
PocketBeagle Walk Through
SanDisk
beagleboard.org®
e-ale

PocketBeagle walk-through

Friendly to novices and experts alike, the Beagle experience tracks mainline u-boot, Linux and Debian development, while augmenting it to enable development to start as quickly as possible. Attendees will get started interacting with the hardware via the command-line, shell scripts, Python and JavaScript. Attendees will be walked through the configuration details for the boot configuration, pin multiplexing, USB networking and other helper scripts they should get to know. Support and development processes within the BeagleBoard.org community will be covered. Exercises will pave the way for the other workshops to dive into their topic without needing to backtrack excessively on PocketBeagle-specific details.

Author and license

- Author
  - Jason Kridner
    Co-founder BeagleBoard.org, Texas Instruments Sitara apps
    https://beagleboard.org/about
- License
  - Creative Commons Attribution – Share Alike 4.0
    https://creativecommons.org/licenses/by-sa/4.0/

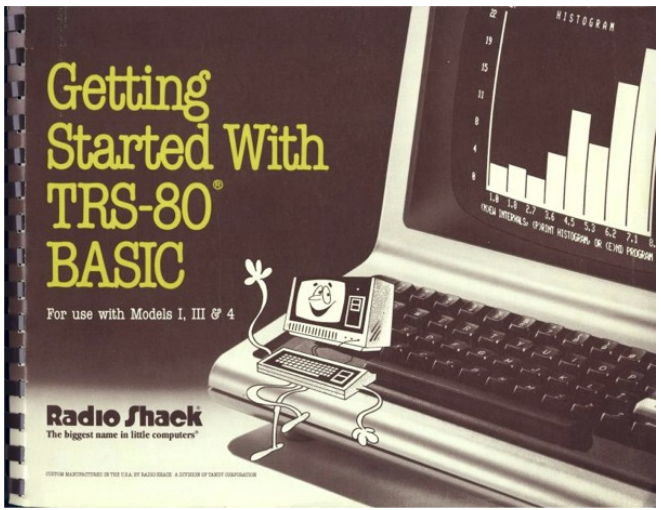
Outline

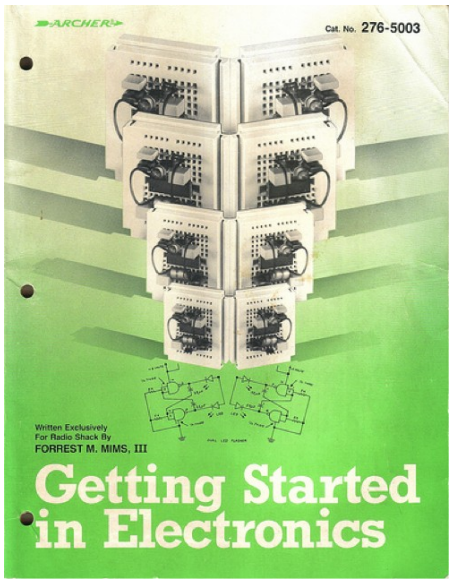
- BeagleBoard.org, PocketBeagle and BaconBits
- Developer experience
  - Command-line and shell script
  - JavaScript and Python
  - C/C++
  - C on PRUs
- Project examples
- Labs

BeagleBoard.org's objectives

- Education
  - Design and use of open source SW/HW
  - Embedded computing
- Collaboration
  - Physical computing
  - Robotics
  - Industrial/machine controls

Inspiration from early PCs

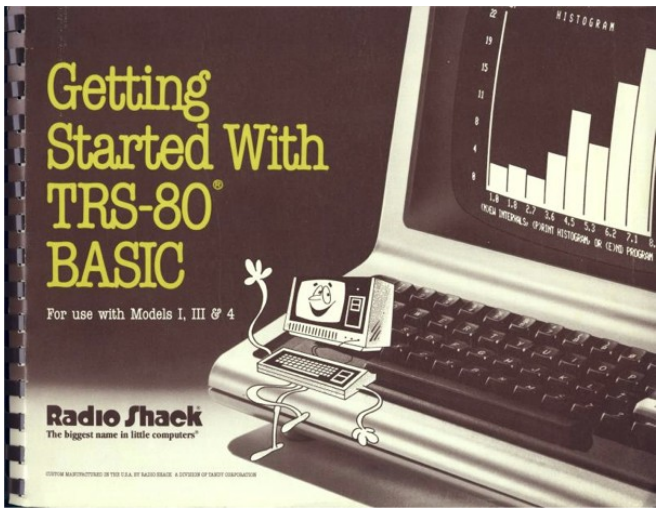


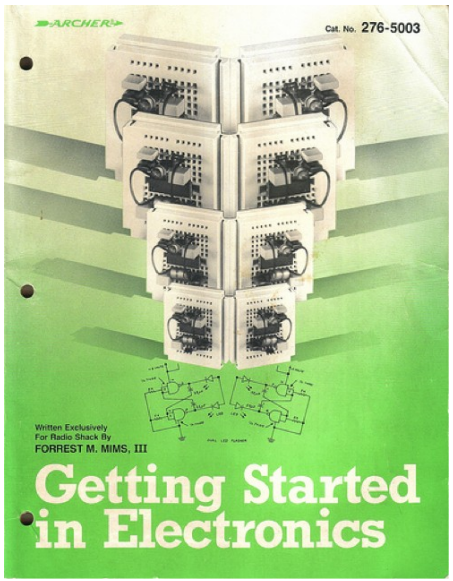


http://newcome.wordpress.com/2009/12/15/make-electronics-the-new-engineers-notebook/
http://www.sandywalsh.com/2012_07_01_archive.html

Inspiration from early PCs

- How do people learn about embedded computers with so much ground to cover?



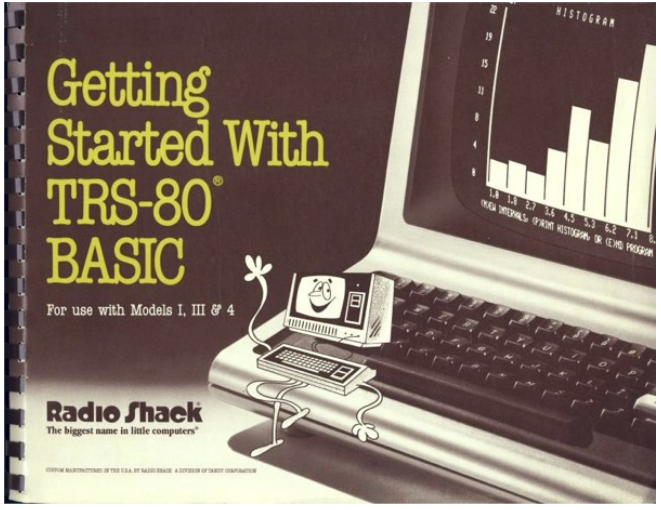


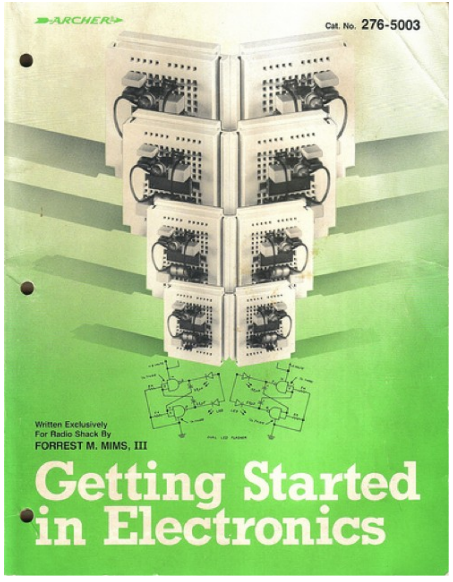
http://newcome.wordpress.com/2009/12/15/make-electronics-the-new-engineers-notebook/
http://www.sandywalsh.com/2012_07_01_archive.html

Inspiration from early PCs

- How do people learn about embedded computers with so much ground to cover?

- Linux keeps history
- Affordable -> hackable
- Open from boot
- High-level languages
- Motivate with hardware





http://newcome.wordpress.com/2009/12/15/make-electronics-the-new-engineers-notebook/
http://www.sandywalsh.com/2012_07_01_archive.html

Vision

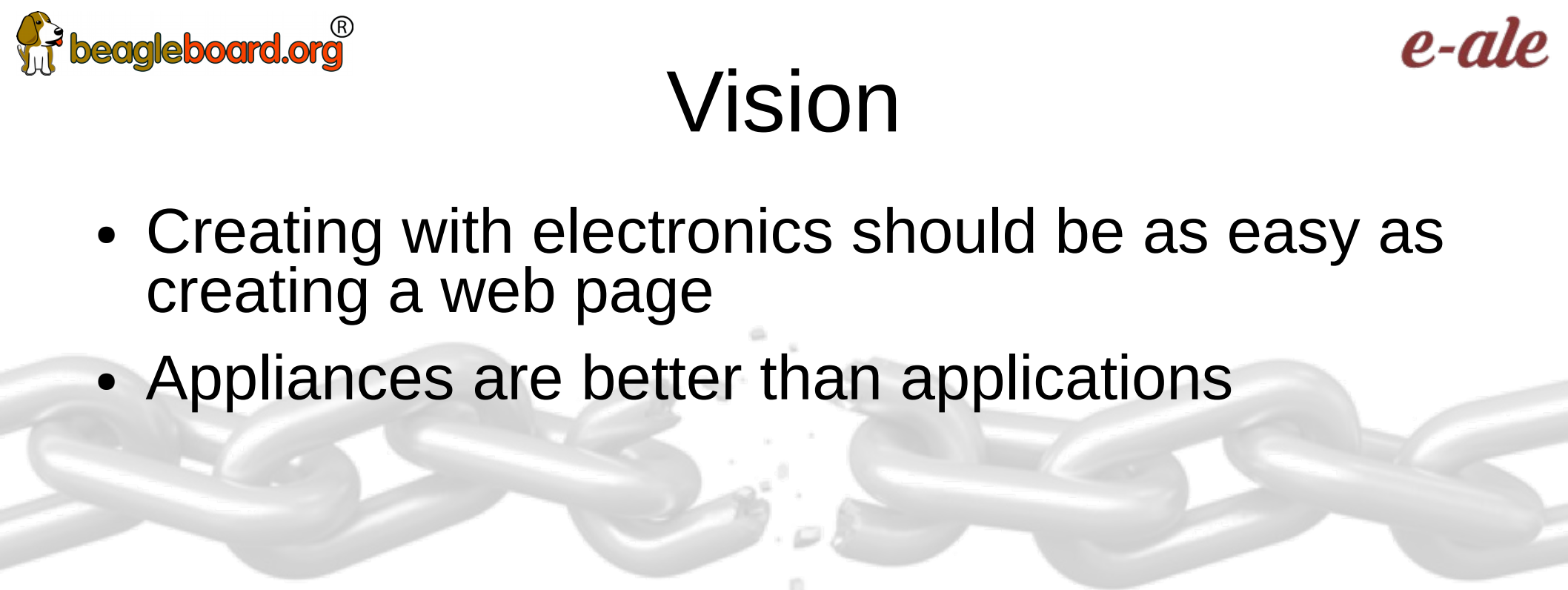
- Creating with electronics should be as easy as creating a web page

beagleboard.org®

e-ale

Vision

- Creating with electronics should be as easy as creating a web page
- Appliances are better than applications



open source hardware

open source™

Vision

- Creating with electronics should be as easy as creating a web page
- Appliances are better than applications
- Open source software and hardware enable
  - Collaboration on the problem
  - Ability to understand and improve the fundamentals

Reality

- Boot-to-browser feels too limiting → booting to Debian distro
- Collaborative programming still complex → collaborate at the kernel
- Many possible development environments
  - command-line/ssh, Cloud9 IDE, node-red, pureData, SuperCollider, LabView, Matlab, Eclipse, Visual Studio, Scratch, Blockly
- Domain specific approaches
  - Machinekit/LinuxCNC, PLC, many IoT toolkits
  - Many rapid sensor approaches: capes, mikroBus, Grove/Grove Zero, PMOD
  - Many rapid build approaches: LEGO, printing/milling, Makeblock, Vex, various other aluminum kits

Approach

- Don’t try to boil the ocean
  - We seek to engage the open source community
- Help where we can
  - Blue supports Grove cables
  - PocketBeagle supports mikroBus click pinout
  - Many “BeagleBoard Compatible” devices targeting specific application areas

Board history

Fanless open computer (BeagleBoard)

$249

Mint tin sized with industrial peripherals (BeagleBone)

$69

Application focused BeagleBones

$79

Smalls mint tin sized with super-flexible design - PocketBeagle

$25

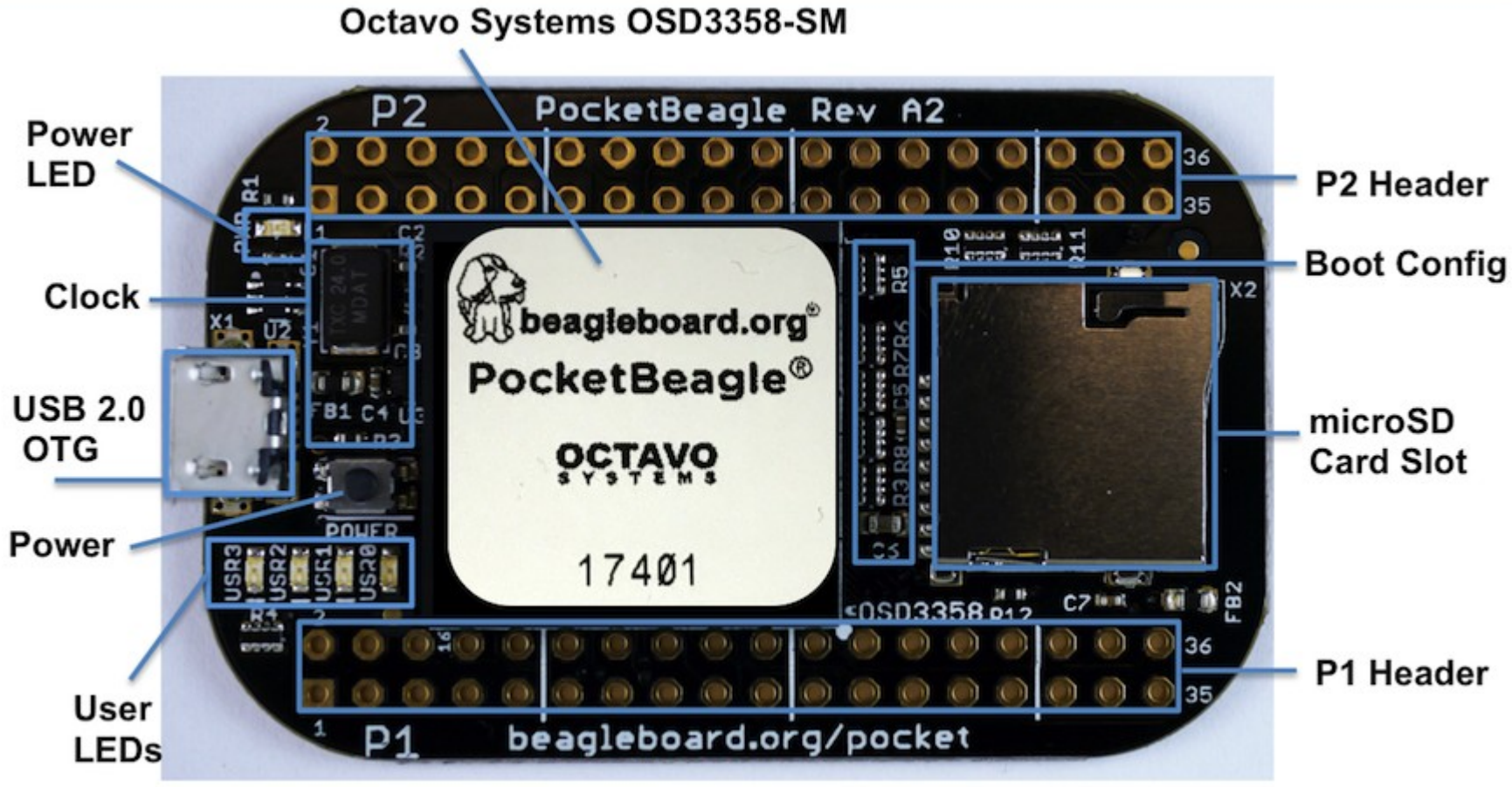
PocketBeagle objectives

- Get simple
  - 4-layer PCB done in both Kicad and EAGLE
  - Every expansion header pin has a useful predefined mode
- Get flexible
  - USB to holes, no on-board pin consumption, no header soldered
  - Support for 2 mikroBus Click boards (over 300 already exist)
- Get small
  - Stick with mint-tin survival-kit theme, but go to “smalls” (35mm x 55mm)
- Get low cost
  - System-in-package approach has can lower build costs
  - Launched/sustainable at $25

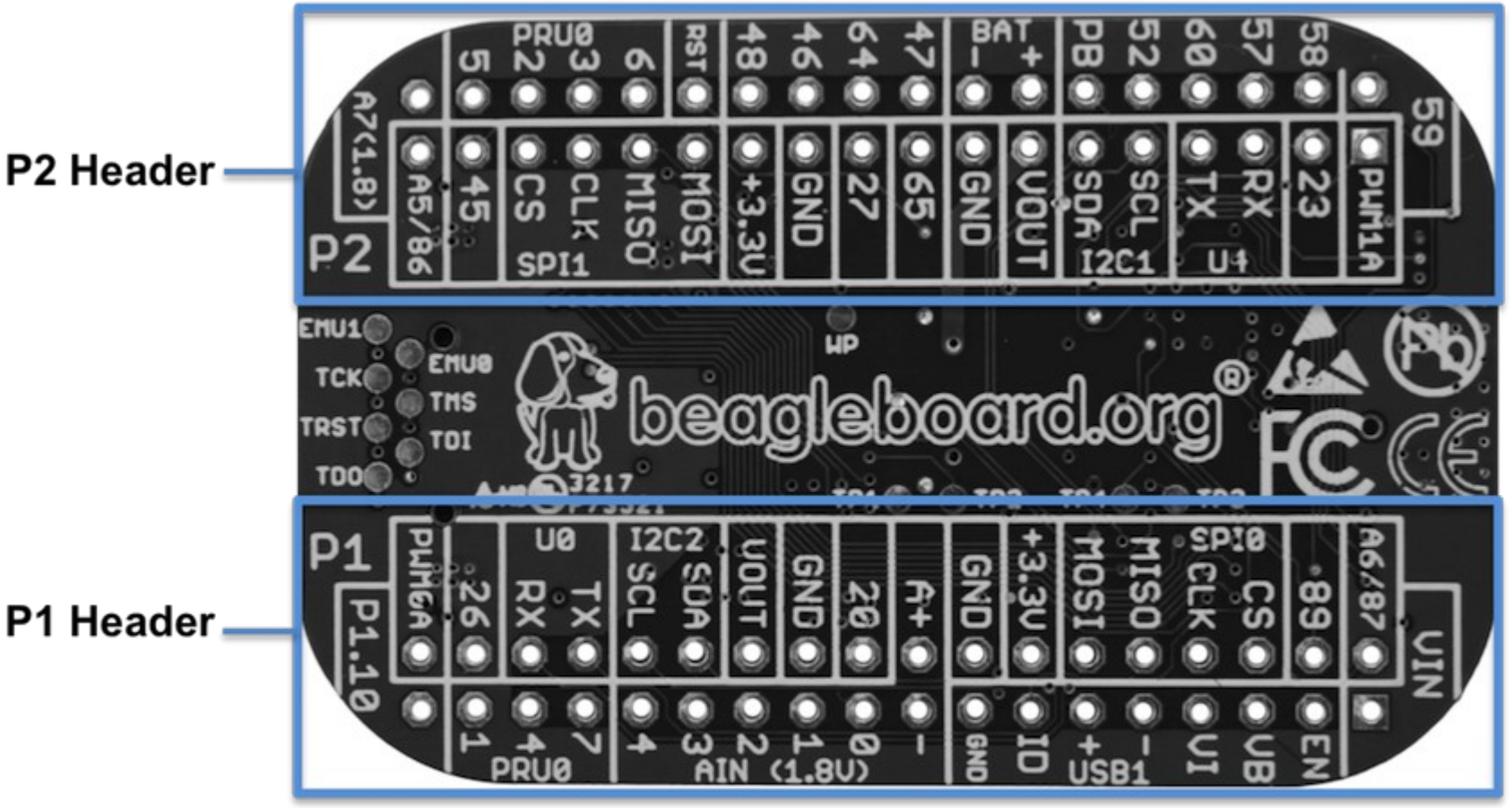
PocketBeagle key features

- Processing
  - 1-GHz ARM Cortex-A8 processor
  - 2x200-MHz programmable real-time units (PRUs)
  - ARM Cortex-M3 microcontroller for power and security
  - SGX530 graphics processor (OpenGLES)
- Memory
  - 512-MB DDR3
  - 4-KB I2C EEPROM
- Interfaces
  - USB 2.0 OTG
  - microSD
- 72 expansion header pins
  - 8 analog inputs (6@1.8V, 2@3.3V)
  - 44 digital I/Os (18 enabled)
  - 3 UARTs (2 enabled)
  - 2 I2C ports
  - 2 SPI ports
  - 2 quadrature encoders accessible
  - 2 CAN bus controllers accessible
  - USB, power/reset buttons, battery/DC

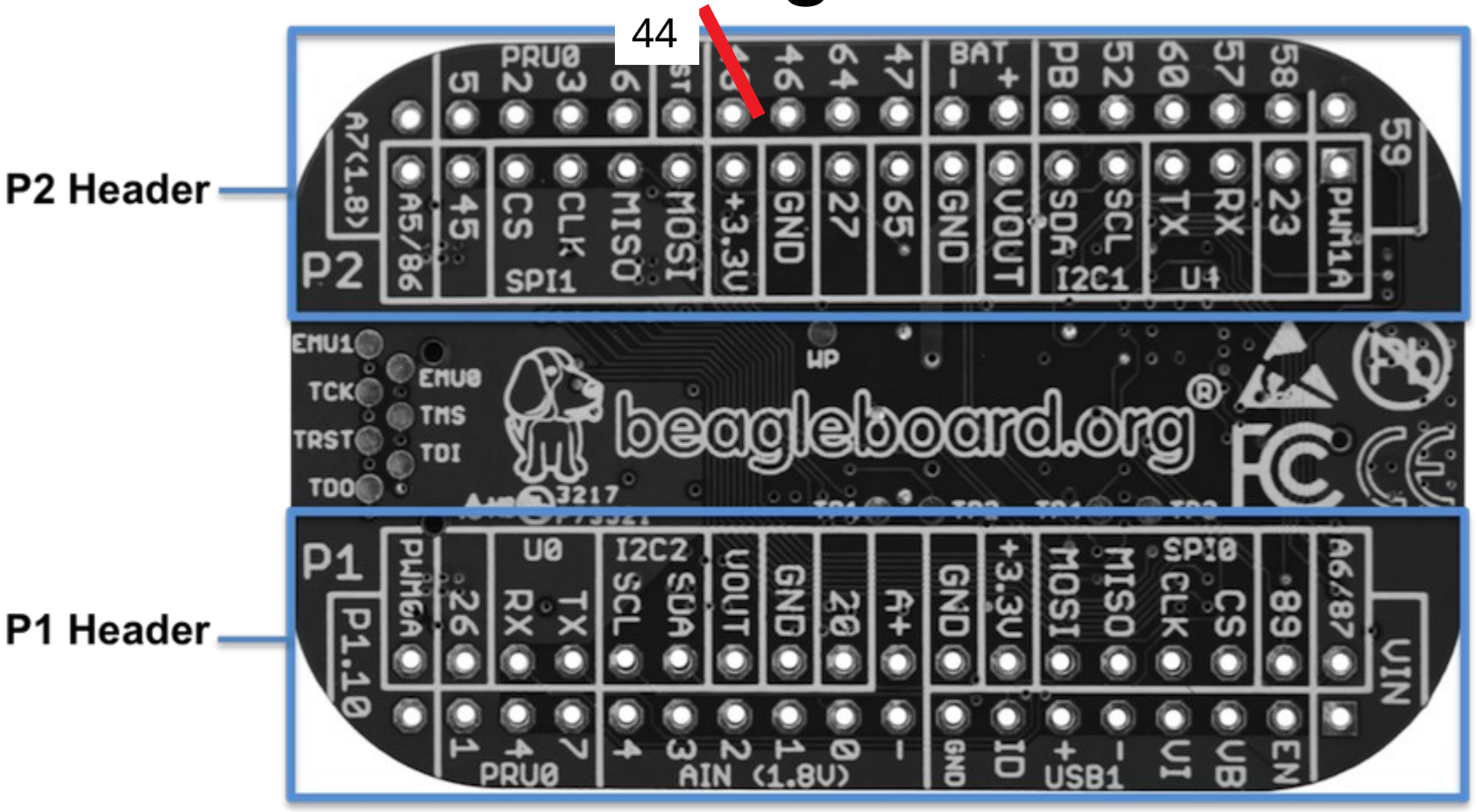
PocketBeagle top



PocketBeagle bottom

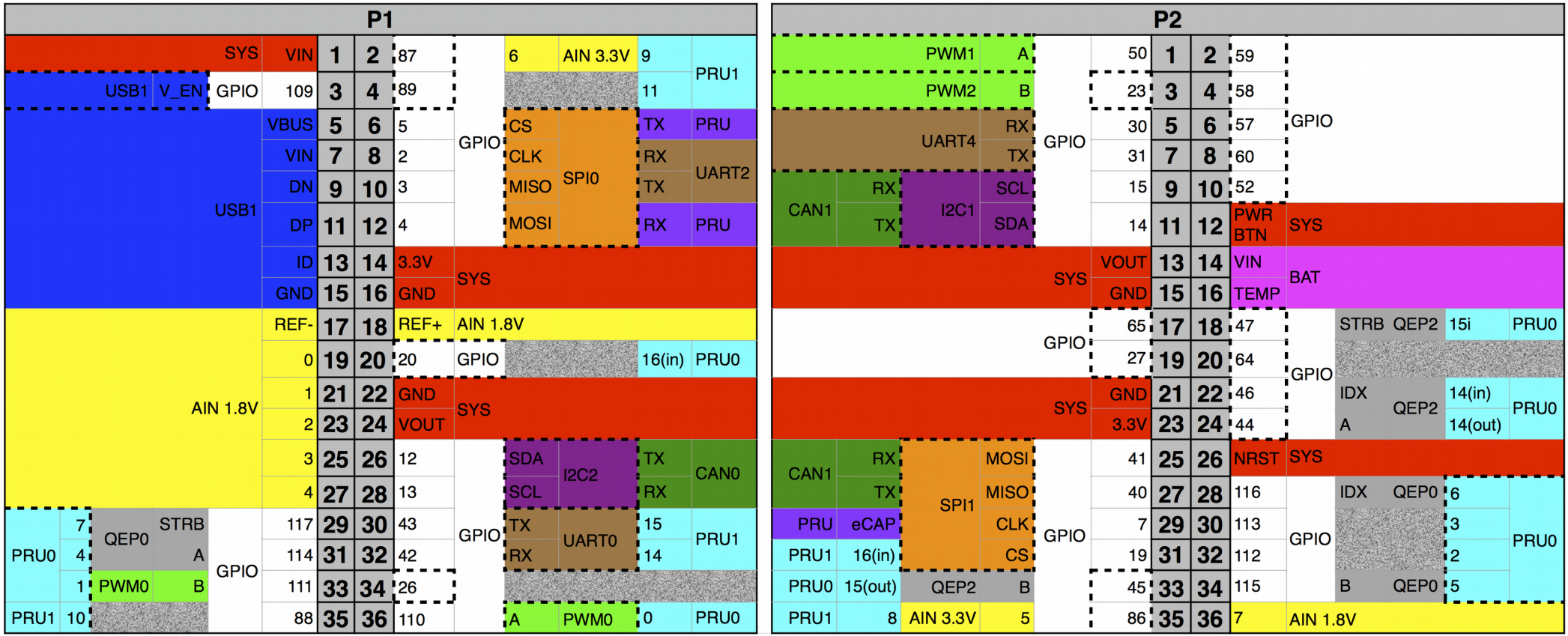


PocketBeagle bottom

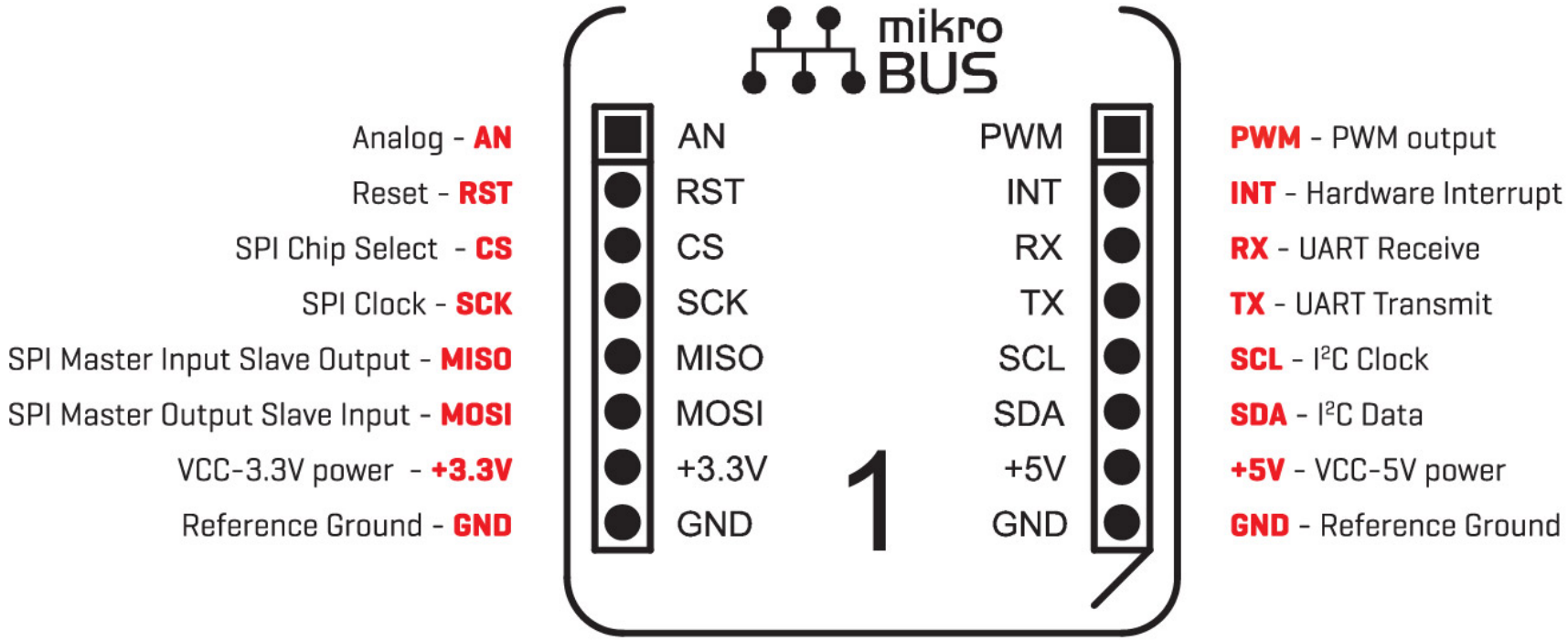


PocketBeagle expansion

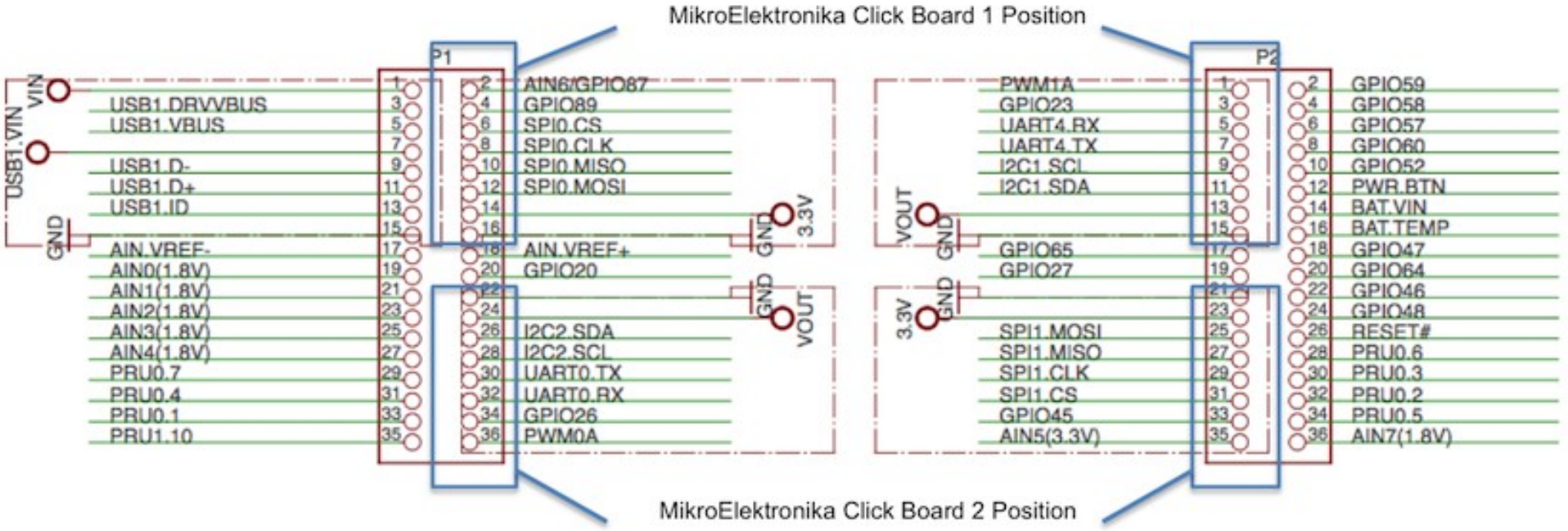
PocketBeagle Expansion Headers (Rev A2a)



mikroBus Click



Connecting mikroBus Clicks



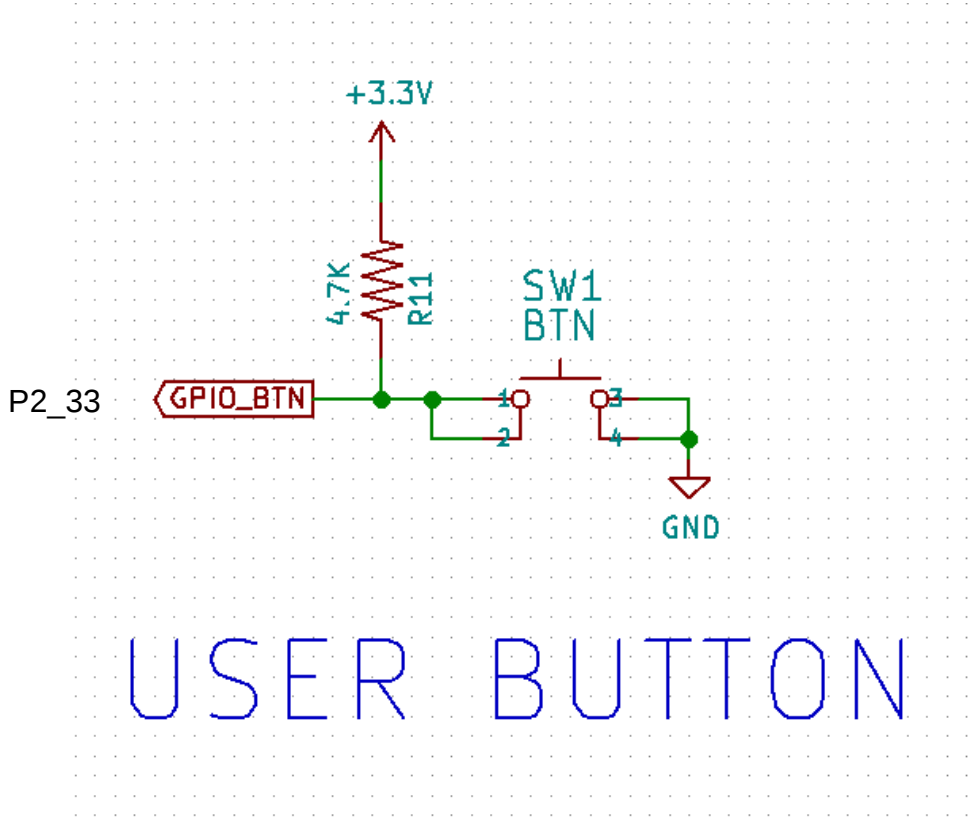
BaconBits objectives

- Designed specifically for e-ale training
- Inspired by “Bacon Cape” by Dave Anders
  - Designed for similar purpose on BeagleBone
- Provides target for common embedded interfaces
  - SPI, I2C, GPIO, PWM, ADC, USB, serial
- Avoided buying several modules

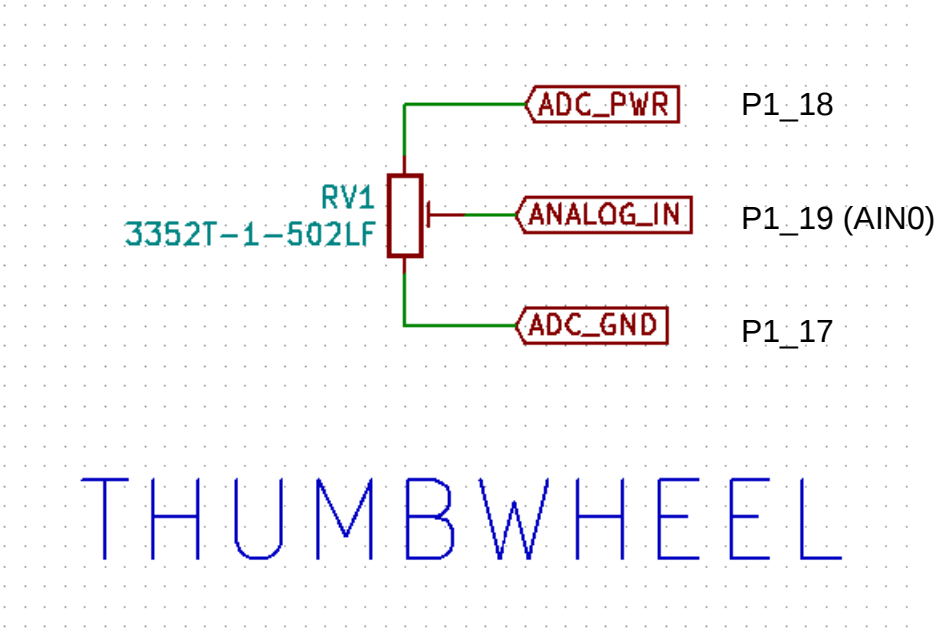
BaconBits features

- USB-to-Serial micro B
- USB Host A with power
- Power and Reset buttons
- GPIO push button
- ADC potentiometer thumbwheel
- PWM tri-color LED
- SPI 2-digit 7-segment display
- I2C accelerometer

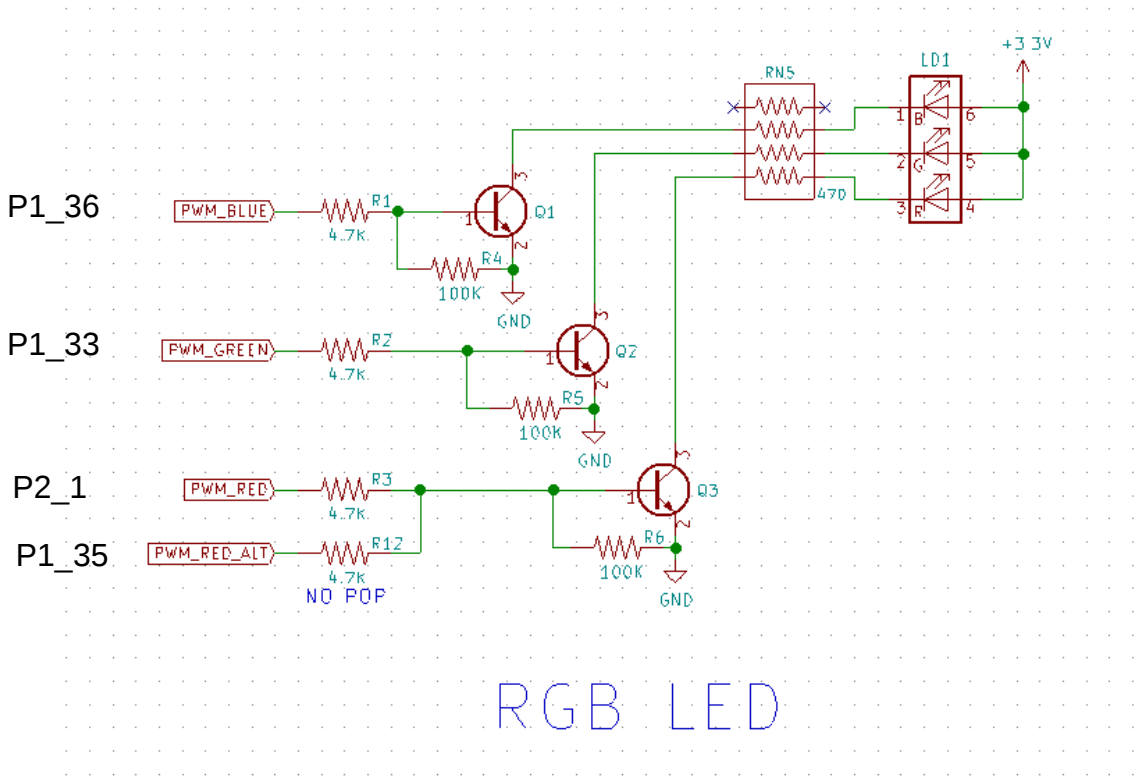
BaconBits GPIO button



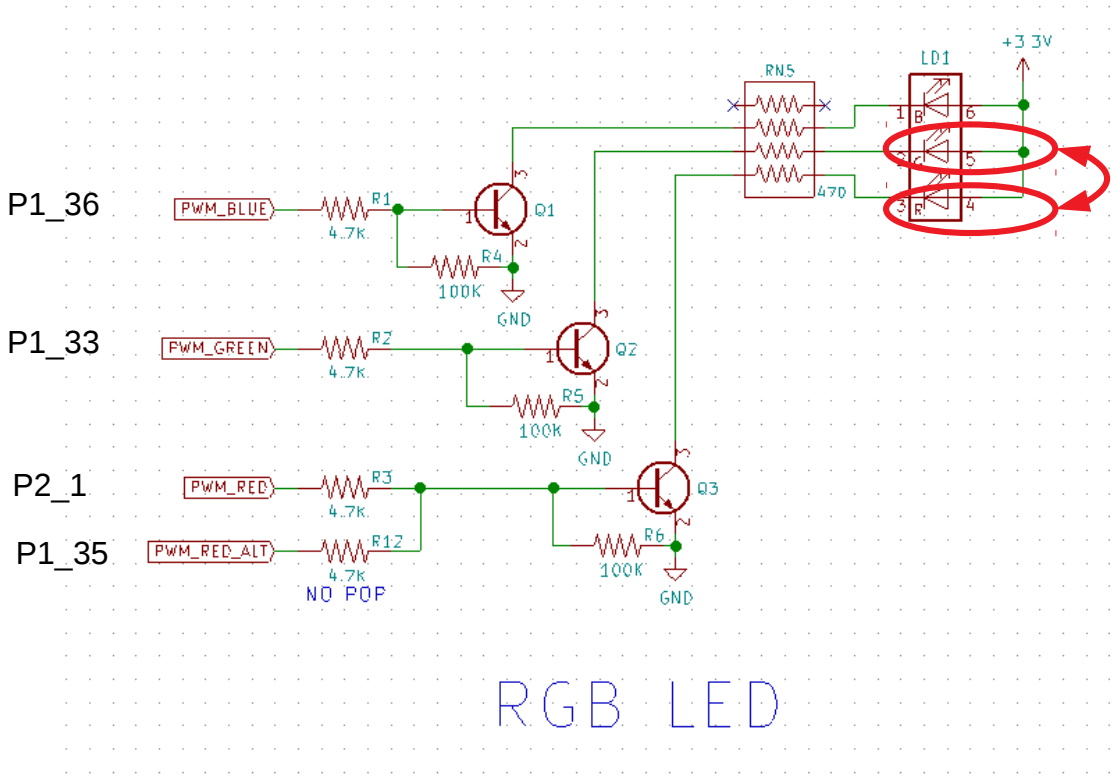
BaconBits ADC thumbwheel



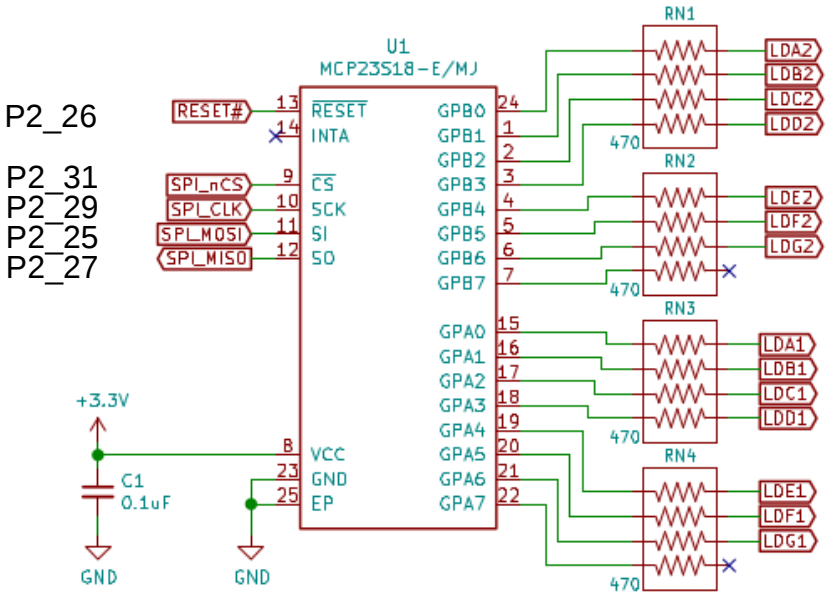
BaconBits RGB LED

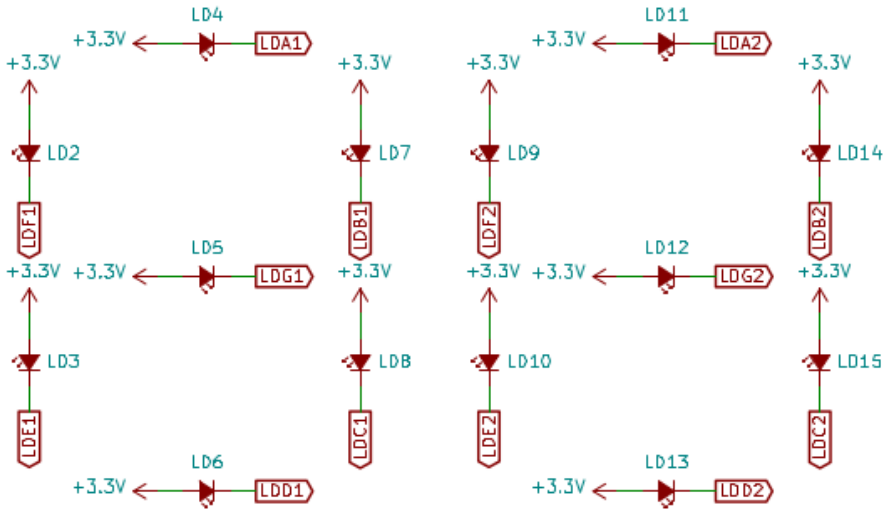


BaconBits RGB LED



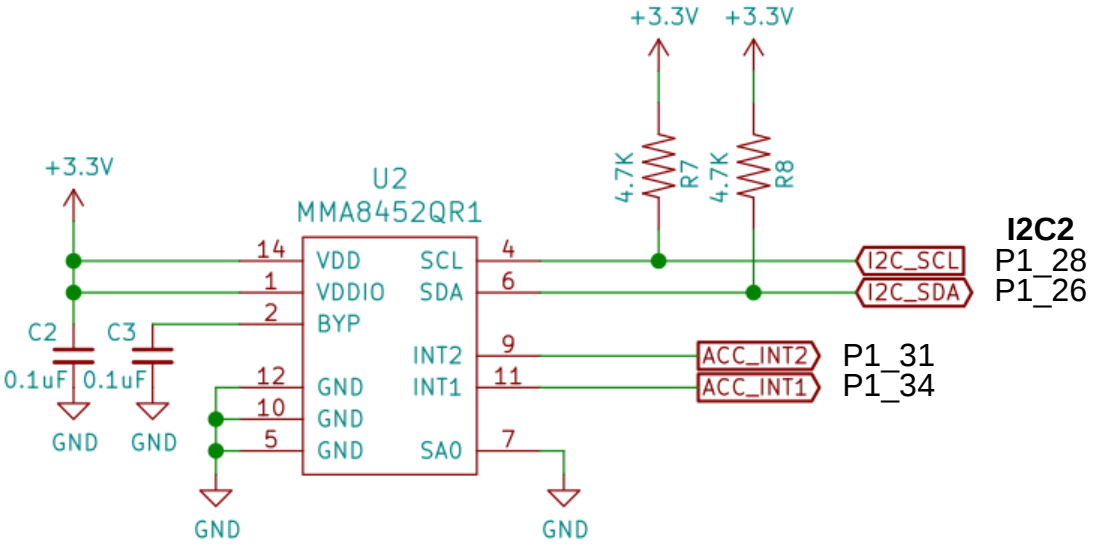
BaconBits SPI 7-segment display





EMULATED 7 SEGMENTS

BaconBits I2C acceleromter



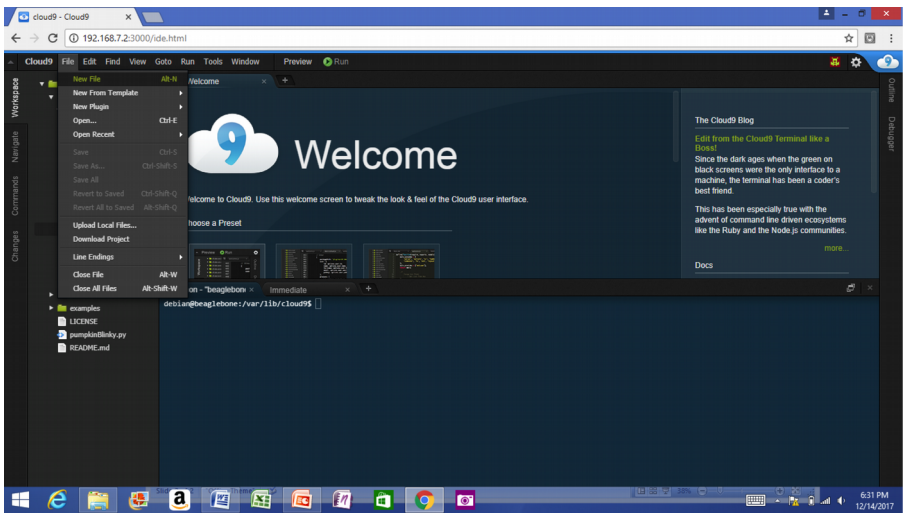
ACCELEROMETER

Developer experience

- Customized Debian images – bbb.io/latest
- Self-hosted tools for ARMs and PRU
- Libraries for various high-level languages
- Scripts for common tasks
- Sources for bootloader, device tree, etc.
- Servers for network-based development

Single cable development

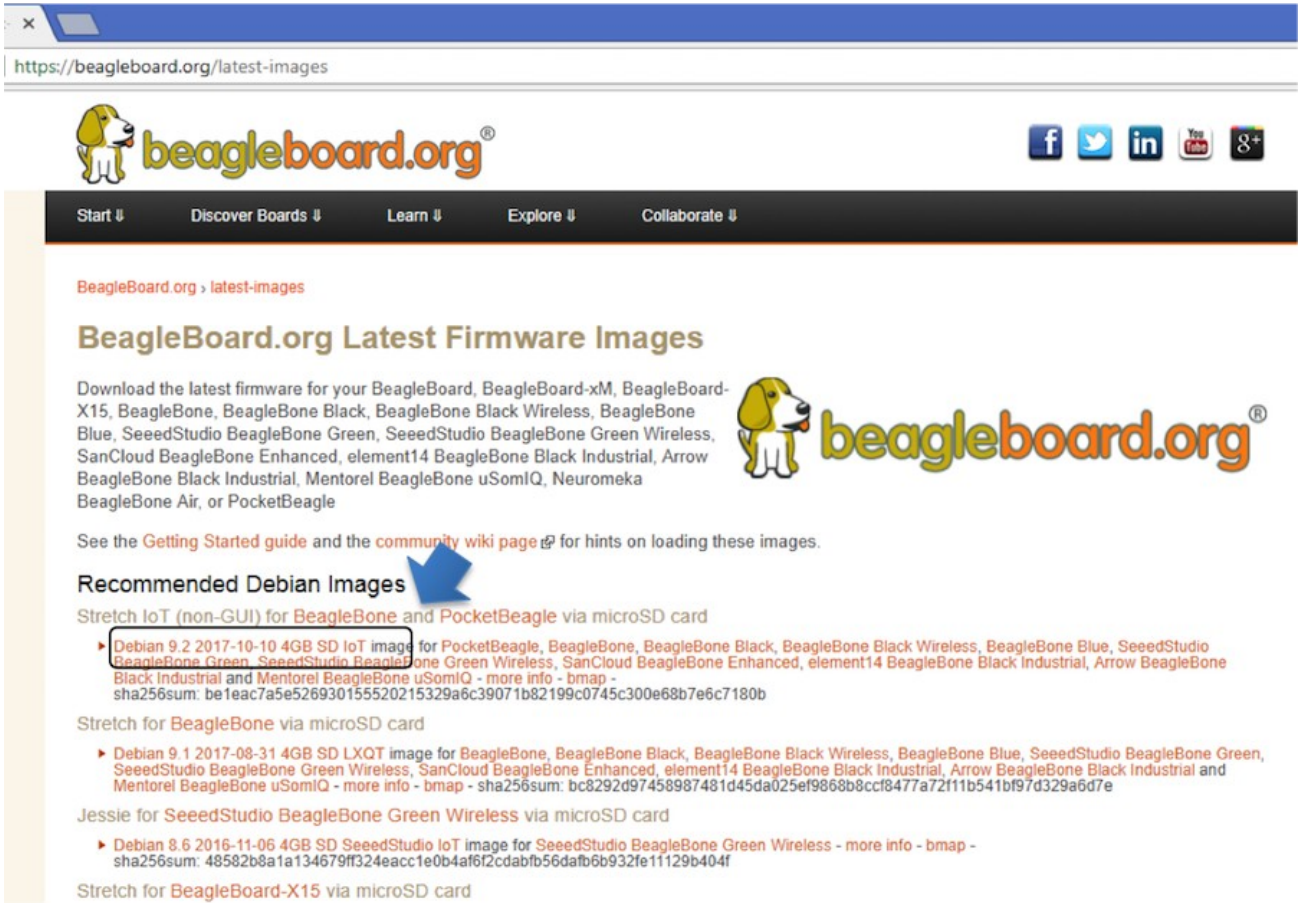
- Power, network, develop
- You can add a network and power many other ways



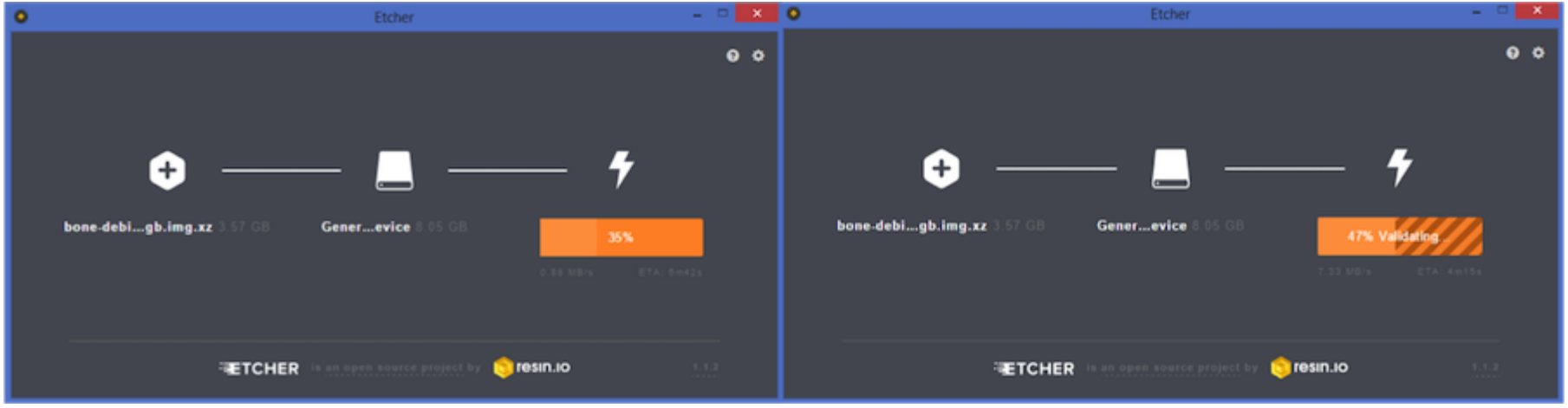
Some work in progress

- Add proxy for various services (in Buster IoT images today)
- Integrate common web-based WiFi provisioning
  - SeeedStudio BealgeBone Green Wireless ships with ‘wifidog’ → we will unify approach
- Cross-platform distro installer app
  - See USB NETCONSOLE presentation
- Support for Grove modules and mikroBus clicks
  - Focus on device-tree overlays and kernel patches
- Integration alignment with complete domain solutions
  - Intelligent Agent Replicape/Revolve, Bela Mini, BeagleLogic, etc.
- Improved and integrated PRU examples
- Move to distro friendly approaches for customizations

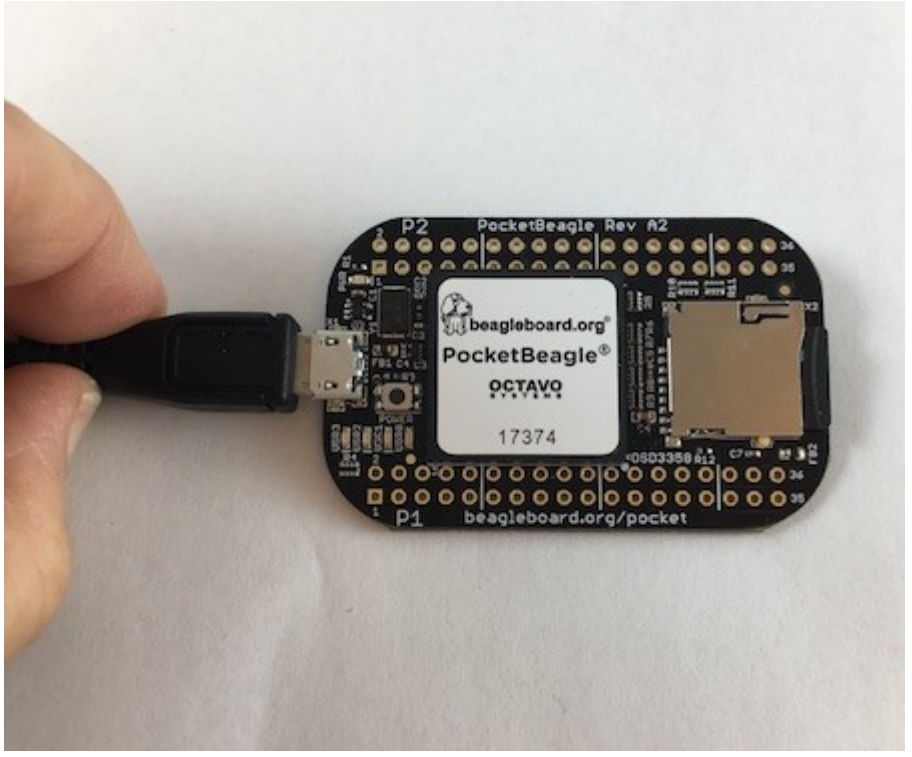
Download image



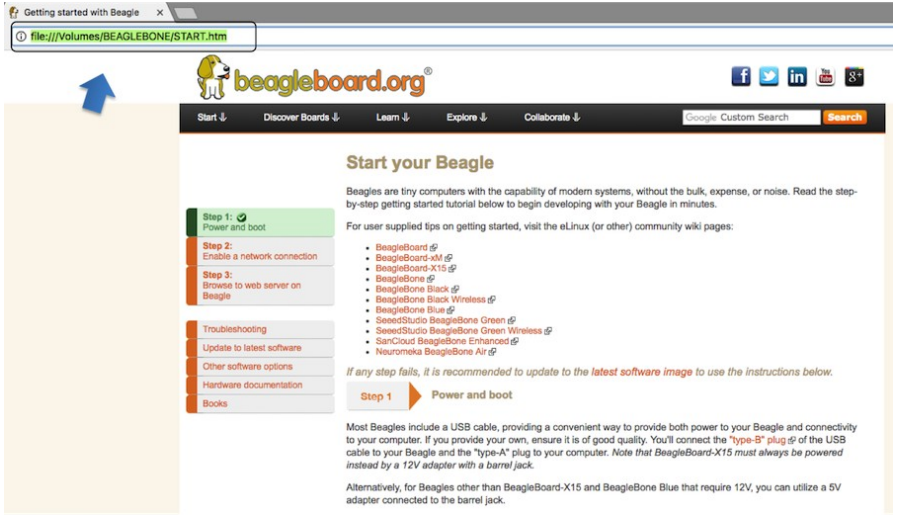
Write image to microSD with Etcher

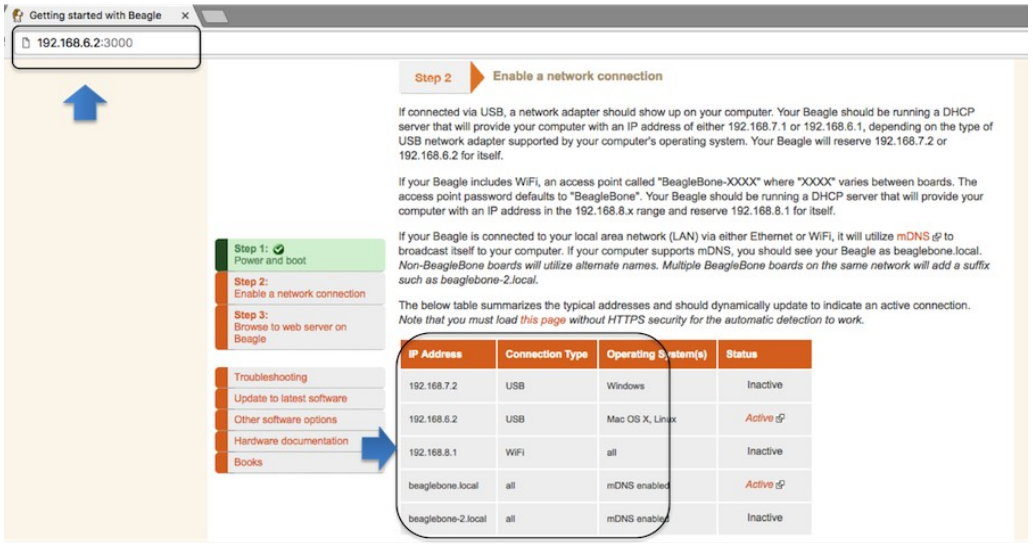


Insert microSD and boot

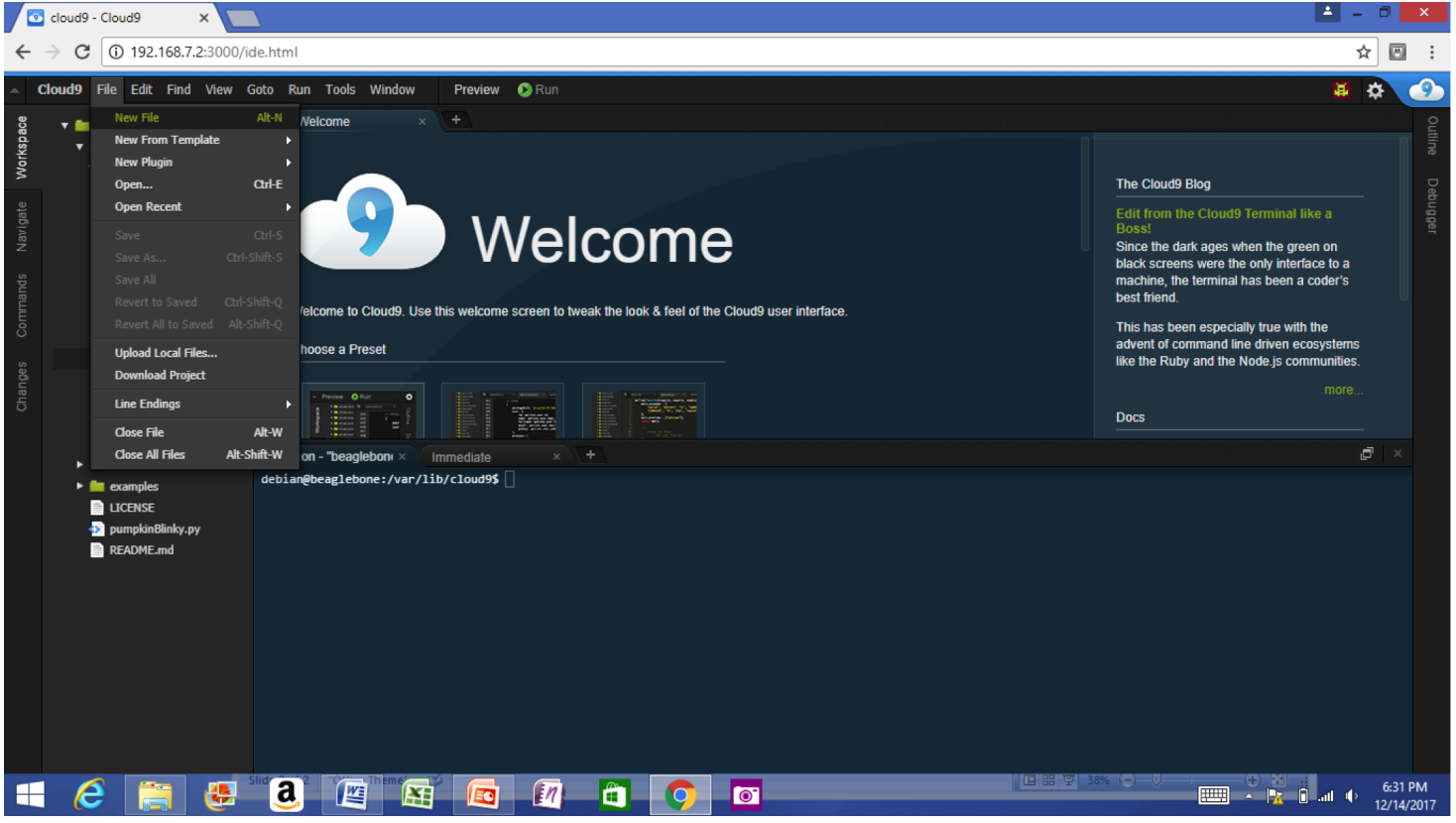


Connect to the USB network





Open the IDE


cloud9 - Cloud9
192.168.7.2:3000/ide.html
Cloud9
File
Edit
Find
View
Goto
Run
Tools
Window
Preview
Run
New File Alt-N
New From Template
New Plugin
Open... Ctrl-E
Open Recent
Save Ctrl-S
Save As... Ctrl-Shift-S
Save All
Revert to Saved Ctrl-Shift-Q
Revert All to Saved Alt-Shift-Q
Upload Local Files...
Download Project
Line Endings
Close File Alt-W
Close All Files Alt-Shift-W
Welcome
Welcome
elcome to Cloud9. Use this welcome screen to tweak the look & feel of the Cloud9 user interface.
hoose a Preset
The Cloud9 Blog
Edit from the Cloud9 Terminal like a Boss!
Since the dark ages when the green on black screens were the only interface to a machine, the terminal has been a coder's best friend.
This has been especially true with the advent of command line driven ecosystems like the Ruby and the Node.js communities.
more...
Docs
Immediate
debian@beaglebone:/var/lib/cloud9$
examples
LICENSE
pumpkinBlinky.py
README.md
Workspace
Navigate
Commands
Changes
Outline
Debugger
6:31 PM
12/14/2017

USB gadgets

- Linux name for device/slave drivers
  - ie., when not host
- USB devices have “classes”
  - Mass storage
  - Camera
  - Audio
  - Printer
  - “HID” or human-interface device like mouse and keyboard
  - Communications

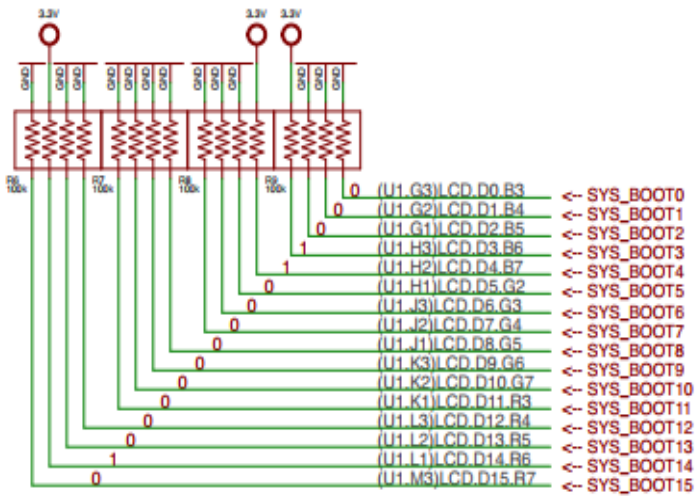
USB gadgets

- Default image USB gadgets
  - Virtual mass storage
    - Serves you up README.htm
  - Virtual serial
    - Provides access to console after kernel boot
  - Virtual network
    - Enables access to ssh and web servers

TI AM335x: bootrom

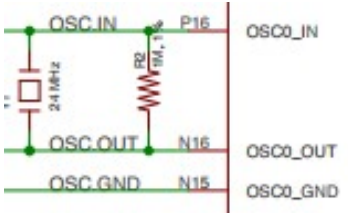
From: (AM335x and AMIC110 Sitara™ Processors Technical Reference Manual (Rev. P))

- http://www.ti.com/lit/ug/spruh73p/spruh73p.pdf (page 5032)



SYSBOOT[15:14] = 01 = 24Mhz
SYSBOOT[4:0] = 11000

1. SPI0
2. MMC0 - going to use today
3. USB0 - (node-beagle-boot)
4. UART0



TI AM335x: bootrom

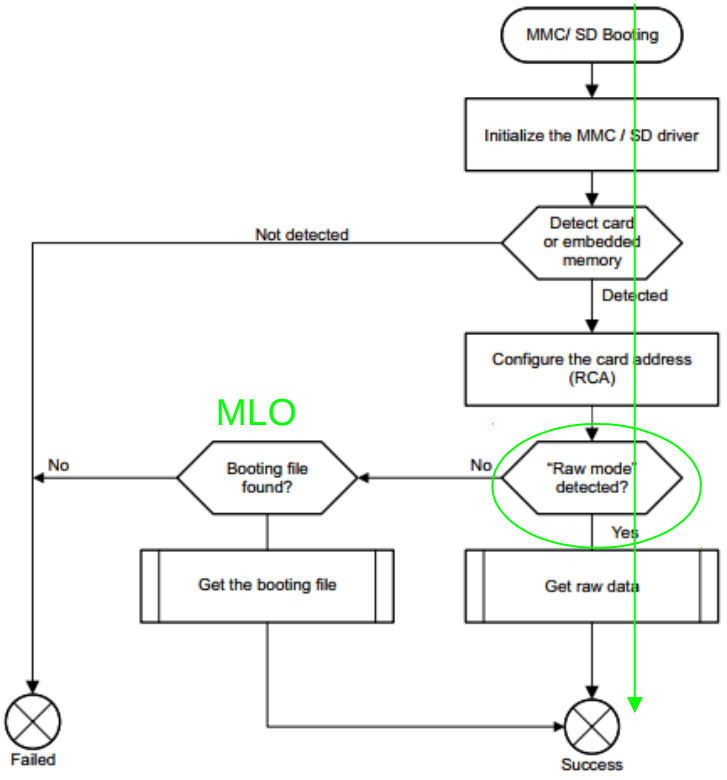
http://www.ti.com/lit/ug/spruh73p/spruh73p.pdf

Page: 5053

26.1.8.5.3 Booting Procedure

The high level flowchart of the eMMC / eSD and MMC/SD booting procedure is depicted in Figure 26-22.

Figure 26-22. MMC/SD Booting



TI AM335x: bootrom: raw mode:

http://www.ti.com/lit/ug/spruh73p/spruh73p.pdf (Page: 5054)

1. 0x0 <- (FAT Boot Sector, let's leave it blank...)
2. 0x20000 (128KB) <- We are going to use this location
3. 0x40000 (256KB) <- (2nd “backup” location)
4. 0x60000 (384KB) <- (3rd “backup” location)

Only 128KB in size… (hint, only 128KB of SRAM)

Das U-Boot (the Universal Boot Loader) U-Boot

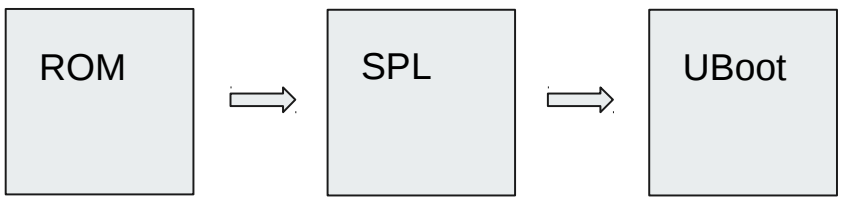
Original Author: Wolfgang Denk, now maintained by Tom Rini

- https://www.denx.de/wiki/U-Boot
- http://git.denx.de/?p=u-boot.git;a=summary
- https://en.wikipedia.org/wiki/Das_U-Boot

U-Boot: AM335x

Outputs two files for TI am335x targets:

- MLO = SPL (or Secondary Program Loader)
- u-boot.img (or u-boot-dtb.img) (U-Boot)



U-Boot: SPL

1. Initializes main memory (DDRx for am335x)
2. Loads full (U-Boot) into DDR memory

Or:

3. Initializes main memory (DDRx for am335x)
4. Loads Linux Kernel into DDR memory (aka: Falcon mode, faster boot mode/etc)

U-Boot:

- Network
- USB
- MMC
- File System (fat/extX)
- Shell

Sometimes you don’t need a full OS, have U-Boot init and then have U-Boot load/run your application.

U-Boot:

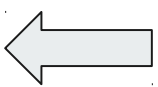
CPU : AM335X-GP rev 2.1
I2C: ready
DRAM: 512 MiB
Some drivers were not found
Reset Source: Power-on reset has occurred.
MMC: OMAP SD/MMC: 0, OMAP SD/MMC: 1
Using default environment

Board: BeagleBone Black
<ethaddr> not set. Validating first E-fuse MAC
BeagleBone Black:
Model: SeeedStudio BeagleBone Green:

U-Boot: microSD

Insert USB-microSD adapter, and type “lsblk”

```
voodoo@hestia:~/Supercon-2017-PocketBeagle$ lsblk
NAME        MAJ:MIN RM   SIZE RO TYPE MOUNTPOINT
sda           8:0    0 465.8G  0 disk
└─sda1        8:1    0 465.8G  0 part /
sde           8:64   1   7.4G  0 disk
└─sde1        8:65   1   7.4G  0 part
```



That’s our 8GB USB Flash Drive

Open: system.sh change: MMC=/dev/sde

U-Boot: Format microSD

```
sudo dd if=/dev/zero of=${MMC} bs=1M count=10

sudo sfdisk ${MMC} <<-__EOF__
4M,,L,*
__EOF__

sudo mkfs.ext4 -L rootfs ${MMC}1

voodoo@hestia:~/Supercon-2017-PocketBeagle$ ./scripts/format_drive.sh
```

U-Boot: (refresh for your memory)

http://www.ti.com/lit/ug/spruh73p/spruh73p.pdf (Page: 5054)

1. 0x0
2. 0x20000 (128KB) <- We are going to use this location
3. 0x40000 (256KB)
4. 0x60000 (384KB)

```
sudo dd if=./deploy/MLO of=${MMC} count=1 seek=1 bs=128k
sudo dd if=./deploy/u-boot.img of=${MMC} count=2 seek=1 bs=384k
```

Base Rootfs: Debian 9.x (Stretch)

Maintainer: ~~me~~ (with lots of help from all the Debian Developers and 1000’s of other users)

- https://elinux.org/Beagleboard:BeagleBoneBlack_Debian#2017-11-05_-_Debian_9_.28Stretch.29_-_Weekly
- https://www.debian.org/
- https://github.com/beagleboard/image-builder

Device Tree

- See kernel documentation for bindings
  - devicetree/bindings/eeprom/eeprom.txt
- Local copies enable you to extend on the fly
  - /opt/source/dtb-4.9-ti
  - /opt/source/bb.org-overlays
- Overlays loaded in u-boot, but also possible via kernel configfs

config-pin

- config-pin -i p1.36
- config-pin -q p1.36
- config-pin p1.36 pruout

show-pins.pl

- perl /opt/scripts/device/bone/show-pins.pl -v

mikroBus Click usage

- See bbb.io/pbmb
- Supported with device-tree overlays loaded in u-boot

Enabling PRU

- 2 possible drivers: remoteproc or uio
- Enabled via device tree at boot
  - Different systems might have different defaults

Demonstrations

- BaconBits demo
-

Some current projects

- Bela Mini
- PocketPilot

Contributions and issues

- Cape/add-on support
  - https://github.com/beagleboard/bb.org-overlays
- Image deltas
  - https://github.com/beagleboard/image-builder
- In-system examples
  - https://github.com/beagleboard/bone101

Questions?

e-ale


beagleboard.org®
e-ale
Thank you!
SanDisk
e-ale
e-ale