

Day 2-1

Assignment:

Lab 02, Due Today

Homework 03, Due Tuesday (once written)

Today's Topics:

Analog In

PWM

Reading a Switch from a web page

- JSFIDDLE

03-2 IoT

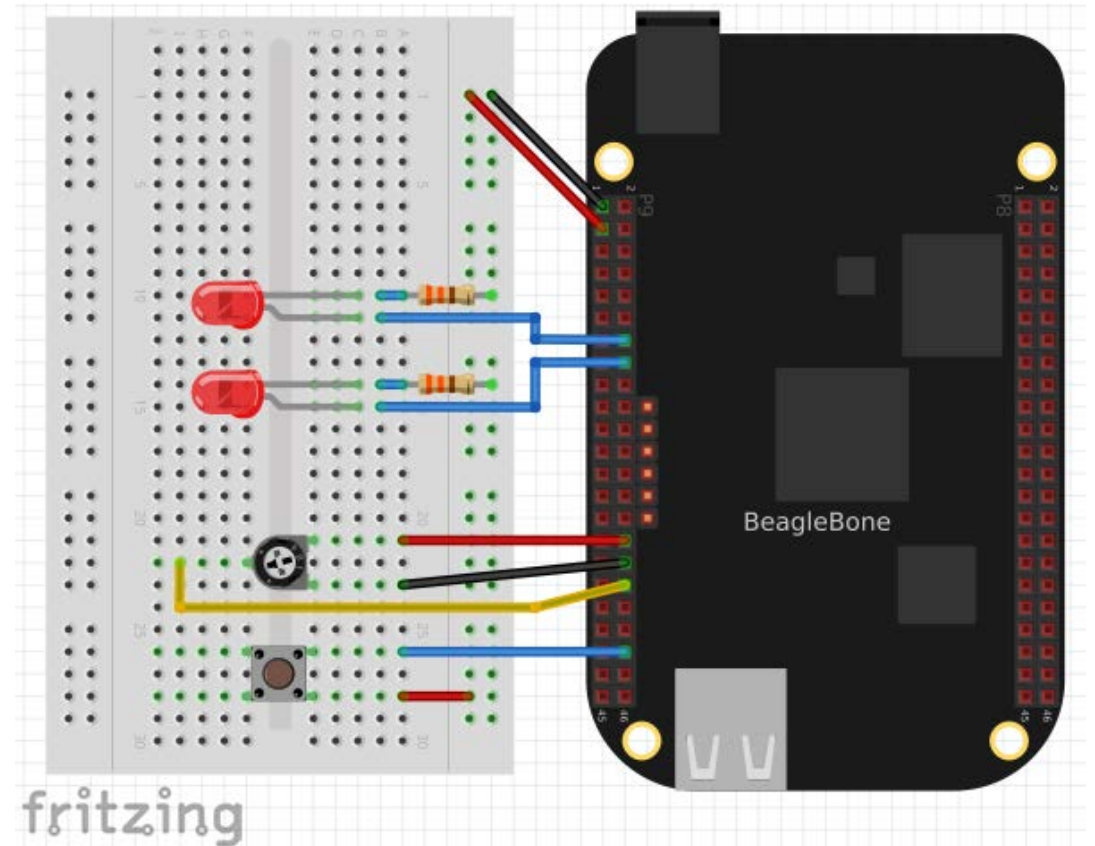
CONNECTING TO THE WEB

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 displays the jsfiddle.net interface for a demo titled "BoneScript jQuery Demo". The browser address bar shows the URL `jsfiddle.net/3Pe6f/109/`. The interface is divided into several sections:

- Frameworks & Extensions:** A sidebar on the left containing a dropdown menu with "jQuery 1.9.1" selected, and checkboxes for "Migrate 1.1.0", "jQuery UI 1.9.2", "jQuery Mobile 1.3.0b1", and "jQuery Mobile 1.2.0". Below this are sections for "Fiddle Options", "External Resources", "Languages", "Ajax Requests", and "Legal, Credits and Links".
- HTML:** A text area containing the following code:

```
<h1>BoneScript jQuery Demo</h1>
<p>sliderStatus = <span id="sliderStatus"></span>
</p>
<p>buttonStatus = <span id="buttonStatus"></span>
</p>
```
- CSS:** A text area on the right, currently empty.
- JavaScript:** A large text area in the center containing the following code:

```
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);
    }
  }

  setTimeout(getSliderStatus, 20);
}
```
- Result:** A section on the right showing the output of the code, which is:

```
sliderStatus = -
buttonStatus = -
```

At the bottom of the interface, there is a "Follow @jsfiddle" button indicating 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 interface for a project titled "BoneScript jQuery Demo". The left sidebar contains sections for "Frameworks & Extensions" (jQuery 1.9.1, Migrate 1.1.0, jQuery 1.9.2, jQuery Mobile 1.3.0b1, jQuery Mobile 1.2.0), "Fiddle Options" (onLoad), "External Resources", "Languages", "Ajax Requests", and "Legal, Credits and Links". The central code editor has three tabs: "HTML", "JavaScript", and "CSS". The "HTML" tab shows the following code:

```
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>
```

The "JavaScript" tab shows the following code:

```
1 setTargetAddress('beaglebone.local', {
2   initialized: run
3 });
4 setTargetAddress('192.168.7.2', {
5   initialized: run
6 });
7
8 function run() {
9   var b = require('bonescript');
10  var SLIDER = 'P9_36';
11  var BUTTON = 'P9_18';
12  b.pinMode(BUTTON, b.INPUT);
13
14  getSliderStatus();
15
16  function getSliderStatus() {
17    b.analogRead(SLIDER, onSliderRead);
18  }
19
20  function onSliderRead(x) {
21    if (!x.err) {
22      $('#sliderStatus').html(x.value.toFixed(3));
23    }
24    getButtonStatus()
25  }
26
27  function getButtonStatus() {
28    b.digitalRead(BUTTON, onButtonRead);
29  }
30
31  function onButtonRead(x) {
32    if (!x.err) {
33      $('#buttonStatus').html(x.value);
34    }
35  }
36 }
```

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 "Run" button in the top toolbar to the "Run" button, and another arrow points from the "Then hit Run" text to the JavaScript code in the central 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>
```

