

智能小车实践

Version by Hollow Man

一、小车的安装

1.智能小车部件

小车的零部件如下两图所示：

		
Arduino Mega 2560 电路板及其扩展板	树莓派 3b 主板	
		
电机	电池	充电器
		
主动轮	电源开关	超声波感应器
		
底座 A、B 板	亚克力板	万向轮

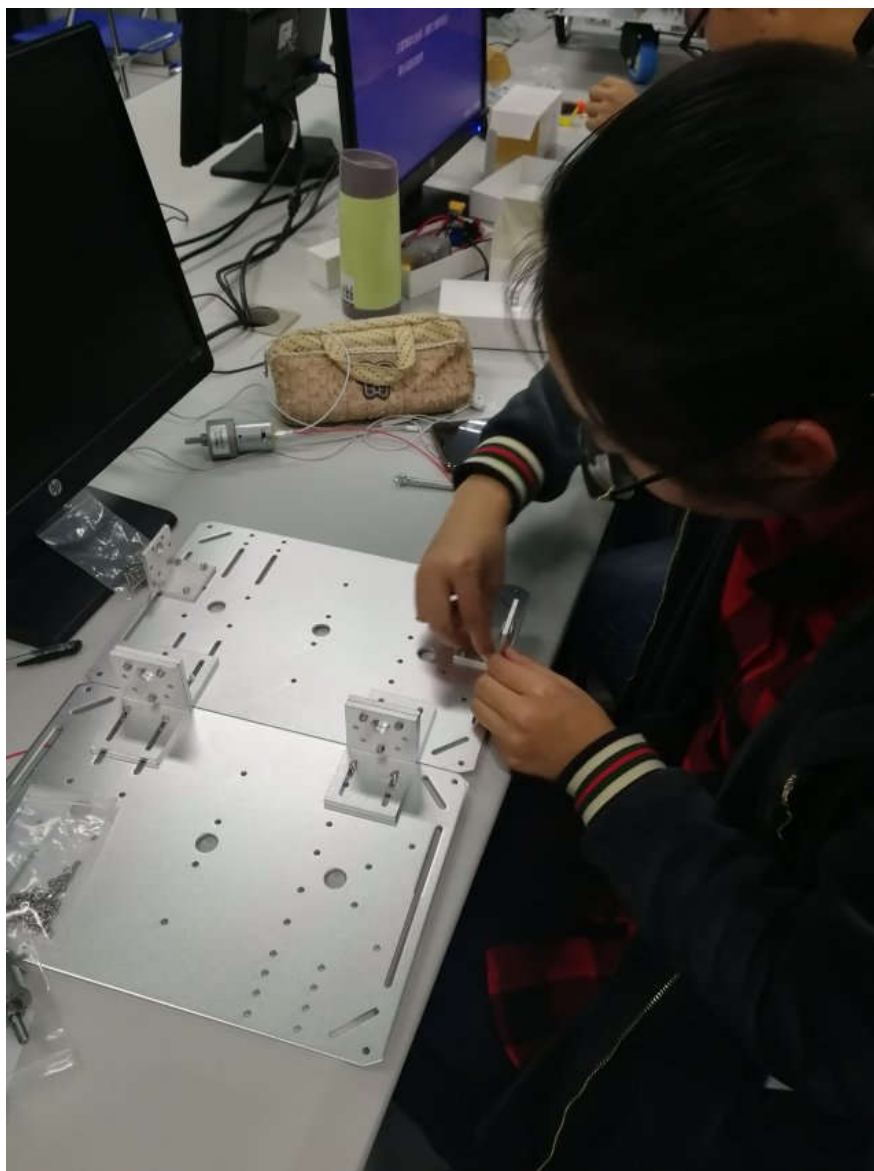
主要部件图示



次要部件图示

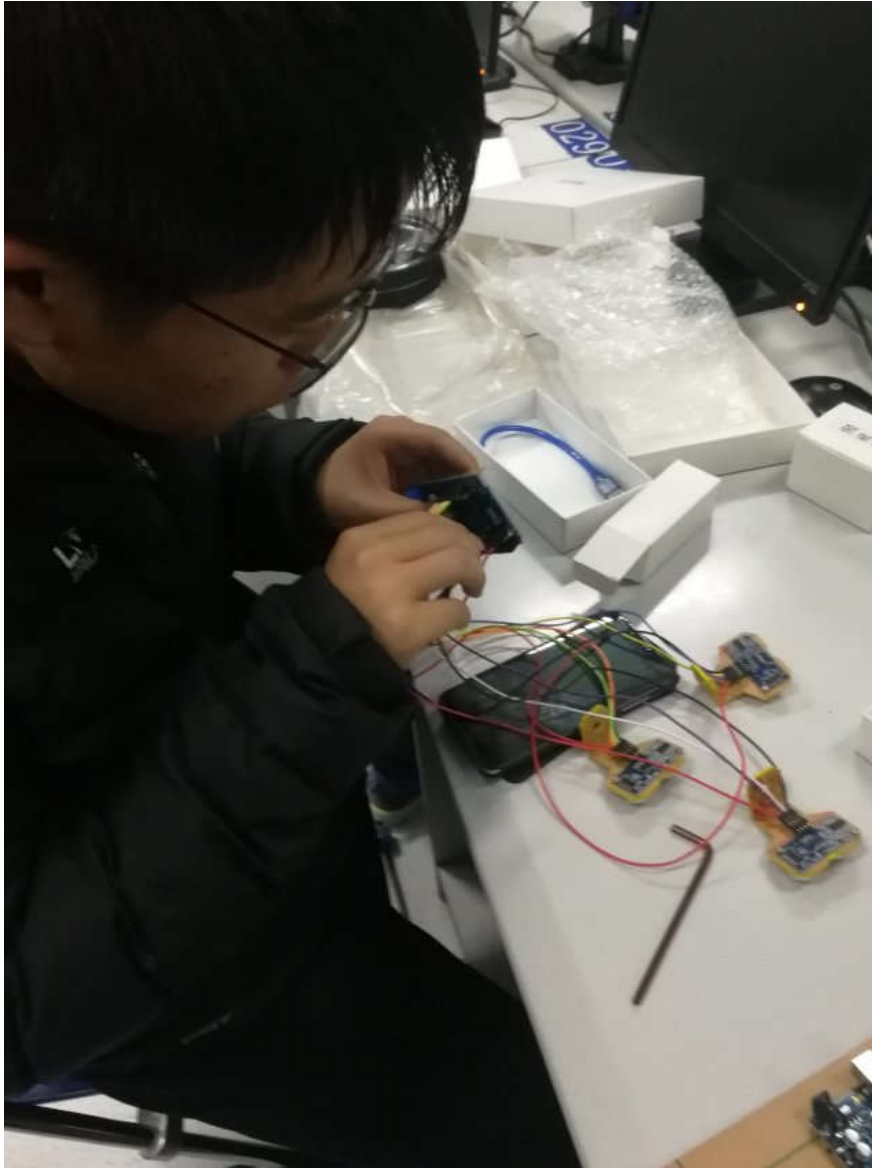
2.分工组装小车中

经过分工与协作，小组成员们正在认真地组装小车：



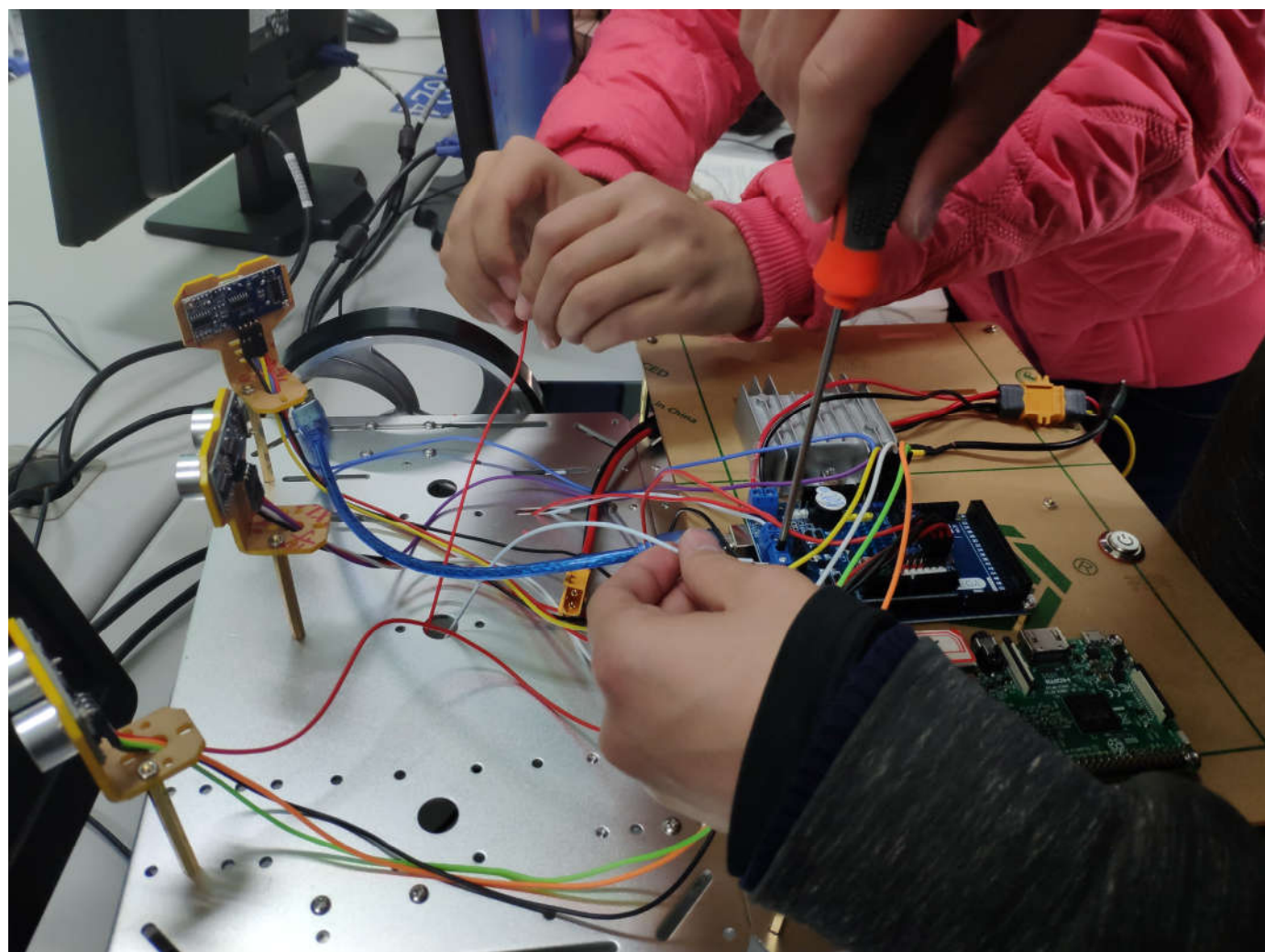
底座安装

对每个超声波传感器，选取红黑杜邦线各一根，其他颜色杜邦线容量共两根，合计四根线通过超声波传感器的下方的圆孔插入进来，再接到超声波传感器的四个引脚上。黑色杜邦线接到从左到右第一个引脚上，红色杜邦线接在第四个引脚上，剩余两根线的母头接在另外两个引脚上，这两个引脚**分别对应的是Echo引脚和Trig引脚**（从左到右）。将所有黑线和红线分别接到Arduino主板上的**GND和Vcc**引脚上，再将剩余的线（公头）分组接到1-9号引脚上。



超声波传感器接线

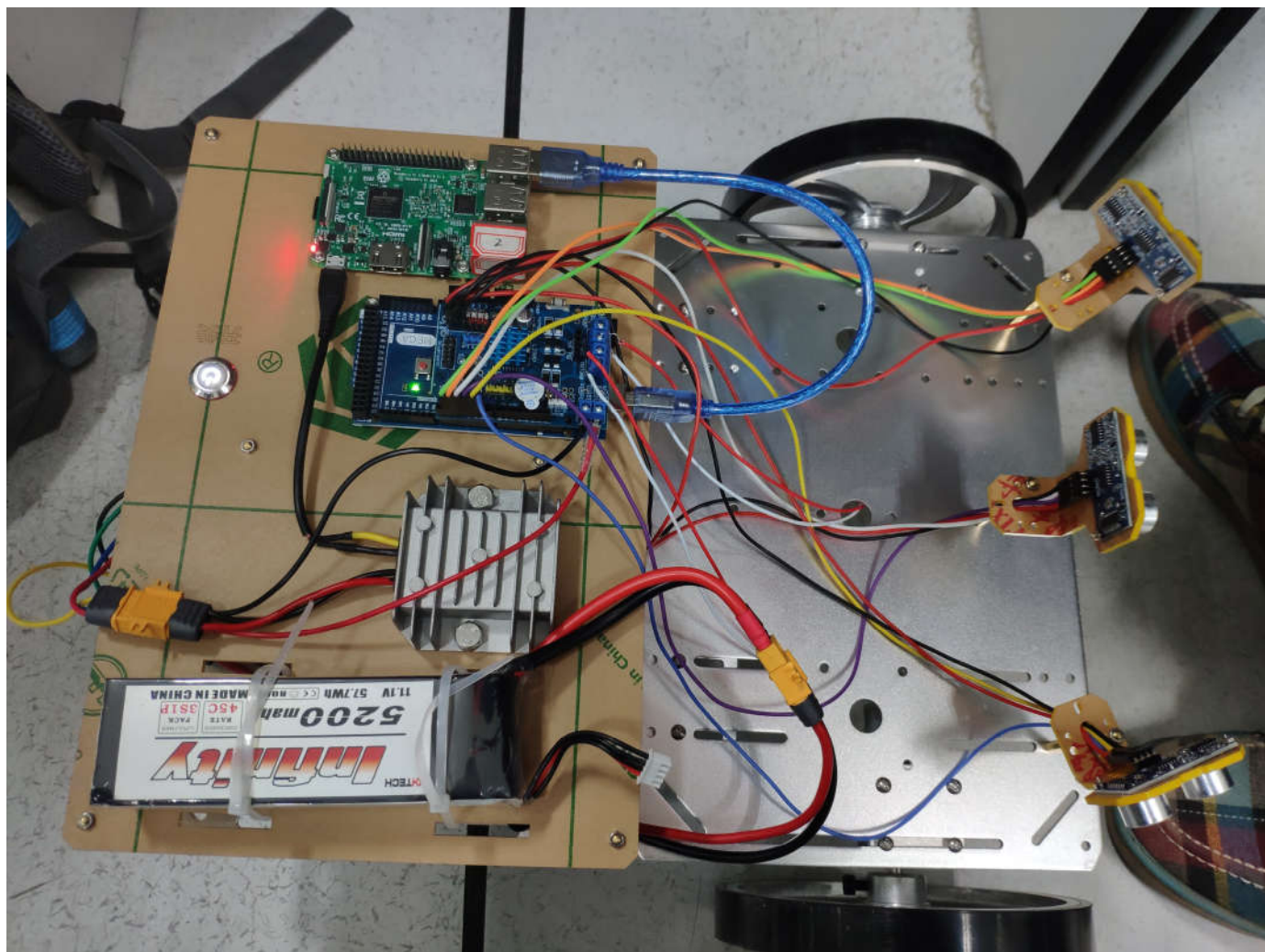
将红色和白色的电机线分两组分别连接电机与 Arduino 扩展板，对于每个电机各有一对儿红白电机线，将带橡胶保护套的一端接在电机上，另一端剪掉大约 3-5mm 的橡胶皮后将裸露的金属丝接在 Arduino 扩展板上如图所示的位置（MOTOTB 和 MOTORA 接口），从左到右分别为白红白红颜色电机线。其中从左到右第一组红白线接小车右轮，第二组红白线接小车左轮。（小车车头为两个主动轮所在位置，车尾为万向轮所在位置）



电动机接线

3.最终成果

经过了安装和调试之后，小车的最终样式如下图所示。



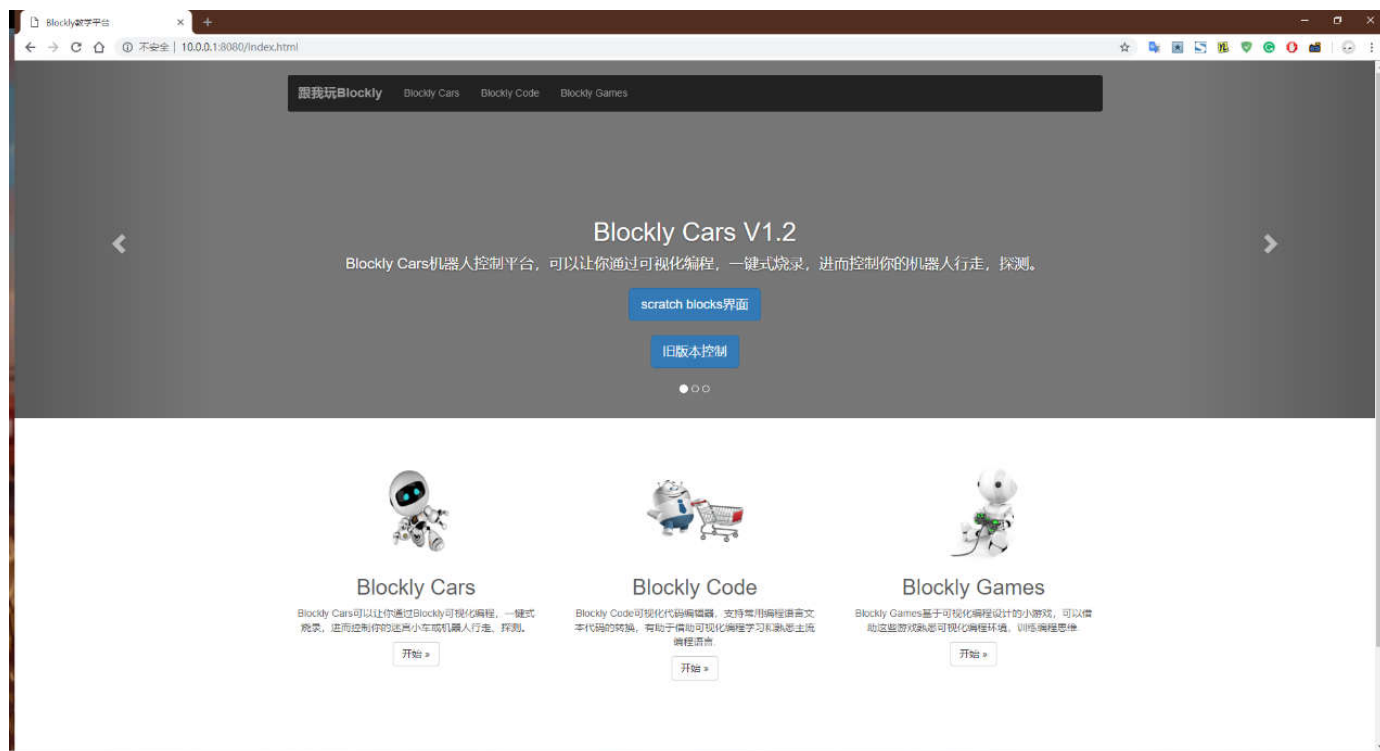
成果展示

二、小车的编程

 参考自《第三代Freduino智能小车安装指导说明书》。

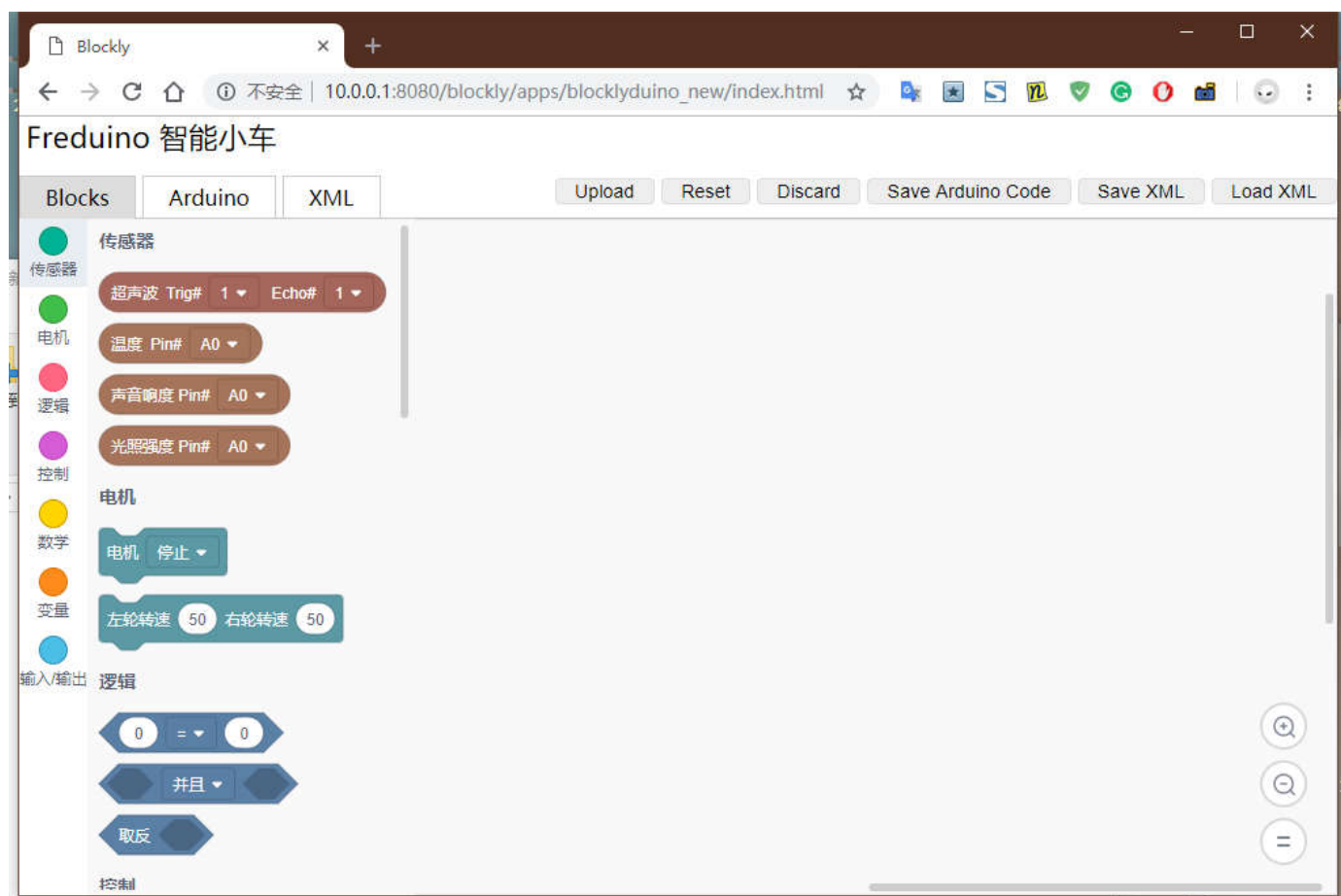
1.开始编程

小车正确启动后，该系统板会提供一个WIFI热点，供手机、平板等设备连接，该WIFI热点名称为Freduino_编号，此编号为小车所配系统板上的编号。我们组的WIFI热点名称为Freduino_2，热点密码为12345678。通过电脑连接到WIFI后，在终端浏览器中输入网址：10.0.0.1:8080，打开。



首页

选择“Scratch blocks界面”进入编程界面



Scratchblocks编程界面

在左侧的代码块工具箱中选择“电机”一栏中的电机(停止) 代码块（或者在代码块区域滑动鼠标滚轮即可找到该代码块），可以将它拖动到编辑区域当中。

Freduino 智能小车



电机控制模块

将停止改为前进

Freduino 智能小车



前进

然后单机右上角Upload 按钮，将代码上传到Arduino 控制板。

Freduino 智能小车



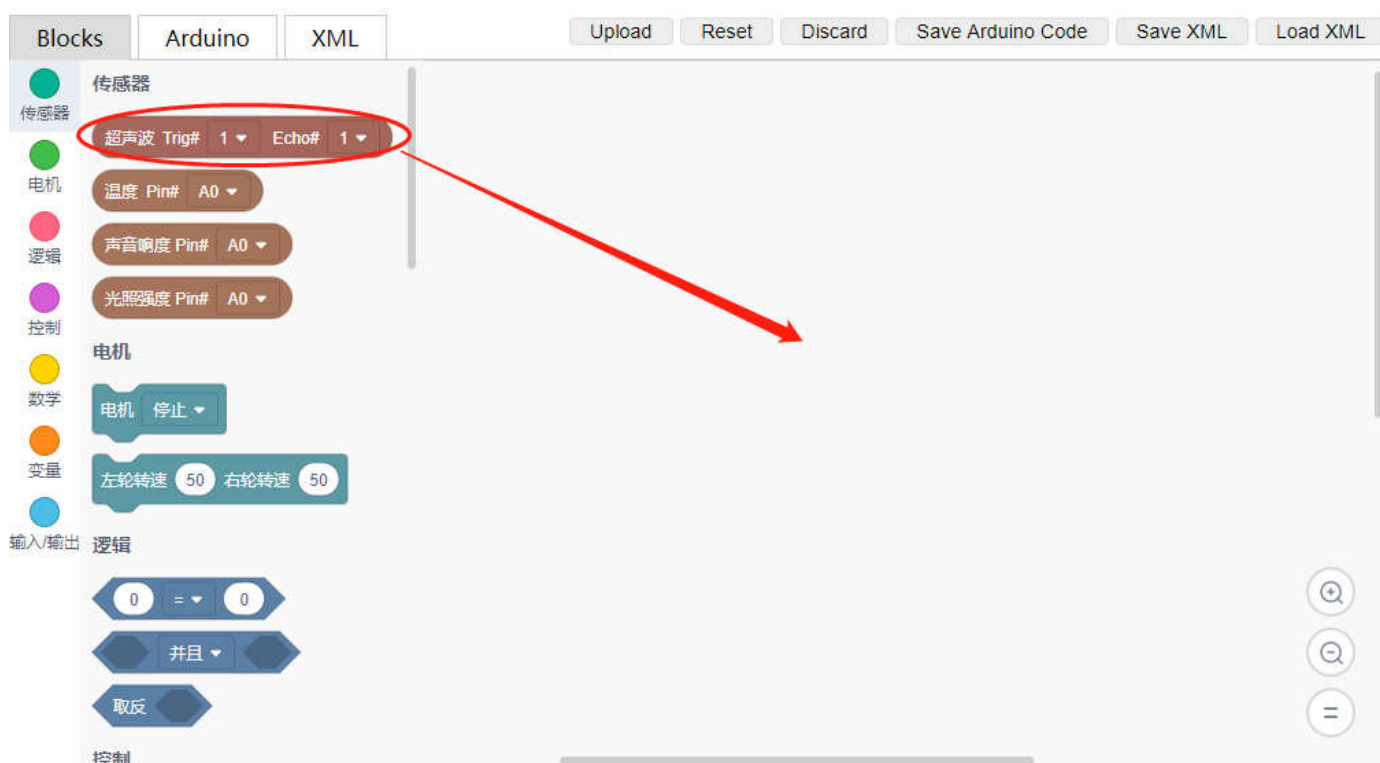
上传程序到小车

需要小车停止时，可以如上图所示的方法上传代码“电机停止”，也可以直接点击“Upload”按钮右边的“Reset”按钮。

2. 读取传感器数据

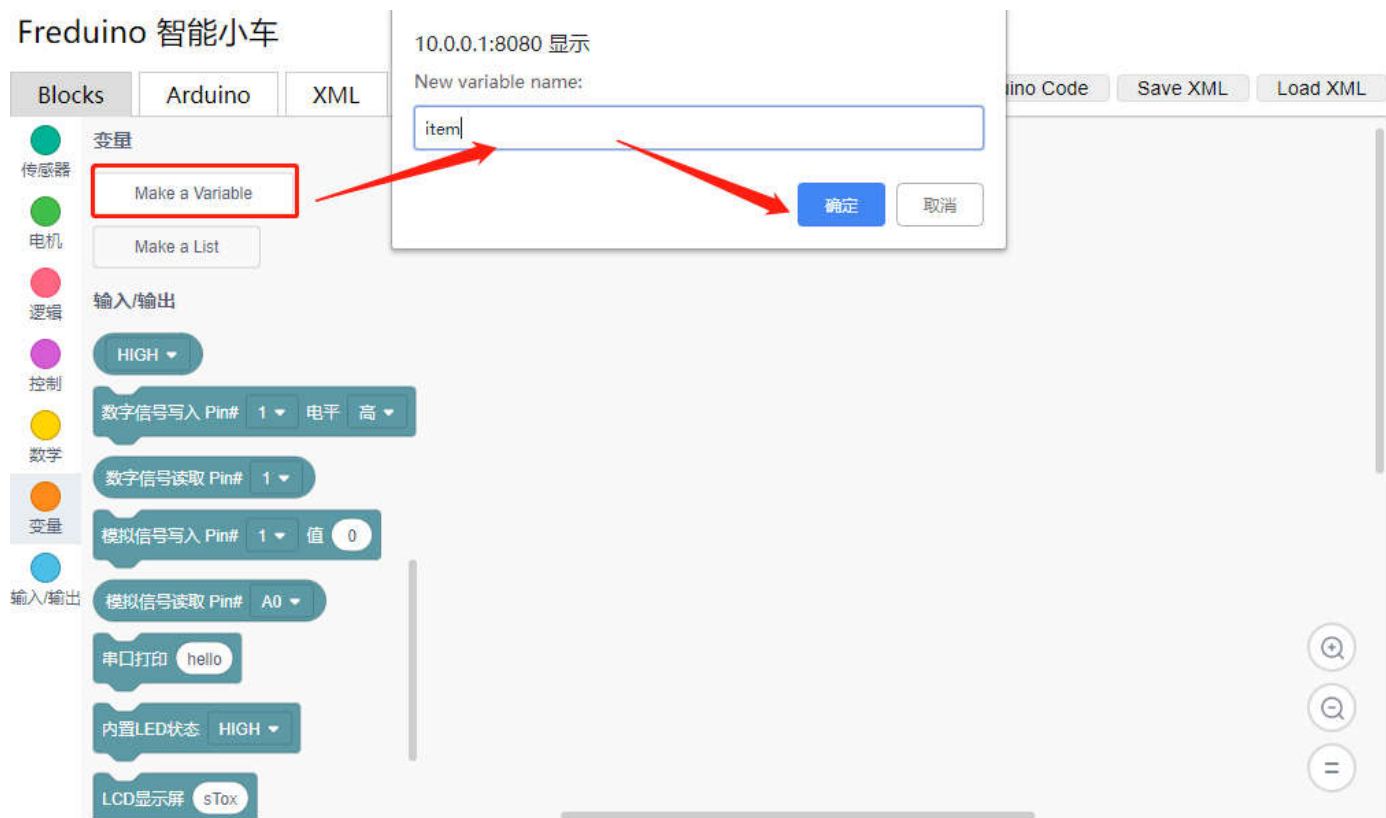
方法：为了让小车拥有对环境变化作出响应的能力，我们可以让小车通过超声波传感器探测前方或左右侧是否有障碍物，并对探测到的不同结果作出响应。 **首先，我们将代码块工具箱传感器一栏的超声波传感器拖入编辑区内。**

Freduino 智能小车



超声波传感器

超声波传感器这一代码块会返回超声波传感器探测到距离前方障碍物的距离，为了对这个距离进行进一步的判断，可以直接使用这一模块中的数据，也可以用一个变量来保存所获得的距离数值。在代码块工具箱变量一栏中点击“Make a Variable”，然后输入变量名称，点击确定，即可创建一个变量。这里创建了一个新变量item。



创建变量

之后，在变量一栏中，出现了三个新的选项“item”、“set item to”和“change item by”，从中将“set item to ”拖入编程区，将超声波传感器的代码块拖入标有数字零的圈中，即可将超声波传感器传回的数据保存到变量item中，需要使用超声波传感器中的数据的时候，可以直接使用变量“item”。

Freduino 智能小车

The screenshot shows the Freduino IDE interface with the 'Blocks' tab selected. The left sidebar contains a category list: 变量 (Variables), 传感器 (Sensors), 电机 (Motors), 逻辑 (Logic), 控制 (Control), 数学 (Math), 数字 (Numbers), 输入/输出 (Input/Output), and 变量 (Variables). The main workspace contains the following blocks:

- Make a Variable** block.
- item** block (selected).
- set item to 0** block.
- change item by 1** block.
- set item to 超声波 Trig# 1 Echo# 1** block.
- Make a List** block.
- HIGH** block.
- 数字信号写入 Pin# 1 电平 高** block.
- 数字信号读取 Pin# 1** block.
- 模拟信号写入 Pin# 1 值 0** block.
- 模拟信号读取 Pin# A0** block.

The 'item' variable is highlighted in the left sidebar, and the 'set item to' block is highlighted in the main workspace.

设置变量

为了对不同变量进行识别，可以对item 变量的变量名进行修改，**点击item 按钮，选择“Rename variable”选项，在弹出的对话框中输入相应的新变量名，即可对变量名进行修改：**

Freduino 智能小车

The screenshot shows the Freduino IDE interface with the 'Blocks' tab selected. The left sidebar contains a category list: 变量 (Variables), 传感器 (Sensors), 电机 (Motors), 逻辑 (Logic), 控制 (Control), 数学 (Math), 数字 (Numbers), 输入/输出 (Input/Output), and 变量 (Variables). The main workspace contains the following blocks:

- Make a Variable** block.
- item** block (selected).
- set item to 0** block.
- change item by 1** block.
- set item to 超声波 Trig# 1 Echo# 1** block.
- Make a List** block.
- HIGH** block.
- 数字信号写入 Pin# 1 电平 高** block.
- 数字信号读取 Pin# 1** block.
- 模拟信号写入 Pin# 1 值 0** block.
- 模拟信号读取 Pin# A0** block.

The 'item' variable is highlighted in the left sidebar, and the 'set item to' block is highlighted in the main workspace. A dialog box is open over the 'item' block, showing the following options:

- ☒ **item**
- Rename variable**
- Delete the "item" variable**

重命名变量

注意：传感器的信息主要通过Echo引脚和Trig引脚传输信息，此处的Echo和Trig引脚必须选择正确！

3. 读取传感器数据方法

在左侧代码块区域还有“逻辑”、“控制”、“数学”以及“输入/输出”板块。其中，“逻辑”板块主要用于逻辑判断，如大小判断、与或非等，可以在编程中用于条件语句。“控制”板块中有重复执行和“if”语句，以及在小车编程中非常重要的“延时”控制语句。“输入/输出”板块的内容稍有些复杂，但功能强大。主要用于直接读取各引脚传来的数据。

