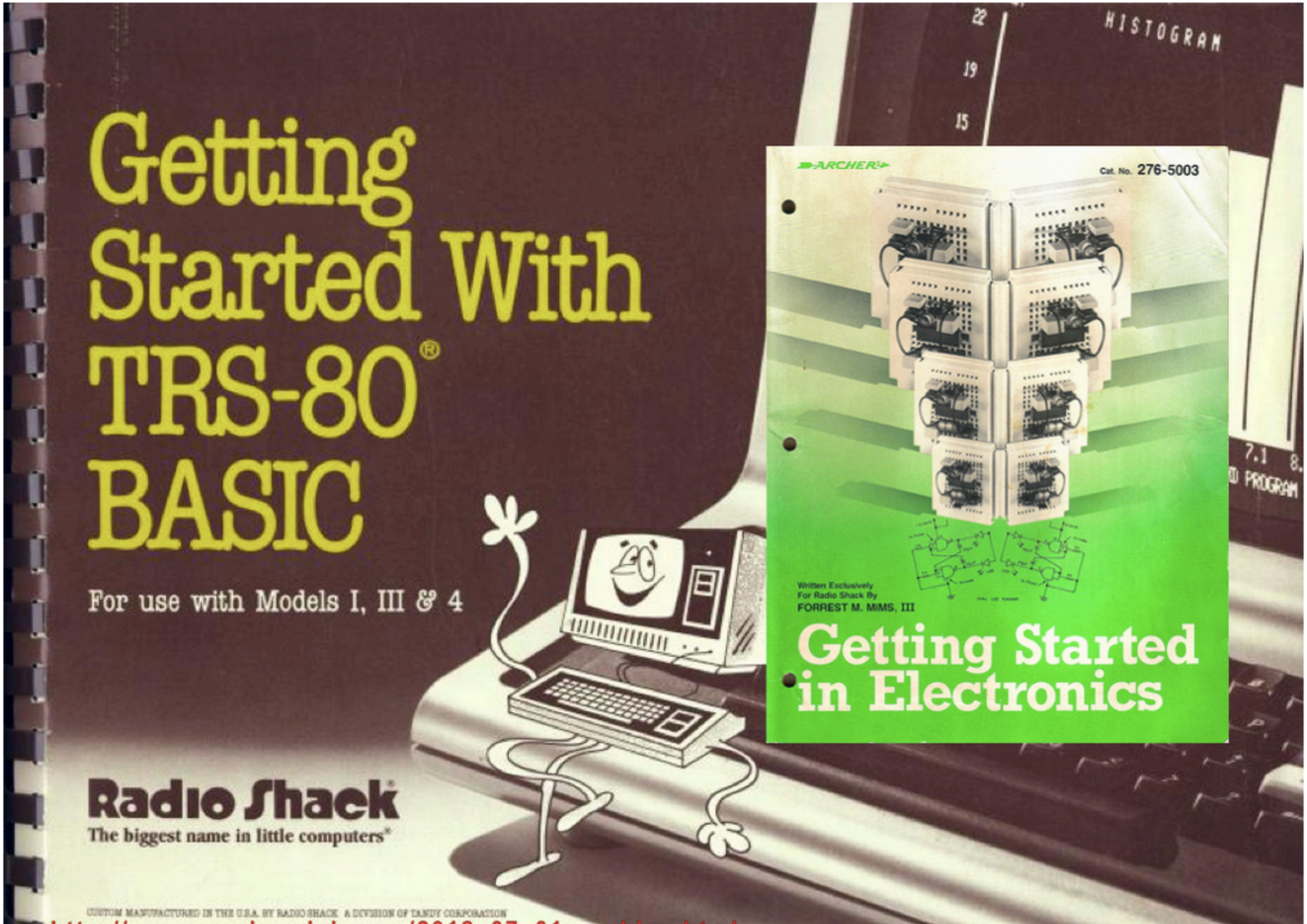
BeagleBoard.org

Linux and web servers for teaching electronics

Jason Kridner

Co-Founder

BeagleBoard.org



Getting Started With TRS-80® BASIC

For use with Models I, III & 4

Radio Shack®

The biggest name in little computers®

CUSTOM MANUFACTURED IN THE U.S.A. BY RADIO SHACK A DIVISION OF TANDY CORPORATION

ARCHER®

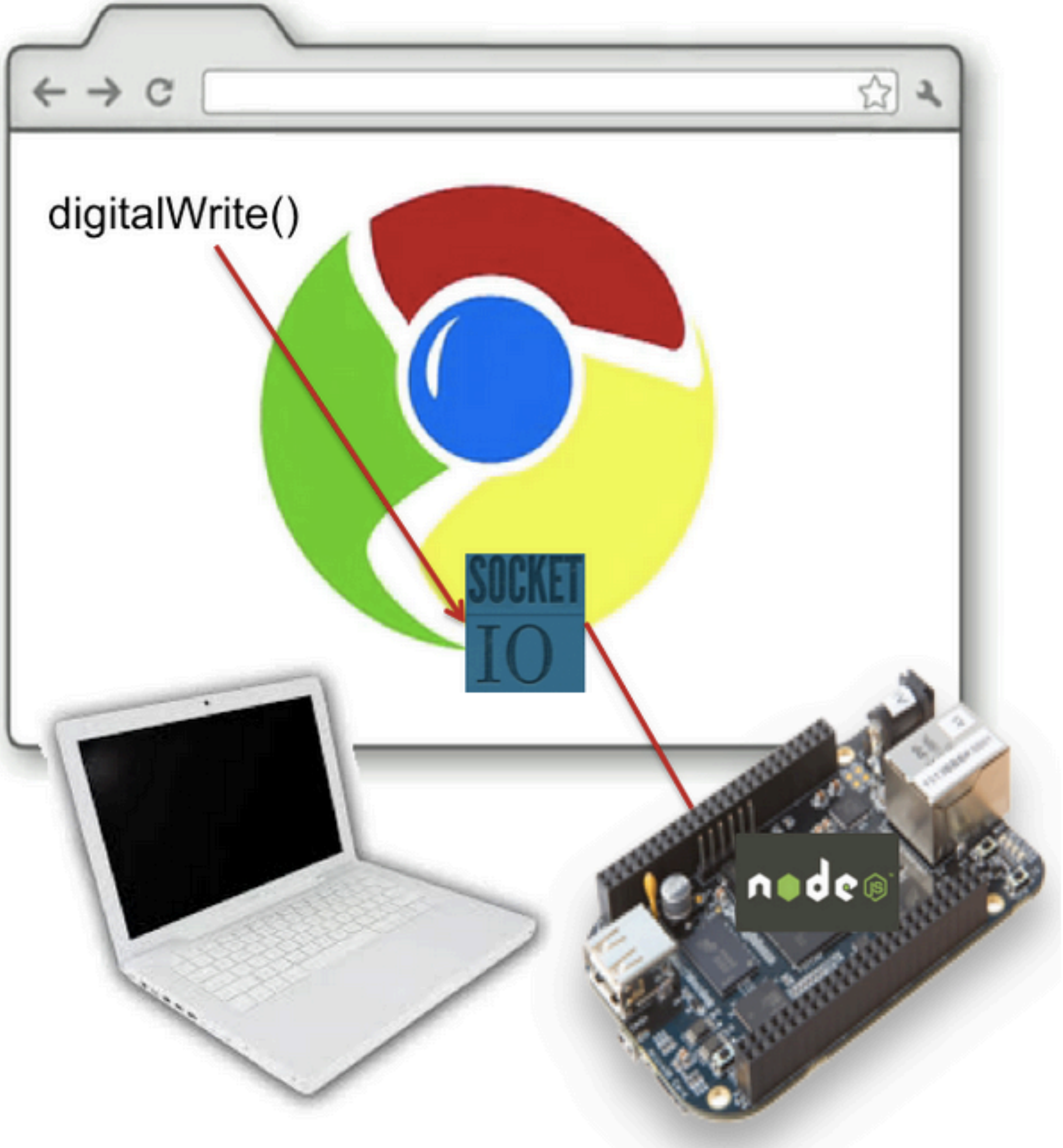
Cat. No. 276-5003

Written Exclusively
For Radio Shack By
FORREST M. MIMS, III

Getting Started in Electronics

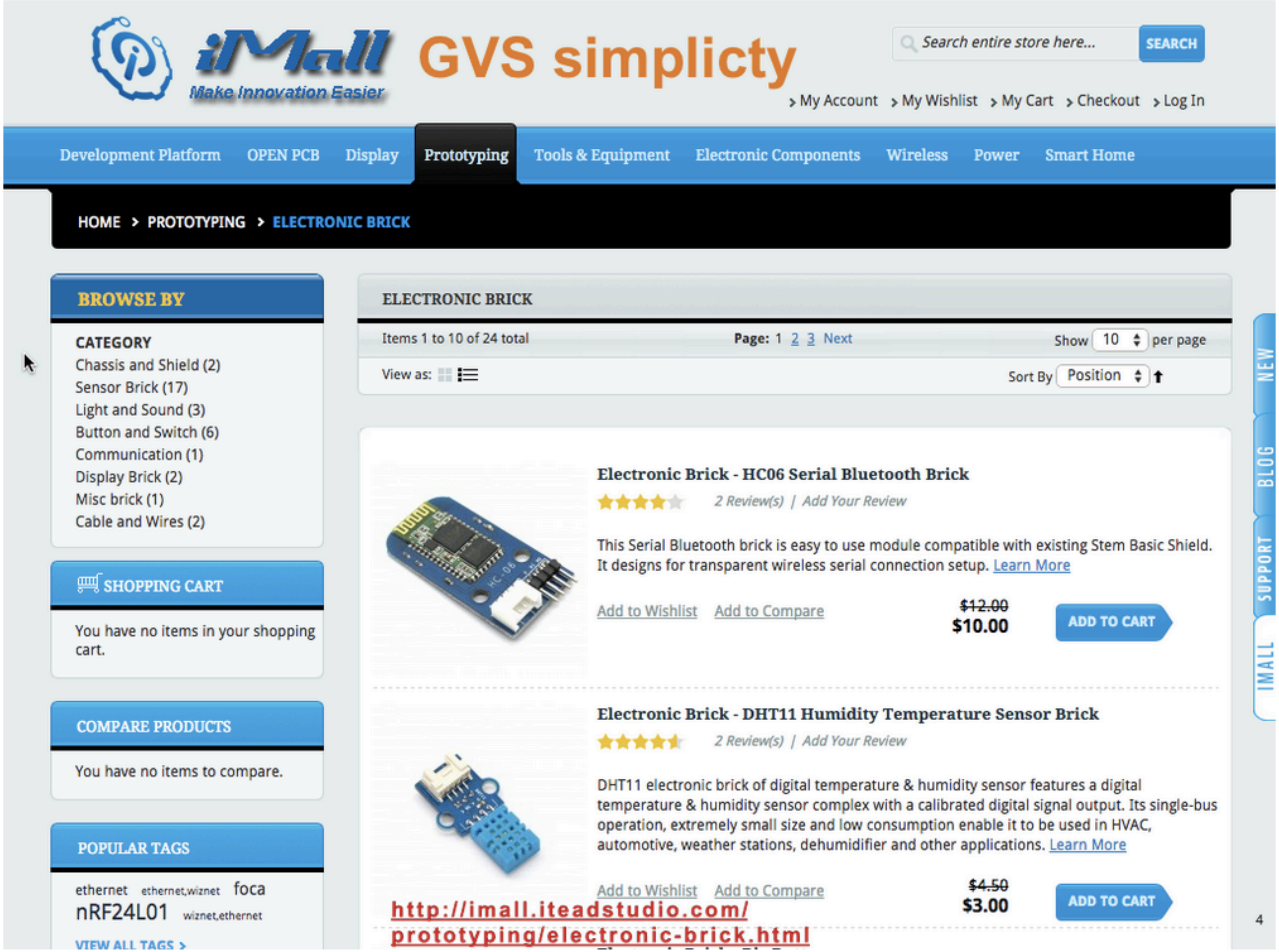
http://www.sandywalsh.com/2012_07_01_archive.html

http://newcome.wordpress.com/2009/12/15/make-electronics-the-new-engineers-notebook/


digitalWrite()

http://www.adafruit.com/

beagleboard.org



GVS simplicty

HOME > PROTOTYPING > ELECTRONIC BRICK

BROWSE BY

CATEGORY

- Chassis and Shield (2)
- Sensor Brick (17)
- Light and Sound (3)
- Button and Switch (6)
- Communication (1)
- Display Brick (2)
- Misc brick (1)
- Cable and Wires (2)

SHOPPING CART

You have no items in your shopping cart.

COMPARE PRODUCTS

You have no items to compare.

POPULAR TAGS

ethernet ethernet,wiznet foca nRF24L01 wiznet,ethernet

VIEW ALL TAGS >

ELECTRONIC BRICK

Items 1 to 10 of 24 total   Page: 1 2 3 Next   Show 10 per page

View as: Sort By Position

Electronic Brick - HC06 Serial Bluetooth Brick

★★★★☆ *2 Review(s) | Add Your Review*

This Serial Bluetooth brick is easy to use module compatible with existing Stem Basic Shield. It designs for transparent wireless serial connection setup. Learn More

Add to Wishlist   Add to Compare   ~~$12.00~~ $10.00   ADD TO CART

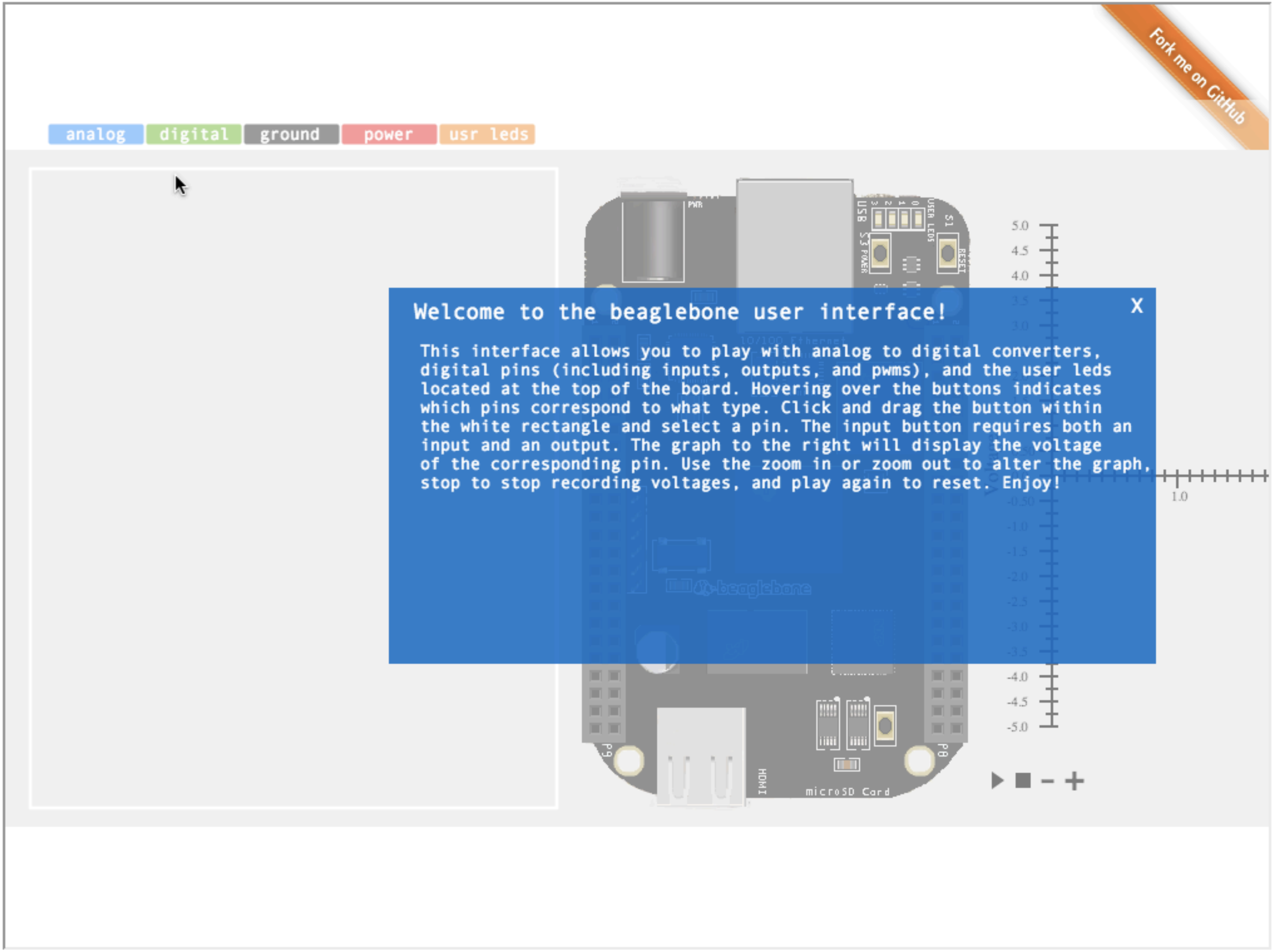
Electronic Brick - DHT11 Humidity Temperature Sensor Brick

★★★★★ *2 Review(s) | Add Your Review*

DHT11 electronic brick of digital temperature & humidity sensor features a digital temperature & humidity sensor complex with a calibrated digital signal output. Its single-bus operation, extremely small size and low consumption enable it to be used in HVAC, automotive, weather stations, dehumidifier and other applications. Learn More

Add to Wishlist   Add to Compare   ~~$4.50~~ $3.00   ADD TO CART

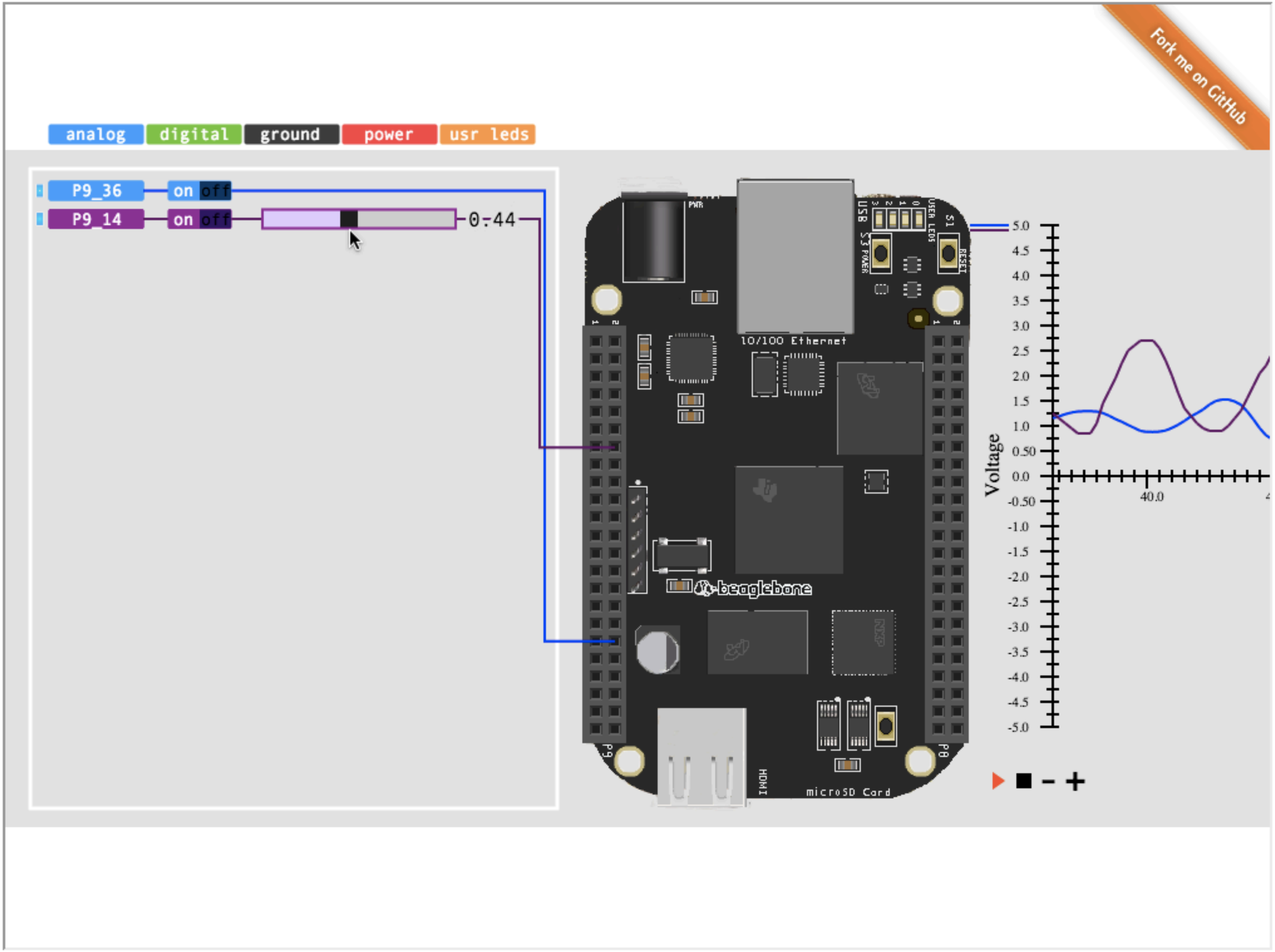
http://imall.iteadstudio.com/prototyping/electronic-brick.html

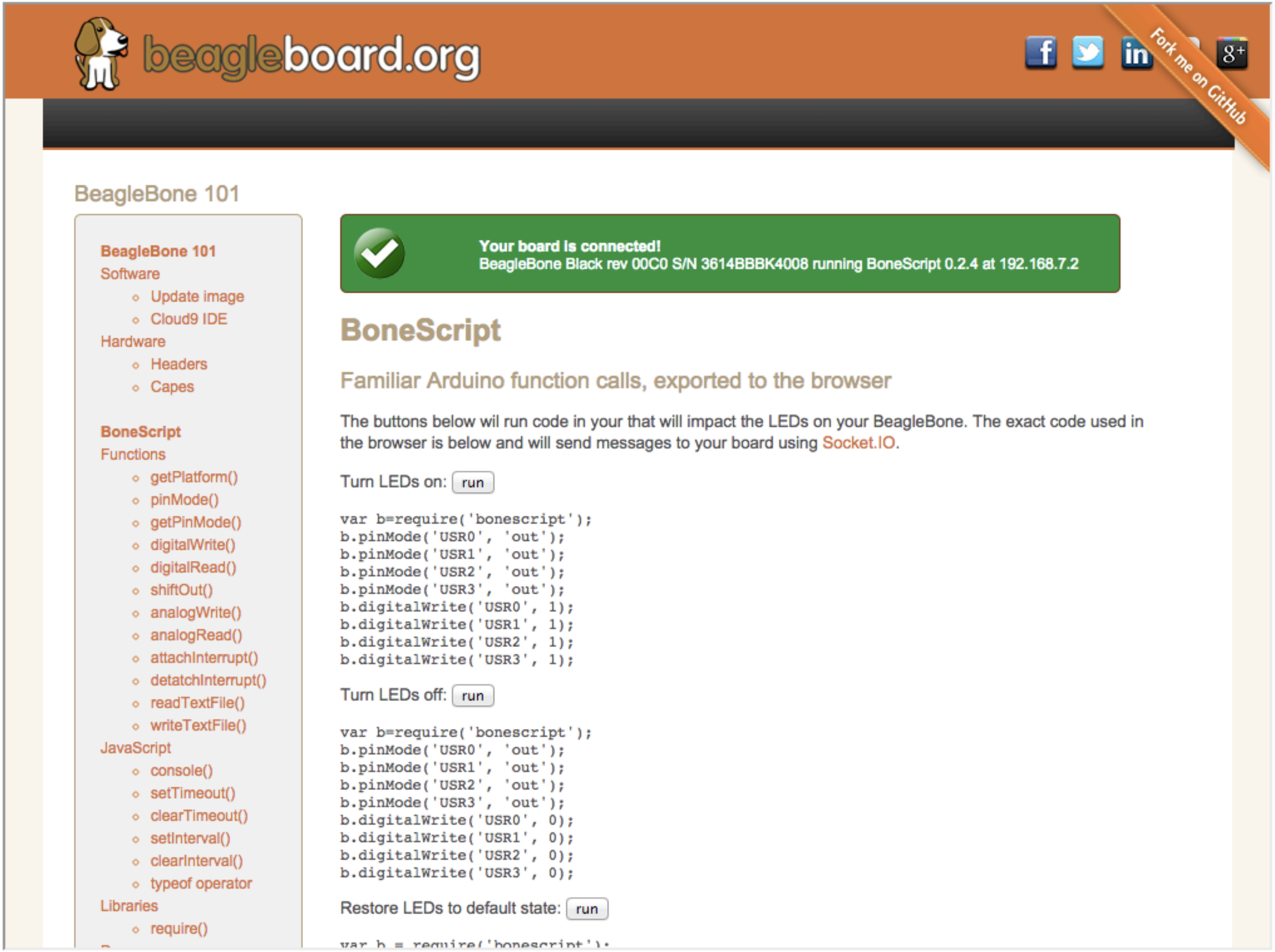


Welcome to the beaglebone user interface!

X

This interface allows you to play with analog to digital converters, digital pins (including inputs, outputs, and pwms), and the user leds located at the top of the board. Hovering over the buttons indicates which pins correspond to what type. Click and drag the button within the white rectangle and select a pin. The input button requires both an input and an output. The graph to the right will display the voltage of the corresponding pin. Use the zoom in or zoom out to alter the graph, stop to stop recording voltages, and play again to reset. Enjoy!


Fork me on GitHub
analog
digital
ground
power
usr leds
P9_36
on off
P9_14
on off
0.44
PWR
USB
3 2 1 0
USER LEDS
S1
S3 POWER
RESET
10/100 Ethernet
beaglebone
P9
P8
HDMI
microSD Card
5.0
4.5
4.0
3.5
3.0
2.5
2.0
1.5
1.0
0.50
0.0
-0.50
-1.0
-1.5
-2.0
-2.5
-3.0
-3.5
-4.0
-4.5
-5.0
Voltage
40.0
▶ ■ − +



Your board is connected!
BeagleBone Black rev 00C0 S/N 3614BBBK4008 running BoneScript 0.2.4 at 192.168.7.2

BoneScript

Familiar Arduino function calls, exported to the browser

The buttons below wil run code in your that will impact the LEDs on your BeagleBone. The exact code used in the browser is below and will send messages to your board using Socket.IO.

Turn LEDs on: run

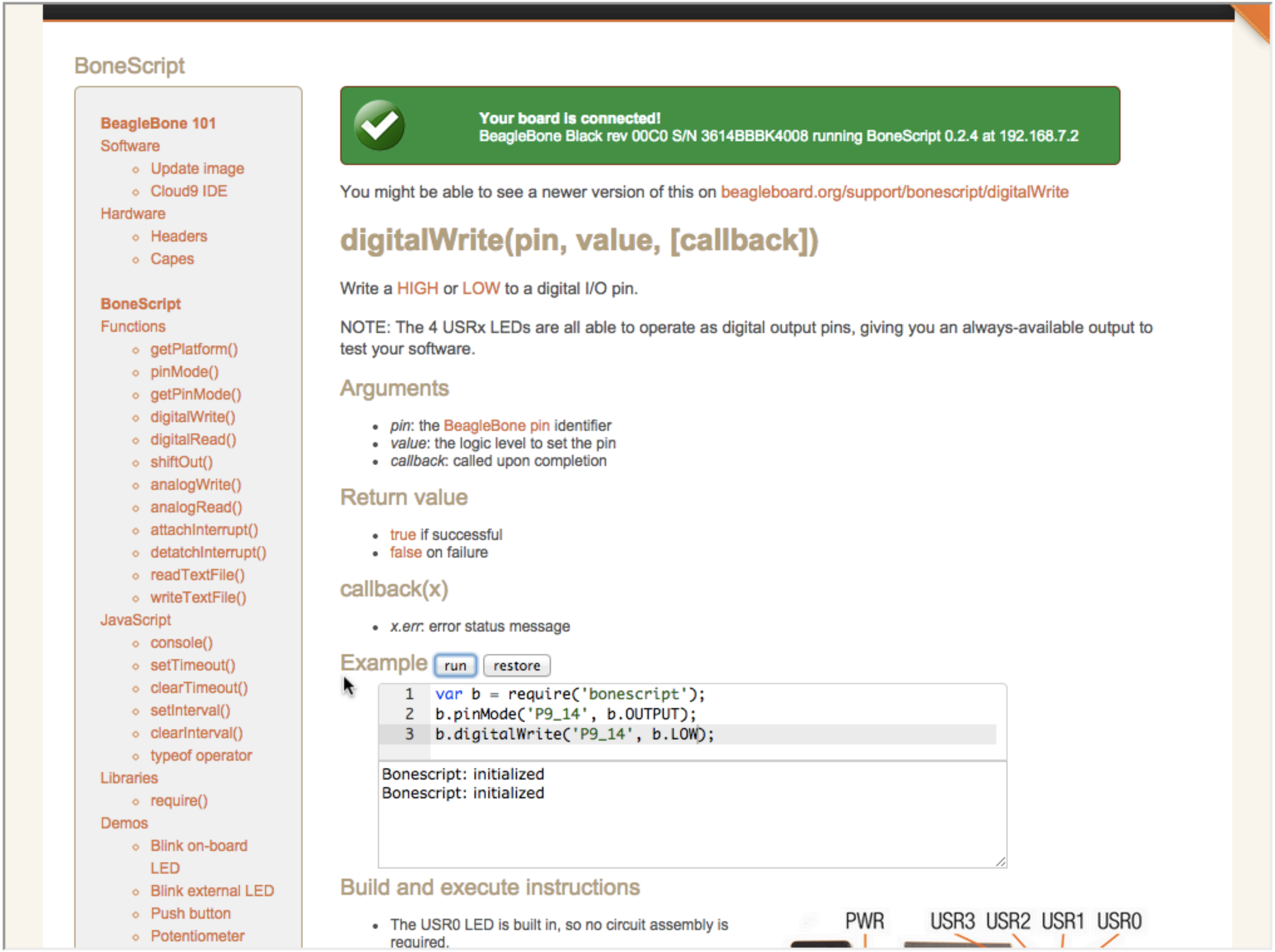
```
var b=require('bonescript');
b.pinMode('USR0', 'out');
b.pinMode('USR1', 'out');
b.pinMode('USR2', 'out');
b.pinMode('USR3', 'out');
b.digitalWrite('USR0', 1);
b.digitalWrite('USR1', 1);
b.digitalWrite('USR2', 1);
b.digitalWrite('USR3', 1);
```

Turn LEDs off: run

```
var b=require('bonescript');
b.pinMode('USR0', 'out');
b.pinMode('USR1', 'out');
b.pinMode('USR2', 'out');
b.pinMode('USR3', 'out');
b.digitalWrite('USR0', 0);
b.digitalWrite('USR1', 0);
b.digitalWrite('USR2', 0);
b.digitalWrite('USR3', 0);
```

Restore LEDs to default state: run

```
var b = require('bonescript');
```



BoneScript

BeagleBone 101

Software

- Update image
- Cloud9 IDE

Hardware

- Headers
- Capes

BoneScript

Functions

- getPlatform()
- pinMode()
- getPinMode()
- digitalWrite()
- digitalRead()
- shiftOut()
- analogWrite()
- analogRead()
- attachInterrupt()
- detatchInterrupt()
- readTextFile()
- writeTextFile()

JavaScript

- console()
- setTimeout()
- clearTimeout()
- setInterval()
- clearInterval()
- typeof operator

Libraries

- require()

Demos

- Blink on-board LED
- Blink external LED
- Push button
- Potentiometer

Your board is connected!
BeagleBone Black rev 00C0 S/N 3614BBBK4008 running BoneScript 0.2.4 at 192.168.7.2

You might be able to see a newer version of this on beagleboard.org/support/bonescript/digitalWrite

digitalWrite(pin, value, [callback])

Write a HIGH or LOW to a digital I/O pin.

NOTE: The 4 USRx LEDs are all able to operate as digital output pins, giving you an always-available output to test your software.

Arguments

- *pin*: the BeagleBone pin identifier
- *value*: the logic level to set the pin
- *callback*: called upon completion

Return value

- true if successful
- false on failure

callback(x)

- *x.err*: error status message

Example

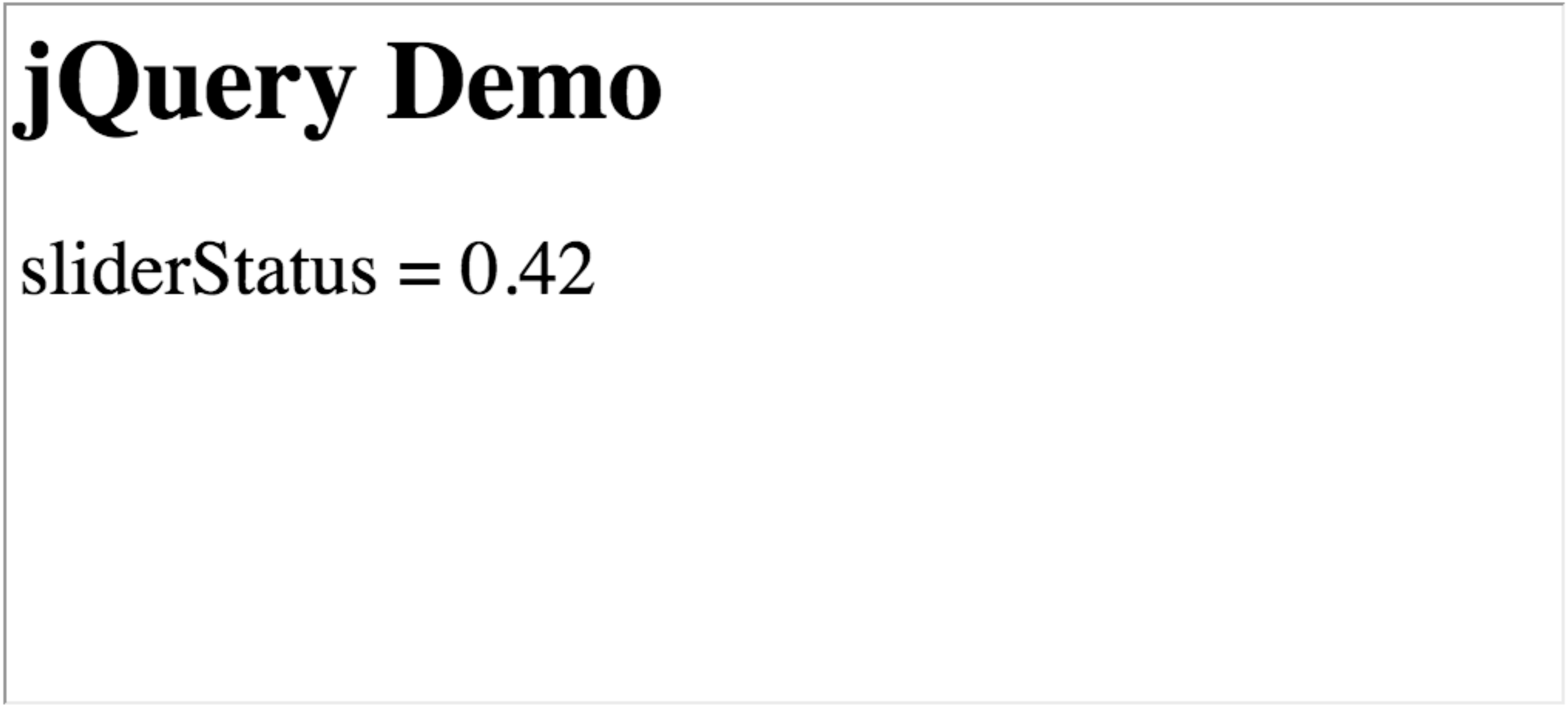
```
var b = require('bonescript');
b.pinMode('P9_14', b.OUTPUT);
b.digitalWrite('P9_14', b.LOW);
```

```
Bonescript: initialized
Bonescript: initialized
```

Build and execute instructions

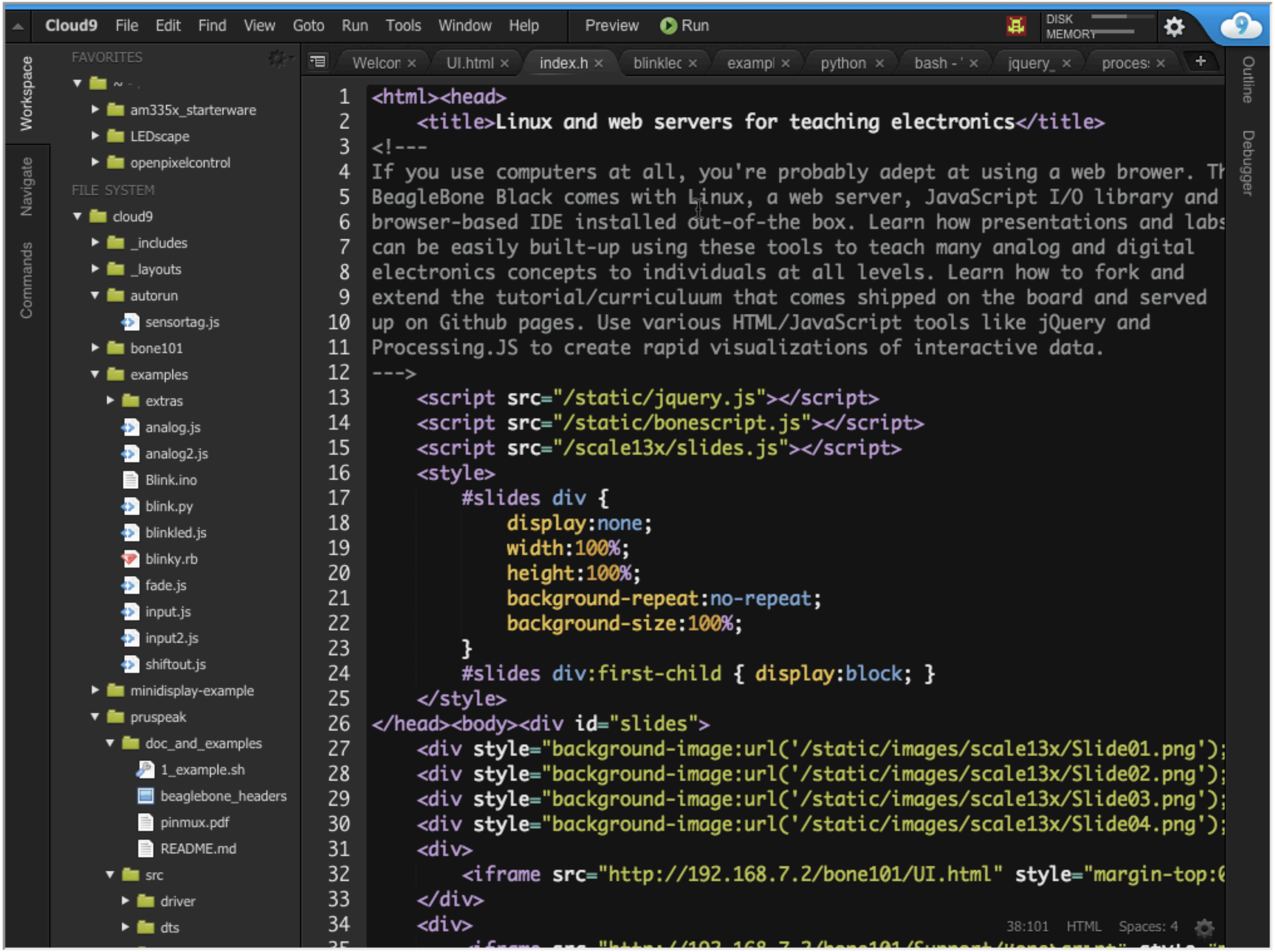
- The USR0 LED is built in, so no circuit assembly is required.

Bring in the web interactions

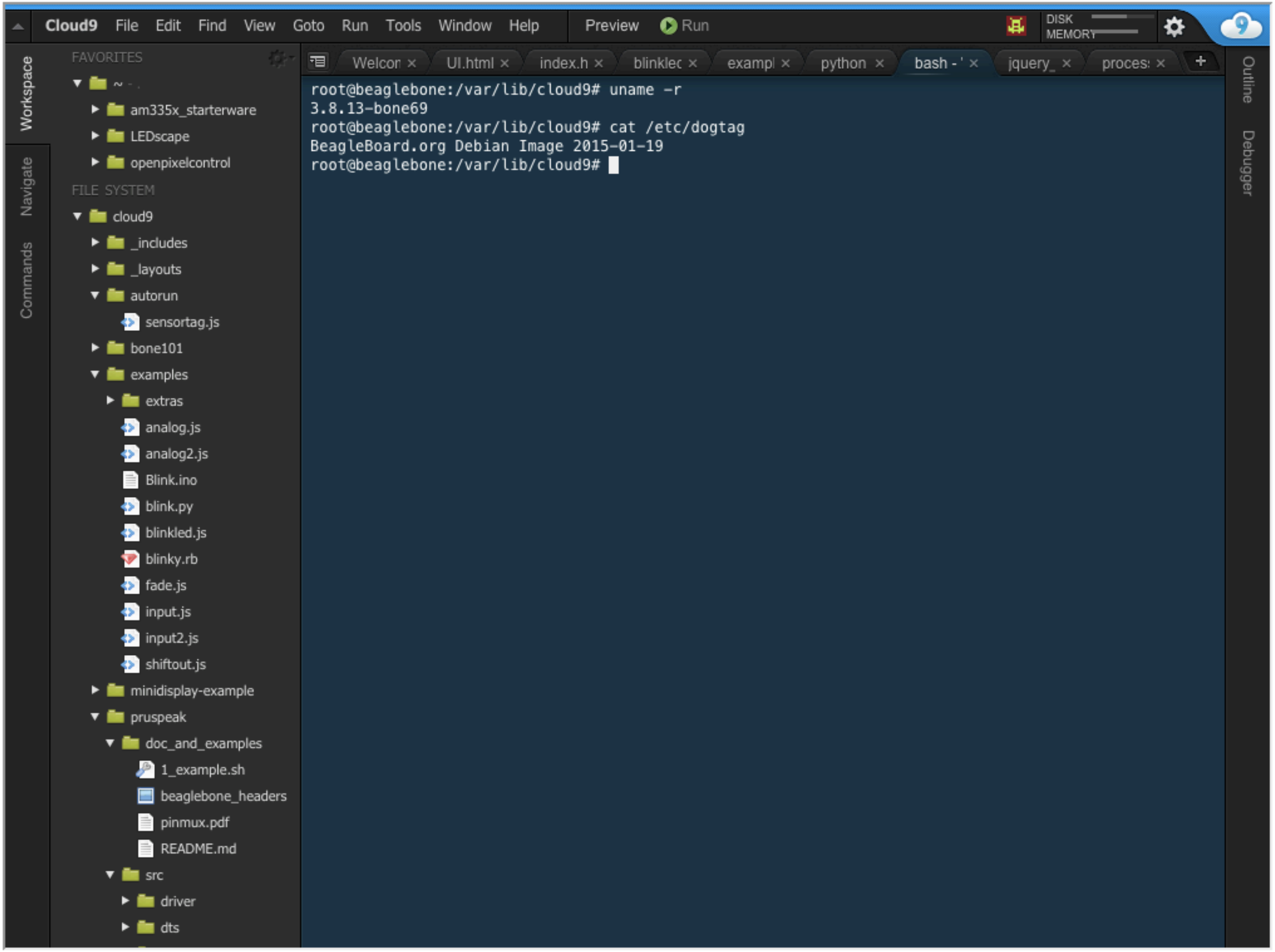


jQuery Demo

sliderStatus = 0.42

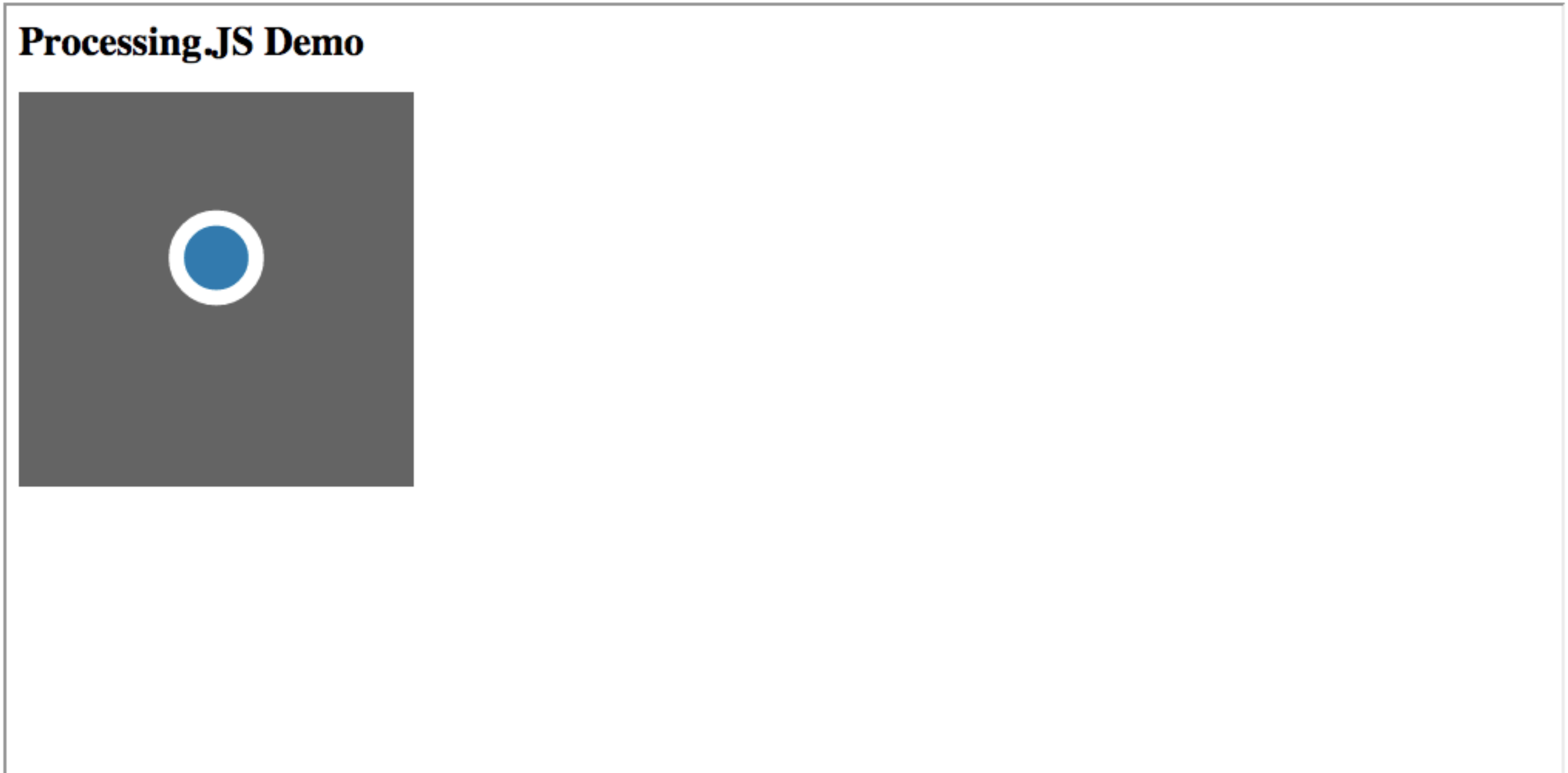


```
<html><head>
    <title>Linux and web servers for teaching electronics</title>
<!---
If you use computers at all, you're probably adept at using a web brower. Th
BeagleBone Black comes with Linux, a web server, JavaScript I/O library and
browser-based IDE installed out-of-the box. Learn how presentations and labs
can be easily built-up using these tools to teach many analog and digital
electronics concepts to individuals at all levels. Learn how to fork and
extend the tutorial/curriculuum that comes shipped on the board and served
up on Github pages. Use various HTML/JavaScript tools like jQuery and
Processing.JS to create rapid visualizations of interactive data.
--->
    <script src="/static/jquery.js"></script>
    <script src="/static/bonescript.js"></script>
    <script src="/scale13x/slides.js"></script>
    <style>
        #slides div {
            display:none;
            width:100%;
            height:100%;
            background-repeat:no-repeat;
            background-size:100%;
        }
        #slides div:first-child { display:block; }
    </style>
</head><body><div id="slides">
    <div style="background-image:url('/static/images/scale13x/Slide01.png');
    <div style="background-image:url('/static/images/scale13x/Slide02.png');
    <div style="background-image:url('/static/images/scale13x/Slide03.png');
    <div style="background-image:url('/static/images/scale13x/Slide04.png');
    <div>
        <iframe src="http://192.168.7.2/bone101/UI.html" style="margin-top:(
    </div>
    <div>
```



```
root@beaglebone:/var/lib/cloud9# uname -r
3.8.13-bone69
root@beaglebone:/var/lib/cloud9# cat /etc/dogtag
BeagleBoard.org Debian Image 2015-01-19
root@beaglebone:/var/lib/cloud9# 
```

Easy to extend with other visualizations



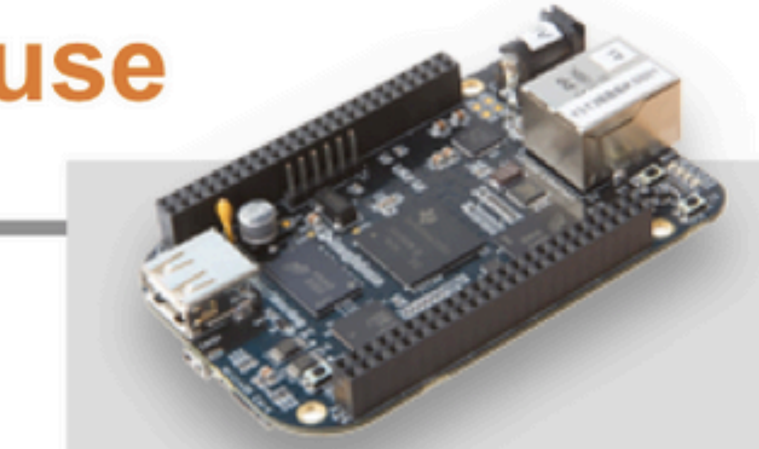
Processing.JS Demo

BeagleBone Black
1 GHz performance, ready to use

Truly flexible open hardware and software development platform

All you need is in the box

Proven ecosystem from prototype to product



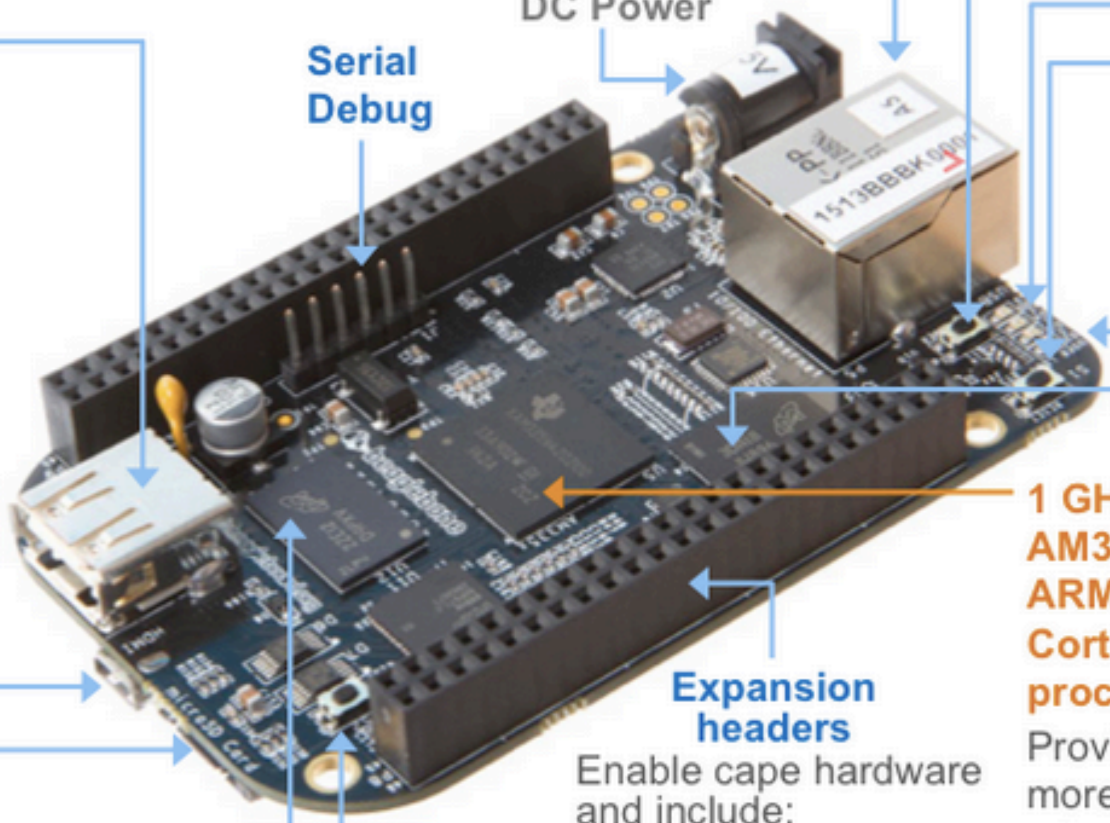
BeagleBone Black

- Ready to use: ~$50
- 1 GHz performance and embedded microcontrollers
- On-board HDMI to connect directly to TVs and monitors
- 512MB DDR3-800 RAM
- On-board 4GB flash storage with Debian frees up the microSD card slot
- Support for existing Cape plug-in boards: http://beaglebonecapes.com

Most affordable and proven open hardware Linux platform available

beagleboard.org

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

Serial Debug

DC Power

Boot Button

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!

Power Button

LEDS

Reset Button

USB Client
Development interface and directly powers board from PC

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

4GB on-board storage using eMMC

- Pre-loaded with Ångström 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

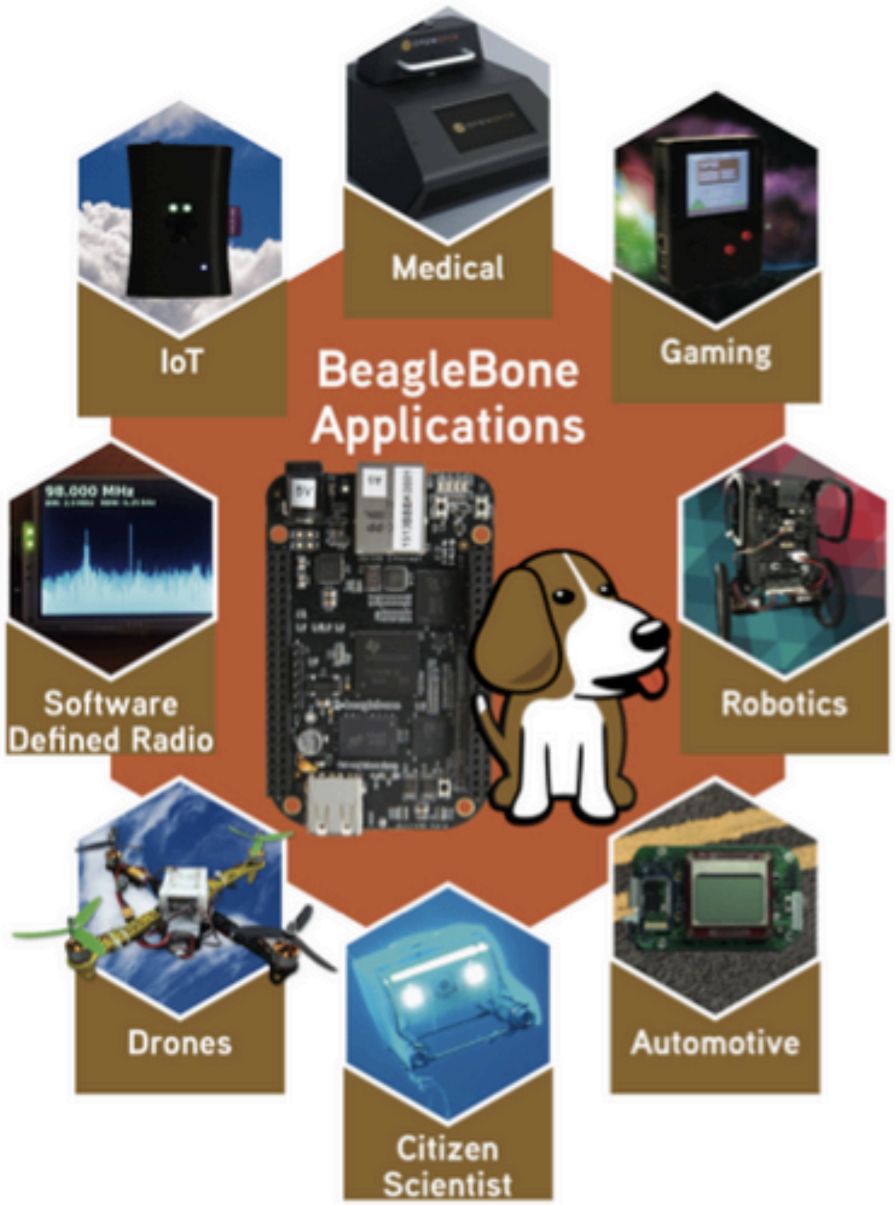
Money saving extras:

- Power over USB
- Included USB cable
- 4GB on-board storage
- Built-in PRU microcontrollers

beagleboard.org

Huge base of existing projects

Making it fun and easy to bring ideas to life

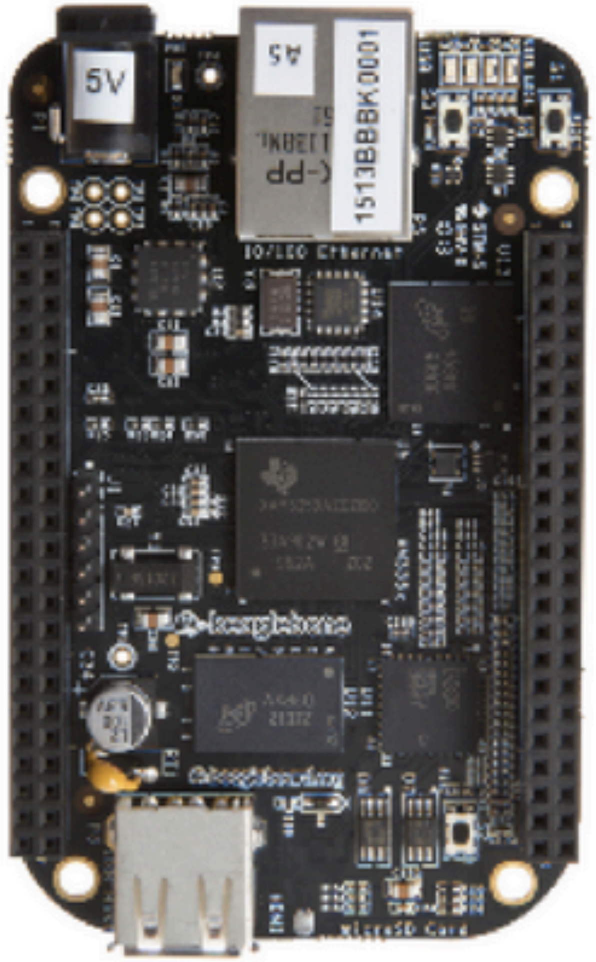


- Medical analysis, assistance and information management
- Home information, automation and security systems
- Home and mobile entertainment and educational systems
- New types of communications systems
- Personal robotic devices for cleaning, upkeep and manufacturing
- Remote presence and monitoring
- Automotive information management and control systems
- Personal environmental exploration and monitoring

http://beagleboard.org/project

Cape Expansion Headers

DGND	1	2	DGND
VDD_3V3	3	4	VDD_3V3
VDD_5V	5	6	VDD_5V
SYS_5V	7	8	SYS_5V
PWR_BUT	9	10	SYS_RESETN
UART4_RXD	11	12	GPIO_60
UART4_TXD	13	14	EHRPWM1A
GPIO_48	15	16	EHRPWM1B
SPI0_CS0	17	18	SPI0_D1
I2C2_SCL	19	20	I2C2_SDA
SPI0_D0	21	22	SPI0_SCLK
GPIO_49	23	24	UART1_TXD
GPIO_117	25	26	UART1_RXD
GPIO_115	27	28	SPI1_CS0
SPI1_D0	29	30	GPIO_122
SPI1_SCLK	31	32	VDD_ADC
AIN4	33	34	GNDA_ADC
AIN6	35	36	AIN5
AIN2	37	38	AIN3
AIN0	39	40	AIN1
GPIO_20	41	42	ECAPPWM0
DGND	43	44	DGND
DGND	45	46	DGND



LEGEND
POWER/GROUND/RESET
AVAILABLE DIGITAL
AVAILABLE PWM
SHARED I2C BUS
RECONFIGURABLE DIGITAL
ANALOG INPUTS (1.8V)

DGND	1	2	DGND
MMC1_DAT6	3	4	MMC1_DAT7
MMC1_DAT2	5	6	MMC1_DAT3
GPIO_66	7	8	GPIO_67
GPIO_69	9	10	GPIO_68
GPIO_45	11	12	GPIO_44
EHRPWM2B	13	14	GPIO_26
GPIO_47	15	16	GPIO_46
GPIO_27	17	18	GPIO_65
EHRPWM2A	19	20	MMC1_CMD
MMC1_CLK	21	22	MMC1_DAT5
MMC1_DAT4	23	24	MMC1_DAT1
MMC1_DAT0	25	26	GPIO_61
LCD_VSYNC	27	28	LCD_PCLK
LCD_HSYNC	29	30	LCD_AC_BIAS
LCD_DATA14	31	32	LCD_DATA15
LCD_DATA13	33	34	LCD_DATA11
LCD_DATA12	35	36	LCD_DATA10
LCD_DATA8	37	38	LCD_DATA9
LCD_DATA6	39	40	LCD_DATA7
LCD_DATA4	41	42	LCD_DATA5
LCD_DATA2	43	44	LCD_DATA3
LCD_DATA0	45	46	LCD_DATA1

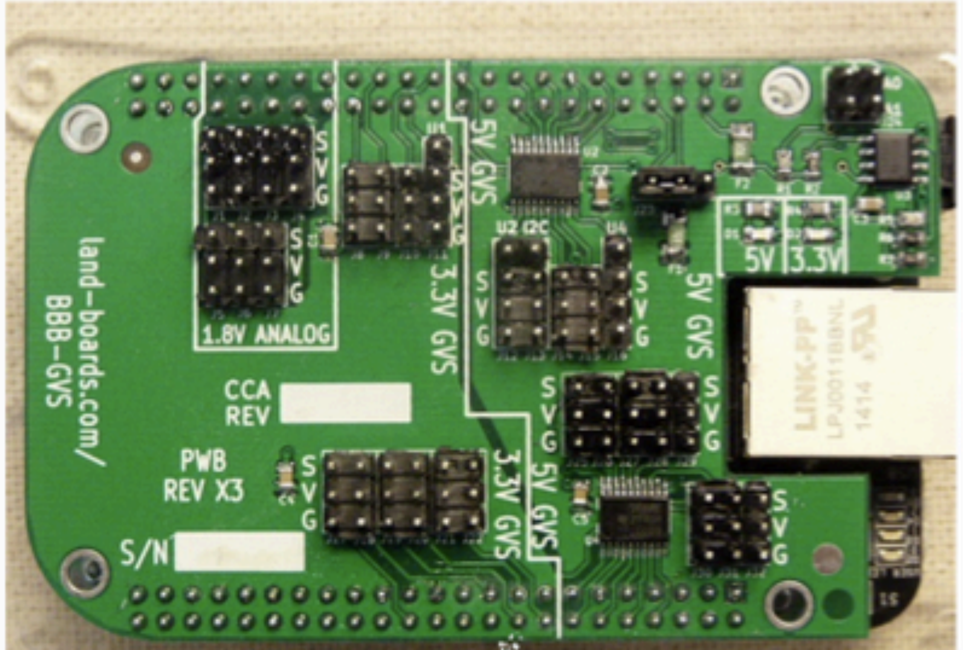
beagleboard.org

Capes to make wiring even easier

BBB-GVS

I SELL ON tindie

GVS

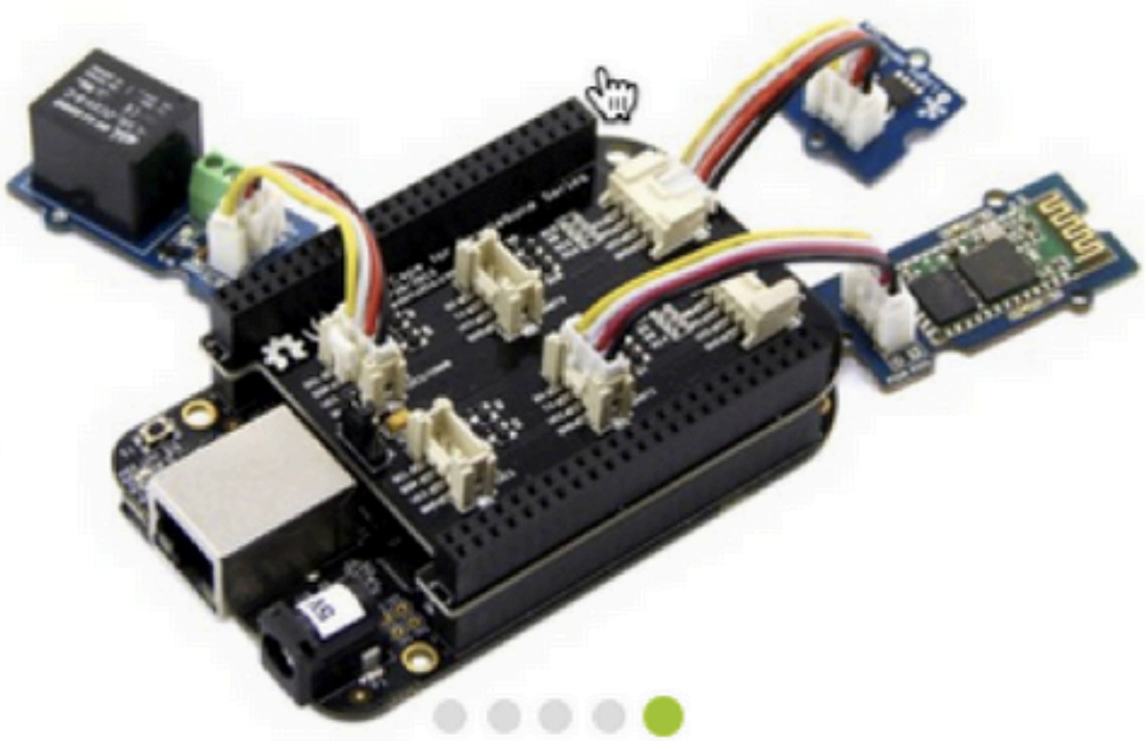


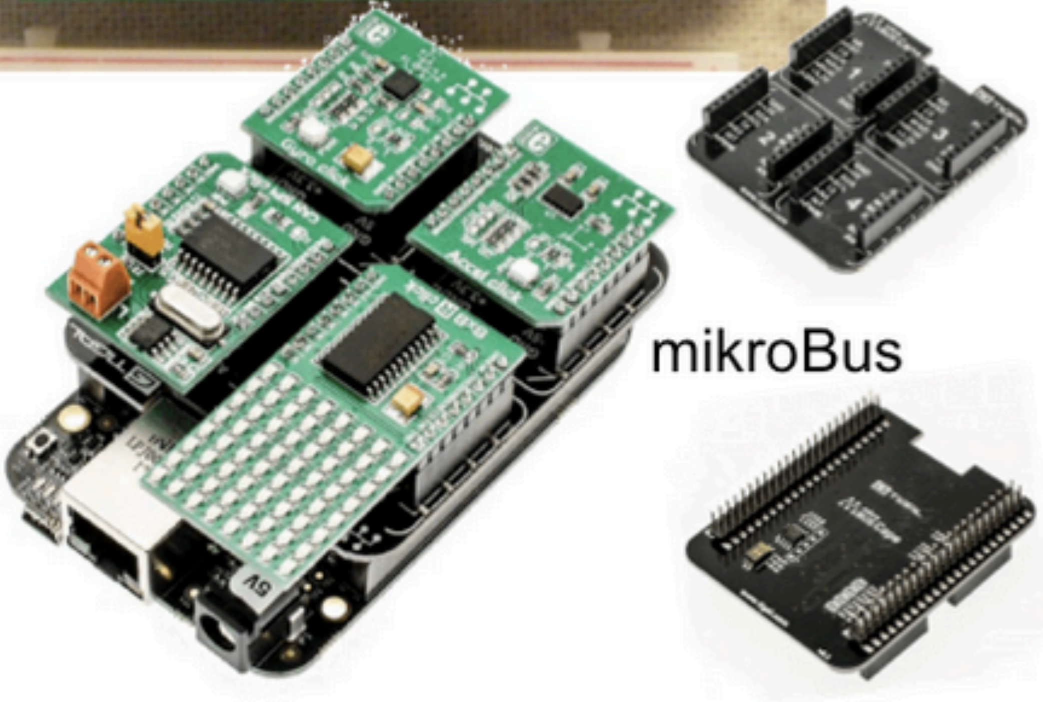
Grove Cape for BeagleBone Series

SKU: 811001001

USD ▾ $19.90

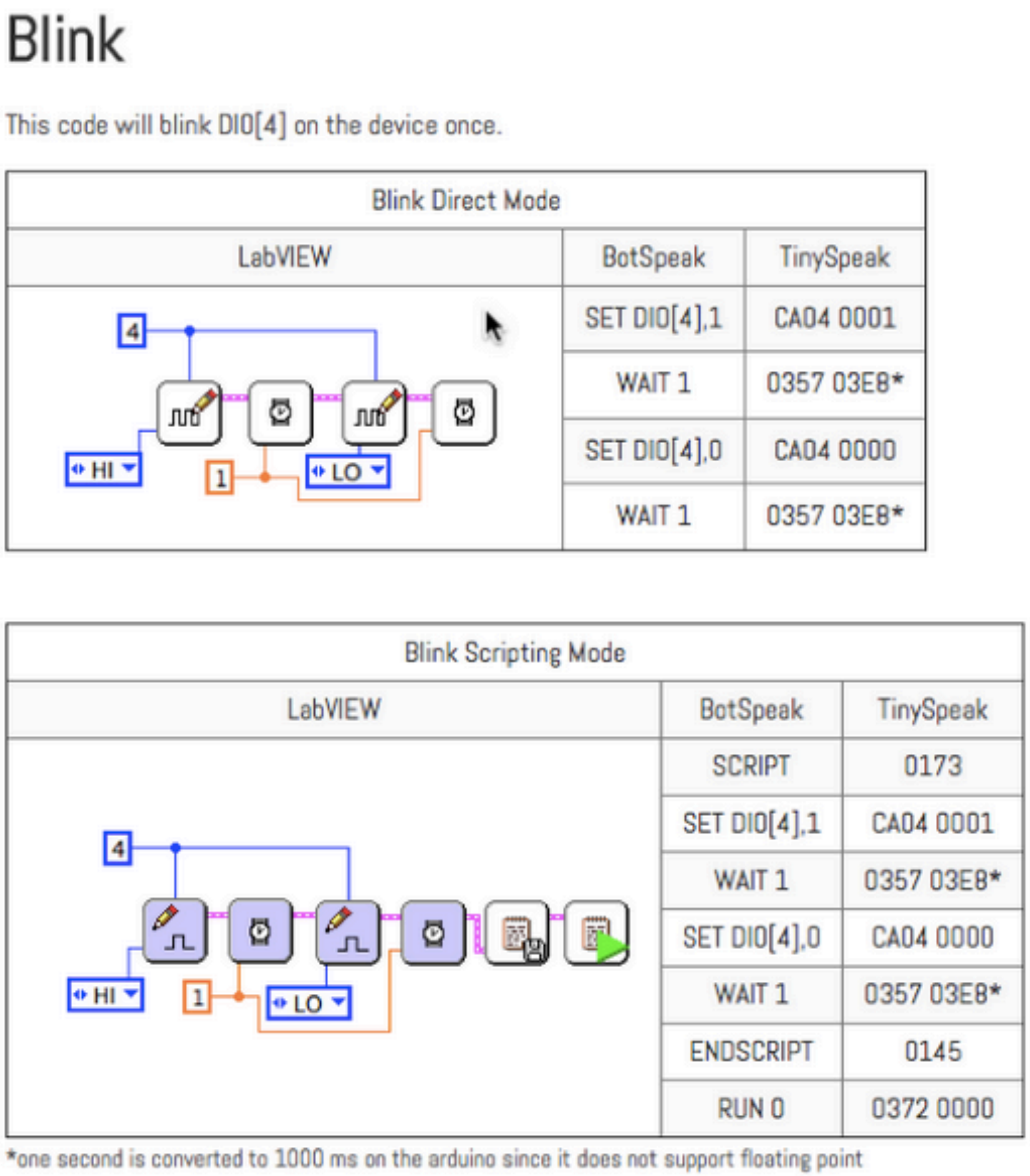
Grove





mikroBus

http://botspeak.org/



Blink

This code will blink DIO[4] on the device once.

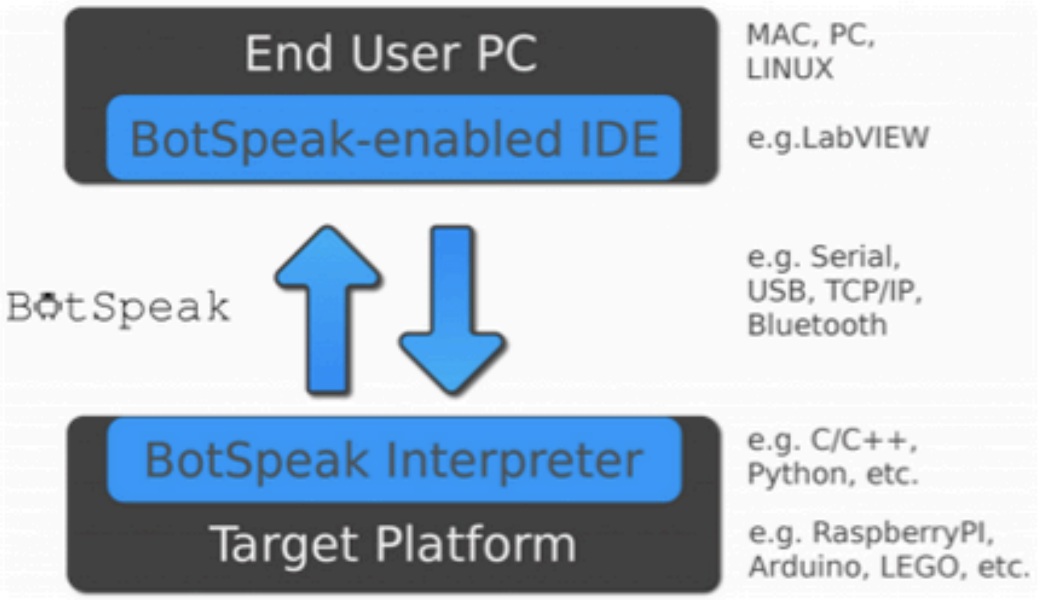
Blink Direct Mode		
LabVIEW	BotSpeak	TinySpeak
	SET DIO[4],1	CA04 0001
	WAIT 1	0357 03E8*
	SET DIO[4],0	CA04 0000
	WAIT 1	0357 03E8*

Blink Scripting Mode		
LabVIEW	BotSpeak	TinySpeak
	SCRIPT	0173
	SET DIO[4],1	CA04 0001
	WAIT 1	0357 03E8*
	SET DIO[4],0	CA04 0000
	WAIT 1	0357 03E8*
	ENDSCRIPT	0145
	RUN 0	0372 0000

*one second is converted to 1000 ms on the arduino since it does not support floating point

Speak

BOT



beagleboard.org

What are PRUs

- “Programmable Real-time Units”
- 32-bit RISC processors at 200MHz with single-cycle pin access for hard real-time
- Optimized for packet processing/switching and software implementations of peripherals
- Part of the PRU-ICSS, “Industrial Communications SubSystem”

PRU: Programmable Real-time Unit

Architecture

- Two 32-bit RISC cores for real-time functions each running at 200MHz
- 8KB IRAM, 8KB DRAM, 12KB Shared RAM
- **Single-cycle execution**
- **Direct I/O interface sampling at ~5ns**
- Logic, Control and arithmetic instructions
- 32-bit MULT and Interrupt controller
- Efficient bit/byte/word manipulations

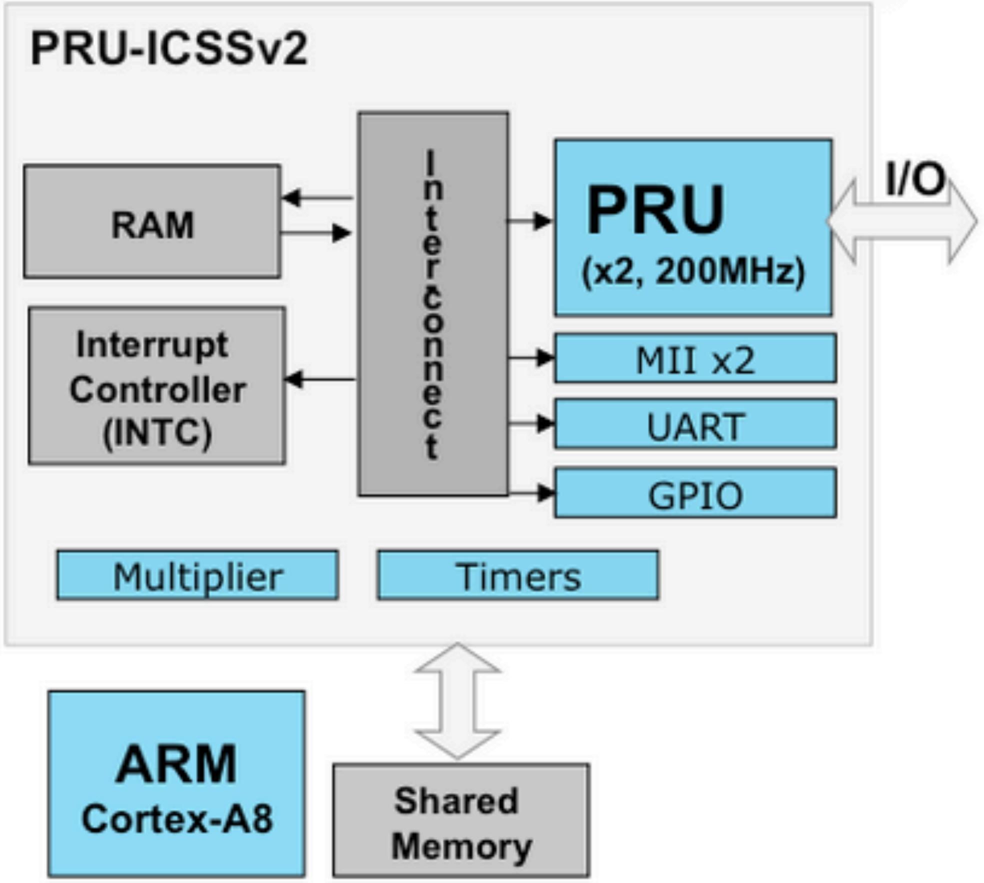
Capabilities

- Implement **Real-time communication interfaces** : PROFIBUS, EtherCAT, PROFINET & Ethernet/IP
- Implement **custom IP** (such as EnDAT 2.2, SINC3 decimation, PWMs, DP Memory, Manchester Coding, 9 bit UART or a Backplane bus)

Advantages

- Completely programmable & Flexible
- Reduce system cost & complexity

AM335x SoC: ARM + PRU



beagleboard.org

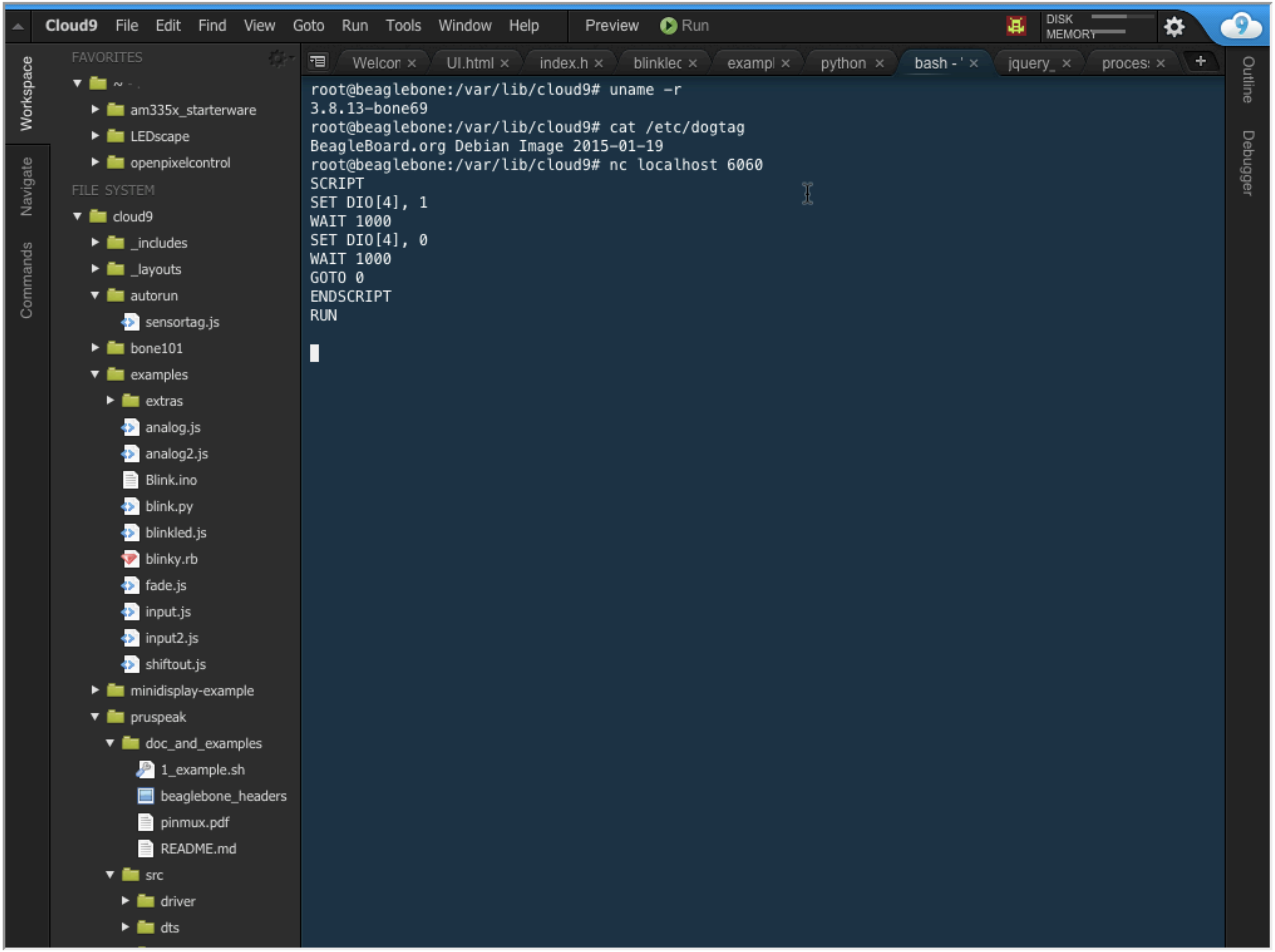
25 PRU low-latency I/Os

P9

DGND	1	2	DGND
VDD_3V3	3	4	VDD_3V3
VDD_5V	5	6	VDD_5V
SYS_5V	7	8	SYS_5V
PWR_BUT	9	10	SYS_RESETN
GPIO_30	11	12	GPIO_60
GPIO_31	13	14	GPIO_50
GPIO_48	15	16	GPIO_51
GPIO_5	17	18	GPIO_4
I2C2_SCL	19	20	I2C2_SDA
GPIO_3	21	22	GPIO_2
GPIO_49	23	24	GPIO_15
PRU0_7	25	26	PRU1_16 IN
PRU0_5	27	28	PRU0_3
PRU0_1	29	30	PRU0_2
PRU0_0	31	32	VDD_ADC
AIN4	33	34	GNDA_ADC
AIN6	35	36	AIN5
AIN2	37	38	AIN3
AIN0	39	40	AIN1
PRU0_6	41	42	PRU0_4
DGND	43	44	DGND
DGND	45	46	DGND

P8

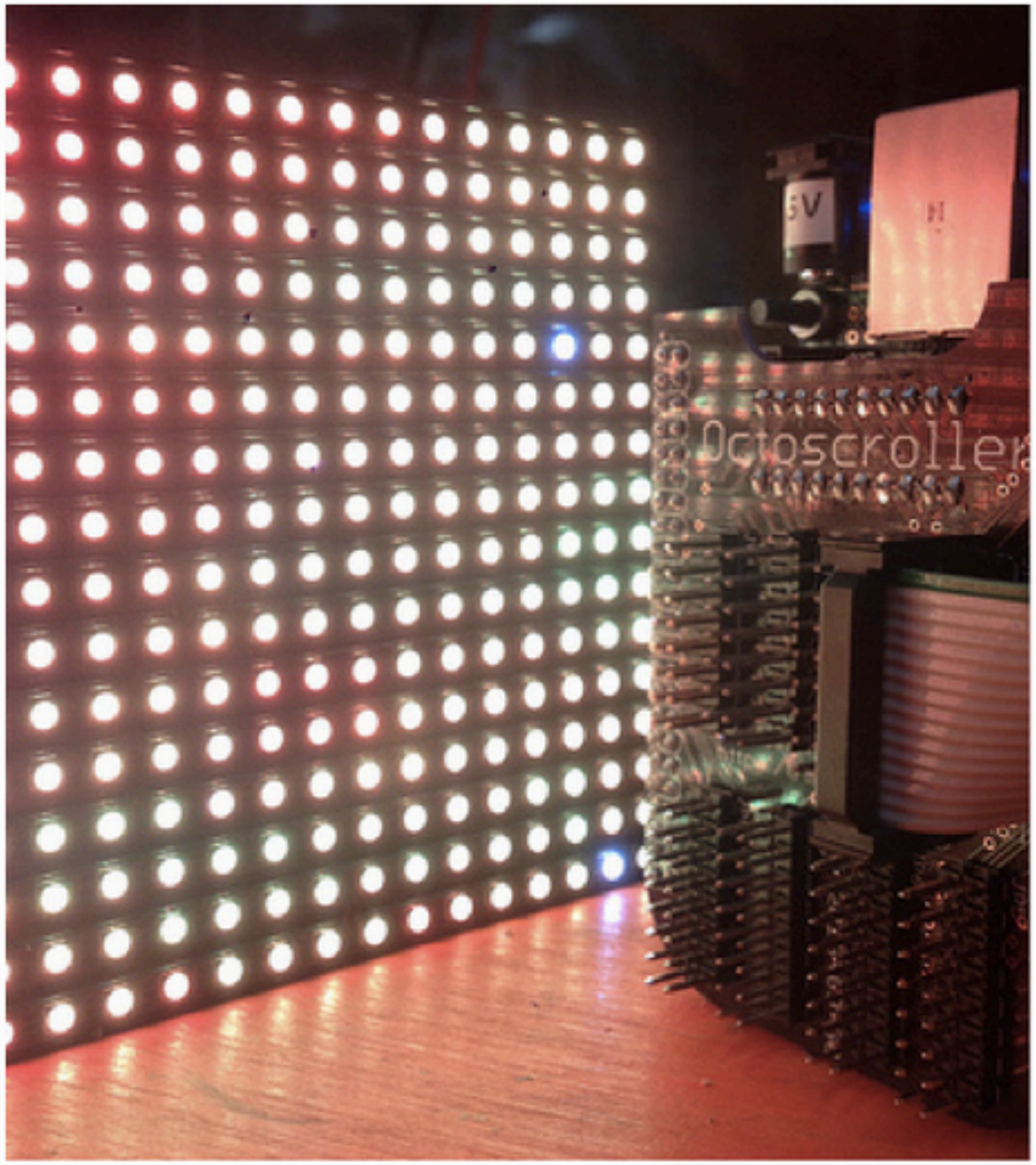
DGND	1	2	DGND
GPIO_38	3	4	GPIO_39
GPIO_34	5	6	GPIO_35
GPIO_66	7	8	GPIO_67
GPIO_69	9	10	GPIO_68
PRU0_15 OUT	11	12	PRU0_14 OUT
GPIO_23	13	14	GPIO_26
GPIO_47	15	16	GPIO_46
GPIO_27	17	18	GPIO_65
GPIO_22	19	20	PRU1_13
PRU1_12	21	22	GPIO_37
GPIO_36	23	24	GPIO_33
GPIO_32	25	26	GPIO_61
PRU1_8	27	28	PRU1_10
PRU1_9	29	30	PRU1_11
GPIO_10	31	32	GPIO_11
GPIO_9	33	34	GPIO_81
GPIO_8	35	36	GPIO_80
GPIO_78	37	38	GPIO_79
PRU1_6	39	40	PRU1_7
PRU1_4	41	42	PRU1_5
PRU1_2	43	44	PRU1_3
PRU1_0	45	46	PRU1_1



```
root@beaglebone:/var/lib/cloud9# uname -r
3.8.13-bone69
root@beaglebone:/var/lib/cloud9# cat /etc/dogtag
BeagleBoard.org Debian Image 2015-01-19
root@beaglebone:/var/lib/cloud9# nc localhost 6060
SCRIPT
SET DIO[4], 1
WAIT 1000
SET DIO[4], 0
WAIT 1000
GOTO 0
ENDSCRIPT
RUN
```

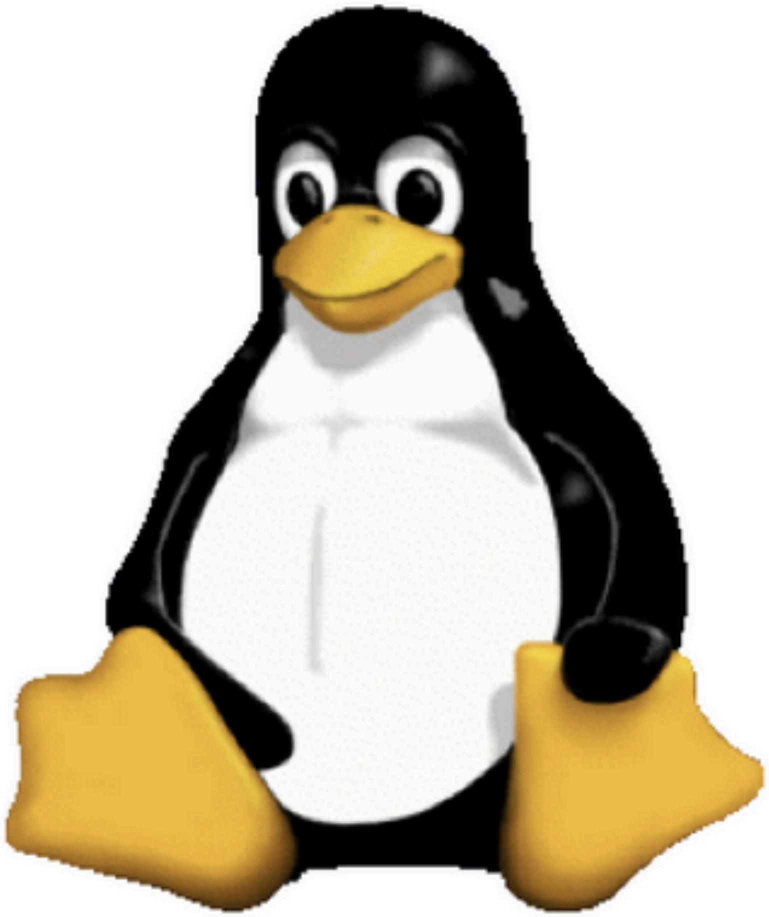
LEDscape and Octoscroller

- Works with Adafruit 32x16 LED panels
- 12-bit color supported through PRU-based pulse-width modulation
- Open source software and hardware
- Content delivered using network packets (Python)
- Supports 64 panels each



Must teach operating systems concepts

- Where are my bits?
- What is a command line and why do I care?
- In Linux, everything is a file



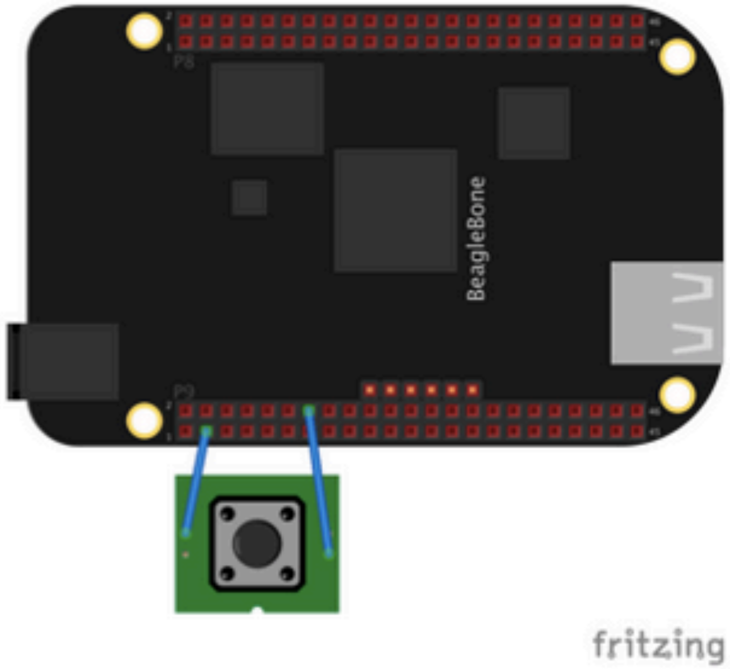
Some quick interfaces to hack

- LEDs

```
root@beaglebone:~# cd /sys/class/leds/beaglebone\:green\:usr0
root@beaglebone:/sys/class/leds/beaglebone:green:usr0# cat trigger
none nand-disk mmc0 mmc1 timer oneshot [heartbeat] backlight gpio cpu0 default-on transient
root@beaglebone:/sys/class/leds/beaglebone:green:usr0# echo none > trigger
root@beaglebone:/sys/class/leds/beaglebone:green:usr0# echo 1 > brightness
root@beaglebone:/sys/class/leds/beaglebone:green:usr0# echo 0 > brightness
```

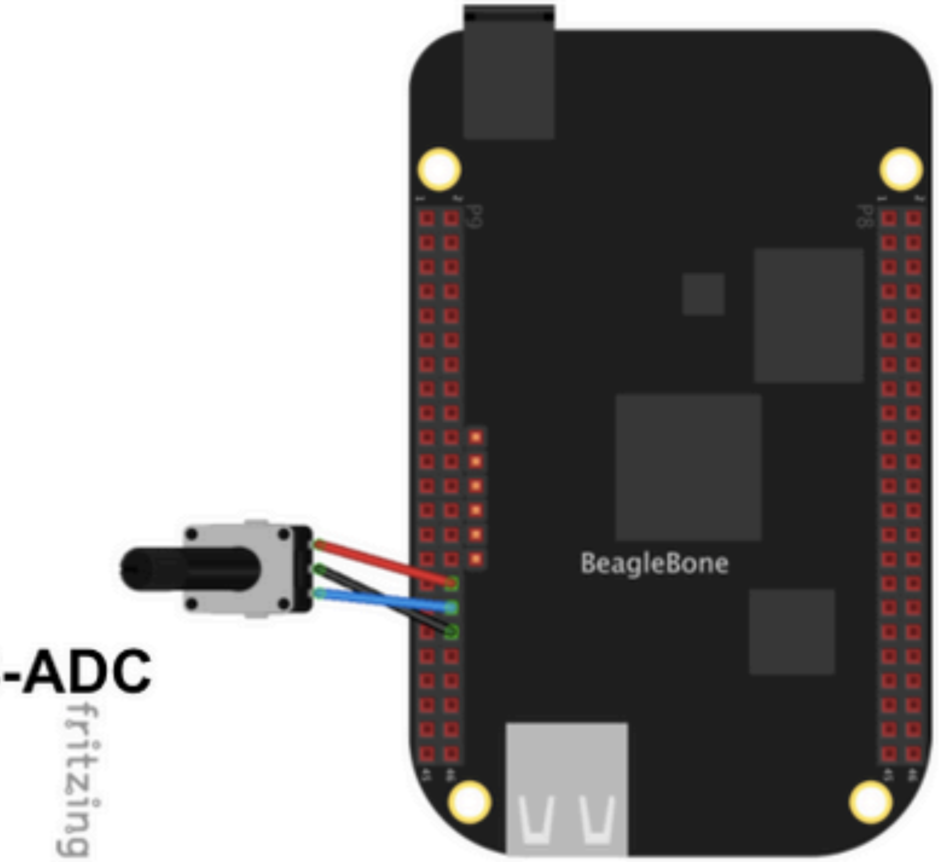
- GPIOs

```
root@beaglebone:~# config-pin overlay cape-universaln
Loading cape-universaln overlay
root@beaglebone:~# config-pin p9.14 gpio_pd
root@beaglebone:~# config-pin -q p9.14
P9_14 Mode: gpio_pd Direction: in Value: 0
root@beaglebone:~# cd /sys/class/gpio
root@beaglebone:/sys/class/gpio# cat gpio50/direction
in
root@beaglebone:/sys/class/gpio# cat gpio50/value
0
```



Hacking with the ADC

- Don’t forget it is only **1.8V**
- The overlay will load an ADC driver



- Steps

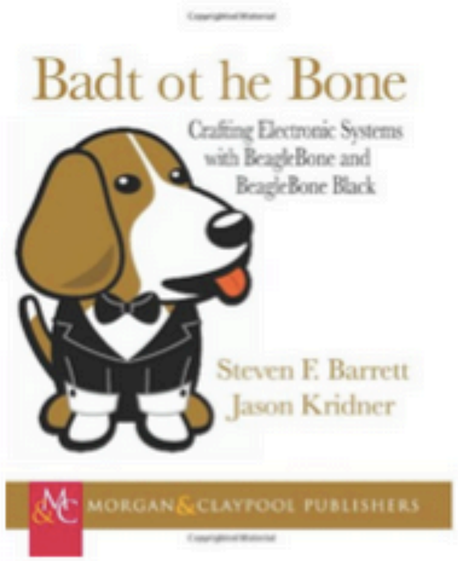
```
root@beaglebone:~# config-pin overlay BB-ADC
Loading BB-ADC overlay
root@beaglebone:~# config-pin -q p9.36
Pin is not modifyable: P9_36 AIN5
root@beaglebone:~# cd /sys/bus/iio/devices/iio\:device0
root@beaglebone:/sys/bus/iio/devices/iio:device0# cat in_voltage5_raw
1850
```

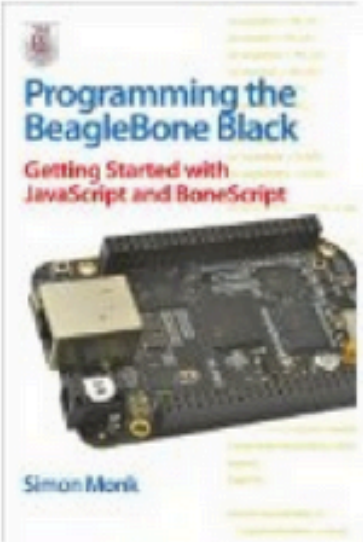
Some BeagleBoard.org related books

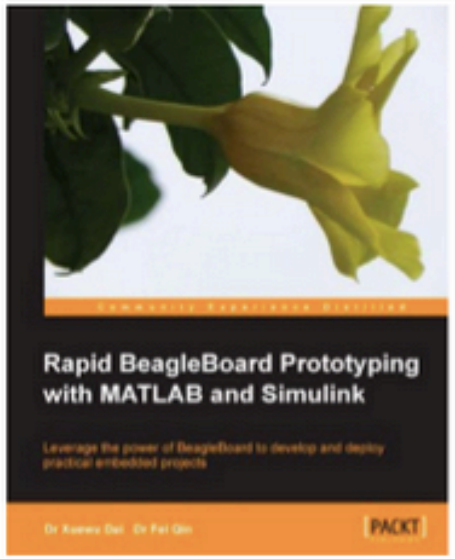
Make:
Getting Started with BeagleBone
Linux-Powered Electronic Projects With Python and JavaScript
Matt Richardson

BeagleBone Home Automation
Live your sophisticated dream with home automation using BeagleBone
Juha Lumme
PACKT

BeagleBone Robotic Projects
Create complex and exciting robotic projects with the BeagleBone Black
Richard Grimmett
PACKT


Badt ot he Bone
Crafting Electronic Systems with BeagleBone and BeagleBone Black
Steven F. Barrett
Jason Kridner
MORGAN & CLAYPOOL PUBLISHERS


Programming the BeagleBone Black
Getting Started with JavaScript and BoneScript
Simon Monk


Rapid BeagleBoard Prototyping with MATLAB and Simulink
Leverage the power of BeagleBoard to develop and deploy practical embedded projects
Dr Xuewu Dai Dr Fei Qin
PACKT

http://bit.ly/bbb-books

beagleboard.org

Stop