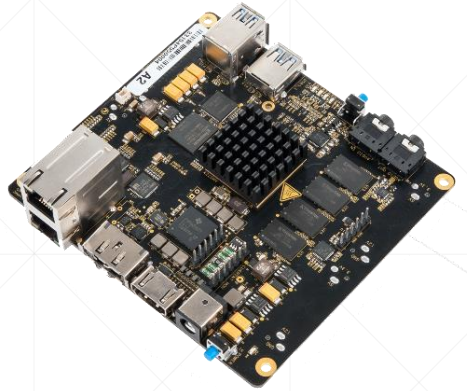
System-in-Package Technology:

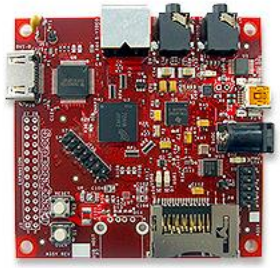
Making it Easier to Build Your Own Linux Computer

Jason Kridner
Erik Welsh
03/12/2018

The SBC Prototyping Revolution

- Proliferation of prototyping boards
  - Huge array of Processors
  - Every kind of connector
  - Add-on boards for additional functionality
- Developing communities
  - Support for new users
  - Collaboration for experienced developers
- Exposure to Linux
  - Development of drivers
  - Open source projects

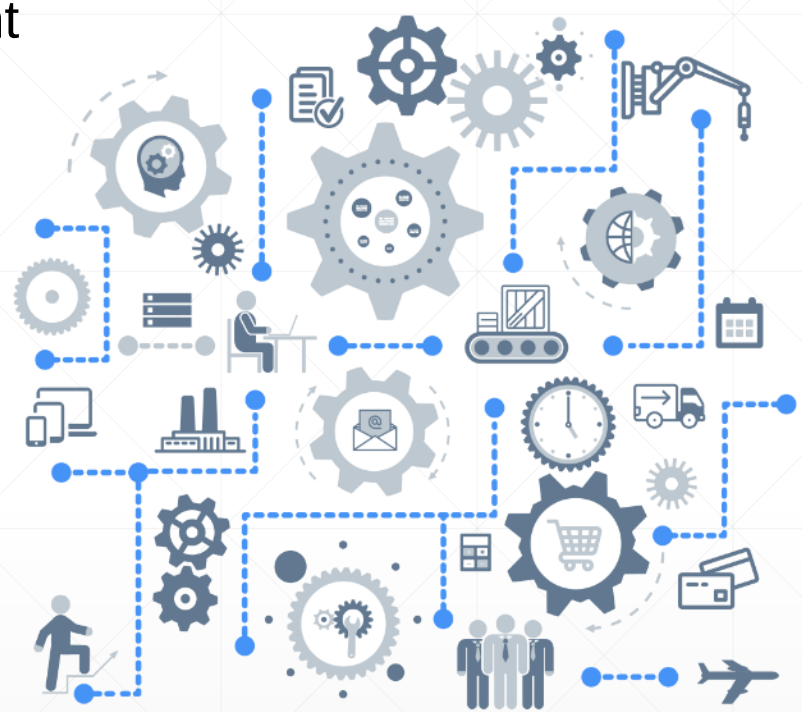







Software Drives Hardware Decisions

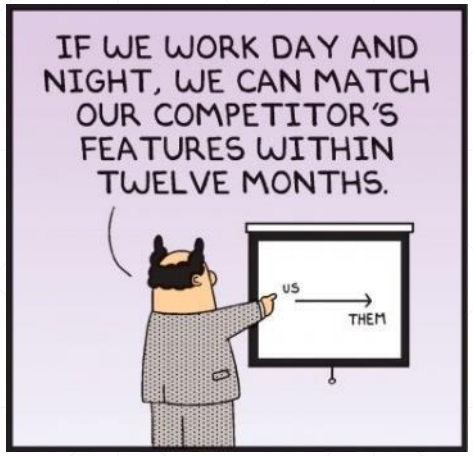
- Software developers must be involved in Hardware development
  - If there is no SW for a piece of HW, then don’t use it
  - Many platforms provide great starting point for SW development
    - Focus on value added feature differentiation
    - Re-writing drivers does not add value
  - Don’t allow changes in HW for the sake of changing HW
    - SW impact needs to be understood



- With great power, comes great responsibility …
  - Choosing a platform with Open Hardware
  - Using components that can be obtained from Distribution / in small quantities
  - Hardware should also focus on value added features
    - Routing DDR does not add value

“Mind the Gap” Moving from Prototype to Product

- Developing custom PCB
  - Smaller is better … but smaller is harder
  - Open hardware help development; known good solutions
- Migrating Software
  - Porting from development board to final components
  - Bring up & Provisioning
- Doing more with less
  - Smaller teams
  - Need tools that reduce time and effort



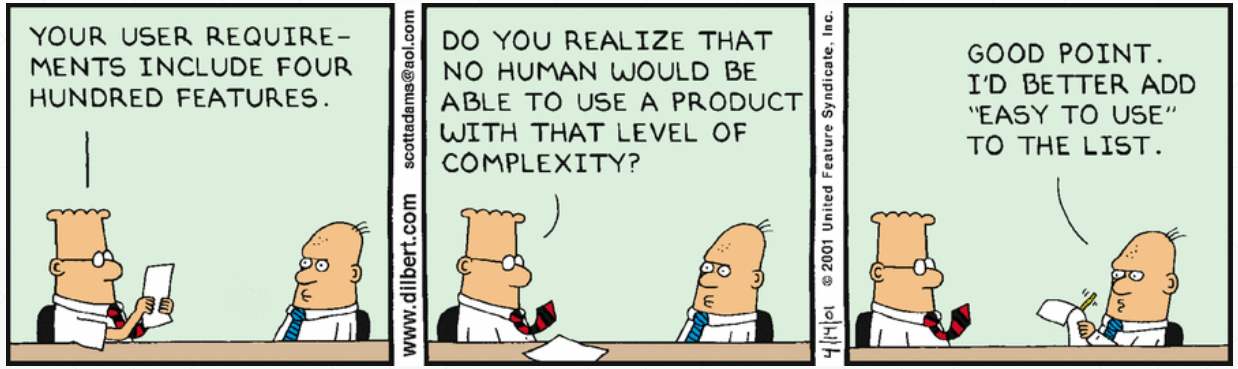
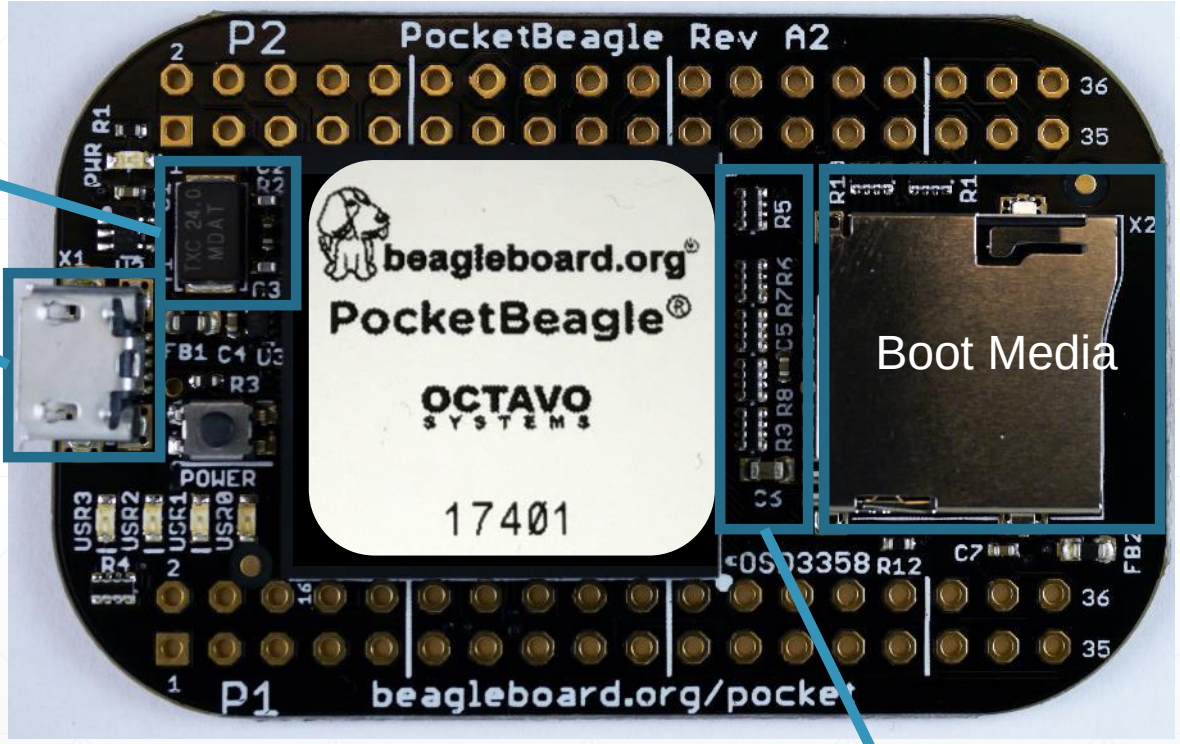


Photo credits: Dilbert by Scott Adams

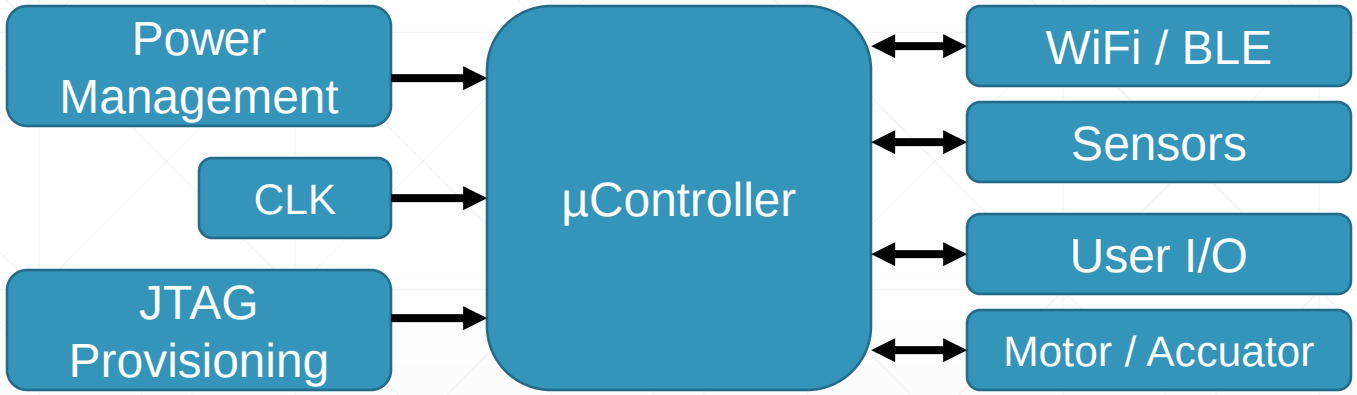
System-in-Package provides simple Linux HW Solutions

- Minimum Hardware Required to Run Linux
  - Connect power inputs
  - Connect clock inputs
  - Select boot mode
  - Provide Linux boot image
- Proven Linux solution
- Like working with a microcontroller but now with the power of Linux



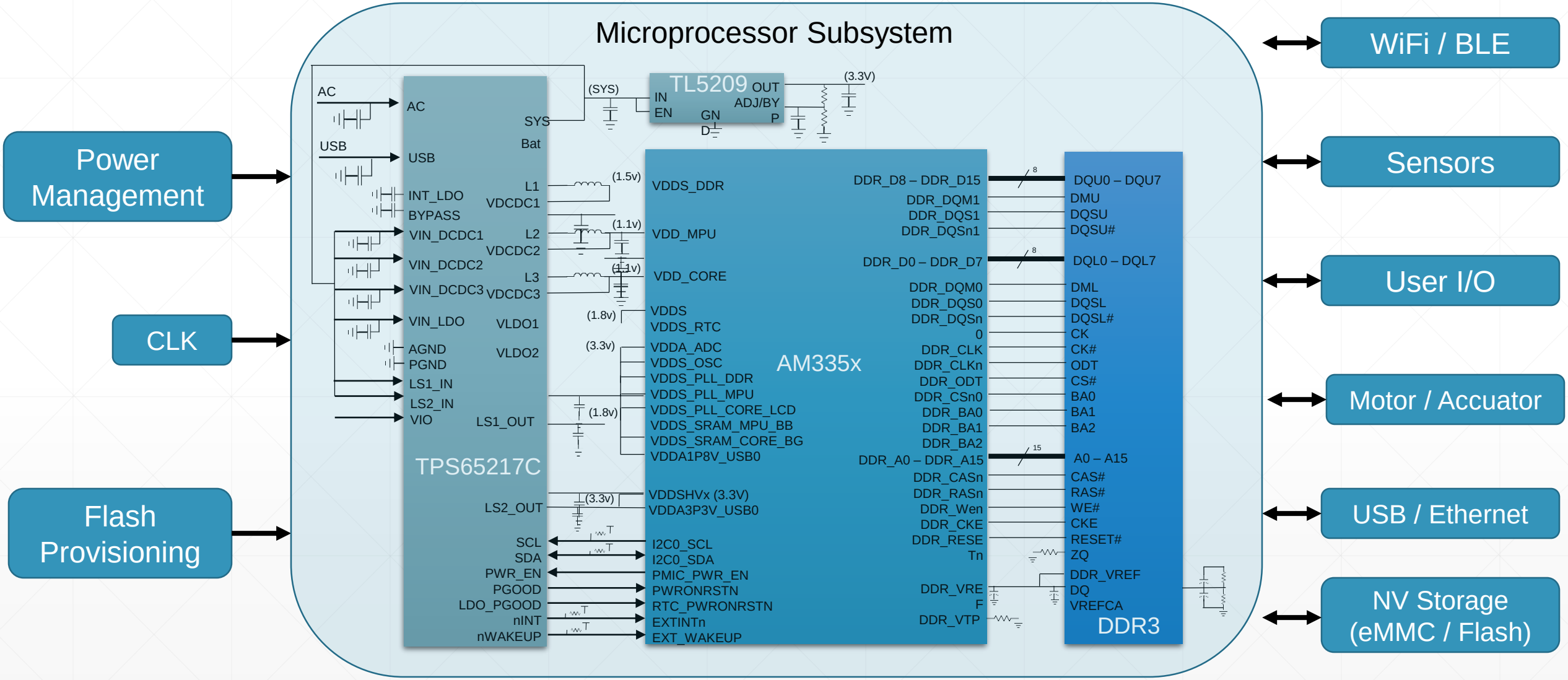
Moving up from a microcontroller can be scary …

Microcontroller System Block Diagram



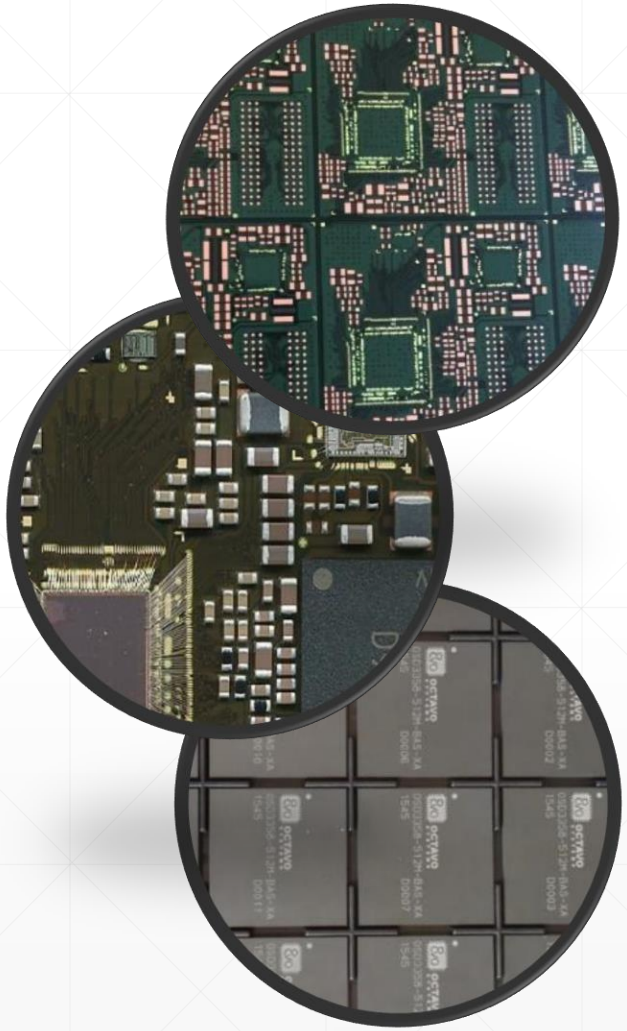
Becomes ….

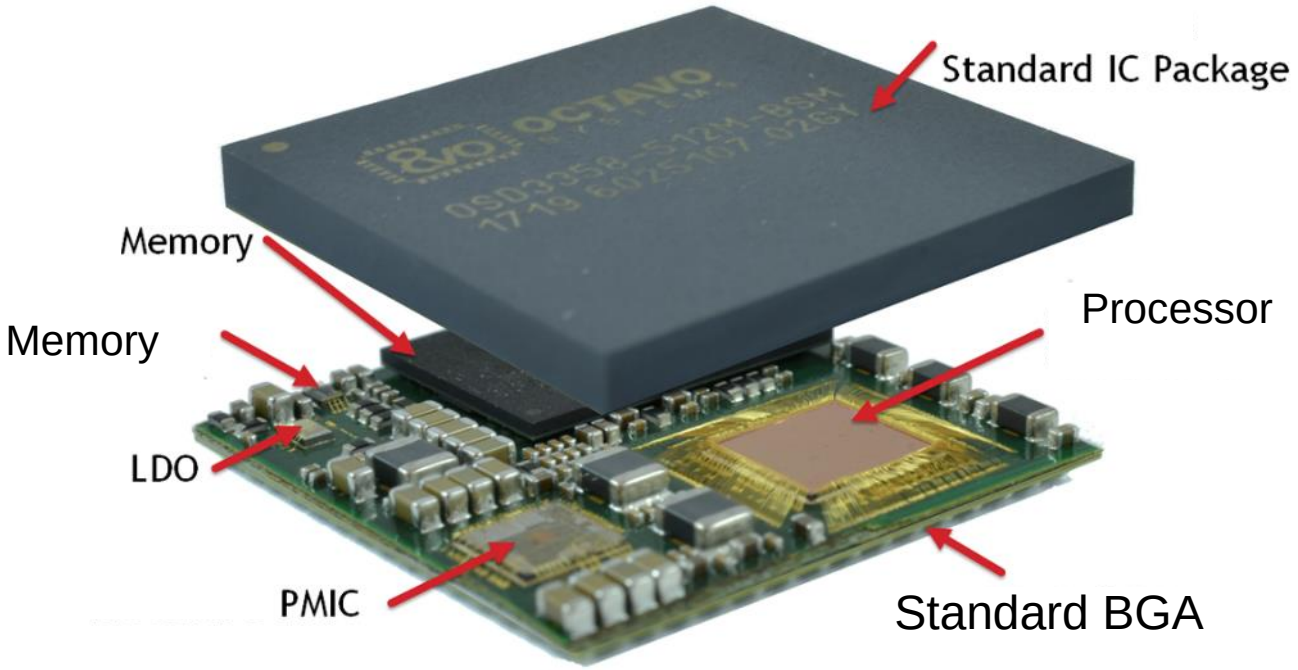
Typical Microprocessor System Block Diagram



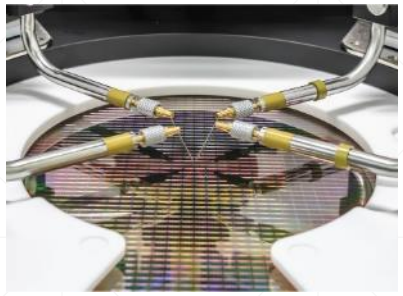
How can we simplify this complexity?

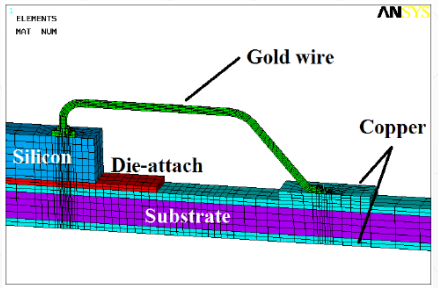
System-in-Package





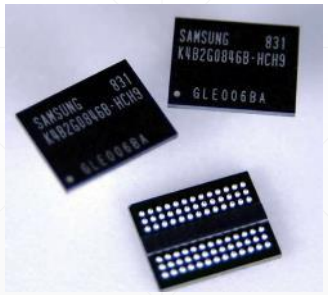
What is System-in-Package





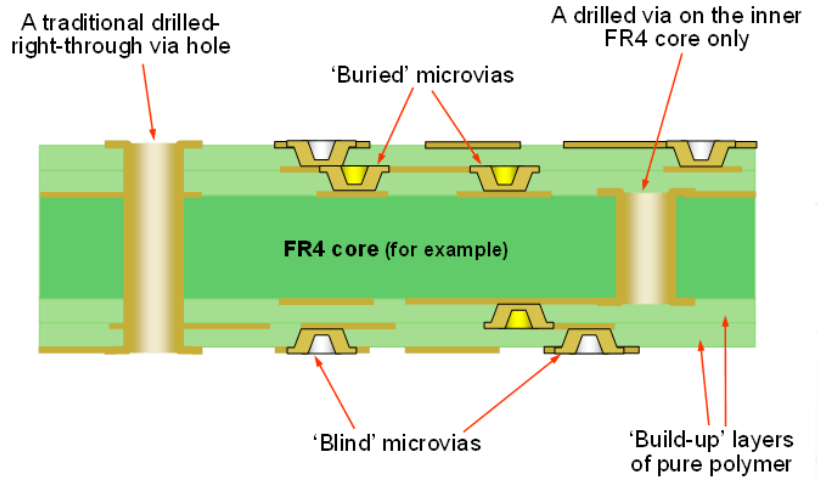
Attached Die

+



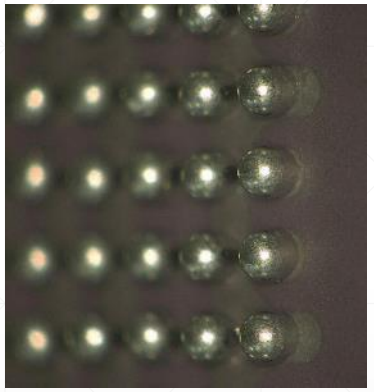
Discrete Components

+



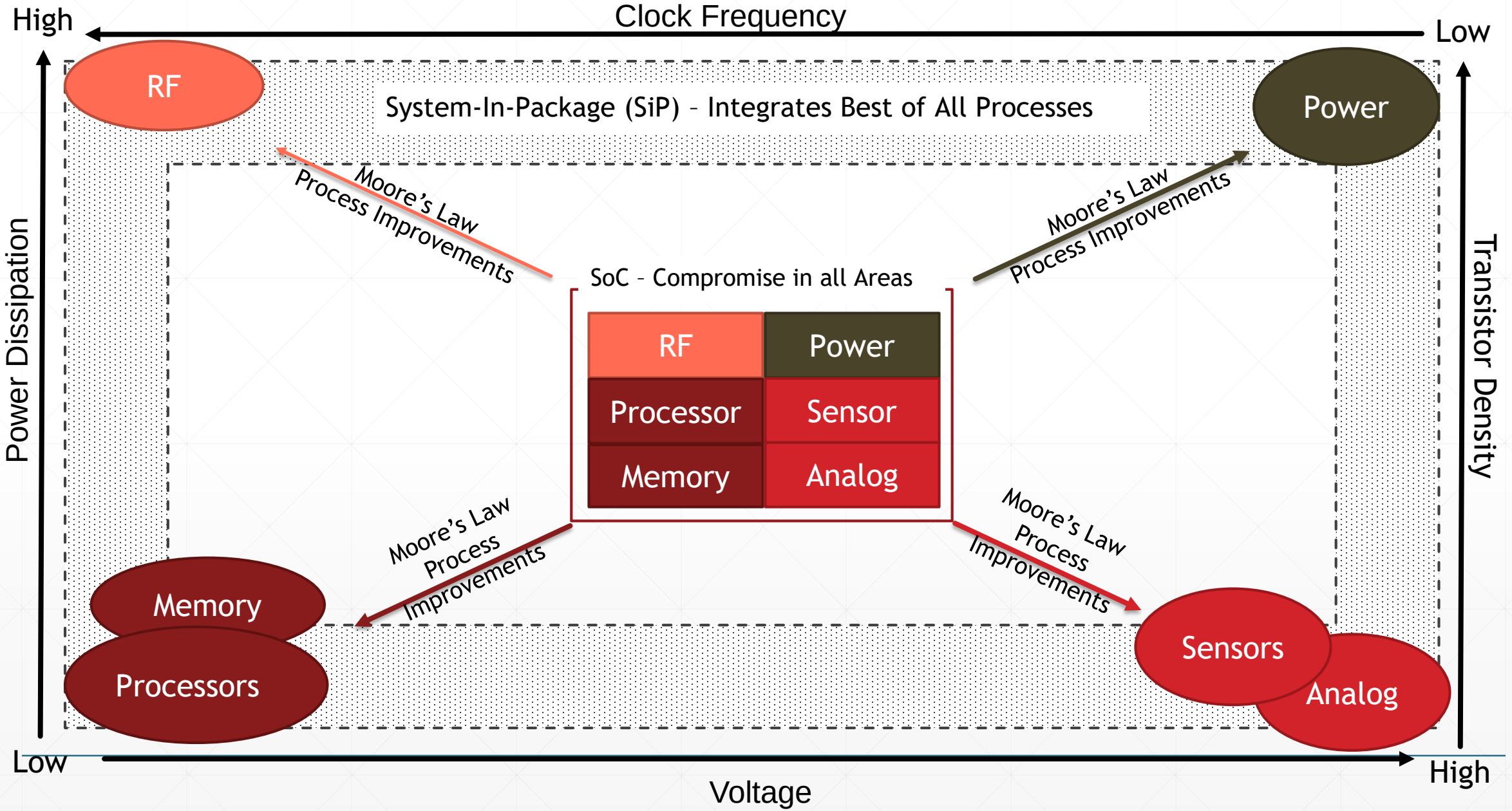
Substrate

+

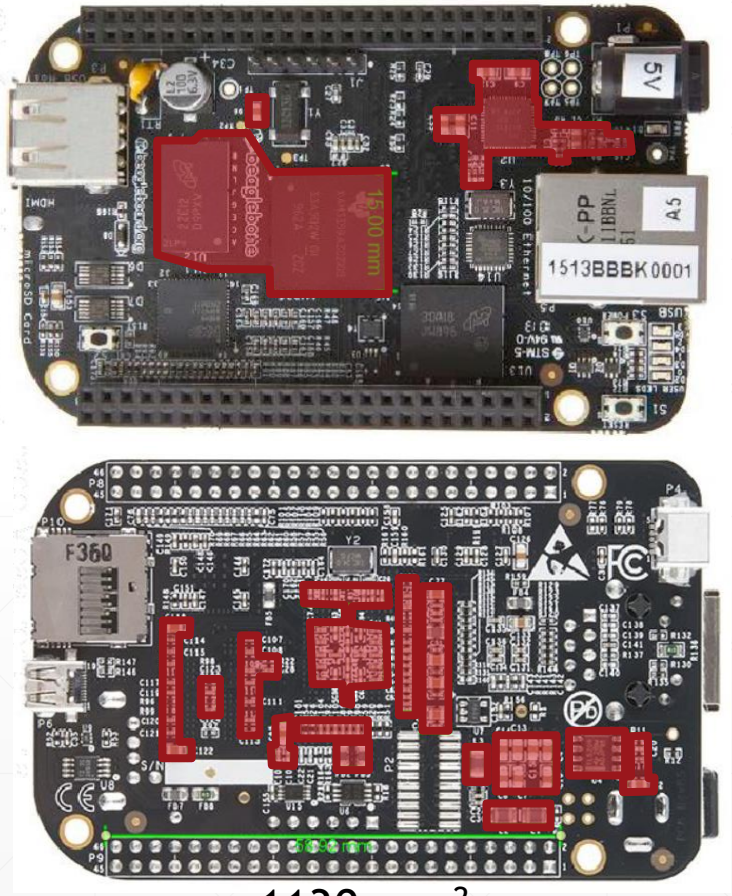


Pins

Why Can't We Just Use an SoC?

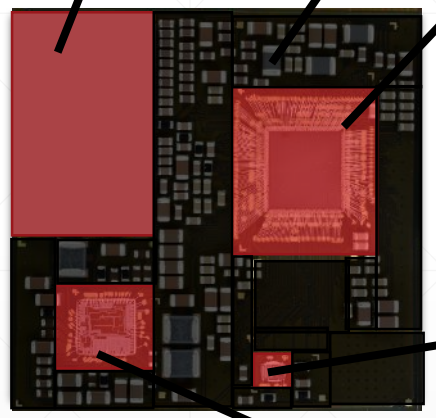


OSD3358 SiP Integration



1130 mm^2

BeagleBone Black Board



OSD3358 SiP

DDR3 SDRAM
800MHz

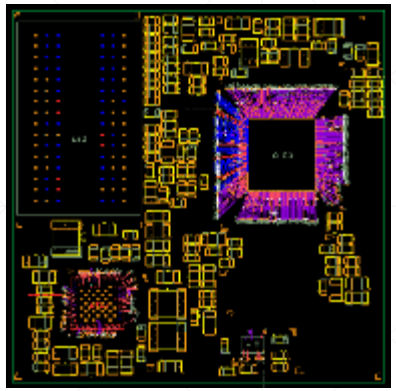
All Needed Resistors,
Capacitors, and
Inductors

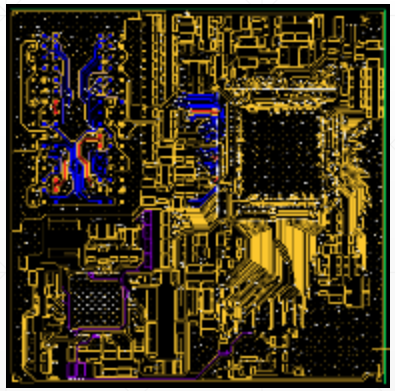
TI Sitara AM335x
Cortex-A8
1GHz

TI TL5209 LDO
3.3V Out

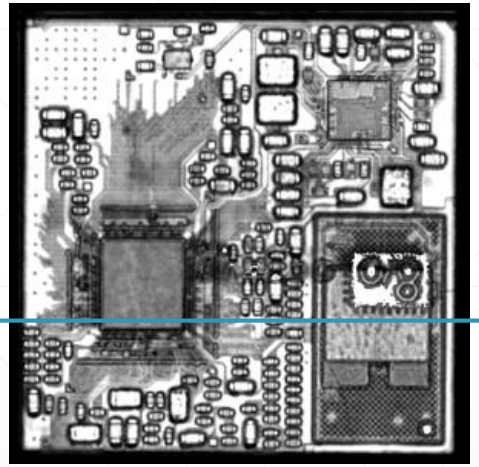
TI TPS65217C PMIC
Vin: Battery, 5VDC, USB
Vout: 1.8V, 3.3V, Sys_Vout

A Closer Look at a SiP

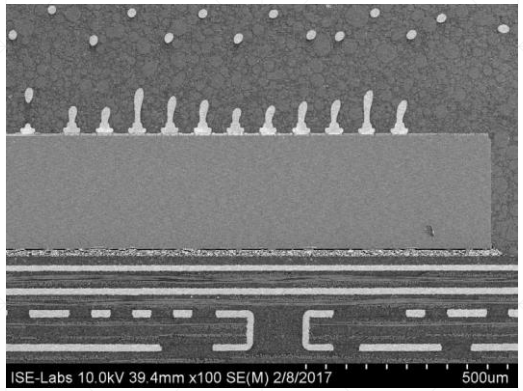


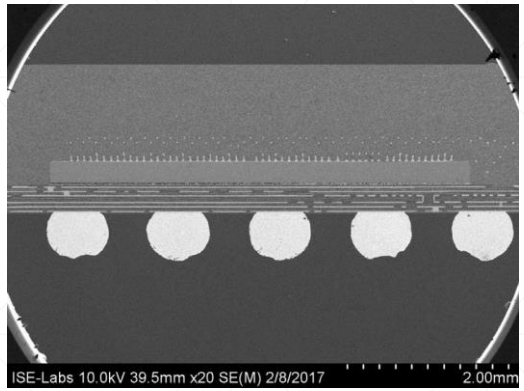


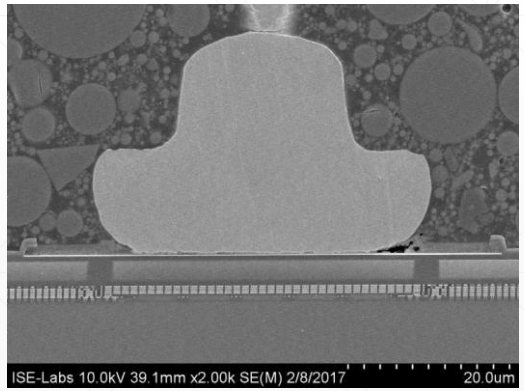
6 Layer Substrate



Manufactured SiP



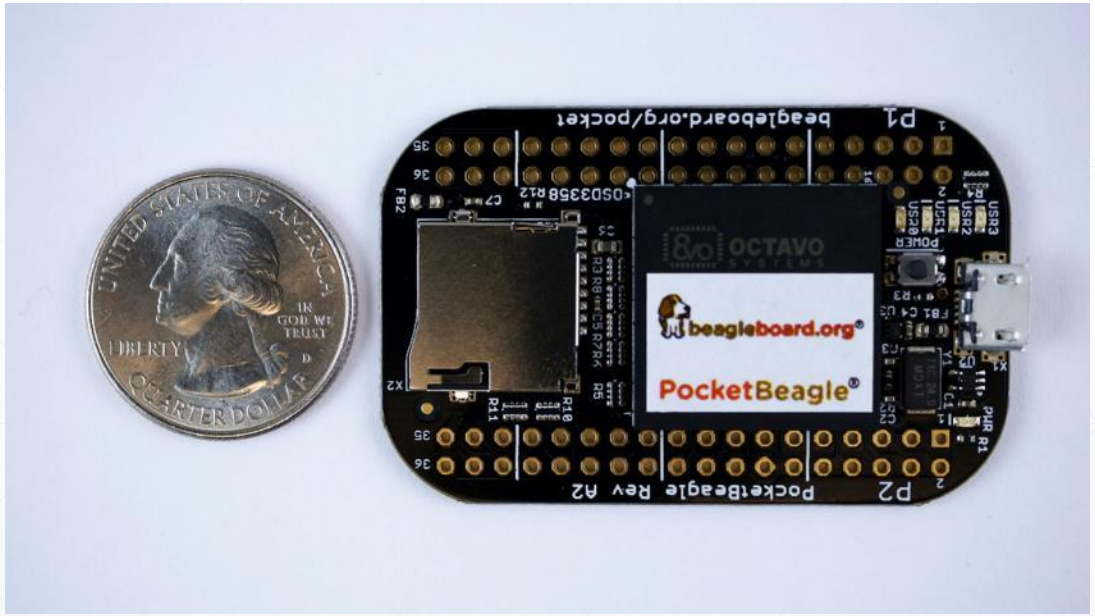




Cross-Section SEM Picture of SiP

PocketBeagle

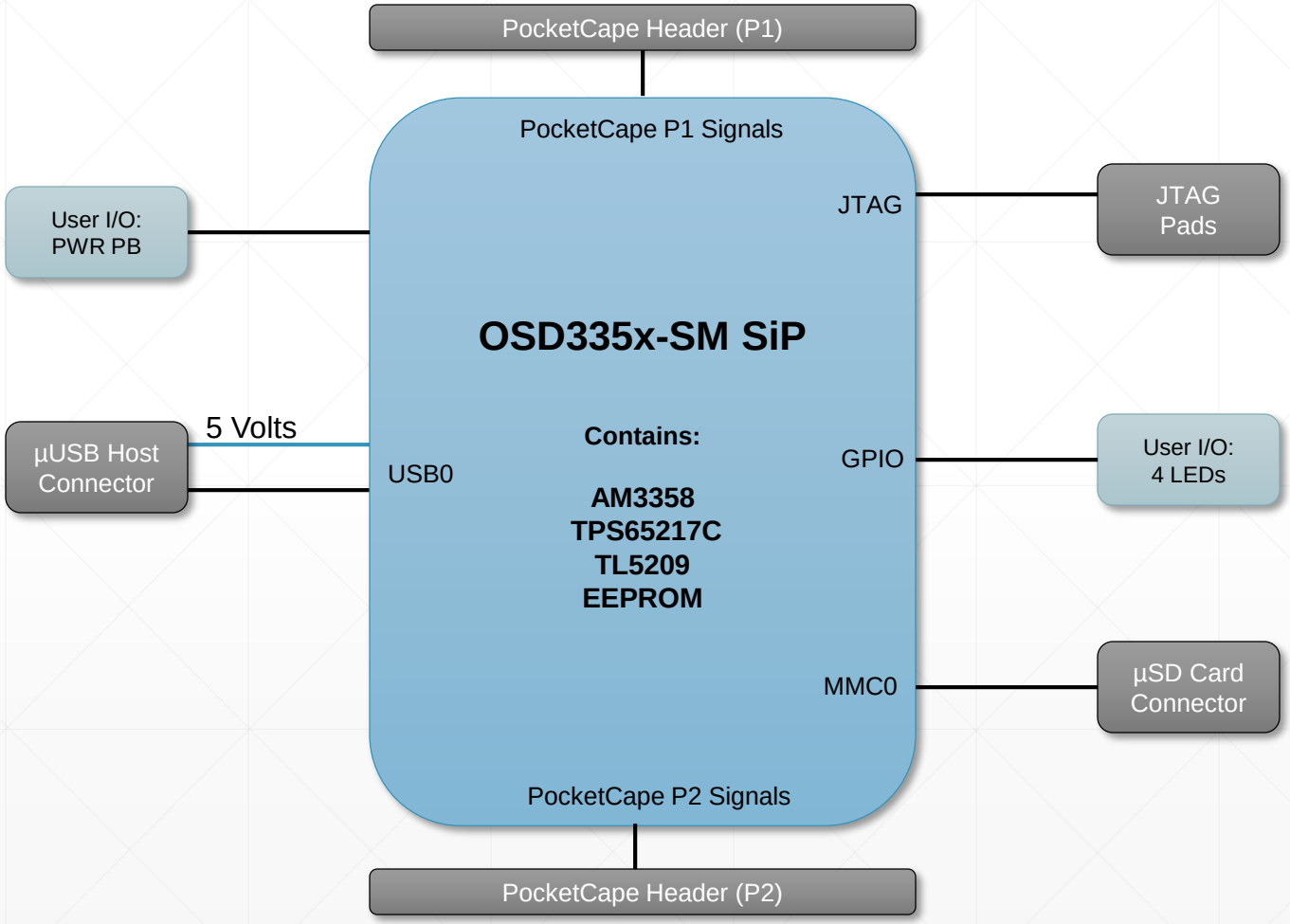
- http://bbb.io/pocket
  - Forums: http://bbb.io/discuss
  - News: http://bbb.io/news

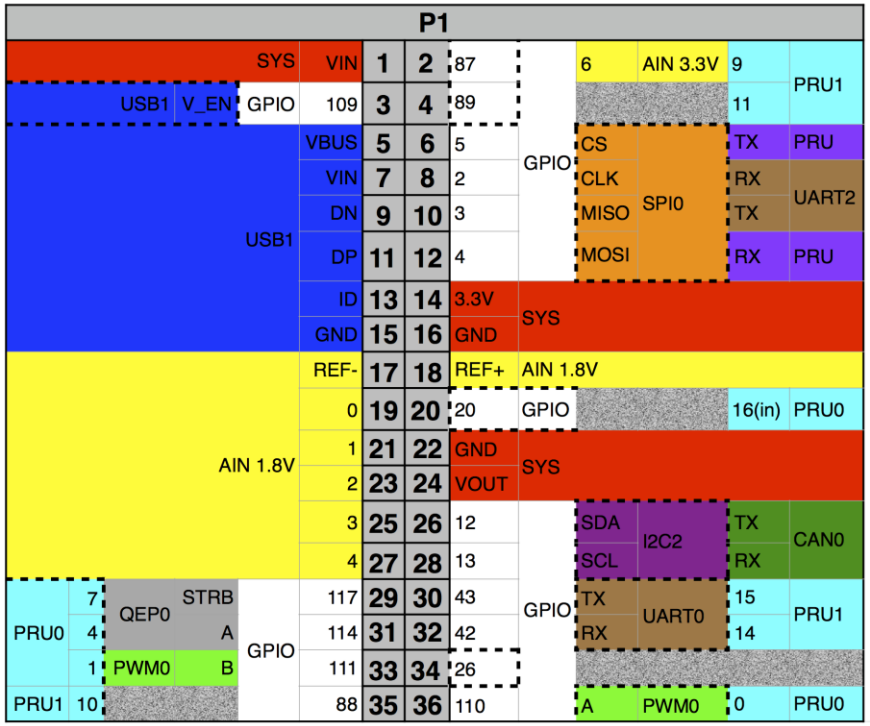


- Based on Octavo Systems OSD3358-SM SiP
  - ARM Cortex-A8 @ 1-GHz
  - 512 MB DDRs RAM integrated
  - ARM Cortex-M3
  - 2×200-MHz RISC Programmable Real-time Units (PRU)
  - Integrated power management
- Connectivity
  - Bootable microSD card slot
  - High speed USB 2.0 OTG (host/client) control signals
  - Dual 36-pin expansion headers
    - 8 analog inputs (6 @ 1.8V and 2 @ 3.3V)
    - 44 digital GPIOs
    - 3 UARTS
    - 2 I2C
    - 2 SPI
    - 4 PWM
    - 2 QEP
    - 2 CAN
- $25

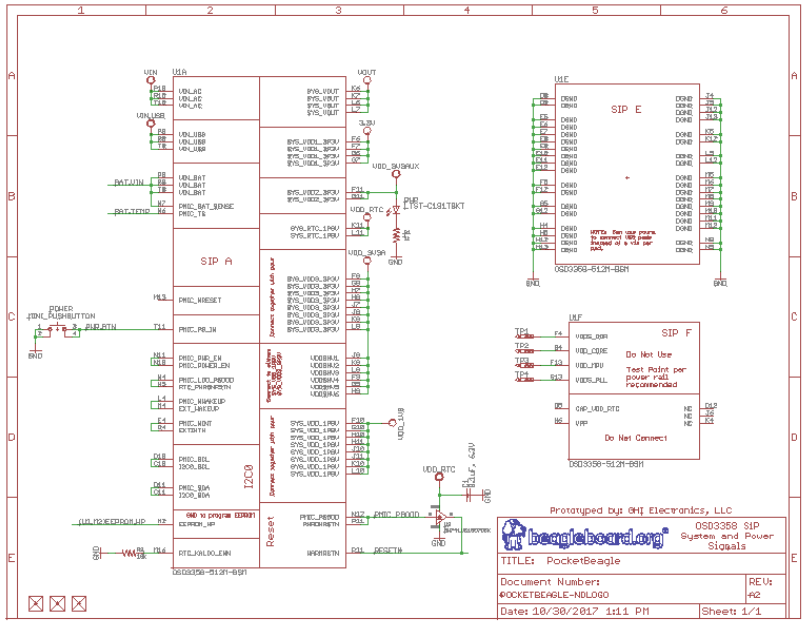
56mm x 35mm x 5mm

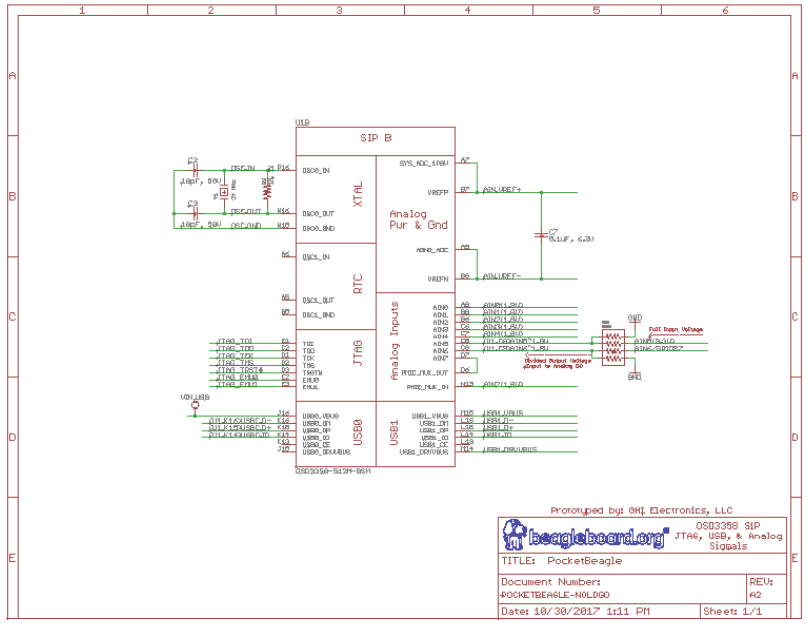
PocketBeagle Block Diagram

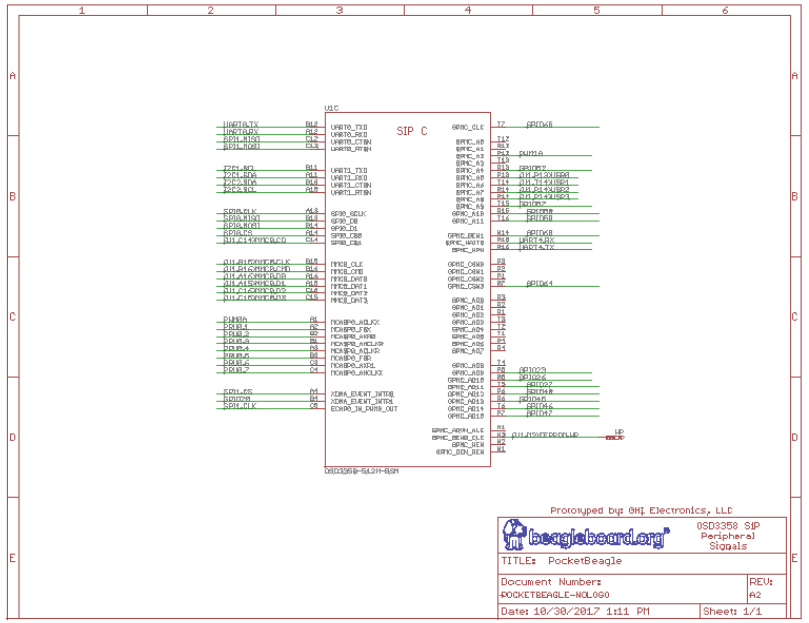


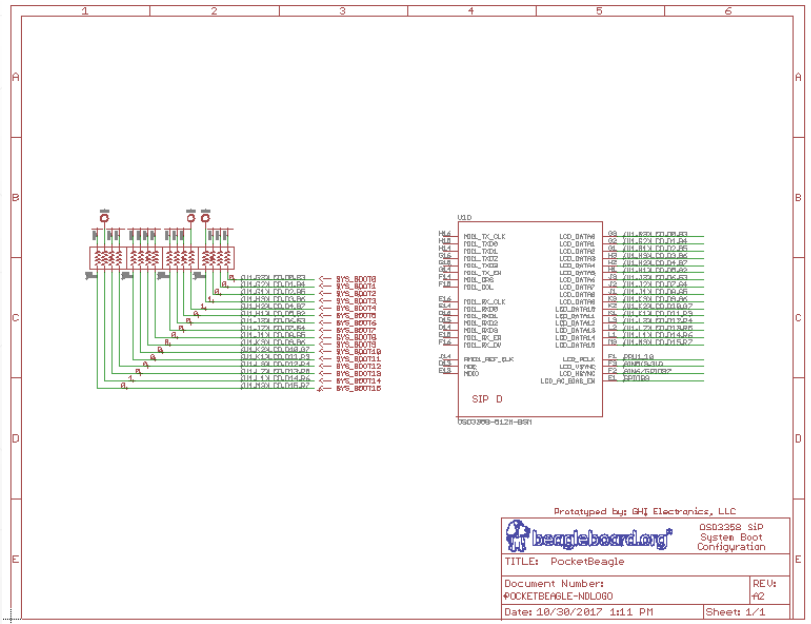


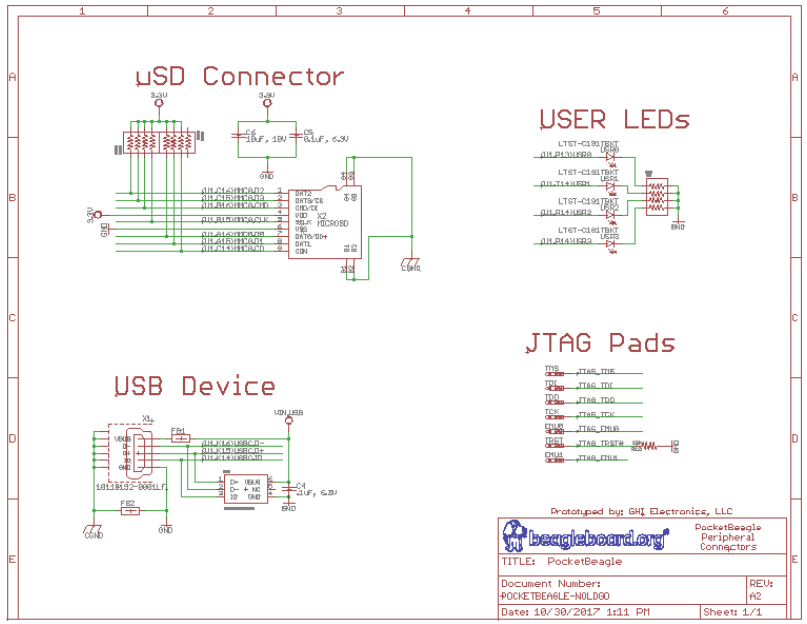
PocketBeagle Schematics

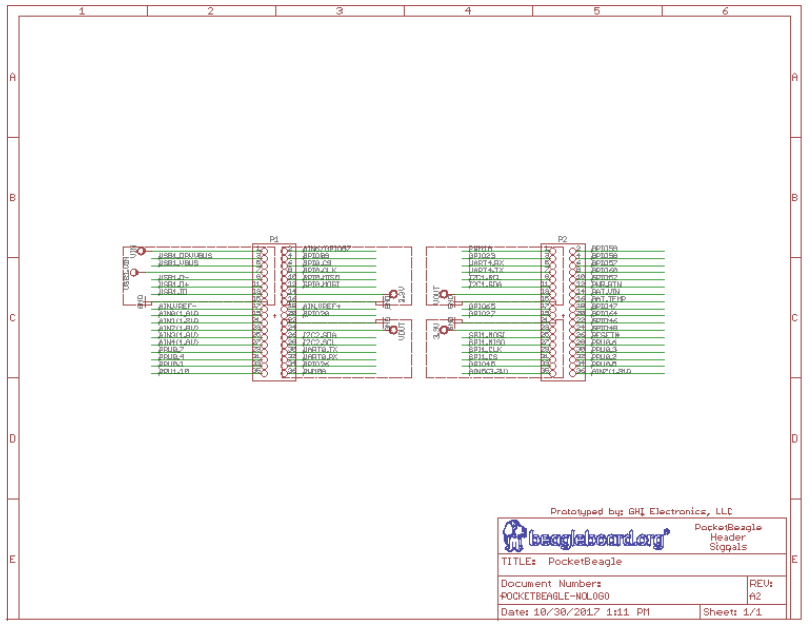




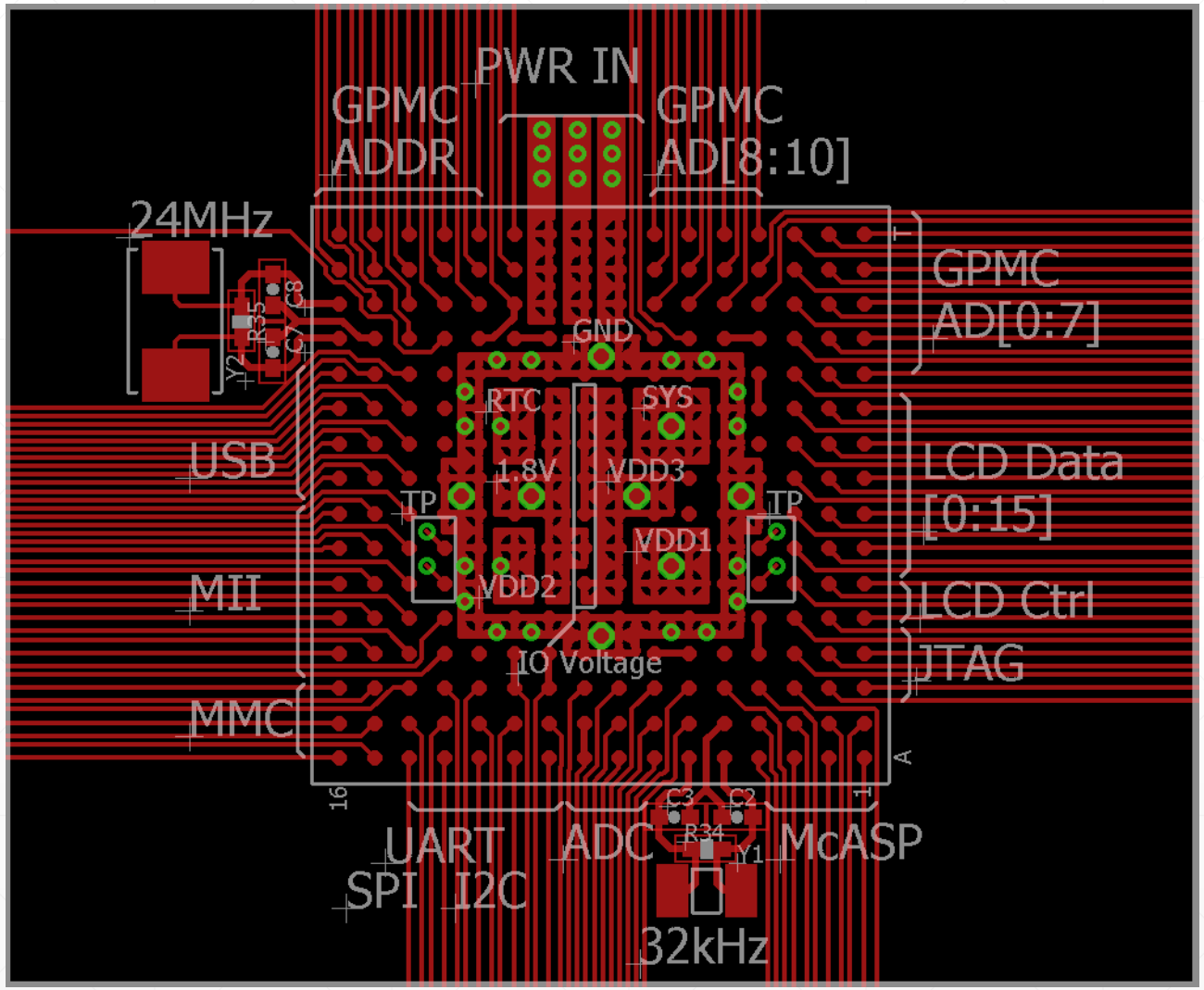








Simplified Layout



All Signals Escaped in a Single Layer

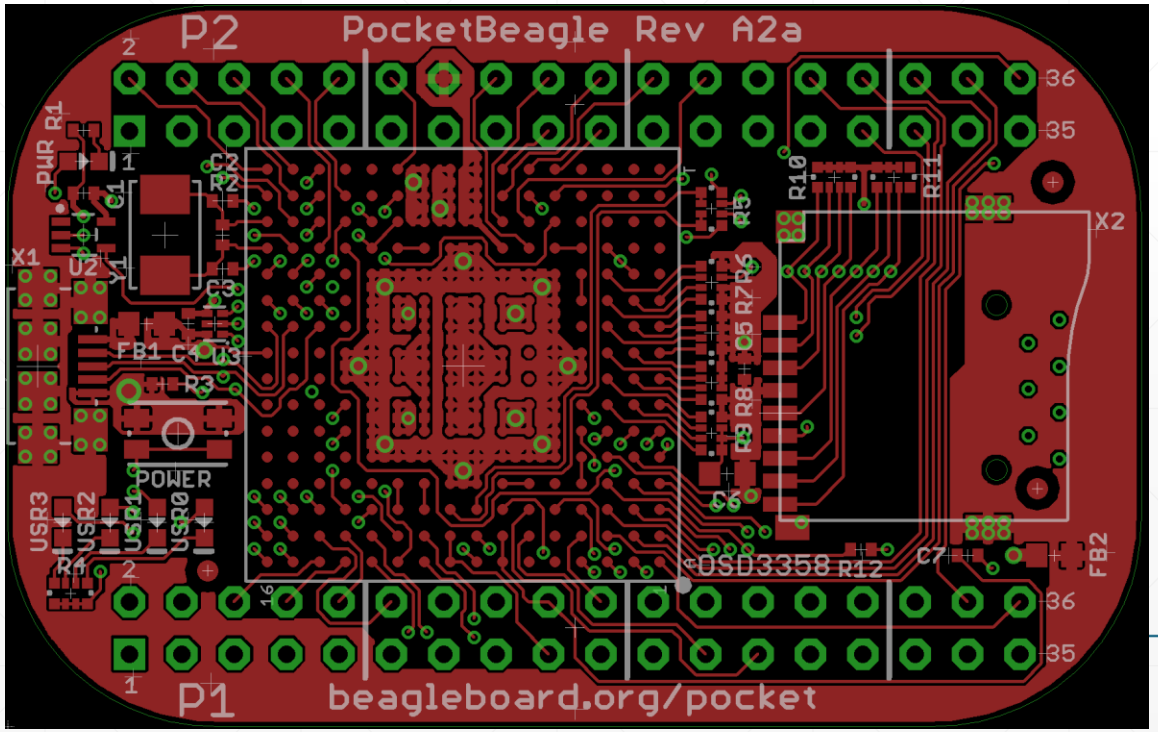
- 6 mil Trace Width
- 6 mil Space

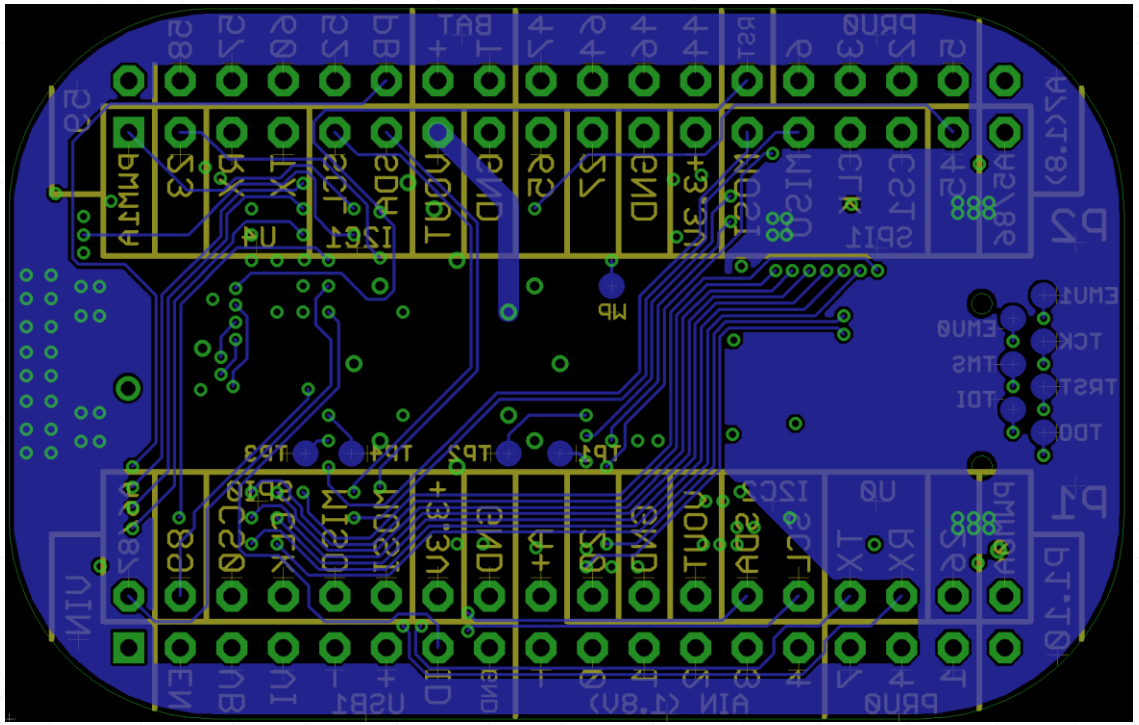
All Power Domains and Internal Signals located in the center for easy connection

https://octavosystems.com/app_notes/osd335x-sm-layout-guide/

PocketBeagle Layout

- Open Source Schematics & Layout
- 4 layers PCB
  - 6 mil trace / 6 mil space
  - 15 mil drill / 25 mil via



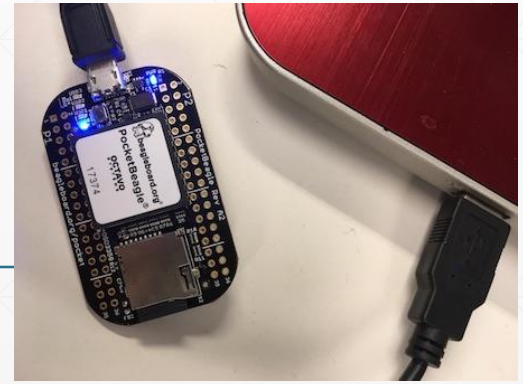


Simplified Board Bring Up Process

- Hardware Bring Up
  - Verify Power Isolation (ie your power rails are not shorted to ground – Don't release the magic smoke)

- Software Bring Up
  - Download Latest Image from BeagleBoard.org
  - Modify the device tree to meet your needs
  - Power up the board and check that everything boots properly

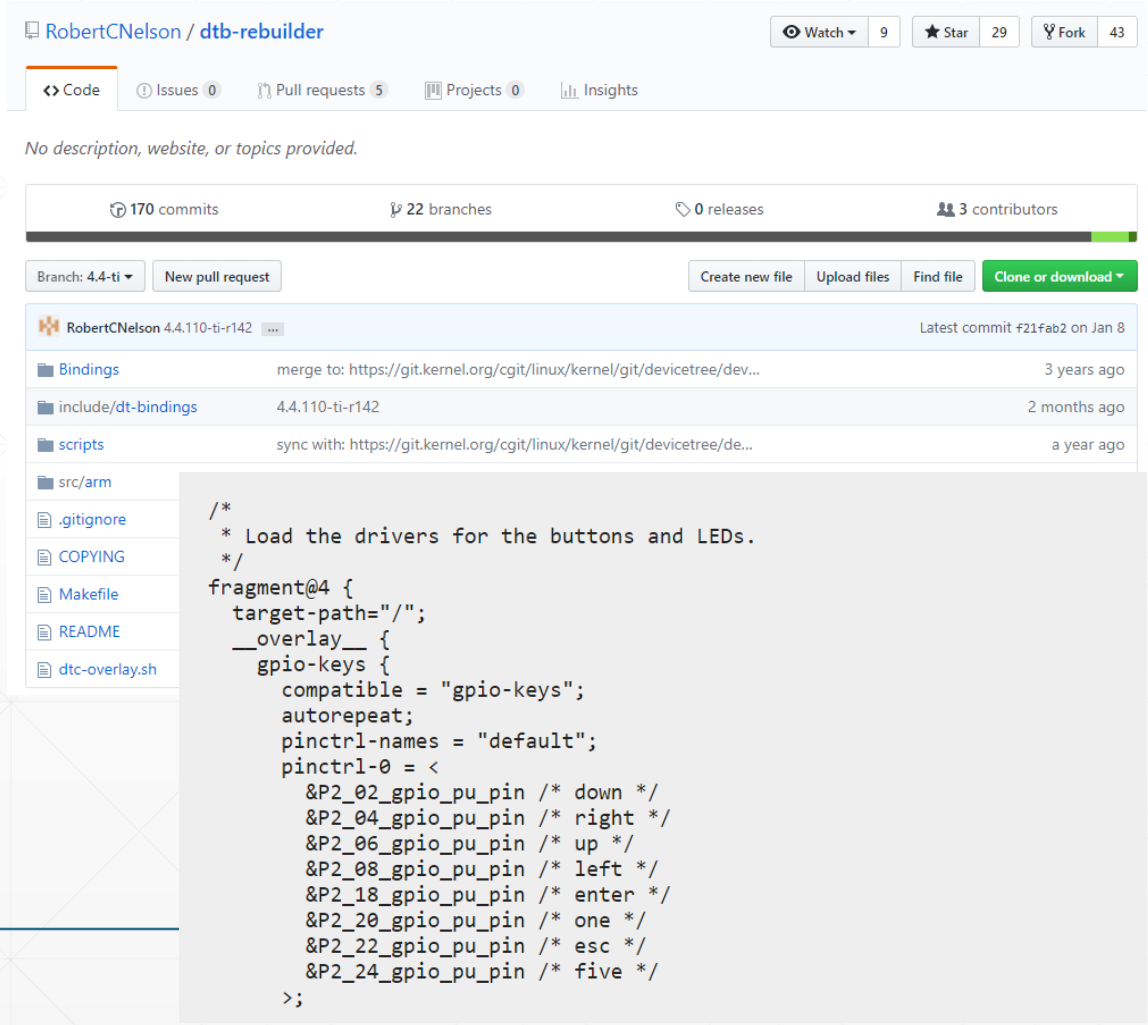
- You don't worry about
  - Bad voltages to the processor or DDR
  - DDR not working



Modifying Your Device Tree

- Development boards provide good device tree infrastructure
  - Majority of your device tree is already done for you
  - Only update the items that are different for your board
  - Many examples to mine for information / help
    - https://github.com/RobertCNelson/dtb-rebuilder

- Prototyping can be done with device tree overlay
  - Allows testing on your SBC prototyping board



Using a SiP in your Linux Computer Design Will:

- Bring you 100+ components in one package
  - Makes board design faster, simpler and easier to add your own new features
  - Ensures easy board bring-up
- Give you the heart of the Computer Hardware in a single BGA package
  - Lower cost PCB, fewer board layers, single sided
  - Easy to manufacture with - Some have even hand soldered it!
- Bridge the gap between Prototype and Production
  - Open Hardware + Open Source Software
  - Easy migration from SBC prototyping board to your custom PCB

Thank You

For more information come to our table at ELC Technical Showcase