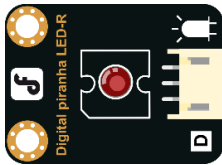


Project 3: Mini Lamp

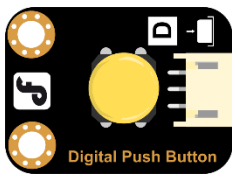
A lamp is something we use on a daily basis. It employs a light source and a simple switch. Let's make our own.

COMPONENT LIST:

1× Digital Piranha LED-R



1× Digital Push Button



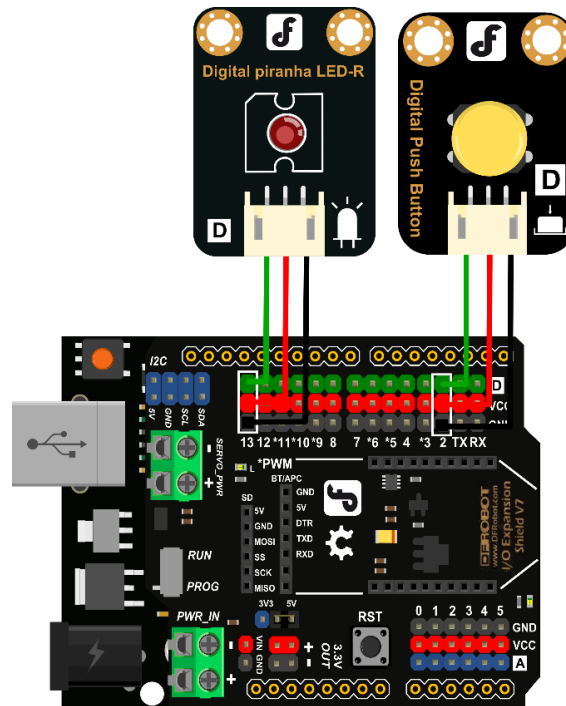
HARDWARE CONNECTIONS

Connect the Digital Push Button to digital pin 2

Connect the Digital Piranha LED-R to digital pin 13

Be sure that your power, ground and signal connections are correct or you risk damaging your components.

When you have connected the components and checked your connections, plug the 101 in to your PC with the USB cable so you can upload a program.



CODE INPUT

Sample Code 3-1:

```
// Item Three-Mini Lamp
int buttonPin = 2;           // button to be connected to digital pin No.2
int ledPin = 13;            // LED to be connected to digital pin No.13

int ledState = HIGH;        // ledState records the state of LED light
int buttonState;            //buttonState records the stat of the button

// lastbuttonState records the previous status of the button
int lastButtonState = LOW;

long lastDebounceTime = 0;
long debounceDelay = 50;    // eliminate the oscillation time

void setup() {
    pinMode(buttonPin, INPUT);
    pinMode(ledPin, OUTPUT);

    digitalWrite(ledPin, ledState);
}
```

```

void loop() {
    //reading stores buttonPin's data
    int reading = digitalRead(buttonPin);

    //record the time when data changes
    if (reading != lastButtonState) {
        lastDebounceTime = millis();
    }

    // Wait for 50ms and check if it's in line with the button's current status
    // If it's not in line with the button's current status, change the button's
    status

    //Meanwhile, if the button is set to be in the mode of HIGH (meaning the
    button has been pressed), led's status shall be changed.
    if ((millis() - lastDebounceTime) > debounceDelay) {
        if (reading != buttonState) {
            buttonState = reading;

            if (buttonState == HIGH) {
                ledState = !ledState;
            }
        }
    }

    digitalWrite(ledPin, ledState);

    //Change the button's previous state value
    lastButtonState = reading;
}

```

Use this sample code to implement the behavior we want.

You can copy and paste it in to the Arduino IDE, but if you want to develop your skills we recommend typing it out.

When you have finished, click “Verify” to check the code for syntax errors. If the code verifies successfully, you can upload it to your 101.

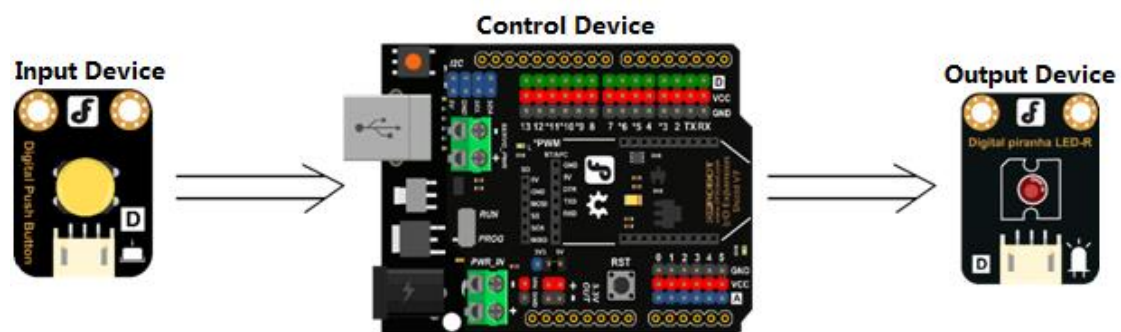
Once the code has uploaded, press the button. The LED should turn on.

When the button is pressed again, the LED should turn off.

HARDWARE ANALYSIS (DIGITAL INPUT – DIGITAL OUTPUT)

In the control device we have made:

The digital push button is the input unit and the LED is the output unit. Like our last experiment, we have one digital input and one digital output.



CODE ANALYSIS

We saw that the button is the input device and the LED is the output device.

```
pinMode(buttonPin, INPUT);  
pinMode(ledPin, OUTPUT);
```

`digitalWrite()` reads the state of the button

```
int reading = digitalRead(buttonPin);
```

When the button switches between HIGH and LOW states, there will be a brief period of oscillation. This is visually represented below:



To avoid signal errors such as this, we will wait when the signal changes and read it some time after.

Function `millis()` will record the time when data collected has changed

```
if (reading != lastButtonState) {  
    lastDebounceTime = millis();  
}
```

“millis()” is a function which measures duration. It is measured in milliseconds (1 second = 1000 milliseconds). This particular function will start counting when the 101 start running.

Wait for another 50ms and check if the button’ s signal has the same value with its current state. If not, change the state of the button. If the button is in the state of HIGH (meaning the button has been pressed), change the state of LED.

```
if ((millis() - lastDebounceTime) > debounceDelay) {  
    if (reading != buttonState) {  
        buttonState = reading;  
  
        if (buttonState == HIGH) {  
            ledState = !ledState;  
        }  
    }  
}
```

TRY IT YOURSELF

Doorbell with a Mini Lamp

Nowadays, people are always listening to music on their headphones. This means that if they are at home and the doorbell rings, they cannot hear it.

A doorbell with a mini lamp would solve this problem. If the doorbell button is pressed, the lamp will light up.

This idea could also work for old people and the hearing-impaired.