

Arduino基础入门篇23—串口通讯

原创

TonyIOT

2018-09-26 08:21:41

© 14887

☆ 收藏

47

收起

分类专栏： [Arduino](#)

版权声明：本文为博主原创文章，遵循 CC 4.0 BY-SA 版权协议，转载请附上原文出处链接和本声明。

本文链接：<https://blog.csdn.net/TonyIOT/article/details/82848062>

在很多时候，Arduino需要和其他设备相互通讯，而最常见最简单的方式就是串口通讯。本篇介绍Arduino硬件串口通讯，了解相关函数的使用。

1. 硬件串口

在PC机上最常见的串行通讯协议是RS-232串行协议，而各种微控制器(单片机)上采用的是TTL串行协议，两者电平不同，需要经过相应电平转换才能进行相互通讯。

Arduino Uno R3开发板上，硬件串口位于Rx(0)和Tx(1)引脚上，Arduino的USB口通过转换芯片与这两个引脚连接。该转换芯片会通过USB接口在PC机上虚拟出一个用于Arduino通讯的串口，我们下载程序也是通过串口进行的。



2. 相关函数介绍

Arduino提供的串口通讯函数非常丰富，详细介绍参考[Arduino官网串口使用指南](#)。

Serial.begin()

- 描述：开启串口，通常置于setup()函数中。
- 原型：
 - **Serial.begin(speed)**
 - **Serial.begin(speed, config)**
- 参数：
 - speed：波特率，一般取值9600,115200等。
 - config：设置数据位、校验位和停止位。默认SERIAL_8N1表示8个数据位，无校验位，1个停止位。
- 返回值：无。

Serial.end()

- 描述：禁止串口传输。此时串口Rx和Tx可以作为数字IO引脚使用。
- 原型：**Serial.end()**
- 参数：无。
- 返回值：无。

Serial.print()

- 描述：串口输出数据，写入字符数据到串口。
- 原型：
 - **Serial.print(val)**
 - **Serial.print(val, format)**
- 参数：
 - val：打印的值，任意数据类型。
 - config：输出的数据格式。BIN(二进制)、OCT(八进制)、DEC(十进制)、HEX(十六进制)。对于浮点数，此参数指定要使用的小数位数。
- 示例：
 - Serial.print(78, BIN) 得到 "1001110"
 - Serial.print(78, OCT) 得到 "116"
 - Serial.print(78, DEC) 得到 "78"
 - Serial.print(78, HEX) 得到 "4E"
 - Serial.print(1.23456, 0) 得到 "1"
 - Serial.print(1.23456, 2) 得到 "1.23"
 - Serial.print(1.23456, 4) 得到 "1.2346"
 - Serial.print('N') 得到 "N"
 - Serial.print("Hello world.") 得到 "Hello world."
- 返回值：返回写入的字节数。

Serial.println()

- 描述：串口输出数据并换行。
- 原型：
 - **Serial.println(val)**
 - **Serial.println(val, format)**
- 参数：
 - val：打印的值，任意数据类型。
 - config：输出的数据格式。
- 返回值：返回写入的字节数。

Serial.available()

- 描述：判断串口缓冲区的状态，返回从串口缓冲区读取的字节数。
- 原型：**Serial.available()**
- 参数：无。
- 返回值：可读取的字节数。

Serial.read()

- 描述：读取串口数据，一次读一个字符，读完后删除已读数据。
- 原型：**Serial.read()**
- 参数：无。
- 返回值：返回串口缓存中第一个可读字节，当没有可读数据时返回-1，整数类型。

Serial.readBytes()

- 描述：从串口读取指定长度的字符到缓存数组。
- 原型：**Serial.readBytes(buffer, length)**
- 参数：
 - buffer：缓存变量。
 - length：设定的读取长度。
- 返回值：返回存入缓存的字符数。

3. 实验步骤

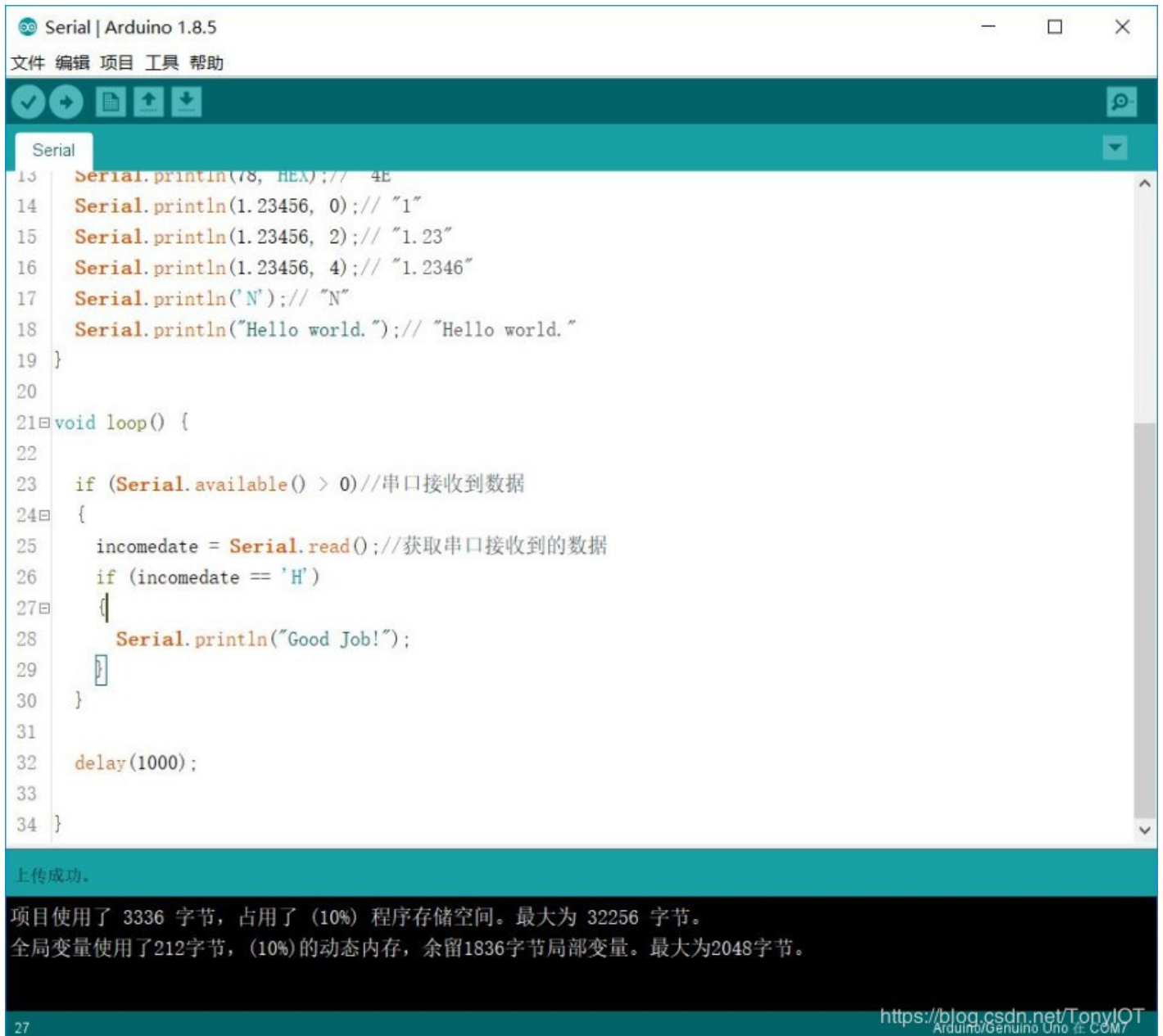
1. 通过数据线连接开发板到PC机。

2. 新建sketch，拷贝如下代码替换自动生成的代码并进行保存。

```
1  /*
2  |   Serial
3  |   串口通讯实验
   */
```

```
4  int incomedate = 0;
5  void setup() {
6
7      Serial.begin(9600); //设置串口波特率9600
8
9      Serial.println(78, BIN); // "1001110"
10     Serial.println(78, OCT); // "116"
11     Serial.println(78, DEC); // "78"
12     Serial.println(78, HEX); // "4E"
13     Serial.println(1.23456, 0); // "1"
14     Serial.println(1.23456, 2); // "1.23"
15     Serial.println(1.23456, 4); // "1.2346"
16     Serial.println('N'); // "N"
17     Serial.println("Hello world."); // "Hello world."
18 }
19
20 void loop() {
21
22     if (Serial.available() > 0) // 串口接收到数据
23     {
24         incomedate = Serial.read(); // 获取串口接收到的数据
25         if (incomedate == 'H')
26         {
27             Serial.println("Good Job!");
28         }
29     }
30
31     delay(1000);
32
33 }
34
```

3. 设置好对应端口号和开发板类型进行程序下载。



```
Serial | Arduino 1.8.5
文件 编辑 项目 工具 帮助

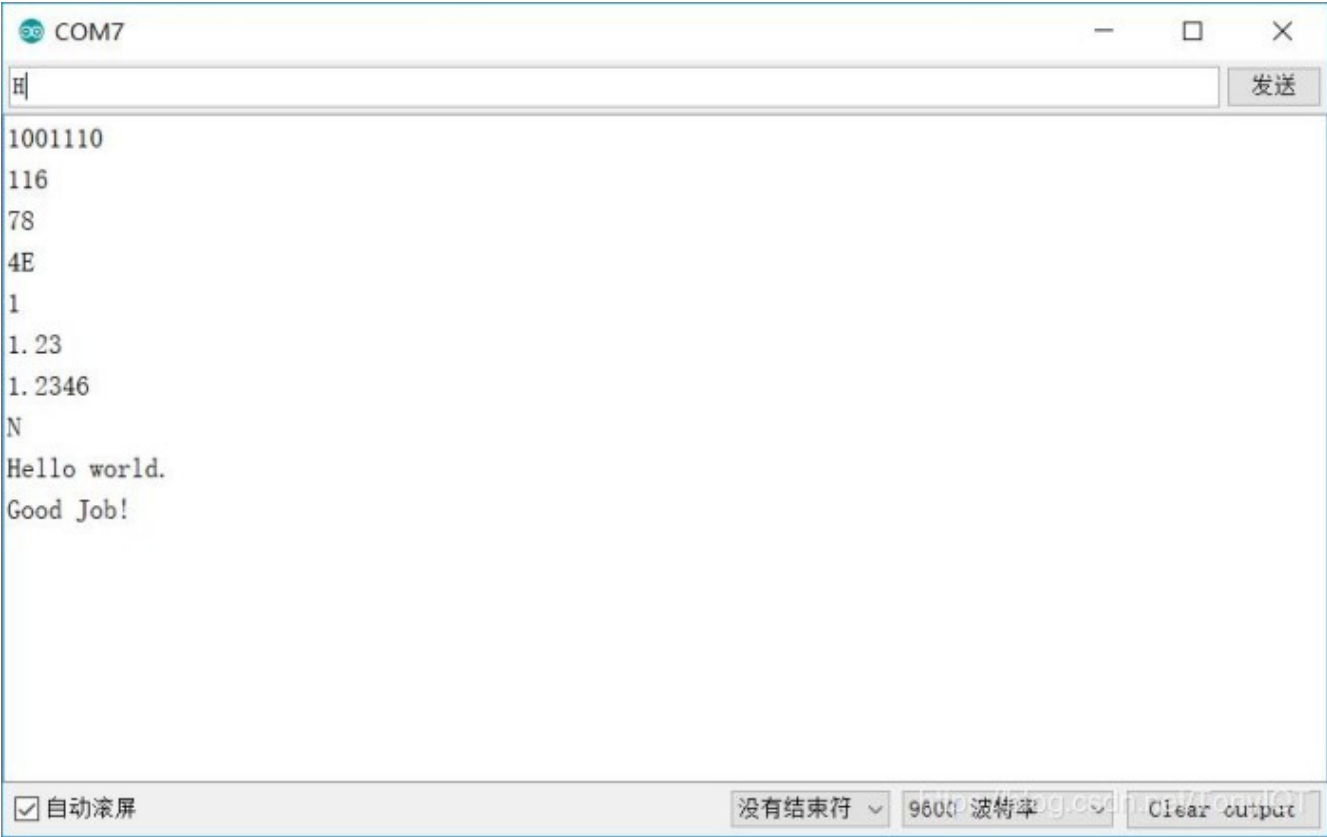
Serial
13 Serial.println(78, HEX); // 4E
14 Serial.println(1.23456, 0); // "1"
15 Serial.println(1.23456, 2); // "1.23"
16 Serial.println(1.23456, 4); // "1.2346"
17 Serial.println('N'); // "N"
18 Serial.println("Hello world."); // "Hello world."
19 }
20
21 void loop() {
22
23   if (Serial.available() > 0) // 串口接收到数据
24   {
25     incomedate = Serial.read(); // 获取串口接收到的数据
26     if (incomedate == 'H')
27     {
28       Serial.println("Good Job!");
29     }
30   }
31
32   delay(1000);
33
34 }
```

上传成功。

项目使用了 3336 字节，占用了 (10%) 程序存储空间。最大为 32256 字节。
全局变量使用了 212 字节，(10%) 的动态内存，余留 1836 字节局部变量。最大为 2048 字节。

27 <https://blog.csdn.net/TonyIoT> Arduino/Genuino Uno 在 COM7

4. 打开串口监视器，设置波特率9600，观察串口打印信息。输入"H"，串口返回"Good Job!"。



更多内容，欢迎关注我的公众号。微信扫一扫下方二维码即可关注：



TonyIOT   博客专家
原创文章 77 获赞 283 访问量 29万+

已关注

私信