

**MIT AI2 204**

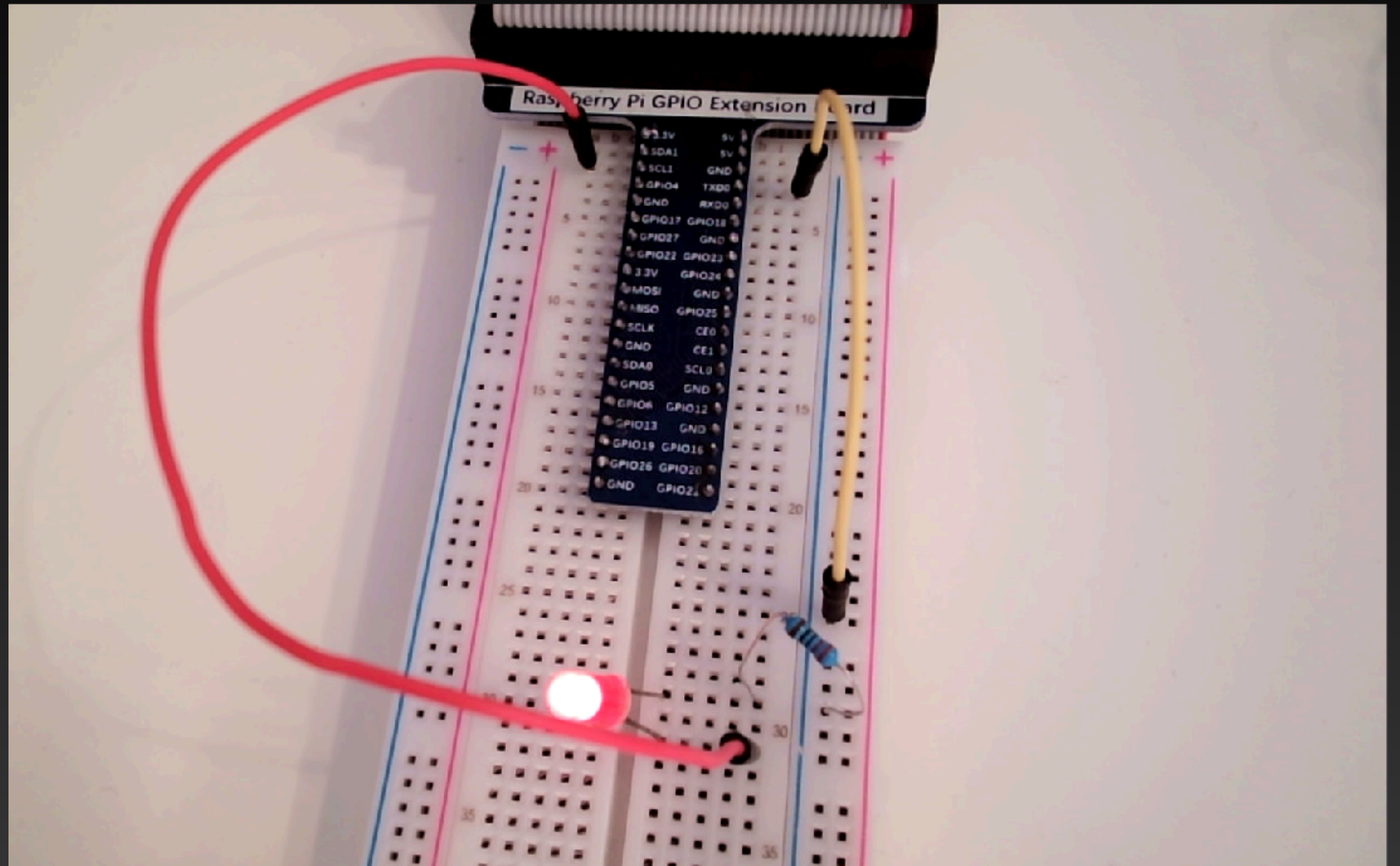
# **IoT with MIT App Inventor**

**Fundamental**

X. Tang

# Flashing LED - Accessing the External World

Connect the wires use GPIO2 as output.



## Raspberry Pi Wi-Fi based projects

```
sudo apt remove python3-rpi.gpio
```

```
sudo apt install python3-rpi-lgpio
```

# Raspberry Pi Wi-Fi based projects

## Project 1 - Getting and displaying the local WiFi parameters

Description: In this project, local Wi-Fi parameters, such as the IP address, MAC address, and signal strength of the Android device are obtained and displayed on its screen.

The Taifun WiFi component extension is used to get the local Wi-Fi parameters.



# Getting and Displaying Local Wi-Fi Parameters

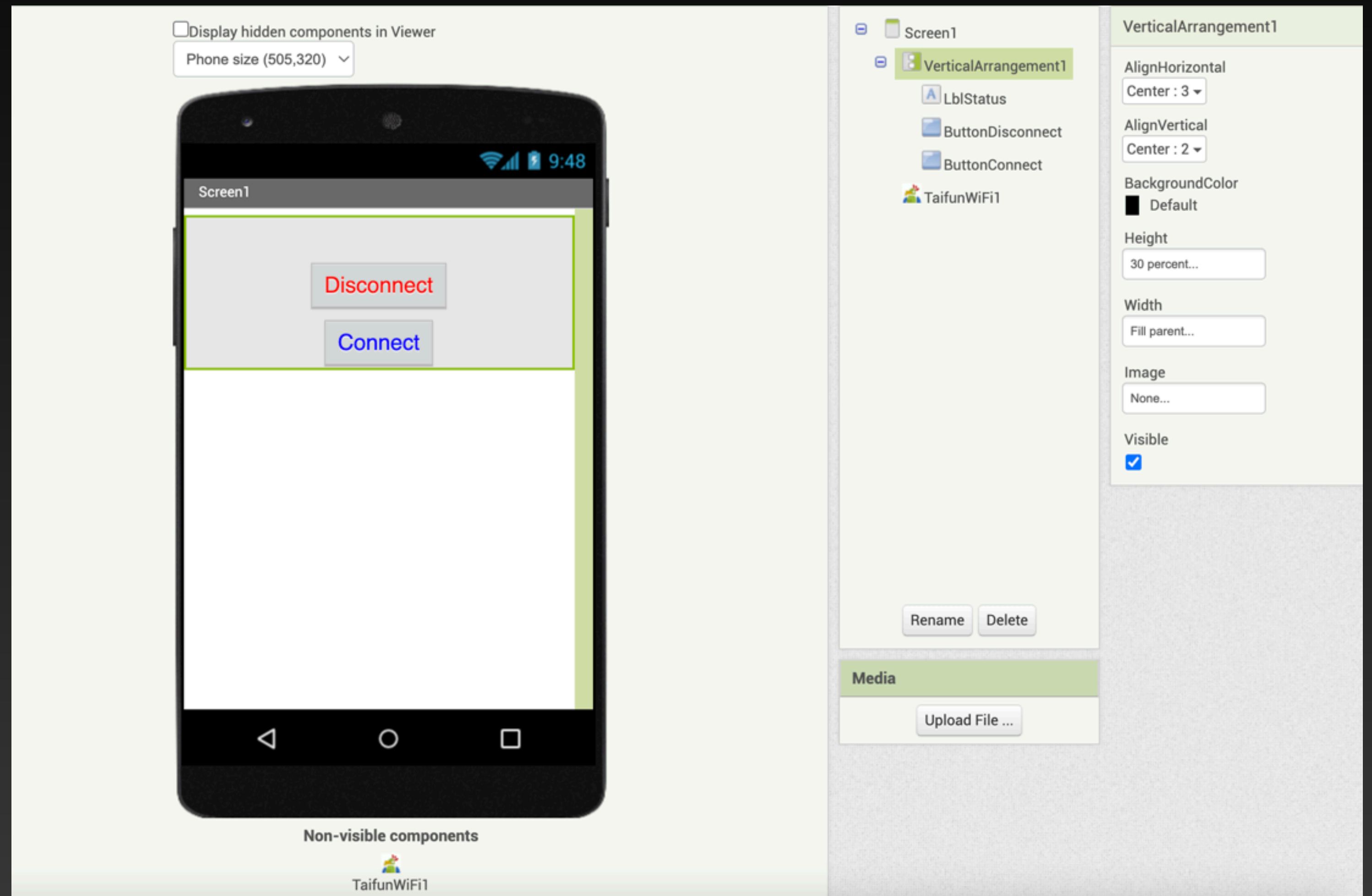
- Create a new project and name it as WIFI\_TEST
- Insert a VerticalArrangement with the following configuration:

AlignHorizontal: Center: 3

AlignVertical: Center: 2

Height: 30 percent

Width: Fill parent



# Getting and Displaying Local Wi-Fi Parameters

- Insert a Label with the following configuration. This Label will display the local Wi-Fi parameters:

Name: LblStatus

BackColor: None

FontBold: ticked

FontSize: 20

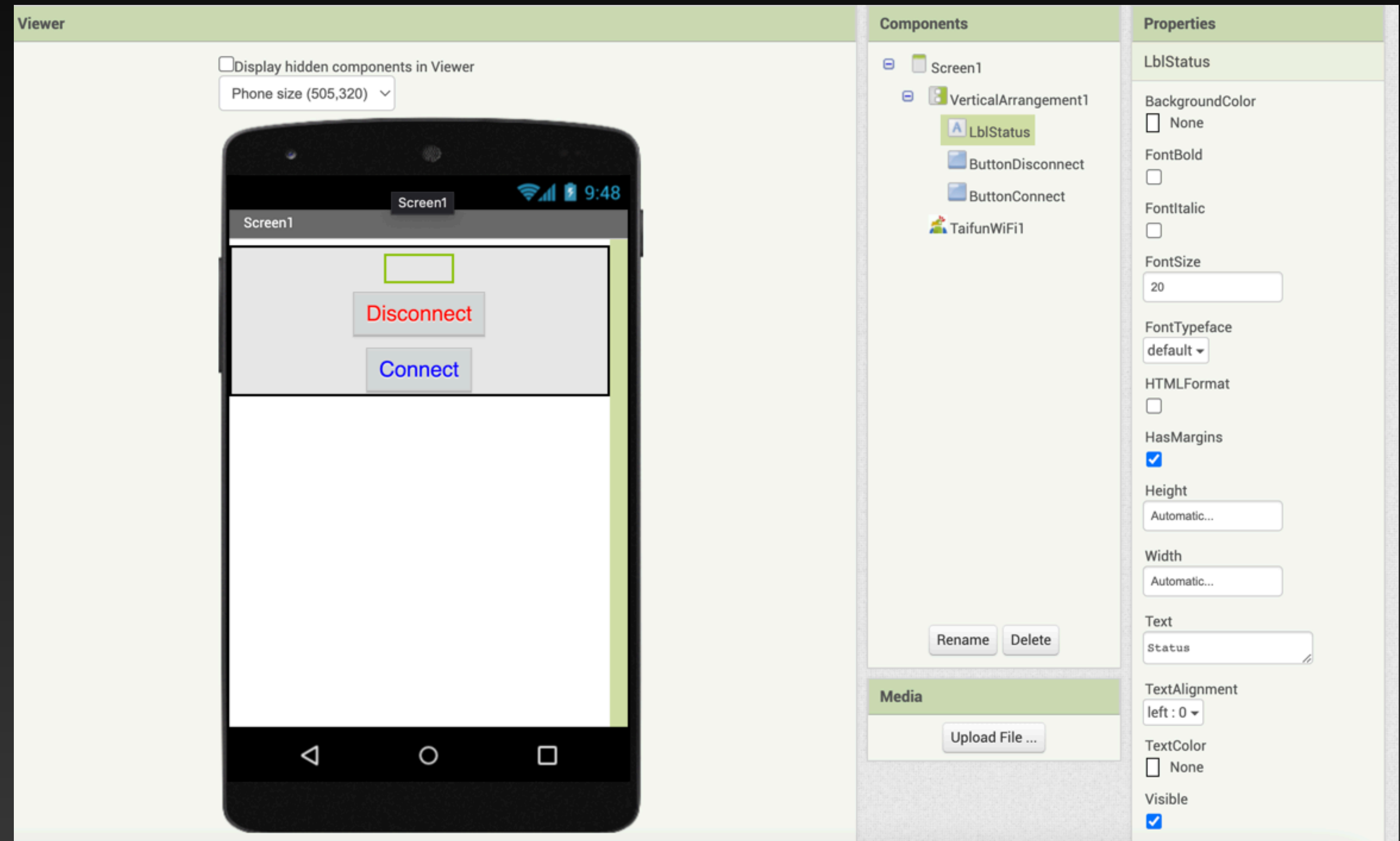
Height: Automatic

Width: Automatic

Text: Status

TextColor: None

Visible: ticked



# Getting and Displaying Local Wi-Fi Parameters

- Insert a Button with the following configuration. This button will disconnect from Wi-Fi when clicked:

Name: ButtonDisconnect

BackColor: Default

FontBold: ticked

FontSize: 20

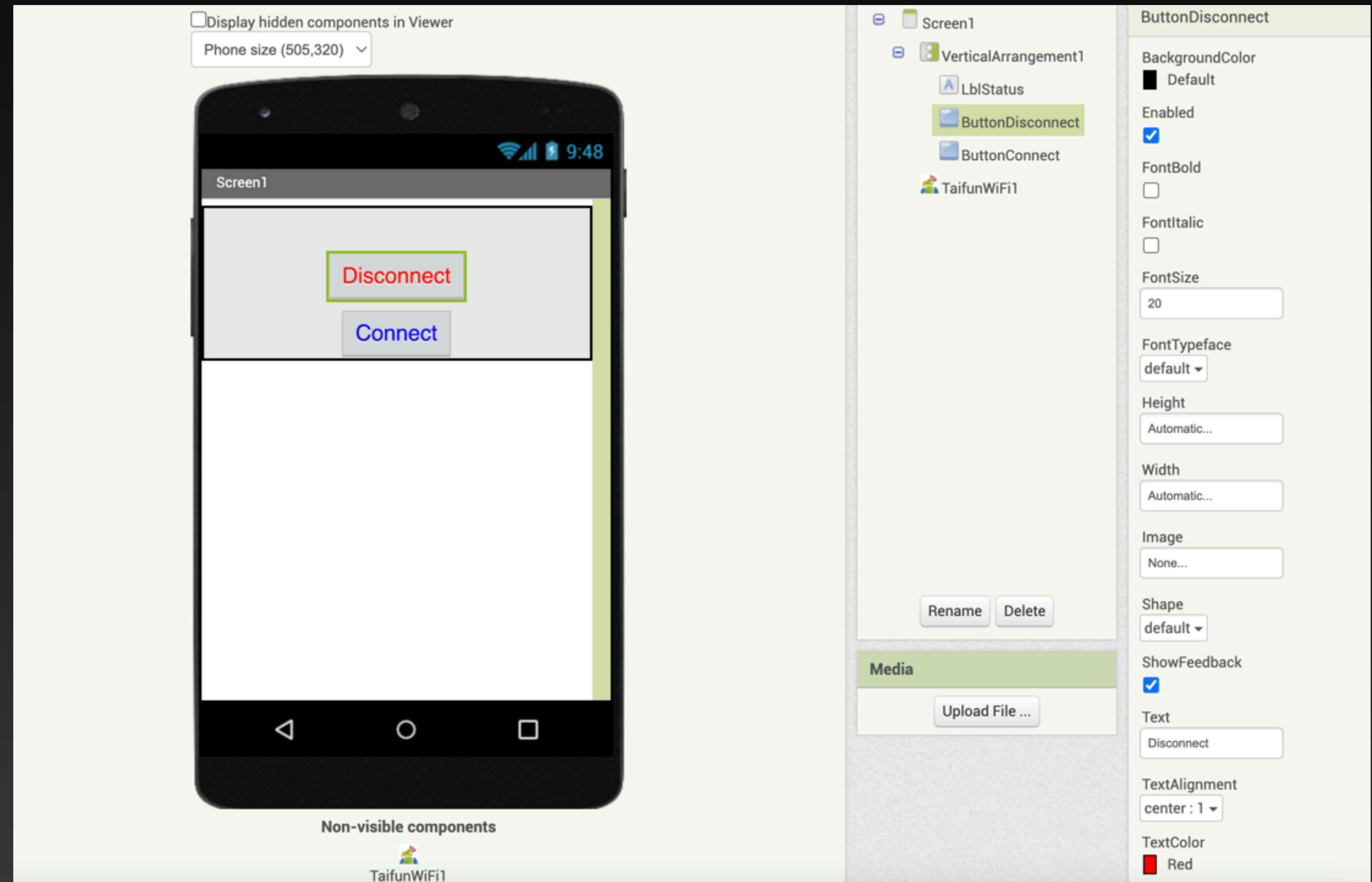
Height: Automatic

Width: Automatic

Text: Disconnect

TextColor: Red

Visible: ticked





# Getting and Displaying Local Wi-Fi Parameters

- Insert a Button with the following configuration. This button will connect to Wi-Fi when clicked:

Name: ButtonConnect

BackgroudColor: Default

FontBold: ticked

FontSize: 20

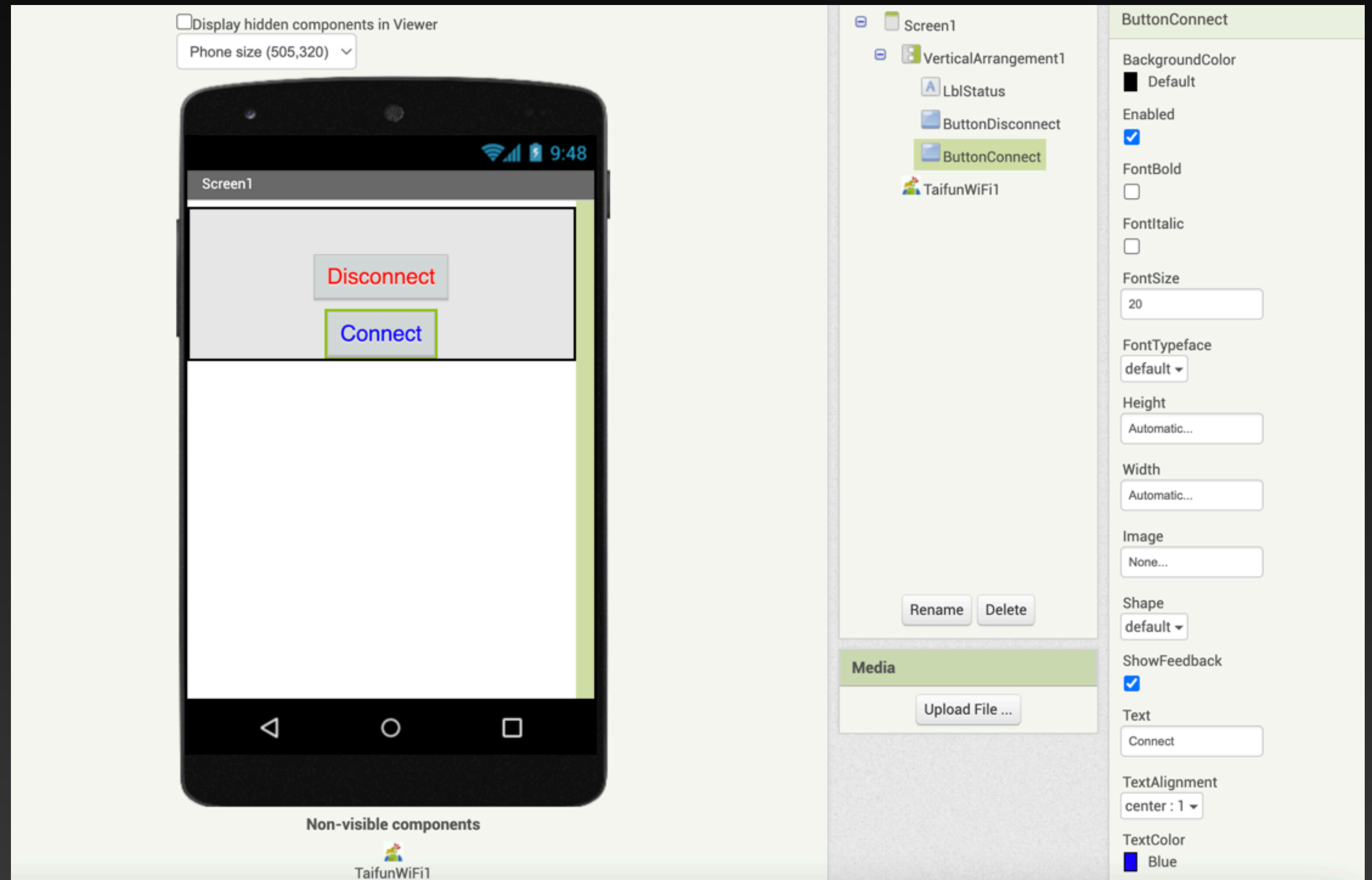
Height: Automatic

Width: Automatic

Text: Connect

TextColor: Blue

Visible: ticked






# Getting and Displaying Local Wi-Fi Parameters


- Go to the following website:  
<https://puravidaapps.com/wifi.php>
- Go to the end of the site and click Download TaifunWifi extension (aix file), download it to a folder

Pura Vida Apps Snippets Tutorials Extensions Links Search Privacy Policy Contact

 App Inventor Extensions

Search owner of a phone number Powered By BeenVerified

Enter Area Code:  -

 WiFi Manag

See the [App Inventor Extensions](#)


For questions about this extension


For feature requests [please contact](#) USD! With your contribution you will


Download

Developing and maintaining snippets, tutorials and extensions for App Inventor takes a lot of time. I hope it saved some of your time. If yes, then you might consider to donate a small amount!

Donation amount: 5 USD



or donate some mBTC to  Address: **1Jd8kXLHu2Vkuhi15TWHiQm4uE9AGPYxi8**




Thank you! Taifun

[Download TaifunWifi extension \(aix file\)](#) ←

[Download WiFi Test \(aia file\)](#)

[Download Available SSIDs Test \(aia file\)](#)

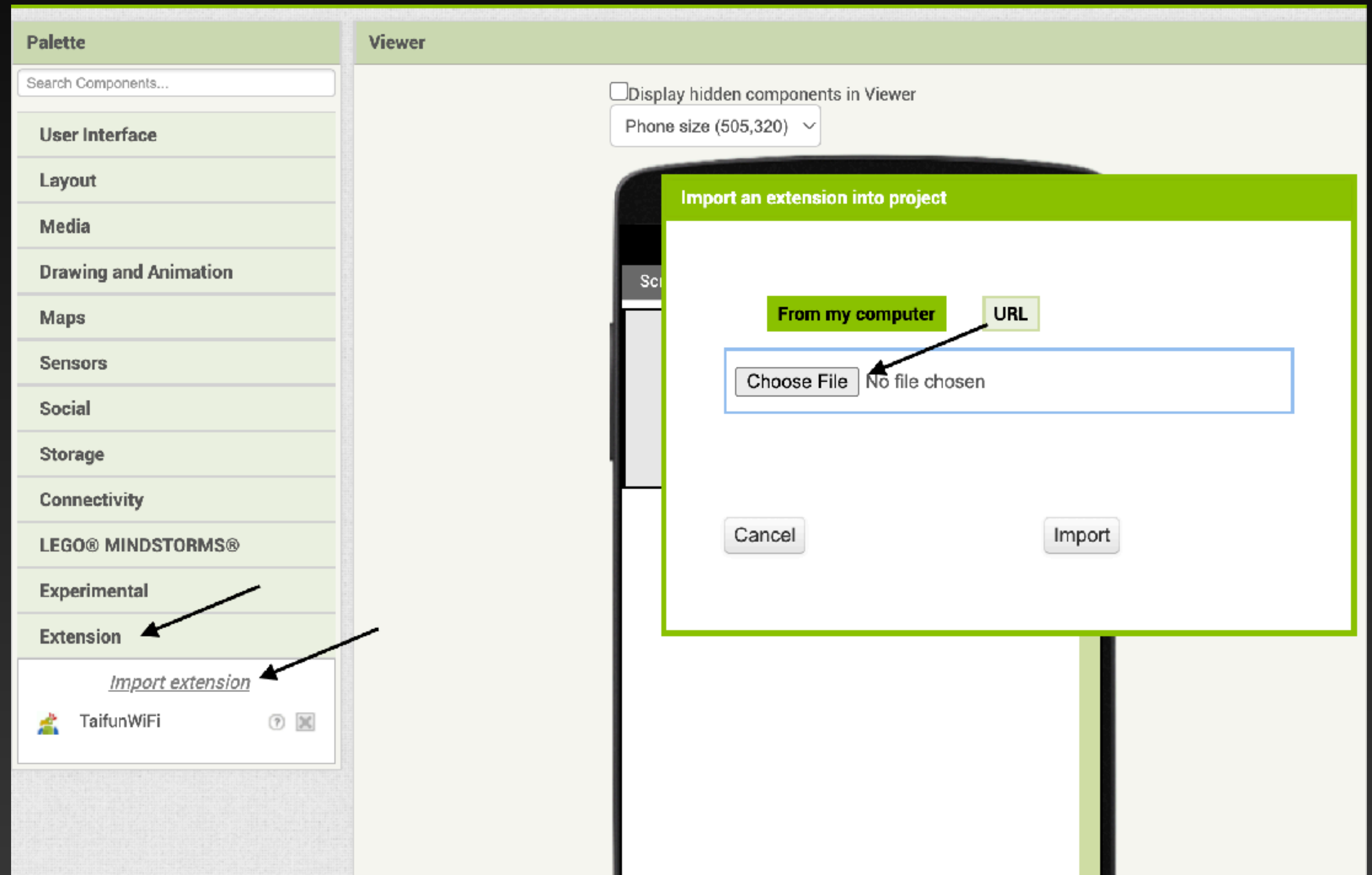
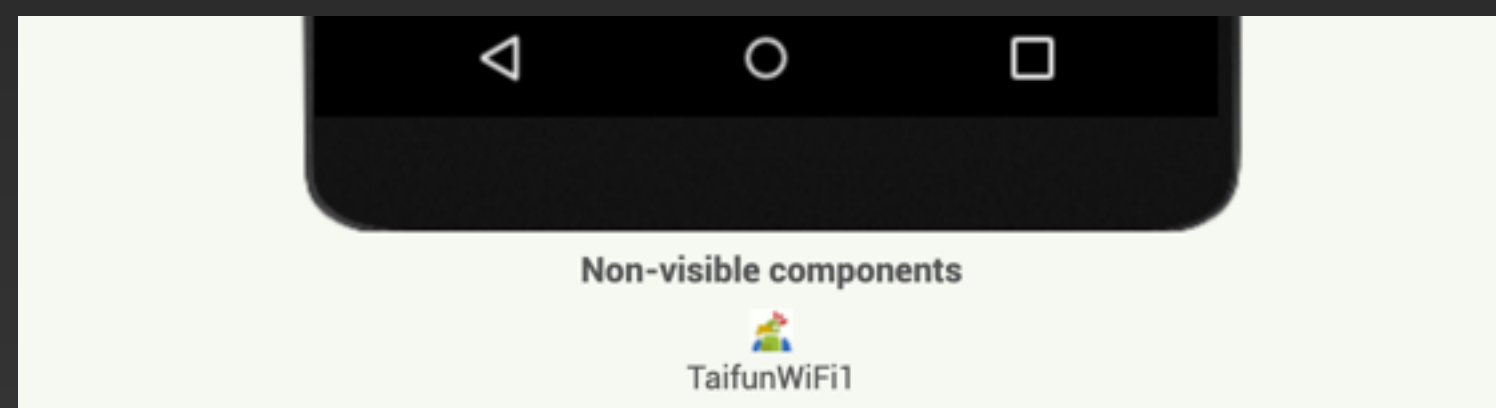
[Back to top of page ...](#)



This work by Pura Vida Apps is licensed under a [Creative Commons Attribution-ShareAlike 3.0 Unported License](#) with attribution (name=Pura Vida Apps and link to the source site) required.

# Getting and Displaying Local Wi-Fi Parameters

- Click Extension tab on the left-hand side of your app inventor project
- Click Import extension and browse to the file you just downloaded
- You should see a new component called TaifunWiFi under Extension. Click and drop it on your Viewer. This is a hidden component and will only be displayed under the phone image as TaifunWiFi1



# Getting and Displaying Local Wi-Fi Parameters

- Initialize two variables named ssid and password, Enter YOUR Wi-Fi SSID name and password into these blocks respectively.
- Click ButtonConnect and select when ButtonConnect.Click do. This block will be executed when button Connect is clicked.
- Click on TaifunWiFi1 and select call TaifunWiFi1.ConnectSSID and join blocks ssid and password to this block. This block will establish a connection to the local Wi-Fi router.

The image displays a Scratch script for managing a local Wi-Fi connection using the TaifunWiFi1 library. It includes two main event-driven blocks: one for connecting and one for disconnecting.

**Initialization:**

- `initialize global ssid to "1234"`
- `initialize global password to "5678"`

**When ButtonConnect.Click do:**

- `call TaifunWiFi1 .ConnectSSID` (with inputs: `ssid` from global, `password` from global)
- `set LblStatus . Text to` `join` ("Connected", "Local IP", `call TaifunWiFi1 .LocalIP`, "\n", "MAC Address", `call TaifunWiFi1 .MacAddress`, "\n", "Signal Strength", `call TaifunWiFi1 .SignalStrength`)
- `set LblStatus . TextColor to` blue

**When ButtonDisconnect.Click do:**

- `call TaifunWiFi1 .Disconnect`
- `set LblStatus . Text to` "Disconnected"
- `set LblStatus . TextColor to` red

**Warning Panel:**

0 0  
Show Warnings



# Getting and Displaying Local Wi-Fi Parameters

- Inset Join block and extend it to 8 connectors. Enter blocks as shown to display the IP address, MAC address, and the signal strength.
- Click Button Disconnect disconnects from the Wi-Fi

The image shows a Scratch script for connecting to a Wi-Fi network and displaying its parameters. The script is organized into two main sections: a 'when ButtonConnect.Click' section and a 'when ButtonDisconnect.Click' section.

**Initialization:**

- `initialize global ssid to "1234"`
- `initialize global password to "5678"`

**Connect Section (when ButtonConnect.Click):**

- `do` block containing:
  - `call TaifunWiFi1.ConnectSSID` with inputs `ssid` (global) and `password` (global).
  - `set LblStatus.Text to` a join block with 8 connectors:
    - Connector 1: `"Connected"`
    - Connector 2: `"Local IP"`
    - Connector 3: `call TaifunWiFi1.LocalIP`
    - Connector 4: `"\n"`
    - Connector 5: `"MAC Address"`
    - Connector 6: `call TaifunWiFi1.MacAddress`
    - Connector 7: `"\n"`
    - Connector 8: `"Signal Strength"`
  - `set LblStatus.TextColor to` a blue color block.

**Disconnect Section (when ButtonDisconnect.Click):**

- `do` block containing:
  - `call TaifunWiFi1.Disconnect`
  - `set LblStatus.Text to "Disconnected"`
  - `set LblStatus.TextColor to` a red color block.

**Warning Panel:**

At the bottom left, there is a 'Show Warnings' panel with two warning icons (yellow triangle and red X) and a count of 0 for each.



# Getting and Displaying Local Wi-Fi Parameters

- Connect your program through AI companion, and click on connect in your app.
- What did you see on your mobile screen?

# Raspberry Pi Wi-Fi based projects

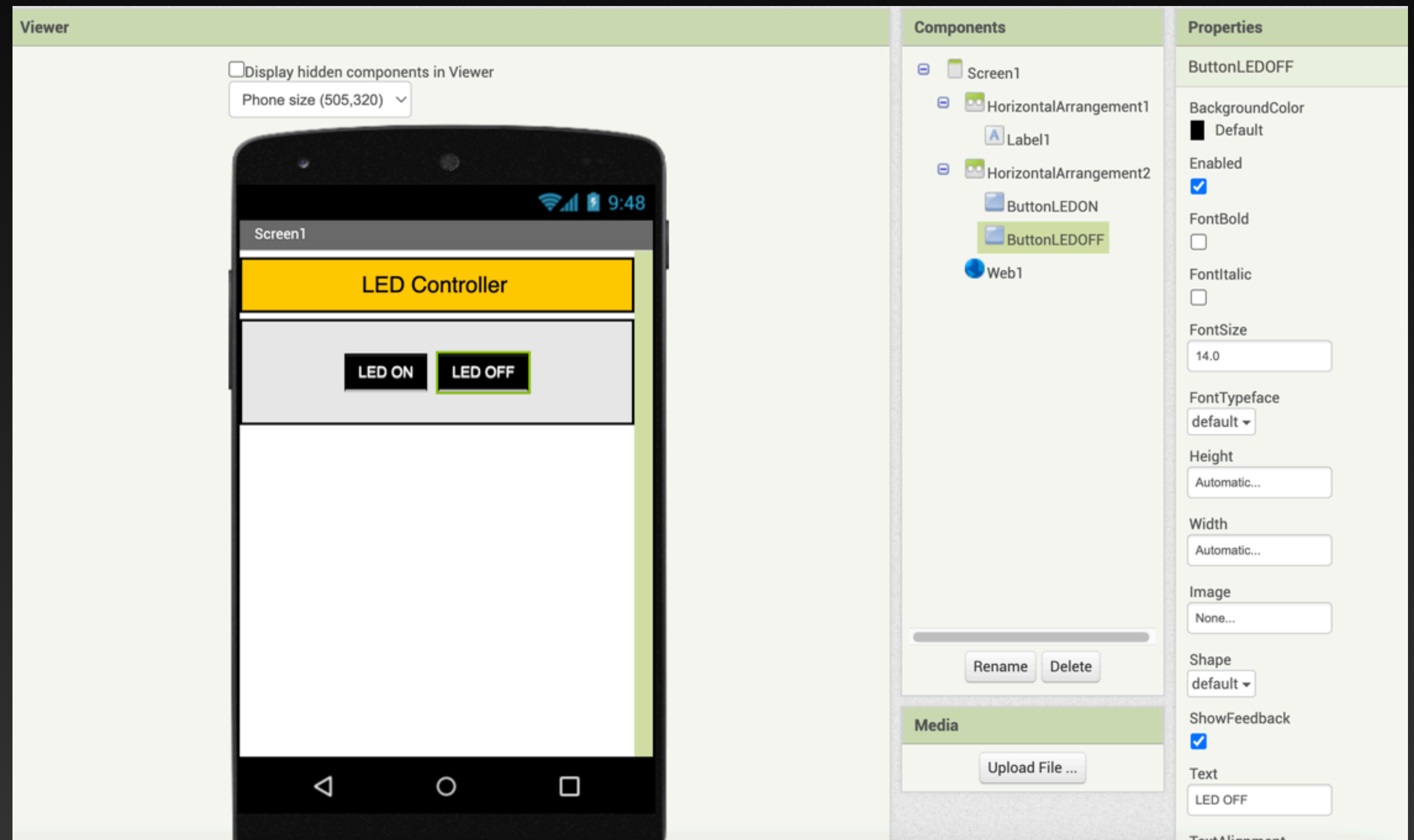
## Project 2 - Web server to control LED

Description: In this project, an LED is connected to the Raspberry Pi and is controlled from an Android mobile phone using a web server application.

Use the same circuit program from previous project to setup LED on raspberry Pi.

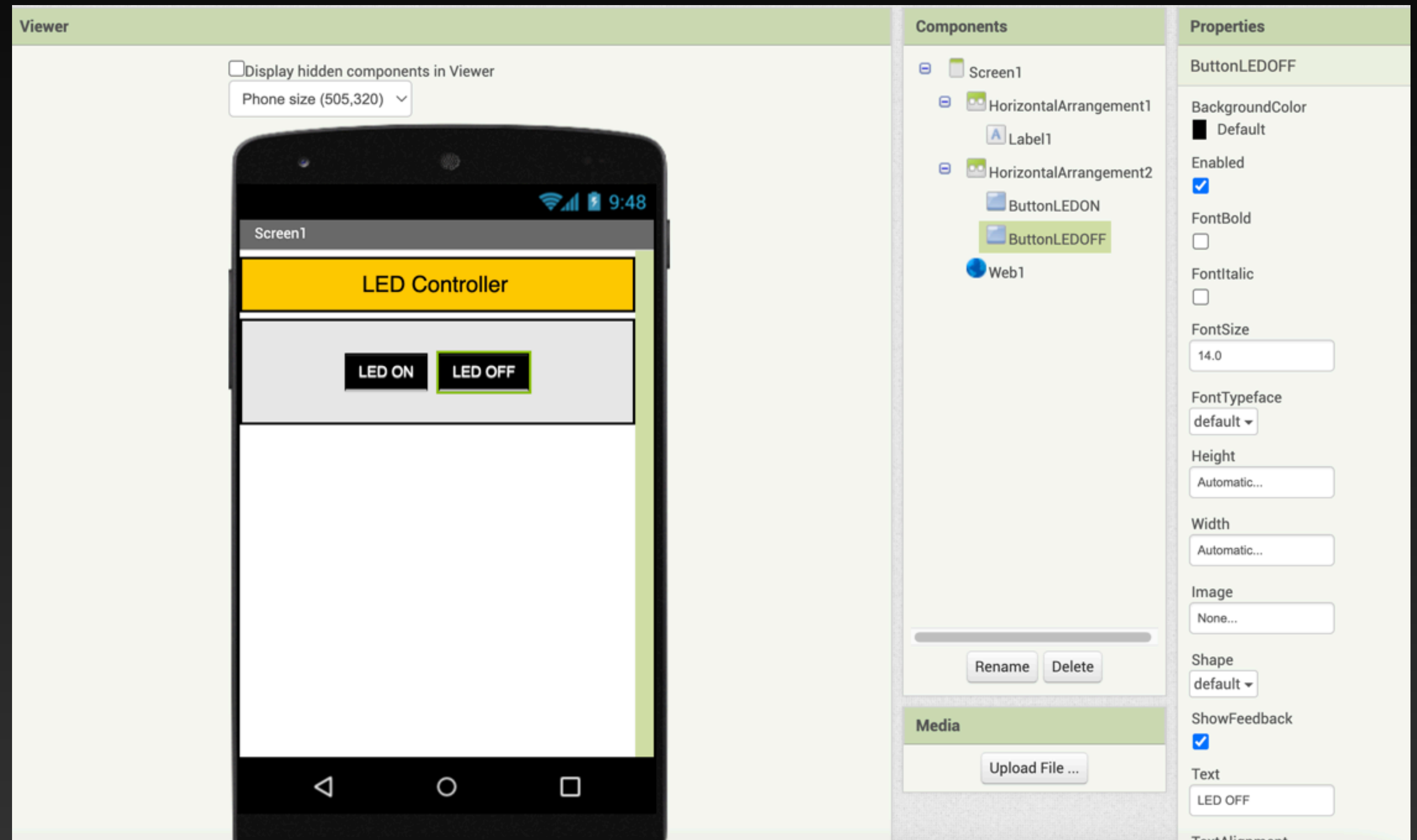
# Web Server to Control LED

- Create a new project and name it as WEB\_LED
- Insert a HorizontalArrangement and insert a Label on it with its Text set to LED CONTROLLER
- Insert another HorizontalArrangement



# Web Server to Control LED

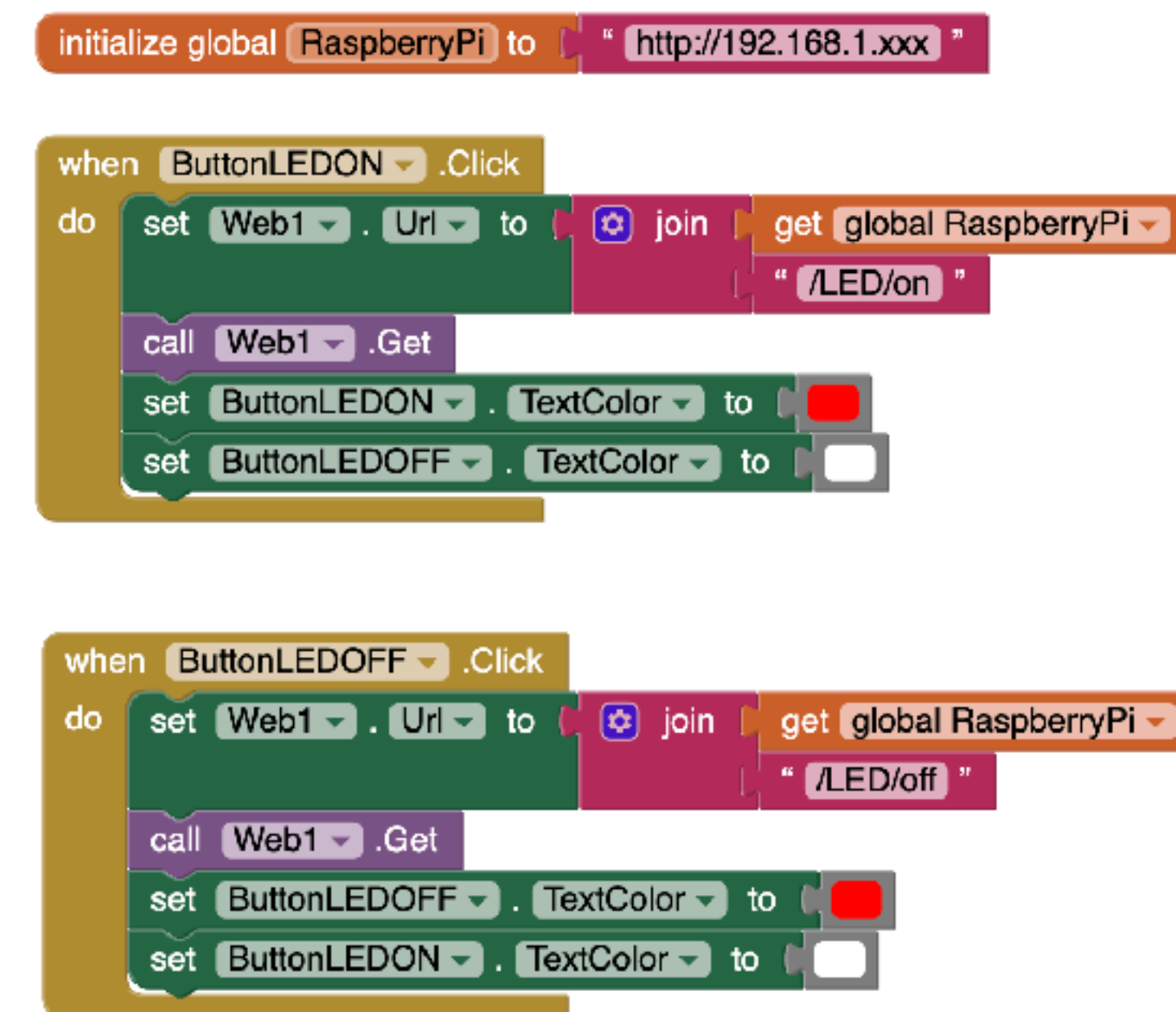
- Insert two buttons on the HorizontalArrangement with the names ButtonON and ButtonOFF, with their texts set to LED ON and LED OFF respectively
- Click the Connectivity tab and insert a Web component on the Viewer. This is a hidden component.
- Initialize a variable called RaspberryPi and set it to the IP address of your RaspberryPi





# Web Server to Control LED

- Initialize a variable called RaspberryPi and set it to the IP address of your RaspberryPi
- Click ButtonON and select when ButtonON.Click do. This block will be executed when button LED ON is clicked.
- Click on Web1 and select set Web1.Url to.
- Insert a Join block and set the URL to the IP address of your URL and add string /LED/on to this block.



# Web Server to Control LED

- Click on Web1 and select call Web1.Get. In this project, the URL is set to <http://192.168.1.xxx/LED/on>
- Set the cooler of ButtonON to red , and the color of ButtonOFF to white
- Repeat for the ButtonOFF as shown in the second group of blocks. But this time set the URL to <http://192.168.1.xxx/LED/off>

