

# Arduino Platform and Genuino101

## Introduction to Arduino Platform

Arduino is an open-source electronics hardware and software platform for everyone to build devices and interactive objects that sense and control the physical world.

Arduino microcontroller is designed to be working together with various kinds of sensors and actuators by receiving and sending out signal through its analog or digital I/O (input/output) pins. Code for programming the Arduino boards is based on C language. The program is edited and uploaded to an Arduino board through the Arduino IDE software (A free integrated development environment software based on Processing). Arduino can either work independently as a single-board microcontroller, or communicate with software such as Flash, Processing or MaxMSP, which makes it an ideal development tool for students, computer programmers, hobbyists and industrial artists.

The following interactive projects may help you better understand what Arduino can do:

- Coffee maker that plays music when coffee gets ready.
- Smart gadget that rings your phone when e-mail comes.
- A shinning fluffy toy that changes pattern when being touched.
- Professor X' s wheelchair that talks to you and brings you water and drinks.
- Portable monitor that records your heart rate when riding a bicycle.
- A robot that draws lines of your command on the ground.

**With Arduino, all these fascinating projects are just within touch!**

## The Birth of Arduino

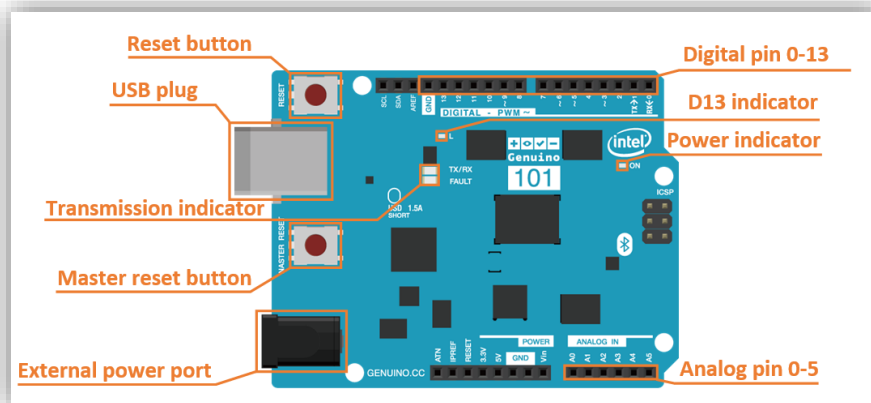
Arduino started as a project for students at the Interaction Design Institute Ivrea, in Italy. Members of Arduino's core development team were Massimo Banzi, David Cuartielles, Tom Igoe, Gianluca Martino, David Mellis and Nicholas Zambetti.

Massimo Banzi's students often complained that they couldn't find a cheap and easy-to-use micro-controller. In the winter of 2005, Banzi mentioned this issue while talking with David Cuartielles, a Spanish chip engineer who was a visiting scholar to Massimo's university. They decided to design their own circuit board and brought Banzi's student David Mellis into this project to design the programming language for their board. Mellis finished the source code within two days. It took them another three days to etch the circuit board. They named it Arduino.

Now, Arduino has become an extremely popular development tool around the world. Although the most classical model, the Arduino UNO, still remains popular, Arduino has been collaborating with other companies and releasing modified versions of Arduino boards to fulfill different needs. In the tutorial, we will be working on one of Arduino's latest release, the Genuino 101.

## Meet Genuino 101

Genuino 101(outside USA) and Arduino 101 (USA only) is the latest Arduino developing platform based on the Intel Curie module, a latest micro computing unit which includes two-core processor, supporting both x86(Quark) and ARC architecture. Compare to Arduino UNO, the most classical Arduino microcontroller, 101 is significantly better in performance and power consumption. More than that, 101 is equipped with Bluetooth 4.1 module and six-axis accelerometer and gyroscope, featuring functions such as gesture recognition and Bluetooth communication. Meanwhile, 101 has also kept the simplicity of Arduino. It is designed to have the same I/O pin layout as Arduino UNO, so that you can connect any Arduino UNO compatible components or expansion shields to Genuino 101 and program it through Arduino IDE.



101 comes with 14 digital I/O pins (4 PWM output), 6 analog input pins, micro USB connector for serial communication and sketch upload, a power jack, an ICSP header and I2C pins.

However, a major difference between Arduino UNO and 101 is that the operating voltage of Genuino 101 is rated as low as 3.3V (but still 5V tolerant), which means it may not be enough for some 5V compatible components (mostly output components). Nevertheless, you can still use most of them without safety concern.

You may also notice that 101 comes with both a "RESET" button and a "MASTER RESET" button. This is the result of the dual-core structure of the Intel Cherry module. Different from most of other Arduino boards, 101 runs on a complete operating system (ViperOS RTOS) under the x86 architecture core, whereas its ARM core manages its I/O pin signals. Therefore, when the "master reset" button is pressed, the entire board starts rebooting, which not only restarts its Arduino program, but also the operating system and Bluetooth connection. However, "RESET" button only restarts the Arduino program.

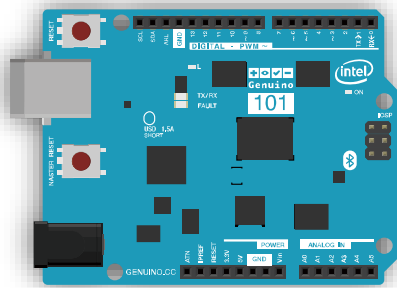
## Get Started with Genuino101

For experienced users, you can follow the guide listed on the main page of Genuino 101 to get started.

Genuino 101 main page: <https://www.arduino.cc/en/Guide/Arduino101>

However, if you are new to Arduino, we also provide a step-by-step tutorial to guide you through downloading and installation of the Arduino IDE software, and eventually teach you how to program 101.

Before we start, make sure you have the following things by your hand:



Genuino 101 board



USB A to B Cable

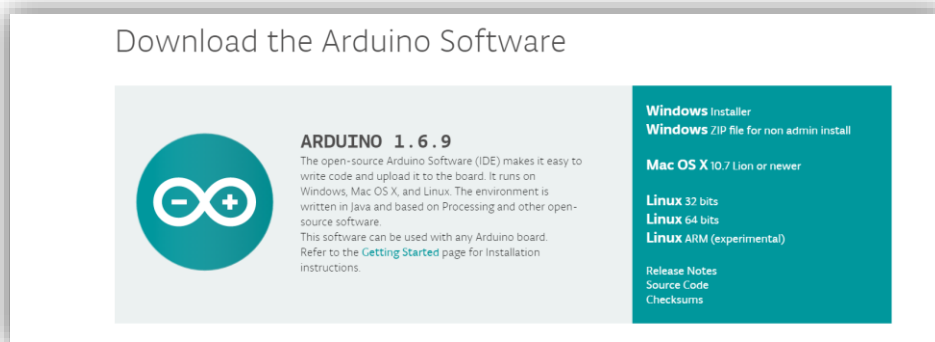
Computer that runs Windows/Mac OS/Linux with internet connection.

### 1. Download the latest Arduino IDE software

The following installation procedure is based on Windows OS. Similar steps may apply to other OS.

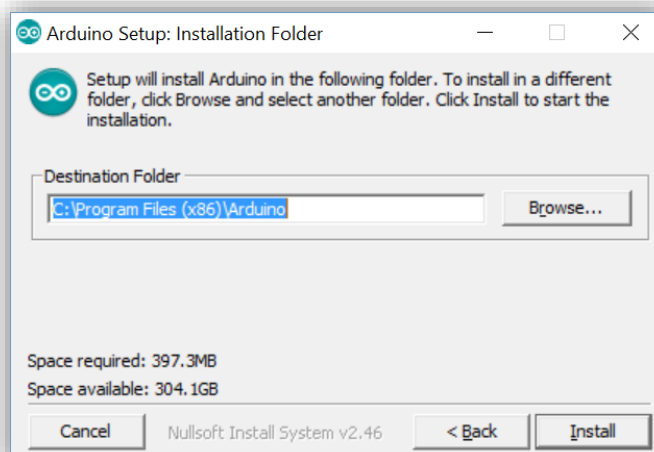
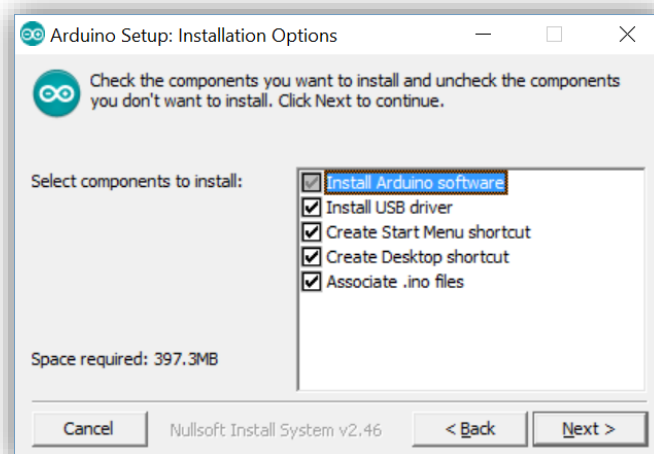
To begin with, you will first need to download the **latest Arduino IDE software** from the official website. Need to notice that any version earlier than **1.6.7** does NOT have Genuino 101 support added.

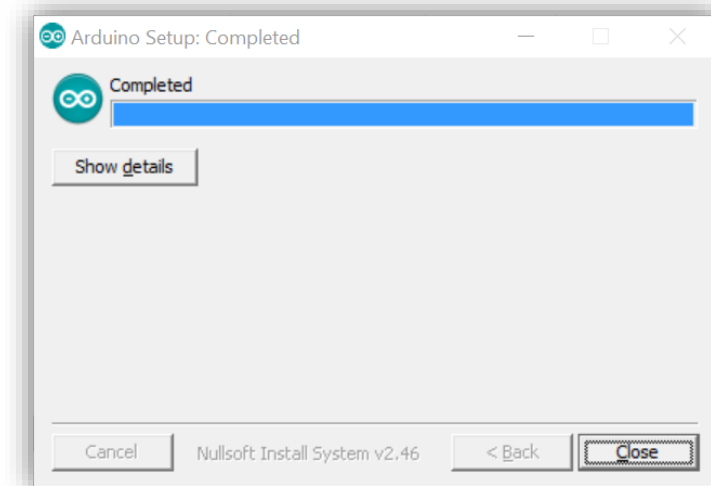
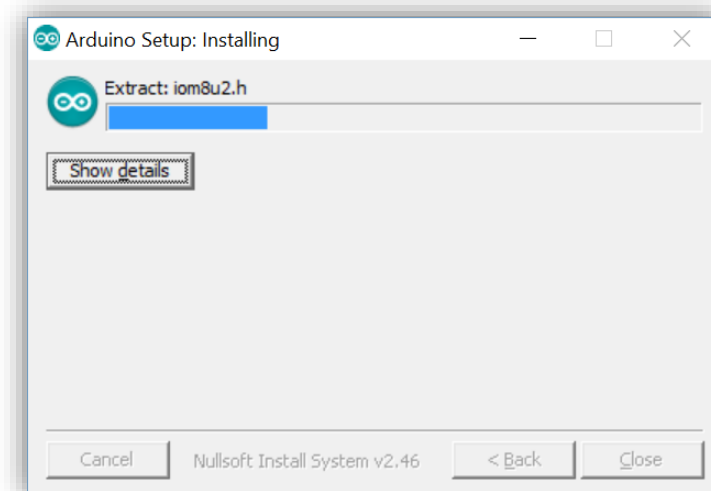
Download link: <http://arduino.cc/en/Main/Software>



Download the installation package for your operating system from the right side of the section. For Windows users, either download the **Windows installer** (recommended for first time user) or Windows ZIP package (You will need to manually install the driver).

For Windows installer, when the file is ready, open the package and follow the wizard to finish installation, drivers will be automatically installed when installation completes.



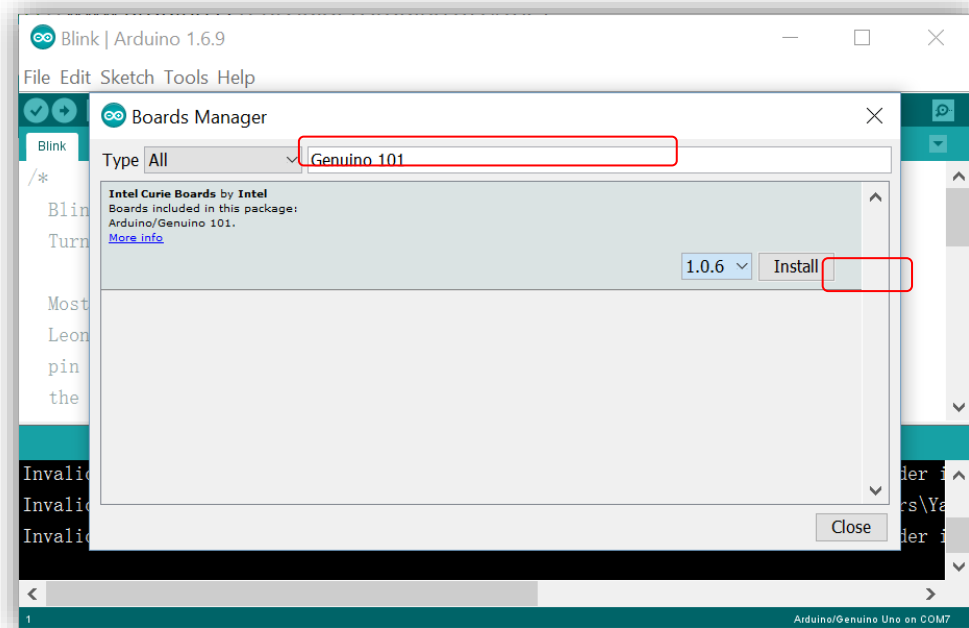
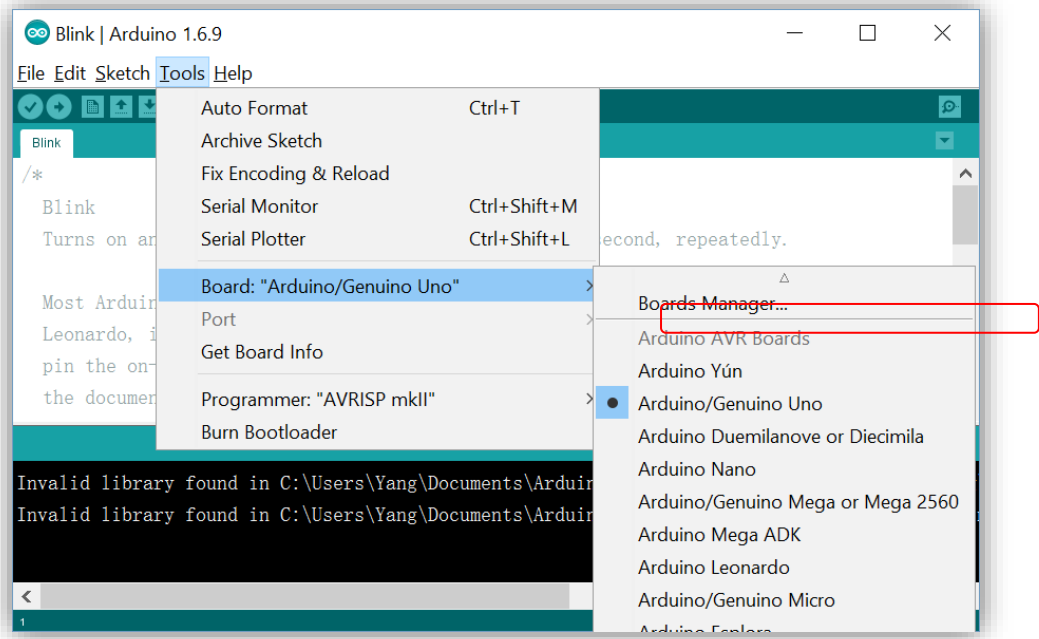


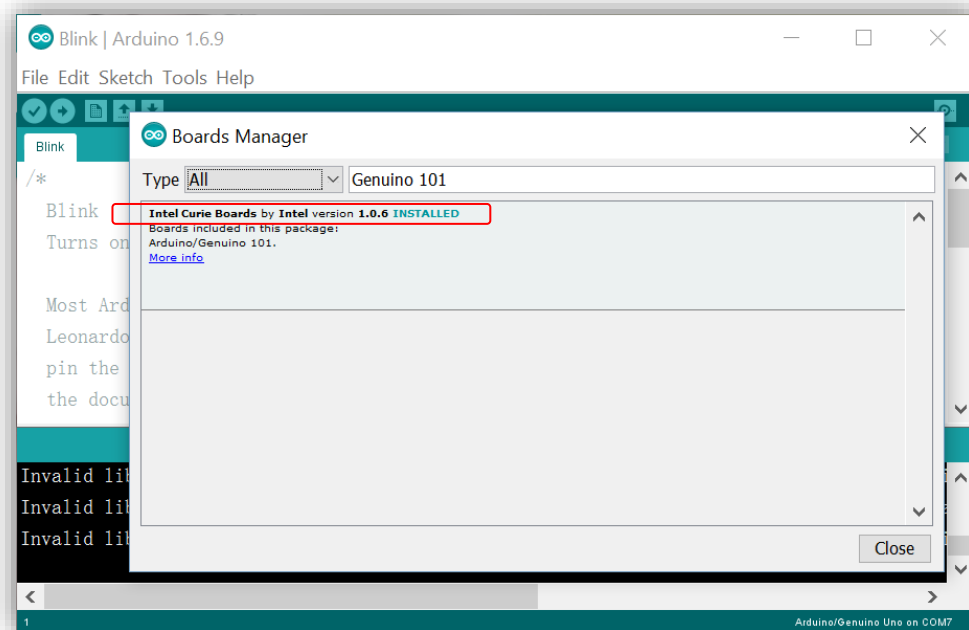
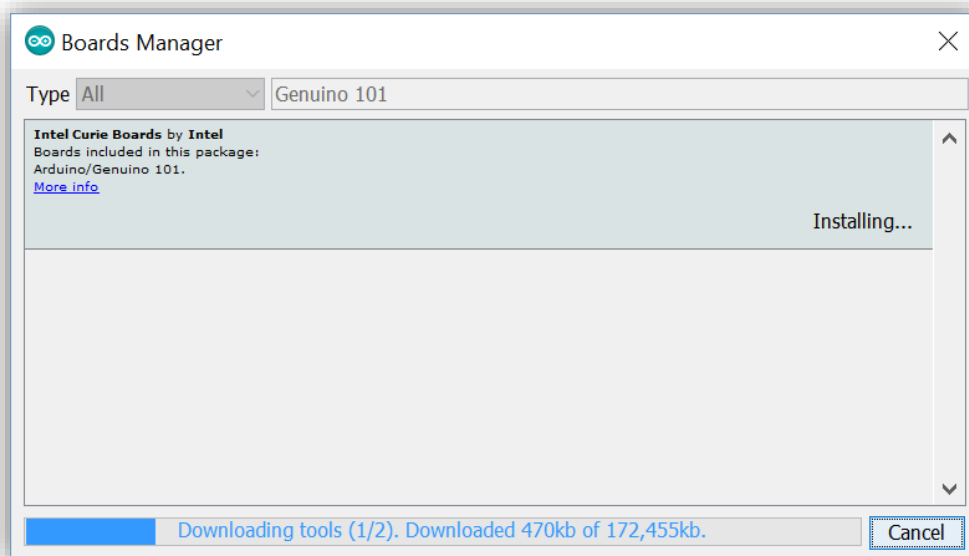
## 2. Install the Intel Curie Boards core in Arduino Boards Manager

Arduino IDE doesn't come with Intel Curie Boards core preinstalled by default. To add support for Genuino 101, you will need to install **Intel Curie boards core** using the **Arduino Boards Manager**. To do this, navigate to

**Menu Bar > Tools > Board: xxx > Boards Manager.**

In the popped out window, type in **"Genuino 101"** and wait till its information shows up. Select version 1.0.6 (or the latest version), then click **"install"** and wait the progress bar goes to the end. The whole process may take 5-10 minutes depending on your internet connection. When successfully installed, the board info will be marked as **"INSTALL"**



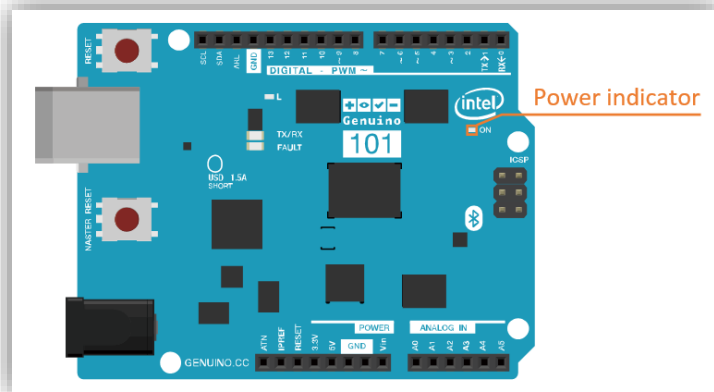


During Installation, some key process may be blocked by the firewall or anti-virus software. Select to set allowing making changes and add them to the white list. You may need to remove and reinstall Intel Curie Boards core if 101 doesn't work normally in next steps.



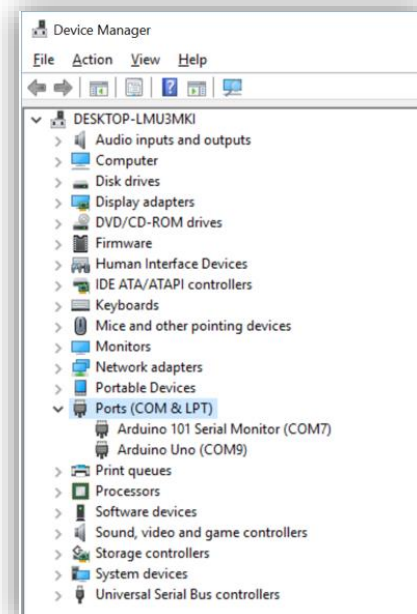
### 3. Connect Genuino 101 to the computer

Once the Arduino IDE and Intel Curie Boards core are correctly installed, you can now connect your Genuino 101 to the computer using the USB A to B Cable. You will see the LED light on top right corner lights up when Genuino is powered.



Before programming, we will need to make sure that the board is recognized by your computer and find out which **COM port** (provides serial communication interface) it's on. It can be checked through the following procedure.

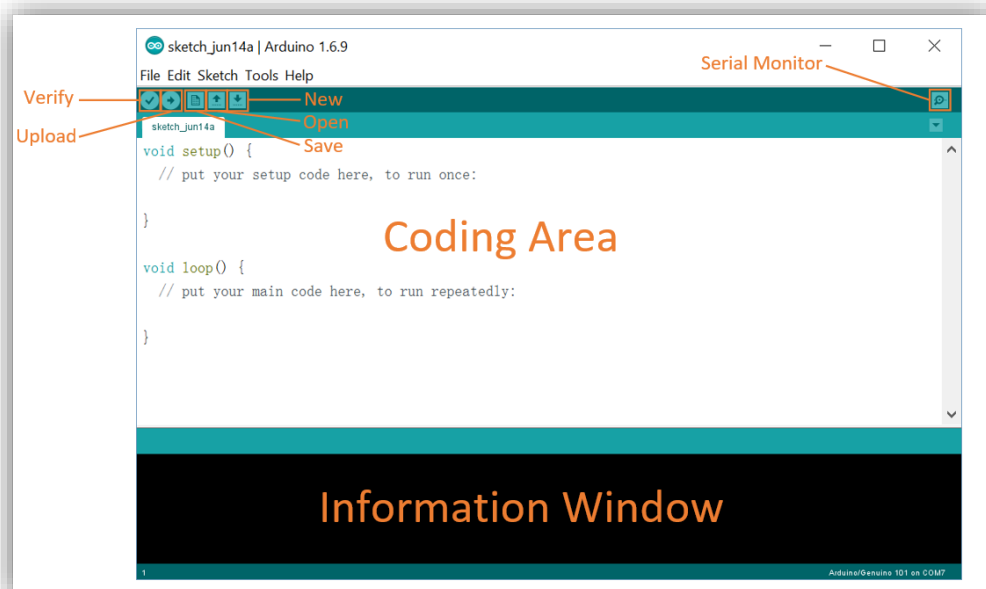
Open "**Control Panel**" and navigate to "**Device Manager**", expand "**Ports (COM & LPT)**". The COM port for Genuino (Arduino) 101 will then show up on the list (COM7 in our case).



#### 4. Start programming in Arduino IDE

Run the Arduino IDE software installed in the previous steps and it opens up a window as following. This window allows you to edit and upload code to your Arduino boards. You can also communicate with the boards using the integrated serial monitor through COM port.

Now let's take a close look of the Arduino IDE interface.

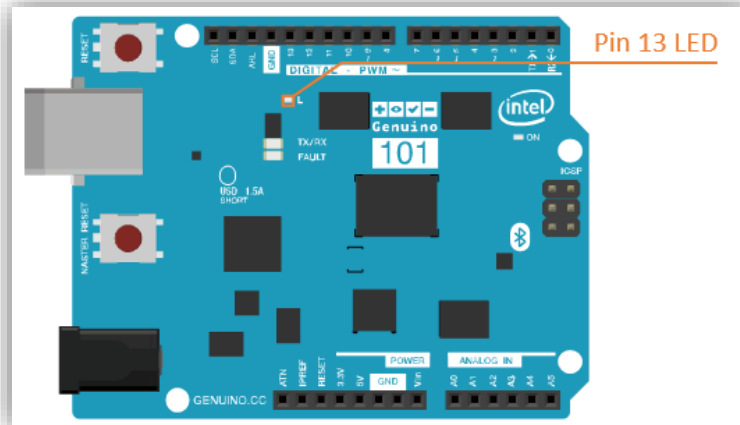


The white area is where you put the code in. Code for initialization the program is put under **"void setup()"** and it only be running once after the board boots up. After that, code under **"void loop()"** run repeatedly, allowing board to exchange data in real time.

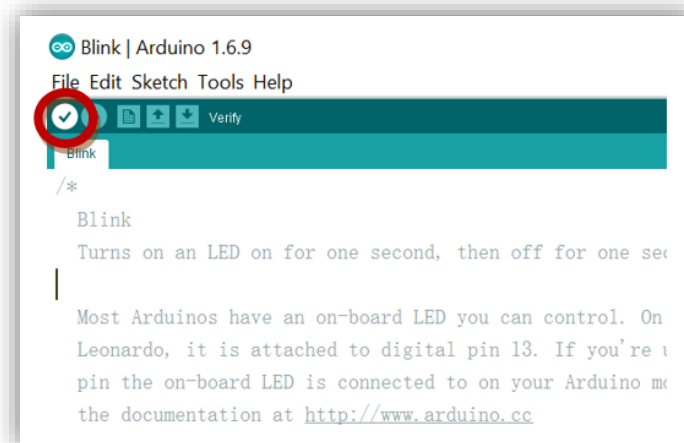
The black area below is the information window, where you can read notification or error information when code is being compiled or uploaded.

#### 5. Upload program to Genuino 101

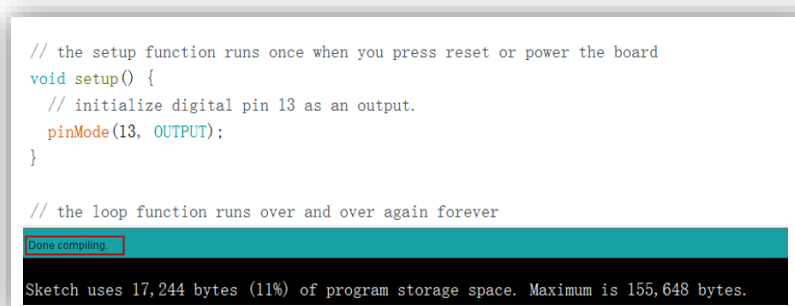
In this step, we will upload the Arduino example program "Blink" to Genuino 101. The "Blink" program turns on and off the LED light attached to digital pin 13 for every 1 second. Just like most of other Arduino boards, Genuino 101 comes with an on-board LED indicator already attached to digital pin 13, which means no further components will be needed for this example. The LED indicator can be found next to the digital pin headers.



Before uploading, you should first make sure that there is not error in your code. To do this, click **"verify"** .



Wait for a few seconds, if nothing goes wrong, a **"Done Compiling"** message will show up in the information window. If not, you may need to go back and check your code.

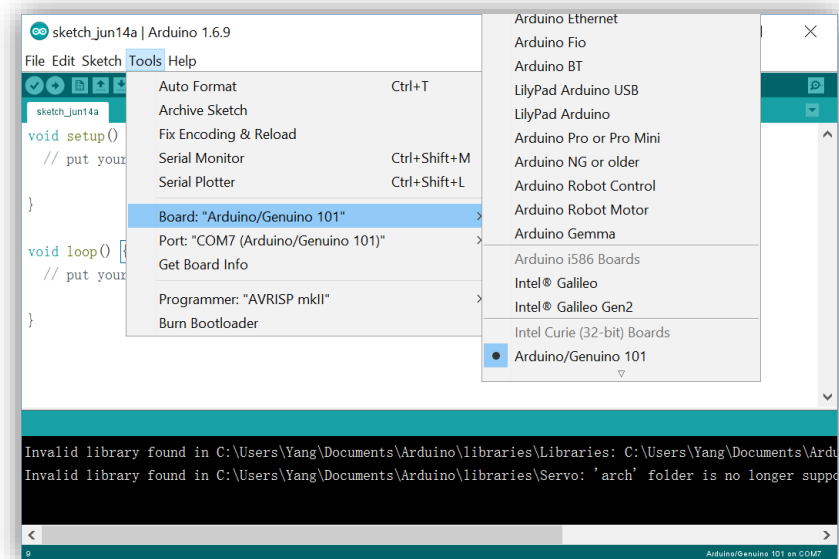


Then we need to choose the **board type** and **COM port**.

To choose the Genuino 101 as our board type, navigate to

Menu Bar > Tools > Board: and Choose **"Arduino/Genuino 101"** .

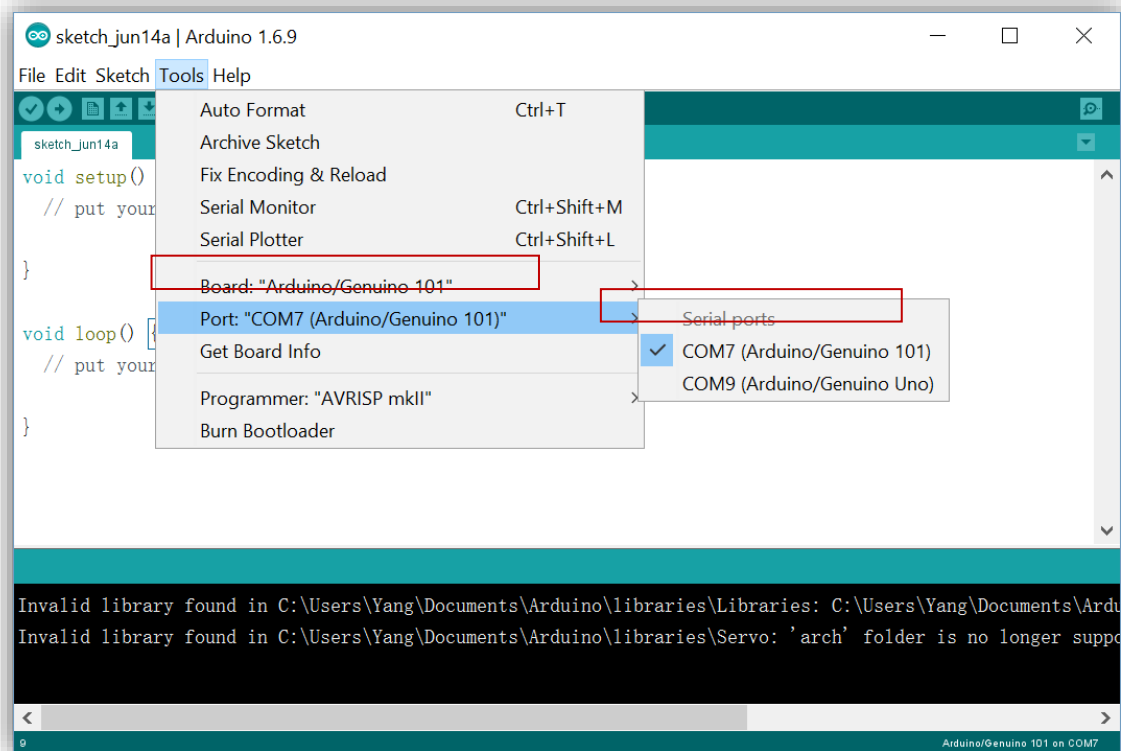
Normally this only needs to be done once unless you have switched between other types of boards.



Then we will navigate to Menu bar >Tools > Port.

As we checked in step 3, we should select **"COM7"** as the serial port. The COM port options will only show up when boards are connected and recognized by the computer. The same board may be assigned with different COM port, which means you always have to make sure the right COM port has been chosen before uploading.

Once selected, the board's information and its COM port will show up in the **bottom right** of the window.



Finally, hit “Upload” to upload the code to 101.



As 101 runs on an independent operating system, sometimes it may prevent it from updating the code. In this case, the following message will show up in the information window. To solve it, press “Master Reset” button on the board.

```
Device is not responding, check the right serial port is selected or RESET the board right before exporting
Starting download script...
Flashing is taking longer than expected
Try pressing MASTER_RESET button
ERROR: Device is not responding.
Device is not responding, check the right serial port is selected or RESET the board right before exporting
```

When successfully uploaded, the message **"Done uploading"** will appear in the information window. Now you should be able to see the LED light on D13 starts blinking.

```
// the setup function runs once when you press reset
void setup() {
  // initialize digital pin 13 as an output.
  pinMode(13, OUTPUT);
}

// the loop function runs over and over again forever
Done uploading.
Invalid library found in C:\Users\Yang\Documents\Arduino\libraries\
Invalid library found in C:\Users\Yang\Documents\Arduino\libraries\
Invalid library found in C:\Users\Yang\Documents\Arduino\libraries\
Invalid library found in C:\Users\Yang\Documents\Arduino\libraries\
18
```

In sum, there are always 3 things to do before upload:

1. Verify code
2. Select board and COM port
3. UPLOAD!

That' d be all you need to get your 101 on the run. Feel free to contact us in our forum for any question or concern.

**Link:** <http://www.dfrobot.com/forum/>

Remember to check out DFRobot' s community for more tutorial and exciting projects. We also hope that you can post your project or thoughts on the forum. Welcome to become one of us!