Make Your Computer Ready

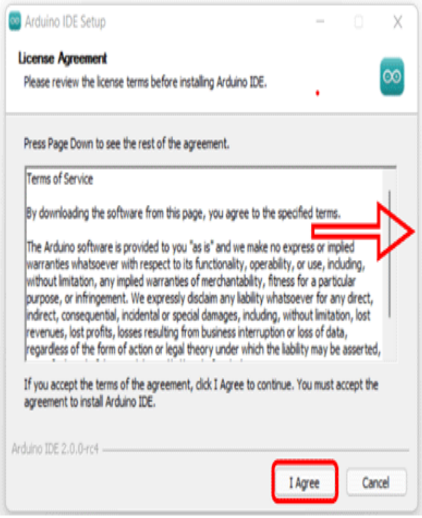
Introduction to Arduino Programming

What is the Arduino IDE? As we know we need a text/code editor to write the code, a compiler to convert that code to machine code or binary files so that the microcontroller can understand, and also programming software to load these firmware files onto the microcontroller. When we combine all these with some additional features like debugging support, console support, etc, that's what we call an IDE (Integrated Development Environment) or in simple terms the Arduino Software. Arduino IDE, as the name states, is a development IDE for the Arduino boards. It consists of a feature-rich code editor, compiler, programmer, serial console, serial plotter, and many other features. It is simple and easy to use.
Arduino IDE is cross-platform, and it can run on operating systems from Microsoft, Linux, and Windows. Furthermore, you can program the boards using the Arduino IDE and Arduino Language, which is a derivative of C/C++.

How to Install Arduino IDE?

Installing Arduino IDE is pretty straightforward. Go to Arduino IDE to download the recent version of Arduino IDE. There are multiple versions available for different operating systems such as Windows, Mac, and Linux. Not only that, nowadays Arduino IDE comes in two variants Arduino IDE 1.x and Arduino IDE 2.x. For this tutorial, we will be talking about the Classic 1.X variant. Basically, both have almost same functionality with a different GUI plus some additional features such as auto code completion.

- Download the installer for your operating system from the above-given link.
- Once the download is done open the .exe file.
- Agree to the licence agreement and select if the IDE should be installed to all users or not and click on “Next” to continue.
- Select the location in which you want the IDE to install if you want to change the location or keep it default and click on “Install”
- Wait for the installer to finish installation, and click on “Close”.


Arduino IDE Setup

License Agreement

Please review the license terms before installing Arduino IDE.

Press Page Down to see the rest of the agreement.

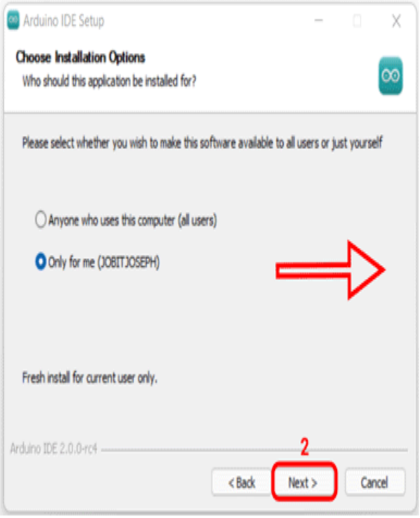
Terms of Service

By downloading the software from this page, you agree to the specified terms.

The Arduino software is provided to you "as is" and we make no express or implied warranties whatsoever with respect to its functionality, operability, or use, including, without limitation, any implied warranties of merchantability, fitness for a particular purpose, or infringement. We expressly disclaim any liability whatsoever for any direct, indirect, consequential, incidental or special damages, including, without limitation, lost revenues, lost profits, losses resulting from business interruption or loss of data, regardless of the form of action or legal theory under which the liability may be asserted,

If you accept the terms of the agreement, click I Agree to continue. You must accept the agreement to install Arduino IDE.

Arduino IDE 2.0.0-rc4


Arduino IDE Setup

Choose Installation Options

Who should this application be installed for?

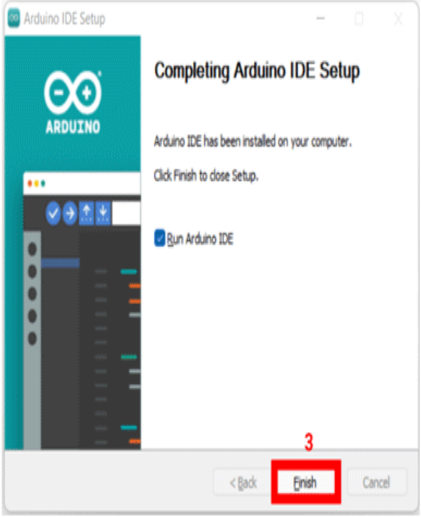
Please select whether you wish to make this software available to all users or just yourself

○ Anyone who uses this computer (all users)

● Only for me (JOBITJOSEPH)

Fresh install for current user only.

Arduino IDE 2.0.0-rc4


Arduino IDE Setup

Completing Arduino IDE Setup

Arduino IDE has been installed on your computer.

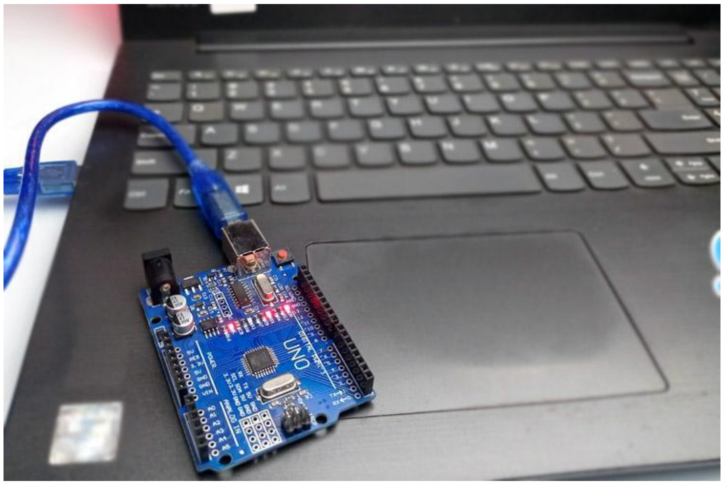
Click Finish to close Setup.

☑ Run Arduino IDE

Arduino Driver Installation

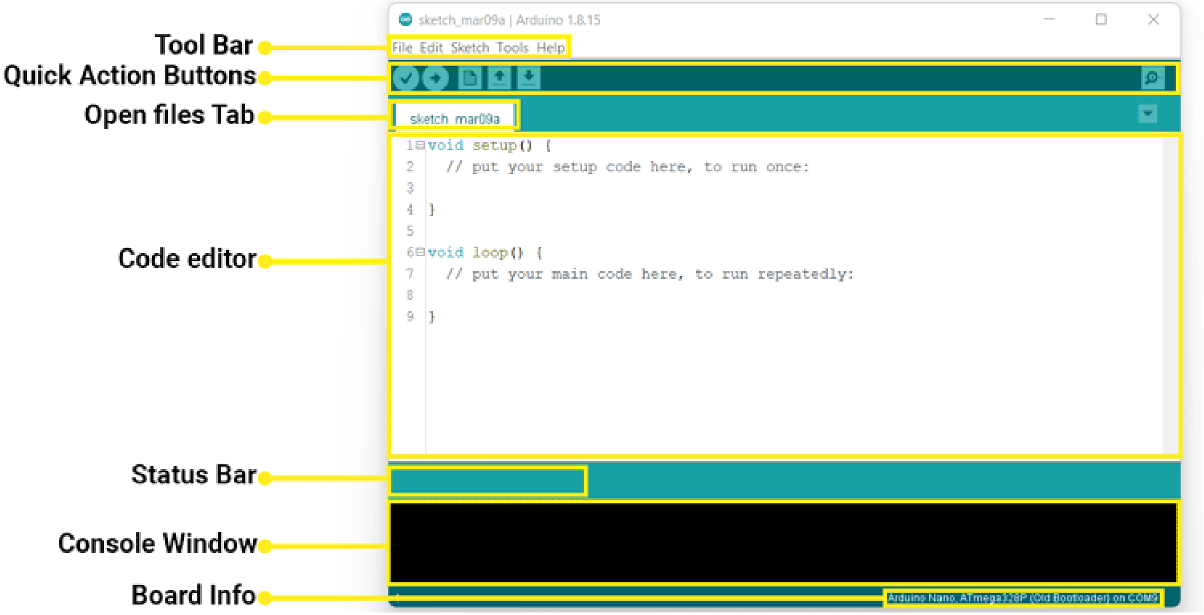
One common question that everyone gets when starting with Arduino Software is about the board driver installation. But the truth is that you don’t have to worry about it. The installer will load all the necessary files to the computer and will install the drivers automatically as soon as you connect the board to the computer.
Connecting Arduino Board to the Computer

To connect the Arduino board to the computer, simply connect the appropriate cable to the Arduino board and connect the other end to the USB port of your PC. The power LED will glow indicating the board is powered. The system will automatically install the driver for the board.



Arduino IDE – Basics

After the Arduino IDE installation, you can launch the Arduino IDE by double-clicking the Arduino IDE shortcut on the Desktop or from the Start Menu. Now the Arduino IDE will be opened. The below image shows the Arduino IDE interface.

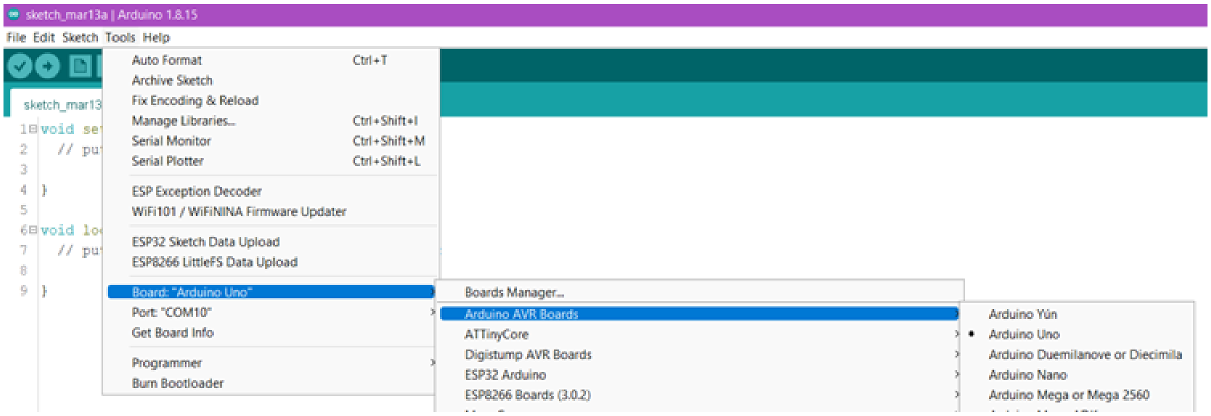


```
void setup() {
  // put your setup code here, to run once:

}

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

}
```

Arduino Nano, ATmega328P (Old Bootloader) on COM9

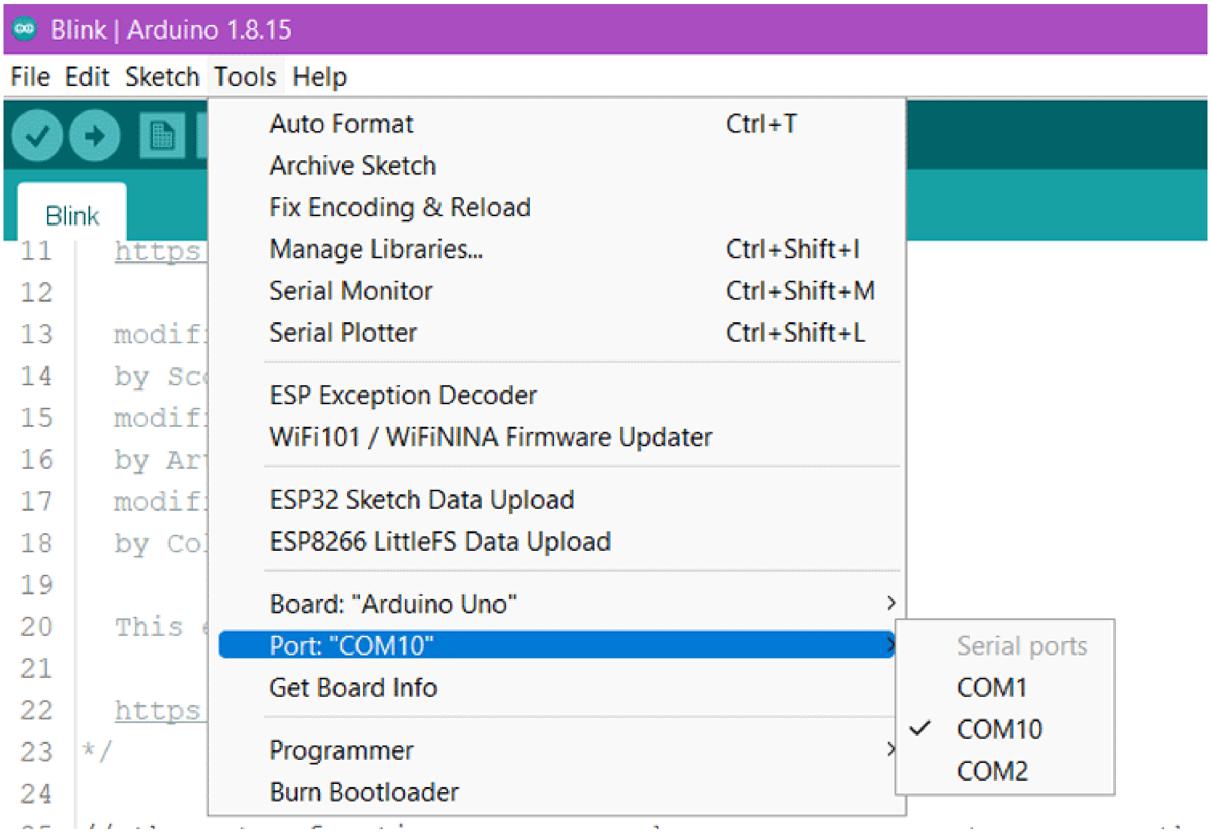
Selecting the right Board on Arduino

Now let’s select the proper board. It is very important to select the proper board before compilation because the compiler will use this in the compile directives. To do that Click on the “Tools” -> “Board” -> “Arduino AVR Boards” and select your board from it.



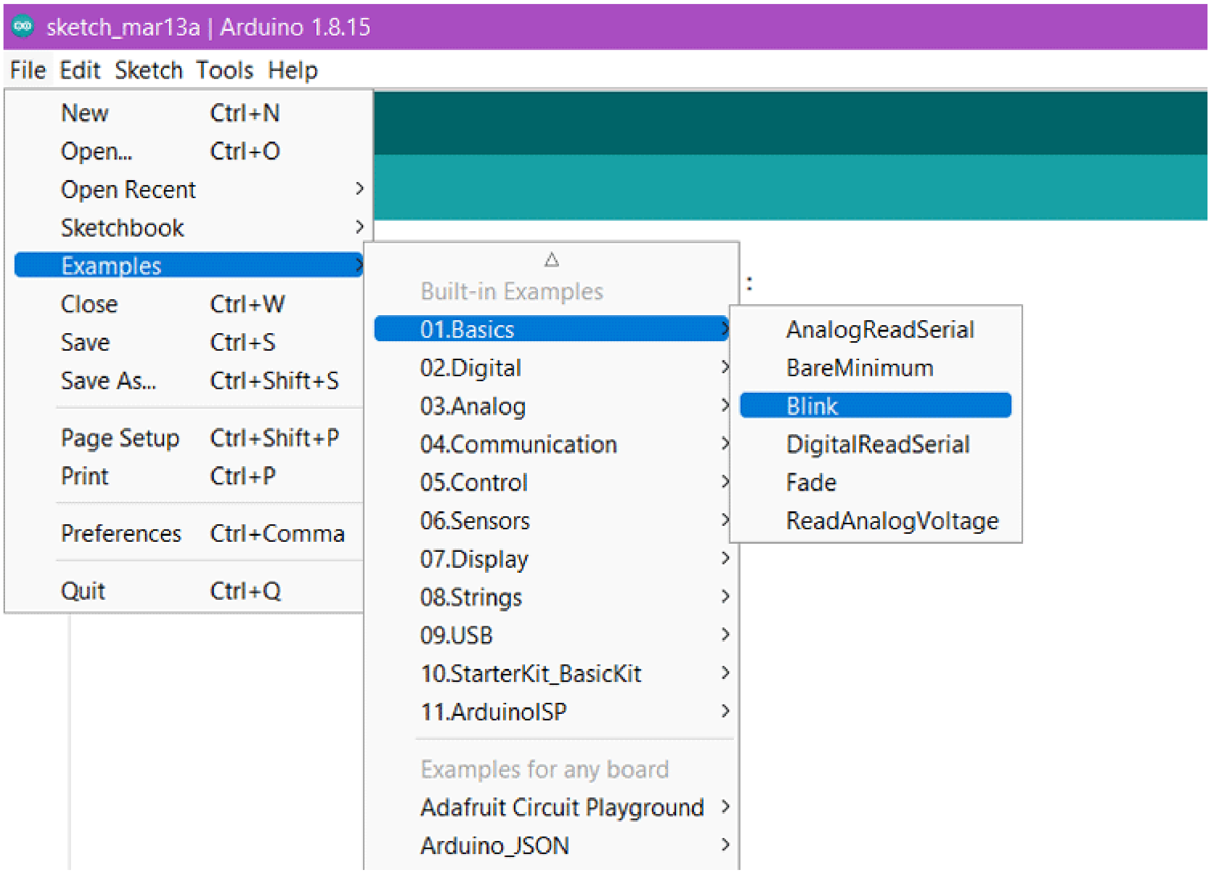
Select the Arduino Serial Port

It’s also important to select the proper serial port to which the Arduino board is connected. Otherwise, you won’t be able to upload the code to the board. To do that click on “Tools” -> “Port” And select the correct COM port. If there are multiple COM ports and you are not sure which one to select, disconnect the Arduino board from the USB port and reopen the menu. The COM port entry that disappeared will be the correct COM port. Reconnect the Board to the same USB port and select that port.



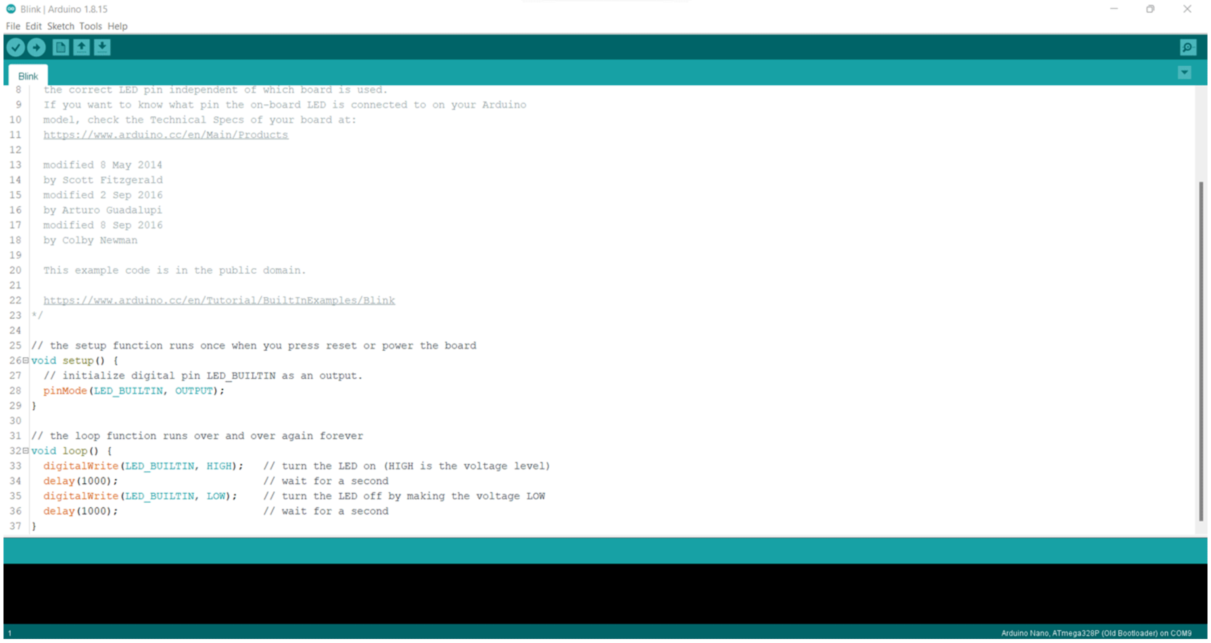
Arduino Example Codes

To start with we will be using the LED Blink example that is already provided with the Arduino IDE. To open the Blink example, go to Files -> Examples -> Basics -> Blink.



Example Blink Program for Arduino

Let’s open an example code to understand it further. For that click on File -> Examples -> Basics -> Blink.
A new window with the example blink code will be open. In the code area, you can see two functions, void setup() and void loop(). The void setup() function is the first function that will run when the Arduino is powered on. Normally we will add any initialization codes in this function. The void loop() will run in a loop, and we will add any code that needs to be repeated to this function.


Blink | Arduino 1.8.15
File Edit Sketch Tools Help
Blink
1
Arduino Nano, ATmega328P (Old Bootloader) on COM9

```
  the correct LED pin independent of which board is used.
  If you want to know what pin the on-board LED is connected to on your Arduino
  model, check the Technical Specs of your board at:
  https://www.arduino.cc/en/Main/Products

  modified 8 May 2014
  by Scott Fitzgerald
  modified 2 Sep 2016
  by Arturo Guadalupi
  modified 8 Sep 2016
  by Colby Newman

  This example code is in the public domain.

  https://www.arduino.cc/en/Tutorial/BuiltInExamples/Blink
*/

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

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

As mentioned earlier, in the void setup() function there is an initialization code.

```
pinMode(LED_BUILTIN, OUTPUT);
```

In this line with pinMode(), we set the pin which is connected to the built-in LED as an output. pinMode() is used to set a specific pin as INPUT or OUTPUT. The syntax is pinMode (pin, mode). In which pin is the pin number, and the mode is INPUT or OUTPUT. In the example we are setting the built-in LED pin, i.e., pin number 13, which is defined in the Arduino board definition file, to an output.

```
Void loop () {
digitalWrite (LED_BUILTIN, HIGH);   // turn the LED on (HIGH is the voltage level)
delay(1000);                        //wait for a second
digitalWrite(LED_BUILTIN, LOW);   //turn the LED off by making the voltage Low
delay(1000);                      // wait for a second
}
```

If we look at the Loop function, we are calling two functions repeatedly digitalWrite() and delay().
The digitalWrite() function is used to change the state of a specific pin, i.e. we change the pin's state to high or low. The syntax is digitalWrite(pin, value) in which the pin is the corresponding pin number and the value is the state or value we want to set to the pin, i.e., HIGH or LOW.

Activity 1

Print "Hello World"

Component Required

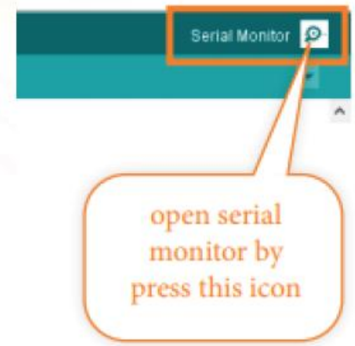
- Arduino Uno
- Arduino Uno Cable

Procedure to upload the code

- Connect an arduino board to your computer/laptop using an arduino cable.
- Open arduino IDE.
- Open program file in arduino IDE.
- Go to

File->Open>Robotic Starter Kit>Select hello_world.

- Go to Tools->Board->Select Arduino Uno.



open serial monitor by press this icon

- Go to Tools->Port->Select Com Port(Arduino uno).

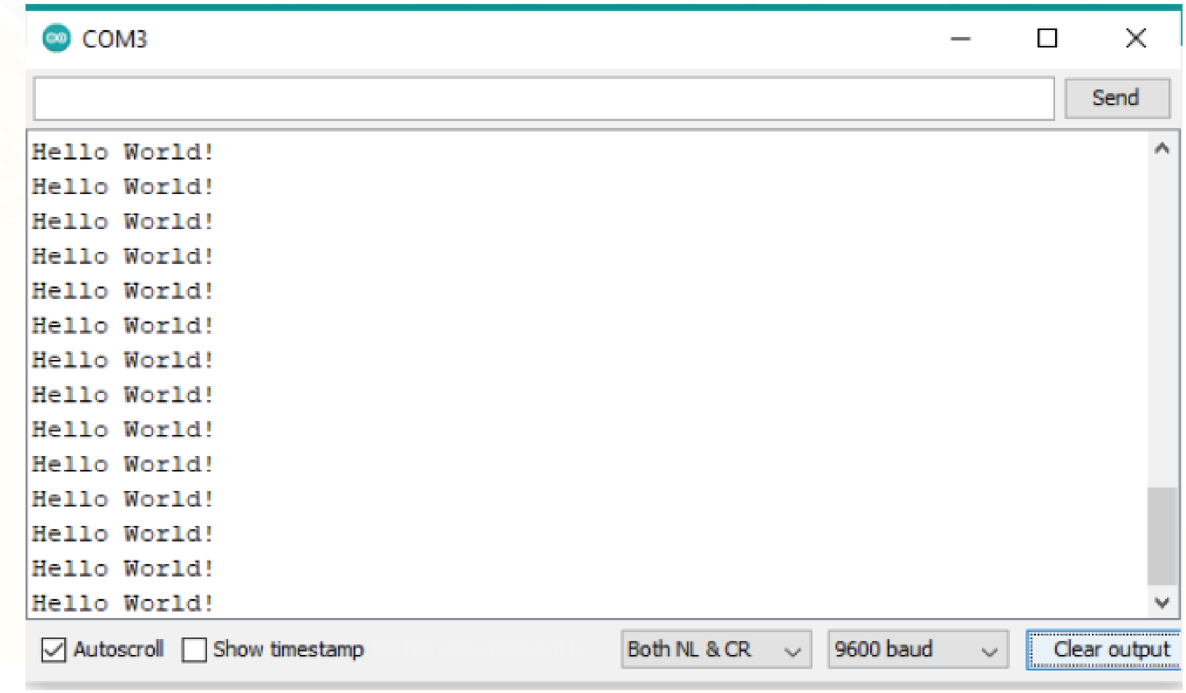
- Click on the compile/verify option.
- Click on the upload option.

Program

```
void setup(){
Serial.begin(9600);
}
void loop(){
Serial.println(“Hello World!”);
}
```

Output

1. To see the text on the serial monitor, click on this icon in your arduino IDE.
2. You can see your text Hello World! there.



```
Hello World!
Hello World!
Hello World!
Hello World!
Hello World!
Hello World!
Hello World!
Hello World!
Hello World!
Hello World!
Hello World!
Hello World!
Hello World!
Hello World!
```

☑ Autoscroll □ Show timestamp Both NL & CR 9600 baud

Note:

Select baud rate 9600 baud.