

Arduino 与 Processing 通讯二：从 Arduino 中接收 Processing 的数据

Phodal Huang

October 24, 2017

目录

步骤 1: From Processing...	3
步骤 2: ...to Arduino	4

玩点什么: <https://www.wandianshenme.com>

原文链接:<https://www.wandianshenme.com/play/arduino-processing-communication-part-2-arduino>

在上一个玩法《[Arduino 与 Processing 通讯一：从 Processing 中接收 Arduino 的数据](#)》里，我们介绍了如何从 Processing 中接收 Arduino 的数据。在这一篇玩法里，我们将介绍如何从 Arduino 里接收 Processing 发送的数据。

步骤 1: From Processing...

所以，我们已经将数据从 Arduino 发送给 Processing，但是如果我们能以另一种方式发送数据：从 Processing 到 Arduino 呢？小菜一碟！

让我们先从 Processing 方面开始。它的开始过程非常像我们的最后一个程序：我们导入串行 (Serial) 库，并在顶部为我们的端口，声明一个全局 Serial 对象变量。在我们的 setup() 方法中，我们找到我们的端口，并在该端口上初始化串行通信，并使用我们的 Serial 变量 9600 波特。我们还将使用 size() 命令给我们一个点击窗口，这将触发我们的程序通过串口发送一些东西到 Arduino。

```
1 import processing.serial.*;
2
3 Serial myPort; // Create object from Serial class
4
5 void setup()
6 {
7   size(200,200); //make our canvas 200 x 200 pixels big
8   String portName = Serial.list()[0]; //change the 0 to a 1 or 2 etc. to
   match your port
9   myPort = new Serial(this, portName, 9600);
10 }
```

在我们的 draw() 循环中，我们使用 Processing Serial 库中的 write 方法，发送我们想要的任何东西到串口上。对于此程序，每当我们在“Processing”窗口中单击鼠标时，我们将发送一个“1”。我们还会在控制台上打印出来，看看我们实际上是在发送一些东西。如果我们没有点击，我们会发送一个‘0’。

```
1 void draw() {
2   if (mousePressed == true)
3   { //if we clicked in the window
4     myPort.write('1'); //send a 1
5     println("1");
```

```
6 } else
7 {
    //otherwise myPort.write('0'); //send
    a 0
8 }
9 }
```

如果您运行此代码，则只要在窗口中单击鼠标，就会看到一组 **1** 显示在控制台区域中。整齐的排列着！但是我们如何从 **Arduino** 寻找这些 **1**？我们可以做些什么呢？

步骤 2: ...to Arduino

好！在这个页面上，我们将从 **Processing** 处理这些 **1**，如果我们看到它们，我们将在 **13** 引脚上打开一个 **LED**（在一些 **Arduino** 上，如 **Uno** 开发上的 **13** 引脚连接着 **LED**，所以你不需要一个外部 **LED** 才能看到）。

在我们的 **Arduino** 程序的顶部，我们需要两个全局变量：

- 一个用于保存来自 **Processing** 的数据
- 另一个用于告诉 **Arduino** 我们的 **LED** 被引导到哪个引脚。

```
1 char val; // Data received from the serial port
2 int ledPin = 13; // Set the pin to digital I/O 13
```

接下来，在我们的 `setup()` 方法中，我们将 **LED** 引脚设置为输出，因为我们为 **LED** 供电，我们将以 **9600** 波特率开始串行通信。

```
1 void setup() {
2   pinMode(ledPin, OUTPUT); // Set pin as OUTPUT
3   Serial.begin(9600); // Start serial communication at 9600 bps
4 }
```

最后，在 `loop()` 方法中，我们将看看输入的串行数据。如果我们看到一个 `'1'`，我们将 **LED** 设置为高（或开），如果不是（我们看到一个 `'0'`），我们将关闭 **LED**。在循环结束时，我们将添加一个短暂的延迟，以帮助 **Arduino** 跟上串行数据流。

```
1 void loop() {
2   if (Serial.available())
3   { // If data is available to read,
4     val = Serial.read(); // read it and store it in val
```

```
5   }
6   if (val == '1')
7 { // If 1 was received    digitalWrite(ledPin, HIGH); // turn the LED on
8   } else {
9     digitalWrite(ledPin, LOW); // otherwise turn it off
10  }
11  delay(10); // Wait 10 milliseconds for next reading
12 }
```

瞧!

如果我们将此代码加载到我们的 **Arduino** 上，并从上一页运行 **Processing** 程序，那么只需点击 **Processing** 画布，就可以打开 **Arduino** 的第 13 脚连接的 **LED**。

原文链接: [https:// learn.sparkfun.com/ tutorials/ connecting-arduino-to-processing#from-processing](https://learn.sparkfun.com/tutorials/connecting-arduino-to-processing#from-processing)

原文链接: <https://www.wandianshenme.com/play/arduino-processing-communication-part-2-ardu>