**Introduction**

You should've learnt how to install Arduino IDE in the above **Get Started**. Now you can start with a simple experiment to learn the basic operation and code in the IDE.

**Components**

- 1 \* SunFounder Uno board

- 1 \* USB cable

- 1 \* Resistor (220Ω)

- 1 \* 1 LED

- 1 \* Breadboard

- Jumper wires

**Principle**

In this experiment, the operating voltage of the LED is 1.8V and the operating current is 10mA-20mA. The SunFounder Uno board can supply 5V or 3.3V power. In this experiment 5V is adopted, so the minimum resistance of the current limiting resistor should be (5 V to 1.8 V)/20 = 160 ohm. The 220ohm offered in the kit is suitable and you can also choose other resistors that meet the condition. The larger the resistance is, the dimmer the LED will get.

The schematic diagram:



**Principle:** Connect one end of the 220ohm resistor to pin 9 of the Uno and the other end to the anode (the long pin) of the LED, and the cathode (the short pin) of the LED to GND. When the pin 9 outputs high level, the current gets through the current limiting resistor to the anode of the LED. And since the cathode of the LED is connected to GND, the LED will light up.  When pin 9 outputs low level, the LED goes out.

**Experimental Procedures**

**Step 1:**Build the circuit (the pin with a curve is the anode of the LED)



**Step 2**: Open the code file

In the folder **Super Kit V2.0 for Arduino**you just downloaded and unzipped, you can see the folder for the corresponding lesson which includes two folders: code (the sketch) and circuit (Fritzing file). Go to the code folder and find the .ino file (.ino is the format of the sketch). Then double click to open it. You can also open an empty .ino file and type in the code we provide.

**Step 3:** Select the Board and Port

Before uploading the code, you need to select the **Board** and **Port**.

Click **Tools** ->**Board** and select **Arduino/Genuino Uno**. If your board is Mega2560, then select **Arduino/Genuino Uno Mega**or**Mega2560**. If it's Nano, select **Arduino** **Nano**.

 

Then select **Tools** ->**Port**. Your port should be different from mine.



**Step 4:** Upload the sketch to the SunFounder Uno board

Click the **Upload** icon to upload the code to the control board.



If "Done uploading" appears at the bottom of the window, it means the sketch has been successfully uploaded.

You should now see the LED blinking.



**Experiment Summary**

Through this experiment, you have learned how to turn on an LED. You can also change the blinking frequency of the LED by changing the num value in the delay function delay (num). For example, change it to **delay (250)**and you will find that the LED blinks more quickly.

**Code Analysis**

**Code Analysis 1-1 Define variables**

****

You should define every variable before using in case of making mistakes. This line defines a constant variable ledPin for the pin 9. In the following code, ledPin stands for pin 9. You can also directly use pin 9 instead.

**Code Analysis 1-2** **setup() function**

A typical Arduino program consists of two subprograms: setup() for initialization and loop() which contains the main body of the program.

The setup() function is usually used to initialize the digital pins and set them as input or output as well as the baud rate of the serial communication.

The loop() function contains what the MCU will run circularly. It will not stop unless something happens like power outages.

****

The setup() function here sets the ledPin as OUTPUT.

**pinMode(Pin)**: Configures the specified pin to behave either as an input or an output.

The void before the setup means that this function will not return a value. Even when no pins need to be initialized, you still need this function. Otherwise there will be errors in compiling.

You can write:

****

**Code Analysis 1-3  loop function**

****

This program is to set ledPin as HIGH to turn on the LED, with a delay of 500ms. Set ledPin as LOW to turn the LED off and also delay 500ms. The MCU will run this program repeatedly and you will see that the LED brightens for 500ms and then dims for 500ms. This on/off alternation will not stop until the control board runs out of energy.

**digitWrite**(Pin): Write a [HIGH](https://www.arduino.cc/en/Reference/Constants) or a [LOW](https://www.arduino.cc/en/Reference/Constants) value to a digital pin. When this pin has been set as output in pinModel(), its voltage will be set to the corresponding value: 5V (or 3.3V on 3.3V boards) for HIGH, 0V (ground) for LOW.