14 December 2019

Introduction to Git and Embedded Programming (Input device)

Git 介绍和嵌入式编程（输入设备）

Week 3 assignments: 第三周作业

- Publish, modify and re-publish several times the page using git workflow
  使用 Git 工作流程发布、修改、反复发布你的网页
- Sketch of your final project
  学期末会有一个最终作品，请初步规划你的最终作品的功能等。
- Create a program to turn ON LED when then button is pressed and keep it ON after it is released.
  创建一个程序，当按下按钮时将 LED 点亮，并在释放后使其保持点亮状态。
- Create a page to document your progress of week 3 创建一个网页去记录本周的收获，拍下项目成功运行的照片上传。
  - What did you do and learn? 你做了什么？学到了什么？
  - What are your experiences? (problems, solution, etc.).你经历了什么？（苦难有哪些？解决方案是？等）
  - Upload a picture and the programming code to your website.上传照片，编你的网站。
- Create a page to document your progress of week 3 创建一个网页去记录本周的收获，拍下项目成功运行的照片上传。

Please bring with you: 带上

- Your personal laptop 个人笔记本电脑
- A mouse 鼠标

And please install the following software: 下载下面软件

- Arduino is an open source electronic prototyping platform. Arduino 是一个开源的电子快速原型平台。
  https://www.arduino.cc/en/Main/Software

Do not hesitate to contact us if you have any question. 有疑问可以随时联系老师。

Reference: 参考案例网站，里面有很多代码范例

https://wiki.dfrobot.com/

重力传感器代码案例

https://wiki.dfrobot.com/Gravity__DS18B20_Temperature_Sensor__Arduino_Compatible__V2_SKU__DFR0024

光线传感器代码案例

https://wiki.dfrobot.com/DFRobot_Ambient_Light_Sensor_SKU_DFR0026

Getting Started With Arduino Arduino 入门

Before you start controlling the world around you, you will need to set up the software to program your board. 在开始 “控制周围世界” 之前，您需要设置软件以对开发板进行编程。

The Arduino Software (IDE) allows you to write programs and upload them to your board. The programming cycle on Arduino is basically as follows:
Arduino 软件（IDE）允许您编写程序并将其上传到开发板上。 Arduino 上的编程周期基本上如下：

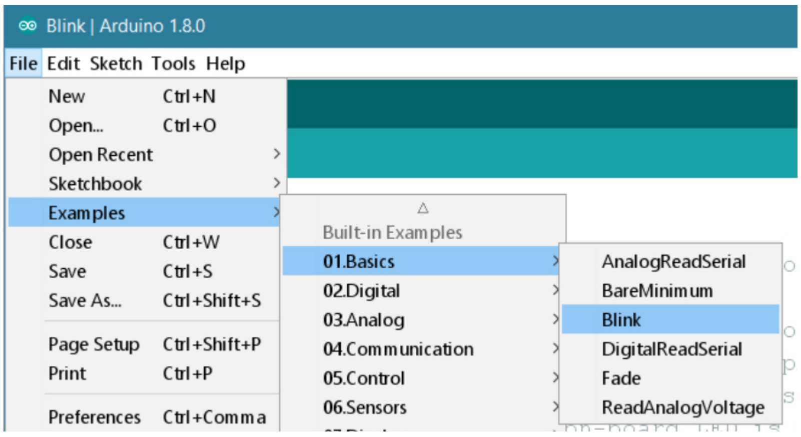
1. Plug your board into a USB port on your computer
2. Write a sketch that will bring the board to life.
3. Upload this sketch to the board through the USB connection and wait a couple of seconds for the board to restart.
4. The board executes (performs) the sketch that you wrote.

1.将开发板插入计算机上的 USB 端口
2.编写代码，让电路板动起来。
3.通过 USB 连接将此代码上传到电路板上，并等待几秒钟以使电路板重新启动。
4.电路板执行你编写的代码。

Open your first sketch 打开你的第一个代码程序

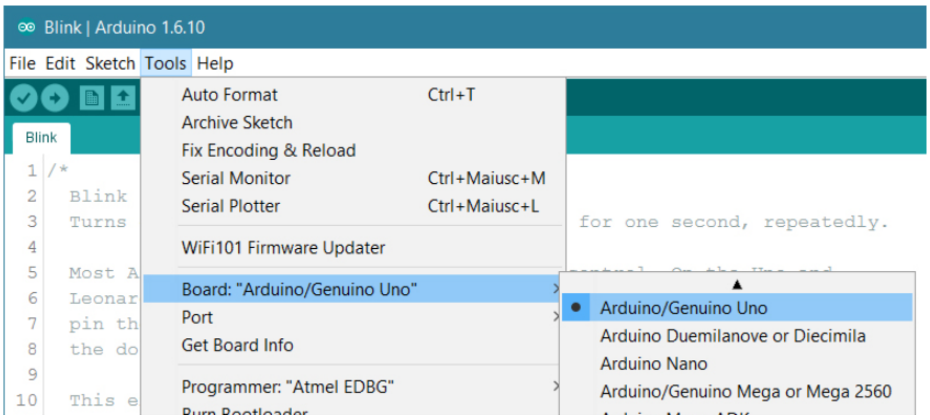
The LED blinking sketch is the first program that you should run to test whether your Arduino board is working and is configured correctly.
LED 闪烁的代码是您应该运行的第一个程序，用于测试 Arduino 板是否正常工作并正确配置。

Open the LED blink example sketch: **File>Examples>01.Basics>Blink**.打开下面路径



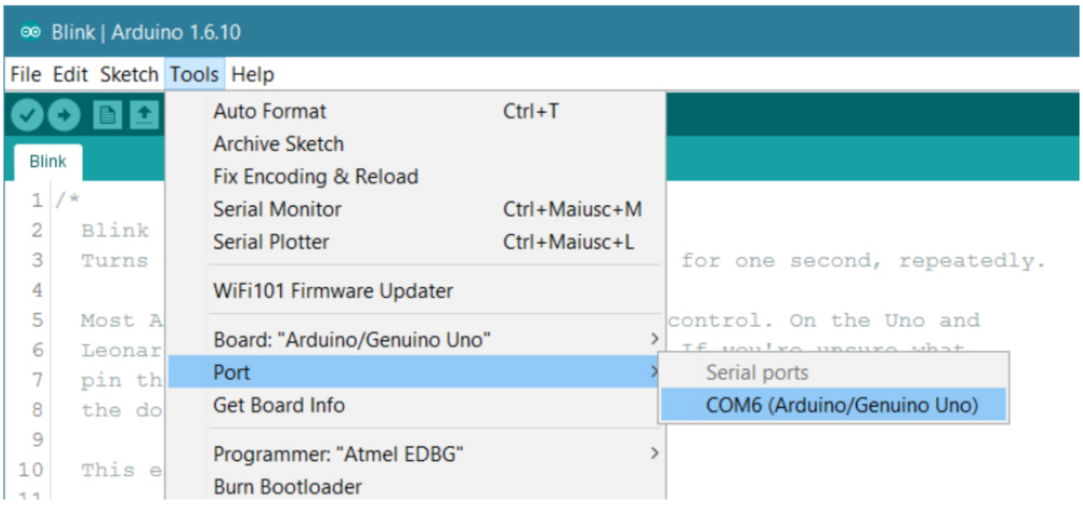
Select your board type and port 选择您的板子类型和端口

You'll need to select the entry in the Tools > Board menu that corresponds to your Arduino board. 您需要在 Tools> Board 菜单中选择与 Arduino 开发板相对应的条目。



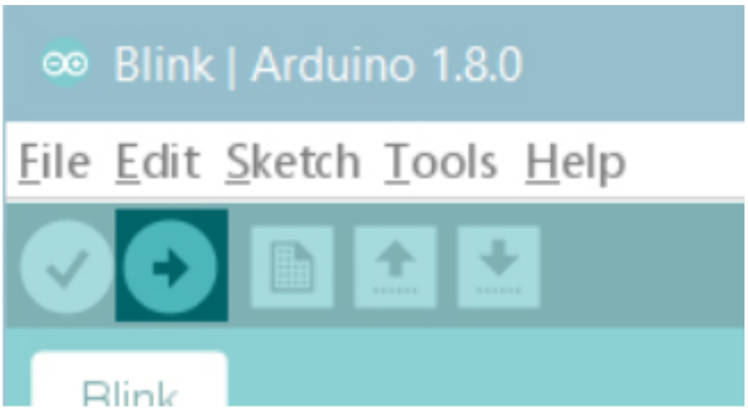
```
/*
  Blink
  Turns                                              for one second, repeatedly.

  Most A
  Leonar
  pin th
  the do

  This e
```

Select the serial device of the board from the Tools /Serial Port menu. This is likely to be COM3 or higher (COM1 and COM2 are usually reserved for hardware serial ports). To find out, you can disconnect your board and re-open the menu; the entry that disappears should be the Arduino board. Reconnect the board and select that serial port.
从工具/面板中选择串行设备。 串行端口菜单。 这可能是 COM3 或更高版本（COM1 和 COM2 通常为硬件串行端口保留）。 要找出答案，您可以断开板子并重新打开菜单； 消失的条目应该是 Arduino 开发板。 重新连接电路板，然后选择该串行端口。



Upload the program 上传程序

Now, simply click the "Upload" button in the environment. Wait a few seconds - you should see the RX and TX leds on the board flashing. If the upload is successful, the message "Done uploading." will appear in the status bar.



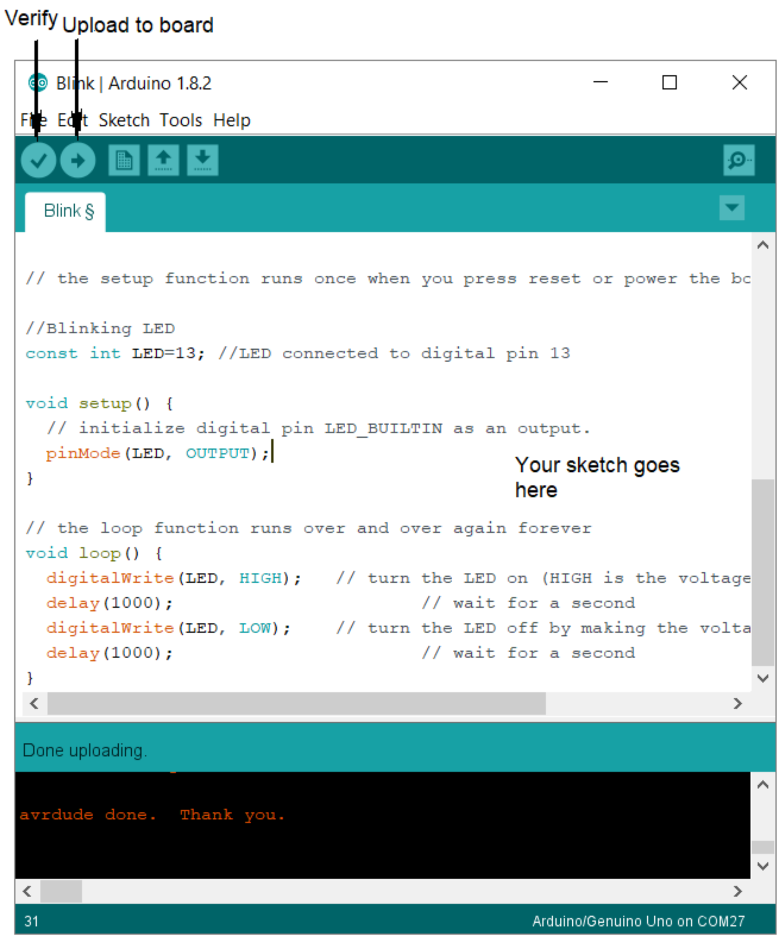
现在，只需单击环境中的 “上传” 按钮。 等待几秒钟-您应该看到板上的 RX 和 TX LED 闪烁。 如果上传成功，则显示消息 “完成上传” 。 将显示在状态栏中。

A few seconds after the upload finishes, you should see the pin 13 (L) LED on the board start to blink (in orange).
上传完成几秒钟后，您应该看到板上的针脚 13（L）LED 开始闪烁（橙色）。

Example 4-1. Blinking LED 代码范例

```
//Blinking LED
const int LED=13; //LED connected to digital pin 13

void setup() {
  // initialize digital pin LED_BUILTIN as an output.
  pinMode(LED, OUTPUT);
}

// the loop function runs over and over again forever
void loop() {
  digitalWrite(LED, HIGH);   // turn the LED on (HIGH is the voltage level)
  delay(1000);                       // wait for a second
  digitalWrite(LED, LOW);    // turn the LED off by making the voltage LOW
  delay(1000);                       // wait for a second
}
```

The Arduino IDE with your first sketch loaded. 第一个程序上传。



```
// the setup function runs once when you press reset or power the bc

//Blinking LED
const int LED=13; //LED connected to digital pin 13

void setup() {
  // initialize digital pin LED_BUILTIN as an output.
  pinMode(LED, OUTPUT);
}

// the loop function runs over and over again forever
void loop() {
  digitalWrite(LED, HIGH);   // turn the LED on (HIGH is the voltage
  delay(1000);                       // wait for a second
  digitalWrite(LED, LOW);    // turn the LED off by making the volta
  delay(1000);                       // wait for a second
}
```

```
avrdude done.  Thank you.
```

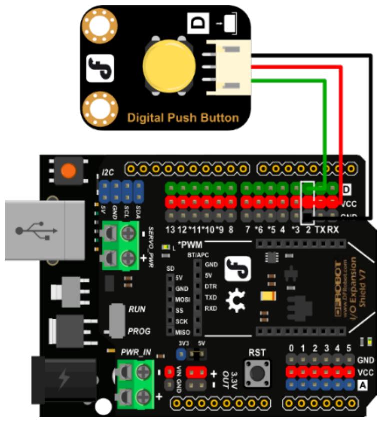
To Sum up, this program does this:
• Turns pin 13 into an output (just once at the beginning)
• Enters a loop
• Switches on the LED connected to pin 13
• Waits for a second
• Switches off the LED connected to pin 13

• Waits for a second
• Goes back to beginning of the loop

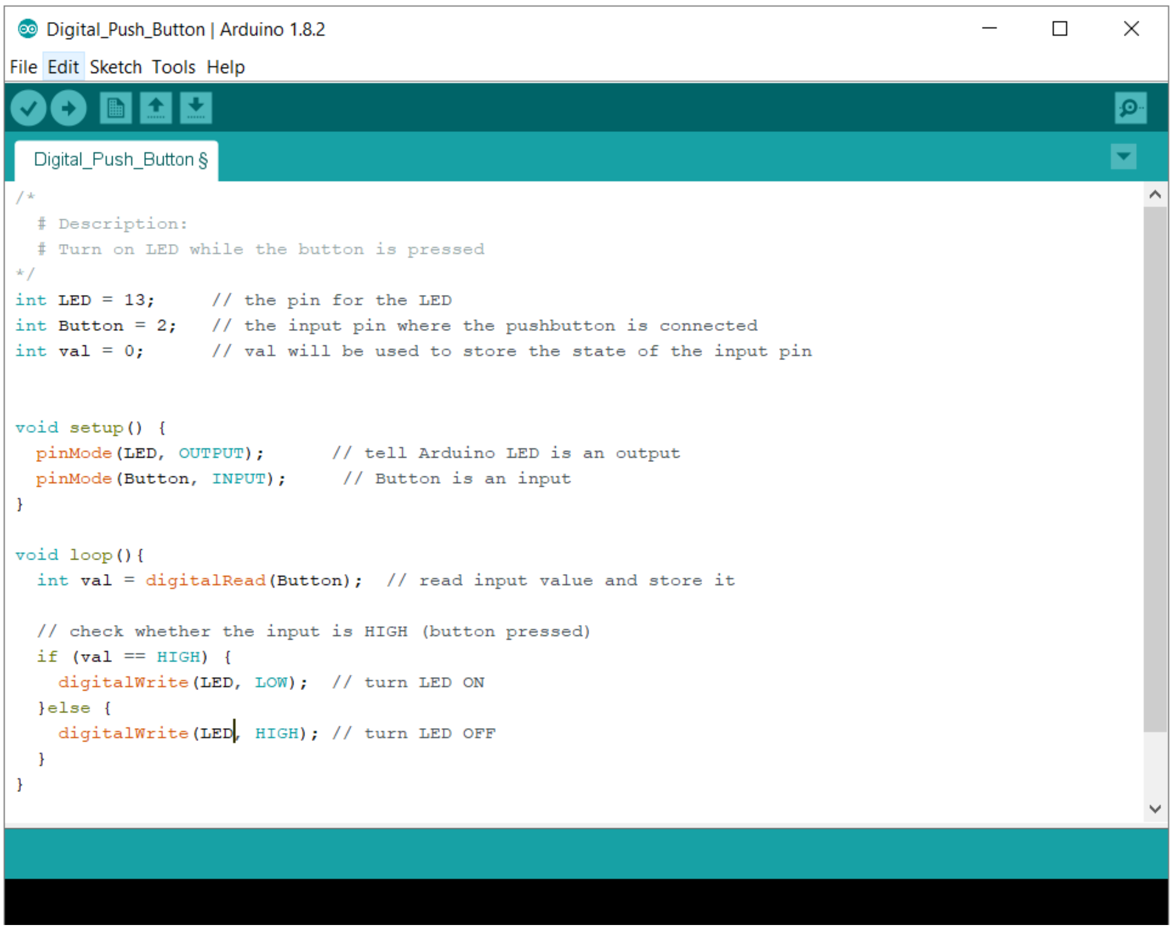
总而言之，该程序执行以下操作：
•将引脚 13 变成输出（仅在开始时一次）
•进入循环
•接通连接到引脚 13 的 LED
•等待一秒钟
•关闭连接到引脚 13 的 LED
•等待一秒钟
•返回循环的开始

Using a Pushbutton to Control the LED 使用按钮控制 LED

Hooking up a pushbutton 连接按钮



digitalRead() checks to see whether there is any voltage applied to the pin that you specify between parentheses, and returns a value of HIGH or LOW, depending on its findings. With digitalRead(), you can “ask a question” of Arduino and receive an answer that can be stored in memory somewhere and used to make decisions immediately or later.
digitalRead（）检查在括号之间指定的引脚上是否施加了任何电压，并根据发现的结果返回 HIGH 或 LOW 值。 到目前为止，您使用的其他说明还没有返回任何信息，它们只是执行了我

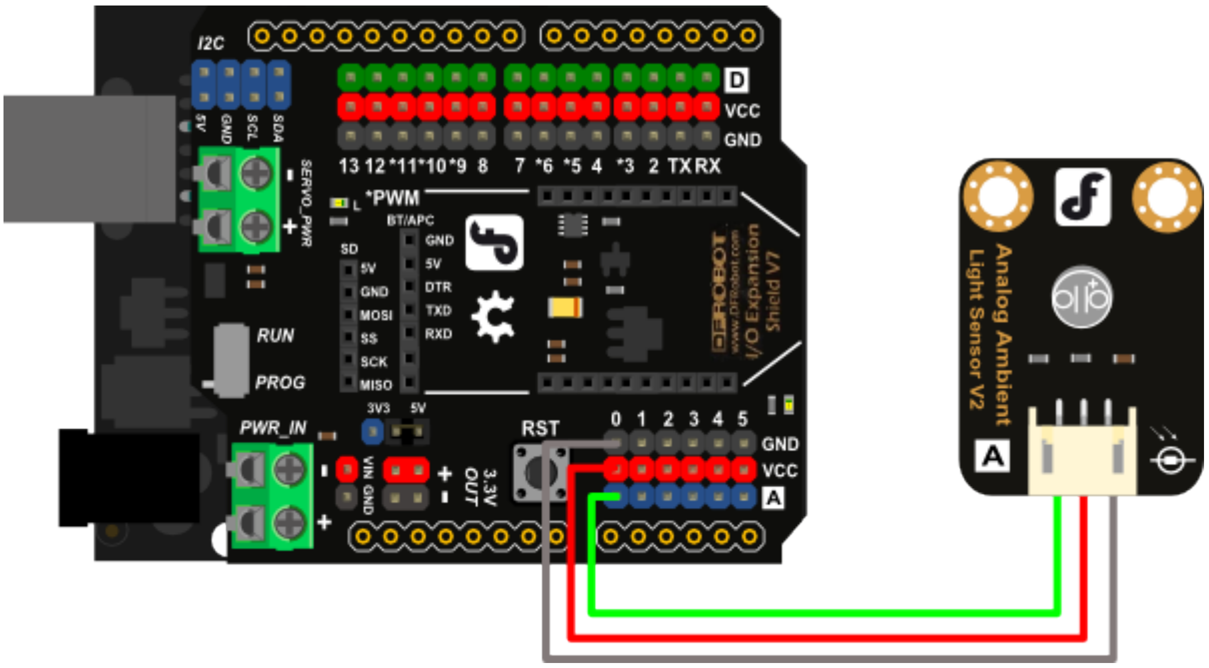

Digital_Push_Button | Arduino 1.8.2
File Edit Sketch Tools Help
Digital_Push_Button §

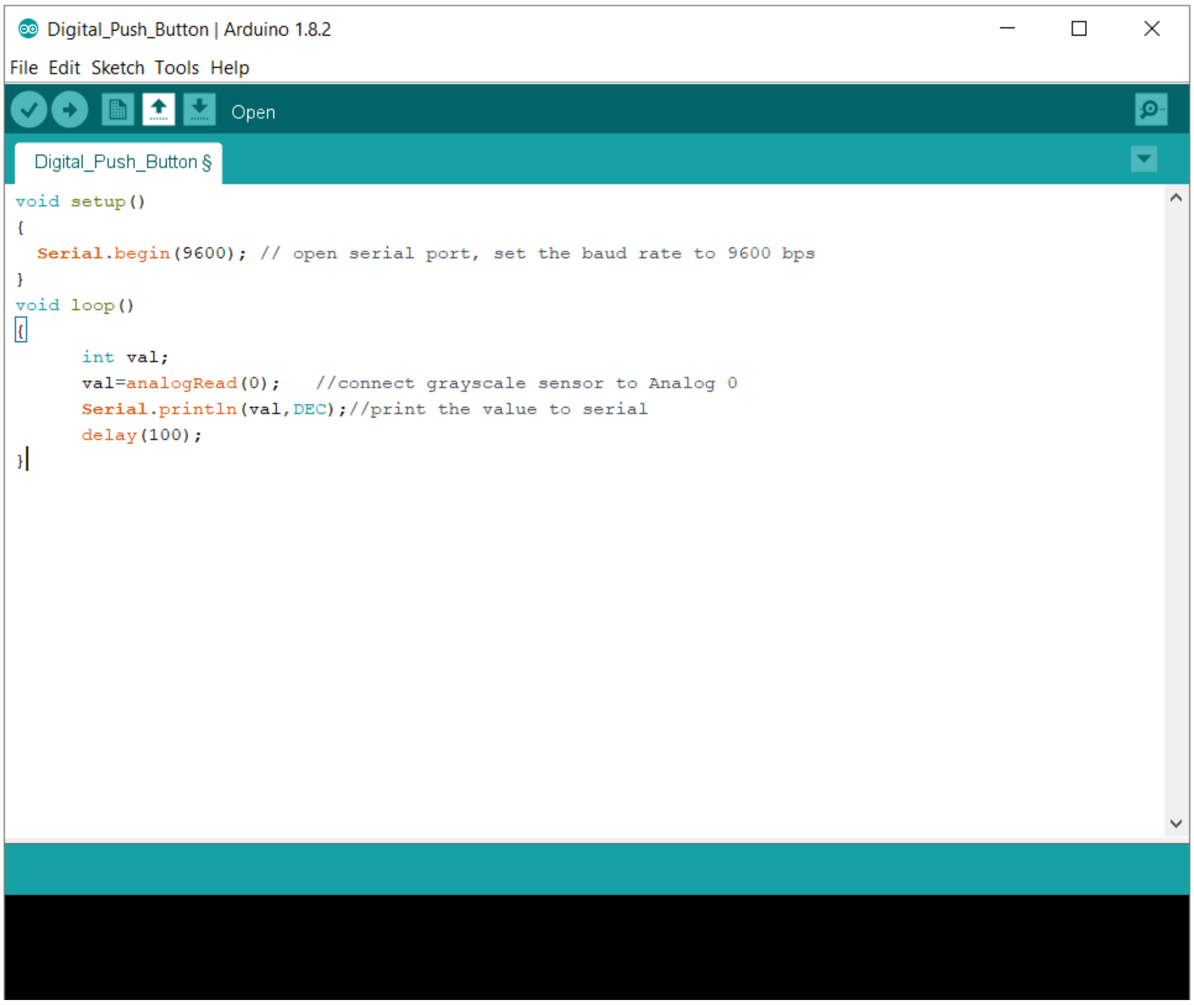
```
/*
  # Description:
  # Turn on LED while the button is pressed
*/
int LED = 13;     // the pin for the LED
int Button = 2;   // the input pin where the pushbutton is connected
int val = 0;      // val will be used to store the state of the input pin


void setup() {
  pinMode(LED, OUTPUT);       // tell Arduino LED is an output
  pinMode(Button, INPUT);      // Button is an input
}

void loop(){
  int val = digitalRead(Button);  // read input value and store it

  // check whether the input is HIGH (button pressed)
  if (val == HIGH) {
    digitalWrite(LED, LOW);  // turn LED ON
  }else {
    digitalWrite(LED, HIGH); // turn LED OFF
  }
}
```

们要求他们执行的操作。 但是这种功能有点受限制，因为它将迫使您坚持非常可预测的指令序列，而无需外界输入。 使用 digitalRead（），您可以“向 Arduino 问一个问题”，并接收一个答案，该答案可以存储在某个地方的内存中，并用于立即或稍后做出决策。

```
/*
  Turn on LED when the button is pressed and keep it on after it is released
*/
int LED = 7;  // the pin for the LED
int BUTTON = 4; // the input pin where the push button is connected
int value = 0;  // value will be used to store the state
int old_value = 0;  // this variable stores the previous value of "value"
int state = 0; // 0 =LED off and 1 =LED on
// the setup function runs once when you press reset or power the board
void setup() {
  // initialize digital pin LED_BUILTIN as an output.
  pinMode(LED, OUTPUT);
  pinMode(BUTTON, INPUT);
}

// the loop function runs over and over again forever
void loop() {
  value = digitalRead(BUTTON);//read input value and store it, yum, fresh

  //check if there was a transition
  //check if the button is pressed ( input is HIGH) and change the state
  if((value == HIGH)&&(old_value == LOW)){
    state = 1 - state;
    delay(50);
  }

  old_value =value; //valus is now old, let's store it

  if(state == 1){
    digitalWrite(LED, HIGH);    // turn the LED on
  }else{
    digitalWrite(LED, LOW);    // turn the LED off
  }
```

Using Analog Ambient Light Sensor 使用模拟环境光传感器




Digital_Push_Button | Arduino 1.8.2
File Edit Sketch Tools Help
Open
Digital_Push_Button §

```
void setup()
{
  Serial.begin(9600); // open serial port, set the baud rate to 9600 bps
}
void loop()
{
      int val;
      val=analogRead(0);   //connect grayscale sensor to Analog 0
      Serial.println(val,DEC);//print the value to serial
      delay(100);
}
```