

# App Inventor + IoT: Lego EV3 robot Sensor panel



(including robot assembly)

**Level: advanced**

## Introduction

This tutorial demonstrates how to use App Inventor as a monitor to show Lego EV3 robot's sensor status (color, ultrasonic and touch sensor).

Notice: You don't have to write a program for the EV3, just power it on and pair with your Android phone. App Inventor's EV3 components are talking with the Lego EV3 brick with a special protocol called [Lego EV3 Direct Command](#). These commands are actually byte arrays which can be executed directly by the Lego EV3 brick.

- [Source .aia](#)
- Reference: [Button Control of Lego EV3 Robot](#)

Lego EV3 Robot has 5 different kinds of sensors, including:

1. **Color Sensor**: detect surrounding light level (0-100) or color code of object ahead in numeric format(0-100).
2. **Ultrasonic Sensor**: detect distance of object ahead (measured in cm/inch).
3. **IR Sensor**: return IR message from IR controller (text and number).
4. **Touch Sensor**: detect whether it is pressed or not (logic).
5. **Gyroscopic Sensor**: detect motor's rotational speed and changes in its direction. It has two modes: **Angle mode** measures rotation(in degrees) and **Gyro mode** measures rotational speed (range from 0~440 degrees/second. Sample rate of 1 kHz).



Figure 1. Lego EV3 core components

Note: There is an encoder within EV3 motors to detect the motor speed and direction, we will introduce them in later topics.

Turn on your EV3 and pair it with your Android phone (default key: **1234**). You can check whether the robot's EV3 is on by the Bluetooth icon at the upper-left screen corner.

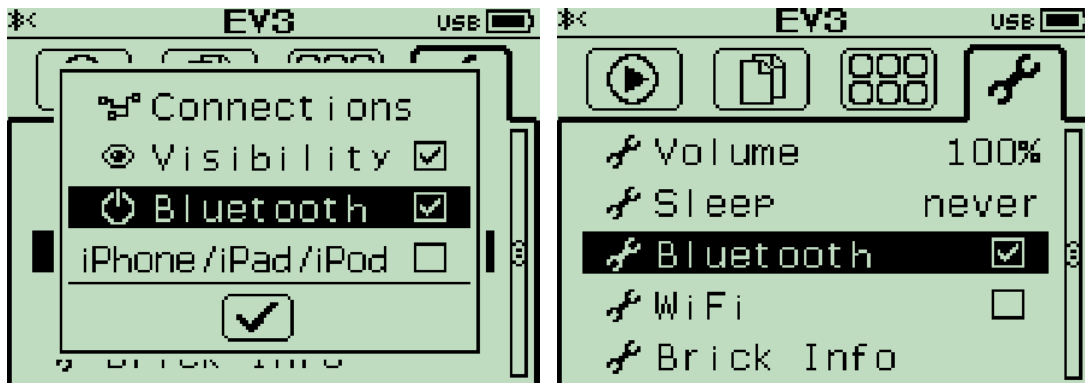


Figure 1c, 1d. Check EV3 Bluetooth is turned on.

## App Inventor

This app will update Lego EV3's sensor value (**color sensor, touch sensor and ultrasonic sensor**) value on screen. Now log into [MIT App Inventor site](https://www.mit.edu/~appinventor/) and create a new project.

## Designer

In the Designer page of App Inventor platform, please add these components and place them as shown in Figure 2. For details please refer to Table 1.

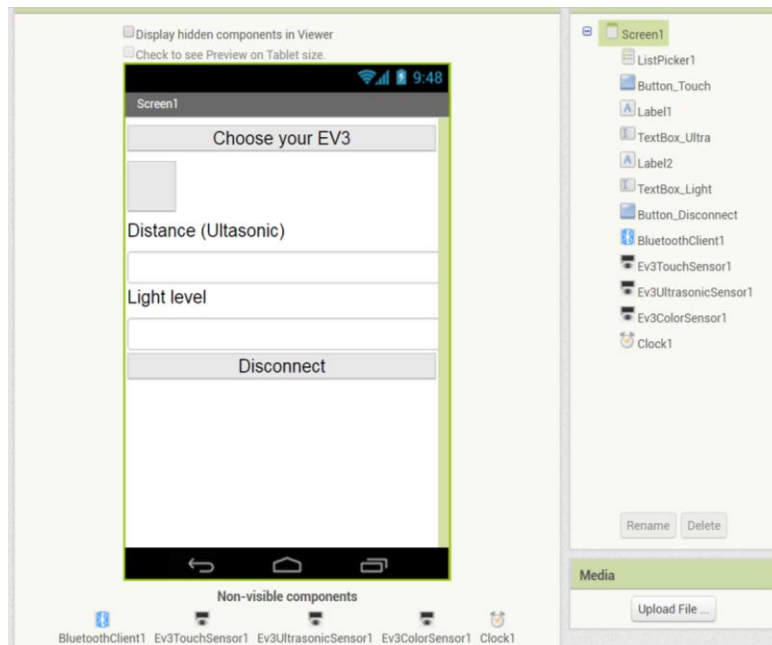


Figure 2. Designer

Table 1. Required components of this project

Name	Palette	Settings	Description
ListPicker1	User Interface	set Text to <b>"Choose your EV3"</b> (empty)	
Button_Touch	User Interface	Set the backgroundColor to <b>light gray</b>	
Label1	User Interface	Set Text to <b>"Distance (Ultrasonic) "</b>	

TextBox_Ultra	User Interface	Set Text to "" (empty)	Show distance by ultrasonic sensor.
Label2	User Interface	Set Text to " <b>Light level</b> "	
TextBox_Light	User Interface	Set Text to "" (empty)	Show light intensity by light sensor.
Button_Disconnect	User Interface	Set Text to " <b>Disconnect</b> "	Click to disconnect from EV3 brick.
Ev3TouchSensor1	LEGO® MINDSTORMS®	Set the BluetoothClient to <b>BluetoothClient1</b>	Get status of Ev3TouchSensor
Ev3UltrasonicSensor1	LEGO® MINDSTORMS®	Set the BluetoothClient to <b>BluetoothClient1</b>	Get status of Ev3UltrasonicSensor
Ev3ColorSensor1	LEGO® MINDSTORMS®	Set the BluetoothClient to <b>BluetoothClient1</b>	Get status of Ev3ColorSensor
BluetoothClient1	Connectivity	no need to change	Communication between AI2 and Lego EV3 brick
Clock1	Sensors	Set <b>timerinterval</b> to 200, which means updating 5 times per second.	Read EV3 sensor data periodically.

## Blocks

### STEP1 Preparation

First, we need to set the content of **ListPicker1(ListPicker1.Elements)** to **BluetoothClient1.AddressAndNames**, so we should choose the paired EV3 brick in this ListPicker.



In **ListPicker1.Elements.AfterPicking** event, we connect with what we chose (**ListPicker.Selection**). If we connect successfully, then it will activate the **Clock.Timer** and set the **Screen.Title** to "**Connected**".

```
when ListPicker1 .AfterPicking
do
  if call BluetoothClient1 .Connect address ListPicker1 . Selection
  then
    set Clock1 . TimerEnabled to true
    set Screen1 . Title to " Connected "
```

Figure 3. Use ListPicker to select paired Bluetooth device.

### STEP2 Show sensor value regularly

In **Clock1.Timer** event, we are polling sensors status 5 times per second, which means every 0.2 second. For ultrasonic and color sensor, we simply show the value directly on Textbox. And we have one special button to change the background color according to whether the touch sensor is pressed or not.

```
when Clock1 .Timer
do
  set TextBox_Ultra . Text to call Ev3UltrasonicSensor1 .GetDistance
  set TextBox_Light . Text to call Ev3ColorSensor1 .GetLightLevel
  if call Ev3TouchSensor1 .IsPressed
  then
    set Button_Touch . BackgroundColor to red
  else
    set Button_Touch . BackgroundColor to grey
```

Figure 4. Updating sensor value 5 times per second

### STEP3 Disconnect

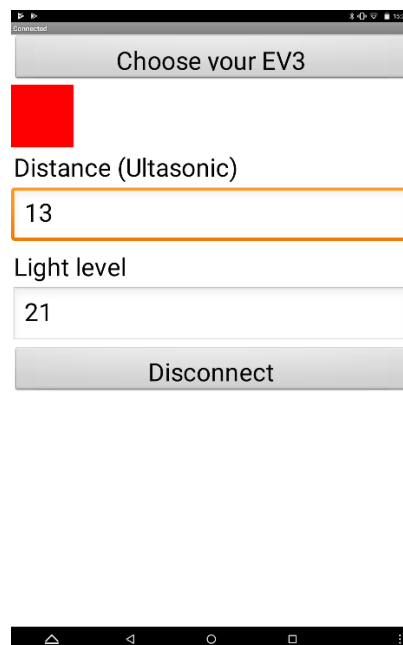
When **Button\_Disconnect** is pressed (**Button\_Disconnect.Click** event), we ask the **BluetoothClient** to disconnect from the EV3 brick, disable **Clock Timer** and set the relevant message on the **Screen title**.

```
when Button_Disconnect .Click
do
  call BluetoothClient1 .Disconnect
  set Clock1 . TimerEnabled to false
  set Screen1 . Title to " Disconnected "
```

Figure 5. Disconnect from Bluetooth device

## Tips

You must pair your EV3 robot to your Android device first, then press its power button and check that EV3's Bluetooth is switched on. Press the [**Choose your EV3**] ListPicker and choose the name of the paired EV3, you should see varying number on your screen. Try to press Lego EV3's touch sensor to check whether the button has been red or not~.



## Brainstorming

1. Try to play a short sound when the distance (using Ultrasonic sensor) is less than 15 cm.
2. Try to change a component's (like Canvas or button) background color according to what EV3 color sensor detected.