

OPEN-SMART

Name: Rich UNO R3 user manual

Version: v1.0

Date: 2017.06.12

©OPEN-SMART

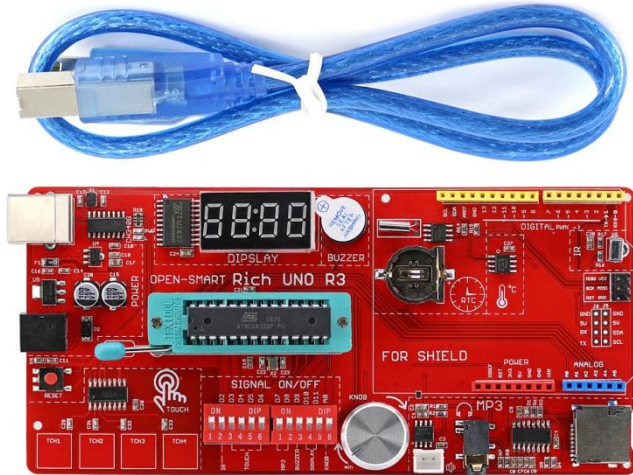
1 Description	- 5 -
2 Specification.....	- 7 -
3 Interface	- 8 -
4 Before wonderful projects.....	- 10 -
4.1 Get Arduino IDE	- 10 -
4.2 Install library	- 12 -
4.3 Get voice sources	- 13 -
4.4 Install driver for CH340G	- 14 -
4.5 Select board and serial port	- 16 -
5 Lesson 1: Touch and sound	- 17 -
5.1 Overview	- 17 -
5.2 Upload code for lesson 1	- 17 -
5.3 Explain nouns.....	- 19 -
5.4 Learn the functions.....	- 19 -
6 Lesson 2: Display number.....	- 21 -
6.1 Overview	- 21 -
6.2 Upload code for lesson 2.....	- 21 -
6.3 Learn the functions.....	- 23 -
7 Lesson 3: Touch and display	- 25 -
7.1 Overview	- 25 -
7.2 Upload code for lesson 3.....	- 25 -
7.3 Learn the functions.....	- 27 -
8 Lesson 4: Touch and debounce	- 29 -
8.1 Overview	- 29 -
8.2 Upload code for lesson 4.....	- 29 -
8.3 Learn the functions.....	- 31 -
9 Lesson 5: Temperature and display	- 33 -
9.1 Overview	- 33 -
9.2 Upload code for lesson 5.....	- 33 -
9.3 Learn the functions.....	- 35 -
10 Lesson 6: Number flashing	- 36 -
10.1 Overview	- 36 -
10.2 Upload code for lesson 6.....	- 36 -
10.3 Learn the functions.....	- 38 -
11 Lesson 7: Knob angle display	- 40 -
11.1 Overview	- 40 -
11.2 Upload code for lesson 7	- 40 -
11.3 Learn the functions.....	- 42 -
12 Lesson 8: Knob control brightness.....	- 43 -
12.1 Overview	- 43 -
12.2 Upload code for lesson 8.....	- 43 -
12.3 Learn the functions.....	- 45 -
13 Lesson 9: Clock input	- 47 -
13.1 Overview	- 47 -

13.2 Upload code for lesson 9.....	- 47 -
13.3 Learn the functions.....	- 51 -
14 Lesson 10: Clock display	- 53 -
14.1 Overview	- 53 -
14.2 Upload code for lesson 10.....	- 53 -
14.3 Learn the functions.....	- 55 -
15 Lesson 11: Test infrared remote keys	- 56 -
15.1 Overview	- 56 -
15.2 Upload code for lesson 11	- 56 -
15.3 Learn the functions.....	- 58 -
16 Lesson 12: Infrared remote control and display	- 60 -
16.1 Overview	- 60 -
16.2 Upload code for lesson 12.....	- 60 -
16.3 Learn the functions.....	- 62 -
17 Lesson 13: Play a song.....	- 63 -
17.1 Overview	- 63 -
17.2 Upload code for lesson 13.....	- 63 -
17.3 Learn the functions.....	- 65 -
18 Lesson 14: Play song with its file name	- 66 -
18.1 Overview	- 66 -
18.2 Upload code for lesson 14.....	- 66 -
18.3 Learn the functions.....	- 68 -
19 Lesson 15: Knob control volume	- 69 -
19.1 Overview	- 69 -
19 Upload code for lesson 15	- 69 -
19.3 Learn the functions.....	- 71 -
20 Lesson 16: Touch control MP3	- 72 -
20.1 Overview	- 72 -
20.2 Upload code for lesson 16.....	- 72 -
20.3 Learn the functions.....	- 74 -
21 Lesson 17: IR control MP3	- 76 -
21.1 Overview	- 76 -
21.2 Upload code for lesson 17.....	- 76 -
21.3 Learn the functions.....	- 78 -
22 Lesson 18: Speak voice.....	- 80 -
22.1 Overview	- 80 -
22.2 Upload code for lesson 18.....	- 80 -
22.3 Learn the functions.....	- 82 -
23 Lesson 19: Speak pi	- 84 -
23.1 Overview	- 84 -
23.2 Upload code for lesson 19.....	- 84 -
23.3 Learn the functions.....	- 86 -
24 Lesson 20: Speak temperature	- 87 -
24.1 Overview	- 87 -

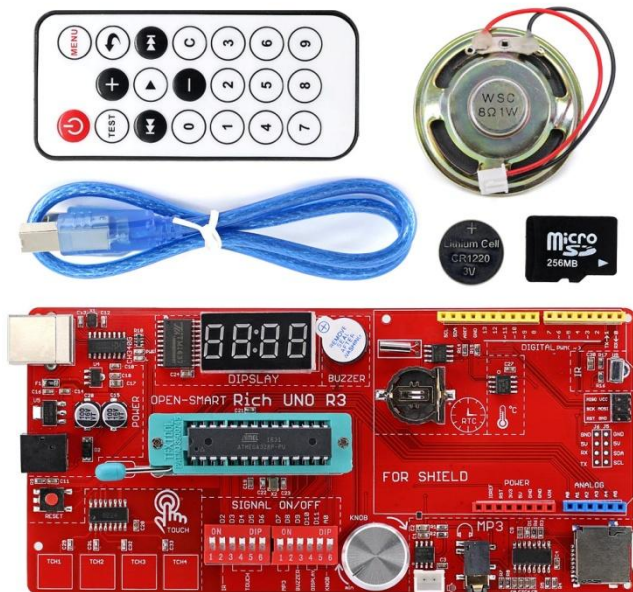
24.2 Upload code for lesson 20.....	- 87 -
24.3 Learn the functions.....	- 89 -
25 Lesson 21: Speak clock.....	- 91 -
25.1 Overview	- 91 -
25.2 Upload code for lesson 21	- 91 -
25.3 Learn the functions.....	- 93 -
26 Lesson 22: Speak time and temperature.....	- 95 -
26.1 Overview	- 95 -
26.2 Upload code for lesson 22.....	- 95 -
26.3 Learn the functions.....	- 97 -
27 Part List.....	- 99 -

1 Description

OPEN-SMART UNO R3 Board top view:



OPEN-SMART UNO R3 Type-A Kit top view:



Overview

Compatible with Arduino UNO R3, OPEN-SMART Rich UNO R3 is an ATMEGA328P development board with rich peripherals. Peripherals include 4 digit display, DS1307 clock, LM75 temperature sensor, infrared receiver, serial MP3 player, rotation angle sensor, 4-channel touch sensor. Hardware resources are very rich, do not need any DuPont wires, and the usage is very simple. It is very suitable for Arduino learners.

4 digit display: 4 digit tube (0.36 inches) which can display the clock point, it needs D10/D11 pins to control and display the integer, clock, stopwatch, score and so on.

DS1307 clock: based on DS1307 high-precision real-time clock module, I2C interface, the address is 0x68.

LM75 temperature sensor: I2C interface temperature sensor, not only can measure the temperature, it can also set the temperature protection temperature, the address is 0x48.

Infrared receiver: use D2 pin, It can receive the modulated infrared signal of 38KHz that is sent by the IR transmitter module and demodulate it into logic level, and it can complete the infrared remote control with the codec program.

Serial MP3: MP3 music player module is based on high-quality MP3 music chip, use D7 / D8 pins to be software serial port, you can send commands to switch songs, change the volume and play mode and other operations.

Rotation angle sensor: 10K ohm adjustable potentiometer knob angle sensor, use A0 pin, can be used for MP3 volume adjustment, 4-digit display brightness adjustment.

4-CH touch sensor: capacitive touch switch, only when you touch the corresponding position with your fingers, the module corresponding pin (D3 / D4 / D5 / D6) will output high level, otherwise it outputs low.

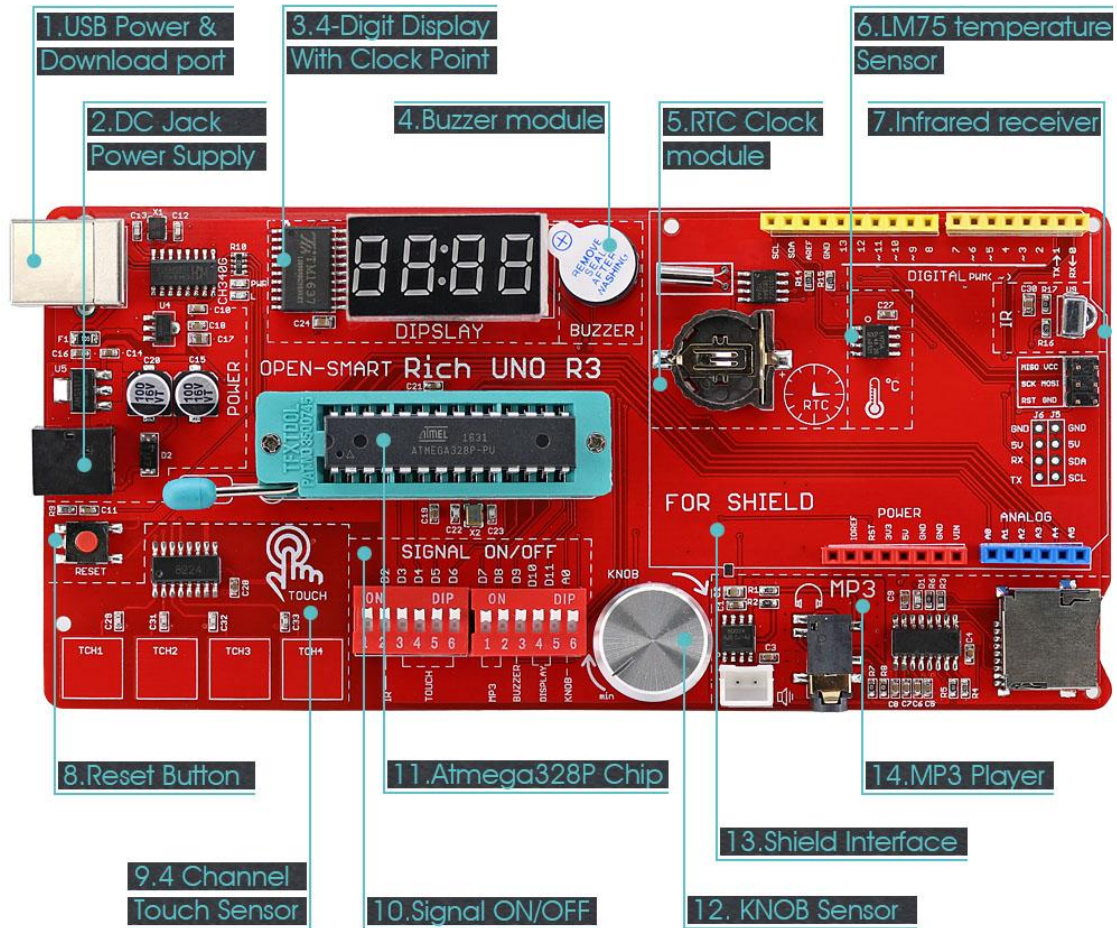
Features:

- Use Arduino UNO bootloader
- It is 100% compatible with Arduino UNO R3 program, expansion shields, IDE.
- Use Through Hole Mount type B USB connector, consistent with Arduino UNO R3, ruggedness and long service life.
- You can burn the program with the type A male to type B male cable.
- Onboard 500mA resettable fuse to protect power supply from the USB port and the DC jack at the same time.
- USB interface driver chip: CH340G, compatible with win7, win8, linux, MAC OS.
- Microcontroller: Atmel ATmega328P
- Working voltage: 5V
- IO logic voltage: 5V
- Working current: 500mA (Max)
- Digital I/O Pins: 14 (of which 6 provide PWM output)
- Flash Memory: 32 KB of which 2 KB used by bootloader
- SRAM: 2KB
- EEPROM: 1KB
- Clock Speed: 16 MHz
- Onboard DIP switch, you can disconnect the connection between the peripheral module on the board and the Atmega328P.
- Onboard Arduino Shield interface, can plug the compatible expansion shield.

2 Specification

Item	Min	Typical	Max	Unit
Power Supply(VCC)	3.7	5	5.25	VDC
DC Jack Power	7	/	12	VDC
Current (@VCC=5V)	/	/	500	mA
Speaker Power	/	/	3	W
Logic interface	5V TTL			/
Supported Card Type	Micro SD card(<=2G); Mirco SDHC card(<=32G)			/
File system format	Fat16 / Fat32			/
Dimensions	165x75x27			mm
Net Weight	107			g

3 Interface



1. **USB Power & Download port:** supply 5V power or connect to PC so that you can upload the sketch code to the board.
2. **DC Jack Power Supply:** you can use the battery case with DC Jack 5.5*2.5 head to supply power for the board, should be 7~12VDC.
3. **4-Digit Display:** 4 digit tube (0.36 inches) which can display the clock point, it needs D10/D11 pins to control and display the integer, clock, stopwatch, score and so on.
4. **Buzzer:**
5. **RTC Clock:** based on DS1307 high-precision real-time clock module, I2C interface, the address is 0x68, it needs the CR1220 battery to keep running when the board is power off.
6. **LM75 Temperature Sensor:** I2C interface temperature sensor, not only can measure the temperature, it can also set the temperature protection temperature, the address is 0x48.
7. **Infrared Receiver:** use D2 pin, It can receive the modulated infrared signal of 38KHz that is

sent by the IR transmitter module and demodulate it into logic level, and it can complete the infrared remote control with the codec program.

8. Reset Button:

9. 4-Channel Touch Sensor: capacitive touch switch, only when you touch the corresponding position with your fingers, the module corresponding pin (D3 / D4 / D5 / D6) will output high level, otherwise it outputs low.

10. Signal ON/OFF: You can disconnect the connection between the peripheral module on the board and the Atmega328P.

11. Atmega328P Chip: 32KB flash with 2KB EEPROM, 16MHz, the same as Arduino UNO R3

12. Knob Sensor: 10K ohm adjustable potentiometer knob angle sensor, use A0 pin, can be used for MP3 volume adjustment, 4-digit display brightness adjustment.

13. Shield Interface: Standard Arduino shield interface, you can plug your shield onto it.

14. MP3 Player: MP3 music player module is based on high-quality MP3 music chip, use D7 / D8 pins to be software serial port, you can send commands to switch songs, change the volume and play mode and other operations.

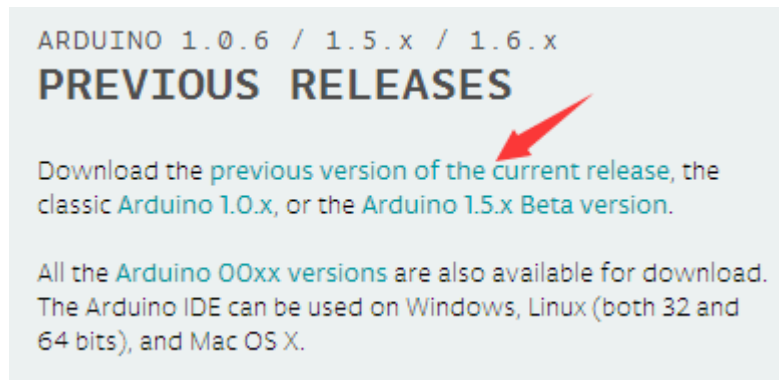
4 Before wonderful projects

4.1 Get Arduino IDE

1) If you do not get the Arduino IDE, please down it [here](#). **Please make sure you will put the IDE in the English path.**

2) We recommend you to download Arduino 1.6.5, because our projects use this version.

Click [previous version of the current release](#) and select 1.6.5 for your PC system, mine is win10 64bit, so I click [Windows](#).



ARDUINO 1.0.6 / 1.5.x / 1.6.x

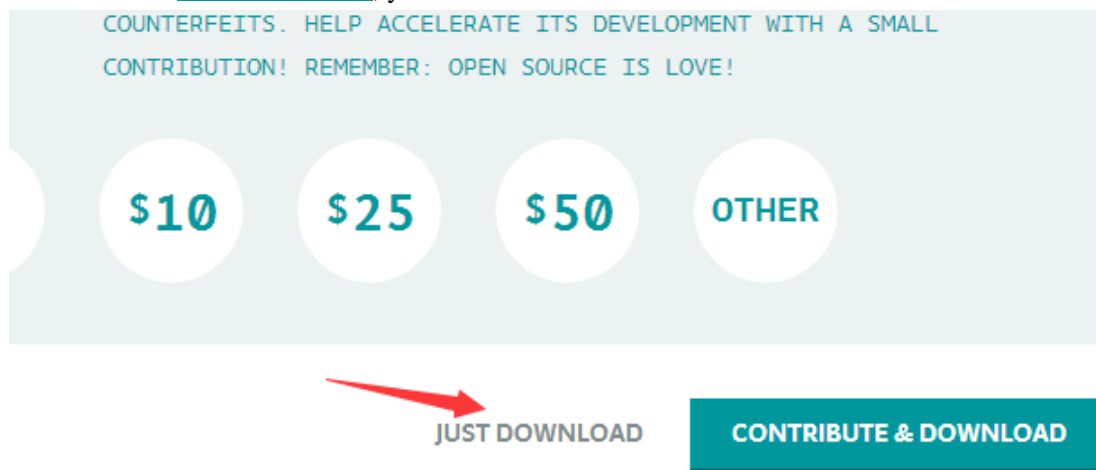
PREVIOUS RELEASES

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.

1.6.5	Windows Windows Installer	MAC OS X	Linux 32 Bit Linux 64 Bit	Source code on Github
-------	------------------------------	----------	------------------------------	--------------------------

And then click [JUST DOWNLOAD](#), you do not need to install the IDE for this version I select.



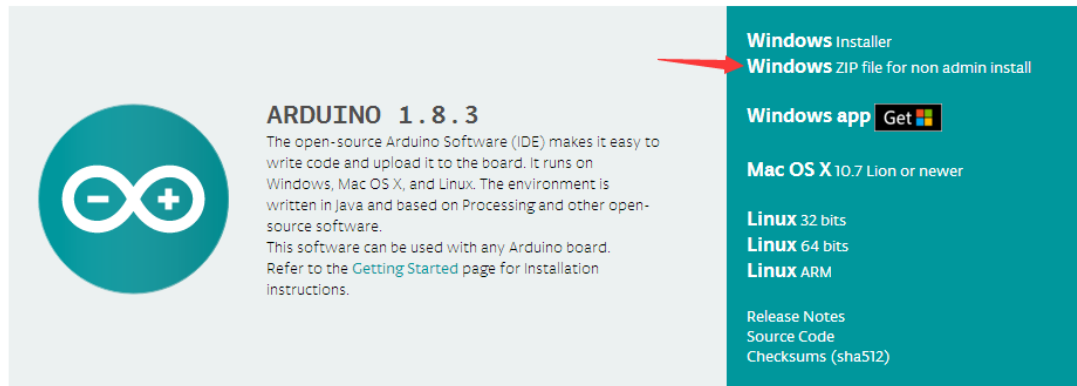
COUNTERFEITS. HELP ACCELERATE ITS DEVELOPMENT WITH A SMALL CONTRIBUTION! REMEMBER: OPEN SOURCE IS LOVE!

\$10 **\$25** **\$50** **OTHER**

JUST DOWNLOAD **CONTRIBUTE & DOWNLOAD**

3) If you want to download the Arduino 1.8.3, you can click [Windows ZIP file for non admin install](#) and then click [JUST DOWNLOAD](#)

Download the Arduino IDE



ARDUINO 1.8.3

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
- Windows** ZIP file for non admin install
- Windows app** [Get](#)
- Mac OS X** 10.7 Lion or newer
- Linux** 32 bits
- Linux** 64 bits
- Linux** ARM
- [Release Notes](#)
- [Source Code](#)
- [Checksums \(sha512\)](#)

Support 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 **16,227,205** 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](#)

4.2 Install library

Why you need install library?

Almost all the Rich UNO R3 board and Arduino peripheral module need corresponding library so that you can use them more easily and you do not have to write too much code to complete the project.

Install all the libraries the Rich UNO R3 needs:

- 1) Get the libraries from our google drive [here](#). The name is **Arduino libraries for Rich UNO R3.rar**
- 2) Unzip it and all the four libraries in it must be put in the directory of `..\Arduino-1.xx\libraries`, and then restart the IDE. All the examples in the directory of `D:\arduino-1.6.5-r2\libraries\RichUNO\examples` are the 22 lessons we are going to do and study Arduino and Rich UNO R3.
- 3) Restart the Arduino: If you have open the IDE, after you have install all the libraries, please close all the windows of the IDE.

4.3 Get voice sources

We get the voices sources from [here](#). You can input the word and setting something and it will output the voice you want. It is very good, easy and free.

- 1) We have get some voice we need in the 22 projects, and you can download from our google drive [here](#). The name is **Voice Sources for TF card.rar**
- 2) Format your TF card (should not larger than 32GB), unzip **Voice Sources for TF card.rar** . Find the 01 and 02 folder and put them into your TF card. 01 folder has two songs in it, one is Yesterday Once More, and the other is More Than I Can Say. 02 folder has 40 voice segments include some numbers and prompt voice.

4.4 Install driver for CH340G

1) If you are using win7 operating system, download the CH340G driver from [here](#).

If your system is MAC OS, download [here](#).

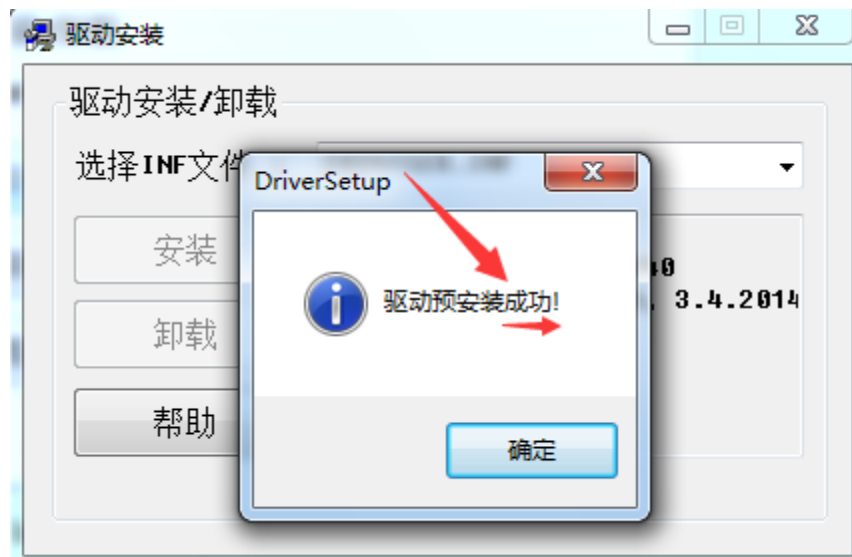
If yours is linux, download [here](#).

NOTE: Before plugging the USB cable to the PC, you should install the driver first.

2) When you have downloaded it, click the CH341SER.EXE

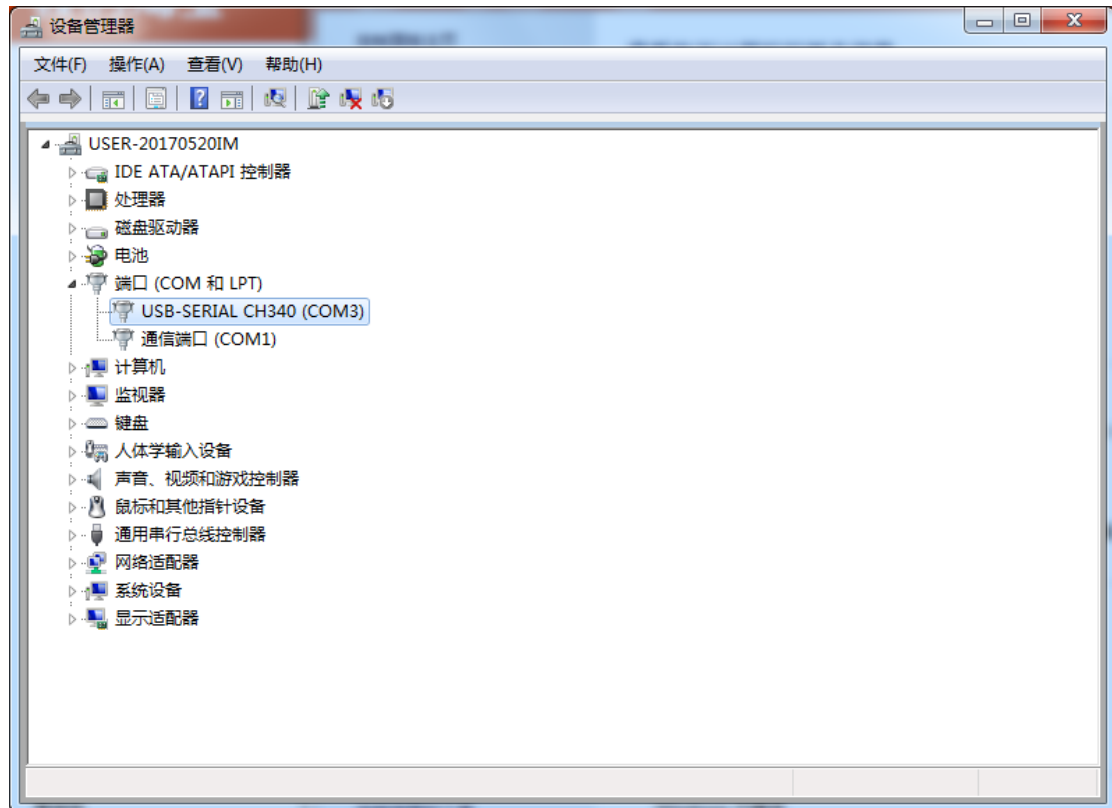


(3) Click the Install Button to pre-install the driver and then it is successfully installed



(4) Close the window and plug the module to your PC with the USB cable and it will automatically install USB interface driver.

(5) Then you can find its serial port number, mine is COM3, you should find yours and then you should select it in the Arduino IDE.



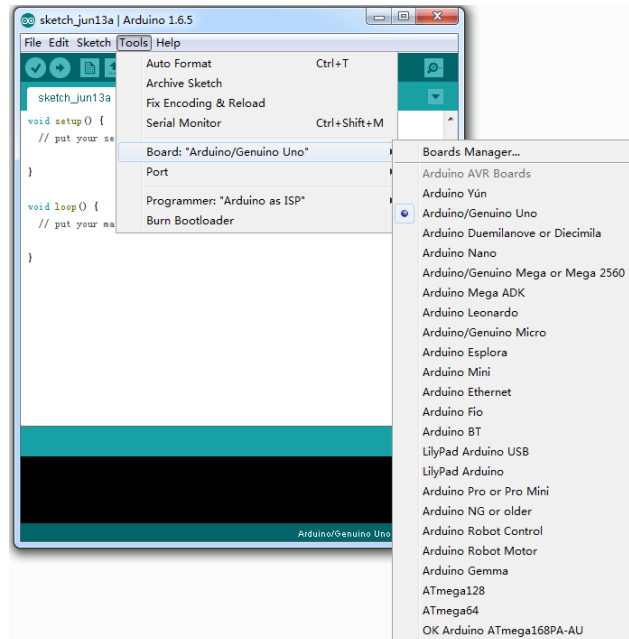
4.5 Select board and serial port

You can refer to the video Select board and serial port. If you plug the board into the same USB port, after you have set board and serial port, you do not need to do it again the next time. If you plug it into different USB port you should select again.

1) Select board:

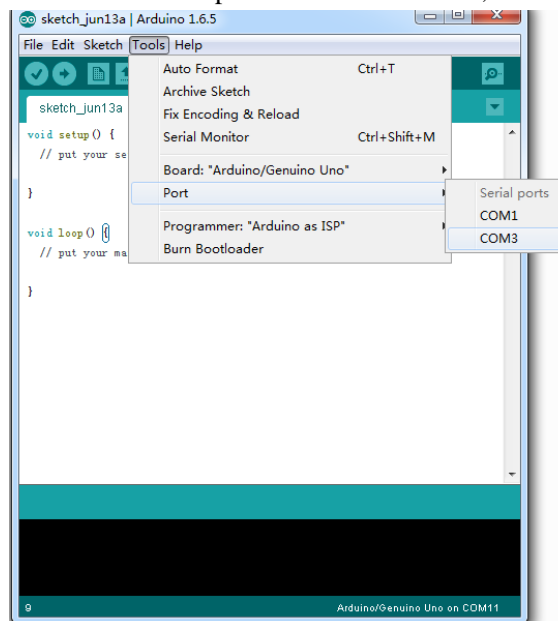
Click arduino.exe in the directory of D:\arduino-1.6.5-r2 to start the *Arduino IDE*.

Click Tools to select Arduino/Genuino Uno for our Rich UNO R3



2) Select serial port

Get to know what serial port you find at the end of **4.4 Install driver for CH340G**, and then Click Tools to select the port and mine is COM3, so I click COM3.



Now all the preparatory work has been completed, please enjoy all the projects in the next section.

5 Lesson 1: Touch and sound

5.1 Overview

We select capacitive touch sensor instead of traditional buttons to reduce the pain of pressing the button and increase the service life. The *buzzer* has only two status such sound and silence.

In this lesson, you can learn about the touch pad control method, how to turn on and turn off the buzzer.

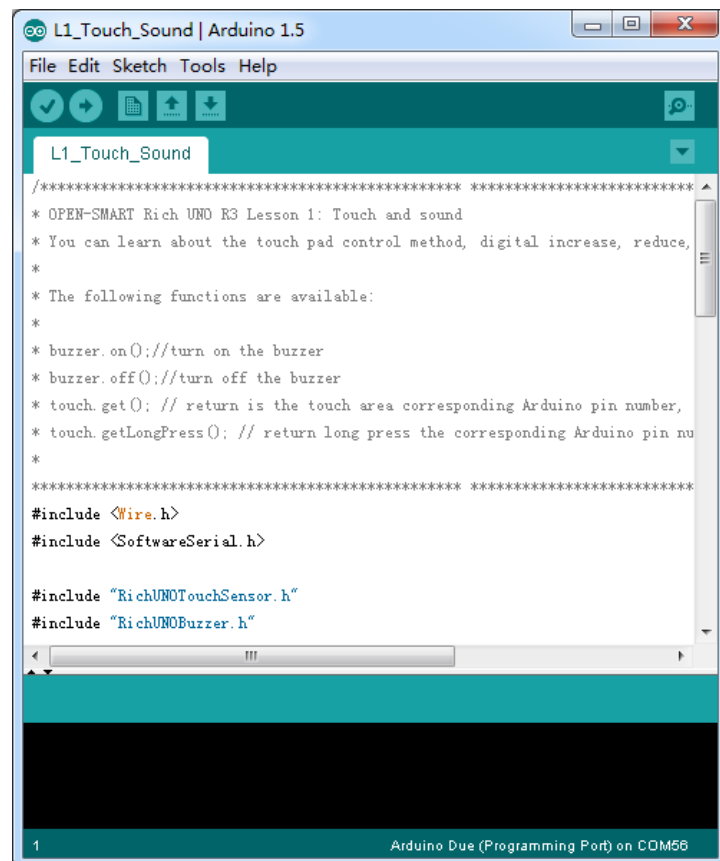
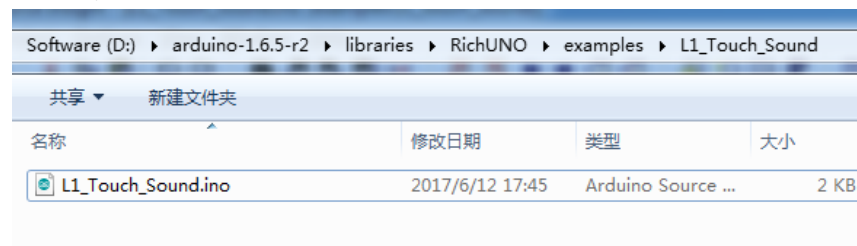
5.2 Upload code for lesson 1

1) Open lesson1 *example*

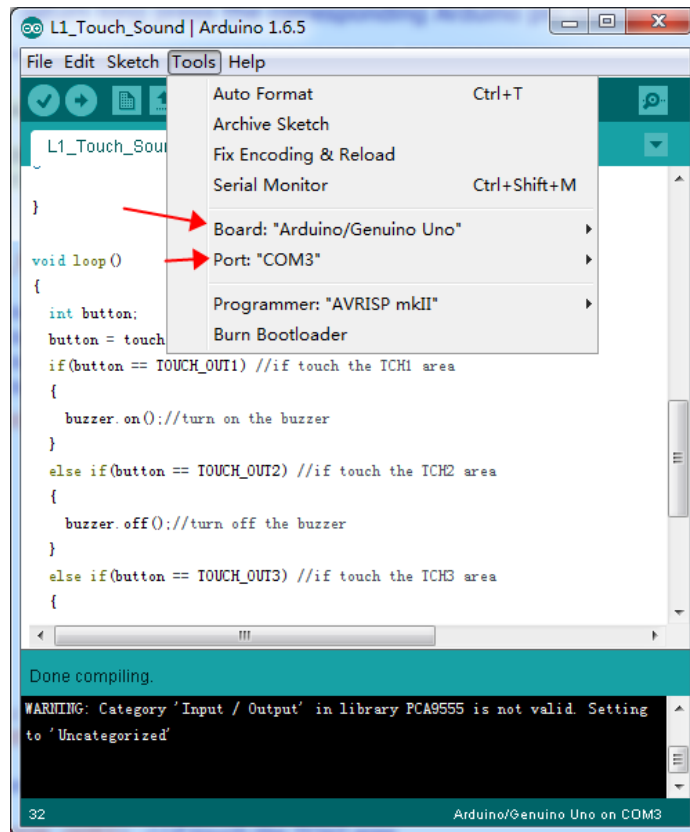
Double click the file in the directory of

\\arduino-1.6.5-r2\\libraries\\RichUNO\\examples\\L1_Touch_Sound

So that you find the window below:

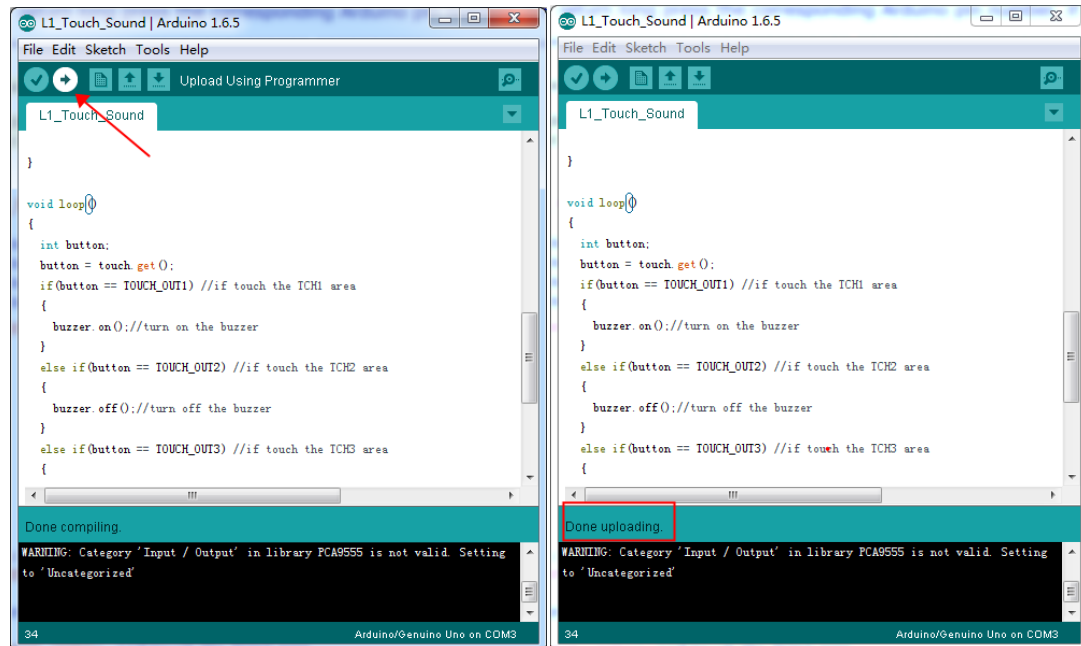


2) Check the Board and Serial Port



3) Click Upload button to upload the code to Rich UNO R3 board

When you see that “Done uploading”, congratulations, you have upload the code to the board successfully.



4) Check whether the board is running as you want

In this lesson, you only need to test the touch sensor and the buzzer.

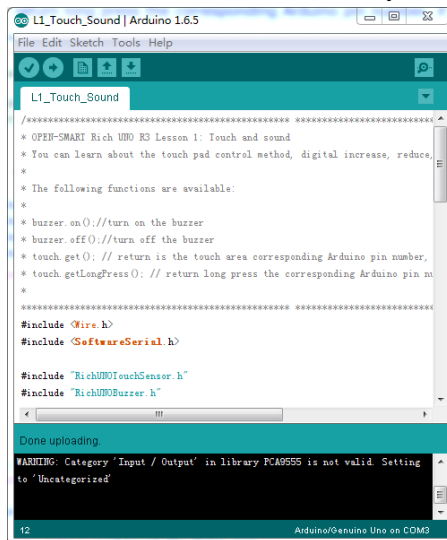
Touch TCH1 area with your finger, the buzzer continues to sound.

Touch TCH2 area, the buzzer stops sounding.

5.3 Explain nouns

Each lesson may explain some nouns in this section.

Arduino IDE: Integrated development environment for Arduino series board, it contains all the files in arduino-1.6.5-r2 directory. Usually we call the IDE window as IDE.



buzzer: the buzzer onboard is a simple sounding component, and the sound frequency is fixed.

example: somebody call it sketch code. It is a program you want to and you can upload to the board

5.4 Learn the functions

1) You can find that at the beginning of all lessons code is

```
#include <Wire.h>
```

```
#include <SoftwareSerial.h>
```

Because the RichUNO library files use Wire.h and SoftwareSerial.h which is not in the RichUNO directory, if the example code wants to use RichUNO library, it should add those two line.

2) Other header files

In lesson 1 we use the buzzer and 4-channel touch sensor, so it should include their header files to control buzzer and touch sensor more easily.

So you should add:

```
#include "RichUNOTouchSensor.h"
```

```
#include "RichUNOBuzzer.h"
```

3) Define class object

We have defined the buzzer and touch sensor class in the RichUNO library, and at any time you should define a object before you use it, and some object has parameter.

For example:

Buzzer buzzer(9);//define a Buzzer object, and the buzzer connect to D9 of Rich UNO R3 board

4) Two necessary functions

Each example code should include the two necessary functions:

```
void setup(){ }
```

```
void loop(){} 
```

Usually we write initialization code in setup function, and then write the code what you want the board to do in loop function which runs cyclically.

5) Functions

```
*buzzer.on();//turn on the buzzer
```

```
*buzzer.off();//turn off the buzzer
```

You can try to use delay function to make intermittent sound .

For example:

```
buzzer.on();
```

```
delay(200);//sound for 200ms
```

```
buzzer.off();
```

```
*touch.get(); // return is the touch area corresponding Arduino pin number, if not then return -1
```

```
*touch.getLongPress(); // return long press for more than 3 seconds the corresponding Arduino  
pin //number, if not return -1
```

6 Lesson 2: Display number

6.1 Overview

4-digit 7-segment display with clock point can clearly display number in the sun so it is often use for watches / Clock / Induction Cooker.

In this lesson, you can learn how to display numbers, which can be displayed in bits, or you can display integers directly.

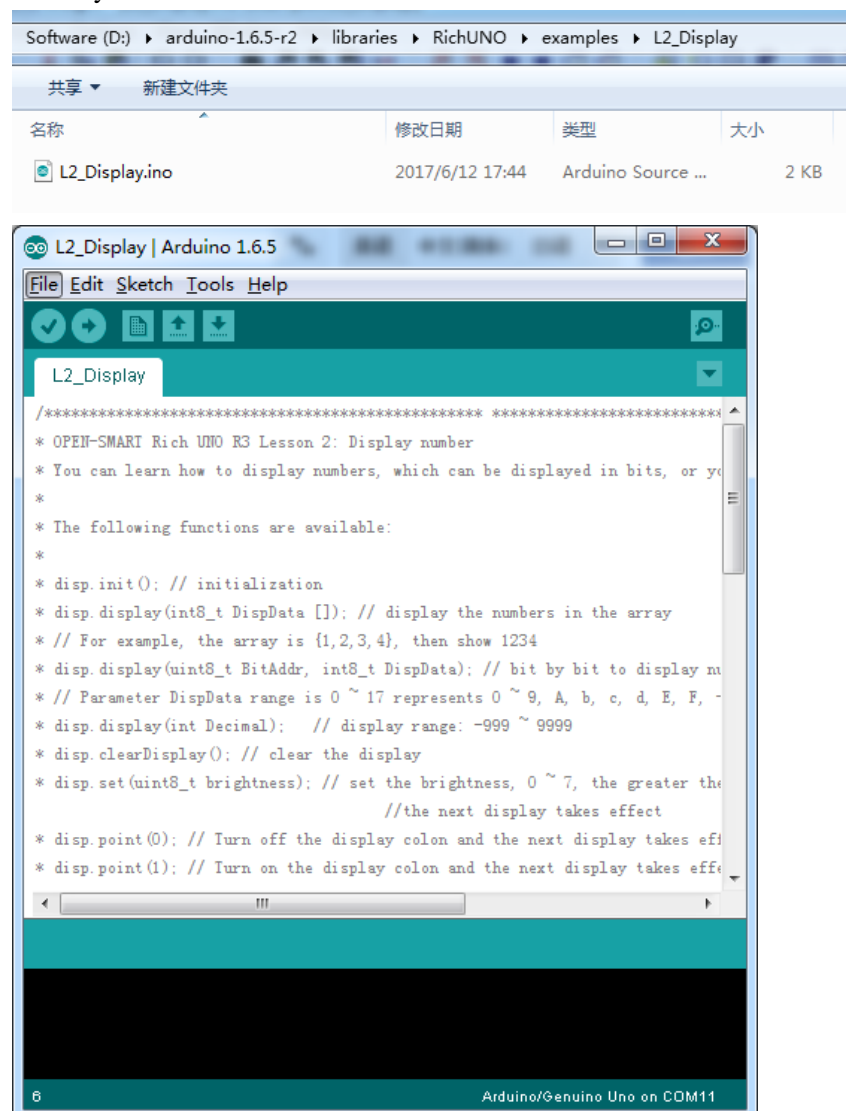
6.2 Upload code for lesson 2

1) Open lesson2 example

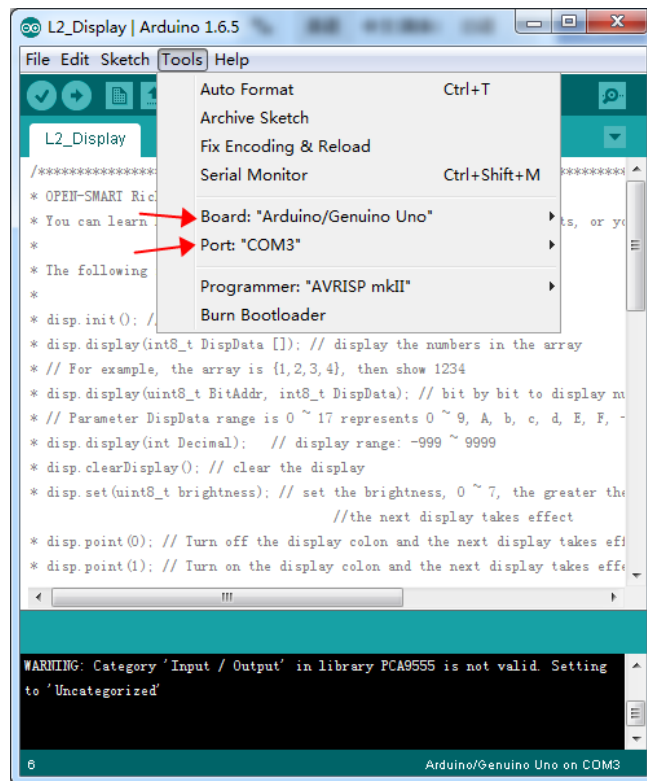
Double click the file in the directory of

\\arduino-1.6.5-r2\\libraries\\RichUNO\\examples\\L2_Display

So that you find the window below:

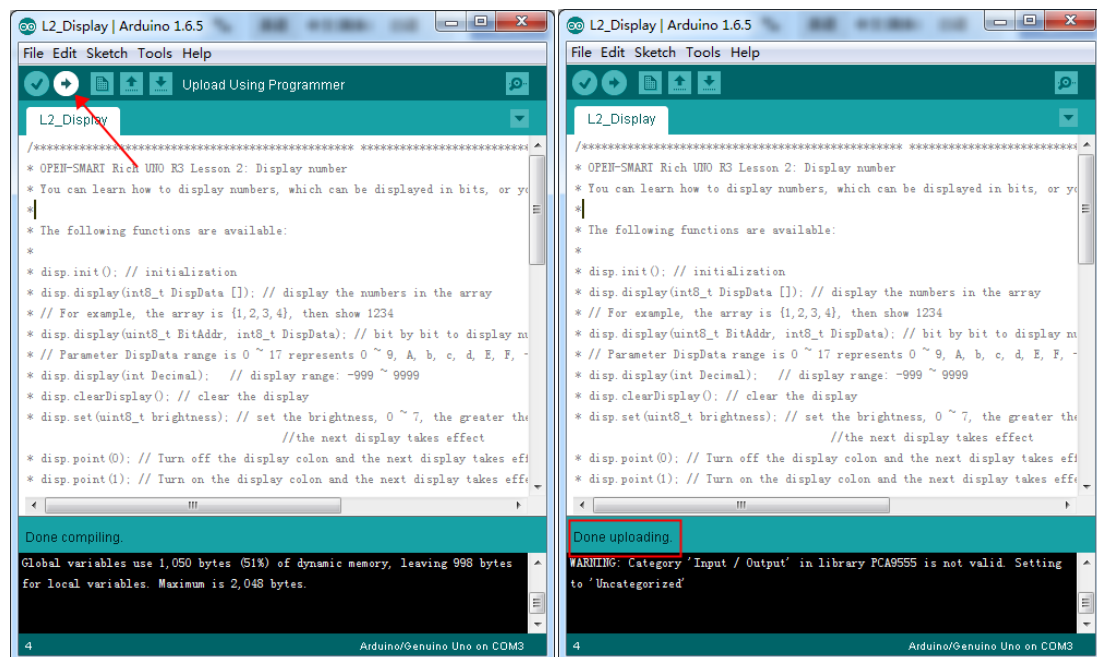


2) Check the Board and Serial Port



3) Click Upload button to upload the code to Rich UNO R3 board

When you see that “Done uploading”, congratulations, you have upload the code to the board successfully.



4) Check whether the board is running as you want

In this lesson, you can see it displays integer of 3456 / -100 / 25 cyclically.

6.3 Learn the functions

1) You can find that at the beginning of all lessons code is

```
#include <Wire.h>
```

```
#include <SoftwareSerial.h>
```

Because the RichUNO library files use Wire.h and SoftwareSerial.h which is not in the RichUNO directory, if the example code wants to use RichUNO library, it should add those two line.

2) Other header files

In lesson 2 we use 4-digit display, so it should include their header files to control display more easily.

So you should add:

```
#include "RichUNOTM1637.h"
```

3) Define class object

We have defined the display class in the RichUNO library, and at any time you should define a object before you use it, and some object has parameter.

For example:

```
TM1637 disp(10,11);
```

//define a TM1637 object, and display connect to D10 / D11 of Rich UNO R3 board

4) Two necessary functions

Each example code should include the two necessary functions:

```
void setup(){ }
```

```
void loop(){} 
```

Usually we write initialization code in setup function, and then write the code what you want the board to do in loop function which runs cyclically.

5) Functions

```
* disp.init(); // initialization
```

You should write it in the setup function initialize it.

```
* disp.display(int8_t DispData []); // display the numbers in the array
```

// For example, the array is {1,2,3,4}, then show 1234

```
* disp.display(uint8_t BitAddr, int8_t DispData); // bit by bit to display numbers, characters,
```

//BitAddr range is 0 ~ 3,Parameter DispData range is 0 ~ 17 represents

//0 ~ 9, A, b, c, d, E, F, -, space

You can try to write " disp.display(0, 10); " to display "A"

```
* disp.display(int Decimal); // display range: -999 ~ 9999
```

```
* disp.clearDisplay(); // clear the display, nothing display
```

```
* disp.set(uint8_t brightness); // set the brightness, 0 ~ 7, the greater the value, the higher the  
    //brightness, the next display takes effect  
* disp.point(0); // Turn off the display colon and the next display takes effect  
* disp.point(1); // Turn on the display colon and the next display takes effect
```

7 Lesson 3: Touch and display

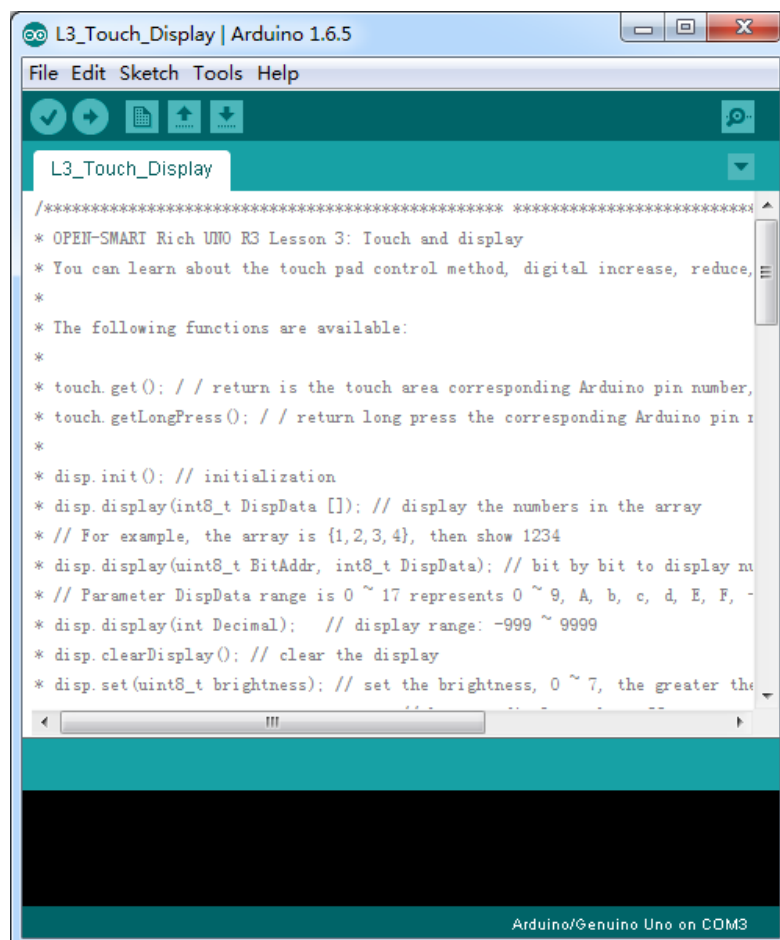
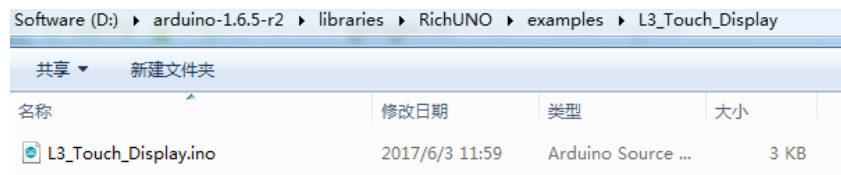
7.1 Overview

The board has 4 touch area, and you can distinguish which area you touch refer to the code. You can learn about the touch pad control method, digital increase, reduce, clear operation of the digital display.

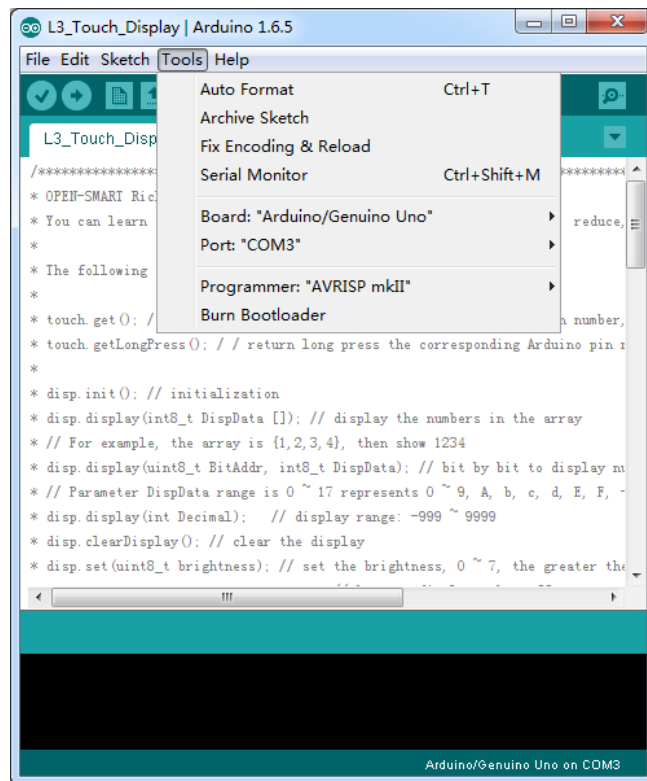
7.2 Upload code for lesson 3

1) Open lesson3 example

Double click the file in the directory of
 \arduino-1.6.5-r2\libraries\RichUNO\examples\L3_Touch_Display
 So that you find the window below:

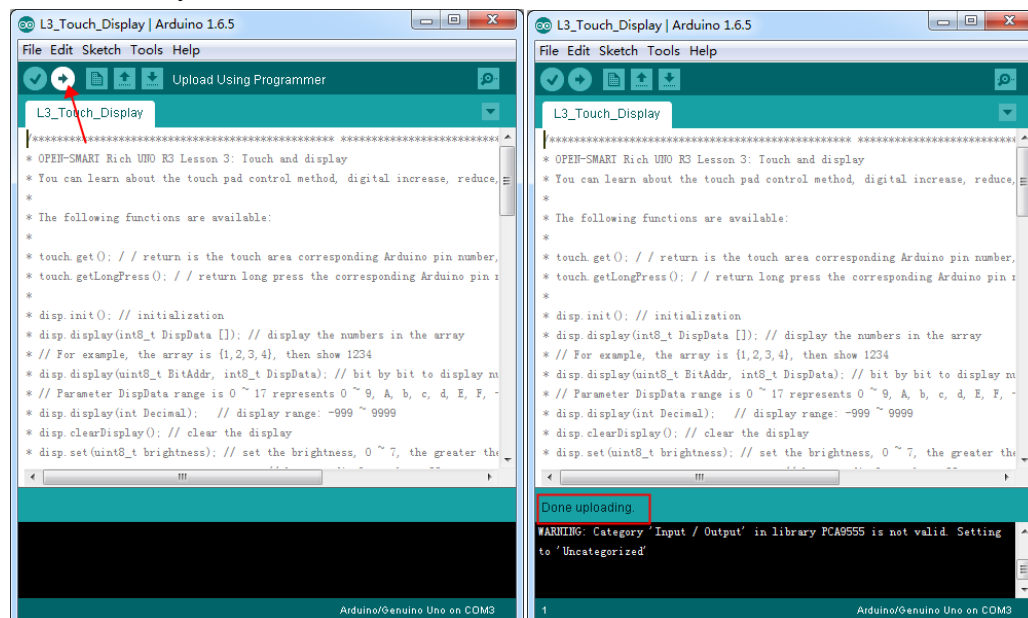


2) Check the Board and Serial Port



3) Click Upload button to upload the code to Rich UNO R3 board

When you see that “Done uploading”, congratulations, you have upload the code to the board successfully.



4) Check whether the board is running as you want

In this lesson, when you touch TCH1, the number increase. When you touch TCH2, the number decrease. When you touch TCH3, clear number to be "0". When you touch TCH4, it displays maximum number "9999".

7.3 Learn the functions

1) You can find that at the beginning of all lessons code is

```
#include <Wire.h>
```

```
#include <SoftwareSerial.h>
```

Because the RichUNO library files use Wire.h and SoftwareSerial.h which is not in the RichUNO directory, if the example code wants to use RichUNO library, it should add those two line.

2) Other header files

In lesson 3 we use 4-digit display and touch sensor, so it should include their header files to control display and touch sensor more easily.

So you should add:

```
#include "RichUNOTouchSensor.h"
```

```
#include "RichUNOTM1637.h"
```

3) Define class object

We have defined the display class in the RichUNO library, and at any time you should define a object before you use it, and some object has parameter.

For example:

```
TM1637 disp(10,11); //define a TM1637 object, and display connect to D10 / D11 of Rich UNO R3 board
```

4) Two necessary functions

Each example code should include the two necessary functions:

```
void setup(){ }
```

```
void loop(){} }
```

Usually we write initialization code in setup function, and then write the code what you want the board to do in loop function which runs cyclically.

5) Functions

** touch.get(); // return is the touch area corresponding Arduino pin number, if not then return -1
If you touch TCH1, it will return 3. TCH2 is 4, TCH3 is 5, and TCH4 is 6.*

You may try to write this function to get 1,2,3,4 for TCHx.

```
int get()
```

```
{
```

```
    int touchNum;
```

```
    int _pin;
```

```
    _pin = touch.get();
```

```
    switch(_pin)
```

```
    {
```

```
        case TOUCH_OUT1: touchNum = 1; break;
```

```
        case TOUCH_OUT2: touchNum = 2; break;
```

```
        case TOUCH_OUT3: touchNum = 3; break;
```

```
        case TOUCH_OUT4: touchNum = 4; break;
```

```
        default: touchNum = _pin; break;
```

```
    }
```

```
    return touchNum;
```

```
}
```

** `disp.init();` // initialization*

You should write it in the setup function initialize it and it has set the default brightness to be 3(0~7, number larger, brightness higher).

** `disp.display(int Decimal);` // display range: -999 ~ 9999*

If your number is not in the range, it will ignore your operation.

8 Lesson 4: Touch and debounce

8.1 Overview

Maybe you have found that when you touch TCH1 the number increases too fast in lesson 3 and the number is not that you want to display. So we can add debounce code to avoid this problem.

In this lesson, you can learn about software debounce method to achieve the number increasing step by step and the step maybe one, two or more.

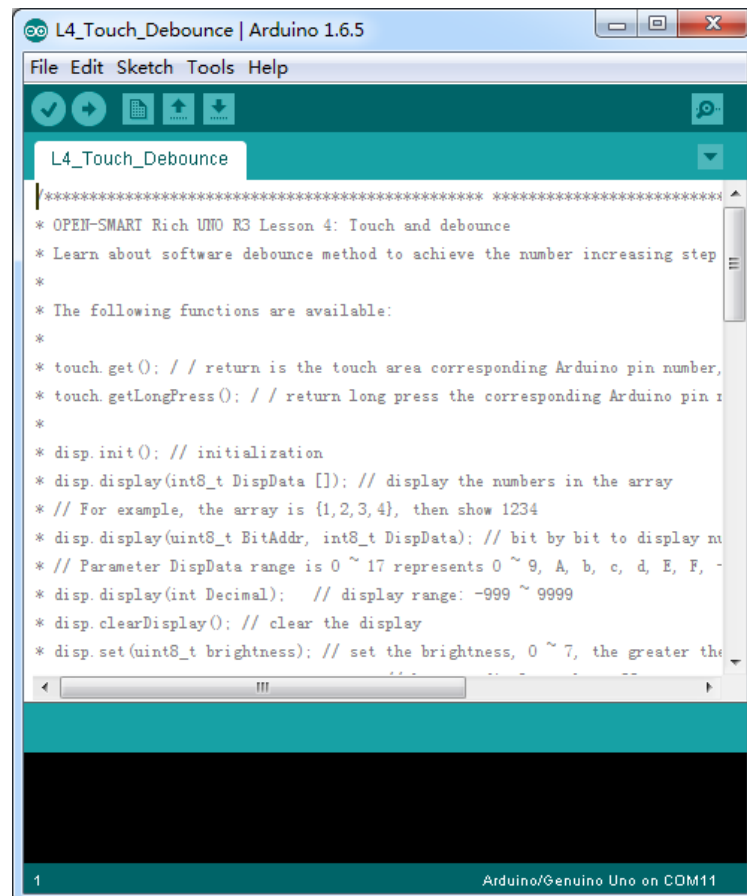
8.2 Upload code for lesson 4

1) Open lesson4 example

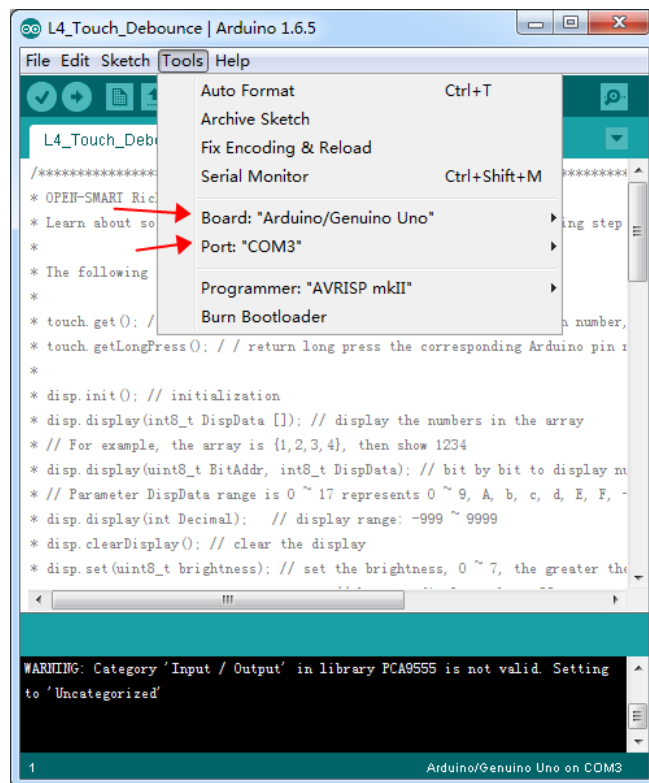
Double click the file in the directory of

\\arduino-1.6.5-r2\\libraries\\RichUNO\\examples\\L4_Touch_Debounce

So that you find the window below:

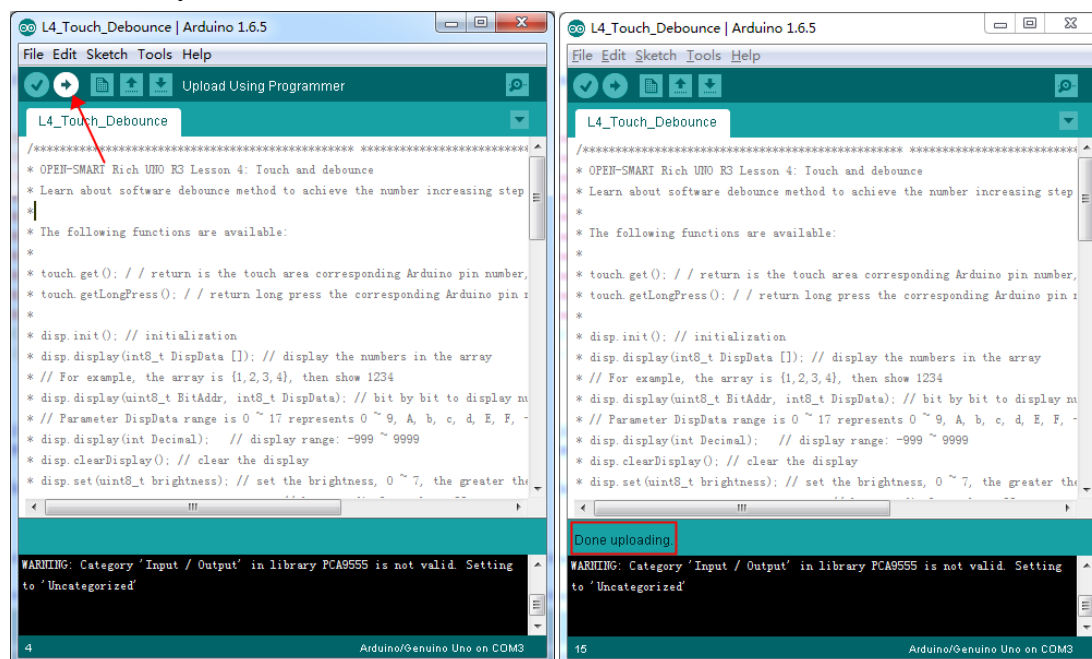


2) Check the Board and Serial Port



3) Click Upload button to upload the code to Rich UNO R3 board

When you see that “Done uploading”, congratulations, you have upload the code to the board successfully.



4) Check whether the board is running as you want

In this lesson, you can touch TCH1, and the number is incremented step by step.

8.3 Learn the functions

1) You can find that at the beginning of all lessons code is

```
#include <Wire.h>
```

```
#include <SoftwareSerial.h>
```

Because the RichUNO library files use Wire.h and SoftwareSerial.h which is not in the RichUNO directory, if the example code wants to use RichUNO library, it should add those two line.

2) Other header files

In lesson 4 we use 4-digit display and touch sensor, so it should include their header files to control display more easily.

So you should add:

```
#include "RichUNOTouchSensor.h"
```

```
#include "RichUNOTM1637.h"
```

3) Define class object

We have defined the display class and touch sensor class in the RichUNO library, and at any time you should define a object before you use it, and some object has parameter.

For example:

```
TouchSensor touch(3,4,5,6);//define a touch sensor object, and display connect to D3 / D4 / D5 / D6 of Rich UNO R3 board
```

4) Two necessary functions

Each example code should include the two necessary functions:

```
void setup(){ }
```

```
void loop(){} 
```

Usually we write initialization code in setup function, and then write the code what you want the board to do in loop function which runs cyclically.

5) Functions

** touch.get(); // return is the touch area corresponding Arduino pin number, if not then return -1*

If you touch TCH1, it will return 3. TCH2 is 4, TCH3 is 5, and TCH4 is 6.

You may try to write this function to get 1,2,3,4 for TCHx.

```
int get()
```

```
{
```

```
    int touchNum;
```

```
    int _pin;
```

```
    _pin = touch.get();
```

```
    switch(_pin)
```

```
    {
```

```
        case TOUCH_OUT1: touchNum = 1;break;
```

```
        case TOUCH_OUT2: touchNum = 2;break;
```

```
        case TOUCH_OUT3: touchNum = 3;break;
```

```
        case TOUCH_OUT4: touchNum = 4;break;
```

```
default: touchNum = _pin;break;  
}  
return touchNum;  
}
```

** disp.init(); // initialization*

You should write it in the setup function initialize it and it has set the default brightness to be 3(0~7, number larger, brightness higher).

** disp.display(int Decimal); // display range: -999 ~ 9999*

If your number is not in the range, it will ignore your operation.

9 Lesson 5: Temperature and display

9.1 Overview

Usually we need to know the temperature of many places, such as baby room, warehouse, greenhouse, incubator, fish tank.

LM75A temperature sensor we used is a high-speed I2C interface temperature sensor and it can converted the temperature into a digital signal directly in the temperature range of $-55^{\circ}\text{C} \sim +125^{\circ}\text{C}$. MCU can read data from its internal registers directly through the I2C bus, and operate the 4 data registers to set different operation modes. The module can not only measure the temperature but you can set the over-temperature protection temperature. It is widely used.

In this lesson, you can learn how to get temperature and display integer part of the temperature on the 4-digit tube.

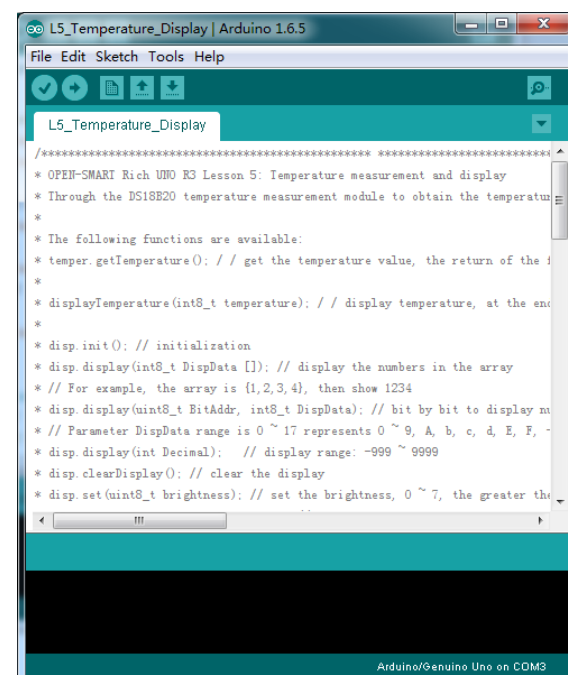
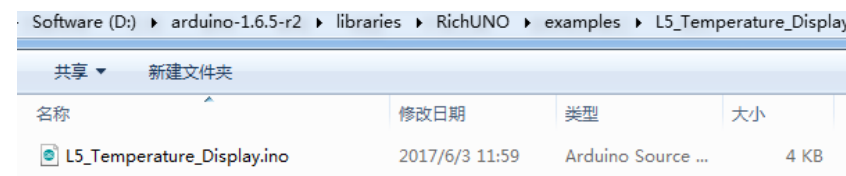
9.2 Upload code for lesson 5

1) Open lesson5 example

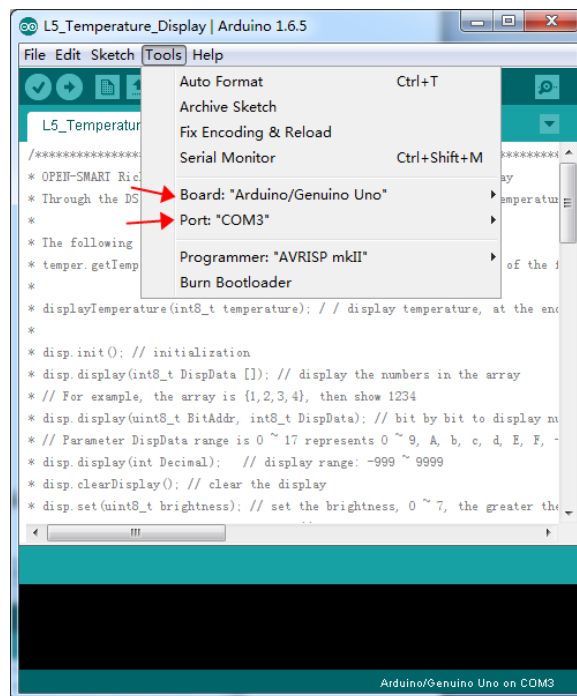
Double click the file in the directory of

\\arduino-1.6.5-r2\\libraries\\RichUNO\\examples\\L5_Temperature_Display

So that you find the window below:

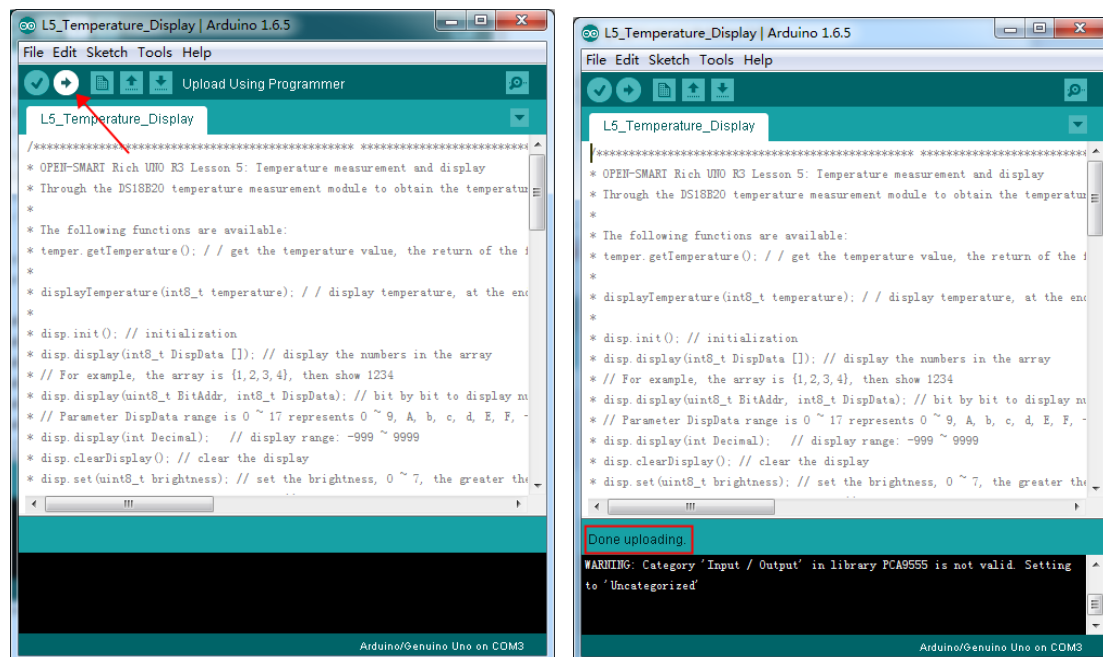


2) Check the Board and Serial Port



3) Click Upload button to upload the code to Rich UNO R3 board

When you see that “Done uploading”, congratulations, you have upload the code to the board successfully.



4) Check whether the board is running as you want

In this lesson, you can see it displays the temperature on the 4-digit tube, and the temperature unit is in degrees Celsius. You can cover the temperature sensor with your fingers, the temperature will rise.

9.3 Learn the functions

1) You can find that at the beginning of all lessons code is

```
#include <Wire.h>
```

```
#include <SoftwareSerial.h>
```

Because the RichUNO library files use Wire.h and SoftwareSerial.h which is not in the RichUNO directory, if the example code wants to use RichUNO library, it should add those two line.

2) Other header files

In lesson 5 we use 4-digit display and temperature sensor, so it should include their header files to control display more easily.

So you should add:

```
#include "RichUNOTM1637.h"
```

```
#include "RichUNOLM75.h"
```

3) Define class object

We have defined the display class and LM75 class in the RichUNO library, and at any time you should define a object before you use it, and some object has parameter and some does not.

For example:

```
TM1637 disp(10,11); //define a TM1637 object, and display connect to D10 / D11 of Rich UNO R3 board
```

```
LM75 temper; // define an LM75 object "temper" for temperature
```

4) Two necessary functions

Each example code should include the two necessary functions:

```
void setup(){ }
```

```
void loop(){} 
```

Usually we write initialization code in setup function, and then write the code what you want the board to do in loop function which runs cyclically.

5) Functions

** temper.getTemperature(); // get the temperature value, the return of the float is the decimal, unit is in degrees Celsius.*

For example:

```
float celsius;
```

```
celsius = temper.getTemperatue(); //get temperature
```

** disp.init(); // initialization*

You should write it in the setup function initialize it and it has set the default brightness to be 3(0~7, number larger, brightness higher).

** disp.display(int8_t DispData []); // display the numbers in the array*

** // For example, the array is {1,2,3,4}, then show 1234*

10 Lesson 6: Number flashing

10.1 Overview

4-digit 7-segment display with clock point can clearly display number in the sun so it is often use for watches / Clock / Induction Cooker.

In this lesson, learn how to use the internal timer method to make the displayed number flash.

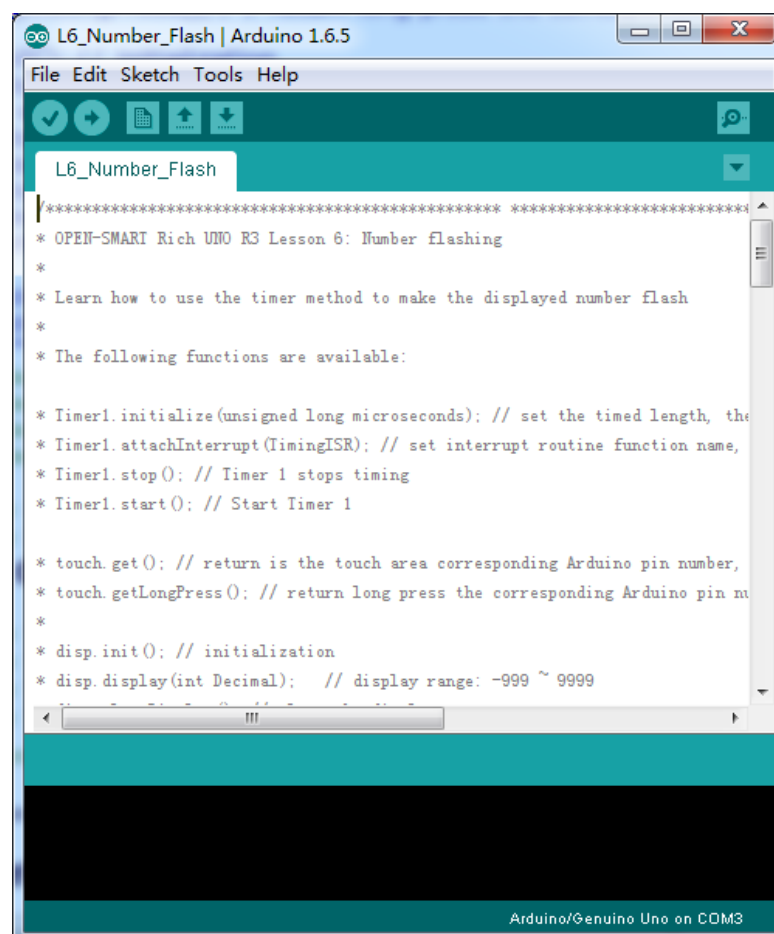
10.2 Upload code for lesson 6

1) Open lesson6 example

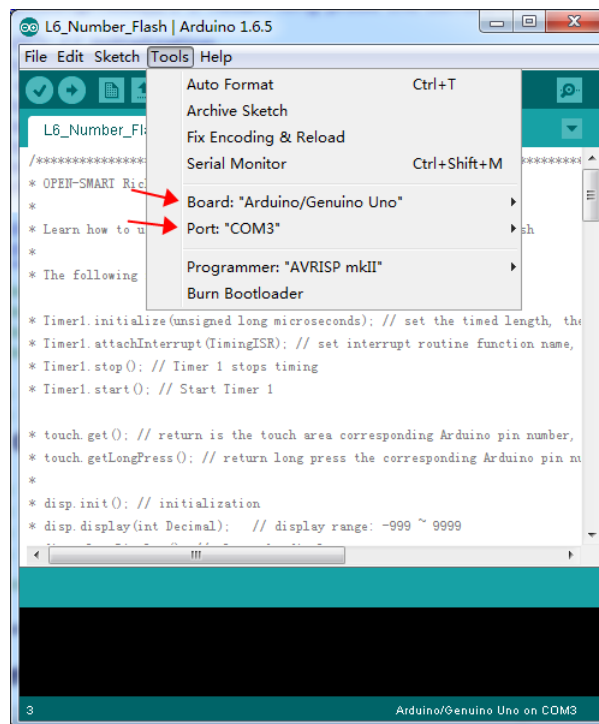
Double click the file in the directory of

\\arduino-1.6.5-r2\\libraries\\RichUNO\\examples\\L6_Number_Flash

So that you find the window below:

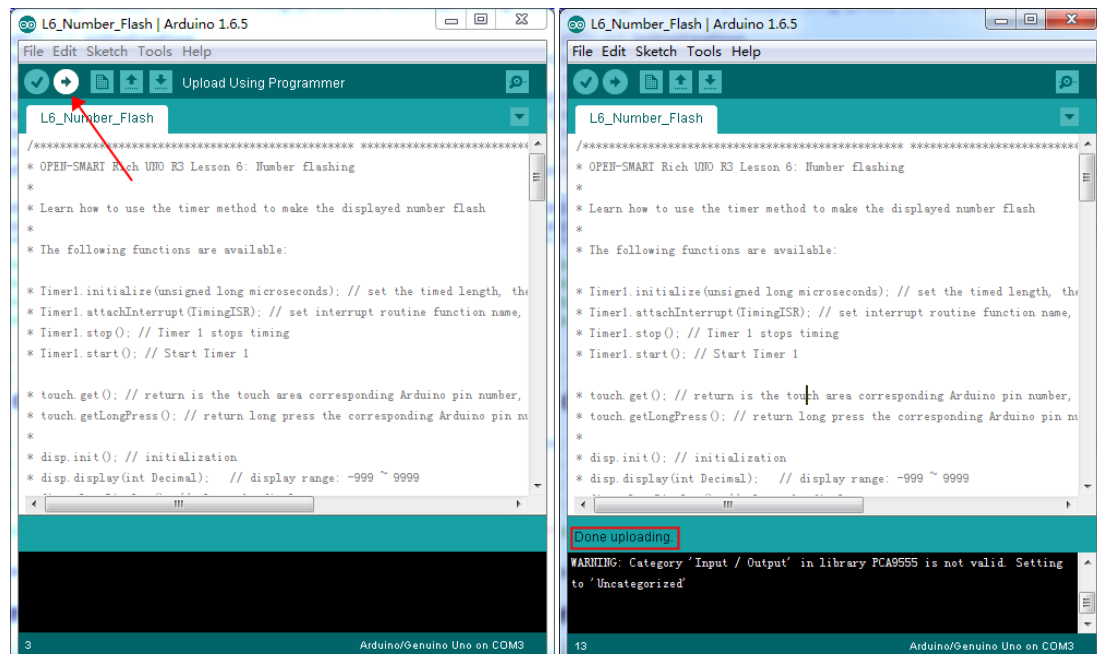


2) Check the Board and Serial Port



3) Click Upload button to upload the code to Rich UNO R3 board

When you see that “Done uploading”, congratulations, you have upload the code to the board successfully.



4) Check whether the board is running as you want

In this lesson, you can see it displays 2017 first, and when you touch TCH4, it is flashing, so you can change the number. Touch TCH1 to increase number and TCH2 to decrease number. When you touch TCH3, the number stops flashing and you can not change it.

10.3 Learn the functions

1) You can find that at the beginning of all lessons code is

```
#include <Wire.h>
```

```
#include <SoftwareSerial.h>
```

Because the RichUNO library files use Wire.h and SoftwareSerial.h which is not in the RichUNO directory, if the example code wants to use RichUNO library, it should add those two line.

2) Other header files

In lesson 6 we use 4-digit display, touch sensor and timer one, so it should include their header files to control display more easily.

So you should add:

```
#include <TimerOne.h>
```

```
#include "RichUNOTM1637.h"
```

```
#include "RichUNOTouchSensor.h"
```

3) Define class object

We have defined the display class in the RichUNO library, and at any time you should define a object before you use it, and some object has parameter. Timer1 is an object defined in <TimerOne.h>, you can use it directly.

For example:

```
TM1637 disp(10,11); //define a TM1637 object, and display connect to D10 / D11 of Rich UNO R3 board
```

4) Two necessary functions

Each example code should include the two necessary functions:

```
void setup(){ }
```

```
void loop(){} 
```

Usually we write initialization code in setup function, and then write the code what you want the board to do in loop function which runs cyclically.

5) Functions

```
* Timer1.initialize(unsigned long microseconds); // set the timed length, the unit is microseconds
```

```
* Timer1.attachInterrupt(TimingISR); // set interrupt routine function name, is the timing interrupt entry, when the time is up, it will call TimingISR function you write.
```

```
* Timer1.stop(); // Timer 1 stops timing
```

```
* Timer1.start(); // Timer 1 start
```

```
* touch.get(); // return is the touch area corresponding Arduino pin number, if not then return -1
```

If you touch TCH1, it will return 3. TCH2 is 4, TCH3 is 5, and TCH4 is 6.

You may try to write this function to get 1,2,3,4 for TCHx.

```
int get()
```

```
{
```

```
    int touchNum;
```

```
int _pin;
_pin = touch.get();
switch(_pin)
{
    case TOUCH_OUT1: touchNum = 1;break;
    case TOUCH_OUT2: touchNum = 2;break;
    case TOUCH_OUT3: touchNum = 3;break;
    case TOUCH_OUT4: touchNum = 4;break;
    default: touchNum = _pin;break;
}
return touchNum;
}

* disp.init(); // initialization
  You should write it in the setup function initialize it.

* disp.display(int Decimal); // display range: -999 ~ 9999

* disp.clearDisplay(); // clear the display, nothing display
```

11 Lesson 7: Knob angle display

11.1 Overview

The knob angle can tell you the angle between 0 with 280 degrees. It is usually used to control volume of speaker, voltage of power and speed of motors.

In this lesson, you can learn get the angle of the knob(rotary angle sensor) and display on the 4-digit display.

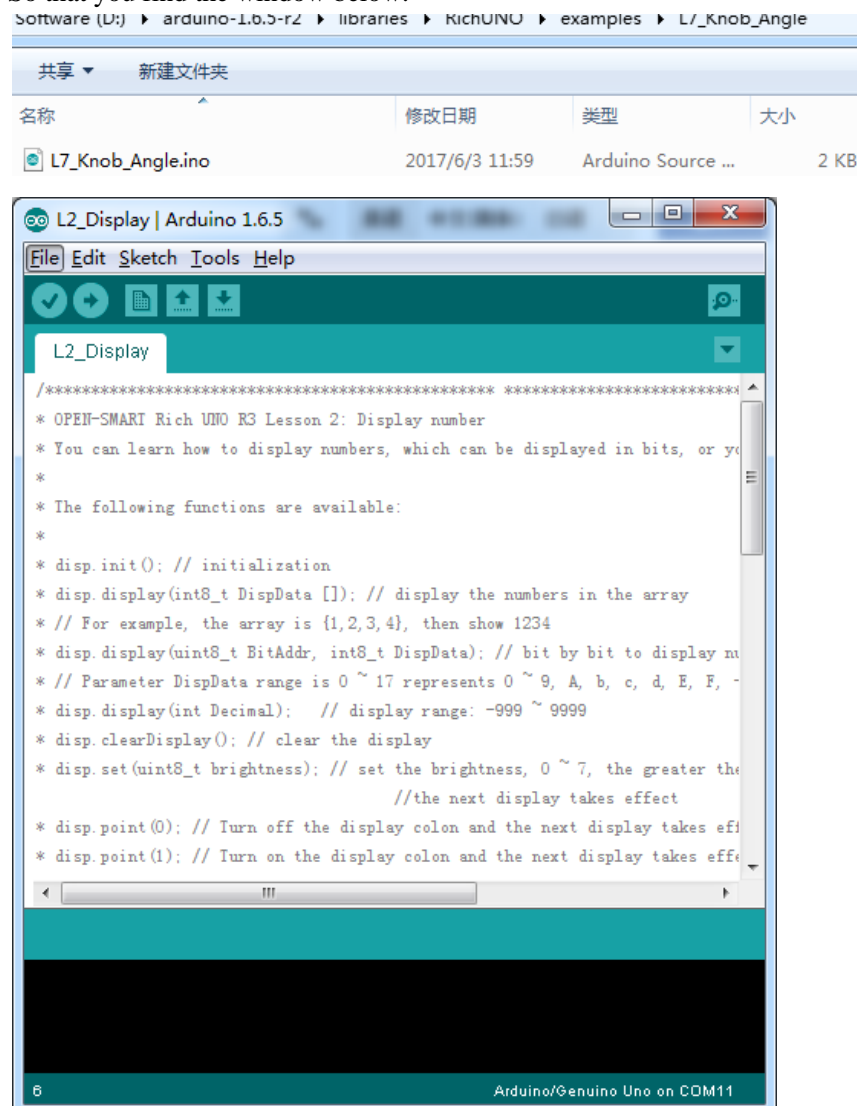
11.2 Upload code for lesson 7

1) Open lesson7 example

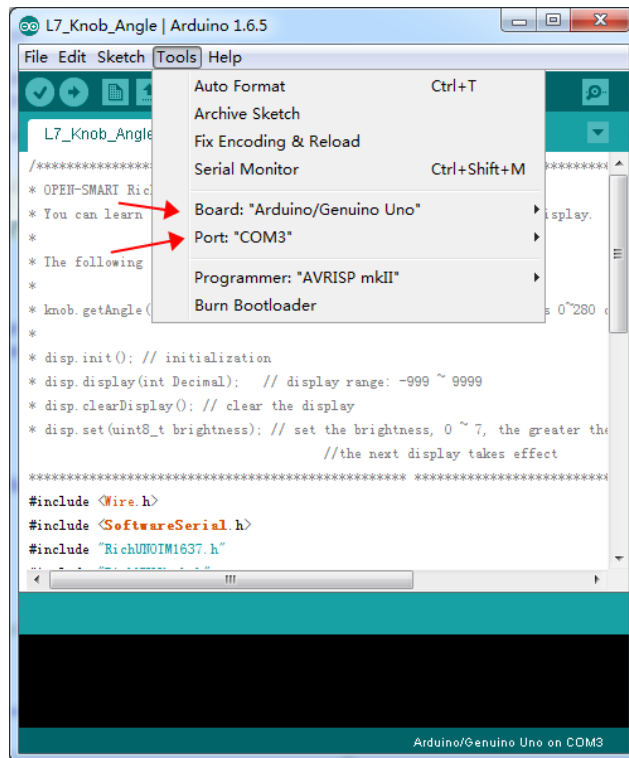
Double click the file in the directory of

\\arduino-1.6.5-r2\\libraries\\RichUNO\\examples\\L7_Knob_Angle

So that you find the window below:

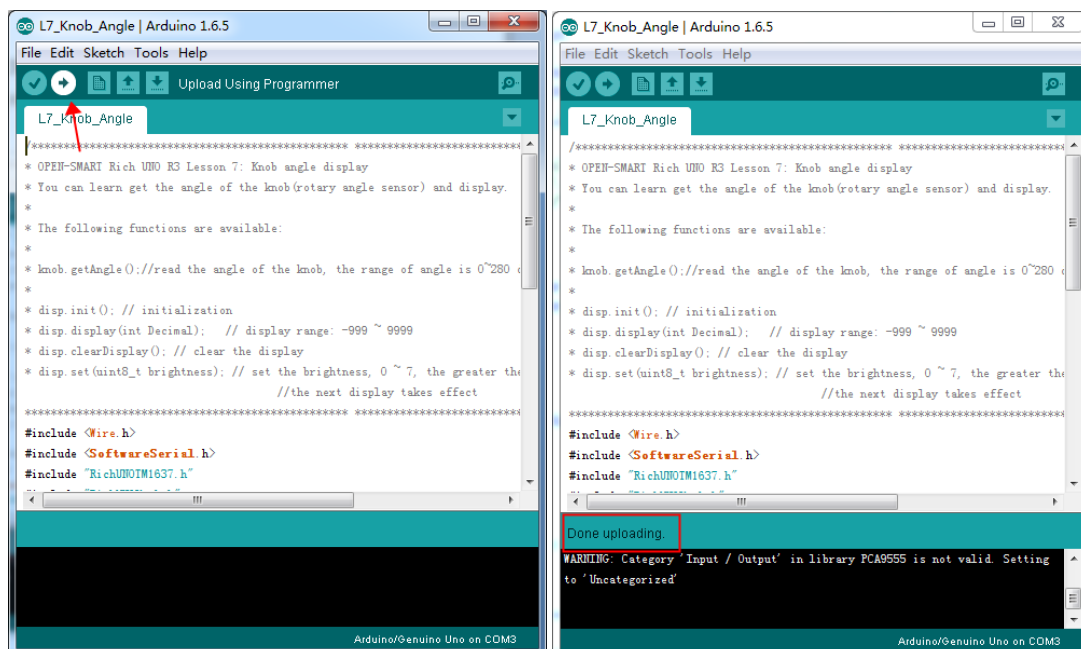


2) Check the Board and Serial Port



3) Click Upload button to upload the code to Rich UNO R3 board

When you see that “Done uploading”, congratulations, you have upload the code to the board successfully.



4) Check whether the board is running as you want

In this lesson, you can see it displays the angle when you rotate the knob. When the nick goes to the min position, it displays 0. When the nick goes to the other way, it displays max angle 280.

11.3 Learn the functions

1) You can find that at the beginning of all lessons code is

```
#include <Wire.h>
```

```
#include <SoftwareSerial.h>
```

Because the RichUNO library files use Wire.h and SoftwareSerial.h which is not in the RichUNO directory, if the example code wants to use RichUNO library, it should add those two line.

2) Other header files

In lesson 7 we use 4-digit display, so it should include their header files to control display more easily.

So you should add:

```
#include "RichUNOTM1637.h"
```

```
#include "RichUNOKnob.h"
```

3) Define class object

We have defined the display class in the RichUNO library, and at any time you should define a object before you use it, and some object has parameter.

For example:

```
TM1637 disp(10,11); //define a TM1637 object, and display connect to D10 / D11 of Rich UNO R3 board
```

```
Knob knob(A0); //define a Knob object, the signal of knob connect to A0 of Rich UNO R3
```

4) Two necessary functions

Each example code should include the two necessary functions:

```
void setup(){ }
```

```
void loop(){} 
```

Usually we write initialization code in setup function, and then write the code what you want the board to do in loop function which runs cyclically.

5) Functions

** knob.getAngle(); //read the angle of the knob, the range of angle is 0~280 degrees*

For example:

```
int angle;
```

```
angle = knob.getAngle();
```

So you can get the knob angle.

** disp.init(); // initialization*

You should write it in the setup function initialize it.

** disp.display(int Decimal); // display range: -999 ~ 9999*

12 Lesson 8: Knob control brightness

12.1 Overview

The knob angle can tell you the angle between 0 with 280 degrees. It is usually used to control brightness of display, volume of speaker, voltage of power and speed of motors.

In this lesson, you can learn to use the knob(rotary angle sensor) to control the brightness of the display.

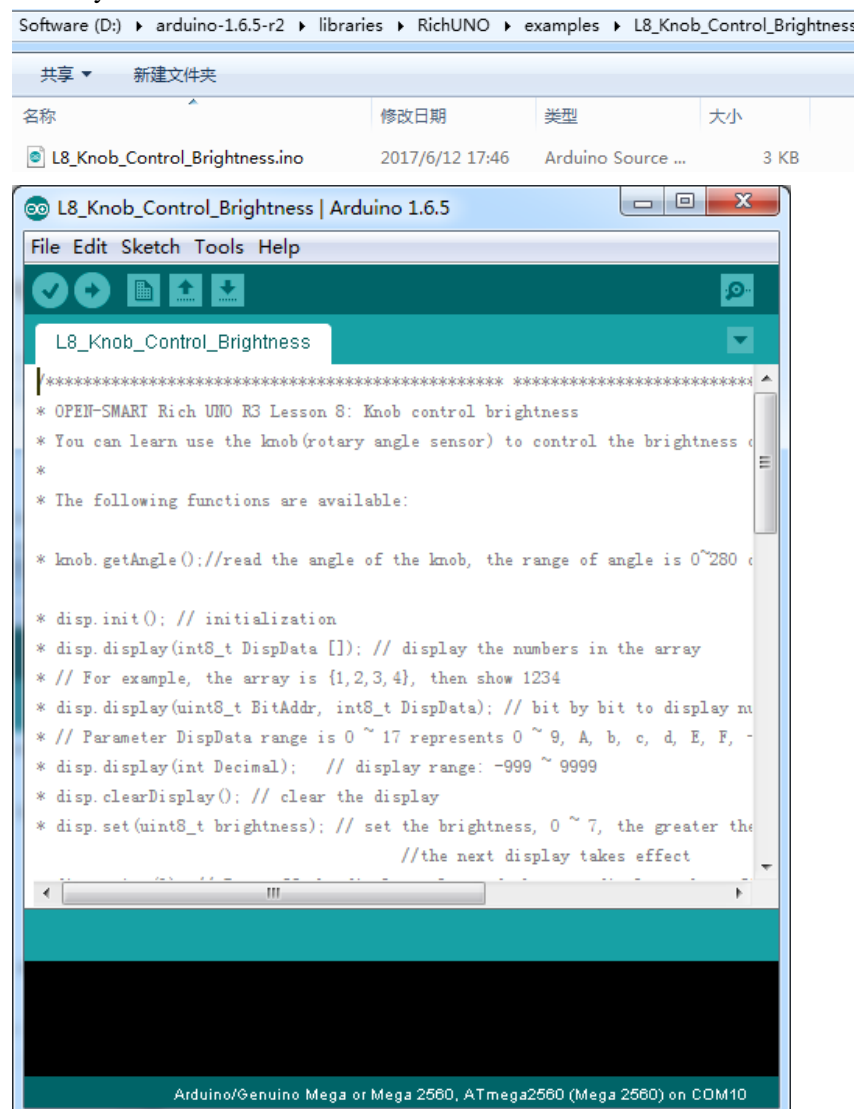
12.2 Upload code for lesson 8

1) Open lesson8 example

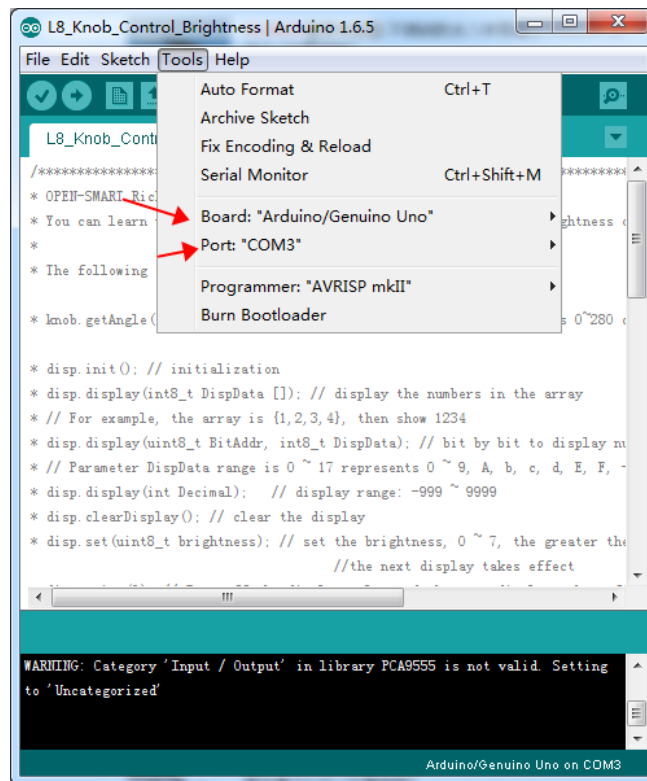
Double click the file in the directory of

\\arduino-1.6.5-r2\\libraries\\RichUNO\\examples\\L8_Knob_Control_Brightness

So that you find the window below:

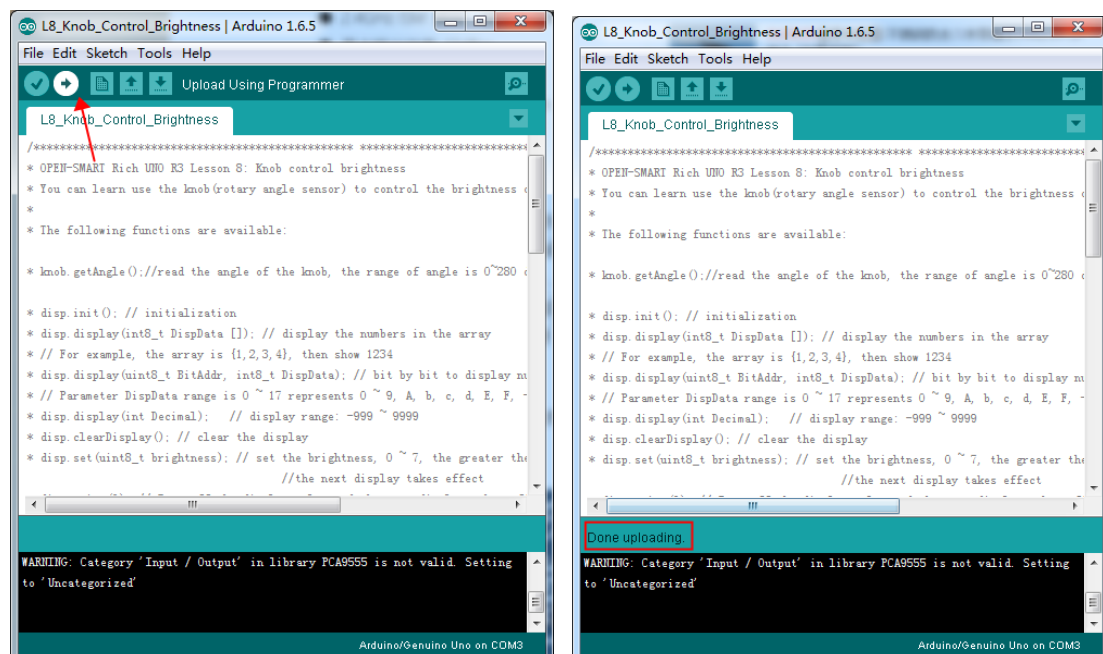


2) Check the Board and Serial Port



3) Click Upload button to upload the code to Rich UNO R3 board

When you see that “Done uploading”, congratulations, you have upload the code to the board successfully.



4) Check whether the board is running as you want

In this lesson, you can see it displays 88:88 and when you rotate clockwise, the brightness becomes high, otherwise the brightness decreases

12.3 Learn the functions

1) You can find that at the beginning of all lessons code is

```
#include <Wire.h>
```

```
#include <SoftwareSerial.h>
```

Because the RichUNO library files use Wire.h and SoftwareSerial.h which is not in the RichUNO directory, if the example code wants to use RichUNO library, it should add those two line.

2) Other header files

In lesson 8 we use 4-digit display, so it should include their header files to control display more easily.

So you should add:

```
#include "RichUNOTM1637.h"
```

```
#include "RichUNOKnob.h"
```

3) Define class object

We have defined the display class in the RichUNO library, and at any time you should define a object before you use it, and some object has parameter.

For example:

```
TM1637 disp(10,11); //define a TM1637 object, and display connect to D10 / D11 of Rich UNO R3 board
```

```
Knob knob(A0); //define a Knob object, the signal of knob connect to A0 of Rich UNO R3
```

4) Two necessary functions

Each example code should include the two necessary functions:

```
void setup(){ }
```

```
void loop(){} 
```

Usually we write initialization code in setup function, and then write the code what you want the board to do in loop function which runs cyclically.

5) Functions

```
* knob.getAngle(); //read the angle of the knob, the range of angle is 0~280 degrees
```

For example:

```
int angle;
```

```
angle = knob.getAngle();
```

So you can get the knob angle.

```
* disp.init(); // initialization
```

You should write it in the setup function initialize it.

```
* disp.display(int8_t DispData []); // display the numbers in the array
```

// For example, the array is {1,2,3,4}, then show 1234

```
* disp.set(uint8_t brightness); // set the brightness, 0 ~ 7, the greater the value, the higher the  
//brightness, the next display takes effect
```

** disp.point(0); // Turn off the display colon and the next display takes effect*
** disp.point(1); // Turn on the display colon and the next display takes effect*

13 Lesson 9: Clock input

13.1 Overview

Note: Make sure you have plugged the CR1220 battery onto the battery holder. In the Rich UNO R3 Kit A, the battery comes with the kit. If you do not have one, please go to buy or get one.

In this lesson, you can learn to use the touch button to enter the year, month, day, day of week, hour, minute, to initialize the clock module data, and you can see whether your setting is ok from the serial monitor of Arduino IDE and it will tell you what you are setting.

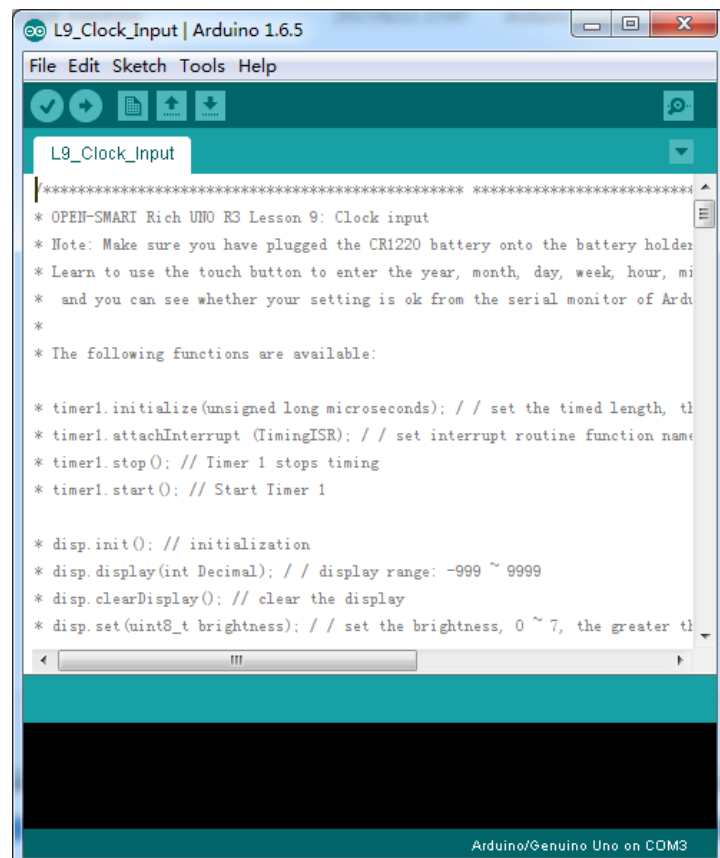
13.2 Upload code for lesson 9

1) Open lesson9 example

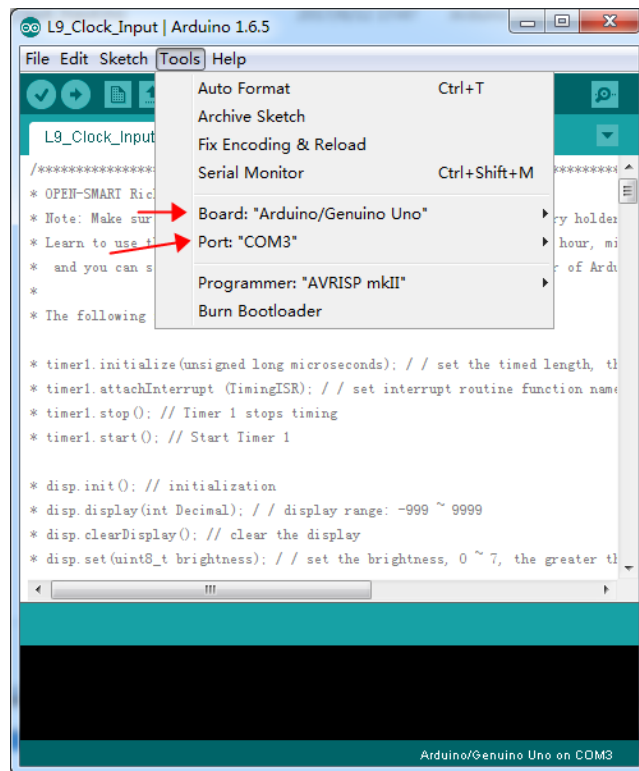
Double click the file in the directory of

\arduino-1.6.5-r2\libraries\RichUNO\examples\L9_Clock_Input

So that you find the window below:

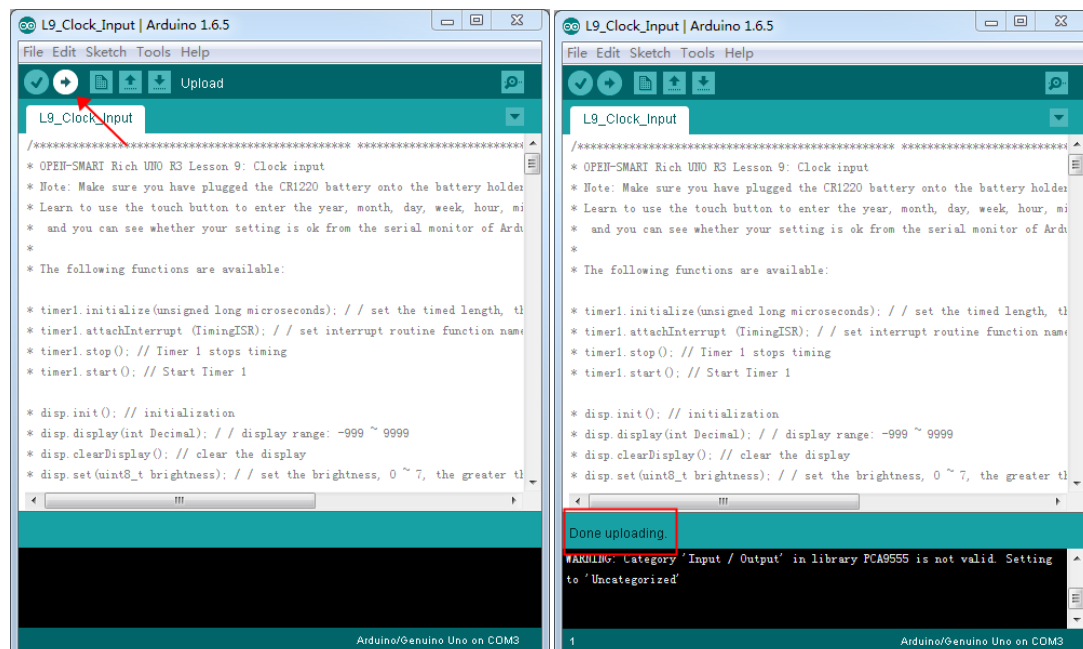


2) Check the Board and Serial Port



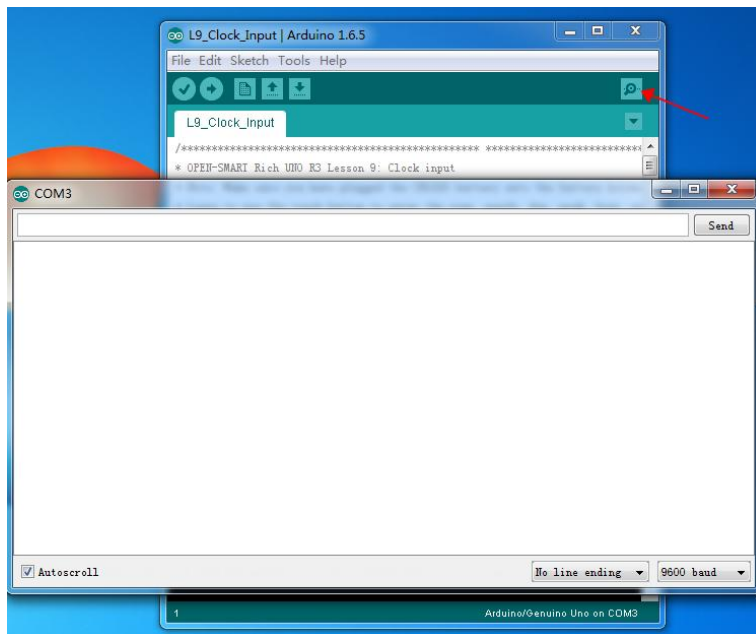
3) Click Upload button to upload the code to Rich UNO R3 board

When you see that “Done uploading”, congratulations, you have upload the code to the board successfully.



4) Check whether the board is running as you want

Firstly, click the serial monitor button in the upper right corner, then you can see the serial monitor window.



Secondly, you can see the board displays 2017. Now you should know that

TCH1 area corresponds to the number increase

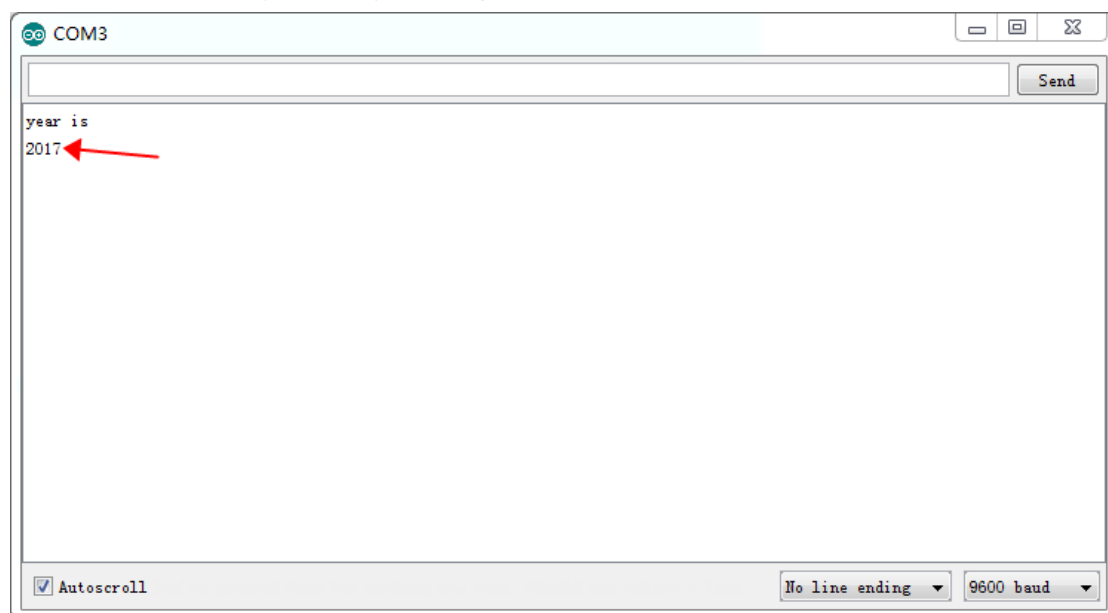
TCH2 area corresponds to the number decrease

TCH3 area corresponds to confirm the number you set (OK Button)

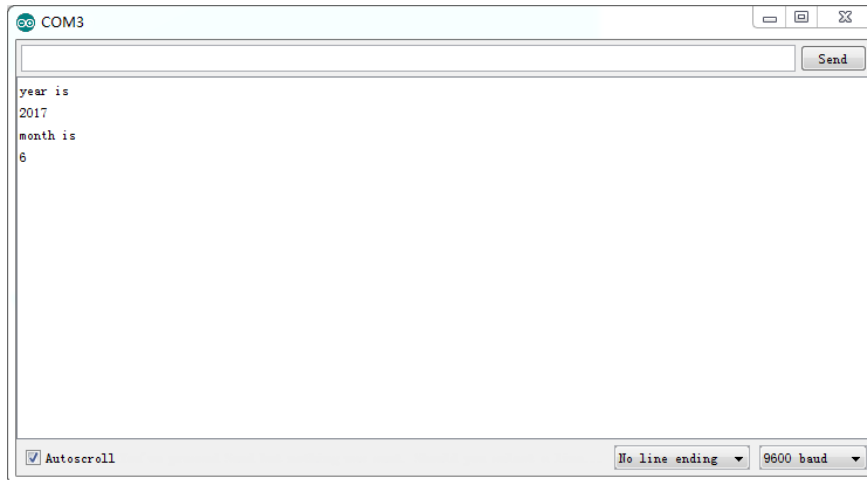
TCH4 area corresponds to make it enter next step("next step button")

Every time you have set the date and time, you should touch OK button to confirm it and touch next step button to set next item.

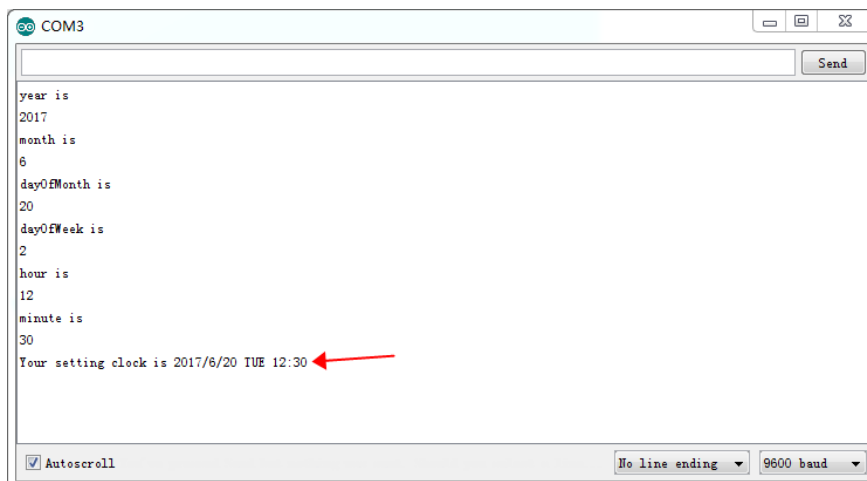
Thirdly, touch TCH4 to set year, and if you have set it ok, please touch TCH3 to confirm and the serial monitor will tell you what you have just set.



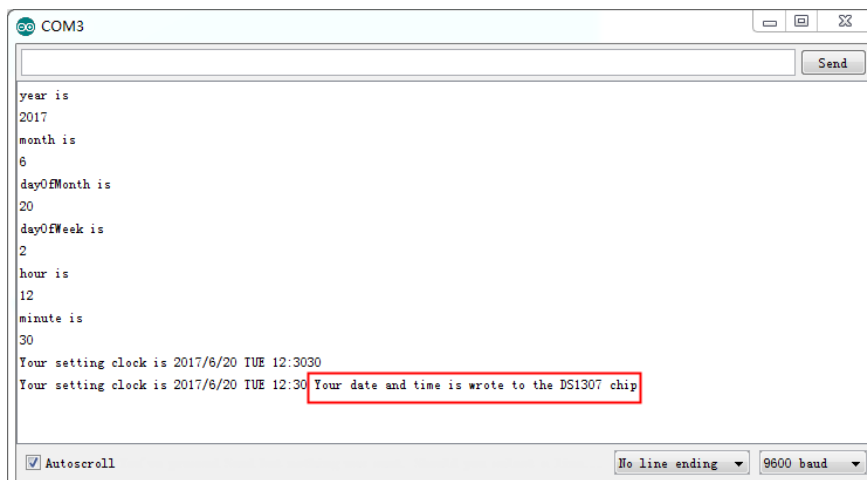
Then touch TCH4 again to set month, you can touch TCH1 to increase the number and TCH2 to decrease and if you have set it ok, please touch TCH3 to confirm and the serial monitor will tell you what you have just set.



Again and again, then day of month, day of week, hour, minute, all the six items have been set, the monitor will tell you what date you set.



And then touch TCH3 to confirm the date and it will update the date to the DS1307 chip.



In this lesson, you can see the date and time have been set successfully.

13.3 Learn the functions

1) You can find that at the beginning of all lessons code is

```
#include <Wire.h>
```

```
#include <SoftwareSerial.h>
```

Because the RichUNO library files use Wire.h and SoftwareSerial.h which is not in the RichUNO directory, if the example code wants to use RichUNO library, it should add those two line.

2) Other header files

In lesson 9 we use 4-digit display, touch sensor and DS1307 clock, so it should include their header files to control more easily.

So you should add:

```
#include <TimerOne.h>
```

```
#include "RichUNOTM1637.h"
```

```
#include "RichUNOTouchSensor.h"
```

```
#include "RichUNODS1307.h"
```

3) Define class object

We have defined the basic class in the RichUNO library, and at any time you should define a object before you use it, and some object has parameter.

For example:

```
TM1637 disp(10,11); //define a TM1637 object, and display connect to D10 / D11 of Rich UNO R3
```

```
DS1307 clock; //define a object of DS1307 class, it connects to I2C port of Rich UNO R3
```

4) Two necessary functions

Each example code should include the two necessary functions:

```
void setup(){ }
```

```
void loop(){} 
```

Usually we write initialization code in setup function, and then write the code what you want the board to do in loop function which runs cyclically.

5) Functions

```
* clock.begin(); // The clock is on and the function must be called first
```

```
* clock.startClock(void); // clock start, if you stop the clock need to call this function
```

```
* clock.stopClock(void); // clock stop timing
```

```
* clock.getTime(void); // read the date and time, will be saved in the clock class variable
```

```
* clock.fillByHMS(uint8_t _hour, uint8_t _minute, uint8_t _second); // write time
```

```
* clock.fillByYMD(uint16_t _year, uint8_t _month, uint8_t _day); // write date
```

```
* clock.fillDayOfWeek(uint8_t _dow); // write day of week
```

```
* clock.setTime(void); // write the clock and time that have been written by filling functions to the  
clock chip
```

```
* Timer1.initialize(unsigned long microseconds); // set the timed length, the unit is microseconds
```

```
* Timer1.attachInterrupt(TimingISR); // set interrupt routine function name, is the timing interrupt  
entry, when the time is up, it will call TimingISR function you write.
```

** Timer1.stop(); // Timer 1 stops timing*

** Timer1.start(); // Timer 1 start*

** touch.get(); // return is the touch area corresponding Arduino pin number, if not then return -1*

If you touch TCH1, it will return 3. TCH2 is 4, TCH3 is 5, and TCH4 is 6.

You may try to write this function to get 1,2,3,4 for TCHx.

int get()

{

int touchNum;

int _pin;

_pin = touch.get();

switch(_pin)

{

case TOUCH_OUT1: touchNum = 1;break;

case TOUCH_OUT2: touchNum = 2;break;

case TOUCH_OUT3: touchNum = 3;break;

case TOUCH_OUT4: touchNum = 4;break;

default: touchNum = _pin;break;

}

return touchNum;

}

** disp.init(); // initialization*

You should write it in the setup function initialize it.

** disp.display(int Decimal); // display range: -999 ~ 9999*

** disp.clearDisplay(); // clear the display, nothing display*

14 Lesson 10: Clock display

14.1 Overview

All of us are very concerned about the time to wake you up, time You can use 4-digit tube to display the clock and of course you can use our [2.2 inch TFT shield](#) or [2.8 inch TFT Shield](#).

In this lesson, you can learn how to get the date and time from DS1307 module and display time on the 4-digit display.

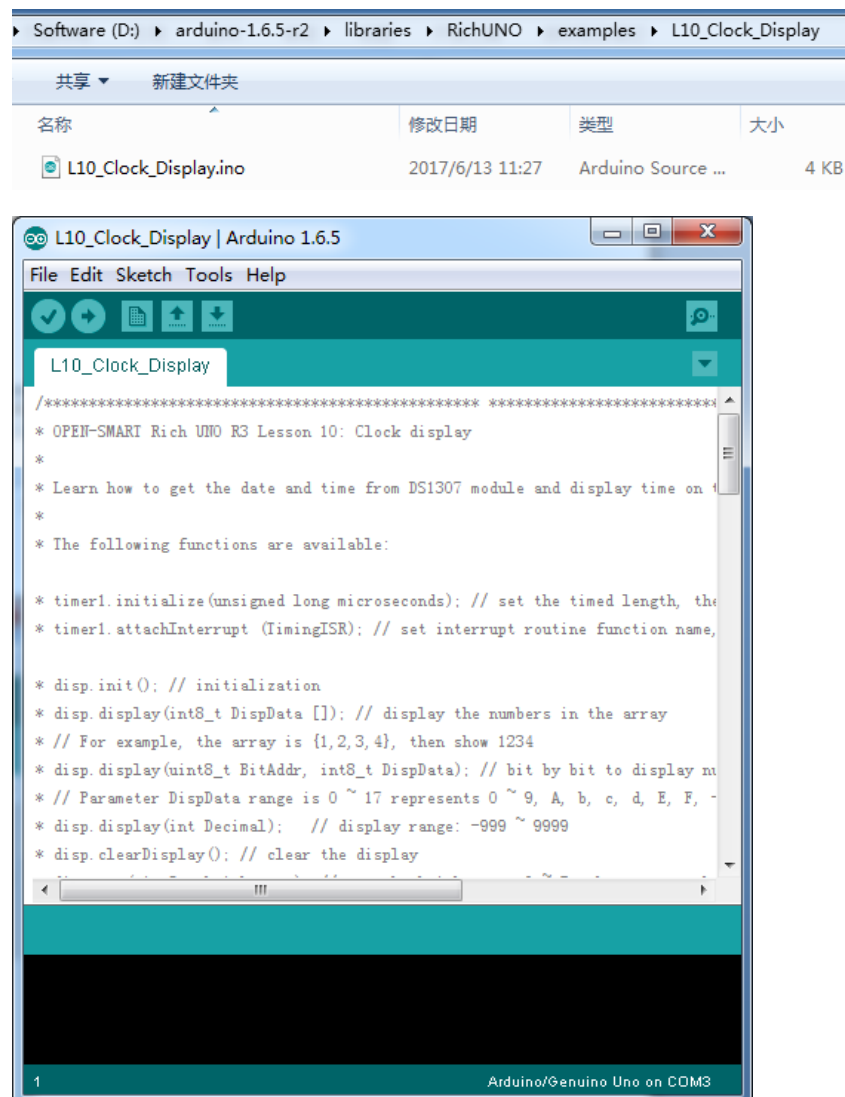
14.2 Upload code for lesson 10

1) Open lesson10 example

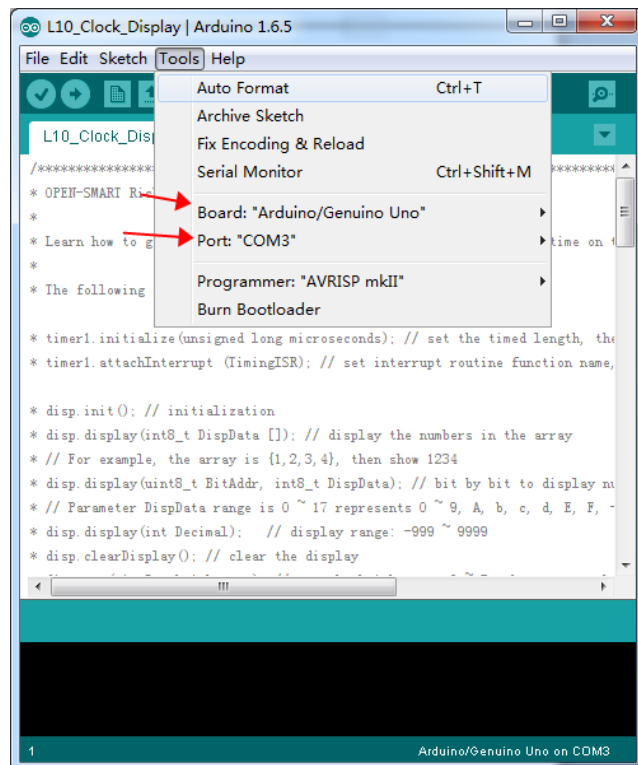
Double click the file in the directory of

\arduino-1.6.5-r2\libraries\RichUNO\examples\ L10_Clock_Display

So that you find the window below:

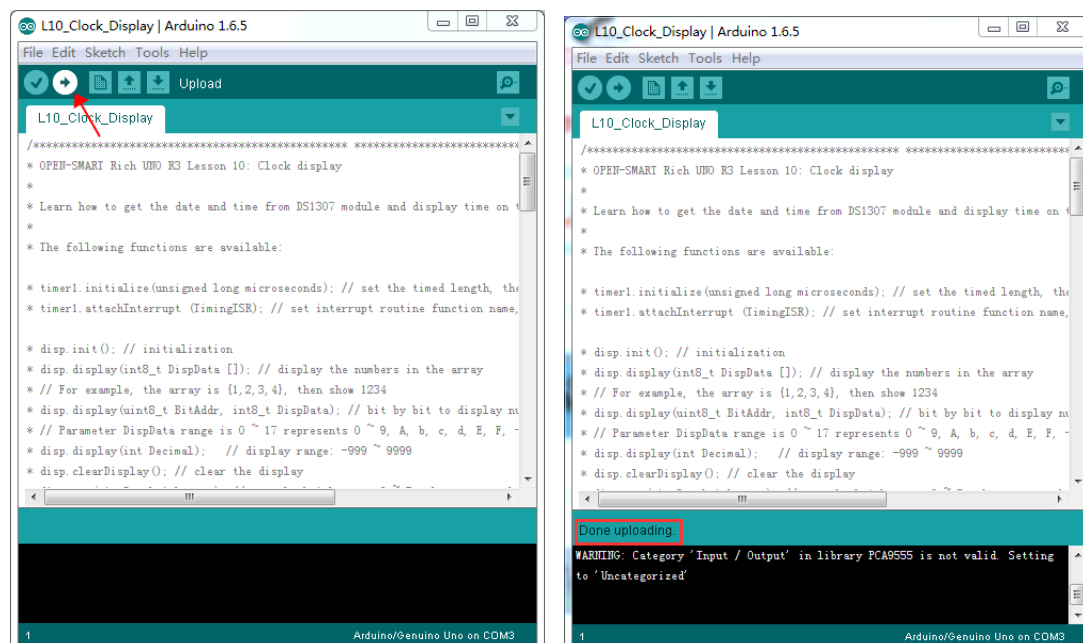


2) Check the Board and Serial Port



3) Click Upload button to upload the code to Rich UNO R3 board

When you see that “Done uploading”, congratulations, you have upload the code to the board successfully.



4) Check whether the board is running as you want

In this lesson, you can see it displays the time and the clock point is flashing every second.

14.3 Learn the functions

1) You can find that at the beginning of all lessons code is

```
#include <Wire.h>
```

```
#include <SoftwareSerial.h>
```

Because the RichUNO library files use Wire.h and SoftwareSerial.h which is not in the RichUNO directory, if the example code wants to use RichUNO library, it should add those two line.

2) Other header files

In lesson 10 we use DS1307 clock and 4-digit display, so it should include their header files to control display more easily.

So you should add:

```
#include <TimerOne.h>
```

```
#include "RichUNOTM1637.h"
```

```
#include "RichUNODS1307.h"
```

3) Define class object

We have defined the class in the RichUNO library, and at any time you should define a object before you use it, and some object has parameter and some does not.

For example:

```
TM1637 disp(10,11); //define a TM1637 object, and display connect to D10 / D11 of Rich UNO R3 board
```

```
DS1307 clock; //define a object of DS1307 class
```

4) Two necessary functions

Each example code should include the two necessary functions:

```
void setup(){ }
```

```
void loop(){ }
```

Usually we write initialization code in setup function, and then write the code what you want the board to do in loop function which runs cyclically.

5) Functions

** clock.begin(); // The clock is on and the function must be called first*

** clock.getTime(void); // read the date and time, will be saved in the clock class variable*

** Timer1.initialize(unsigned long microseconds); // set the timed length, the unit is microseconds*

** Timer1.attachInterrupt(TimingISR); // set interrupt routine function name, is the timing interrupt entry, when the time is up, it will call TimingISR function you write.*

** disp.init(); // initialization. You should write it in the setup function initialize it.*

** disp.display(int8_t DispData []); // display the numbers in the array
// For example, the array is {1,2,3,4}, then show 1234*

** disp.point(0); // Turn off the display colon and the next display takes effect*

** disp.point(1); // Turn on the display colon and the next display takes effect*

15 Lesson 11: Test infrared remote keys

15.1 Overview

Almost all the remote control fans, air conditioners, televisions use infrared remote control.

Infrared reception angle is small, not susceptible to interference.

In this lesson, you can learn how to get keys code of the buttons on the IR remote and display on the serial monitor of Arduino IDE.

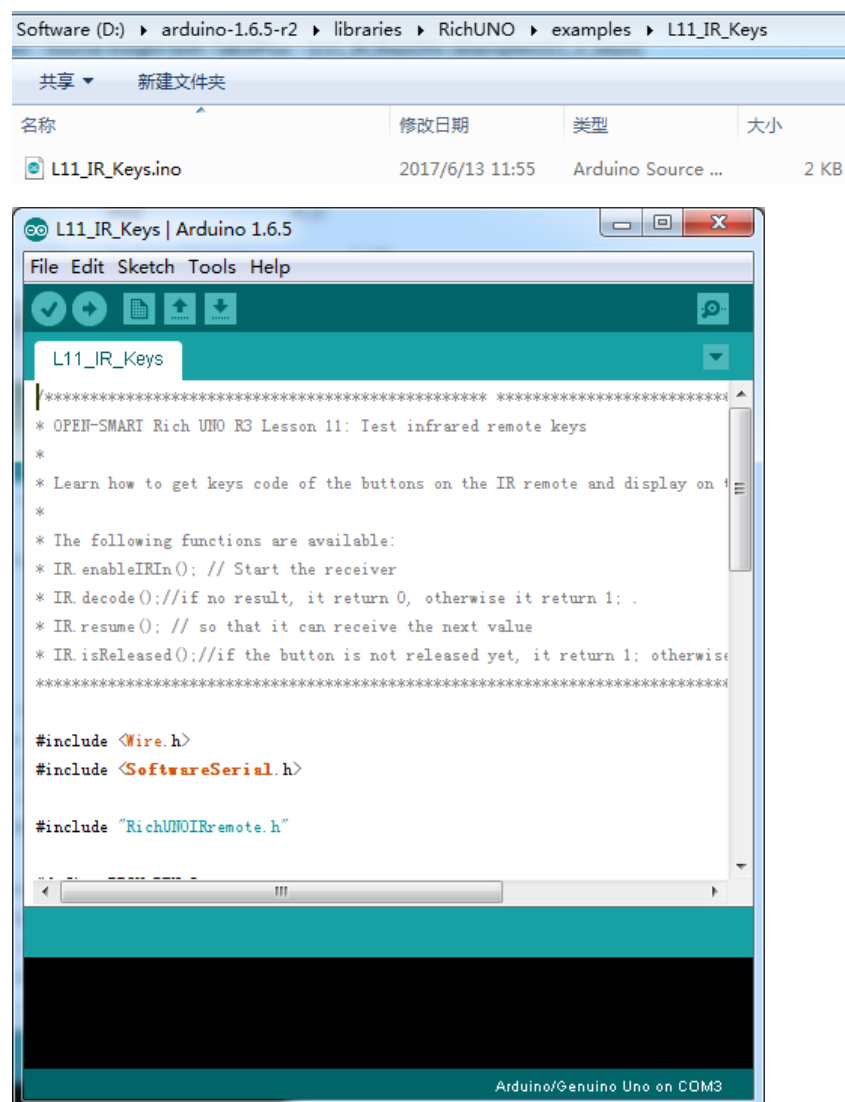
15.2 Upload code for lesson 11

1) Open lesson11 example

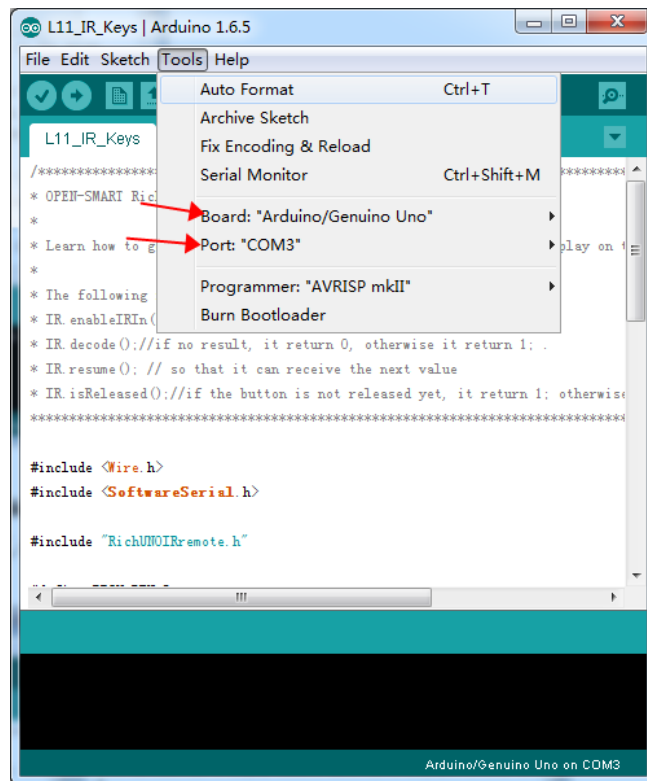
Double click the file in the directory of

\\arduino-1.6.5-r2\\libraries\\RichUNO\\examples\\ L11_IR_Keys

So that you find the window below:

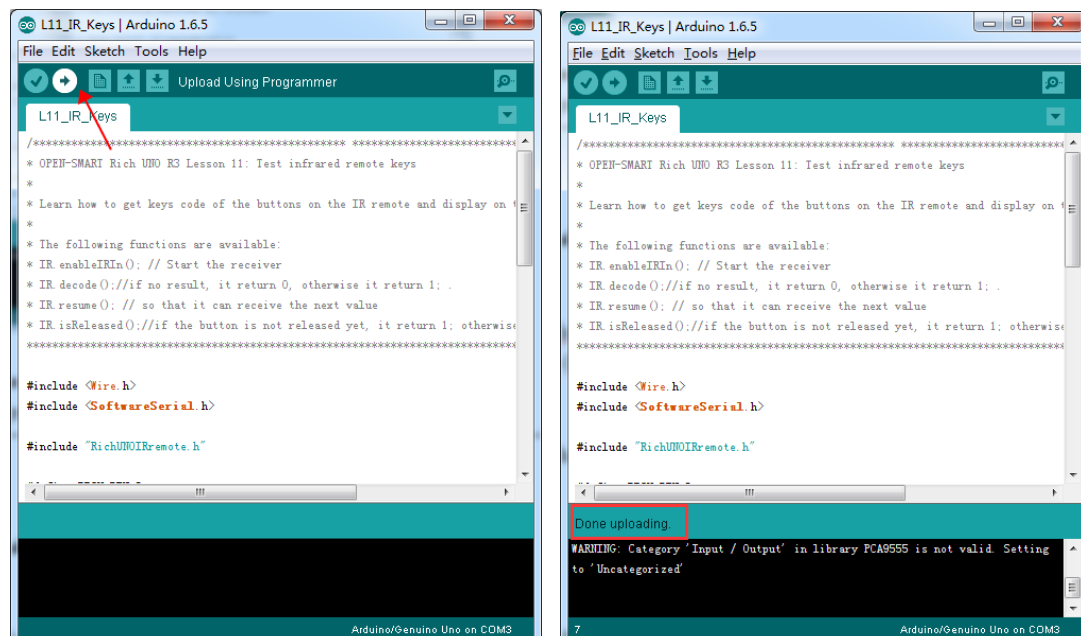


2) Check the Board and Serial Port



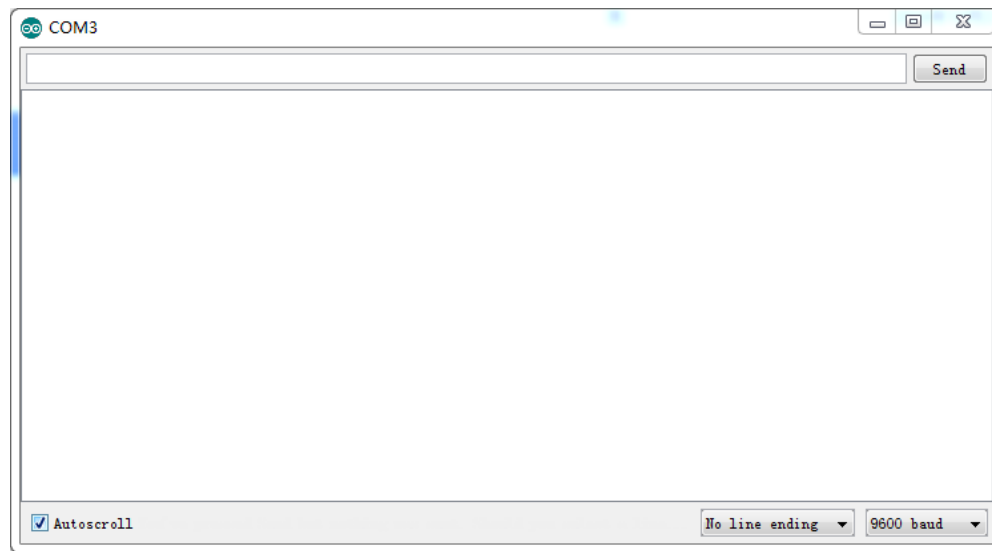
3) Click Upload button to upload the code to Rich UNO R3 board

When you see that “Done uploading”, congratulations, you have upload the code to the board successfully.

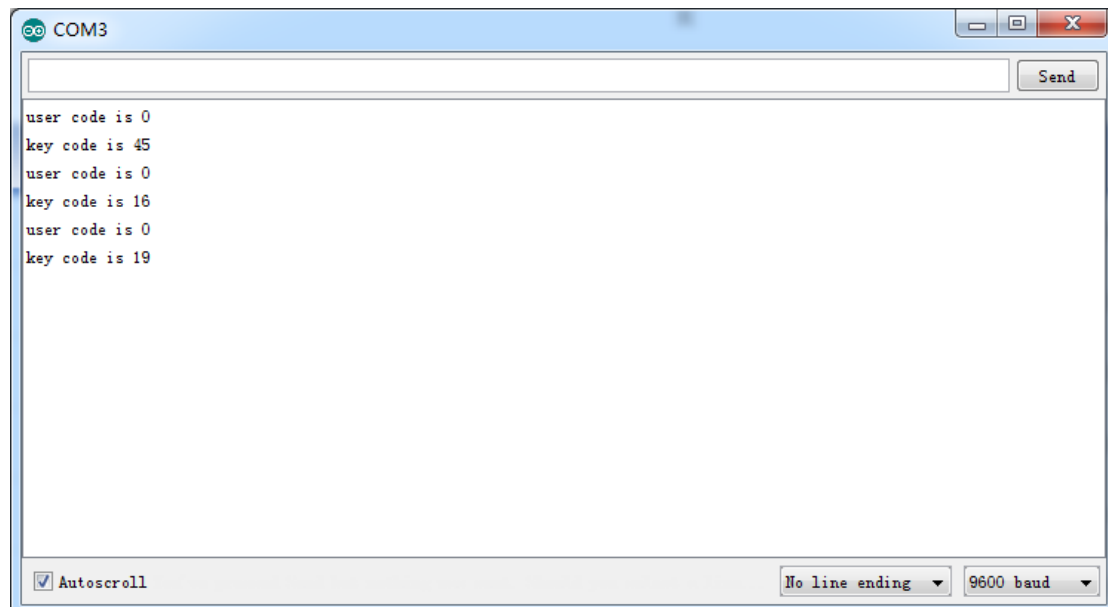


4) Check whether the board is running as you want

Firstly, click the serial monitor button in the upper right corner, then you can see the serial monitor window.



In this lesson, you can refer to the IR remote key code.jpg from [here](#). You can see that when press the button on the IR remote, the serial monitor will output the user code of the remote and key code of the button. So you can get what button has been pressed according to the key code and then do some operations.



15.3 Learn the functions

1) You can find that at the beginning of all lessons code is

```
#include <Wire.h>
```

```
#include <SoftwareSerial.h>
```

Because the RichUNO library files use Wire.h and SoftwareSerial.h which is not in the RichUNO directory, if the example code wants to use RichUNO library, it should add those two line.

2) Other header files

In lesson 11 we use IR remote, so it should include their header files to control remote more easily.

So you should add:

```
#include "RichUNOIRremote.h"
```

3) Define class object

We have defined the IR remote class in the RichUNO library, and at any time you should define a object before you use it, and some object has parameter.

For example:

```
IRrecv IR(2); //define an IR receiver object to get the key code, the receiver pin connect to D2 of Rich UNO R3
```

4) Two necessary functions

Each example code should include the two necessary functions:

```
void setup(){ }  
void loop(){}
```

Usually we write initialization code in setup function, and then write the code what you want the board to do in loop function which runs cyclically.

5) Functions

- * *IR.enableIRIn(); // Start the receiver*
- * *IR.decode(); //if no result, it return 0, otherwise it return 1; .*
- * *IR.resume(); // so that it can receive the next value*
- * *IR.isReleased(); //if the button is not released yet, it return 1; otherwise it return 0;*
- * *IR.keycode // this will return the value of key code you just press*
- * *IR.usercode // this will return the value of user code you just press*

- * *Serial.begin(9600); //set serial output baud rate is 9600bps, so your serial monitor should be the same, and should be called at the setup() function.*
- * *Serial.print("key code is "); //print the string "key code is" on the serial monitor*
- * *Serial.println(IR.keycode,HEX); // print the string on the serial monitor and line feed*

16 Lesson 12: Infrared remote control and display

16.1 Overview

Almost all the remote control fans, air conditioners, televisions use infrared remote control.

Infrared reception angle is small, not susceptible to interference.

In this lesson, how to recognize the key on the IR remote control and display the number on the display.

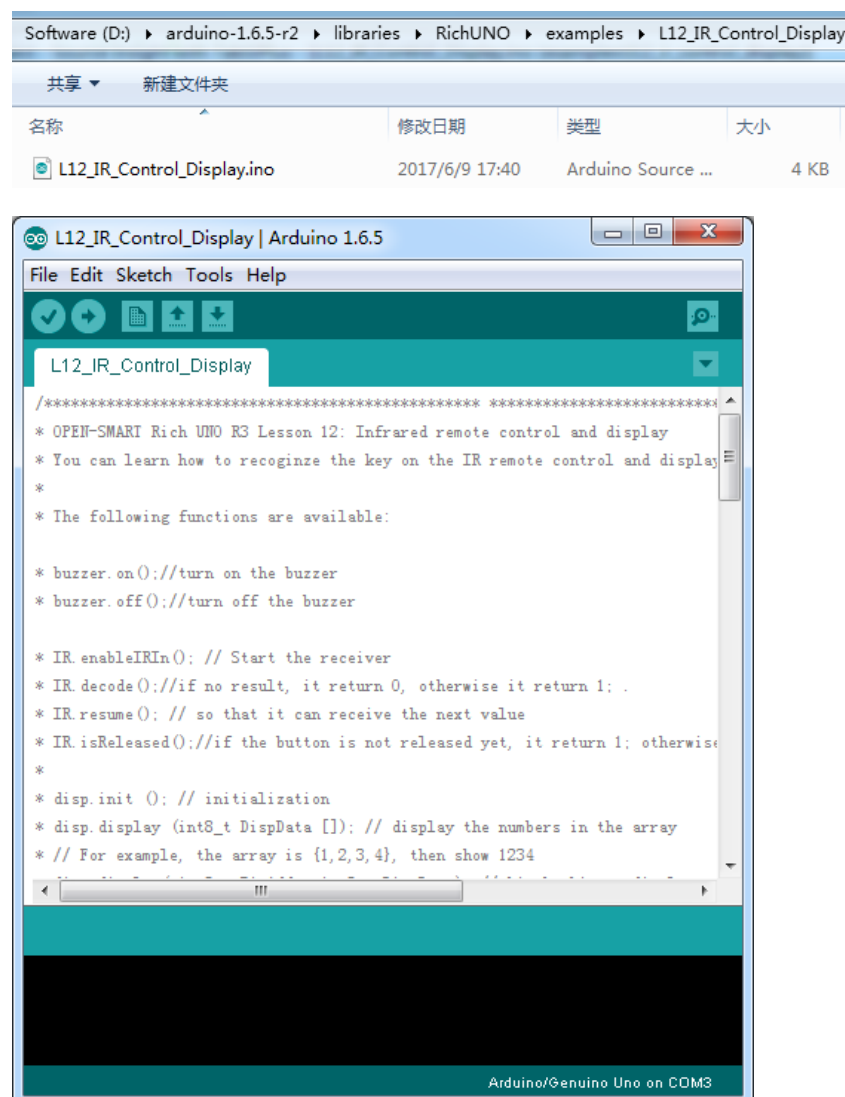
16.2 Upload code for lesson 12

1) Open lesson12 example

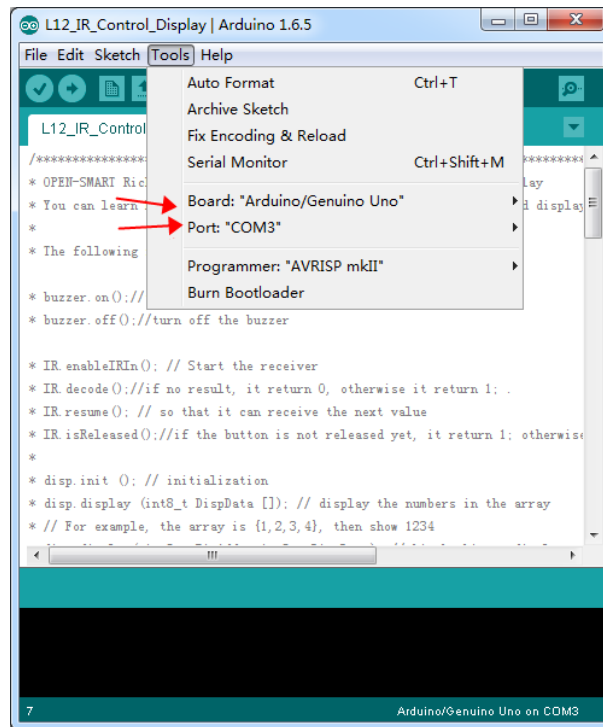
Double click the file in the directory of

\\arduino-1.6.5-r2\\libraries\\RichUNO\\examples\\ L12_IR_Control_Display

So that you find the window below:

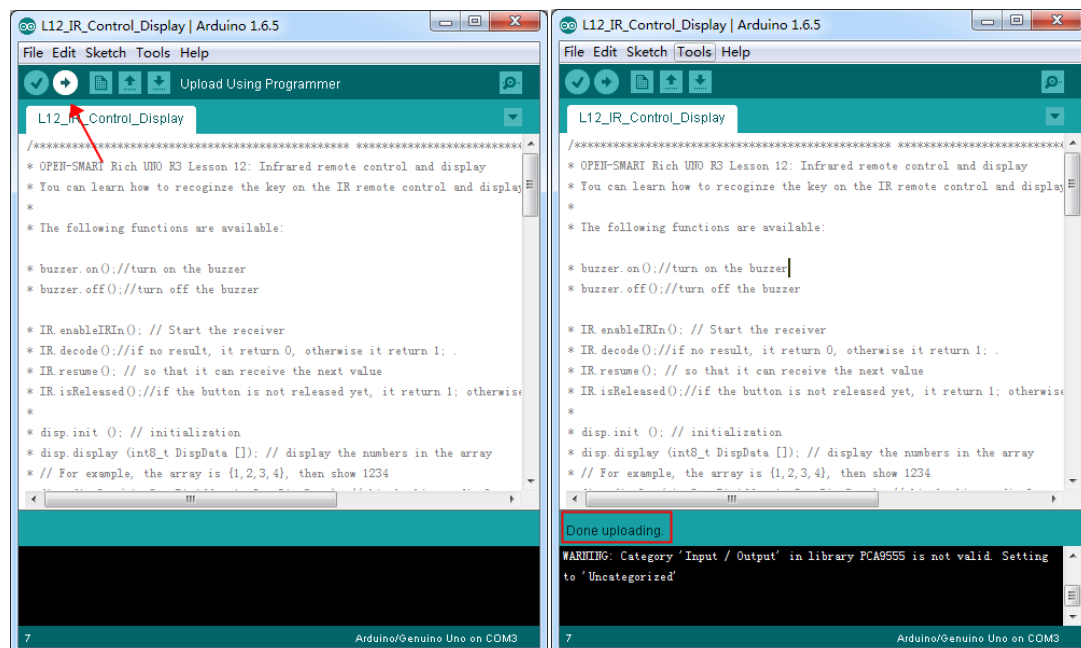


2) Check the Board and Serial Port



3) Click Upload button to upload the code to Rich UNO R3 board

When you see that “Done uploading”, congratulations, you have upload the code to the board successfully.



4) Check whether the board is running as you want

In this lesson, only number 0~9 and “+” “-” buttons are valid.

Every time you press the button, the buzzer will beep for a short time.

Every time you press the number 0~9, the 4-digit tube displays the number.

When you press “+”, the number increases by one. Pressing “-”, the number decreases by one.

16.3 Learn the functions

1) You can find that at the beginning of all lessons code is

```
#include <Wire.h>
```

```
#include <SoftwareSerial.h>
```

Because the RichUNO library files use Wire.h and SoftwareSerial.h which is not in the RichUNO directory, if the example code wants to use RichUNO library, it should add those two line.

2) Other header files

In this lesson, we use 4-digit display, IR remote and buzzer, so it should include their header files to control display more easily.

So you should add:

```
#include "RichUNOTM1637.h"
```

```
#include "RichUNOIRremote.h"
```

```
#include "RichUNOBuzzer.h"
```

3) Define class object

We have defined the class in the RichUNO library, and at any time you should define a object before you use it, and some object has parameter.

For example:

```
TM1637 disp(10,11); //define a TM1637 object, and display connect to D10 / D11 of Rich UNO R3 board
```

```
Buzzer buzzer(9); //define Buzzer object, it connects to D9 of Rich UNO R3
```

4) Two necessary functions

Each example code should include the two necessary functions:

```
void setup(){ }
```

```
void loop(){} 
```

Usually we write initialization code in setup function, and then write the code what you want the board to do in loop function which runs cyclically.

5) Functions

```
* buzzer.on(); //turn on the buzzer
```

```
* buzzer.off(); //turn off the buzzer
```

```
* IR.enableIRIn(); // Start the receiver
```

```
* IR.decode(); //if no result, it return 0, otherwise it return 1; .
```

```
* IR.resume(); // so that it can receive the next value
```

```
* IR.isReleased(); //if the button is not released yet, it return 1; otherwise it return 0;
```

```
* disp.init(); // initialization
```

You should write it in the setup function initialize it.

```
* disp.display(int Decimal); // display range: -999 ~ 9999
```

17 Lesson 13: Play a song

17.1 Overview

NOTE!!! First of all you should download the voice resources from our google drive:

<https://drive.google.com/drive/folders/0B6uNNXJ2z4CxaXFpakMxR0p1Unc?usp=sharing>

Then unzip it and find the 01 and 02 folder and put them into your TF card (should not larger than 32GB).

MP3 and songs are important for us now. We may feel happy and relax when we listen to some beautiful music.

In this lesson, you can learn how to play a song with its index in the TF card.

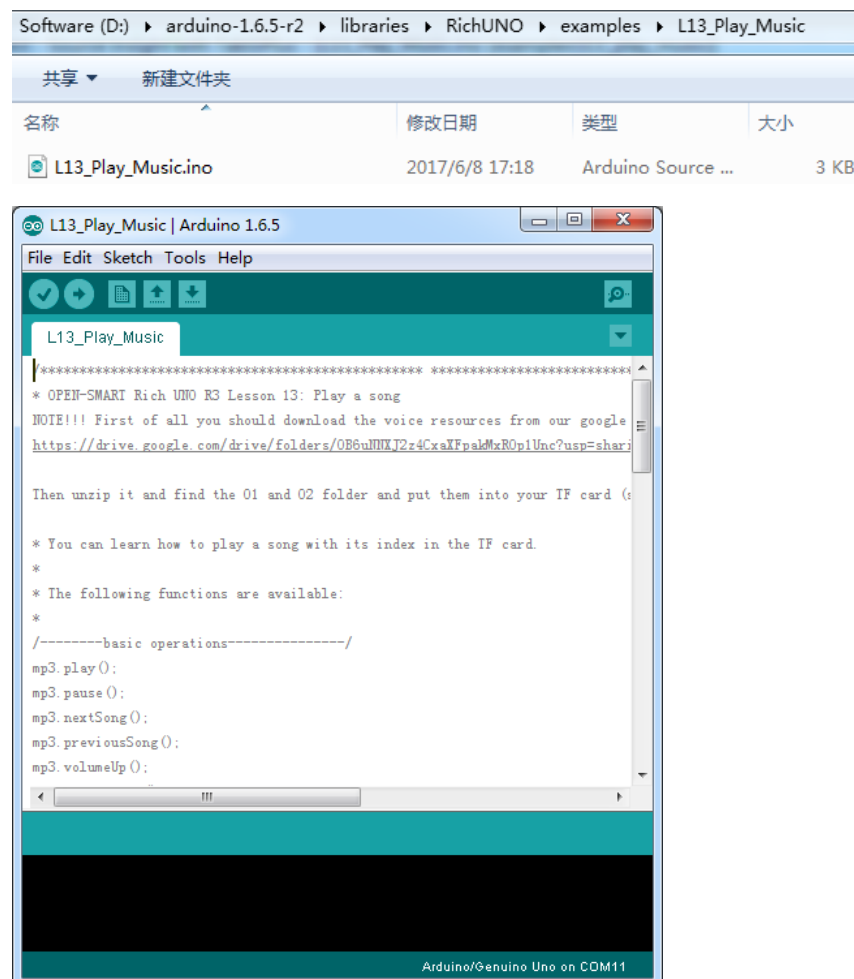
17.2 Upload code for lesson 13

1) Open lesson13 example

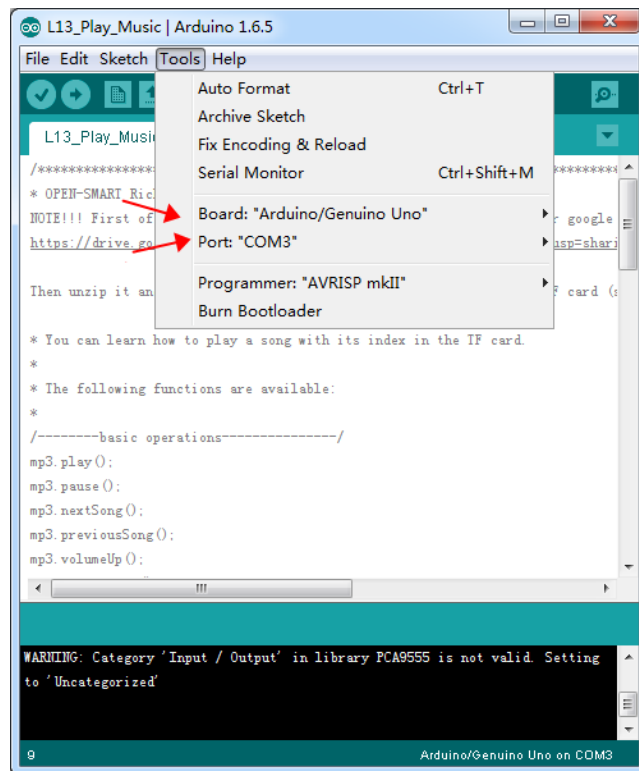
Double click the file in the directory of

\\arduino-1.6.5-r2\\libraries\\RichUNO\\examples\\ L13_Play_Music

So that you find the window below:

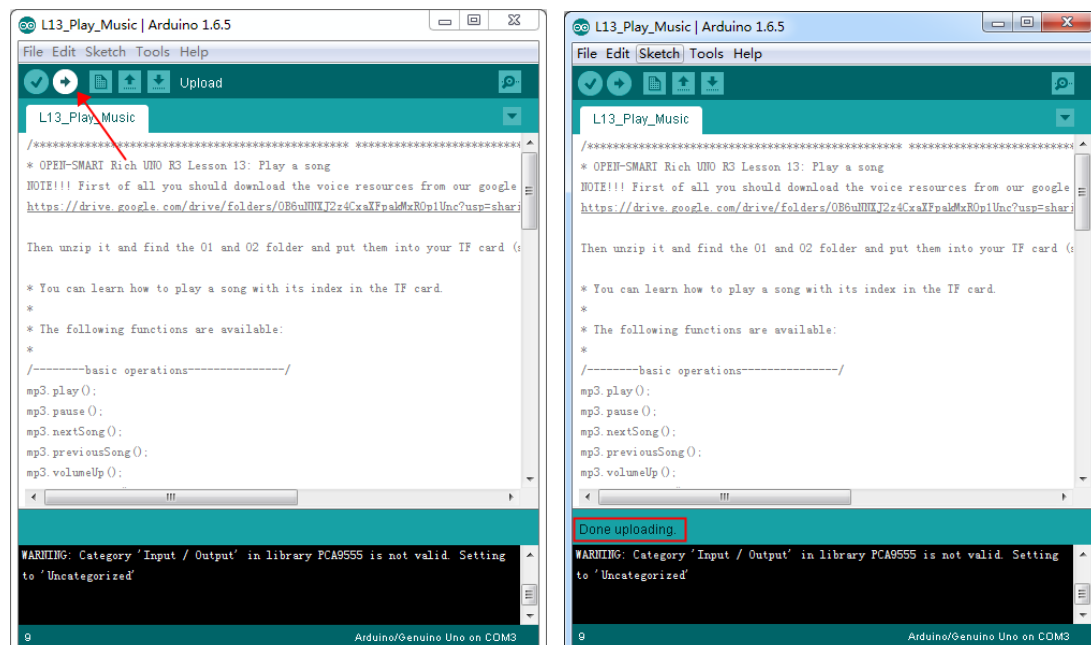


2) Check the Board and Serial Port



3) Click Upload button to upload the code to Rich UNO R3 board

When you see that “Done uploading”, congratulations, you have upload the code to the board successfully.



4) Check whether the board is running as you want

In this lesson, you can see it play the first song in the TF card.

17.3 Learn the functions

1) You can find that at the beginning of all lessons code is

```
#include <Wire.h>
```

```
#include <SoftwareSerial.h>
```

Because the RichUNO library files use Wire.h and SoftwareSerial.h which is not in the RichUNO directory, if the example code wants to use RichUNO library, it should add those two line.

2) Other header files

In this lesson we use MP3, so it should include their header files to control MP3 more easily.

So you should add:

```
#include "RichUNOMP3.h"
```

3) Define class object

We have defined the class in the RichUNO library, and at any time you should define a object before you use it, and some object has parameter.

For example:

```
MP3 mp3(7, 8); //define the MP3 object and connect to D7/D8 of Rich UNO R3
```

4) Two necessary functions

Each example code should include the two necessary functions:

```
void setup(){}
```

```
void loop(){}
```

Usually we write initialization code in setup function, and then write the code what you want the board to do in loop function which runs cyclically.

5) Functions

```
* mp3.playWithVolume(int8_t index, int8_t volume); //play the song according to the physical  
index of song in the TF card and the volume set
```

For example:

```
mp3.playWithVolume(1, 20); //play the first song with volume 20 (max is 30)
```

18 Lesson 14: Play song with its file name

18.1 Overview

NOTE!!! First of all you should download the voice resources from our google drive:

<https://drive.google.com/drive/folders/0B6uNNXJ2z4CxaXFpakMxR0p1Unc?usp=sharing>

Then unzip it and find the 01 and 02 folder and put them into your TF card (should not larger than 32GB).

In this lesson, you can learn how to play a song with folder name and file name. This is very important for voice broadcast projects.

18.2 Upload code for lesson 14

1) Open lesson14 example

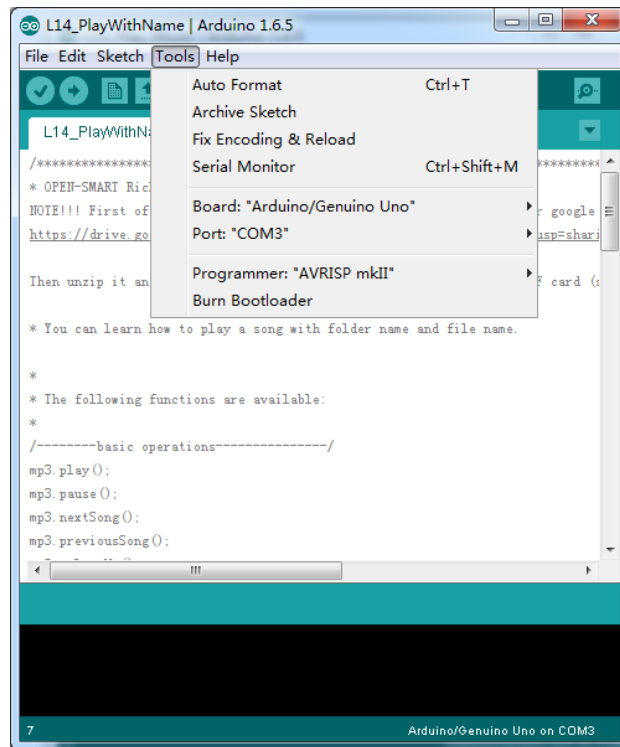
Double click the file in the directory of

\arduino-1.6.5-r2\libraries\RichUNO\examples\ L14_PlayWithName

So that you find the window below:

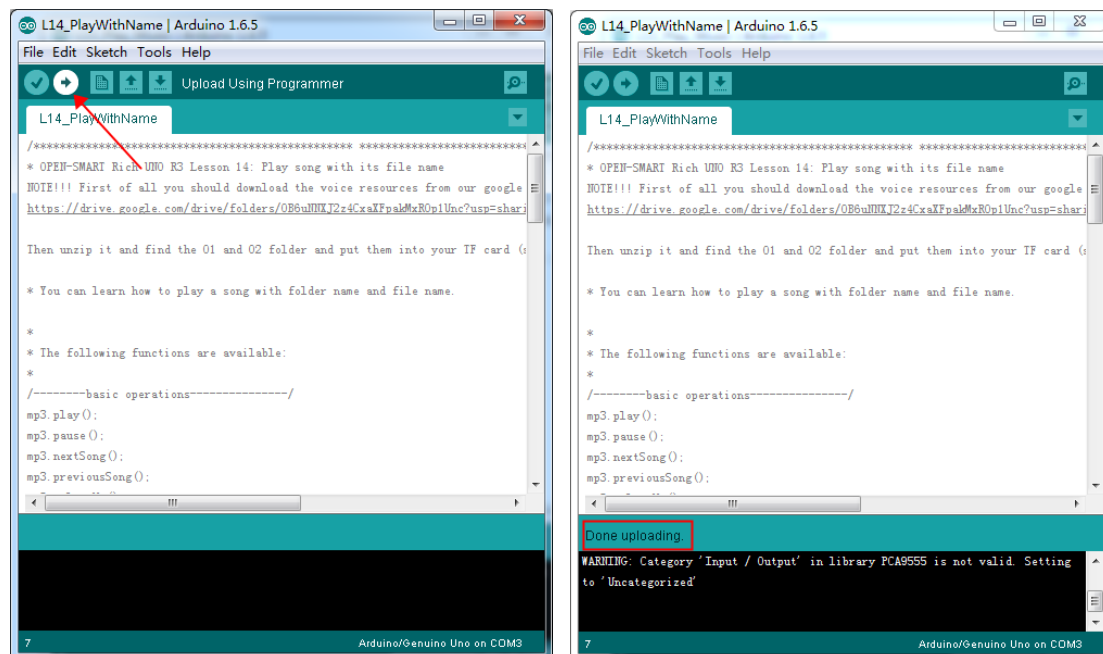


2) Check the Board and Serial Port



3) Click Upload button to upload the code to Rich UNO R3 board

When you see that “Done uploading”, congratulations, you have upload the code to the board successfully.



4) Check whether the board is running as you want

In this lesson, you can hear that it plays the song with the folder name and file name you set.

18.3 Learn the functions

1) You can find that at the beginning of all lessons code is

```
#include <Wire.h>
```

```
#include <SoftwareSerial.h>
```

Because the RichUNO library files use Wire.h and SoftwareSerial.h which is not in the RichUNO directory, if the example code wants to use RichUNO library, it should add those two line.

2) Other header files

In this lesson we use MP3, so it should include their header files to control MP3 more easily.

So you should add:

```
#include "RichUNOMP3.h"
```

3) Define class object

We have defined the class in the RichUNO library, and at any time you should define a object before you use it, and some object has parameter.

For example:

```
MP3 mp3(7, 8); //define the MP3 object and connect to D7/D8 of Rich UNO R3
```

4) Two necessary functions

Each example code should include the two necessary functions:

```
void setup(){ }
```

```
void loop(){} }
```

Usually we write initialization code in setup function, and then write the code what you want the board to do in loop function which runs cyclically.

5) Functions

```
mp3.playWithFileName(int8_t directory, int8_t file); //play a song according to the folder name and  
                                                    prefix of its file name
```

```
//directory (folder name) must be 01 02 03...09 10...99
```

```
//prefix of file name must be 001...009 010...099
```

```
mp3.setVolume(int8_t vol); //vol is 0~0x1e, 30 adjustable level
```


19 Lesson 15: Knob control volume

19.1 Overview

NOTE!!! First of all you should download the voice resources from our google drive:

<https://drive.google.com/drive/folders/0B6uNNXJ2z4CxaXFpakMxR0p1Unc?usp=sharing>

Then unzip it and find the 01 and 02 folder and put them into your TF card (should not larger than 32GB).

When you play a song, you may want to control the volume up and down, and the silver knob can help you.

In this lesson, you can learn how to use the knob to control volume of the song.

19 Upload code for lesson 15

1) Open lesson15 example

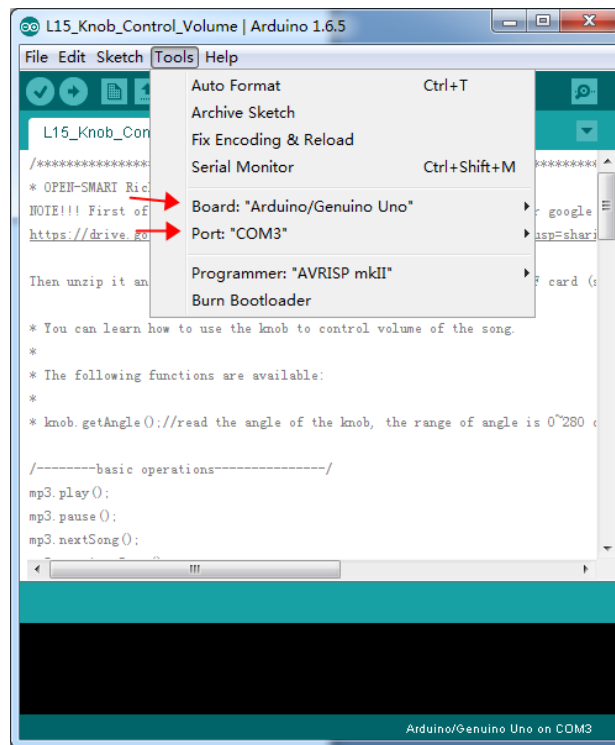
Double click the file in the directory of

\\arduino-1.6.5-r2\\libraries\\RichUNO\\examples\\L15_Knob_Control_Volume

So that you find the window below:

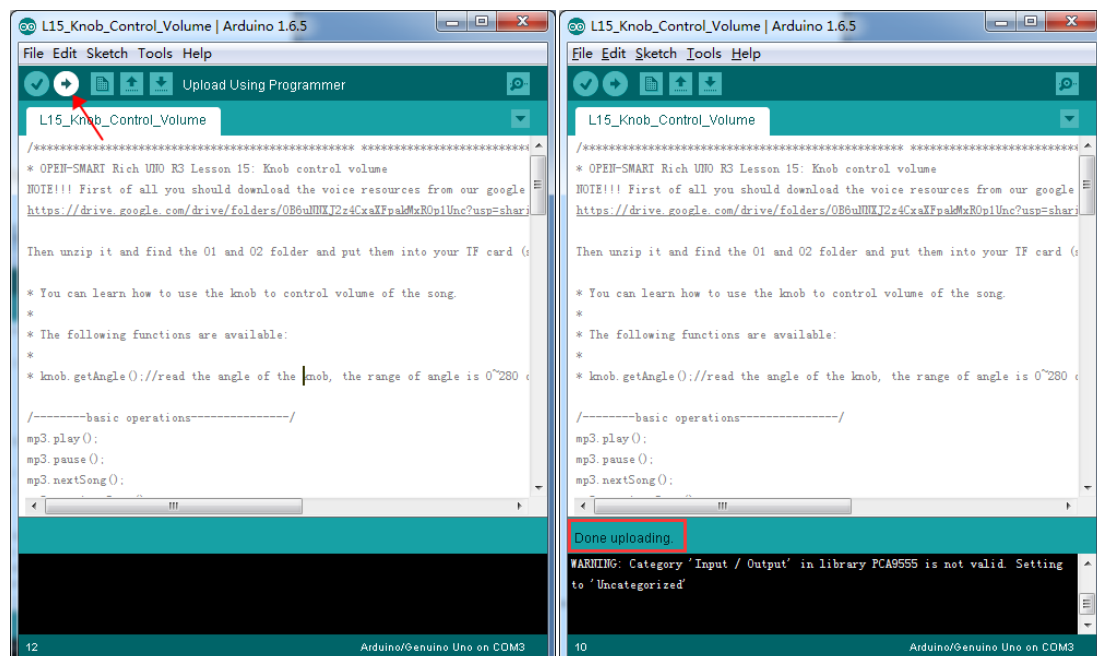


2) Check the Board and Serial Port



3) Click Upload button to upload the code to Rich UNO R3 board

When you see that “Done uploading”, congratulations, you have upload the code to the board successfully.



4) Check whether the board is running as you want

In this lesson, you can hear that it plays the song with the folder name and file name you set. And when you rotate clockwise, the volume is increased, otherwise the volume goes down.

19.3 Learn the functions

1) You can find that at the beginning of all lessons code is

```
#include <Wire.h>
```

```
#include <SoftwareSerial.h>
```

Because the RichUNO library files use Wire.h and SoftwareSerial.h which is not in the RichUNO directory, if the example code wants to use RichUNO library, it should add those two line.

2) Other header files

In this lesson we use MP3 and the knob, so it should include their header files to control MP3 more easily.

So you should add:

```
#include "RichUNOMP3.h"
```

```
#include "RichUNOKnob.h"
```

3) Define class object

We have defined the display class in the RichUNO library, and at any time you should define a object before you use it, and some object has parameter.

For example:

```
Knob knob(A0);
```

```
MP3 mp3(7, 8); //define the MP3 object and connect to D7/D8 of Rich UNO R3
```

4) Two necessary functions

Each example code should include the two necessary functions:

```
void setup(){ }
```

```
void loop(){} 
```

Usually we write initialization code in setup function, and then write the code what you want the board to do in loop function which runs cyclically.

5) Functions

```
* mp3.playWithFileName(int8_t directory, int8_t file); //play a song according to the folder name  
and prefix of its file name
```

```
//directory (folder name) must be 01 02 03...09 10...99
```

```
//prefix of file name must be 001...009 010...099
```

```
* mp3.setVolume(int8_t vol); //vol is 0~0x1e, 30 adjustable level
```

```
* knob.getAngle(); //read the angle of the knob, the range of angle is 0~280 degrees
```

And you should convert the angle to the volume range.

20 Lesson 16: Touch control MP3

20.1 Overview

NOTE!!! First of all you should download the voice resources from our google drive:

<https://drive.google.com/drive/folders/0B6uNNXJ2z4CxaXFpakMxR0p1Unc?usp=sharing>

Then unzip it and find the 01 and 02 folder and put them into your TF card (should not larger than 32GB).

In this lesson, you can learn how to use the 4-channel touch sensor to control the MP3, such as play / pause / next song / previous song operations.

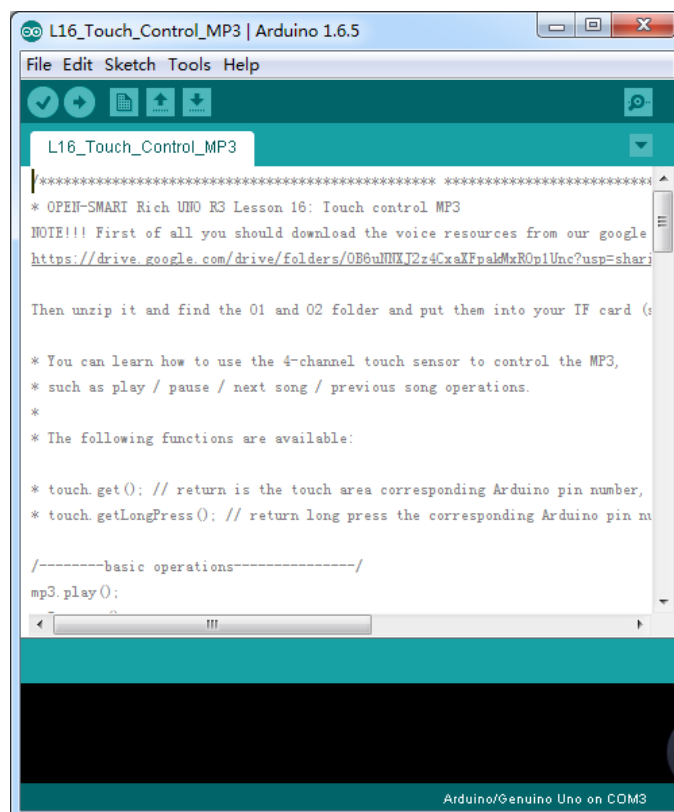
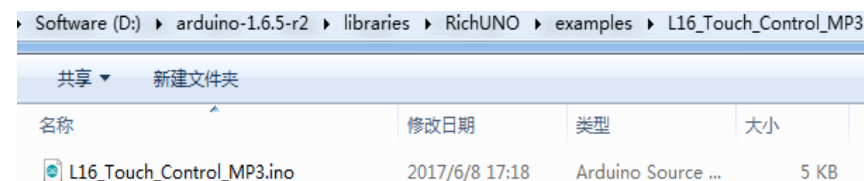
20.2 Upload code for lesson 16

1) Open lesson16 example

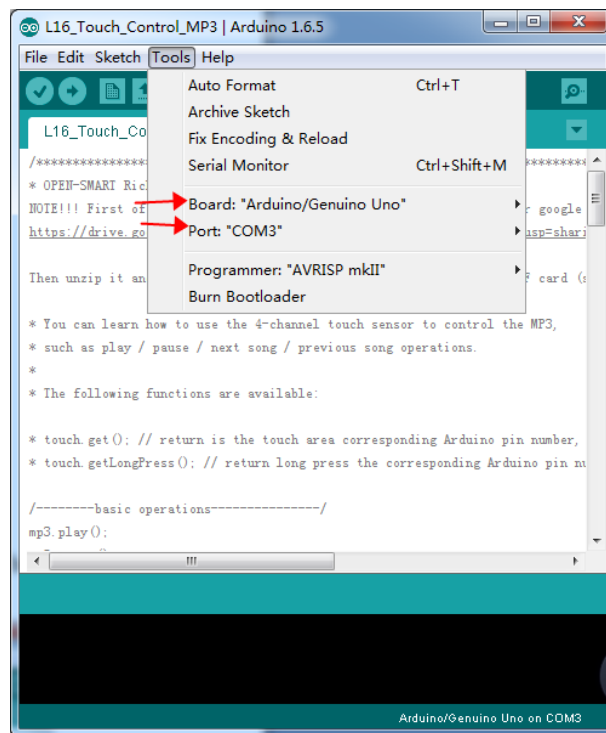
Double click the file in the directory of

\\arduino-1.6.5-r2\\libraries\\RichUNO\\examples\\L16_Touch_Control_MP3

So that you find the window below:

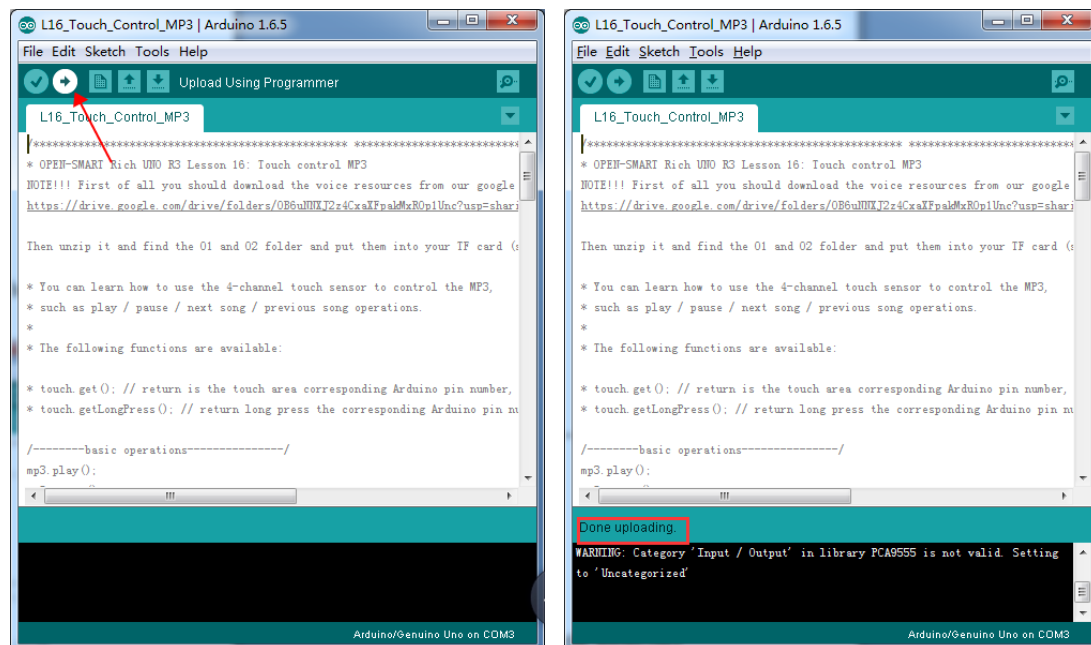


2) Check the Board and Serial Port



3) Click Upload button to upload the code to Rich UNO R3 board

When you see that “Done uploading”, congratulations, you have upload the code to the board successfully.



4) Check whether the board is running as you want

In this lesson, you can see it plays a song “Yesterday once more”. If you touch TCH2, the song pause; If you touch TCH1, it will play. Touch TCH3 to play next song and TCH4 to play previous song.

20.3 Learn the functions

1) You can find that at the beginning of all lessons code is

```
#include <Wire.h>
```

```
#include <SoftwareSerial.h>
```

Because the RichUNO library files use Wire.h and SoftwareSerial.h which is not in the RichUNO directory, if the example code wants to use RichUNO library, it should add those two line.

2) Other header files

In this lesson we use MP3 and touch sensor, so it should include their header files to control MP3 more easily.

So you should add:

```
#include "RichUNOMP3.h"
```

```
#include "RichUNOTouchSensor.h"
```

3) Define class object

We have defined the display class in the RichUNO library, and at any time you should define a object before you use it, and some object has parameter.

For example:

```
MP3 mp3(7, 8); //define the MP3 object and connect to D7/D8 of Rich UNO R3
```

4) Two necessary functions

Each example code should include the two necessary functions:

```
void setup(){ }
```

```
void loop(){ }
```

Usually we write initialization code in setup function, and then write the code what you want the board to do in loop function which runs cyclically.

5) Functions

** touch.get(); // return is the touch area corresponding Arduino pin number, if not then return -1*

If you touch TCH1, it will return 3. TCH2 is 4, TCH3 is 5, and TCH4 is 6.

You may try to write this function to get 1,2,3,4 for TCHx.

```
int get()
```

```
{
```

```
    int touchNum;
```

```
    int _pin;
```

```
    _pin = touch.get();
```

```
    switch(_pin)
```

```
    {
```

```
        case TOUCH_OUT1: touchNum = 1; break;
```

```
        case TOUCH_OUT2: touchNum = 2; break;
```

```
        case TOUCH_OUT3: touchNum = 3; break;
```

```
        case TOUCH_OUT4: touchNum = 4; break;
```

```
        default: touchNum = _pin; break;
```

```
    }
```

```
    return touchNum;
```

```
}
```

** mp3.play();//if the song is set pause, this function can make to back to play.*

** mp3.pause();//if the song is playing, this function can set it to pause.*

** mp3.nextSong();//only can play next song in the same folder*

For example: if the playing song is in folder 01, nextSong() function only can play the next song in 01 folder, and if there is only one song in it, this operation will play the same song again.

** mp3.previousSong();//only can play previous song in the same folder like nextSong() function.*

** mp3.setVolume(int8_t vol);//vol is 0~0x1e, 30 adjustable level*

** mp3.playWithFileName(int8_t directory, int8_t file);//play a song according to the folder name
and prefix of its file name*

//directory (folder name) must be 01 02 03...09 10...99

//prefix of file name must be 001...009 010...099

21 Lesson 17: IR control MP3

21.1 Overview

NOTE!!! First of all you should download the voice resources from our google drive:

<https://drive.google.com/drive/folders/0B6uNNXJ2z4CxaXFpakMxR0p1Unc?usp=sharing>

Then unzip it and find the 01 and 02 folder and put them into your TF card (should not larger than 32GB).

In this lesson, how to use the Infrared remote control to control the MP3, and when you press the button, the buzzer beep.

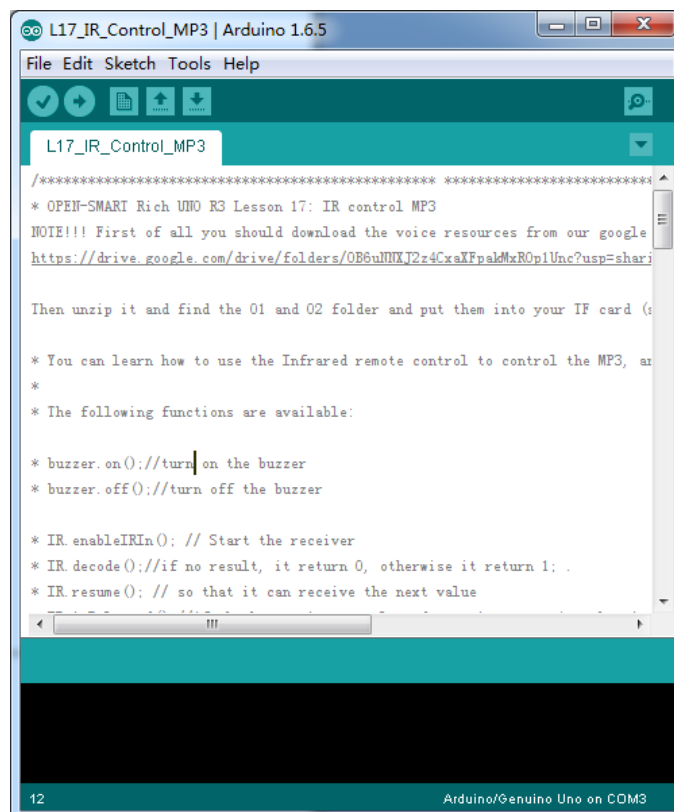
21.2 Upload code for lesson 17

1) Open lesson17 example

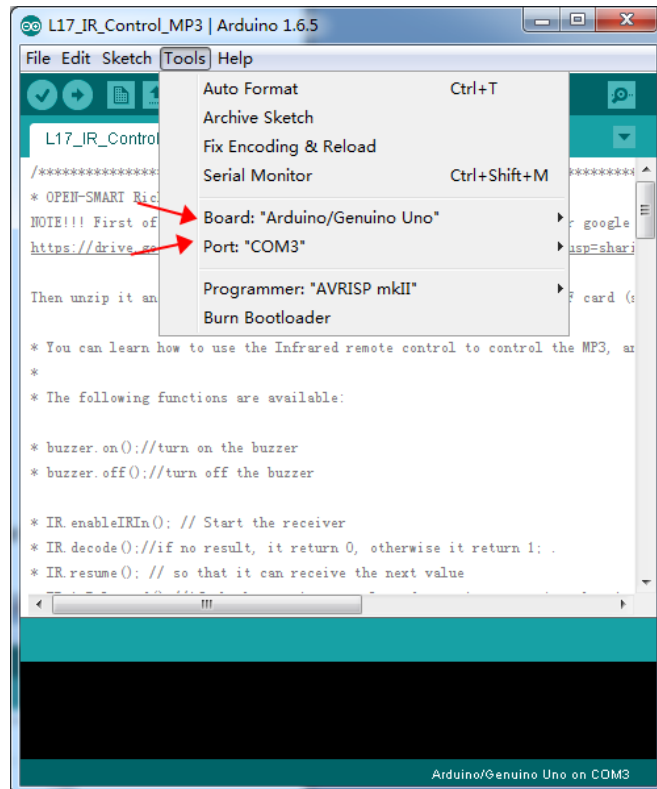
Double click the file in the directory of

\arduino-1.6.5-r2\libraries\RichUNO\examples\ L17_IR_Control_MP3

So that you find the window below:

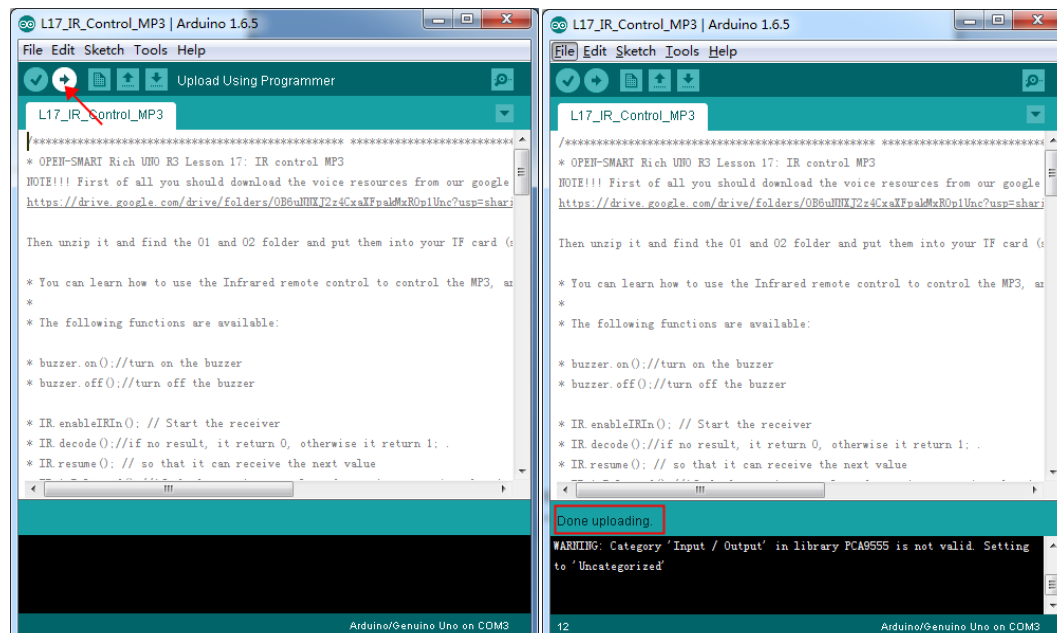


2) Check the Board and Serial Port



3) Click Upload button to upload the code to Rich UNO R3 board

When you see that “Done uploading”, congratulations, you have upload the code to the board successfully.



4) Check whether the board is running as you want

In this lesson, you can press 1 to play the first song in the TF card and press 2 to play second song. Press “+” and “-” to adjust the volume. Press PLAY key to set song to play or pause. Press NEXT key to play next song and PREV key to play previous song.

21.3 Learn the functions

1) You can find that at the beginning of all lessons code is

```
#include <Wire.h>
```

```
#include <SoftwareSerial.h>
```

Because the RichUNO library files use Wire.h and SoftwareSerial.h which is not in the RichUNO directory, if the example code wants to use RichUNO library, it should add those two line.

2) Other header files

In this lesson we use MP3, IR remote and buzzer, so it should include their header files to control display more easily.

So you should add:

```
#include "RichUNOMP3.h"
```

```
#include "RichUNOIRremote.h"
```

```
#include "RichUNOBuzzer.h"
```

3) Define class object

We have defined the display class in the RichUNO library, and at any time you should define a object before you use it, and some object has parameter.

For example:

```
MP3 mp3(7, 8); //define the MP3 object and connect to D7/D8 of Rich UNO R3
```

4) Two necessary functions

Each example code should include the two necessary functions:

```
void setup(){ }
```

```
void loop(){} 
```

Usually we write initialization code in setup function, and then write the code what you want the board to do in loop function which runs cyclically.

5) Functions

** mp3.play(); //if the song is set pause, this function can make to back to play.*

** mp3.pause(); //if the song is playing, this function can set it to pause.*

** mp3.nextSong(); //only can play next song in the same folder*

For example: if the playing song is in folder 01, nextSong() function only can play the next song in 01 folder, and if there is only one song in it, this operation will play the same song again.

** mp3.previousSong(); //only can play previous song in the same folder like nextSong() function.*

** mp3.setVolume(int8_t vol); //vol is 0~0x1e, 30 adjustable level*

** mp3.playWithIndex(int8_t index); //play the song according to the physical index of song in the TF card*

** buzzer.on(); //turn on the buzzer*

** buzzer.off(); //turn off the buzzer*

** IR.enableIRIn(); // Start the receiver*
** IR.decode();//if no result, it return 0, otherwise it return 1; .*
** IR.resume(); // so that it can receive the next value*
** IR.isReleased();//if the button is not released yet, it return 1; otherwise it return 0;*
** IR.keycode // this will return the value of key code you just press*

22 Lesson 18: Speak voice

22.1 Overview

NOTE!!! First of all you should download the voice resources from our google drive:

<https://drive.google.com/drive/folders/0B6uNNXJ2z4CxaXFpakMxR0p1Unc?usp=sharing>

Then unzip it and find the 01 and 02 folder and put them into your TF card (should not larger than 32GB).

When you want to do wonderful voice broadcast project, you should first know the voice resources you have.

In this lesson, you can learn to know all the voice sources in the 02 folder so you can make your voice broadcast project.

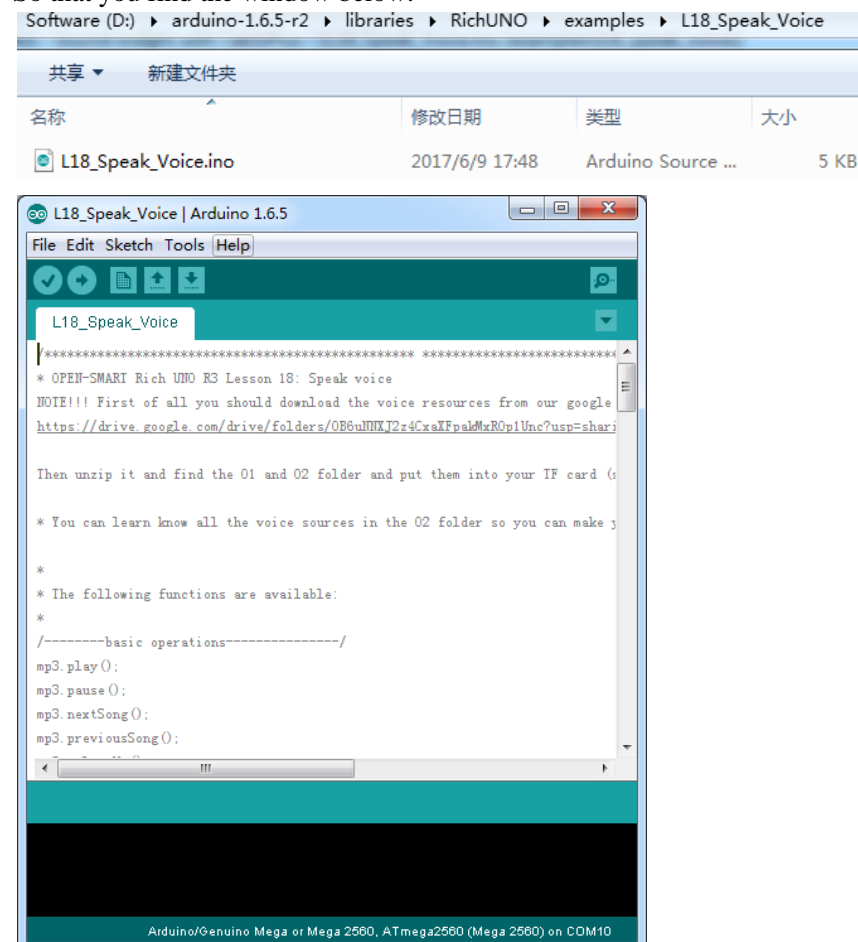
22.2 Upload code for lesson 18

1) Open lesson18 example

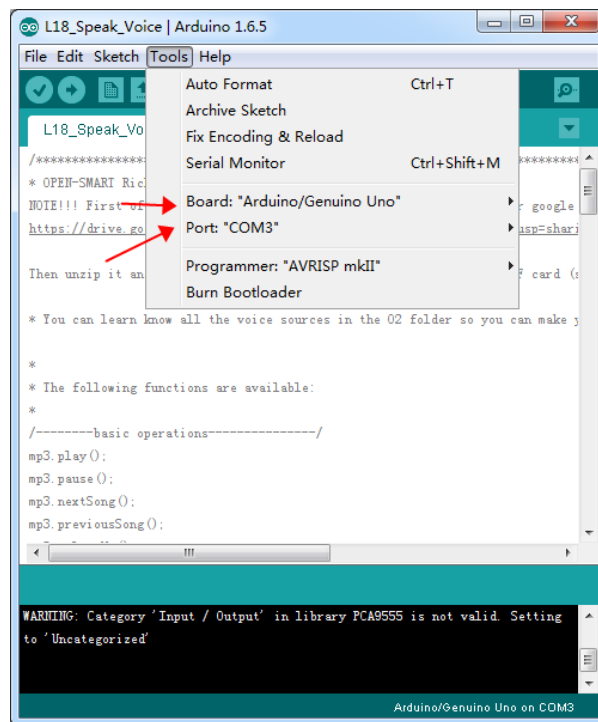
Double click the file in the directory of

\arduino-1.6.5-r2\libraries\RichUNO\examples\ L18_Speak_Voice

So that you find the window below:

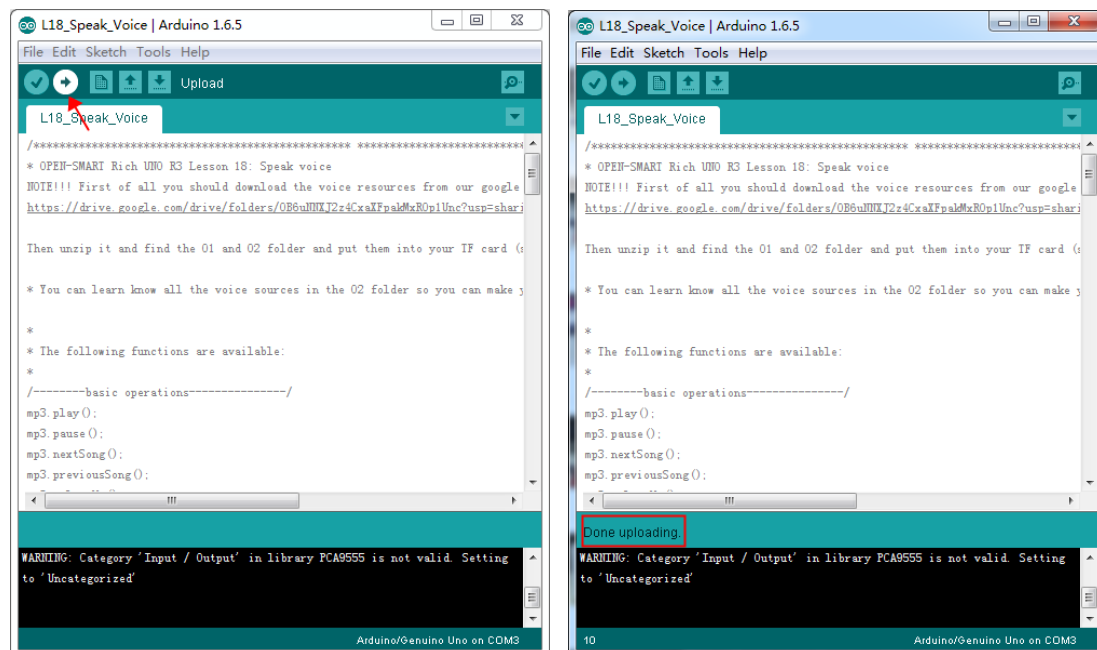


2) Check the Board and Serial Port



3) Click Upload button to upload the code to Rich UNO R3 board

When you see that “Done uploading”, congratulations, you have upload the code to the board successfully.



4) Check whether the board is running as you want

In this lesson, firstly you can touch TCH3 to speak the first voice and it displays 1.

Then touch TCH1 to increase the number (voice file name) one by one, and TCH2 to decrease, and every time you touch, it will speak the voice according to the voice file name.

22.3 Learn the functions

1) You can find that at the beginning of all lessons code is

```
#include <Wire.h>
```

```
#include <SoftwareSerial.h>
```

Because the RichUNO library files use Wire.h and SoftwareSerial.h which is not in the RichUNO directory, if the example code wants to use RichUNO library, it should add those two line.

2) Other header files

In this lesson we use MP3, 4-digit display and touch sensor, so it should include their header files to control display more easily.

So you should add:

```
#include "RichUNOMP3.h"
```

```
#include "RichUNOTouchSensor.h"
```

```
#include "RichUNOTM1637.h"
```

3) Define class object

We have defined the display class in the RichUNO library, and at any time you should define a object before you use it, and some object has parameter.

For example:

```
MP3 mp3(7, 8); //define the MP3 object and connect to D7/D8 of Rich UNO R3
```

4) Two necessary functions

Each example code should include the two necessary functions:

```
void setup(){ }
```

```
void loop(){} }
```

Usually we write initialization code in setup function, and then write the code what you want the board to do in loop function which runs cyclically.

5) Functions

** touch.get(); // return is the touch area corresponding Arduino pin number, if not then return -1*

If you touch TCH1, it will return 3. TCH2 is 4, TCH3 is 5, and TCH4 is 6.

You may try to write this function to get 1,2,3,4 for TCHx.

```
int get()
```

```
{
```

```
    int touchNum;
```

```
    int _pin;
```

```
    _pin = touch.get();
```

```
    switch(_pin)
```

```
    {
```

```
        case TOUCH_OUT1: touchNum = 1; break;
```

```
        case TOUCH_OUT2: touchNum = 2; break;
```

```
        case TOUCH_OUT3: touchNum = 3; break;
```

```
        case TOUCH_OUT4: touchNum = 4; break;
```

```
        default: touchNum = _pin; break;
```

```
    }
```

```
    return touchNum;
```

```
}
```

** mp3.setVolume(int8_t vol); // vol is 0~0x1e, 30 adjustable level*

** mp3.playWithFileName(int8_t directory, int8_t file); // play a song according to the folder name
and prefix of its file name*

// directory (folder name) must be 01 02 03...09 10...99

// prefix of file name must be 001...009 010...099

** disp.init(); // initialization*
You should write it in the setup function initialize it.

** disp.display(int Decimal); // display range: -999 ~ 9999*

23 Lesson 19: Speak pi

23.1 Overview

NOTE!!! First of all you should download the voice resources from our google drive:

<https://drive.google.com/drive/folders/0B6uNNXJ2z4CxaXFpakMxR0p1Unc?usp=sharing>

Then unzip it and find the 01 and 02 folder and put them into your TF card (should not larger than 32GB).

When you want to do wonderful voice broadcast project, you should know how to play multiple tone in the order set.

In this lesson, you can learn how to play a number (Pi) according to the value and the filename of digit.

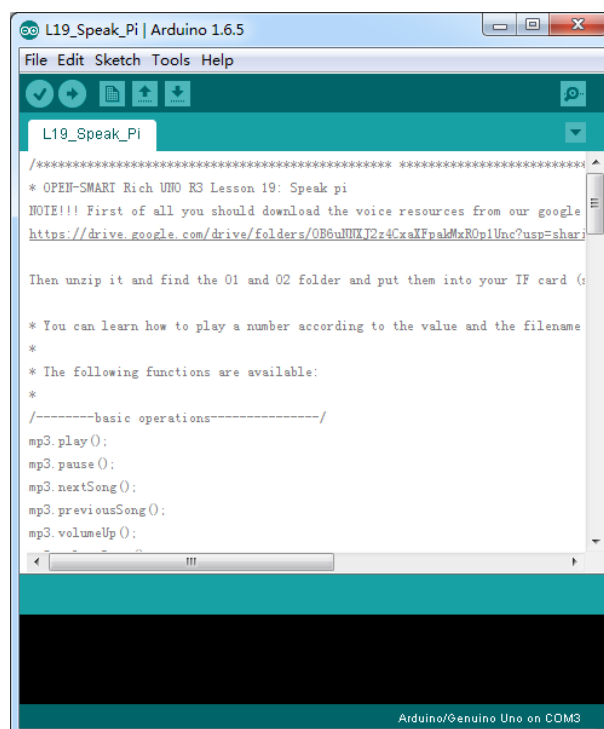
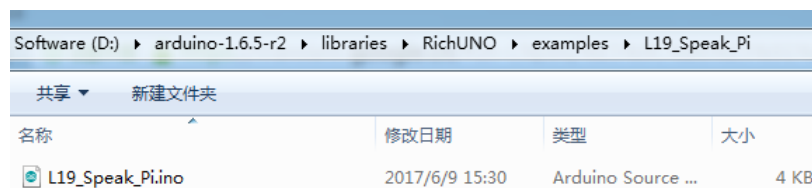
23.2 Upload code for lesson 19

1) Open lesson19 example

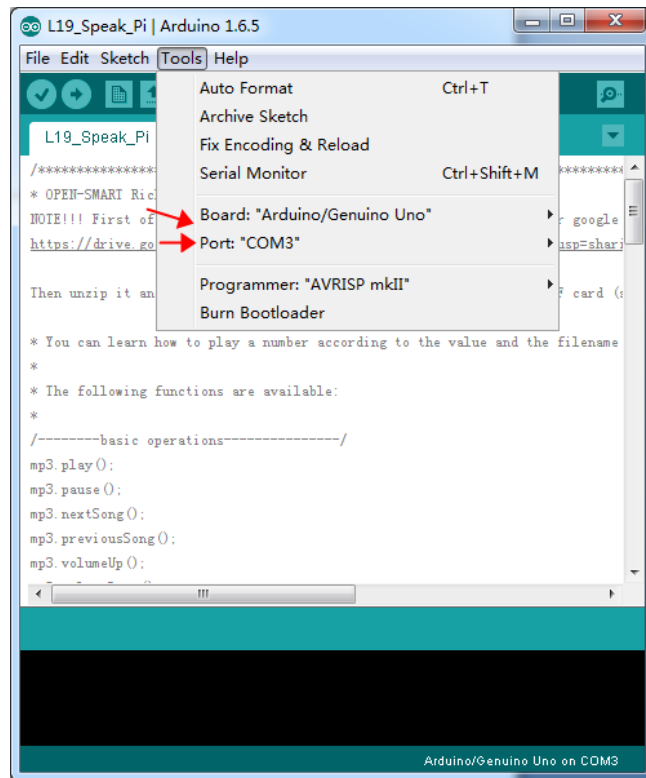
Double click the file in the directory of

\arduino-1.6.5-r2\libraries\RichUNO\examples\L19_Speak_Pi

So that you find the window below:

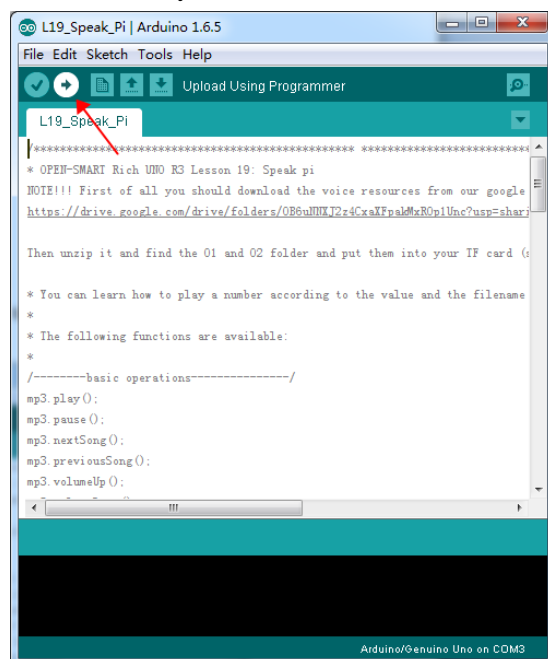


2) Check the Board and Serial Port



3) Click Upload button to upload the code to Rich UNO R3 board

When you see that “Done uploading”, congratulations, you have upload the code to the board successfully.



4) Check whether the board is running as you want

In this lesson, you can hear that it speaks 9 digit of pi.

23.3 Learn the functions

1) You can find that at the beginning of all lessons code is

```
#include <Wire.h>
```

```
#include <SoftwareSerial.h>
```

Because the RichUNO library files use Wire.h and SoftwareSerial.h which is not in the RichUNO directory, if the example code wants to use RichUNO library, it should add those two line.

2) Other header files

In this lesson we use MP3, so it should include their header files to control display more easily.

So you should add:

```
#include "RichUNOMP3.h"
```

3) Define class object

We have defined the display class in the RichUNO library, and at any time you should define a object before you use it, and some object has parameter.

For example:

```
MP3 mp3(7, 8); //define the MP3 object and connect to D7/D8 of Rich UNO R3
```

4) Two necessary functions

Each example code should include the two necessary functions:

```
void setup(){ }
```

```
void loop(){} }
```

Usually we write initialization code in setup function, and then write the code what you want the board to do in loop function which runs cyclically.

5) Functions

```
* mp3.setVolume(int8_t vol); //vol is 0~0x1e, 30 adjustable level
```

```
* mp3.playWithFileName(int8_t directory, int8_t file); //play a song according to the folder name  
and prefix of its file name
```

```
//directory (folder name) must be 01 02 03...09 10...99
```

```
//prefix of file name must be 001...009 010...099
```

```
mp3.getStatus(); //it will return the status of the mp3 player
```

```
#define STATUS_STOP 0
```

```
#define STATUS_PLAY 1
```

```
#define STATUS_PAUSE 2
```

```
#define STATUS_FORWARD 3
```

```
#define STATUS_REWIND 4
```

NOTE: If it is playing, it will return 1, you should wait until it return 0, then you can play next voice.

24 Lesson 20: Speak temperature

24.1 Overview

NOTE!!! First of all you should download the voice resources from our google drive:

<https://drive.google.com/drive/folders/0B6uNNXJ2z4CxaXFpakMxR0p1Unc?usp=sharing>

Then unzip it and find the 01 and 02 folder and put them into your TF card (should not larger than 32GB).

In this lesson, You can learn how to speak the temperature you get from the temperature sensor according to the value and the filename of digit /beep tone. And only when you touch TCH1 area, it will speak temperature.

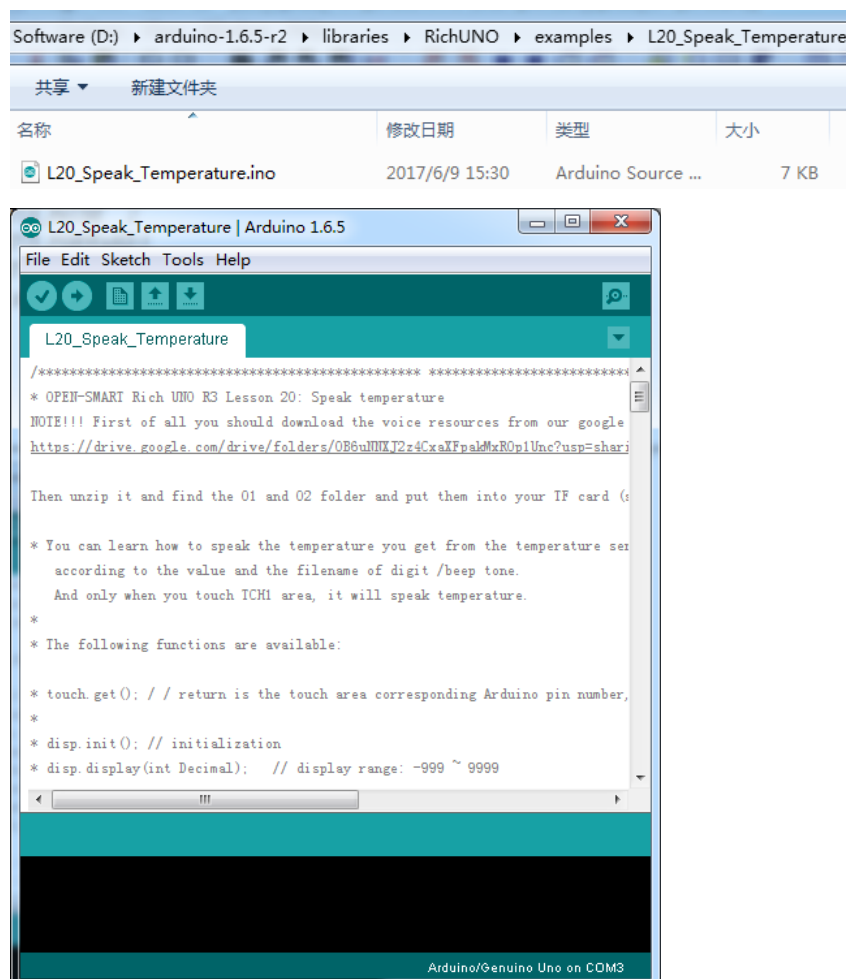
24.2 Upload code for lesson 20

1) Open lesson20 example

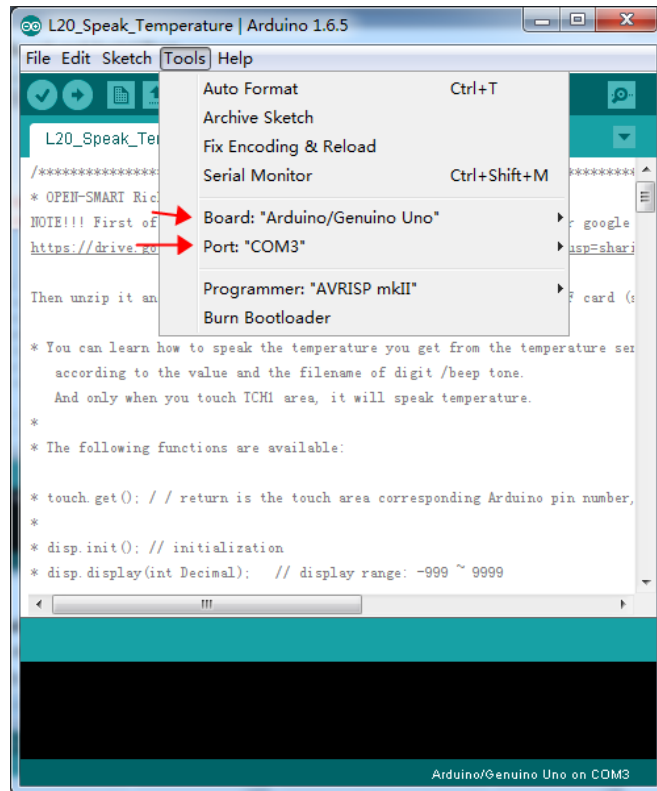
Double click the file in the directory of

\\arduino-1.6.5-r2\\libraries\\RichUNO\\examples\\ L20_Speak_Temperature

So that you find the window below:

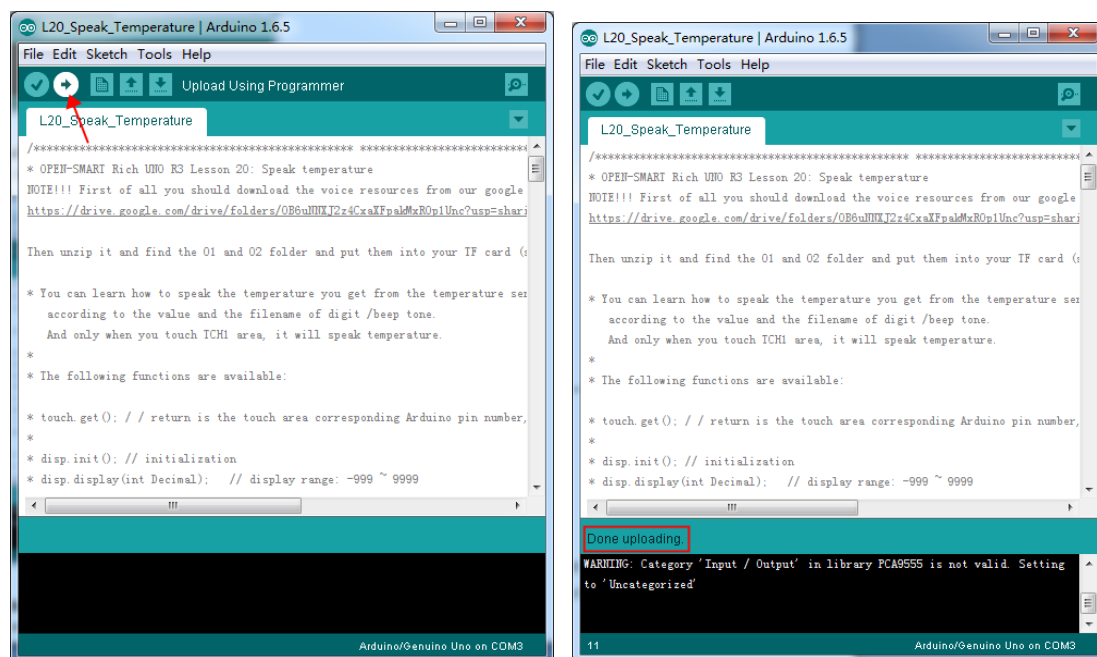


2) Check the Board and Serial Port



3) Click Upload button to upload the code to Rich UNO R3 board

When you see that “Done uploading”, congratulations, you have upload the code to the board successfully.



4) Check whether the board is running as you want

In this lesson, you can touch TCH1 so that OPEN-SMART will tell you the temperature.

24.3 Learn the functions

1) You can find that at the beginning of all lessons code is

```
#include <Wire.h>
```

```
#include <SoftwareSerial.h>
```

Because the RichUNO library files use Wire.h and SoftwareSerial.h which is not in the RichUNO directory, if the example code wants to use RichUNO library, it should add those two line.

2) Other header files

In this lesson we use MP3, touch sensor, LM75A temperature sensor and 4-digit tube, so it should include their header files to control display more easily.

So you should add:

```
#include "RichUNOMP3.h"
```

```
#include "RichUNOLM75.h"
```

```
#include "RichUNOTouchSensor.h"
```

```
#include "RichUNOTM1637.h"
```

3) Define class object

We have defined the display class in the RichUNO library, and at any time you should define a object before you use it, and some object has parameter.

For example:

```
MP3 mp3(7, 8); //define the MP3 object and connect to D7/D8 of Rich UNO R3
```

4) Two necessary functions

Each example code should include the two necessary functions:

```
void setup(){ }
```

```
void loop(){} 
```

Usually we write initialization code in setup function, and then write the code what you want the board to do in loop function which runs cyclically.

5) Functions

```
* touch.get(); // return is the touch area corresponding Arduino pin number, if not then return -1
```

```
* mp3.setVolume(int8_t vol); //vol is 0~0x1e, 30 adjustable level
```

```
* mp3.playWithFileName(int8_t directory, int8_t file); //play a song according to the folder name  
and prefix of its file name
```

```
//directory (folder name) must be 01 02 03...09 10...99
```

```
//prefix of file name must be 001...009 010...099
```

```
mp3.getStatus(); //it will return the status of the mp3 player
```

```
#define STATUS_STOP 0
```

```
#define STATUS_PLAY 1
```

```
#define STATUS_PAUSE 2
```

```
#define STATUS_FORWARD 3
```

```
#define STATUS_REWIND 4
```

NOTE: If it is playing, it will return 1, you should wait until it return 0, then you can play next voice.

** temper.getTemperature(); // get the temperature value, the return of the float is the decimal, unit is in degrees Celsius.*

For example:

float celsius;

celsius = temper.getTemperatue();//get temperature

** disp.init(); // initialization*

You should write it in the setup function initialize it and it has set the default brightness to be 3(0~7, number larger, brightness higher).

** disp.display(int8_t DispData []); // display the numbers in the array*

** // For example, the array is {1,2,3,4}, then show 1234*

Wire.begin();//you should call it in the setup() function because LM75A temperature sensor use I2C library

25 Lesson 21: Speak clock

25.1 Overview

NOTE!!! First of all you should download the voice resources from our google drive:

<https://drive.google.com/drive/folders/0B6uNNXJ2z4CxaXFpakMxR0p1Unc?usp=sharing>

Then unzip it and find the 01 and 02 folder and put them into your TF card (should not larger than 32GB).

In this lesson, you can learn how to speak the time you get from the RTC module according to the value and the filename of digit /beep tone.

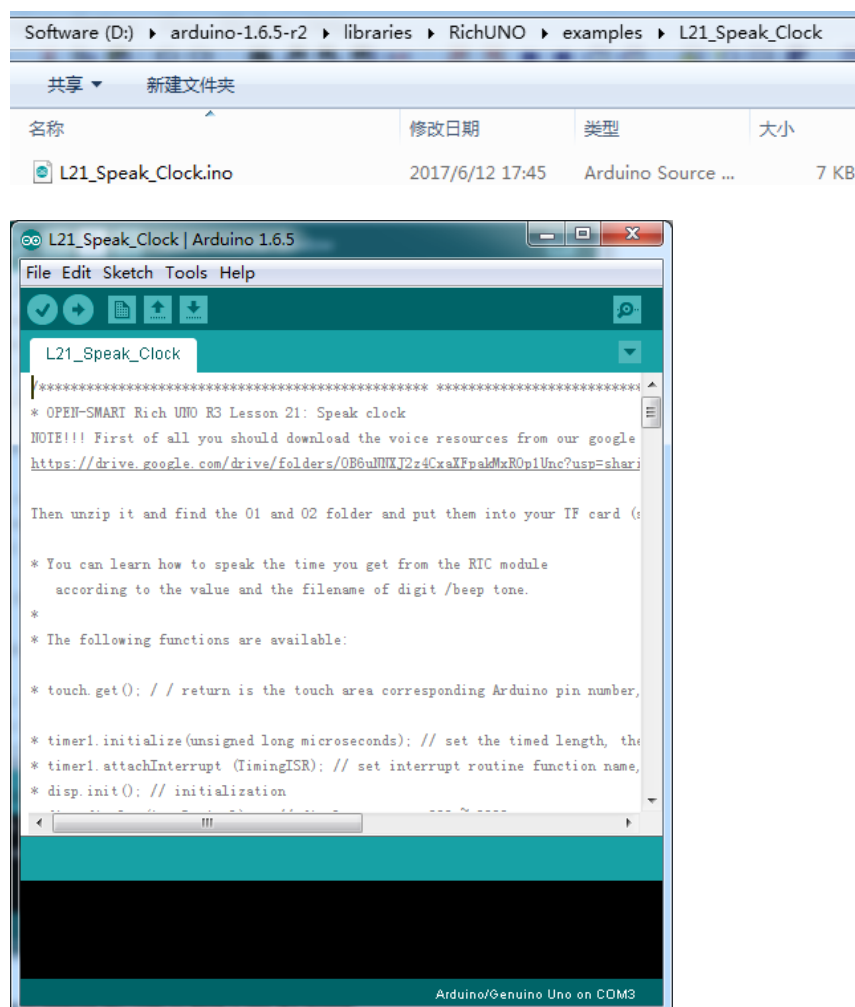
25.2 Upload code for lesson 21

1) Open lesson21 example

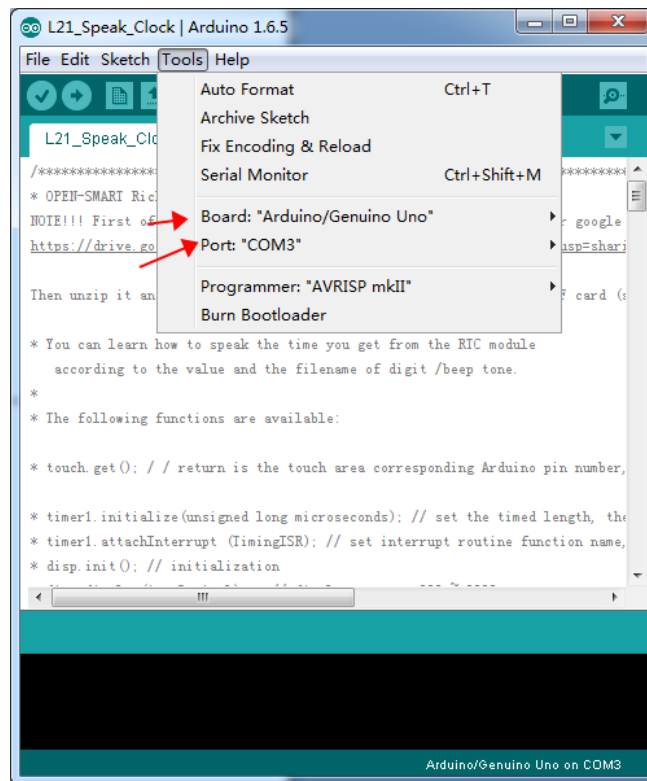
Double click the file in the directory of

\arduino-1.6.5-r2\libraries\RichUNO\examples\ L21_Speak_Clock

So that you find the window below:

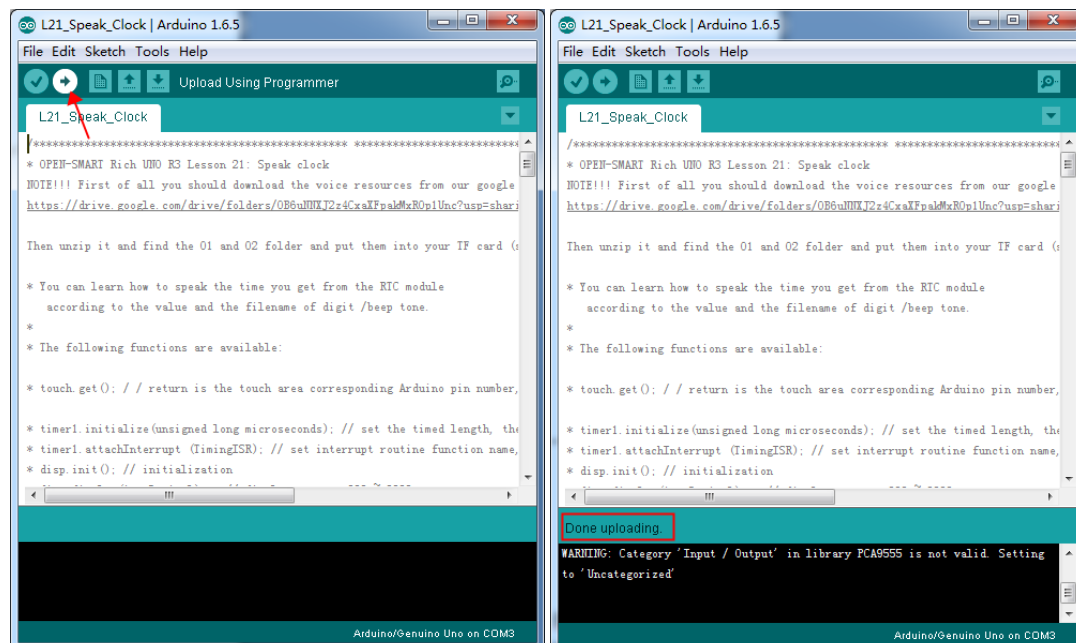


2) Check the Board and Serial Port



3) Click Upload button to upload the code to Rich UNO R3 board

When you see that “Done uploading”, congratulations, you have upload the code to the board successfully.



4) Check whether the board is running as you want

In this lesson, you can see it displays the time and the clock point is flashing every second. Then you can touch TCH2 to hear the time it tells you.

25.3 Learn the functions

1) You can find that at the beginning of all lessons code is

```
#include <Wire.h>
```

```
#include <SoftwareSerial.h>
```

Because the RichUNO library files use Wire.h and SoftwareSerial.h which is not in the RichUNO directory, if the example code wants to use RichUNO library, it should add those two line.

2) Other header files

In this lesson we use MP3, 4-digit display, RTC and touch sensor, so it should include their header files to control display more easily.

So you should add:

```
#include "RichUNOMP3.h"
```

```
#include "RichUNOTouchSensor.h"
```

```
#include "RichUNOTM1637.h"
```

```
#include "RichUNODS1307.h"
```

3) Define class object

We have defined the class in the RichUNO library, and at any time you should define a object before you use it, and some object has parameter.

For example:

```
TM1637 disp(10,11); //define a TM1637 object, and display connect to D10 / D11 of Rich UNO R3 board
```

```
DS1307 clock; //define a object of DS1307 class
```

4) Two necessary functions

Each example code should include the two necessary functions:

```
void setup(){ }
```

```
void loop(){} 
```

Usually we write initialization code in setup function, and then write the code what you want the board to do in loop function which runs cyclically.

5) Functions

```
* touch.get(); // return is the touch area corresponding Arduino pin number, if not then return -1
```

```
* mp3.setVolume(int8_t vol); //vol is 0~0x1e, 30 adjustable level
```

```
* mp3.playWithFileName(int8_t directory, int8_t file); //play a song according to the folder name  
and prefix of its file name
```

```
//directory (folder name) must be 01 02 03...09 10...99
```

```
//prefix of file name must be 001...009 010...099
```

```
mp3.getStatus(); //it will return the status of the mp3 player
```

```
#define STATUS_STOP 0
```

```
#define STATUS_PLAY 1
```

```
#define STATUS_PAUSE 2
```

```
#define STATUS_FORWARD 3
```

```
#define STATUS_REWIND 4
```

NOTE: If it is playing, it will return 1, you should wait until it return 0, then you can play next voice.

```
* clock.begin(); // The clock is on and the function must be called first
```

```
* clock.getTime(void); // read the date and time, will be saved in the clock class variable
```

After you call get time function, you can get the hour and minute from clock.hour and clock.minute

```
* Timer1.initialize(unsigned long microseconds); // set the timed length, the unit is microseconds
```

```
* Timer1.attachInterrupt(TimingISR); // set interrupt routine function name, is the timing interrupt  
entry, when the time is up, it will call TimingISR function you write.
```

```
* disp.init(); // initialization. You should write it in the setup function initialize it.
```

```
* disp.display(int8_t DispData []); // display the numbers in the array
```

// For example, the array is {1,2,3,4}, then show 1234

```
* disp.point(0); // Turn off the display colon and the next display takes effect
```

```
* disp.point(1); // Turn on the display colon and the next display takes effect
```

26 Lesson 22: Speak time and temperature

26.1 Overview

NOTE!!! First of all you should download the voice resources from our google drive:

<https://drive.google.com/drive/folders/0B6uNNXJ2z4CxaXFpakMxR0p1Unc?usp=sharing>

Then unzip it and find the 01 and 02 folder and put them into your TF card (should not larger than 32GB).

In this lesson, this is integrated project.

You can learn how to speak the time and temperature you get from the RTC module according to the value and the filename of digit /beep tone.

Each time you touch the buzzer beep a time. The knob controls the volume of the speak tone before you touch.

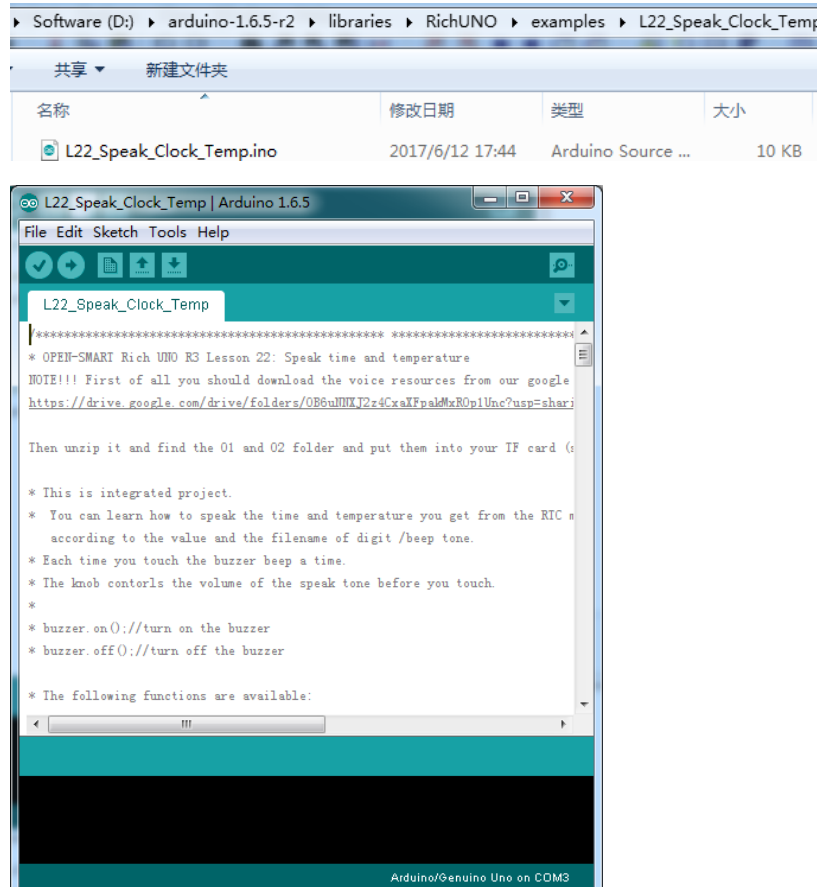
26.2 Upload code for lesson 22

1) Open lesson22 example

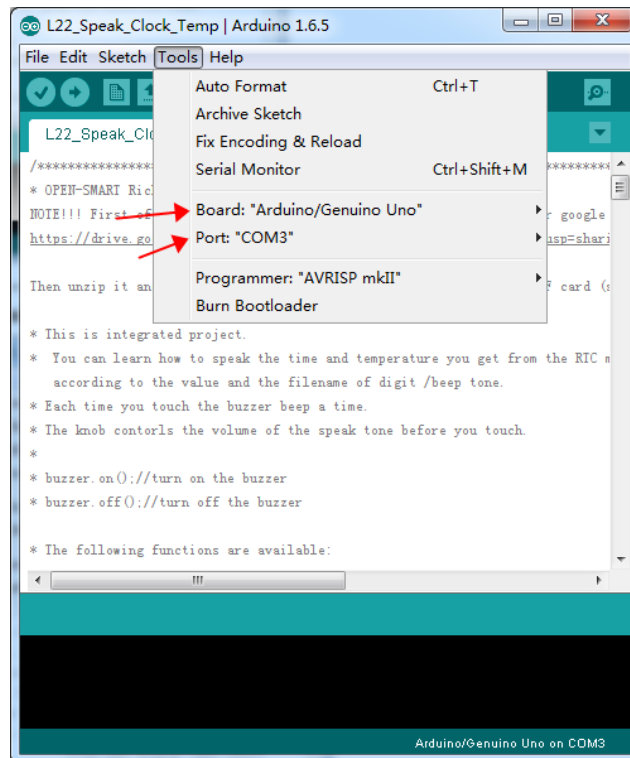
Double click the file in the directory of

\arduino-1.6.5-r2\libraries\RichUNO\examples\ L22_Speak_Clock_Temp

So that you find the window below:

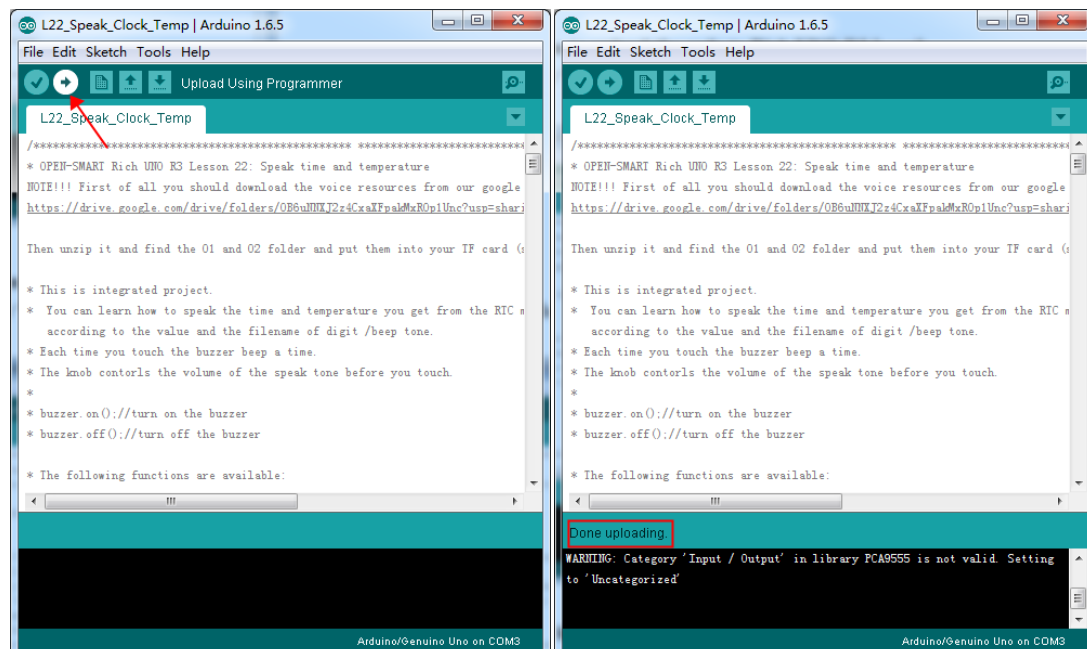


2) Check the Board and Serial Port



3) Click Upload button to upload the code to Rich UNO R3 board

When you see that “Done uploading”, congratulations, you have upload the code to the board successfully.



4) Check whether the board is running as you want

In this lesson, you can touch TCH1 to hear temperature and touch TCH2 to hear the time. You can rotate the knob to adjust the volume. Every time you touch, the buzzer will beep.

26.3 Learn the functions

1) You can find that at the beginning of all lessons code is

```
#include <Wire.h>
```

```
#include <SoftwareSerial.h>
```

Because the RichUNO library files use Wire.h and SoftwareSerial.h which is not in the RichUNO directory, if the example code wants to use RichUNO library, it should add those two line.

2) Other header files

In this lesson we use MP3, 4-digit display, RTC and touch sensor, so it should include their header files to control display more easily.

So you should add:

```
#include "RichUNOMP3.h"
```

```
#include "RichUNOTouchSensor.h"
```

```
#include "RichUNOTM1637.h"
```

```
#include "RichUNODS1307.h"
```

```
#include "RichUNOBuzzer.h"
```

```
#include "RichUNOLM75.h"
```

```
#include "RichUNOKnob.h"
```

3) Define class object

We have defined the class in the RichUNO library, and at any time you should define a object before you use it, and some object has parameter.

For example:

```
TM1637 disp(10,11); //define a TM1637 object, and display connect to D10 / D11 of Rich UNO R3 board
```

```
LM75 temper; // initialize an LM75 object "temper" for temperature
```

4) Two necessary functions

Each example code should include the two necessary functions:

```
void setup(){ }
```

```
void loop(){ }
```

Usually we write initialization code in setup function, and then write the code what you want the board to do in loop function which runs cyclically.

5) Functions

```
* buzzer.on(); //turn on the buzzer
```

```
* buzzer.off(); //turn off the buzzer
```

```
* knob.getAngle(); //read the angle of the knob, the range of angle is 0~280 degrees
```

```
* updateVolume(); //used to update volume according to the angle of the knob
```

```
* touch.get(); // return is the touch area corresponding Arduino pin number, if not then return -1
```

** mp3.setVolume(int8_t vol); // vol is 0~0x1e, 30 adjustable level*
** mp3.playWithFileName(int8_t directory, int8_t file); // play a song according to the folder name and prefix of its file name*

// directory (folder name) must be 01 02 03...09 10...99

// prefix of file name must be 001...009 010...099

mp3.getStatus(); // it will return the status of the mp3 player

#define STATUS_STOP 0

#define STATUS_PLAY 1

#define STATUS_PAUSE 2

#define STATUS_FORWARD 3

#define STATUS_REWIND 4

NOTE: If it is playing, it will return 1, you should wait until it return 0, then you can play next voice.

** clock.begin(); // The clock is on and the function must be called first*

** clock.getTime(void); // read the date and time, will be saved in the clock class variable*

After you call get time function, you can get the hour and minute from clock.hour and clock.minute

** Timer1.initialize(unsigned long microseconds); // set the timed length, the unit is microseconds*

** Timer1.attachInterrupt(TimingISR); // set interrupt routine function name, is the timing interrupt entry, when the time is up, it will call TimingISR function you write.*

** disp.init(); // initialization*

You should write it in the setup function initialize it.

** disp.display(int8_t DispData []); // display the numbers in the array*

// For example, the array is {1,2,3,4}, then show 1234

** disp.clearDisplay(); // clear the display, nothing display*

** disp.point(0); // Turn off the display colon and the next display takes effect*

** disp.point(1); // Turn on the display colon and the next display takes effect*

** temper.getTemperature(); // get the temperature value, the return of the float is the decimal, unit is in degrees Celsius.*

For example:

float celsius;

celsius = temper.getTemperatue(); // get temperature

SpeakTime(int8_t time[]); // used to speak the time

SpeakTemp(float temp); // used to speak the temperature "temp"

displayTemperature(int8_t temperature); // used to display temperature

27 Part List

Documents download link:

<https://drive.google.com/drive/folders/0B6uNNXJ2z4CxaXFpakMxR0p1Unc?usp=sharing>

OPEN-SMART Official Store:

<https://open-smart.aliexpress.com/store/1199788>

Technical Support: catalex_inc@163.com