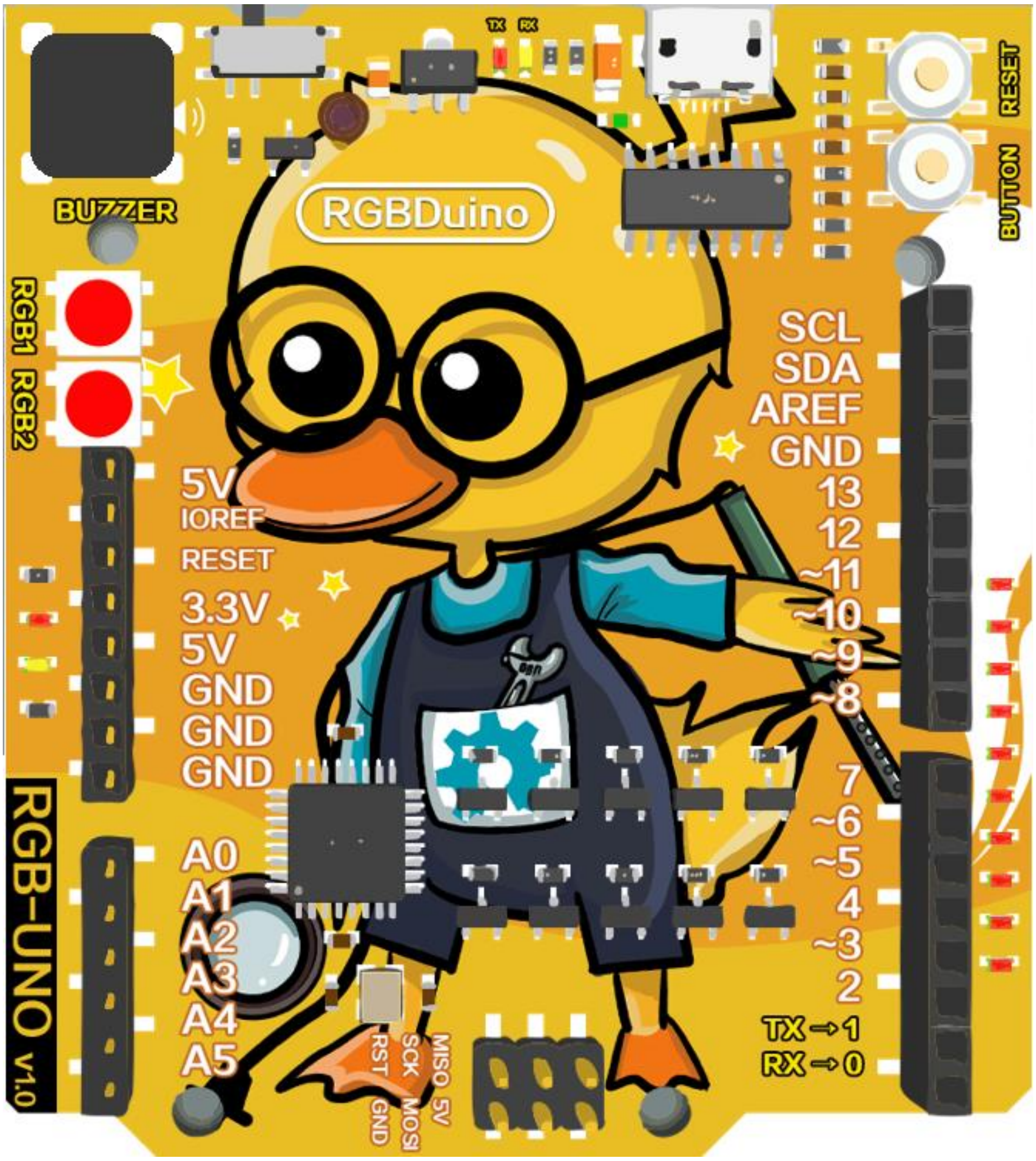
Start learning
RGBDuino UNO



Start marking something!

CONTENTS

INTRODUCTION

IntronductiontoComponents

- -RGBDuino Uno
- - RGBDuino Uno Board

Setting up

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

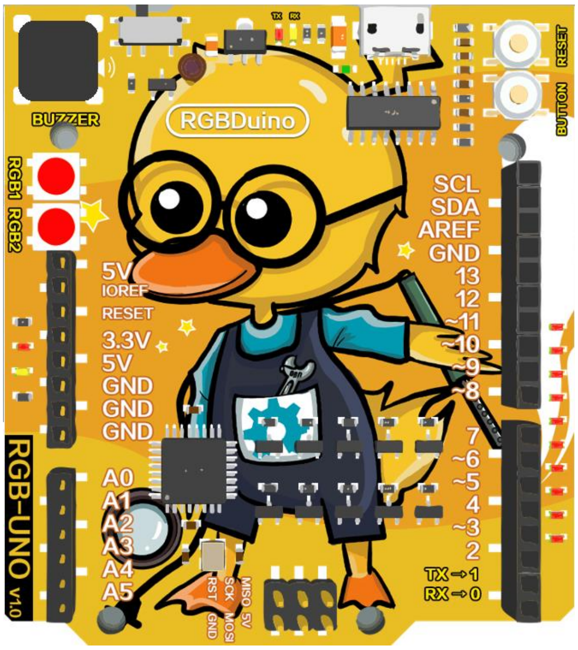
LESSONS

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.

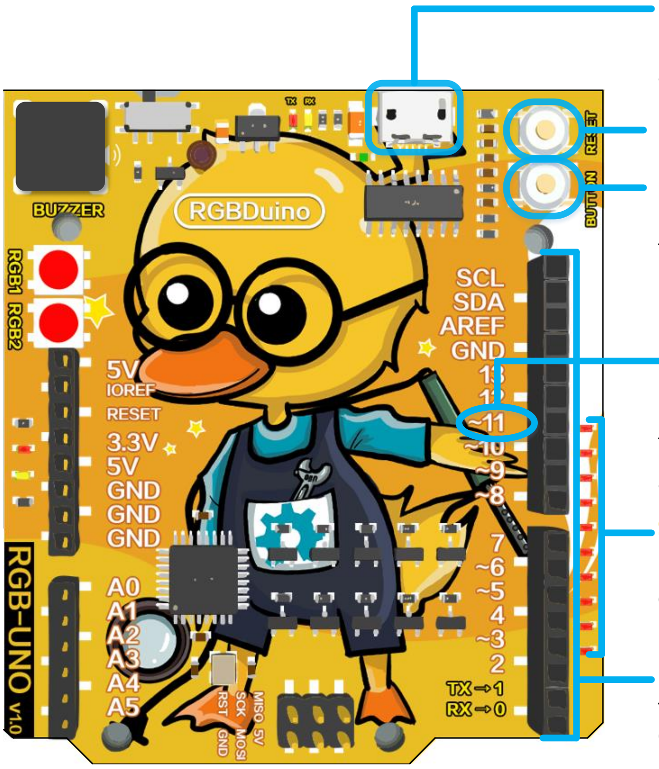
-10 blue LEDs and two RGB programmable lamps

-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!



Micro USB Connector

Main supply for RGBDuino Uno,Used for program and debug purpose too.

Reset Button

Button to restart RGBDuino Uno program.

Programmable Button

This button is connected to Pin 2 and GND. To use it ,user need to configure it as INPUT_PULLUP

PWM Pin ~

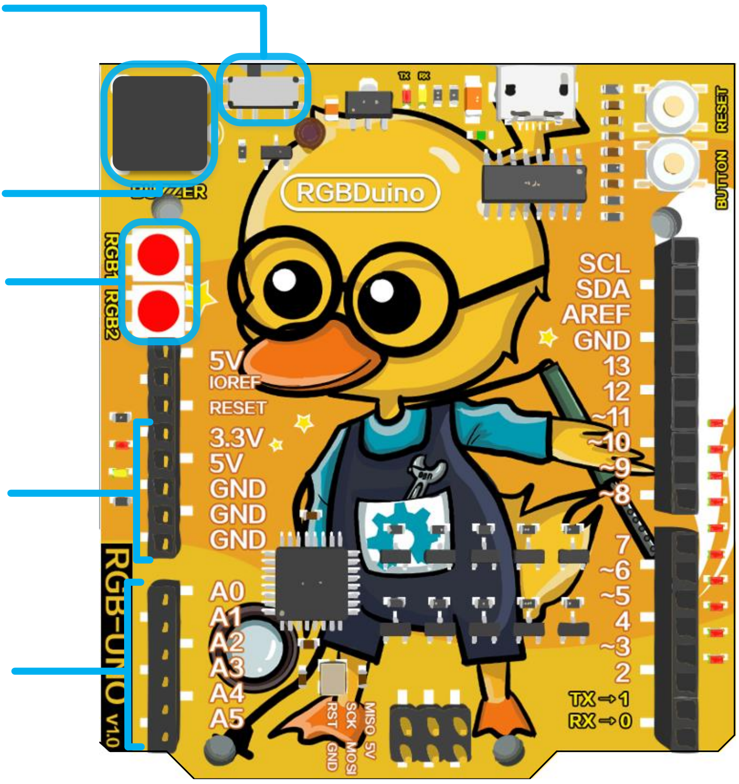
The digital pin that has this symbol can only use analogWrite(); to control the output.

Series of LED for Digital I/O

Every digital IO is equipped with LED. where you can control it as indicator for input.

Digital Pin

This pin can be used with :digitalRead();as an input
digitalWrite();as an output.



Piezo Buzzer Slide Swithc

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

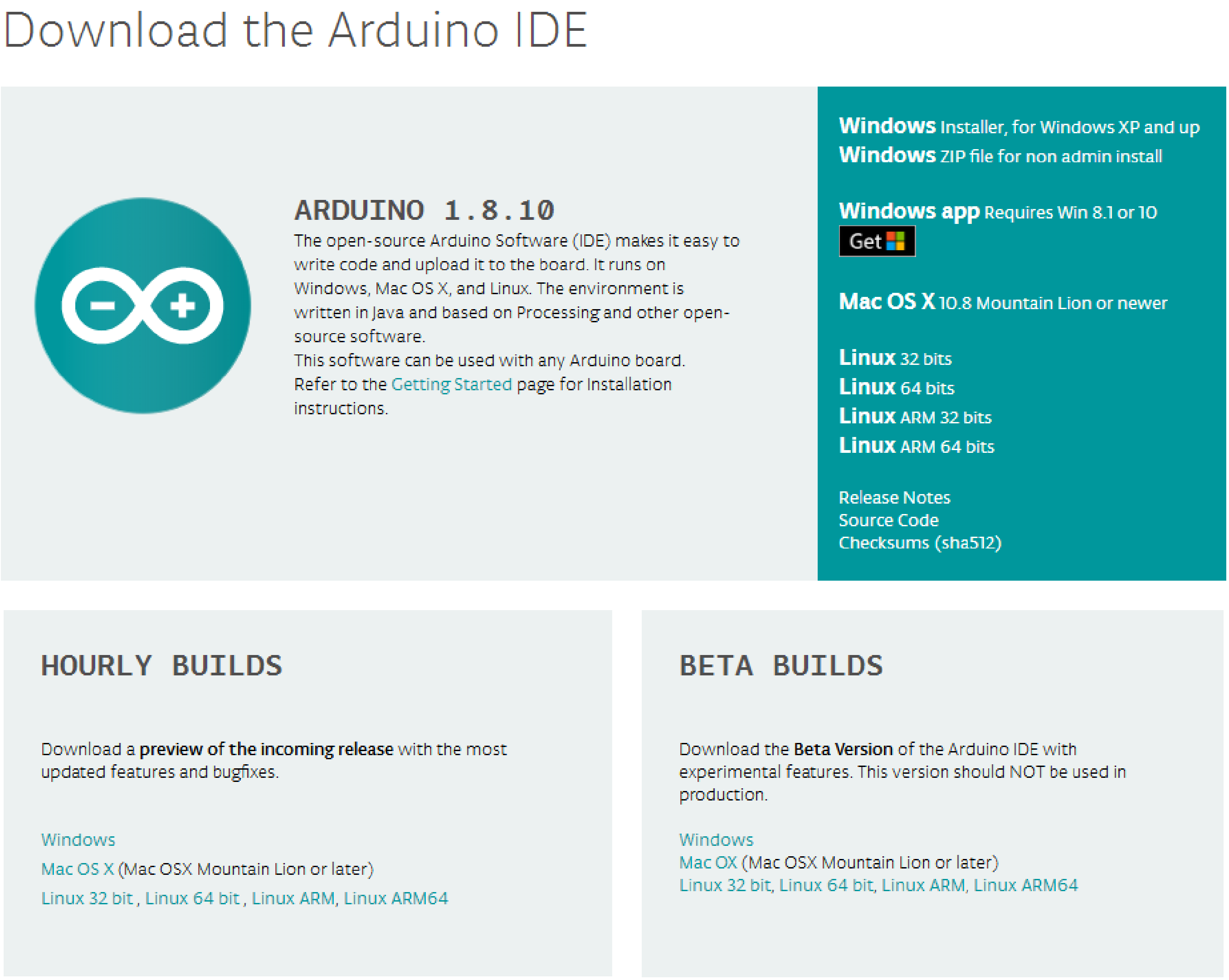
GND-Ground Pins
5V-Regulated 5V output
3V3-Regualted 3.3v supply

Analog Pin

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

DOWLOAD Aruino 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
Get

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 OX (Mac OSX Mountain Lion or later)
Linux 32 bit, Linux 64 bit, Linux ARM, Linux ARM64

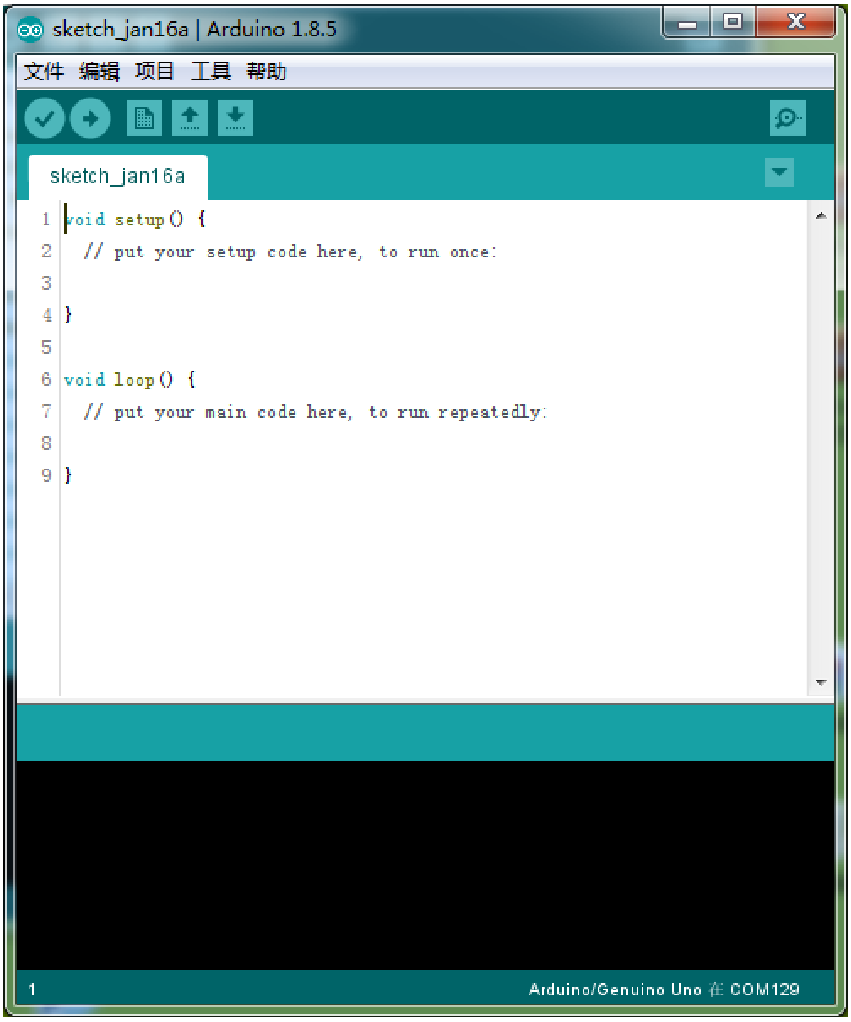
Arduino IDE is compatible with Windows, Mac OS X and also Linux. You just need to choose the appropriate operating system installation package for your computer. If you are a Windows user, it is recommended that you choose Windows (installer).

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

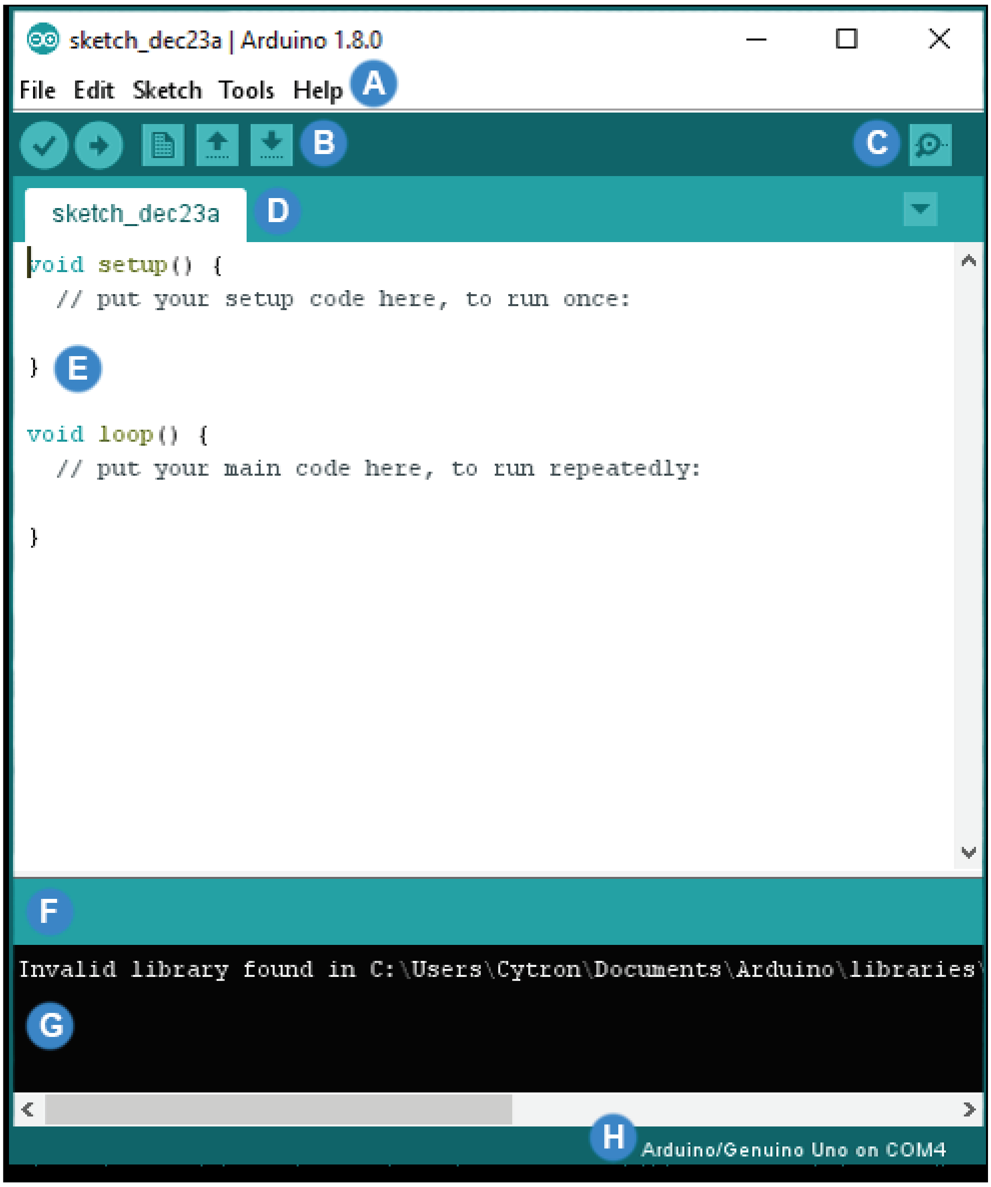
Arduino IDE.
You will have arduino-1.8.x-windows.exe software after finish downloading for Windows OS user while for Mac OS user, you will get a zip file of arduino-1.8.x-macosx zip file as shown below ：

*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.



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

}

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

}
```



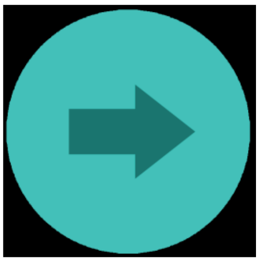
```
void setup() {
  // put your setup code here, to run once:

}

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

}
```

```
Invalid library found in C:\Users\Cytron\Documents\Arduino\libraries\
```

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

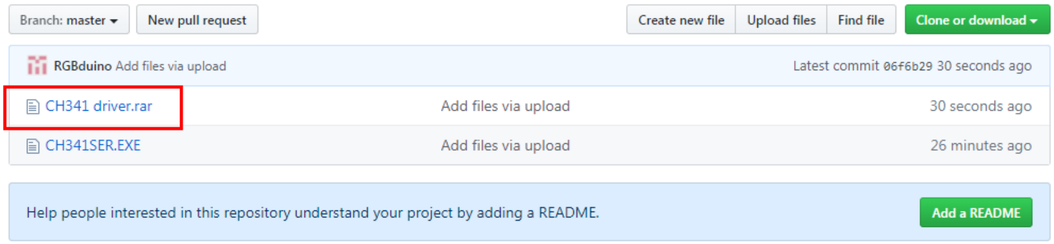
This saves the currently active sketch.

Save

Open Serial Monitor.

Installing RGBDuino UNO driver

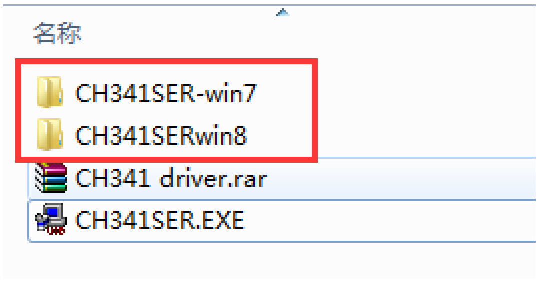
Download RGBDuino UNO driver at RGBDuino Uno product page (under Attachment tab).



RGBduino Add files via upload — Latest commit 06f6b29 30 seconds ago

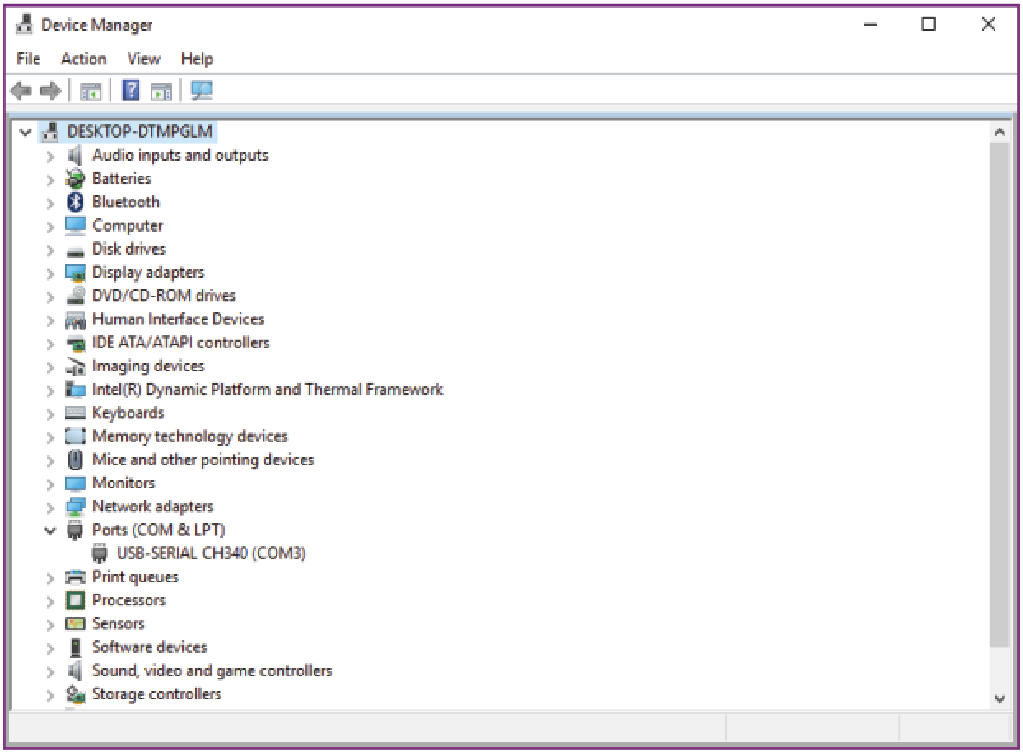
CH341 driver.rar	Add files via upload	30 seconds ago
CH341SER.EXE	Add files via upload	26 minutes ago

Help people interested in this repository understand your project by adding a README.



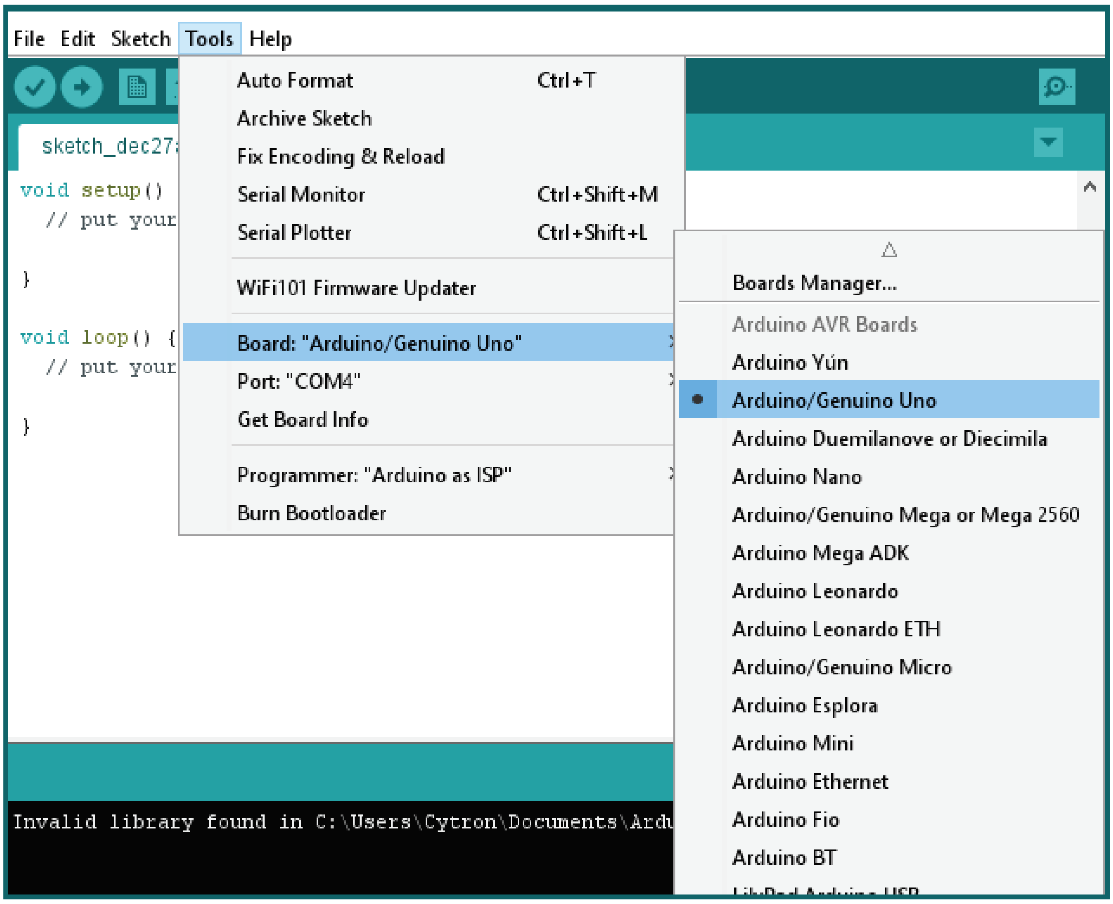
CH341SER-win7
CH341SERwin8
CH341 driver.rar
CH341SER.EXE

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



- DESKTOP-DTMPGLM
  - Audio inputs and outputs
  - Batteries
  - Bluetooth
  - Computer
  - Disk drives
  - Display adapters
  - DVD/CD-ROM drives
  - Human Interface Devices
  - IDE ATA/ATAPI controllers
  - Imaging devices
  - Intel(R) Dynamic Platform and Thermal Framework
  - Keyboards
  - Memory technology devices
  - Mice and other pointing devices
  - Monitors
  - Network adapters
  - Ports (COM & LPT)
    - USB-SERIAL CH340 (COM3)
  - Print queues
  - Processors
  - Sensors
  - Software devices
  - Sound, video and game controllers
  - Storage controllers

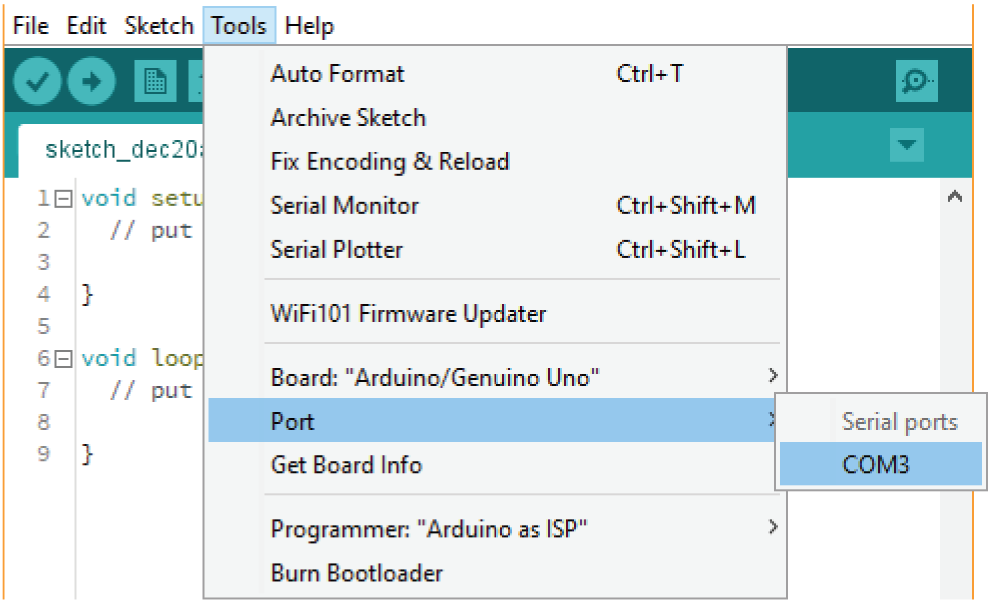
Select Board :



```
void setup()
  // put your

}

void loop() {
  // put your

}
```

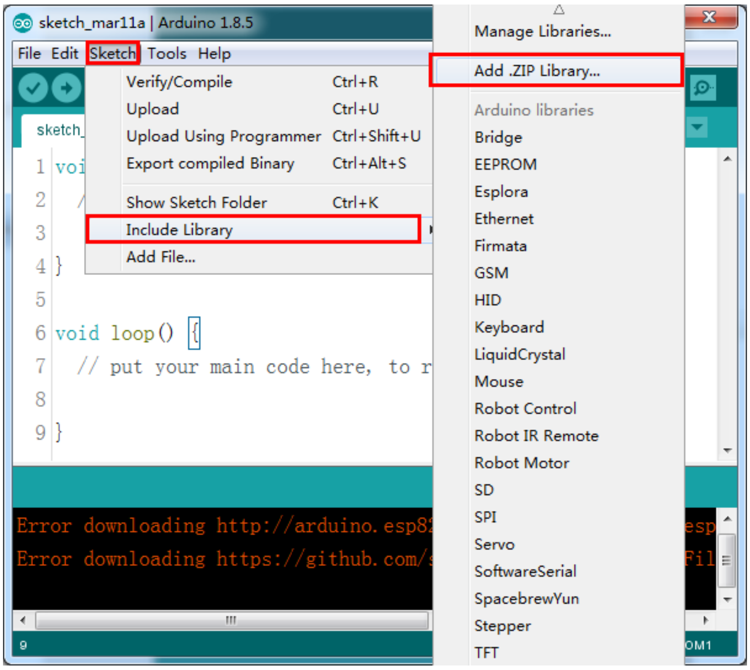
```
Invalid library found in C:\Users\Cytron\Documents\Ardu
```

Select Serial Port :



```
void setu
  // put

}

void loop
  // put

}
```

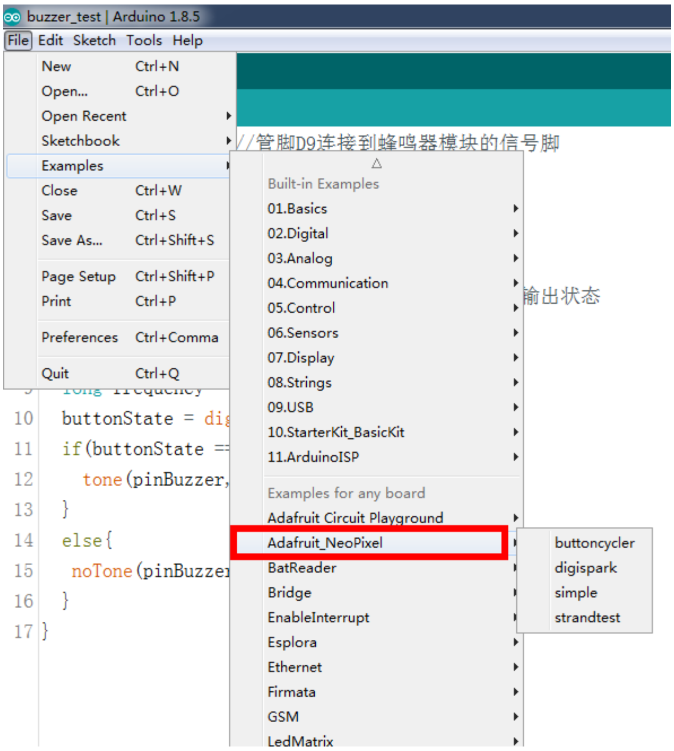
Install NeoPixelmaster library



```
void loop() {
  // put your main code here, to r
}
```

```
Error downloading http://arduino.esp8
Error downloading https://github.com/
```

The library added to your libraries. Check “include library”menu



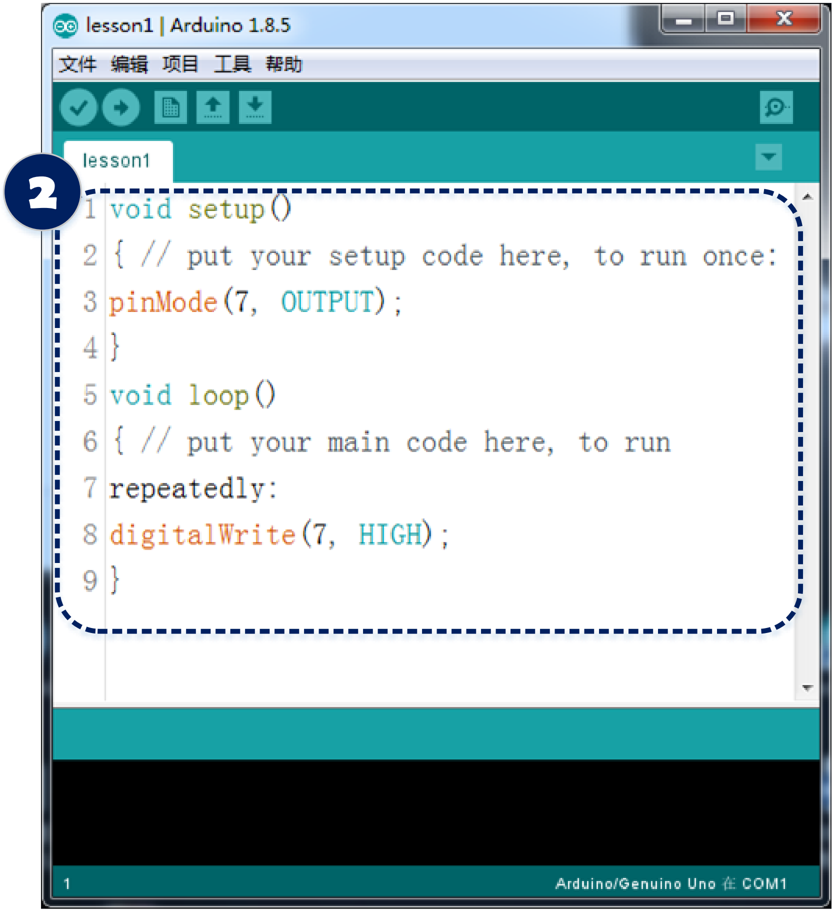
```
//管脚D9连接到蜂鸣器模块的信号脚
输出状态
long frequency
  buttonState = dig
  if(buttonState ==
    tone(pinBuzzer,
  }
  else{
   noTone(pinBuzze
  }
}
```

Then you can run the test code normally.

LESSON 1 :

THE LED (DIGITAL OUTPUT)

LESSON 1 :
LIGHT UP THE LED (IDE)



```
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);
}
```

i

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

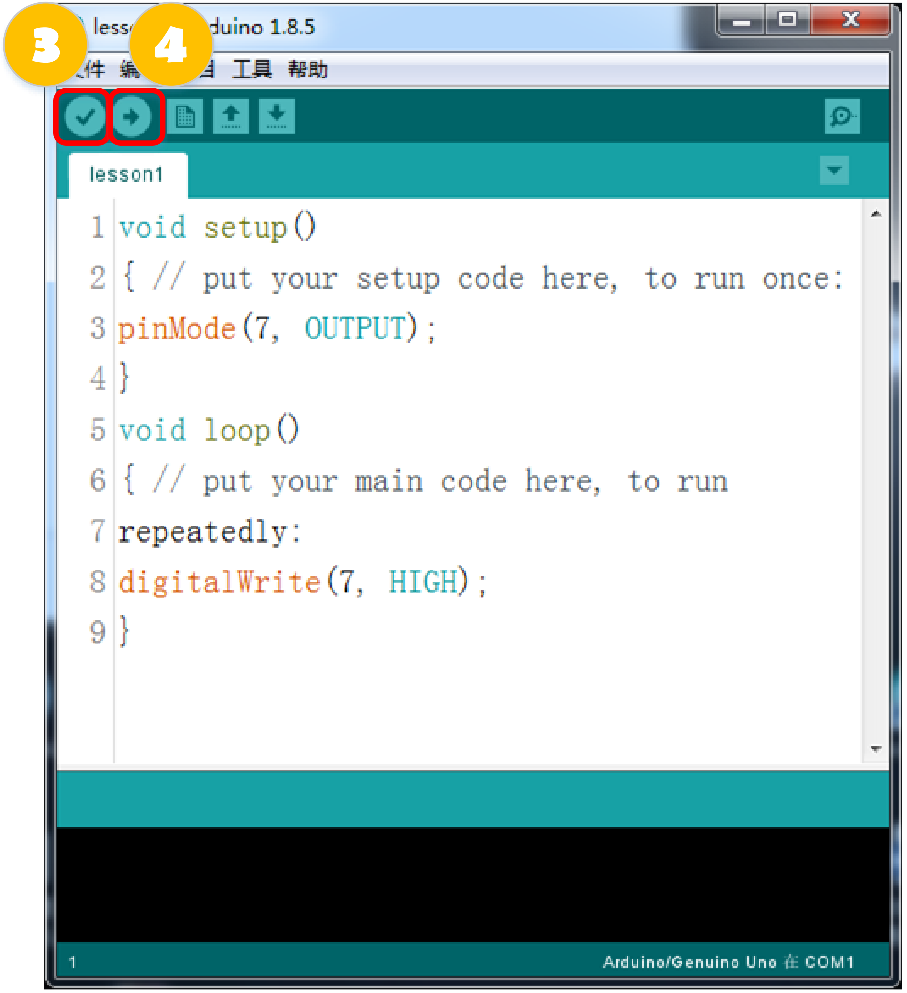
1

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);
}
```



```
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

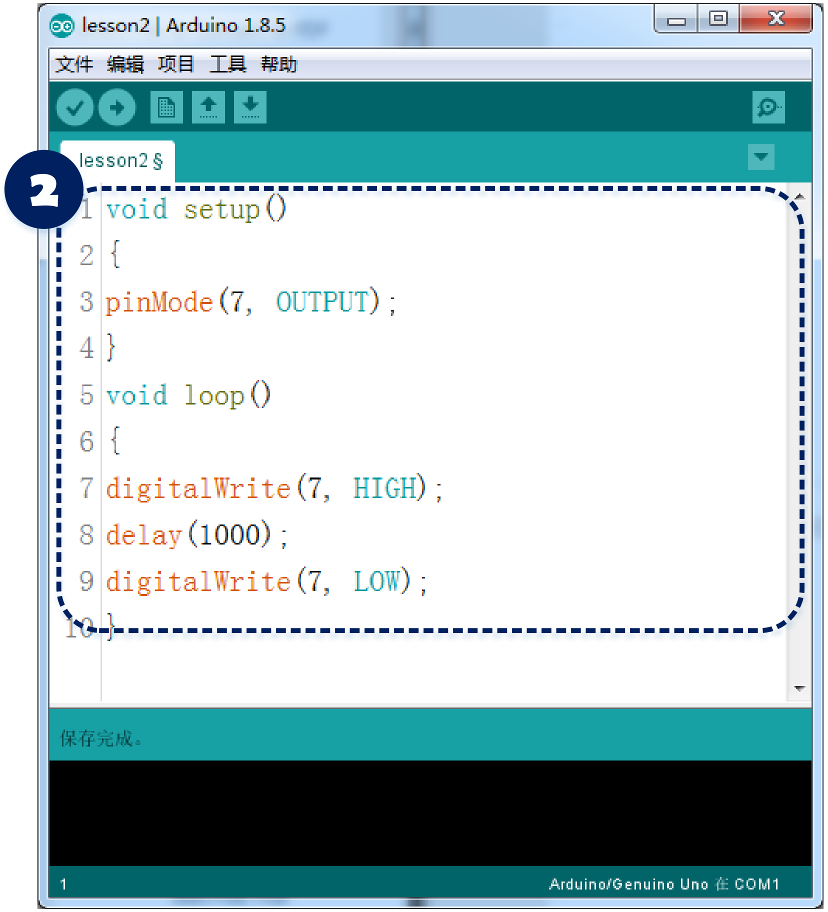
The void setup() runs once when the Maker UNO is powered on. The code in the void setup() usually use to con_gure the pin as INPUT or OUTPUT using pinMode();

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)



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

保存完成。

i

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

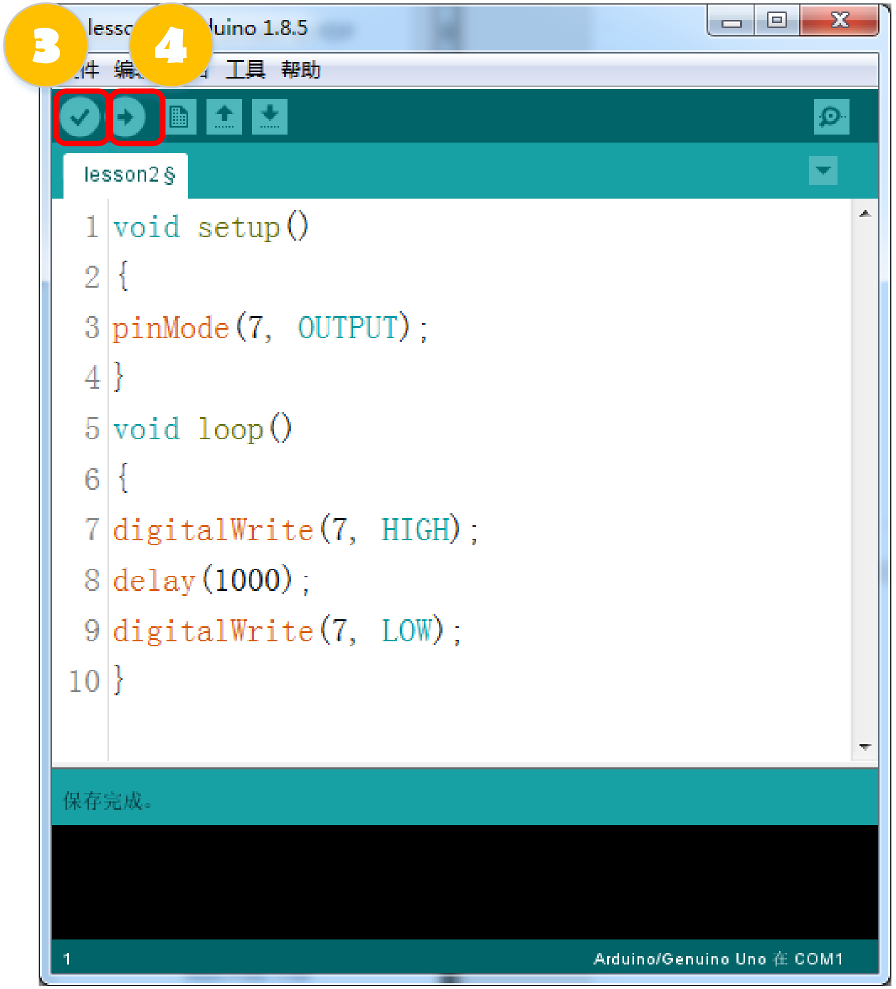
1

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);
  }
```



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

3 Compile the file.

4 Upload the sketch.

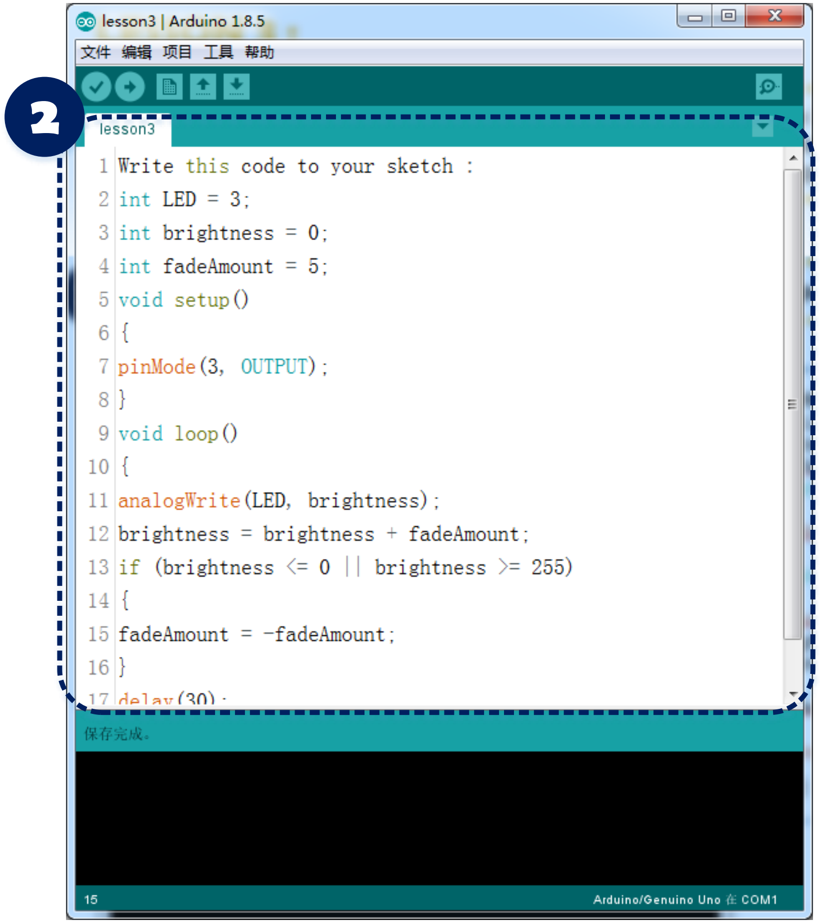
5 You will see status of "Done Uploading"if everything is correct your LED at pin 7 will light

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 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 3:

FADE AN LED

LESSON 3 :
FADE AN LED



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);
```

保存完成。

15 Arduino/Genuino Uno 在 COM1

i

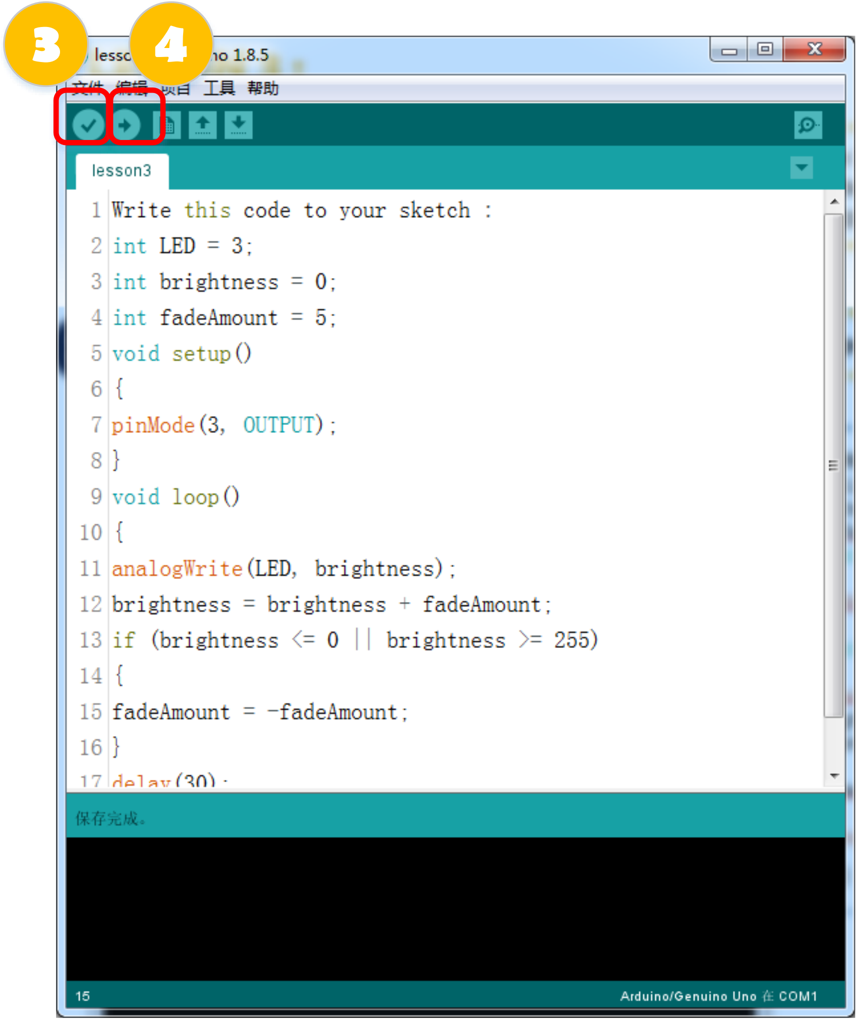
The LED will fade using analogWrite() function using Pulse Width Modulation (PWM) which make a digital output acting as analog output.

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);

    }
```



```
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);
```

```
保存完成。
```

```
15                                   Arduino/Genuino Uno 在 COM1
```

3 Compile the file.

4 Upload the sketch.

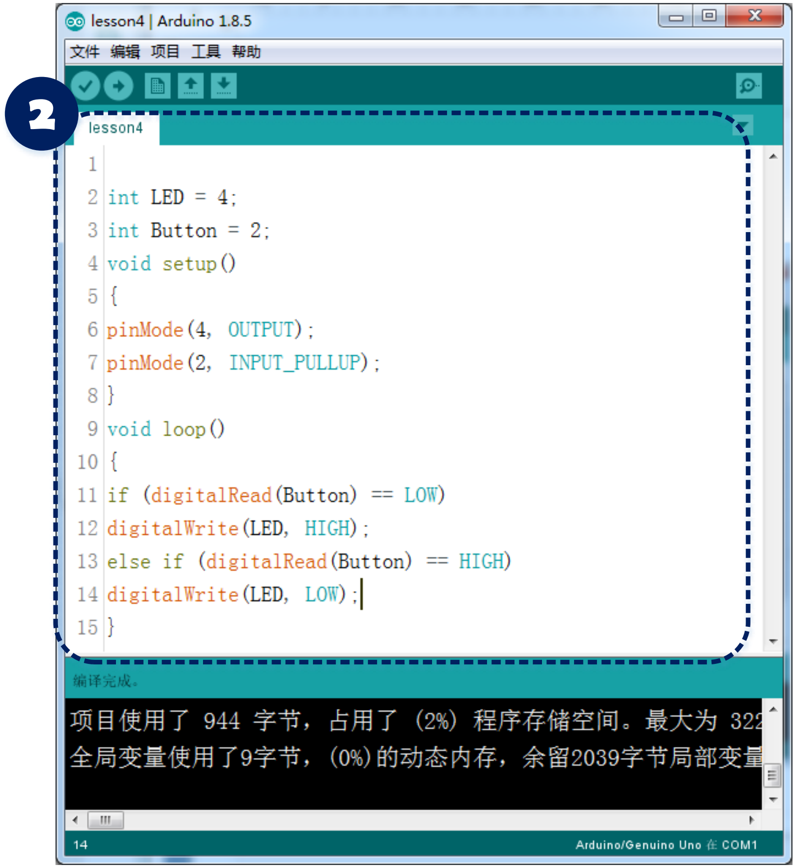
5 You will see status of “Done Uploading”if everything is correct your LED at pin 7 will light

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 identi?ed 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



```
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);
}
```

```
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
```

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!

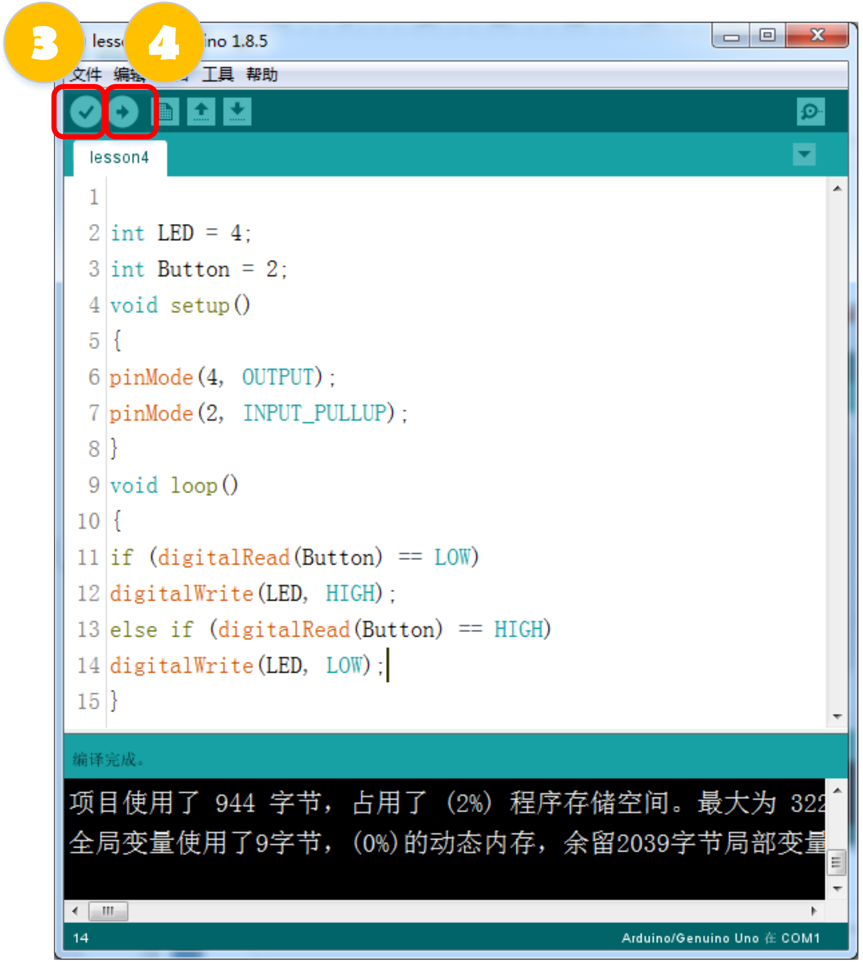
1

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);

   }
```



```
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);
}
```

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 con?guration 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.

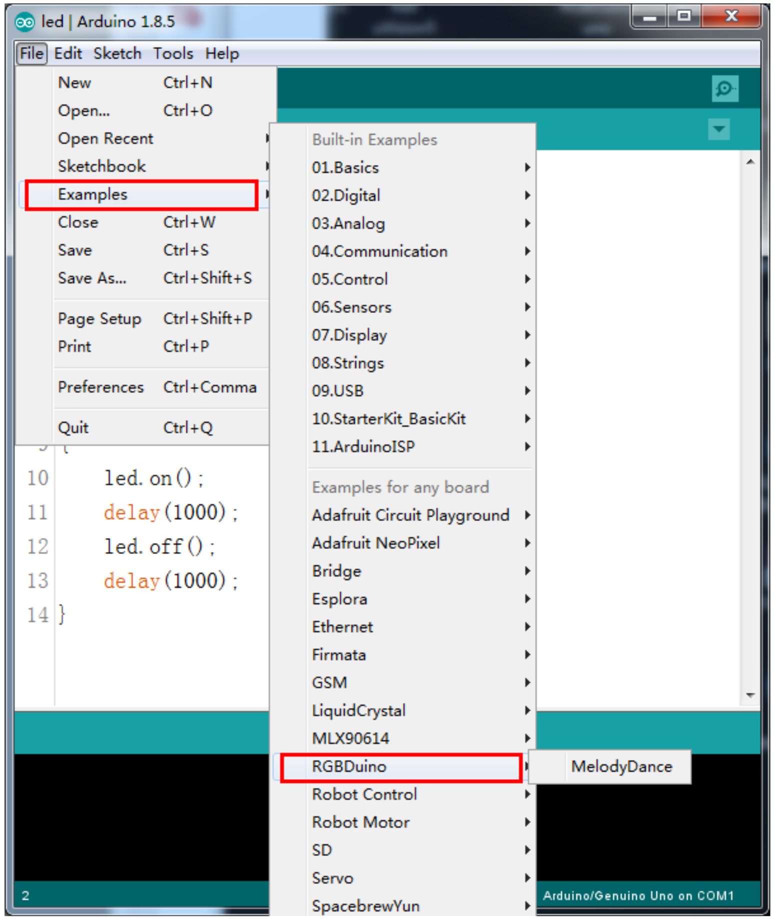
The else if() statement is use to set other condition than if() statement.

The digitalRead(Button) == LOW); will read the button input. If the button is pushed, the INPUT will be LOW.

LESSON 5:

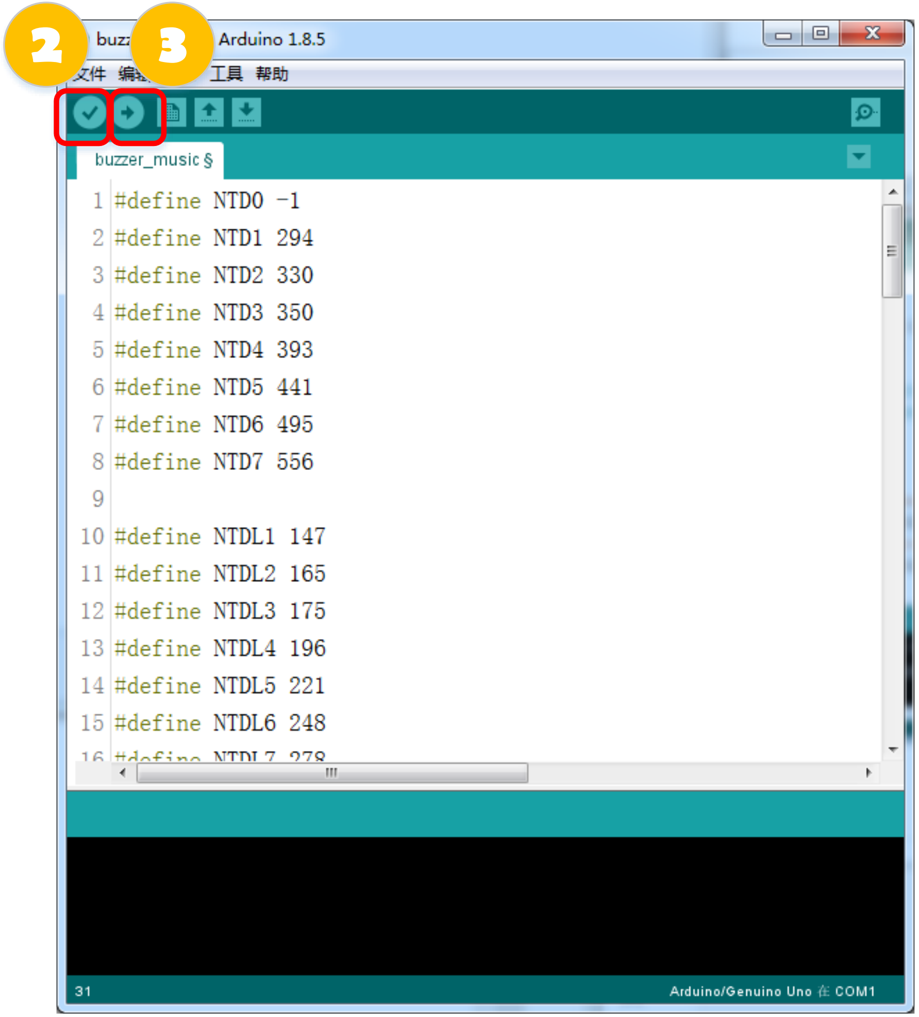
MELODY DANCE

LESSON 5 :
MELODY DANCE



```
    led.on();
    delay(1000);
    led.off();
    delay(1000);
}
```

1 Go to File > Examples > RGBDuino > MelodyDance



```
#define NTD0 -1
#define NTD1 294
#define NTD2 330
#define NTD3 350
#define NTD4 393
#define NTD5 441
#define NTD6 495
#define NTD7 556

#define NTDL1 147
#define NTDL2 165
#define NTDL3 175
#define NTDL4 196
#define NTDL5 221
#define NTDL6 248
#define NTDL7 278
```

2 Compile the file.

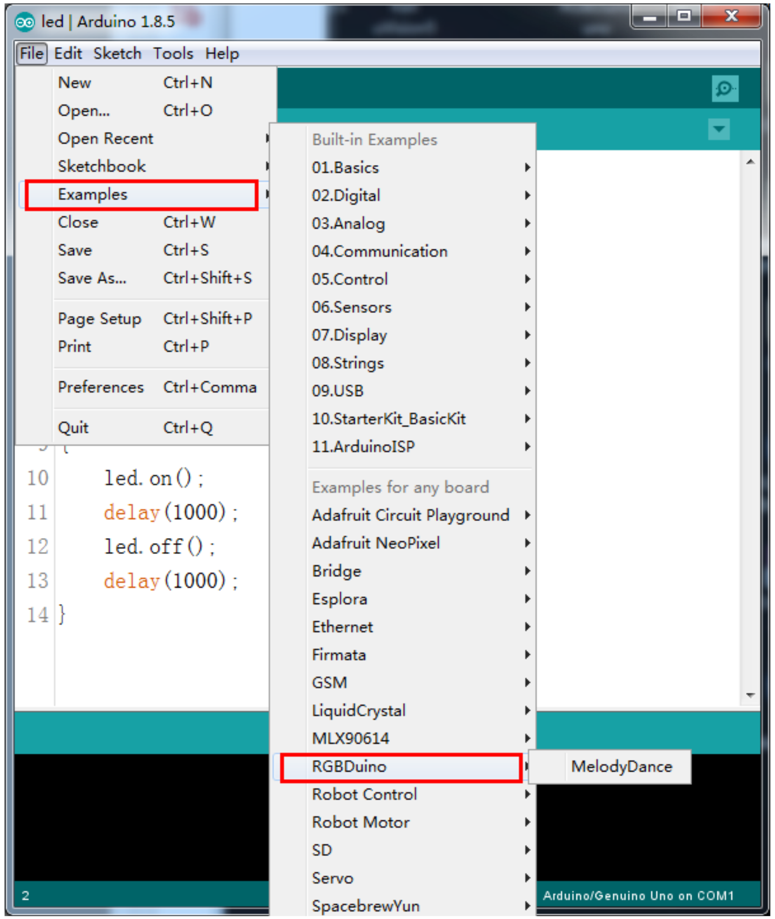
3 Upload the sketch.

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

LESSON 6:

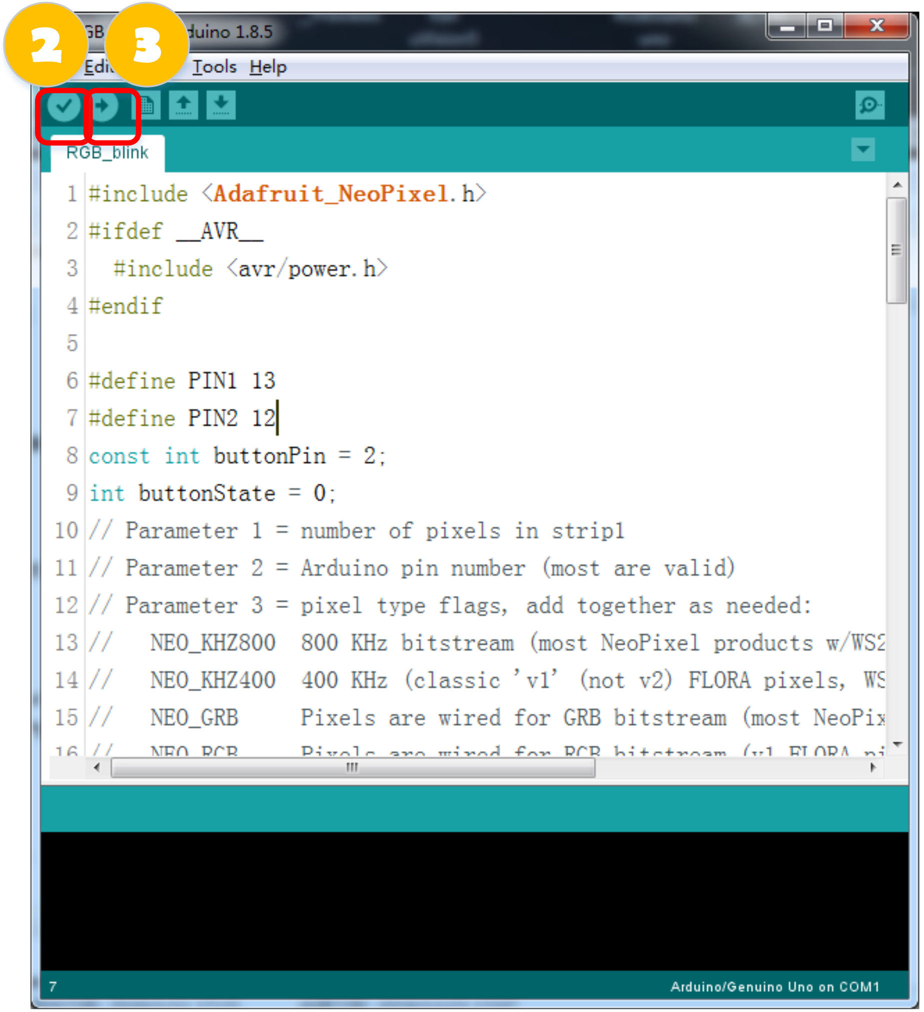
RGB BLINK

LESSON 6 :
RGB BIINK



```
{
    led.on();
    delay(1000);
    led.off();
    delay(1000);
}
```

1 Go to File > Examples > RGBDuino > RGBBLINK



```
#include <Adafruit_NeoPixel.h>
#ifdef __AVR__
  #include <avr/power.h>
#endif

#define PIN1 13
#define PIN2 12
const int buttonPin = 2;
int buttonState = 0;
// Parameter 1 = number of pixels in strip1
// 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/WS2
//   NEO_KHZ400  400 KHz (classic 'v1' (not v2) FLORA pixels, WS
//   NEO_GRB     Pixels are wired for GRB bitstream (most NeoPix
//   NEO_RGB     Pixels are wired for RGB bitstream (v1 FLORA pi
```

2 Compile the file.

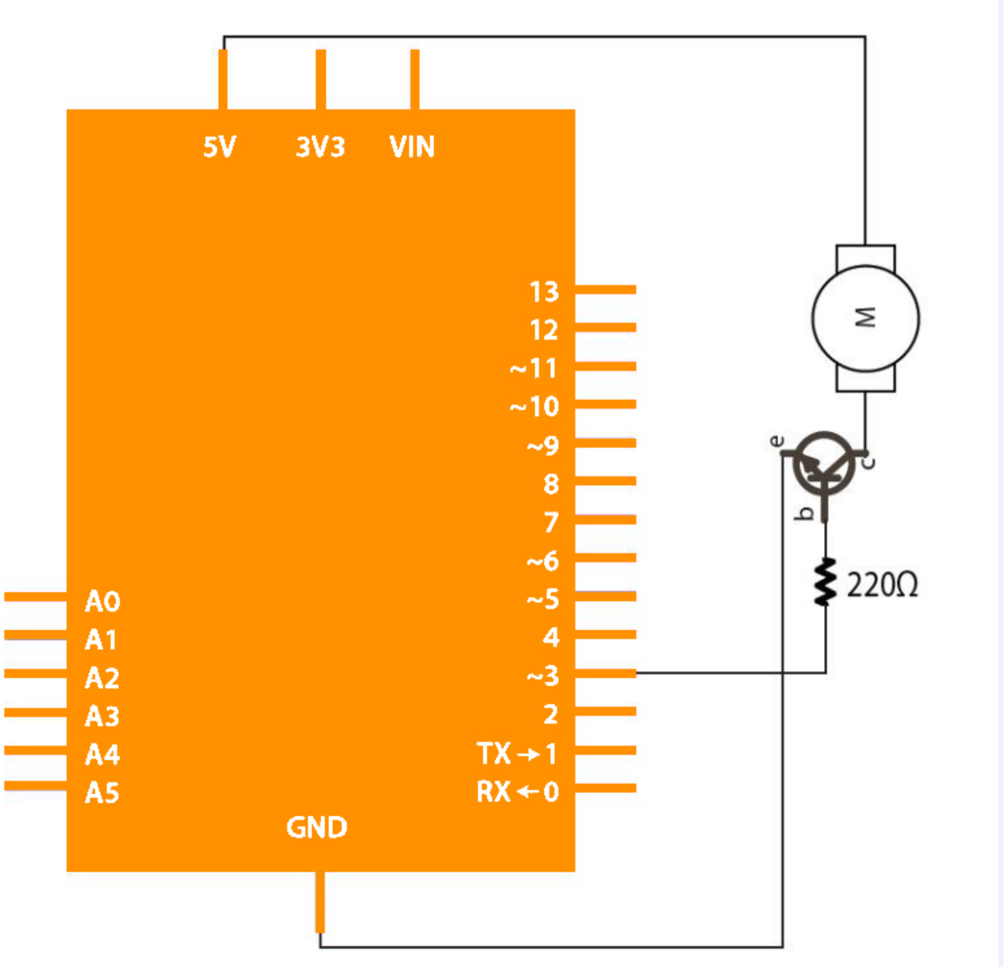
3 Upload the sketch.

4 You will see two RGB lights changing colors flashing

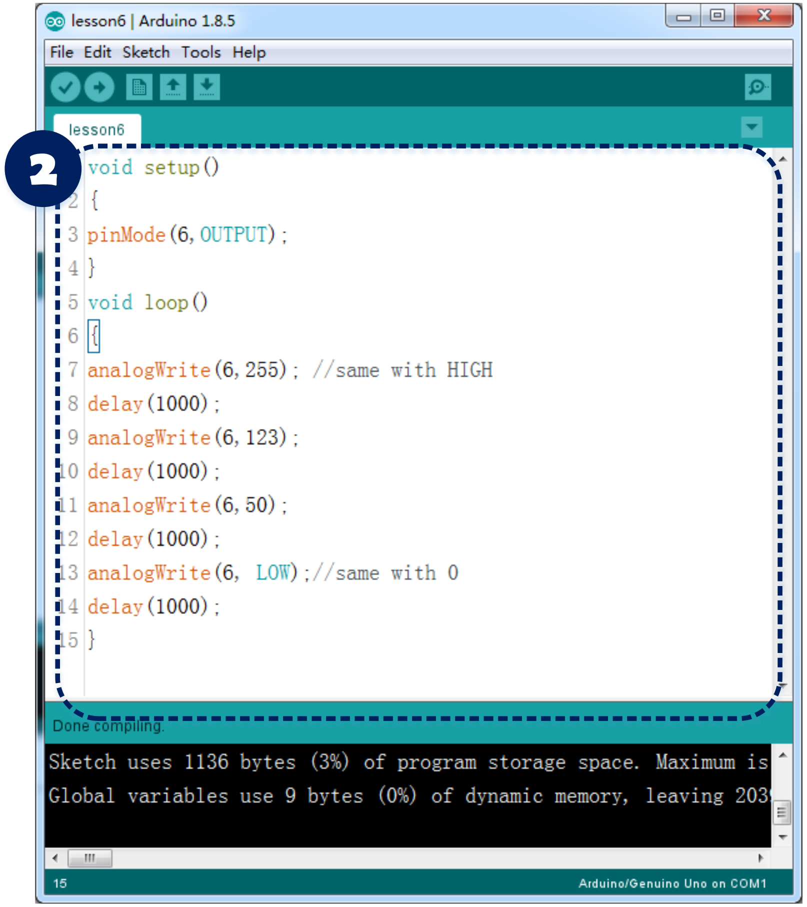
LESSON 7:

CONTROLLING MOTOR

LESSON 7 : SCHEMATIC



LESSON 7 :
CONTROLLING MOTOR



```
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);
}
```

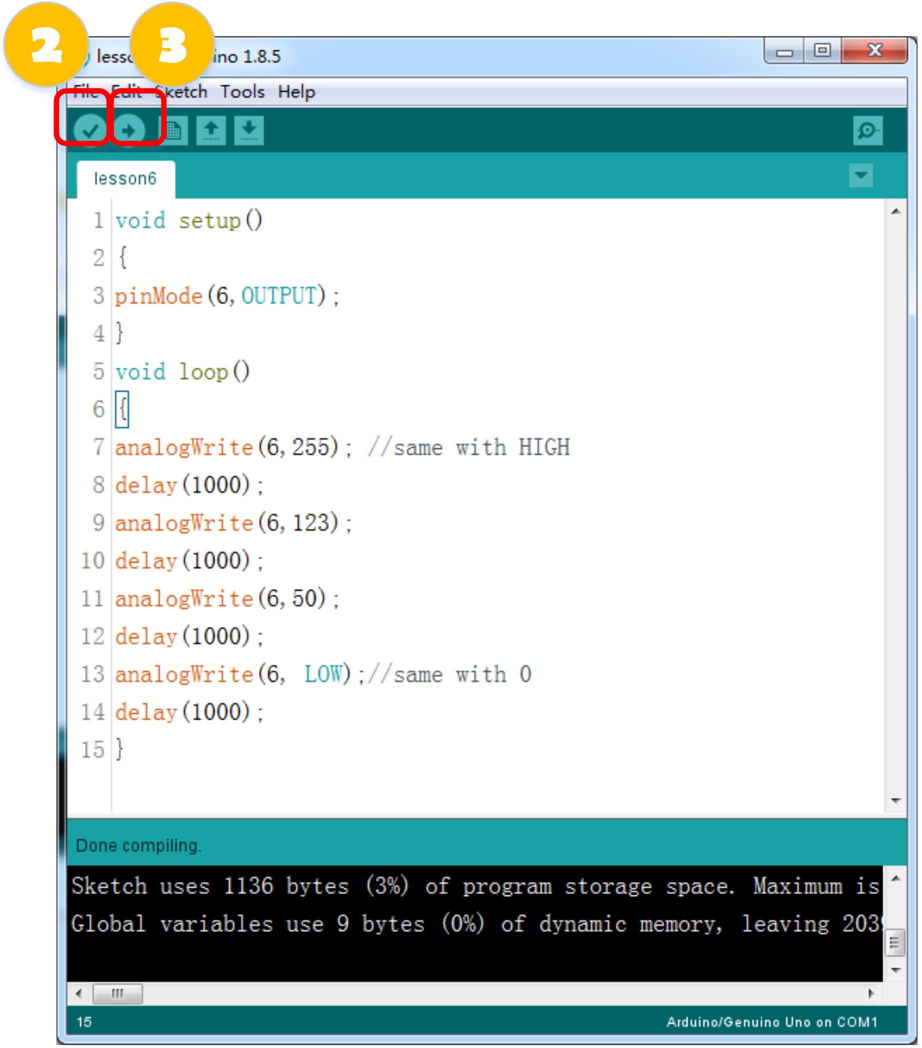
Done compiling.

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

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);
}
```



```
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);
}
```

Done compiling.

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

2 Compile the file.

3 Upload the sketch.

4 The motor will rotate with 4 different speed.

www.RGBDuino.com

Email:

RGBDuino@163.com