BeagleBone Cookbook Webinar Series
Recipe #2
Displaying GPIO status in a Web Browser

November 3, 2015
Jason Kridner
Co-author of BeagleBone Cookbook
Board member at BeagleBoard.org Foundation
Sitara Applications Engineering at Texas Instruments
BeagleBone Black
Ready to explore and use in minutes

Truly flexible open hardware and software development platform

All you need is in the box

Proven ecosystem from prototype to product

BeagleBone Black – the most flexible solution in open-source computing

• Ready to use
  • USB client network
  • Built-in tutorials
  • Browser based IDE
  • Fla$$ed w/Debian
• Fast and flexible
  • 1-GHz Sitara ARM
  • 2x200-MHz PRUs
  • 512-MB DDR3
  • On-board HDMI
  • 65 digital I/O
  • 7 analog inputs
• Support for numerous Cape plug-in boards

http://beaglebone-akes.com

~$50
BeagleBone Black board features

10/100 Ethernet

USB Host
Easily connects to almost any everyday device such as mouse or keyboard

microHDMI
Connect directly to monitors and TVs

microSD
Expansion slot for additional storage

512MB DDR3
Faster, lower power RAM for enhanced user-friendly experience

Boot Button

Serial Debug

DC Power

Reset Button

Power Button

LEDS

USB Client
Development interface and directly powers board from PC

USB Host

Expansion headers
Enable cape hardware and include:
- 65 digital I/O
- 7 analog
- 4 serial
- 2 SPI
- 2 I2C
- 8 PWMs
- 4 timers
- And much much more!

1-GHz Sitara AM335x ARM® Cortex™-A8 processor
Provides a more advanced user interface and up to 150% better performance than ARM11

Money saving extras:
- Power over USB
- Included USB cable
- 4-GB on-board storage
- Built-in PRU microcontrollers

Power Button

4-GB on-board storage using eMMC
- Pre-loaded with Debian Linux Distribution
- 8-bit bus accelerates performance
- Frees the microSD slot to be used for additional storage for a less expensive solution than SD cards
Simple browser-based interactions

http://beagleboard.github.io/bone101
Cloud9 IDE hosted locally
Zero install and exposes command-line
10,000s of developers building connected devices today

- Medical analysis, assistance and information management
- Home information, automation and security systems
- Home and mobile entertainment and educational systems
- New types of communications systems
- Personal robotic devices for cleaning, upkeep and manufacturing
- Remote presence and monitoring
- Automotive information management and control systems
- Personal environmental exploration and monitoring
BeagleBone Cookbook
http://beagleboard.org/cookbook

- 99 recipes covering
  - Basics
  - Sensors
  - Displays and outputs
  - Motors
  - Internet of things
  - Kernel
  - Real-time I/O
  - Capes
Prerequisites

• Connect to the board per recipe 1.2
  – [http://beagleboard.org/getting-started](http://beagleboard.org/getting-started)

• Verify the software image per recipe 1.3 and potentially updating per recipe 1.9
Connect a button to GPIO P8_19
http://beagleboard.org/Support/bone101/#headers
Recipe 6.6: Continuously Displaying the GPIO Value


```html
<html>
<head>
  <title>BoneScript jQuery Demo</title>
  <script src="/static/jquery.js"></script>
  <script src="/static/bonescript.js"></script>
  <script src="/static/jqueryDemo.js"></script>
</head>

<body>
  <h1>BoneScript jQuery Demo</h1>
  <p>buttonStatus = <span id="buttonStatus">-</span></p>
</body>
</html>
```

https://github.com/BeagleBoneCookbook/firstEdition/blob/master/06iot/jQueryDemo.js

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

function run() {
    var b = require('bonescript');
    b.pinMode('P8_19', b.INPUT);
    getButtonStatus();
    function getButtonStatus() {
        b.digitalRead('P8_19', onButtonRead);
    }
    function onButtonRead(x) {
        $('#buttonStatus').html(x.value);
        setTimeout(getButtonStatus, 20);
    }
}
```
Stepping back to recipe 6.3
Interacting with the Bone via a Web Browser

https://github.com/BeagleBoneCookbook/firstEdition/blob/master/06iot/server.js

```javascript
var port=9090, h=require('http'),
    u=require('url'), f=require('fs');
var s=h.createServer(servePage);
s.listen(port);

function servePage(req, res) {
    var p = u.parse(req.url).pathname;
    f.readFile(__dirname+p,
        function (err, data) {
            if (err) return;
            res.write(data, 'utf8');
            res.end();
        });
}
```

- BeagleBone Black ships with Debian and Node.JS
- Using Node.JS is easy to serve up a simple web page
- Run with: node server.js
- Browse to port 9090 and a local file
Recipe 6.4 adds hardware interaction
https://github.com/BeagleBoneCookbook/firstEdition/blob/master/06iot/GPIOserver.js

```javascript
var h=require('http'), f=require('fs'),
b=require('bonescript'),
g='P8_19', p=9090;

var htmlStart = "<!DOCTYPE html>\n<html><body><h1>" + g + "</h1>data = ";
var htmlEnd = "</body></html>";
var s = h.createServer(servePage);

b.pinMode(g, b.INPUT);
s.listen(p);

function servePage(req, res) {
  var data = b.digitalRead(g);
  res.write(htmlStart + data + htmlEnd, 'utf8');
  res.end();
}

• Builds on simple Node.JS web server
• BoneScript library utilized on server
• Content served using variables, not files
• Full example uses URL path
  – distinguish content
• Refresh manually
```
Recipe 6.5 introduces jQuery

http://jsfiddle.net/n5j3p32o/1/

- Great tool to make content dynamic
- jsfiddle.net provides a playground for learning
- Learn more about the API at jquery.com
How BoneScript works in the browser
http://beagleboard.org/static/bonescript.js

• Provides a `setTargetAddress()` function to define the global require() function

• Utilizes the built-in Node.JS based web server built into the BeagleBone Black default image
  https://github.com/jadonk/bonescript/blob/master/src/server.js

• On-board bonescript.js provides the require() function and utilizes socket.io to define remote procedure calls
  https://github.com/jadonk/bonescript/blob/master/src/bonescript.js
Connect a potentiometer to ADC P9_36

http://beagleboard.org/Support/bone101/#headers
Recipe 6.7: Plotting Data

• See demo code at

• This is just the beginning
  – Lots of different types of hardware interactions
  – Lots of different visualizations possible in the browser
More

• JavaScript tricks
  – http://beagleboard.org/project/javascript-tricks/

• Shortcuts to updates and examples from the book
  – http://beagleboard.org/cookbook