

App Inventor + IoT: Wheeled robot control with LinkIt 7697(Wi-Fi)



(including Arduino
IDE setup)

Level: advanced

This tutorial will help you get started with App Inventor + IoT and control a two-wheeled robot through LinkIt 7697 (Arduino compatible) with buttons on your app!

- [source.ino](#) / [source .aia](#)

Note: This is an advanced project, we assume you have intermediate understanding of App Inventor, Arduino and Bluetooth. Please check the [MIT App Inventor IoT website](#) for help getting started.

Hardware

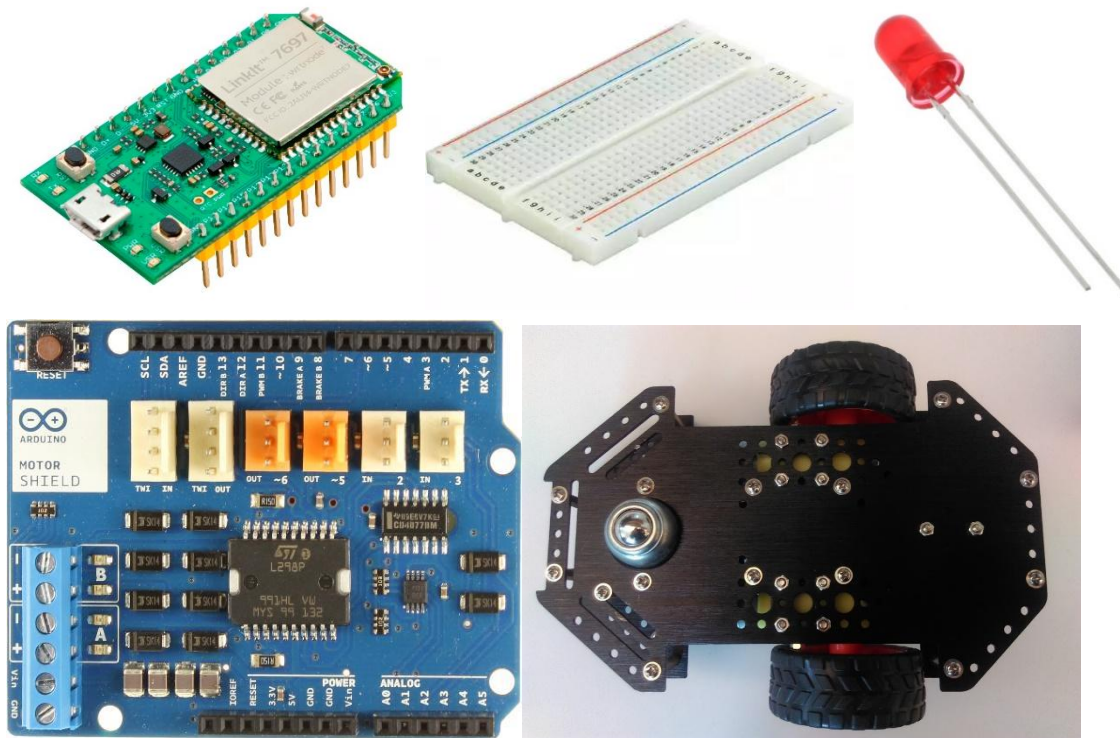
[LinkIt 7697](#) is an Arduino compatible dev board with Wi-Fi / BLE. You can use it like just like any other Arduino, and interface with App Inventor through its Wi-Fi communication.

In this project, we are going to drive a two-wheeled robot based on a LinkIt 7697. By four buttons on your app, you can move the robot forwards, backwards, left, and right, as well as stopping it.

LinkIt 7697 and our sample robot is like below:

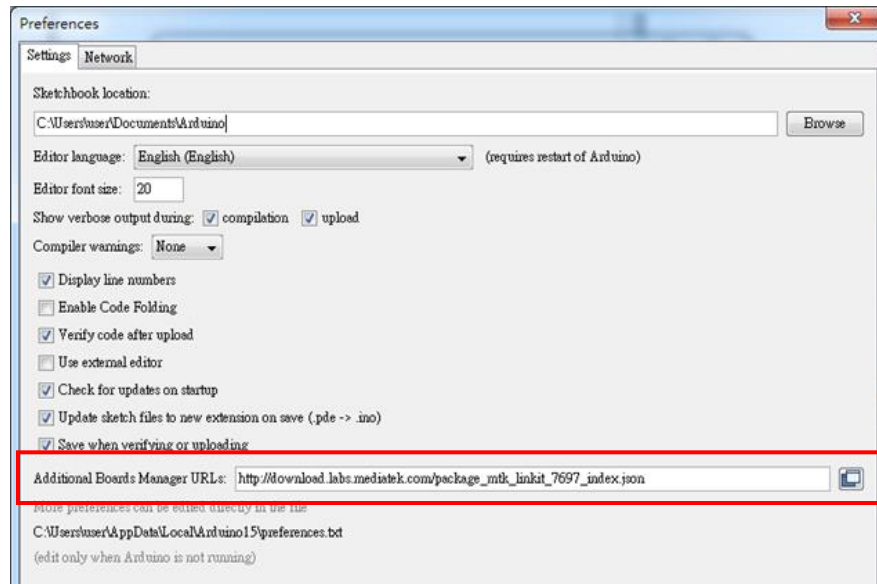
- LinkIt 7697 dev board, 1
- breadboard, 1
- LED, 2
- wires, several
- DC-motor extension shield, 1

- robot chassis (any version with 2 DC-motor and wheels)



Arduino IDE Setup

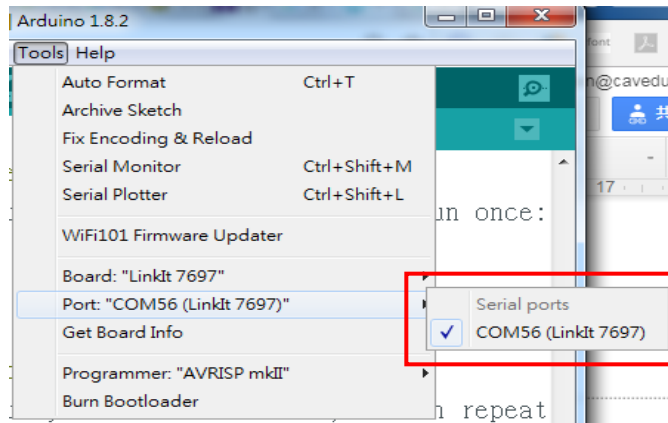
1. First, go to the [Arduino IDE 1.8.x](https://www.arduino.cc/en/software) version. Download the .zip file, unzip and click arduino.exe to open the IDE.
2. From **File**→ **Preference** menu, enter the link below in the Additional Boards Manager URLs field:
 - http://download.labs.mediatek.com/package_mtk_linkit_7697_index.json



3. Open **Tools/ Board/ Board Manager**, then search "7697" and install the latest version of 7697 SDK.



4. Download and install the [CP2102N driver](#) ([Windows](#) / [MAC/OSX](#)) , then check the COM port in your Device manager. Look for "**Silicon Labs CP210 USB to UART Bridge(COMXX)**", this is the COM port number of your LinkIt 7697.
5. For MAC users, it should be something like **"/dev/tty.usbserialXXX..."** and for Windows users, please check the picture below:

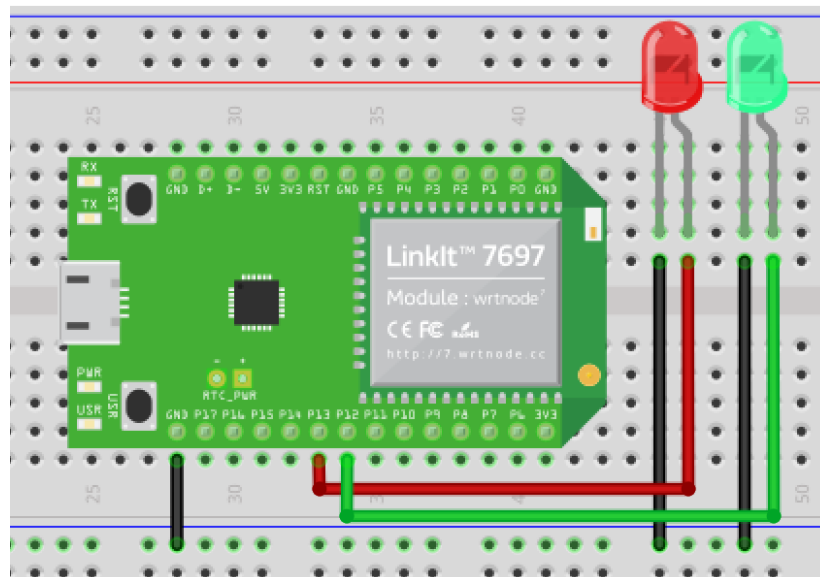


Hardware Assembly

Hardware of this project is a bit complicated, we separated it into three parts: LED headlights, motor drive circuit and chassis.

1. LED headlights

Our robot has two LED as headlights. Your circuit should look like the picture below, and please check the table for more details.

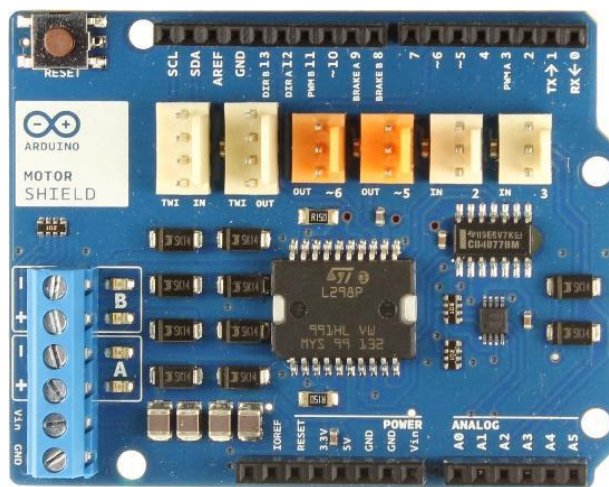


LinkIt 7697	LED
P13	LED1 + (left)
P12	LED2 + (right)

GND	LED1 - / LED2 -
-----	-----------------

2. Motor drive circuit

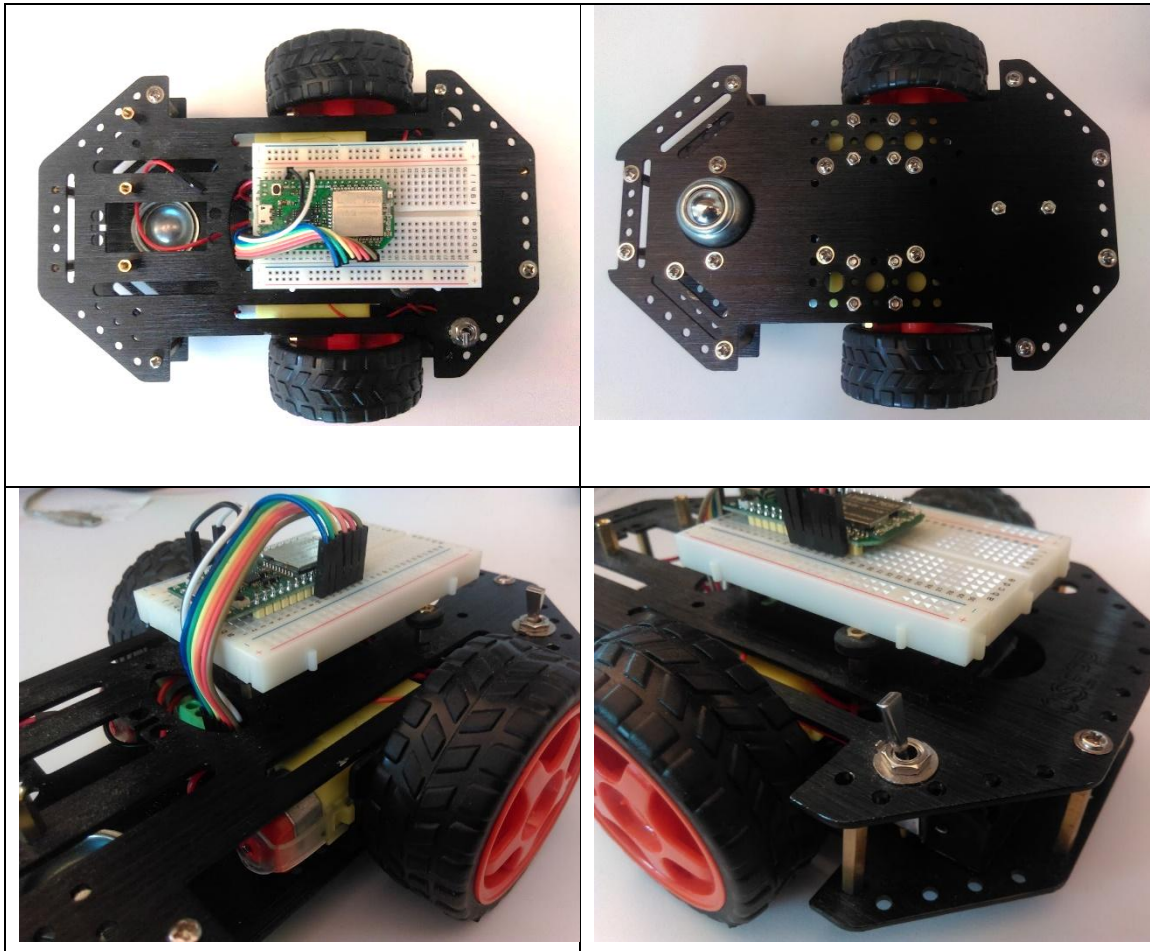
We recommend that you use a motor extension shield (like the [Arduino motor shield](#)) rather than a single H-bridge DC-motor control chip. Since motor shield has an integrated chip, you can save a lot of time with wiring. Additionally, the shield has protection for your board against surge. Any model of Arduino-compatible DC-motor shield should work fine with the LinkIt 7697.



[Arduino motor shield](#)

Your circuit should look like the picture below, and please check the table for more details. Please notice that you should power the motor shield and the microcontroller board with separate power, this is because the moment when the motor start rotating will need higher current and the microcontroller board may not be able to provide enough current and will reboot repeatedly.

Here are some photos of our sample robot:



App Inventor

The purpose of this project is to control the robot's actions and LEDs via Wi-Fi. The main idea is that App Inventor will open LinkIt 7697's URL with different commands, this will light up the LED in different colors.

- /F: robot go forward
- /B: robot go backward
- /R: robot turn right
- /L: robot turn left
- /S: robot stop
- /l: left LED on/off
- /r: right LED on/off

Now log in to your [MIT App Inventor account](#) and create a new project.

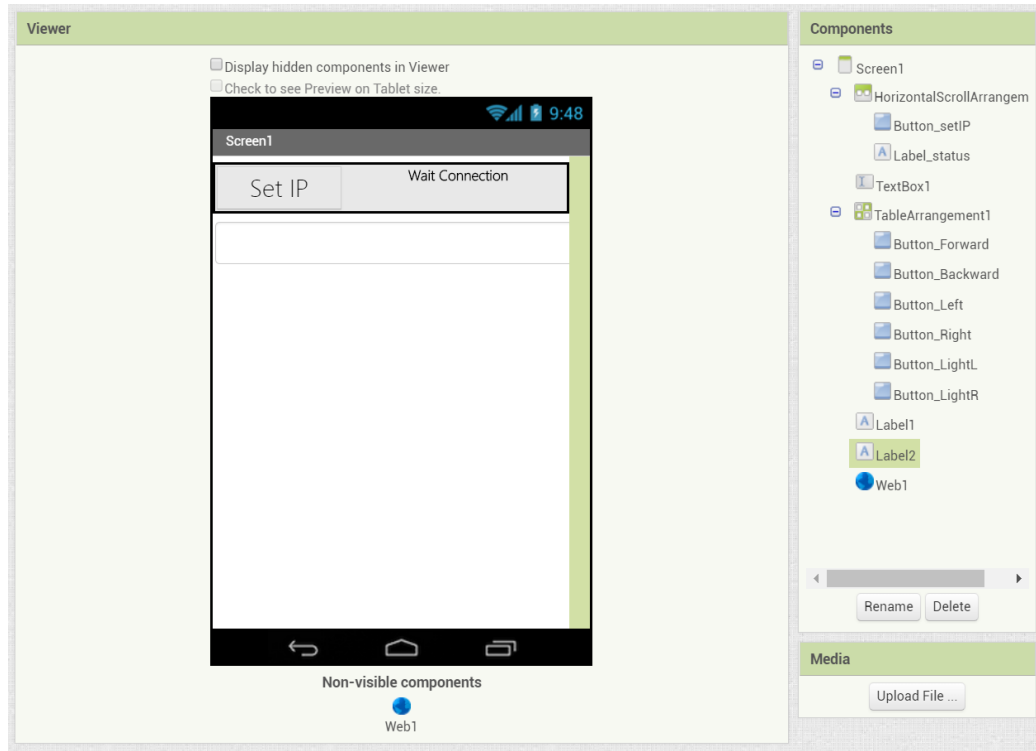
Designer

1. The most used components in this project are buttons (to trigger actions) and labels (to show relevant messages).
2. Add a **TableArrangement** component, set its width to **Fill parent**, height to **200** pixels, visible to **false**, Row to **2** and Column to **3**.
3. For robot control, add four **buttons** into previous **TableArrangement** component. Give each a width of 33% and a height of 100 pixels. And modify the Text to "↑", "↓", "←" and "→", representing different directions for the robot.
4. For LED control, add two **buttons** to the previous **TableArrangement** component. Give each a width of 33% and a height of 100 pixels. And modify the Text to "L" and "R", representing the left and right LEDs.
5. Add two labels, set their **Text** properties to "I / F / f" and "L / B / R" and visible to **false**. They are just reminders for each

button's action.

6. Add a **Web** component for sending text to LinkIt 7697 through Wi-Fi.

After some adjusting, your designer should look similar to this. It doesn't have to be exactly the same. Feel free to modify the component's background color, position and text size.



Blocks

Let's take a look at our blocks step by step:

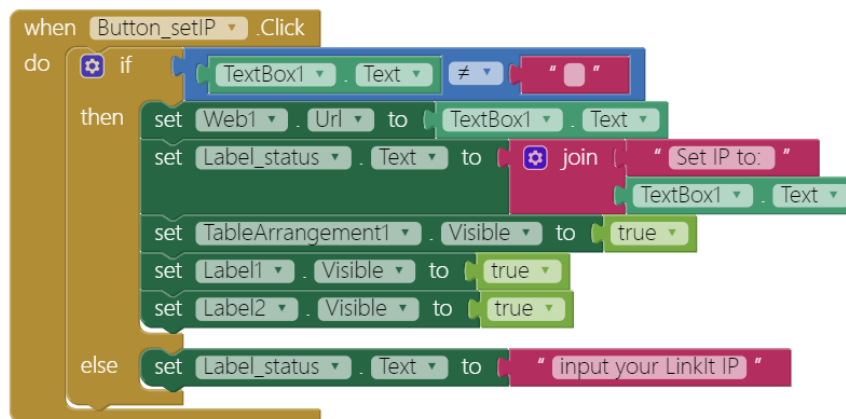
1. Variable for Bluetooth address

Please replace this with what you get from Arduino's Serial Monitor (Open in toolbar or press Ctrl + Shift + m). This is LinkIt 7697's IP address after connection to the Internet.

initialize global ip to "http://192.168.1.73"

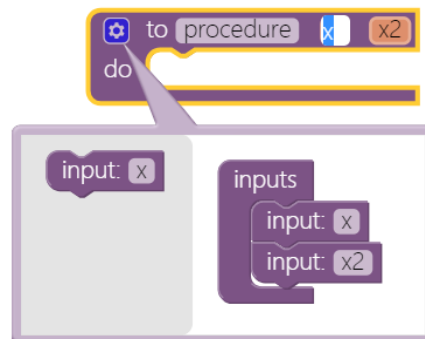
2. Setting the IP

Button_setIP is used to update the LinkIt 7697's IP. This is very convenient when you want to control more than one LinkIt board. When you click the button (**Button_setIP.Click** event), it will check whether the TextBox is empty. If so, then it will set the TextBox's content to the Web component's Url (LinkIt 7697's IP address). If not, it will show a message that tells the user to input LinkIt IP in the form of "XXX.XXX.XXX.XXX", such as "192.168.1.73".

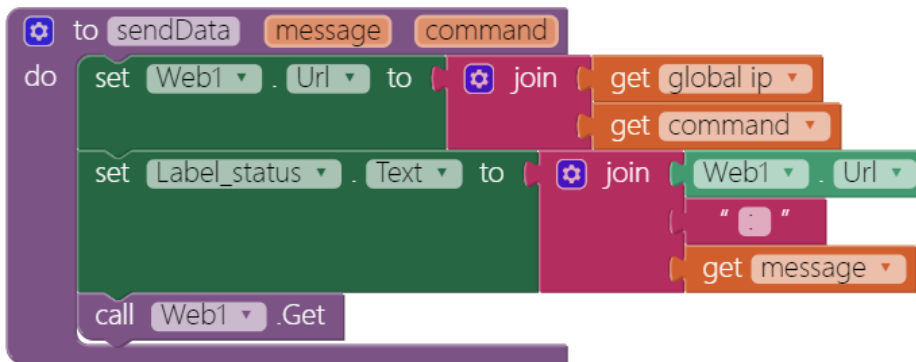


3. Procedure to update the Url of LinkIt 7697

Here we use a procedure to manage what we send and update the message when we click each button. Please add a procedure and click the blue gear to add two parameters. Rename this procedure to "**sendData**", and two parameters as "**message**" and "**command**." The **message** parameter will update the label and the **command** parameter will send different commands to LinkIt 7697.



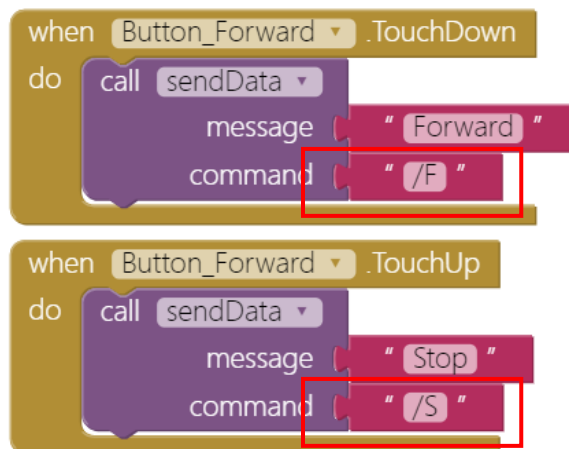
The **command** parameter of the **sendData** procedure will be combined with the IP variable into a complete URL, like "**http://192.168.1.73/R**". The last letter is a different command sent to LinkIt 7697. Finally, we use the Web component to get this URL, by sending out a specific text to LinkIt 7697.



4. Button to drive the robot forward and stop

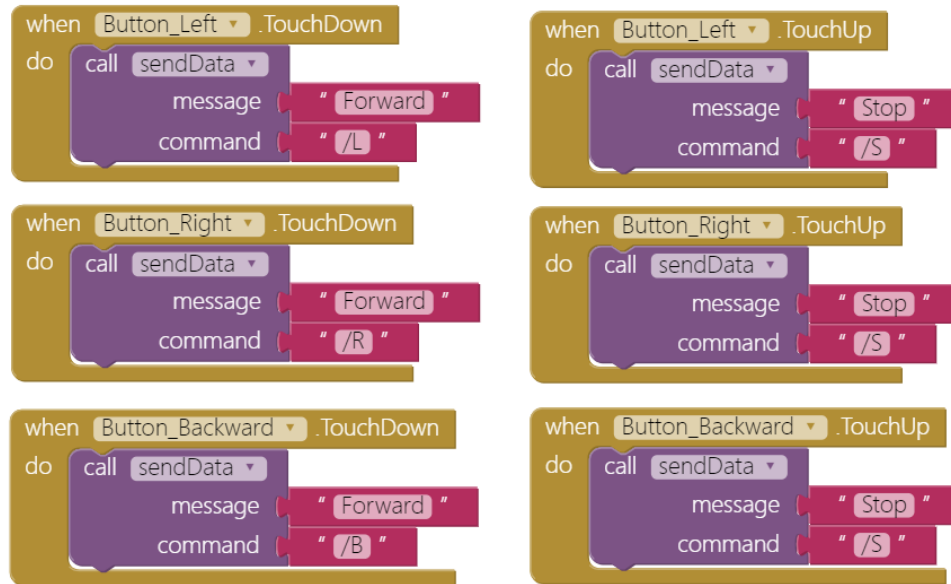
When each button is clicked, we are calling **sendData** procedures with different parameters. Take **Button_Forward** for example. the url is "<http://192.168.1.73/F>" to drive robot go forward. And it is "<http://192.168.1.73/S>" to stop the robot, we are going to do exactly the same situation as user finger is moved away from other three buttons.

Taking good advantage of the procedures will make your code simply more readable and manageable.



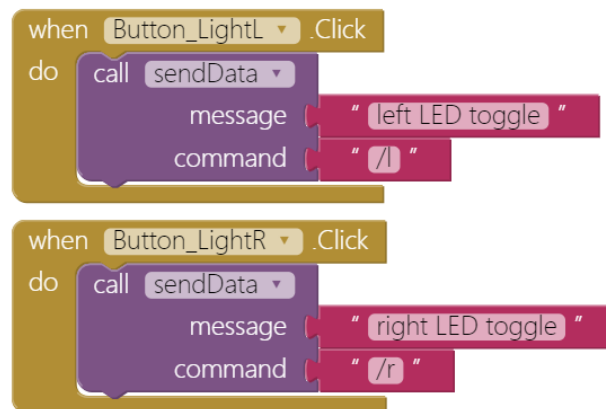
5. Other buttons for robot actions

When we click the other buttons, we also use **sendData** procedures with different texts. This happens both when they are pressed (**TouchDown** event), and with the same text **/S** when user finger is moved away.



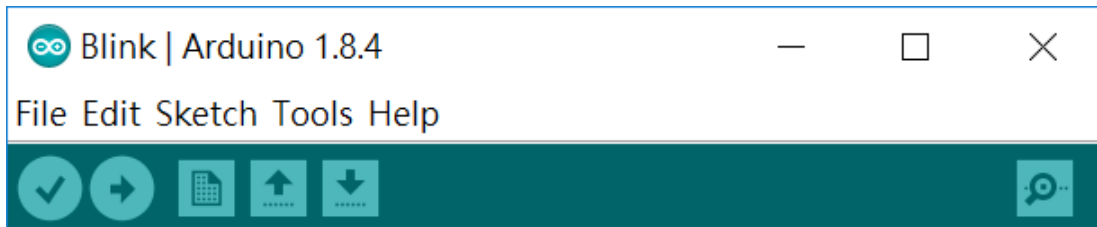
6. Buttons to control LED

For the two buttons to control LEDs, we use **sendData** procedure with text **/l** and **/r** to toggle the two LEDs, which means the first time we click the button will light up a LED, and next time click the same button will light off.



Arduino code

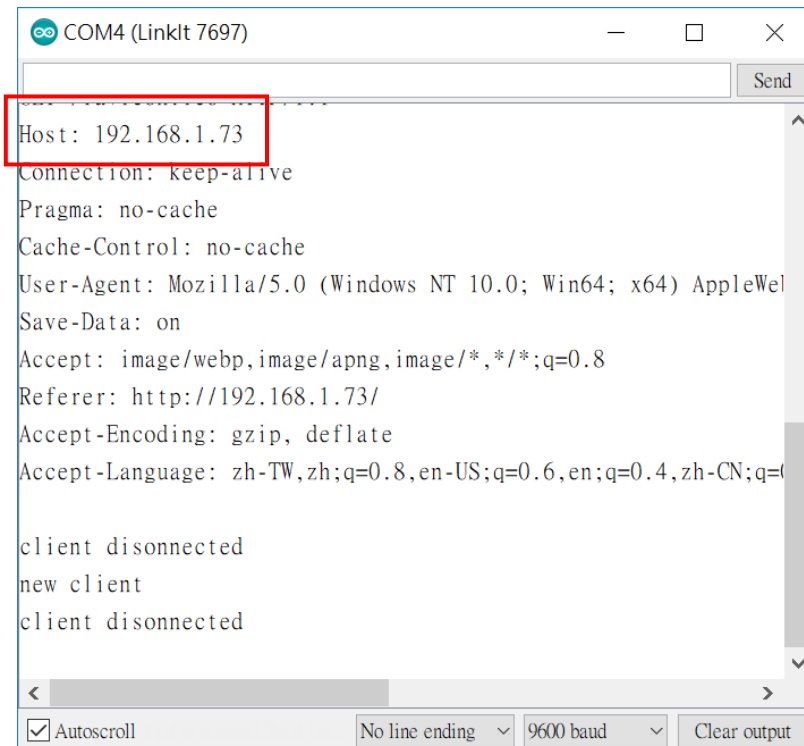
Please download the code [here](#) and upload it to your LinkIt 7697. Press the "**Upload**" right-arrow button, this will compile and upload the Arduino sketch in Arduino IDE to your LinkIt 7697. Please make sure you see the "**done uploading**" message in the console below.



Tips

Make sure your LinkIt 7697 is running correctly as a Webserver. Open Serial Monitor in Arduino IDE, check the IP address of your LinkIt 7697.

Modify App Inventor's **IP** variable value with this, then click each button to get your robot going!



Brainstorming

1. Add a **Speechrecognizer** component and use voice command to drive your robot like "**go forward**", "**go backward**", "**turn left**", "**turn right**" and "**stop**". (reference: [LED control tutorial](#))
2. Note: the robot's behavior depends on which platform you choose. Factors like chassis length, wheel, and motor torque are all going to affect how your robot move. Try playing with different kinds of robot chassis (tank is good!) and adjust the code, you will discover more!