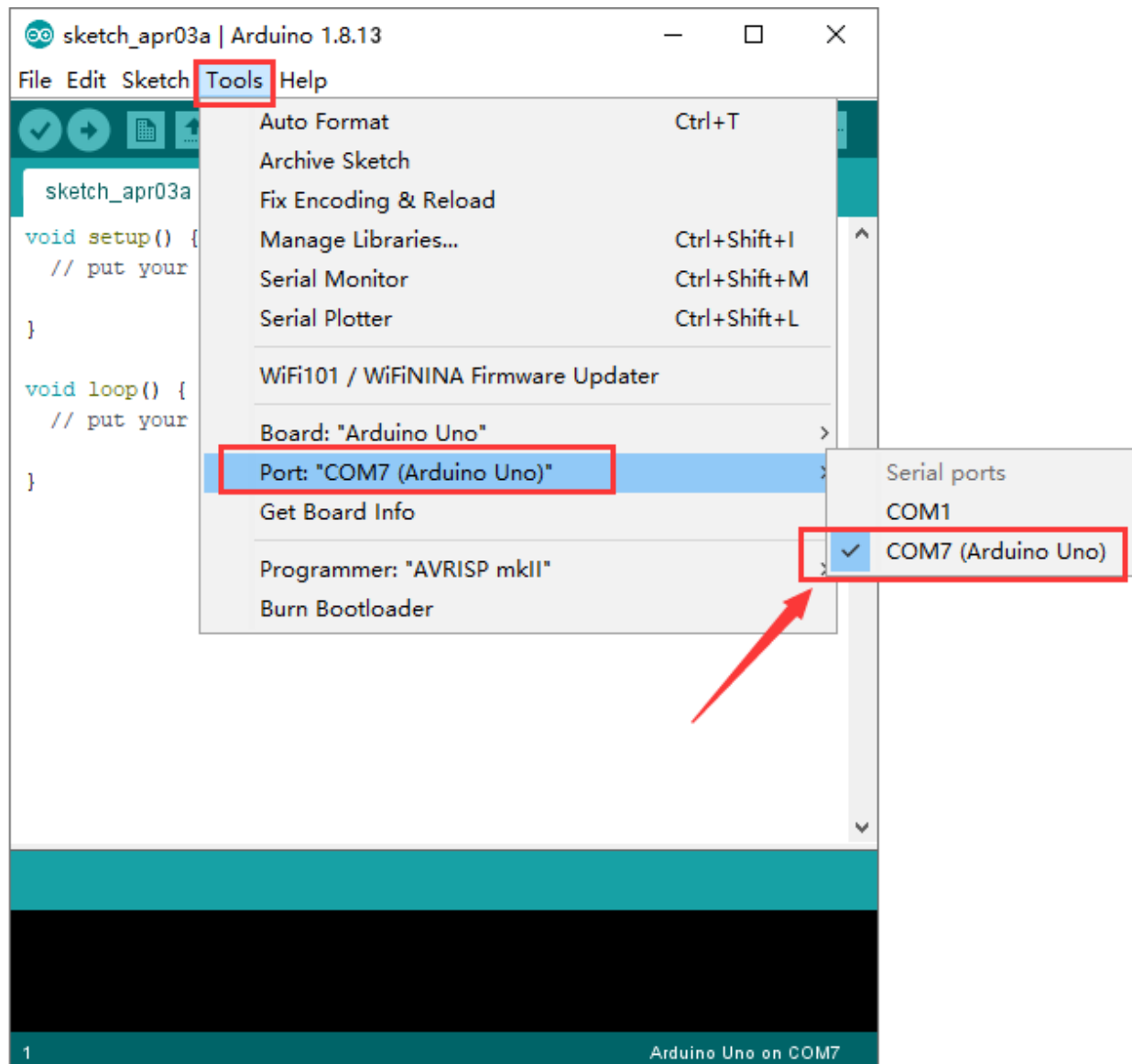
Click the icon and open Arduino IDE.



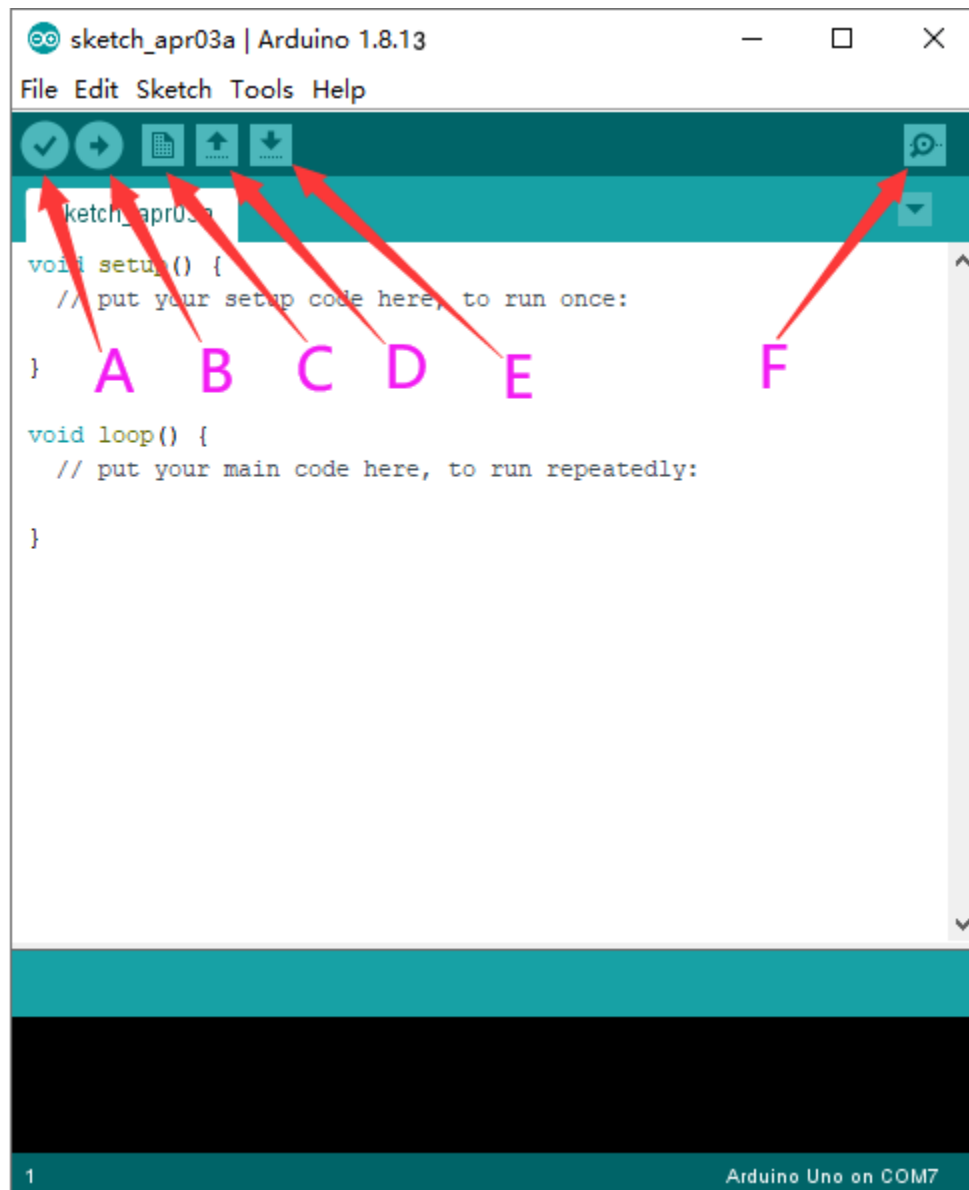
When downloading the sketch to the board, you must select the correct name of Arduino board that matches the board connected to your computer. As shown below;



Then select the correct COM port (you can see the corresponding COM port after the driver is successfully installed)



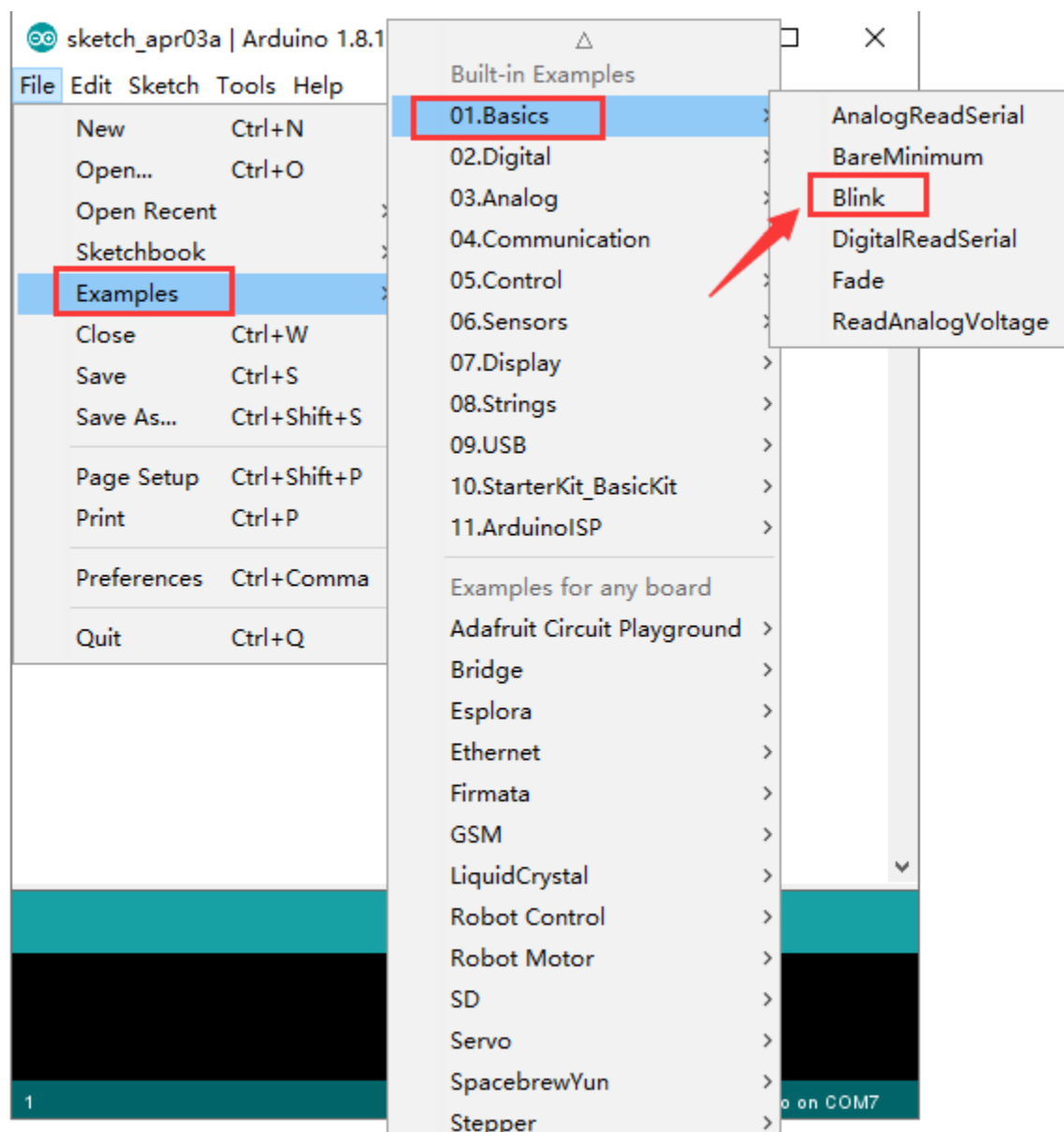




- A- Used to verify whether there is any compiling mistakes or not.
- B- Used to upload the sketch to your Arduino board.
- C- Used to create shortcut window of a new sketch.
- D- Used to directly open an example sketch.
- E- Used to save the sketch.
- F- Used to send the serial data received from board to the serial monitor.

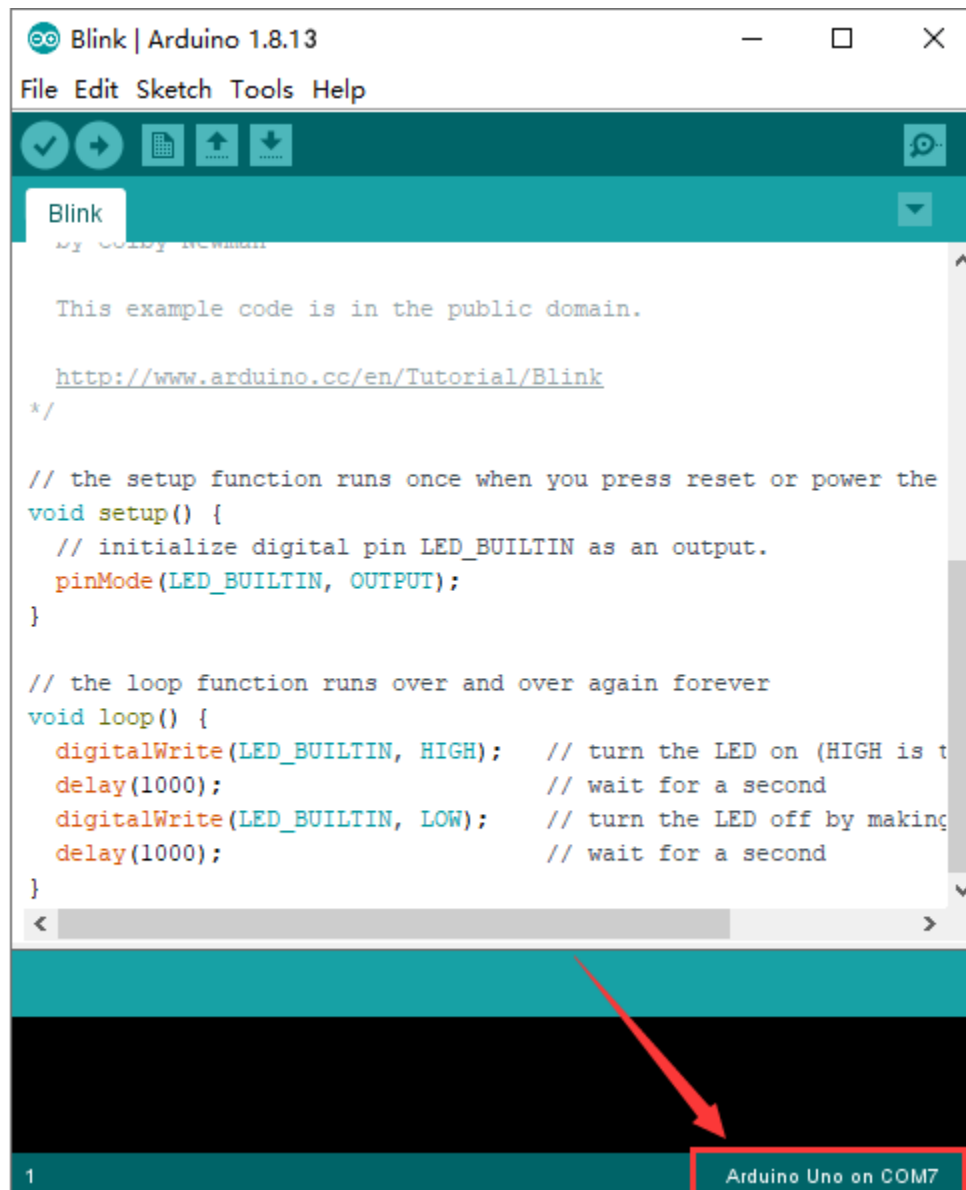
#### 4Start First Program


Open the file to select Example, choose BLINK from BASIC, as shown below:





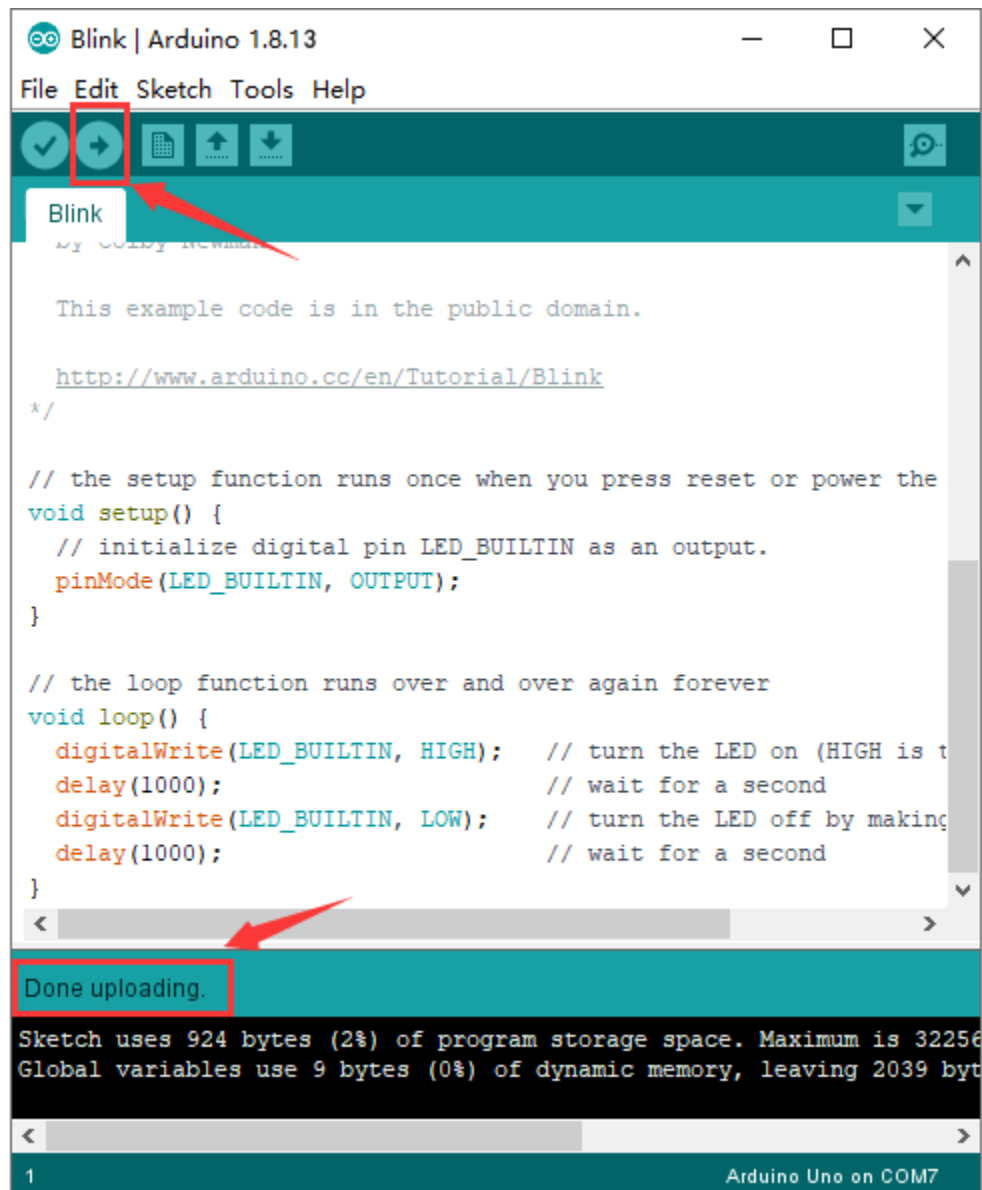
Set board and COM port, the corresponding board and COM port are shown on the lower right of IDE.



Click  to start compiling the program, and check errors.



Click  to upload the program



Upload the program successfully, the onboard LED lights on for 1s, lights off for 1s. Congratulation, you finish the first program.

## 5. Max development board Project

### Project 1 : Hello world



### Project instruction:

As the beginning of all programming languages, Hello world occupies an unchangeable position, so the first project of our Arduino tutorial starts with hello world.

The “Hello, world” program refers to a computer program that outputs the string “Hello, world” on the computer screen.

In this project, the computer is the Max control board and the screen is the serial monitor. This project only needs one development board and one USB cable.

#### 1. Tips:

After connecting the Max development board, open the Arduino IDE and ensure two points:

1. Select the correct development board version in Tools>Board (Max development board is compatible with Arduino UNO, so choose Arduino UNO for the board model in the software)
2. In Tools>Port, select the serial port where the Max development board is located.

#### 1. Project code:

```
/*
keyestudio Max Development Board
Project 1
hello world
http://www.keyestudio.com
*/

int val;//define variable val

void setup()
{
  Serial.begin(9600);//Set the baud rate to 9600, which must be consistent with the software settings.
}

void loop()
{
```

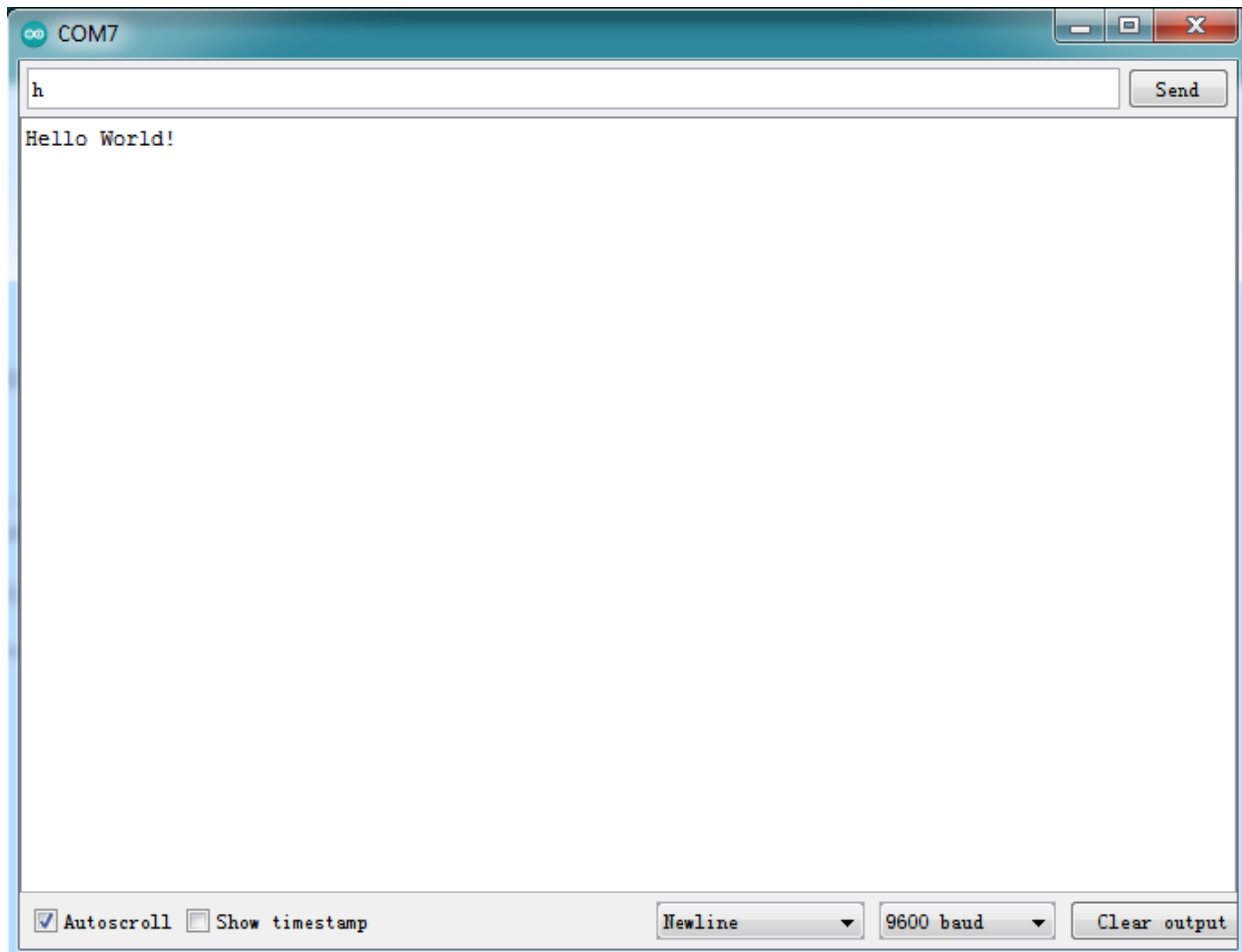
```
val=Serial.read();//Read the instructions or characters sent by the PC to the Arduino and assign them to val
if(val=='h')//Judging whether the received command or character is "h".
{
  //If the received is "h"
  Serial.println("Hello World!");//Display "Hello World!"
}
}
//*****
```

### 1. Project results

Open the Arduino IDE, copy and paste the code in the IDE, click Verify to compile, and check the program for errors. At this time, the code has not been uploaded to the Max board.

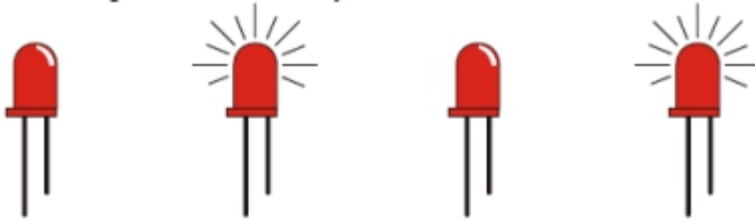
Click upload, you can see the TX and RX LEDs on the Max board flash a few times, and the IDE prompts Done uploading.

Upload the code to the development board, when enter "h" in the Serial Monitor, it will display "Hello, world!".



### Project 2: LED blink

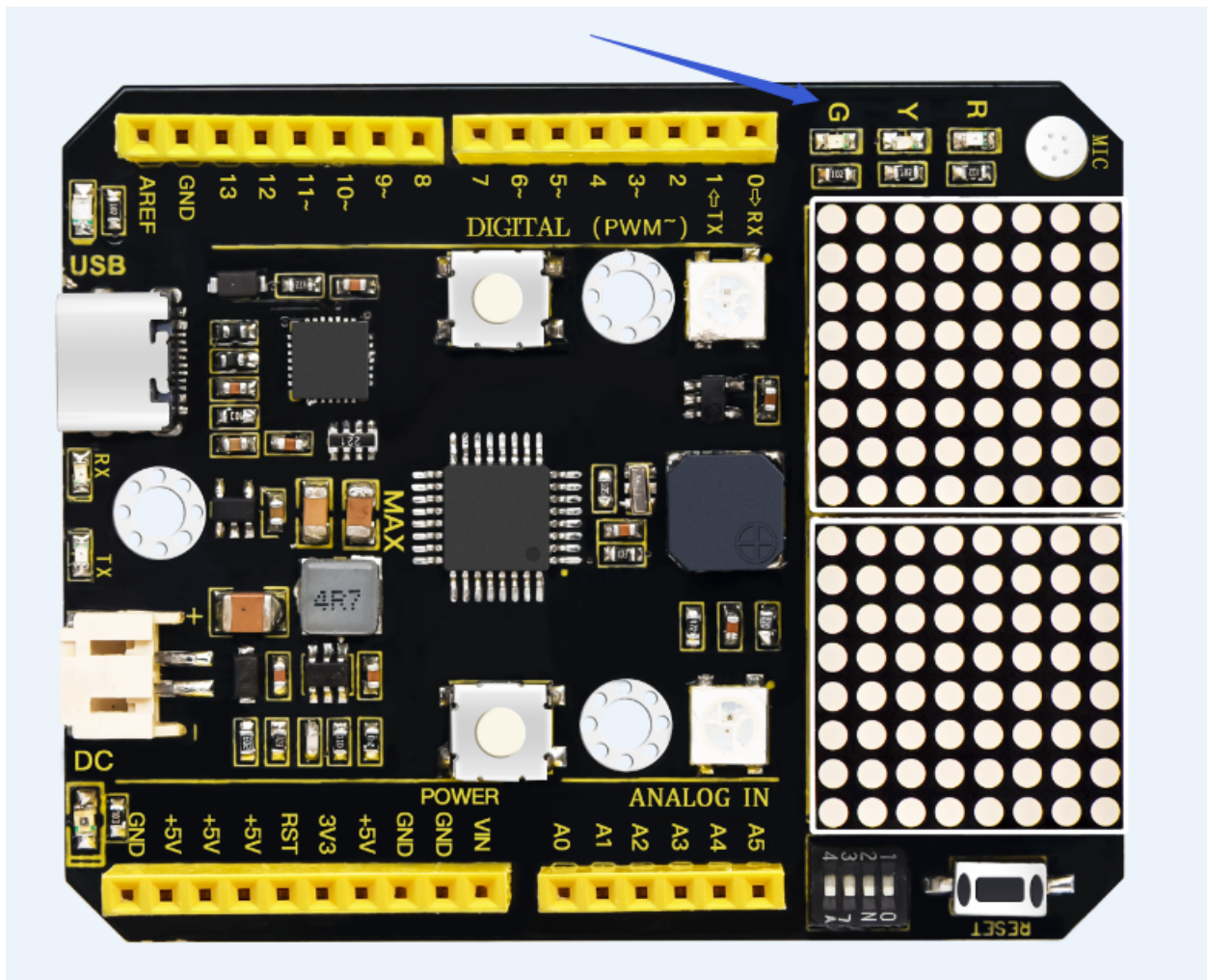




### 1. Project instruction

The next step is to start a very classic project-blink. This project is to light up an LED and control the delay time. It is an entry-level project for Arduino beginners.

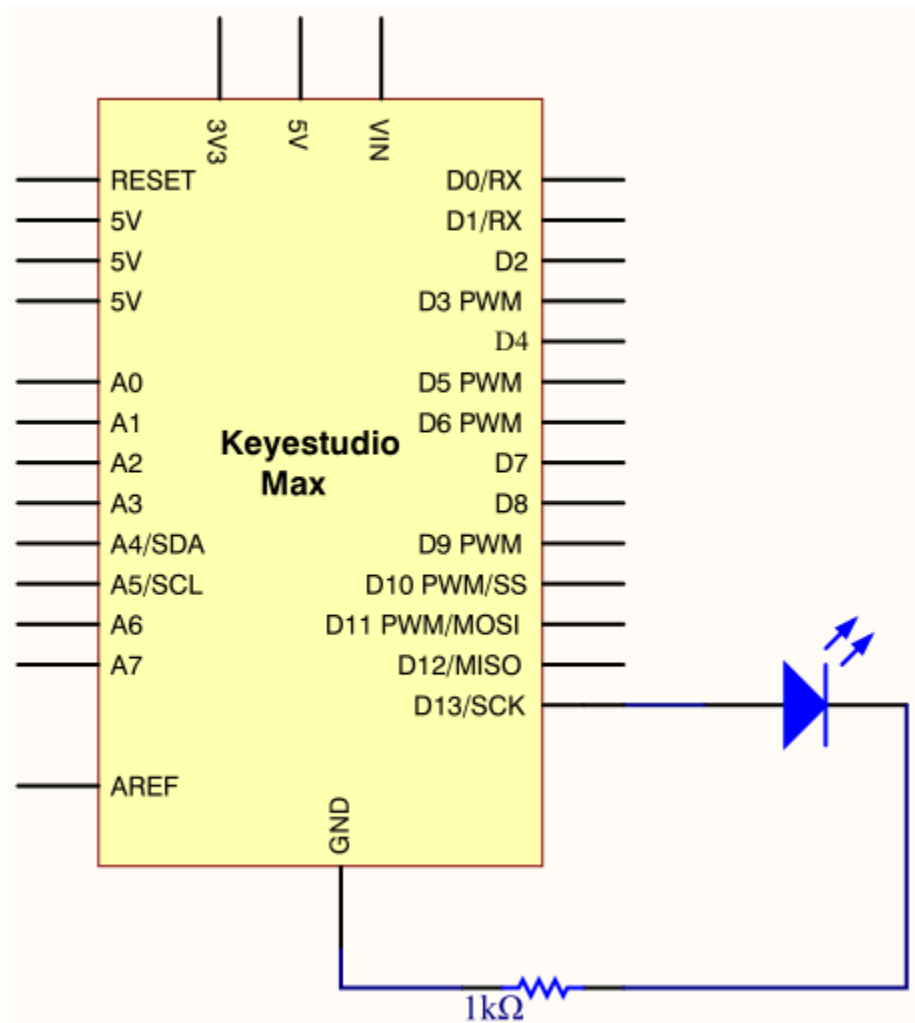
In this example, the LED we use is an LED onboard MAX, As shown in the picture below, it is directly connected to the D13 pin.



### 1. Project Principle

LED is a type of semiconductor called “Light Emitting Diode” which is an electronic device made of semiconductor materials (silicon, selenium, germanium, etc.). It is dubbed indicator, digital and word display in circuit and device.

### 1. Project circuit



### 1. Project code

```

/*
keyestudio Max Development Board
Project 2
Blink
http://www.keyestudio.com
*/

// the setup function runs once when you press reset or power the board
void setup() {
  // initialize digital pin 13 as an output.
  pinMode(13, OUTPUT);
}

// the loop function runs over and over again forever
void loop() {

```

```
digitalWrite(13, HIGH); // turn the LED on (HIGH is the voltage level)
delay(1000); // wait for a second
digitalWrite(13, LOW); // turn the LED off by making the voltage LOW
delay(1000); // wait for a second
} //*****
```

When the upload is successful, it will prompt “Done uploading”. After everything is successful, you can see the LED of D13 is on for one second and off for one second.

### 1. Code description

setup() function is the basic function of the Arduino program and must be included. It will only be executed once when the program is running. It is commonly used to define pins and declare variables.

loop(): loop function is also a necessary function of the Arduino program. Its function is to run after setup() is executed once, and it keeps repeating.

digitalWrite(13, HIGH) Set the output voltage of the pin to high or low. Here D13 is high level, so that the LED can light up.

delay(1000) The delay function is a delay function, the value here is 1000, the unit is milliseconds (ms), 1000ms is 1 second.

What if you want to make the LED flash faster? Yes, you only need to modify the delay. Decrease the value, it means that the waiting time is shorter, that is, the flashing is faster. or, the flash is slower.

### Project 3: SOS light



#### 1. Project instruction

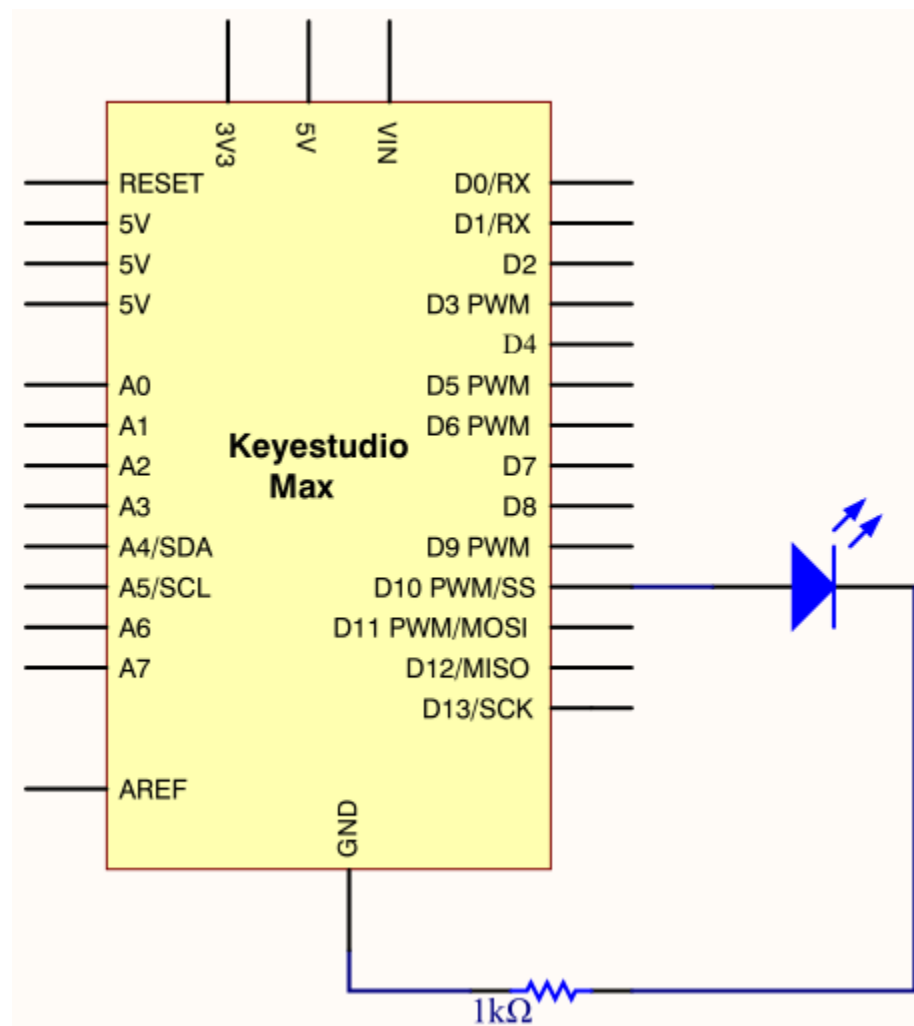
SOS is a Morse code distress signal (), used internationally, that was originally established for maritime use. In formal notation SOS is written with an overscore line, to indicate that the Morse code equivalents for the individual letters of “SOS” are transmitted as an unbroken sequence of three dots / three dashes / three dots, with no spaces between the letters.[1] In International Morse Code three dots form the letter “S” and three dashes make the letter “O”, so “S O S” became a common way to remember the order of the dots and dashes. (IWB, VZE, 3B, and V7 form equivalent sequences, but traditionally SOS is the easiest to remember.)

#### Project Principle

We could produce a letter through LED on and off state, long and short flash respectively represent dot and dash.

Through Morse code table, “S” is represented by three dots, therefore, we replace three dots with short flash, dash symbolizes letter “O”, replace three dashes with long flash.

### Project circuit



### Project code

```
\/*  
keyestudio Max Development Board  
  
Project 3  
  
s o s  
  
http://www.keyestudio.com  
  
\*/  
  
int ledPin = 10;  
  
void setup() {
```

(continues on next page)

(continued from previous page)

```
pinMode(ledPin, OUTPUT);

}

void loop() {

// stand for“ S”with three quick flash
for(int x=0;x<3;x++){

digitalWrite(ledPin,HIGH); //set LED on
delay(150); //delay in 150ms

digitalWrite(ledPin,LOW); //set LED off
delay(100); //delay 100ms

}

// time gap to produce next letter
delay(100);

//stand for“ O”with three short flash
for(int x=0;x<3;x++){

digitalWrite(ledPin,HIGH); //set LED on
delay(400); //delay in 400ms

digitalWrite(ledPin,LOW); //set LED off
delay(100); //delay in 100ms

}

// time gap to produce next letter
delay(100);

// stand for“ S”with three quick flash
for(int x=0;x<3;x++){

digitalWrite(ledPin,HIGH); //set LED on
delay(150); //delay in 150ms

digitalWrite(ledPin,LOW); //set LED off
```

(continues on next page)

(continued from previous page)

```

delay(100); //delay in 100ms

}

// wait 5s before repeating S.O.S signal

delay(5000);

}

```

Upload code to Arduino, we will view LED flash S.O.S signal and wait for 5s and flash again. Plug external power in Arduino, and place them in a waterproof box, then S.O.S signal can be sent.

S.O.S signal is usually applied in maritime use and mountain-climbing

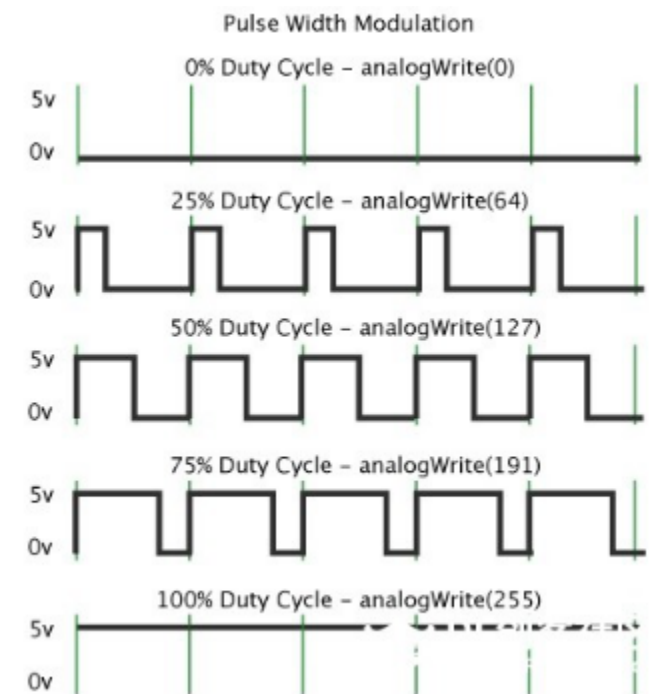
### Project 4: Breathing LED

#### Project instruction

We have learned how to control LED on and off, besides, the light intensity of LED can be controlled by program. The pins marked with "~" on Arduino board indicate they have PWM function. In this project, we will control LED brightness through PWM to simulate breathing effect.

#### Project Principle

PWM is a means of controlling the analog output via digital means. Digital control is used to generate square waves with different duty cycles (a signal that constantly switches between high and low levels) to control the analog output. In general, the input voltage of port are 0V and 5V. What if the 3V is required? Or what if switch among 1V, 3V and 3.5V? We can't change resistor constantly. For this situation, we need to control by PWM.

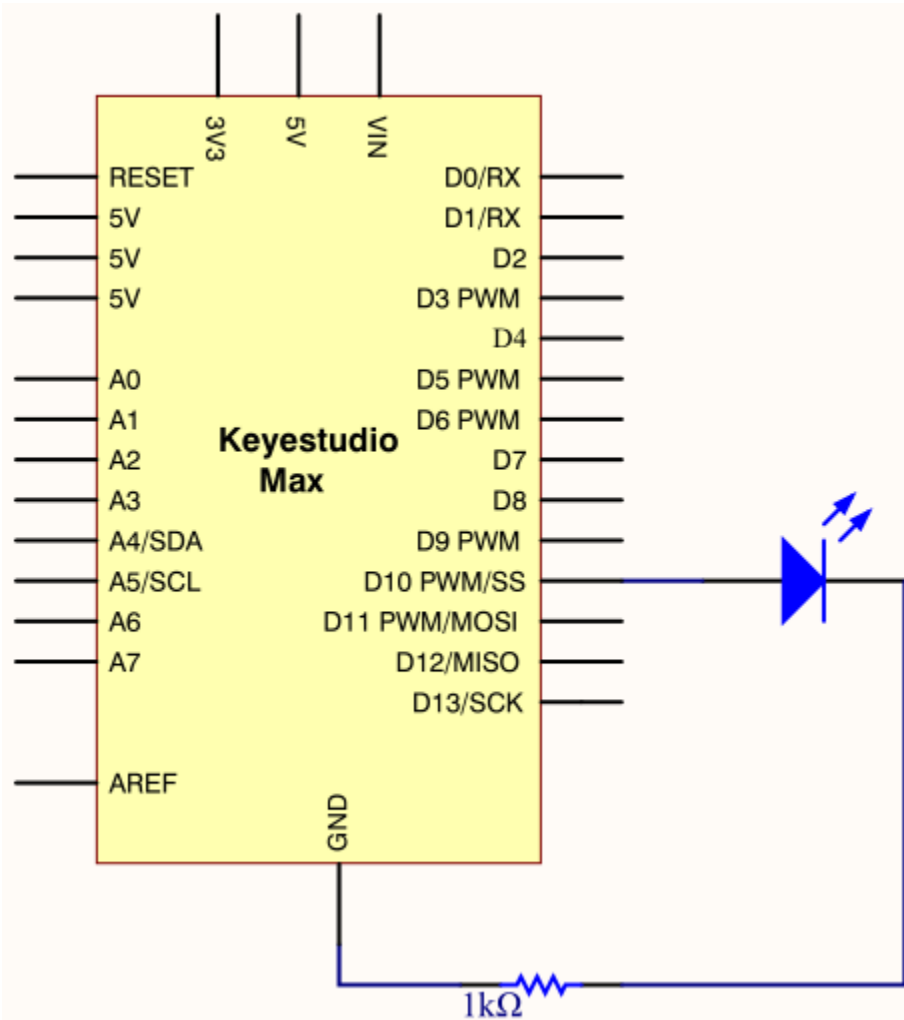


In the above figure, the green line represents a period, and value of analogWrite() corresponds to a percentage which is called Duty Cycle as well. Duty cycle implies that high-level duration is divided by low-level duration in a cycle. From top to bottom, the duty cycle of first square wave is 0% and its corresponding value is 0. The LED brightness is

lowest, that is, turn off. The more time high level lasts, the brighter the LED. Therefore, the last duty cycle is 100%, which correspond to 255, LED is brightest. 25% means darker.

PWM mostly is used for adjusting the LED brightness or rotation speed of motor.

### Project circuit



### Project code

```

/\*
keyestudio Max Development Board

Project 4

Breathing LED

http://www.keyestudio.com

\*/

int ledPin = 10;

```

(continues on next page)

(continued from previous page)

```
int value;

void setup() {
  pinMode(ledPin,OUTPUT);
}

void loop()
{
  for (value = 0 ; value \< 255; value=value+1)
  {
    analogWrite(ledPin, value);
    delay(5);
  }
  for (value = 255; value \>0; value=value-1)
  {
    analogWrite(ledPin, value);
    delay(5);
  }
}
```

Upload test code successfully, LED gradually becomes brighter then darker, like human breath, rather than turn on and off immediately.

**Code description**

Function analogWrite()

We know that digital port only has two state of 0 and 1. So how to send an analog value to a digital value? Here, this function is needed. Let's observe the Arduino board and find 6 pins marked "~" which can output PWM signals.

The formula as follows

analogWrite(pin,value)

analogWrite() is used to write an analog value from 0~255 for PWM port, so the value is in the range of 0~255. Attention that you only write the digital pins with PWM function, such as pin 3, 5, 6, 9, 10, 11.

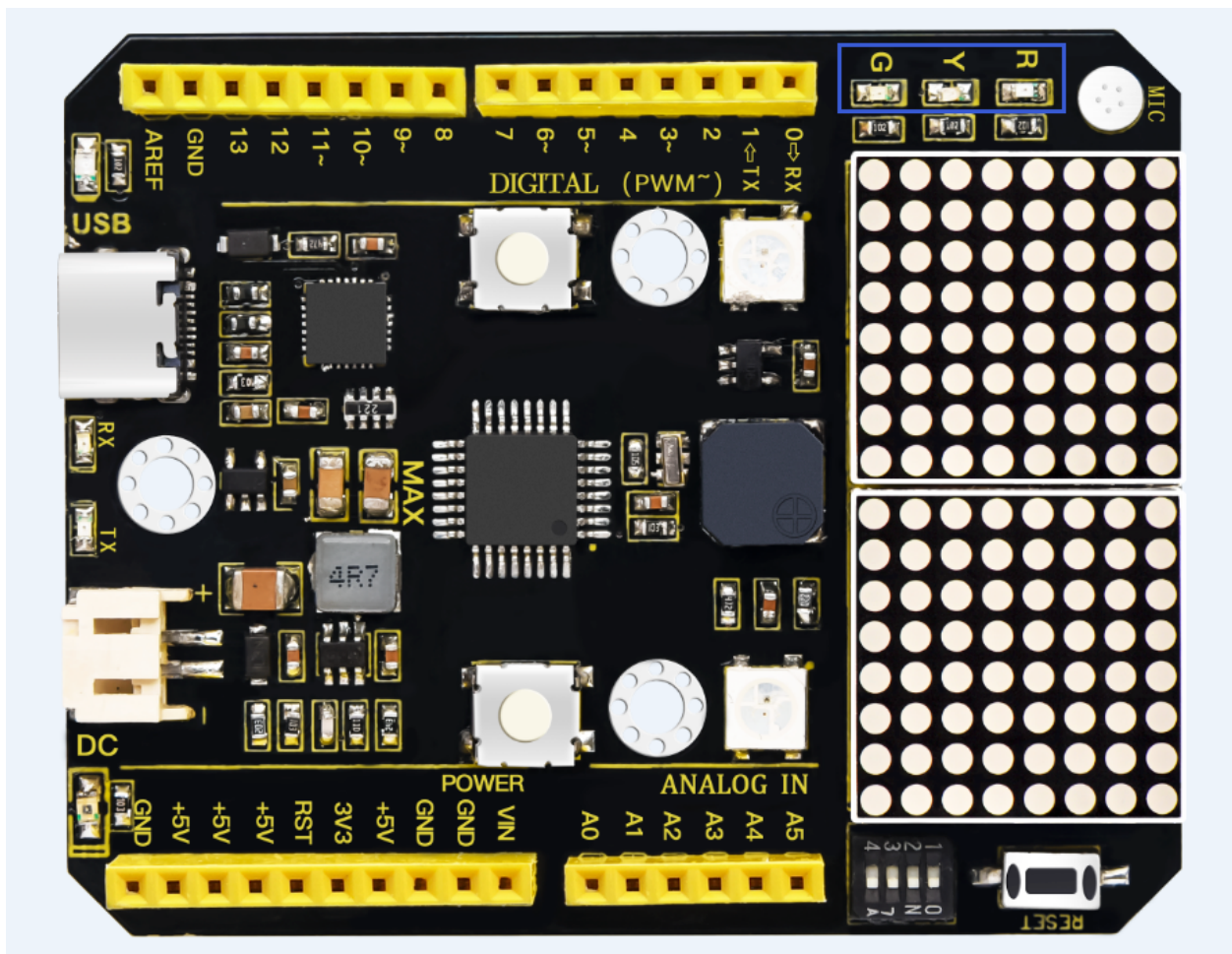
**Project 5: Traffic light**





### 1. Project instruction

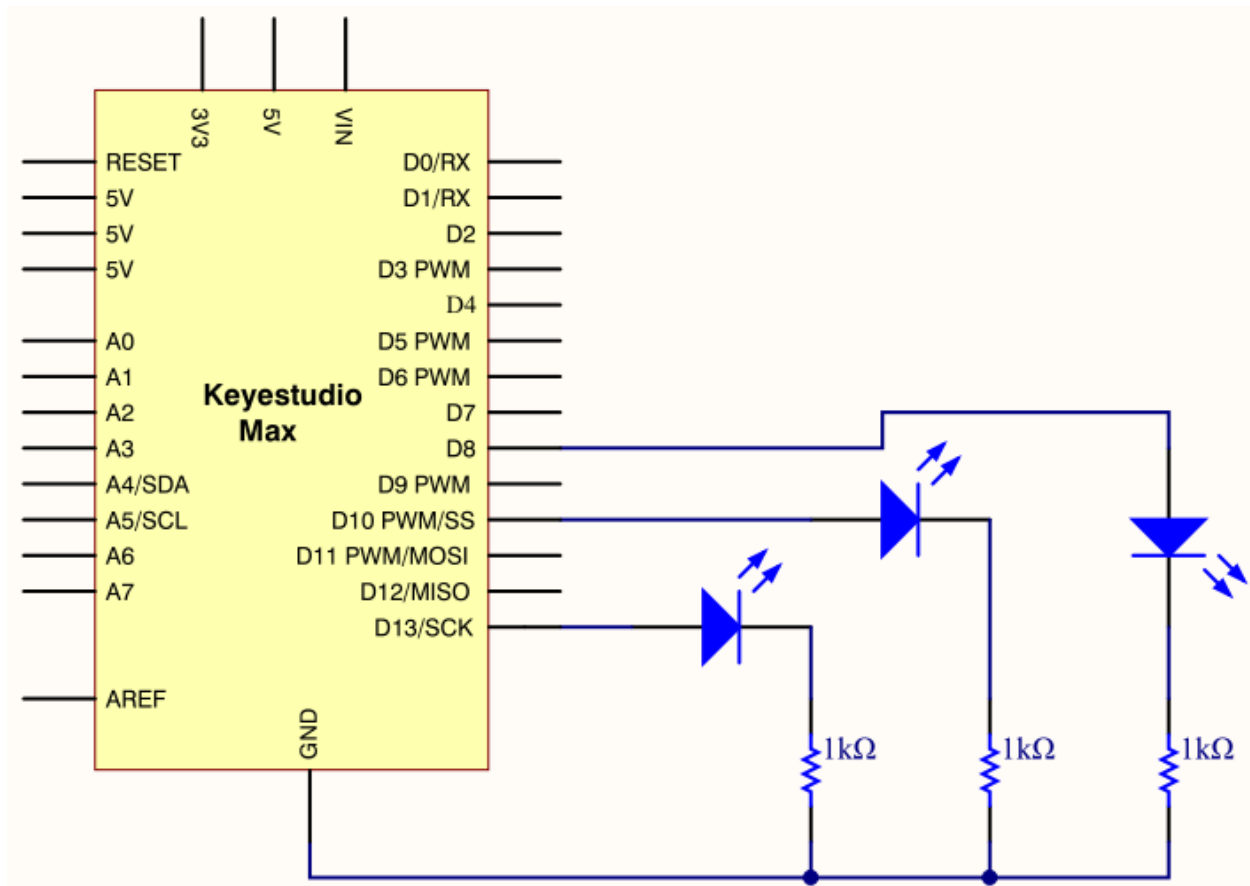
In the previous program, we have done the LED blinking experiment with one LED. Now, it's time to up the stakes to do a bit more complicated experiment-traffic light. Actually, these two experiments are similar. While in this traffic light experiment, we use three LEDs with different colors rather than an LED.



### 1. Project Principle

Max board imitates traffic light by controlling flashing time of LED through program.

### 1. Project circuit



#### 1. Project code

```

/*
keyestudio Max Development Board
Project 5
Traffic light
http://www.keyestudio.com
*/

int redled =8; // initialize digital pin 8.
int greenled =10; // initialize digital pin 10.
int yellowled =13; // initialize digital pin 13.

void setup()
{
  pinMode(redled, OUTPUT); // set the pin with red LED as“output”
  pinMode(yellowled, OUTPUT); // set the pin with yellow LED as“output”
  pinMode(greenled, OUTPUT); // set the pin with green LED as“output”
}

void loop()

```

```
{  
digitalWrite(greenled, HIGH);/// turn on green LED  
delay(5000);// wait 5 seconds  
digitalWrite(greenled, LOW); // turn off green LED  
for(int i=0;i<3;i++)// blinks for 3 times  
{  
delay(500);// wait 0.5 second  
digitalWrite(yellowled, HIGH);// turn on yellow LED  
delay(500);// wait 0.5 second  
digitalWrite(yellowled, LOW);// turn off yellow LED  
}  
delay(500);// wait 0.5 second  
digitalWrite(redled, HIGH);// turn on red LED  
delay(5000);// wait 5 second  
digitalWrite(redled, LOW);// turn off red LED  
}
```

### Project 6: LED Chasing Effect

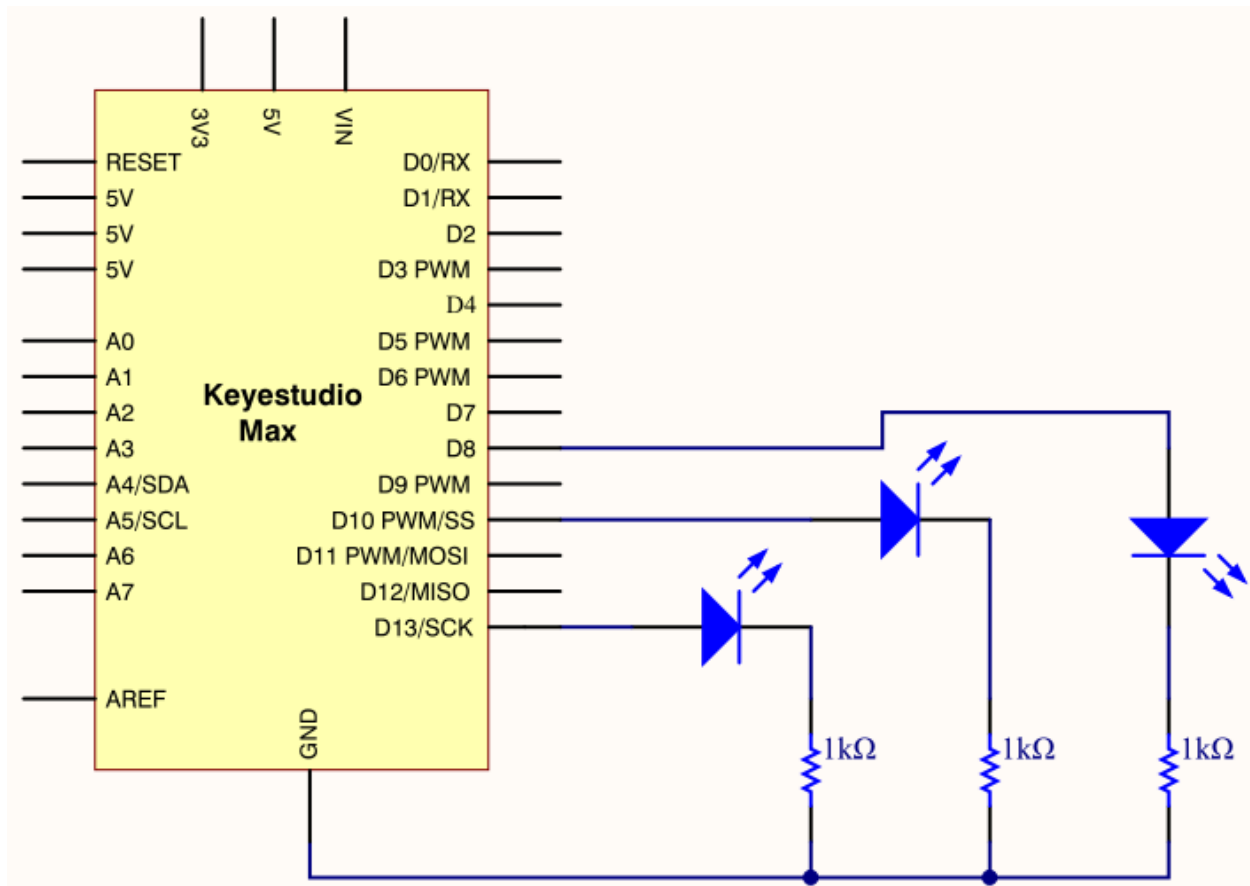


#### 1. Project instruction

Flow light is commonly viewed in our daily life, such as on building, advertisement board and so on.

We will show you how to make a flow light in this part.

#### 1. Project circuit



### 1. Project code

```

/*
keyestudio Max Development Board
Project 6
LED Chasing Effect
http://www.keyestudio.com
*/

int redled = 8;
int greenled = 10;
int yellowled = 13;

void setup(){
  pinMode(redled, OUTPUT);
  pinMode(greenled, OUTPUT);
  pinMode(yellowled, OUTPUT);
}

void loop()
{

```

```

digitalWrite(redled, HIGH);//turn on red light
delay(100);//delay in 0.1s
digitalWrite(redled, LOW); //turn off red light
delay(100);//delay in 1s
digitalWrite(greenled, HIGH);//turn on green light
delay(100);//delay in 0.1s
digitalWrite(greenled, LOW);//turn off green light
delay(100);//delay in 0.1s
digitalWrite(yellowled, HIGH);//turn on yellow light
delay(100);//delay in 0.1s
digitalWrite(yellowled, LOW);//turn off yellow light
delay(100);//delay in 0.1s
}

```

### Project 7: Play Music



#### 1. project instruction

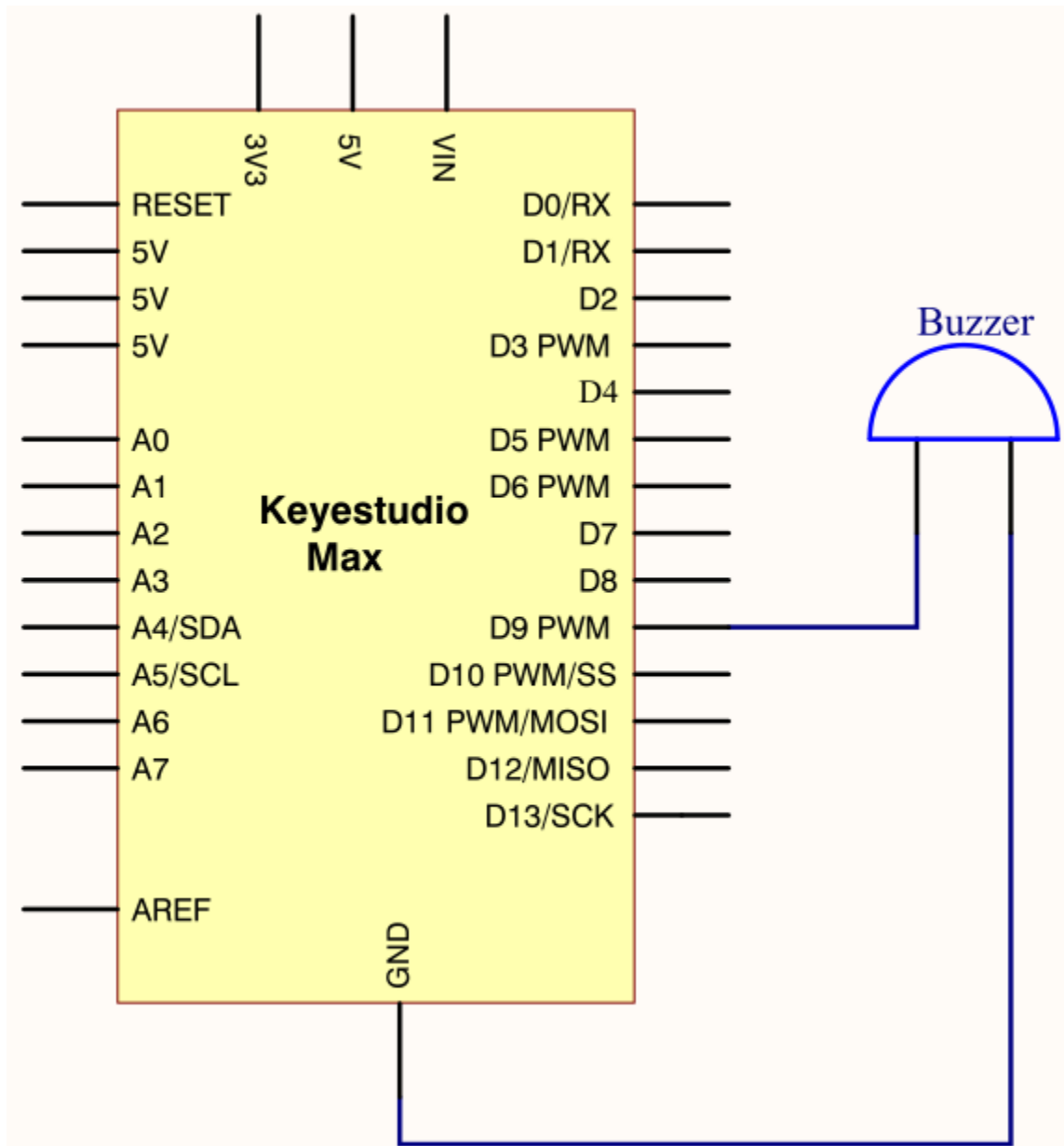
We can use Arduino to make many interactive works. The most commonly used one is acoustic-optic display. All the previous experiment has something to do with LED. However, the circuit in this experiment can produce sound. Normally, the experiment is done with a buzzer but not a speaker while buzzer is more simpler and easier to use.

The buzzer we introduced here is a passive buzzer. It cannot be actuated by itself, but by external pulse frequencies. Different frequency produces different sound. We can use Arduino to code the melody of a song, which is quite fun and simple.

#### 1. Project Principle

Passive buzzer is an integrated electronic buzzer without vibration source inside. It must be driven by 2K-5K square wave instead of direct current signals. There is little difference between the two buzzers, but when the pins of the two buzzers are placed up, the passive buzzer comes with green circuit board, and the one sealed with vinyl is an active buzzer.

#### 1. Project circuit



### 1. Project code

```

/*
keyestudio Max Development Board
Project 7
Play Music
http://www.keyestudio.com
*/

int tonepin=9;// set pin9 connecting buzzer

void setup()
{

```

```

pinMode(tonepin,OUTPUT);//set IO port to output
}
void loop()
{
  unsigned char i,j;
  while(1)
  {
    for(i=0;i<80;i++)//output a sound with frequency
    {
      digitalWrite(tonepin,HIGH);//emit a sound
      delay(1);//delay in 1ms
      digitalWrite(tonepin,LOW);//don't emit a sound
      delay(1);//delay 1 ms
    }
    for(i=0;i<100;i++)// output another sound with frequency, 100 controls frequency and can be adjusted also by yourselves
    {
      digitalWrite(tonepin,HIGH);
      delay(2);
      digitalWrite(tonepin,LOW);
      delay(2);
    }
  }
}

```

Seen from the above code, 80 and 100 in the for statement effects frequency, and delay controls time, alike beats in music.

## Project 8: WS2812 RGB LED

### 1. project instruction

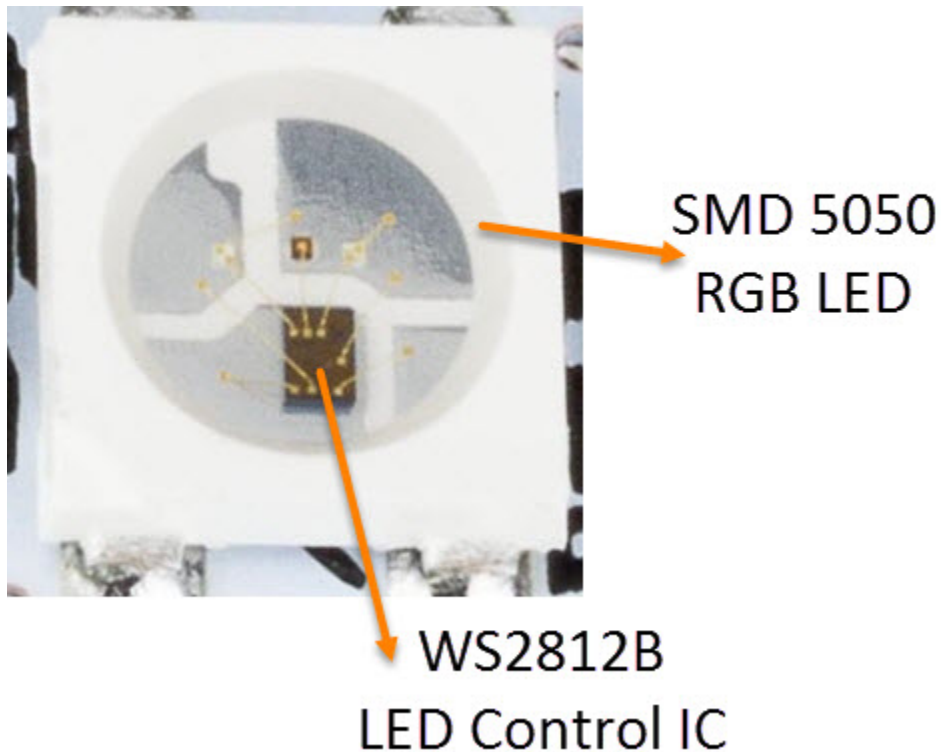
In this project, we will take a look at an interesting product called the WS2812B Addressable LEDs. They are individually addressable RGB LEDs and are also called by different names like NeoPixel by Adafruit, for example. In this tutorial, we will learn about the WS2812B LEDs, its internal structure and construction and also how to control individually addressable LEDs using Arduino.

### 1. Project Principle

The WS2812B Addressable LED Strip is an intelligent light source that contains a control IC and an RGB LED in a same package, usually in SMD 5050 form factor. The following image shows a strip of WS2812B LEDs.



At first glance, it looks like a regular LED strip. But upon a closer inspection, you can find the WS2812B LED Control IC integrated into the SMD 5050 RGB LED Package. The following blown up image shows the same.



Individual WS2812B LED Block has four pins namely VDD, VSS (GND), DIN and simple function description of the pins.

Pin	Function
VDD	Power Supply for LED
VSS	Ground
DIN	Control Data Signal Input
DOUT	Control Data Signal Output



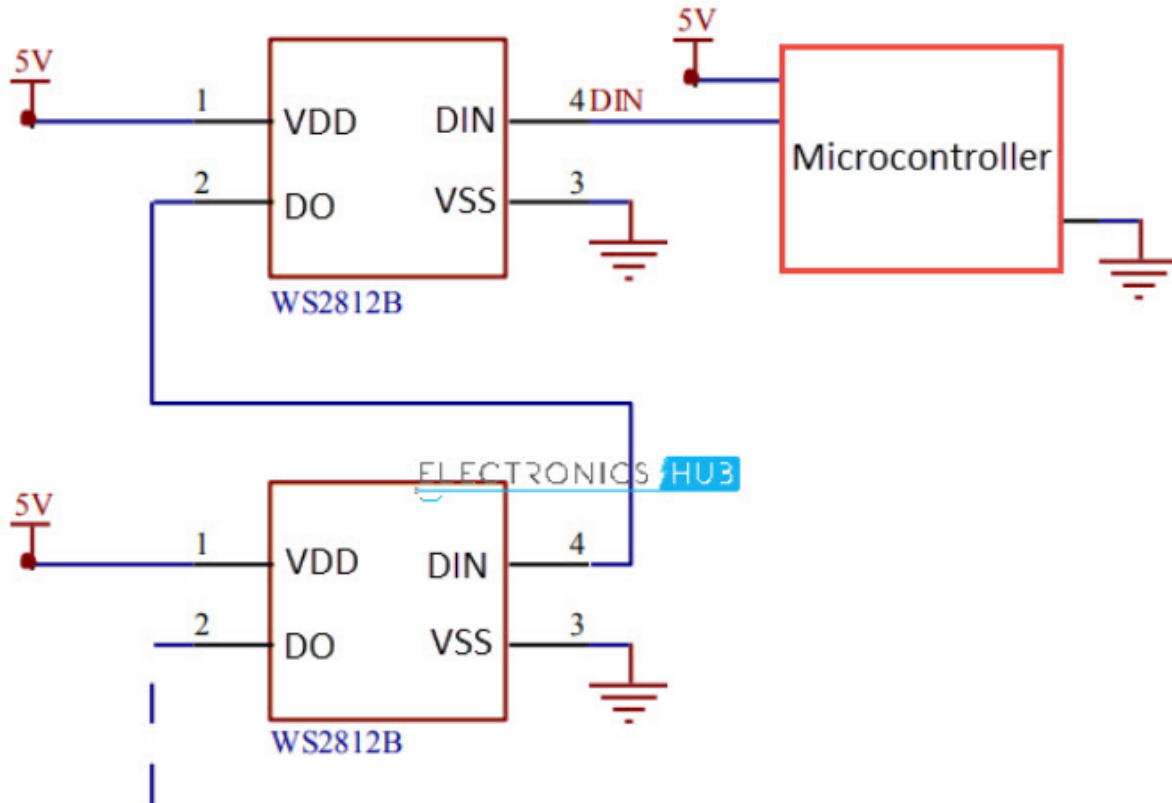
The power supply for the WS2812B IC is also supplied through the VDD Pin.

DOOUT. The functions of these pins are very simple and the following table gives a

### WS2812B LED Application Circuit

To understand how the WS2812B LEDs can be individually controlled, the following application circuit will be useful. All the WS2812B LEDs are connected in a cascade manner where the DO of the first LED is connected to the DIN of the second LED and so on.

But the first LED has to receive data through its DIN pin from a Microcontroller like Arduino. The following image shows the typical application circuit block diagram.

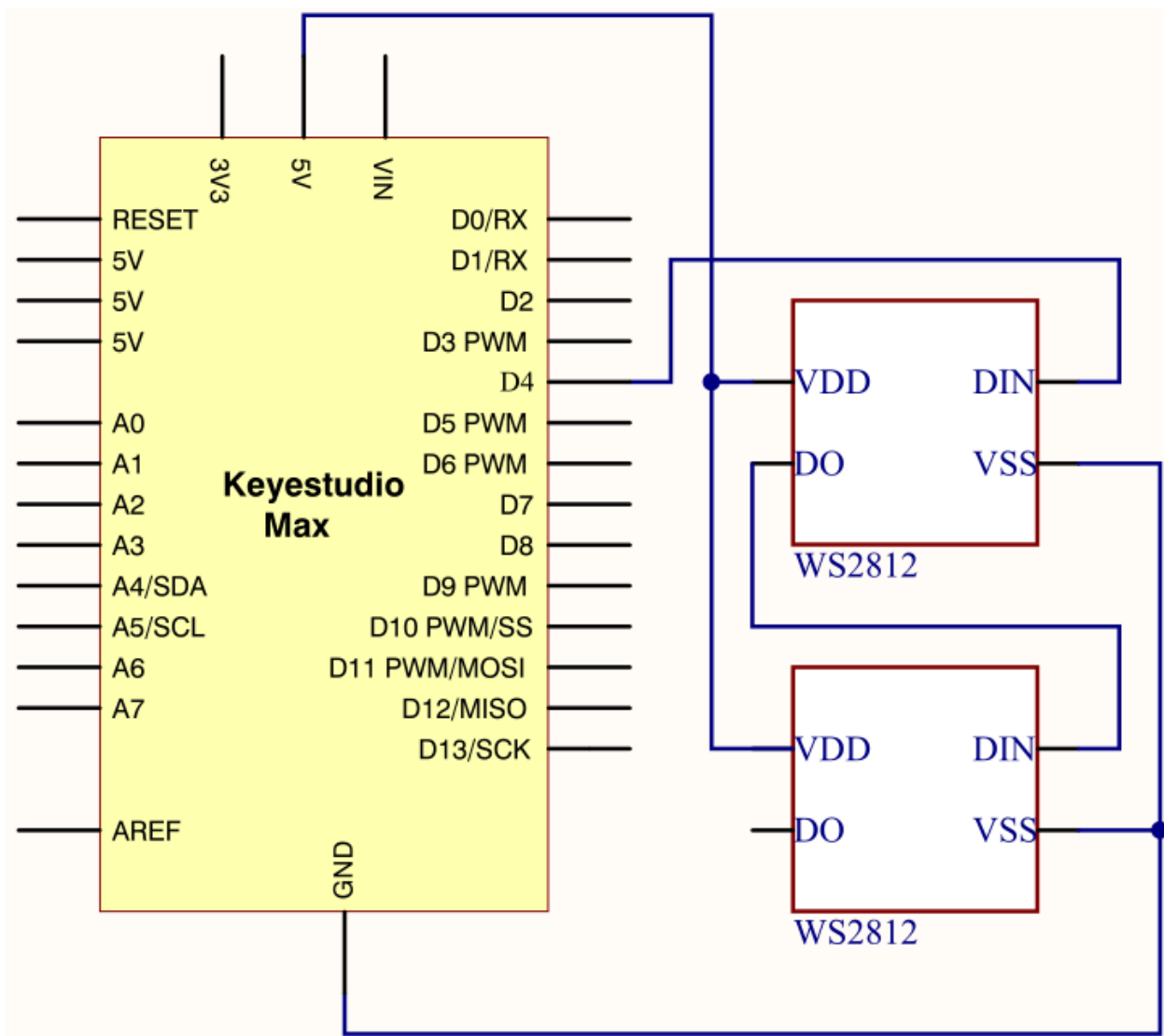


The data transfer protocol used by the WS2812B LEDs is NRZ Mode. The first DIN Port of the WS2812B LED array receives data from the microcontroller. The data for individual pixel is of 24-bit that consists of individual Red, Green and Blue LED control data of 8-bits each. The order of data must be GRB and the composition of the 24-bit data is shown below. Note that HIGH bit data is sent first.

G7	G6	G5	G4	G3	G2	G1	G0	R7	R6	R5	R4	R3	R2	R1	R0	B7	B6	B5	B4	B3	B2	B1	B0
----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----

Once the first WS2812B Block receives the first 24-bit data, the data is sent to its internal latch for further decoding. The remaining data is reshaped by its signal reshaping and amplification circuit and is passed to the next pixel in the cascade through the DO pin.

#### 1. Project circuit



### 1. Project code

```

/*
keyestudio Max Development Board
Project 8
WS2812 RGB LED
http://www.keyestudio.com
*/
#include <Adafruit_NeoPixel.h>
#ifdef _AVR_
#include <avr/power.h>
#endif
#define PIN 4

```

---

```

// Parameter 1 = number of pixels in strip
// Parameter 2 = Arduino pin number (most are valid)
// Parameter 3 = pixel type flags, add together as needed:
// NEO_KHZ800 800 KHz bitstream (most NeoPixel products w/WS2812 LEDs)
// NEO_KHZ400 400 KHz (classic 'v1' (not v2) FLORA pixels, WS2811 drivers)
// NEO_GRB Pixels are wired for GRB bitstream (most NeoPixel products)
// NEO_RGB Pixels are wired for RGB bitstream (v1 FLORA pixels, not v2)
Adafruit_NeoPixel strip = Adafruit_NeoPixel(2, PIN, NEO_GRB + NEO_KHZ800);
// IMPORTANT: To reduce NeoPixel burnout risk, add 1000 uF capacitor across
// pixel power leads, add 300 - 500 Ohm resistor on first pixel's data input
// and minimize distance between Arduino and first pixel. Avoid connecting
// on a live circuit...if you must, connect GND first.
void setup() {
// This is for Trinket 5V 16MHz, you can remove these three lines if you are not using a Trinket
#ifdef (AVR_ATtiny85)
if (F_CPU == 16000000) clock_prescale_set(clock_div_1);
#endif
// End of trinket special code
strip.begin();
strip.show(); // Initialize all pixels to 'off'
}
void loop() {
// Some example procedures showing how to display to the pixels:
colorWipe(strip.Color(255, 0, 0), 50); // Red
colorWipe(strip.Color(0, 255, 0), 50); // Green
colorWipe(strip.Color(0, 0, 255), 50); // Blue
// Send a theater pixel chase in...
theaterChase(strip.Color(127, 127, 127), 50); // White
theaterChase(strip.Color(127, 0, 0), 50); // Red
theaterChase(strip.Color(0, 0, 127), 50); // Blue
rainbow(20);
rainbowCycle(20);
theaterChaseRainbow(50);
}
// Fill the dots one after the other with a color
void colorWipe(uint32_t c, uint8_t wait) {

```

---

```
for(uint16_t i=0; i<strip.numPixels(); i++) {
  strip.setPixelColor(i, c);
  strip.show();
  delay(wait);
}

void rainbow(uint8_t wait) {
  uint16_t i, j;
  for(j=0; j<256; j++) {
    for(i=0; i<strip.numPixels(); i++) {
      strip.setPixelColor(i, Wheel((i+j) & 255));
    }
    strip.show();
    delay(wait);
  }
}

// Slightly different, this makes the rainbow equally distributed throughout
void rainbowCycle(uint8_t wait) {
  uint16_t i, j;
  for(j=0; j<256*5; j++) { // 5 cycles of all colors on wheel
    for(i=0; i< strip.numPixels(); i++) {
      strip.setPixelColor(i, Wheel(((i * 256 / strip.numPixels()) + j) & 255));
    }
    strip.show();
    delay(wait);
  }
}

//Theatre-style crawling lights.
void theaterChase(uint32_t c, uint8_t wait) {
  for (int j=0; j<10; j++) { //do 10 cycles of chasing
    for (int q=0; q < 3; q++) {
      for (int i=0; i < strip.numPixels(); i=i+3) {
        strip.setPixelColor(i+q, c); //turn every third pixel on
      }
      strip.show();
      delay(wait);
    }
  }
}
```

```

for (int i=0; i < strip.numPixels(); i=i+3) {
strip.setPixelColor(i+q, 0); //turn every third pixel off
}
}
}
}

//Theatre-style crawling lights with rainbow effect
void theaterChaseRainbow(uint8_t wait) {
for (int j=0; j < 256; j++) { // cycle all 256 colors in the wheel
for (int q=0; q < 3; q++) {
for (int i=0; i < strip.numPixels(); i=i+3) {
strip.setPixelColor(i+q, Wheel( (i+j) % 255)); //turn every third pixel on
}
strip.show();
delay(wait);
for (int i=0; i < strip.numPixels(); i=i+3) {
strip.setPixelColor(i+q, 0); //turn every third pixel off
}
}
}
}

// Input a value 0 to 255 to get a color value.
// The colours are a transition r - g - b - back to r.
uint32_t Wheel(byte WheelPos) {
WheelPos = 255 - WheelPos;
if(WheelPos < 85) {
return strip.Color(255 - WheelPos * 3, 0, WheelPos * 3);
}
if(WheelPos < 170) {
WheelPos -= 85;
return strip.Color(0, WheelPos * 3, 255 - WheelPos * 3);
}
WheelPos -= 170;
return strip.Color(WheelPos * 3, 255 - WheelPos * 3, 0);
}

```

### Project 9: To make a lamp

### 1. Project instruction

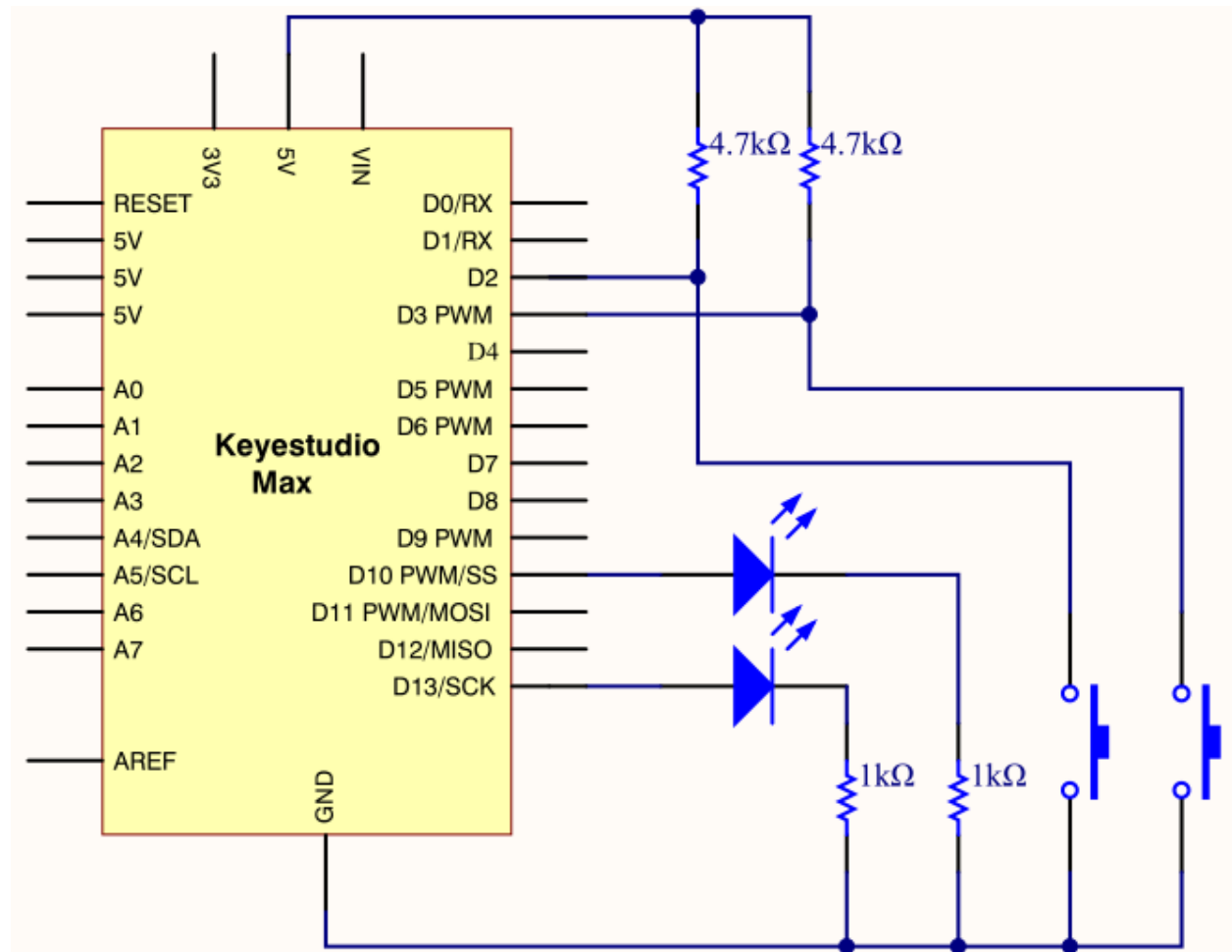
In previous lessons, we have known the output function of Arduino. In this experiment, we will make you have better understanding of I/O function through an LED experiment.

### 1. Project principle

The push button switch is usually used to turn on and off the control circuit, and it is a kind of control switch appliance that is widely used.

The push button switch can complete basic controls such as start, stop, forward and reverse rotation, speed change and interlock. Usually each push button switch has two pairs of contacts. Each pair of contacts consists of a NO contact and a NC contact. When the button is pressed, the two pairs of contacts act simultaneously, the NC contact is disconnected, and the NO contact is closed.

### 1. Project circuit



### 1. Project code

```
/*
```

```
keyestudio Max Development Board
```

```
Project 9
```

```
To make a lamp
```

```
http://www.keyestudio.com
```

```

*/
int ledpin1=13;//define digital 13
int ledpin2=10;//define digital 10
int buttonpin1=2;//define digital 2
int buttonpin2=3;//define digital 3
int val1;//define variable val1
int val2;//define variable val2
void setup()
{
  pinMode(ledpin1,OUTPUT);//define LED pin1 to output
  pinMode(ledpin2,OUTPUT);//define LED pin2 to output
  pinMode(buttonpin1,INPUT);//define button pin1 to input
  pinMode(buttonpin2,INPUT);//define button pin2 to input
}
void loop()
{
  val1=digitalRead(buttonpin1);//read the level of digital 2 and set to val
  val2=digitalRead(buttonpin2);//read the level of digital 3 and set to val
  if(val1==HIGH)//detect if the button is pressed, LED on when pressed
  {
    digitalWrite(ledpin1,LOW);
  }
  else
  {
    digitalWrite(ledpin1,HIGH);
  }
  if(val2==HIGH)//detect if the button is pressed, LED on when pressed
  {
    digitalWrite(ledpin2,LOW);
  }
  else
  {
    digitalWrite(ledpin2,HIGH);}
}

```

### 5. Project results:

This principle is applied widely in all kinds of circuit and household appliances, such as push switch and indicator.

### Project 10: To make a night light



#### 1. Project instruction

A photoresistor, relies on light to work. In the dark environment, it has high resistance. The stronger the light intensity is, the smaller its value is, thereby, we could check the light intensity through reading its resistance value.

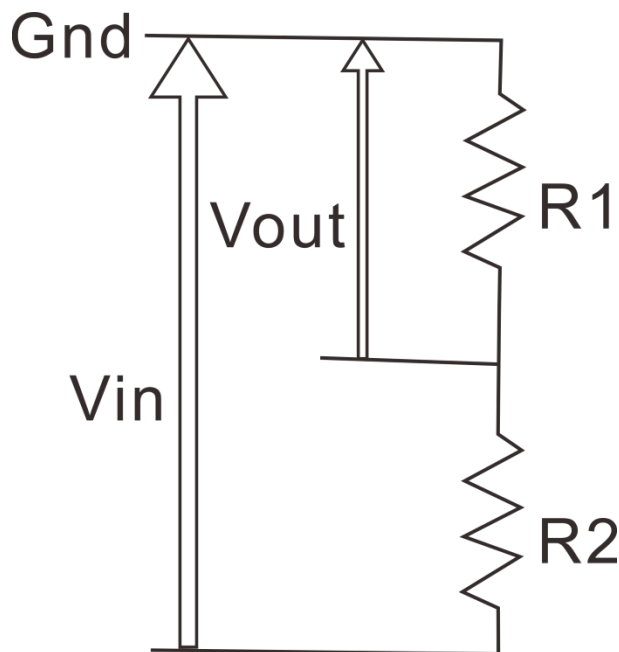
We will make a photo-sensitive light which will adjust its light intensity with the ambient environment.

#### 1. Project Principle

Photoresistor, a kind of special electronic component, can transfer light signals into electronic signals. It contains special light-conductive material and is sealed with plastic and glass packaging. As the light is irradiated on this kind of photo-conductive material, its the resistance value will get small rapidly.

The photoresistor has a very high resistance in a dark environment. The stronger the light, the lower the resistance value. As the resistance value at both ends decreases, the  $V_{out}$  voltage increases (the value read from the analog port also increases, and the value of the analog port 0~1023 corresponds to the voltage value of 0~5V).

Why does the voltage decrease? Then we need to explain through physics knowledge-the principle of partial pressure. Let us look at a typical voltage divider circuit and see how it works.



Input voltage  $V_{in}$ (5V) connected to two resistors, only measures the voltage of  $V_{out}$  on resistor  $R1$ . Its voltage is less than input voltage,



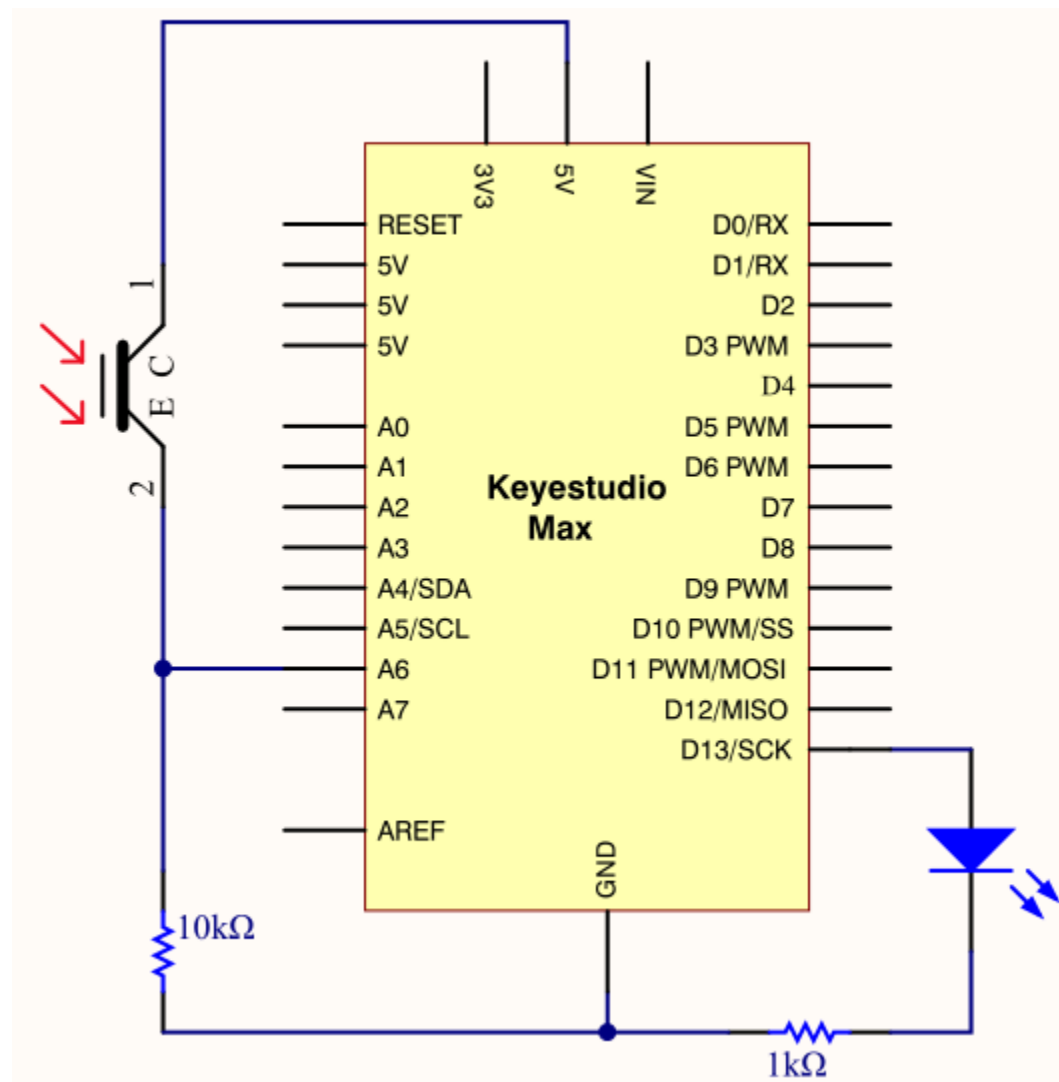
The formula of the voltage of  $V_{out}$  is as follows

$$V_{out} = \frac{R_1}{R_1 + R_2} \times V_{in}$$

$R_1$  represents 10k resistor,  $R_2$  is photoresistor. When  $R_2$  is in the dark environment, its value gets larger but  $V_{out}$ 's value is close to 0v. Once there is light, the value of  $R_2$  will get smaller and the value of  $V_{out}$  increases instead.

Judged from the above formula,  $R_1$  should be in 1k~10k, otherwise, the value of  $V_{out}$  is not apparent.

#### 1. Project circuit



#### 1. Project code

```
/*
```

```
keystudio Max Development Board
```

```
Project10
```

To make a night light

<http://www.keyestudio.com>

\*/

```
int LED = 13; // set LED to pin 13
```

```
int val = 0; //set analog pin 0 to read the voltage value of photosensitive diode
```

```
void setup(){
```

```
  pinMode(LED,OUTPUT); // set LED to output
```

```
  Serial.begin(9600); // set serial baud rate to 9600
```

```
}
```

```
void loop(){
```

```
  val = analogRead(A6); // read the voltage value 0~1023
```

```
  Serial.println(val); // check the change of voltage value in serial
```

```
  if(val>300)
```

```
  { // once exceed the set value, LED will be off
```

```
    digitalWrite(LED,LOW);
```

```
  }else
```

```
  { // otherwise, LED is on
```

```
    digitalWrite(LED,HIGH);
```

```
}
```

```
  delay(10); //delay in 10ms
```

```
}
```

### 1. Project results:

After wiring up hardware, open Arduino IDE to input code and LED will be on.

Turn on flashlight and point at photoresistor, LED is off. Once you move flashlight away, LED will be on again.

Project expansion:

For the test code, we read analog value of photoresistive diode through analog port.

Once the light is irradiated, the analog value read decreases. As the analog value is more than 300, LED will be off, on the contrary, the analog is less than and equivalent to 300, LED on

## Project 11: Voice-activated

### 1. Project instruction

Akin to a microphone, sound sensor can receive sound waves and show the vibration image of sound but not to measure the intensity of noise.

It has a built-in condenser electret microphone that is sensitive to sound.

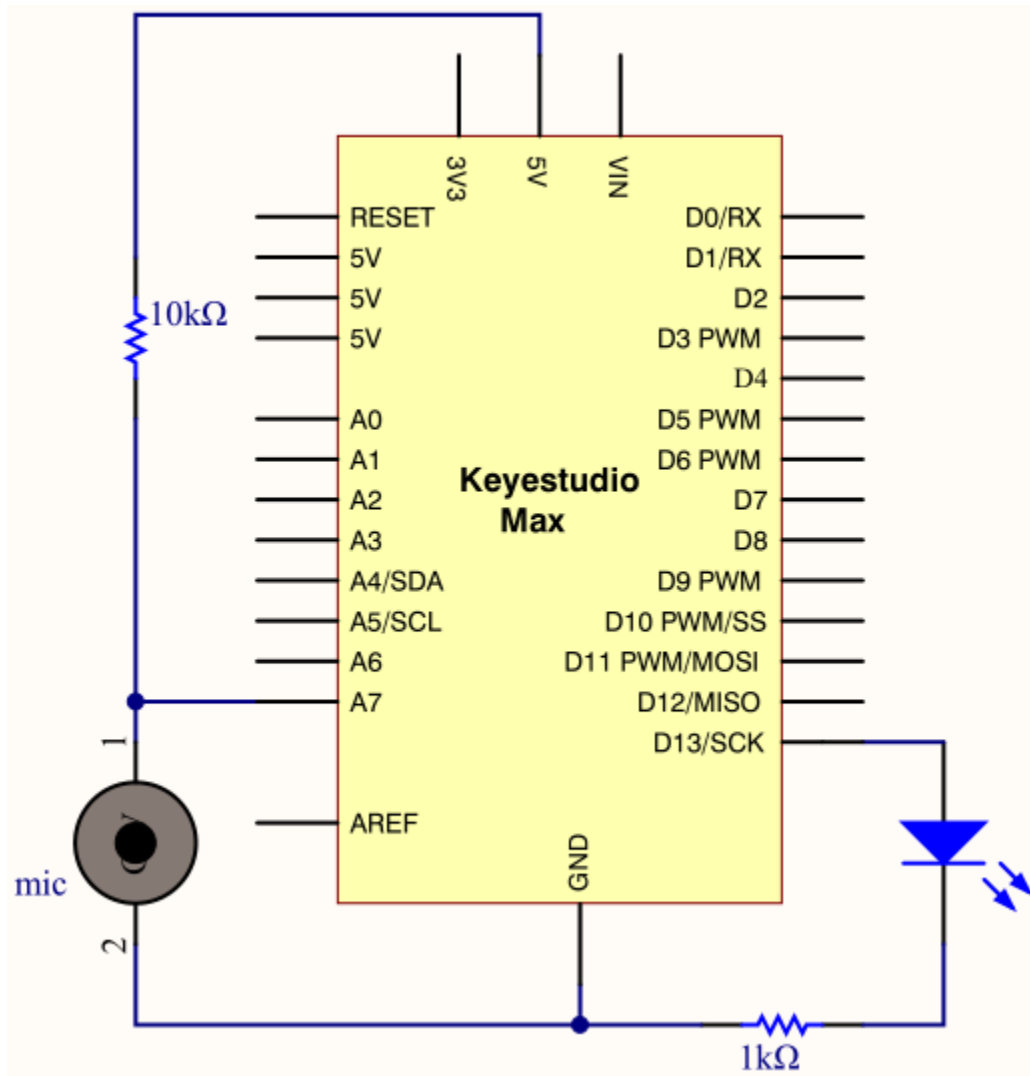
The sound waves vibrate the electret film in the microphone, which causes the change of capacitance, and generates a correspondingly changed tiny voltage. This voltage is then converted into a 0-5V voltage, received by the data collector after A/D conversion, and sent to Arduino UNO.

### 1. Project Principle

- Use 5v DC power supply (working voltage 3.3V-5V)
- With analog output AO, real-time microphone voltage signal output
- High sensitivity, electret condenser microphone (ECM) sensor
- Can detect the sound intensity of ambient environment.

Note: This sensor can only identify the presence or absence of sound (according to the principle of vibration) and cannot identify the size of the sound or the sound of a specific frequency.

#### 1. Project circuit



#### 1. Project code

/\*

keyestudio Max Development Board

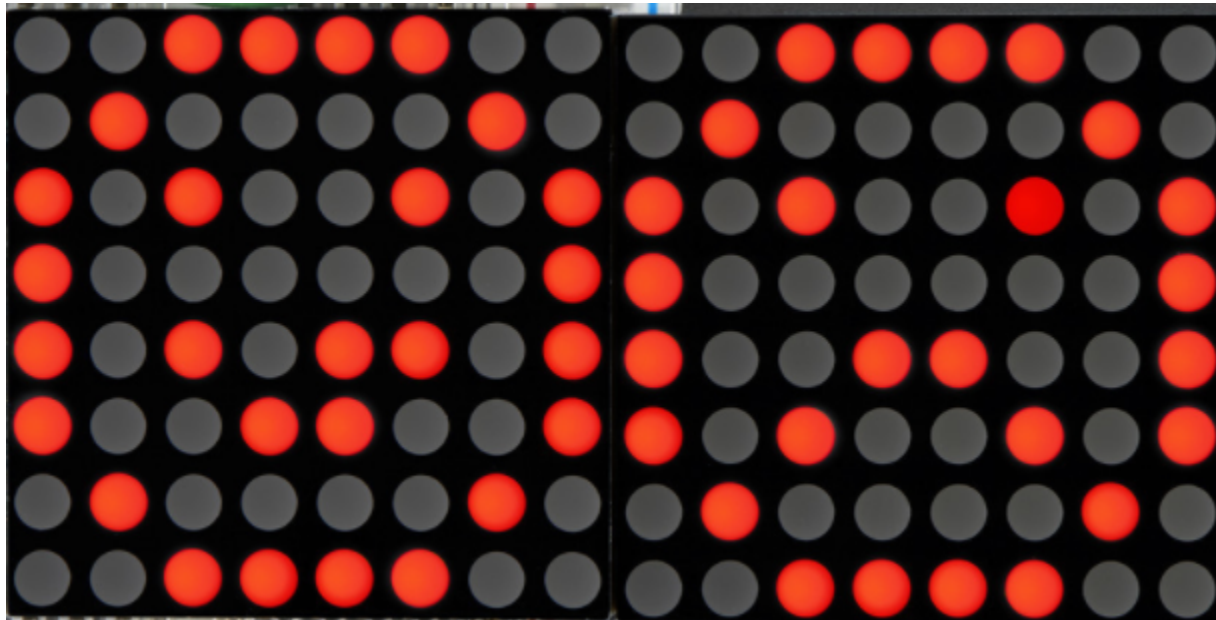
Project11

Voice-activated

<http://www.keyestudio.com>

```
*/  
const int SOUND_PIN = A7;  
const int LED_PIN = 13;  
void setup() {  
  pinMode(LED_PIN, OUTPUT);  
  Serial.begin(9600);  
}  
void loop() {  
  // read A7 analog signal  
  int value = analogRead(SOUND_PIN);  
  // output signal value  
  Serial.println(value);  
  if(value > 5){  
    // more than threshold, pin 13 is high level and LED on digitalWrite(LED_PIN, HIGH);  
    delay(100);  
  }else{  
    // less than threshold, pin 13 is low level and LED off  
    digitalWrite(LED_PIN, LOW);  
    delay(100);  
  }  
}
```

### Project 12: 8\*16 dot matrix-turn on a led



#### 1. Project instruction

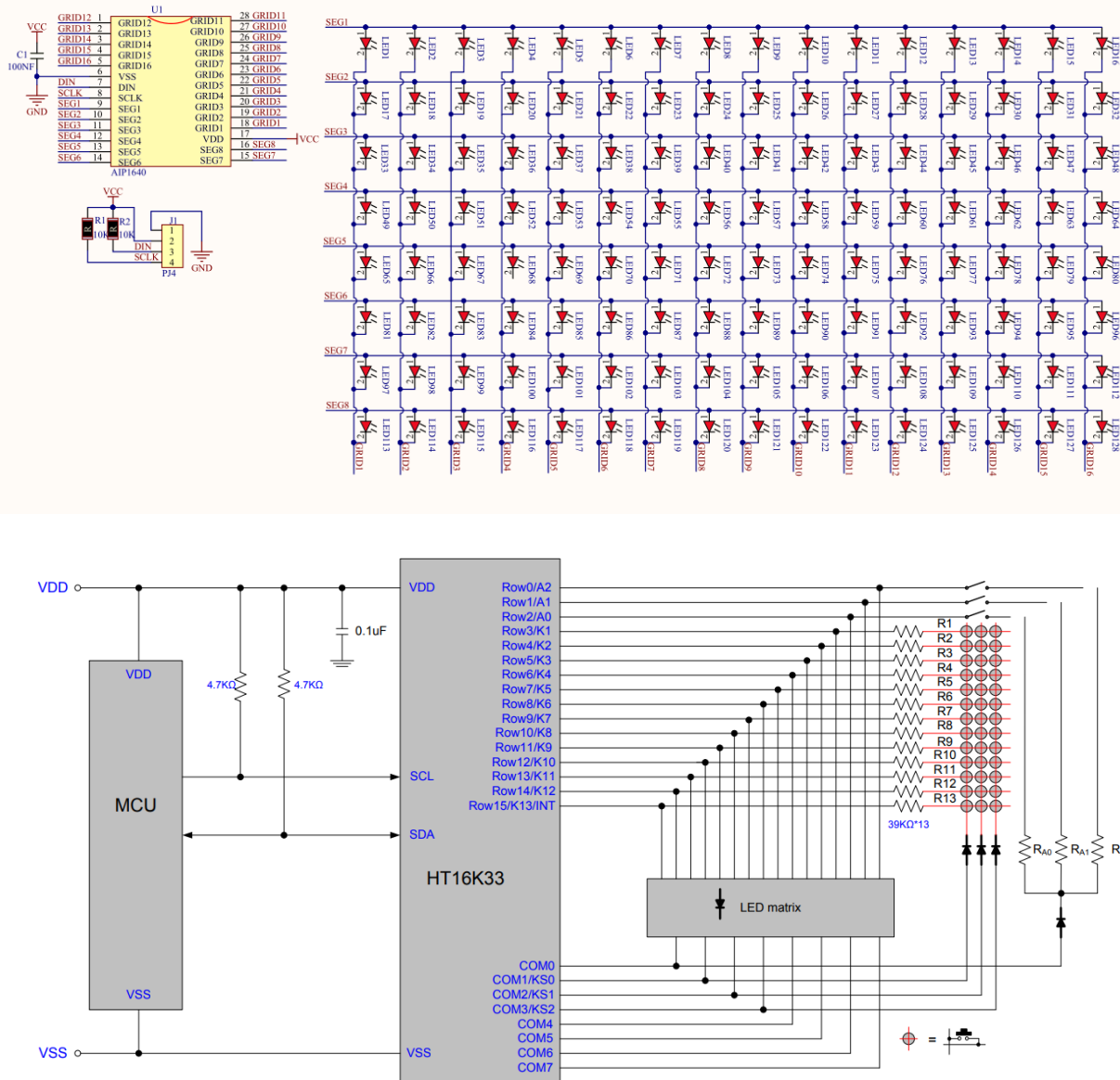
The 8\*16 dot matrix integrated on the MAX board uses I2C communication. As long as 2 signal pins, up to 128 LEDs can be controlled, and interesting displays such as numbers, characters, and graphics can be displayed.

The 8\*16 dot matrix is also great for making scrolling displays or small video displays. In our example, we set it up so words flow from one matrix to the other. Can be used for advertising display.

The 8\*16 dot matrix uses a HT16K33 driver chip that does all the heavy lifting for you. This chip has a built in clock so they multiplex the display. Through a simple I2C interface, you can control the chip work and drive 8\*16 Dot matrix screen.

Now we are about to start many 8\*16 dot matrix projects, let's first turn on a light on the dot matrix.

## 1. Project circuit

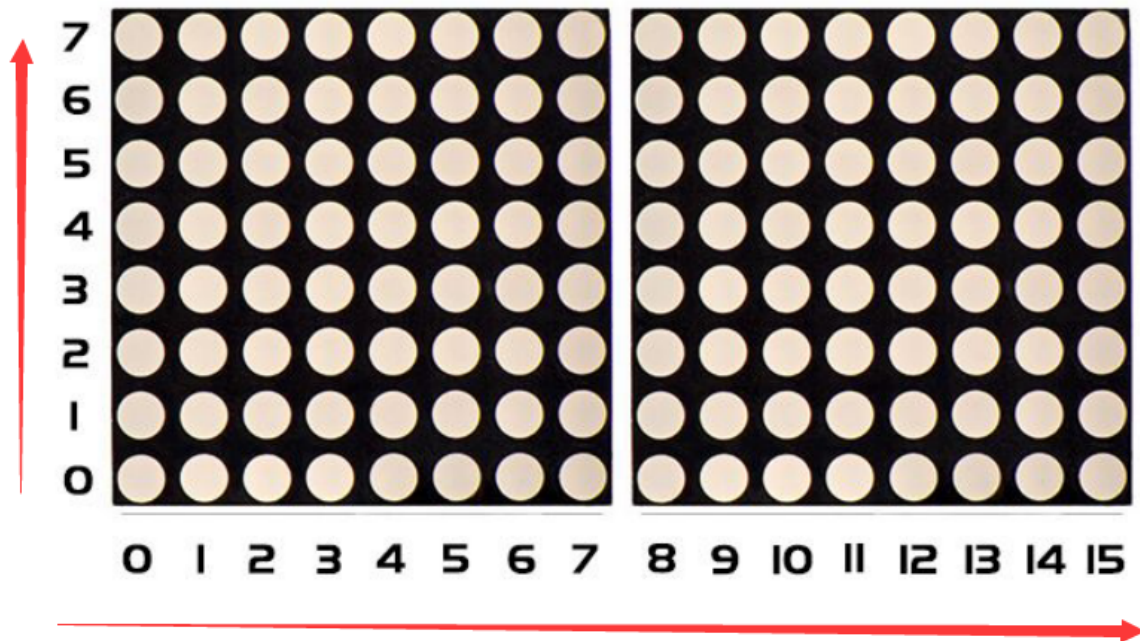


## 1. Project Principle

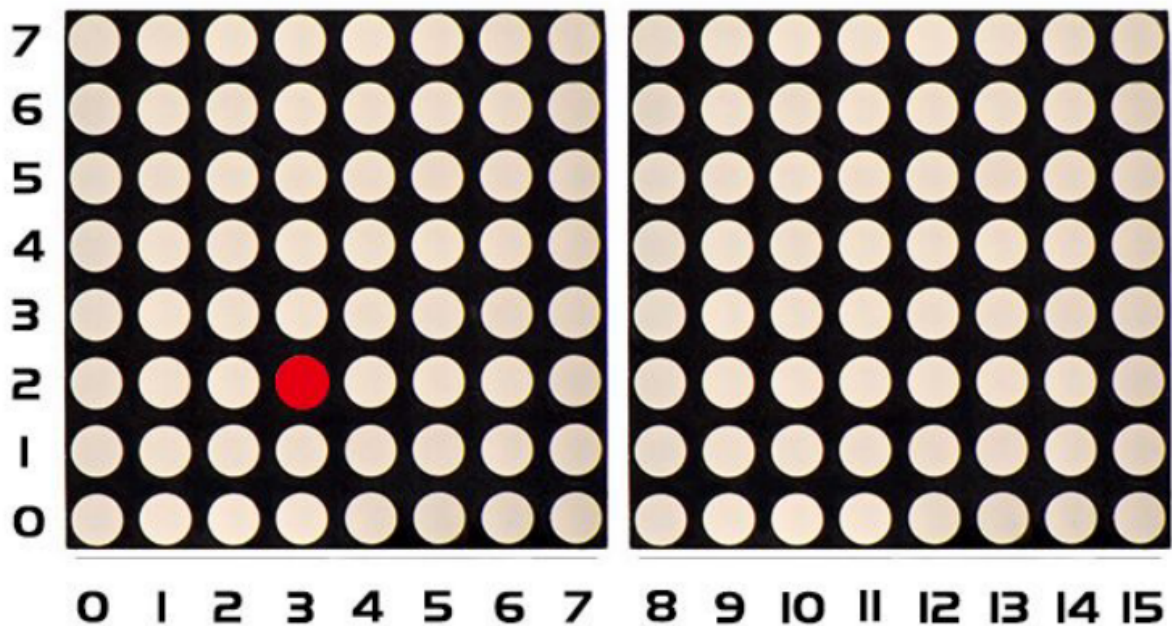
<http://www.google.com/about>

The display principle of 8\*16 dot matrix is very simple. The single-chip microcomputer can drive 8\*16 dot matrix

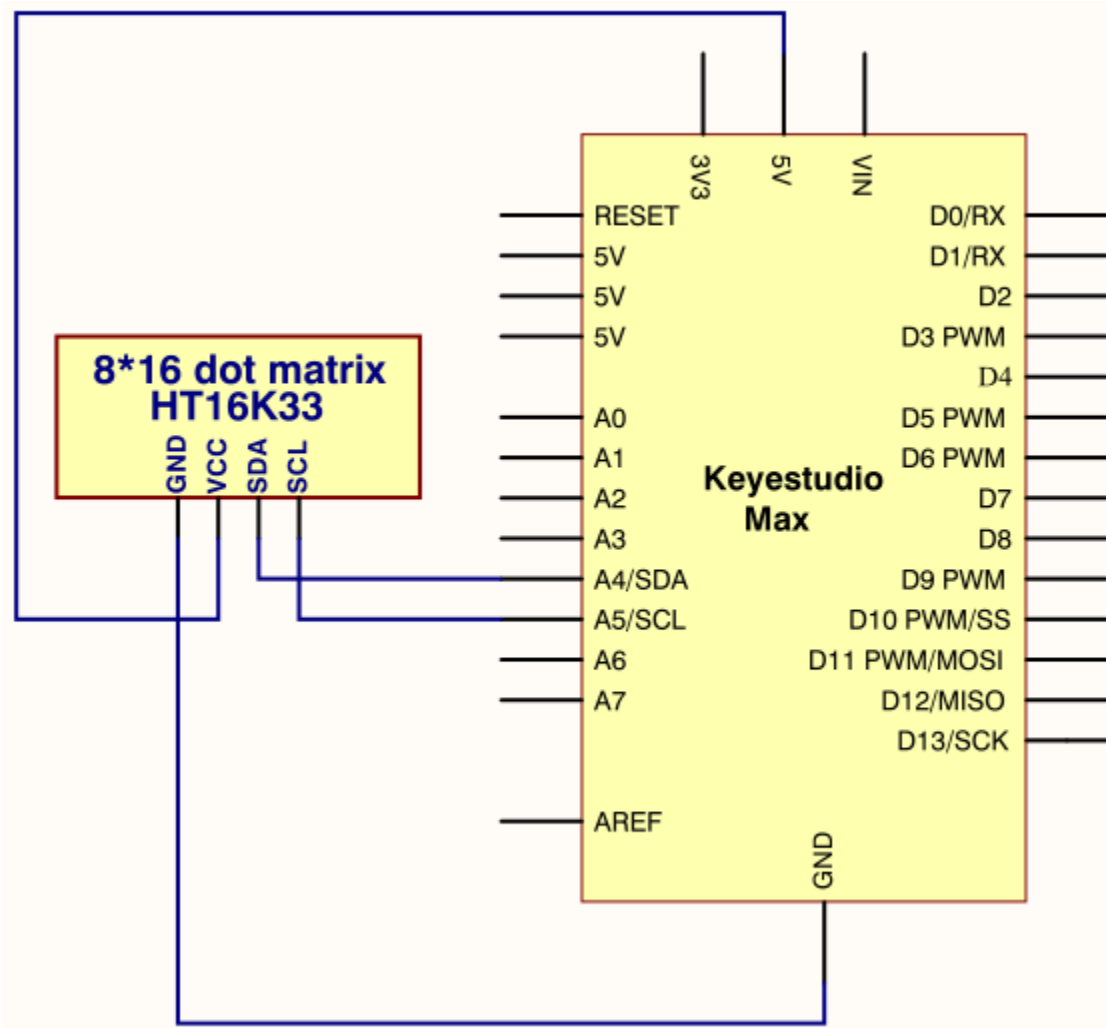
through HT16K33 chip according to above circuit diagram. The arrangement of the dot matrix is as follows: 8 LED lights in the horizontal direction and 16 LED lights in the vertical direction. The total number of lights is 128. In order to quickly find the position of the light, we number the horizontal lights from 0-7, and the vertical lights from 0-15.



How to turn on a LED on dot matrix?



According to above picture, the coordinate of red point is 2,3, and we put coordinate value in the following code.



### 1. Project code

```

/*
keyestudio Max Development Board
Project 12
8*16 dot matrix-turn on a led
http://www.keyestudio.com
*/

#include <Wire.h>
#include "Keyestudio_LEDBackpack.h"
#include "Keyestudio_GFX.h"
Keyestudio_8x16matrix matrix = Keyestudio_8x16matrix();
void setup()
{
matrix.begin(0x70); // pass in the address

```



```
}  
void loop()  
{  
  matrix.clear(); // clear display  
  matrix.drawPixel(2, 3, LED_ON);  
  matrix.writeDisplay(); // write the changes we just made to the display  
}
```

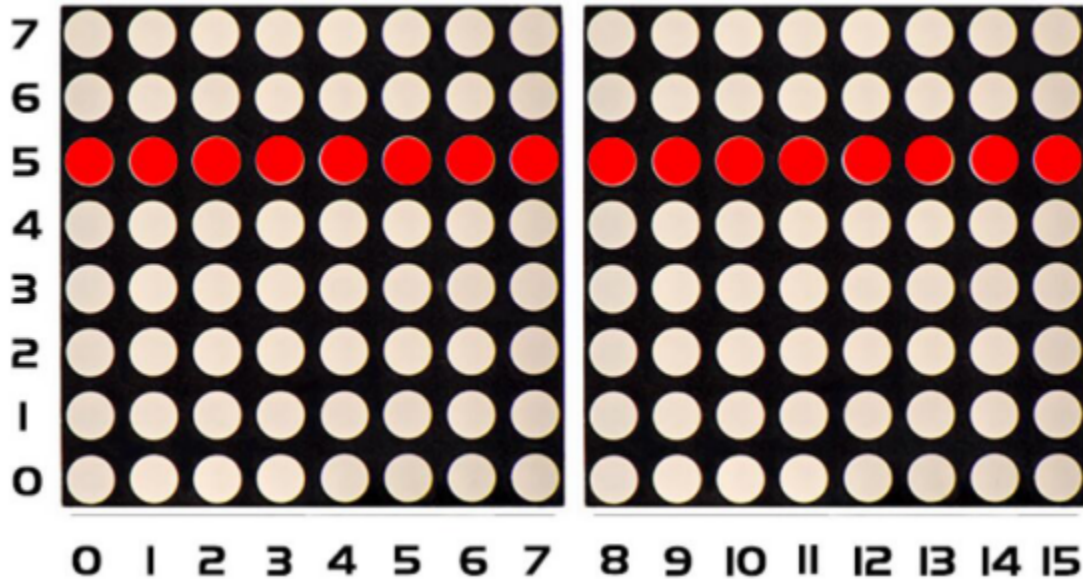
### Project 13: 8\*16 dot matrix-turn on one line

#### 1. project instruction

In previous program, we light up a coordinate point.

In this project, we will light up a row of LEDs.

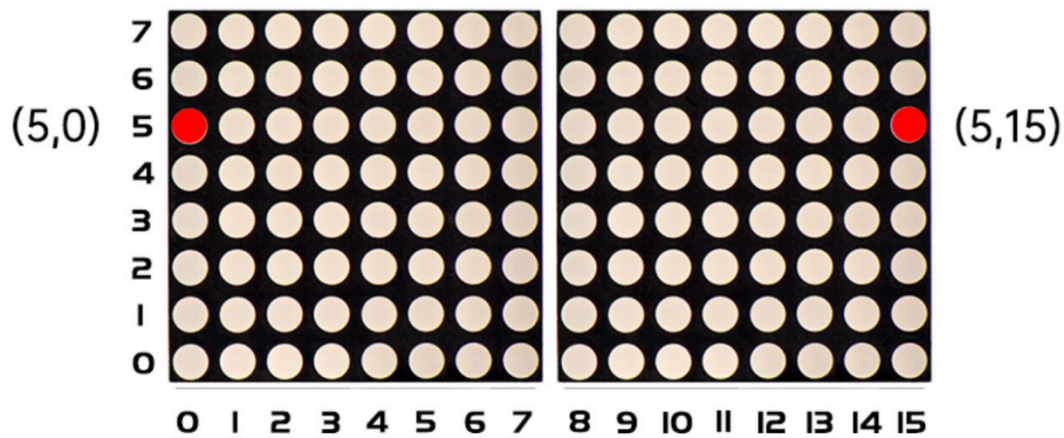
#### 1. Project Principle



To light a row of LEDs, we need to make use of function `matrix.drawLine`.

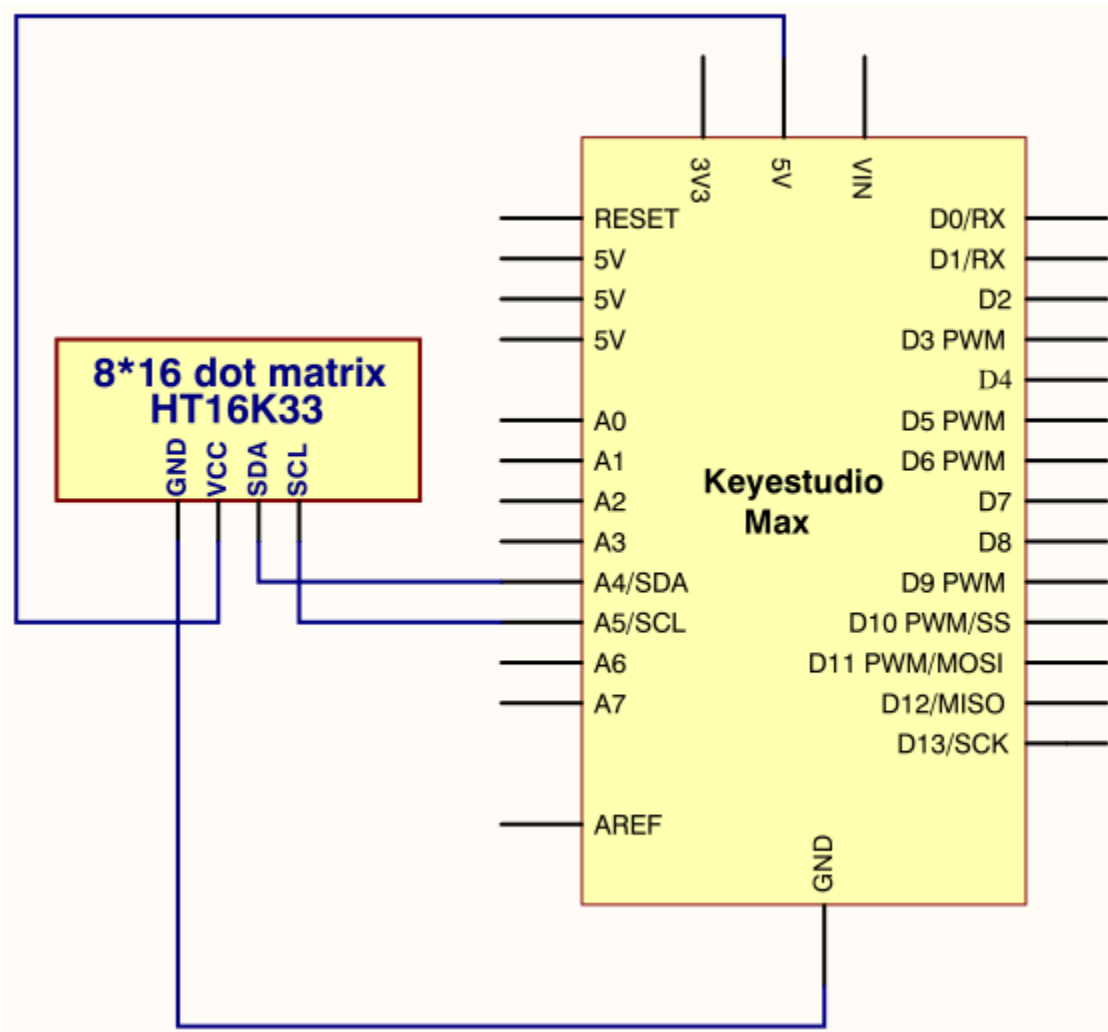
Lighting up the start and terminal point on a row of LEDs indicates all LEDs on this row are on, therefore, we need to decide the start and terminal point of this row of LEDs





Seeing from the above picture, the start point locates (5,0) and the end one is (5,15). Next, we place the value in the below code

#### 1. Project circuit



#### 1. Project code

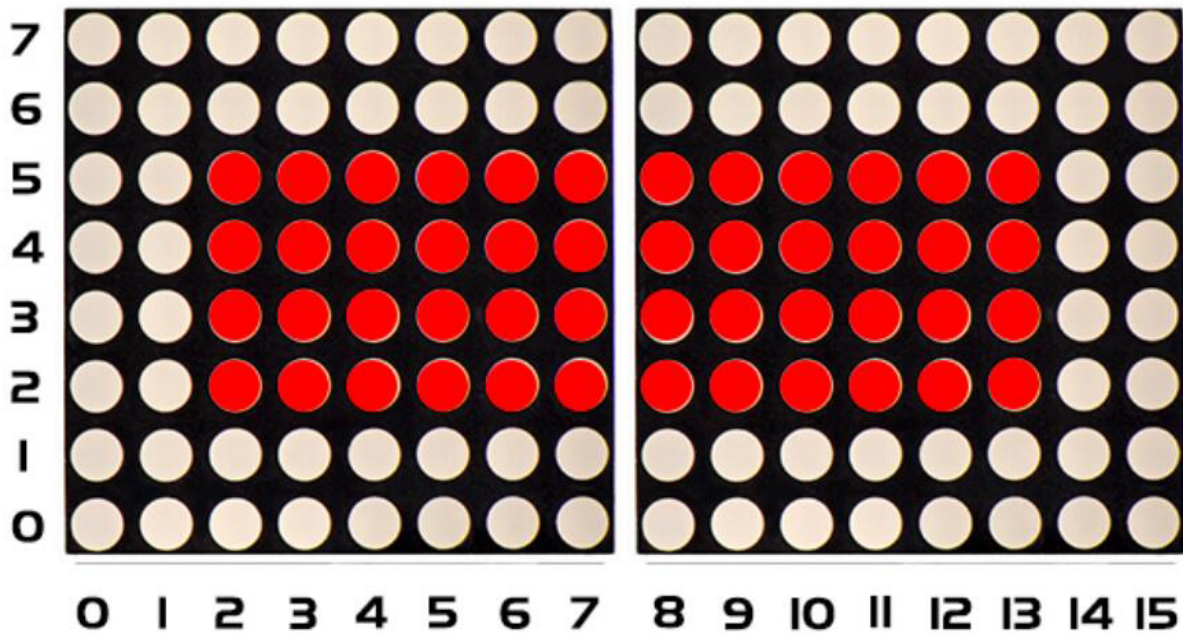
```
/*
keyestudio Max Development Board
Project 13
8*16 dot matrix-turn on one line
http://www.keyestudio.com
*/
#include <Wire.h>
#include "Keyestudio_LEDBackpack.h"
#include "Keyestudio_GFX.h"
Keyestudio_8x16matrix matrix = Keyestudio_8x16matrix();
void setup()
{
matrix.begin(0x70); // pass in the address
}
void loop()
{
matrix.clear();
matrix.drawLine(5,0, 5,15, LED_ON);
matrix.writeDisplay(); // write the changes we just made to the display
}
```

### **Project 14: 8\*16 dot matrix-draw a rectangle**

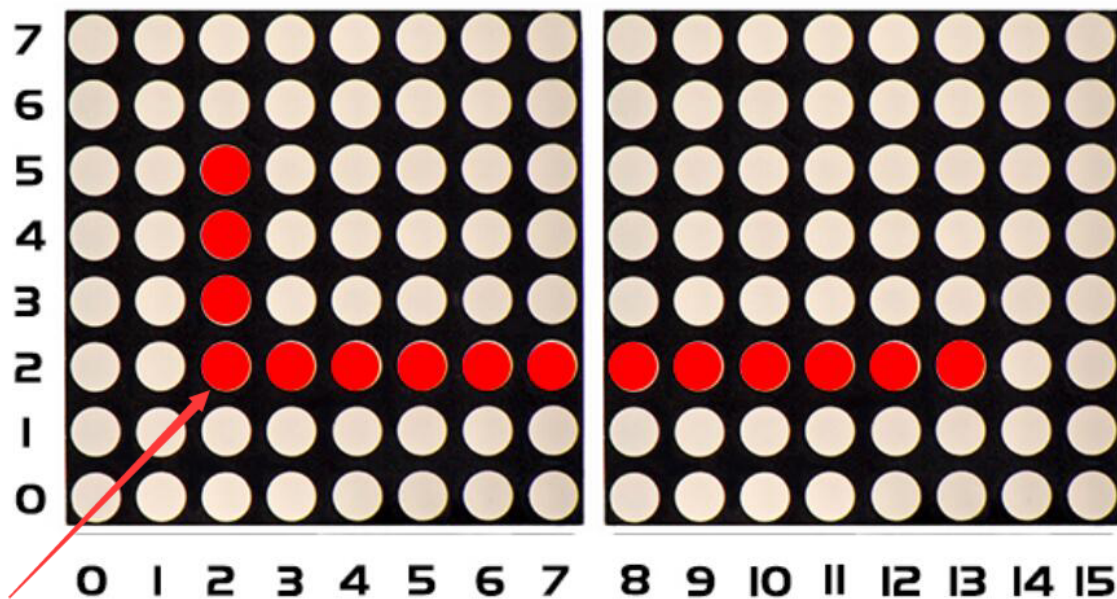
#### **1. Project instruction**

I believe that you get more interested in our programs. In this part, we will display a rectangle on 8\*16 dot matrix.

#### **1. Project Principle**



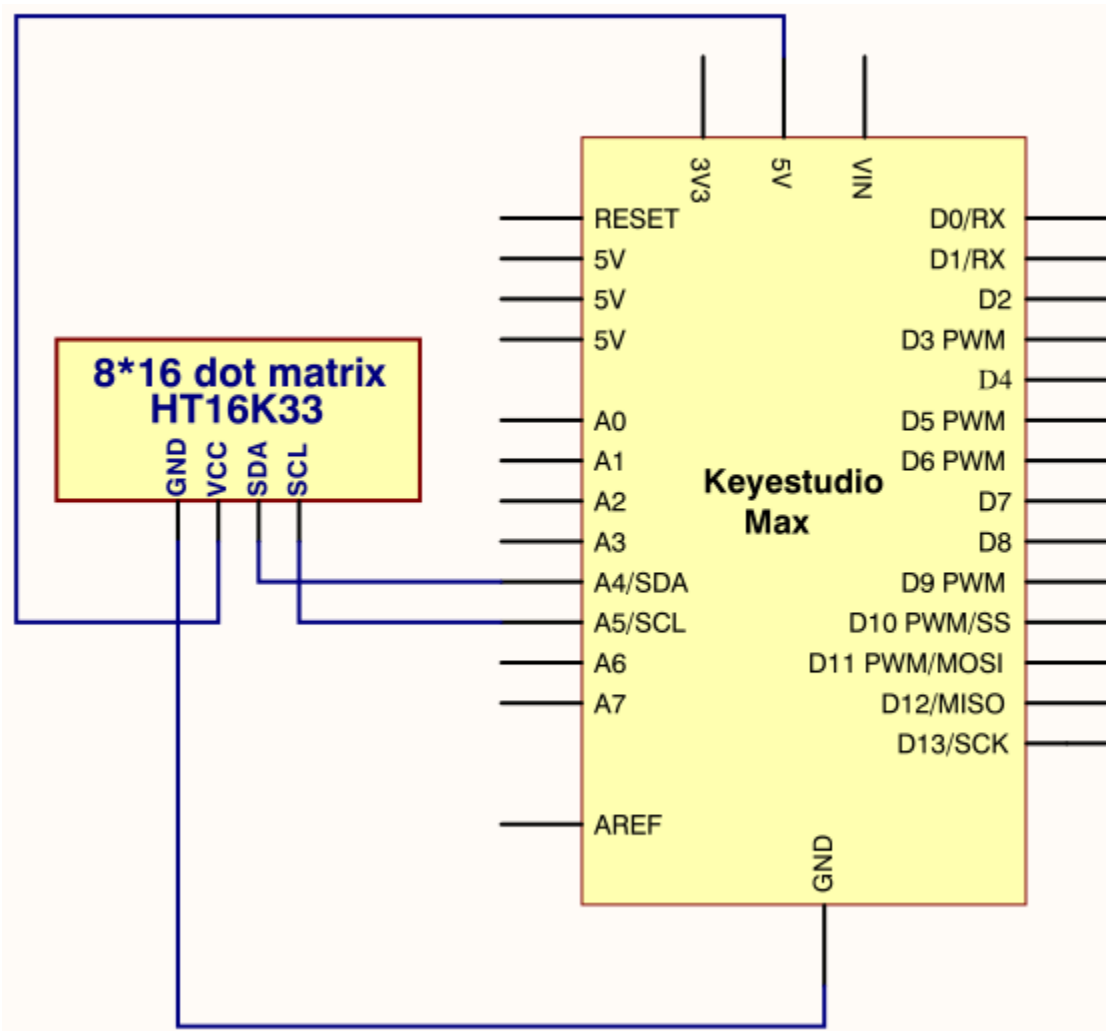
So, how to show a rectangle? We need use a function `matrix.fillRect`, which implies that you have to make sure a point on dot matrix so as to draw a square or rectangle, as shown below:



For example, we will draw a rectangle 4 in width and length in 12 from the point (2,2). What we need to do is put the value of width and length behind the point (2,2). Therefore, we get the code `matrix.fillRect(2,2,4,4)`.

If display a square with 4 in width and length, its code are supposed to be `matrix.fillRect(2,2,4,4)`.

#### 1. Project circuit



### 1. Project code

```

/*
keyestudio Max Development Board
Project 14
8*16 dot matrix-turn on a rectangle
http://www.keyestudio.com
*/
#include <Wire.h>
#include "Keyestudio_LEDBackpack.h"
#include "Keyestudio_GFX.h"
Keyestudio_8x16matrix matrix = Keyestudio_8x16matrix();
void setup()
{
matrix.begin(0x70); // pass in the address

```

```

}
void loop()
{
  matrix.clear();
  matrix.fillRect(2,2, 4,12, LED_ON);
  matrix.writeDisplay(); // write the changes we just made to the display
}

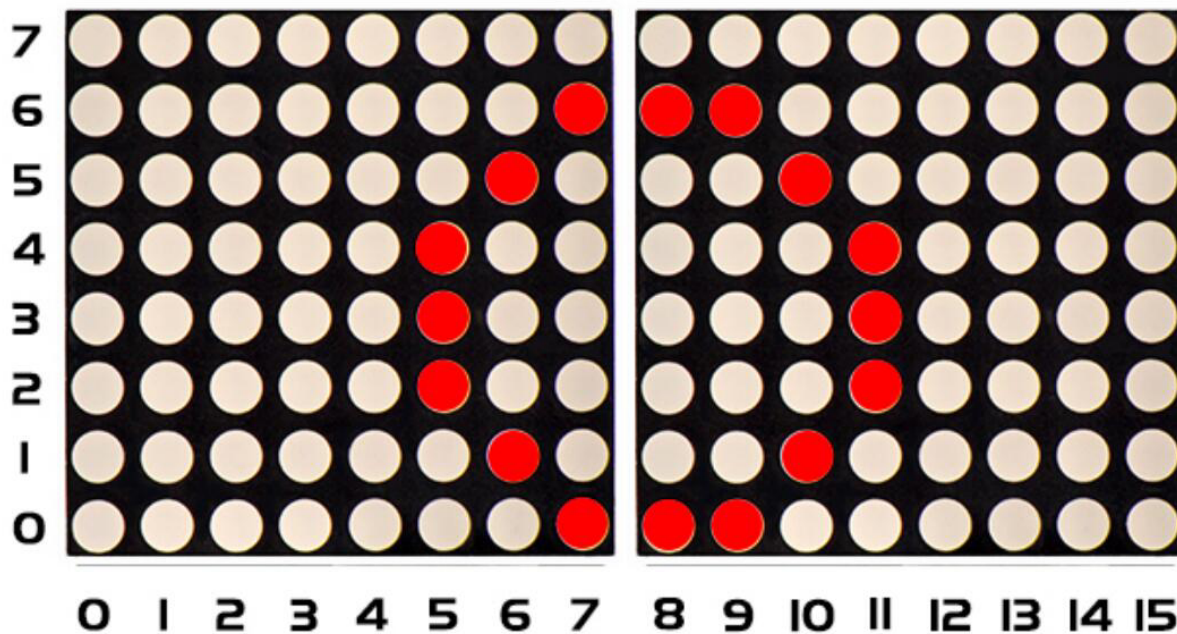
```

### Project 15: 8\*16 dot matrix-produce a circle

#### 1. Project instruction

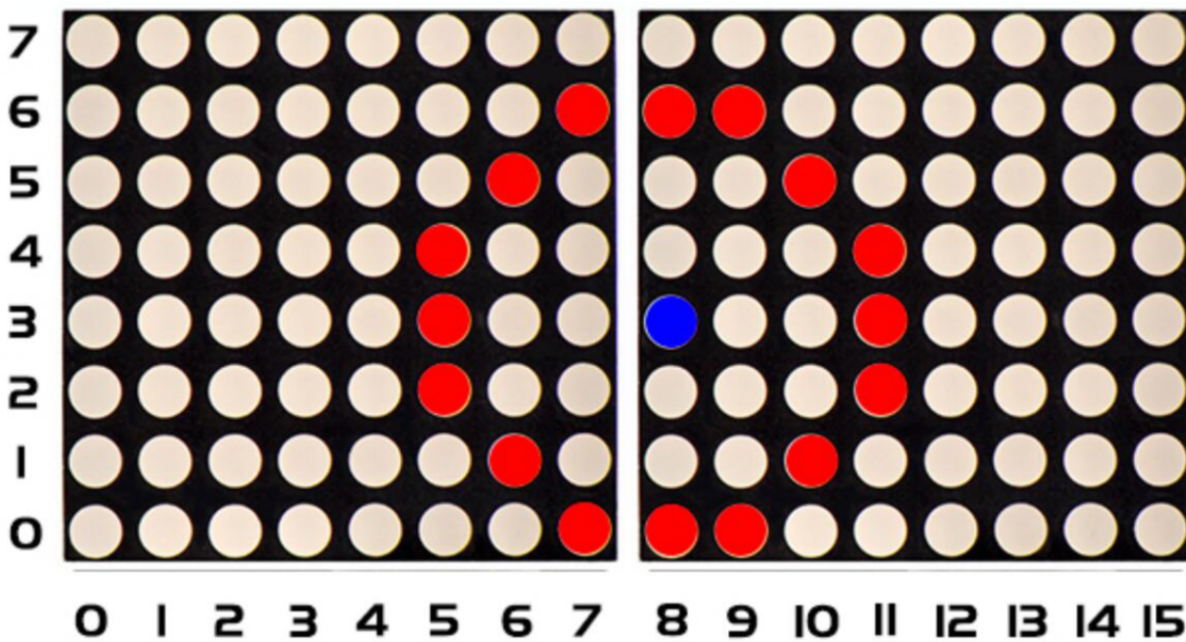
In this program, we will generate a circle on 8\*16 dot matrix.

#### 1. Project Principle



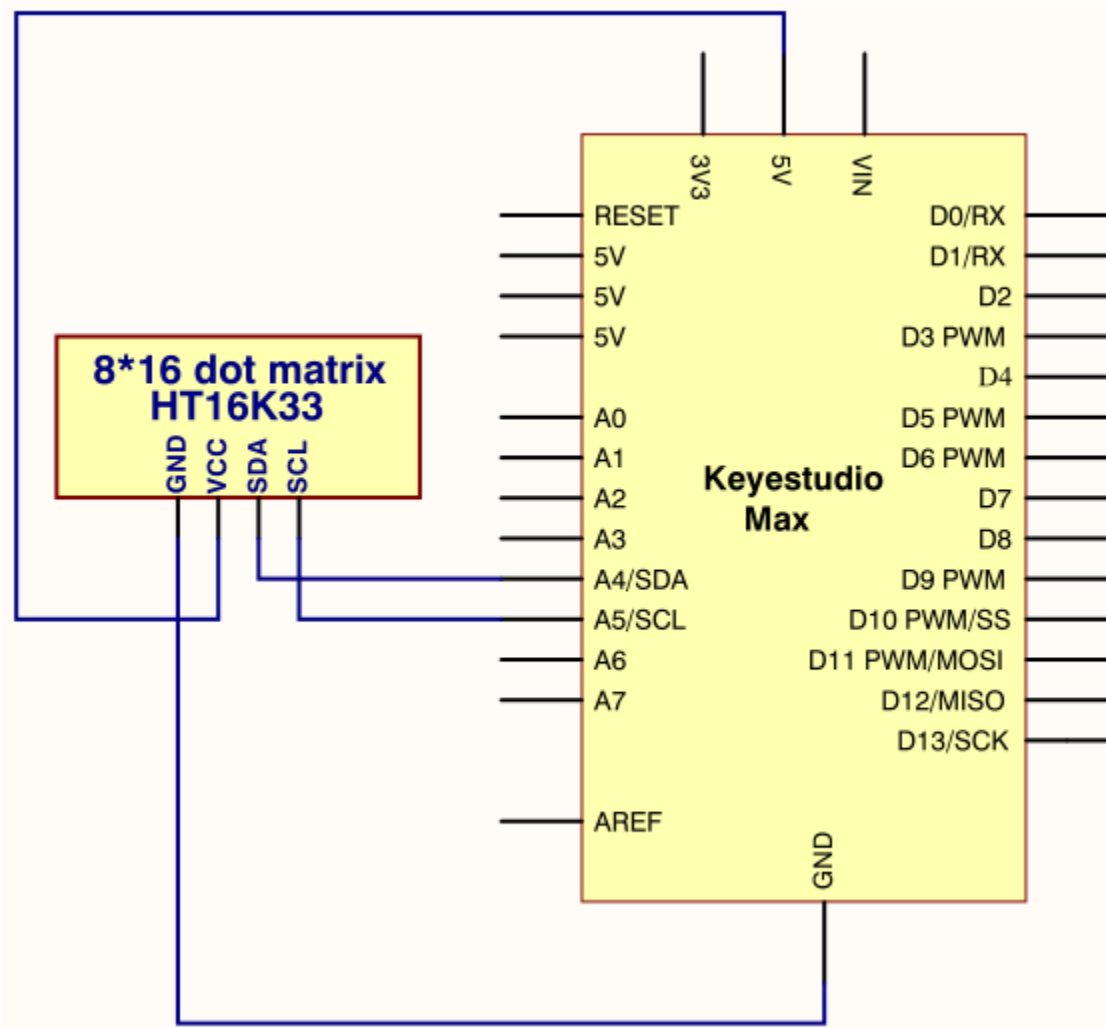
To show a circle with a radius of 3, we could use a function `matrix.drawCircle`. Select a central point to decide its coordinate and check radius value, as shown below:





The coordinate of center point of circle is (3,8), its radius is 3. We could put value (3,8, 3), in the following code. If we want to draw a circle with a constant center point and a radius of 2, we just need to change the value behind coordinate of center point into 2.

1. **Project circuit**



### 1. Project code

```

/*
keyestudio Max Development Board
Project 15
8*16 dot matrix-turn on a circle
http://www.keyestudio.com
*/

#include <Wire.h>
#include "Keyestudio_LEDBackpack.h"
#include "Keyestudio_GFX.h"
Keyestudio_8x16matrix matrix = Keyestudio_8x16matrix();
void setup()
{
matrix.begin(0x70); // pass in the address

```

```
}  
void loop() {  
  matrix.clear();  
  matrix.drawCircle(3,8, 3, LED_ON);  
  matrix.writeDisplay(); // write the changes we just made to the display  
  delay(500);  
}
```

### Project 16: 8\*16 dot matrix-display text and numbers

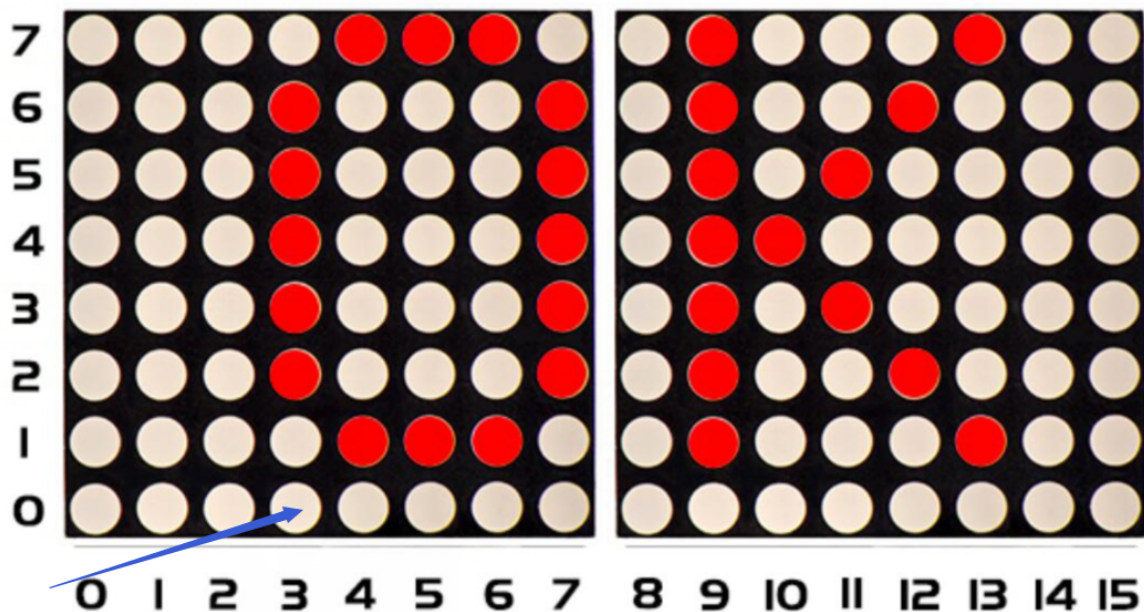
#### 1. Project instruction

In this project, we will make 8\*16 dot matrix produce some numbers and texts, like advertisement board at bus station.

#### 1. Project Principle

This program is pretty easy. We only need to confirm the coordinate of texts and numbers. Let's conduct an experiment to illustrate.

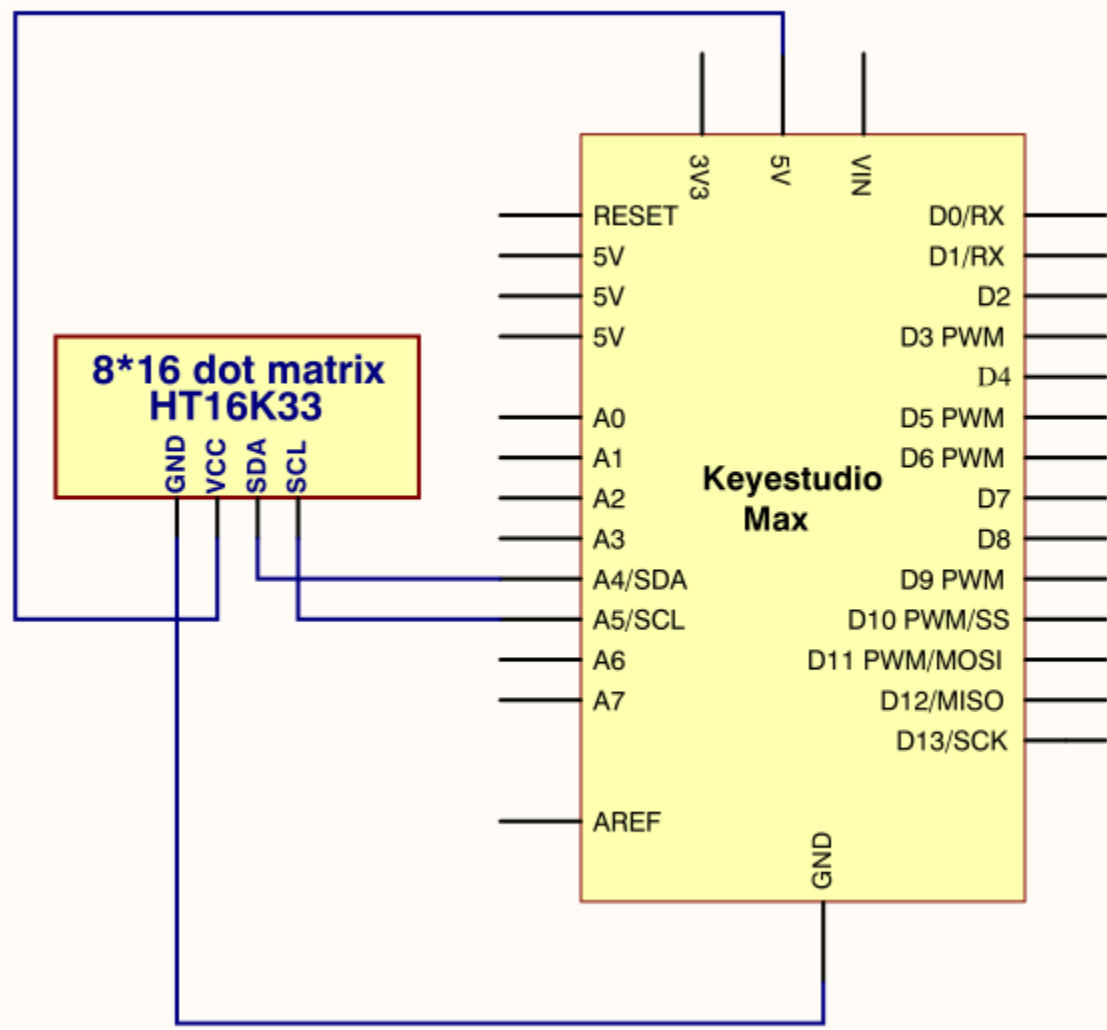
We set the display area to start from number 3 on horizontal coordinate and number zero on vertical coordinate, as shown below



We add the texts to be shown in the code `matrix.print` after getting the coordinate of text.

Note: numbers can be shown directly, but texts need adding double quotes.





### 1. Project code

```

/*
keyestudio Max Development Board
Project 16
8*16 dot matrix-display text and numbers
http://www.keyestudio.com
*/

#include <Wire.h>
#include "Keyestudio_LEDBackpack.h"
#include "Keyestudio_GFX.h"
Keyestudio_8x16matrix matrix = Keyestudio_8x16matrix();
void setup()
{
matrix.begin(0x70); // pass in the address

```

```
}  
void loop() {  
  matrix.setTextSize(1);// set size of text  
  matrix.setTextWrap(false); // we dont want text to wrap so it scrolls nicely  
  matrix.setTextColor(LED_ON);  
  matrix.setRotation(1);// show rotation  
  matrix.clear();  
  matrix.setCursor(3,0);//set the coordinate of text  
  matrix.print("OK");//show OK  
  matrix.writeDisplay();  
  delay(1000);  
}
```

### Project 17: 8\*16 dot matrix-display images

#### 1. Project instruction

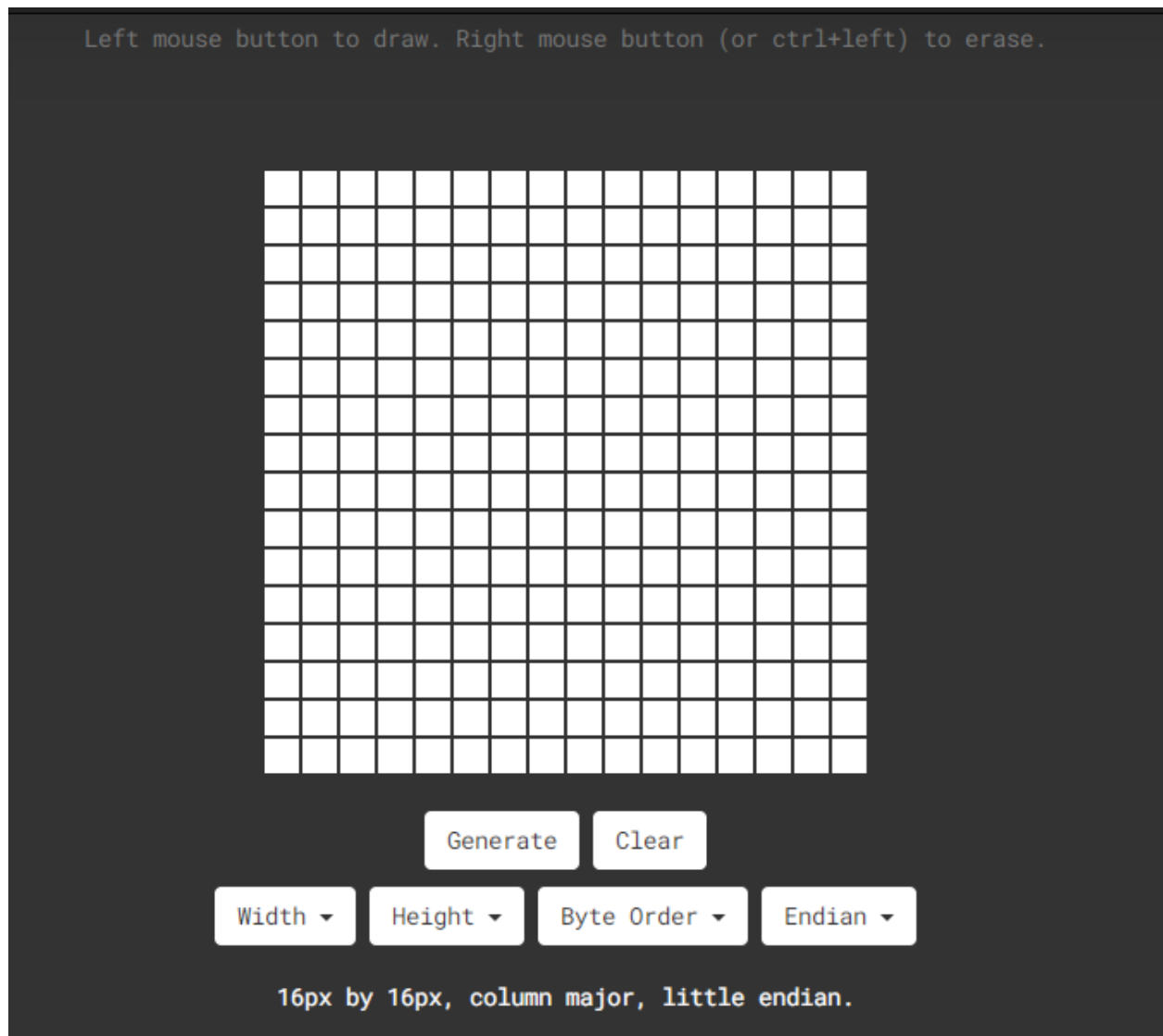
A large number of display devices, like cellphone, computer screen and advertisement board, are made of tiny light-emitting unit. The 8\*16 dot matrix aren't able to show exquisite and high pixel pictures but some cute and customized heart-shape picture, smiley face and so on, even though consisting of 128 pcs light-emitting units.

#### 2. Project Principle

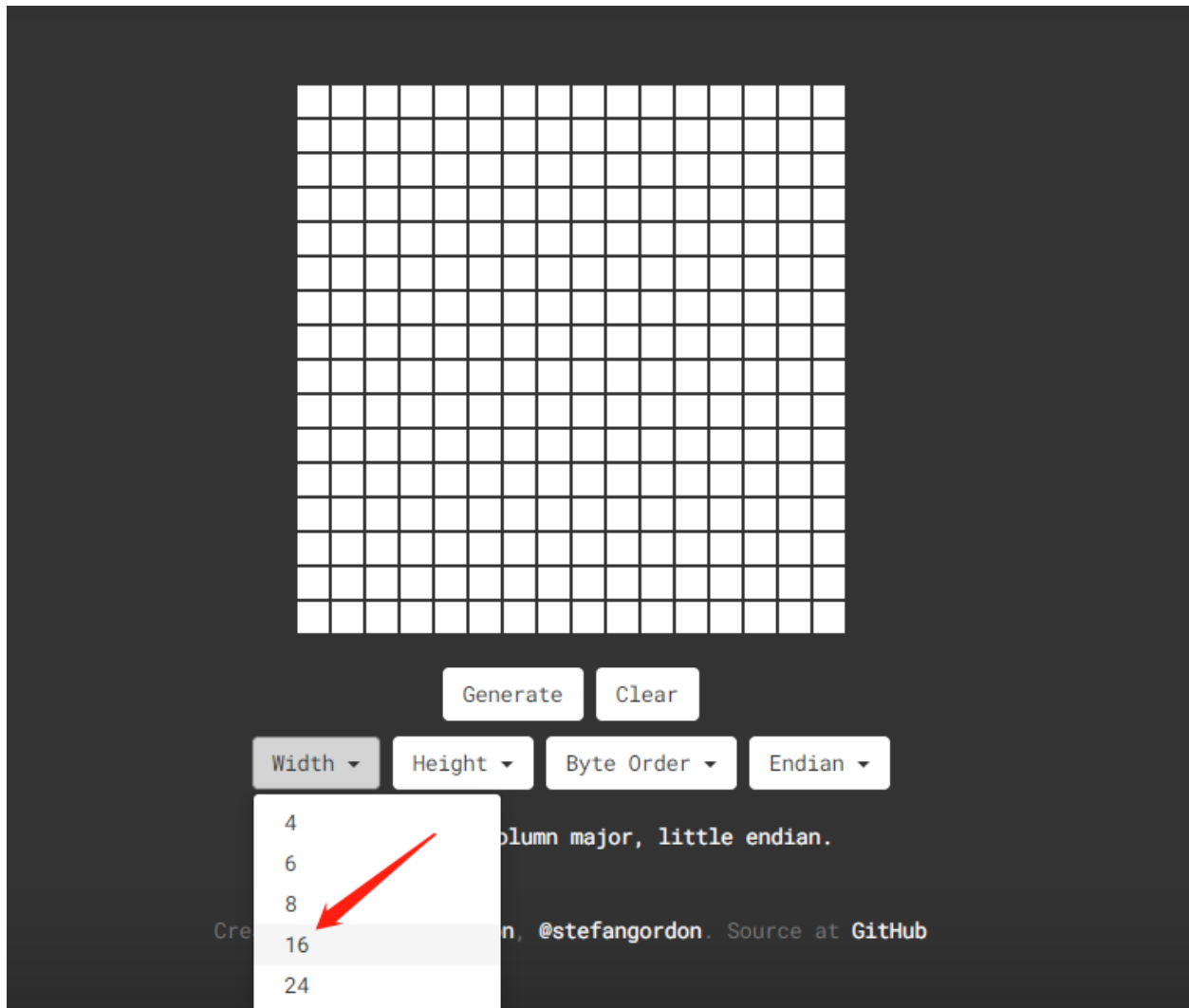
Introduction for Modulus Tool

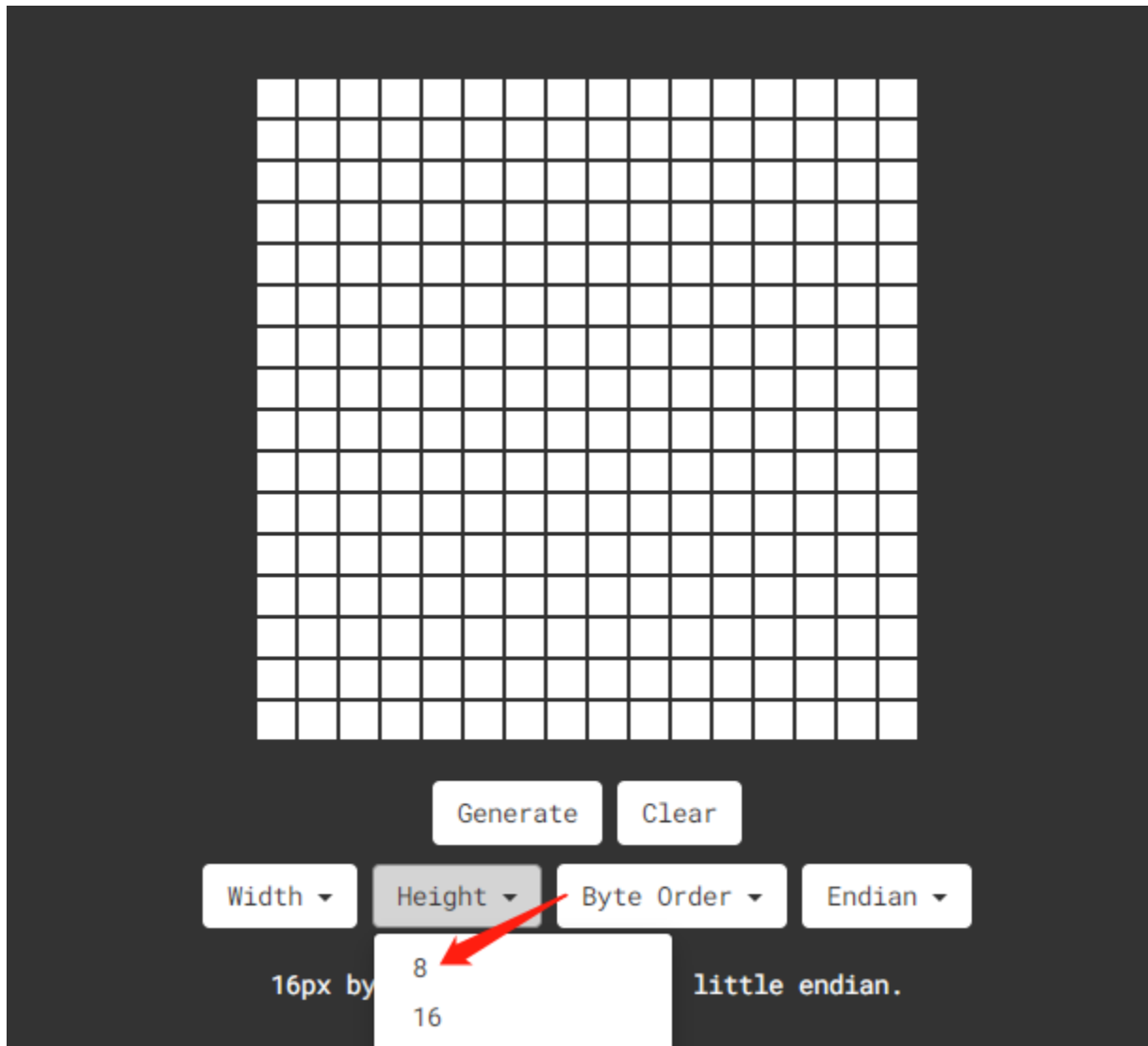
The online version of dot matrix modulus tool:

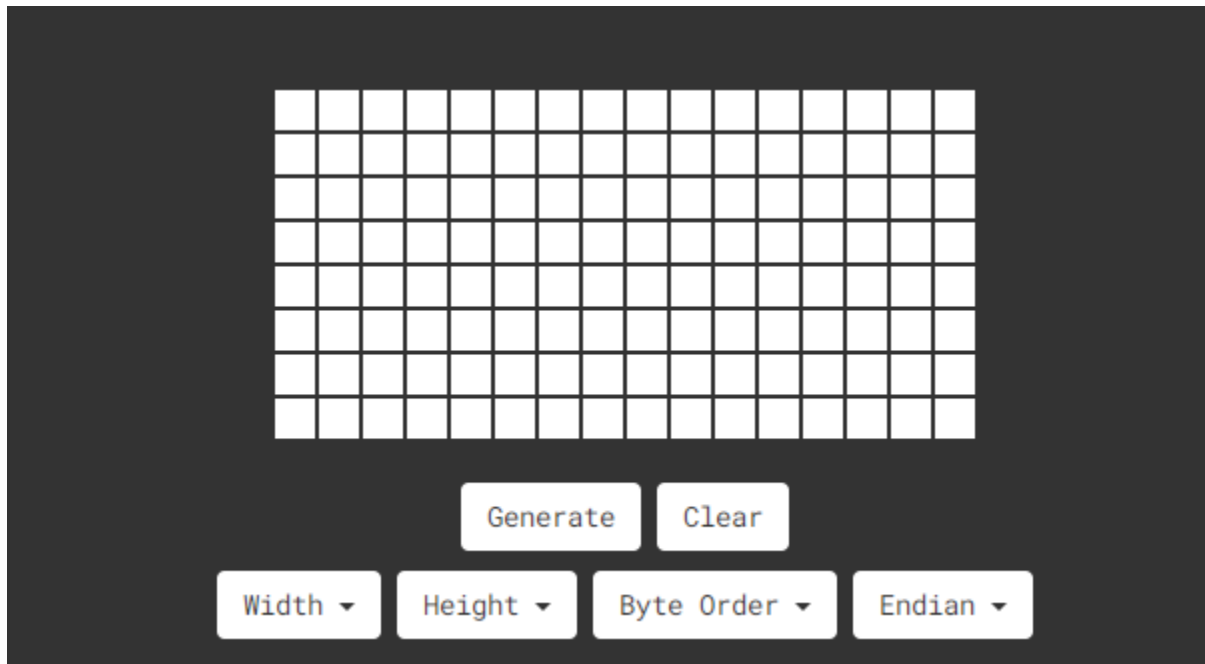
<http://dotmatrixtool.com/#>



The dot matrix is 8\*16 in this project, thereby, set the height to 8, width to 16, as shown below.

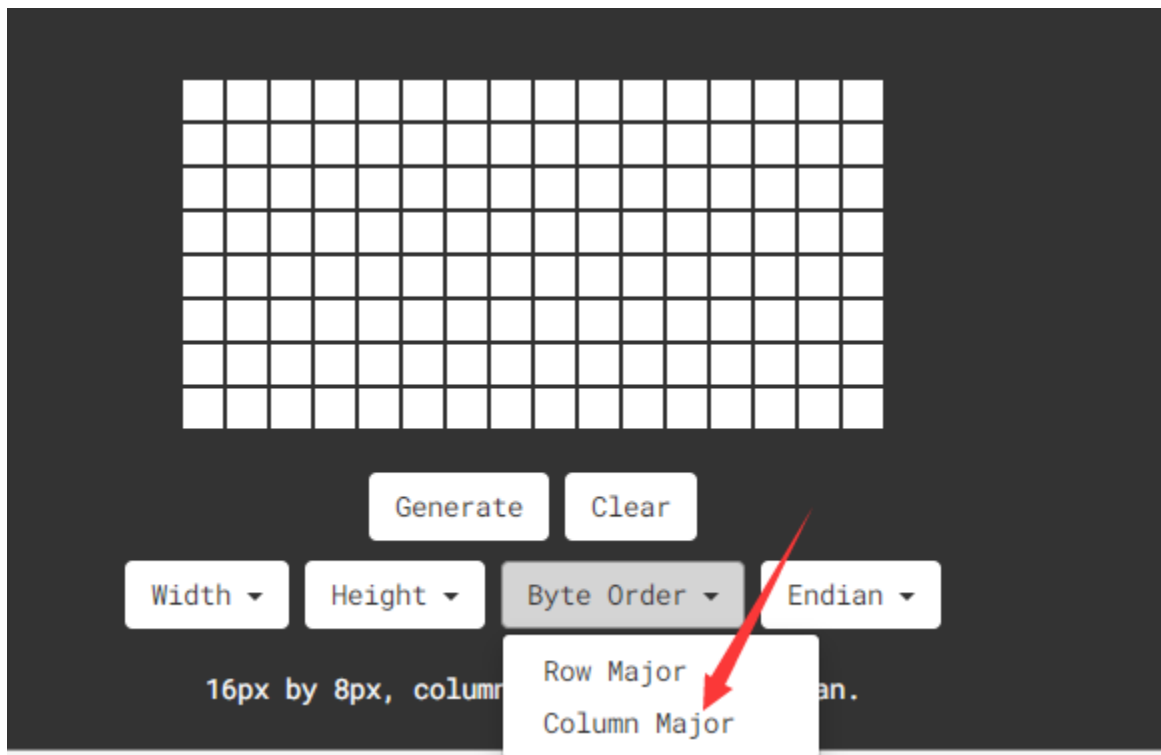




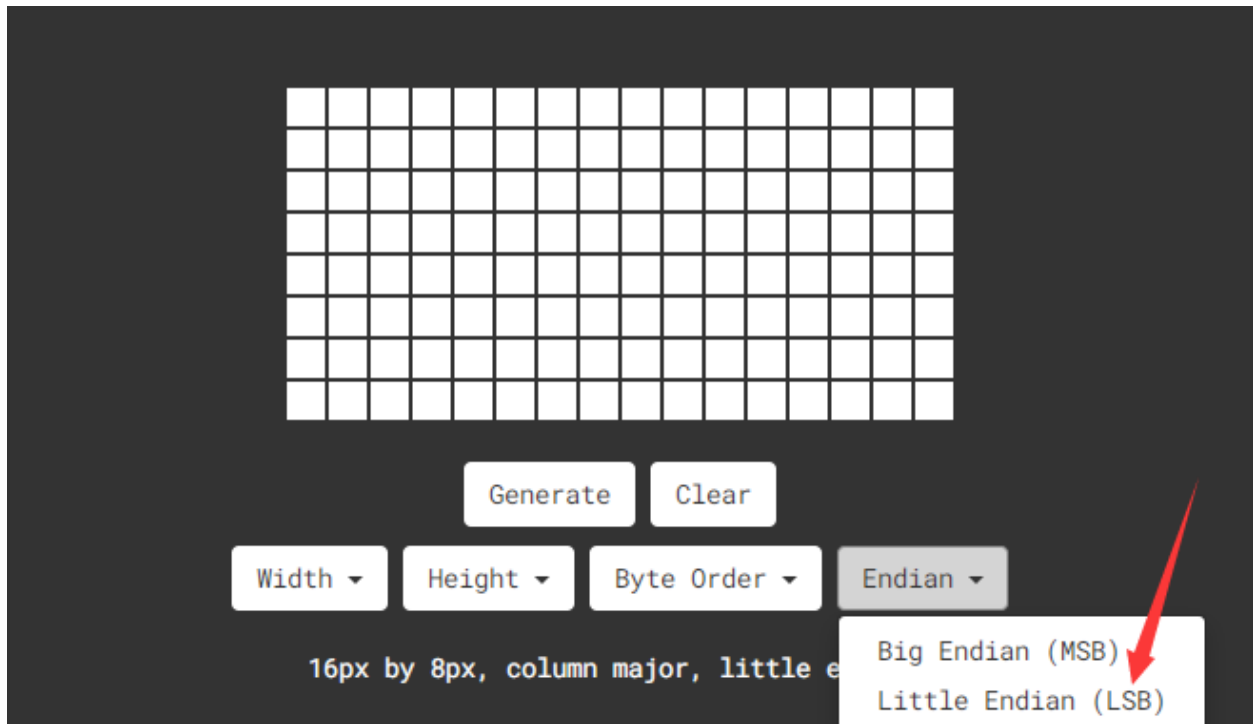


As same as dot matrix on Max board

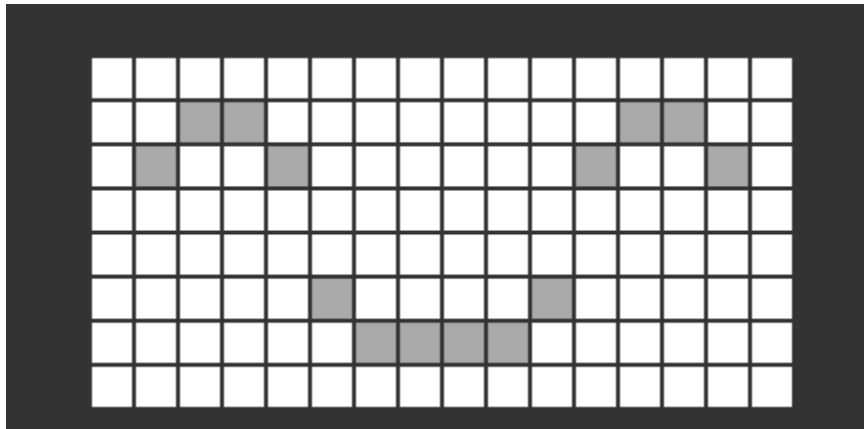
Click Byte order to select “**column major**”



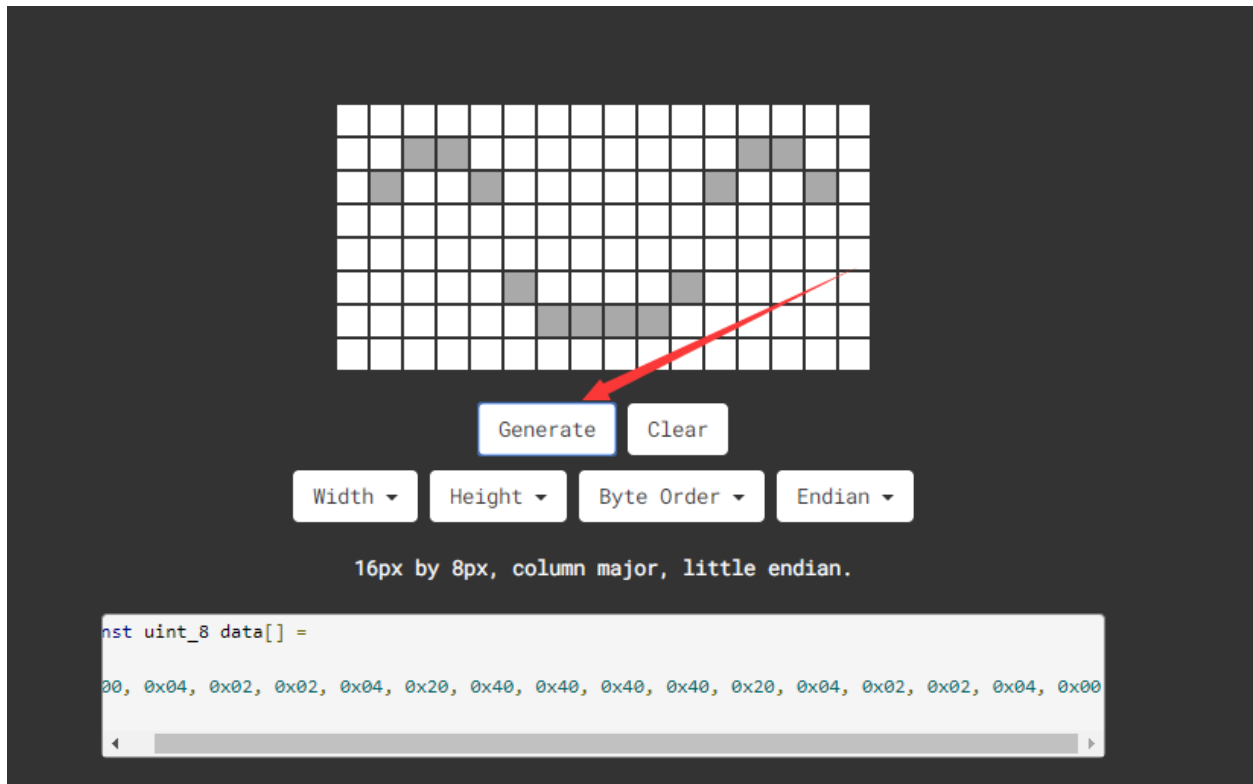
Tap endian to set little endian(lsb)



Then we draw the pattern we need.



Click generate to get hexadecimal code.

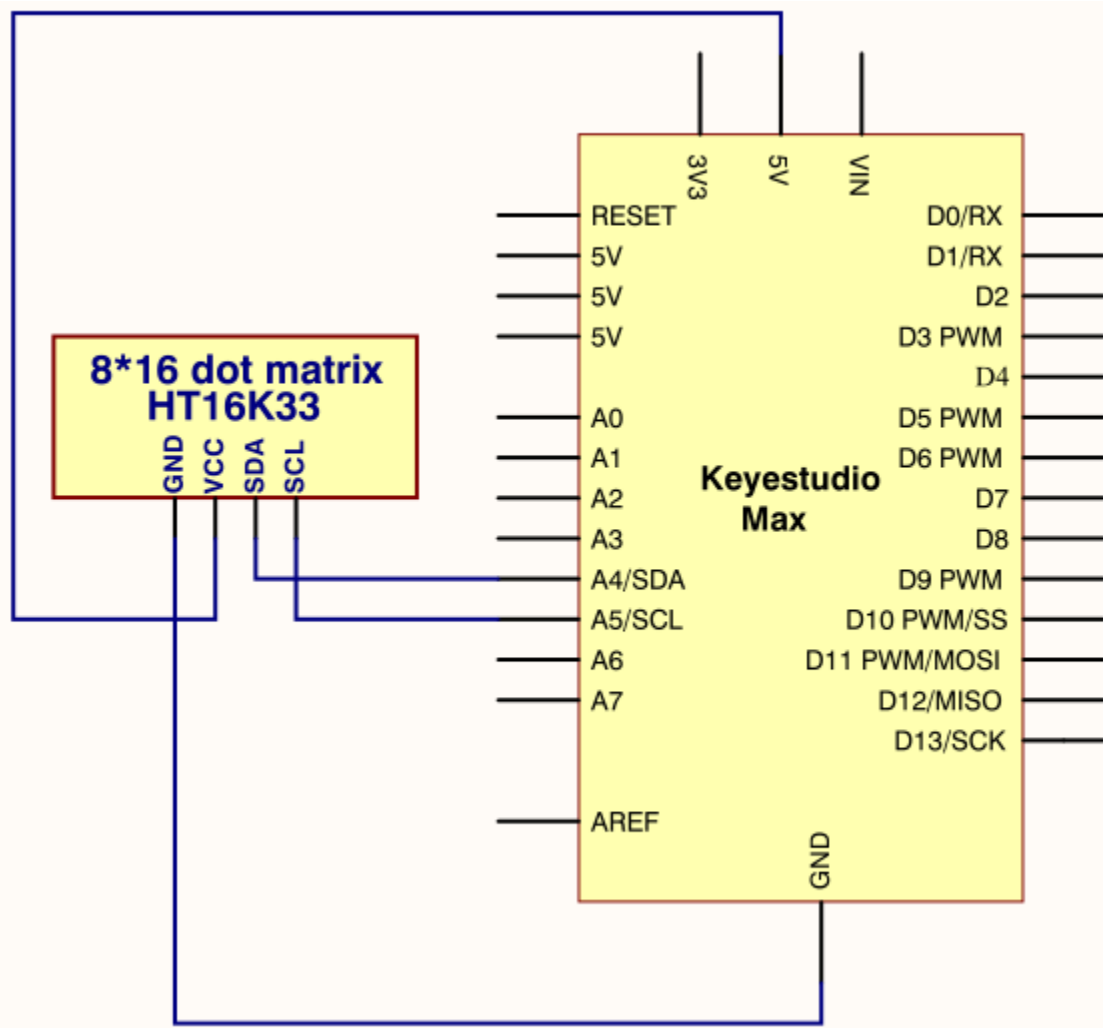


0x00, 0x04, 0x02, 0x02, 0x04, 0x20, 0x40, 0x40, 0x40, 0x40, 0x20, 0x04, 0x02, 0x02, 0x04, 0x00};

Move the above hexadecimal code into the code of the below program.

### 3. Project circuit





#### 4. Project code

```

/*
keyestudio Max Development Board
Project 17
8*16 dot matrix-display images
http://www.keyestudio.com
*/

#include <Wire.h>
#include "Keyestudio_LEDBackpack.h"
#include "Keyestudio_GFX.h"
Keyestudio_8x16matrix matrix = Keyestudio_8x16matrix();
void setup()
{
matrix.begin(0x70); // pass in the address

```

```
}  
static const uint8_t PROGMEM  
smile_bmp[] =  
{ 0x00, 0x04, 0x02, 0x02, 0x04, 0x20, 0x40, 0x40, 0x40, 0x40, 0x20, 0x04, 0x02, 0x02, 0x04, 0x00};  
void loop() {  
  //matrix.setRotation(1);  
  matrix.clear();  
  matrix.drawBitmap(0, 0, smile_bmp, 8, 16, LED_ON);  
  matrix.writeDisplay();  
  delay(1000);  
}
```

### **Project 18: 8\*16 dot matrix-knob control**

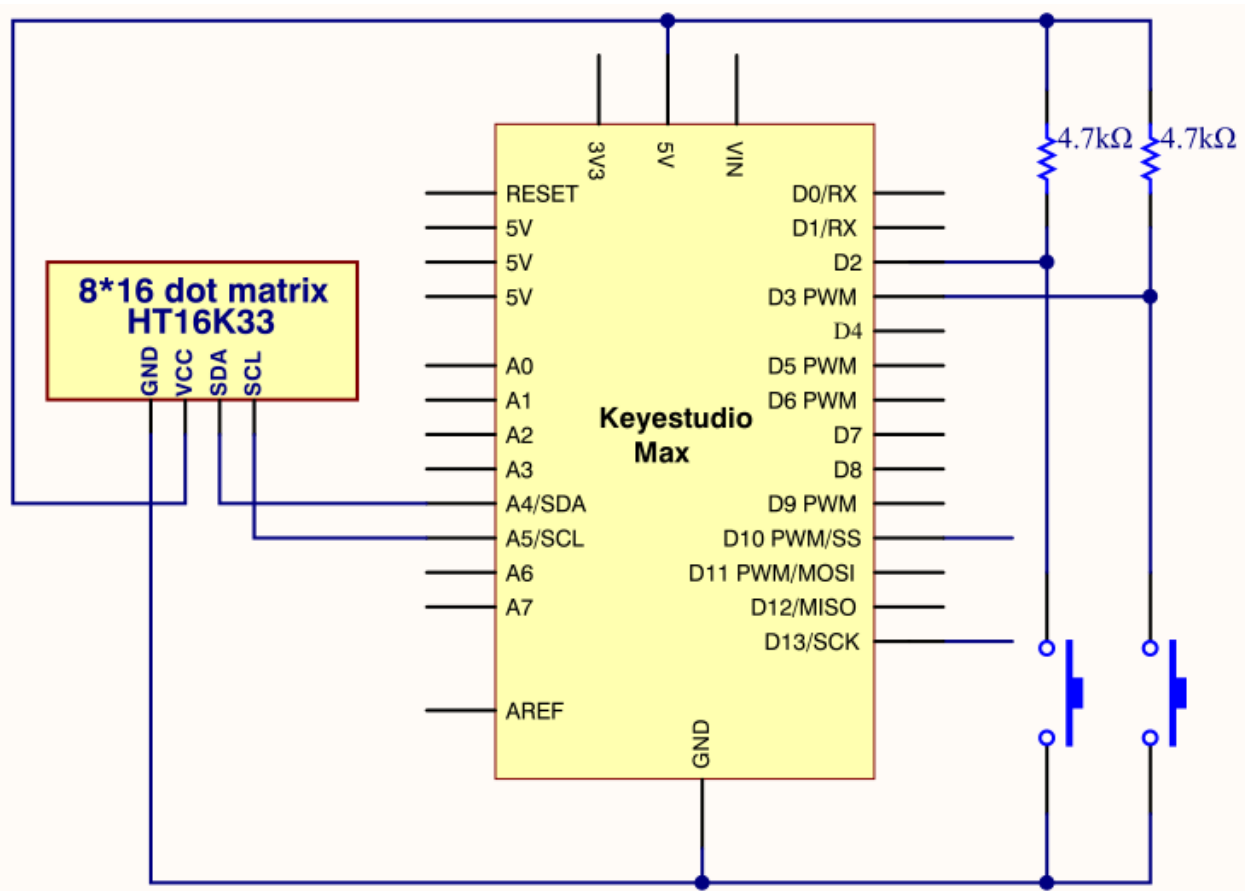
#### **1. Project instruction**

There are on-board button switches on Max board, we will combine button switch and 8\*16 dot matrix to make an interactive display.

#### **1. Project principle**

The signal pins of two buttons are connected to D2 and D3. Press left button, 8\*16 dot matrix shows“L”; press right button, 8\*16 dot matrix shows“R”.

#### **1. Project circuit**



### 1. Project code

```

/*
keyestudio Max Development Board
Project 18
8*16 dot matrix-knob control
http://www.keyestudio.com
*/
#include <Wire.h>
#include "Keyestudio_LEDBackpack.h"
#include "Keyestudio_GFX.h"
Keyestudio_8x16matrix matrix = Keyestudio_8x16matrix();
int K1=3;
int K2=2;
int x;
void setup()
{
matrix.begin(0x70); // pass in the address

```

```
pinMode(K1,INPUT);
pinMode(K2,INPUT);
matrix.drawCircle(3,8, 3, LED_ON);
matrix.writeDisplay(); // write the changes we just made to the display
}
void loop()
{
int K1_level=digitalRead(K1);
int K2_level=digitalRead(K2);
if(K1_level==0)
{
matrix.setTextSize(1);
matrix.setTextWrap(false); // we dont want text to wrap so it scrolls nicely
matrix.setTextColor(LED_ON);
matrix.setRotation(1);
matrix.clear();
matrix.setCursor(2,0);
matrix.print("L");
matrix.writeDisplay();
}
if(K2_level==0)
{
matrix.setTextSize(1);
matrix.setTextWrap(false); // we dont want text to wrap so it scrolls nicely
matrix.setTextColor(LED_ON);
matrix.setRotation(1);
matrix.clear();
matrix.setCursor(9,0);
matrix.print("R");
matrix.writeDisplay();
}}
}
```

### **Project 19: 8\*16 dot matrix-light control**

#### **1. project instruction**

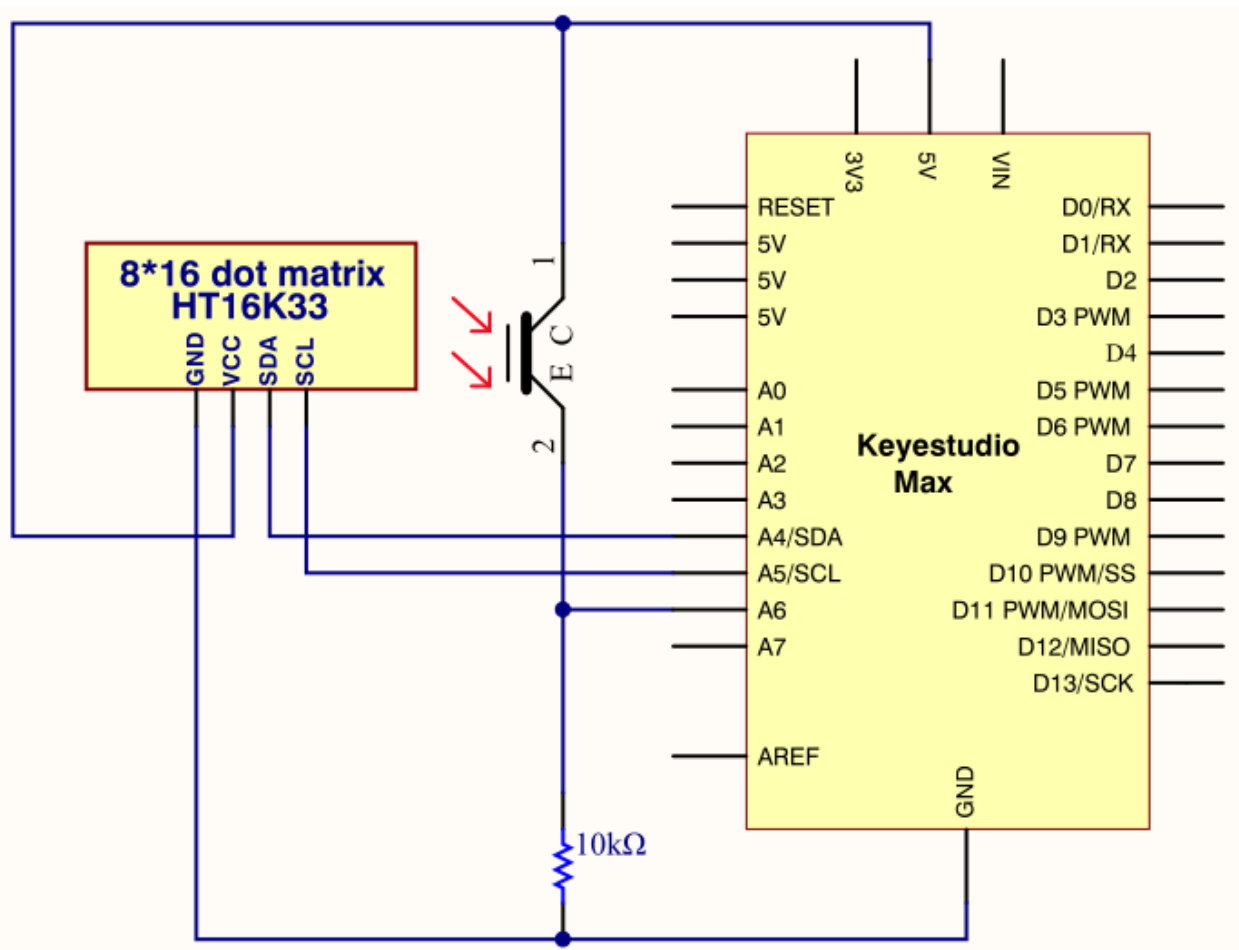
The light sensor feature that resistance is inverse proportion to to light intensity. Based on this characteristics, we made a night light. In this chapter, we will show you light column on dot matrix.

#### **1. Project Principle**

The signal pin of light sensor is linked with A6 of MAX development board.

8\*16 dot matrix will show light column, the darker the light intensity is, the shorter the height of light column.

### 1. Project circuit



### 1. Project code

```

/*
keyestudio Max Development Board
Project 19
8*16 dot matrix-light control
http://www.keyestudio.com
*/

#include <Wire.h>
#include "Keyestudio_LEDBackpack.h"
#include "Keyestudio_GFX.h"

int light = A6;
int Light_val;

Keyestudio_8x16matrix matrix = Keyestudio_8x16matrix();

```

```
void setup() {  
  Serial.begin(9600);  
  Serial.println("16x8 LED Matrix Test");  
  pinMode(A6,INPUT);  
  matrix.begin(0x70); // pass in the address  
}  
void loop() {  
  Light_val=analogRead(A6);  
  Light_val=map(Light_val,0,1023,0,15);  
  matrix.clear();  
  matrix.drawLine(3,0, 3,Light_val, LED_ON);  
  matrix.writeDisplay(); // write the changes we just made to the display  
  delay(10);  
}
```

### **Project 20: 8\*16 dot matrix-sound control**

#### **1. Project instruction**

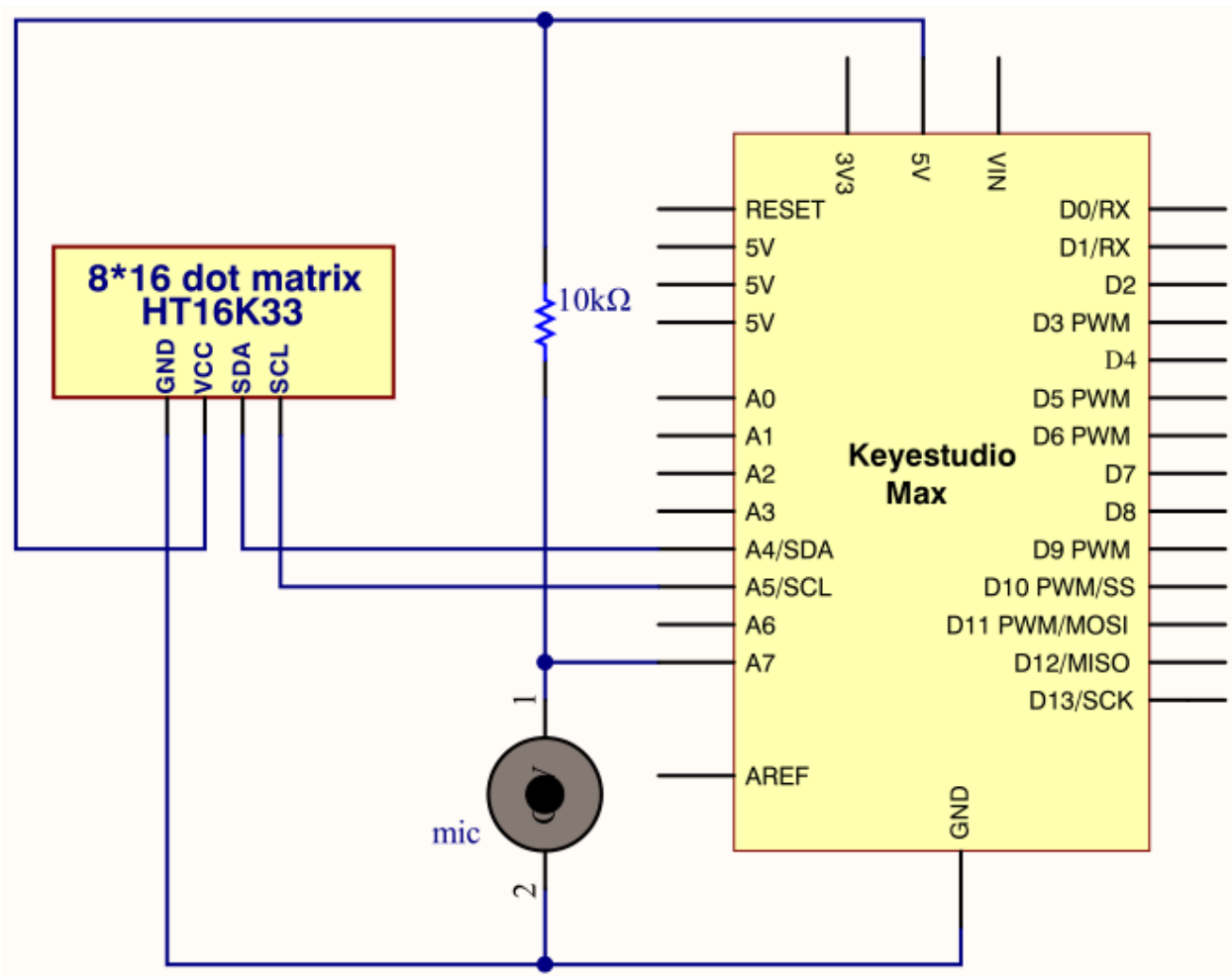
In this lesson, we will create more fantastic programs with microphone sensor and 8\*16 dot matrix

#### **1. Project Principle**

The signal pin of microphone sensor is connected to A7 of Max development.

The circle on dot matrix will enlarge with the sound intensity. The stronger the sound intensity, the larger the circle gets.

#### **1. Project circuit**



### 1. Project code

```

/*
keyestudio Max Development Board
Project 20
8*16 dot matrix-sound control
http://www.keyestudio.com
*/
#include <Wire.h>
#include "Keyestudio_LEDBackpack.h"
#include "Keyestudio_GFX.h"
int mic = A7;
Keyestudio_8x16matrix matrix = Keyestudio_8x16matrix();
int Mic_val;
void setup() {
  Serial.begin(9600);

```

```
Serial.println("16x8 LED Matrix Test");
pinMode(mic,INPUT);
matrix.begin(0x70); // pass in the address
}
void loop() {
Mic_val=analogRead(mic);
Mic_val=map(Mic_val,0,1023,0,10);
matrix.clear();
matrix.drawCircle(3,8,Mic_val, LED_ON);
matrix.writeDisplay(); // write the changes we just made to the display
delay(10);
}
```

### **Project 21: 8\*16 dot matrix-game project**

#### **1. Project instruction**

We could play games through Max development board. Can you imagine that?

Let's get started.

#### **1. Project Principle**

Game 1: gluttonous snake

Press two buttons on max board, the gluttonous snake moves clockwise and anticlockwise.

Note: 1.8.12 version Wire library is not compatible with code when testing code of game 2, you need change library in arduino-1.8.12\hardware\arduino\avr\libraries

Game 2: Break the bricks

Press buttons simultaneously to start.

#### **1. Project circuit**





```
// buttons
const int RIGHTBUTTON = 0;
const int LEFTBUTTON = 1;

// direction
const int TOP = 0;
const int RIGHT = 1;
const int BOTTOM = 2;
const int LEFT = 3;

// Snake
const int MAX_SNAKE_LENGTH = 100;

// Variables
//Keyestudio_8x8matrix matrix = Keyestudio_8x8matrix(); // Display
Keyestudio_8x16matrix matrix = Keyestudio_8x16matrix();
int direction = TOP; // direction of movement
int snakeX[MAX_SNAKE_LENGTH]; // X-coordinates of snake
int snakeY[MAX_SNAKE_LENGTH]; // Y-coordinates of snake
int snakeLength = 1; // nr of parts of snake
boolean buttonRead = false; // is button already read in this loop
unsigned long prevTime = 0; // for gamedelay (ms)
unsigned long delayTime = 500; // Game step in ms
int fruitX, fruitY;
unsigned long fruitPrevTime = 0;
unsigned long fruitBlinkTime = 1000/250;
int fruitLed = LED_ON;

void setup(){
  Serial.begin(9600);
  Serial.println("Snake is started");
  randomSeed(analogRead(0));
  // Init led matrix
  matrix.begin(0x70);
  // init buttons
  int buttonpins[] = {buttonRightPin, buttonLeftPin};
  initButtons(buttonpins, 2);
  // init snake
  snakeX[0] = 4;
  snakeY[0] = 7;
```

```
for(int i=1; i<MAX_SNAKE_LENGTH; i++){
snakeX[i] = snakeY[i] = -1;
}
makeFruit();
}
void loop(){
checkButtons();
unsigned long currentTime = millis();
if(currentTime - prevTime >= delayTime){
nextstep();
buttonRead = false;
prevTime = currentTime;
}
draw();
}
void initButtons(int pinNumbers[], int length){
Serial.println("initButtons");
// Copy PIN numbers
buttonPin = new int[length];
for(int i=0; i<length; i++){
buttonPin[i] = pinNumbers[i];
}
// Initialize button state
buttonState = new int[length];
for(int i=0; i<length; i++){
buttonState[i] = LOW;
}
for(int i=0; i<length; i++)
{
Serial.print(buttonPin[i]);
Serial.print(buttonState[i]);
}
}
boolean buttonClicked(int index){
// current state
int state = digitalRead(buttonPin[index]);
```

```
boolean buttonClick = false;
// button has pressed
if(buttonState[index] == HIGH){
  if(state == LOW){
    buttonClick = true;
  }
}
buttonState[index] = state;
return buttonClick;
}

void checkButtons(){
  if(!buttonRead){
    int currentDirection = direction;
    if(buttonClicked(LEFTBUTTON)){
      direction--;
      if(direction < 0){
        direction = LEFT;
      }
    }
    else if(buttonClicked(RIGHTBUTTON)){
      direction++;
      if(direction > 3){
        direction = TOP;
      }
    }
    buttonRead = (currentDirection != direction);
  }
}

void draw(){
  matrix.clear();
  drawSnake();
  drawFruit();
  matrix.writeDisplay();
}

void drawSnake(){
  for(int i=0; i<snakeLength; i++){
```

```
matrix.drawPixel(snakeX[i], snakeY[i], LED_ON);
}
}
void drawFruit(){
if(inPlayField(fruitX, fruitY)){
unsigned long currenttime = millis();
if(currenttime - fruitPrevTime >= fruitBlinkTime){
fruitLed = (fruitLed == LED_ON) ? LED_OFF : LED_ON;
fruitPrevTime = currenttime;
}
matrix.drawPixel(fruitX, fruitY, fruitLed);
}
}
boolean inPlayField(int x, int y){
return (x>=0) && (x<8) && (y>=0) && (y<16);
}
void nextstep(){
for(int i=snakeLength-1; i>0; i-){
snakeX[i] = snakeX[i-1];
snakeY[i] = snakeY[i-1];
}
switch(direction){
case TOP:
snakeY[0] = snakeY[0]-1;
break;
case RIGHT:
snakeX[0] = snakeX[0]+1;
break;
case BOTTOM:
snakeY[0] = snakeY[0]+1;
break;
case LEFT:
snakeX[0]=snakeX[0]-1;
break;
}
if((snakeX[0] == fruitX) && (snakeY[0] == fruitY)){
```

```
snakeLength++;
if(snakeLength < MAX_SNAKE_LENGTH){
makeFruit();
}
else {
fruitX = fruitY = -1;
}
}
}

void makeFruit(){
int x, y;
x = random(0, 7);
y = random(0, 15);
while(isPartOfSnake(x, y)){
x = random(0, 7);
y = random(0, 15);
}
fruitX = x;
fruitY = y;
}

boolean isPartOfSnake(int x, int y){
for(int i=0; i<snakeLength-1; i++){
if((x == snakeX[i]) && (y == snakeY[i])){
return true;
}
}
return false;
}

//*****
```

### **1. Project code2**

```
*****Break the bricks*****

/*
keyestudio Max Development Board
Project 21
8*16 dot matrix-break the bricks
http://www.keyestudio.com
*/
```

```
#include <Wire.h>
#include "Keyestudio_LEDBackpack.h"
#include "Keyestudio_GFX.h"
#include "Breakout.h"
Keyestudio_8x16matrix matrix = Keyestudio_8x16matrix();
Breakout breakout;
int count = 0;
void setup() {
  Serial.begin(9600);
  Serial.println("8x16 LED Matrix Test");
  pinMode(L_PIN, INPUT);
  pinMode(R_PIN, INPUT);
  matrix.begin(0x70); // pass in the address
  matrix.setBrightness(0);
}
void loop() {
  // action depends on game state - ideally the state should be a part of the Breakout class or a separate logic class
  switch(state){
    // setup game
    case SETUP:
      breakout.restart();
      breakout.speed = START_SPEED;
      state = START;
      break;
    case START:
      // ready to start the game, changed to PLAY if the action button is pushed.
      // Change here if paddle should be able to move before you shoot the first ball and start the game.
      checkActions();
      if(actions == SEL){
        state = PLAY;
      }
      break;
    case LOST:
      // game lost
      breakout.lost();
      reDraw();
```

```
blink(10, 100);
breakout.restart();
breakout.speed = START_SPEED;
state = START;
break;
case WON:
// game won
breakout.won();
reDraw();
blink(10, 100);
breakout.restart();
state = START;
break;
case PLAY:
// game in play
{
if(count > breakout.speed){
checkControls();
breakout.play(controls);
count = 0;
if(breakout.bricks == 128){
state = WON;
}
if(breakout.ball.y == 7)
state = LOST;
}
delay(20);
count++;
}
}
reDraw();
}
void reDraw(){
for(int i = 0; i < 8; i++){
for(int j = 0; j < 16; j++){ //change 8 into 16
if(breakout.level[i][16-j-1] != 'E'){
```



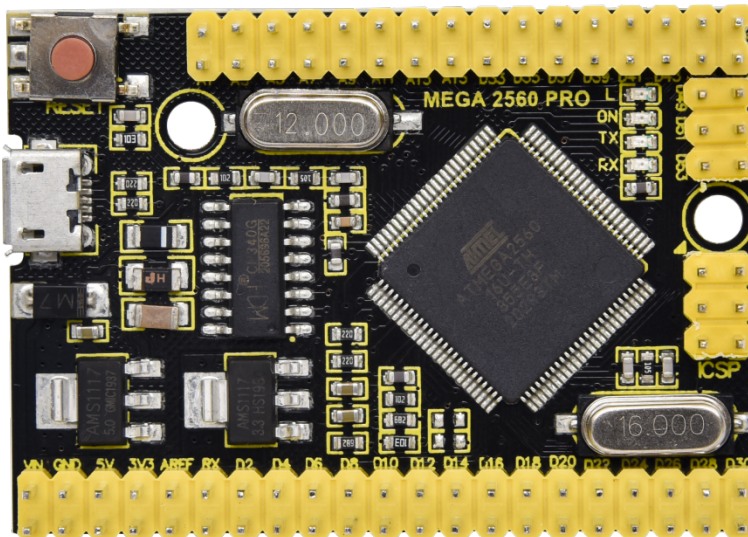
```
matrix.drawPixel(i,j,1);
}
else
matrix.drawPixel(i,j,0);
}
}
matrix.writeDisplay();
}
void checkActions(){
int count = 0;
while(count < 5 && (!digitalRead(R_PIN))&&(!digitalRead(L_PIN))){
count++;
}
if(count > 4){
actions = SEL;
return;
}
actions = NO;
}
void checkControls(){
int count = 0;
while(count < 5 && (!digitalRead(L_PIN))){
count++;
}
if(count > 4){
controls = LEFT;
return;
}
count = 0;
while(count < 5 && (!digitalRead(R_PIN))){
count++;
}
if(count > 4){
controls = RIGHT;
return;
}
```

```
controls = NONE;
}
void blink(int times, int wait){
for(int i = 0; i < times; i++){
matrix.setBrightness(2);
matrix.writeDisplay();
delay(wait);
matrix.setBrightness(0);
matrix.writeDisplay();
delay(wait);
}
}
```

### 1. Resource:

<https://fs.keyestudio.com/KS0500>

## KS0502 KEYESTUDIO MEGA PRO DEVELOPMENT BOARD



### Introduction

The MEGA control board, among the series of MCUs , is the most popular since it has numerous pins.

However, a substantial number of pins don't meet the space demand of DIY design. To tackle this issue, we roll out the Keyestudio MEGA PRO development board. In fact, its use method is as same as the official mega board, in addition to the different volume.

Its processor core is ATMEGA2560-16AU. In the meantime, it has 54 digital input/output pins (of which 15 can be used as PWM outputs), 16 analog inputs, 4 channel serial communication ports, a USB connection, 1 ICSP header, and a reset button. And all ports are extended by pins with the interval of 2.54mm.

What's more, you can burn the firmware for ATMEGA2560-16AU through the built-in ICSP port. The firmware of this chip is burnt well before delivery, therefore, you don't need to burn the firmware.

The power can be supplied through USB cable, port 5V , GND 5V, as well as Vin GND (DC 7-12V).

### Specification

Microcontroller: ATMEGA2560-16AU

USB to serial chip CH340G

Operating Voltage: 5V

Input Voltage (recommended): DC 7-12V

Digital I/O Pins: 54 (D0-D53)

PWM Digital I/O Pins 15 (D2-D13 D44-D46)

## Board for Arduino

---

Analog Input Pins: 16(A0-A15)

DC Current per I/O Pin: 20 mA

DC Current for 3.3V Pin: 50 mA

Flash Memory: 256 KB of which 8 KB used by bootloader

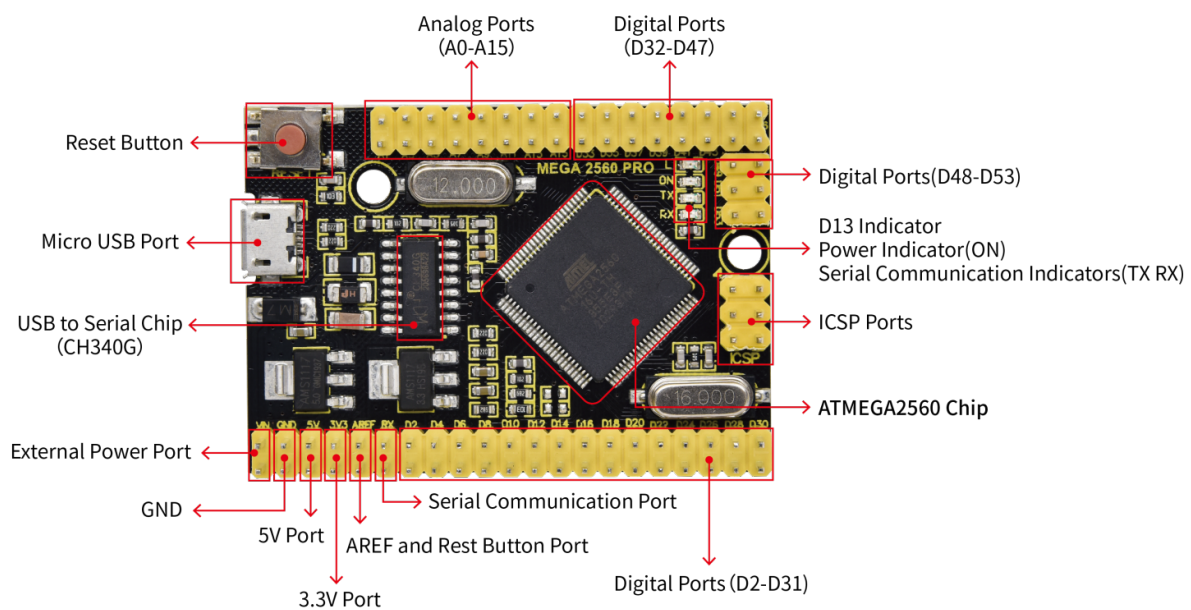
SRAM: 8 KB

EEPROM: 4 KB

Clock Speed: 16 MHz

LED\_BUILTIN:D13

### Pinout Diagram:



1	External Power Ports	External input: DC 7-12V
2	Ground	GND
3	5V	DC 5V input/output voltage supply power for control board when inputting 5V
4	3.3V	Provide DC 3.3V output voltage
5	AREF	Analog reference. Used to set the external reference voltage(0-5V)
6	Reset Button Port	Can be connected to press button, as same as reset button
7	Serial communication port	The default serial communication port RX responds to D0(digital port), TX responds to D1(digital port)
8	Digital Ports	Have 54 digital input/output pins (of which 15 can be used as PWM outputs). These pins can be configured as digital input pin to read the logic value (0 or 1). Or used as digital output pin to drive different modules like LED, relay, etc.
9	Analog Ports	16 analog pins (A0-A15)
10	AT-MEGA2560 Micro-controller	Each board has its own microcontroller,. The MCU of this board is ATMEGA2560-16AU.
11	ICSP Pin	the AVR, an Arduino micro-program header consisting of MOSI, MISO, SCK, RESET, VCC, and GND. It is often called the SPI (serial peripheral interface) and can be considered an “extension” of the output. In fact, slave the output devices to the SPI bus host. When connecting to PC, program the firmware to ATMEGA2560-16AU.
12	L Indicator	When D13 is high level, LED will be on; when it is low level, LED will be off
13	ON Indicator	LED is on when control board is plugged in power, otherwise, it will be off
14	TX Indicator	When Arduino board communicates via serial port and sends the message, TX led will flash
15	RX Indicator	When Arduino board communicates via serial port and receive the message, RX led will flash.
16	Reset Button	Reset your control board
17	Micro USB	Supply power for control board and upload code
18	USB to serial chip	CH340G, transform USB signals of computer to serial signals

### Specialized Functions of Some Pins:

Serial Communication Port(4-channel)SerialD0 corresponds to RX0, D1 is equivalent to TX0, Serial1D19 corresponds RX1, D18 is equivalent to TX1), Serial2 (D17 corresponds to RX2, D16 stands for TX2) and Serial3D15 stands for RX3, D14 corresponds to TX3.

RXD0and TXD1are connected to the USB to serial chip of CH340G

- **Serial Communication:** D0 (RX0) and D1 (TX1); Serial 1: D19 (RX1) and D18 (TX1); Serial 2: D17 (RX2) and D16 (TX2); Serial 3: D15 (RX3) and D14 (TX3).

- **PWM Pins (Pulse-Width Modulation):** D2 to D13, and D44 to D46.
- **External Interrupts:** D2 (interrupt 0), D3 (interrupt 1), D18 (interrupt 5), D19 (interrupt 4), D20 (interrupt 3), and D21 (interrupt 2).
- **SPI communication:** D53 (SS), D52 (SCK), D51 (MOSI), D50 (MISO).
- **IIC communication:** D20 (SDA); D21 (SCL).

## 4. Windows System

### 4.1 Download Arduino IDE

When getting this control board, we need to install Arduino IDE

Enter the website <https://www.arduino.cc/> and click **SOFTWARE** > **DOWNLOADS**



The screenshot shows the Arduino.cc website. The top navigation bar includes links for PROFESSIONAL, EDUCATION, and STORE. Below this, a secondary navigation bar has links for HARDWARE, SOFTWARE (highlighted with a red box), CLOUD, DOCUMENTATION, COMMUNITY, BLOG, and ABOUT. A search bar is located on the right. Below the navigation bar, there are buttons for CODE ONLINE and GETTING STARTED. A section titled 'Downloads' (highlighted with a red box) features the Arduino IDE 1.8.13 logo and a description: 'The open-source Arduino Software (IDE) makes it easy to write code and upload it to the board. This software can be used with any Arduino board. Refer to the Getting Started page for Installation instructions.' Below this, it mentions 'SOURCE CODE' and states that active development is hosted by GitHub. To the right, a 'DOWNLOAD OPTIONS' section lists various download methods: Windows (Win 7 and newer, ZIP file, and Windows app Win 8.1 or 10), Linux (32 bits, 64 bits, ARM 32 bits, and ARM 64 bits), and Mac OS X (10.10 or newer). A 'Get' button is next to the Windows app option.

You can select the latest version—1.8.13. Alternatively, the previous release is your another choice.

In this project, we use 1.8.12 version.

## Previous Releases

Download the previous version of the current release, the classic 1.0.x, or old beta releases.

### DOWNLOAD OPTIONS

**Previous Release (1.8.12)**

**Arduino 1.0.x**

**Arduino 1.5.x beta**

**Arduino 1.9.x beta**

Click **Previous Release (1.8.12)** to enter the new page. As shown below;

The **Windows installer** needs installing manually. Yet , the **Windows zip file for non admin install** a zip file of Arduino 1.8.12 version, can be directly downloaded and installed.

---

# Previous IDE Releases

## ARDUINO 1.8.12

Arduino IDE that can be used with any Arduino board, including the Arduino Yún and Arduino DUE. Refer to the [Getting Started](#) page for Installation instructions.

[See the release notes.](#)

**Windows** [Installer](#)

**Windows** [ZIP file for non admin install](#)

**Mac OS X** [10.8 Mountain Lion or newer](#)

**Linux** [32 bits](#)

**Linux** [64 bits](#)

**Linux** [ARM 32](#)

**Linux** [ARM 64](#)

**Source**

### 4.2 Install Driver

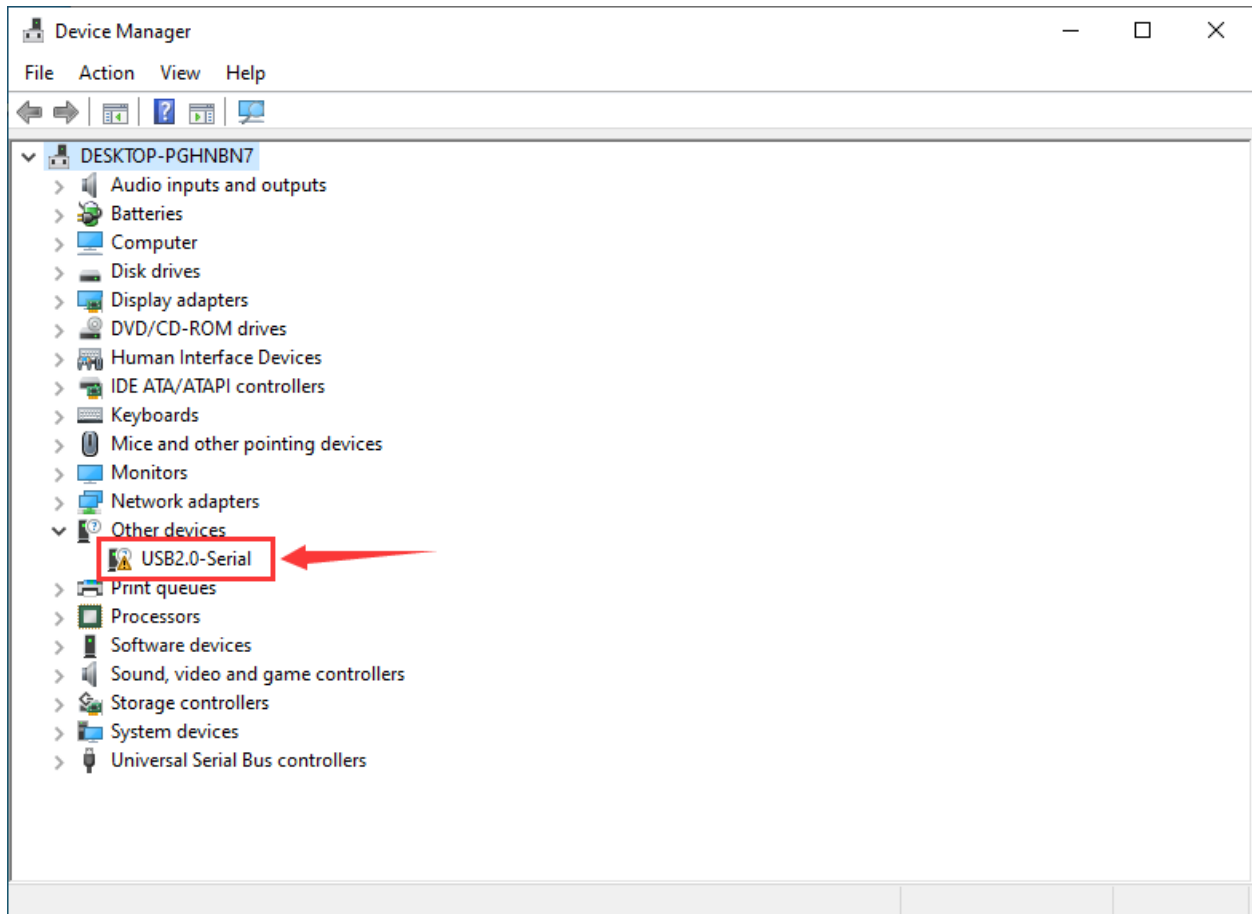
For this part, we need to install the driver of Arduino IDE


First, let's attach USB cable to computer. The driver can be installed automatically if the PC system is Windows 10, however, you need to install the driver manually if the PC system is other version.

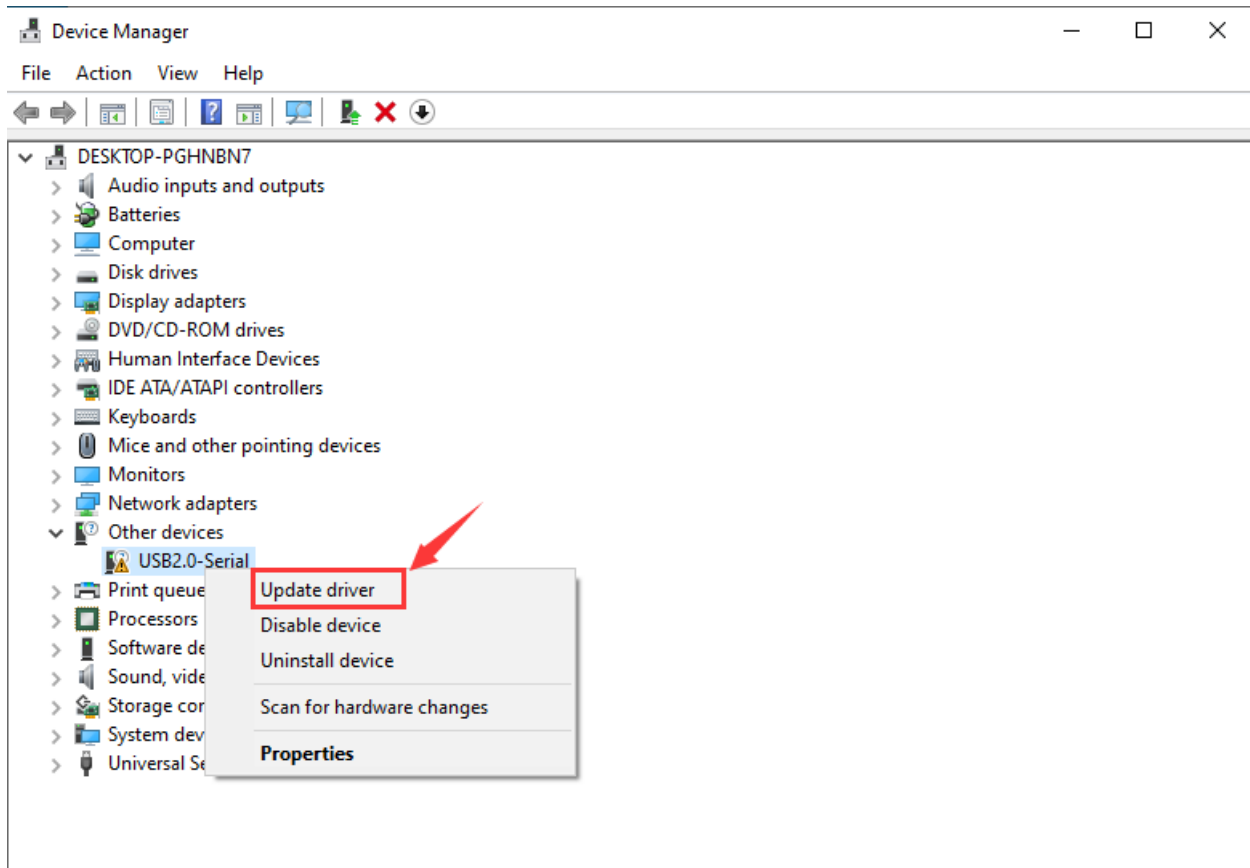
The USB to serial chip of control board is CH340G, therefore, we will install its driver(usb\_ch341\_3.1.2009.06).

Click Computer— Properties— Device Manager, as shown below:

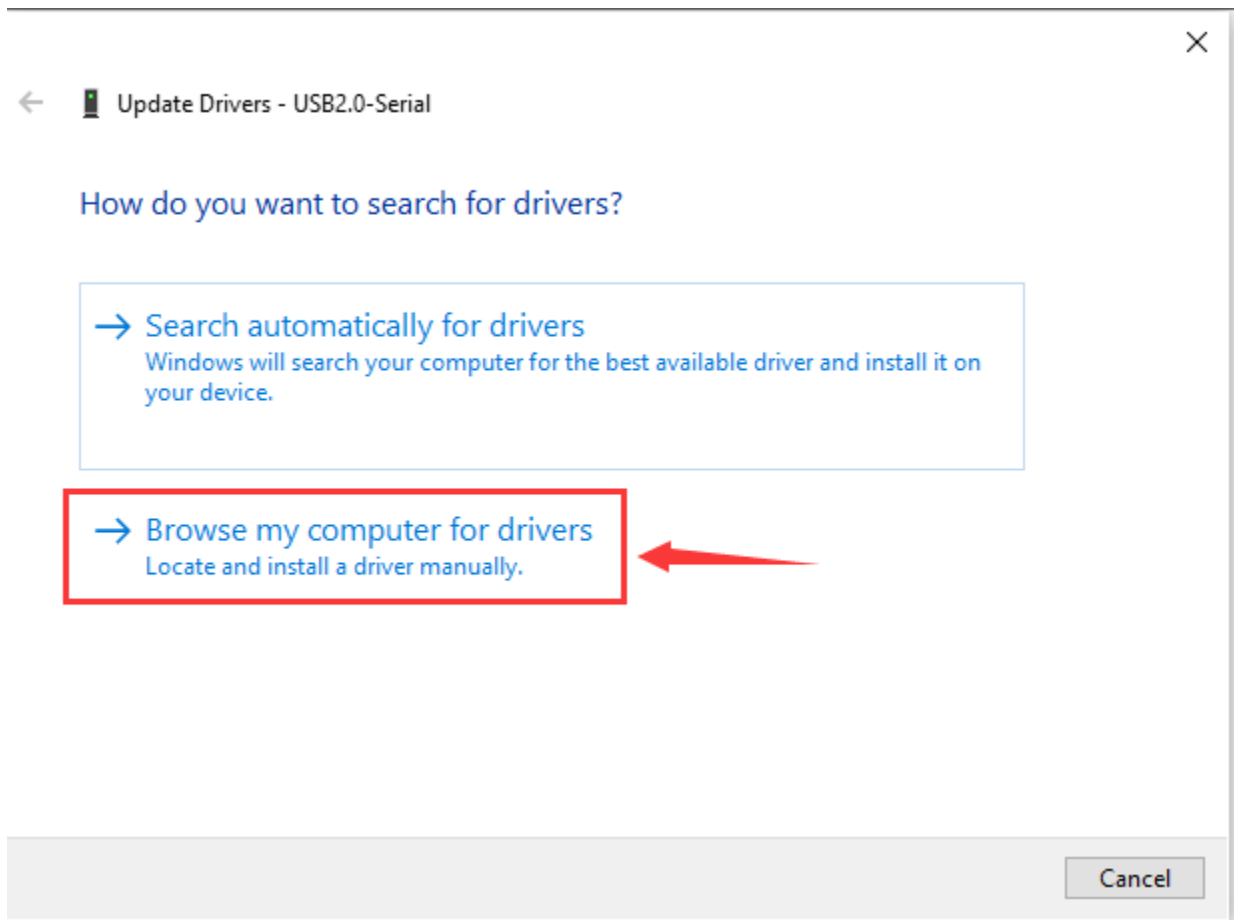




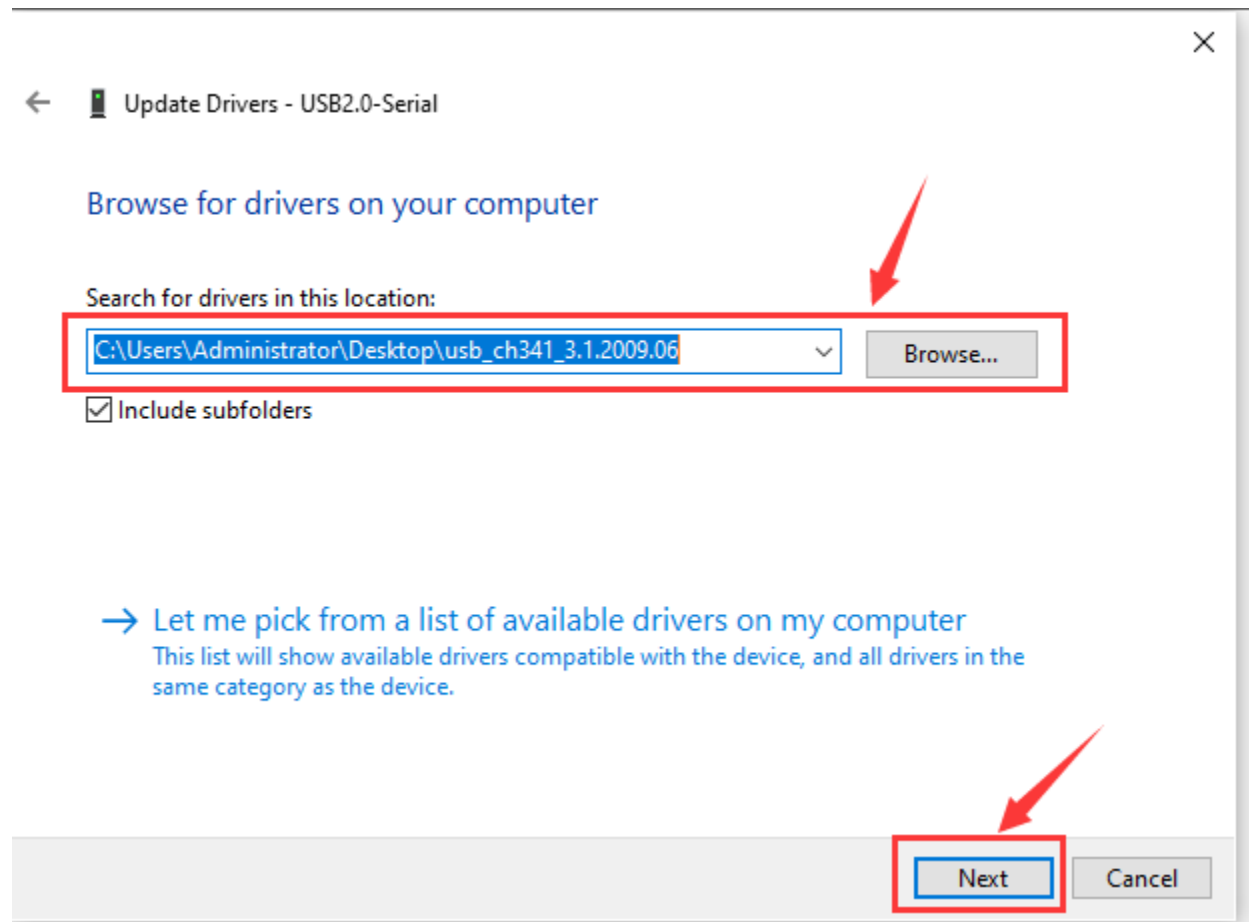
Click  **USB Serial** and “Update Driver Program”



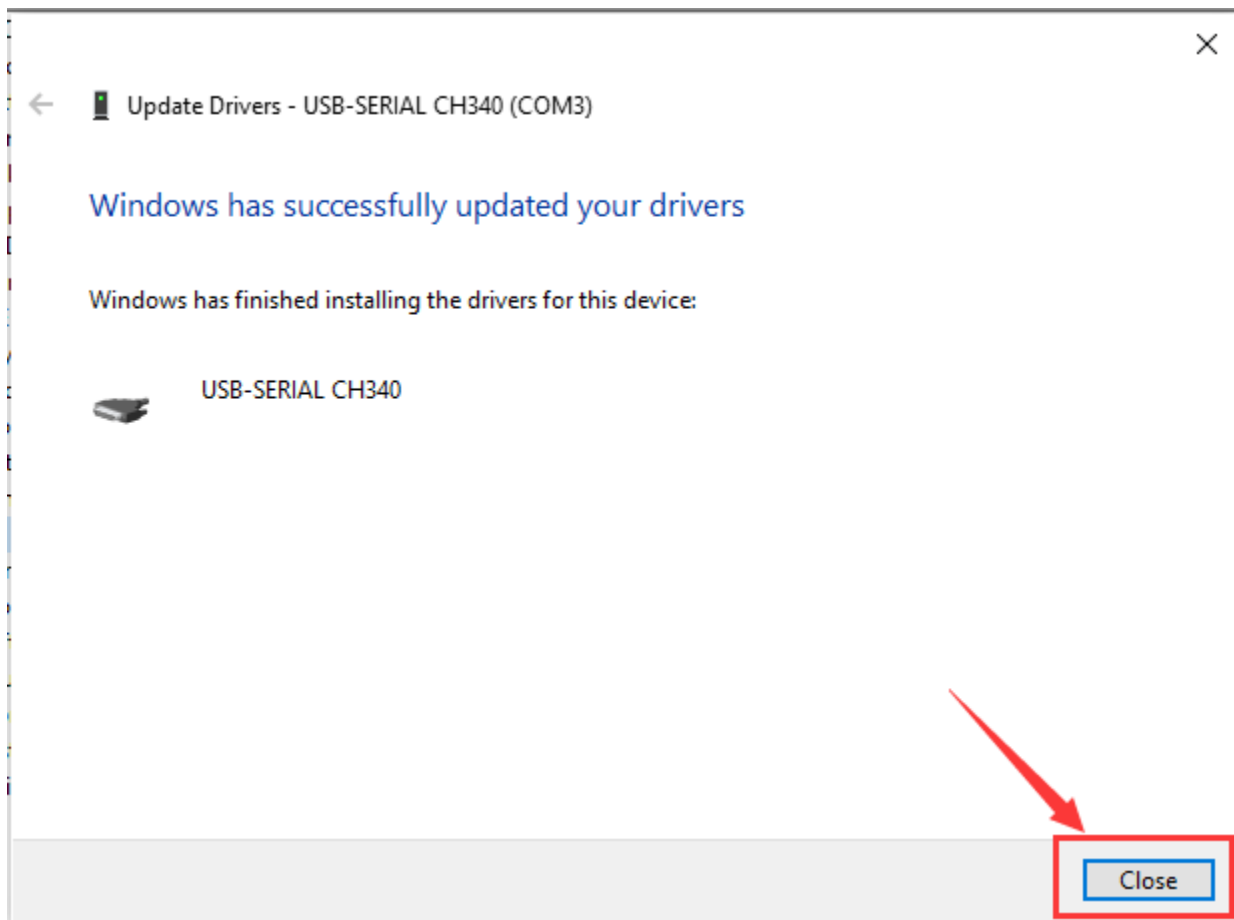
Jump into the following page and select “Browse my computer for driver software”.



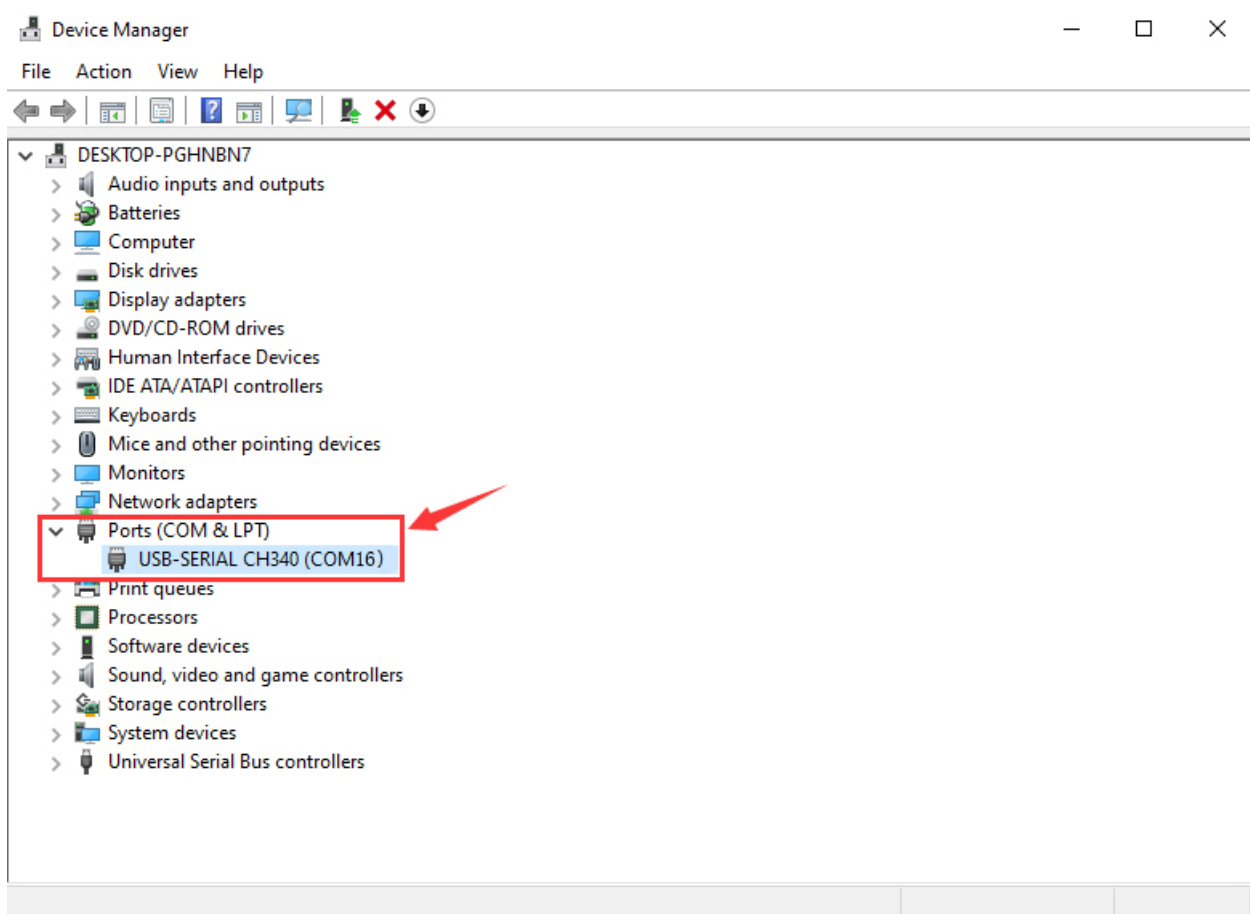
Search the **usb\_ch341\_3.1.2009.06** folder



After the driver is installed, you need to click **Close**.




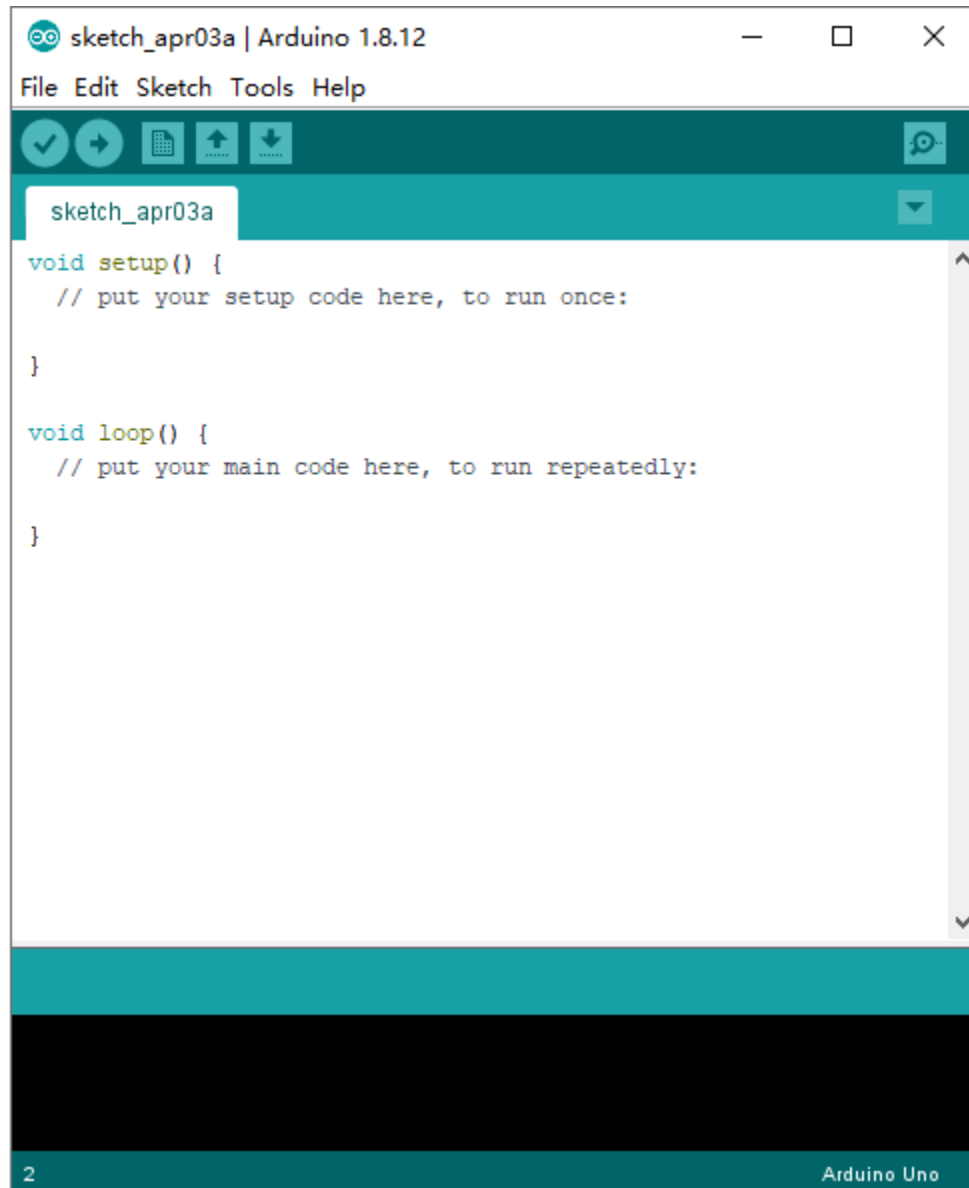
Click Computer— Properties— Device Manager, as shown below



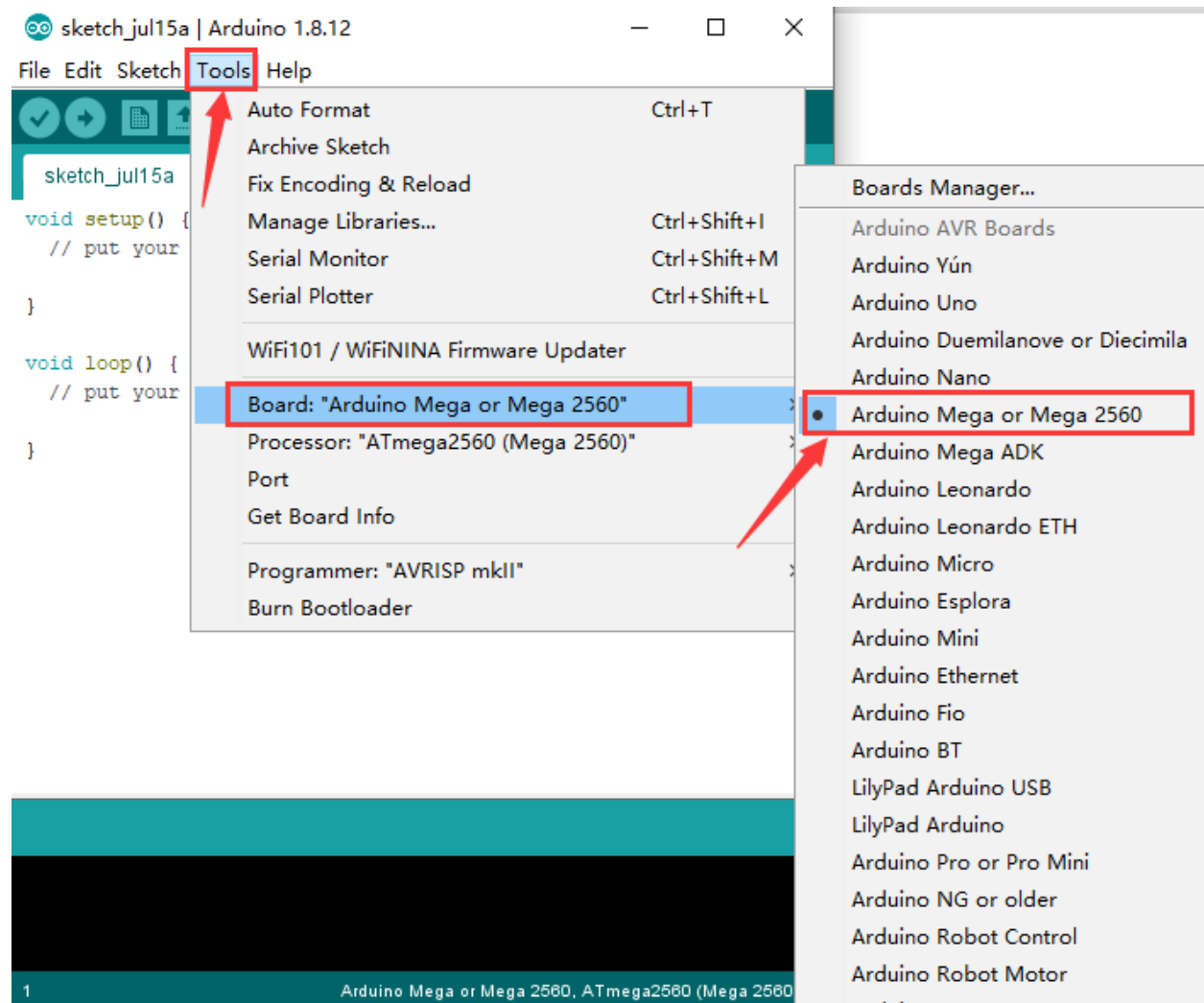
### 4.3 Arduino IDE Setting



Click  icon to pen Arduino IDE.

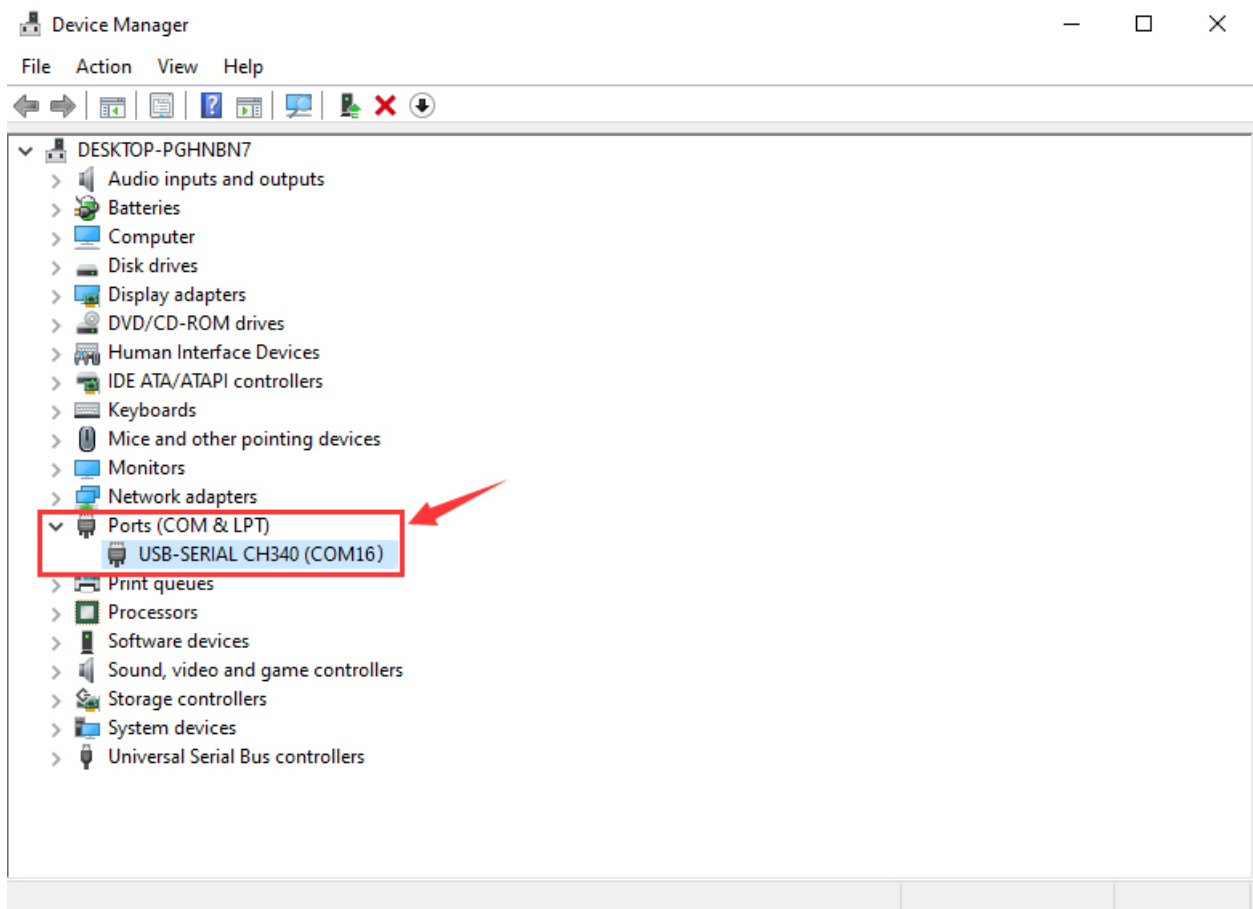


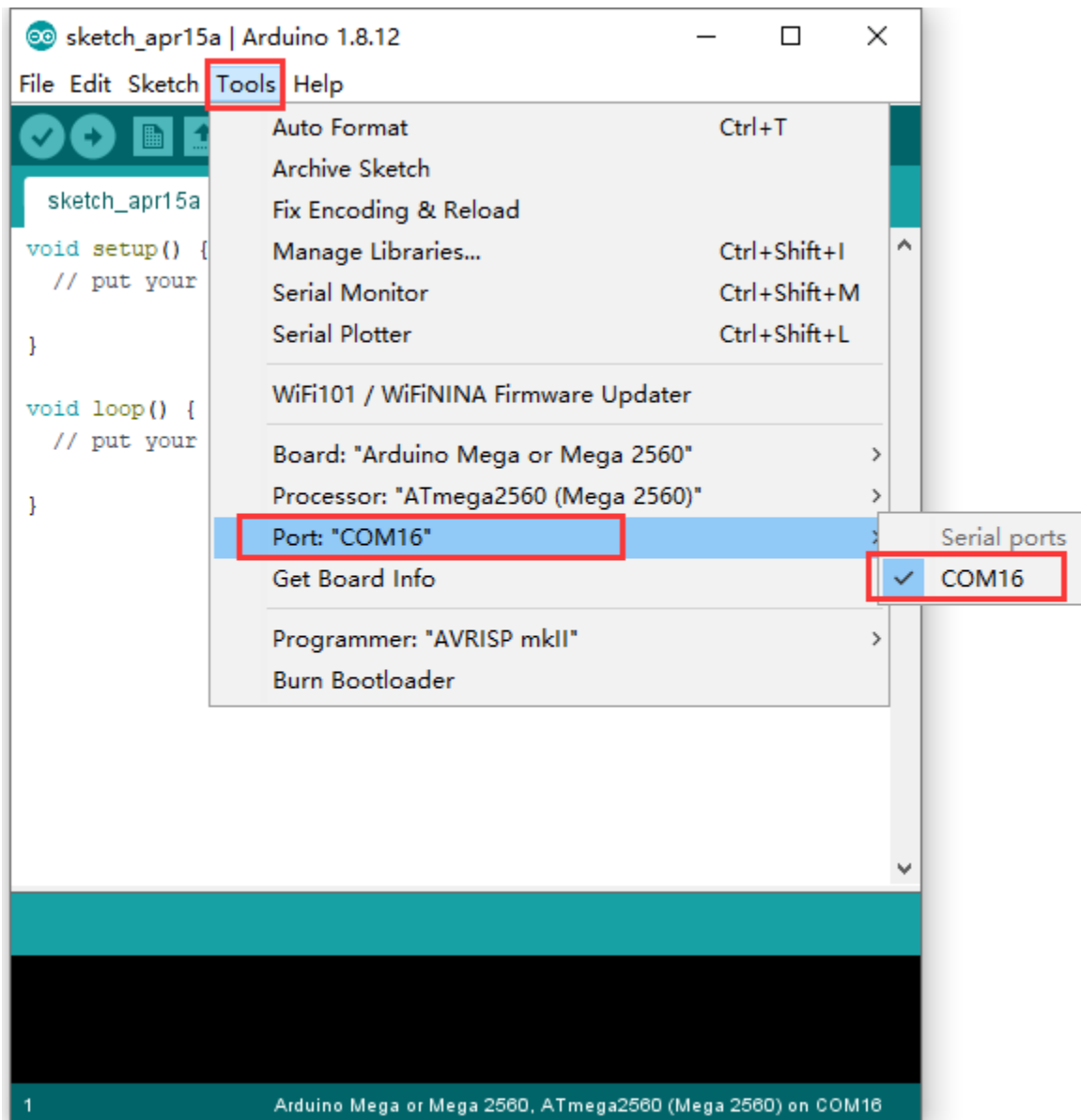
When downloading the sketch to the board, you must select the correct name of Arduino board that matches the board connected to your computer. As shown below;

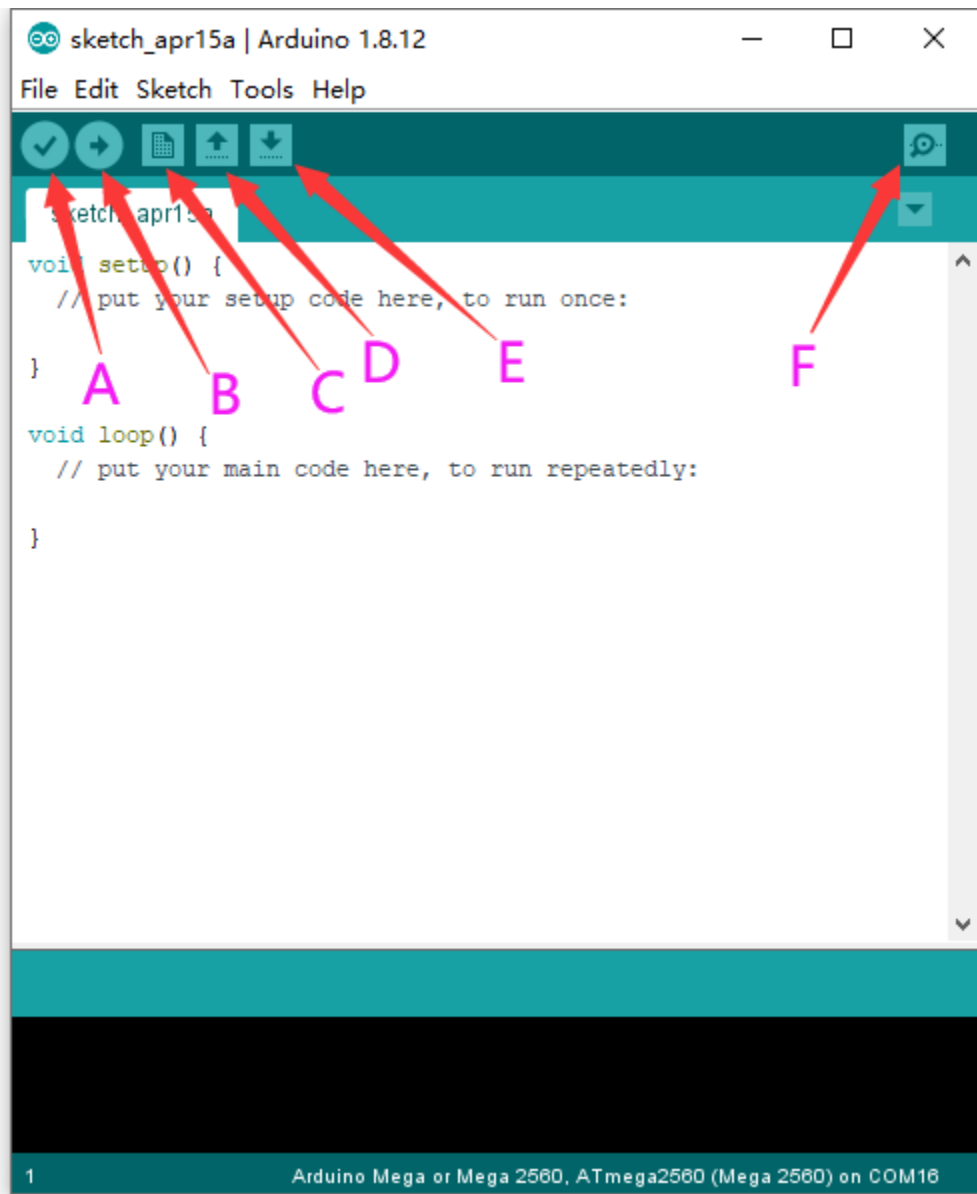


Then select the correct COM port (you can see the corresponding COM port after the driver is successfully installed).





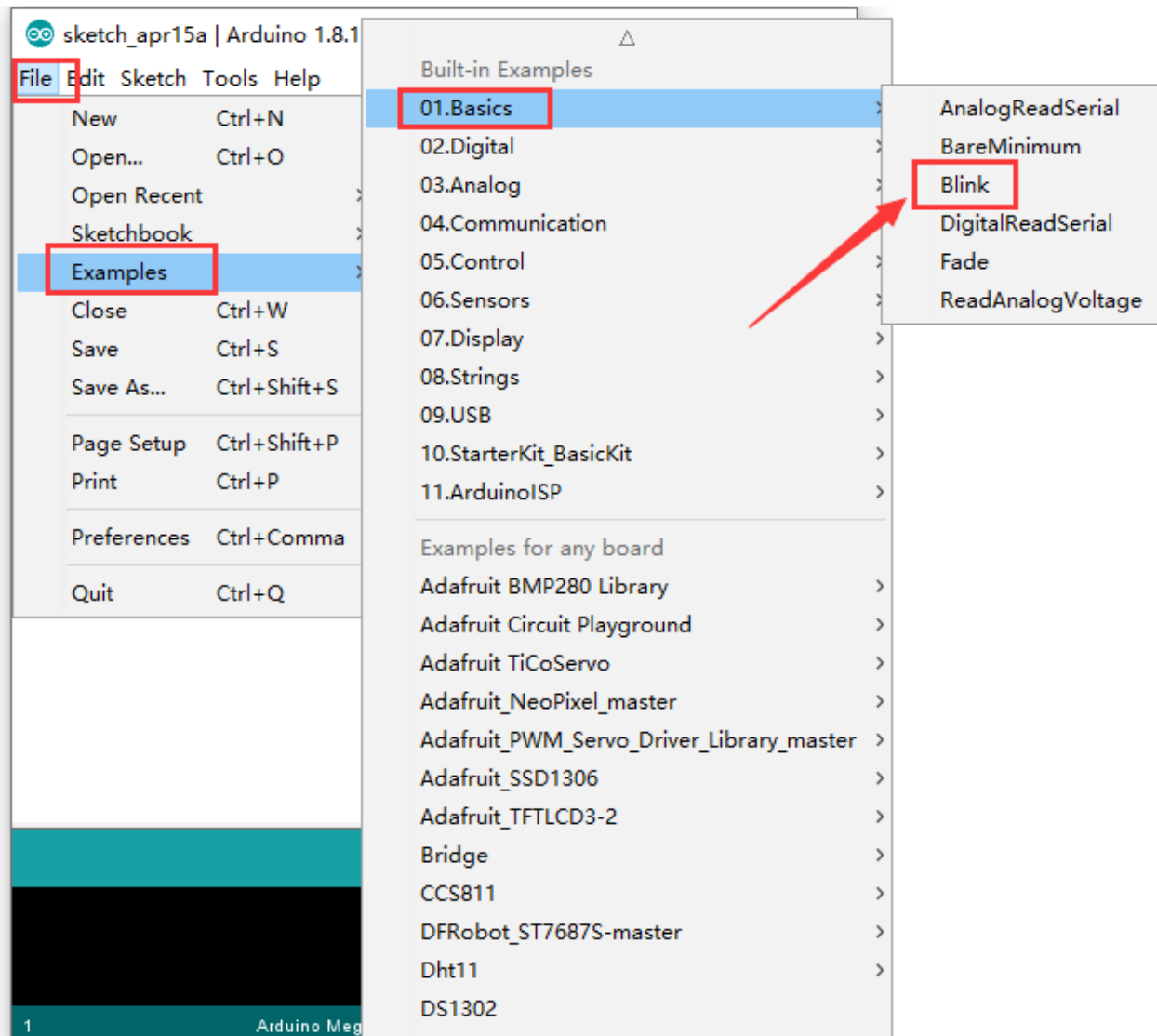


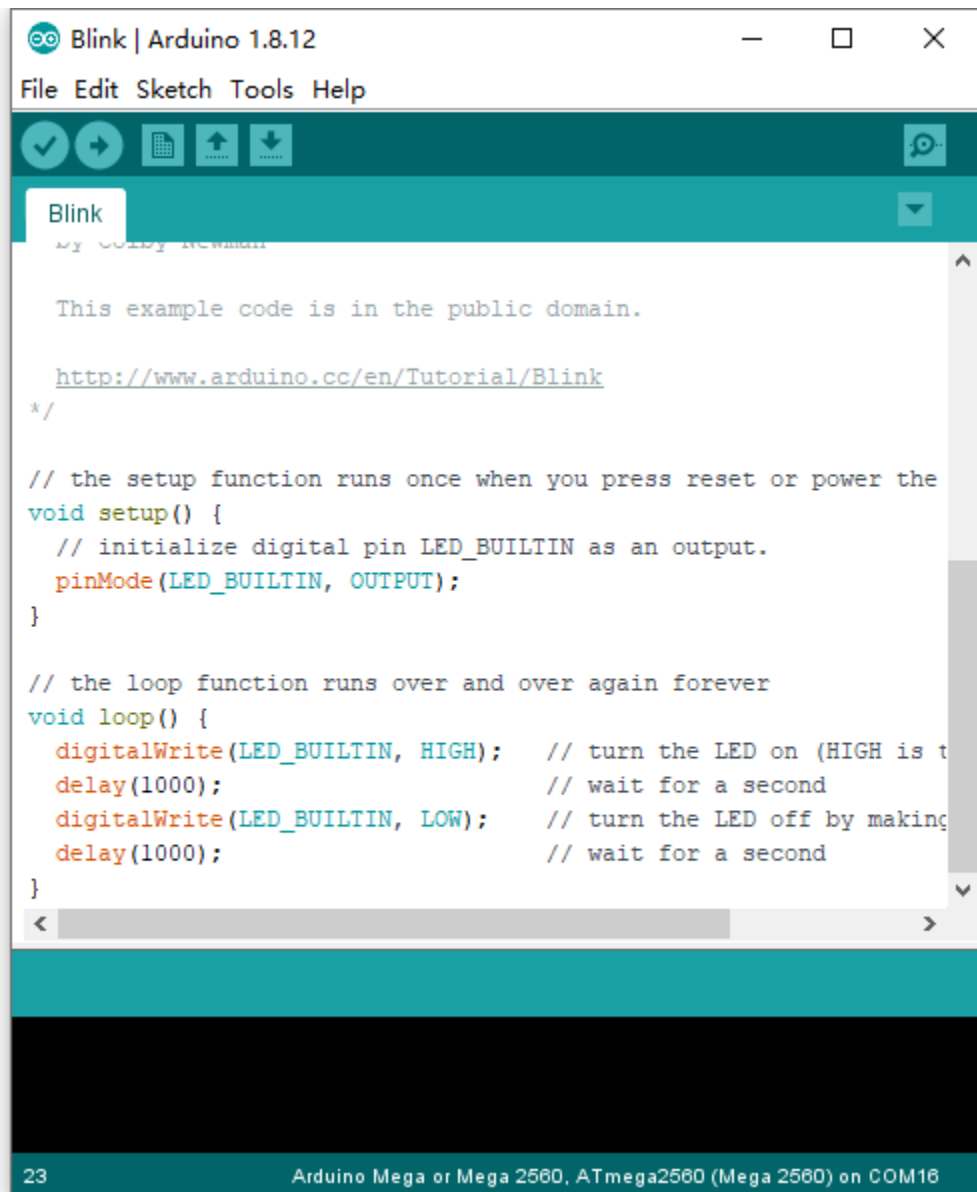


- A- Used to verify whether there is any compiling mistakes or not.
- B- Used to upload the sketch to your Arduino board.
- C- Used to create shortcut window of a new sketch.
- D- Used to directly open an example sketch.
- E- Used to save the sketch.
- F- Used to send the serial data received from board to the serial monitor.

#### 4.4 Start A Program


Open the file to select **Example**, and click **BASIC>BLINK**, as shown below:

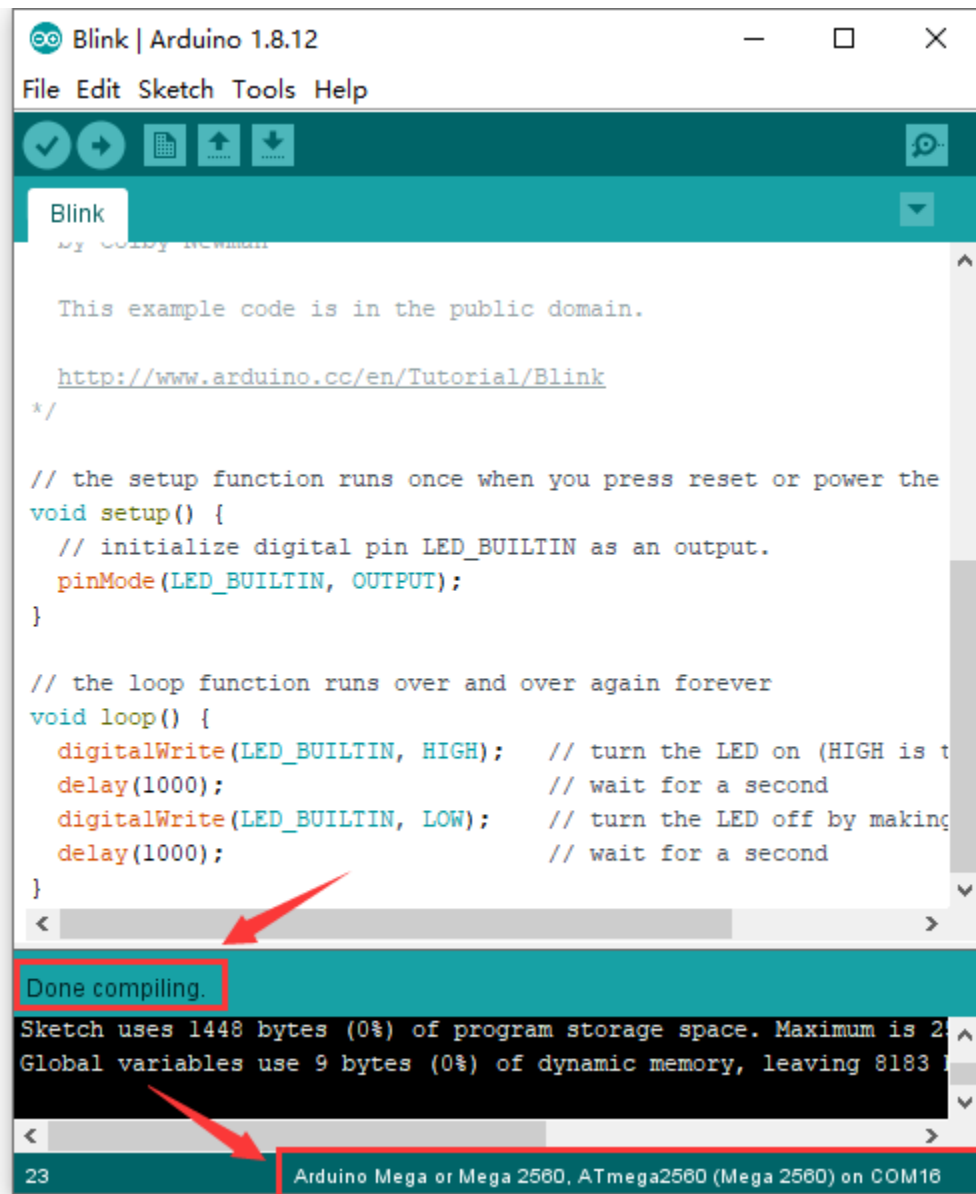




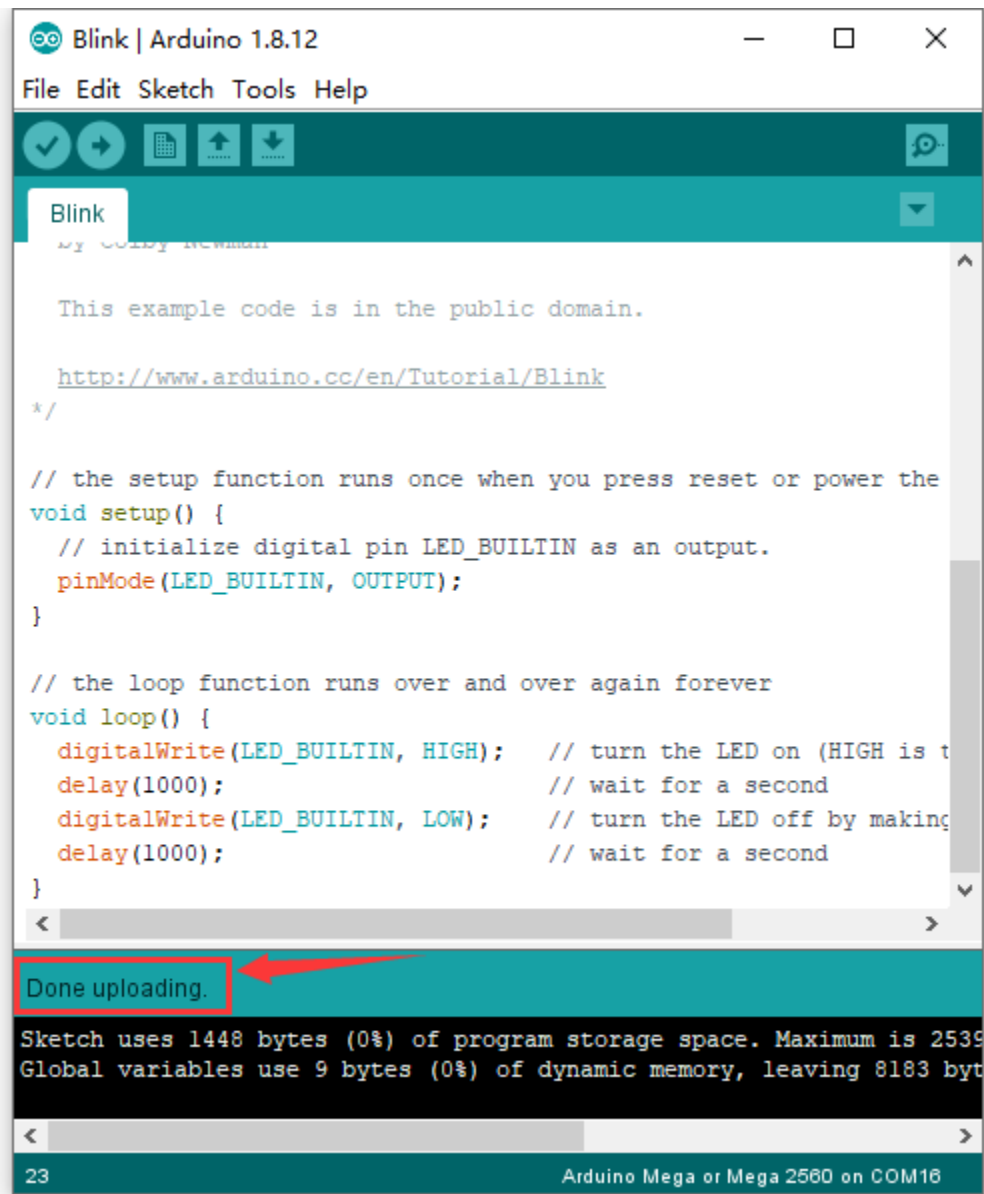
Set the correct **COM port**, and the corresponding board and COM port are shown on the lower right of IDE.



Click  to start compiling the program, and check errors.



Click  to upload the program



After the program is uploaded successfully, the onboard LED blinks. Congratulation, you finish the first program.

## 5. MAC System

### 5.1 Install Arduino IDE on MAC System

The installation instruction is as same as the chapter 4.1, as shown below:



## ARDUINO 1.8.12

Arduino IDE that can be used with any Arduino board, including the Arduino Yún and Arduino DUE. Refer to the [Getting Started](#) page for Installation instructions.

[See the release notes.](#)

**Windows** Installer

**Windows** ZIP file for non admin install

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

**Linux** 32 bits

**Linux** 64 bits

**Linux** ARM 32

**Linux** ARM 64

**Source**

5.2 Download the Driver of CH340

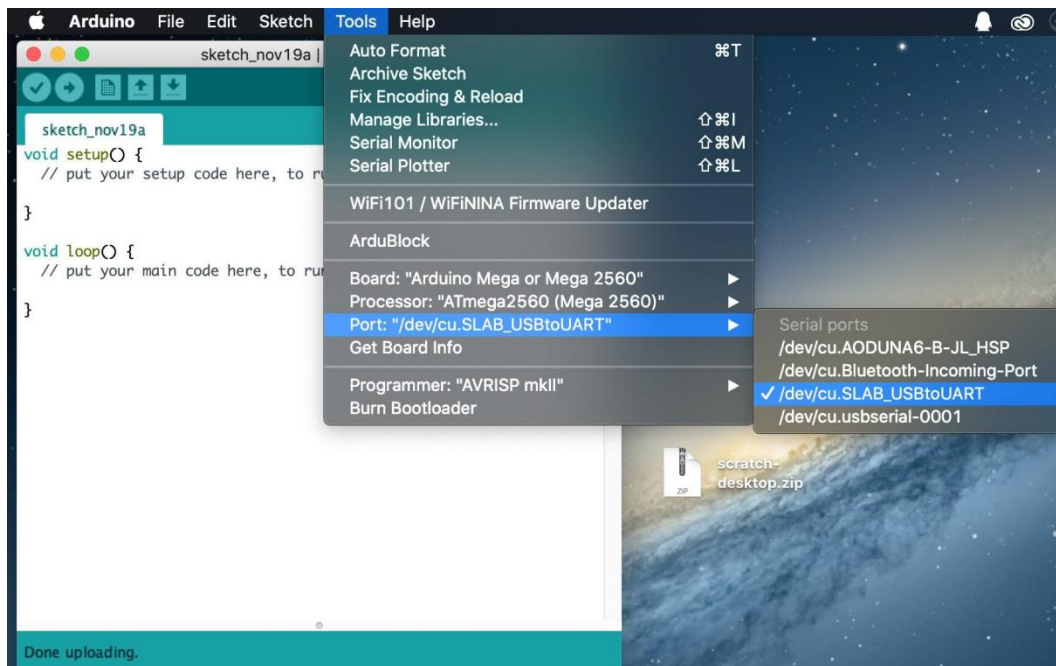
<https://fs.keyestudio.com/CH340-MAC>

5.3 How to Install the Driver of CH340

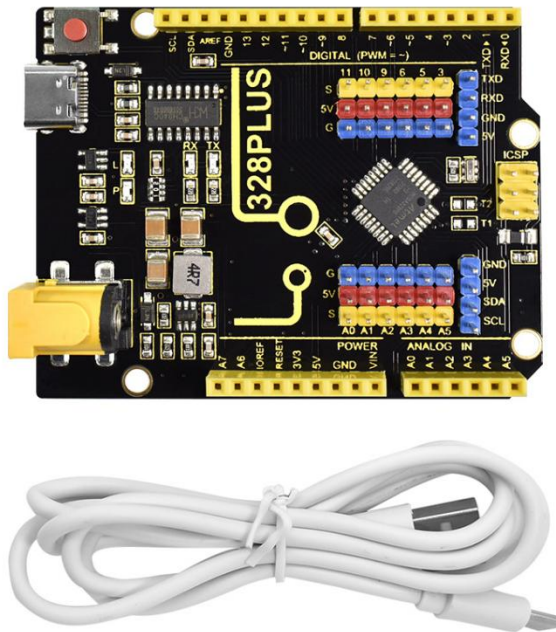
[https://wiki.keyestudio.com/Download\\_CH340\\_Driver\\_on\\_MAC\\_System](https://wiki.keyestudio.com/Download_CH340_Driver_on_MAC_System)

5.4 Arduino IDE Setting

The setting method is as same as the chapter 4.3 except from COM port, as shown below.



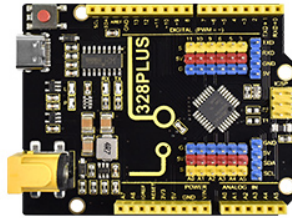
## KS0546 KEYESTUDIO 328 PLUS DEVELOPMENT BOARD



### 1.Description

Doing experiments with electronic products, we often program on the Arduino IDE development environment with Arduino series microcontrollers.

Keyestudio 328 PLUS control board is fully compatible with Arduino IDE development environment. It is as same as the Arduino UNO R3 board. Moreover, some improvements we made highly strengthen its function(as shown below). In order to wire efficiently, we equip it with a 1m USB cable with type-c interface for you.



Keyestudio 328 PLUS

VS



Arduino UNO R3

1. USB CH340C, stability and compatibility are better for-to-turn chip

2. Two more IO mouth, A6,A7, the best

3. Extended serial communication and I2C interface, can be easily connected to similar devices

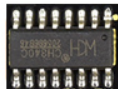
4. Unique DC-DC power supply design, working voltage 5 V, the current of 2 A, can directly drive some high current load, such as steering gear and motor

5. Extension of 6 PWM ports and 6 analog port interfaces to connect sensors directly

6. Extended serial communication and I2C interface, can be easily connected to similar devices

7. Input voltage 6-15 V, wider voltage range, choose more

8. Choose the current more popular type-c interface, beautiful and generous, faster transmission speed



1. USB turnstile chip is 16 U2, Some systems are not compatible



2. Design did not leave these 2 IO ports, defective



3. No expansion port, more complex connection required



4. Working voltage 5 V, the current only 1 A, can not drive the equipment with high current.



5. No expansion port, more complex connection required



6. No expansion port, more complex connection required



7. Input voltage 7-12 V, optional power supply mode is not



8. Adopt traditional square USB interface, more common

## 2.Specification

Microcontroller ATMEGA328PB

USB to serial chip: CH340C

Working voltage: 5V

External power: DC 6-15V (recommend 9V)

Digital I / O pins: 14 (D0-D13)

PWM channel: 6 (D3 D5 D6 D9 D10 D11)

Analog input channel (ADC): 8 (A0-A7)

Each I / O Port of DC output capability : 10 mA

Output capability of 3.3V port: MAX150 mA

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

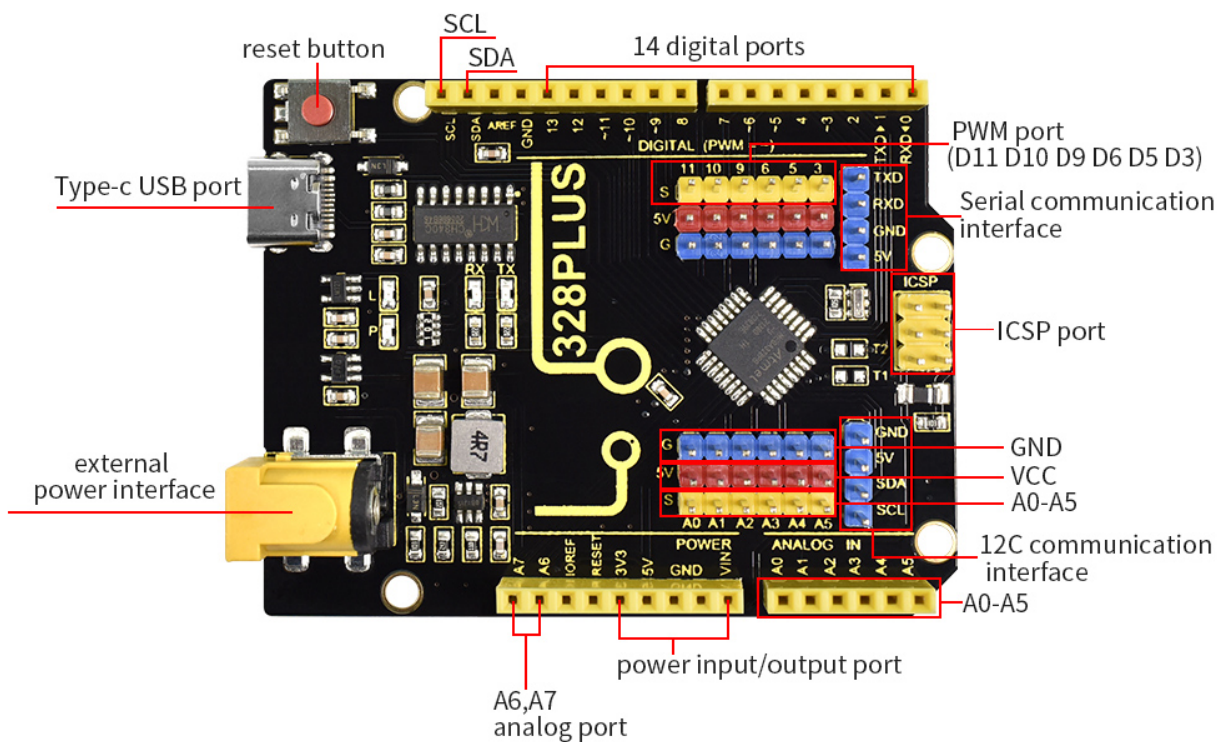
SRAM: 2 KB (ATMEGA328PB)

EEPROM: 1 KB (ATMEGA328PB)

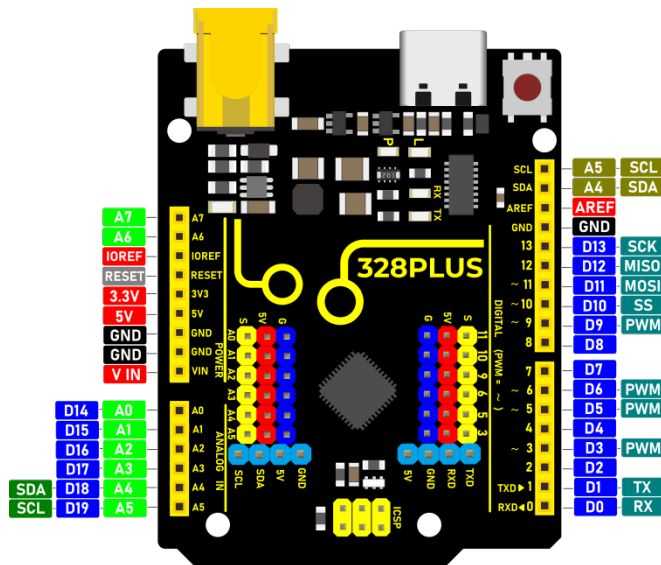
Clock speed: 16MHz

On-board LED pin: D13

### 3. Interfaces



### 4. Specialized Functions of Some Pins:



Serial communication interface: D0 is RX, D1 is TX

PWM interface (pulse width modulation): D3 D5 D6 D9 D10 D11

External interrupt interface: D2 (interrupt 0) and D3 (interrupt 1)

SPI communication interface: D10 is SS, D11 is MOSI, D12 is MISO, D13 is SCK

IIC communication port: A4 is SDA, A5 is SCL

## 5. Install Arduino IDE and Driver

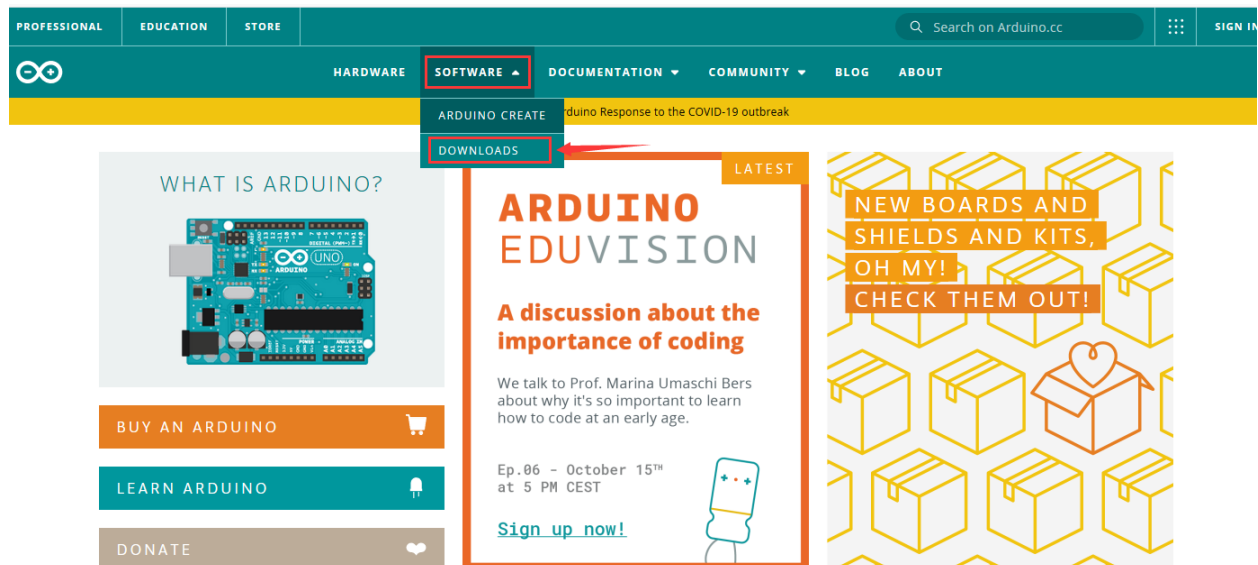
### 5.1 Download the Arduino IDE

When getting this control board, we need to install Arduino IDE.

Enter the website <https://www.arduino.cc/> click

SOFTWARE

DOWNLOADS



Download the version you want and the latest one is available for downloading too.

## Download the Arduino IDE



**ARDUINO 1.8.13**

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 7 and up

**Windows** ZIP file for non admin install

**Windows app** Requires Win 8.1 or 10

**Get**

**Mac OS X** 10.10 or newer

**Linux** 32 bits

**Linux** 64 bits

**Linux** ARM 32 bits

**Linux** ARM 64 bits

Release Notes

Source Code

Checksums (sha512)

Alternatively, you could select previous release.

In this project, we use 1.8.12 version

### Previous Releases

Download the [previous version of the current release](#) the classic [Arduino 1.0.x](#), or the [Arduino 1.5.x Beta version](#).

All the [Arduino 00xx versions](#) are also available for download. The Arduino IDE can be used on Windows, Linux (both 32 and 64 bits), and Mac OS X.

Click [previous version of the current release](#) to view the below page

**ARDUINO 1.8.12**

Arduino IDE that can be used with any Arduino board, including the Arduino Yún and Arduino DUE. Refer to the [Getting Started](#) page for Installation instructions.

[See the release notes.](#)

**Windows** Installer

**Windows** ZIP file for non admin install

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

**Linux** 32 bits

**Linux** 64 bits

**Linux** ARM 32

**Linux** ARM 64

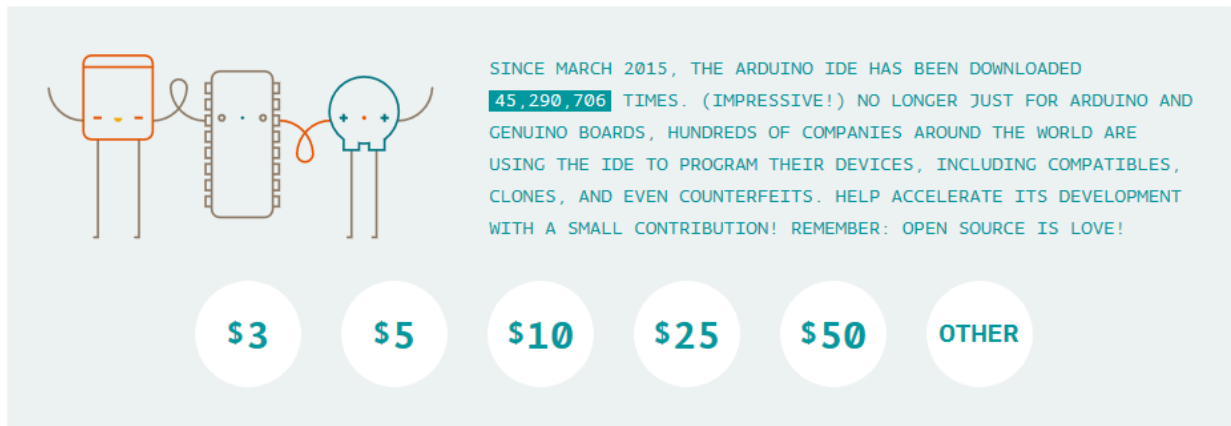
**Source**



Click [Windows Installer](#) to download an installer of Arduino 1.8.12 version which needs to be installed manually. When you tap [Windows ZIP file for non admin install](#) a zip file of Arduino 1.8.12 version will be directly downloaded, and you only need to unzip it to finish the installation.

## Contribute to the Arduino Software

Consider supporting the Arduino Software by contributing to its development. (US tax payers, please note this contribution is not tax deductible). [Learn more on how your contribution will be used.](#)



JUST DOWNLOAD

CONTRIBUTE & DOWNLOAD

Click icon

to download Arduino IDE.

### 5.2 Installing Driver

#### 1. Windows system

Download CH340 driver

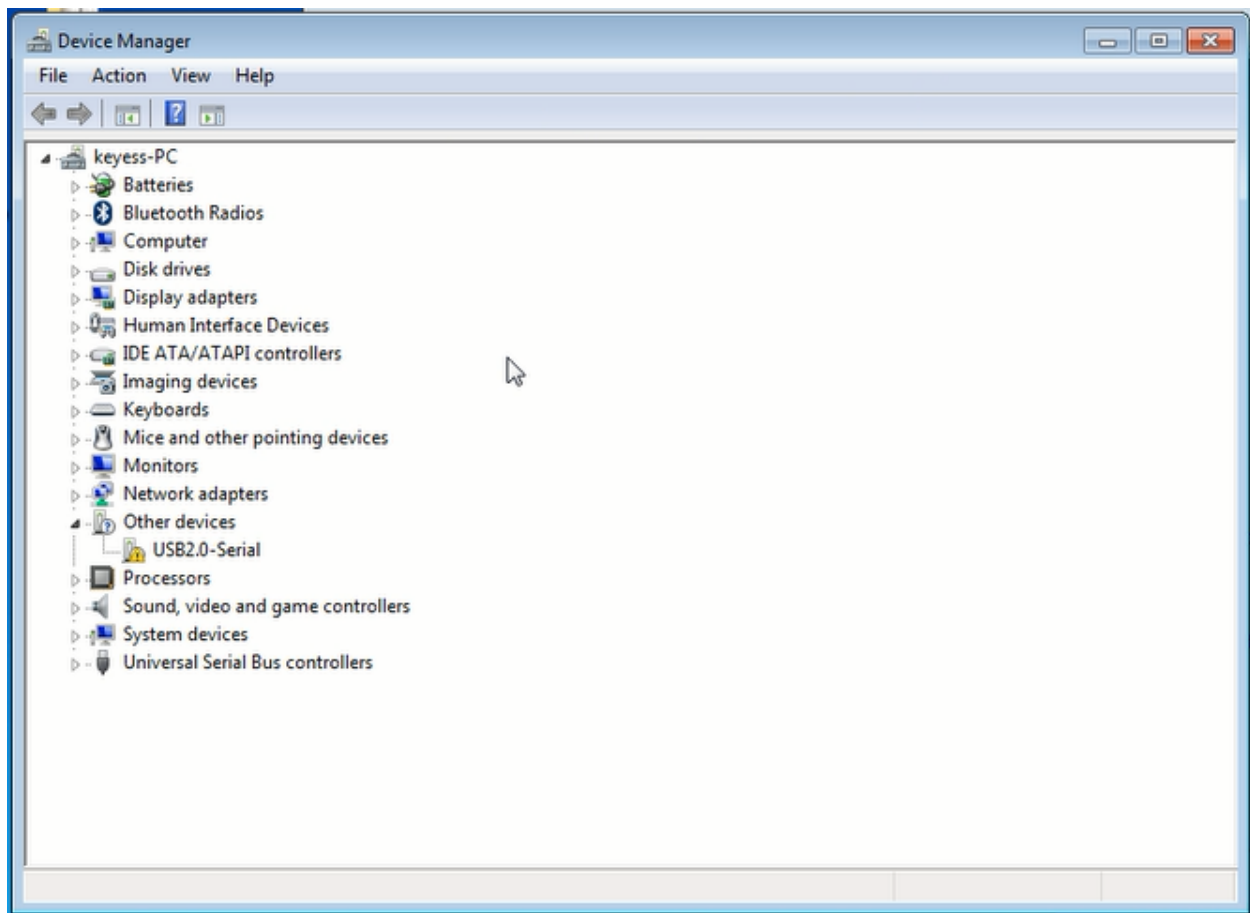
<https://sparks.gogo.co.nz/ch340.html>

Next, let's install Arduino driver.

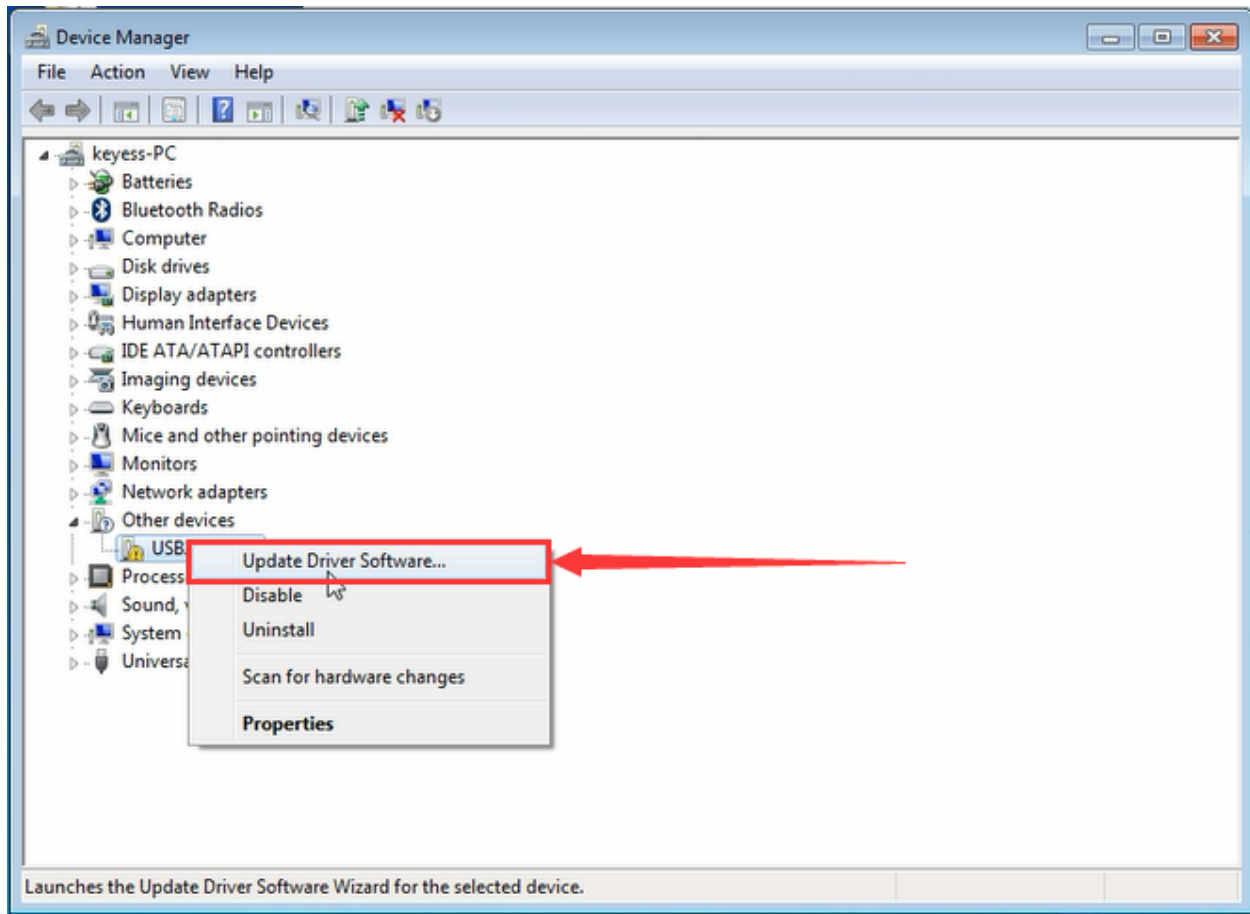
For different operating system, there may be slight difference in installation method. Below is an example in WIN 7.

When you connect Arduino Uno to your computer the first time, right click "Computer" —> "Properties" —> "Device manager", you can see "USB2.0-Serial".

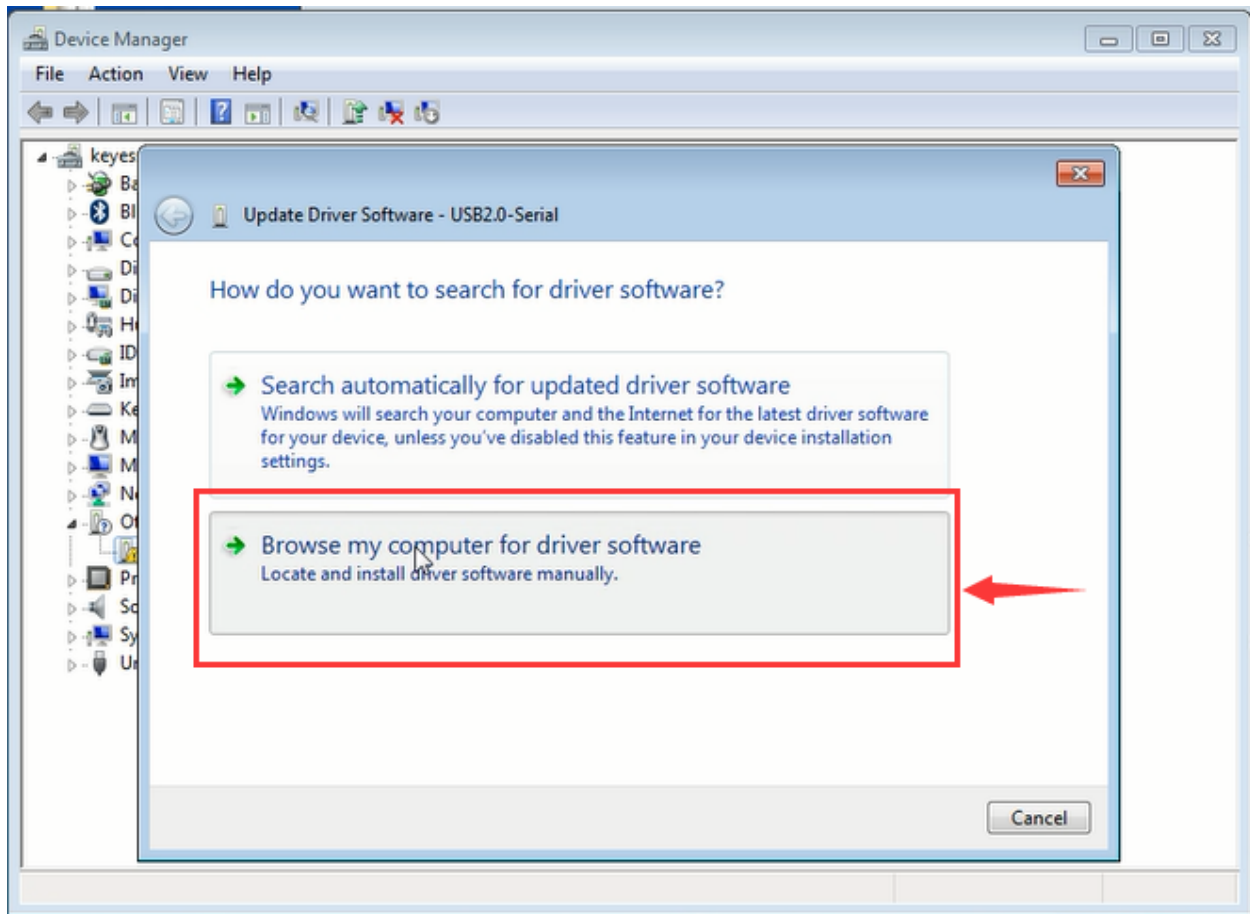




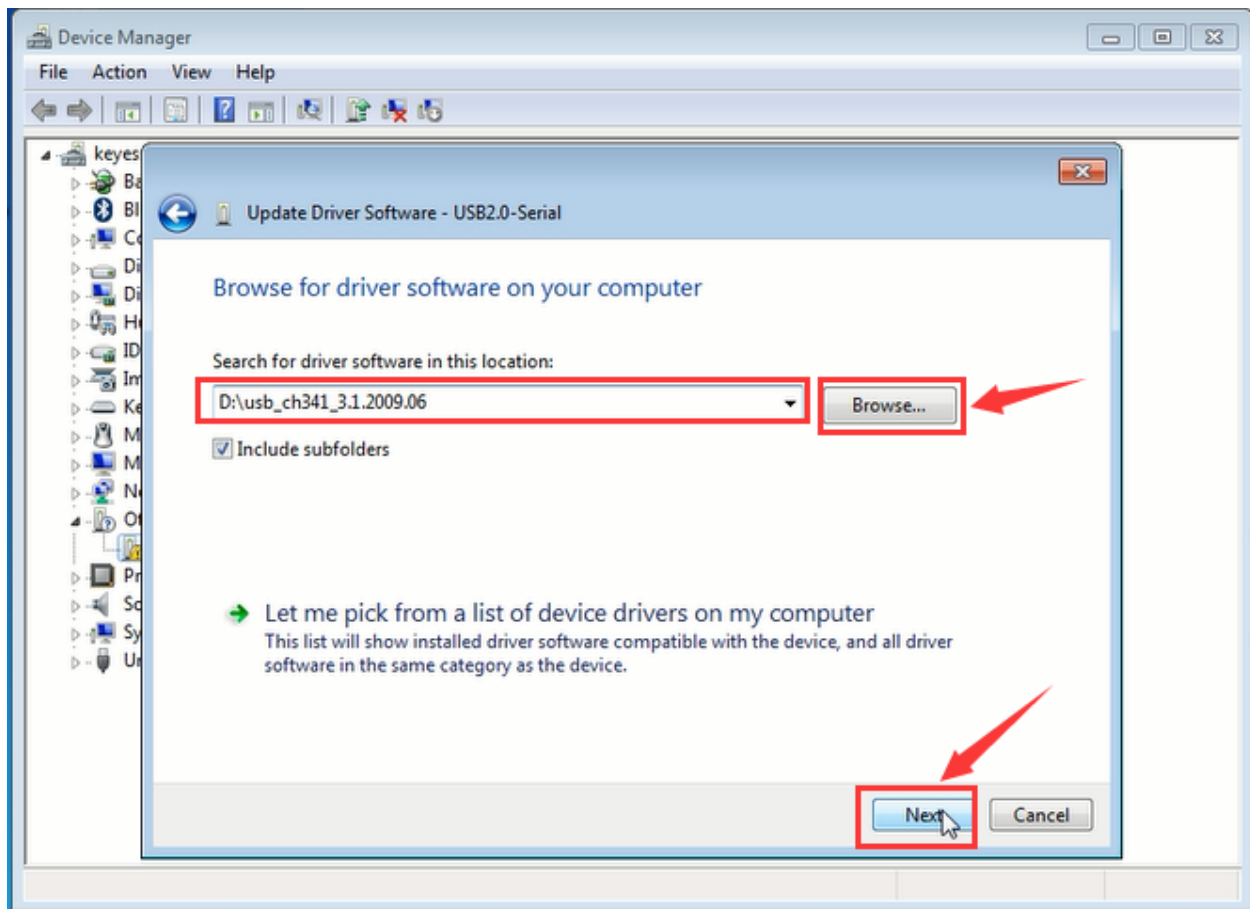
Click “USB2.0-Serial”, select “Update Driver software”.



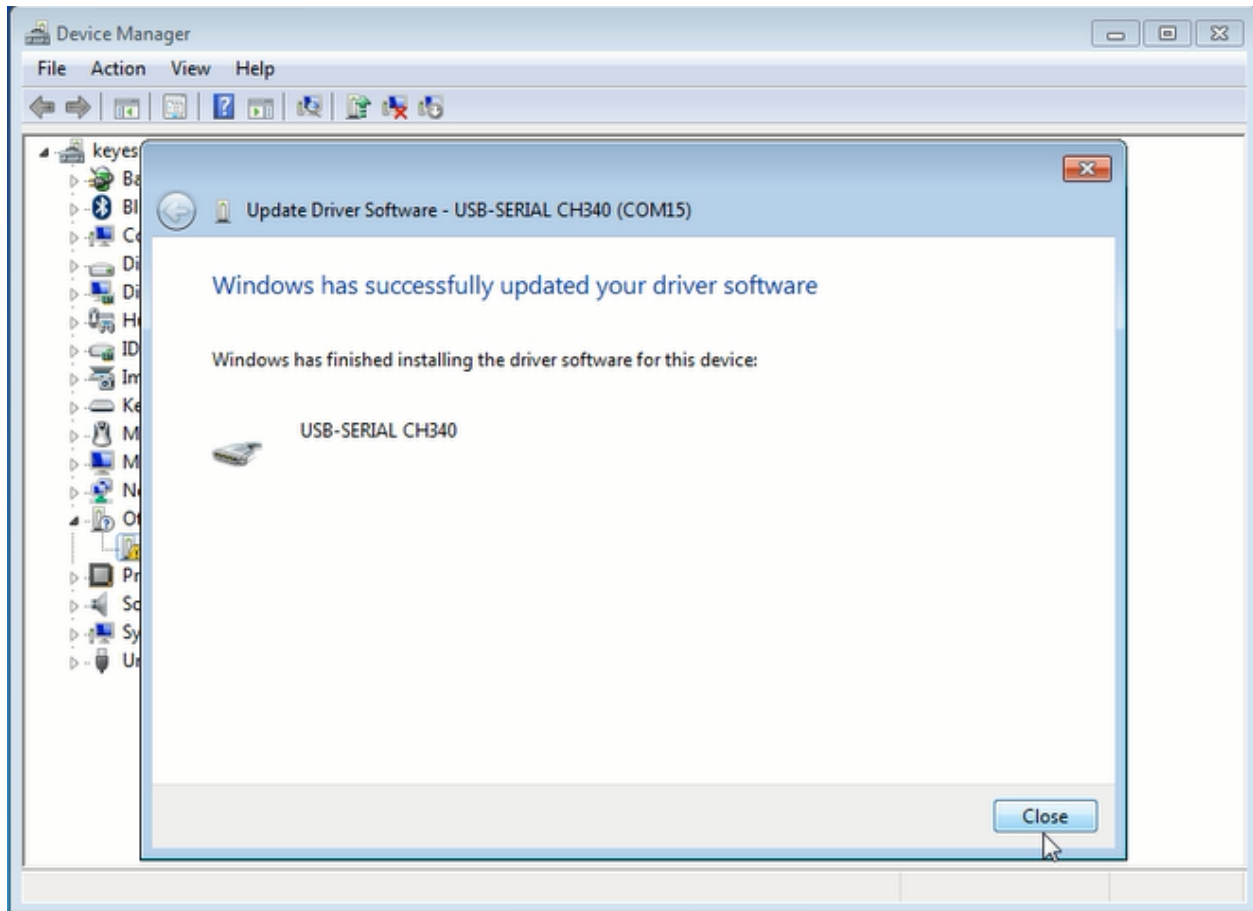
In this page, click “Browse my computer for driver software”.



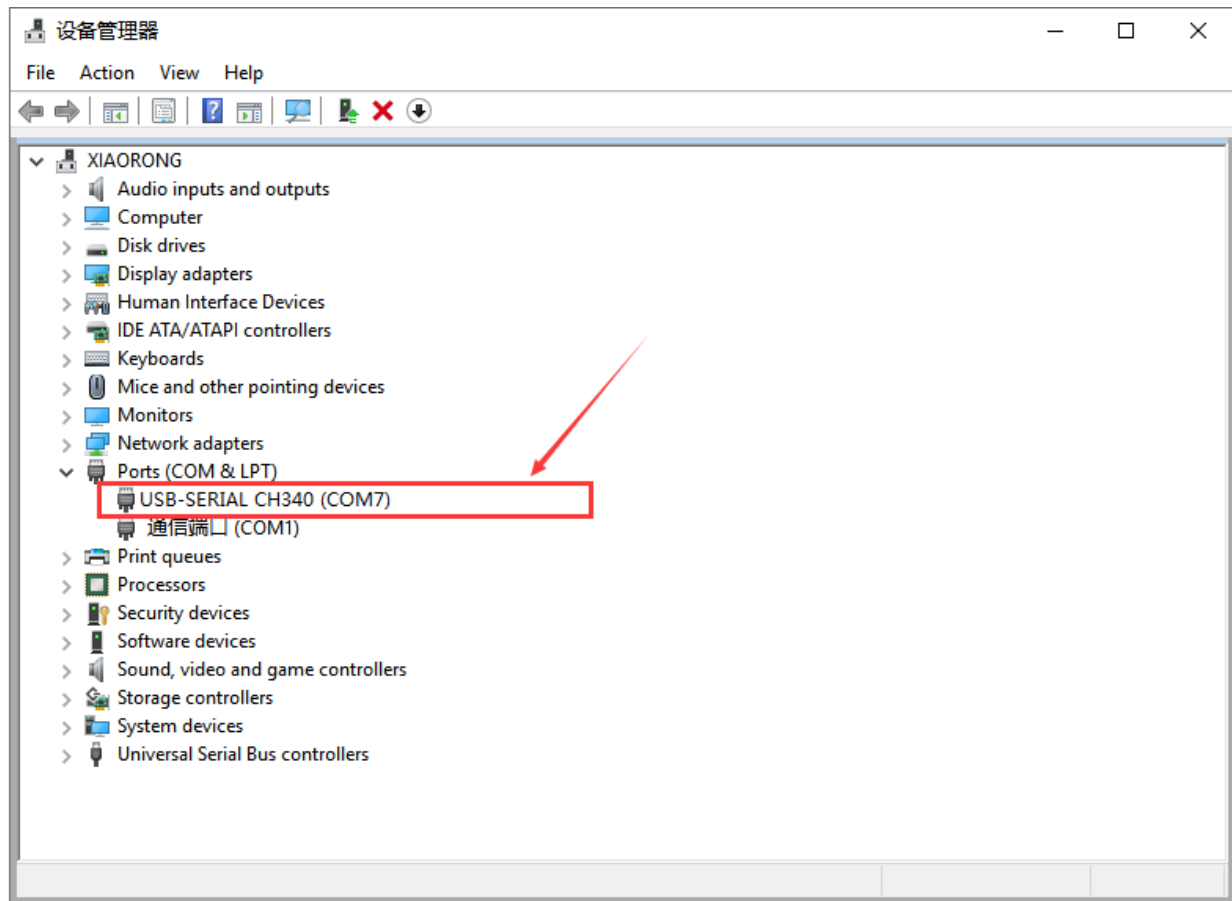
Find the “usb\_ch341\_3.1.2009.06” file, Click “Next”.



Installation completed; click “Close”.



After driver is installed, go to "Device manager" again. Right click "Computer" —> "Properties"—> "Device manager", you can see CH340 device as below figure shows, also the Com port info.



## 2.MAC system

Download CH340 driver

<https://sparks.gogo.co.nz/ch340.html>

1. Click V1.5 CH340 MacOS Driver package

## Older Windows Driver Version and Instructions

- Download the [Windows CH340 Driver](#)
- Unzip the folder.
- **If you are running a 64Bit Windows:** — run the SETUP\_64.EXE installer.
- **If you are running a 32Bit Windows:** — run the SETUP\_32.EXE installer.
- If you don't know, try the 64 bit and if it doesn't work, the 32-bit.
- In the Arduino IDE when the CH340 is connected you will see a COM Port in the Tools > Serial Port menu, the COM number for your device may vary depending on your system.

## Macintosh

(Manufacturer's Chinese Info Link)

The following github has up to day pkg files for 1.3, 1.4 and 1.5 at time of writing, thanks to Joshua Wallis for bringing this to my attention...

<https://github.com/adrianmihalko/ch340g-ch34g-ch34x-mac-os-x-driver>

Direct Download Links (from the github repo above):

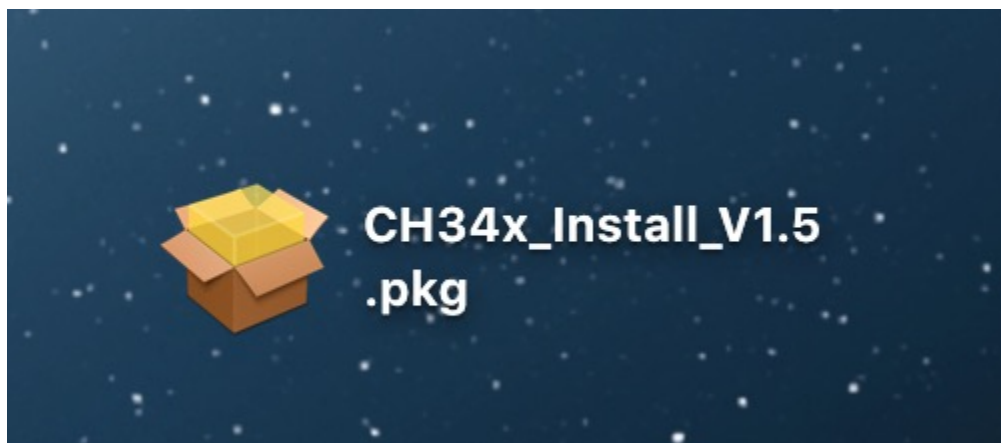
[V1.5 CH340 MacOS Driver Pkg](#)

[V1.4 CH340 MacOS Driver Pkg](#)

[V1.3 CH340 MacOS Driver Pkg](#)

See the github link above for installation instructions if you need them.

1. After the download, seen as below:



1. click installation package and tap Continue

## Older Windo

- Download the [Window](#)
- Unzip the folder.
- **If you are running a**
- **If you are running a**
- If you don't know, try t
- In the Arduino IDE wh

## Macintosh

(Manufacturer's Chine

The following github h

<https://github.>

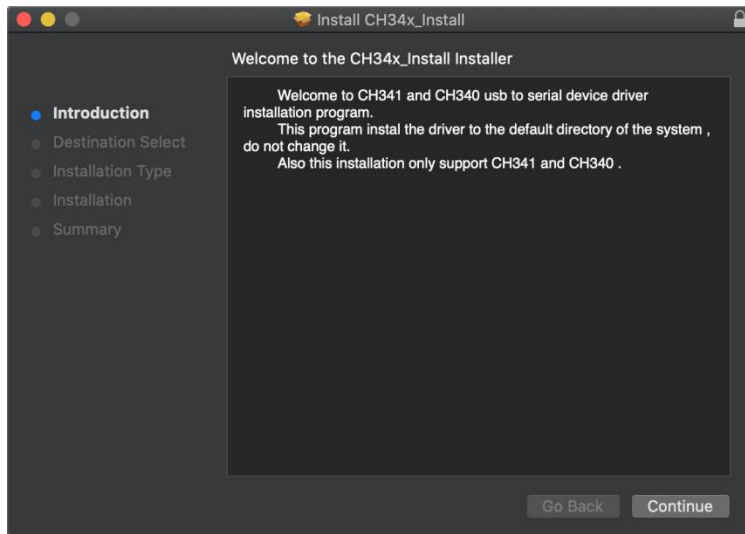
Direct Download Lin

[V1.5 CH340 MacOS D](#)

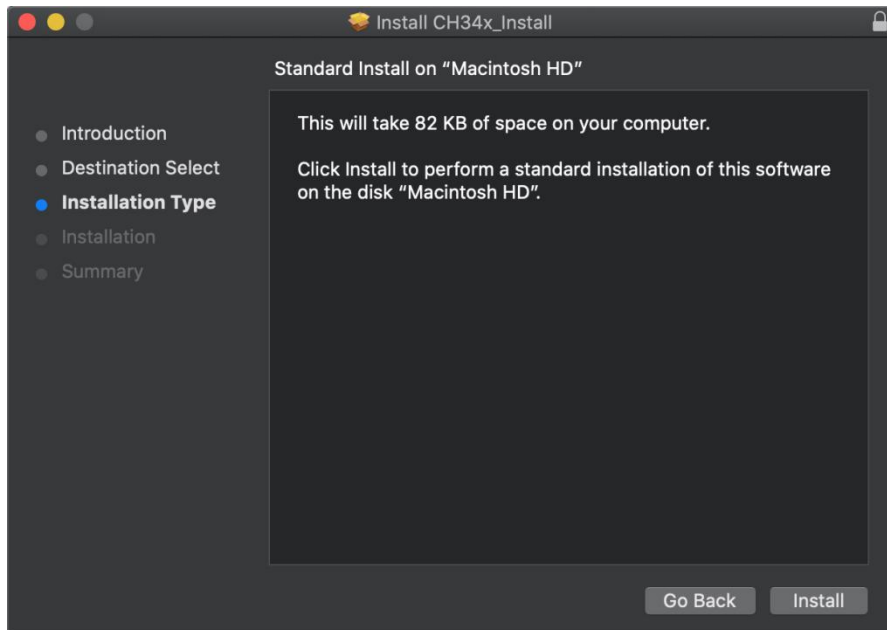
[V1.4 CH340 MacOS D](#)

[V1.3 CH340 MacOS D](#)

See the github link ab

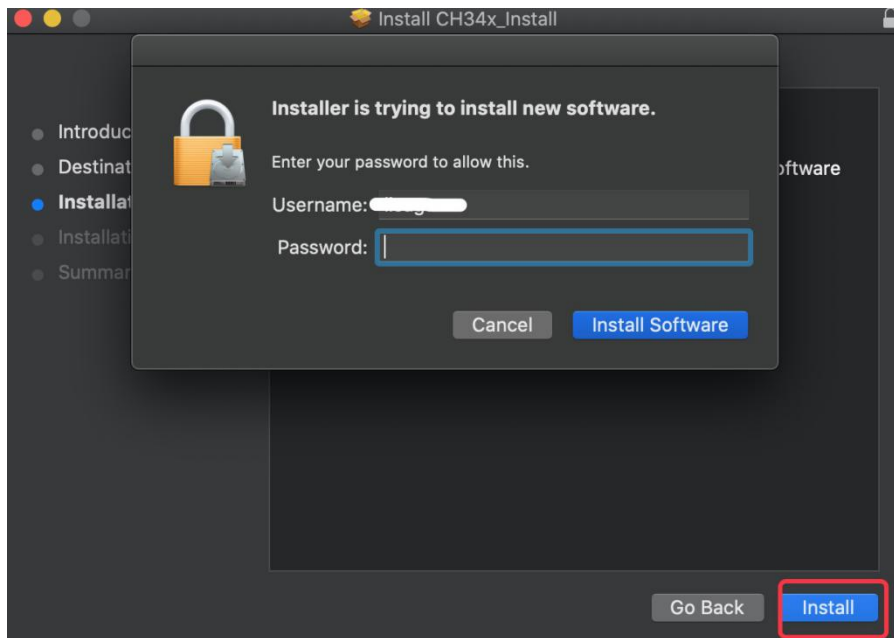


1. Click Install

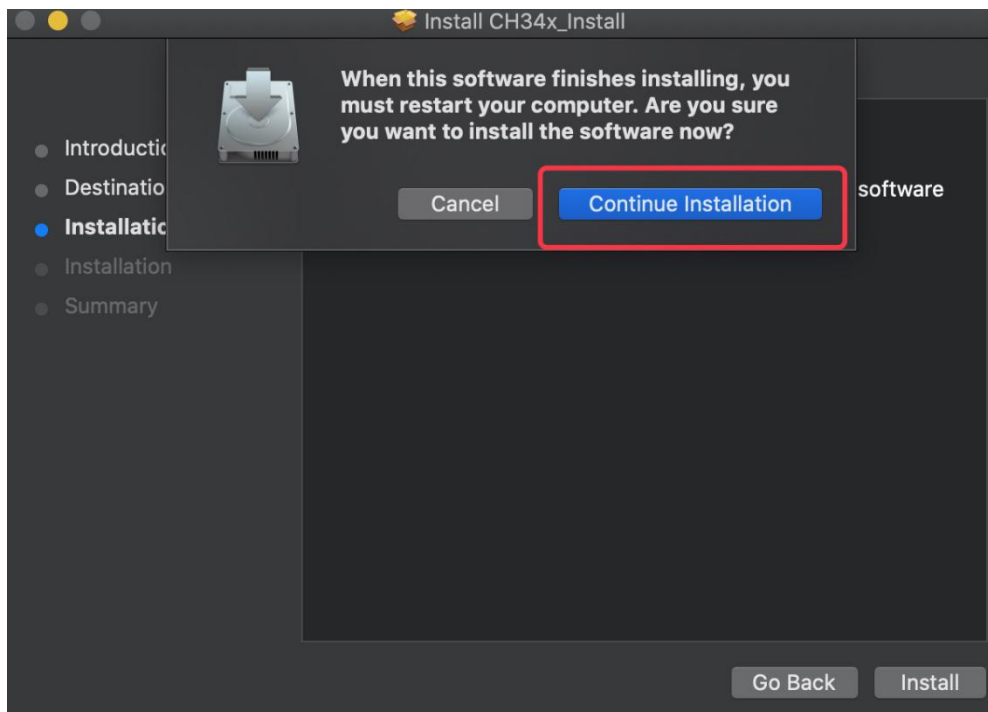


1. Input your user password and click Install Software

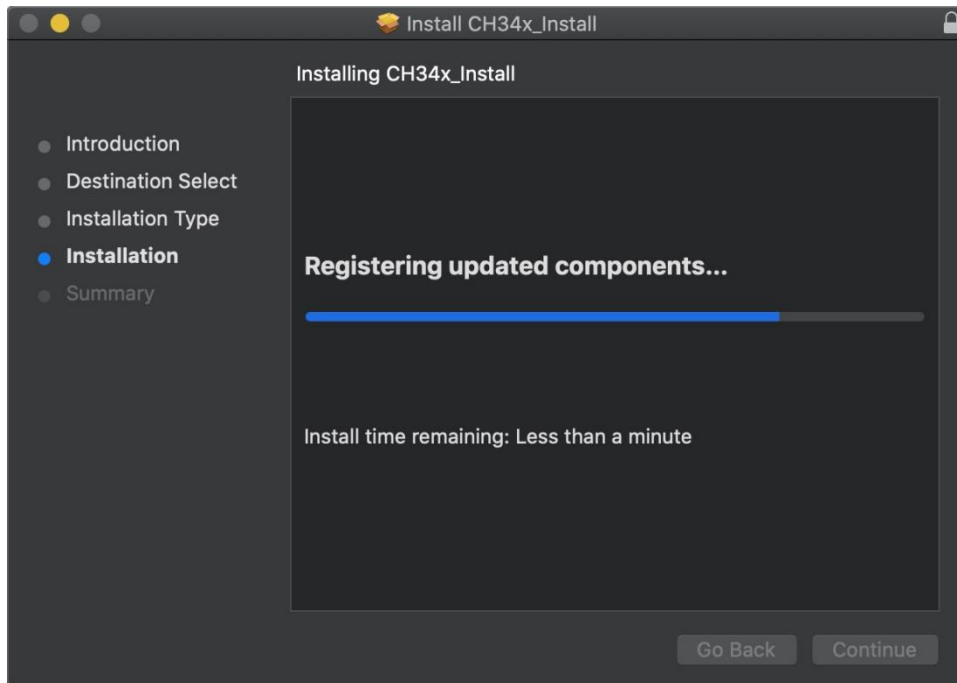




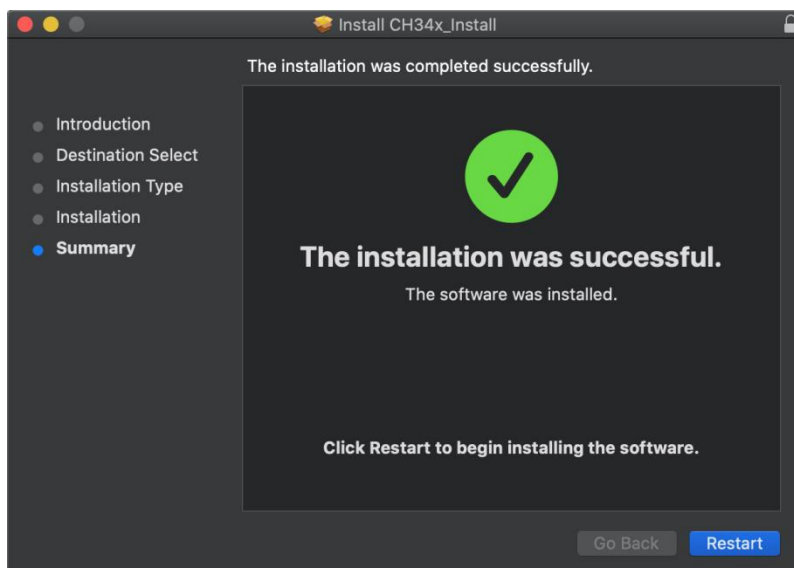
1. Tap Continue Installation



1. Wait to install




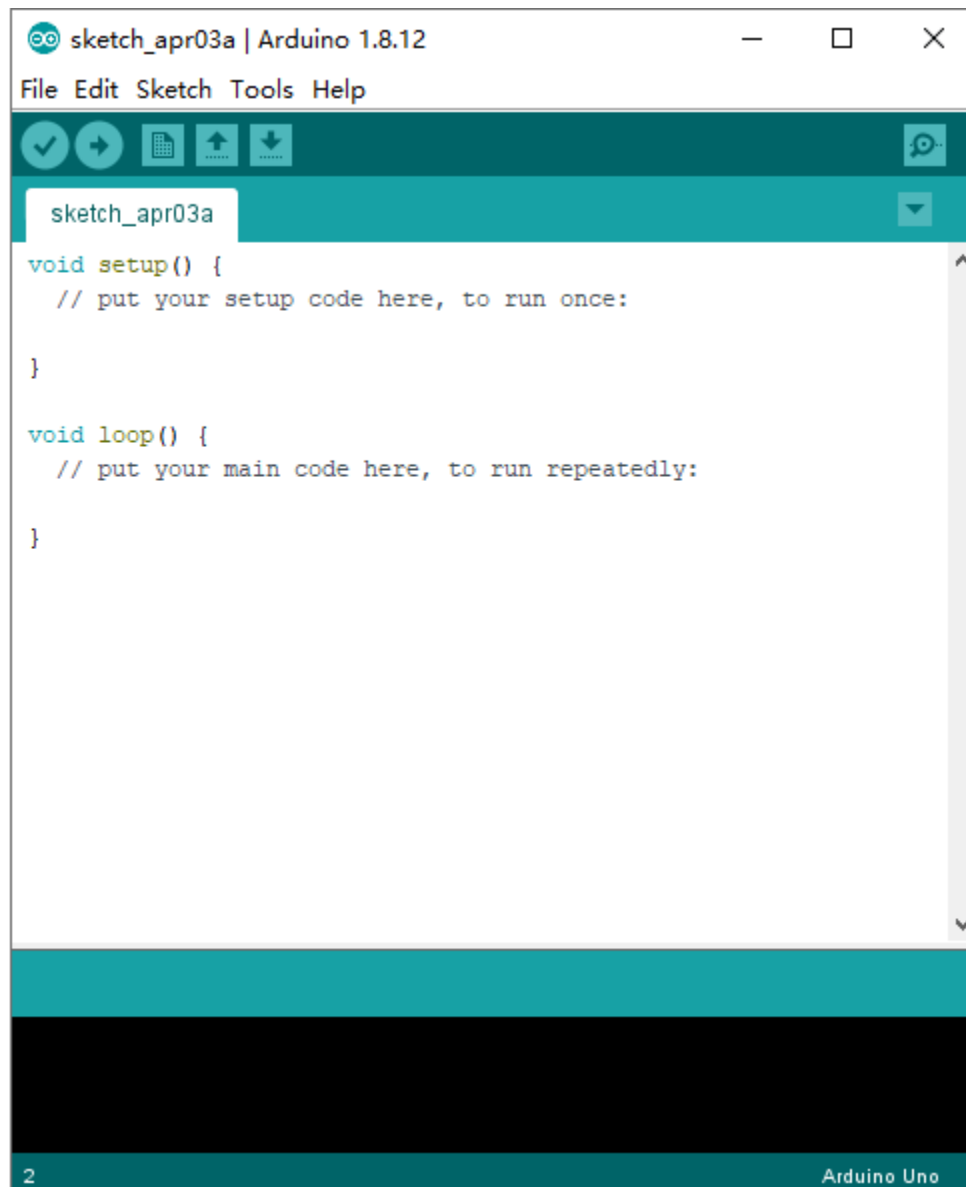
1. Click Restart after the installation is finished



### 5.3 Arduino IDE Setting

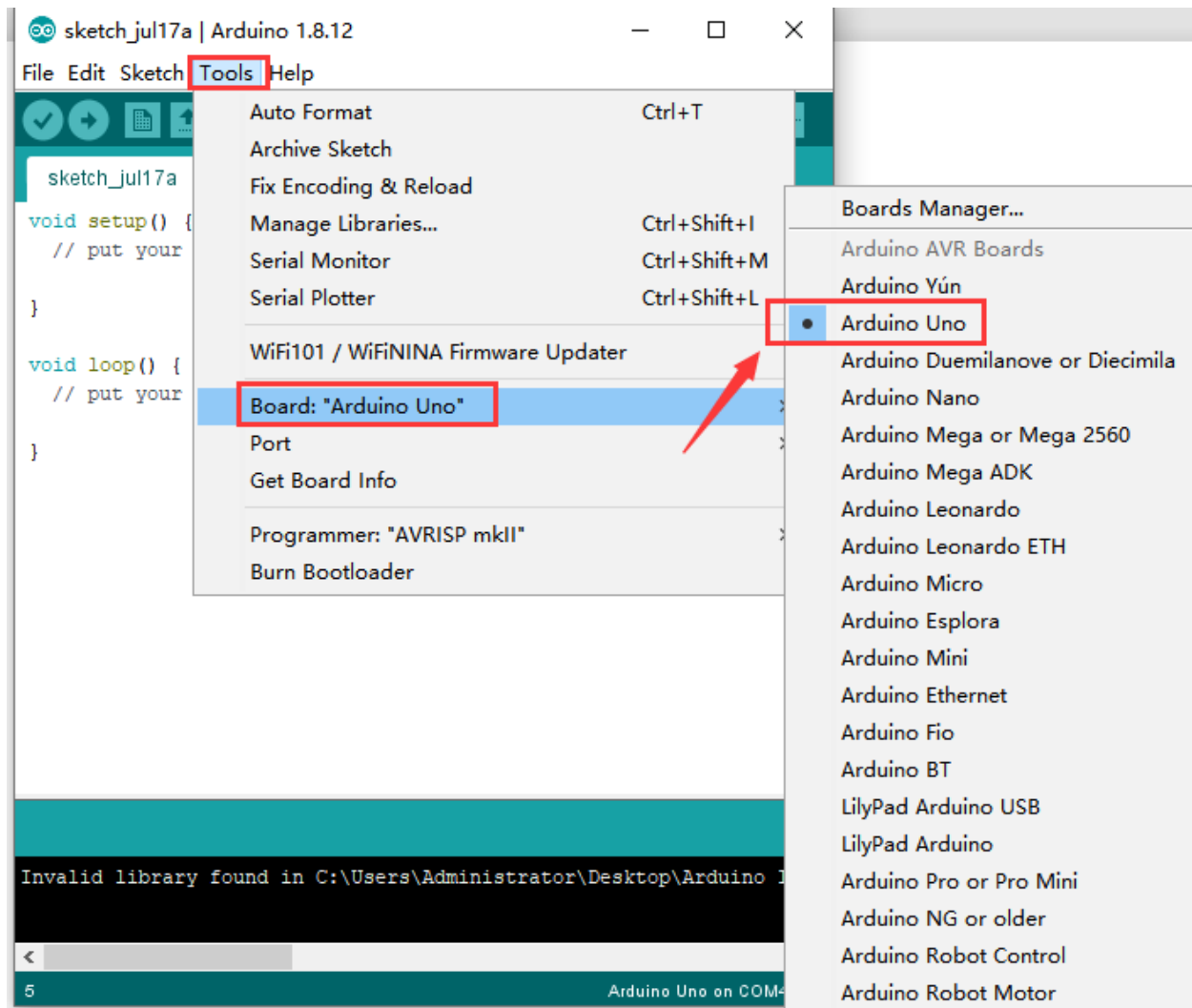


Click  icon open Arduino IDE.

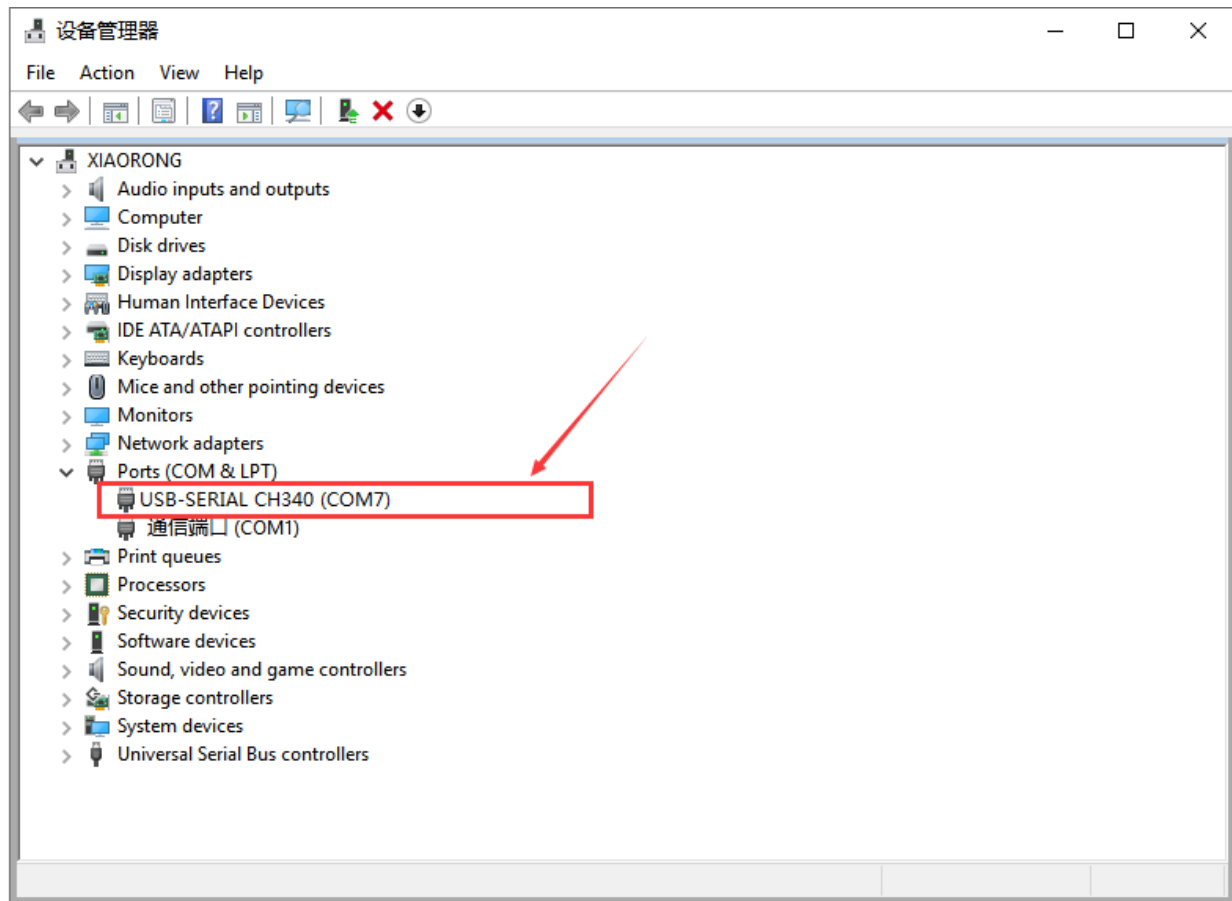


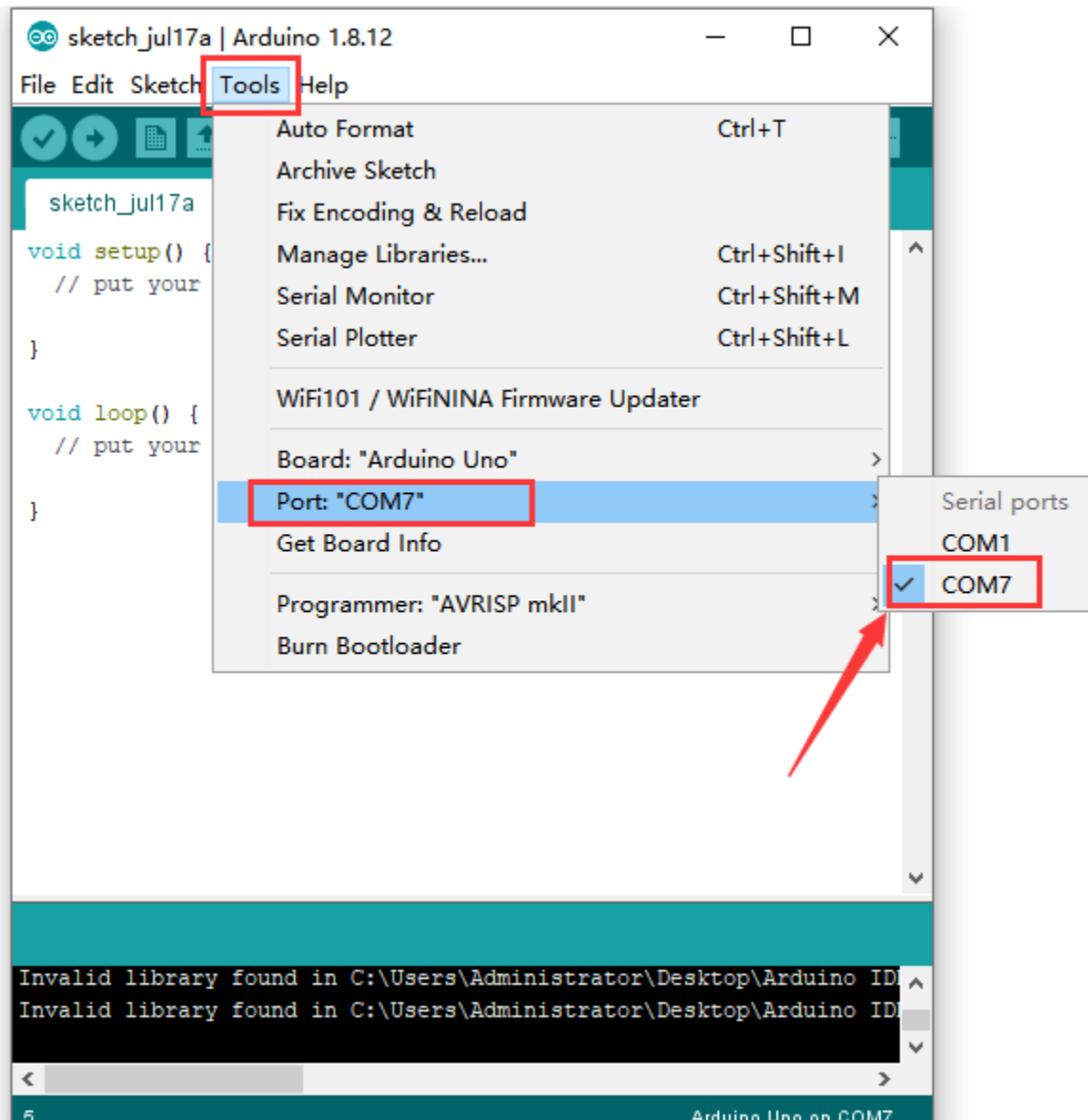
To avoid the errors when uploading the program to the board, you need to select the correct Arduino board that matches the board connected to your computer.

Then come back to the Arduino software, you should click Tools→Board, to select the board. (as shown below)

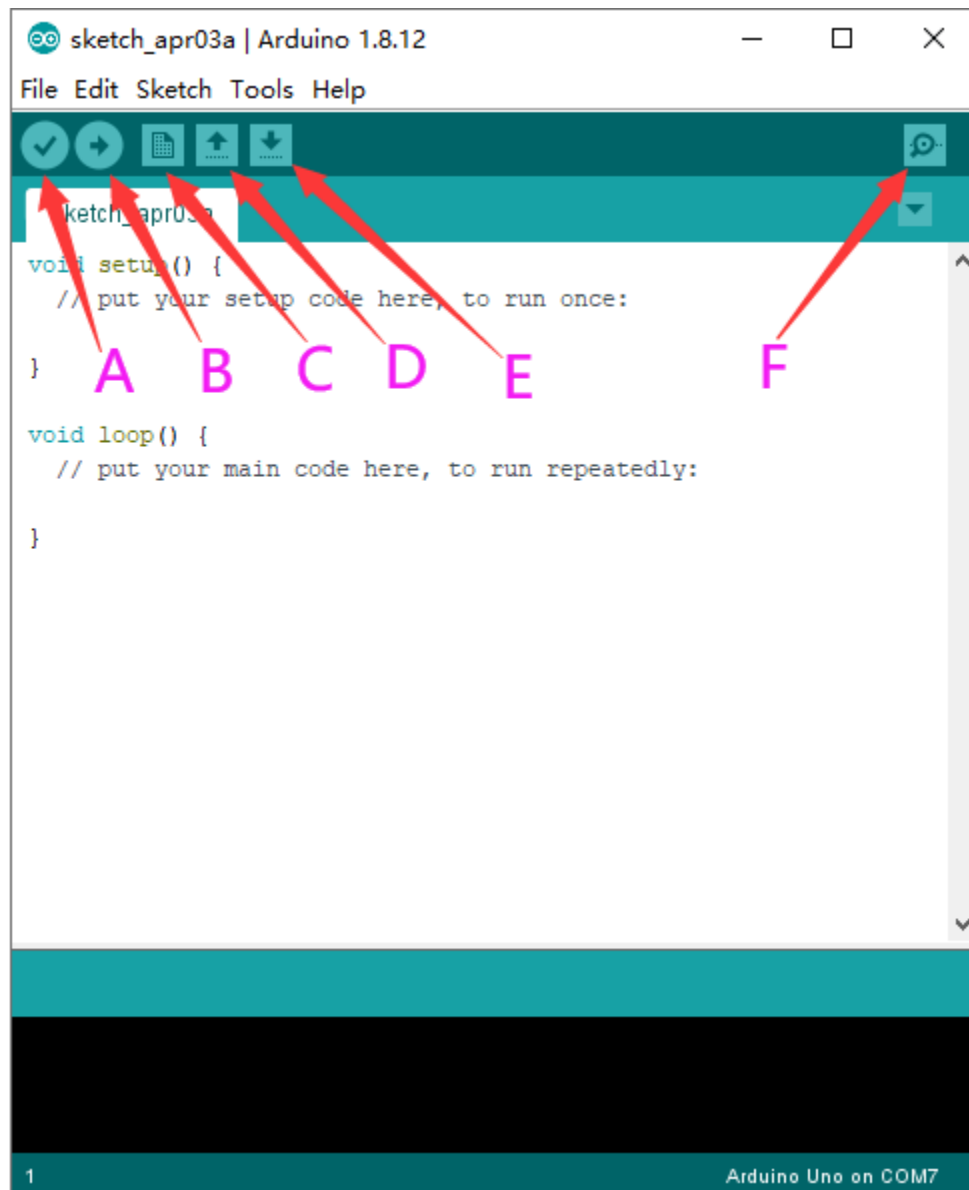


Then select the correct COM port (you can see the corresponding COM port after the driver is successfully installed)





Before uploading the program to the board, let's demonstrate the function of each symbol in the Arduino IDE toolbar.



A- Used to verify whether there is any compiling mistakes or not.

B- Used to upload the sketch to your Arduino board.

C- Used to create shortcut window of a new sketch.

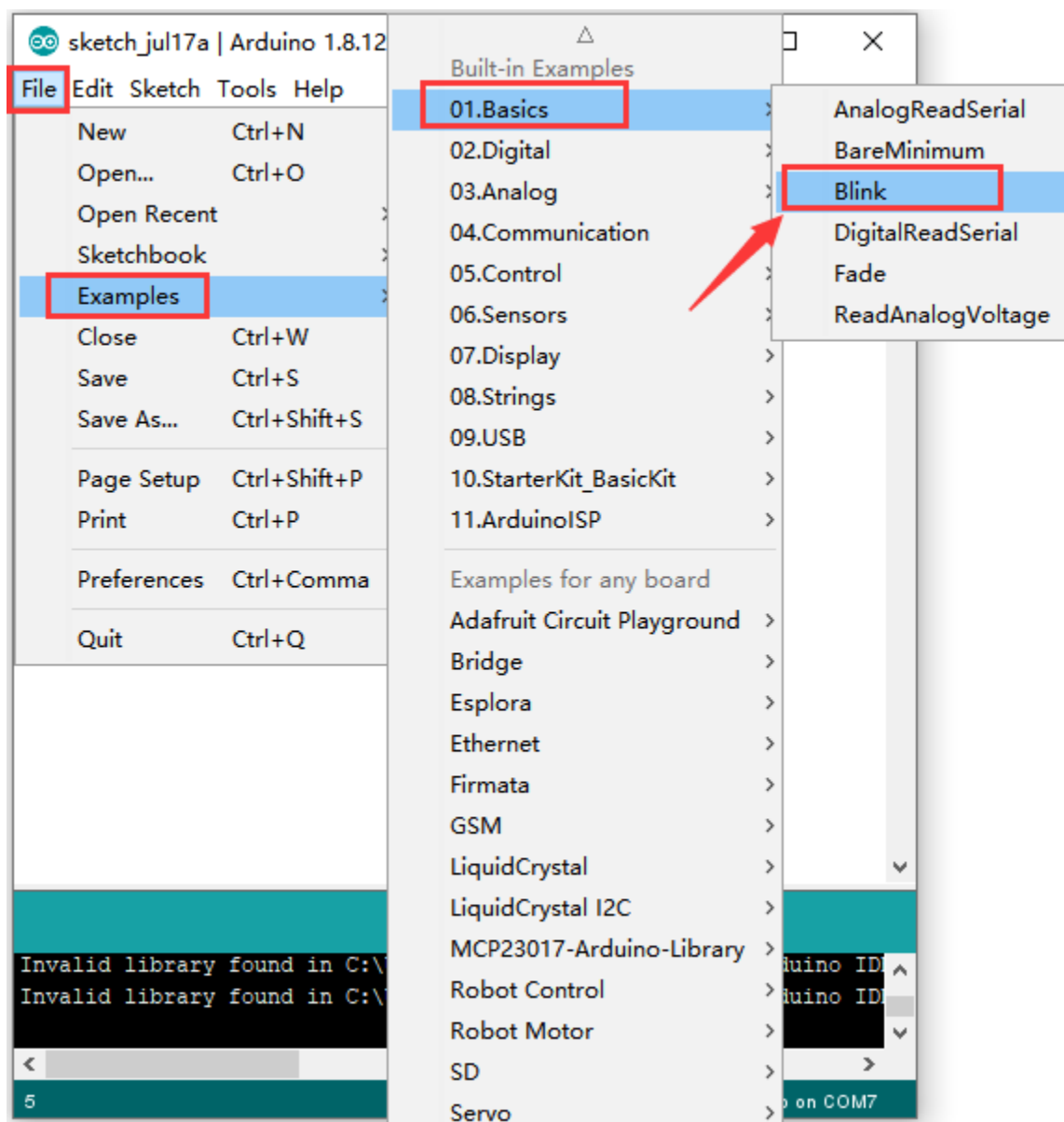
D- Used to directly open an example sketch.

E- Used to save the sketch.

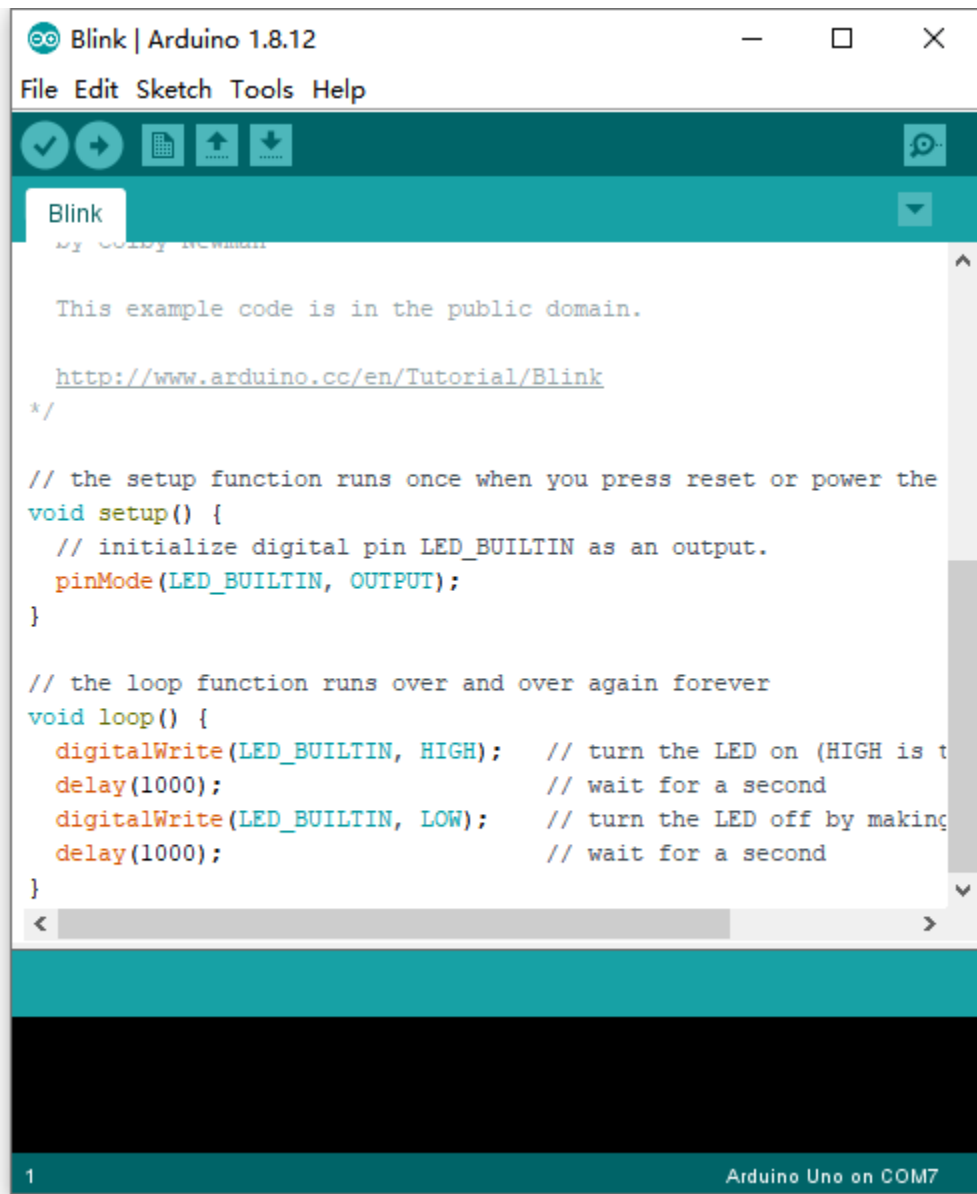
F- Used to send the serial data received from board to the serial monitor.

### 5.4 Start First Program

Open the file to select Example, and choose BLINK from BASIC, as shown below:




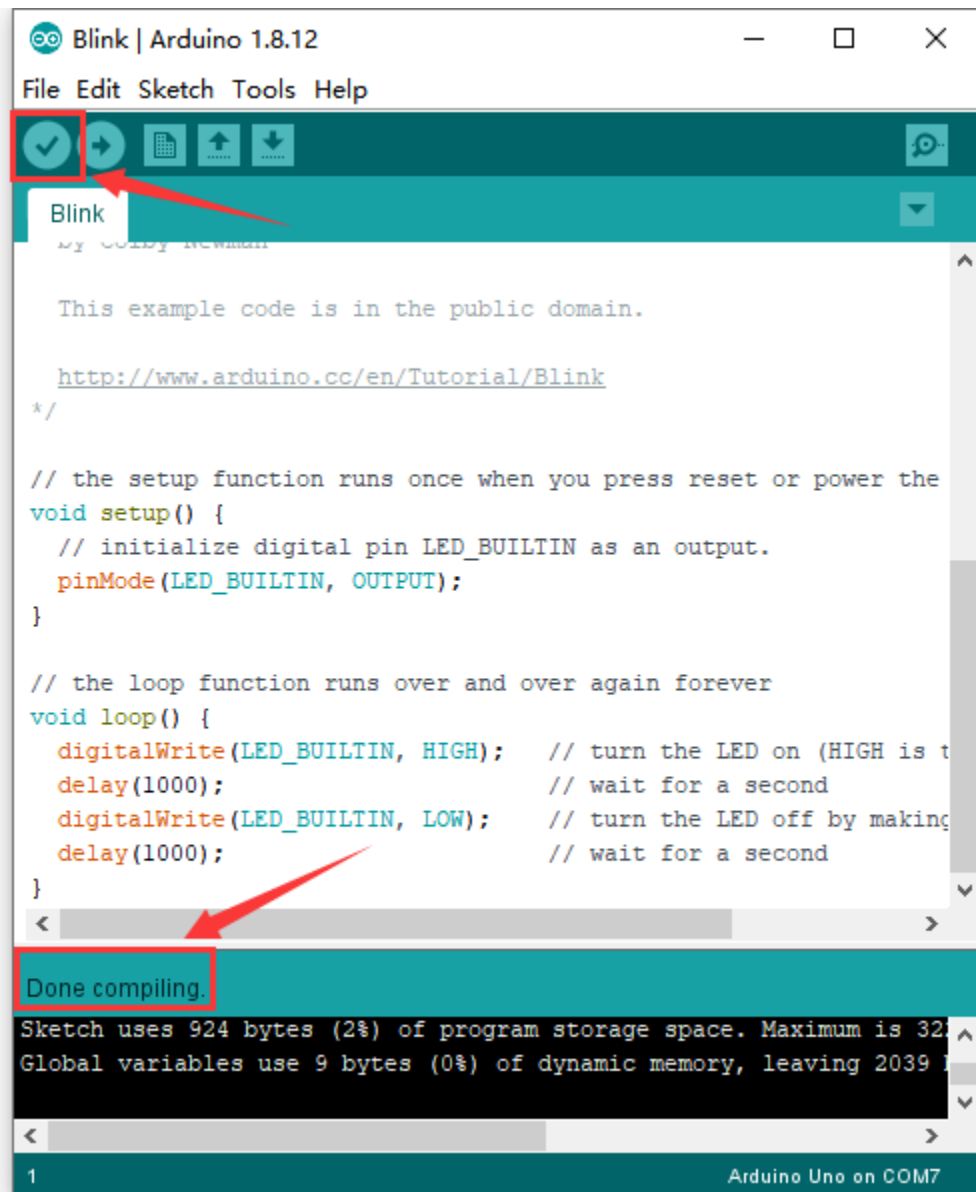





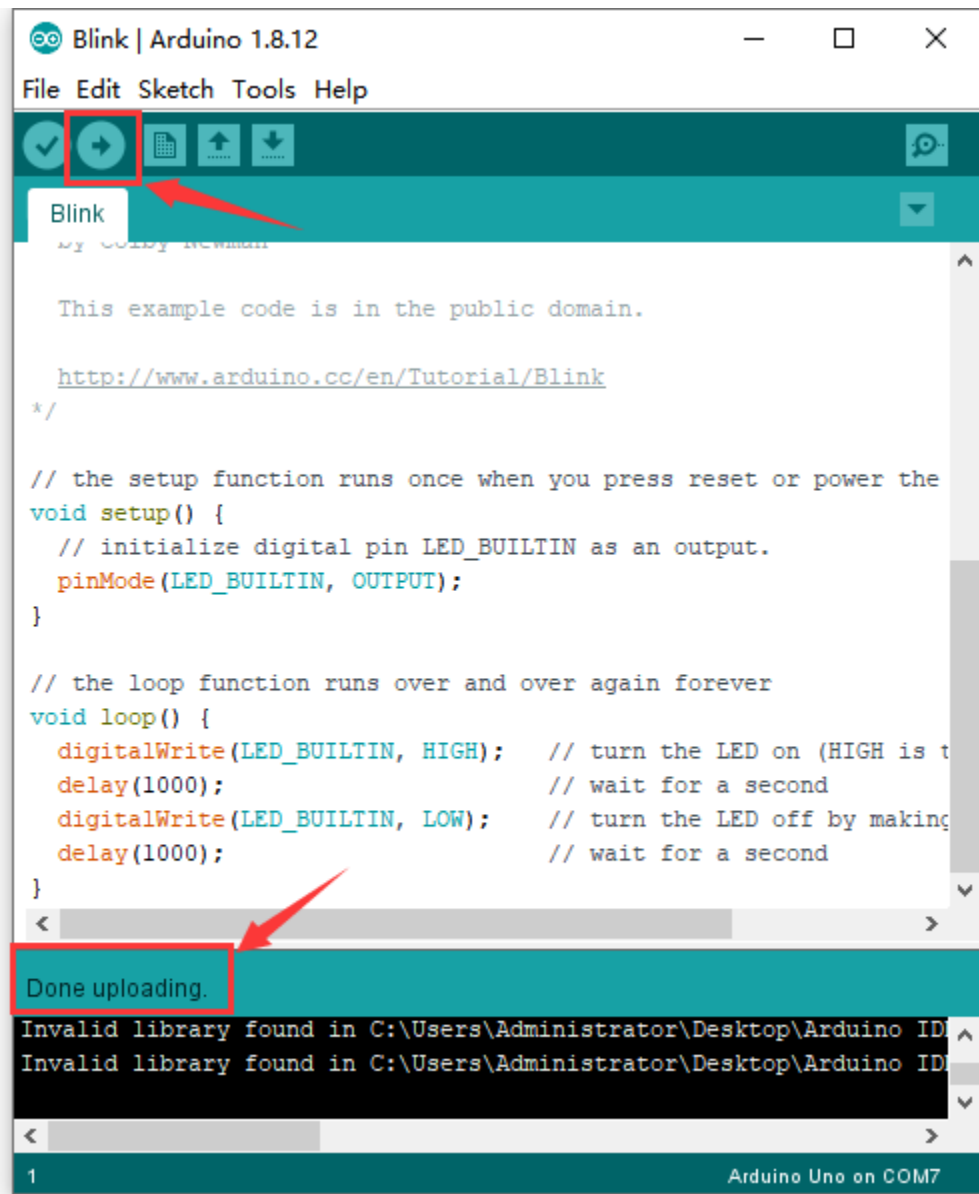
Set board and COM port, the corresponding board and COM port are shown on the lower right of IDE.



Click  to start compiling the program, check errors.



Click  to upload the program, upload successfully.




Upload the program successfully, the onboard LED lights on for 1s, lights off for 1s. Congratulation, you finish the first program.

## 6. Troubleshooting

1. If test code can't be uploaded, click

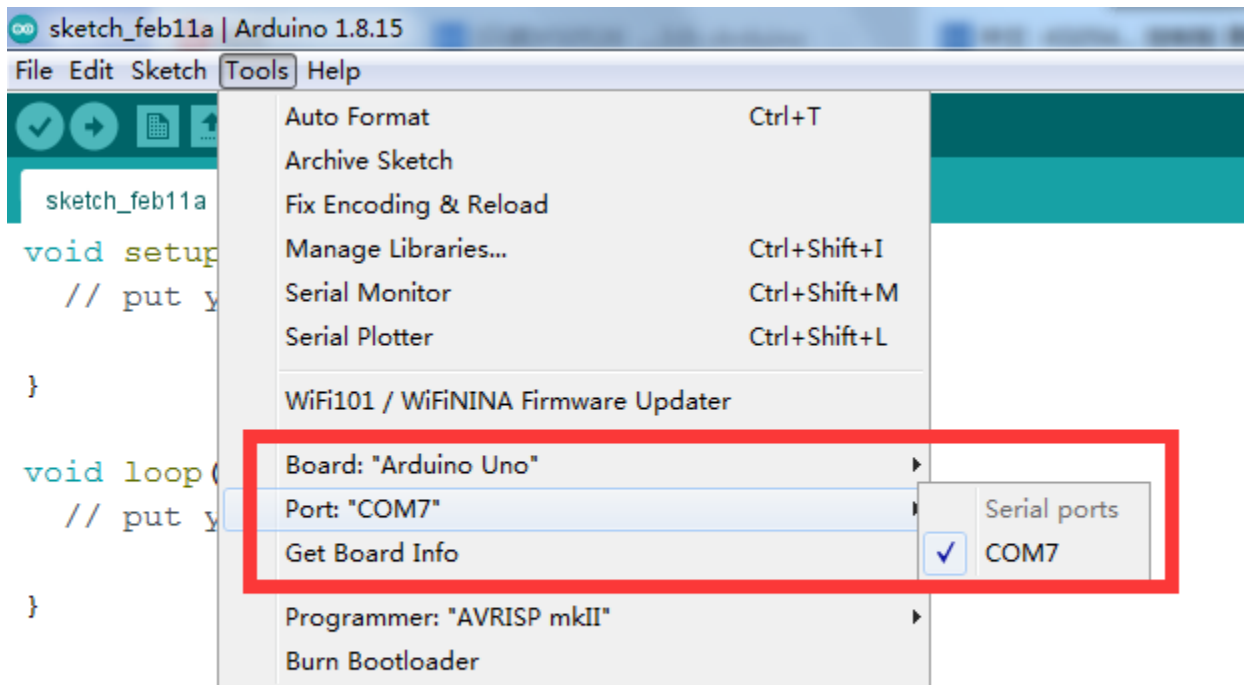


However, if clicking  (verifying test code) doesn't work, there is something wrong with the test code or Arduino IDE.



2.If test code can be verified but not be uploaded, check the correct development board or COM port you choose.

3.If you can't select the correct COM port, just check if the driver of the development board is installed successfully.(refer to 5.2)



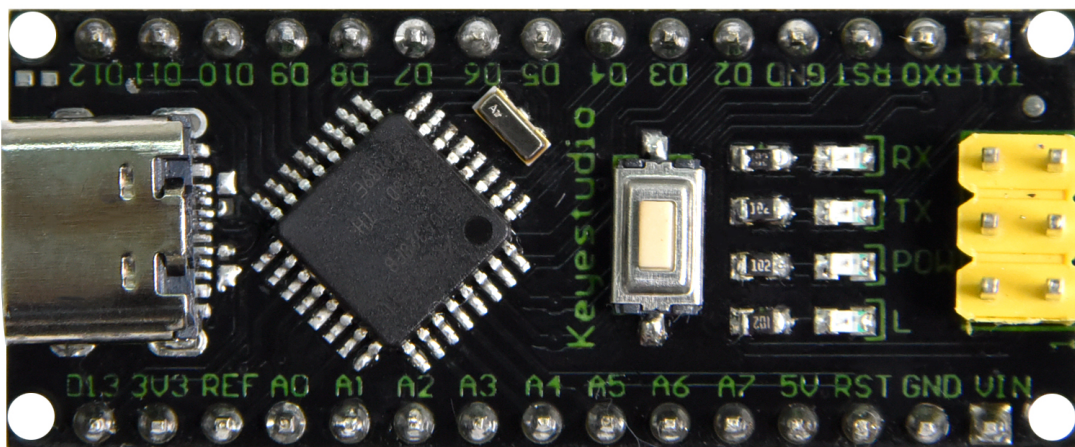
4.Since unstable power supply from USB, you can change different USB ports if the development board can't be recognized.

5.When uploading test code, TXRX of the development board is not allowed to be used(fail to upload test code).

If the Bluetooth module is used, upload test code then install the BT module



## KS0547 KEYESTUDIO NANO PLUS DEVELOPMENT BOARD



### 1. Description

The processor core of Keyestudio NANO PLUS is ATMEGA328P-AU. It is as same as the official Arduino Nano in addition to driver file and USB to serial chip (CH340G).

It also has 14 digital input / output interfaces (6 of which can be used as PWM output), 8 analog input interfaces, 1 16MHz crystal oscillator, 1 mini USB port, 1 ICSP interface and a reset button.

The ICSP interface is used to program the Atmega328P-Au. We can supply power with a type-CUS cable or the port VIN GND (DC 7-12V)

### 2. Specification

Microcontroller: ATMEGA328P-AU

Operating Voltage: 5V

Input Voltage (recommended): DC 7-12V

Digital I/O Pins: 14 (D0-D13)

PWM Digital I/O Pins6 (D3 D5 D6 D9 D10 D11)

Analog Input Pins: 8(A0-A7)

DC Current per I/O Pin: 40 mA

Flash Memory: 32 KB of which 2 KB used by bootloader

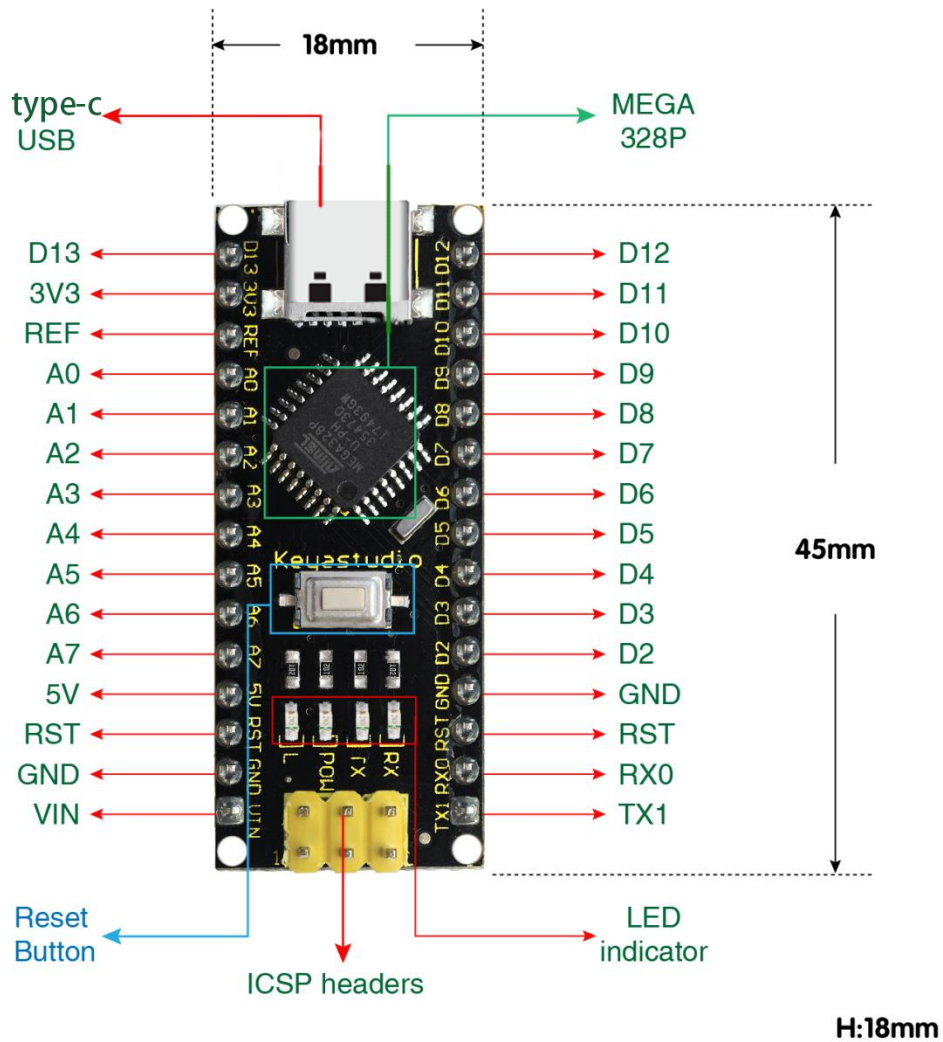
SRAM:2 KB

EEPROM: 1 KB

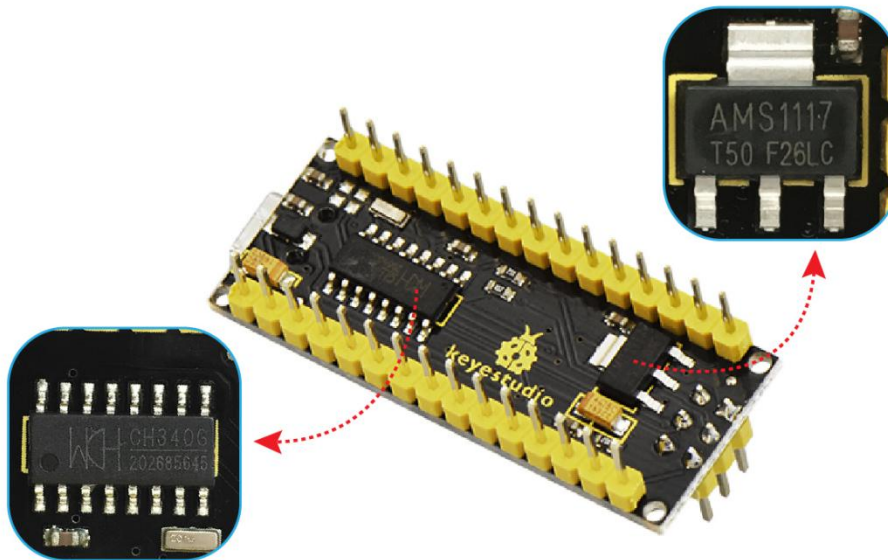
Clock Speed:16 MHz

LED\_BUILTIN:D13

### Pins







1	<b>ICSP Header</b>	<b>ICSP (In-Circuit Serial Programming) Header</b> ICSP is the AVR, an micro-program header consisting of MOSI, MISO, SCK, RESET, VCC, and GND. It is often called the SPI (serial peripheral interface) and can be considered an “extension” of output. In fact, slave the output devices under the SPI bus host. When connecting to PC, program the firmware to ATMEGA328P-AU.
2	<b>LED indicator RX</b>	Onboard you can find the label: RX(receive ) When control board communicates via serial port, receive the message, RX led flashes.
3	<b>LED indicator TX</b>	Onboard you can find the label: TX (transmit) When control board communicates via serial port, send the message, TX led flashes.
4	<b>LED indicator POW</b>	Power up the control board, LED on, otherwise LED off.
5	<b>LED indicator L</b>	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.
6	<b>RX0D0 TX1D1 D2-D13</b>	It has 14 digital input/output pins D0-D13 (of which 6 can be used as PWM outputs). These pins can be configured as digital input pin to read the logic value (0 or 1). Or used as digital output pin to drive different modules like LED, relay, etc.
7	<b>RST</b>	Reset pin: connect external button. The function is the same as RESET button.
8	<b>MEGA 328P</b>	Each board has its own microcontroller. You can regard it as the brain of your board. Microcontrollers are usually from ATMEL. Before you load a new program on the Arduino IDE, you must know what IC is on your board. This information can be checked at the top surface of IC. The board's microcontroller is ATMEGA328P-AU. More info. see the <a href="#">datasheet</a>
9	<b>MINI USB</b>	The board can be powered via Mini-B USB connection. Also upload the program to the board via USB port.
10	<b>3V3 pin</b>	Provides 3.3V voltage output
11	<b>REF</b>	Reference external voltage (0-5 volts) for the analog input pins. Used with <a href="#">analogReference()</a> .
12	<b>A0-A7</b>	The Nano has 8 Analog Pins, labeled A0 through A7.
13	<b>5V pin</b>	Provides 5V voltage output
14	<b>GND</b>	Ground pin
15	<b>VIN</b>	Input an external voltage DC7-12V to power the board.
16	<b>Reset Button</b>	Used to reset the control board
17	<b>CH340G</b>	USB-to-serial port chip, converting the USB signal into Serial port signal.
18	<b>AMS1117</b>	Convert the external voltage input DC7-12V into DC5V, then transfer it to the processor and other elements.

### 3.Specialized Functions of Some Pins

- **Serial communication:** 0 (RX) and 1 (TX). Used to receive (RX) and transmit (TX) TTL serial data.
- **PWM (Pulse-Width Modulation):** D3, D5, D6, D9, D10, D11

- **External Interrupts:** D2 (interrupt 0) and D3 (interrupt 1). 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.
- **SPI communication:** D10 (SS), D11 (MOSI), D12 (MISO), D13 (SCK).
- **IIC communication:** A4 (SDA); A5(SCL)

## 4.Windows System

### 4.1 Download the Arduino IDE

When getting this control board, we need to install Arduino IDE

Enter the website <https://www.arduino.cc/> and click



You can select the latest version—1.8.13. Alternatively, the previous release is your another choice.

In this project, we use 1.8.12 version.

## Previous Releases

Download the previous version of the current release, the classic 1.0.x, or old beta releases.

### DOWNLOAD OPTIONS

**Previous Release (1.8.12)**

**Arduino 1.0.x**

**Arduino 1.5.x beta**

**Arduino 1.9.x beta**

Click **Previous Release (1.8.12)** to enter the new page. As shown below;

The **Windows installer** needs installing manually. Yet , the **Windows zip file for non admin install** a zip file of Arduino 1.8.12 version, can be directly downloaded and installed.

# Previous IDE Releases

## ARDUINO 1.8.12

Arduino IDE that can be used with any Arduino board, including the Arduino Yún and Arduino DUE. Refer to the [Getting Started](#) page for Installation instructions.

[See the release notes.](#)

**Windows** Installer

**Windows** ZIP file for non admin install

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

**Linux** 32 bits

**Linux** 64 bits

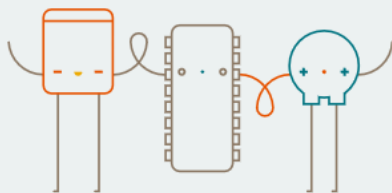
**Linux** ARM 32

**Linux** ARM 64

**Source**

## Contribute to the Arduino Software

Consider supporting the Arduino Software by contributing to its development. (US tax payers, please note this contribution is not tax deductible). [Learn more on how your contribution will be used.](#)



SINCE MARCH 2015, THE ARDUINO IDE HAS BEEN DOWNLOADED **45,290,706** TIMES. (IMPRESSIVE!) NO LONGER JUST FOR ARDUINO AND GENUINO BOARDS, HUNDREDS OF COMPANIES AROUND THE WORLD ARE USING THE IDE TO PROGRAM THEIR DEVICES, INCLUDING COMPATIBLES, CLONES, AND EVEN COUNTERFEITS. HELP ACCELERATE ITS DEVELOPMENT WITH A SMALL CONTRIBUTION! REMEMBER: OPEN SOURCE IS LOVE!

\$3

\$5

\$10

\$25

\$50

OTHER

JUST DOWNLOAD

CONTRIBUTE & DOWNLOAD

JUST DOWNLOAD

CONTRIBUTE & DOWNLOAD

Click icon

to download Arduino IDE.

### 4.2 Download Driver of CH340

<https://fs.keyestudio.com/CH340-WIN>

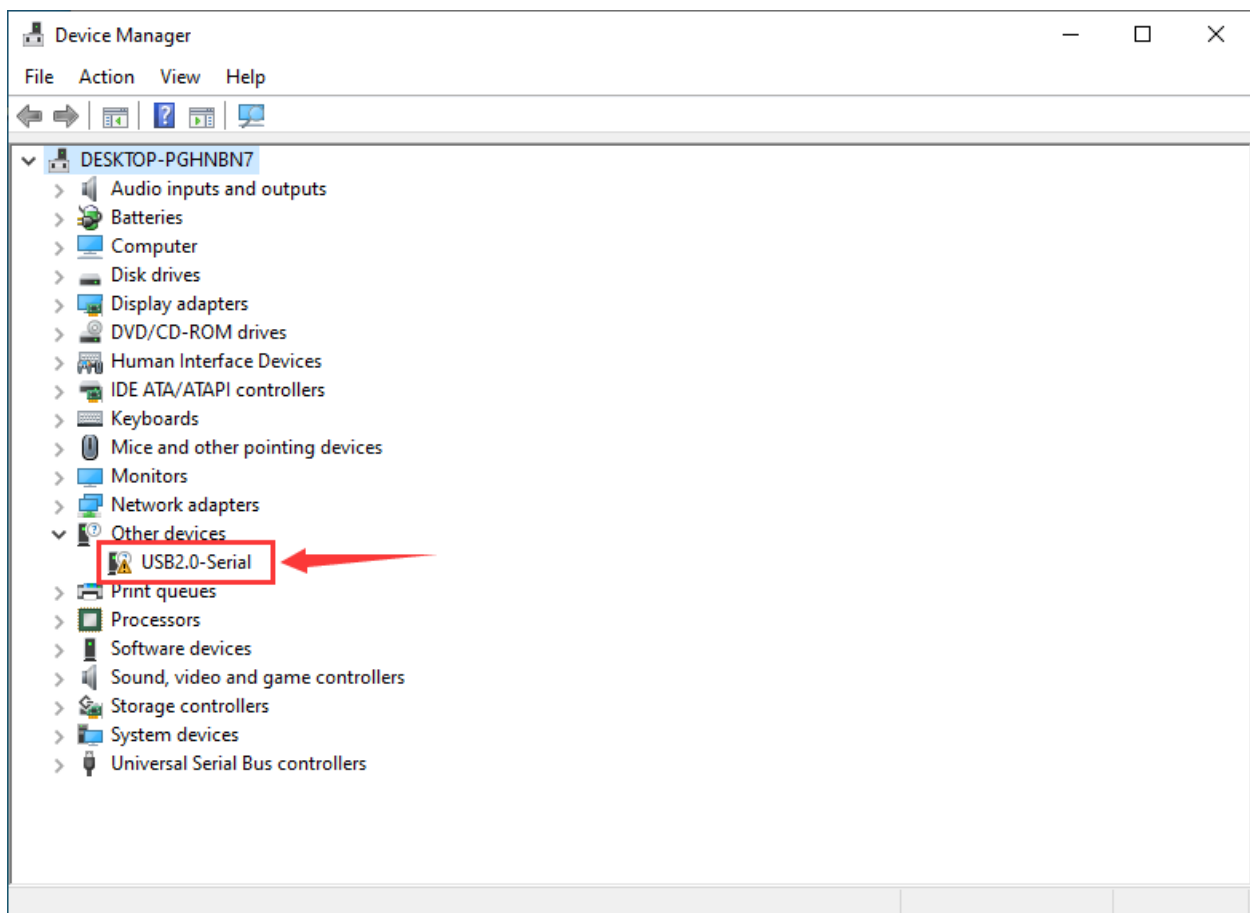
### 4.3 Install the Driver


For this part, we need to install the driver of Arduino IDE

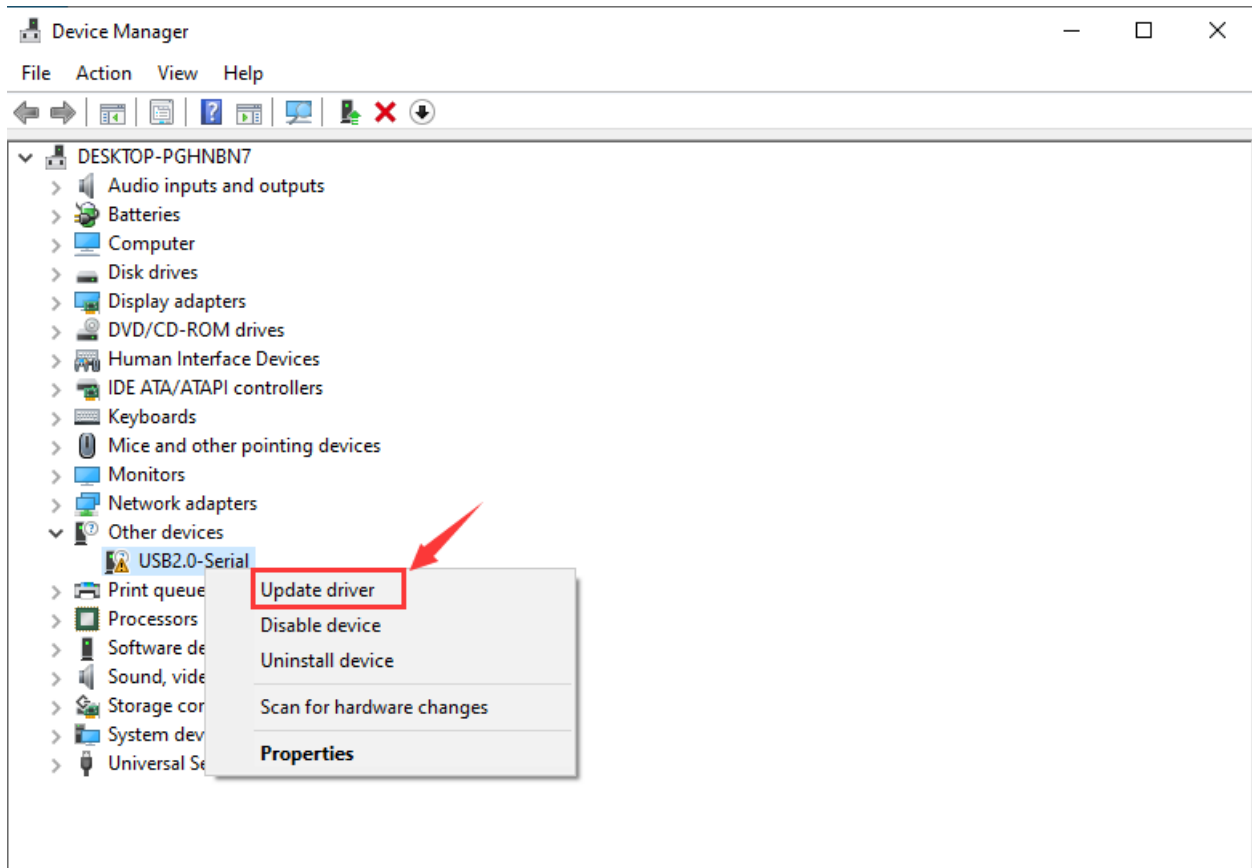
First, let's attach USB cable to computer. The driver can be installed automatically if the PC system is Windows 10, however, you need to install the driver manually if the PC system is other version.

The USB to serial chip of control board is CH340G, therefore, we will install its driver(usb\_ch341\_3.1.2009.06).

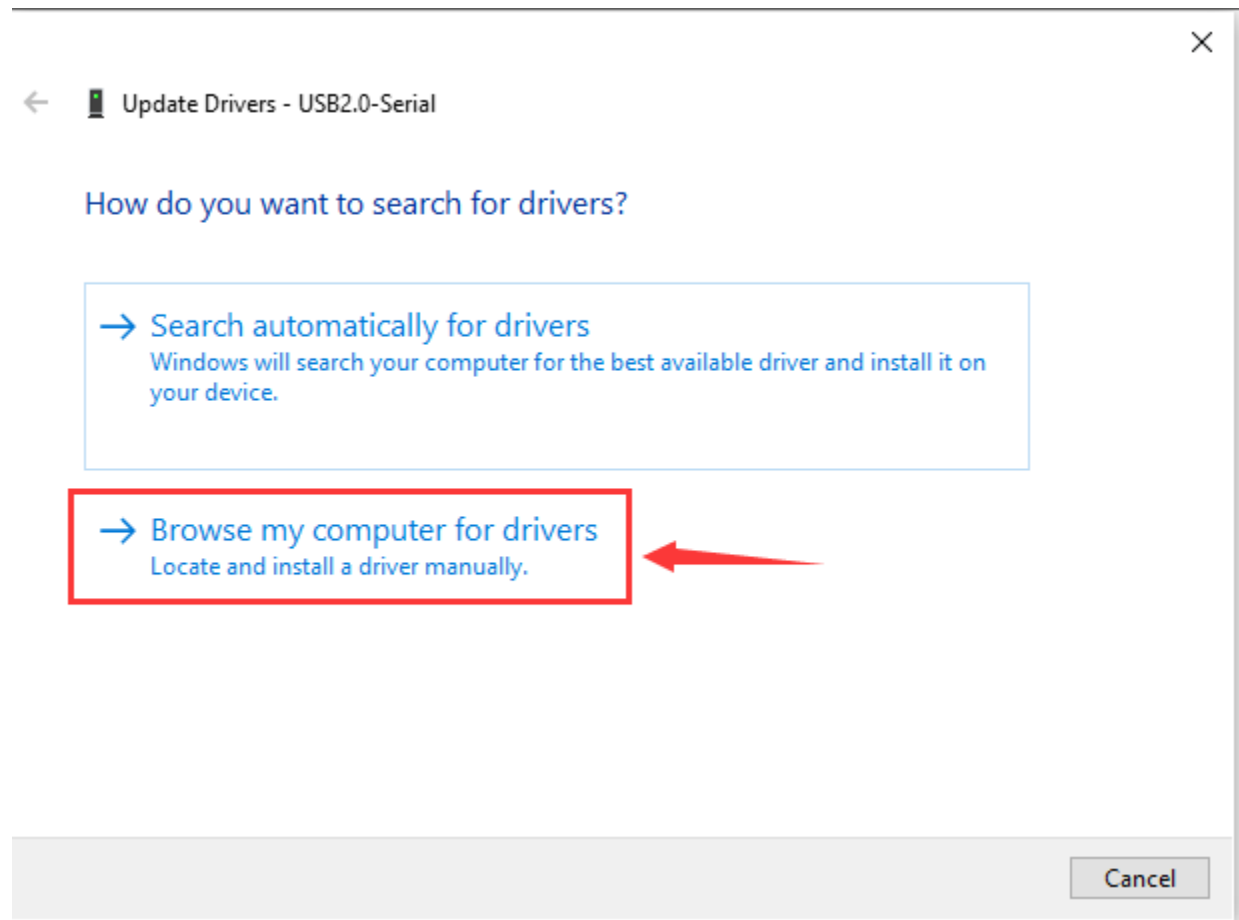
Click Computer— Properties— Device Manager, as shown below:



Click  USB Serial and "Update Driver Program"

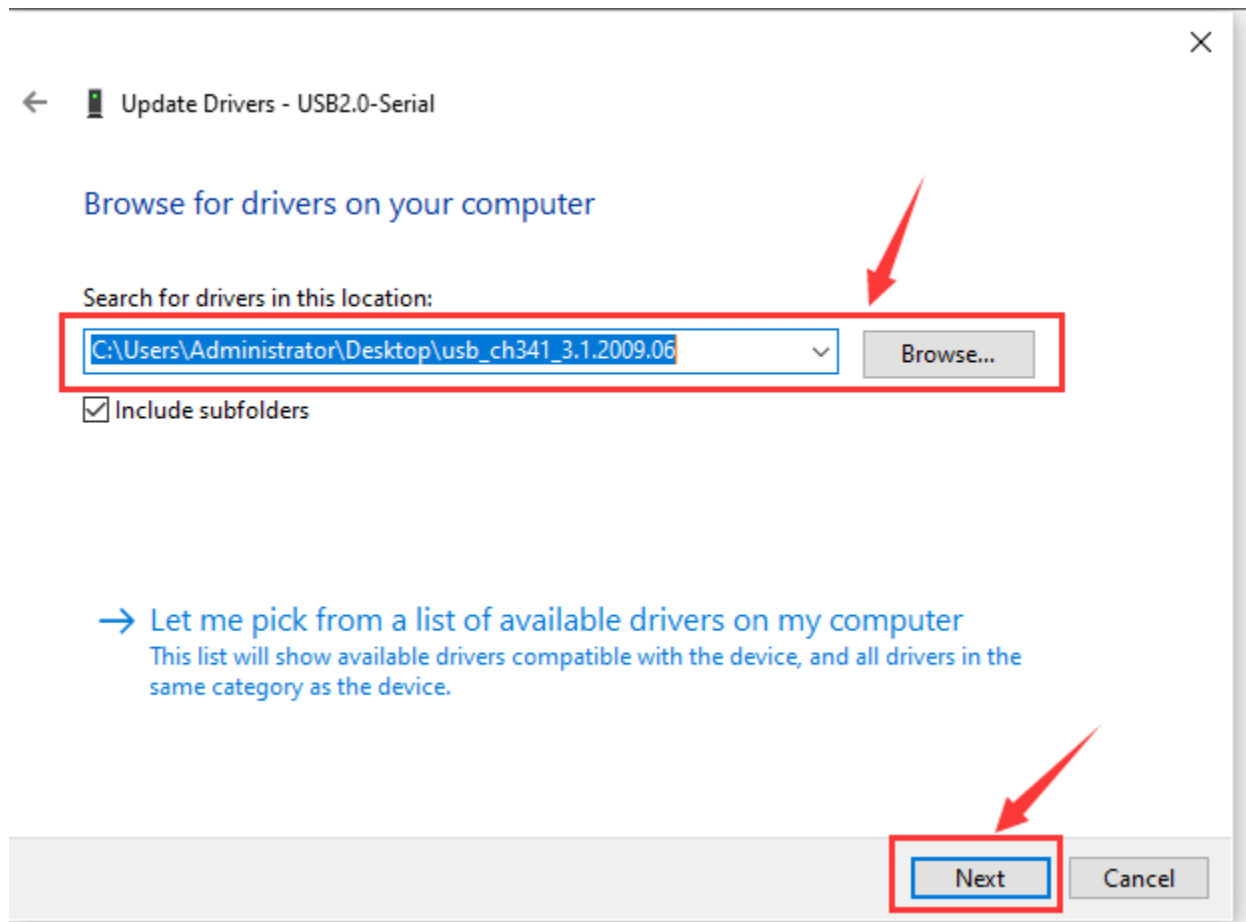


Jump into the following page and select “Browse my computer for driver software”.



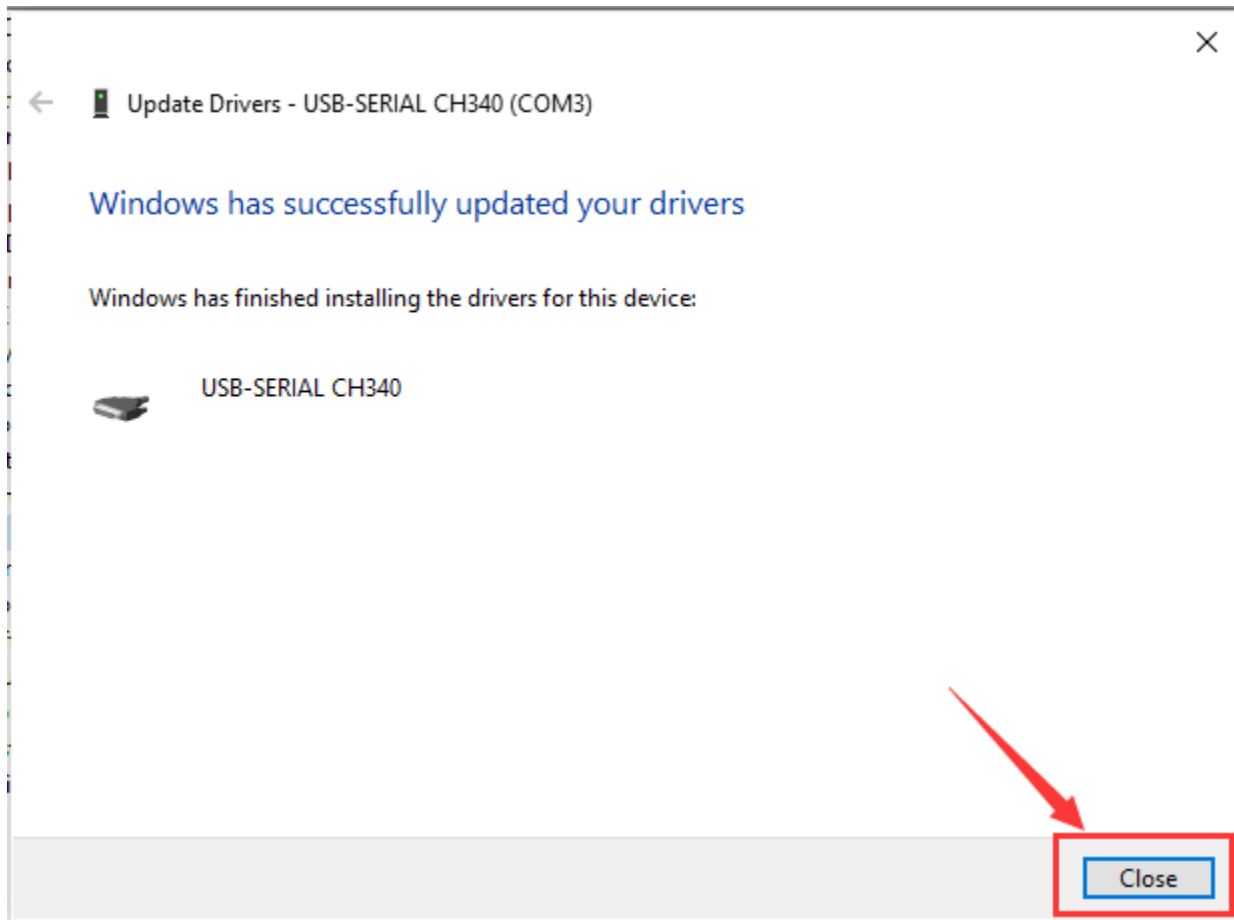
Search the **usb\_ch341\_3.1.2009.06** folder



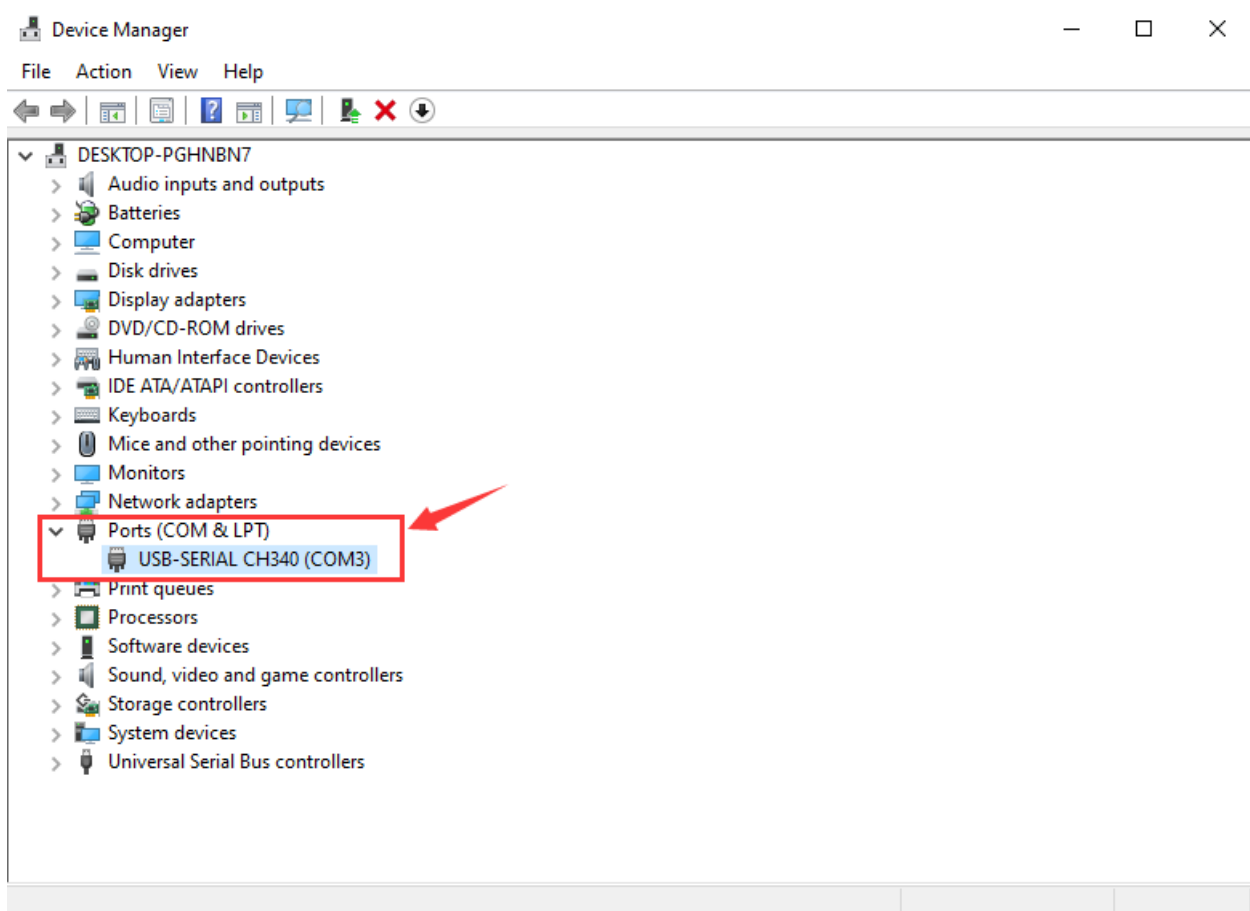


After the driver is installed, you need to click **Close**.

[http://wiki.keyestudio.com/index.php/File:Driver\\_6.png](http://wiki.keyestudio.com/index.php/File:Driver_6.png)




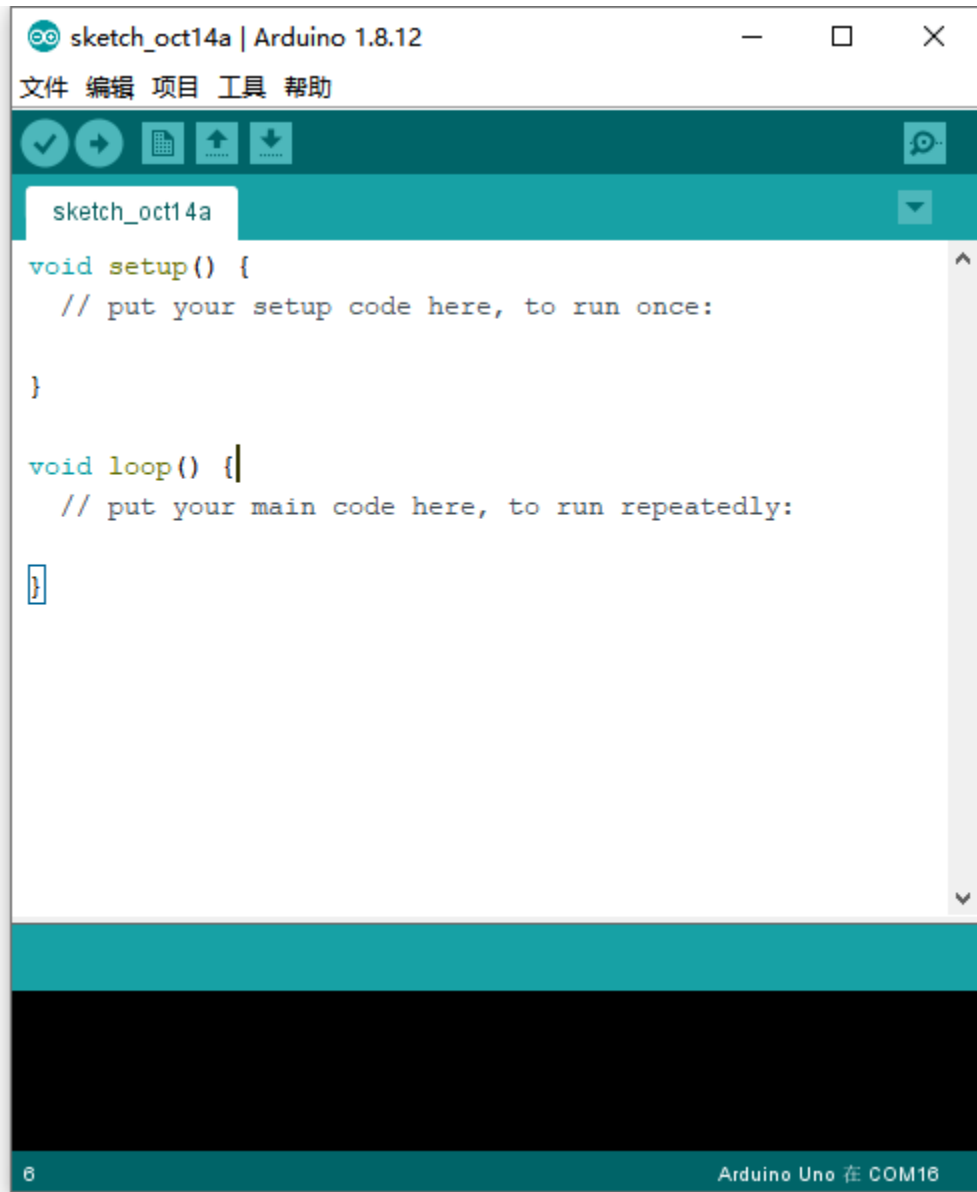
Click Computer— Properties— Device Manager, as shown below.



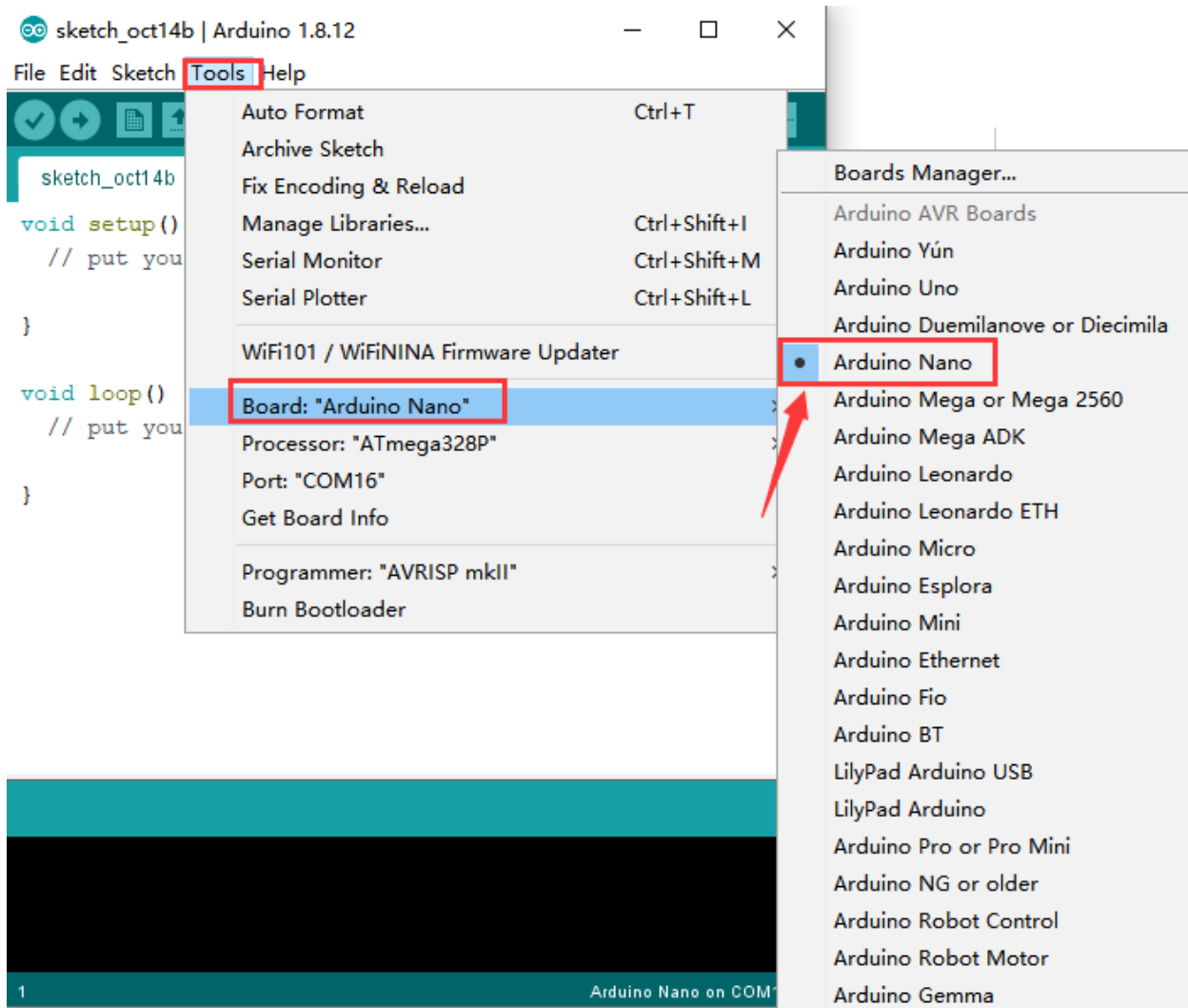
#### 4.4 Arduino IDE Setting

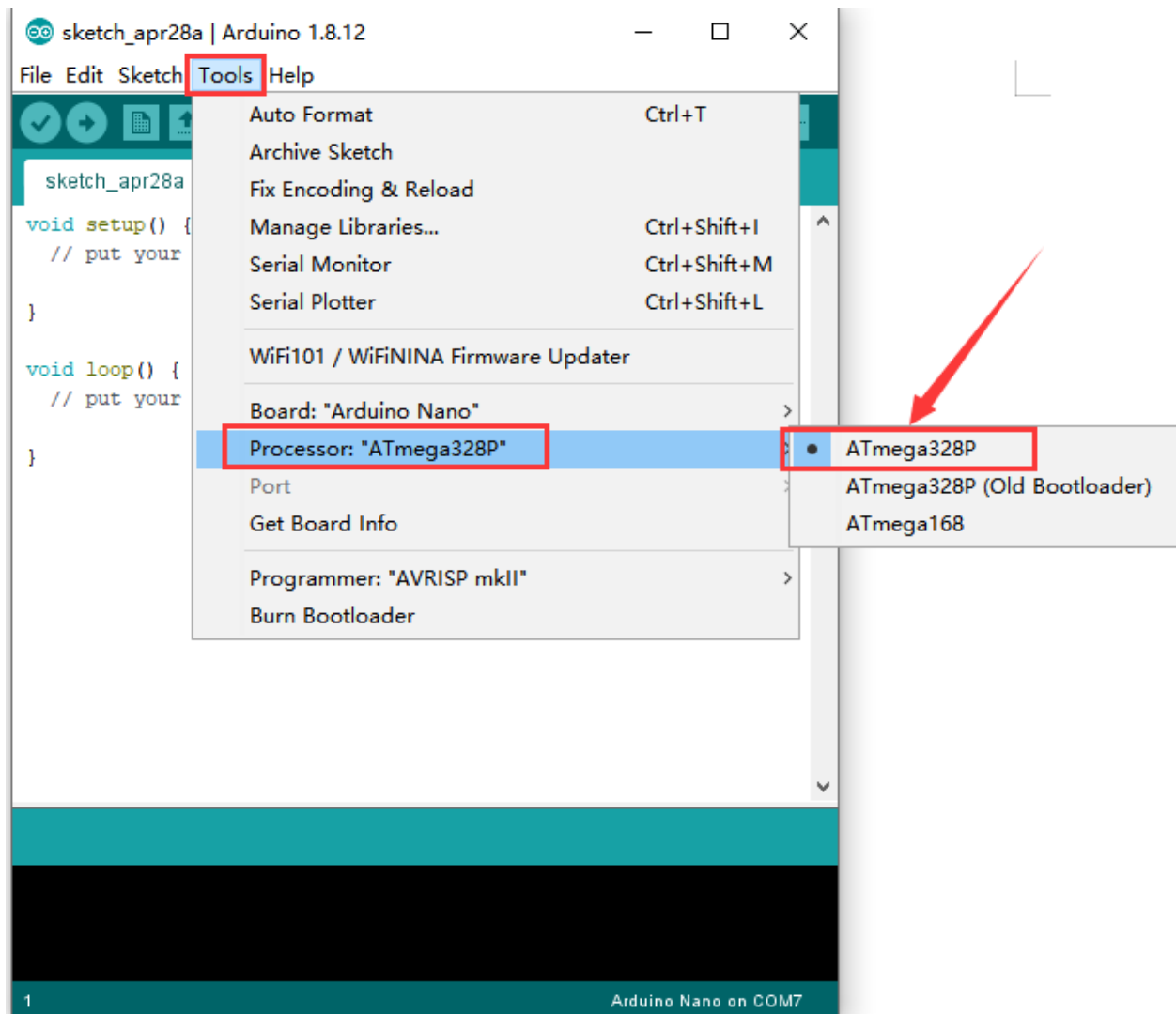


Click  icon to pen Arduino IDE.

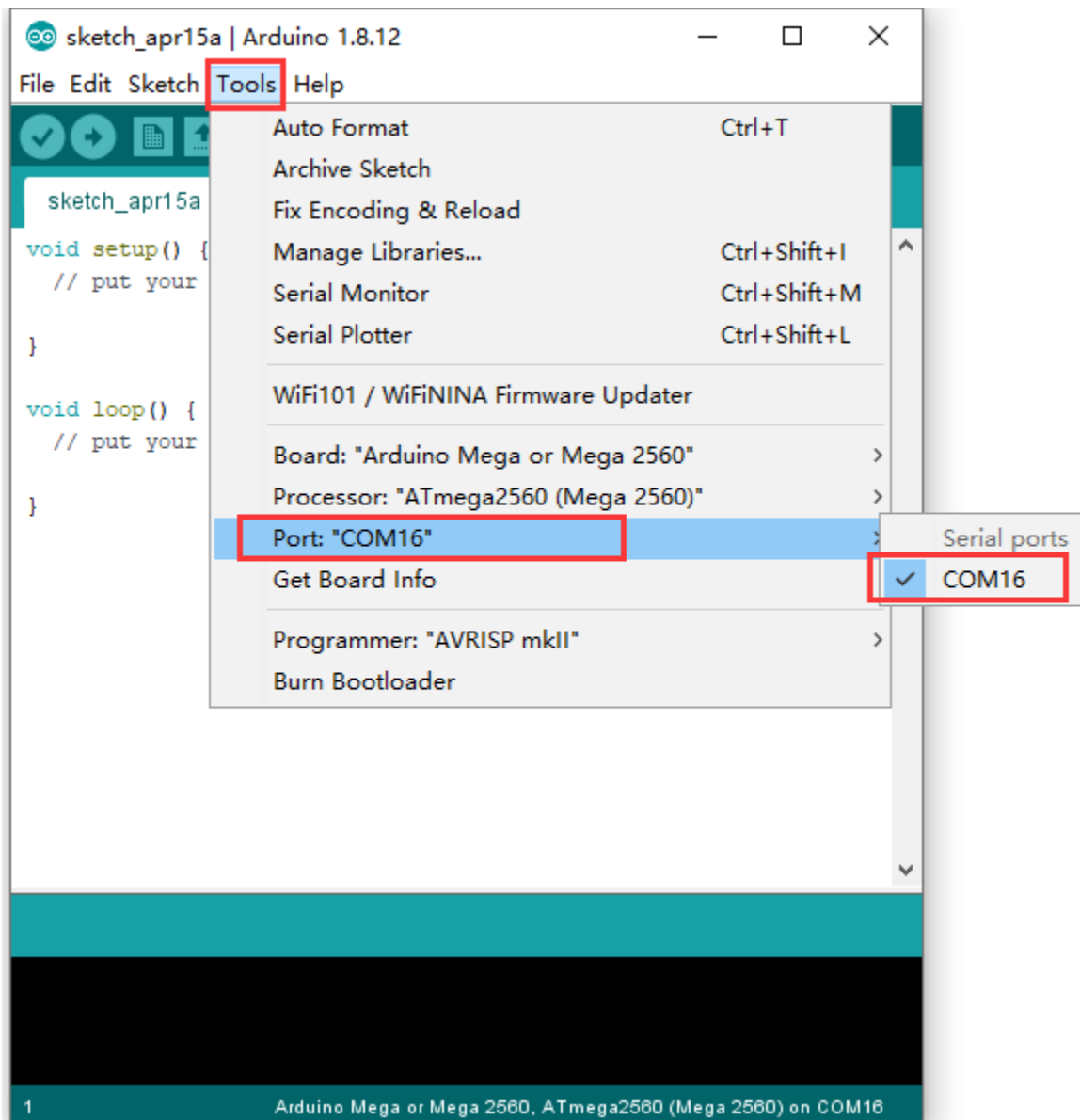


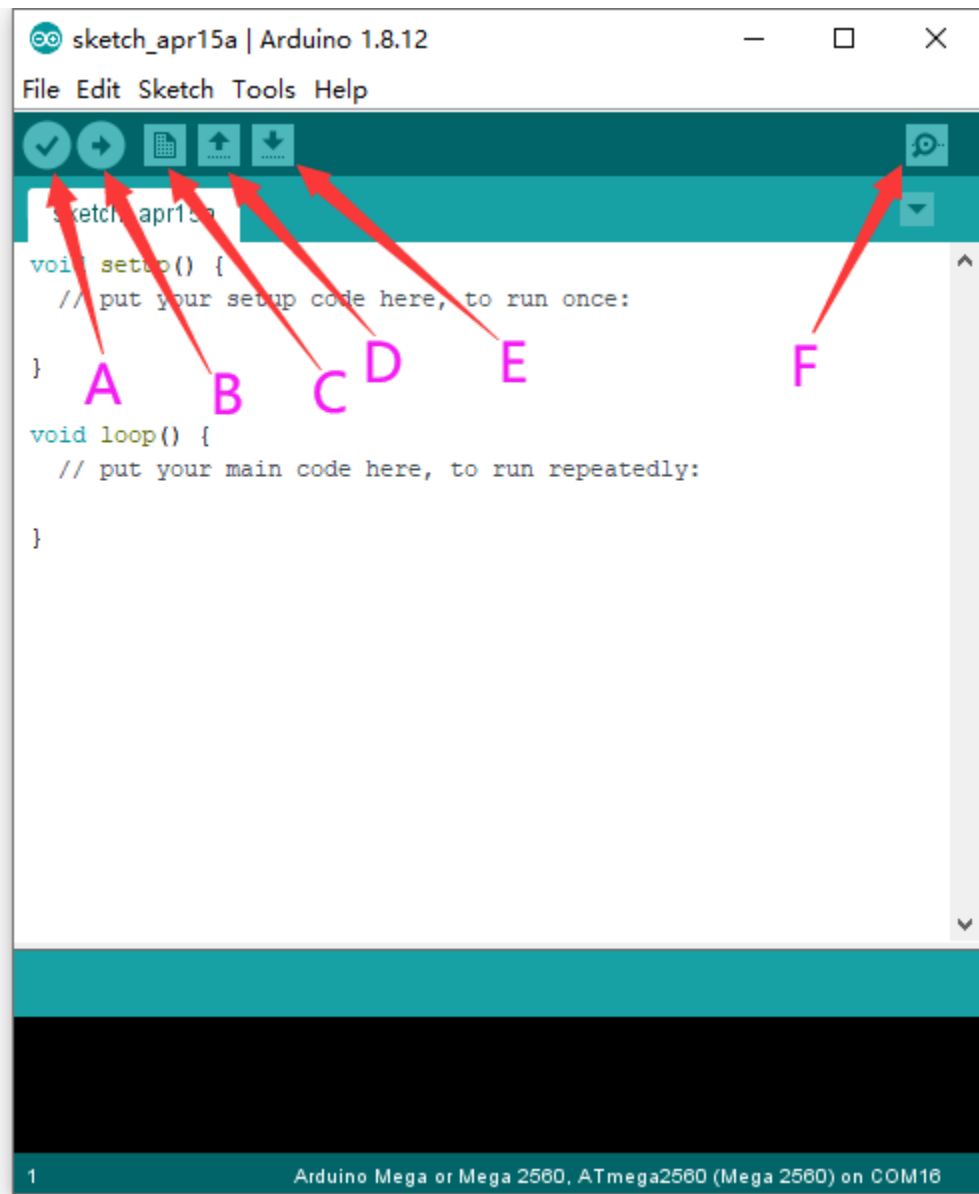
When downloading the sketch to the board, you must select the correct name of Arduino board that matches the board connected to your computer. As shown below;





Then select the correct COM port (you can see the corresponding COM port after the driver is successfully installed).





A- Used to verify whether there is any compiling mistakes or not.

B- Used to upload the sketch to your Arduino board.

C- Used to create shortcut window of a new sketch.

D- Used to directly open an example sketch.

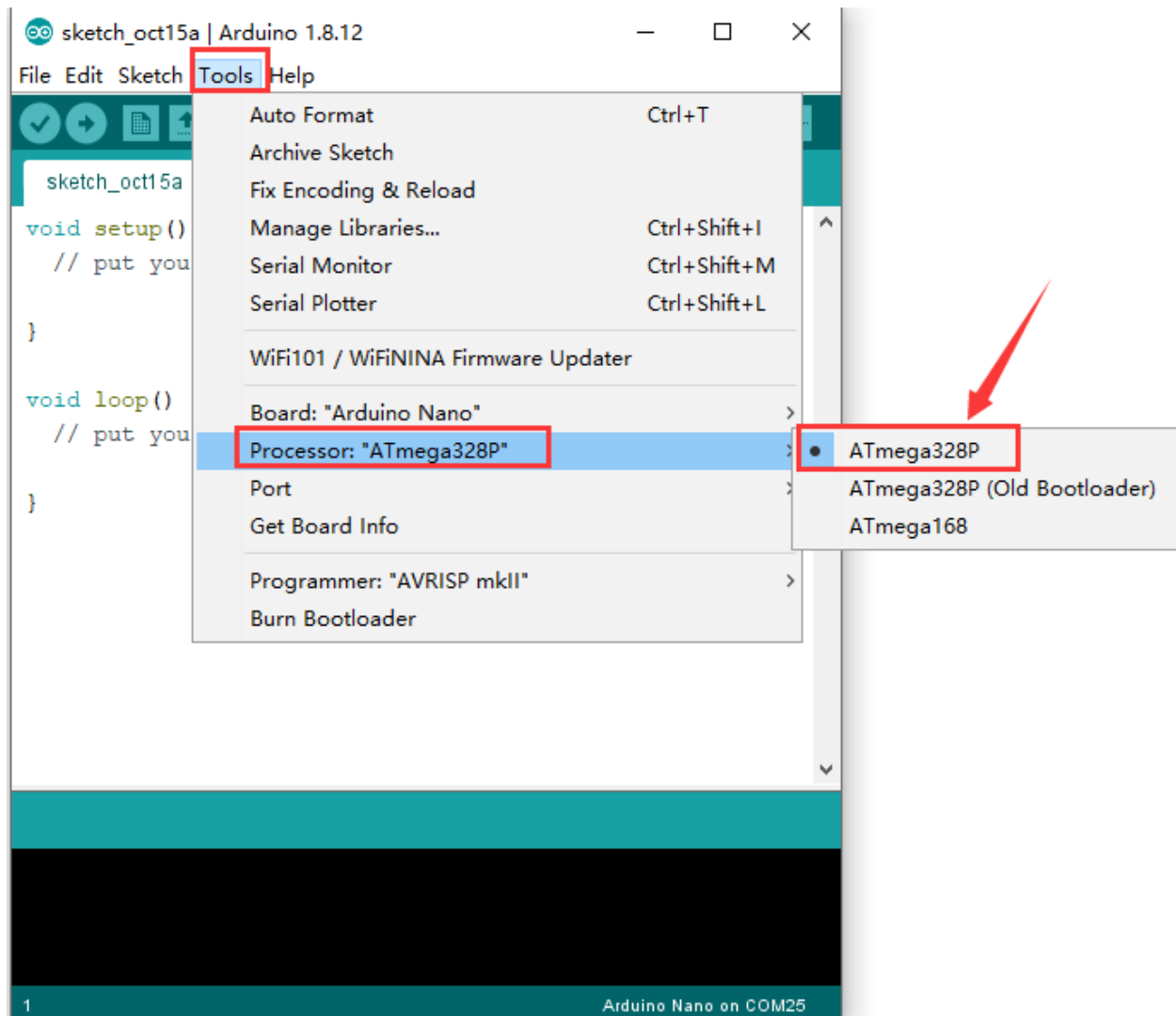
E- Used to save the sketch.

F- Used to send the serial data received from board to the serial monitor.

### 4.5 Setting IDE for New Bootloader

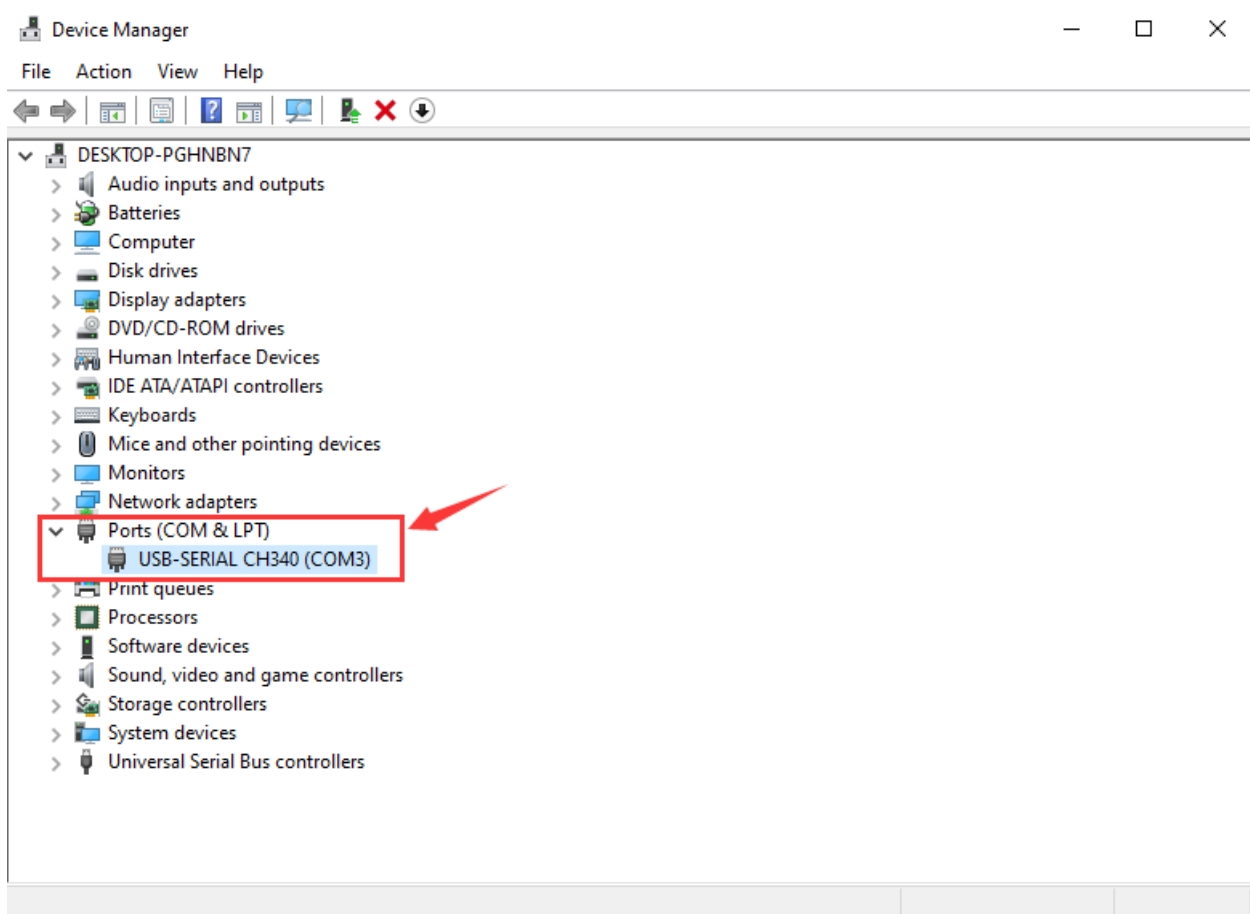
Arduino Nano board could burn new and old bootloader. New bootloader is only compatible with 1.8.9 IDE and above. Yet the old one is compatible with all versions.

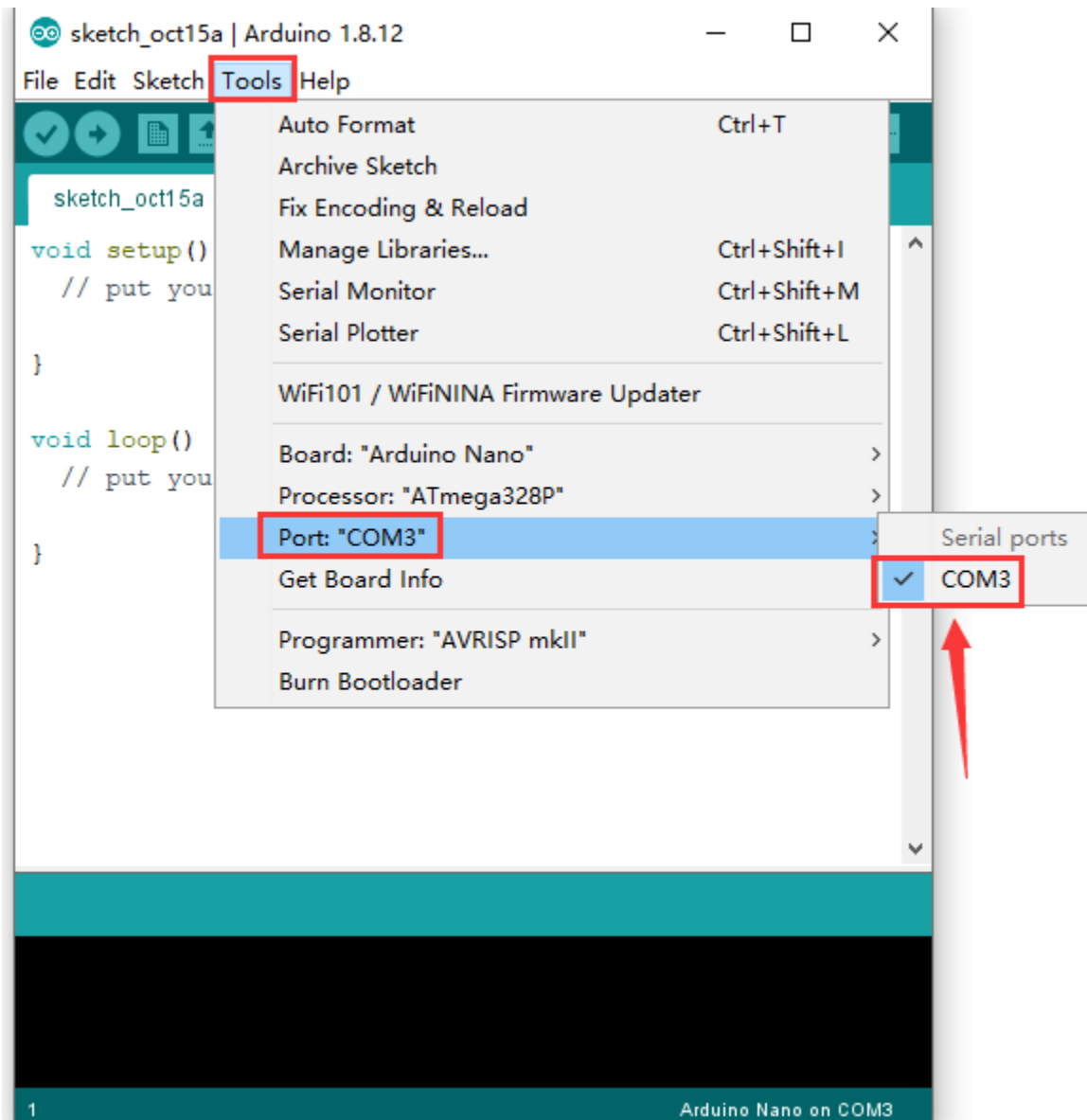




#### 4.6 Select Your Serial Port

Select the serial device of the Arduino board from the **Tools | Serial Port** menu. **Note:** to avoid errors, the COM Port should keep the same as the Ports shown on Device Manager.





#### 4.7 Hello World!

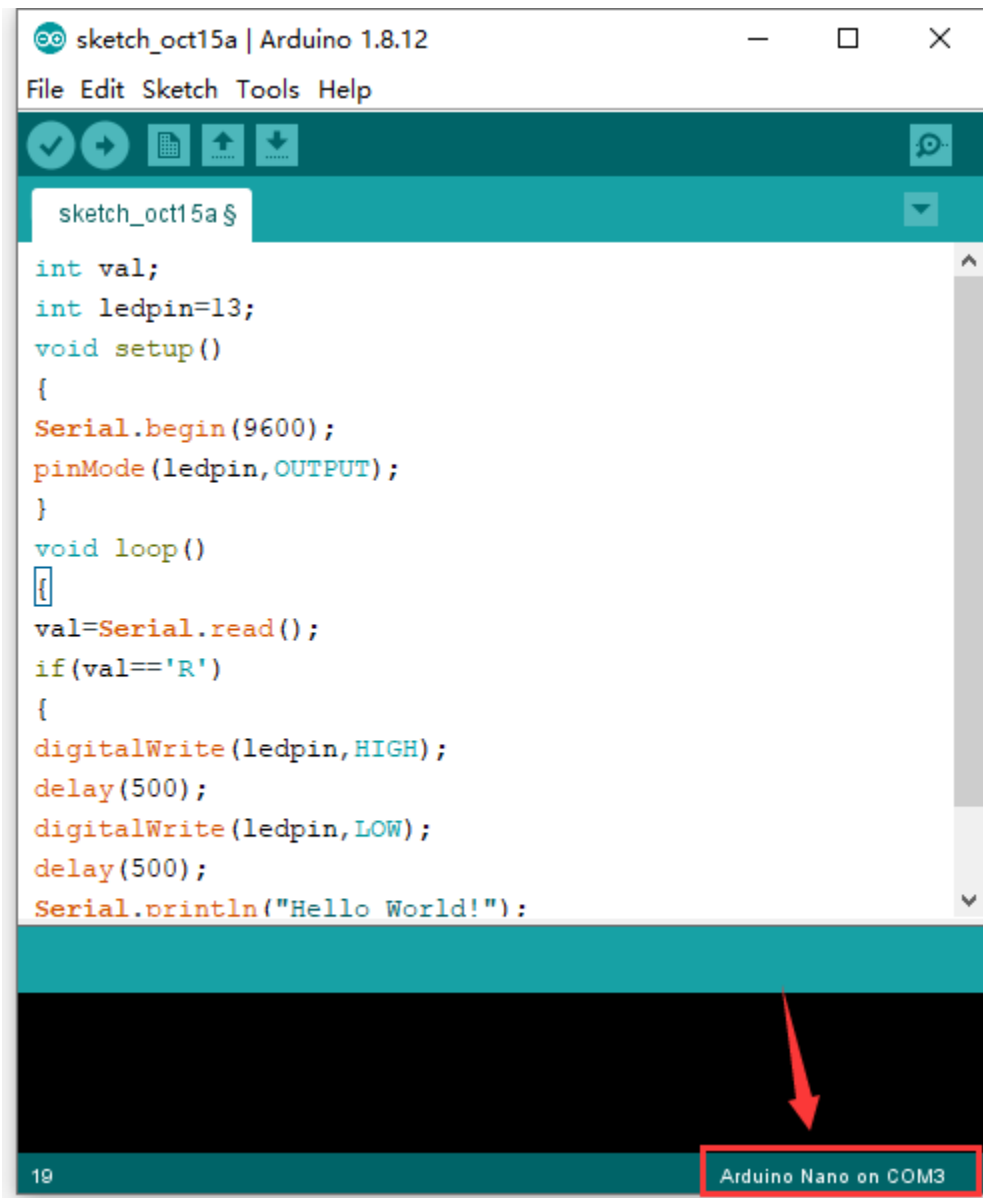
Copy the following code in the Arduino IDE.

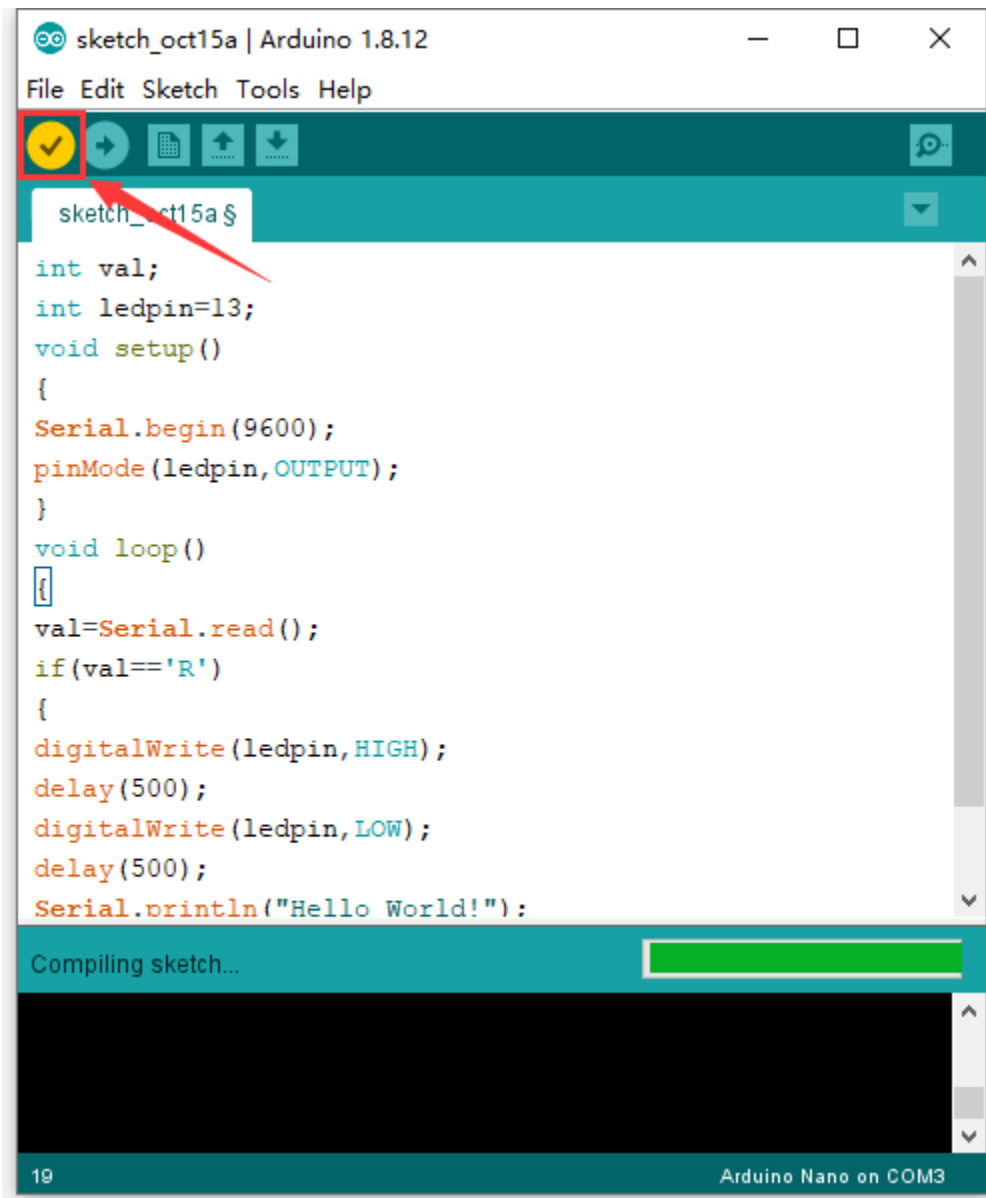
```
int val;  
  
int ledpin=13;  
  
void setup()  
{  
  Serial.begin(9600);  
  pinMode(ledpin,OUTPUT);  
}
```

(continues on next page)

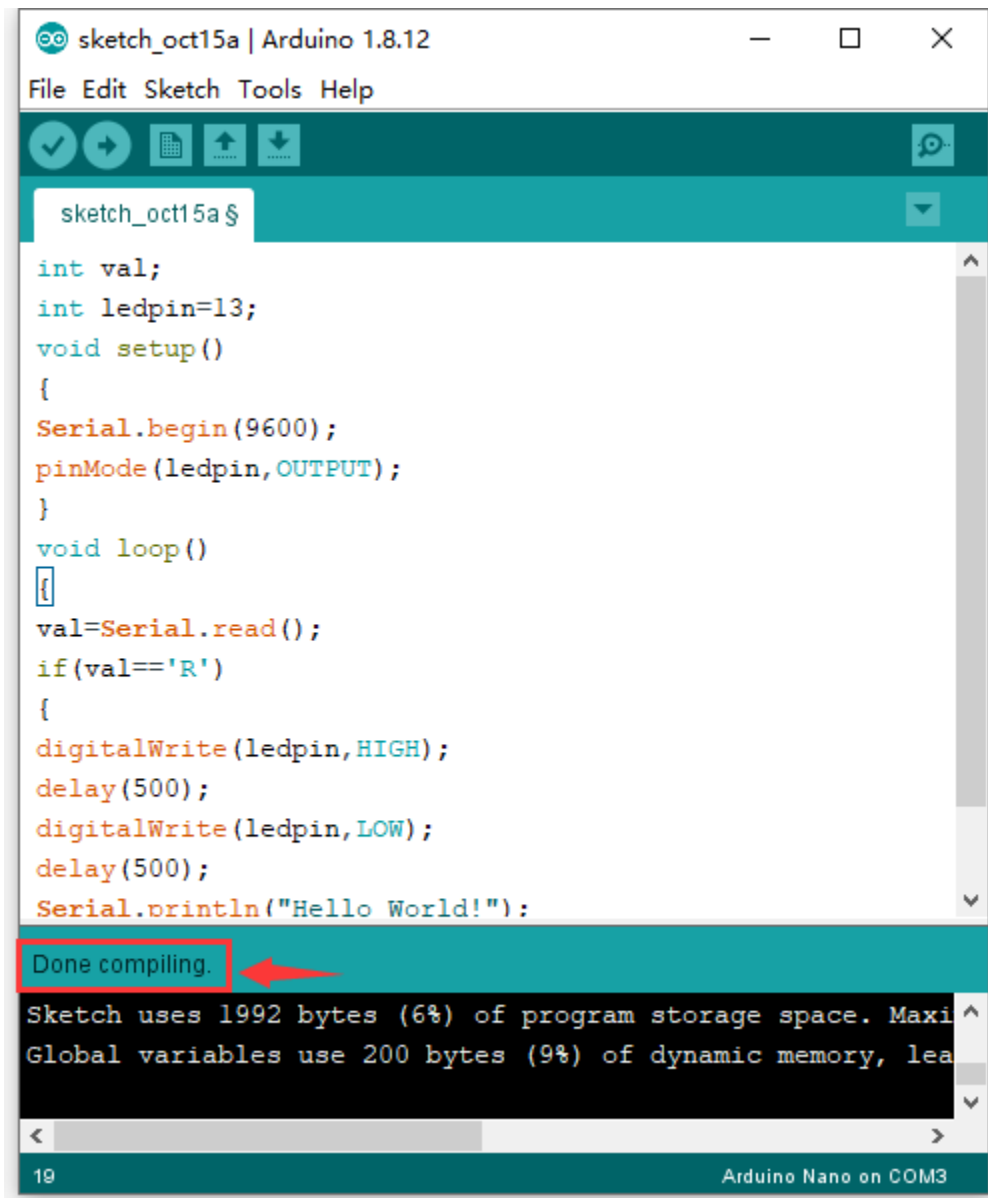
(continued from previous page)

```
}  
  
void loop()  
{  
  val=Serial.read();  
  if(val=='R')  
  {  
    digitalWrite(ledpin,HIGH);  
    delay(500);  
    digitalWrite(ledpin,LOW);  
    delay(500);  
    Serial.println("Hello World!");  
  }  
}
```

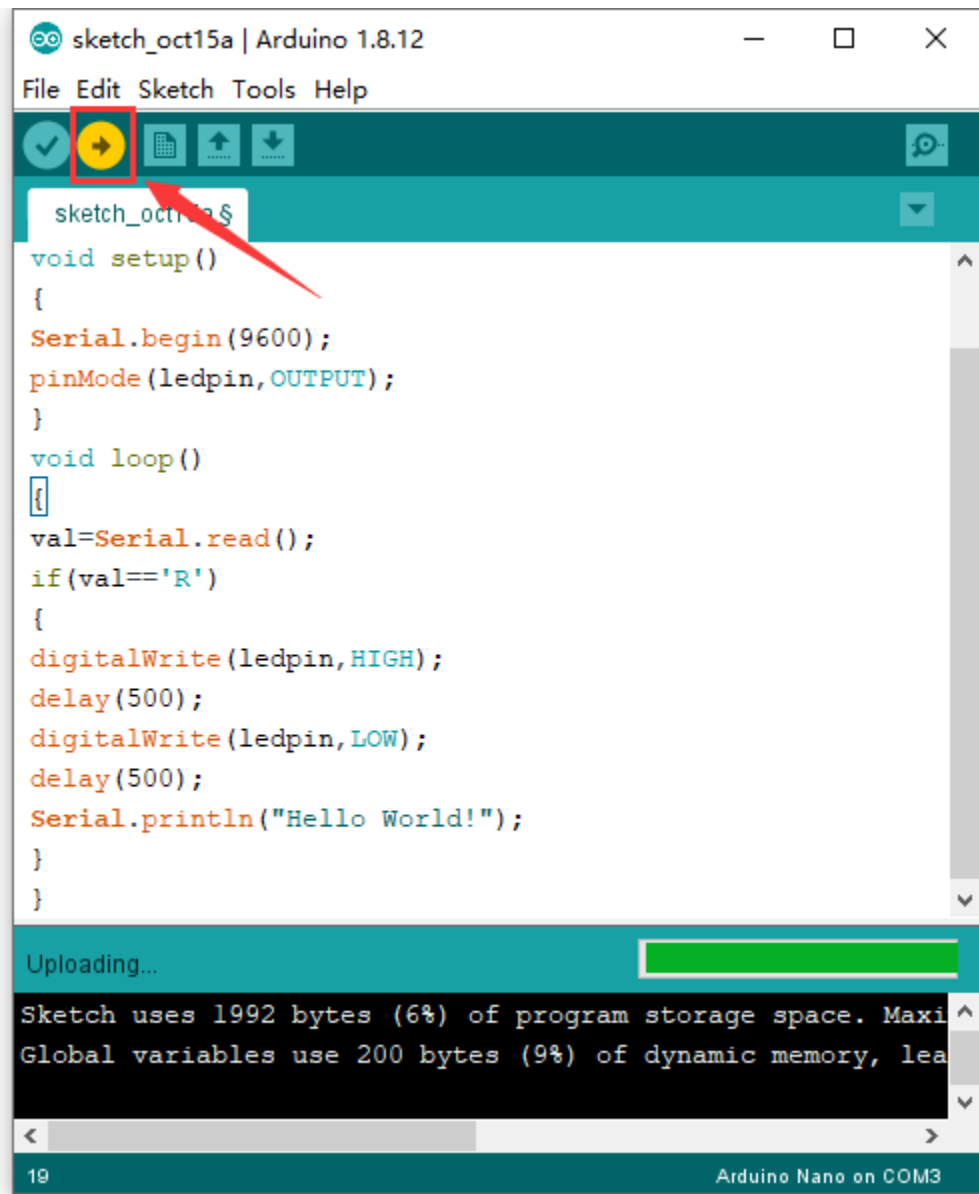




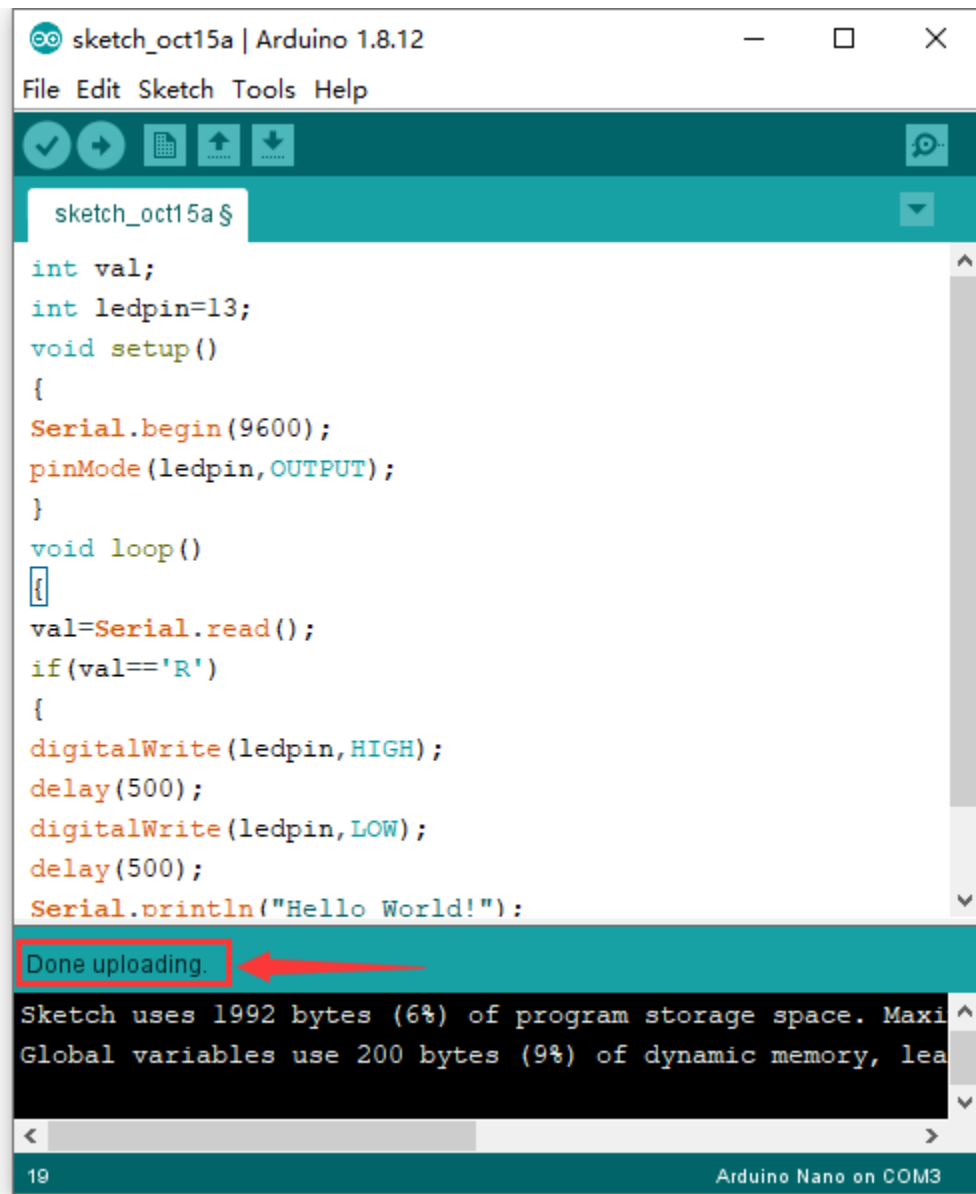
Then click verify button to check the errors. If compiling successfully, the message “Done compiling.” will appear in the status bar.




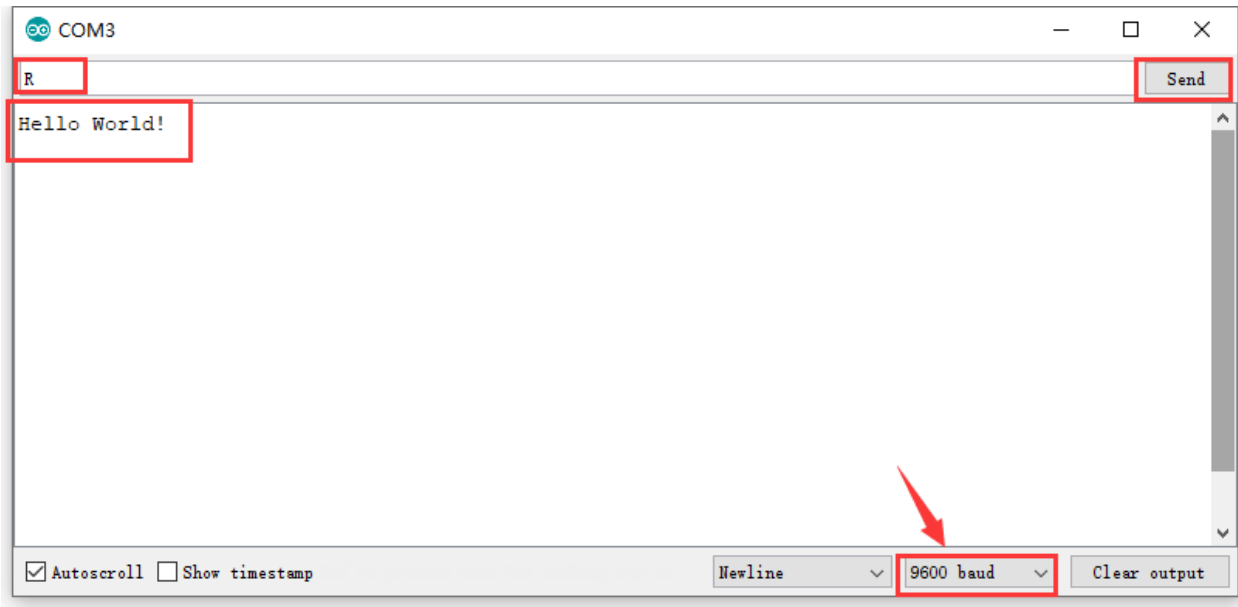
After that, click the “Upload” button to upload the code. If the upload is successful, the message “Done uploading.” will appear in the status bar.







Then click  to open serial monitor and set the baud rate to 9600, enter an “R” and click Send, that is, the computer will send the character R. When NANO board receives it, you should see the RX led on the board flash once, and then D13 led flash once; when keyestudio NANO board sends “Hello World!” to the computer, finally you should see the “Hello World!” is showed on the monitor, and TX led on the board flash once.



## 5. MAC System

### 5.1 Install Arduino IDE on MAC System

The installation instruction is as same as the chapter 4.1, as shown below:



### 5.2 Download Driver of CH340

<https://fs.keyestudio.com/CH340-MAC>

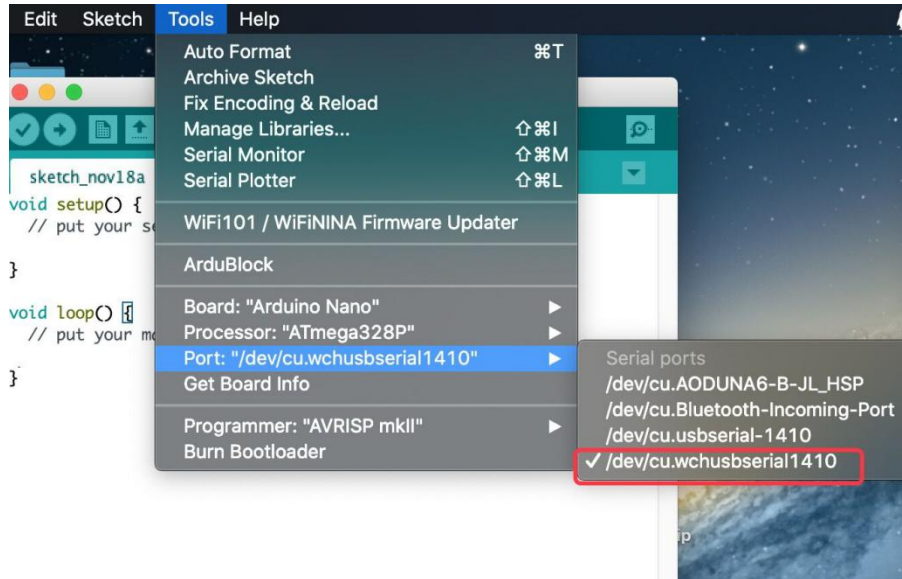
### 5.3 How to Install Driver of CH340

Please refer to the following link:

[https://wiki.keyestudio.com/Download\\_CH340\\_Driver\\_on\\_MAC\\_System](https://wiki.keyestudio.com/Download_CH340_Driver_on_MAC_System)

#### 5.4 Setting Arduino IDE

The setting method is as same as the chapter 4.4 except from COM port, as shown below:



#### 6.Shipping List

Keyestudio NANO PLUS\*1pcs

30cm Blue mini USB\*1pcs