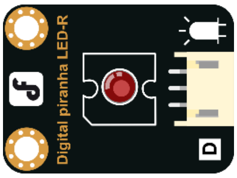
Project 1: Make an LED blink

After knowing the basic idea of what Arduino platform and Genuino 101 is and how digital and analog signals work, it's the time for us build our first project, blink an LED light.

In the very first chapter, we uploaded the blink code to light up the on-board LED light for a simple test. Now, instead of just using the on-board LED, we will attach an LED light module and set it to be blinking in a desired pattern.

COMPPONMENT LIST:

1× Digital Piranha LED-R

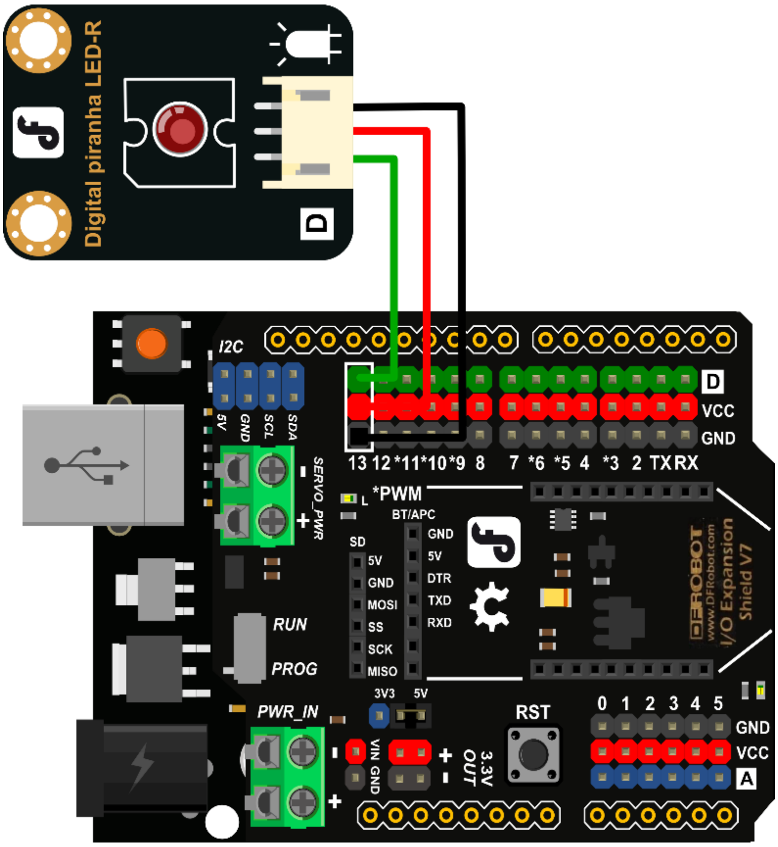


Note: To save more space, the I/O Expansion Shield is not listed here as we will be using it for all rest projects. You can leave it installed on your 101 for convenience.

HARDWARE CONNECTIONS

Connect the Digital Piranha LED-R module to digital pin 13 on the Arduino. Make sure your power, ground and signal connections are correct or you risk damaging your components.

When all your connections are made, connect the 101 to your computer via the USB cable.


Digital piranha LED-R
D
I2C
5V
GND
SCL
SDA
SERVO_PWR
13 12 *11 *10 *9 8 7 *6 *5 4 *3 2 TX RX
D
VCC
GND
*PWM
L
BT/APC
SD
5V
GND
MOSI
SS
SCK
MISO
GND
5V
DTR
TXD
RXD
RUN
PROG
PWR_IN
3V3
5V
RST
VIN
GND
3.3V OUT
0 1 2 3 4 5
GND
VCC
A
DFROBOT
www.DFRobot.com
I/O Expansion Shield V7

CODE INPUT

Open the Arduino IDE. Although you can open the code via “File” > “Examples” > “01.Basic” > “Blink”, we encourage you to type the code out by hand to develop your skills.

The code is displayed below:

Sample Code 1-1:

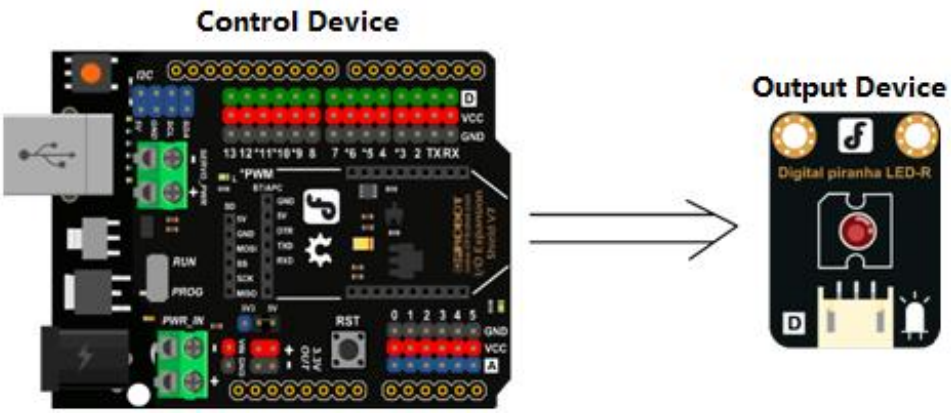
```
     // Item One —— LED Blink
    /*
      Description: LED light on first; LED light off one second later; loop the
above action
    */
    int ledPin = 13;
    void setup() {
          pinMode(ledPin, OUTPUT);
    }
    void loop() {
          digitalWrite(ledPin,HIGH);
          delay(1000);
          digitalWrite(ledPin,LOW);
          delay(1000);
    }
```

Once the code is input, click “Verify” in Arduino IDE. The IDE will check the code for errors. If there are no errors, you are ready to upload. Click “upload” and the IDE will send the code to your 101. When the upload is complete, you external LED will blink!

Now let’s review code and see how it works.

HARDWARE ANALYSIS (DIGITAL OUTPUT)

In the previous sessions, we’ ve mentioned input and output. The whole device is composed of two parts: control and output. The 101 is the control unit and the digital Piranha LED-R module is the output unit. No input unit is included in this system.



CODE ANALYSIS

Arduino’ s code is composed of two main parts:

```
“void setup() {”
```

This part of the code is executed when the device is initialized. It runs only once. It is used to declare outputs and inputs of the device.

```
“void loop() {“
```

This part of the code is executed after “void setup(){ ” once initialized, it runs in a loop forever (or until the device is powered off). This is where all the key arguments of the code are contained.

Check out this code:

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

The code above shows that ledPin is set up as the output mode. The comma in the brackets is the code syntax and needs to be there to separate the parameters.

What is ledPin?

Examine the first line of the code below:

```
int ledPin = 13;
```

We’ ve named pin13 “ledPin”, therefore ledPin represents pin 13 on the 101。 “int” stands for *integer* and means that “ledPin” is an integer

What if you want to attach the LED to pin10 instead of pin13?

```
pinMode(10, OUTPUT);
```

You just have to change the pin in the code from 13 to 10.

Let’ s take a look at the function “loop()”, which contains only one function digitalWrite().

The syntax of function “digitalWrite()” is as follows:

```
digitalWrite(pin,value)
```

When we set pinMode() as OUTPUT, the voltage will be either be HIGH (5V or 3.3V for the control board) or LOW (0V). “HIGH” means on and “LOW” means off.

```
digitalWrite(ledPin,HIGH);   // LED on

digitalWrite(ledPin,LOW);    // LED off
```

When HIGH is included in digitalWrite() just as it is shown above, pin 13 will be HIGH and the LED will be turned on. When LOW is included in digitalWrite() just as it is shown above, pin 13 will be LOW and the LED will be turned off.

Don’ t forget the following line:

```
delay(1000);
```

The figure within the brackets means the amount of milliseconds of delay. There are 1000 milliseconds in 1 second, so this means the duration of delay is 1 second. The LED will turn on for 1 second and then turn off for 1 second continuously.

You might notice "//" and "/*…*/" located ahead of the code:

```
 // Item One—LED Blink
/
   Description: LED on and off alternately every other second
 */
```

Anything written after "//" or inside "/*…*/" will not be compiled. We use this function to write comments about the code in plain English. "//" will comment for one line of code. "/*…*/" will comment for multiple lines. The Arduino IDE will signify this by automatically turning the text grey.

TRY IT YOURSELF

Try to make the LED light blink faster or slower. For example, keep the LED light off for 5 seconds and then quickly blink for 250 milliseconds