**Introduction**

In this experiment, you will learn how to turn on/off an LED by using an I/O port and a button. The "I/O port" refers to the INPUT and OUTPUT port. Here the INPUT port of the SunFounder Uno board is to read the output of an external device. Since the board itself has an LED (connected to Pin 13), you can use the LED for convenience.

**Components**

- 1 \* SunFounder Uno board

- 1 \* USB cable

- 1 \* Button

- 1 \* Resistor (10kΩ)

- 1 \* Capacitor 104

- 1 \* Breadboard

- Jumper wires

**Principle**

**Button**

Buttons are a common component used to control electronic devices. They are usually used as switches to connect or break circuits. Although buttons come in a variety of sizes and shapes, the one used here is a 6mm mini-button as shown in the following pictures.

Pin 1 is connected to pin 2 and pin 3 to pin 4. So you just need to connect either of pin 1 and pin 2 to pin 3 or pin 4.



The following is the internal structure of a button. The symbol on the right below is usually used to represent a button in circuits.





Since the pin 1 is connected to pin 2, and pin 3 to pin 4, when the button is pressed, the 4 pins are connected, thus closing the circuit.

The schematic diagram：



**Principle:**Connect one end of the buttons to pin 12 which connects with a pull-down resistor and a 0.1uF (104) capacitor (to eliminate jitter and output a stable level when the button is working). Connect the other end of the resistor to GND and one of the pins at the other end of the button to 5V. When the button is pressed, pin 12 is 5V (HIGH). Set the pin 12 as High level by programming and pin 13 (integrated with an LED) as High at the same time. Then release the button (pin 12 changes to LOW) and pin 13 is Low. So we will see the LED lights up and goes out alternately as the button is pressed and released.

**Experimental Procedures**

**Step 1:**Build the circuit



**Step 2**: Open the code file

**Step 3:** Select correct Board and Port

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

Now, press the button, and the LED on the SunFounder Uno board will light up.



**Experiment Summary**

You can also change the code to: when the button is pressed, if (buttonState=HIGH). The LED goes out (digitalWrite(ledPin, LOW)). When the button is released (the else), the LED lights up ((digitalWrite(ledPin, HIGH)). You only need to replace the code in**if** with those in **else**.

**Code Analysis**

**Code Analysis 2-1  Define variables**

****

Connect the button to pin 12. LED has been connected to pin 13. Define a variable buttonState to restore the state of the button.

**Code Analysis 2-2  Set the input and output status of the pins**

****

We need to know the status of the button in this experiment, so here set the buttonPin as INPUT; to set HIGH/LOW of the LED, we set LedPin as OUTPUT.

**Code Analysis 2-3  Read the status of the button**

****

buttonPin(Pin12) is a digital pin; here is to read the value of the button and store it in buttonState.

**digitalRead (Pin)**: Reads the value from a specified digital pin, either HIGH or LOW.

**Code Analysis 2-4  Turn on the LED when the button is pressed**

****

In this part, when the **buttonState** is High level, write ledPin as High and the LED will be turned on. As one end of the button has been connected to 5V and the other end to pin 12, when the button is pressed, pin 12 is 5V (HIGH). And then determine with theif(conditional); if the conditional is true, then the LED will light up.

Else means that when the if(conditional) is determined as false, run the code in else.