

uKit Explore

用户使用手册

V0.3

目录

目录	2
第一章：简介	3
uKit Explore 介绍	3
Explore 主控简介	4
uKitRoute 智能灰度传感器简介	6
第二章：快速入门	7
uKit Explore 硬件	7
规格参数	7
Arduino IDE for Explore 主控	9
STEP1: 下载 Arduino IDE 软件	9
STEP2: Explore 驱动程序安装	10
STEP3: 连接 Explore 主控至电脑	11
STEP4: 在Arduino IDE 中进行编程	11
STEP5: 上传代码至 Explore 主板	12
PIN 连接以及接线	15
1、使用3Pin线缆连接舵机	15
2、接线	16
第三章: Explore 基础使用	17
项目一 串口实验	17
项目二 炫彩RGB灯	19
项目三 蜂鸣器	21
项目四 4KG舵机 舵机模式控制	23
项目五 4KG舵机 轮模式控制	25
项目六 舵机回读控制	27
第四章: UKit Explore库	29
项目一 NOC库介绍	29
项目二 NOC库使用	33

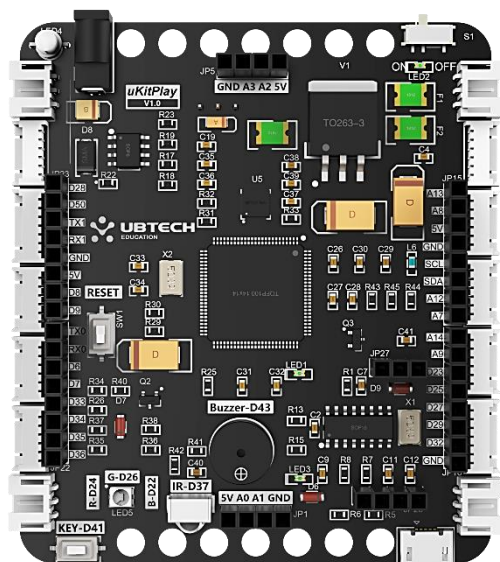
第一章：简介

uKit Explore 介绍

简单来说，uKitExplore是一款是基于Arduino开源设计的竞赛产品，它是精心为初学者设计的竞赛主控，它可以带您走进丰富多彩的电子世界，让您体验到电子科技无穷的乐趣。在整个过程中，能很好的与uKit器件结合，也有丰富的传感器模块接口。另外，还附带了几节实验教程，这些教程都是从初学者的角度考虑的，每一节实验都配有图文结合的实验说明文档和非常有趣的例子程序，有很大的空间让学习者发挥自己的创意。

uKit Explore 系列，有两个大类，包括：电子类、结构类。其中，电子类包括了Explore主控、舵机、传感器等。而结构类包括了外观结构件、连接结构件等。

注意：uKit Explore系列相关程序仍然在保持更新中



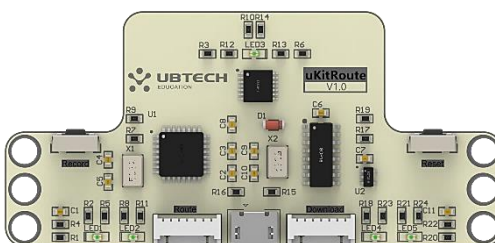
Explore主控



uKit按键传感器



超声波传感器

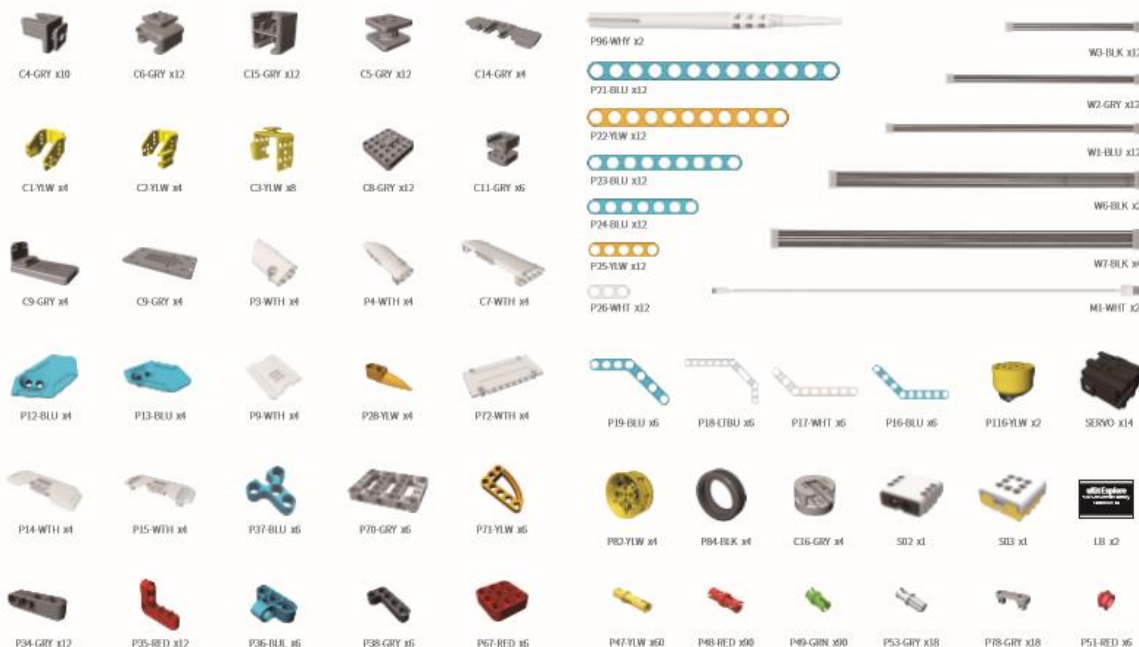


智能灰度传感器



4KG舵机.

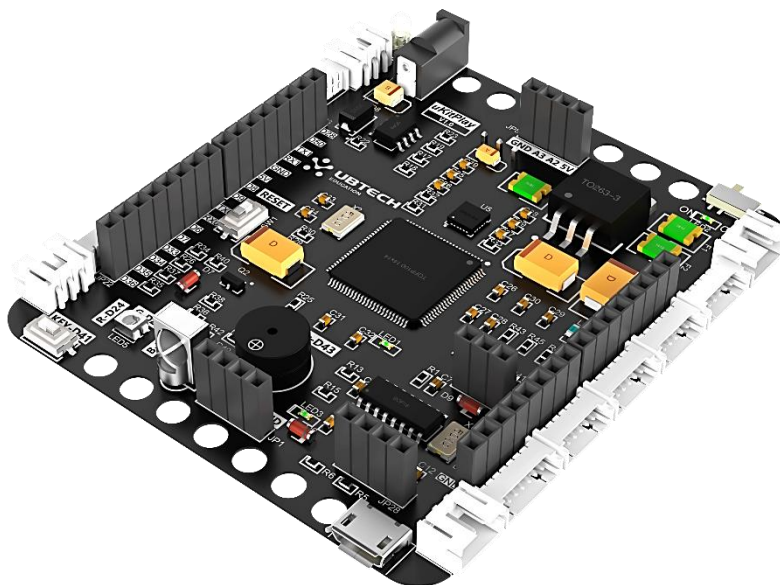
uKit Explore电子件



UIKit Explore结构件

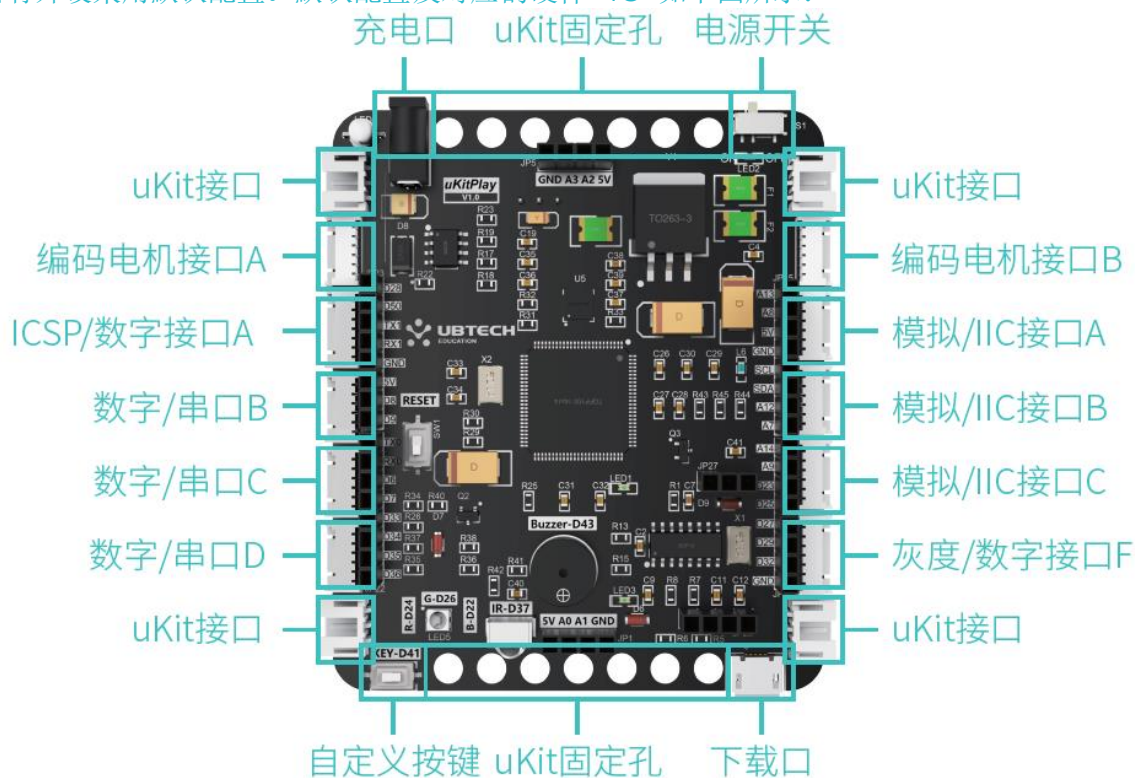
Explore 主控简介

Explore主控是优必选科技为uKit拼搭积木设计的开源平台硬件，兼容Arduino mega 2560。同时支持C/C++、blockly编程。可以控制uKit的舵机实现轮模式和舵机模式、板载提供蜂鸣器、RGB多彩灯、按钮和陀螺仪等丰富的扩展接口。支持 USB 及锂电池供电，支持板载锂电池充电功能，是移动设备、可穿戴电子产品、IOT 应用的最佳选择，可以直接应用于低功耗项目。

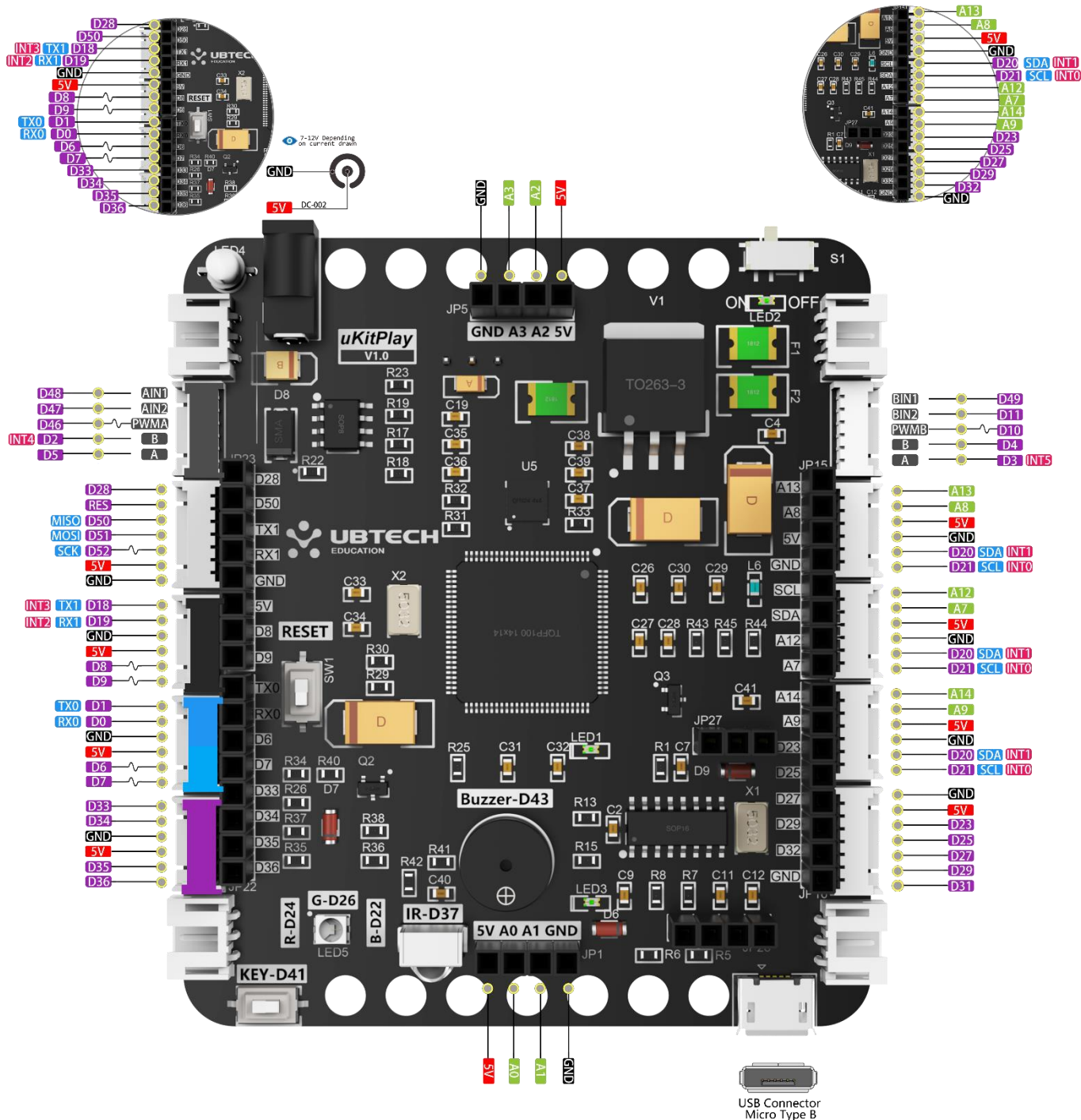


除此以外，uKit Explore 主控还具备丰富的外设，ADC，I2C，I2S，SPI，UART 等，且编程方式完全兼容 Arduino IDE 编程。

注意：uKit Explore 主控所有外设都可以配置到任意引脚，在使用 Arduino IDE 进行编程时，如果没有特殊配置，所有外设采用默认配置。默认配置及对应的硬件 IO 如下图所示：

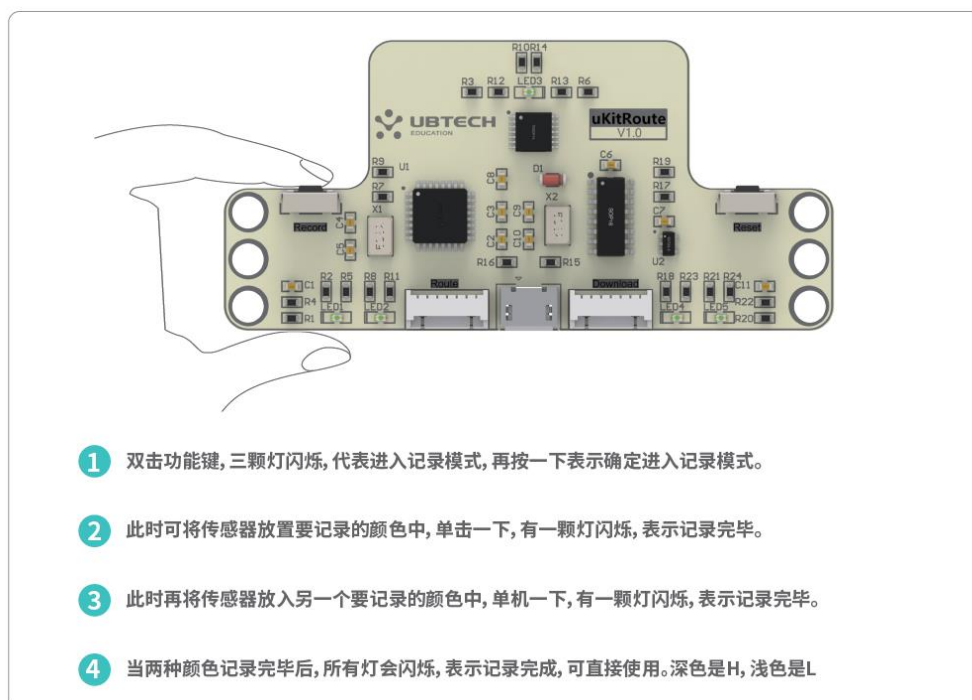
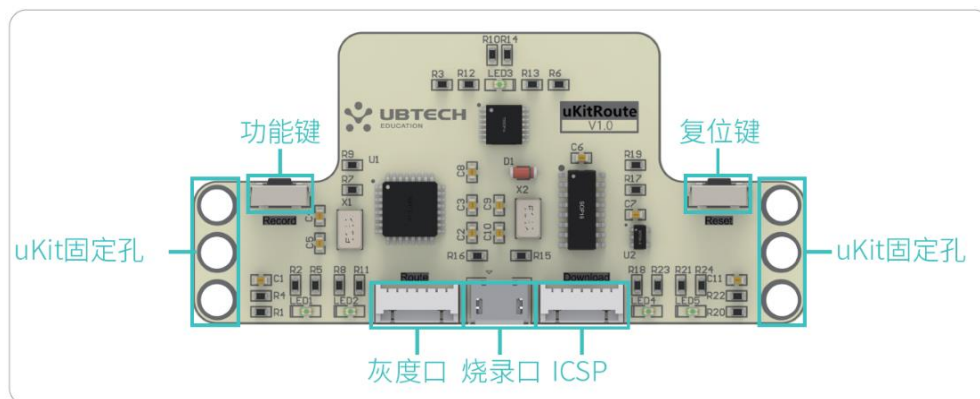


uKitPlay PINOUT



uKitRoute 智能灰度传感器简介

uKitRoute 智能灰度传感器是一款颠覆传统的高智能灰度传感器，是科技与艺术的完美结合。具有多检测点、智能化、多功能化的特点。市面上流行的寻线传感器、灰度传感器等要么只能识别特定的两种颜色、。其功能极其有限，随着技术日新月异的发展，传统的这些传感器已经远远无法满足使用要求。为了满足更多用户更多的需求uKitRoute 智能灰度传感器便应运而生。 uKitRoute 智能灰度传感器通过颜色采集，可以感知任意两种颜色然后输出相应的电平值。采集到的颜色具有断电保持特性，如果是同样的两种颜色的分辨无需再次采集可直接使用。灵活的颜色采集方式可实现一次采集多次使用，亦可根据需要随时采集。可以用在寻迹小车，智能机器人等与颜色互动的产品上面。让你的DIY创意插上翅膀，让你随时随地体验比赛的酣畅与轻松...



注意，读取数据时，需要5个点都在同一个颜色中，这样读出来的数据会比较准确，还有读取数据时放置一格的连接件读取数据会更加准确哦。



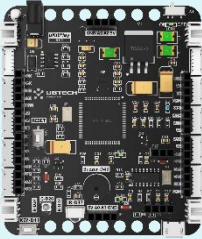
详细的传感器读取操作请查看视频：[视频链接](#)

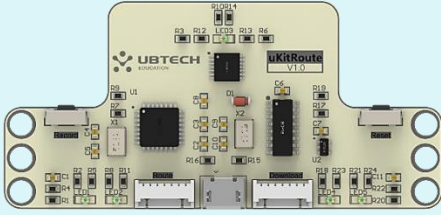
第二章：快速入门


uKit Explore 硬件


在第一章简介中，已经介绍了各个引脚的对应和默认 Arduino IDE 的外设配置引脚，这里就不再撰述，在使用时可以参考第一章简介内容进行相关引脚的特殊配置。

规格参数

uKit Explore	
参考图示	
工作电压	5V
输入电压	7.4V~9V
工作温度	-40℃~+85℃
主控芯片	ATMEGA2560 8bit 16MHz
蜂鸣器	5V 无源
LED	可编程RGB
按键	开机、复位
陀螺仪	MPU6050
保护电路	短接保护、过流保护
调试	硬串口x2
预留接口	I2C、SPI、UART、ADC、GPIO、PWM、uKit
尺寸	78×65×15mm
编程软件	ArduinoIDE
编程语言支持	C/C++、Blockly

巡线传感器	
参考图示	
工作电压	3-5V
探测距离	1~5cm
信号类型	数字输出
信号输出	黑H,白L
分辨率	5点
接口类型	MX1.25-7P

超声波传感器	
参考图示	
工作电压	5V
静态电流	小于2mA
感应角度	不大于15度
探测距离	2-60cm
尺寸	40*30 mm
接口类型	HRD2.45-4P

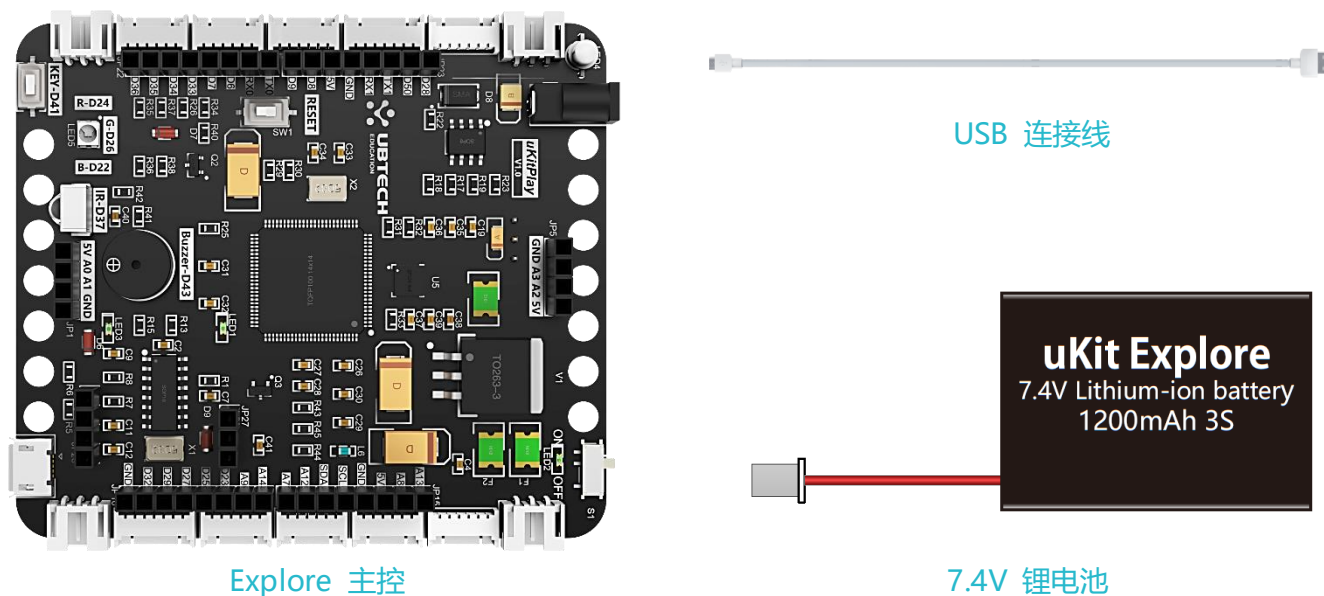
触碰传感器	
参考图示	
工作电压	3-5V
相应时间	60~220ms
信号类型	数字输出
信号输出	触碰H,无触碰L
接口类型	PH2.0-3P

Arduino IDE for Explore 主控

如果你是第一次接触 Arduino 开发平台，同样可以通过以下教程学习如何下载安装 Arduino IDE 软件，以及如何通过 Arduino IDE 编写 Explore 主控程序，完成您自己的项目。

开始之前，请确认您手中有一块 Explore 主控以及 USB 连接线、7.4V 锂电池，除此之外，您还需要一台运行 Windows/Mac OS/Linux 操作系统并且有网络连接的电脑。

我们需要的 Explore 主控 及 USB 连接线、7.4V 锂电池，如下图所示：



Arduino IDE 就是 uKit Explore 控制板的编程软件，通过以下步骤，可以快速搭建开发环境。


STEP1: 下载 Arduino IDE 软件

以下的步骤是基于 Windows 操作系统，如果你使用的是其他操作系统，可以将其作为参考。

首先，你需要从官网下载最新版本的 Arduino IDE 软件。下载链接：<http://arduino.cc/en/Main/Software>

如无法下载可访问百度网盘：<http://pan.baidu.com/s/1jIwniQi>

Download the Arduino IDE




ARDUINO 1.8.5

The open-source Arduino Software (IDE) makes it easy to write code and upload it to the board. It runs on Windows, Mac OS X, and Linux. The environment is written in Java and based on Processing and other open-source software.

This software can be used with any Arduino board. Refer to the [Getting Started](#) page for Installation instructions.

Windows Installer
Windows ZIP file for non admin install

Windows app 

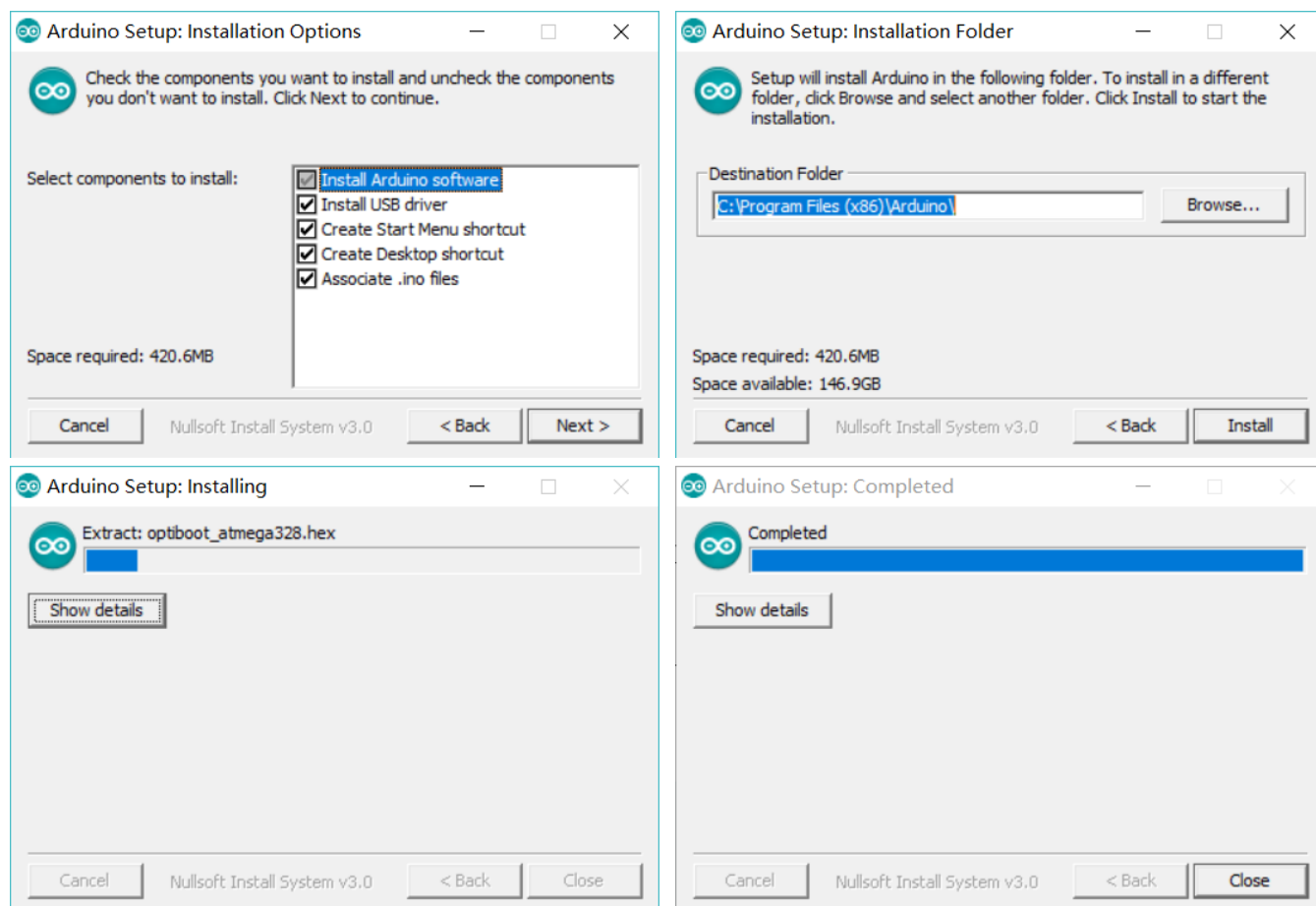
Mac OS X 10.7 Lion or newer

Linux 32 bits
Linux 64 bits
Linux ARM

[Release Notes](#)
[Source Code](#)
[Checksums \(sha512\)](#)

注意：建议使用 **Arduino 1.8.0** 以上版本，教程使用的是 **Arduino IDE 1.8.5** 版本。

在下载页右侧的列表中选择下载对应的安装包。对于 Windows 系统用户既可以选择下载 **Windows installer**（推荐初次使用者下载），也可以下载 Windows ZIP 安装包（需要手动安装驱动）。若选择的是 Windows installer，你可以直接执行安装程序，并跟随安装向导完成配置，驱动会在程序安装完成后自动安装。



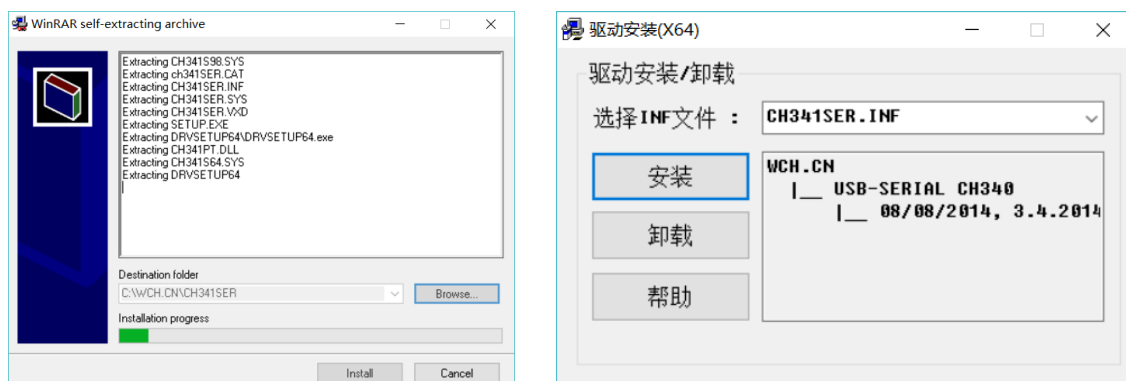
STEP2: Explore 驱动程序安装

安装好 Arduino IDE 之后，你还需要安装 Explore 主控的驱动程序，这样电脑才能识别到 Explore 的端口。

下载链接：

<https://pan.baidu.com/s/1dEZxrCP#list/path=%2FArduino%2Fdrivers%2Fch340&parentPath=%2FArduino%2Fdrivers>

在下载页面选择 Windows，然后把 CH341SER.EXE 下载到电脑。之前你可以直接执行安装程序，并跟随安装向导完成配置，驱动会在程序安装完成后自动安装。

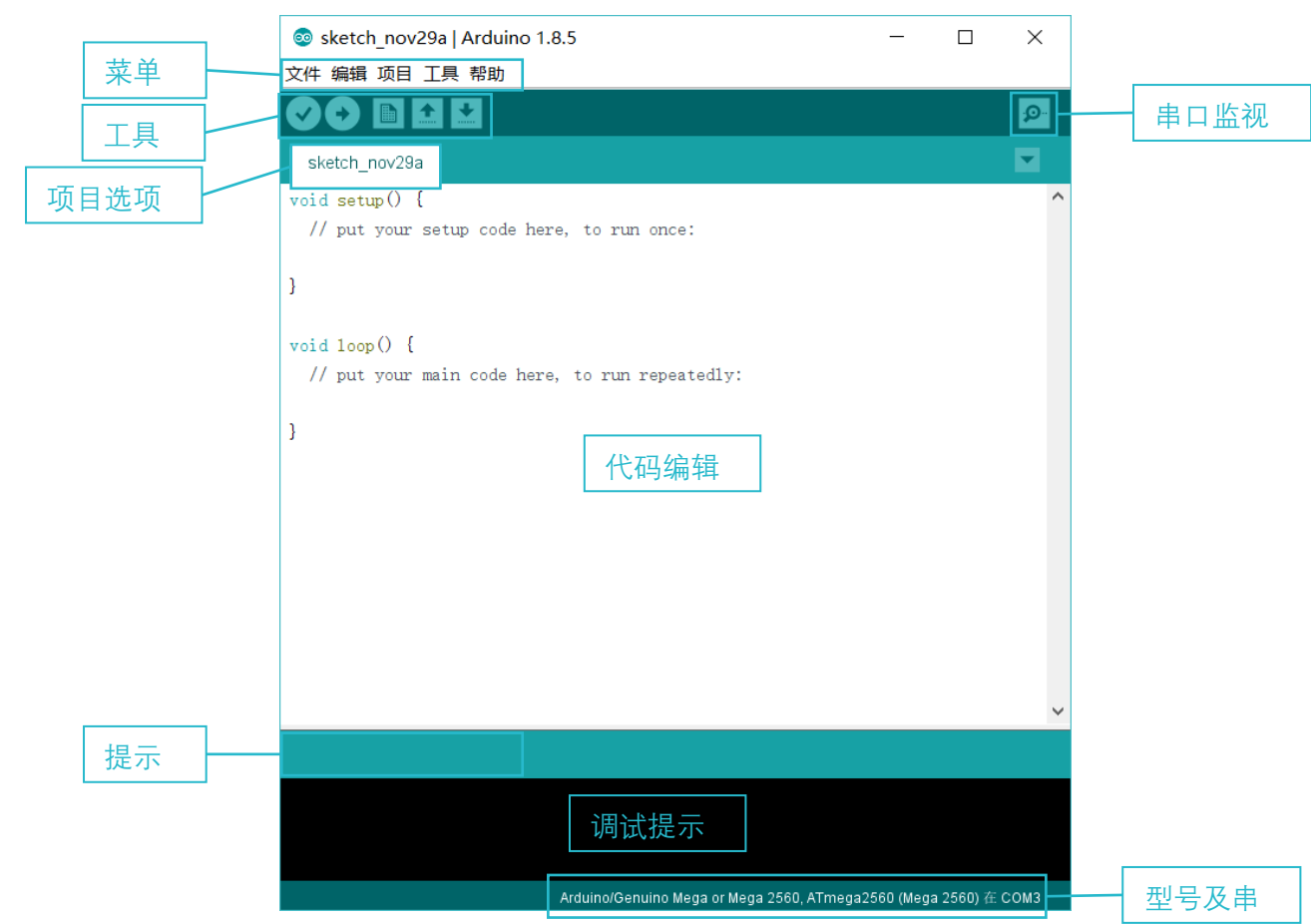


STEP3: 连接 Explore 主控至电脑

正确安装完成 Arduino IDE 和 驱动程序后，即可将 Explore 主控通过 USB 数据线连接至电脑。正确连接时电脑会发出“噔”的一声，表示识别到了端口。

STEP4: 在Arduino IDE 中进行编程

Arduino IDE 软件安装完成后，运行软件打开编程窗口。你可以在这个窗口里编辑并上传代码到 Arduino 开发板上，或是使用内置的串口监视器通过串口与开发板通信。现在让我们仔细看下 Arduino IDE 界面。



在工具栏上，Arduino IDE提供了常用功能的快捷键。如下表格

工具栏功能按键		
	校验（Verify）	验证程序是否编写无误，无误则编译
	下载（Upload）	下载程序到Arduino控制器上。
	新建（New）	新建一个项目。
	打开（Open）	打开一个项目。
	保存（Save）	保存当前项目。
	串口监视器（Serial Monitor）	用它可以查看串口发送接收到的数据

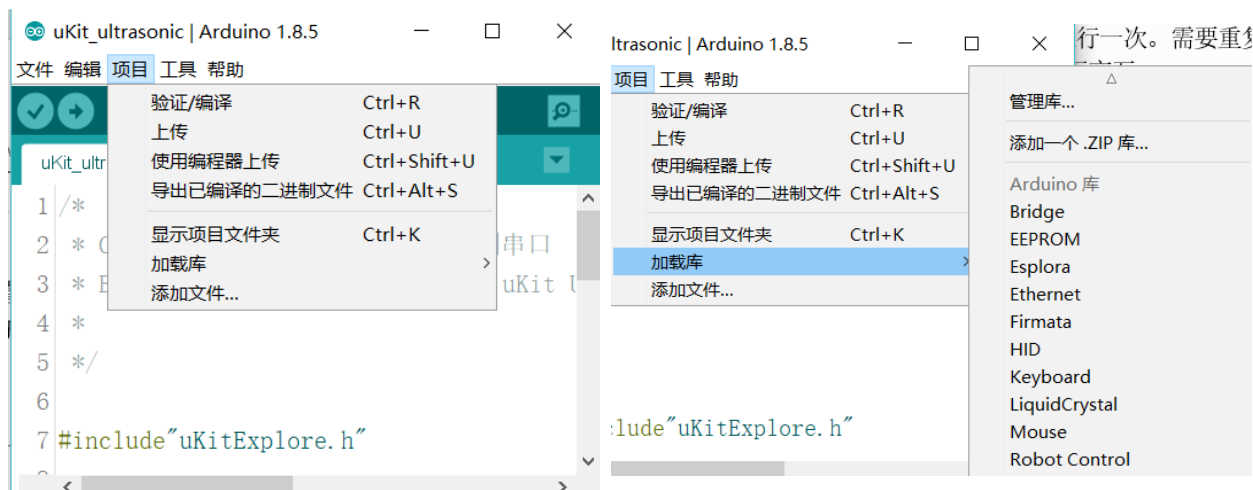
与常规 C 语言程序不同的是，一段用于 Arduino 的程序通常由 `void setup()` 部分与 `void loop()` 两部分构成。

"void setup()"用于放置初始化程序的代码，这部分代码在开发板上电后仅运行一次。需要重复运行的代码需要放置在"void loop()"中，的这些代码会一直重复运行，使得开发板时时与外部进行交互。

白色区域是编程区域。编程区域以下的黑色部分是信息窗口，用于显示代码上传以及编译的信息。你可以通过文件>首选项，设置相关打印信息。

STEP5: 添加 uKit Explore 库文件

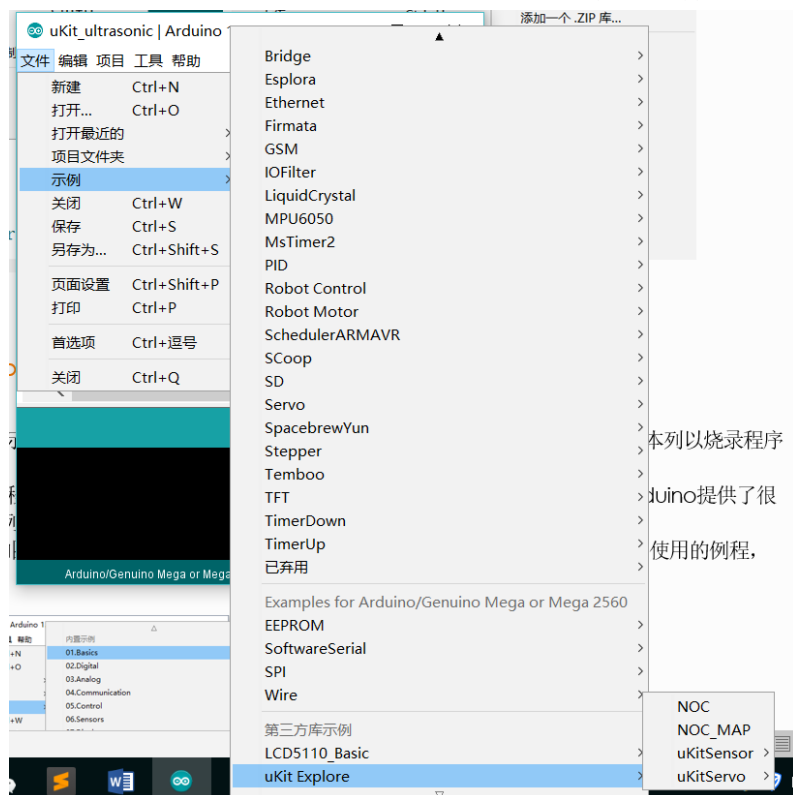
首先先下载一个名叫ubtrobot.zip的压缩包。然后打开Arduino IDE。找到项目->加载库->添加一个.ZIP库。找到你下载的ubtrobot.zip添加进去即可。然后你就可以在文件>示例里找到uKit Explore的代码示例了。



STEP6: 上传代码至 Explore 主板

在这一步，我们将演示如何上传示例程序"NOC"到Explore 主板。首先烧录程序需要有代码，本列以烧录程序为主。暂时不讲代码部分。代码可以打开实例程序。

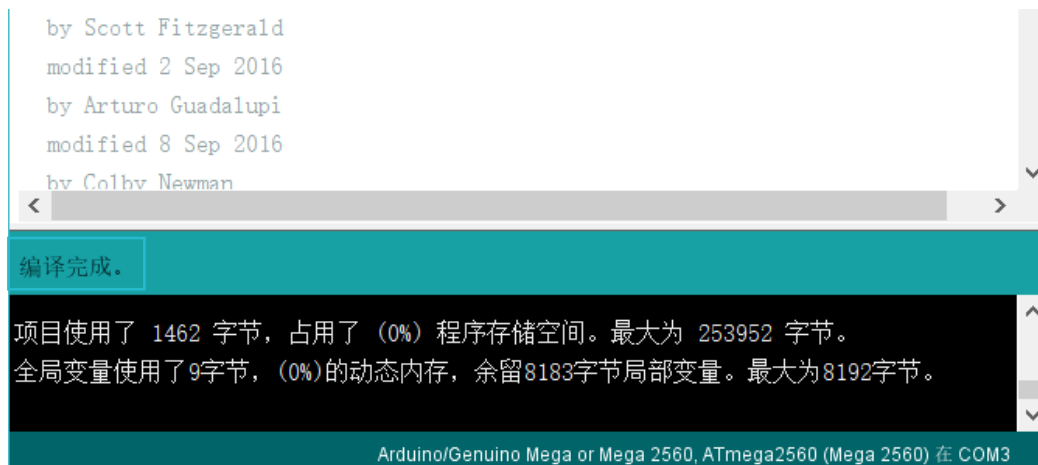
打开Arduino IDE。如图4.3.1，通过 Arduino IDE菜单>文件>示例>uKit Explore>NOC找到我们要使用的例



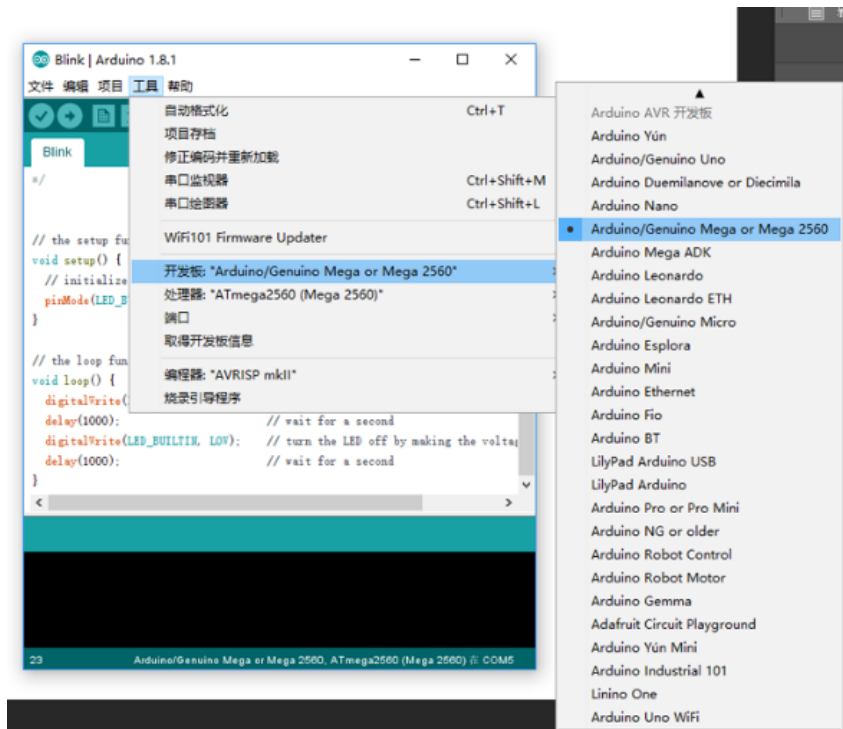
本列以烧录程序
uino提供了很
使用的例程，

程，单击便可打开

在上传之前，你应该首先确认代码中没有错误。点击“[编译](#)”确认。等待几秒钟，如果没有错误的话，一条“编译完成”信息会在信息窗口显示,表示 “编译成功”。如果出现错误可以返回检查程序是否完整。

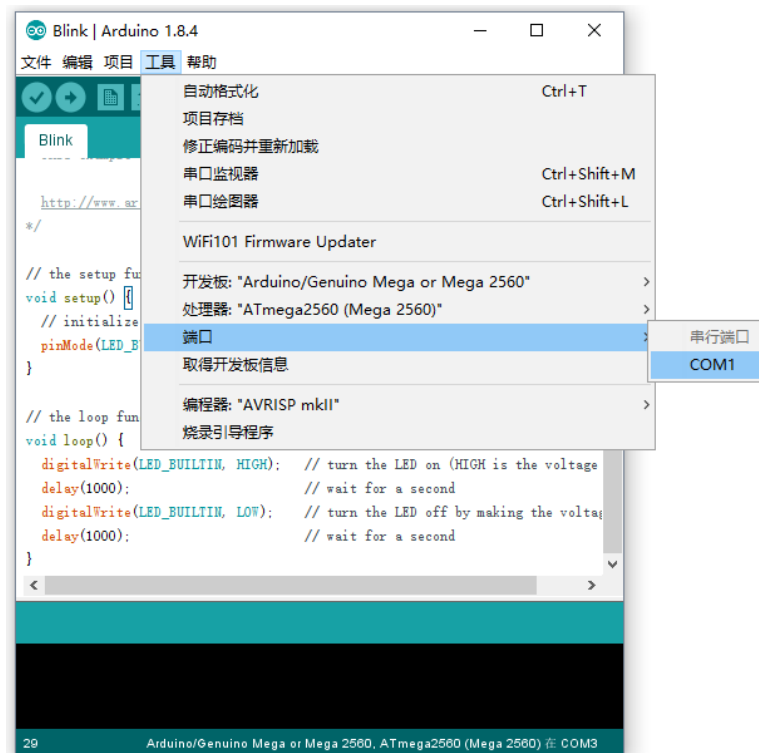



编程成功后，选择工具->开发板->Arduino/Genuino Mega or Mega 2560，通常在不切换到使用其他型号的开发板时，这个步骤仅需在第一次使用时执行一次即可。



切换到工具栏>工具>端口，，我们应该选择"COM1"作为通信端口。通信端口只有当开发板连接至电脑并被成功识别时才会出现。同一块开发板在插拔后可能会占用不同的端口上，所以我们每次上传前都要重复确认。

COM 选择完成后，开发板的信息和端口就会在窗口右下角显示。



最后，点击下载 （Upload）按键，调试提示区会显示“编译程序中”，很快该提示会变成“下载中”，此时Arduino 2560上标有TX、RX的两个LED会快速闪烁，这说明你的程序正在被写入Arduino 2560中。



成功上传后，“上传成功”消息会出现在信息窗口。



简而言之，为 Arduino 上传代码可以分为以下三个步骤：

- 编译代码；
- 选择开发板型号和端口；
- 上传！

以上就是使用 Explore 主板在 Arduino IDE 上面的一些基本的方法。如果在使用中存在任何疑问或者建议，欢迎访问我们的论坛联系我们。

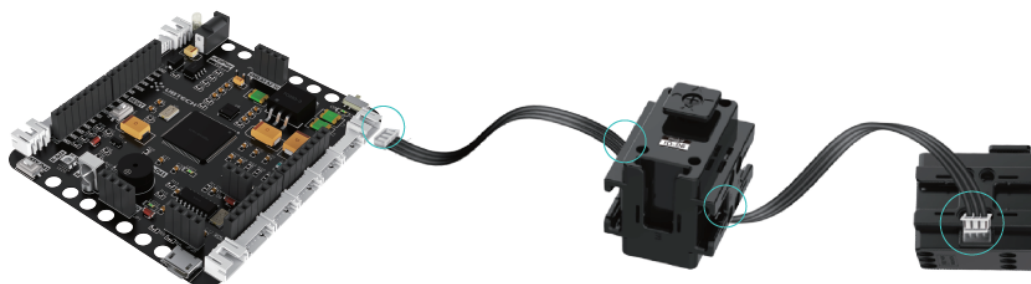
PIN 连接以及接线

我们3Pin的传感器以及舵机，都可用过串联连接，那么接下来给大家介绍如何串联。

1、使用3Pin线缆连接舵机

3 PIN 接口

使用 3 PIN 缆线将主控盒连接至舵机。主控盒与舵机间可传输能源与资讯。



连接完成后，方可使用。

2、接线

在我们的Explore-NT18M中，需要延长我们的连接线，连接地图主控，控制场地。接下来就为大家讲解一下如何连接我们的延长线。

1、我们的Explore-NT18M中有两种长线，分别为：1000mm、1500mm



3Pin, L=1000mm



3Pin, L=1500mm

线材连接线材时,需要注意一正一反的连接。

2、使用转接头，转接头分为正面和反面



正面无靠板



反面有靠板

3、注意，连接错误会导致舵机不运动。

1



第一步将反面与线材相连接。

2



第二步将正面与线材相连接。

注意连接位置。

第三章: Explore 基础使用

本章主要是通过一些示例项目，阐述 Explore 主控板的外设基本使用方法，通过下面的项目，您可以进行修改完成您的自己的项目。

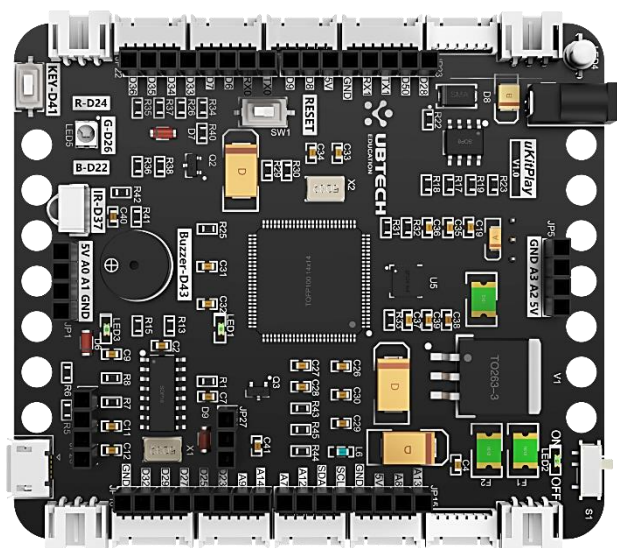
其中 ExPlore 外设主要包括：UART、I2C、SPI、ADC、PWM、DAC，以及内部集成的 RGB 彩灯、陀螺仪、红外遥控器、按键、蜂鸣器等。

项目一 串口实验

在最开始的章节中，现在，我们使用 UART串口，每秒打印一次计时数据。

所需元件

- 1 x Explore 主板
- 1 x USB 数据线



Explore 主控



USB 连接线

硬件连接

该项目不需要连接其他传感器，只需要使用 USB 线连接 Explore 主控和电脑即可。
输入代码

打开 Arduino IDE。我们还是建议您自己手动输入代码熟悉下。

代码如下:

```
void setup() {  
    Serial.begin(115200);  
}  
void loop() {  
    static unsigned long i =  
    0; Serial.println(i++);  
    delay(1000);  
}
```

输入完成后, 点击“[编译](#)”检查代码有无错误。确保没有错误后就可以开始上传了, 点击“[上传](#)”之后 IDE 会把代码发送给 Explore 主板。

注意: 如果你忘了怎么编译和上传代码, 请参考前面的入门教程。样例程序在《Course》文件夹下的 *Item-1* 程序

实验现象

完成之前步骤的上传后, 打开 Arduino IDE 自带的串口监视器, 可以看到如下的打印信息:

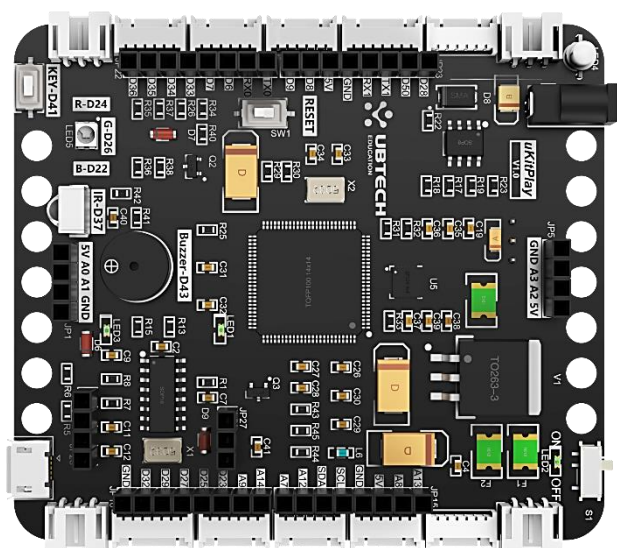


项目二 炫彩RGB灯

现在介绍一种新的LED——RGB LED。之所以叫RGB，是因为这个LED是由红(Red)、绿(Green)和蓝(Blue)三色组成。我们电脑的显示器也是由一个个小的红、绿、蓝点组成的。可以通过调整三个LED中每个灯的亮度就能产生不同的颜色。这个项目就是教你通过一个RGB小灯随机产生不同的炫彩颜色。我们可以先感性的看一下，按下图连接硬件并输入代码。

所需元件

- 1 x Explore 主板
- 1 x USB 数据线

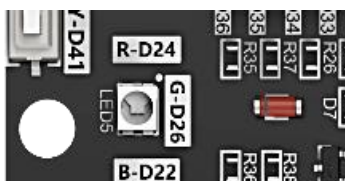


Explore 主控



USB 连接线

硬件连接



该项目不需要连接其他传感器，RGB 彩灯可以使用 Explore 主板上面的 RGB 彩灯，硬件连接方面只需要使用 USB 线连接 Explore 主控和电脑即可。

注意：板载的RGB彩灯连线情况 R-24、G-26、B-22。

输入代码

打开 Arduino IDE。您可以打开 [示例->uKit Explore->Sensor->RGBLED](#) 代码，但我们还是建议您自己手动输入代码熟悉下。

代码如下:

```
/*
 * CN:RGB灯代码
 *
 */

#include "uKitExplore.h"

void setup() {
    Initialization(); //CN:初始化/EN:Initialization
}

//红色亮300ms,绿灯亮300ms, 蓝灯亮300ms, 循环
void loop() {
    colorRGB(255,0,0);
    delay(300);
    colorRGB(0,255,0);
    delay(300);
    colorRGB(0,0,255);
    delay(300);
}
```

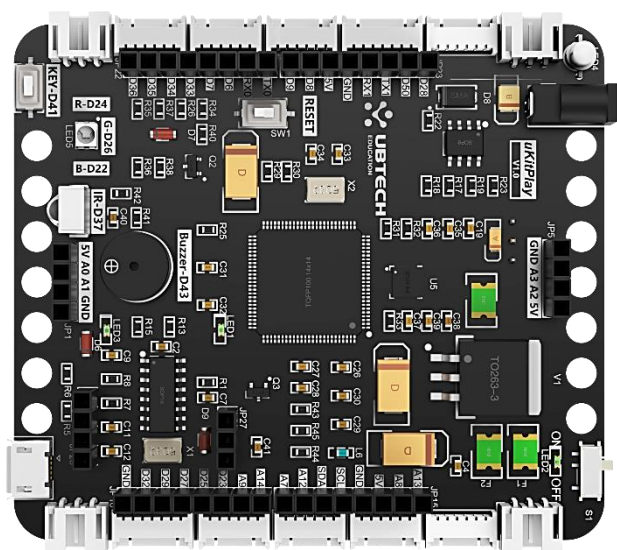
输入完成后, 点击“[编译](#)”检查代码有无错误。确保没有错误后就可以开始上传了, 点击“[上传](#)”之后 IDE 会把代码发送给 Explore 主板。

项目三 蜂鸣器

这里我们要接触一个新的电子元件——蜂鸣器，从字面意思就可以知道，这是一个会发声的元件。这次做一个报警器，通过连接蜂鸣器到Arduino数字输出引脚，并配合相应的程序就可以产生报警器的声音。其原理是利用正弦波产生不同频率的声音。这个项目就是教你通过一个敲打代码让蜂鸣器发声，按下图连接硬件并输入代码。

所需元件

- 1 x Explore 主板
- 1 x USB 数据线



Explore 主控



USB 连接线

硬件连接



该项目不需要连接其他传感器，蜂鸣器可以使用 Explore 主板上面的蜂鸣器，硬件连接方面只需要使用 USB 线连接 Explore 主控和电脑即可。

注意：板载的蜂鸣器连线情况 Buzzer-43。

输入代码

打开 Arduino IDE。您可以打开 [示例->uKit Explore->Sensor->buzzer](#) 代码，但我们还是建议您自己手动输入代码熟悉下。

代码如下:

```
/*
 * CN:蜂鸣器控制代码
 *
 */
#include "uKitExplore.h"

void setup() {
    Initialization(); //CN:初始化/EN:Initialization
}

//红色亮300ms, 绿灯亮300ms, 蓝灯亮300ms, 循环
void loop() {
    tone(300, 300);
    tone(400, 300);
    tone(300, 300);
    tone(400, 300);

}
```

输入完成后, 点击“[编译](#)”检查代码有无错误。确保没有错误后就可以开始上传了, 点击“[上传](#)”之后 IDE 会把代码发送给 Explore 主板。

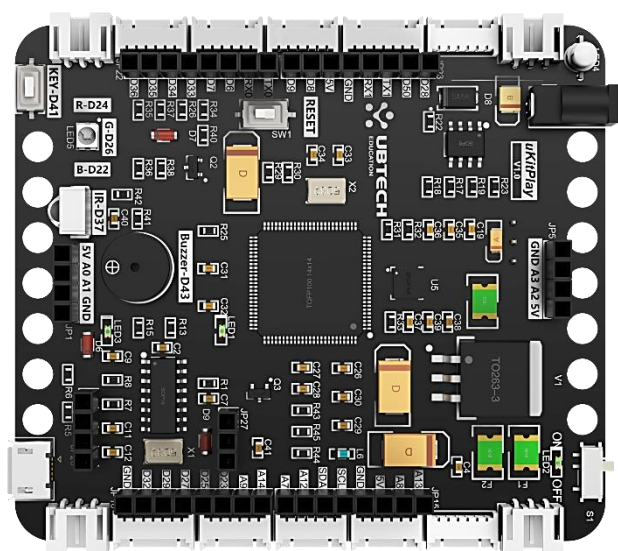
项目四 4KG舵机 舵机模式控制

舵机是一种电机，它使用一个反馈系统来控制电机的位置。可以很好掌握电机角度。大多数舵机是可以最大旋转180°的。也有一些能转更大角度，甚至360°。舵机比较多的用于对角度有要求的场合，比如摄像头，智能小车前置探测器，需要在某个范围内进行监测的移动平台。又或者把舵机放到玩具，让玩具动起来。还可以用多个舵机，做个小型机器人，舵机就可以作为机器人的关节部分。所以，舵机的用处很多。

优必选科技提供的4KG舵机能让舵机实现舵机模式控制，也可以实现电机控制。我们提供了调用库，让我们使用舵机变得更方便了。先从简单入手，套件这个4KG小舵机是-118°~118°的，我们就让它在-118°~118°之间来回转动。这个代码是让1号舵机旋转20°，300ms后舵机旋转-20°，300ms之后停止舵机,300ms后又重新开始。

所需元件

- 1 x Explore 主板
- 1 x USB 数据线
- 1 x 4KG 舵机
- 1 x 7.4V 锂电池



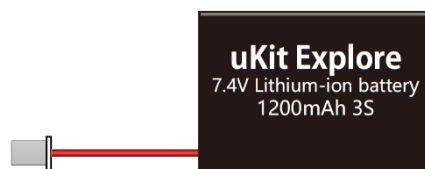
Explore 主控



USB 连接线



4KG舵机



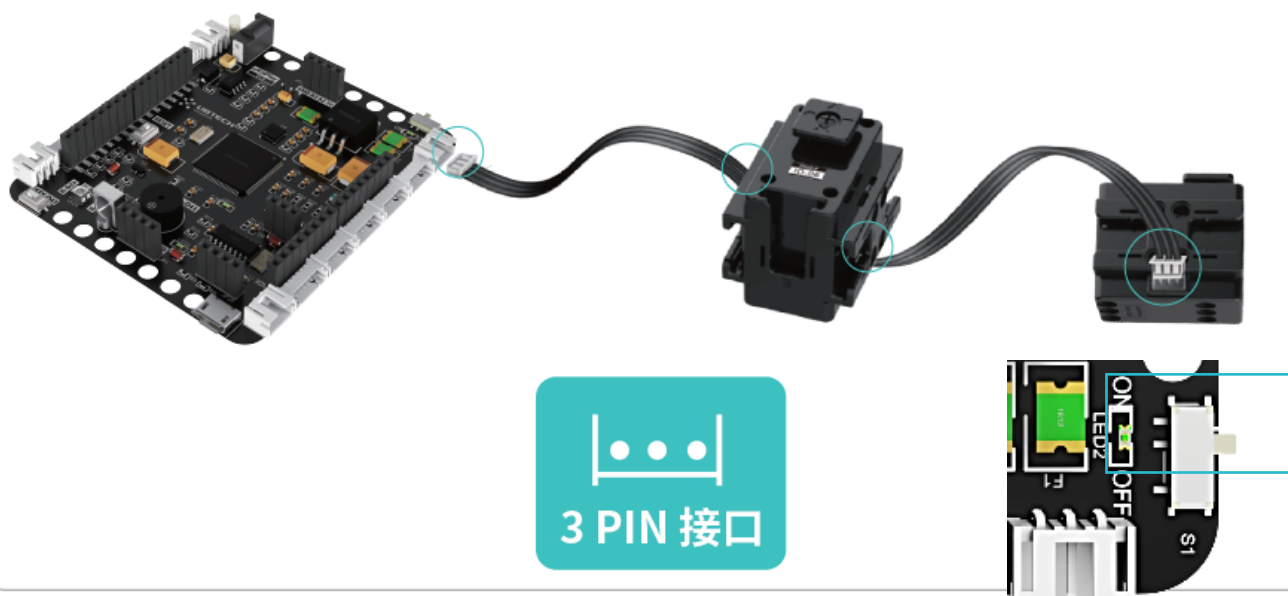
7.4V 锂电池

硬件连接

该项目不需要连接其他传感器，舵机连接在 Explore 主控板的任意一个 3PIN 口即可。舵机可以并联多个。在控制时注意舵机的ID。舵机是依靠ID来控制的。硬件连接方面要使用 USB 线连接 Explore 主控和电脑，3PIN口接入舵机、Explore 主控接入电池即可。这里我们接入1号舵机。

3 PIN 接口

使用 3 PIN 缆线将主控盒连接至舵机。主控盒与舵机间可传输能源与资讯。



注意：烧录程序后，需将开关拨到ON的位置，把电源打开才能驱动舵机。

输入代码

打开 Arduino IDE。您可以打开[示例->uKit Explore->uKitServo->uKitServo_Angle_Mode](#)代码，但我们还是建议您自己手动输入代码熟悉下。

代码如下：

```
#include "uKitExplore.h"
void setup(){
    Initialization();//CN:初始化/EN:Initialization
}
//CN:ServoAngle第一个是di号，第二是角度 (-118°~118°)，第三是运行时间 (100~3000)
void loop(){
    ServoAngle(1,20,300);
    delay(300);
    ServoAngle(1,-20,300);
    delay(300);
    ServoStop(1);
    delay(300);
}
```

输入完成后，点击“[编译](#)”检查代码有无错误。确保没有错误后就可以开始上传了，点击“[上传](#)”之后 IDE 会把代码发送给 Explore 主板。

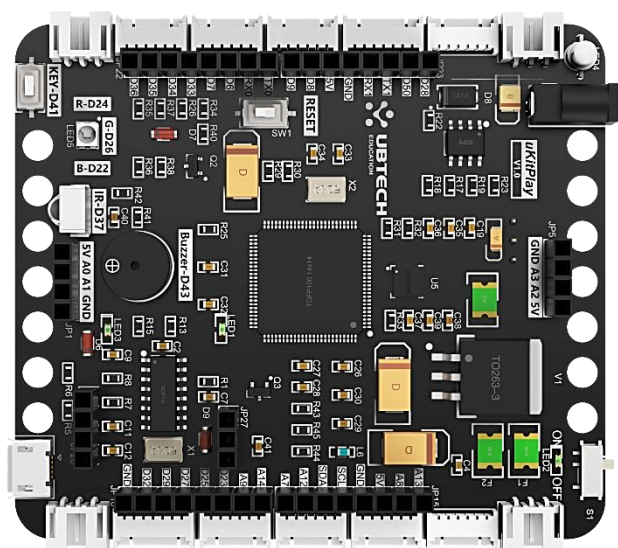
项目五 4KG舵机 轮模式控制

舵机是一种电机，它使用一个反馈系统来控制电机的位置。可以很好掌握电机角度。大多数舵机是可以最大旋转180°的。也有一些能转更大角度，甚至360°。舵机比较多的用于对角度有要求的场合，比如摄像头，智能小车前置探测器，需要在某个范围内进行监测的移动平台。又或者把舵机放到玩具，让玩具动起来。还可以用多个舵机，做个小型机器人，舵机就可以作为机器人的关节部分。所以，舵机的用处很多。

优必选科技提供的4KG舵机能让舵机实现舵机模式控制，也可以实现电机控制。我们提供了"NOC.h"库，让我们使用舵机变得更方便了。先从简单入手，套件这个4KG小舵机是-118°~118°的，我们就让它在-118°~118°之间来回转动。ID-1号uKit舵机以5的速度正转300ms,反转300ms,之后同时正转300ms,同时反转300ms.停止300ms。

所需元件

- 1 x Explore 主板
- 1 x USB 数据线
- 1 x 4KG 舵机
- 1 x 7.4V 锂电池



Explore 主控



USB 连接线



4KG舵机



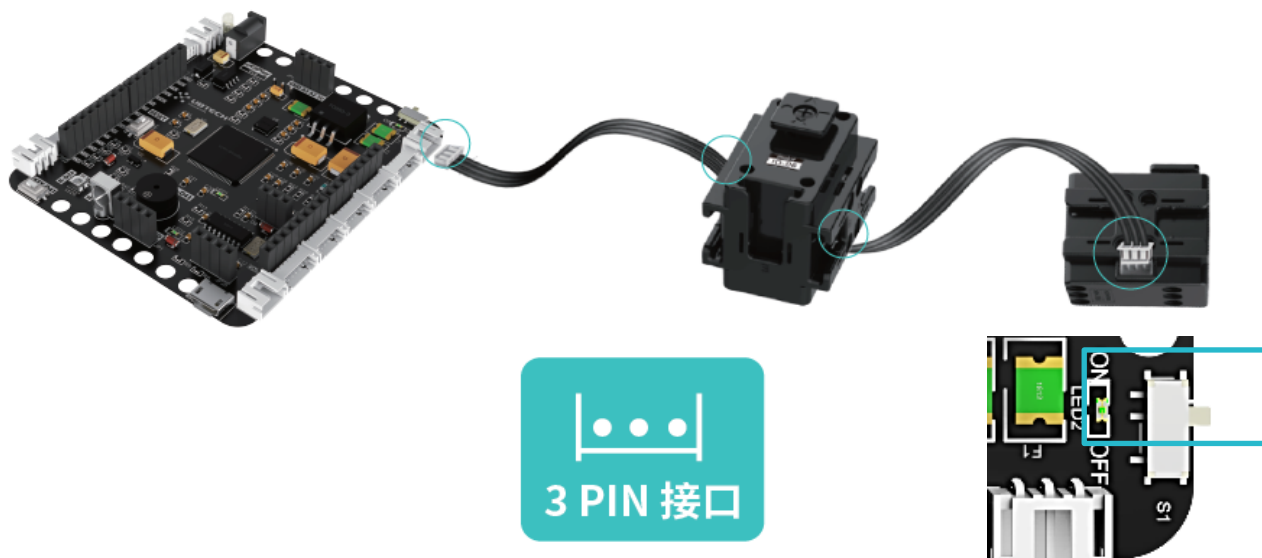
7.4V 锂电池

硬件连接

该项目不需要连接其他传感器，舵机连接在 Explore 主控板的任意一个 3PIN 口即可。舵机可以并联多个。在控制时注意舵机的ID。舵机是依靠ID来控制的。硬件连接方面要使用 USB 线连接 Explore 主控和电脑，3PIN口接入舵机、Explore 主控接入电池即可。这里我们接入1号舵机。

3 PIN 接口

使用 3 PIN 缆线将主控盒连接至舵机。主控盒与舵机间可传输能源与资讯。



注意：烧录程序后，需将开关拨到ON的位置，把电源打开才能驱动舵机。

输入代码

打开 Arduino IDE。您可以打开 [示例->uKit Explore->uKitServo->uKitServo_Rotate_Mode](#) 代码，但我们还是建议您自己手动输入代码熟悉下。

代码如下：

```
#include "uKitExplore.h"

void setup() {
    Initialization();//CN:初始化/EN:Initialization
}

void loop(){
    ServoRotate(1,0,5);
    delay(300);
    ServoRotate(1,1,5);
    delay(300);
    ServoRotate(1,0,5);
    ServoRotate(1,0,5);
    delay(300);
    ServoRotate(1,1,5);
    ServoRotate(1,1,5);
    delay(300);
    ServoStop(1);
    delay(300);
}
```

输入完成后，点击“[编译](#)”检查代码有无错误。确保没有错误后就可以开始上传了，点击“[上传](#)”之后 IDE 会把代码发送给 Explore 主板

项目六 舵机回读控制

我们的机器人之所以能现在一些复杂的动作是因为我们的舵机具有较高的精度，能实现一度的步进。另外还有一个其他舵机没有的功能—回读。我们提供的4KG舵机除了实现轮模式和高精度的舵机模式，还能读取舵机当前的实际值。这就给我们创造机器人动作提供了很大的空间，同时还能做出一些有趣的实验。事不宜迟，马上提供舵机回读的强大吧。

所需元件

- 1 x NOC DEMO 机器人
- 1 x USB 数据线



NOC DEMO机器人



USB 连接线

硬件连接

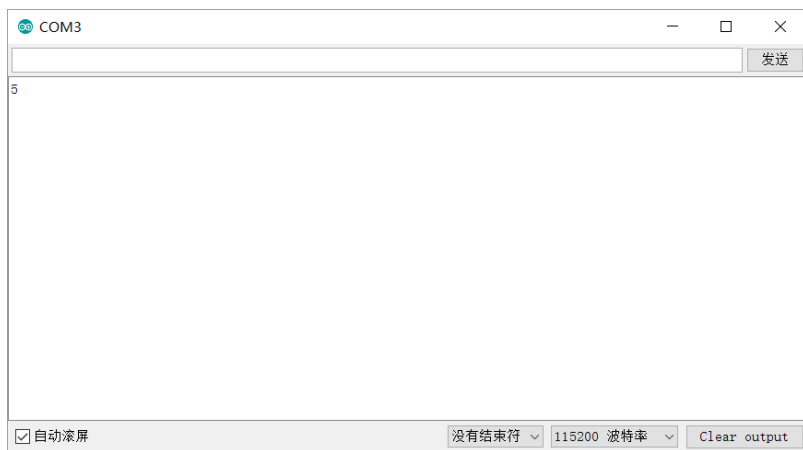
该项目只需要使用 USB 线连接 Explore 主控和电脑即可。

输入代码

打开 Arduino IDE。尽管您可以打开[示例->uKit Explore->uKitServo->ServoRead_PD_More](#)代码。这个代码是通过按主控上的按键来获取舵机的值的。只需在id里填入你想回读的舵机，紧接着打开串口，把波特率调为115200就可以拉。

部分代码代码如下：

```
#include "uKitExplore.h"
unsigned char id[6]={1,2,3,4,5,6};
void setup()
{
    Initialization();
}
void loop()
{
    button1.Update();//读取按键更新
    if(button1.clicks == 1)//按键按一下
    {
        ServoRead_PD_M(id,6);
    }
}
```



输入完成后，点击“[编译](#)”检查代码有无错误。确保没有错误后就可以开始上传了，点击“[上传](#)”之后 IDE 会把代码发送给 Explore 主板。然后机器人开机，上传完成后，数据线不用拔下。然后打开串口监视器。如下图，然后把波特率设置为115200。

此时，你就可以掰动机器人为你想要的动作，然后按下主板上的自定义按键。串口监视器就会显示当前你想要获取舵机的数值。

比如你想读取舵机1，2，3，4，5。那么你只要修改`mm.read_id[]={1,2,3,4,5};`然后再上传程序即可。

那么如何创建一个动作呢？框架如下：

```
#include"uKitExplore.h"
const signed char id[]={1,2,3,9,10,11,4,5,6,14,15,16};
const signed char xxx[]={sizeof(id)/sizeof(id[0])}={{}};
const int xxx_times[]={};
void setup()
{
    Initialization();
}
void loop()
{
    motion_xxx(1);
}

void motion_xxx(char times)//人形模式前进
{
    for(int c=0;c<times;c++)
    {
        for(int i=0;i<sizeof(xxx)/sizeof(xxx[0]);i++)
        {
            for(int t=0;t<sizeof(id)/sizeof(id[0]);t++)
            {
                ServoAngle(id[t],(xxx[i][t]),300);
            }
            delay(xxx_times[i]);
        }
    }
}
```


第四章: uKit Explore库

本章主要是通过一些示例项目，阐述 UKit Explore 库基本使用方法，通过下面的项目，您可以进行修改完成您的自己的项目。

其中 UKit Explore库包括了彩灯、蜂鸣器、巡线、动作等调用。

项目一 uKit Explore库介绍

关于uKit Explore库最主要的函数都在下面了。其他的函数可以不用理解，其他只是舵机与Arduino的通讯方式。主要函数如下：

uKit Exploe库函数说明-Sensor类函数	
<code>void Initialization();</code>	库初始化函数。在使用库时必须加这个函数。
<code>void colorRGB(int red, int green, int blue);</code>	设置板载RGB彩灯颜色的函数。Red范围：0-255、Green范围：0-255、blue范围：0-255
<code>void setcolor(int color);</code>	设置板载RGB彩灯颜色的函数。Color为1是红色，2-绿色，3-蓝色，4-黄色，5灭灯
<code>float Battery_check();</code>	检测电池电量的函数。返回电压值。电池电压低于6.2v时需要充电。
<code>float HcSr04Dis(char jp);</code>	超声波函数。返回cm, JP是位置，超声波若接在JP1, 那么JP为1。
<code>float HcSr04Dis(char EchoPin,char TrigPin);</code>	超声波函数。返回cm, JP是位置，超声波若接在JP1, 那么JP为1。
<code>void tone(uint16_t frequency, uint32_t duration = 0);</code>	调用板载蜂鸣器函数。 Frequency是蜂鸣器发声音频率， duration是响的时间
<code>void tone(int pin, uint16_t frequency, uint32_t duration);</code>	调用板载蜂鸣器函数。Pin蜂鸣器引脚， Frequency是蜂鸣器发声音频率， duration是响的时间。

uKit Exploe库函数说明-uKitServo类函数
void ServoRotate(char id,int dir, int speed);
舵机轮模式控制，id是舵机号，dir是方向（1顺时针，0逆时针），speed是速度（0-5）
void ServoAngle(char id,int angle, int times);
舵机舵机模式，id是舵机号，angle是角度（-118° ~118° ），times是运行时间（300-5000）
void ServoStop(char id);//单个舵机停止函数
单个舵机停止函数
int ServoRead_PD(char id);
单个舵机回读，返回舵机角度值(掉电回读)
void ServoRead_PD_M(char read_id[],char num);
多个舵机回读，返回舵机角度值(掉电回读)
int ServoRead_NPD(char id);
单个舵机回读，返回舵机角度值(不掉电回读)
void ServoRead_NPD_M(char id);
多个舵机回读，返回舵机角度值(不掉电回读)

uKit Exploe库函数说明-uKitSensor类函数
int ukit_infrared(char ID)
ukit红外传感器控制函数，返回cm, (0-20) cm
void uKit_led(char id,char face,int times,int red,int green,int blue);
ukit led传感器。id为led的id号，face为表情种类（0）
int uKit_button(char id);
返回768无操作，返回769是单击，返回770是双击
int uKit_Ultrasonic(char id);
Ukit超声波传感器控制函数，返回距离值

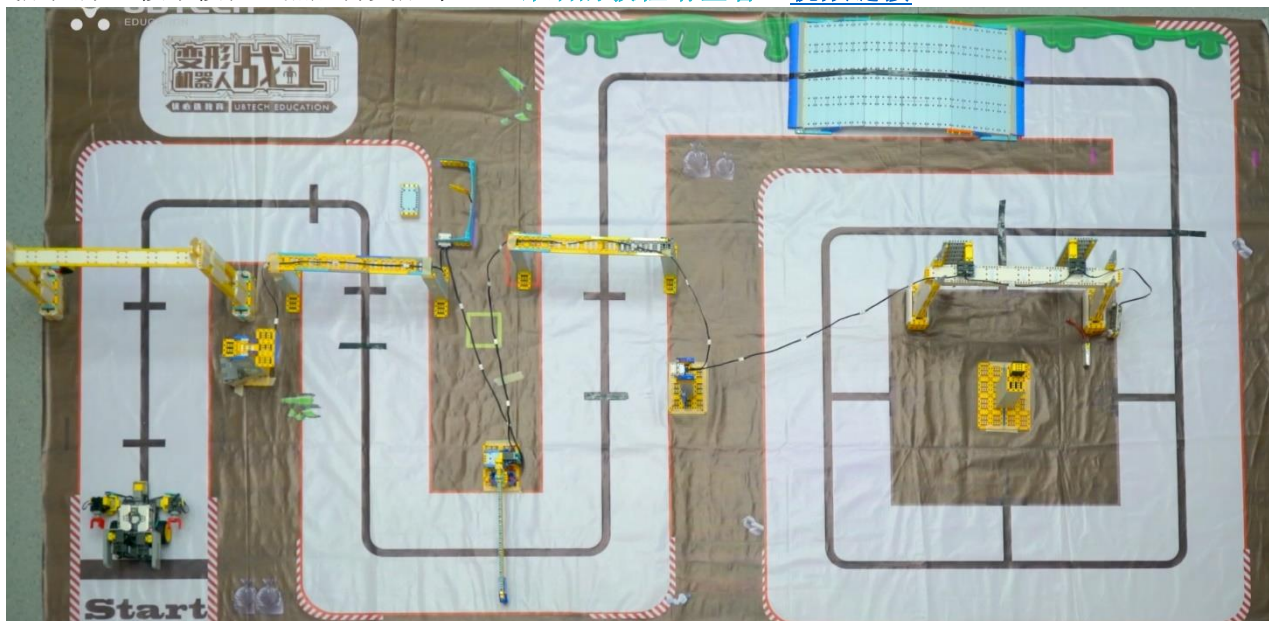
此车型函数和人型函数只针对NOC-DEMO机器人有效，且机器人需严格按照要求搭建DEMO。

uKit Exploe库函数说明-TransforRbot类函数-车型
void forward(int speed);
这是一个让小车前进的函数。前进只支持两轮。speed表示前进速度（速度范围：0~5）
void turnL(int speed);
这是一个让小车左转的函数。speed表示速度（速度范围：0~5）
void turnR(int speed);
这是一个让小车右转的函数。speed表示速度（速度范围：0~5）
void turn_L(int speed);
这是一个让小车左转90度的函数。speed表示速度（速度范围：0~5）
void turn_R(int speed);
这是一个让小车右转90度的函数。speed表示速度（速度范围：0~5）
void turn_U(int speed);
这是一个让小车掉头的函数。speed表示速度（速度范围：0~5）
void back(int speed);
这是一个让小车后退的函数。speed表示速度（速度范围：0~5）
void stop();
这是一个让小车停止的函数。
void findline(int speed);
这是一个让小车循迹的函数。speed表示速度（速度范围：0~765）
void motion_car(char times);
让机器人DEMO保持车形态，times表示执行动作的次数。这个动作原则上只需填1.
void motion_sitdown(char times);
让机器人从人形态变成车形态。times表示执行动作的次数。这个动作原则上只需填1.

uKit Exploe库函数说明-TransforRbot类函数-人型
void motion_getup(char times);
让机器人从车形态变成人形态。 times表示执行动作的次数。这个动作原则上只需填1.
void motion_forward(char times);
让机器人走路前进。 times表示执行动作的次数。一次为一步动作。
void motion_back (char times);
让机器人走路后退。 times表示执行动作的次数。一次为一步动作。
void motion_omotion(char times);
让机器人DEMO保持人形态， times表示执行动作的次数。这个动作原则上只需填1.
void motion_button(char times);
让机器人按下按键。 times表示执行动作的次数。一次为一步动作。
void motion_lift(char times);
让机器人举起右手。 times表示执行动作的次数。一次为一步动作。
void motion_lift_hand(char times);
让机器人举起双手。 times表示执行动作的次数。一次为一步动作。
void motion_head(char times);
让机器人上半身向前倾斜。 times表示执行动作的次数。一次为一步动作。

项目二 NOC编程使用

我们搭建好NOC-DEMO机器人后，需要让机器人执行任务。那么怎么让机器人去执行任务呢？我们就以NOC地图为例子。程序为机器人车形态开到桥洞的位置下，然后变成人形将人体感应炸弹击落，再变成车型态，开到重物旁边，继续前进把重物推入闸门感应区。之后后退回主轨道，继续行驶，开到城门锁定装置旁边，变成人形，按下按钮。然后再变成车型。 [详细的教程请查看：视频链接](#)



所需元件

- 1 x NOC DEMO机器人
- 1 x USB 数据线



NOC DEMO机器人



USB 连接线

硬件连接

该项目只需要使用 USB 线连接 Explore 主控和电脑即可。

输入代码

打开 Arduino IDE。尽管您可以打开 [示例->uKit Explore->NOC](#) 代码。其他无需理会。只需注意void cross(int speed)和void setup()里面的函数即可

部分代码代码如下：

```
void setup()
```

```
{  
    mm.init();//初始化  
    mm.motion_car(1);  
    mm.route[1]=4;  
    mm.route[2]=1;  
    mm.route[3]=2;  
mm.route[3]=3;  
}  
void cross(int speed){  
    switch (mm.route[mm.route_node]){  
        case 1:    //任务1  
            mm.stop();  
            mm.motion_getup(1);  
            delay(1000);  
            mm.motion_forward(3);  
            delay(500);  
            mm.motion_lift(2);  
            delay(1500);  
            mm.motion_sitdown(1);  
            delay(1000);  
            break;  
        case 2:    //任务2  
            mm.forward(speed);  
delay(1300);  
            mm.back(speed);  
            delay(1000);  
            break;  
        case 3:    //任务3  
            mm.stop();  
            mm.motion_getup(1);  
            delay(1000);  
            mm.motion_forward(3);  
            delay(500);  
            mm.motion_lift(2);  
            delay(1500);  
            mm.motion_sitdown(1);  
            delay(1000);  
            break;  
    }  
    case 4:    //任务3  
        mm.forward(speed);  
        delay(600);  
break;  
    mm.route_node = mm.route_node + 1; //准备取优化路径的下一个节点
```


}

输入完成后，点击“[编译](#)”检查代码有无错误。确保没有错误后就可以开始上传了，点击“[上传](#)”之后IDE会把代码发送给 Explore 主板。

代码解析

Case1: 解析

```
mm.stop();           //车停止
mm.motion_getup(1);  //执行车变人一次
delay(1000);         //等待1000ms
mm.motion_forward(3); //人形态前进3步
delay(500);          //等待500ms
mm.motion_lift(2);   //人形态按2次按键动作
delay(1500);         //等待1500ms
mm.motion_sitdown(1); //执行人变车一次
delay(1000);         //等待1000ms
```

这几句话就可以完成机器人车形态开到桥洞的位置下，然后变成人形将人体感应炸弹击落，再变成车型态。

Case2: 解析

```
mm.forward(speed);   //车形态前进
delay(1300);         //前进1300ms
mm.back(speed);      //车形态后退
delay(1000);         //后退1000ms
```

这几句话就可以完成开到重物旁边，继续前进把重物推入闸门感应区。之后后退回主轨道

Case3: 解析

```
mm.stop();           //车停止
mm.motion_getup(1);  //执行车变人一次
delay(1000);         //等待1000ms
mm.motion_forward(4); //人形态前进4步
delay(500);          //等待500ms
mm.motion_lift(2);   //人形态按2次按键动作
delay(1500);         //等待1500ms
mm.motion_sitdown(1); //执行人变车一次
delay(1000);         //等待1000ms
```

这几句话就可以完成开到城门锁定装置旁边，变成人形，按下按键。然后再变成车型。

Case4: 解析

```
mm.forward(speed);   //车形态前进
delay(1000);         //前进1000ms
```

这句话是跳过某些路口用的。

知道这些之后呢？我们看地图可知，车型开到第一个路口后，需继续前进。到达第二个路口后，执行然后变成人形将人体感应炸弹击落，再变成车型态。然后继续到达第三个路口后继续前进把重物推入闸门感应区。之后后退回主轨道。到达第四个路口后，变成人形，按下按键。然后再变成车型。

这些动作我们在case建好了，怎么告诉机器人在第几个路口做什么事情呢？

```
mm.route[1]=4;
mm.route[2]=1;
mm.route[3]=2;
mm.route[3]=3;
```

很简单mm.route[1]表示第一个路口，mm.route[2]表示第二个，以此类推。那么mm.route[1]=4;就是第一个路口前进。mm.route[2]=1;就是第二个路口执行然后变成人形将人体感应炸弹击落，再变成车型态。

注意：如果你忘了怎么编译和上传代码，请参考前面的入门教程。样例程序在《Course》文件夹下的Item-1 程序