

---

# AppInventor 蓝牙小车控制程序 BlueCar 设计报告

## 一. 设计简介

在 2012 年 8 月 16 日小米 2 的发布会上,每个参会者都收到了一辆小米 7 月底发布的“跨界”产品:可用 iPhone 或者 Android 手机遥控的 Phone 飞车,作为智能手机功能延伸的另一个“新领域”,这个产品让我有了浓烈的兴趣和动手制作的冲动,于是我制作了一个可以用蓝牙控制的小车,然后让 Android 手机通过蓝牙来控制小车,于是我使用 AppInventor 设计了一个蓝牙小车控制程序,可以用这个程序来控制小车的移动方向和移动速度。

## 二. 设计方案

### (一) 预计成果

小车控制程序最终由三个模块组成,这三个模块分别对应一个控制模式,按键模式、传感器模式和蓝牙调试模式,在 Android 手机与小车的蓝牙配对之后,进入程序,选择已经配对的小车进行连接,然后选择不同的模式来控制小车的移动,按键模式可以通过按下相应的键来控制小车移动,传感器模式可以通过手机的姿态传感器,根据手机处于不同的方位来控制小车移动,而蓝牙调试模式则是通过向小车发送不同的指令来控制小车。

### (二) 方案概述

#### 1) 手机与蓝牙小车配对并连接

在使用本程序前要先打开手机的蓝牙,然后搜寻周围的蓝牙设备,最后选择小车的蓝牙设备进行配对,进入程序后,程序通过 ListPicker 控件列出所有和手机配对过的设备,然后选择小车的蓝牙设备,将该设备的 MAC 地址储存到全局变量 DeviceMAC 中,然后调用 BluetoothClient 的 Connect 函数就可以连接上小车的蓝牙。

#### 2) 选择模式并控制

1. 按键模式:在手机和小车进行蓝牙配对并连接之后选择按键模式,按键模式一共设置有 7 个按键,分别为前进、后退、左转、右转、加速、减速和停止,按下不同的按键之后,手机会通过蓝牙向小车发送不同的指令,小车收到指令之后会做出不同的动作,小车的初始速度设置为 40,最大速度为 100,可以通过加减速键改变速度,并且速度会实时显示在屏幕上,若在没有连接设备时或没有选择按键模式时按下了按键会有相应的提醒。
2. 传感器模式:在手机和小车进行蓝牙配对并连接之后选择传感器模式,手机内的姿态传感会被启动,通过判断手机在四个方向上偏转的角度向小车发送不同的指令来控制小车的运动方向,同时姿态传感器翻滚角和俯仰角的角度值会实时显示在屏幕上,此时也可以通过加减速键来调节小车的速度。
3. 蓝牙调试模式:在手机和小车进行蓝牙配对并连接之后选择蓝牙调试,这时会打开蓝牙调试的界面,这时手机会收到小车发来的信息并显示在屏幕上,也可以通过文本框向小车发送不同的指令,发送的指令也会显示在屏幕上,在调试模式下可以检测小车和手机的通信是否完整,从而发现程序的问题。

### (三) 主要问题和解决方法

#### 1. 蓝牙的连接问题。

建立一个全局变量 DeviceMAC,通过一个 ListPicker 控件让用户选择已经配对的设备,将设备的 MAC 地址保存在 DeviceMAC 这个全局变量里,并把 MAC 地址存入 TinyDB,如果第一次使用程序时已经连接过小车,那么第二次就可以直接从 TinyDB 中读取设备的 MAC 地址,无需选择设备,直接进行连接。

#### 2. 切换界面蓝牙需要重新连接问题。

由于 AppInventor 是将每一个 screen 作为一个独立的 App 来管理的,因此如果在一个 screen

连接上蓝牙，切换到另一个 screen 蓝牙就需要重新连接，而不同的 screen 之间能够共同使用的是只有 TinyDB，因此可以将设备的 MAC 地址存储到 TinyDB 中，然后在打开一个 screen 的初始化中，将这个 MAC 地址读取然后连接上，这样就避免了在切换 screen 之后需要重新连接蓝牙的问题。

3. 文本超过 TextBox 的高度，无法显示超出部分文本的问题。

由于目前的 TextBox 没有 scroll 属性，所以当文本的高度超过了 TextBox 的高度时，就无法显示超出部分的文本，由于在蓝牙调试模式中需要接受到大量来自小车的信息，所以文本框无法显示接受的信息，我采用了一个折中的办法，就是在 Clock 的 Timer 函数里面设置了 10 秒的倒计时，每隔 10 秒就会清空收到的和发送的数据，虽然不能实时看到接收到的信息，但是能够看到每 10 秒的数据变化。

### 三. 设计成果

#### 1) 程序图标



#### 2) 程序功能

1. 和已经配对蓝牙设备进行连接。
2. 通过按键控制小车的移动。
3. 通过传感器控制小车的移动。
4. 通过命令控制小车的移动。

#### 3) 程序设计

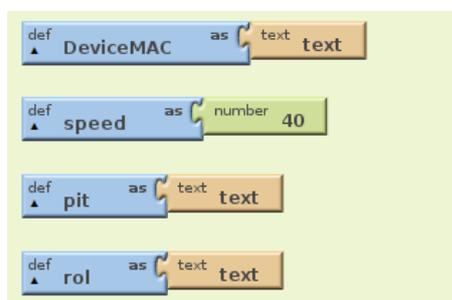
本程序一共有 3 个 Screen，截图如下：（帮助说明界面太过简单未截图）



#### 1. 主控制界面

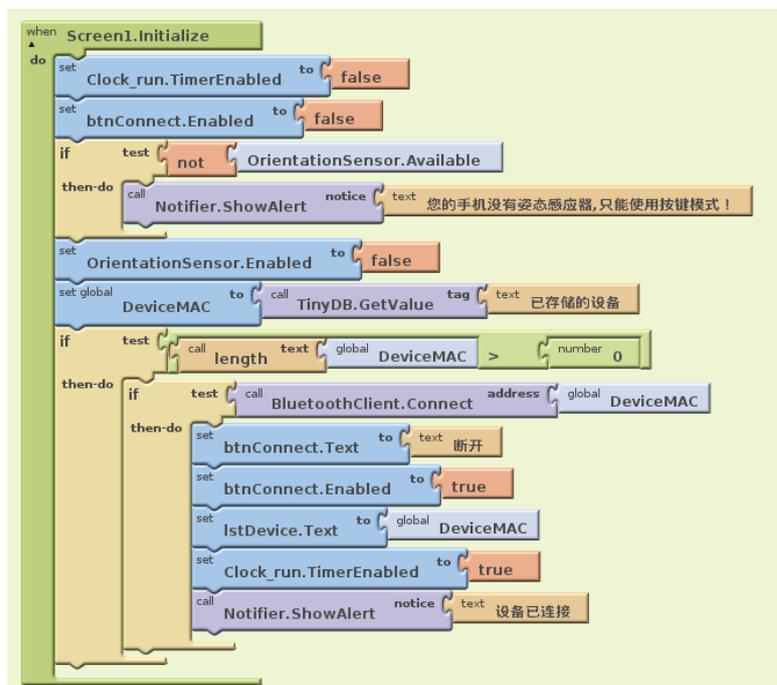


### A. 全局变量



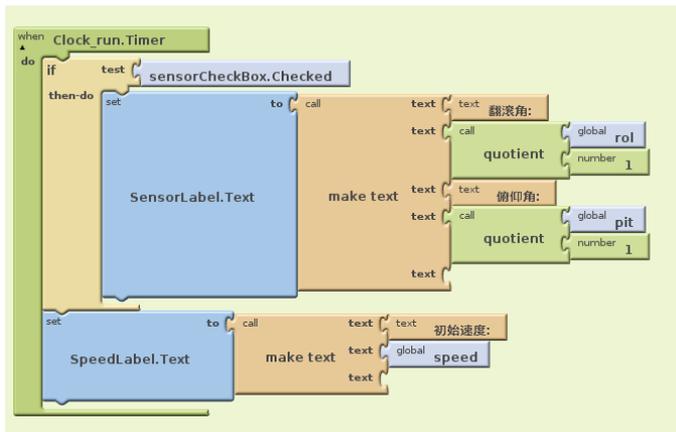
一共定义了 4 个全局变量，DeviceMAC 变量用来存储蓝牙设备的 MAC 地址，speed 变量用来存储小车的速度，pit 变量用来存储手机姿态传感器的俯仰角，rol 变量用来存储手机姿态传感器的俯仰角。

### B. 界面初始化



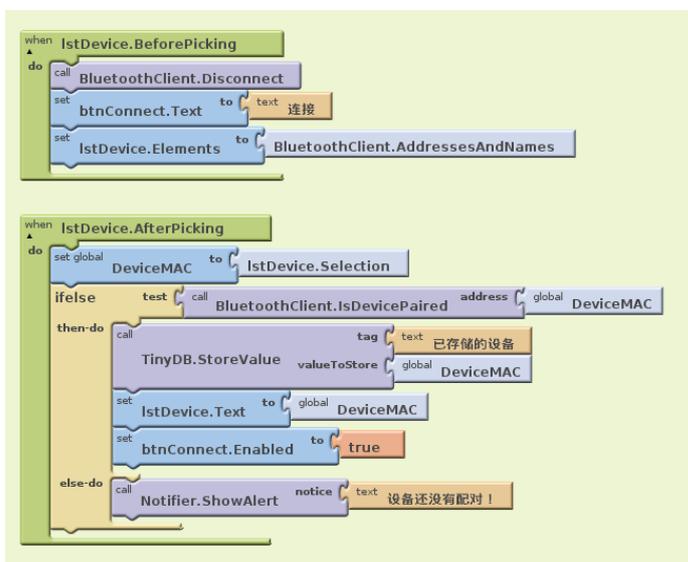
初始化时会检测手机是否支持姿态传感器然后给出提醒，如果曾使用程序连接过设备，程序会自动连接上连接过的设备。

### C. Timer 函数



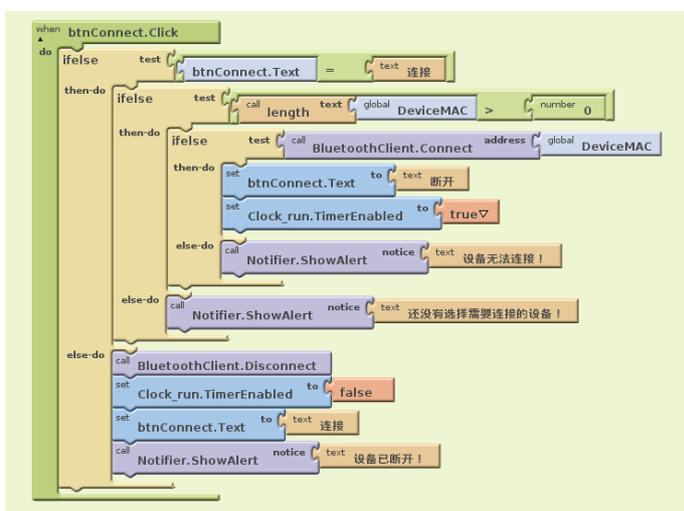
当 Clock 可用时，Timer 函数每秒会刷新传感器的角度值和小车速度。

#### D. 选择可连接的设备



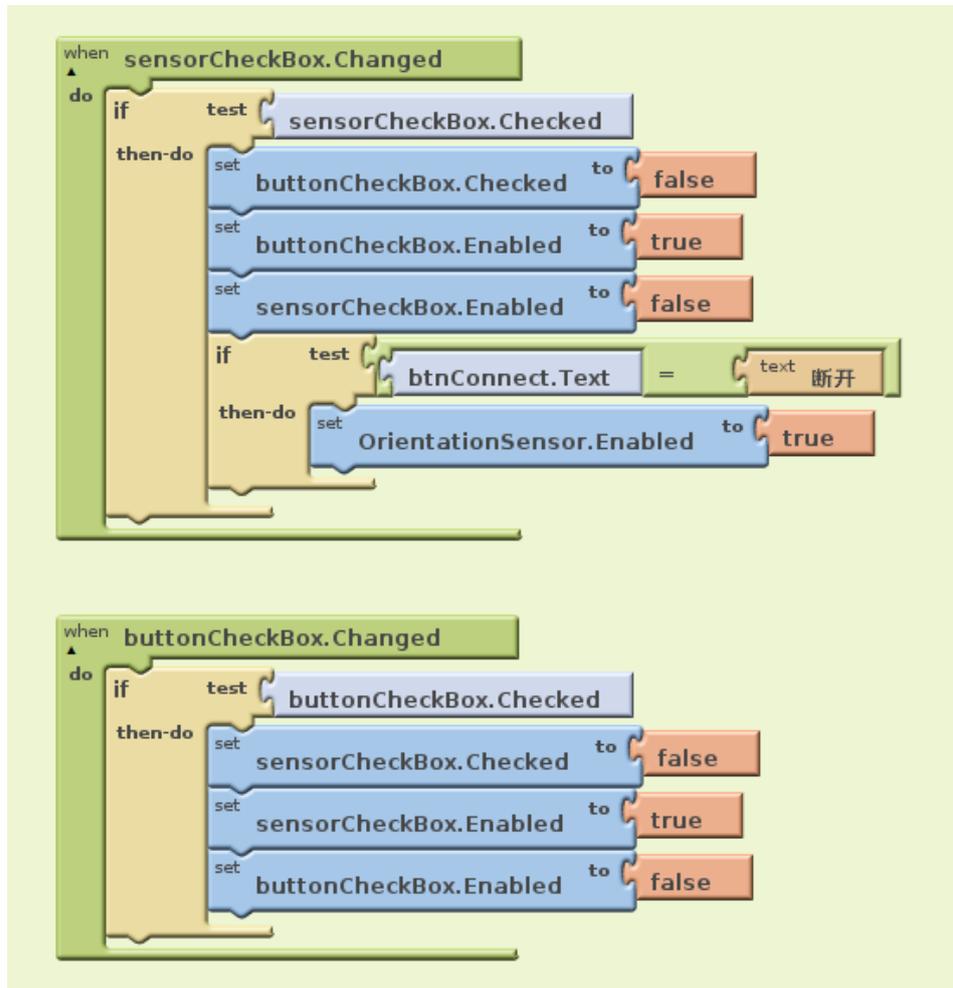
ListPicker 控件会列出所有的蓝牙设备的地址和名称，选定一个蓝牙设备后，其 MAC 地址会存入 TinyDB 中以便以后直接连接，同时把 MAC 地址存储到 DeviceMAC 这个全局变量中，供连接使用。

#### E. 连接蓝牙



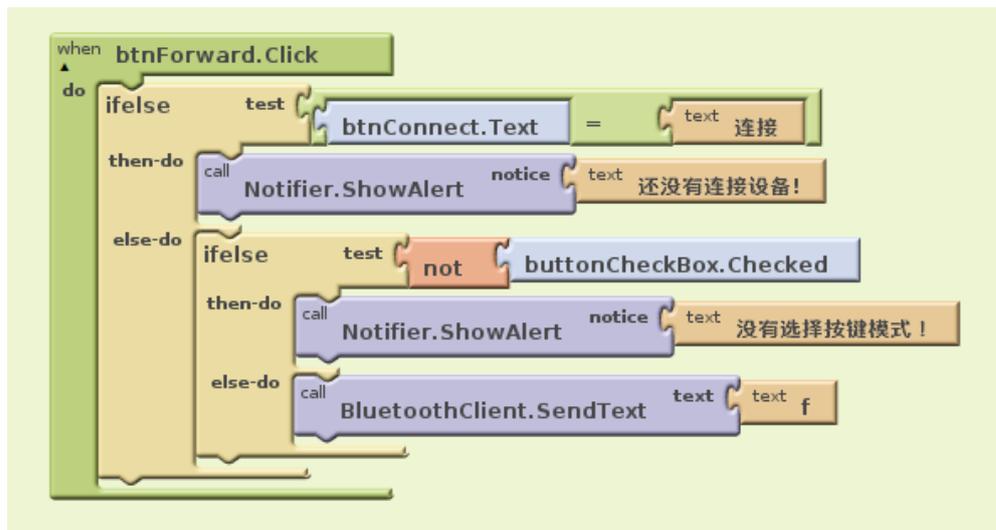
当选择设备之后连接按钮变成可用状态,按下之后手机会尝试连接存储在 DeviceMAC 中蓝牙设备地址,连上之后连接按钮变成断开,再按一次便会断开蓝牙。

#### F. 选择控制模式



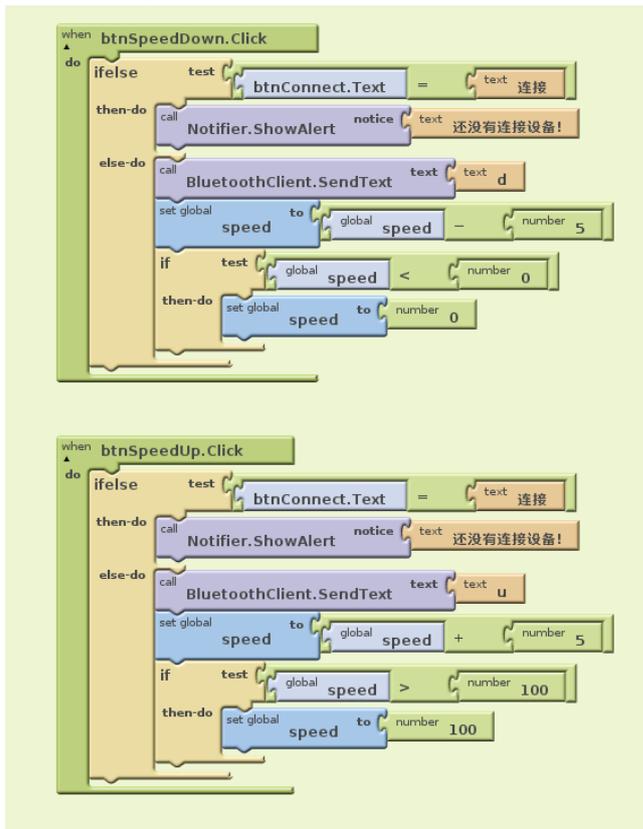
程序默认选择按键模式,选择传感器模式时按键模式会自动取消。

#### G. 方向控制按键



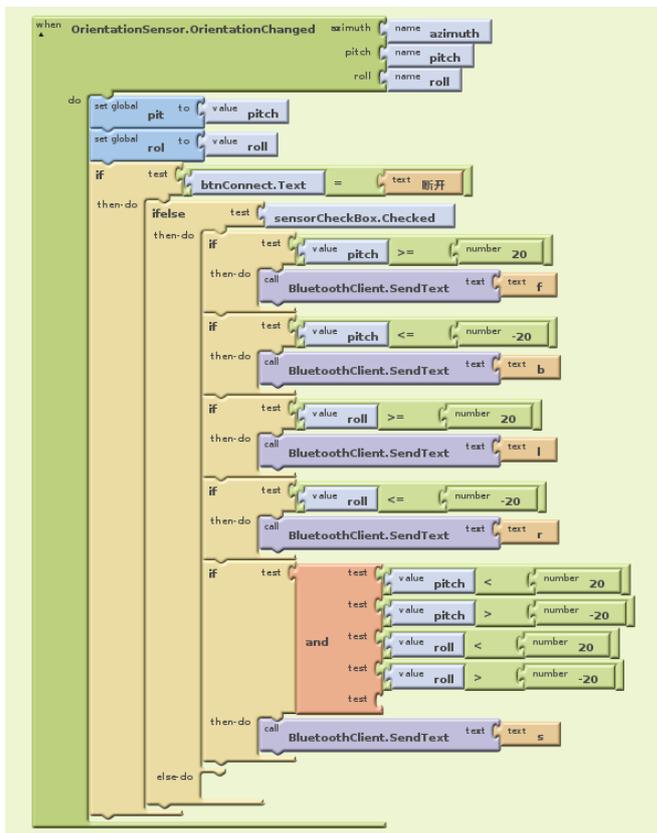
当按下对应的控制按键时会检查是否连接了设备以及是否选择了按键模式,若是,则向小车发送相应的指令。

## H. 加减速键



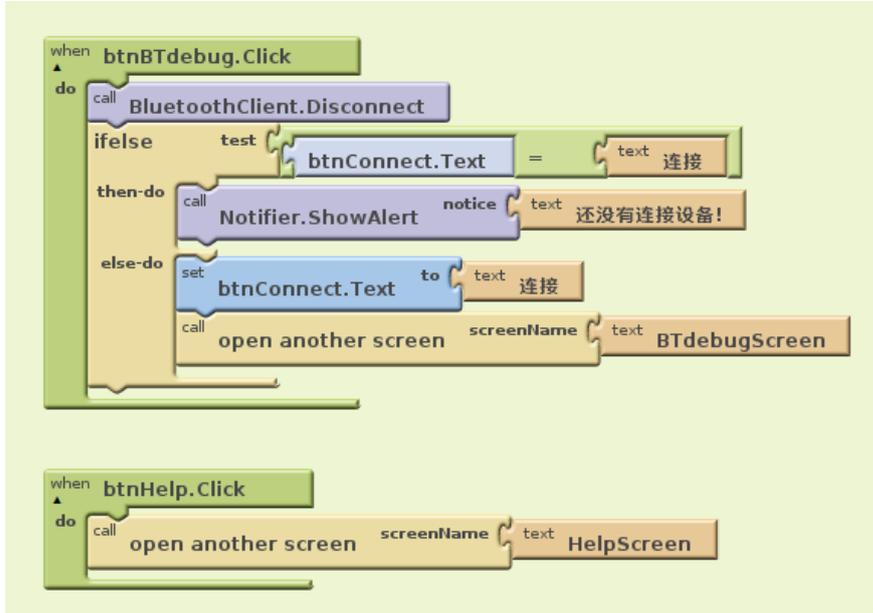
当按下加减速键之后会检查是否连接了设备，并且会改变全局变量 speed 的值，并向小车发送指令。

## I. 传感器控制



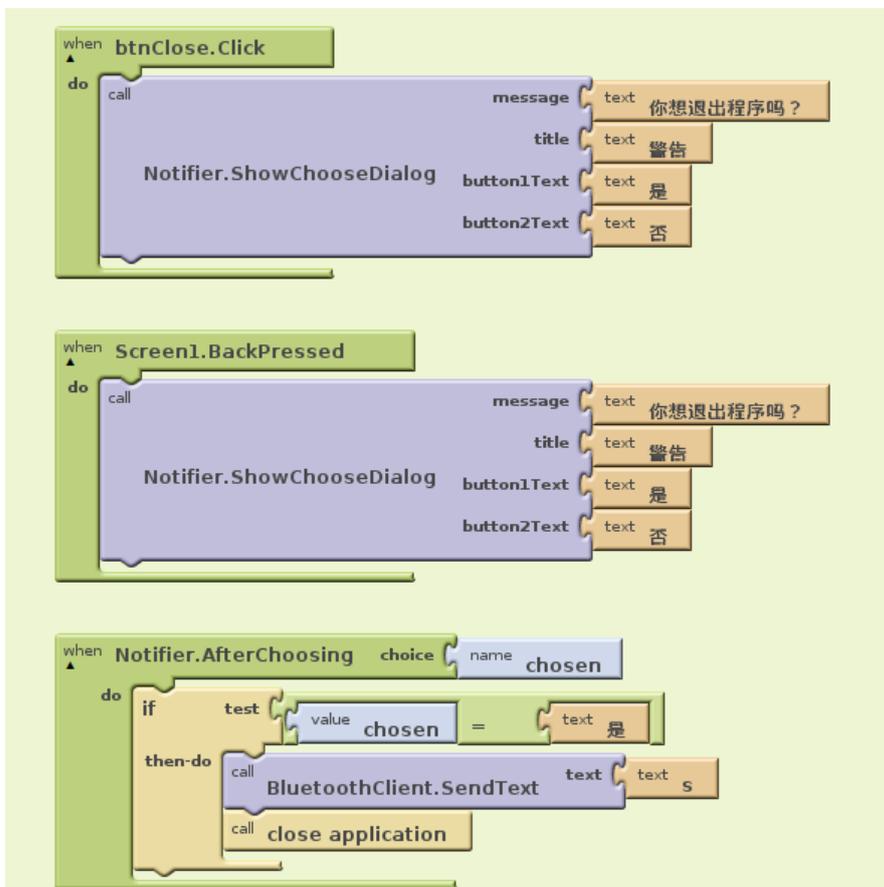
读取姿态传感器的 pitch 和 roll 值存储在全局变量 pit 和 rol 中，通过判断 pit 和 rol 的大小向小车发送不同的指令。

#### J. 打开蓝牙调试和帮助说明界面



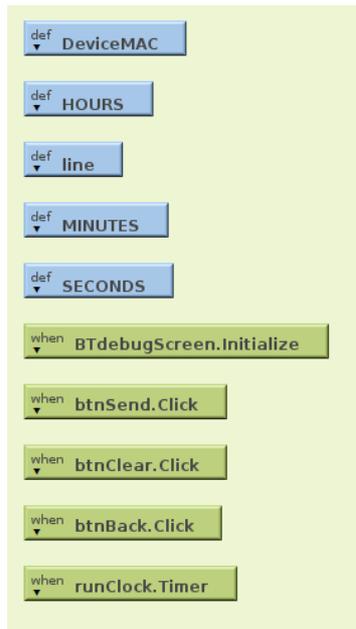
按下对应的按键之后会打开对应的界面，打开蓝牙调试节目前会检测是否连接到蓝牙设备。

#### K. 推出程序按键

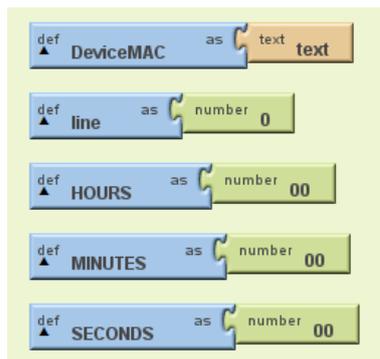


当按下了程序上的推出按键或者 Android 手机的返回按键时，会跳出提示是否确认退出软件，选择是之后会关闭程序。

## 2. 蓝牙调试界面

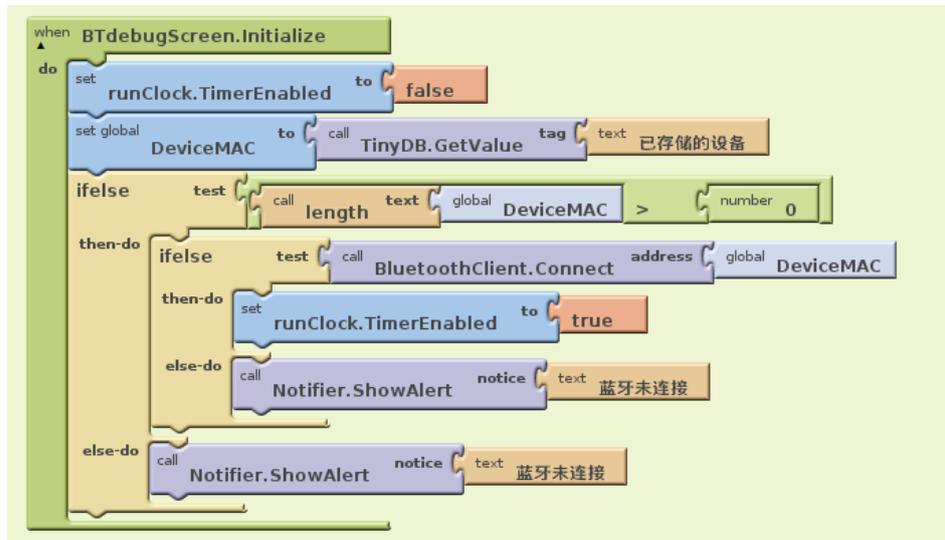


### A. 全局变量



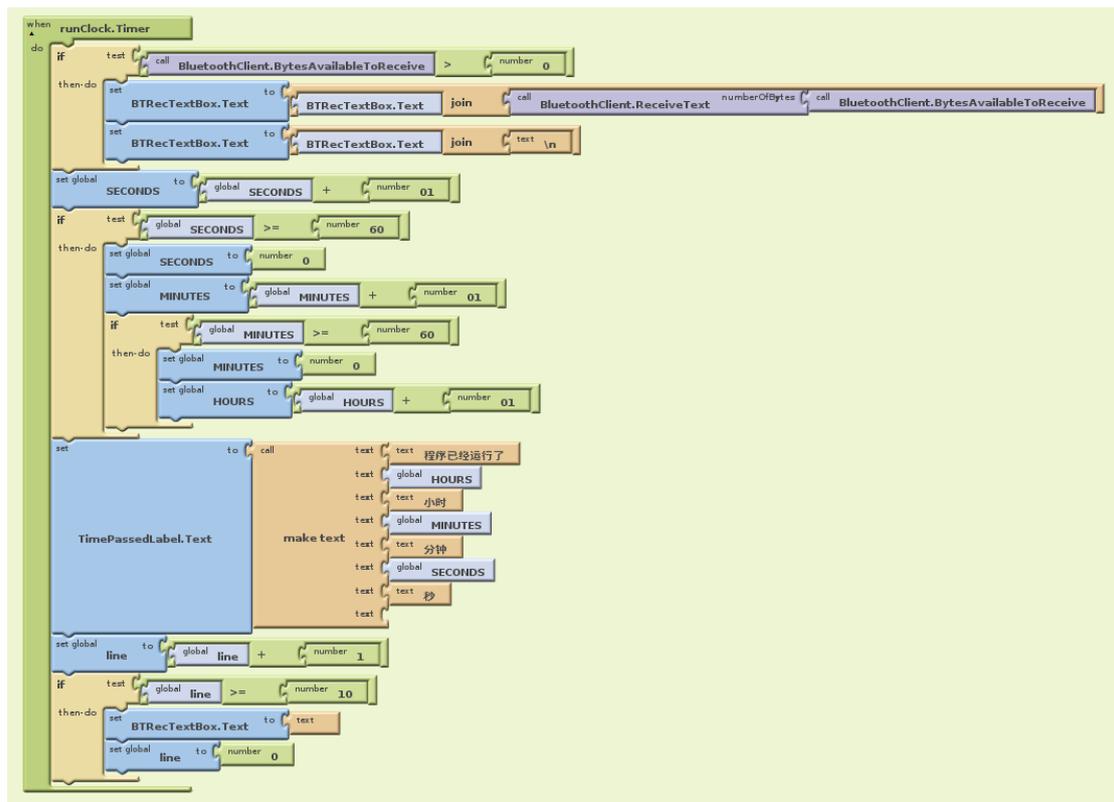
一共定义了 5 个全局变量，DeviceMAC 变量用来存储蓝牙设备的 MAC 地址，line 变量用来存储计数的值，HOURS、MINUTES 和 SECONDS 用来存储程序已经运行的时间。

### B. 界面初始化



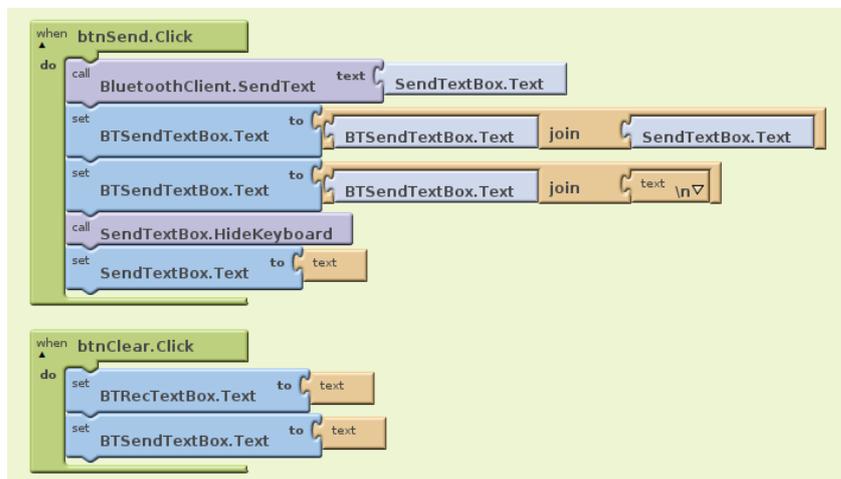
打开蓝牙调试界面时会自动连接上蓝牙设备。

### C. Timer 函数



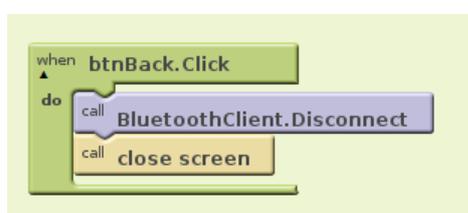
每秒钟都会刷新接受的数据，并且会计算程序已经运行的时间，每 10 秒会清空接收到的数据。

### D. 发送和清除按键



在文本输入框中输入指令后按下发送键之后会向小车发送输入的指令，按下清除键会清除所有已经发送及接收到的数据。

### E. 返回按键



按下返回键之后会关闭蓝牙调试界面，回到控制界面。

### 3. 帮助说明界面（简单，略）

#### 3) 程序运行

##### 1. 主控制界面



##### 2. 蓝牙调试界面



##### 3. 帮助说明界面



#### 四. 总结与思考

整个小车完工大概花了三个月，主要是利用课余时间来完成设计的，小车完成的时候恰好赶上移动编程的比赛，然后我就花了一个月时间来完成这个程序，基本上周末的时间都花在这个程序上，虽然我不是计算机的学生，也没有上过移动编程的课，但是我对编程非常感兴趣，所以经常关注计算机方面的知识，有一定的编程基础，AppInventor 的易用性也给了我巨大的帮助，这里我要感谢我的室友黄暖和，他给予了我很大的帮助，是他教我如何配置 AppInventor 的开发环境，有时一些问题也有向他请教。四个月的时间，基本上占去了一个学期，虽然这个作品不算完美，但是也算是有一定的技术价值，对我以往所学的知识也有了一定的实践。

虽然用 AppInventor 设计 Android 应用有它非常易用的一面，同时能够给我们带来拼图的乐趣，但是设计的应用还是有一定的局限性，比如无法适应不同大小的屏幕，无法使用高级的效果，运行起来比较缓慢，而且还有我发现的 TextBox 无法滚动的问题，如果要解决这个问题，需要我学习更多的 Android 编程知识，通过编程语言来实现。