

# PocketBeagle Walk Through

# 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/>

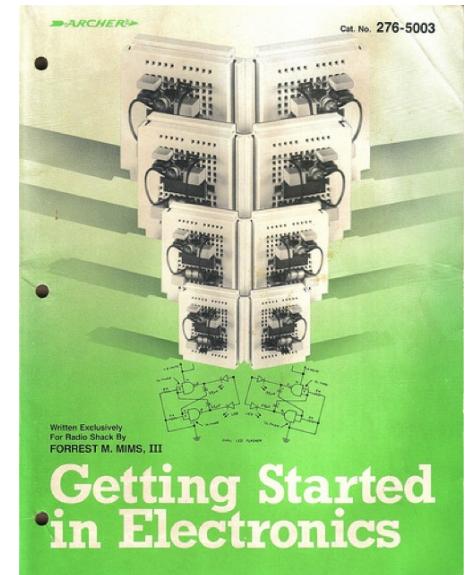
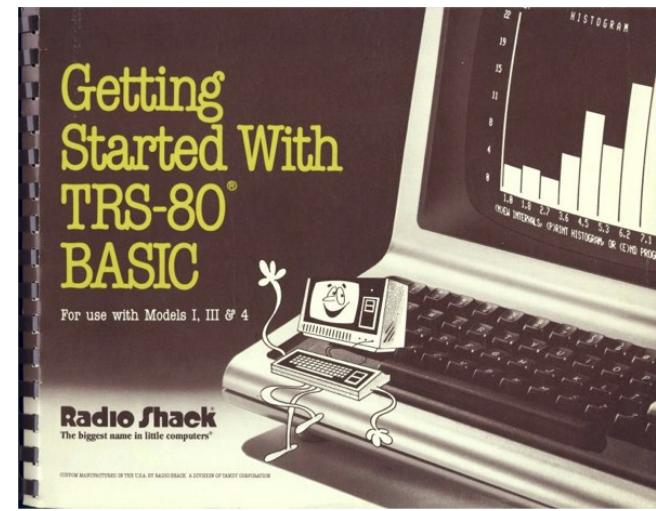
# Outline

- 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

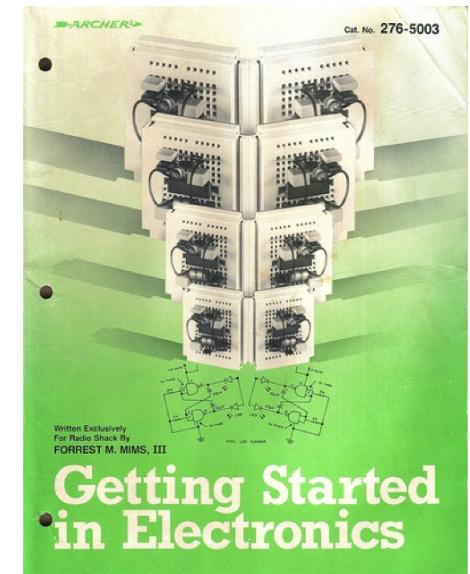
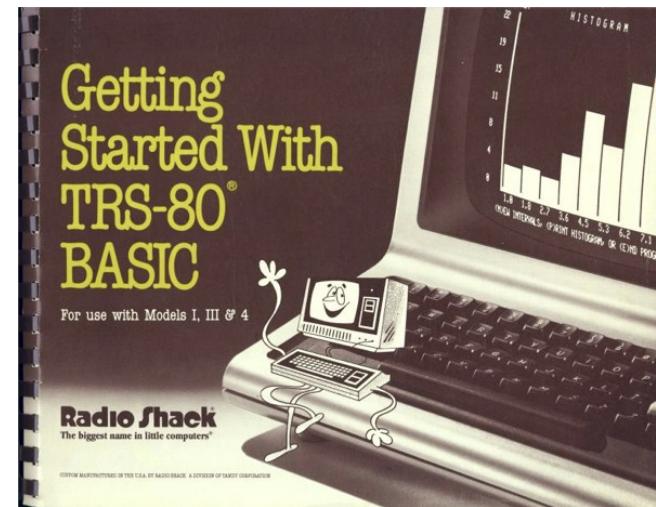
- Education
  - Design and use of open source SW/HW
  - Embedded computing
- Collaboration
  - Physical computing
  - Robotics
  - Industrial/machine controls

# Inspiration from early PCs



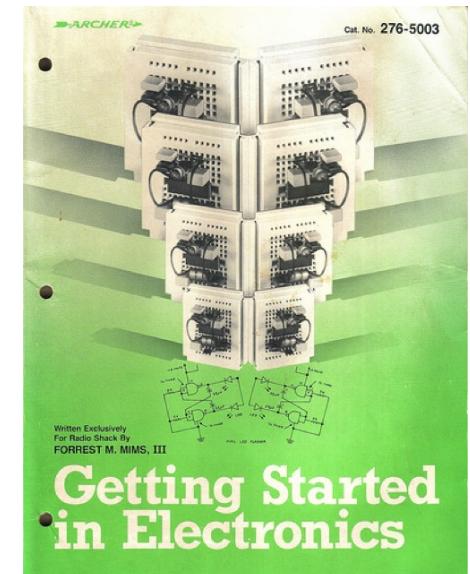
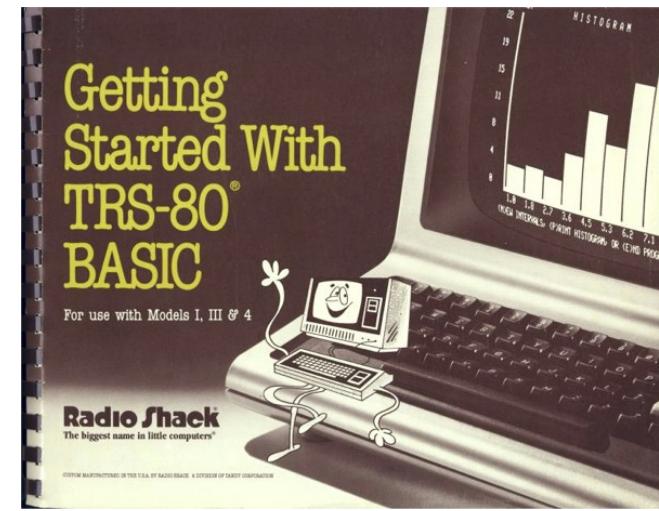
# Inspiration from early PCs

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



# 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



# Vision

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



# Vision

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



# 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)



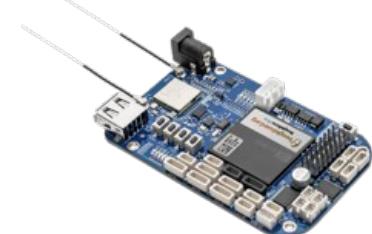
\$249

Mint tin sized with industrial  
peripherals (BeagleBone)



\$69

Application focused BeagleBones



\$79

Smalls mint tin sized with super-  
flexible design - PocketBeagle



\$25

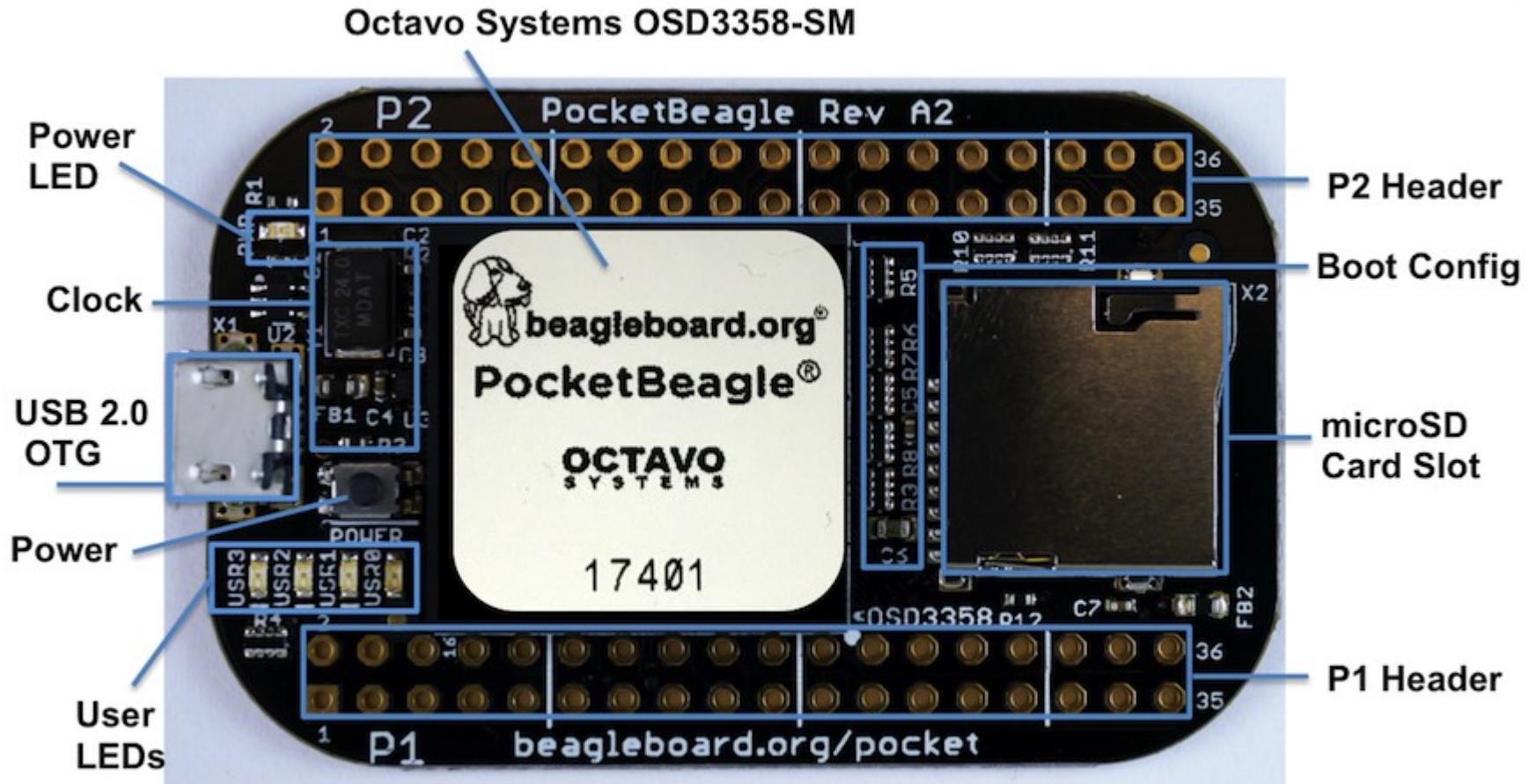
# PocketBeagle objectives

- 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

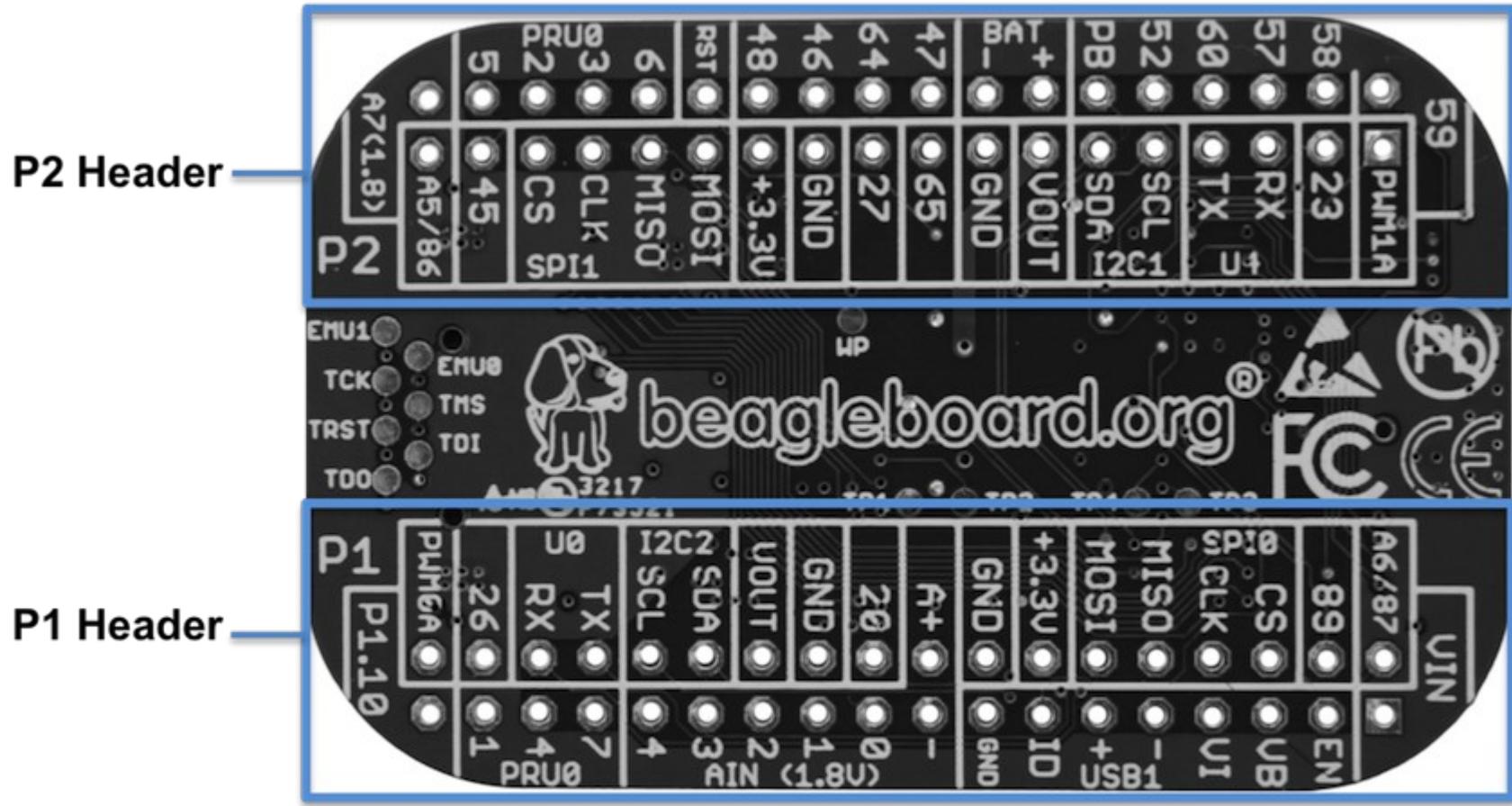
# PocketBeagle key features

- 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 (OpenGL ES)
- 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

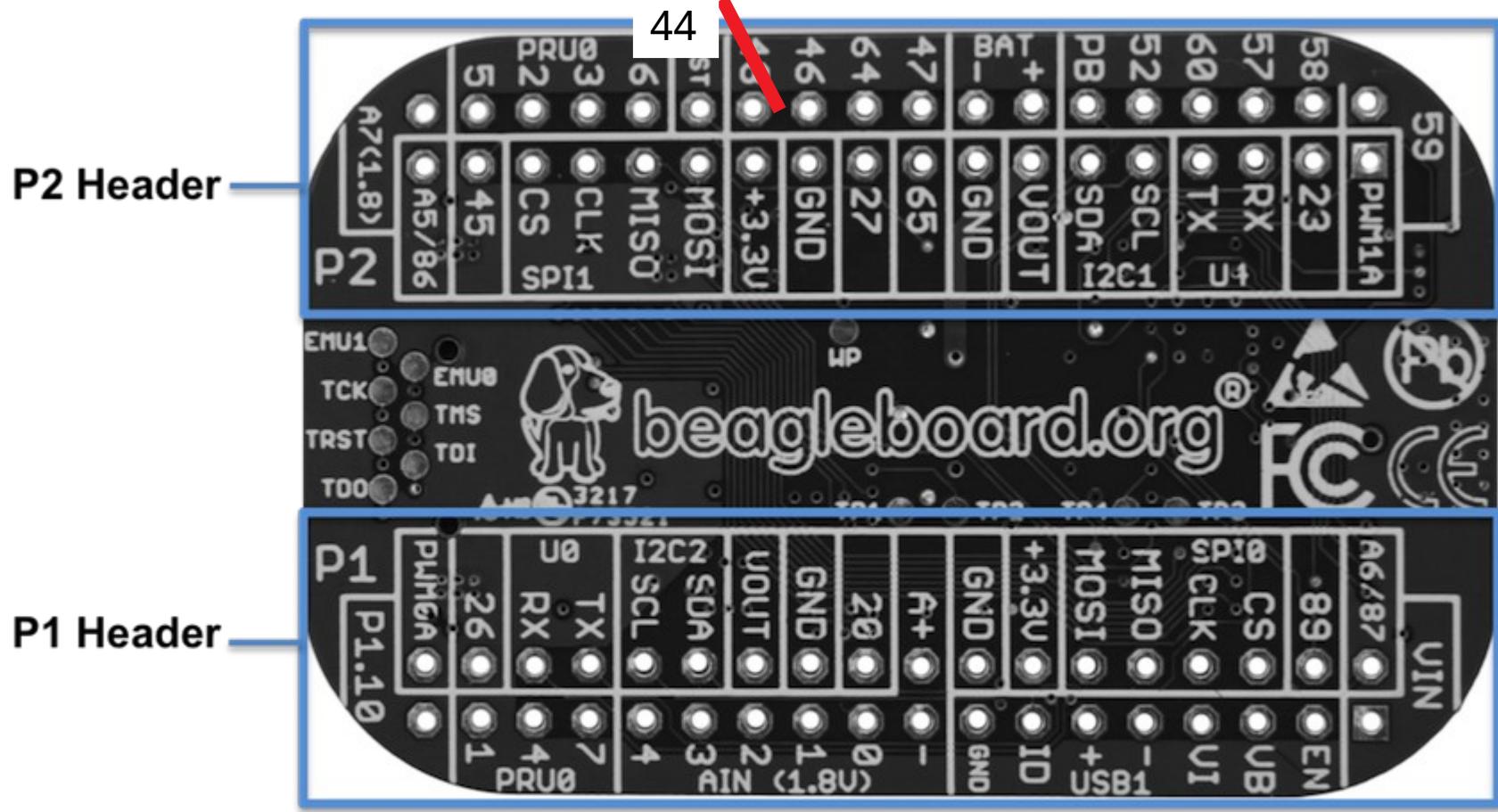
# PocketBeagle top



# PocketBeagle bottom



# PocketBeagle bottom



# PocketBeagle expansion

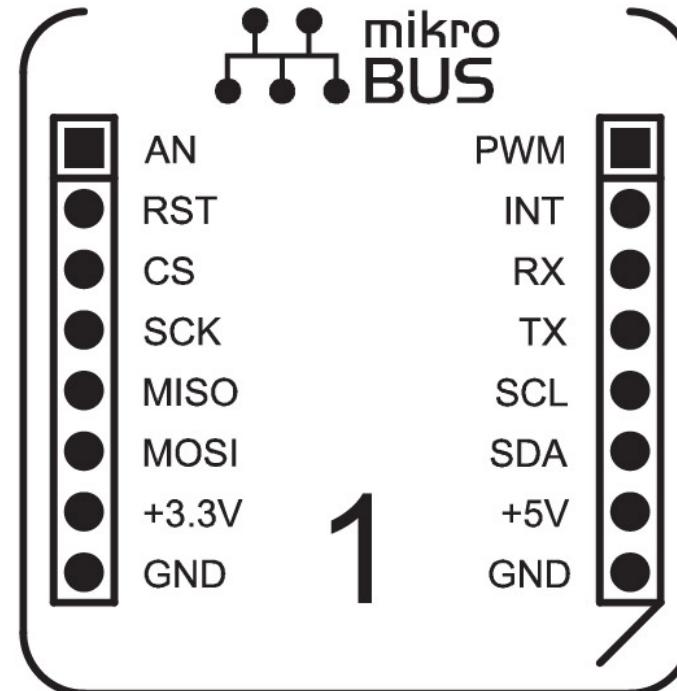
PocketBeagle Expansion Headers (Rev A2a)

		P1			
USB1	SYS	VIN	1	2	87
	USB1	V_EN	109	3	4
		GPIO		89	
		VBUS	5	6	5
		VIN	7	8	2
		DN	9	10	3
		DP	11	12	4
		ID	13	14	3.3V
		GND	15	16	GND
		REF-	17	18	REF+ AIN 1.8V
AIN 1.8V	0	19	20	20	GPIO
	1	21	22	GND	
	2	23	24	VOUT	SYS
	3	25	26	12	
	4	27	28	13	
PRU0	7	QEP0	STRB	A	GPIO
	4	29	30	43	
	114	31	32	42	
	1	33	34	26	
	PRU1	10			
		88	35	36	110
			A	PWM0	0
			PWM0	B	PRU0

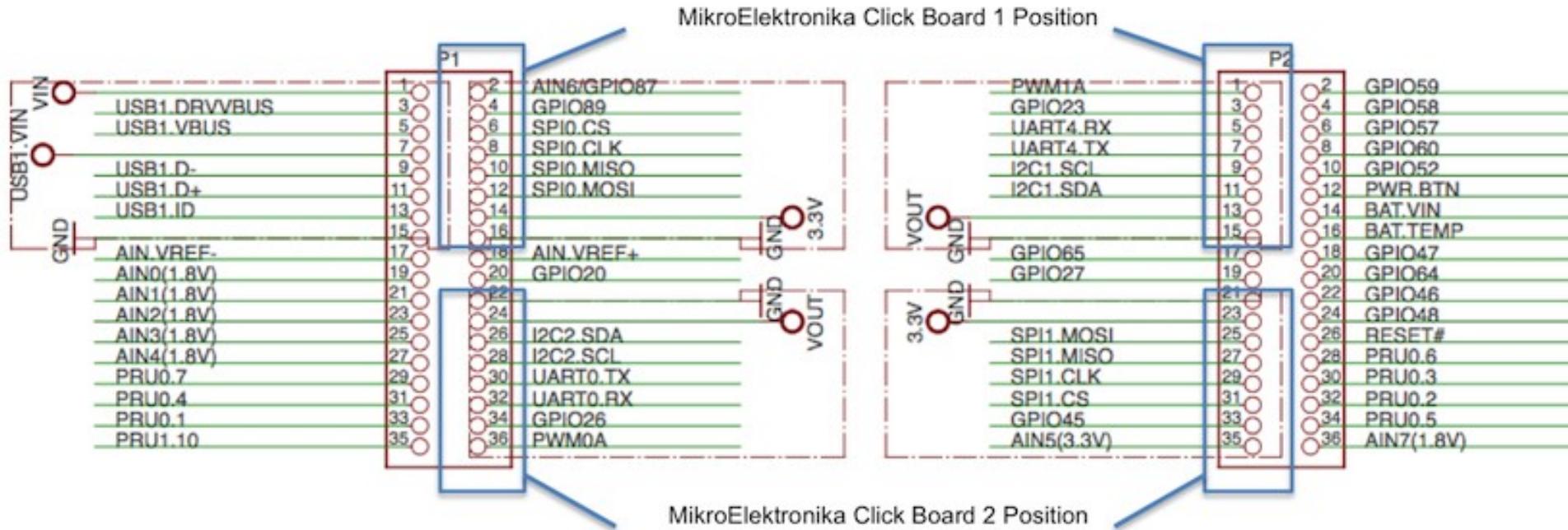
		P2			
GPIO	PWM1	A	50	1	2
	PWM2	B	23	3	4
	UART4	RX TX	30	5	6
	CAN1	RX TX	31	7	8
	I2C1	SCL SDA	15	9	10
			14	11	12
			13	14	VIN
			15	16	TEMP BAT
			65	17	18
			27	19	20
SYS	GPIO		47	STRB	QEP2
			64	15i	PRU0
			46	IDX	QEP2
			44	A	14(in)
			41	25	26
			NRST	SYS	PRU0
			116	IDX	QEP0
			113	6	
			112	3	
			115	2	
PRU1	CAN1	RX TX	7	29	30
	I2C2	SDA SCL	PRU1	16(in)	
	TX	MOSI MISO	PRU1	15(out)	QEP2 B
	UART0	TX RX	PRU1	8	AIN 3.3V
	PRU1	CS CLK	PRU1	5	
			86	35	36
			7		AIN 1.8V

# mikroBus Click

Analog - **AN**  
Reset - **RST**  
SPI Chip Select - **CS**  
SPI Clock - **SCK**  
SPI Master Input Slave Output - **MISO**  
SPI Master Output Slave Input - **MOSI**  
VCC-3.3V power - **+3.3V**  
Reference Ground - **GND**



# Connecting mikroBus Clicks



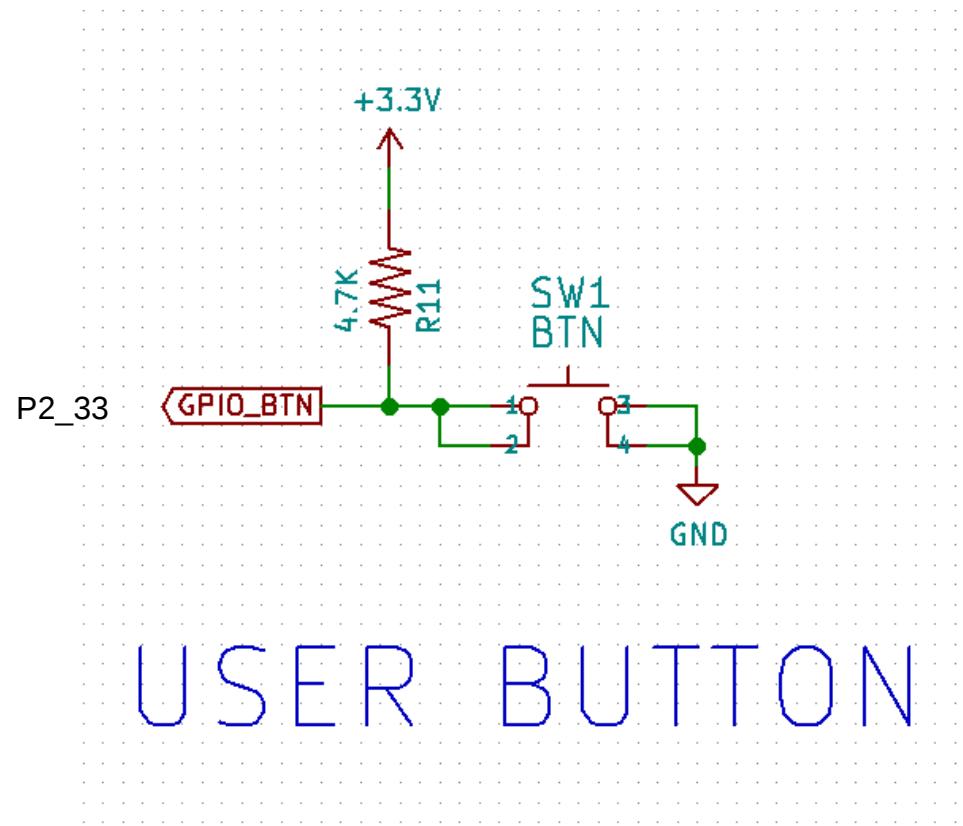
# BaconBits objectives

- 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

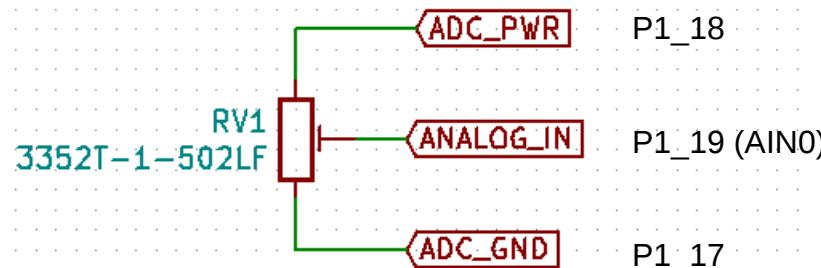
# BaconBits features

- 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

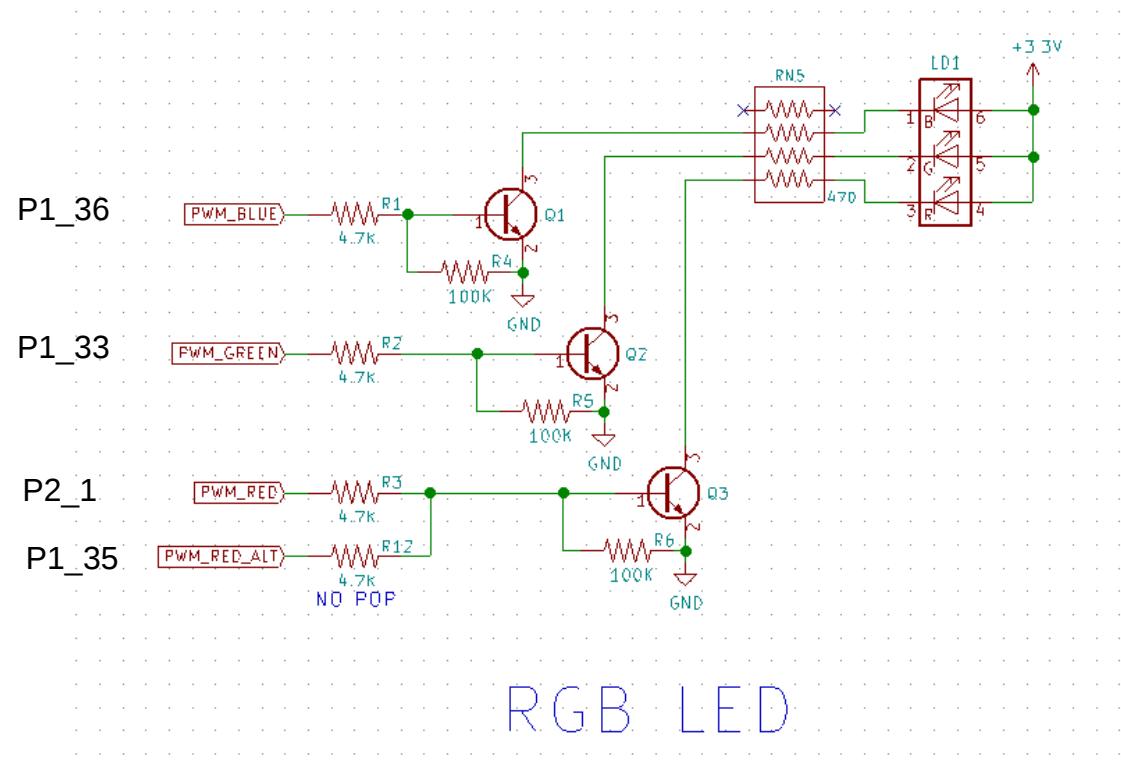


# BaconBits ADC thumbwheel

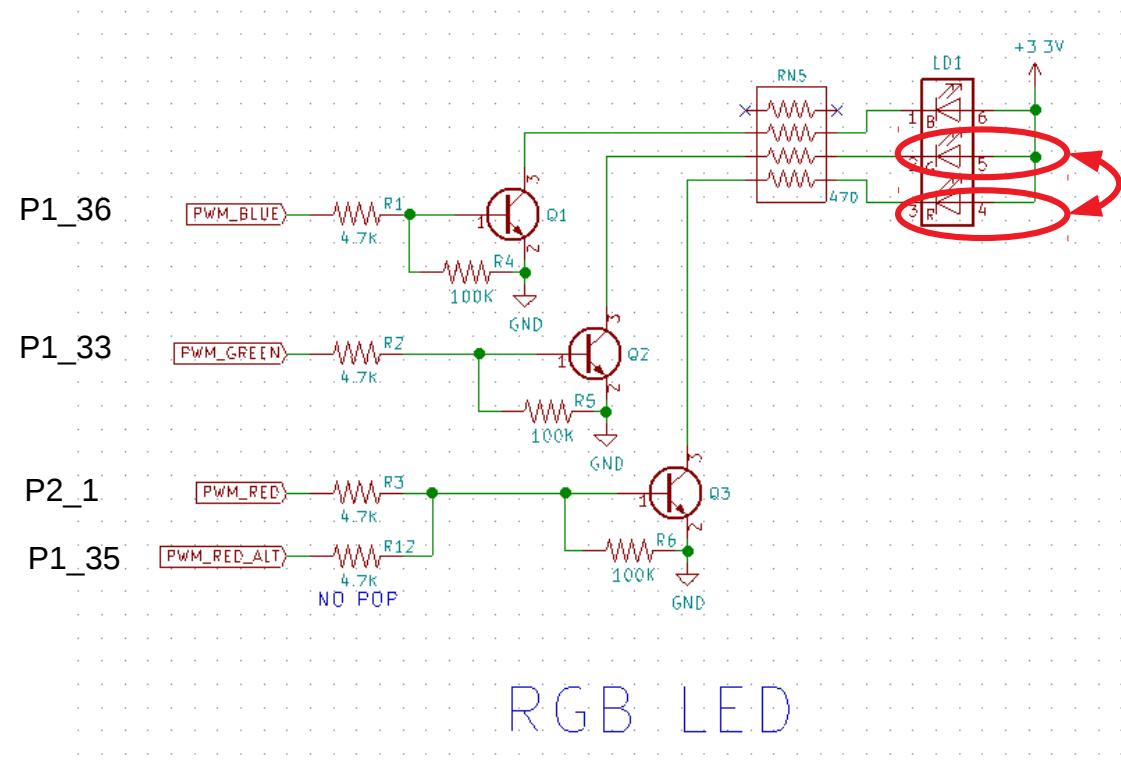


THUMBWHEEL

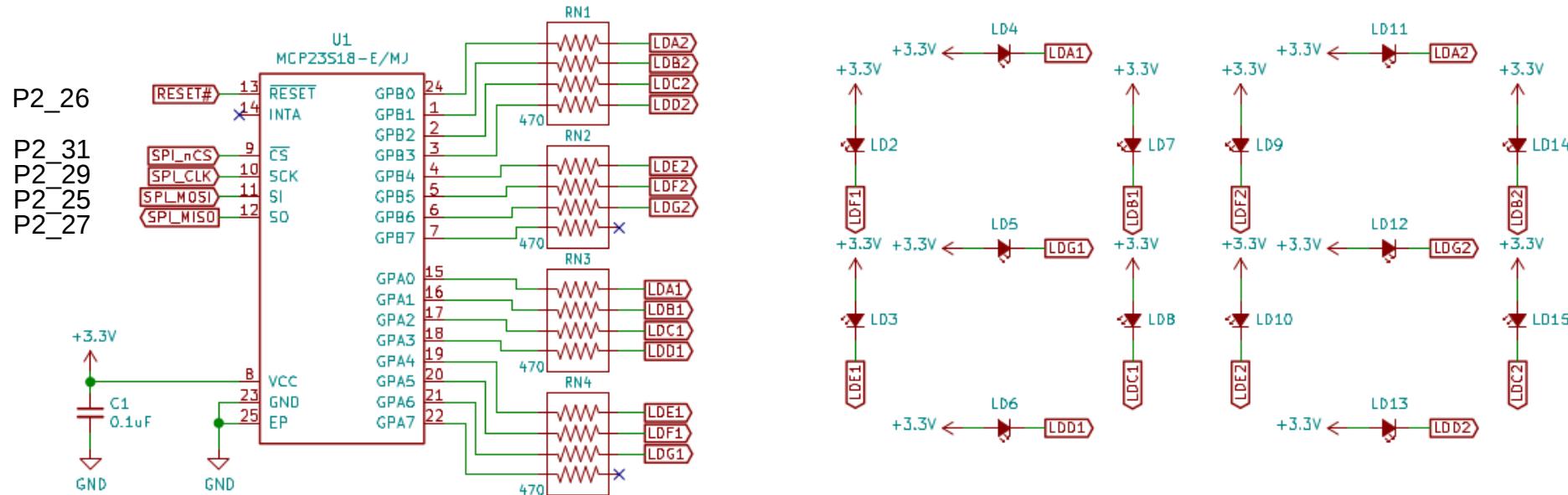
# BaconBits RGB LED



# BaconBits RGB LED

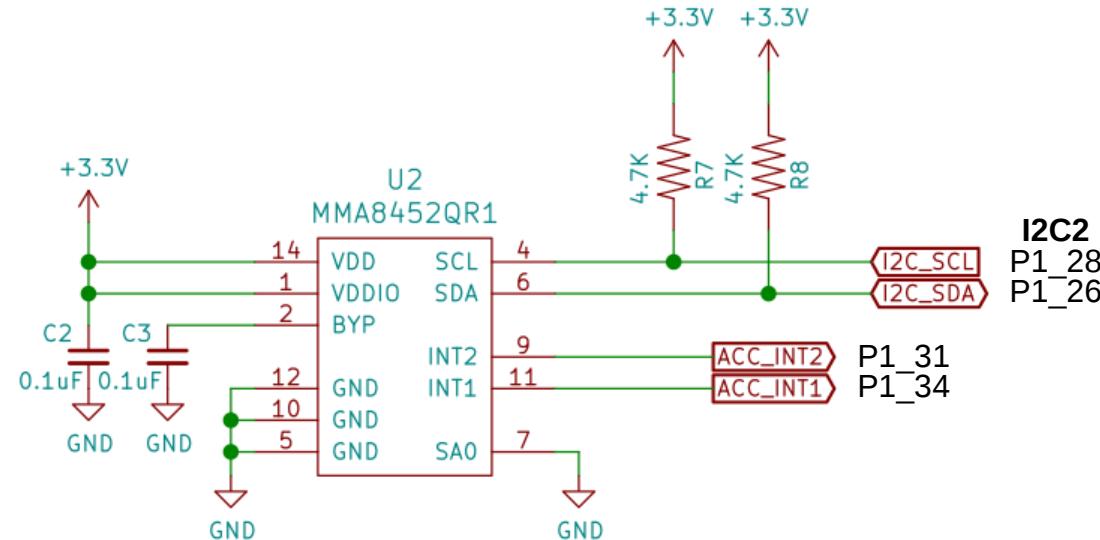


# BaconBits SPI 7-segment display



EMULATED 7 SEGMENTS

# BaconBits I2C accelerometer



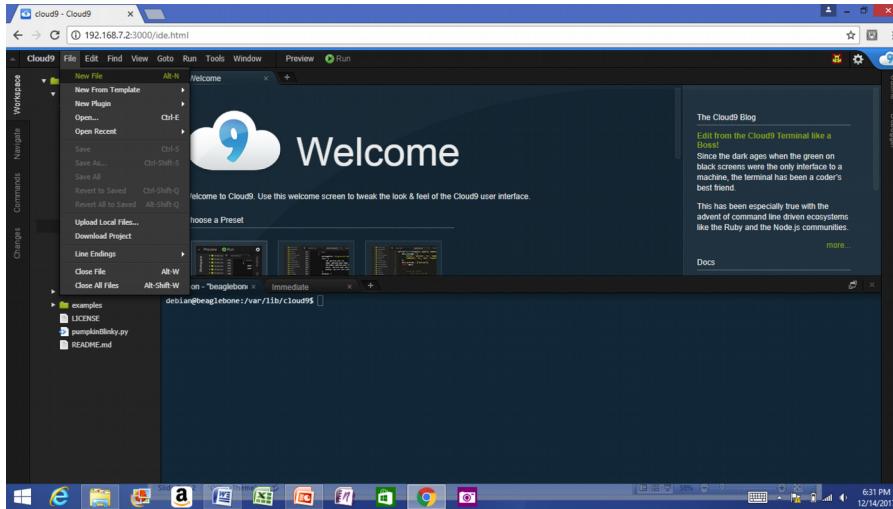
ACCELEROMETER

# Developer experience

- Customized Debian images – [bbb.io/latest](https://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

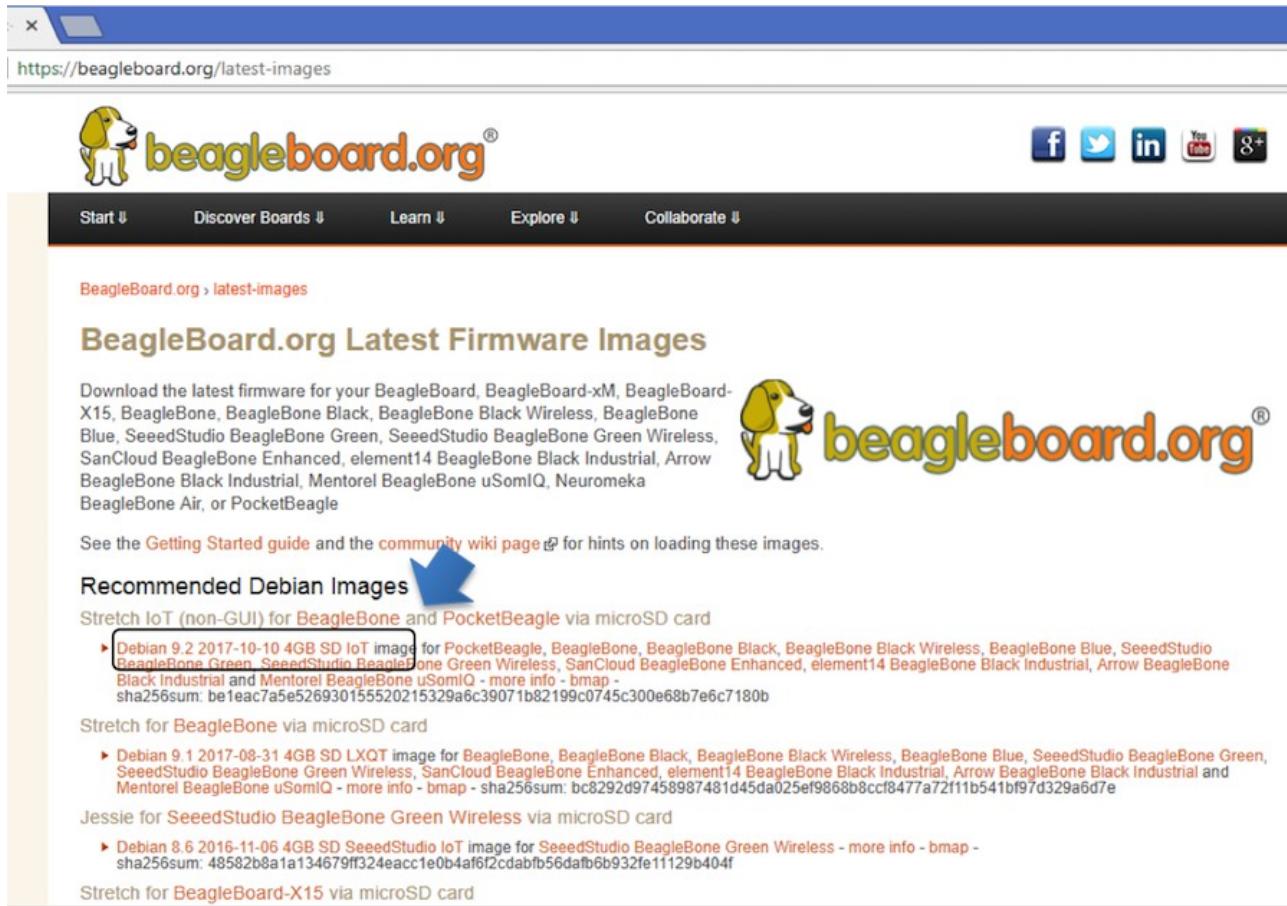
- Power, network, develop
- You can add a network and power many other ways



# Some work in progress

- Add proxy for various services (in Buster IoT images today)
- Integrate common web-based WiFi provisioning
  - SeeedStudio BeagleBone 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



The screenshot shows a web browser window with the URL <https://beagleboard.org/latest-images>. The page content is as follows:

**BeagleBoard.org Latest Firmware Images**

Download the latest firmware for your BeagleBoard, BeagleBoard-xM, BeagleBoard-X15, BeagleBone, BeagleBone Black, BeagleBone Black Wireless, BeagleBone Blue, SeeedStudio BeagleBone Green, SeeedStudio BeagleBone Green Wireless, SanCloud BeagleBone Enhanced, element14 BeagleBone Black Industrial, Arrow BeagleBone Black Industrial, Mentorlel BeagleBone uSomIQ, Neuromeka BeagleBone Air, or PocketBeagle

See the [Getting Started guide](#) and the [community wiki page](#) for hints on loading these images.

**Recommended Debian Images**

Stretch IoT (non-GUI) for BeagleBone and PocketBeagle via microSD card

- [Debian 9.2 2017-10-10 4GB SD IoT image for PocketBeagle, BeagleBone, BeagleBone Black, BeagleBone Black Wireless, BeagleBone Blue, SeeedStudio BeagleBone Green, SeeedStudio BeagleBone Green Wireless, SanCloud BeagleBone Enhanced, element14 BeagleBone Black Industrial, Arrow BeagleBone Black Industrial and Mentorlel BeagleBone uSomIQ - more info - bmap - sha256sum: bfeac7a5e526930155520215329a6c39071b82199c0745c300e68b7e6c7180b](#)

Stretch for BeagleBone via microSD card

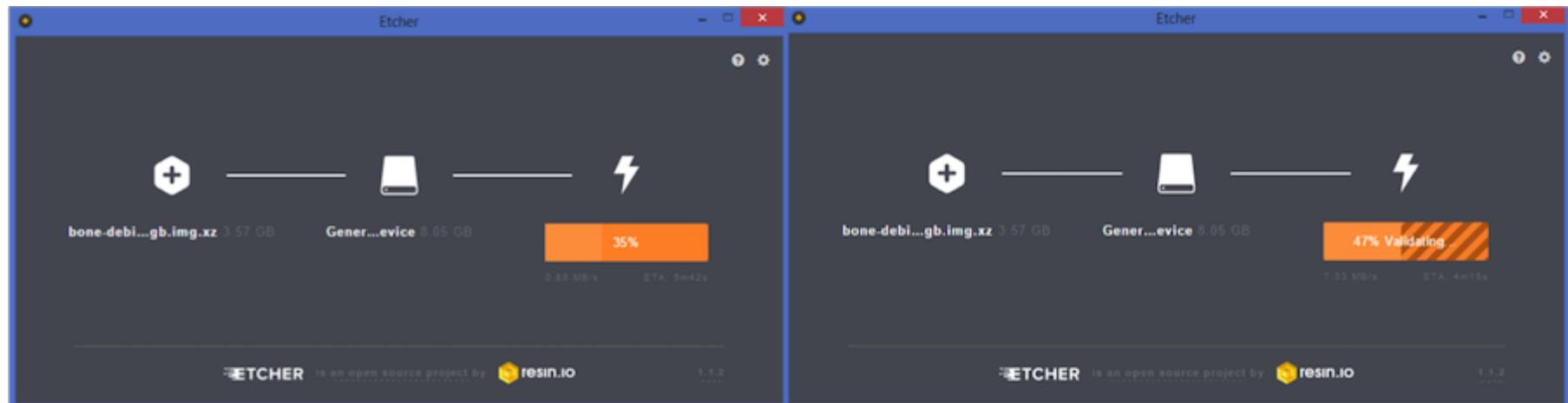
- [Debian 9.1 2017-08-31 4GB SD LXQT image for BeagleBone, BeagleBone Black, BeagleBone Black Wireless, BeagleBone Blue, SeeedStudio BeagleBone Green, SeeedStudio BeagleBone Green Wireless, SanCloud BeagleBone Enhanced, element14 BeagleBone Black Industrial, Arrow BeagleBone Black Industrial and Mentorlel BeagleBone uSomIQ - more info - bmap - sha256sum: bc8292d97458987481d45da025e19868b8ccf8477a72f11b541bf97d329a6d7e](#)

Jessie for SeeedStudio BeagleBone Green Wireless via microSD card

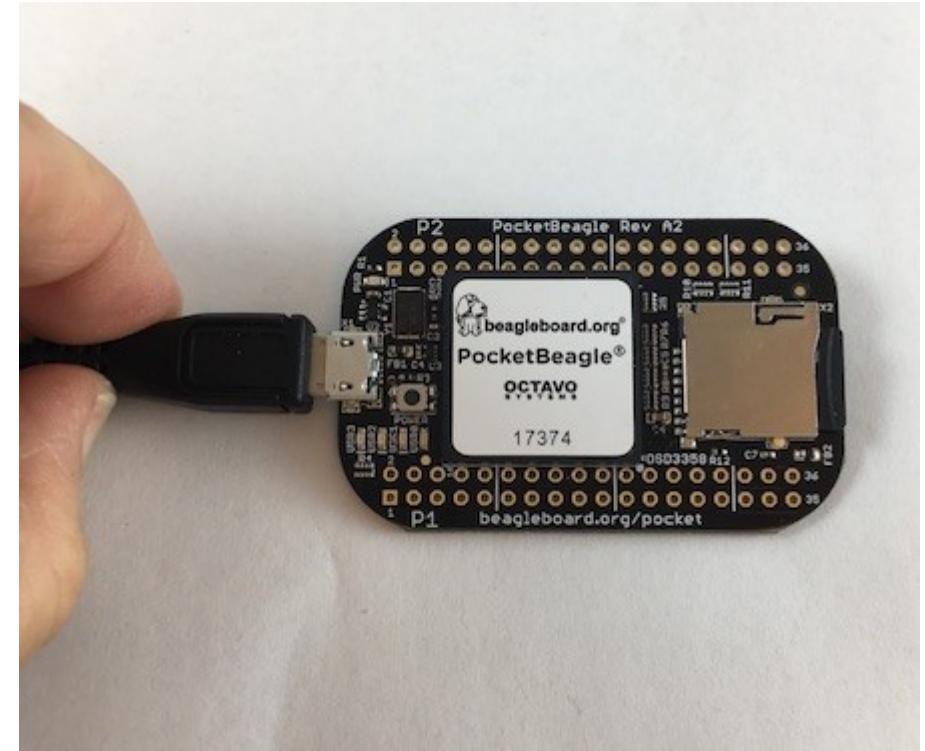
- [Debian 8.6 2016-11-06 4GB SD SeeedStudio IoT image for SeeedStudio BeagleBone Green Wireless - more info - bmap - sha256sum: 48582b8a1a134679f324eacc1e0b4af6f2cdabfb56dafb6b932fe11129b404f](#)

Stretch for BeagleBoard-X15 via microSD card

# Write image to microSD with Etcher



# Insert microSD and boot



# Connect to the USB network

Getting started with Beagle X

file:///Volumes/BEAGLEBONE/START.htm



**Start your Beagle**

Beagles are tiny computers with the capability of modern systems, without the bulk, expense, or noise. Read the step-by-step getting started tutorial below to begin developing with your Beagle in minutes.

For user supplied tips on getting started, visit the eLinux (or other) community wiki pages:

- BeagleBoard
- BeagleBoard-X15
- BeagleBoard-X16
- BeagleBone
- BeagleBone Black
- BeagleBone Green
- BeagleBone Green Wireless
- BeagleBone Blue
- SeedStudio BeagleBone Green
- SeedStudio BeagleBone Green Wireless
- SancCloud BeagleBone Enhanced
- Neuromeka BeagleBone Air

If any step fails, it is recommended to update to the [latest software image](#) to use the instructions below.

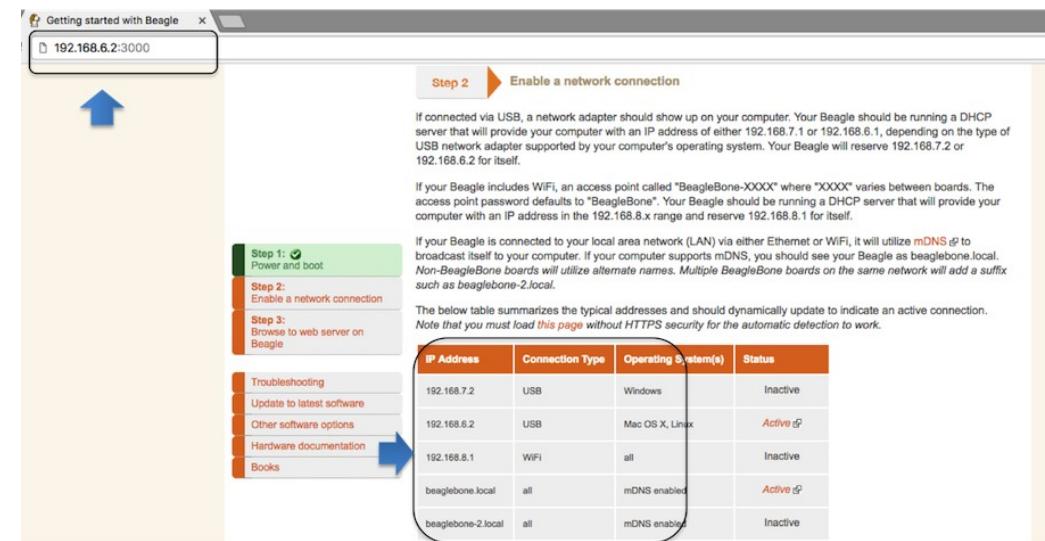
**Step 1** Power and boot

Most Beagles include a USB cable, providing a convenient way to provide both power to your Beagle and connectivity to your computer. If you provide your own, ensure it is of good quality. You'll connect the ["type-B" plug](#) of the USB cable to your Beagle and the ["type-A"](#) plug to your computer. Note that BeagleBoard-X15 must always be powered instead by a 12V adapter with a barrel jack.

Alternatively, for Beagles other than BeagleBoard-X15 and BeagleBone Blue that require 12V, you can utilize a 5V adapter connected to the barrel jack.

Getting started with Beagle X

192.168.6.2:3000



**Step 2** Enable a network connection

If connected via USB, a network adapter should show up on your computer. Your Beagle should be running a DHCP server that will provide your computer with an IP address of either 192.168.7.1 or 192.168.6.1, depending on the type of USB network adapter supported by your computer's operating system. Your Beagle will reserve 192.168.7.2 or 192.168.6.2 for itself.

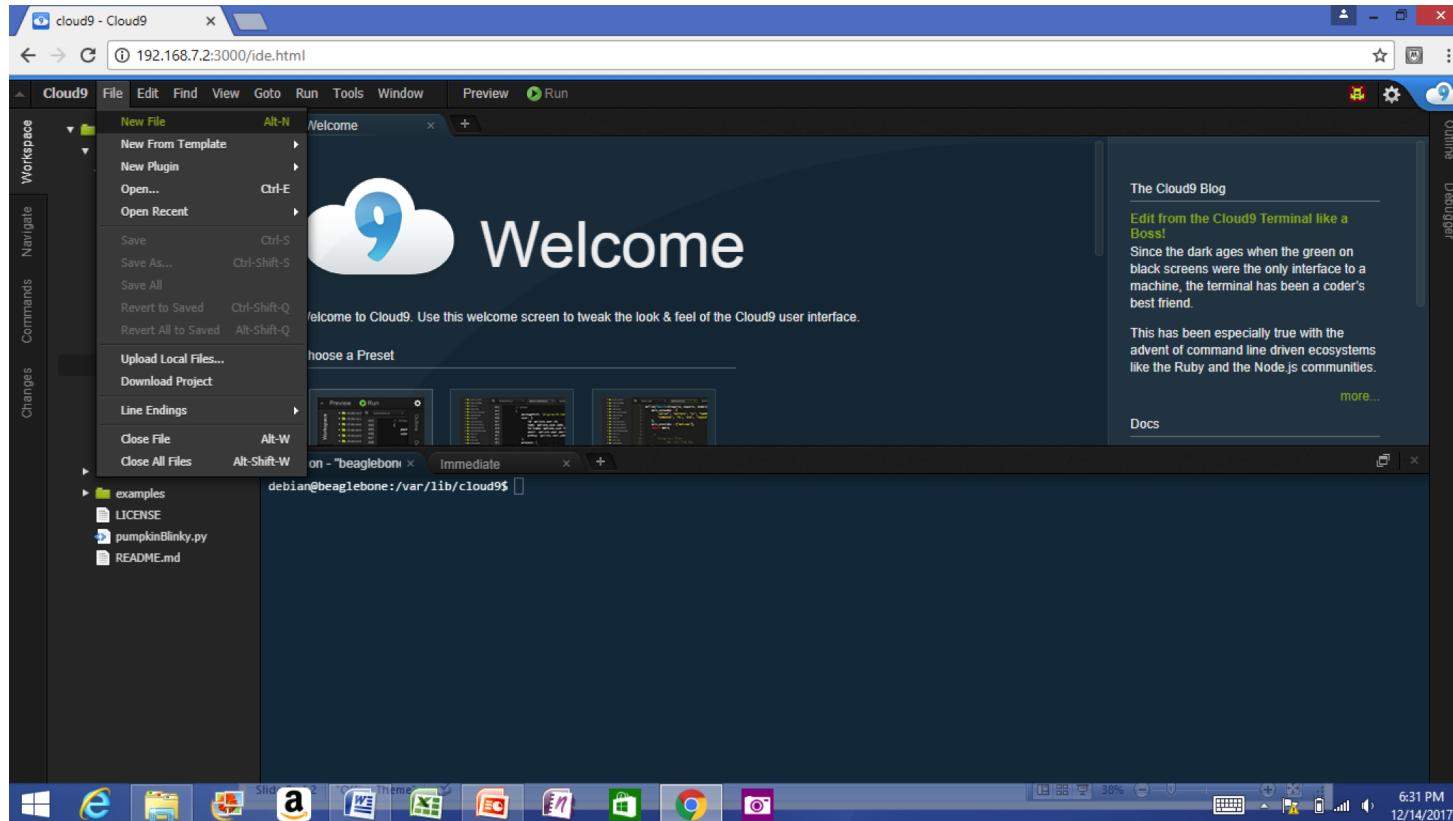
If your Beagle includes WiFi, an access point called "BeagleBone-XXXX" where "XXXX" varies between boards. The access point password defaults to "BeagleBone". Your Beagle should be running a DHCP server that will provide your computer with an IP address in the 192.168.8.x range and reserve 192.168.8.1 for itself.

If your Beagle is connected to your local area network (LAN) via either Ethernet or WiFi, it will utilize mDNS to broadcast itself to your computer. If your computer supports mDNS, you should see your Beagle as beaglebone.local. Non-BeagleBone boards will utilize alternate names. Multiple BeagleBone boards on the same network will add a suffix such as beaglebone-2.local.

The below table summarizes the typical addresses and should dynamically update to indicate an active connection. Note that you must load [this page](#) without HTTPS security for the automatic detection to work.

IP Address	Connection Type	Operating System(s)	Status
192.168.7.2	USB	Windows	Inactive
192.168.8.2	USB	Mac OS X, Linux	Active
192.168.8.1	WiFi	all	Inactive
beaglebone.local	all	mDNS enabled	Active
beaglebone-2.local	all	mDNS enabled	Inactive

# Open the IDE



# 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

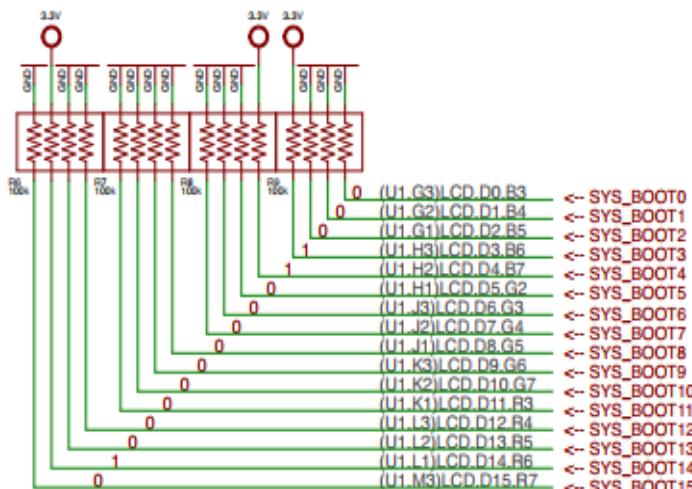
# USB gadgets

- 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 \(Rev. P\)](#))

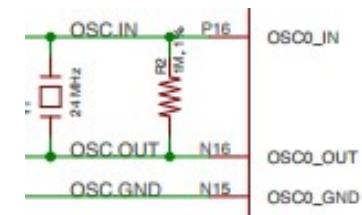
- <http://www.ti.com/lit/ug/spruh73p/spruh73p.pdf> (page 5032)



SYSBOOT[15:14] = 01 = 24Mhz

SYSBOOT[4:0] = 11000

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



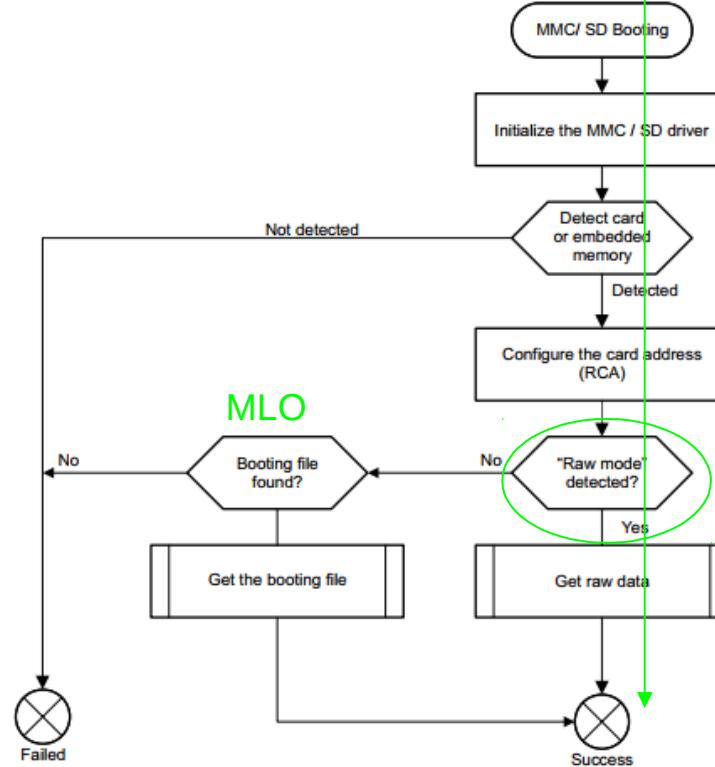
**26.1.8.5.3 Booting Procedure**

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

# TI AM335x: bootrom

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

Page: 5053



---

# **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](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://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](http://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>

A close-up photograph of a BeagleBoard development board. The board is populated with several components: a central BeagleBoard module, a blue microSD card labeled "SanDisk microSD microSDHC microSDXC", a red LED, a green LED, a yellow LED, and a black breadboard. The board is connected to a laptop via a USB cable. The background is dark, making the board components stand out.

# Questions?

# Thank you!

