BeagleBone Cookbook Webinar Series

Recipe #5

I/O with C and mmap()

December 1, 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

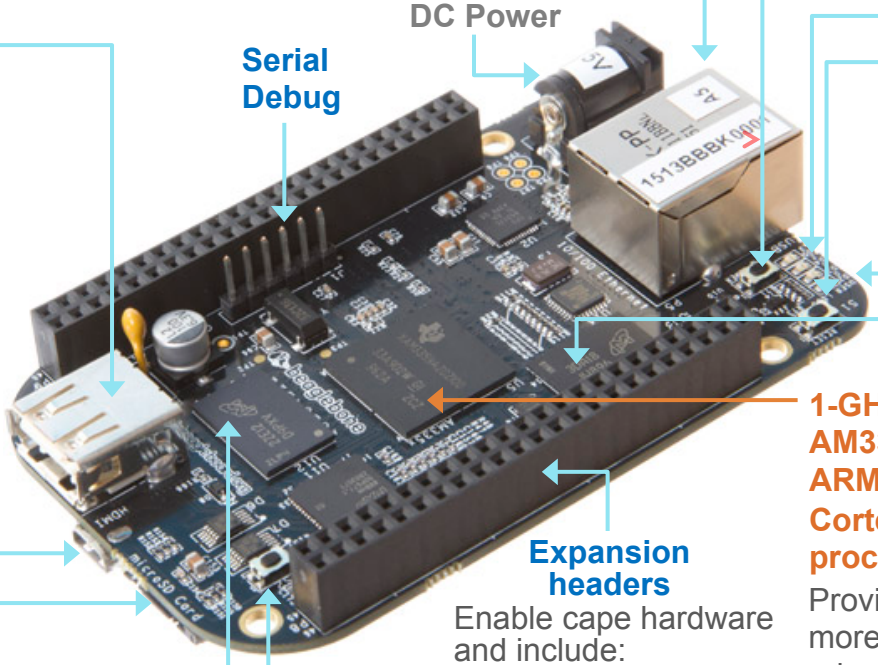
All you need is in the box

Proven ecosystem from prototype to product

- Ready to use
  - USB client network
  - Built-in tutorials
  - Browser based IDE
  - Flashed 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://beaglebonecapes.com

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

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

Power Button

LEDS

Reset Button

USB Client
Development interface and directly powers board from PC

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

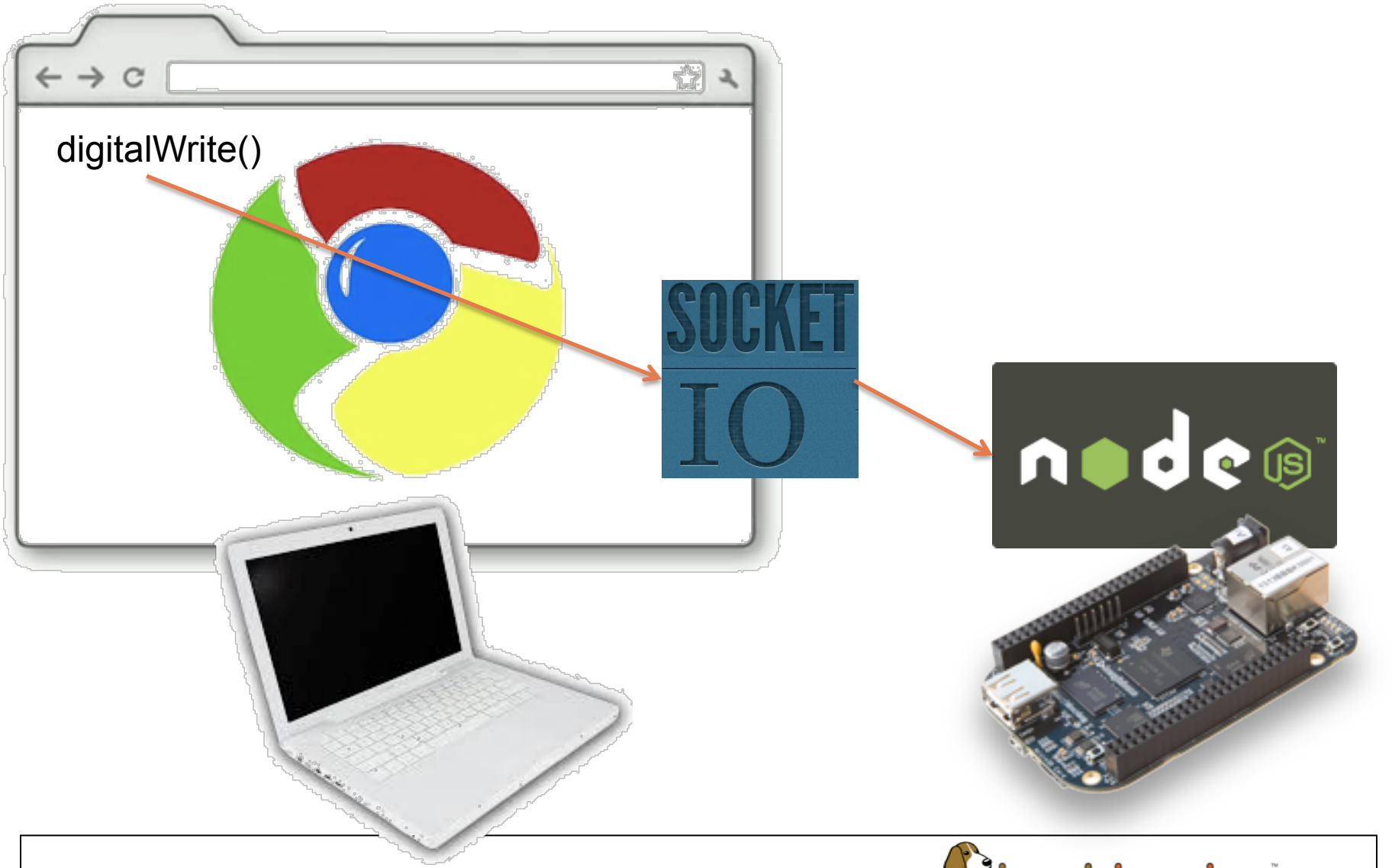
1-GHz Sitara AM335x ARM® Cortex™-A8 processor
Provides a more advanced user interface and up to 150% better performance than ARM11

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!

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

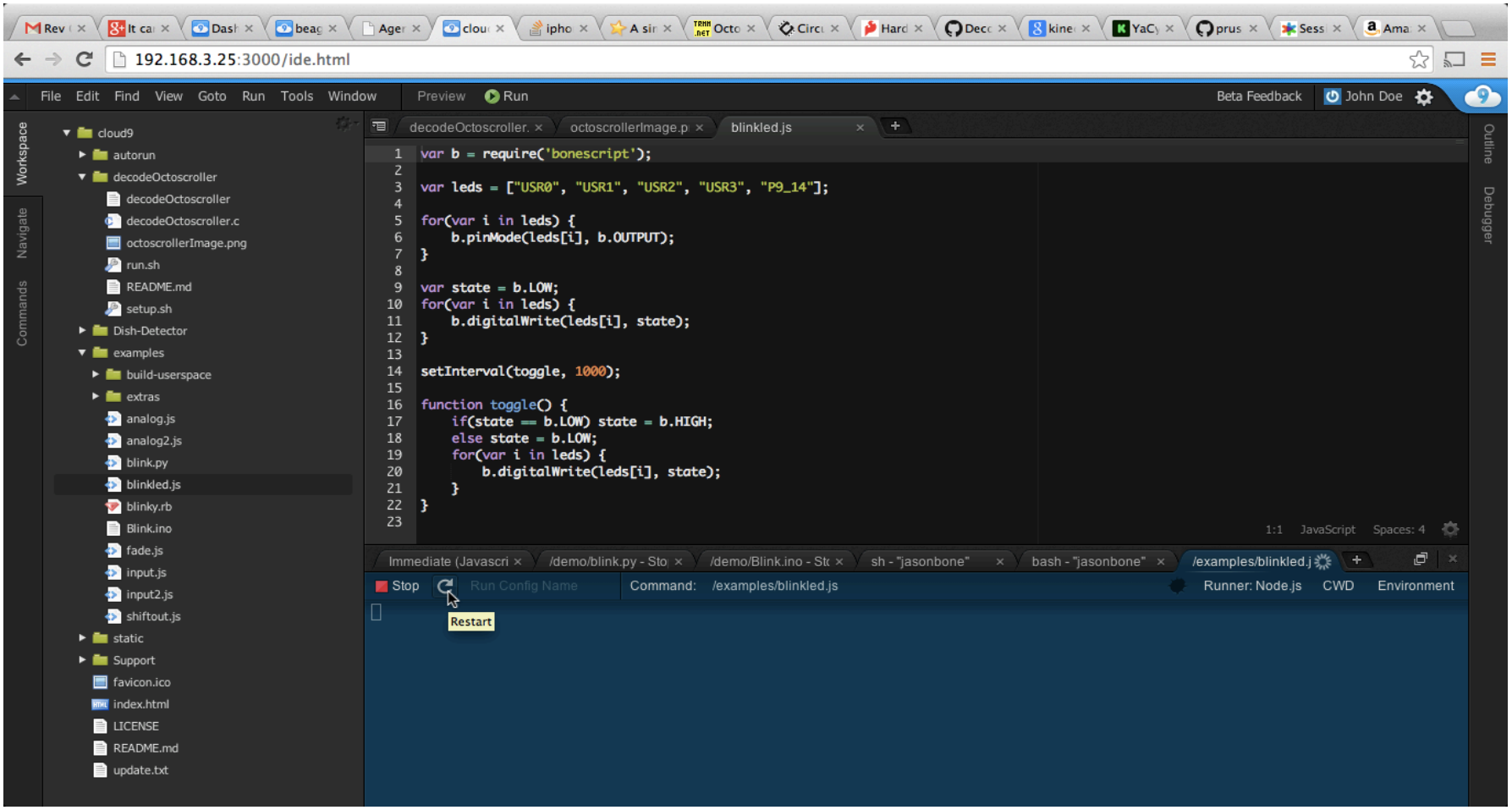
Simple browser-based interactions

http://beagleboard.github.io/bone101

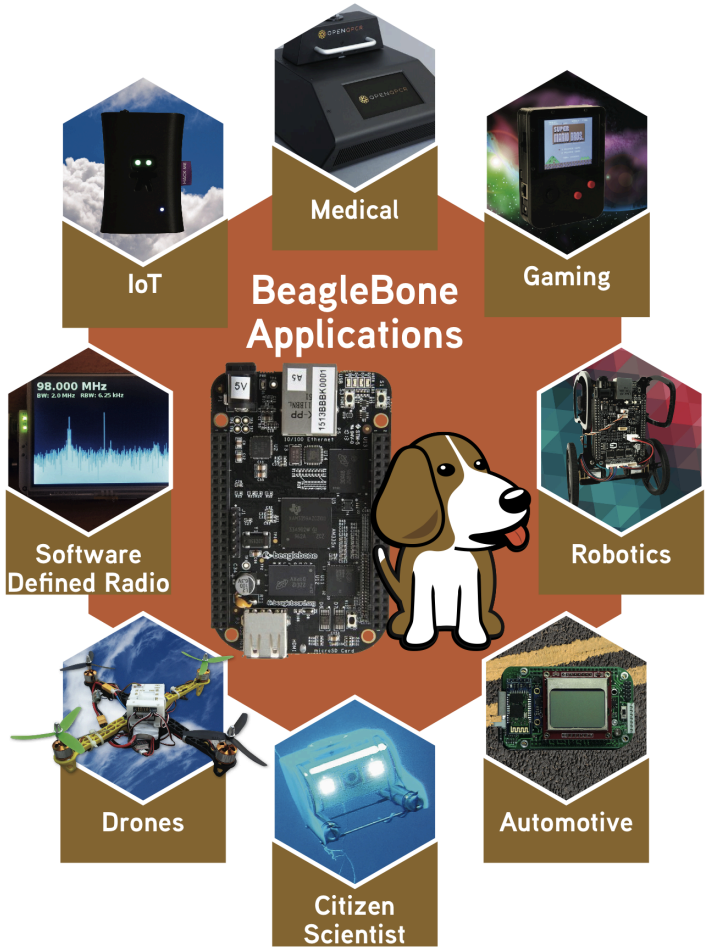


Cloud9 IDE hosted locally

Zero install and exposes command-line



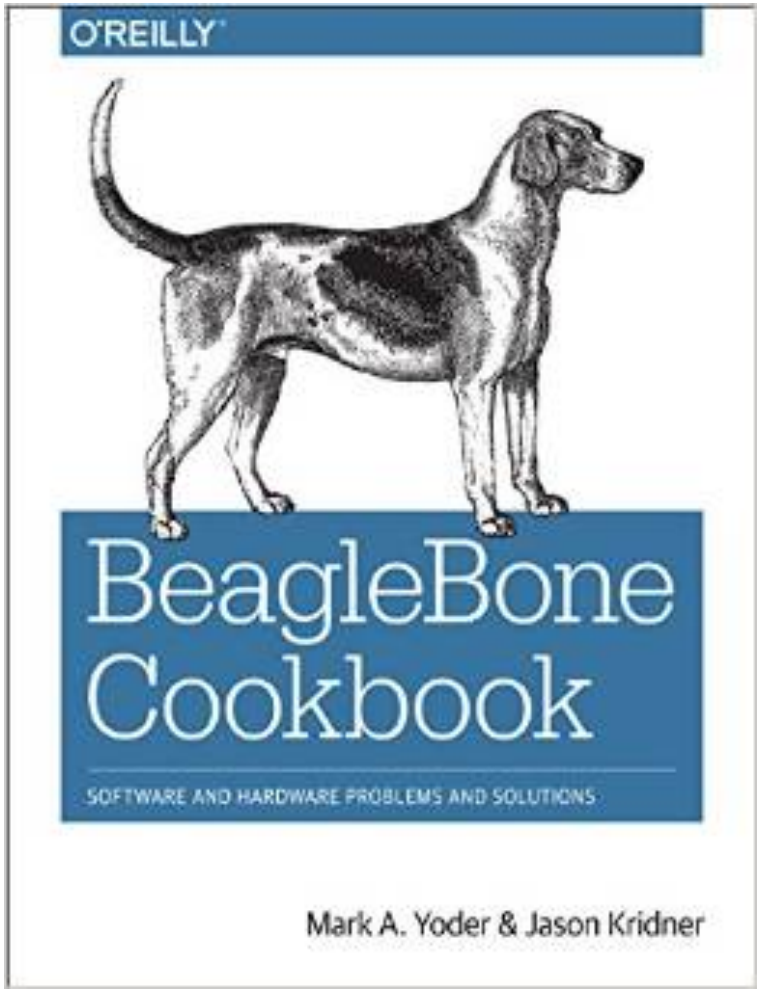
10,000s of developers building connected devices today



- 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



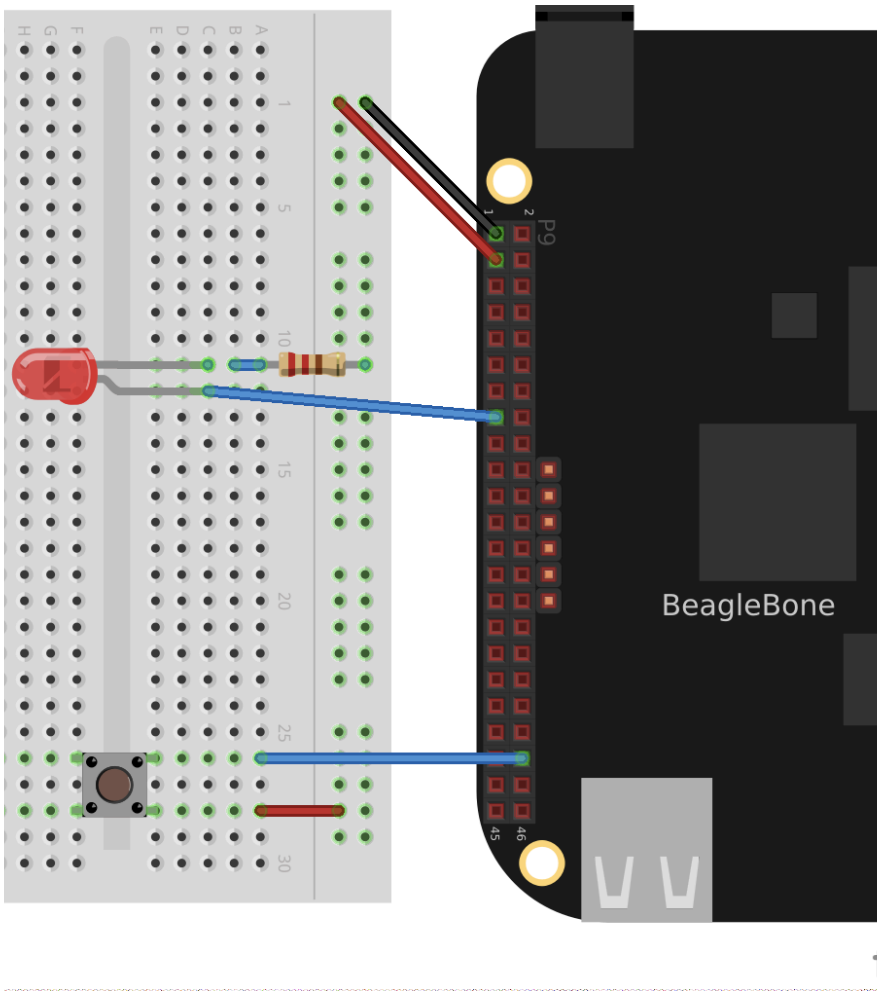
- 99 recipes covering
  - Basics
  - Sensors
  - Displays and outputs
  - Motors
  - Internet of things
  - Kernel
  - Real-time I/O
  - Capes

Prerequisites

- Connect to the board per recipe 1.2
  - http://beagleboard.org/getting-started
- Verify the software image per recipe 1.3 and potentially updating per recipe 1.9
  - http://beagleboard.org/latest-images
- Components
  - BeagleBone Black
  - Push button or 3.3V function generator
  - Jumper wire
  - LED with resistor or (preferred) oscilloscope

Connect a button and an LED

http://beagleboard.org/Support/bone101/#headers-gpio

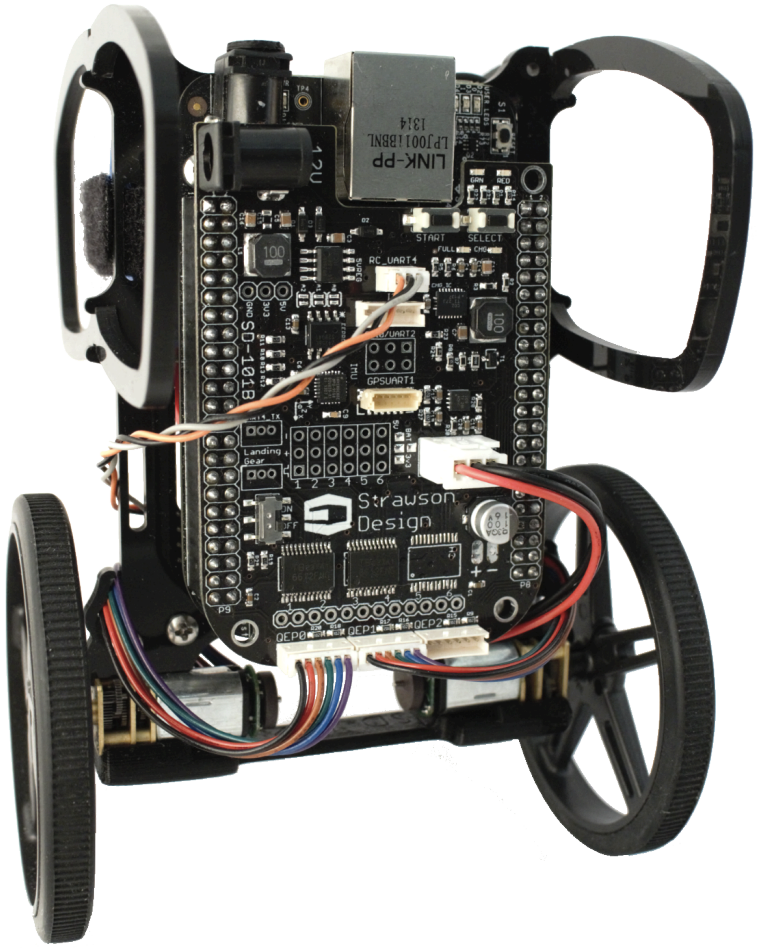


P9

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
GPIO_117	25	26	GPIO_14
GPIO_115	27	28	GPIO_113
GPIO_111	29	30	GPIO_112
GPIO_110	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	GPIO_7
DGND	43	44	DGND
DGND	45	46	DGND

Input on GPIO_7 and output on GPIO_31

Understanding Real-Time



- Throughput vs. latency
- Hard, soft and firm
- Context switching
- Task scheduling
- Linux RT_PREEMPT
- Using ‘strace’ and ‘oprofile’

What are /dev/mem and mmap()?

- /dev/mem is a character device that is an image of the main physical memory of the computer
- mmap() is a system function to map devices into (virtual) memory
- Together, they can be used to provide an application that has only a virtual memory space with access to specific physical addresses
- Directly accessing the registers bypasses system calls and avoids context switches
- This is really just a step towards writing your own device driver

Recipe 8.4: I/O with devmem2

```
bone# wget http://free-electrons.com/pub/mirror/devmem2.c
bone# gcc -o devmem2 devmem2.c && mv devmem2 /usr/local/bin/
bone# ln -s /sys/class/gpio
bone# echo 31 > gpio/export
bone# echo out > gpio/gpio31/direction
bone# echo 1 > gpio/gpio31/value
bone# echo 0 > gpio/gpio31/value
bone# devmem2 0x44E07138
bone# devmem2 0x44E07190 w 0x80000000
bone# devmem2 0x44E07194 w 0x80000000
bone# devmem2 0x44E07138
```

Recipe 8.4: I/O with C and mmap()

bone# wget https://raw.githubusercontent.com/BeagleBoneCookbook/firstEdition/master/08realtime/pushLEDmmap.c

bone# wget https://raw.githubusercontent.com/BeagleBoneCookbook/firstEdition/master/08realtime/pushLEDmmap.h

bone# gcc -O3 –o pushLEDmmap pushLEDmmap.c

bone# ./pushLEDmmap

^C

More

- AM335x Technical Reference Manual
  - http://bit.ly/1B4Cm45
- StarterWare for Sitara
  - http://www.ti.com/tool/starterware-sitara
- Enabling RT_PREEMPT
  - http://elinux.org/Beagleboard:BeagleBoneBlack_Debian#4.1.x-ti
- Learning to write a device driver in Recipe 7.2
- Program GPIO with PRU in Recipe 8.6
- Shortcuts to updates and examples from the book
  - http://beagleboard.org/cookbook