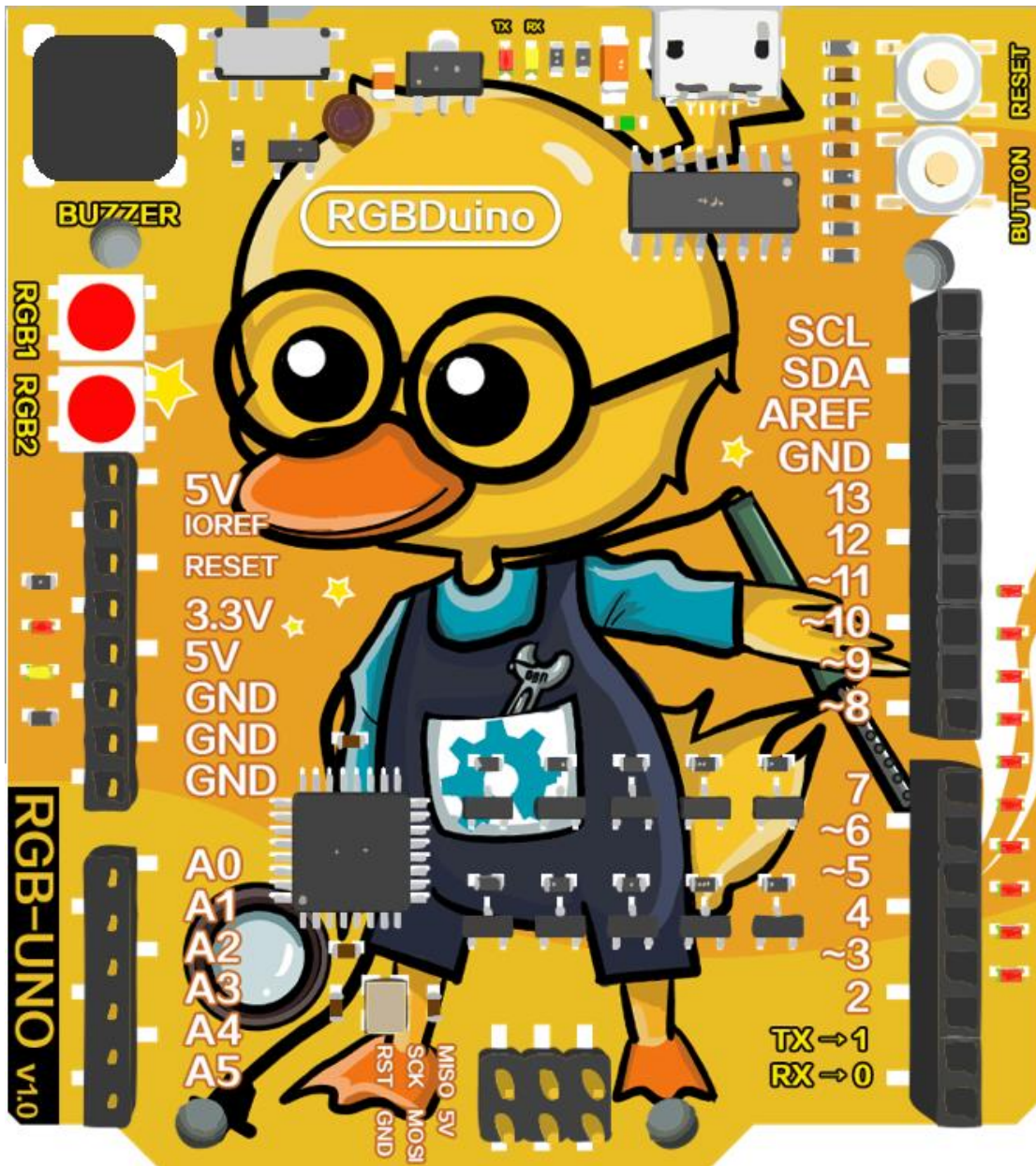


Start learning RGBDuino UNO



Start marking something!

CONTENTS

INTRODUCTION

IntronductiontoComponents

- RGBDuino Uno
- RGBDuino Uno Board

Setting up

- Dowload Arduino IDE
- Install RGBDuino UNO Drivers
- Install RGBDuino UNO libray

LESSONS

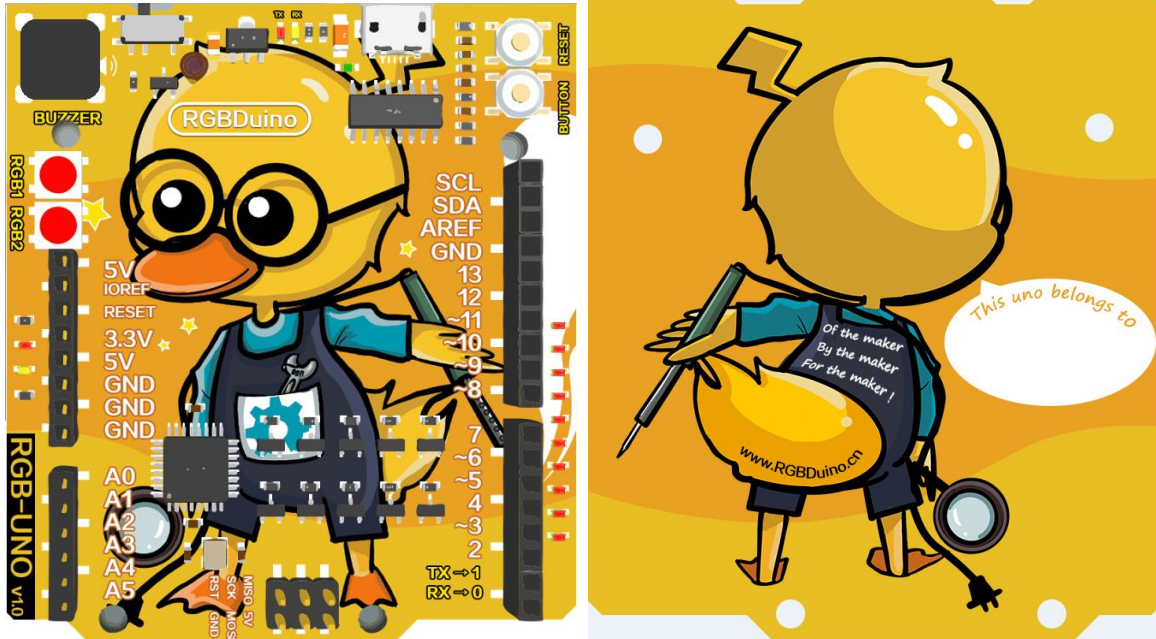
1. The led (Digital OUTPUT)
2. LED Blinking
3. LED OutPut (PWM)
4. Push Button
5. Melody Dance
6. RGB Blink
7. Controlling Motor

INTRODUCTION



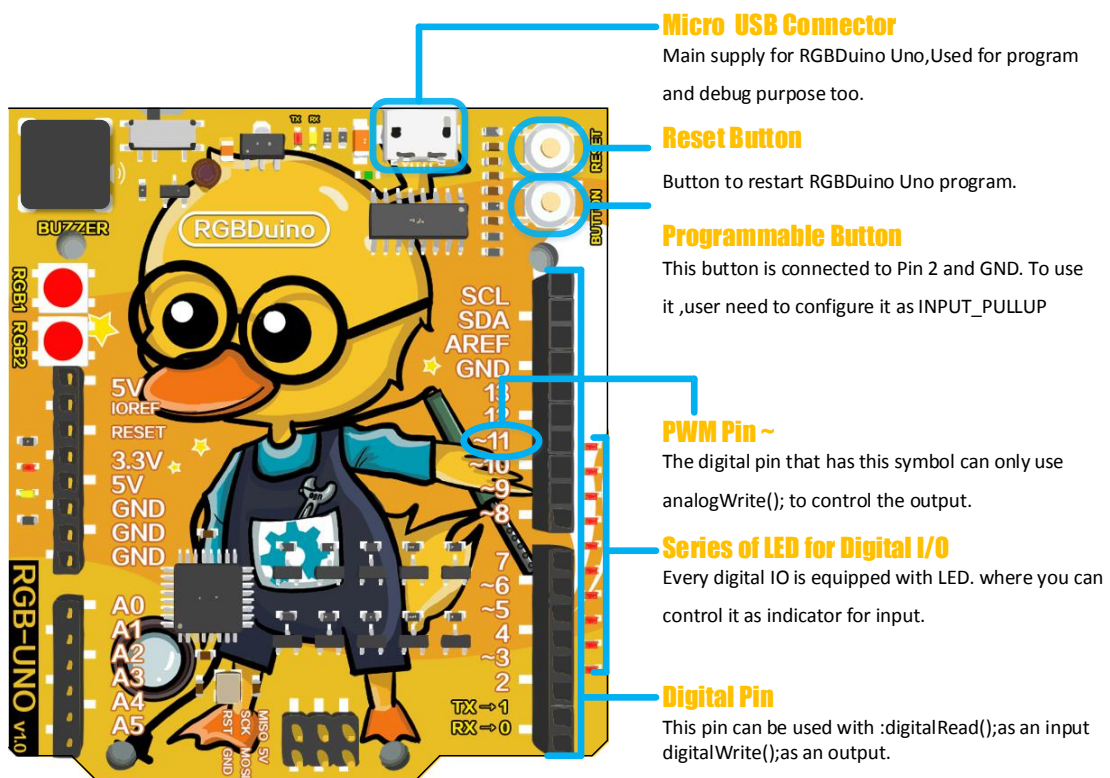
RGBduino UNO

RGBduino UNO compatible board designed and developed specially for students to learn coding and microcontroller. We named it RGBduino UNO to encourage everyone to be a maker by getting started with this amazing board.



UNO Features:

- SMD ATmega328P microcontroller (the same microcontroller on Arduino UNO)
- Input voltage: USB 5V, from computer, power bank or standard USB adapter.
- 500mA (maximum) 3.3V voltage regulator.
- 0-5V outputs with 3.3V compatible inputs.
- 14 Digital I/O Pins (6 PWM outputs).
- 6 Analog Inputs.
- ISP 6-pin Header.
- 32k Flash Memory.
- 16MHz Clock Speed.
- 10 blue LEDs and two RGB programmable lamps
- MICRO USB power and Programming port
- Buzzer that can play music
- More secure plug-in-free design
- On board programmable push button
- Use Micro-usb socket.
- Cute little yellow duck pattern!



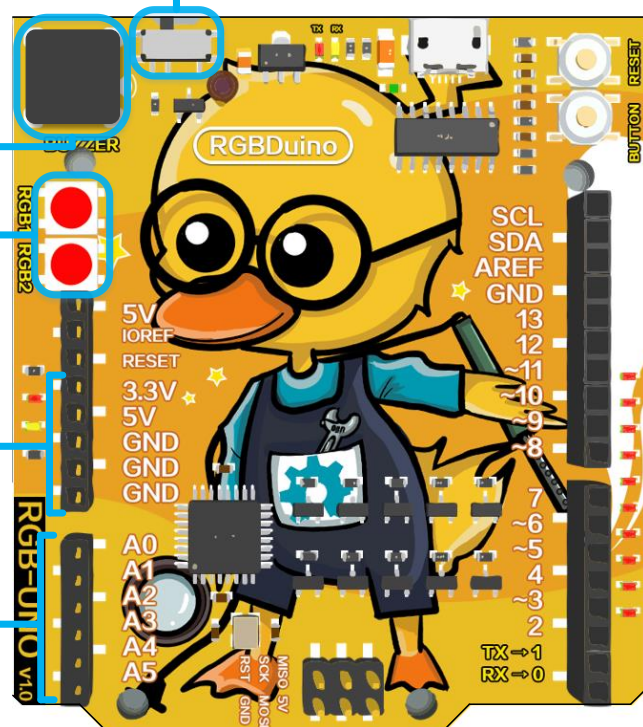
Piezo Buzzer Slide Switche
Slide switch to connect between pin 8 to piezo buzzer. To use piezo buzzer,slide the switch on and program the buzzer.To use pin 8 for other Purpose ,slide the switch off.

Piezo Buzzer
Piezo buzzer is connected to Pin 8 through slide switch.

RGB lamp
Two RGB lamp is connected to Pin12 and Pin13, There is programmable lamps

Power Pin
GND-Ground Pins
5V-Regulated 5V output
3V3-Regualted 3.3v supply

Analog Pin
This pin can be used with analogRead(); to read an input in analog form (0~1023)



DOWNLOAD Arduino IDE

RGBduino UNO requires Arduino software to run. You can download the software from Arduino website (<http://arduino.cc/en/Main/Software>) and it is free to use. www.arduino.cc

Download the Arduino IDE



ARDUINO 1.8.10

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


Mac OS X 10.8 Mountain Lion or newer

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

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

HOURLY BUILDS

Download a **preview of the incoming release** with the most updated features and bugfixes.

[Windows](#)

[Mac OS X](#) (Mac OSX Mountain Lion or later)

[Linux 32 bit](#) , [Linux 64 bit](#) , [Linux ARM](#) , [Linux ARM64](#)

BETA BUILDS

Download the **Beta Version** of the Arduino IDE with experimental features. This version should NOT be used in production.

[Windows](#)

[Mac OS X](#) (Mac OSX Mountain Lion or later)

[Linux 32 bit](#) , [Linux 64 bit](#) , [Linux ARM](#) , [Linux ARM64](#)

Arduino IDE is compatible with Windows, Mac OS X and also Linux. You just need to choose the appropriate operating system installation package for your computer. If you are a Windows user, it is recommended that you choose Windows (installer).

Choose the installer that compatible with your laptop OS and download the

Arduino IDE.

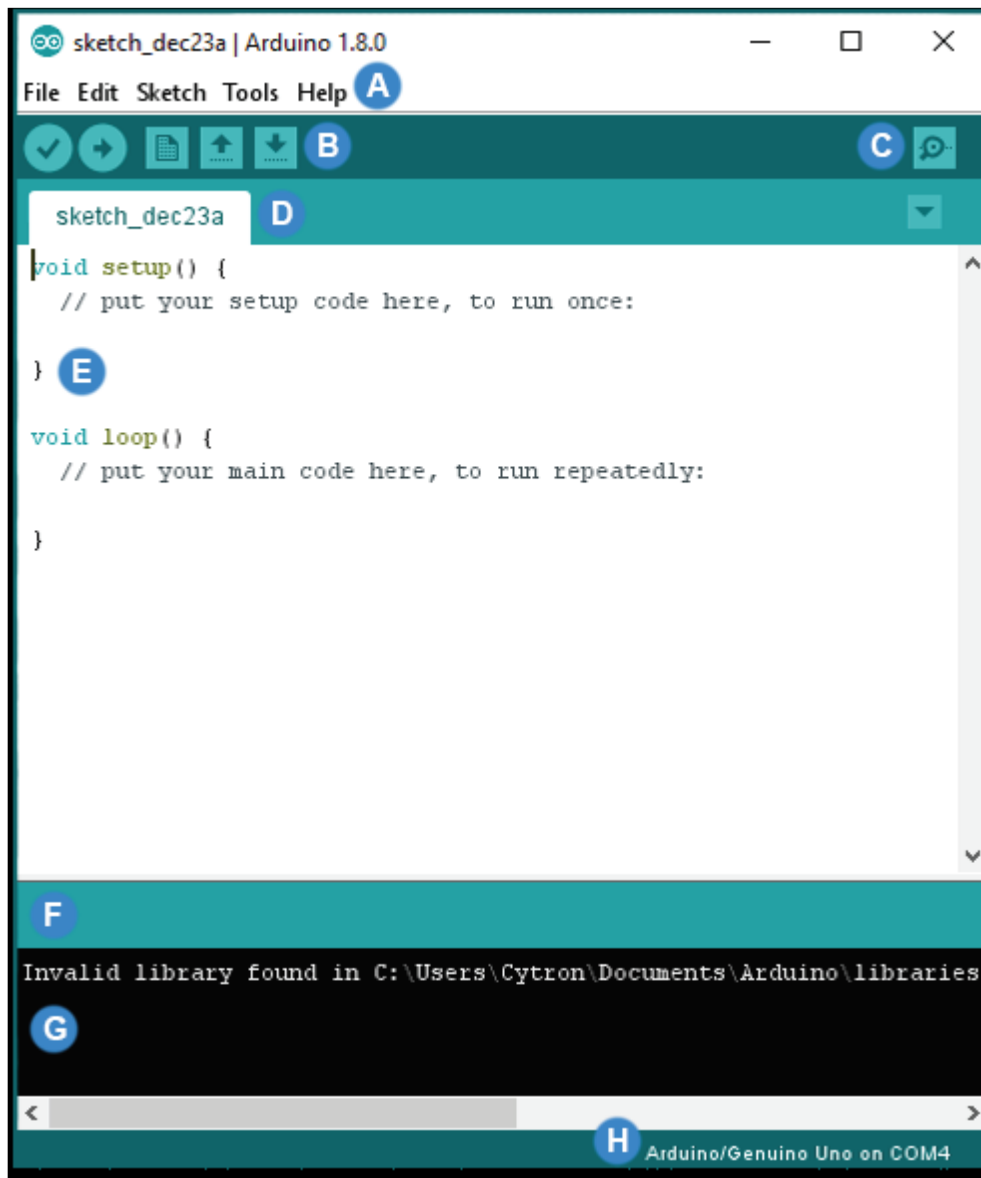
You will have arduino-1.8.x-windows.exe software after finish downloading for Windows OS user while for Mac OS user, you will get a zip file of arduino-1.8.x-macosx zip file as shown below :



***Note:** For latest version of Arduino IDE, go to <https://www.arduino.cc/en/Main/Software>

Double-click on the icon to install Arduino IDE. Complete the download, proceed with the installation as usual. After _nish installing the software, you can start using it by double-click on the icon. Then, you will see this layout of Arduino IDE.





Label	Description	Label	Description
A	Menu Bar	E	Code Area
B	Button Bar	F	Status Bar
C	Serial Monitor	G	IDE Output
D	Sketch Name	H	Board Name and COM Number



Verify

Compiles and approves your code. It will detect errors in syntax (e.g. missing semi colon or parentheses).



Upload

Sends your code to the Maker UNO. When you click it, you should see the lights on your board blink rapidly.



New Sketch

This button opens up a new code window tab.



Open

This button will let you open an existing sketch.



Serial Monitor

This saves the currently active sketch.

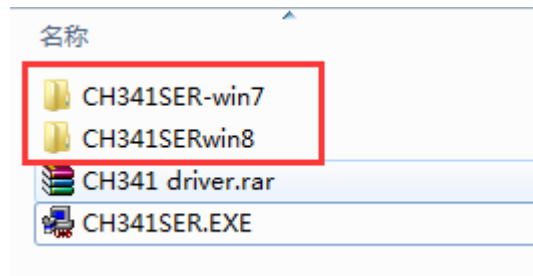
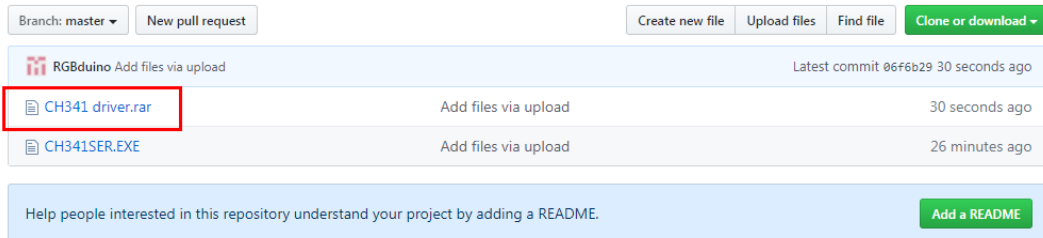


Save

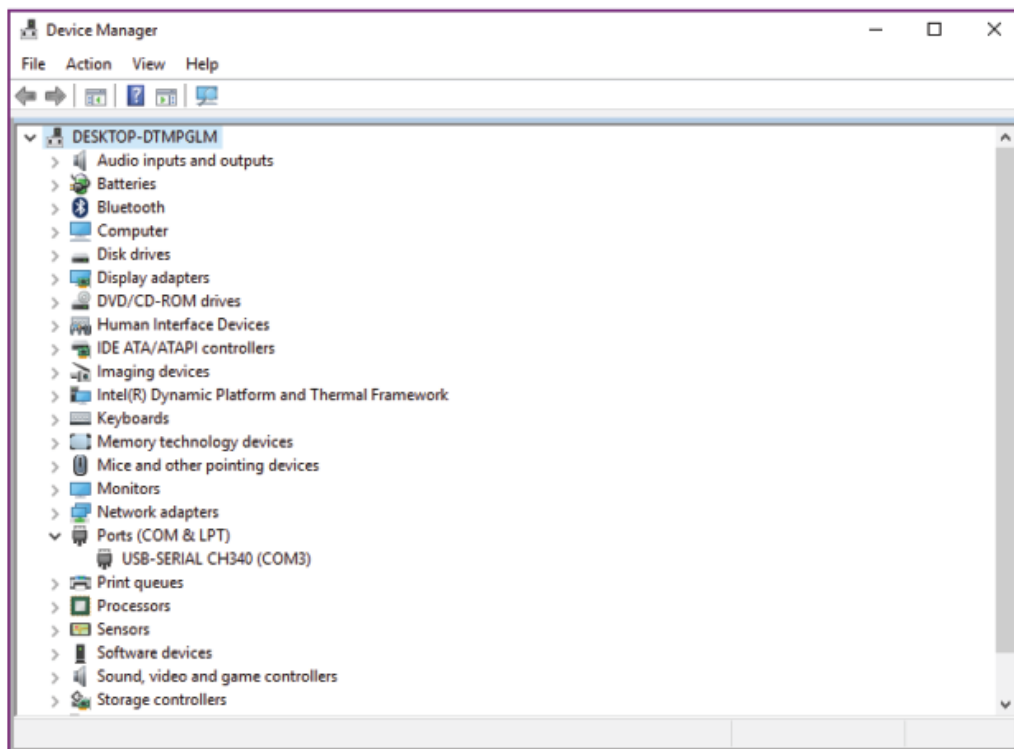
Open Serial Monitor.

Installing RGBduino UNO driver

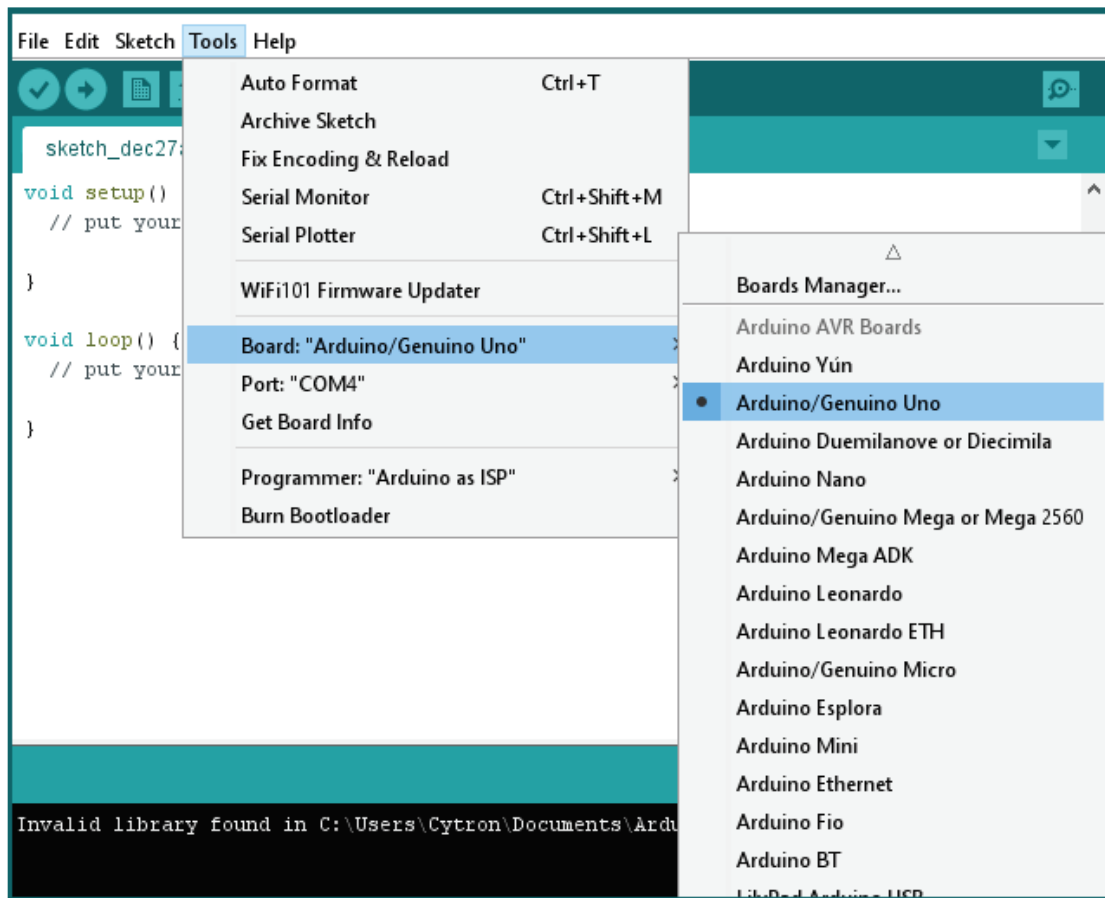
Download RGBduino UNO driver at RGBduino Uno product page (under Attachment tab).



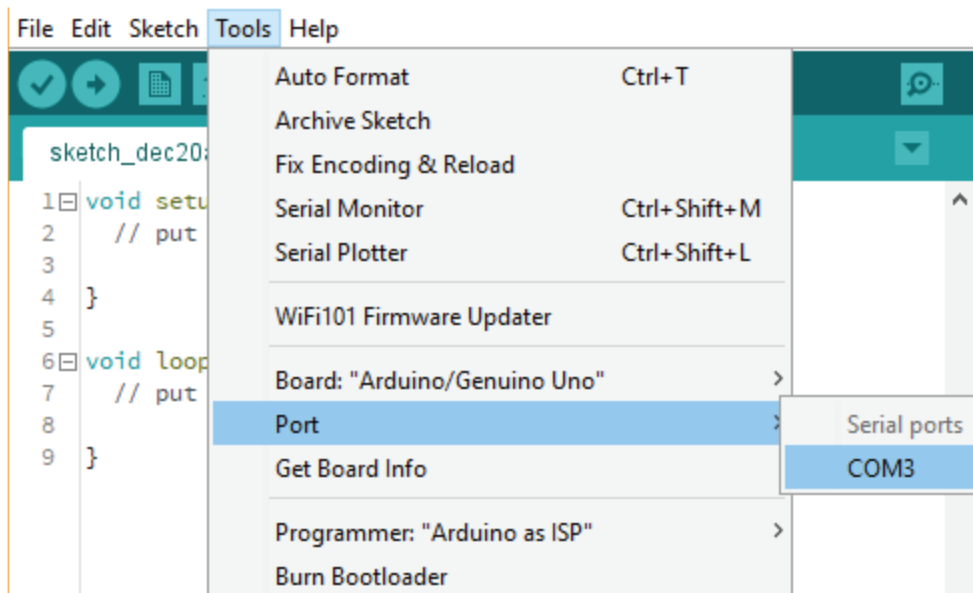
After installation is complete, your RGBduino UNO port should appear at Device Manager under Ports (COM & LPT) - e.g. USB-SERIAL CH340 (COM3). Please remember the port number.



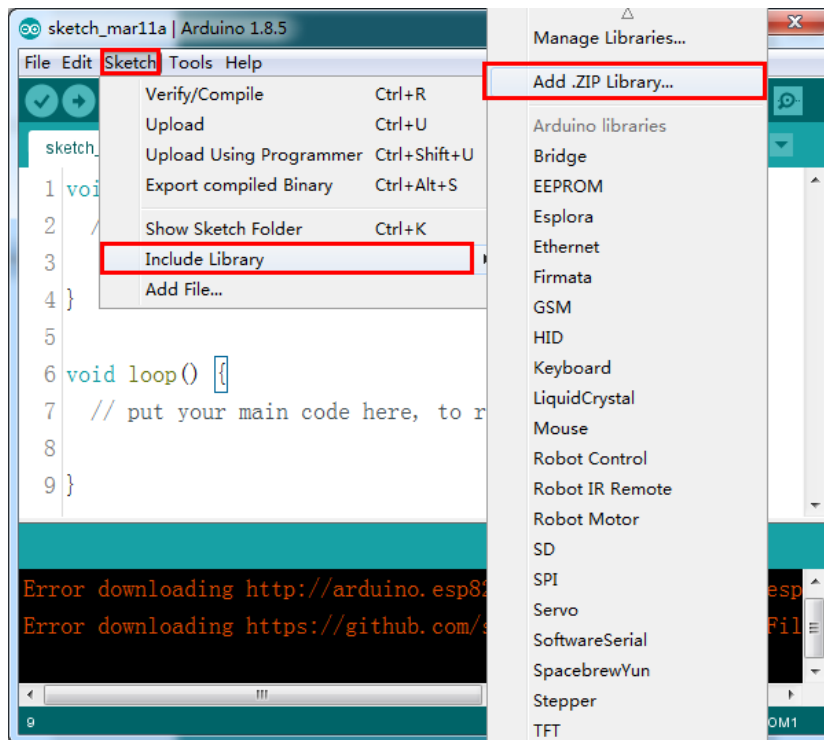
Select Board :



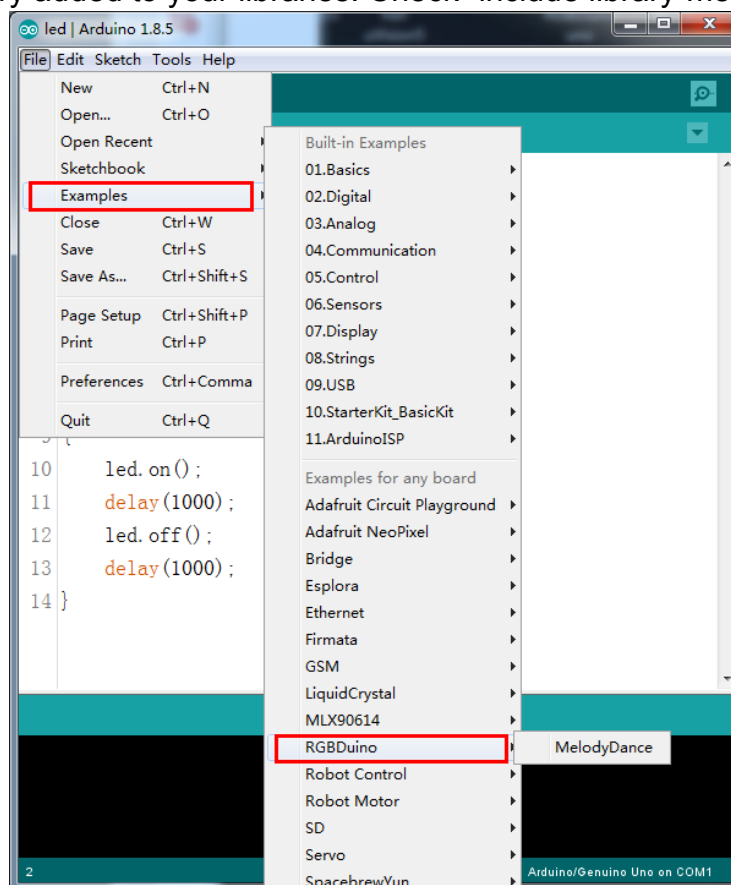
Select Serial Port :



Install RGBduino UNO library



The library added to your libraries. Check "include library" menu



LESSON 1 :

THE LED (DIGITAL OUTPUT)



LESSON 1: LIGHT UP THE LED (IDE)

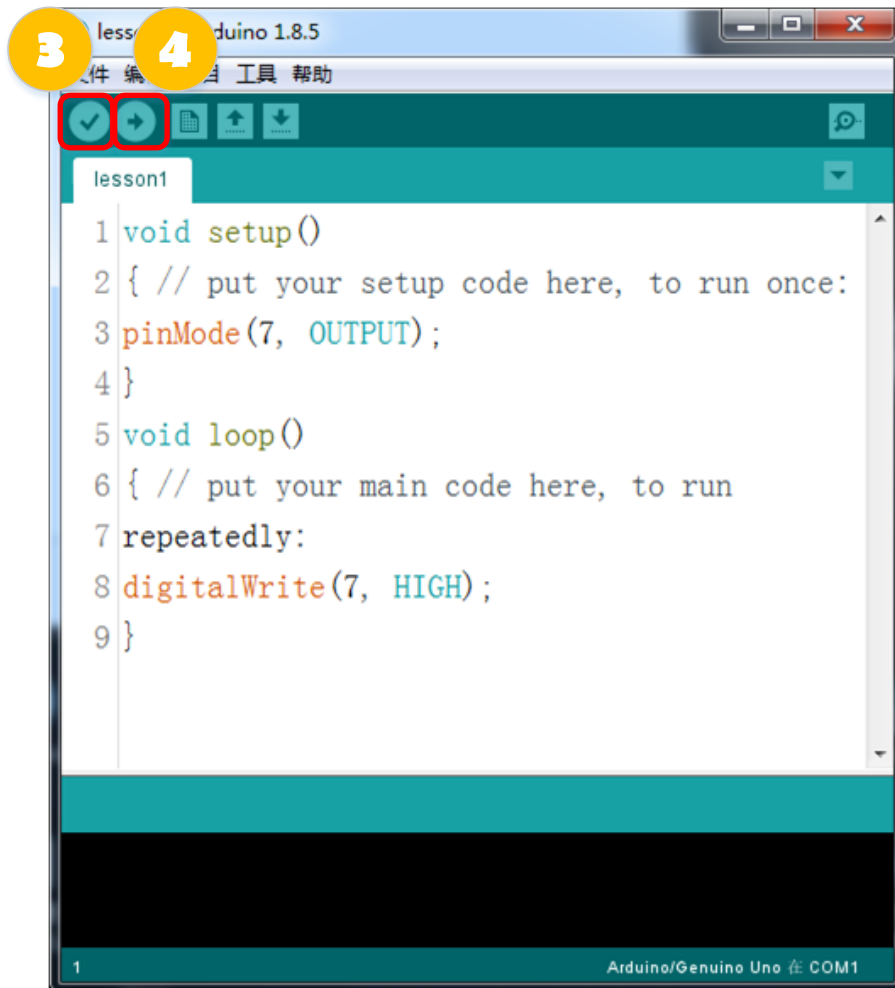
```
lesson1 | Arduino 1.8.5  
文件 编辑 项目 工具 帮助  
lesson1  
1 void setup()  
2 { // put your setup code here, to run once:  
3 pinMode(7, OUTPUT);  
4 }  
5 void loop()  
6 { // put your main code here, to run  
7 repeatedly:  
8 digitalWrite(7, HIGH);  
9 }  
1  
Arduino/Genuino Uno 在 COM1
```

i LED is a light emitting diode. It will light up when a proper voltage is applied in correct direction.

1 Open new sketch on Arduino IDE.

2 Write this code to your sketch:

```
void setup()  
{ // put your setup code here, to run once:  
  pinMode(7, OUTPUT);  
}  
void loop()  
{ // put your main code here, to run  
  repeatedly:  
  digitalWrite(7, HIGH);  
}
```



3 Compile the file.

4 Upload the sketch.

5 You will see status of “Done Uploading” if everything is correct your LED at pin 7 will light

The void setup() runs once when the Maker UNO is powered on. The code in the void setup() usually use to con_gure the pin as INPUT or OUTPUT using pinMode();

The void loop() runs continuously after the voidsetup() has complete. The code in the void loop() usually use to control the INPUT and OUTPUT.The digitalWrite(); is used to set the digital OUTPUT of the pin number to HIGH or LOW.

LESSON 2:

LED (BLINKING)



LESSON 2 : LIGHT UP THE LED (IDE)

```
1 void setup()  
2 {  
3 pinMode(7, OUTPUT);  
4 }  
5 void loop()  
6 {  
7 digitalWrite(7, HIGH);  
8 delay(1000);  
9 digitalWrite(7, LOW);  
10 }
```

保存完成。

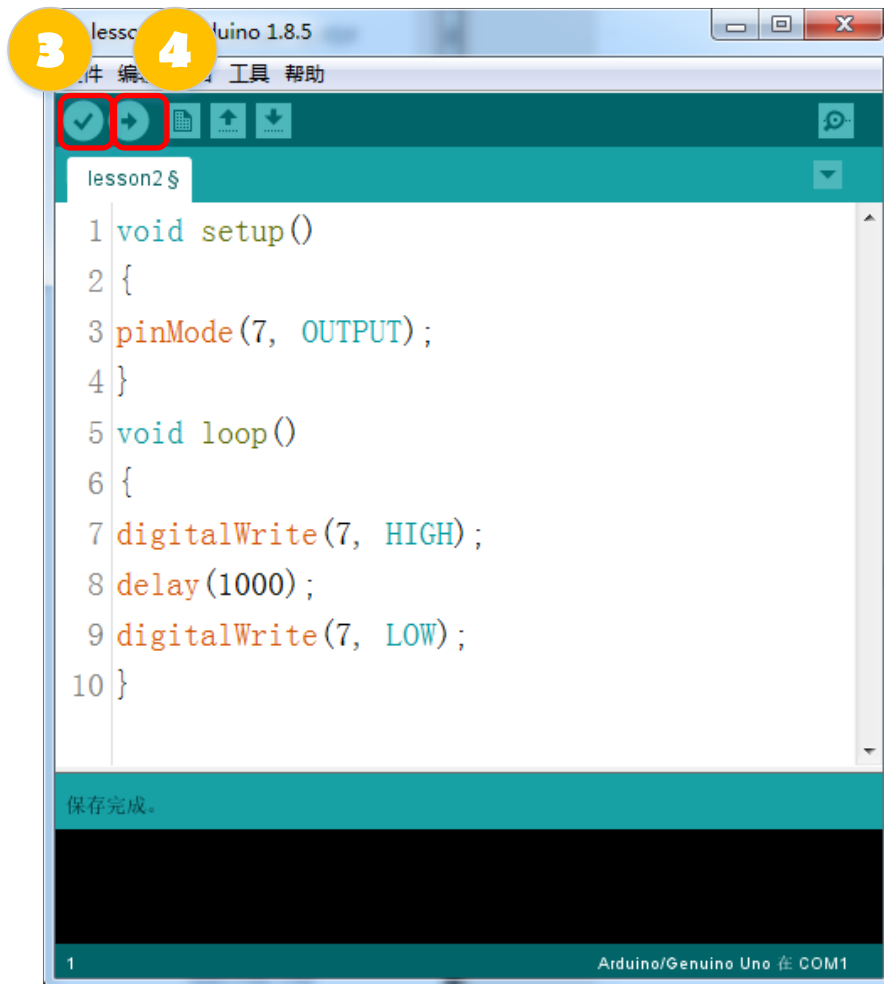
1 Arduino/Genuino Uno 在 COM1

i LED will blink when delay is applied between ON and OFF. Then it will blinking!

1 Open new sketch on Arduino IDE.

2 Write this code to your sketch :

```
void setup()  
{  
pinMode(7, OUTPUT);  
}  
void loop()  
{  
digitalWrite(7, HIGH);  
delay(1000);  
digitalWrite(7, LOW);  
delay(1000);  
}
```



3 Compile the file.

4 Upload the sketch.

5 You will see status of “Done Uploading” if everything is correct your LED at pin 7 will light

The void setup() runs once when the Maker UNO is powered on. The code in the void setup() usually use to configure the pin as INPUT or OUTPUT using pinMode();
The void loop() runs continuously after the void setup() has complete. The code in the void loop() usually use to control the INPUT and OUTPUT. The digitalWrite(); is used to set the digital OUTPUT of the pin number to HIGH or LOW.

LESSON 3:

FADE AN LED



LESSON 3 : FADE AN LED

2



```
1 Write this code to your sketch :
2 int LED = 3;
3 int brightness = 0;
4 int fadeAmount = 5;
5 void setup()
6 {
7   pinMode(3, OUTPUT);
8 }
9 void loop()
10 {
11   analogWrite(LED, brightness);
12   brightness = brightness + fadeAmount;
13   if (brightness <= 0 || brightness >= 255)
14   {
15     fadeAmount = -fadeAmount;
16   }
17   delay(30);

```

保存完成。

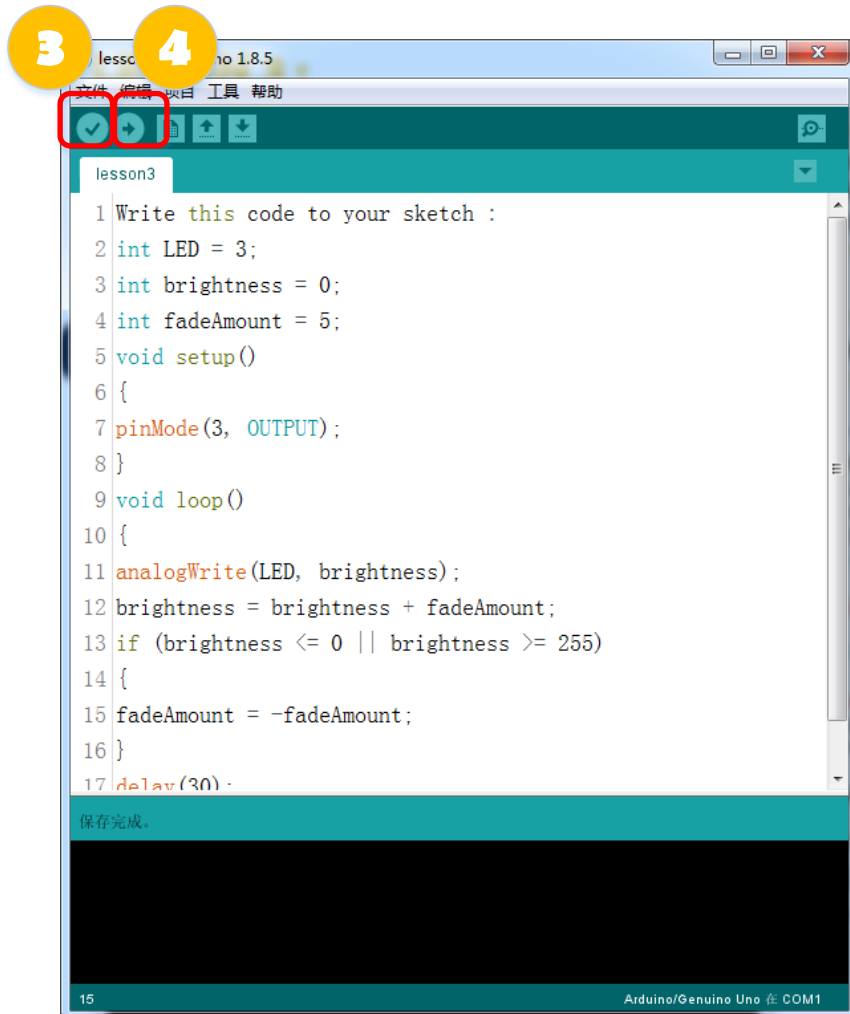
15 Arduino/Genuino Uno 在 COM1

i The LED will fade using `analogWrite()` function using Pulse Width Modulation (PWM) which make a digital output acting as analog output.

1 Open new sketch on Arduino IDE.

2 Write this code to your sketch :

```
int LED = 3;
int brightness = 0;
int fadeAmount = 5;
void setup()
{
  pinMode(3, OUTPUT);
}
void loop()
{
  analogWrite(LED, brightness);
  brightness = brightness + fadeAmount;
  if (brightness <= 0 || brightness >= 255)
  {
    fadeAmount = -fadeAmount;
  }
  delay(30);
}
```



3 Compile the file.

4 Upload the sketch.

5 You will see status of “Done Uploading” if everything is correct your LED at pin 7 will light

The analogWrite() function uses PWM, so if you want to change the pin you're using, be sure to use another PWM capable pin. On most Arduino, the PWM pins are identified with a "~" sign, like ~3, ~5, ~6, ~9, ~10 and ~11. The analogWrite(LED, brightness); set OUTPUT of the pin number 3 to variable “brightness”. The LED will light up based on the amount of variable “brightness”.

LESSON 4:

PUSH BTTON (DIGITAL INPUT)



LESSON 4 : PUSH BUTTON



```
lesson4 | Arduino 1.8.5
文件 编辑 项目 工具 帮助
lesson4
1
2 int LED = 4;
3 int Button = 2;
4 void setup()
5 {
6 pinMode(4, OUTPUT);
7 pinMode(2, INPUT_PULLUP);
8 }
9 void loop()
10 {
11 if (digitalRead(Button) == LOW)
12 digitalWrite(LED, HIGH);
13 else if (digitalRead(Button) == HIGH)
14 digitalWrite(LED, LOW);
15 }
编译完成。
项目使用了 944 字节, 占用了 (2%) 程序存储空间。最大为 322
全局变量使用了9字节, (0%) 的动态内存, 余留2039字节局部变量
14 Arduino/Genuino Uno 在 COM1
```

i Push button act as a digital input device. Maker UNO is able to sense 2 states for digital input, i.e. HIGH and LOW. Push the button and the LED will turn ON!

1 Open new sketch on Arduino IDE.

2 Write this code to your sketch :

```
int LED = 4;
int Button = 2;
void setup()
{
pinMode(4, OUTPUT);
pinMode(2, INPUT_PULLUP);
}
void loop()
{
if (digitalRead(Button) == LOW)
digitalWrite(LED, HIGH);
else if (digitalRead(Button) == HIGH)
digitalWrite(LED, LOW);
}
```

```
1 int LED = 4;
2 int Button = 2;
3 void setup()
4 {
5   pinMode(4, OUTPUT);
6   pinMode(2, INPUT_PULLUP);
7 }
8 void loop()
9 {
10  if (digitalRead(Button) == LOW)
11    digitalWrite(LED, HIGH);
12  else if (digitalRead(Button) == HIGH)
13    digitalWrite(LED, LOW);
14 }
15 }
```

编译完成。

项目使用了 944 字节, 占用了 (2%) 程序存储空间。最大为 322

全局变量使用了9字节, (0%) 的动态内存, 余留2039字节局部变量

14 Arduino/Genuino Uno 在 COM1

3 Compile the file.

4 Upload the sketch.

5 You will see status of “Done Uploading” if everything is correct, when button is pressed, the LED pin 4 will light up.

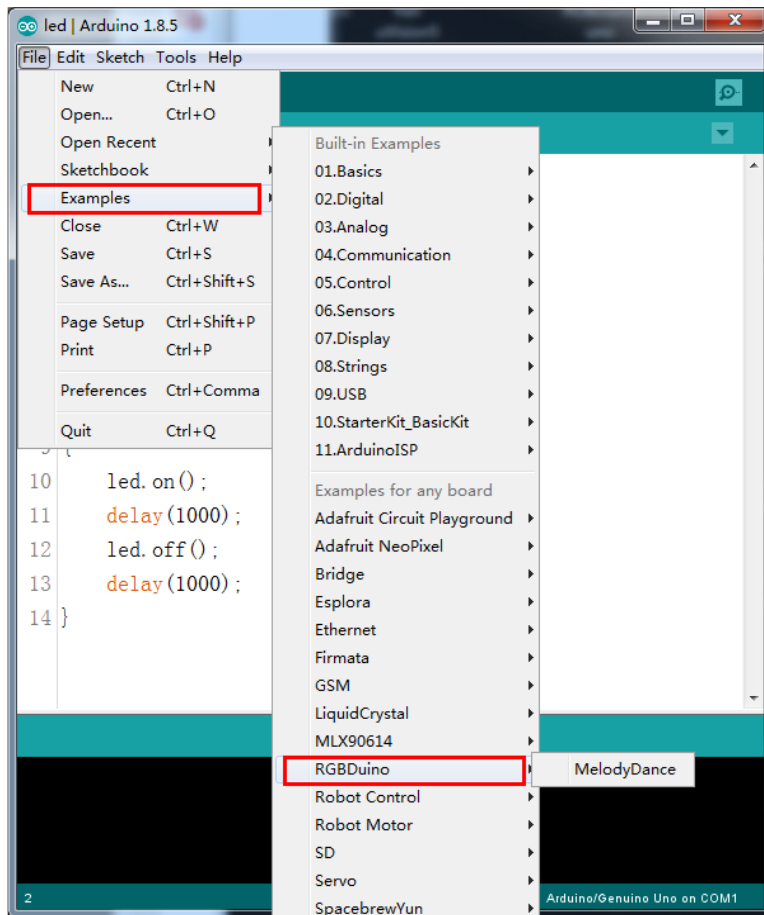
Using `pinMode(INPUT_PULLUP)`, there is an internal 20K-ohm resistor is pulled to 5V. This configuration causes the input to read HIGH when the switch is open, and LOW when it is closed. The `if()` statement is use to compare a condition whether it is TRUE or FALSE. The `else if()` statement is use to set other condition than `if()` statement. The `digitalRead(Button) == LOW`; will read the button input. If the button is pushed, the INPUT will be LOW.

LESSON 5:

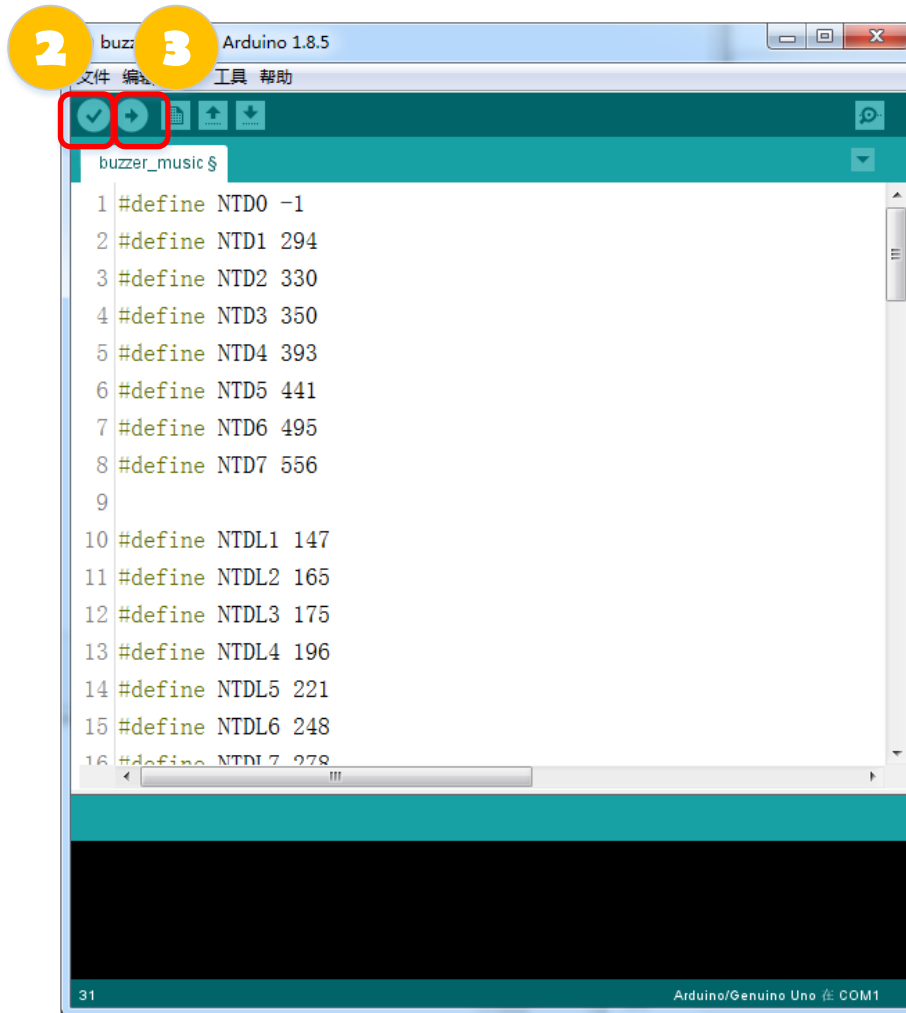
MELODY DANCE



LESSON 5 : MELODY DANCE



1 Go to File > Examples > RGBDuino > MelodyDance



2 Compile the file.

3 Upload the sketch.

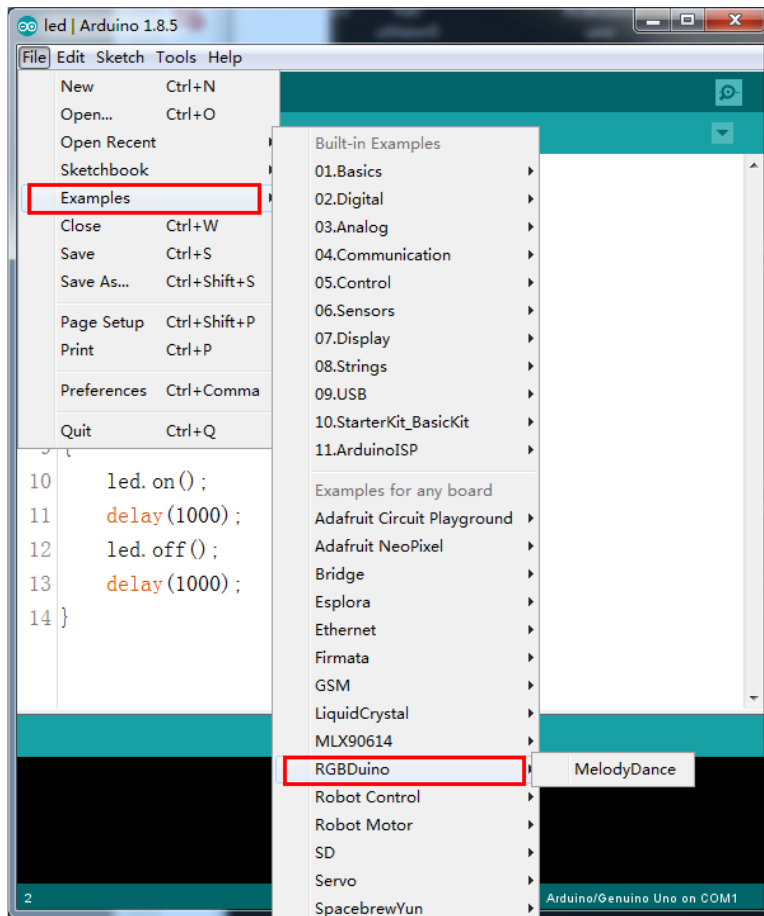
4 You can change the music note based on your preference and enjoy the music tone.

LESSON 6:

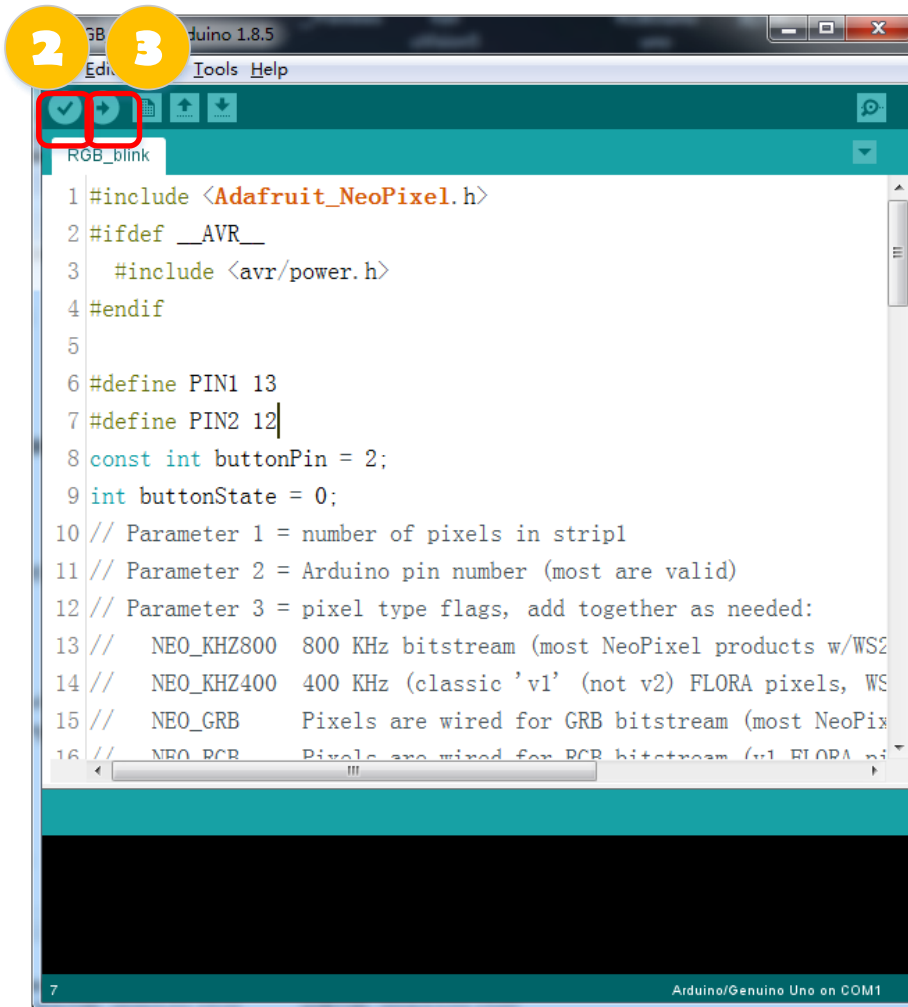
RGB BLINK



LESSON 6 : RGB BIINK



1 Go to File > Examples > RGBDuino > RGBBLINK



2 Compile the file.

3 Upload the sketch.

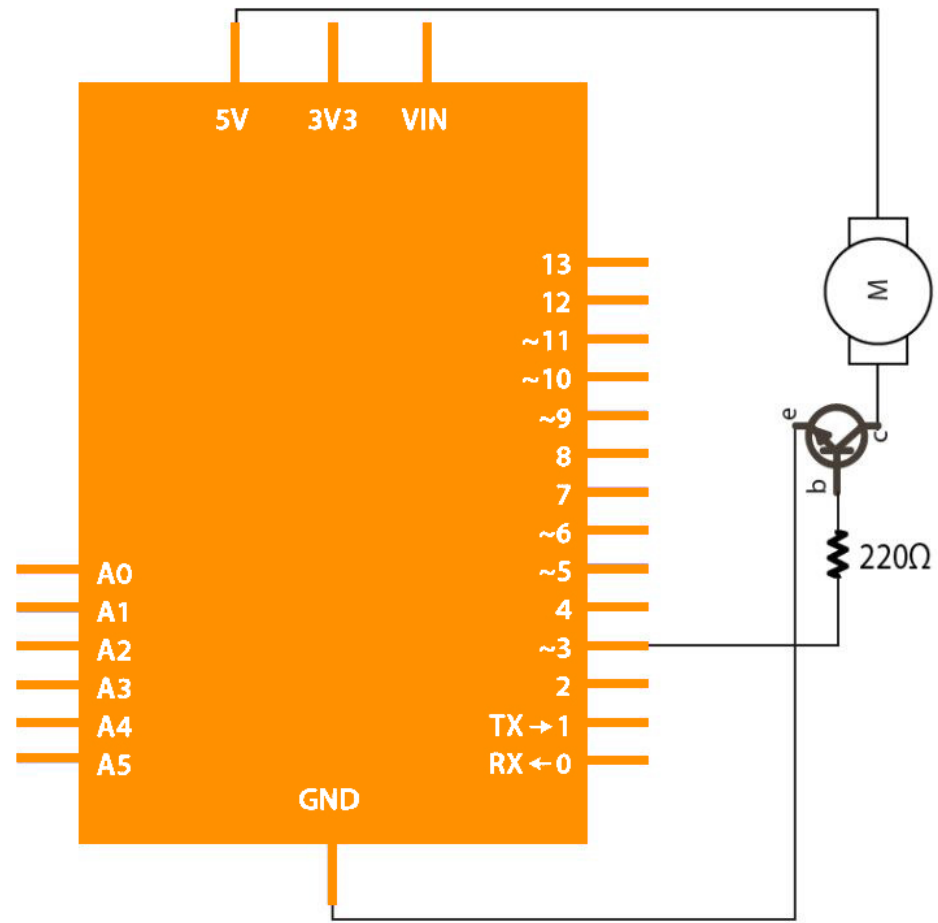
4 You will see two RGB lights changing colors flashing

LESSON 7:

CONTROLLING MOTOR



LESSON 7 : SCHEMATIC



LESSON 7 : CONTROLLING MOTOR



```
1 void setup()
2 {
3   pinMode(6, OUTPUT);
4 }
5 void loop()
6 {
7   analogWrite(6, 255); //same with HIGH
8   delay(1000);
9   analogWrite(6, 123);
10  delay(1000);
11  analogWrite(6, 50);
12  delay(1000);
13  analogWrite(6, LOW); //same with 0
14  delay(1000);
15 }
```

Done compiling

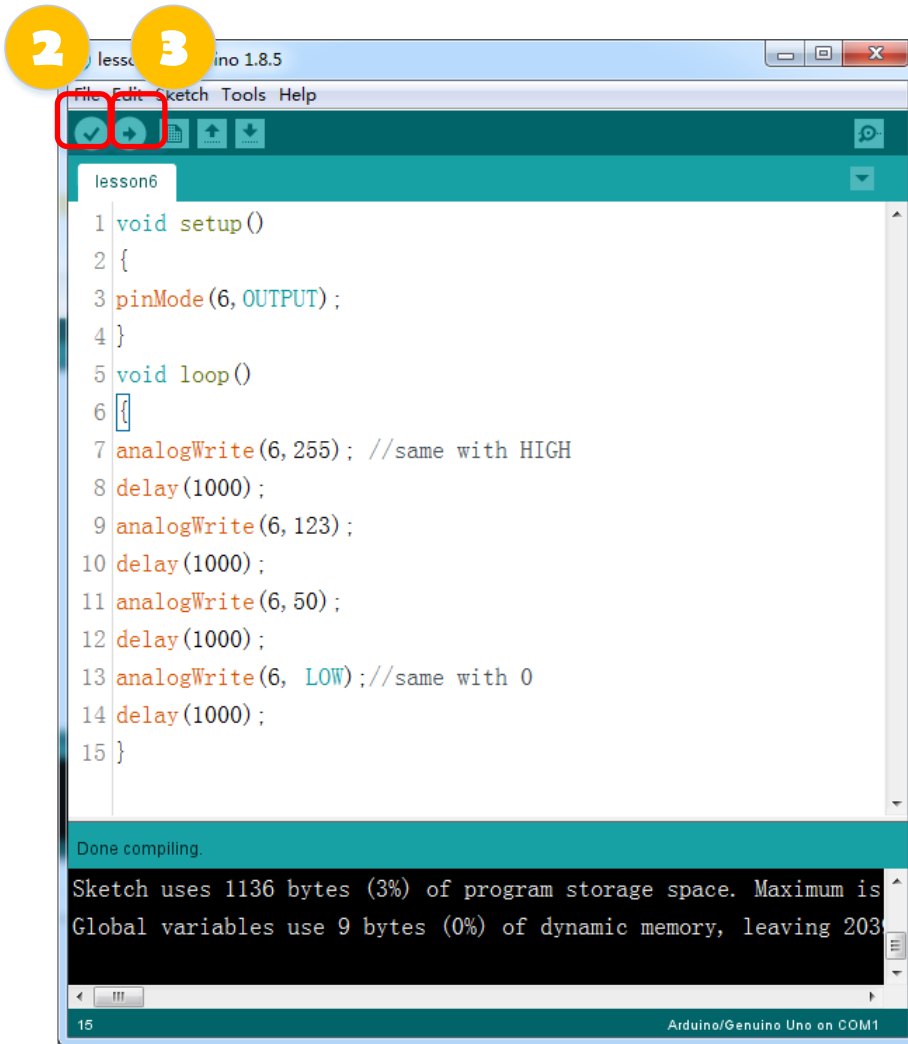
Sketch uses 1136 bytes (3%) of program storage space. Maximum is
Global variables use 9 bytes (0%) of dynamic memory, leaving 203

15 Arduino/Genuino Uno on COM1

1 Open new sketch on Arduino IDE.

2 Write this code to your sketch :

```
void setup()
{
  pinMode(6, OUTPUT);
}
void loop()
{
  analogWrite(6, 255); //same with HIGH
  delay(1000);
  analogWrite(6, 123);
  delay(1000);
  analogWrite(6, 50);
  delay(1000);
  analogWrite(6, LOW); //same with 0
  delay(1000);
}
```



2 Compile the file.

3 Upload the sketch.

4 The motor will rotate with 4 different speed.



www.RGBDuino.com

Email:

RGBDuino@163.com