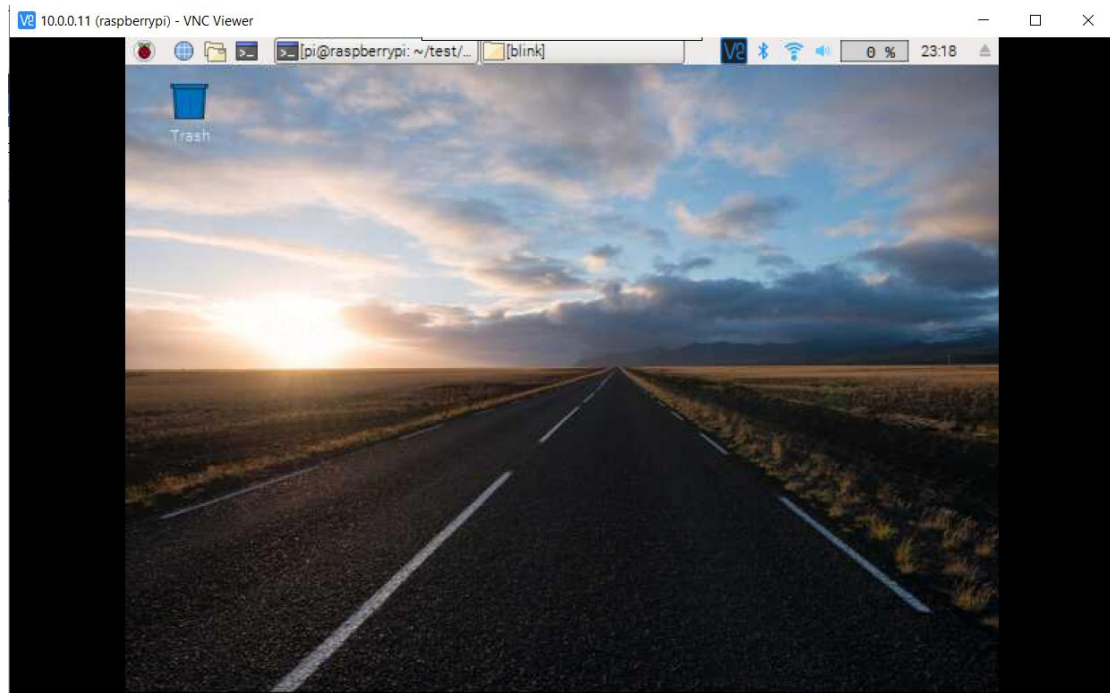


EE517 Internet of Things

Project: Using Websocket to continuously display Temperature

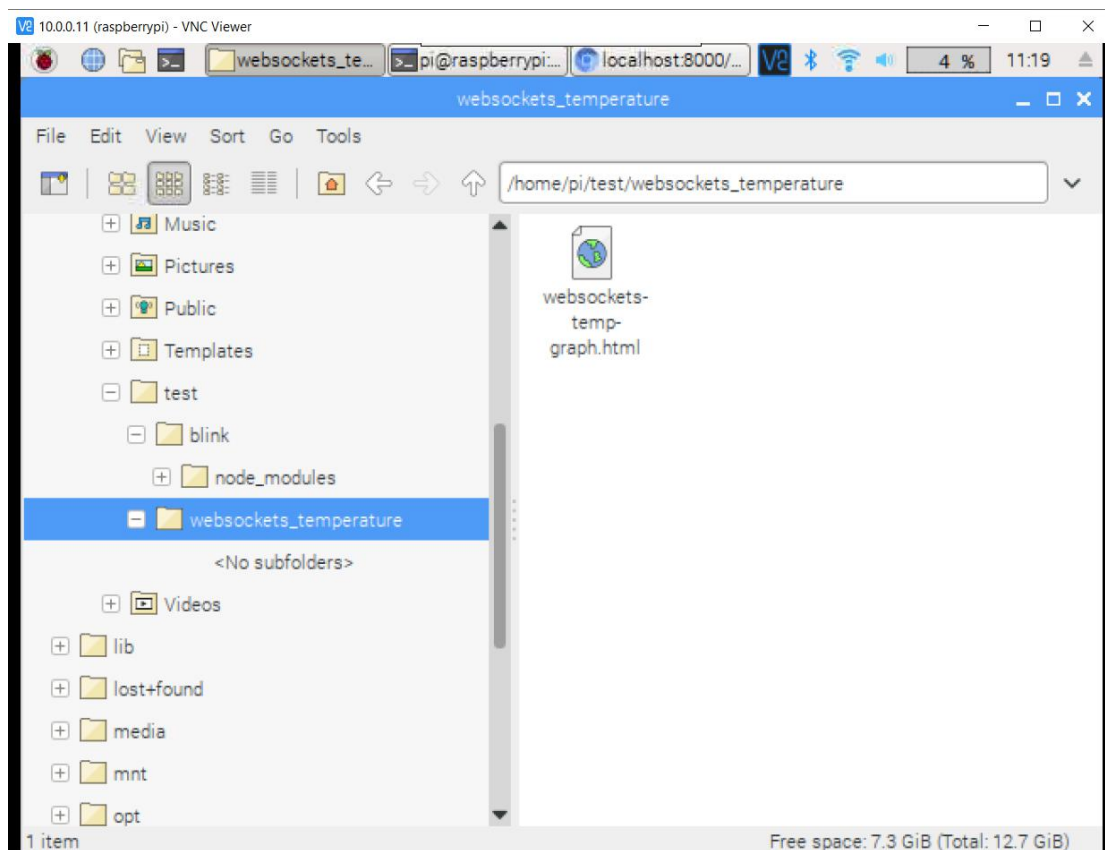
1. Access Raspberry Pi with your computer by using Putty or VNC. Here, I use VNC.



2. Make direction:

```
$ cd ~/test/websocket_temperature
```

3. Create a websockets-temp-graph.html file under your folder



websockets-temp-graph.html

```
<html>
```

```
<head>
```

```
  <script src="https://ajax.googleapis.com/ajax/libs/jquery/1.11.2/jquery.min.js"></script>
```

```
  <script type="text/javascript"
```

```
    src="https://www.google.com/jsapi?autoload={
```

```
      'modules':[{
```

```
        'name':'visualization',
```

```
        'version':'1',
```

```
        'packages':['corechart']
```

```
      ]
```

```
    }"></script>
```

```
</head>
```

```
<body>
```

```
<!-- The area to draw -->
```

```
<div id="chart" style="width: 900px; height: 500px"></div>
```

```
<script type="text/javascript">
```

```
$(document).ready(function () {
```

```
  var maxDataPoints = 10;
```

```
  var chart =
```

```

    // $('#chart') uses jQuery selectors
    new google.visualization.LineChart($('#chart')[0]);

// 2D array data used for drawing the Line Chart
var data =
    google.visualization.arrayToDataTable([
        ['Time', 'Temperature'],
        [getTime(), 0]
    ]);

// The style of the Line Chart
var options = {
    title: 'Temperature',
    curveType: 'function',
    animation: {
        duration: 1000,
        easing: 'in'
    },
    legend: {position: 'bottom'}
};

// Add data collected from WebSocket to the 2D array
function addDataPoint(dataPoint) {
    if (data.getNumberOfRows() > maxDataPoints) {
        data.removeRow(0);
    }
    data.addRow([getTime(), dataPoint.value]);

    chart.draw(data, options);
}

// Return the current time
function getTime() {
    var d = new Date();
    return d.toLocaleTimeString();
}
////////////////////////////////////
// Create a WebSocket subscription to the temperature sensor.
// Note that the URL uses the WebSockets protocol (ws://...)
var socket = new
    WebSocket('ws://devices.webofthings.io/pi/sensors/temperature');

// Register this anonymous function to be called when a message
// arrives on the WebSocket

```

```

socket.onmessage = function (event) {
    var result = JSON.parse(event.data);
    addDataPoint(result);
};

// Register this other anonymous function to be triggered when
// an error occurs on the WebSocket
socket.onerror = function (error) {
    console.log('WebSocket error!');
    console.log(error);
};

////////////////////////////////////

});

</script>
</body>
</html>

```

4. On your terminal, run: `$ python3 -m http.server --cgi`

```

10.0.0.11 (raspberrypi) - VNC Viewer
pi@raspberrypi: ~/test/websockets_temperature
pi@raspberrypi:~ $ cd ~/test/websockets_temperature
pi@raspberrypi:~/test/websockets_temperature $ python3 -m http.server --cgi
Serving HTTP on 0.0.0.0 port 8000 ...
127.0.0.1 - - [27/Feb/2019 10:57:01] "GET /websockets-temp-graph.html HTTP/1.1"
200 -
127.0.0.1 - - [27/Feb/2019 10:57:02] code 404, message File not found
127.0.0.1 - - [27/Feb/2019 10:57:02] "GET /favicon.ico HTTP/1.1" 404 -
127.0.0.1 - - [27/Feb/2019 10:57:16] "GET /websockets-temp-graph.html HTTP/1.1"
200 -
127.0.0.1 - - [27/Feb/2019 10:59:45] "GET /websockets-temp-graph.html HTTP/1.1"
200 -
127.0.0.1 - - [27/Feb/2019 11:00:19] "GET /websockets-temp-graph.html HTTP/1.1"
200 -
127.0.0.1 - - [27/Feb/2019 11:07:43] "GET /websockets-temp-graph.html HTTP/1.1"
200 -
127.0.0.1 - - [27/Feb/2019 11:17:49] "GET /websockets-temp-graph.html HTTP/1.1"
200 -
^C
Keyboard interrupt received, exiting.
pi@raspberrypi:~/test/websockets_temperature $ python3 -m http.server --cgi
Serving HTTP on 0.0.0.0 port 8000 ...

```

on the Raspberry Pi, open a browser and run:
<http://localhost:8000/websockets-temp-graph.html>

