# MIT App Inventor Codi Bot: Wing control

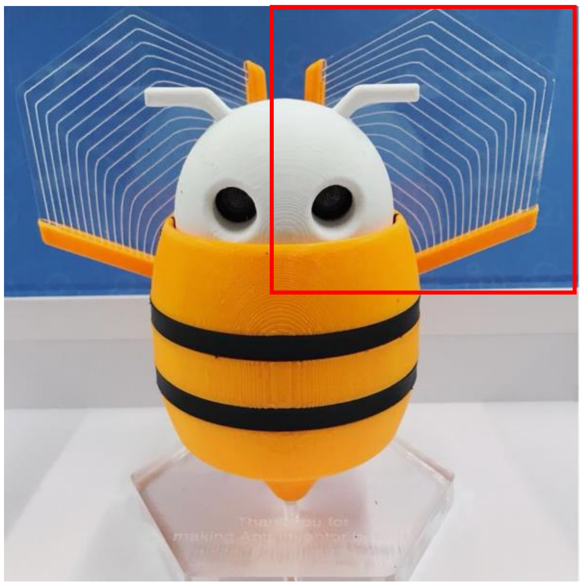
60 mins

**Level: advanced**

This tutorial will help you control one wing of Codi Bot by buttons and a slider, using App Inventor IoT. We have also provided a complete app so you can control both Codi Bot wings.

Note that there may be slight differences due to the tolerances of each servo, you may have to modify the parameters for servo position. Please first check everything is assembled correctly according to the **Codi Bot Standalone Demo tutorial**.

- **source .ino** / **source .aia**
- **complete .aia**



### Function description

This project will show you how to control one of Codi Bot wings, which is actually a small servo motor, with App Inventor through BLE communication. The components used in this tutorial are buttons and a slider.

## Hardware

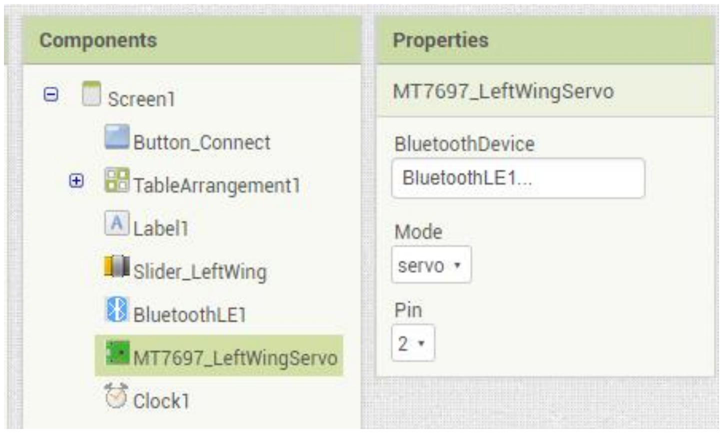
Please follow this **building guide** to assemble your Codi Bot.

## App Inventor

Log in to your App Inventor account and create a new project or directly import **this aia file**.

#### Designer

1. We need to import two extensions from this URL:
   - **Bluetooth low energy**: http://iot.appinventor.mit.edu/assets/resources/edu.mit.appinventor.ble.aix
   - **MT7697pin**: http://iot.appinventor.mit.edu/assets/resources/edu.mit.appinventor.iot.mt7697.aix
2. Add a **BluetoothLE** component to your project. We use this to send commands to Codi Bot through Bluetooth communication.
3. Add an **MT7697pin** component to your project. We use them to control a pin of LinkIt 7697, where the servomotor is also connected.
   - Rename one **MT7697pin** component to "**MT7697_LeftWingServo**". Set the BluetoothDevice property to **BluetoothLE1** (Step 2.), the **Mode** to **servo** and the **Pin** to **2**. This will help us connect the servo signal pin to LinkIt 7697 #2 pin.



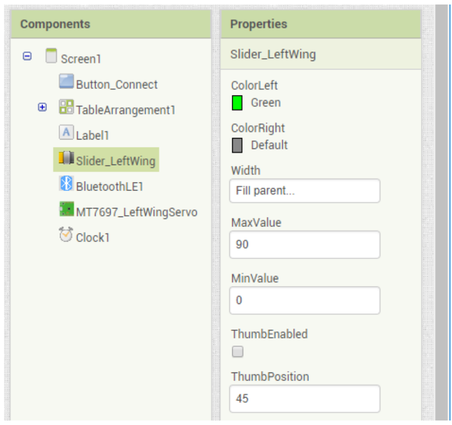
MT7697_LeftWingServo

BluetoothDevice: BluetoothLE1...

Mode: servo

Pin: 2

4. Add a button to establish the Bluetooth connection between your Android phone and LinkIt 7697. Rename it to "**Button_Connect**" and set Text to "**Connect**".
5. Add another four buttons for corresponding wing action. Rename them as "**Button_WingForward**", "**Button_WingBackward**", "**Button_wingFlap**" and "**Button_Home**". Set FontSize to **20**, Width to 50 percent and Text to "**Forward**", "**Backward**", "**Flaping**" and "**Home**" accordingly.
6. Add a TableArrangement component, set the Visible to **false** and both Columns and Rows to **2.** Then put buttons of Step 5 into it.
7. Add a label to show message. Set its Text to "**Left Wing (0 Forward- 90 backward)** ".
8. Add a slider to control the servo motor position. Rename it as "**Slider_LeftWing**". Set its **MaxValue** to **90**, **MinValue** to **0** and **ThumbPosition** to **45**. We do this because the servo can only move up to 90 degrees.


Components

- Screen1
  - Button_Connect
  - TableArrangement1
  - Label1
  - Slider_LeftWing
  - BluetoothLE1
  - MT7697_LeftWingServo
  - Clock1

Properties

Slider_LeftWing

ColorLeft
🟩 Green

ColorRight
■ Default

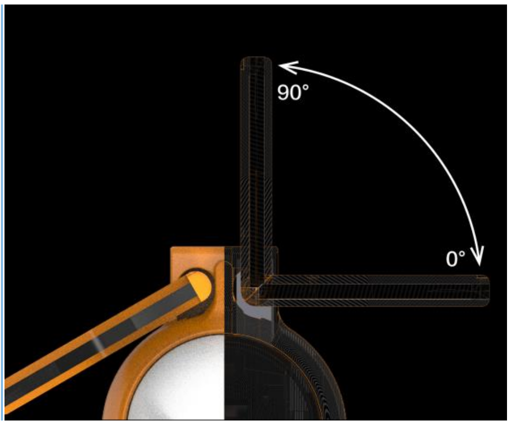
Width
Fill parent...

MaxValue
90

MinValue
0

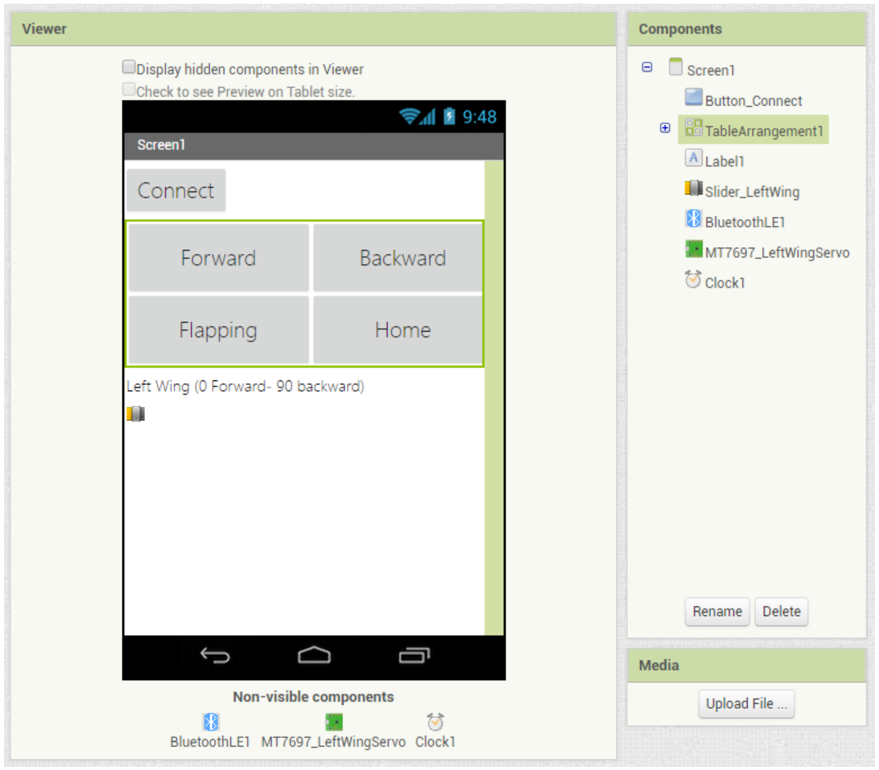
ThumbEnabled
☐

ThumbPosition
45



9. Add a Clock component, uncheck the **TimerEnabled** property and set **TimerInterval** to **1000** (1 second). We use this time duration to make sure the wing can move to its destination.

After some adjusting, your designer page should look similar to the image below. It doesn't have to be exactly the same. Feel free to modify the component's background color, position and text size. *Note: The TableArrangement and components insides are shown here, but don't forget that we've set it to be invisible in step 6.*


Viewer

☐ Display hidden components in Viewer
☐ Check to see Preview on Tablet size.

9:48
Screen1
Connect
Forward | Backward
Flapping | Home
Left Wing (0 Forward- 90 backward)

Non-visible components
BluetoothLE1 MT7697_LeftWingServo Clock1

Components

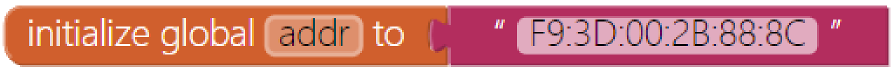
- Screen1
  - Button_Connect
  - TableArrangement1
  - Label1
  - Slider_LeftWing
  - BluetoothLE1
  - MT7697_LeftWingServo
  - Clock1

## Blocks

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

1. **Variable for Bluetooth address**

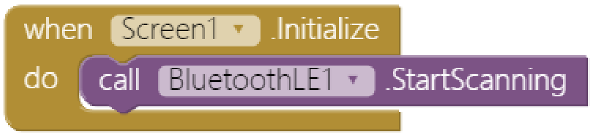
Please replace the **addr** variable with what you get from Arduino Serial Monitor, this is the Bluetooth address of LinkIt 7697. We will show you how to check this information in [Arduino IDE and sketch]() section.



```
initialize global addr to " F9:3D:00:2B:88:8C "
```

2. **Initialize app and scan for nearby Bluetooth devices**

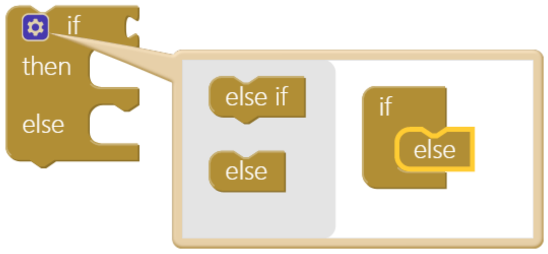
In **Screen1.Initialize** event, we ask the **BluetoothLE** component to scan for BLE devices nearby (**BluetoothLE1.StartScanning**).



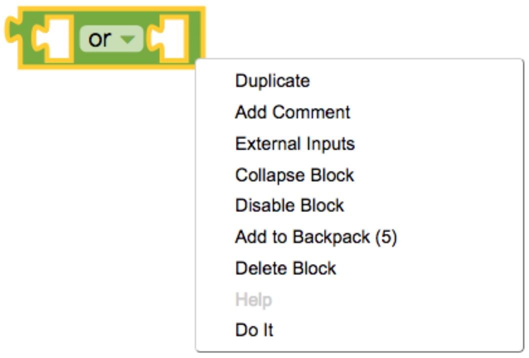
```
when Screen1 .Initialize
do  call BluetoothLE1 .StartScanning
```

3. **Connect and disconnect from Bluetooth device**

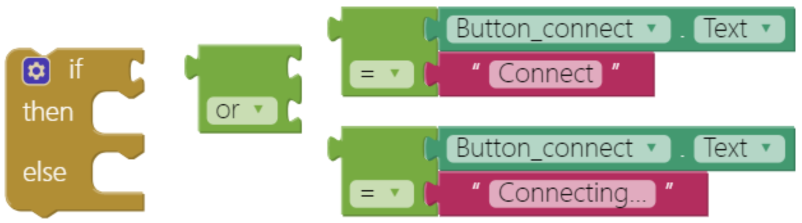
In **Button_Connect.Click** event, we are going to connect or disconnect from Bluetooth device depending on the button text. First, add an **if** condition, then click its blue gear icon to add an **else**.



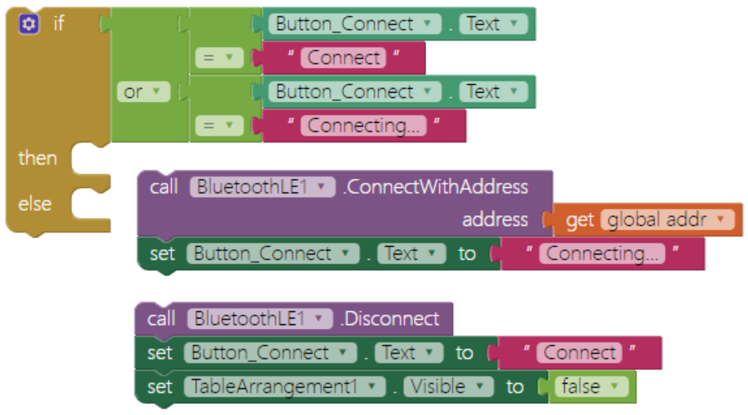
Add an **or** command from logic block, then right-click it and select "**External Inputs**". A drop-down menu will appear on the right-hand side.



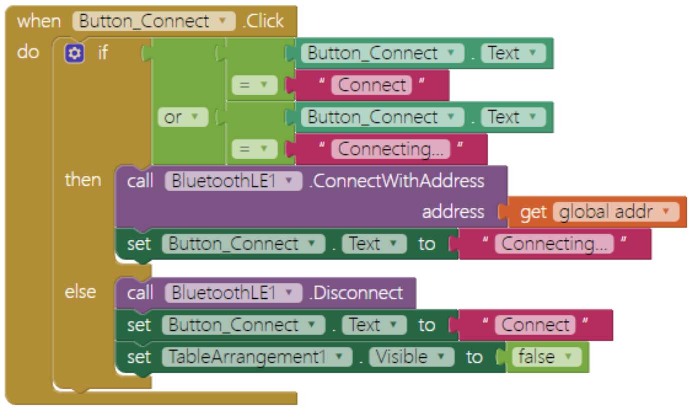
Now we want to check whether the **Button_Connect.Text** status is "**Connect**" OR "**Connecting...**", this is how App Inventor decides whether to connect or disconnect the Bluetooth connection with LinkIt 7697. Please combine these blocks.



Good! When the **Button_Connect**’s text reads "**Connect**" or "**Connecting...**", the app will connect to the specified Bluetooth device (**BluetoothLE1.ConnectwithAddress**), which is our LinkIt 7697.
If the text does not read "**Connect**", the app will disconnect (**BluetoothLE1.Disconnect**) and show a message on Button_Connect, then set the slider Enabled property to **false**.

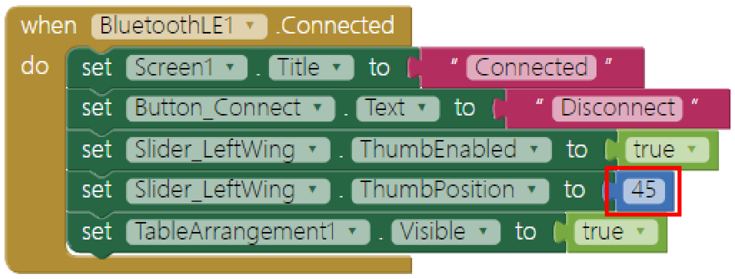

if
or
Button_Connect . Text
=
" Connect "
Button_Connect . Text
=
" Connecting... "
then
call BluetoothLE1 .ConnectWithAddress
address
get global addr
set Button_Connect . Text to " Connecting... "
else
call BluetoothLE1 .Disconnect
set Button_Connect . Text to " Connect "
set TableArrangement1 . Visible to false

Put everything into the **Button_Connect.Click** event, and finish like this:



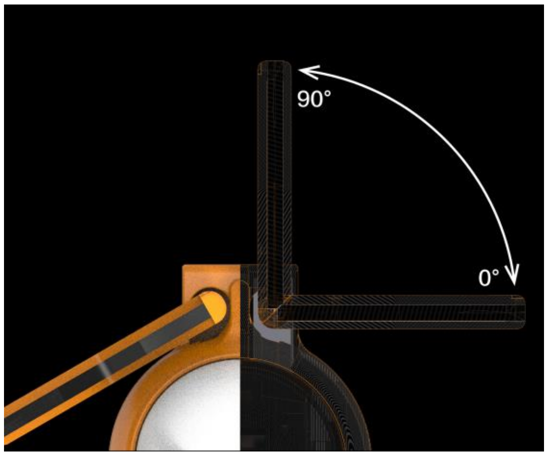
```
when Button_Connect .Click
do  if    or  Button_Connect . Text  =  " Connect "
              Button_Connect . Text  =  " Connecting... "
    then  call BluetoothLE1 .ConnectWithAddress
                                  address  get global addr
          set Button_Connect . Text to " Connecting... "
    else  call BluetoothLE1 .Disconnect
          set Button_Connect . Text to " Connect "
          set TableArrangement1 . Visible to false
```

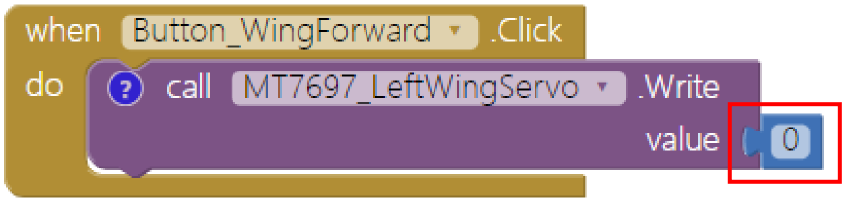
4. **BLE Connected**
   When connected successfully (**BluetoothLE.Connected** event), we will see related messages on several components. Note that we set **Slider_LeftWingServo.ThumbPosition** to 45, this will cause the left wing to move to its home position of 45 degrees.



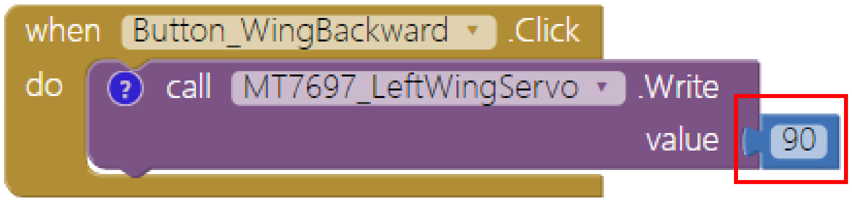
```
when BluetoothLE1 .Connected
do  set Screen1 . Title to " Connected "
    set Button_Connect . Text to " Disconnect "
    set Slider_LeftWing . ThumbEnabled to true
    set Slider_LeftWing . ThumbPosition to 45
    set TableArrangement1 . Visible to true
```

5. **Buttons to make the wing move forward and backward**
   We use several buttons to make the servo motor move the wing to the position we want.
   When **Button_WingForward** is pressed, the servo moves the wing to position **0** by **MT7697_LeftWingServo.Write** method. Similarly, when **Button_WingForward** is pressed, the servo moves the wing to position **90**.



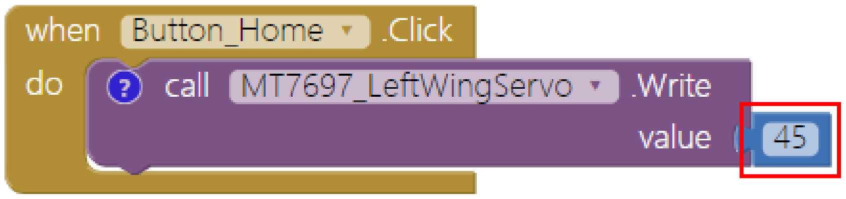


```
when Button_WingForward .Click
do  call MT7697_LeftWingServo .Write
                        value 0
```



```
when Button_WingBackward .Click
do  call MT7697_LeftWingServo .Write
                        value 90
```

6. **Button_Home to move the wing to its home position**
   Similar to the previous step, we move the left wing servo to its home position (45 degree) when **Button_Home** is pressed. This is the same action as when it is connected (Step 4.).

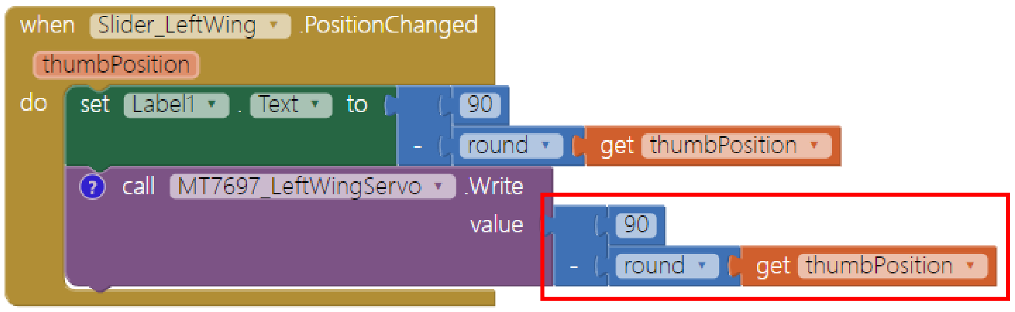


```
when Button_Home .Click
do  call MT7697_LeftWingServo .Write
                        value 45
```

7. **Drag slider to move the wing**
   When the slider is dragged (**Slider_LED_R.PositionChanged event**), we show the position on the label and tell the left wing to

move to this position: (**90 minus thumbPosition**). The 90-degree position of the slider and the wing are in the opposite direction.

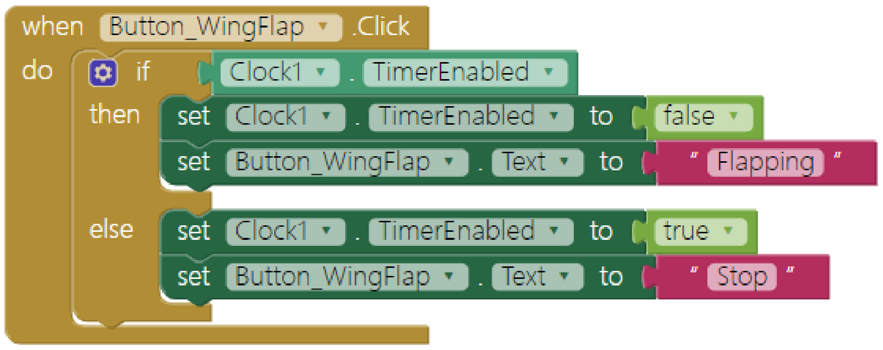
Note that we've use a math **round** method to this is because this method can apply integer only.



```
when Slider_LeftWing .PositionChanged
  thumbPosition
do  set Label1 . Text to  90 - round get thumbPosition
    call MT7697_LeftWingServo .Write
                          value  90 - round get thumbPosition
```

8. **Button to turn on/off the clock timer**

We use **Button_WingFlap** to turn on/off the clock timer, which consequently makes the wing flap or stop. In the **Button_WingFlap** event, we first check whether the Timer is on. If it is on, the app will turn off the timer and change button text to "**Flapping**"; if it is off, the app will turn on the timer and change text to "**Stop**".



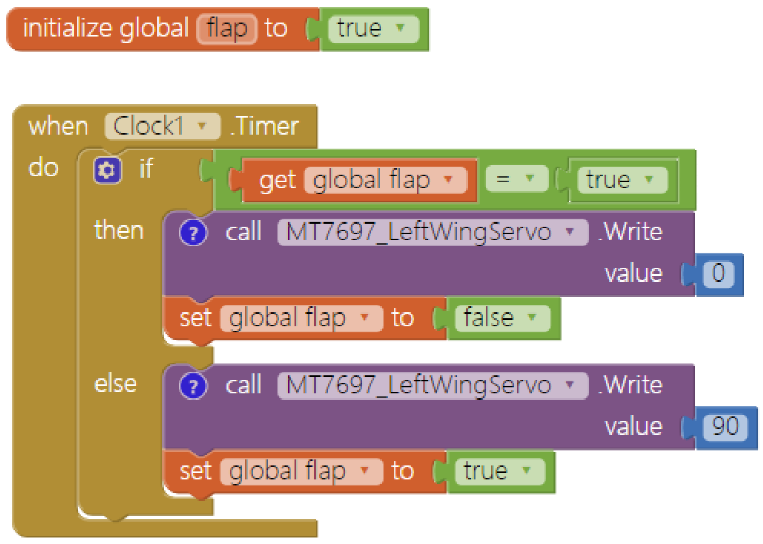
```
when Button_WingFlap .Click
do  if  Clock1 . TimerEnabled
    then  set Clock1 . TimerEnabled to false
          set Button_WingFlap . Text to " Flapping "
    else  set Clock1 . TimerEnabled to true
          set Button_WingFlap . Text to " Stop "
```

9. **Wing flap/stop**

This is the most interesting function of this project. We use a logic variable named **flap** to tell the servo whether it is allowed to move.

In **Clock.Timer** event, we check whether **flap** variable is true. If it is **true**, then make the servo move to position 0 (outward) and set the variable to **false**; if it is **false**, then move to position **90** (backward) and set the variable to **true**.

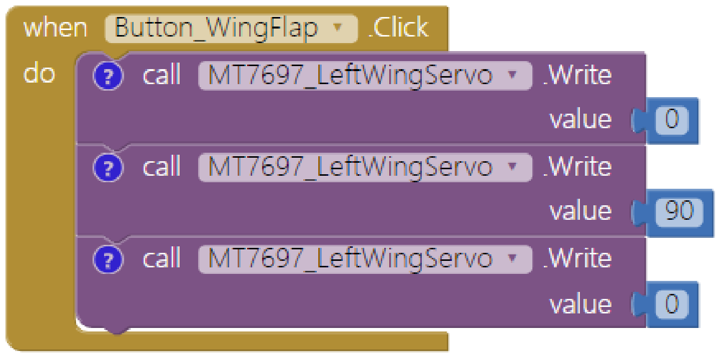
Now, Codi Bot is flapping its left wing!



```
initialize global flap to true

when Clock1 .Timer
do  if  get global flap = true
    then  call MT7697_LeftWingServo .Write
                                   value 0
          set global flap to false
    else  call MT7697_LeftWingServo .Write
                                   value 90
          set global flap to true
```

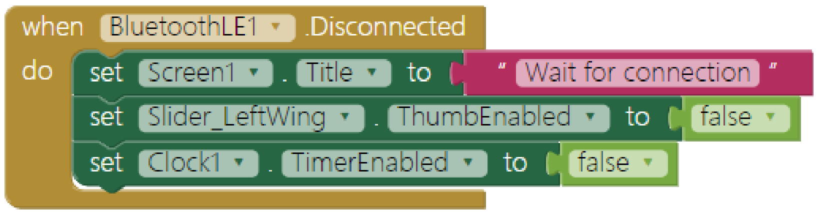
*Note: if you arrange your blocks like the ones below, the servo will do these three things in a instant, which will make it seem like the servo has gone to position 0 directly and never been to position 90.*



```
when Button_WingFlap .Click
do  call MT7697_LeftWingServo .Write
                             value 0
    call MT7697_LeftWingServo .Write
                             value 90
    call MT7697_LeftWingServo .Write
                             value 0
```

## 10. Disconnect

After the Bluetooth connection is closed successfully (trigger in Step1), we reset the app to its initial state to wait for next connect request in the **BluetoothLE1.disconnected** event.

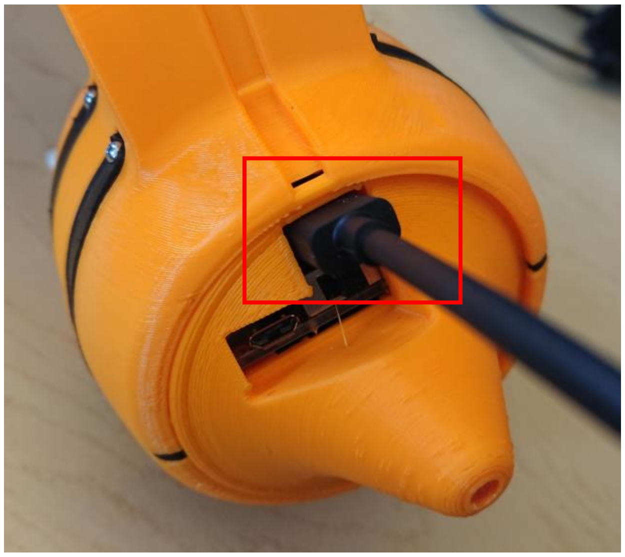


```
when BluetoothLE1 .Disconnected
do  set Screen1 . Title to " Wait for connection "
    set Slider_LeftWing . ThumbEnabled to false
    set Clock1 . TimerEnabled to false
```

## Arduino IDE and sketch

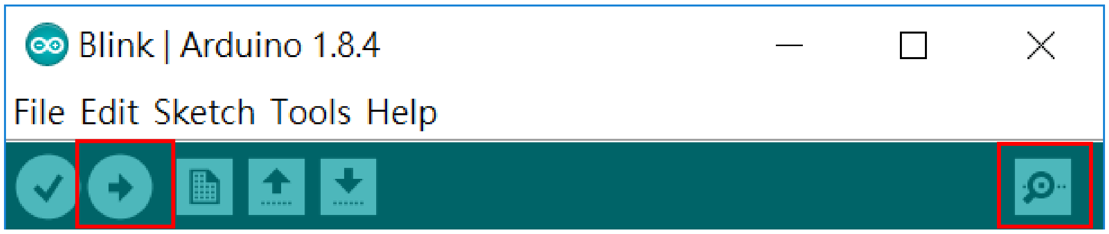
Make sure your computer has Arduino IDE installed and that the LinkIt 7697 SDK and driver are ready. If not, please check Codi Bot Standalone tutorial.

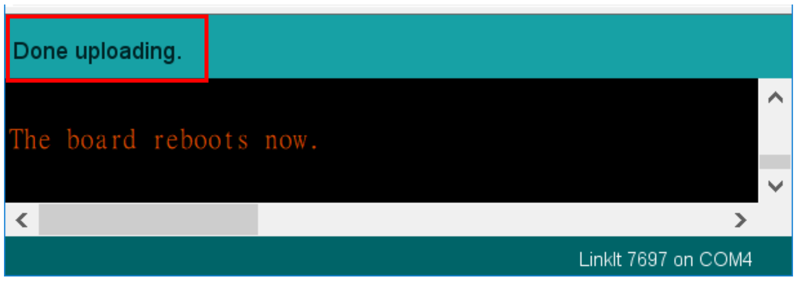
Connect your computer and the LinkIt 7697 with a microUSB cable.



Please download the Arduino sketch **here** and open in your Arduino IDE. This sketch can be used for all Codi Bot projects except the first one "**Standalone demo**", to allow you to focus on building App Inventor projects you enjoy.

Press the "**Upload**" right arrow button of Arduino IDE, this will compile and upload the Arduino sketch to your LinkIt 7697. Please make sure you see the "**done uploading**" message in the console.

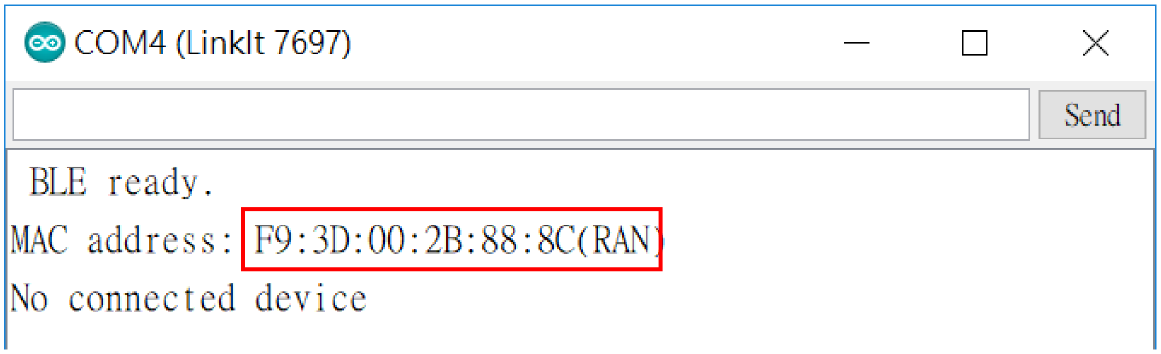




```
Done uploading.

The board reboots now.
```

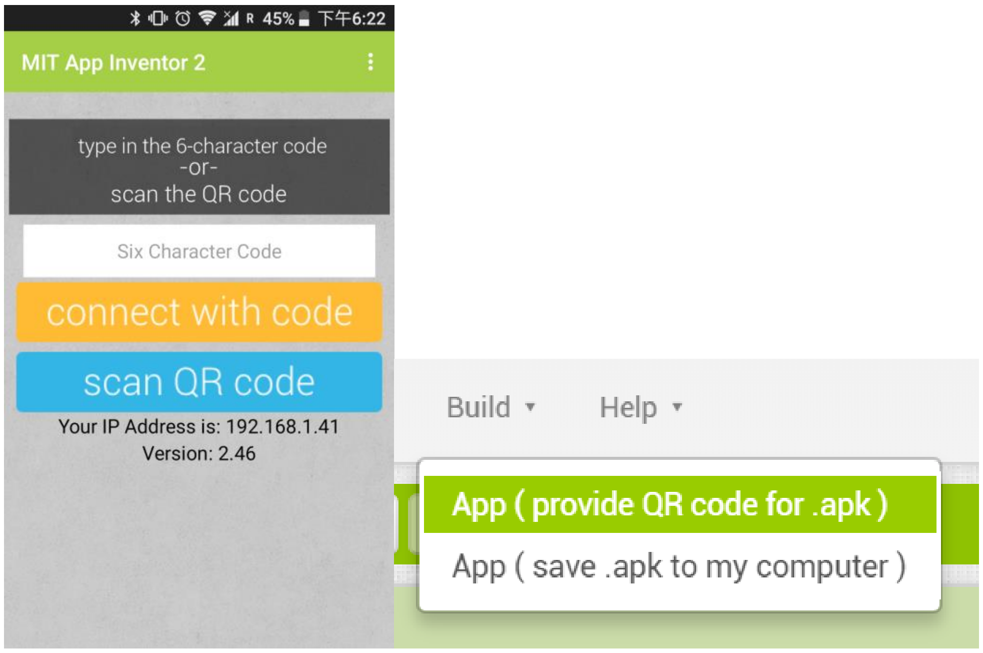
Click the magnifier icon at the upper right corner of Arduino IDE, you should see a message in the pop-up window. The [**XX:XX:XX:XX:XX:XX**] 12-digit string is the Bluetooth address of your LinkIt 7697.  We need to modify the **addr** variable value of your AI2 project.



```
 BLE ready.
MAC address: F9:3D:00:2B:88:8C(RAN)
No connected device
```

## Tips

Make sure your LinkIt 7697 is running correctly. And install App Inventor project on your Android phone by clicking Build / App (provide QR code for .apk), this will show a QR code for the .apk file of this project. Use MIT AI2 Companion to scan this QR code, download the app, and install it.



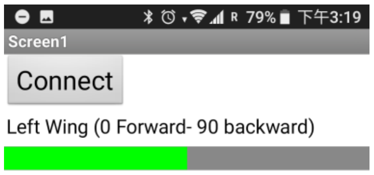
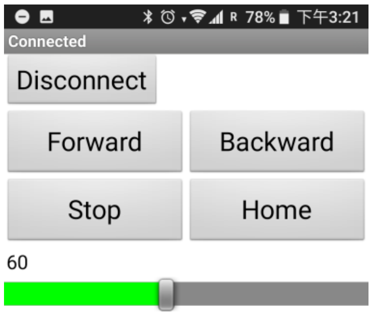
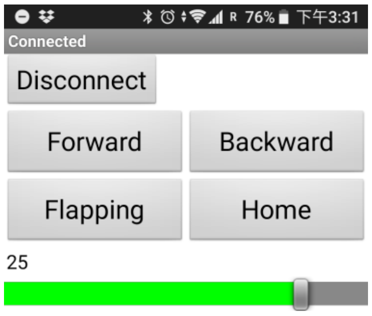
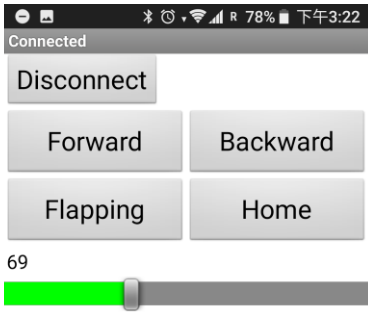
R 45% 下午6:22
MIT App Inventor 2
type in the 6-character code
-or-
scan the QR code
Six Character Code
connect with code
scan QR code
Your IP Address is: 192.168.1.41
Version: 2.46
Build
Help
App ( provide QR code for .apk )
App ( save .apk to my computer )

Open your app (Fig. 1) and click **Connect** button. After a few moments, your app screen title should show "**Connected**", meaning you connected successfully.

Click **Forward** and **Backward** buttons, Codi Bot’s left wing should move correspondingly. Also click the **Flap** button, the wing will move back and forth about once per second (Fig. 2).

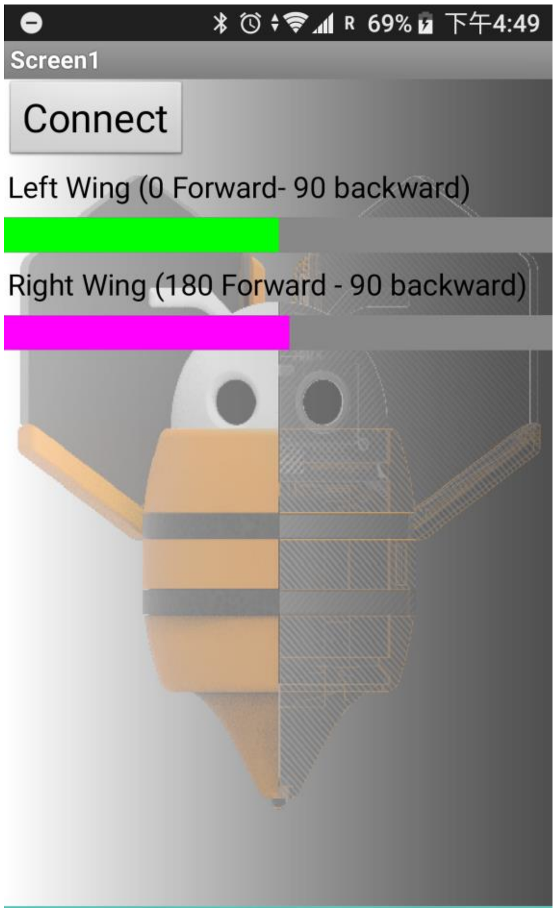
Finally, drag the slider. The wing should move to the position to the position you have selected. Don’t drag too fast or you may damage the servo. It takes time for servo to move to a specific position.

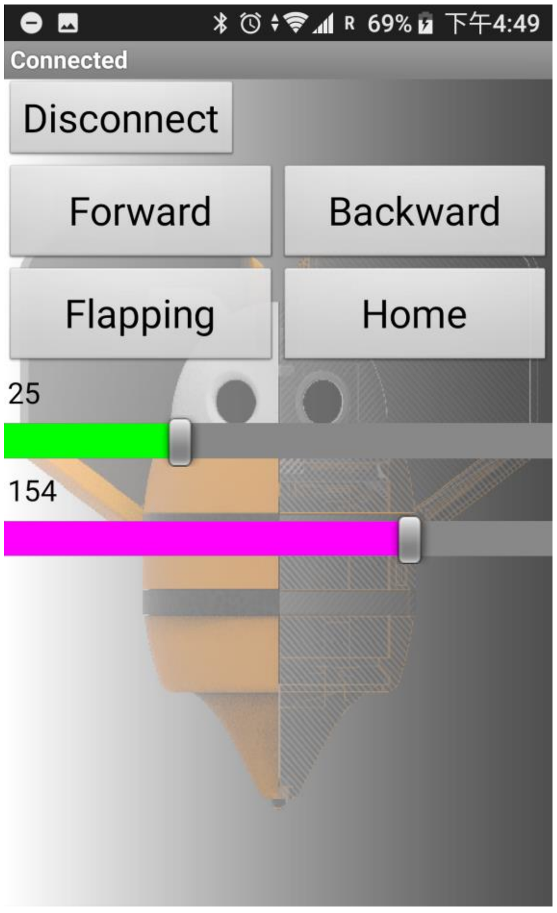
Remember to click the **Disconnect** button when you finish with this project.

| | |
|---|---|
| Fig 1. Initial screen | Fig 2. Press "**Flapping**" button to flap the wing (see its text had changed to "**Stop**"). |
|   79% 下午3:19<br>Screen1<br>Connect<br>Left Wing (0 Forward- 90 backward) |   78% 下午3:21<br>Connected<br>Disconnect<br>Forward<br>Backward<br>Stop<br>Home<br>60 |
| Fig 3. Drag slider to the right (**25** degrees) | Fig 4. Drag slider to the left (**69** degrees) |
|   76% 下午3:31<br>Connected<br>Disconnect<br>Forward<br>Backward<br>Flapping<br>Home<br>25 |   78% 下午3:22<br>Connected<br>Disconnect<br>Forward<br>Backward<br>Flapping<br>Home<br>69 |

## Complete wing control app

We have provided a complete app to control both Codi Bot wings. If you prefer, you can import this **complete .aia** to control your MIT App Inventor Codi Bot.

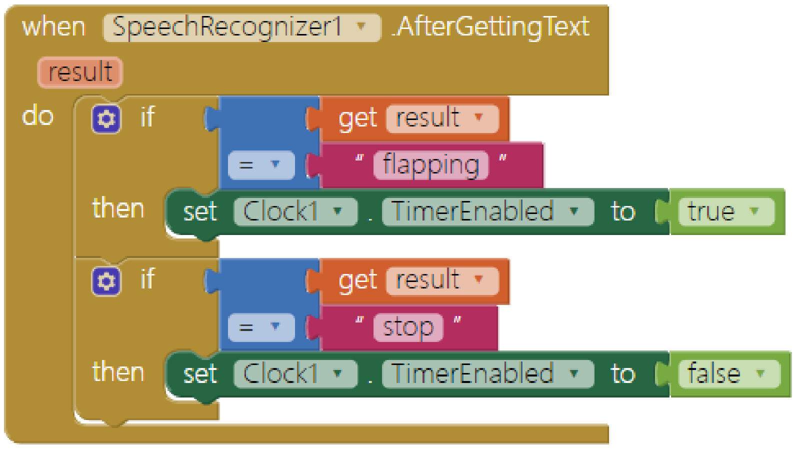

69% 下午4:49
Screen1
Connect
Left Wing (0 Forward- 90 backward)
Right Wing (180 Forward - 90 backward)



## Brainstorming

1. Use App Inventor **SpeechRecognizer** component to make the wing flap or stop.

*Hint: In the* ***SpeechRecognizer.AfterGettingText*** *event, if the desired result is "flapping" then turn on the Clock timer. If it is "stop" then turn the Clock timer off. Other parts of the* ***Clock.Timer*** *event are unchanged.*



```
when SpeechRecognizer1 .AfterGettingText
  result
do  if    get result = " flapping "
    then  set Clock1 . TimerEnabled to true
    if    get result = " stop "
    then  set Clock1 . TimerEnabled to false
```