BeagleBone Cookbook Webinar Series
Recipe #4
Controlling the Speed and Direction of a DC Motor

November 24, 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

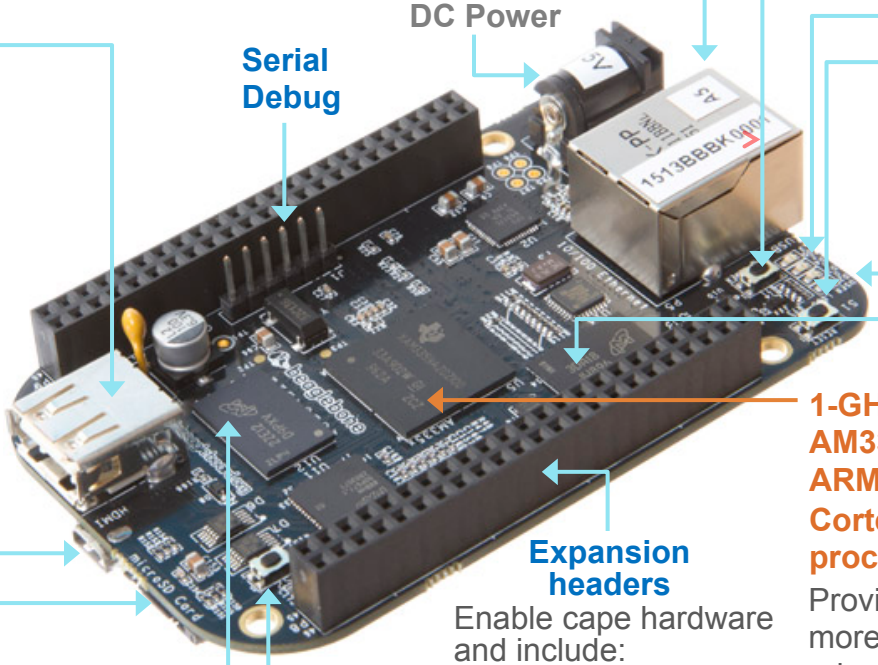
All you need is in the box

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

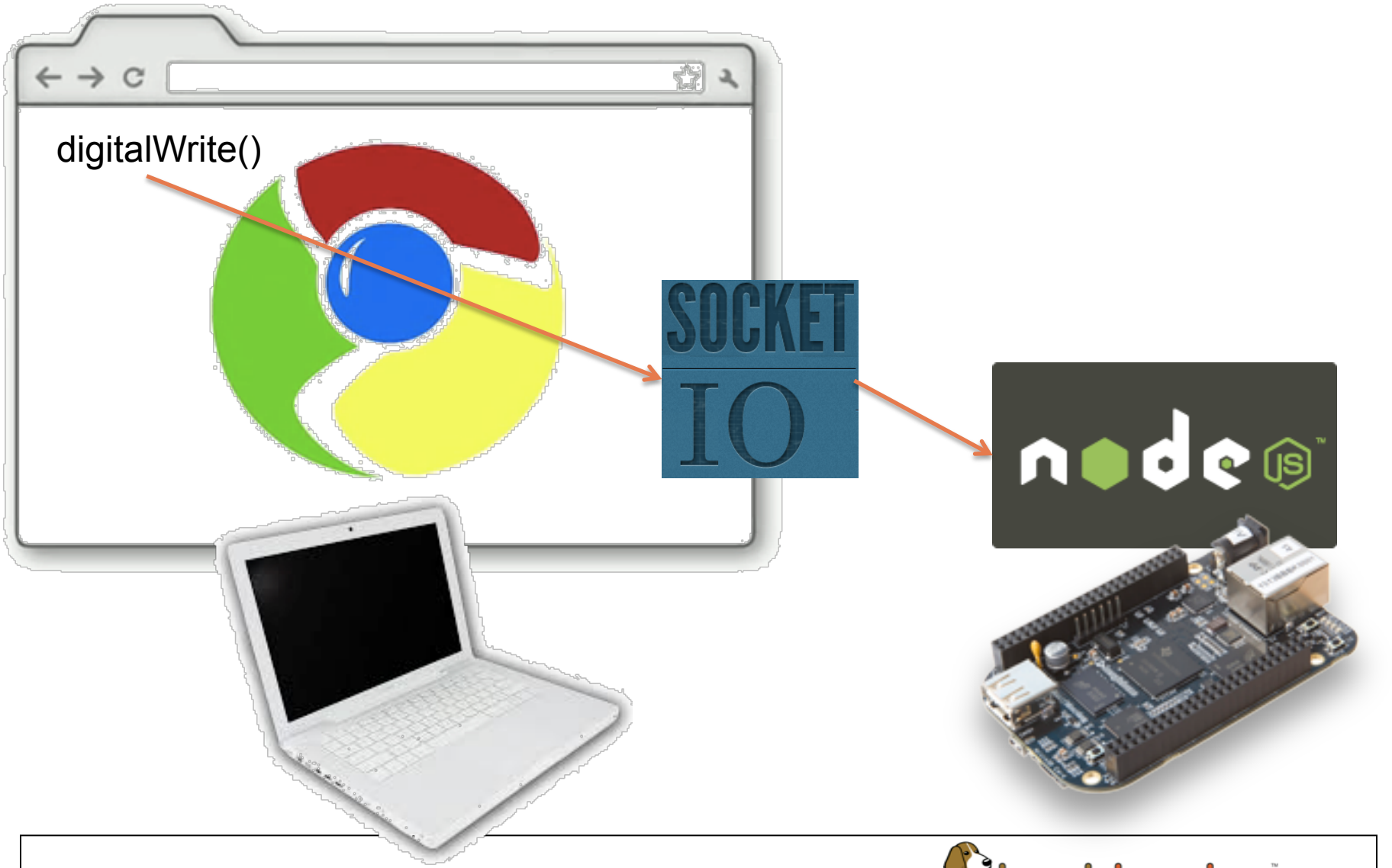
1-GHz Sitara AM335x ARM® Cortex™-A8 processor
Provides a more advanced user interface and up to 150% better performance than ARM11

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

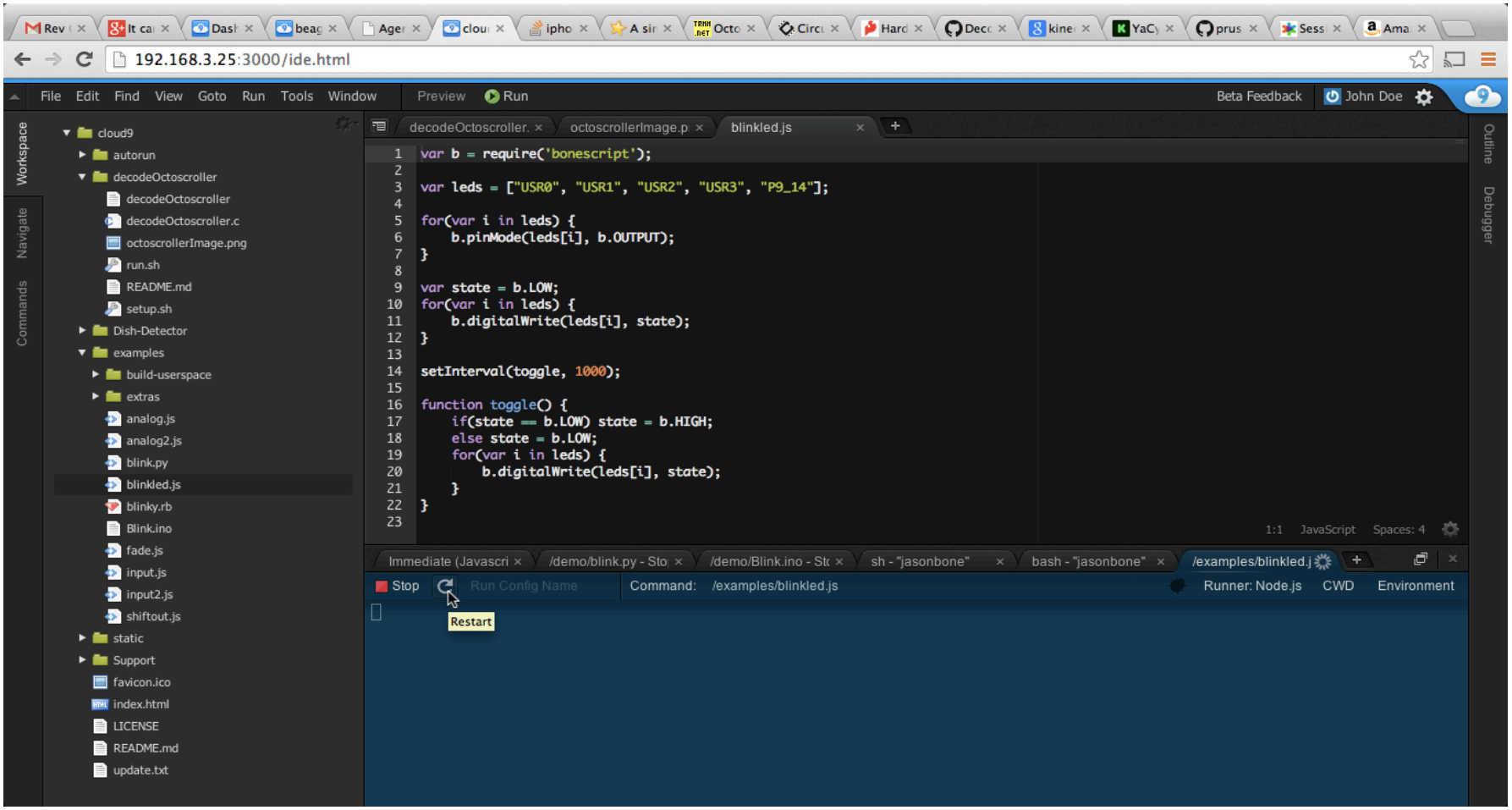
Simple browser-based interactions

http://beagleboard.github.io/bone101

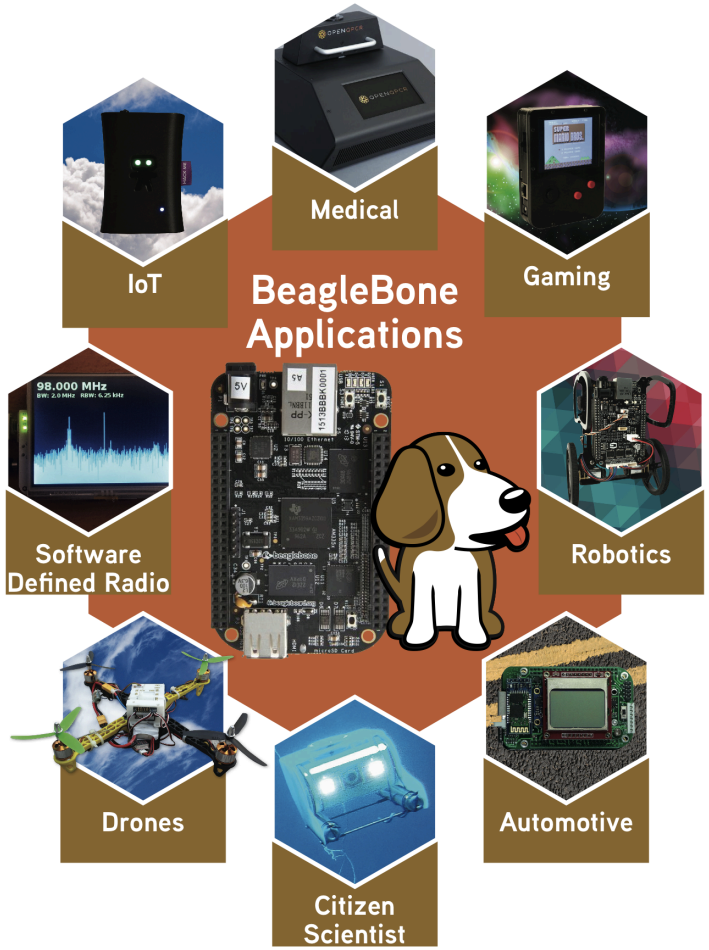


Cloud9 IDE hosted locally

Zero install and exposes command-line



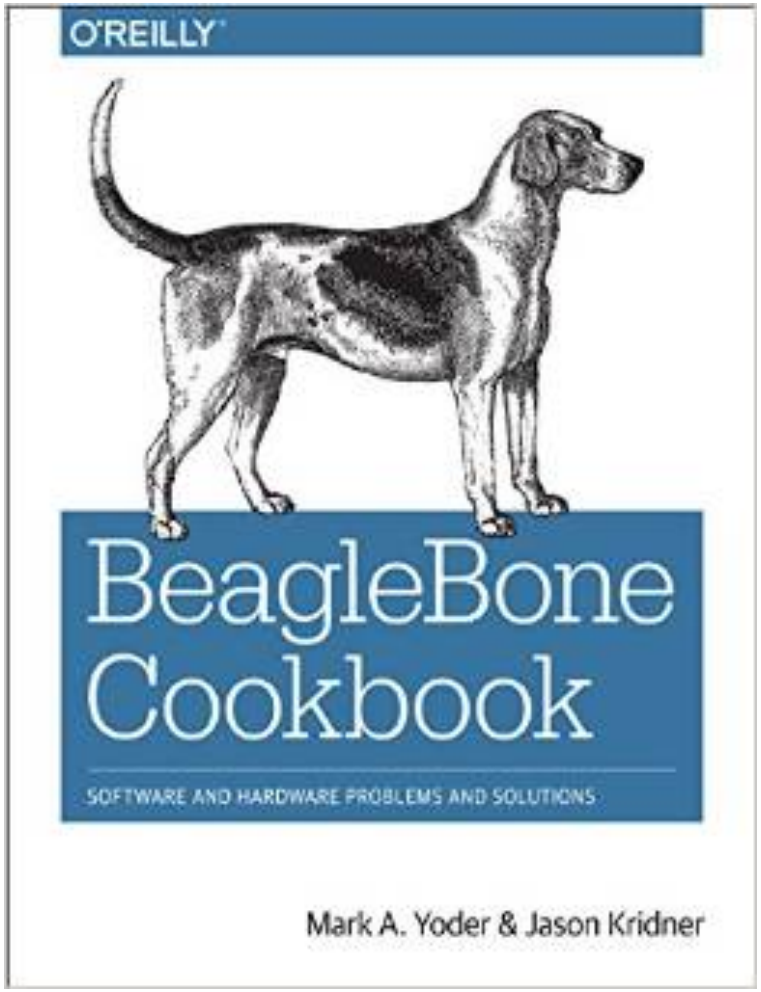
10,000s of developers building connected devices today



- 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



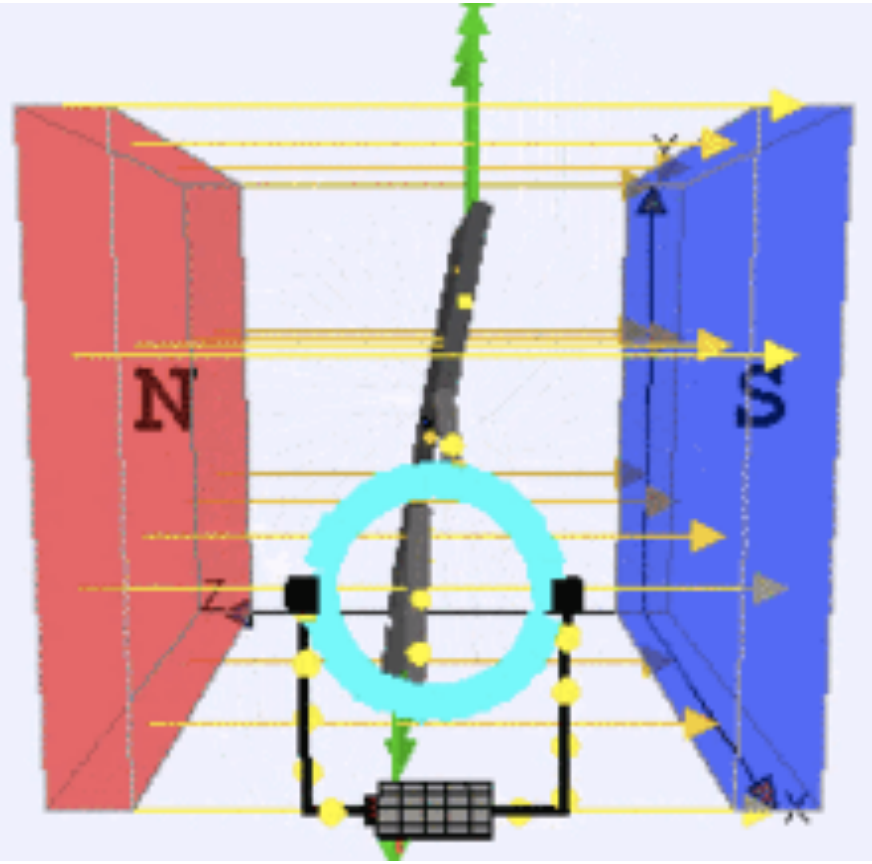
- 99 recipes covering
  - Basics
  - Sensors
  - Displays and outputs
  - Motors
  - Internet of things
  - Kernel
  - Real-time I/O
  - Capes

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
  - L293D H-Bridge IC
  - 5V DC motor
    - For other voltages, verify H-bridge compatibility
  - Breadboard and jumper wire
    - Alternatively, I've had a PCB fabricated

Direct Current (DC) Motor

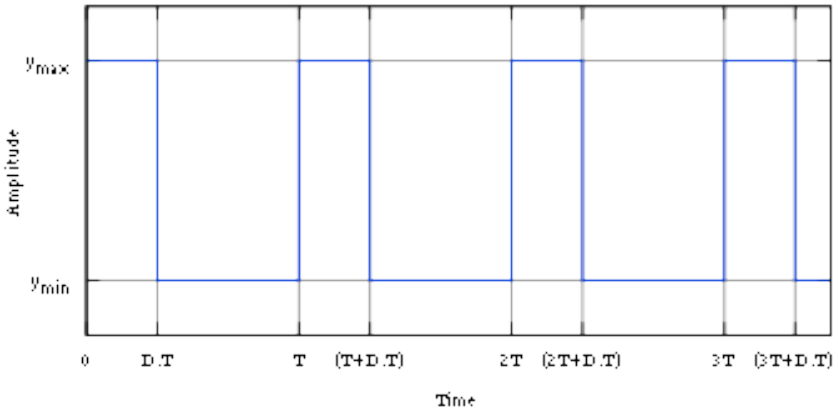
https://en.wikipedia.org/wiki/DC_motor



- DC voltage causes motor to turn
- Brush contact resets drive after partial revolution
- Drive strength is proportional to input voltage
- There's a maximum input voltage
- Reversing voltage reverses direction
- BeagleBone Black doesn't supply enough current on its I/O pins

Pulse-Width Modulation (PWM)

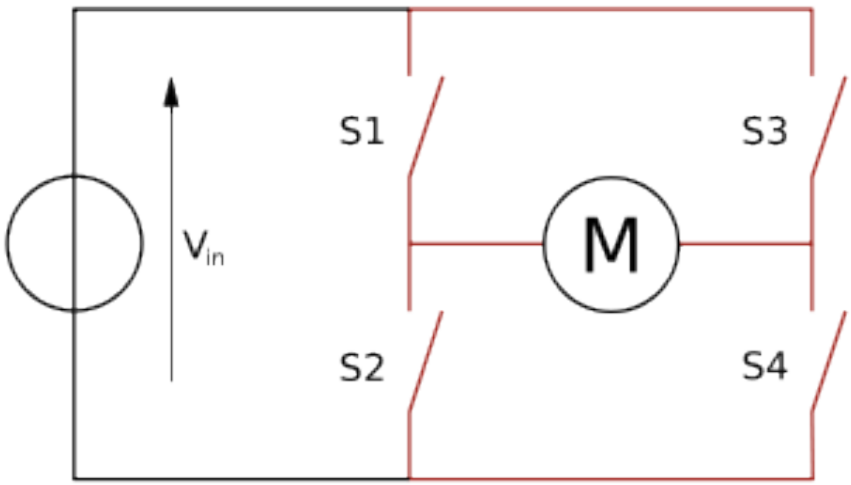
https://en.wikipedia.org/wiki/Pulse-width_modulation



- Enables approximating a voltage by turning on and off quickly
- BeagleBone Black has 8 hardware PWMs
- PRU can produce another 25 more with appropriate firmware

H-Bridge

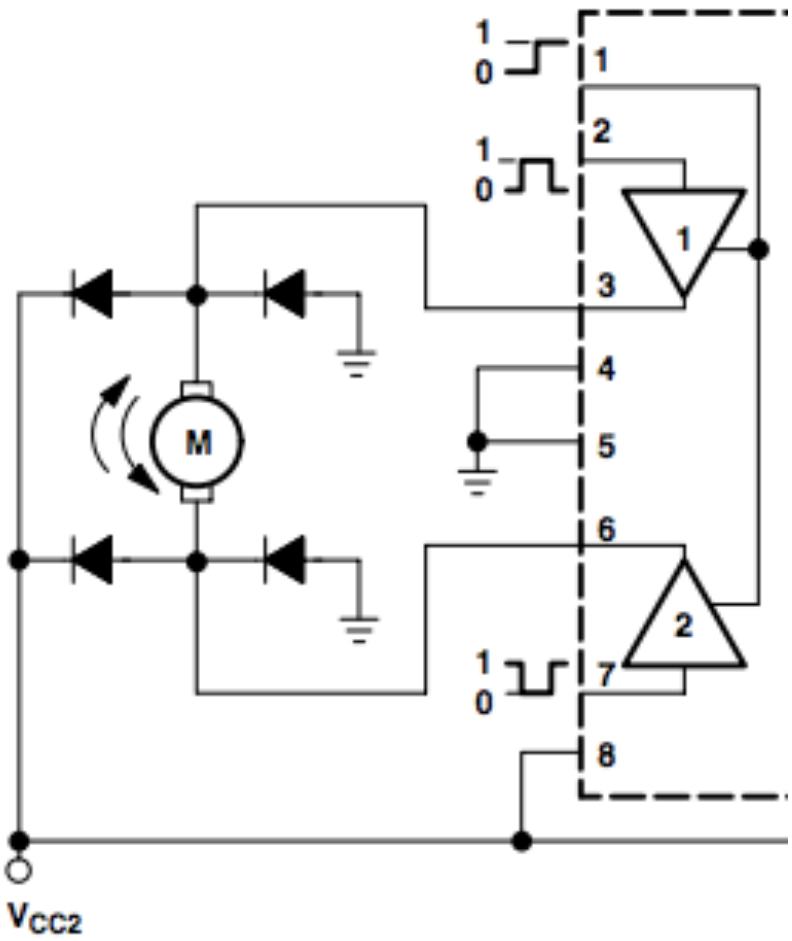
https://en.wikipedia.org/wiki/H_bridge



- Enables reversing direction of the motor
- Integrates driver as well

L293D Block Diagram

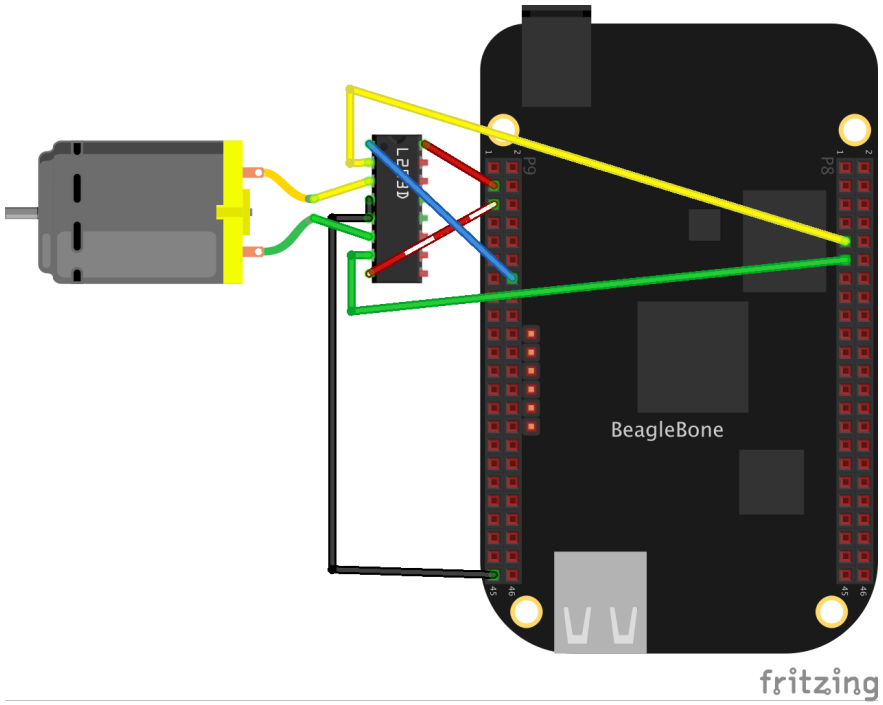
http://www.ti.com/lit/ds/symlink/l293d.pdf



- Pin 1 is the speed control
- Pin 2 is the forward drive
- Pin 7 is the backward drive

Connect your L293D H-bridge

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



- Pin 1 to P9_14 “EN”
- Pin 2 to P8_9 “FWD”
- Pin 3 to “Motor +”
- Pin 4 and 5 to DGND
- Pin 6 to “Motor -”
- Pin 7 to P8_11 “BWD”
- Pin 8 to VDD_5V
- Pin 9 to VDD_3V3

Recipe 4.3: Controlling the motor

https://github.com/BeagleBoneCookbook/firstEdition/blob/master/04motors/h-bridgeMotor.js

```
var b = require('bonescript');
var motor = { SPEED: 'P9_14', FORWARD:
'P8_9', BACKWARD: 'P8_11' };
var FREQ = 50;
var STEP = 0.1;
var count = 0;
var stop = false;

b.pinMode(motor.FORWARD, b.OUTPUT);
b.pinMode(motor.BACKWARD, b.OUTPUT);
b.analogWrite(motor.SPEED, 0, FREQ, 0, 0);

var timer = setInterval(updateMotors, 100);

function updateMotors() {
    var speed = Math.sin(count*STEP);
    count++;
    Mset(motor, speed);
}
```

- Define the pins
- Keep track of state
- Setup pins initially
- Use a 100ms timer to update the motors
- Use a sine wave to increment/decrement the speed for test
- Call ‘Mset’ to update the PWM and direction

Recipe 4.3: Controlling the motor

https://github.com/BeagleBoneCookbook/firstEdition/blob/master/04motors/h-bridgeMotor.js

```
function Mset(motor, speed) {
    speed = (speed > 1) ? 1 : speed;
    speed = (speed < -1) ? -1 : speed;
    //console.log("Setting speed = " + speed);
    b.digitalWrite(motor.FORWARD, b.LOW);
    b.digitalWrite(motor.BACKWARD, b.LOW);
    if(speed > 0) {
     b.digitalWrite(motor.FORWARD, b.HIGH);
    } else if(speed < 0) {
     b.digitalWrite(motor.BACKWARD, b.HIGH);
    }
    b.analogWrite(motor.SPEED,
                        Math.abs(speed), FREQ);
}
```

- Put a cap on the maximum and minimum at 1 and -1
- Set the drive signals for direction
- Adjust PWM based upon the speed

Recipe 4.3: Controlling the motor

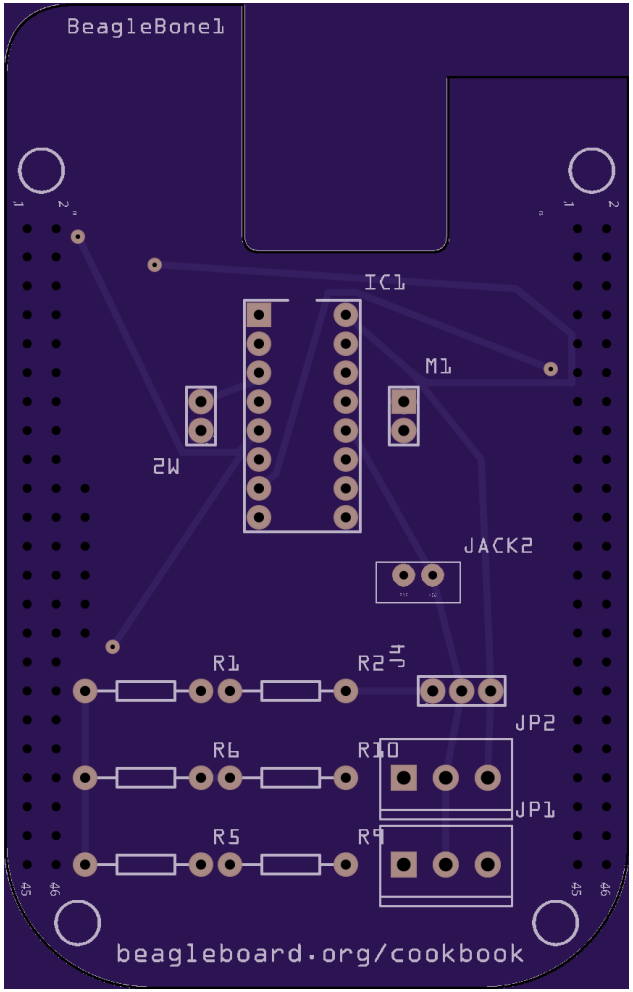
https://github.com/BeagleBoneCookbook/firstEdition/blob/master/04motors/h-bridgeMotor.js

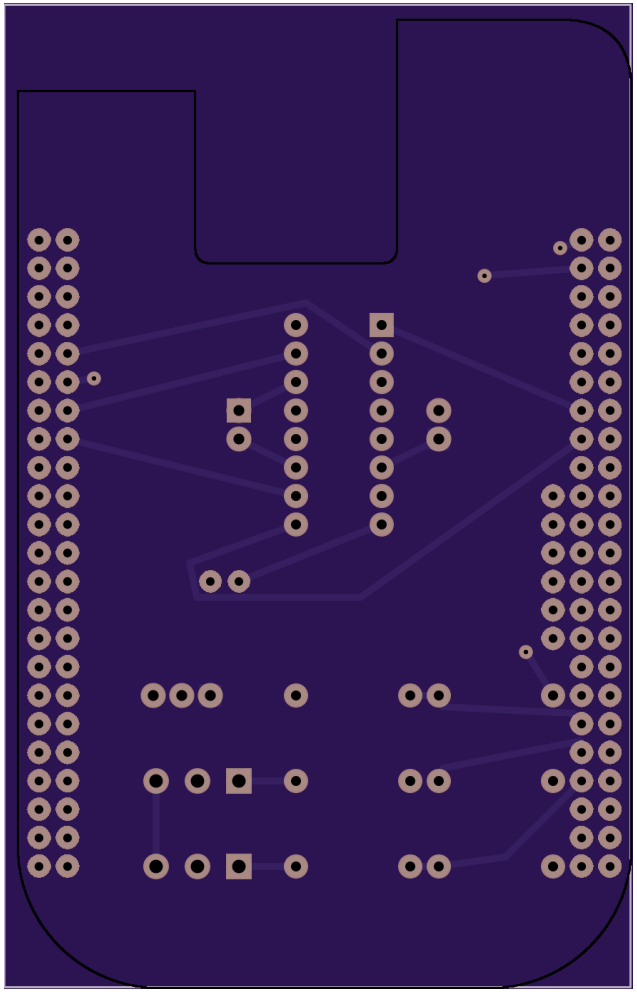
```
function doStop() {
    clearInterval(timer);
    Mset(motor, 0);
}

process.on('SIGINT', doStop);
```

- Detect when program is being stopped by a ^C
- Stop the timer and disable the motor

My quick-hack PCB

See recipe 9.7





More

- Learn more about H-Bridges and motors
  - https://itp.nyu.edu/physcomp/lessons/dc-motors/dc-motors-the-basics/
- My simple PCB
  - https://oshpark.com/shared_projects/Mz40o0aN
- Shortcuts to updates and examples from the book
  - http://beagleboard.org/cookbook