

# 03-2 IoT

---

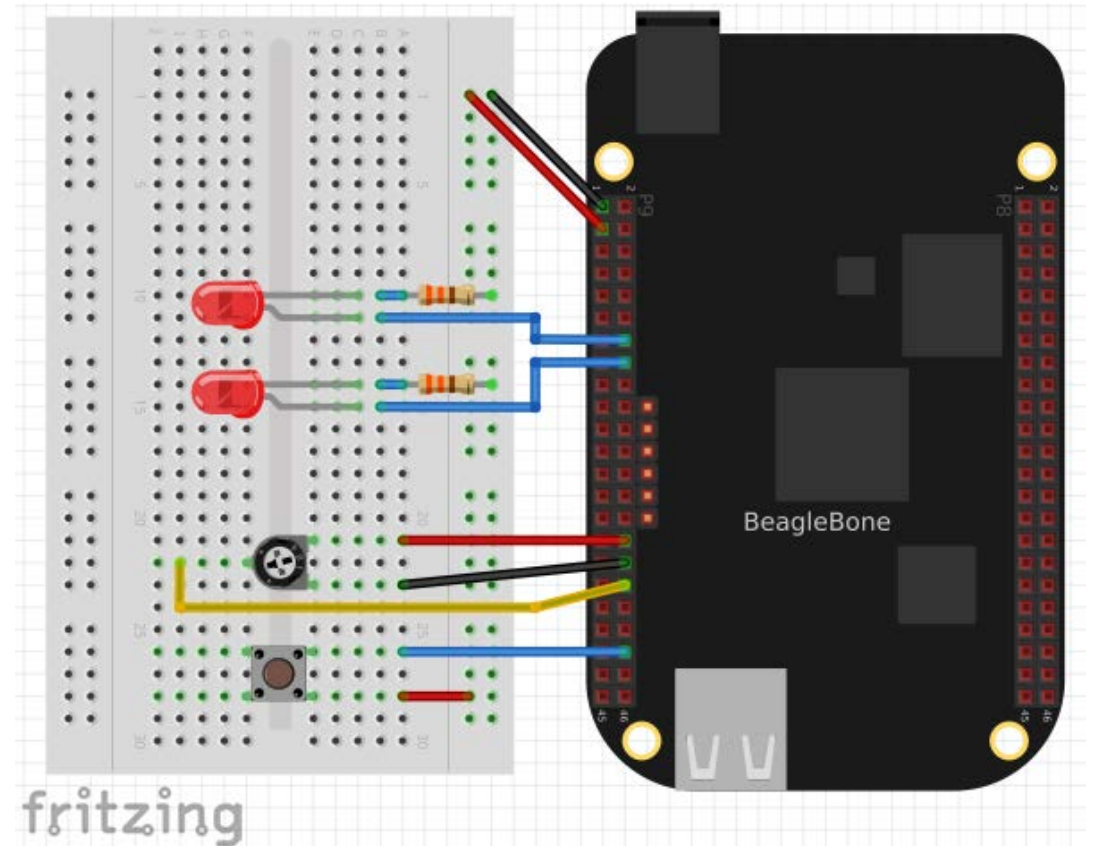
CONNECTING TO THE WEB

# Connecting physical to the web

Now that

- The Bone is on the network and
- You know how to read switches and blink LEDs

It's easy to have a web page read your Bone



# jsfiddle

Jsfiddle is a site for fiddling with your JavaScript

For example: <http://jsfiddle.net/3Pe6f/109/>

The screenshot shows the jsfiddle.net website interface. The browser address bar displays 'jsfiddle.net/3Pe6f/109/'. The website has a blue header with the 'JSFIDDLE' logo and navigation links: Run, Update, Fork, TidyUp, JSHint, Collaboration, and Share. A user profile 'yoder' is visible in the top right. On the left sidebar, there are sections for 'Frameworks & Extensions' (with jQuery 1.9.1 selected), 'Fiddle Options', 'External Resources', 'Languages', 'Ajax Requests', and 'Legal, Credits and Links'. The main content area is divided into three panels: HTML, CSS, and JavaScript. The HTML panel contains a simple structure with a title 'BoneScript jQuery Demo' and two paragraphs. The CSS panel is empty. The JavaScript panel contains the following code:

```
1 setTargetAddress('192.168.7.2', {
2   initialized: run
3 });
4
5 function run() {
6   var b = require('bonescript');
7   var SLIDER = 'P9_36';
8   var BUTTON = 'P9_18';
9   b.pinMode(BUTTON, b.INPUT);
10
11   getSliderStatus();
12
13   function getSliderStatus() {
14     b.analogRead(SLIDER, onSliderRead);
15   }
16
17   function onSliderRead(x) {
18     if (!x.err) {
19       $('#sliderStatus').html(x.value.toFixed(3));
20     }
21   }
22   getButtonStatus()
23
24   function getButtonStatus() {
25     b.digitalRead(BUTTON, onButtonRead);
26   }
27
28   function onButtonRead(x) {
29     if (!x.err) {
30       $('#buttonStatus').html(x.value);
31     }
32     setTimeout(getSliderStatus, 20);
33   }
34 }
```

The rightmost panel shows the 'Result' of the execution, displaying the text 'BoneScript jQuery Demo' and the status values 'sliderStatus = -' and 'buttonStatus = -'. At the bottom of the page, there is a Twitter follow button for '@jsfiddle' with 19.2K followers and a keyboard shortcuts link.

BoneScript jQuery Demo - x

jsfiddle.net/3Pe6f/109/

Apps ★ Bookmarks Mandi Courses Linux Home mp3 Rose SPEN Slickdeals Travel Other » Other bookmarks

JSFIDDLE α Run Update Fork TidyUp JSHint Collaboration Share yoder

Frameworks & Extensions

jQuery 1.9.1

☐ Migrate 1.1.0

☐ jQuery UI 1.9.2

☐ jQuery Mobile 1.3.0b1

☐ jQuery Mobile 1.2.0

onLoad

Fiddle Options

External Resources 1

Languages

Ajax Requests

Legal, Credits and Links

Follow @jsfiddle 19.2K followers

Keyboard shortcuts

HTML

```
1 <h1>BoneScript jQuery Demo</h1>
2
3 <p>sliderStatus = <span id="sliderStatus"></span>
4 </p>
5 <p>buttonStatus = <span id="buttonStatus"></span>
6 </p>
```

CSS

JavaScript

```
1 setTargetAddress('192.168.7.2', {
2   initialized: run
3 });
4
5 function run() {
6   var b = require('bonescript');
7   var SLIDER = 'P9_36';
8   var BUTTON = 'P9_18';
9   b.pinMode(BUTTON, b.INPUT);
10
11   getSliderStatus();
12
13   function getSliderStatus() {
14     b.analogRead(SLIDER, onSliderRead);
15   }
16
17   function onSliderRead(x) {
18     if (!x.err) {
19       $('#sliderStatus').html(x.value.toFixed(3));
20     }
21     getButtonStatus()
22   }
23
24   function getButtonStatus() {
25     b.digitalRead(BUTTON, onButtonRead);
26   }
27
28   function onButtonRead(x) {
29     if (!x.err) {
30       $('#buttonStatus').html(x.value);
31     }
32     setTimeout(getSliderStatus, 20);
33   }
34 }
```

Result

**BoneScript jQuery Demo**

sliderStatus = -

buttonStatus = -

Web page

JavaScript

# BoneScript jQuery Demo

html

```
<h1>BoneScript jQuery Demo</h1>
```

```
<p>
```

```
    sliderStatus = <span id="sliderStatus">-</span>
```

```
</p>
```

```
<p>
```

```
    buttonStatus = <span id="buttonStatus">-</span>
```

```
</p>
```

sliderStatus = -

buttonStatus = -

# js

---

```
setTargetAddress('192.168.7.2', {  
    initialized: run  
});
```

```
function run() {  
    var b = require('bonescript');  
    var SLIDER = 'P9_36';  
    var BUTTON = 'P9_18';  
    b.pinMode(BUTTON, b.INPUT);
```

# js

```
sliderStatus = <span id="sliderStatus">-</span>
buttonStatus = <span id="buttonStatus">-</span>
```

```
getSliderStatus();

function getSliderStatus() {
    b.analogRead(SLIDER, onSliderRead);
}

function onSliderRead(x) {
    if (!x.err) {
        $('#sliderStatus').html(x.value);
    }
    getButtonStatus()
}
```

```
function getButtonStatus() {
    b.digitalRead(BUTTON, onButtonRead);
}

function onButtonRead(x) {
    if (!x.err) {
        $('#buttonStatus').html(x.value);
    }
    setTimeout(getSliderStatus, 200);
}

} // End of run
```

# Hit Run

## Change

- SLIDER and
- BUTTON

## Then hit Run

- sliderStatus and
- buttonStatus

...will update

The screenshot shows the JSFiddle web application. The browser address bar displays `jsfiddle.net/3Pe6f/104/`. The JSFIDDLE toolbar includes buttons for Run, Update, Fork, TidyUp, JSHint, Collaboration, and Share. The left sidebar contains sections for Frameworks & Extensions (jQuery 1.9.1 selected), Fiddle Options, External Resources, Languages, Ajax Requests, and Legal, Credits and Links. The central editor has three tabs: HTML, JavaScript, and CSS. The HTML tab contains the following code:

```
<h1>BoneScript jQuery Demo</h1>
<p>sliderStatus = <span id="sliderStatus"></span>
</p>
<p>buttonStatus = <span id="buttonStatus"></span>
</p>
```

The JavaScript tab contains the following code:

```
setTargetAddress('beaglebone.local', {
  initialized: run
});
setTargetAddress('192.168.7.2', {
  initialized: run
});

function run() {
  var b = require('bonescript');
  var SLIDER = 'P9_36';
  var BUTTON = 'P9_18';
  b.pinMode(BUTTON, b.INPUT);

  getSliderStatus();

  function getSliderStatus() {
    b.analogRead(SLIDER, onSliderRead);
  }

  function onSliderRead(x) {
    if (!x.err) {
      $('#sliderStatus').html(x.value.toFixed(3));
    }
    getButtonStatus();
  }

  function getButtonStatus() {
    b.digitalRead(BUTTON, onButtonRead);
  }

  function onButtonRead(x) {
    if (!x.err) {
      $('#buttonStatus').html(x.value);
    }
  }
}
```

The right sidebar shows the rendered output under the heading "BoneScript jQuery Demo":

```
sliderStatus = 0.344
buttonStatus = 0
```

Green arrows indicate the workflow: one arrow points from the text "Hit Run" to the "Run" button, and another arrow points from the text "Then hit Run" to the JavaScript code editor.



# Move to Bone

---

You have to have an Internet connection to use JSFIDDLE

You can move your code to the Bone so the connection isn't needed.

Add the following to your html:

```
<h1>BoneScript  jQuery Demo</h1>
```

```
<p>
```

```
    sliderStatus = <span id="sliderStatus">-</span>
```

```
</p>
```

```
<p>
```

```
    buttonStatus = <span id="buttonStatus">-</span>
```

```
</p>
```

# Put in jsfiddleDemo.html

```
<html>
  <head>
    <title>BoneScript jsfiddle Demo</title>
    <script src="/static/jquery.js"></script>
    <script src="/static/bonescript.js"></script>
    <script src="jsfiddleDemo.js"></script>
  </head>
  <body>
    <h1>BoneScript jQuery Demo</h1>
    <p>sliderStatus = <span id="sliderStatus">-</span></p>
    <p>buttonStatus = <span id="buttonStatus">-</span></p>
  </body>
</html>
```

These files are already on the Bone

Put js in jsfiddleDemo.js

# Copy to Bone

---

Put `jsfiddleDemo.js` and `jsfiddleDemo.html` in `/var/lib/cloud9`

Browse to

<http://192.168.7.2/jsfiddleDemo.html>

# Add a button

---

Optional: Add a button to the web page that controls an LED

```
<button id="led0" onClick='led(0)'/>LED 0</button>
```

