# Implementing WebIOPi Framework on Raspberry Pi


Project Presentation

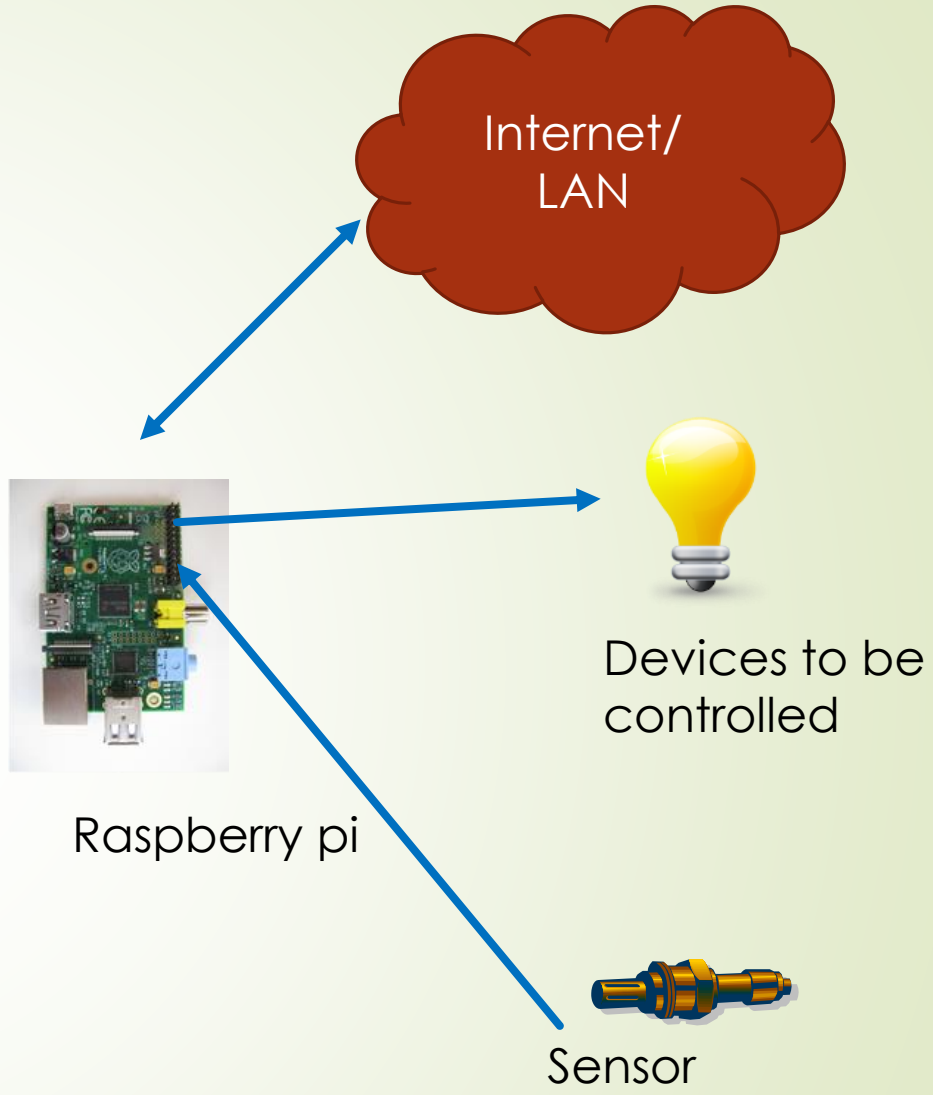
Ubaier Bhat

Ali Belakehal

TALLINNA TEHNIKAÜLIKOOL
TALLINN UNIVERSITY OF TECHNOLOGY

# Objective

- To build an Internet of Things application
  - e.g. be able to control and senesce different devices over via the internet or local area network

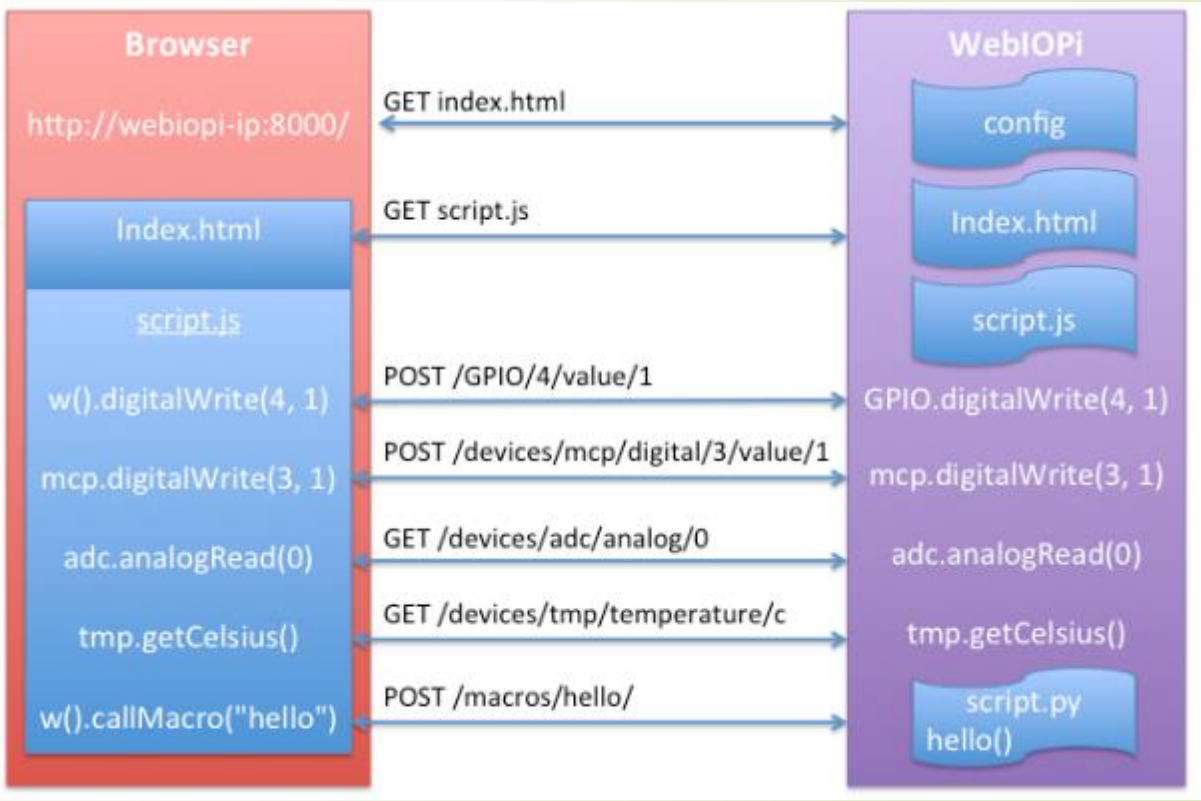


# Components

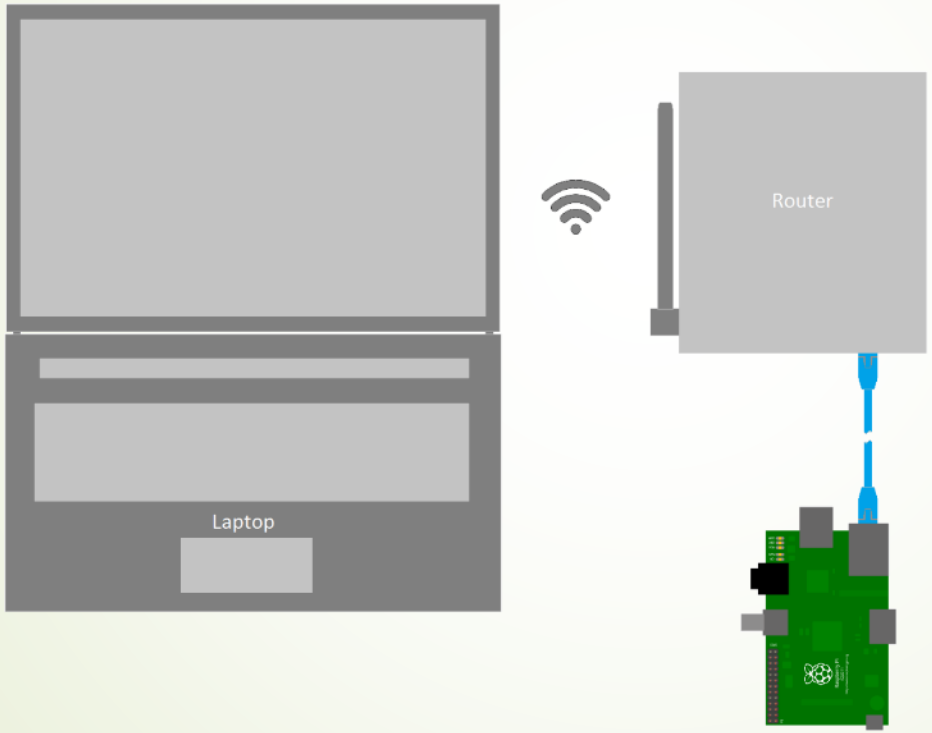
- Hardware
  - Raspberry Pi
    - Model B Revision 2.0
  - For Testing
    - LEDs
    - Relay (Songle SRD-05VDC-SL-C)
- Software
  - Raspbian wheezy
  - Software: WebIOPi
  - Languages: HTML, CSS, JavaScript, Python

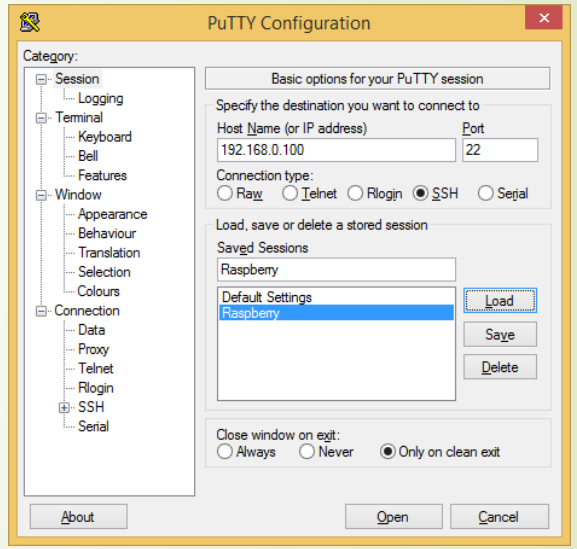
# WebIOPi

- Open Source
- https://code.google.com/p/webiopi/
- Build on Apache web server
- Client-Server framework
- REST verbs (GET, POST)
- HTML, JavaScript, CSS, Python Script



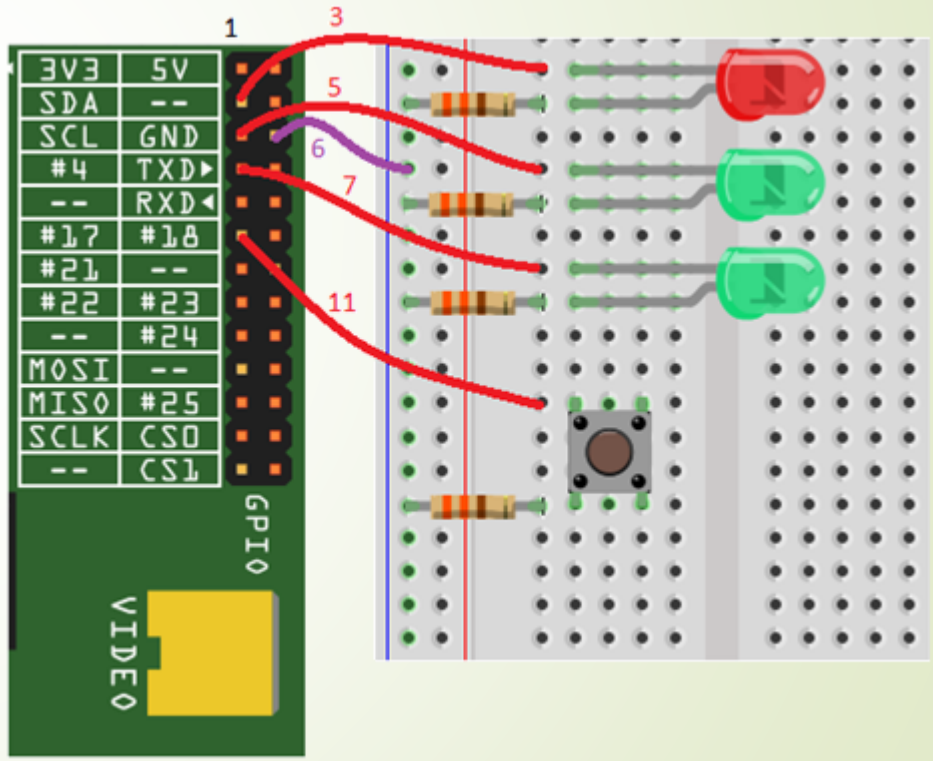
# Configuring Raspberry Pi



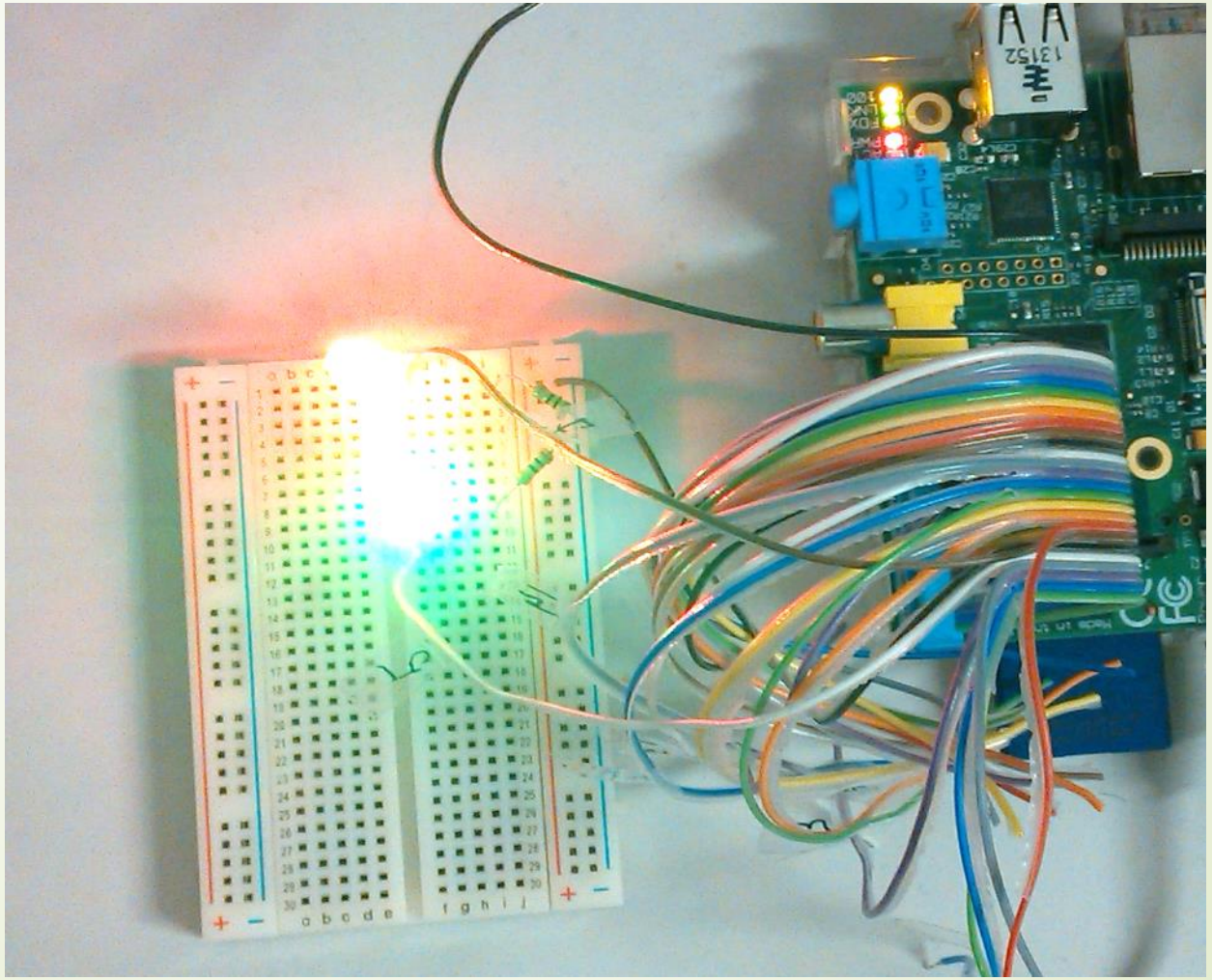



# GPIO connections

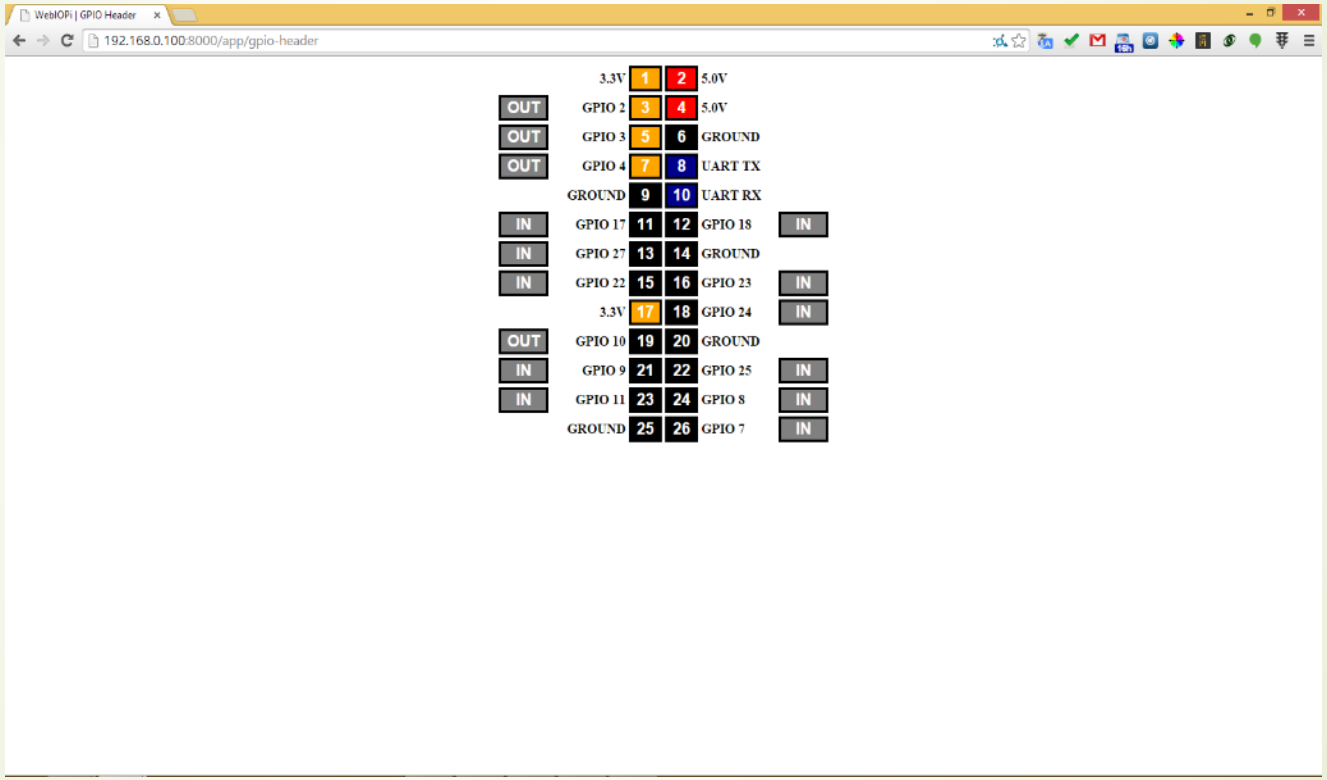
| Device | GPIO # | Pin |
|---|---|---|
| LED 1 (red) | 2 | 3 |
| LED 2 (green) | 3 | 5 |
| LED 3 (green) | 4 | 7 |
| Switch/sensor | 17 | 11 |



# Setting up breadboard



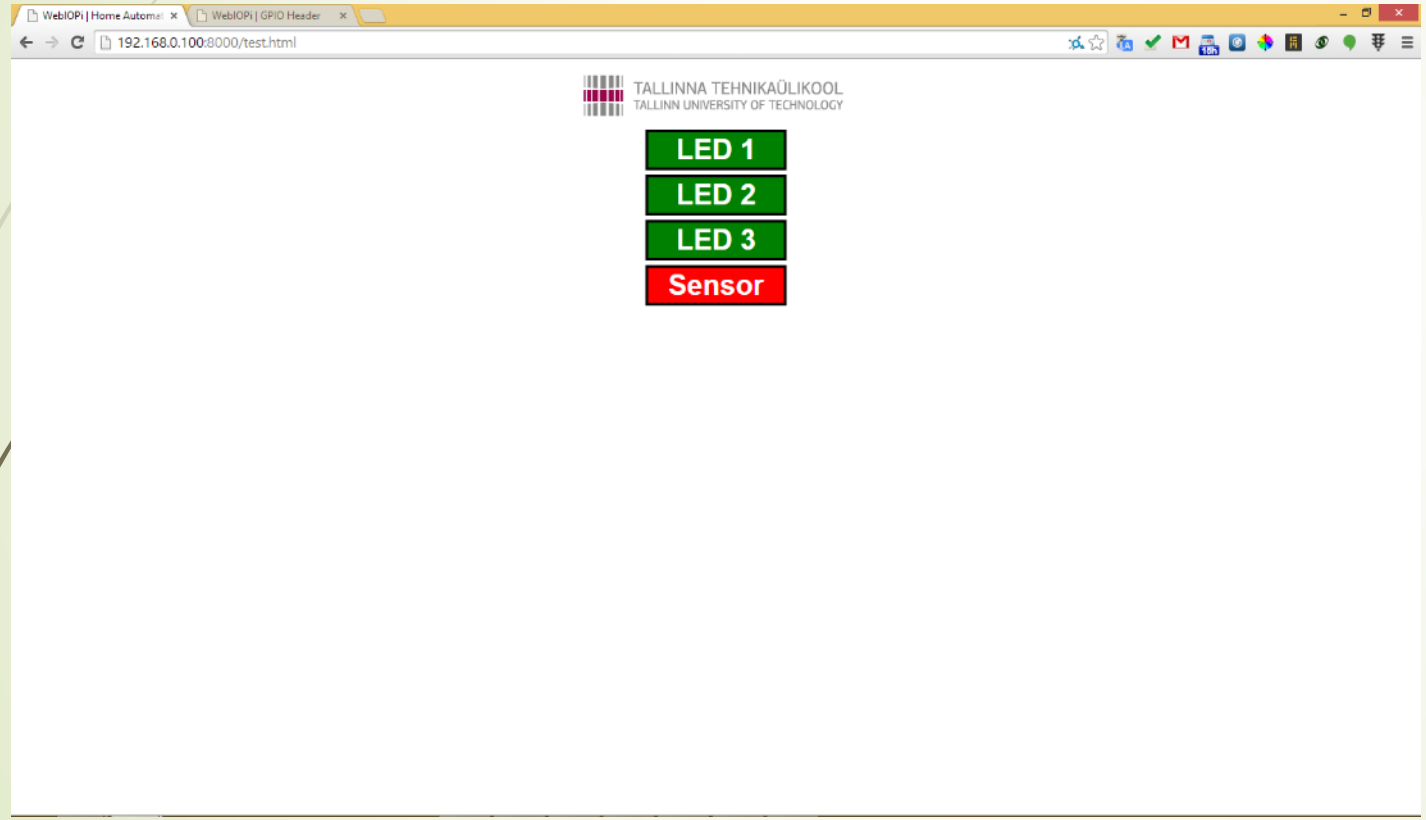
# Default UI


WebIOPi | GPIO Header
192.168.0.100:8000/app/gpio-header
3.3V 1 2 5.0V
OUT GPIO 2 3 4 5.0V
OUT GPIO 3 5 6 GROUND
OUT GPIO 4 7 8 UART TX
GROUND 9 10 UART RX
IN GPIO 17 11 12 GPIO 18 IN
IN GPIO 27 13 14 GROUND
IN GPIO 22 15 16 GPIO 23 IN
3.3V 17 18 GPIO 24 IN
OUT GPIO 10 19 20 GROUND
IN GPIO 9 21 22 GPIO 25 IN
IN GPIO 11 23 24 GPIO 8 IN
GROUND 25 26 GPIO 7 IN

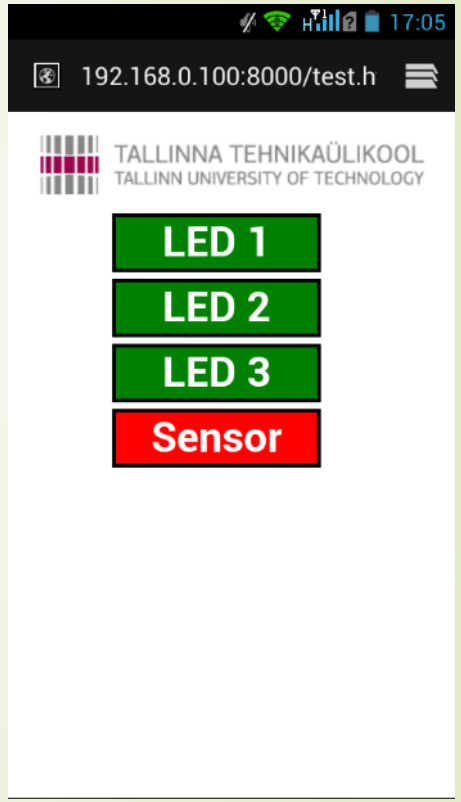
# Customisation of UI

```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
        <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
    <meta content="width=device-width, initial-scale=1, user-scalable = no " name="viewport" />
        <title>WebIOPi | Home Automation Controls</title>
        <script type="text/javascript" src="/webiopi.js"></script>
        <script type="text/javascript">
        webiopi().ready(function() {
                // Settup GPIO Pins
        webiopi().setFunction(2,"out");
        webiopi().setFunction(3,"out");
        webiopi().setFunction(4,"out");
        webiopi().setFunction(17,"in");

        // Create a button title "LED 1" labeled button for GPIO 2
                var button = webiopi().createGPIOButton(2, "LED 1");
        // Append button to HTML element with ID="controls" using jQuery
        $("#controls").append(button);

        // Create a button title "LED 2" labeled button for GPIO 3
        button = webiopi().createGPIOButton(3, "LED 2");
                // Append button to HTML element with ID="controls" using jQuery
        $("#controls").append(button);

        // Create a button title "LED 3" labeled button for GPIO 3
        button = webiopi().createGPIOButton(4, "LED 3");
                // Append button to HTML element with ID="controls" using jQuery
        $("#controls").append(button);

        // Create a status box title "Sensor" labeled button for GPIO 2
                var indicator = webiopi().createGPIOButton(17, "Sensor");
        // Append indicator to HTML element with ID="controls" using jQuery
        $("#controls").append(indicator);


        // Refresh GPIO buttons
                // pass true to refresh repeatedly of false to refresh once
```

# Custom UI




Desktop version




Mobile version

# Thank you

Project repository: https://github.com/ubaierbhat/webiopi-ttu