**Arduino物联网完全开源教程**

徐海辉 韦莎莎 武鹏飞 编著

X X 出 版 社

内 容 简 介

本书采用大量实例（智能家居安防报警系统、植物生长状态监测、GPS传感器、家居环境监控、空气质量采集等）讲解的方式带领读者亲历从开始Arduino创意到最终完成项目的全部过程。采用IoTgo平台，内容由易到难，循序渐进， 精心编排，每个实验均配有电路图和具有详细注释的源程序，方便读者上手练习。

前言

自2005年Arduino诞生到今天，它已经成为了当今世界上最流行的开源硬件。由于Arduino硬件体积小，操作简单，现在不光只有电子爱好者使用，就连艺术家和设计师都开始使用它作为创新互动项目的控制器。关于Arduino的入门学习，相信大家都可以在网络上找到许多免费的资源，在此无需赘述。而笔者更希望和大家分享探讨Arduino与物联网之间的交互应用。

随着电子产品的爆炸性增长，这将直接驱动物联网的发展与完善。什么是物联网呢？其英文名称是“The Internet of Things”。简而言之，就是各种不同的物理设备通过互联网实现了相互通讯，信息共享。虽然概念简单，但意义非凡。试想一下，当你周围的物体能够任由你自己控制而无需受空间或时间的限制，比如家里的空调，你可以在上班的时候控制它开关及温度，这将会给人类的生活带来质的改变。物联网将是实现设备智能化的关键一步。

在过去，设备智能化的概念可能还只是一个构想，基本原因在于硬件的成本高，开发难度大，即使电子设计爱好者有想法，也因为这样的原因而打退堂鼓。然而近年开源硬件的兴起，给许许多多的电子设计爱好者带来希望。仅需低成本的投入，他们就可以使用开源硬件搭建一套简易的数据采集系统或者安防系统。

例如，一位设计师就使用了Arduino Lilypad作为主控器缝制在芭蕾舞鞋的鞋带上，同时也在芭蕾舞鞋鞋底和鞋尖处都安装了压力传感器。当舞者在跳舞的时候，压力传感器将数据发送至Arduino Lilypad，通过数据的编译与转换发送至云平台，最后你将会在手机的app应用上看到舞者的舞步轨迹，从而更好地帮助舞者了解自己的动作和舞步是否到位。这是一个很好的开源硬件物联网应用案例。注意，你会发现这双舞鞋竟然也成了可穿戴设备！现在是否有一种捶胸顿足的冲动？原来可穿戴设备也没什么了不起的。

看到这里，你是否对Arduino与物联网的交互应用产生了兴趣，是否也想自己动手开始制作一个项目出来？这就是本书的宗旨。ITEADStudio团队会在本书深入浅出地分享各种Arduino在物联网中的应用案例，手把手指导零基础的读者，从硬件到云平台，再到软件代码编写，完整地搭建一个属于自己的物联网智能家居系统的原形。在这个过程中，你除了能掌握基本的Arduino相关技术，还能将本书的例子不断深化探索，打造一个属于自己的物联网智能项目。

本书的第一篇将会介绍物联网的基础--硬件设备。硬件主要使用的是Iteaduino系列的Arduino兼容开发板，如Iteaduino UNO, Iteaduino MEGA 2560, WBoard Pro等。使用这些开发板作为案例基础，因为这些开发板兼具了更多的功能和更方便的接口，无需叠加扩展板即可使用。非常适合爱好者快速搭建项目。

本书第二篇将会介绍物联网中的数据传输云平台。本书使用的是IoTgo云平台，一个由ITEADStudio团队自主研发搭建的完全开源物联网平台。在这篇，将会着重介绍云平台的使用方法，应用程序接口以及开放所有源码。你将轻松搭建一个属于自己的云平台，并发掘其更多的可能性。

第三篇是本书的应用实战篇。ITEADStudio团队将会详细地介绍基于IoTgo云平台的两个智能家居的应用基础案例。通过综合案例加深读者对开源硬件Web领域的深刻理解。

准备好了吗？那就开始创造吧！

本书配套了书中所有实例的源代码，读者可以登陆http://arduino.cc的图书专栏的“下载专区”免费下载。

有兴趣的读者可以登陆http://arduino.cc的图书专栏与作者在线交流。

编 者

2014.12

目录

[**Arduino基础篇** - 8 -](#_Toc405470463)

[**第一篇 Arduino相关介绍** - 8 -](#_Toc405470464)

[1.1 入门概述：软硬件介绍 - 9 -](#_Toc405470465)

[1.1.1 Arduino开源硬件介绍 - 9 -](#_Toc405470466)

[1.1.2 Arduino IDE介绍 - 9 -](#_Toc405470467)

[1.2 感知周围环境 - 9 -](#_Toc405470468)

[1.2.1 温湿度测量 - 9 -](#_Toc405470469)

[1.2.2 超声波测距 - 10 -](#_Toc405470470)

[1.3 让一切动起来 - 10 -](#_Toc405470471)

[1.3.1 继电器案例 - 10 -](#_Toc405470472)

[1.3.2 舵机案例 - 10 -](#_Toc405470473)

[1.3.3 电机案例 - 10 -](#_Toc405470474)

[1.4 我的地盘听我的 - 11 -](#_Toc405470475)

[1.4.1 蓝牙无线通信 - 11 -](#_Toc405470476)

[1.4.2 nRF24L01无线通信 - 11 -](#_Toc405470477)

[1.5 我要更大的世界 - 11 -](#_Toc405470478)

[1.5.1 W5500以太网远程通信 - 11 -](#_Toc405470479)

[1.5.2 WIFI远程通信 - 11 -](#_Toc405470480)

[**物联网介绍篇** - 12 -](#_Toc405470481)

[**第二篇 打造自己的物联网** - 12 -](#_Toc405470482)

[2.1 平台介绍：Yeelink， Xively - 13 -](#_Toc405470483)

[2.2 IoTgo:一个开源的物联网 - 13 -](#_Toc405470484)

[2.2.1 IoTgo平台的特点 - 13 -](#_Toc405470485)

[2.2.2 IoTgo平台的架构 - 13 -](#_Toc405470486)

[2.2.3 IoTgo平台的开源资料和更新 - 13 -](#_Toc405470487)

[2.3 IoTgo平台的使用 - 14 -](#_Toc405470488)

[2.3.1 如何注册IoTgo用户 - 14 -](#_Toc405470489)

[2.3.2 如何上传数据到IoTgo平台 - 14 -](#_Toc405470490)

[2.3.3 如何远程控制Arduino设备 - 14 -](#_Toc405470491)

[**Arduion+物联网篇** - 15 -](#_Toc405470492)

[**第三篇 Arduino智能家居实战** - 15 -](#_Toc405470493)

[3.1 家居环境远程监控 - 16 -](#_Toc405470494)

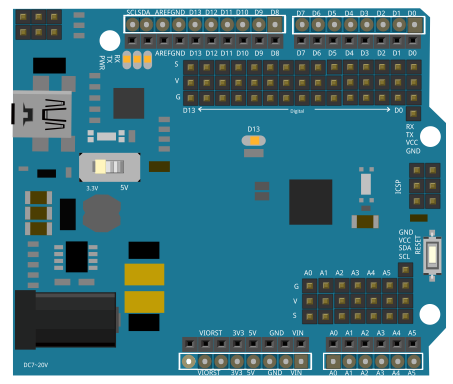
[3.1.1 IoTgo+温湿度传感器 - 16 -](#_Toc405470495)

[3.2 家电远程控制 - 16 -](#_Toc405470496)

[3.2.1 IoTgo + 继电器 - 16 -](#_Toc405470497)

Arduino基础篇

# 第一篇 Arduino相关介绍

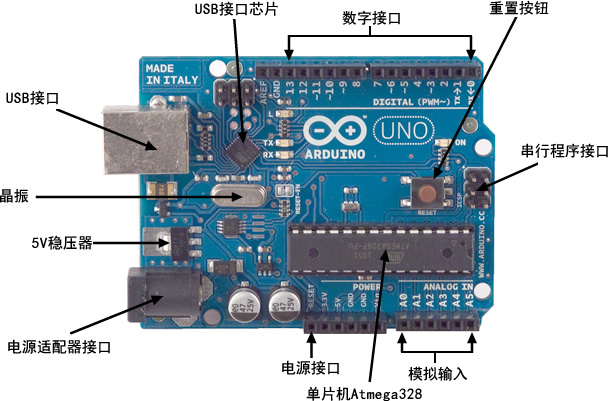
****

## 1.1 入门概述：软硬件介绍

什么是Arduino？它是一个开源的电子平台，这个平台是由灵活易用的硬件和软件组成的。它是面向艺术家，设计师，爱好者和任何对创造交互对象和交互环境有兴趣的人士的。本节将会简要介绍Arduino的硬件与软件。让初学者能够快速掌握arduino的硬件基本情况和如何使用Arduino IDE。掌握这些知识，初学者能够游刃有余地学习后面的应用案例。

### 1.1.1 Arduino开源硬件介绍

下面这款板子是官方版本的Arduino UNO主板，我们简要介绍一下这块板子上的元器件及其作用。

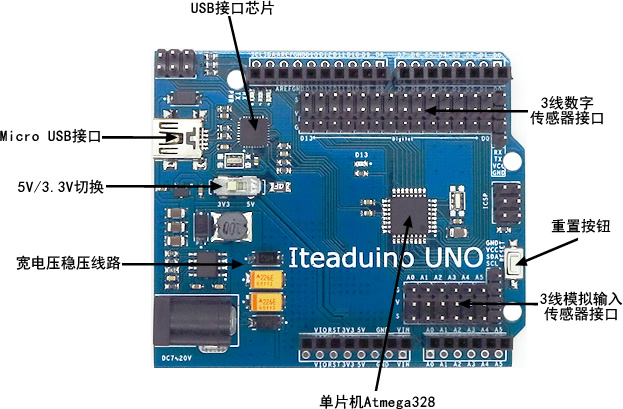


**图1.1-1 Arduino UNO功能示意图**

|  |  |
| --- | --- |
| 元器件 | 作用 |
| USB插口 | 插口作用有3：供电，烧代码，arduino与电脑之间双向传递数据 |
| USB接口芯片 | 它把由电源插孔供给的7~12V电压调至稳定的5V |
| 电源接口 | 第一位置是reset，具有重启的作用；之后的3.3V、5V、GND、9V用于提供不同的电压。 |
| 模拟输入 | A0~A5针脚接口，由于内部电阻大，仅容许少量电流流入，通常作为测量链接在针脚上的电压值，这些接口也可被用作数字信号的输入和输出 |
| 数字接口 | 0~13引脚，是数字输入/输出（I/O）引脚，用来和其他器件发送和接收数据，标有波浪号的引脚可以产生变化的电信号，这种电信号可以用来做灯光效果或电机控制 |
| 单片机ATmega328 | 包含了执行指令的ATmega328处理器，分别保持数据和程序的不同类型的内存，具有多种发送和接收数据的方式 |
| LED | 标有L的LED链接在数字输入输出引脚13上；当Arduino与通过串口或USB链接的设备之间发送货接收数据时，标有RX, TXD LED会亮。 |
| 晶体 | 它每秒振荡16×106次，每振荡一次，单片机可以完成一次运算——加法、减法或其他数学运算。 |
| 重置按钮 | 当一切arduino当机的时候可以按重置按钮重启系统 |
| 串行程序接口 | 提供了一种不使用USB为Arduino编程的方式。一般较少用到 |

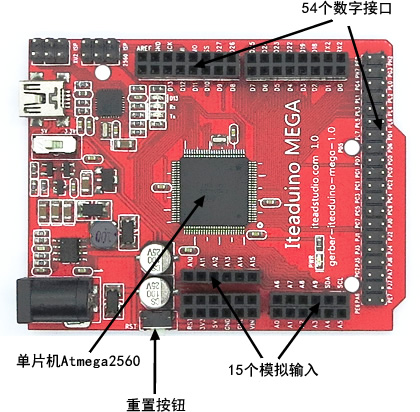
Arduino作为项目控制器最大的优势就是操作简易。你只要用上各种可能的输入输出器件：传感器、指示灯、显示器、电机等，就可以编程实现精确的交互要求，创造出具有特定功能的装置来。比如艺术家可以做出一个颜色根据室外天气变化而变化的室内壁灯；DIY爱好者可以做一些实时读取显示温湿度的闹钟或者智能小车；还有工程师还能用它做家居安防系统。Arduino为你实践自己的想法创造了无限可能。

但是Arduino系列的主板和兼容板有很多种，如何给自己的项目选择合适的主板，那就得看自己的项目需求和Arduino主板的功能能否对应上了。事实上，ITEADUINO UNOEAD团队以arduino为蓝本，开发了一系列的Arduino兼容开发板。之后的案例主要使用Iteaduino UNO 和Iteaduino Mega2560作为主控板的实例，那与Arduino相比，这些板子有什么优势呢？



**图1.1-2 Iteaduino UNO功能示意图**

从实物图可以看出，Iteaduino UNO引出更多的接口，方便连接更多的传感器，即插即用。



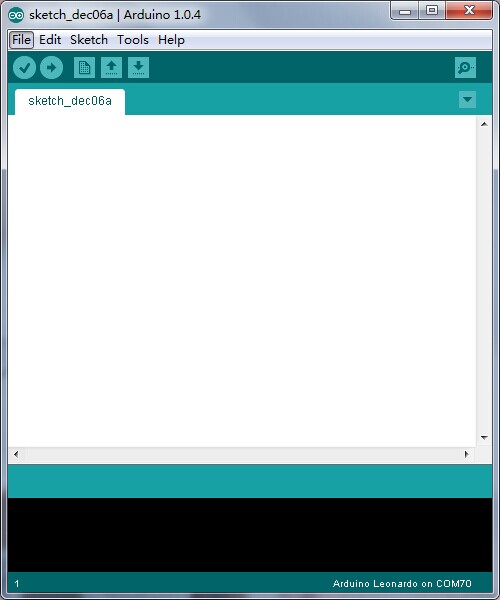
**图1.1-3 Iteaduino MEGA2560功能示意图**

Iteaduino Mega2560兼容更多的UNO扩展板，体积更小，资源更丰富。可以给UNO项目直接升级固件，而不用担心体积问题。

### 1.1.2 Arduino IDE介绍

Arduino 提供了一款非常方便的软件Arduino IDE，这是一个用来编写程序，发送程序到Arduino主板的软件。你可以在Arduino的官网上下载：http://arduino.cc/en/Main/Software。

成功在电脑上安装之后，打开Arduino IDE，你会看到它的界面如下图所示。(请以自己的Arduion IDE版本为准)



**图1.1-4 Arduino IDE界面**

IDE界面分成三部分，顶部是工具栏，代码窗口在中间，消息窗口在底部。在工具栏处，你会看到5个菜单项，6个图标项，下面的表格将会帮助你理解他们的用途。

|  |  |
| --- | --- |
| 菜单项 | 作 用 |
| File | 包含保存、打开和打印程序的功能，你可以打开库，完整的例子程序集以及Preference()子菜单。 |
| EdIteaduino UNO | 包含了文字处理软件中常见的拷贝、粘贴和搜索功能 |
| Sketch | 在里面你可以找到检查程序的功能，还有一些程序目录以及引入库的选项 |
| Tools | 有各种功能，以及选择Arduino板子类型和USB端口的选项 |
| Help | 帮助你如何开始，如何使用，如何解决简单的问题。 |
| 图标项 | 作 用 |
| Verify | 检查程序是否有效，或者有无错误 |
| Upload | 点击这个按钮，将程序直接上传到Arduino板子 |
| New | 创建一个新的空程序 |
| Open | 打开一个已有的程序 |
| Save | 保存当前程序 |
| Serial MonIteaduino UNOor | 显示从Arduino发来的串口数据 |

在使用Arduino IDE来编写自己的程序前，你可能需要先了解一下什么是Arduino Libraries，中文一般简称为库。根据官网的介绍：Arduino的库就像一个程序平台，你可以通过使用它来扩展arduino的环境。它提供了一个便捷的代码分享方式，你还可以在这些库里找到实用的函数。

Arduino的库有两种类型：标准库和用户自己写的库

**标准库（Standard Libraries）**

标准库内包含了基础的通讯函数，和一些硬件比如舵机和字符屏的支持函数。一般，当你下载安装Arduino IDE时，它内部就搭载了一些库，而这些库也完全支持它给出的例子（examples）。这些库文件可以在Arduino/Libraries文件夹里找到，需要注意的是，当你安装不同的Arduino IDE版本，每个版本都有其库文件，不建议去改变这些文件，也不要将自己写的库放在那个标准库文件夹里。

**自安装库（User Installed Libraries）**

因为是一个开源的平台，在网上还有各种各样有用的库文件，比如Arduino官方的playgroun，GIteaduino UNOHub。你可以将有用的库文件安装在Arduino IDE上，简便自己的程序编写，丰富自己的项目内容。ITEADUINO UNOEAD除了提供硬件设备，也为广大初学者编写了这些硬件的支持库，发布在GIteaduino UNOHub上，可以点击这里下载。安装这些库是很简单的，你只需将文件夹解压到Arduino/Libraries文件夹下即可，注意，库文件的命名不能带有中杠线（-），不然Arduino IDE是无法识别的。

**总结**

本章1.1入门概述主要介绍了arduino的硬件和软件两个部分。对于如何编程，未加赘述。实际上，Arduino的程序编写就是Java和C语言，读者只要学会基本的程序编写，就可以做出很棒的交互项目。你可以在国内的极客工坊、Arduino中文社区等论坛上找到Arduino程序编写的入门教程。

## 1.2 感知周围环境

在大致了解了Arduino的软硬件基础知识后，你一定迫不及待地想要动手做一个自己的项目了吧？ 不要着急，先让我们从最简单传感器来开始吧。

人们从外界获取信息必须借助感觉器官，但是每个人的感觉都带有主观性，无法得到一个标准。那么要想量化我们感受到的信息， 就需要借助传感器，通过传感器让电路从外界获取信息,并直观地呈现给我们。

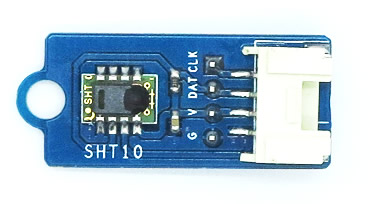
### 1.2.1 温湿度测量

人们总是非常关心自己所处的环境是否舒适安全，因此对环境信息的精确度也提出了更高的要求。本节使用的温湿度传感器是SHT10，该传感器将传感元件和信号处理电路集成在一块微型电路板上，输出完全标定的数字信号。传感器采用专利的CMOSens® 技术，确保产品具有极高的可靠性与卓越的长期稳定性。

这一节我们将使用Iteaduino UNO 来读取SHT10的温湿度值，并直接实时显示在串口输出的1602 LCD显示屏上。

**硬件需求**

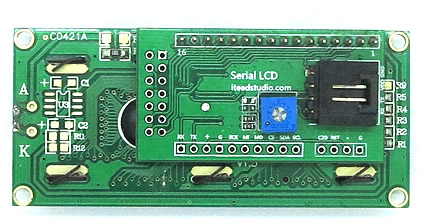
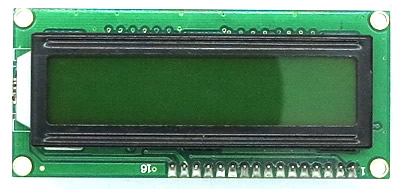
1. Iteaduino UNO主板 1块 （图1.1-2）
2. SHT10温湿度传感器 1块 （图1.2-1）
3. 串口1602 LCD显示屏 1块 （图1.2-2）
4. 4PIN平行杜邦线 1根 （图1.2-3）
5. 4PIN交叉杜邦线 1根 （图1.2-4）



（a）正面图 （b）背面图

**图1.2-1 SHT10温湿度传感器示意图**

如图 1.2-1所示，这款SHT10温湿度传感器电子积木拥有4PIN Grove和4PIN通用杜邦两种接口，可以根据自己的实际情况选择其中一接口即可。



（a）正面图 （b）背面图

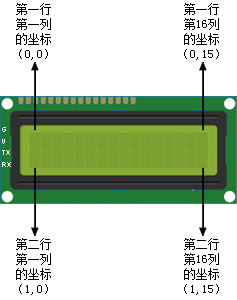
**图1.2-2 串口1602 LCD显示屏示意图**

图 1.2-2是串口1602显示屏，可以用来显示字母、数字以及符号，可同时显示16×2共两行32个字符，波特率是9600。从背面可以看到一个4PIN的UART串行接口，与UNO通信的时候TX和RX是交叉连接的。还有一个蓝色的电位器，是用来调节字符显示的对比度的。调到最小值的时候是看不见字符的。

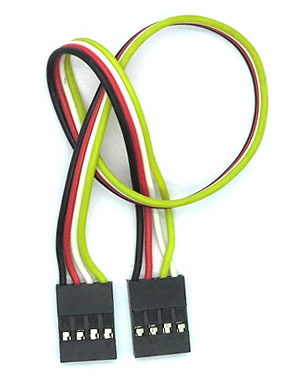
下表是对串口1602 LCD显示屏进行操作的各项指令格式。

|  |  |
| --- | --- |
| 指 令 | 说 明 |
| sc; | 清屏 |
| sb1; | 设置LCD背光灯亮 |
| sb0; | 设置LCD背光灯不亮 |
| sdx,y; | 设置光标的位置，x表示行数(x = [0,1]); y表示列数(y = [0:15])。例如sd0,0;将光标设定在第一行第一列 |
| ss+字符串; | 发送字符串 |
| su0; | 不显示光标 |
| su1; | 显示光标 |
| sf0; | 设置光标不闪烁 |
| sf1; | 设置光标闪烁 |
| sm0; | 将光标向左移动 |
| sm1; | 将光标向右移动 |

图1.2-3可以帮助你更好地理解后面需要编译的代码。



**图1.2-3 1602 LCD显示屏坐标示意图**

**图1.2-4 4PIN平行杜邦线 图1.2-5 4PIN交叉杜邦线**

图 1.2-3和图1.2-4的唯一区别是交叉杜邦线有一头的两根线是位置交叉了。

**连线图**

现在我们要用杜邦线将它们连接到主板上对应的通信接口。

如图1.2-5所示，串口1602 LCD显示屏连接到Iteaduino UNO主板上的UART串口上，SHT10连接到板子上的模拟输入接口

Iteaduino UNO主板 串口1602 LCD显示屏

RX(D0) → TX

TX(D1) → RX

5V → VCC

GND → GND

Iteaduino UNO主板 SHT10温湿度传感器

SCL(A5) → CLK

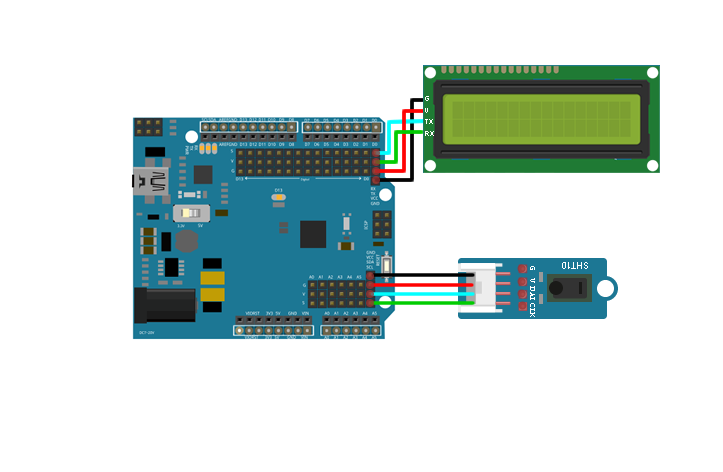
SDA(A4) → DAT

5V → VCC

GND → GND

Iteaduino UNO开发板引出了专用的4PIN UART接口和4PIN

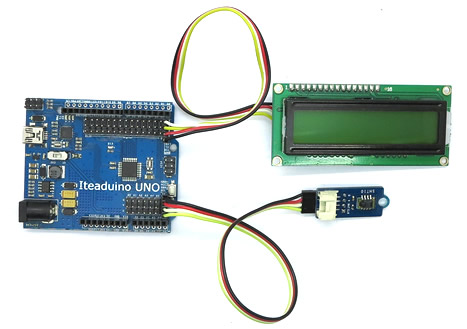
电子积木接口，非常方便。如果使用的是Arduino 官方版UNO就需要使用1PIN的杜邦线一根根连接到对应的引脚或者增加一个I/O扩展板。



**图1.2-6 温湿度测量连接示意图**

**硬件连线实物图**

最终硬件连接如图1.2-6所示，串口1602 LCD显示屏可以通过4PIN交叉杜邦线直接连接到Iteaduino UNO主板上的串口上。SHT10通过4PING平行杜邦线直接连接到板子上的4PIN电子积木接口。



**图1.2-7 温湿度测量连接实物图**

**下载库文件**

要驱动SHT10温湿度传感器需要使用一个封装好的库文件，它可以帮助我们简化大部分的编程工作。本书配套的所有库文件和源代码都可以在github上打包下载：http://arduino.cc

找到下载的SHT1x.zip文件，直接解压得到一个文件夹，将文件名的中划线“-”改为下划线“\_”，放到你的Arduino IDE的libraries文件夹里面，最终放置路径如图1.2-7所示。

1.2-6 库文件放置路径.jpg

**图1.2-8 库文件放置路径**

切记：每次添加新的库文件后都需要重启Arduion IDE才能生效。

**上传代码**

添加了SHT1x的库文件后，我们只需要做最简单的上层通信，将以下代码复制到Arduino IDE中。

**代码1.2.1 温湿度测量代码示意**

/\*Iteaduino UNO连接SHT10实时测量显示温湿度\*/

//调用SHT1x库文件

#include <SHT1x.h>

float temp\_c;

float temp\_f;

float humidity;

//将DAT和CLK分别连接到A4和A5

SHT1x sht1x(A4, A5);

void setup()

{

Serial.begin(9600);

}

void loop()

{

temp\_c = sht1x.readTemperatureC();

temp\_f = sht1x.readTemperatureF();

humidity = sht1x.readHumidity();

//sc是清屏，括号里面的分号表示该条命令结束

Serial.println("sc;");

//在第一行第一列的位置，命令结束

Serial.println("sd0,0;");

//LCD显示字符串SHT10 Hum:，命令结束

Serial.println("ssSHT10 Hum:;");

//在第一行第十一列的位置，命令结束

Serial.println("sd0,10;");

//命令LCD显示字符串，注意，ss括号内没有分号，表示命令还没有结束，所以使用的是print，而不是println

Serial.print("ss");

//接下来要显示的字符串，括号内也没有分号，表示命令还没有结束，仍然是用print

Serial.print(int(humidity));

//显示字符串（%），命令结束

Serial.println("(%);");

Serial.println("sd1,0;");

Serial.println("ssT:;");

Serial.println("sd1,2;");

Serial.print("ss");

Serial.print(int(temp\_c));

Serial.println("(C);");

Serial.println("sd1,10;");

Serial.print("ss");

Serial.print(int(temp\_f));

Serial.println("(F);");

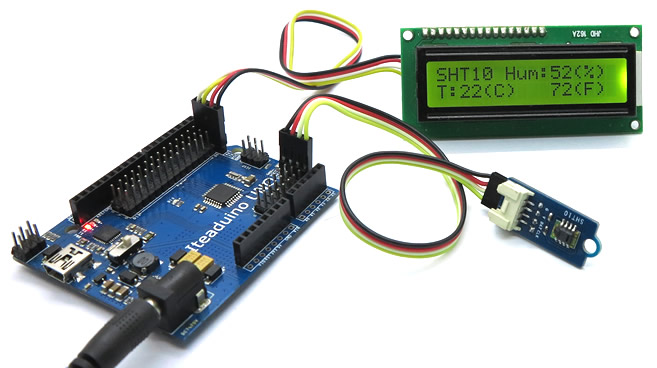
//每隔一秒更新一次温湿度数据

delay(1000);

}

特别注意，在上传代码到Iteaduino UNO的过程中，需要用到TX和RX这两个串口引脚，所以要先断开串口1602 LCD跟Iteaduino UNO的连接，以免占用串口，导致代码不能成功上传。等程序上传完成再连接硬件。

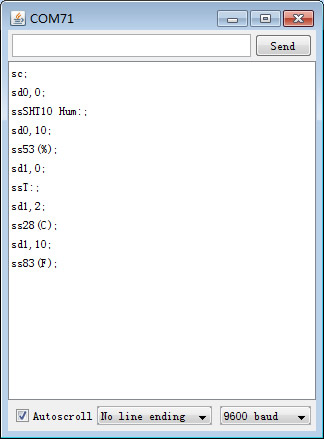
上传代码之前需要选择对应的板子类型和串口号，Iteaduino UNO选择的板子类型是“Arduino UNO”。最终显示效果如图1.2-8所示。



**图1.2-8 温湿度显示最终效果实物图**

在这里不得不说一句，很多刚接触Arduino的小白会以为代码是上传给了功能模块。其实每次上传的代码都是烧写到开发板里面了，由开发板来控制传感器和模块。

用USB线连接到电脑，也可以用Arduino IDE的串口监视器查看输出的信息。如图1.2-9所示。



**图1.2-17 温湿度串口监视器示意图**

### 1.2.2 超声波测距

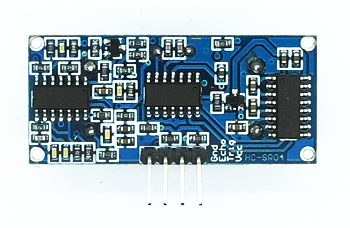
上一节我们学习了如何将温湿度实时显示到串口1602 LCD显示屏的方法，这节我们再来做一个超声波测距显示实例，可以帮助你更好地理解超声波智能避障小车的工作原理，从而开发出更多有趣实用的作品，比如超声波门锁、超声波播放器、倒车雷达等等。

本节使用的超声波传感器是HC-SR04，是DIY爱好者使用非常普遍的一款超声波传感器，可提供2cm-400cm的距离感测，精度3mm，最大的优点就是廉价，是初学爱好者首选。

**硬件需求**

1. Iteaduino UNO主板 1块 （图1.1-2）
2. HC-SR04超声波传感器 1块 （图1.2-9）
3. 串口1602 LCD显示屏 1块 （图1.2-2）
4. 4PIN交叉杜邦线 1根 （图1.2-5）

（5） 1PIN母口杜邦线 4根 （图1.2-10）



（a）正面图 （b）背面图

**图1.2-9 HC-SR04超声波传感器示意图**

HC-SR04超声波传感器是分体式的，板载一个发射端和一个接收端。VCC接5V电源，GND为地线，Trig触发控制信号输入，Echo回响信号输出。

注：此模块不宜带电连接，若要带电连接，则先让模块的GND端先连接，否则会影响模块的正常工作。

1PIN杜邦线是最普遍的通用接线，分为母对母（图1.2-10），公对母（图1.2-11），公对公（图1.2-12）三种类型。



**图1.2-10 1PIN母对母杜邦线示意图**

****

**图1.2-11 1PIN公对母杜邦线示意图**

****

**图1.2-12 1PIN公对公杜邦线示意图**

****

**图1.2-13 1PIN面包线示意图**

面包线也可以替代公对公杜邦线。如图1.2-13所示。

**连线图**

如图1.2-14所示，串口1602 LCD显示屏连接到Iteaduino UNO主板上的UART串口上，HC-SR04连接到板子上的数字信号接口。

Iteaduino UNO主板 串口1602 LCD显示屏

RX(D0) → TX

TX(D1) → RX

5V → VCC

GND → GND

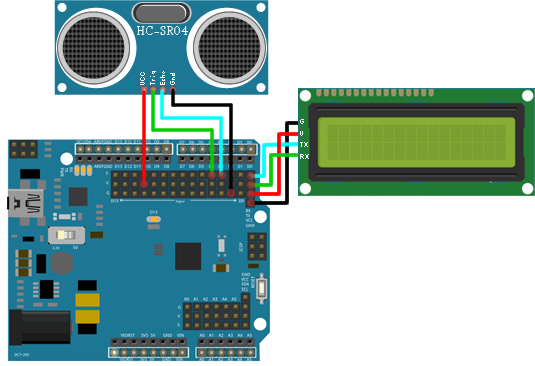
Iteaduino UNO主板 HC-SR04超声波传感器

GND → GND

D2 → Echo

D3 → Trig

5V → VCC



**图1.2-14 超声波测距接线示意图**

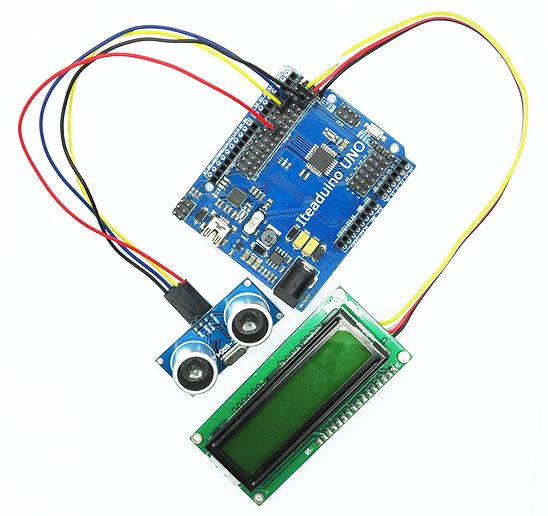
**硬件连线实物图**

Iteaduino UNO引出了3PIN的电子积木接口，如图1.2-15所示，S表示所在的一行都是信号脚，V所标示的一行都是电源接口，G所标示的一行都是地线。所以你可以根据实际情况自由选择需要连接的V和G，而不用完全按照我们的接线示意图连接。



**图1.2-14 3PIN电子积木引脚功能示意图**

硬件连接如图1.2-15所示。



**图1.2-15 超声波测距连线实物图**

**上传代码**

以下是本次实例需要用到的代码，跟上一节不同，HC-SR04不需要用到库文件，可以直接操作使用。

**代码1.2.2 超声波测距代码示意**

/\*Iteaduino UNO连接HC-SR04实时显示测距\*/

//数字信号脚直接用数字表示即可，模拟输入脚需要带A

const int TrigPin = 3;

const int EchoPin = 2;

float cm;

void setup()

{

Serial.begin(9600);

pinMode(TrigPin, OUTPUT);

pinMode(EchoPin, INPUT);

}

void loop()

{

//LOW-HIGH-LOW改变Trig脚的电压，从而发射出超声波

digitalWrite(TrigPin, LOW);

delayMicroseconds(2);

digitalWrite(TrigPin, HIGH);

delayMicroseconds(10);

digitalWrite(TrigPin, LOW);

//超声波和距离换算公式

cm = pulseIn(EchoPin, HIGH) / 58.0;

cm = (int(cm \* 100.0)) / 100.0;

//串口屏显示的信息

Serial.println("sc;");

Serial.println("sd0,0;");

Serial.println("ssHC-SR04:;");

Serial.println("sd0,8;");

Serial.print("ss");

Serial.print(cm);

Serial.println(";");

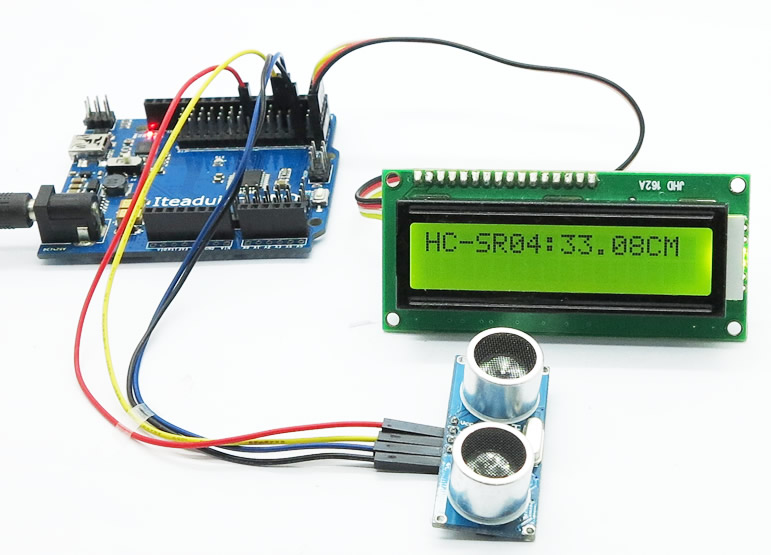
Serial.println("ssCM;");

//每隔一秒更新一次

delay(1000);

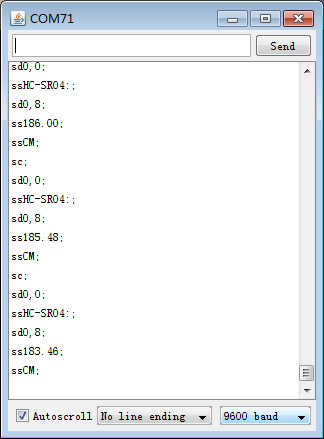
}

上传代码前不能将串口1602 LCD显示屏连接到Iteaduino UNO上，上传完成后再连接。最终显示效果如图1.2-16所示。



**图1.2-16 超声波测距显示效果图**

也可以打开Arduino IDE的串口监视器查看到输出的信息，如图1.2-17所示。



**图1.2-17 超声波测距串口监视器示意图**

**总结**

经过这一节对传感器的学习，我们学会了如何通过传感器获取外界信息。当然现实中还有很多种传感器需要我们学习使用，比如光敏传感器、气敏传感器、土壤湿度传感器、颗粒传感器等等。使用方法大同小异，这里就不一一详述了。当然我们并不满足“五官”的感受，我们还需要主动地改变周围环境，给Arduino装上手和脚，这就是我们下一节的内容。

## 1.3 让一切动起来

在这节，我们要开始让Arduino动起来了。通过学习继电器、舵机和直流电机的工作原理，我们就可以开发出更多的交互式作品，比如遥控家电、智能门锁、机器人、小车等等，用于更好地服务我们的生活。

### 1.3.1 继电器案例

继电器是一种电控制器，具有能反映外界某种激励量（电或非电）的感应机构，通常应用于自动化的控制电路中。简单地理解就是用小电流去控制大电流运作的一种“自动开关”，在电路中起着自动调节、安全保护、转换电路等作用。应用最广泛的就是智能插座、智能开关。

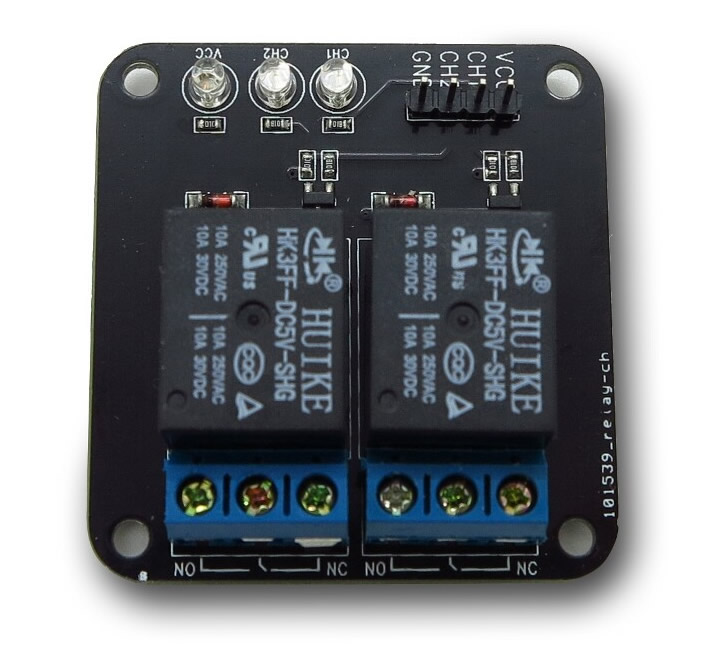
接下来，我们就来做一个用继电器控制灯管亮灭的实例。

**硬件需求**

1. Iteaduino UNO主板 1块 （图1.1-2）
2. 双路继电器模块 1块 （图1.3-1）

（3） 台灯1台

（4） 1PIN母口杜邦线 4根 （图1.2-10）



**图1.3-1 HC-SR04超声波传感器示意图**

这款双路继电器模块，包含有两个继电器，你可以选择使用其中的一个，也可以使用两个。工作电压为5V，可以从Iteaduino UNO开发板上直接取电。板子上还有3个LED指示灯用来显示继电器的状态，分别为CH1、CH2和VCC。每一路继电器都有3个触点，标有NC的是常闭触点，表示继电器线圈未通电时处于接通状态的静触点，标有NO的是常开触点，指的是继电器线圈未通电时处于断开状态的静触点。NC和NO之间的一端是公用端。

**连接图**

如图1.2-10所示，将台灯的插头线的其中一根线剪断，然后将剪断后的线头一端接在继电器的NO触点，另外一端接在同一继电器的公用端。

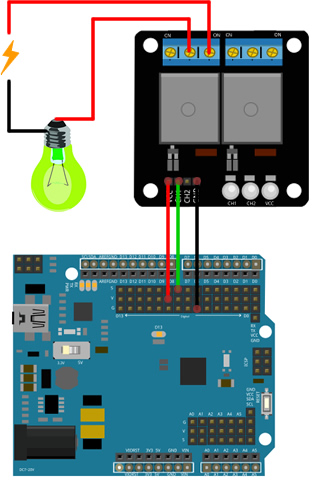
双路继电器模块与Iteaduino UNO的连线如下：

Iteaduino UNO主板 双路继电器模块

5V → VCC

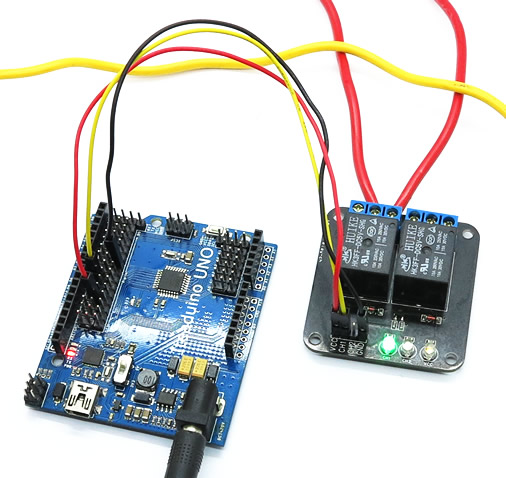
CH1 → D7

GND → GND



**图1.3-2 双路继电器连接示意图**

图1.2-11是硬件连接实物运行效果图。继电器的CH1脚被置高的时候，台灯电路被接通，CH1的指示灯亮绿灯。VCC电源指示灯是亮黄灯。



**图1.3-2 双路继电器连接示意图**

如果把连接在继电器NO一端的线头连到NC则操作就会反过来，继电器被置高的时候灯的电路会被断开。

切记：在每一步手动操作的时候请一定拔掉台灯插头！

**上传代码**

以下是本次实例需要用到的代码。

**代码1.3.1 继电器开关代码示意**

/\*Iteaduino UNO连接双路继电器模块实现开关切换\*/

//定义继电器的CH1这路的控制脚为D7脚

int relay\_CH1 = 7;

void setup()

{

pinMode(relay\_CH1, OUTPUT);

}

void loop()

{

//当D7脚置高，即HIGH=5V左右时，CH1指示灯亮，常开触点被吸合，电路就会被接通

digitalWrite(relay\_CH1, HIGH);

//电路接通保持1秒的时间

delay(1000);

//当D7脚置低，即HIGH=0V时，CH1指示灯灭，弹簧回弹，电路被断开

digitalWrite(relay\_CH1, LOW);

//电路断开保持5秒的时间

delay(5000);

}

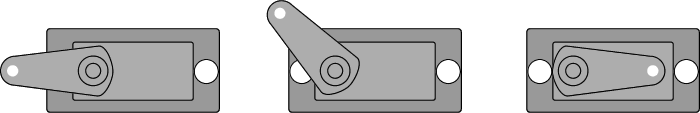
### 1.3.2 舵机案例

舵机是一种位置伺服驱动器，主要是由外壳、电路板、无核心马达、齿轮与位置检测器构成。厂商所提供的舵机规格资料，都会包含外形尺寸(mm)、扭力(kg/cm)、速度(秒/60°)、测试电压(V)及重量(g)等基本资料。扭力的单位是 kg/cm，意思是在摆臂长度 1 公分处，能吊起几公斤重的物体。这就是力臂的观念，因此摆臂长度愈长，则扭力愈小。速度的单位是 sec/60°，意思是舵机转动 60°所需要的时间。舵机一个最大的特点是旋转的角度范围是0°到180°。

舵机有多种规格，但所有的舵机都有三根线，颜色都是标准的棕色、红色、橙色。棕色为地线，红色是电源正极线，橙色是信号线。因为是统一的行业标准，所以可以很方便的区分。

舵机的转动角度是通过调节PWM信号（脉冲信号）的占空比来实现的，标准的PWM信号的周期固定为20描述（50Hz），理论上脉宽分布应为1到2ms，但事实上脉宽可为0.5到2.5ms，不同制造商的产品会有所不同。标准脉宽和舵机的转动角度如图1.3-3所示。

0° 45° 180°



高 高 高

低 1000μs 低 1250μs 低 2000μs

**图1.3-3 舵机角度和脉宽示意图**

接下来，我们就做一个舵机从0°到180°旋转的实例。

**硬件需求**

1. Iteaduino UNO主板 1块 （图1.1-2）

（2） 9克舵机 1个 （图1.3-1）

舵机一般都自带有3pin的母口杜邦线，如图1.3-4所示。我们可以直接连接到Iteaduino UNO开发板上使用。



**图1.3-4 舵机及其配件示意图**

**连接图**

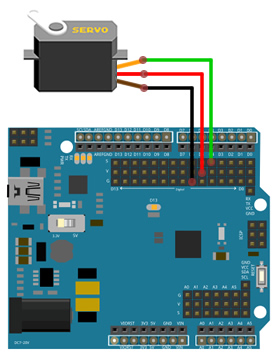
如图1.3-5所示，舵机的接口跟3PIN传感器的接口一样，可以很方便地直接连接到Iteaduino UNO的3PIN接口上。

Iteaduino UNO主板 9克舵机

GND → GND

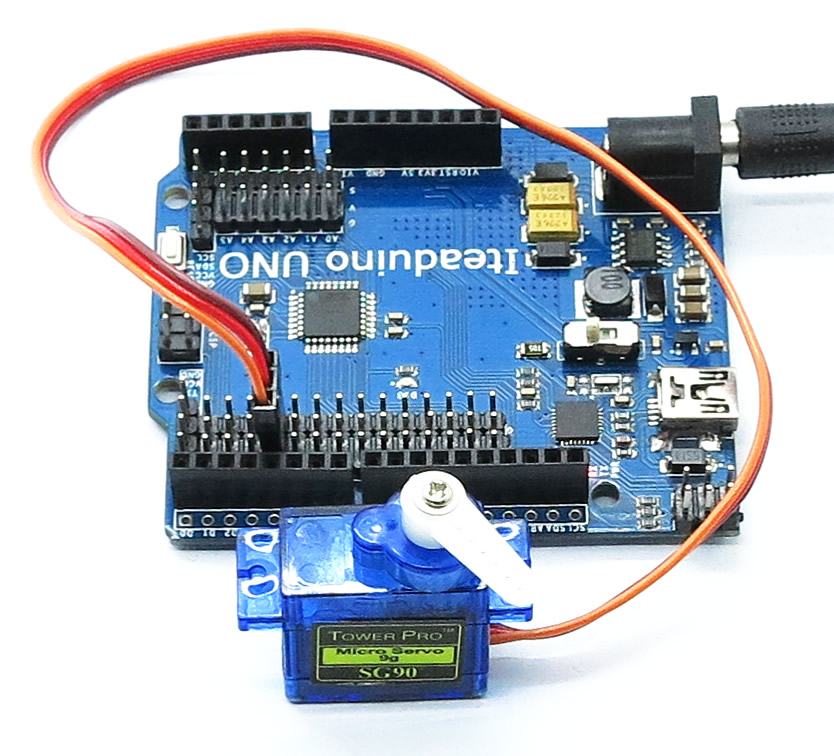
5V → VCC

D5 → 信号脚



**图1.3-5 舵机连线示意图**

硬件实物连接如图1.3-6所示。



**图1.3-5 舵机实物连接效果示意图**

我们通过Iteaduino UNO的普通数字传感器接口产生占空比不同的方波，模拟产生PWM信号进行舵机定位。另外Arduino IDE自带有舵机驱动的Servo函数库，可以直接调用，方便我们编写代码。

**上传代码**

以下是本次实例测试用的代码。

**代码1.3.2 舵机旋转代码示意**

/\*Iteaduino UNO连接连接9克舵机0°到180°旋转\*/

//调用Servo库文件

#include <Servo.h>

//定义一个名为myservo的舵机

Servo myservo;

//定义一个整数型的变量pos

int pos = 0;

//定义舵机的控制脚位D3

int servo\_pin = 3;

void setup()

{

//定义myservo等于前面定义的舵机控制脚

myservo.attach(servo\_pin);

}

void loop()

{

//从0到180递加

for(pos = 0; pos <= 180; pos ++)

{

//输出模拟PWM信号控制舵机

myservo.write(pos);

delay(15);

}

//从0到180递减

for(pos = 180; pos >= 0; pos --)

{

//输出模拟PWM信号控制舵机

myservo.write(pos);

delay(15);

}

}

可定设的数值范围是0到180，如果超过180，超过的脉宽信号将在舵机内部的无核心电机空转，虽然不会损坏舵机，但是不推荐使用。

### 1.3.3 电机案例

前面介绍的舵机只能旋转0°到180°，那么如果要连续转圈或者加速、减速，我们该使用哪种电机呢？

这一节我们将学习如何使用Arduino控制直流电机的加速、减速、正向旋转、反向旋转以及旋转时间。

连续旋转电机有

## 1.4 我的地盘听我的

### 1.4.1 蓝牙无线通信

### 1.4.2 nRF24L01无线通信

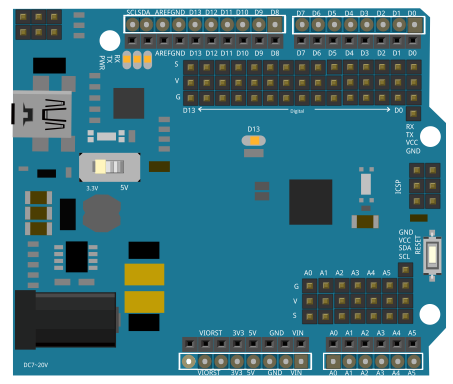
## 1.5 我要更大的世界

### 1.5.1 W5500以太网远程通信

### 1.5.2 WIFI远程通信

物联网介绍篇

# 第二篇 打造自己的物联网

****

## 2.1 平台介绍：Yeelink， Xively

## 2.2 IoTgo:一个开源的物联网

### 2.2.1 IoTgo平台的特点

### 2.2.2 IoTgo平台的架构

### 2.2.3 IoTgo平台的开源资料和更新

## 2.3 IoTgo平台的使用

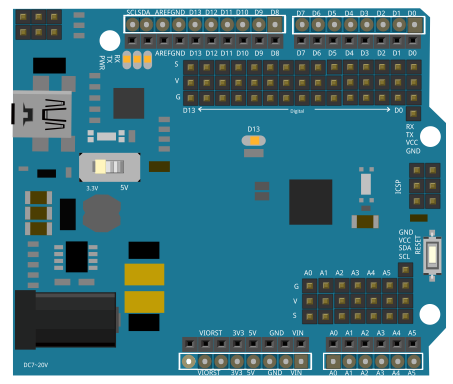
### 2.3.1 如何注册IoTgo用户

### 2.3.2 如何上传数据到IoTgo平台

### 2.3.3 如何远程控制Arduino设备

Arduion+物联网篇

# 第三篇 Arduino智能家居实战

****

## 3.1 家居环境远程监控

### 3.1.1 IoTgo+温湿度传感器

## 3.2 家电远程控制

### 3.2.1 IoTgo + 继电器

后序

本书配套了书中所有实例的源代码，读者可以登陆http://arduino.cc的图书专栏的“下载专区”免费下载。

由于涉及云平台的架构部署，加之编写仓促，错误在所难免，欢迎读者提出宝贵意见。

有兴趣的读者可以登陆http://arduino.cc的图书专栏与作者在线交流。