

# *Embedded Linux Conference 2017:* Google Summer of Code and BeagleBoard.org

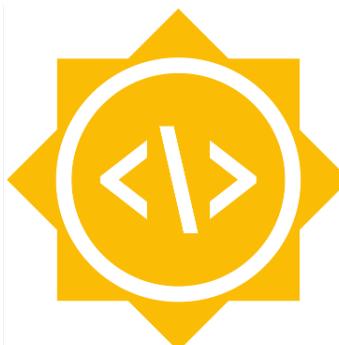


Google  
Summer of Code

Drew Fustini  
<[drew@beagleboard.org](mailto:drew@beagleboard.org)>  
twitter: @pdp7 / @beagleboardorg

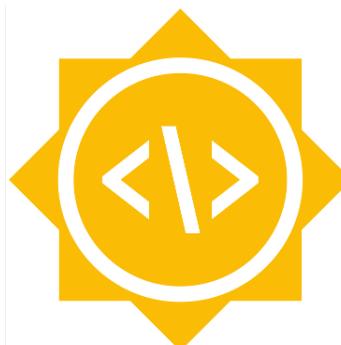
# What is Google Summer of Code?

**"Google Summer of Code is a global program that offers students stipends to write code for open source projects"**



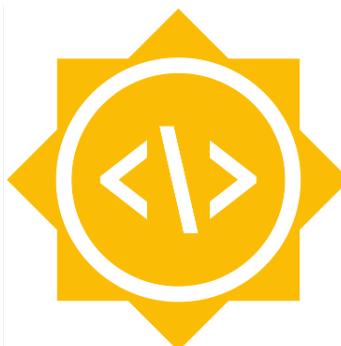
# What is Google Summer of Code?

- 12 years
- 104 countries
- 568 open source projects
- 12,000+ students
- Over 30 million lines of code



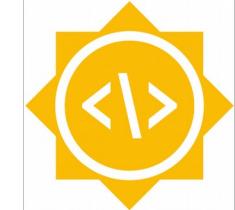
# Google Summer of Code 2016

- 178 open source projects
- 1,206 university students
- 67 countries
- 1,032 students (85.6%) completed
- **\$5,500** to each successful student





# 2017 GSoC Timeline



- **Jan. 19 – Feb. 9:** organizations apply
- **Feb. 27:** accepted organizations published
- **Feb. 27 – March 20:** potential student participants discuss application ideas with orgs
- **March 20 – April 3:** student applications
- **May 4:** accepted student proposals announced
- **May 30:** students begin coding
- **June:** mid-term evaluations
- **August:** final eval & project submissions
- **October:** Mentor Summit at Google



# Student Requirements



- Must be at least 18 years of age
- Must currently be a full or part-time student (or have been accepted for fall term) at an accredited university
- Must be eligible to work in the country you will reside in during the program
- Not already been accepted as a Student in GSoC more than once
- Must reside in a country that is not currently embargoed by the United States



# Student Stipends



- Stipends are paid after each successful evaluation.
  - First Evaluation (paid early July): 30%
  - Second Evaluation (paid early August): 30%
  - Final Evaluation (paid mid September): 40%
  - Stipend amounts are calculated based on your location.
- 2017: Purchasing Power Parity (PPP) based calculation to determine the stipend
  - Minimum 2400 USD & maximum 6600 USD



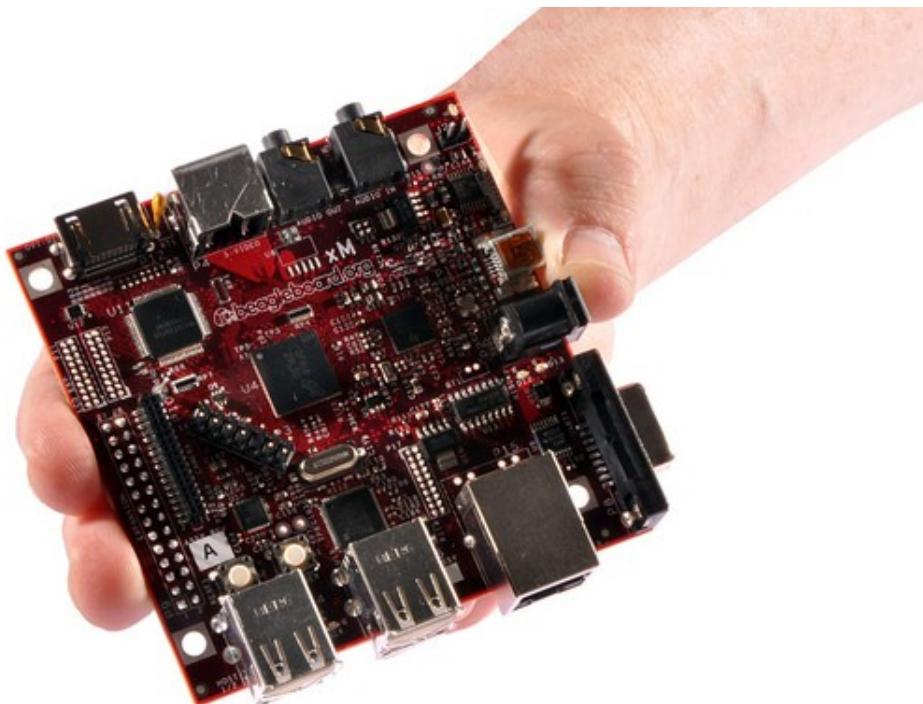
**Open Source Hardware computing**

*for*

**Makers, Educators & Professionals**

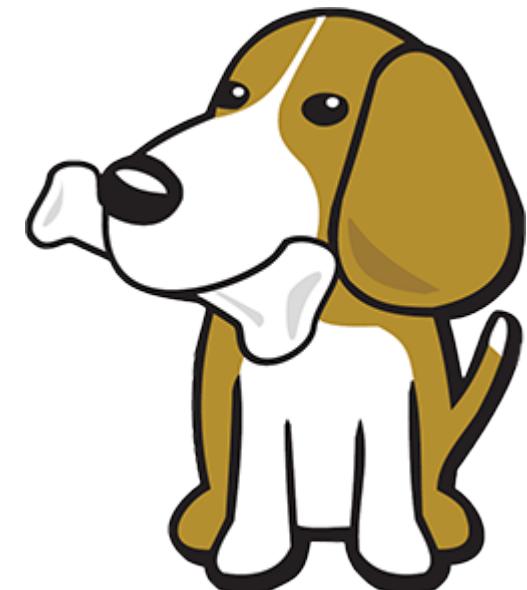


BeagleBoard.org released the first **BeagleBoard**, an affordable, open hardware computer in **2008**





Maker focused, Altoids tin sized  
**BeagleBone** introduced in 2011





More affordable, more powerful  
**BeagleBone Black in 2013**



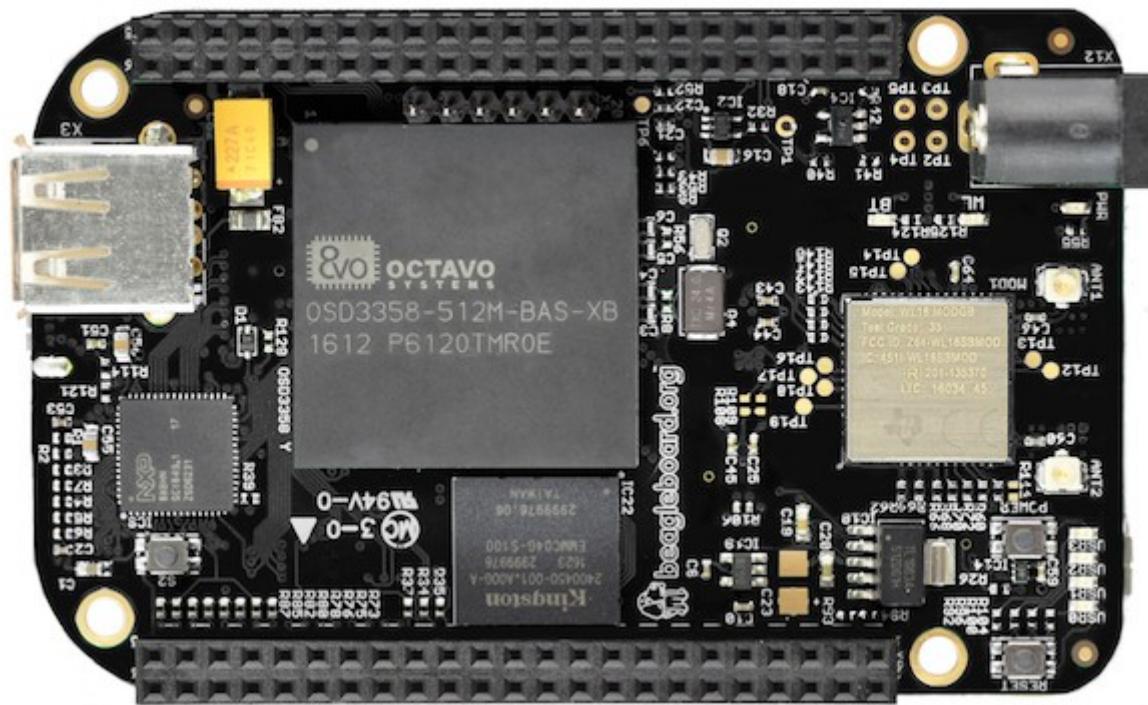


# Open Source Hardware BeagleBone derivatives

	Capes	HDMI	Flash	Special
BeagleBoard.org BeagleBone	Y	N	N	JTAG
BeagleBoard.org BeagleBone Black	Y	Y	Y	-
Arrow BeagleBone Black Industrial	Y	Y	Y	Industrial
Element14 BeagleBone Black Industrial	Y	Y	Y	Industrial
SeeedStudio BeagleBone Green	Y	N	Y	Grove
SanCloud BeagleBone Enhanced	Y	Y	Y	1GB, 1Gbit, wireless
BeagleBoard.org BeagleBone Blue	N	N	Y	Robotics
BeagleBoard.org BeagleBoard-X15	N	Y	N	Big jump in CPUs and I/O



# BeagleBone Black Wireless



- WiFi 802.11b/g/n and Bluetooth 4.1 with BLE
- 1<sup>st</sup> Beagle with Octavo System-in-Package (SiP)
- Designed in EAGLE (*BBB was OrCad/Allegro*)



# Technical Showcase

CE Workgroup Linux Foundation / Embedded Linux Conference

## BeagleBoard.org BeagleBone Blue preview

Jason Kridner

### What is demonstrated

Open hardware Linux computer  
→ Self-hosted web IDE  
→ Debian Linux system

Self-balancing robot run as a userspace task written in C  
→ 5ms inner-loop, <25% load

Connected USB camera  
→ mjpg-streamer over WiFi

Bluetooth speaker for audio

ROS interface



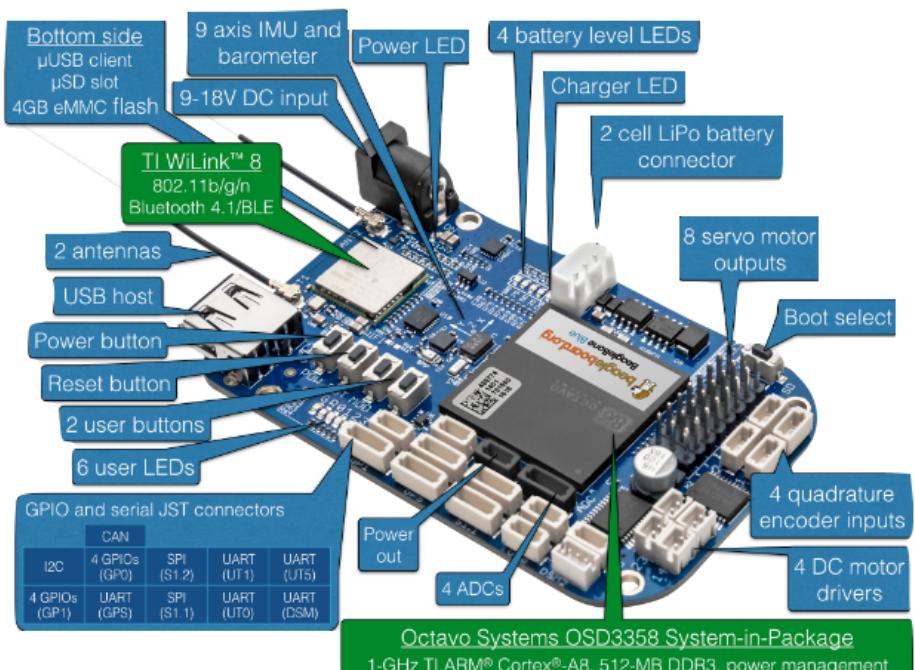
Hardware Information

<https://bbb.io/blue>

1-GHz ARM Cortex-A8, programmable real-time units, 4×DC motor drive, 8×servo motor, WiFi/BT, USB, IMU, 2 cell LiPo battery mgmt, 4×quad enc. barometer

### What was improved

Pre-integrated H/W elements for robotics  
→ pre-integrated software: Ardupilot, ...  
4-layer PCB design  
Quick connections for individual interfaces



Source code or detail technical information availability

<https://github.com/jadonk/beaglebone-blue>

<https://github.com/beagleboard/linux>

<https://github.com/beagleboard/image-builder>



## BeagleBoard.org BeagleBoard-X15

Jason Kridner

### What is demonstrated

Open hardware computer

- ▶ Debian Linux system
- ▶ Open source 2D graphics acceleration
- ▶ Video acceleration
- ▶ OpenCL C66 DSP support
- ▶ Mainline kernel support
- ▶ GCC compiler support
  - ✓ ARM Cortex-A15
  - ✓ ARM Cortex-M4
  - ✓ TI C66x
  - ✓ TI PRU



### What was improved

Fastest BeagleBoard available

More cores and more types of cores

Lots more I/O capability and bandwidth

More RAM (2GB)

Great open hardware ARM build platform

Hardware Information

<https://bbb.io/x15>

Dual-core ARM Cortex-A15, dual C66x DSPs, quad programmable real-time units, 3×USB 3.0, PCIe, 2×gigE, 2GB RAM, 4GB eMMC flash

Source code or detail technical information availability

<https://github.com/beagleboard/beagleboard-x15>

<https://github.com/beagleboard/linux>

<https://github.com/beagleboard/image-builder>



- **BeagleBoard.org Foundation** is US-based non-profit corporation
- Provides education around the design and use of Open Source Software and Open Source Hardware
- Fosters communication between individuals interested in Open Source

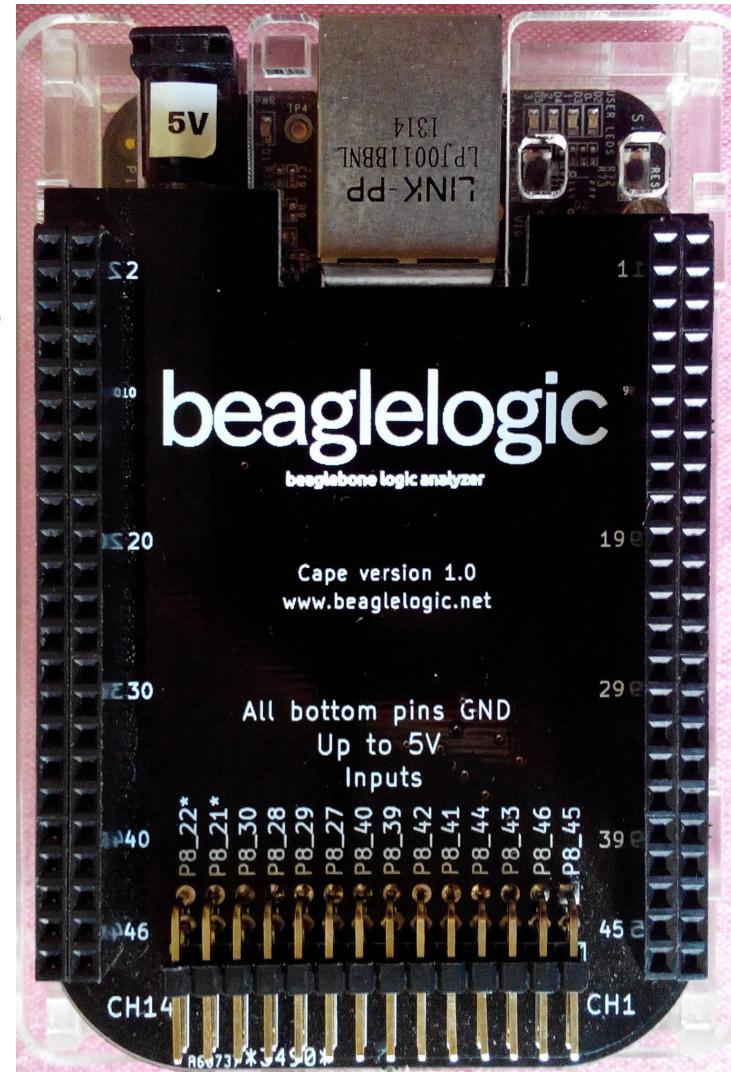


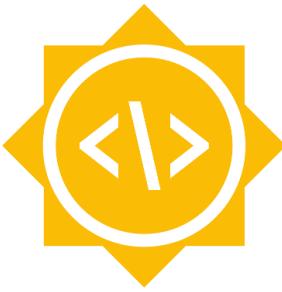
# beaglelogic

beaglebone logic analyzer



- **Kumar Abhishek** created BeagleLogic for GSoC 2014
- BeagleLogic turns BeagleBone into **Logic Analyzer**
- 14-channel, 100Msps
- Web browser user interface
- **Video of final presentation**





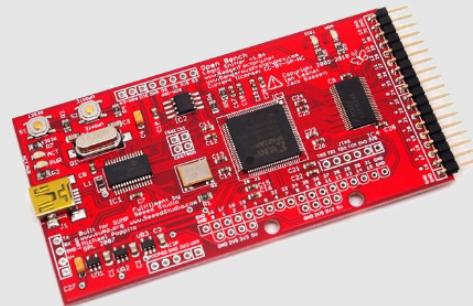
## Why BeagleLogic?

- Best In Class Sample Buffer Size – 320 MiB, 100 Msamples/sec  
(Excluding the cape + \$55 / \$39 for a BeagleBone Black / Green)  
3.3 secs of high-speed 8 channel data!
- Capture and Debug on the same hardware
  - Full-featured **sigrok** software stack preloaded
  - Decode over 30 digital protocols (I<sup>2</sup>C, I<sup>2</sup>S, SPI, UART, WS2812, 1 Wire, nRF24L01, DS1307, ARM SWO Trace ...)
- Networking Capabilities enabling remote debugging
- Web Interface
- Open Source Software & Open Hardware



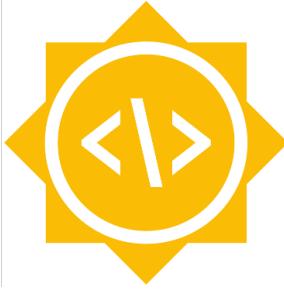
# beaglelogic

beaglebone logic analyzer



	OLS	Saleae Logic 8	BeagleLogic
Price	\$50	\$219	Cape + \$39(BeagleBone Green)
Max Channels	32	8	14
Max Sample Rate	200 MS/s	100 MS/s	100 MS/s
Sample Depth (8 ch)	16KSamples (0.008sec)	Limited by PC	320 MSamples (3.2sec)
Triggering	Hardware	Software	Software
Interface	USB 2.0 12Mbps, UART	USB 2.0, 480Mbps	USB 2.0 480Mbps RNDIS, LAN
Remote Debug	No	No	Yes

BeagleLogic – Best feature/price balance in the sub-100\$ category



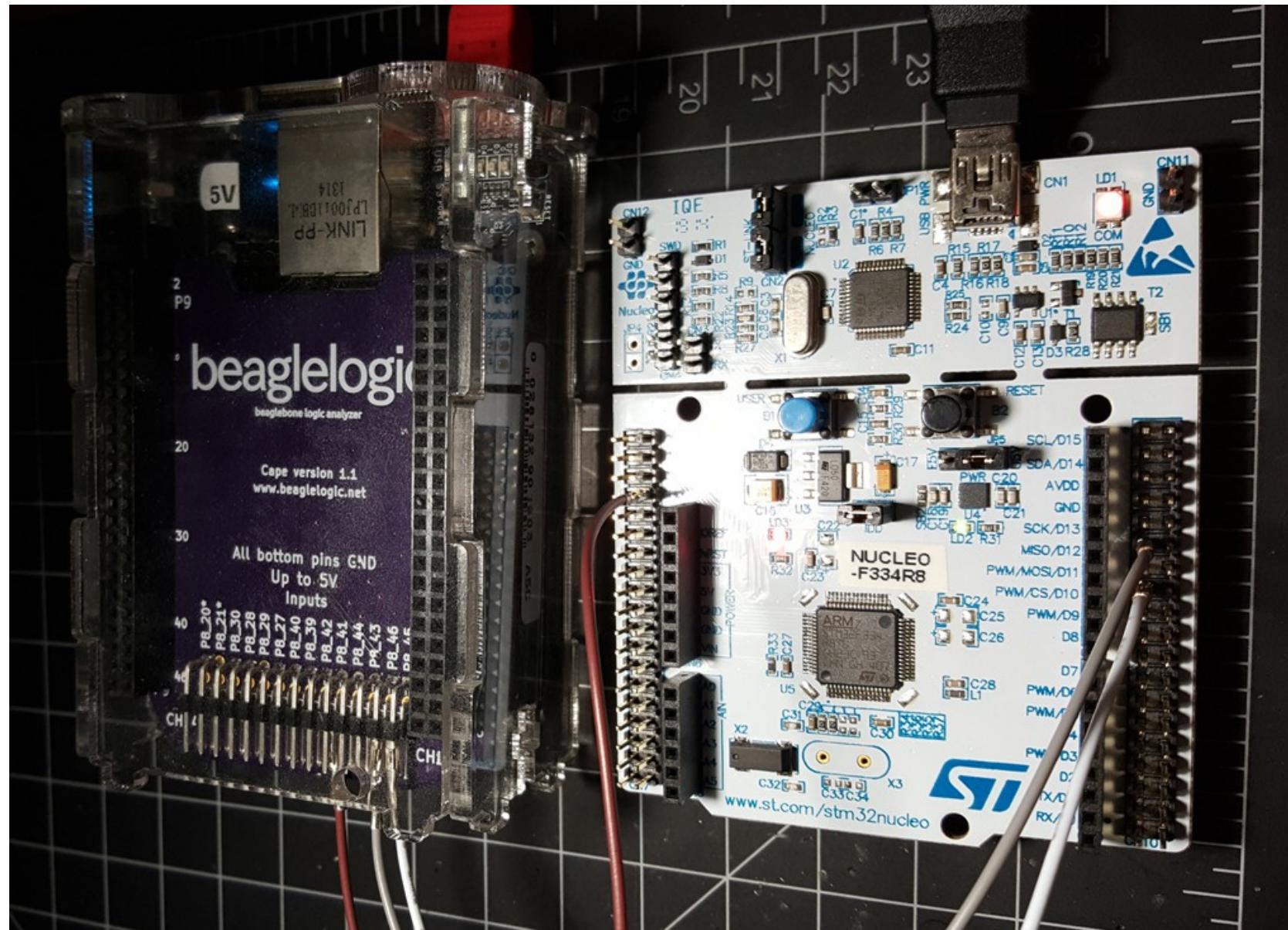
## Quick Start

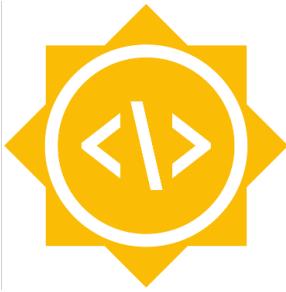
- Flash the BeagleLogic System Image (Debian distribution, 900 MB download) onto a SD card.
- Insert the SD Card into the BeagleBone Black / Green.
- Connect the BeagleBone to your PC via USB and install drivers (if on Windows)
- Navigate to <http://192.168.7.2:4000> (USB) or <http://<ip-of-beaglebone>:4000/> (LAN)



# beaglelogic

beaglebone logic analyzer





# beaglelogic

beaglebone logic analyzer



BeagleLogic - Google Chrome

Messages • Hackada x BeagleLogic • Hacka x BeagleLogic

192.168.7.2:4000

BeagleLogic A logic analyzer on the BeagleBone Black Help About

Configuration

Sample Rate: 5 MHz

Sample Limit: 1000

Input Selection and Annotation

P8_19	<input type="checkbox"/>	P8_20
P8_21	<input type="checkbox"/>	P8_22
P8_23	<input type="checkbox"/>	P8_24
P8_25	<input type="checkbox"/>	P8_26
P8_27	<input type="checkbox"/>	P8_28
P8_29	<input type="checkbox"/>	P8_30
P8_31	<input type="checkbox"/>	P8_32
P8_33	<input type="checkbox"/>	P8_34
P8_35	<input type="checkbox"/>	P8_36
P8_37	<input type="checkbox"/>	P8_38
P8_39	<input type="checkbox"/>	P8_40
P8_41	<input type="checkbox"/>	P8_42
P8_43	<input checked="" type="checkbox"/>	P8_44
P8_45	<input checked="" type="checkbox"/>	P8_46

Rendered in 486 ms.

▶ Begin Capture    ⌂ Save Capture    ⚡ Dump Raw Data

P8\_45

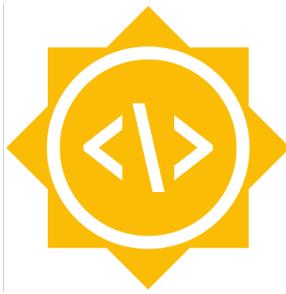
P8\_46

P8\_43

Requesting capture.  
Received 3080 bytes of data.  
Rendering... This may take a couple of seconds, and make the browser window non-responsive. Please be patient!

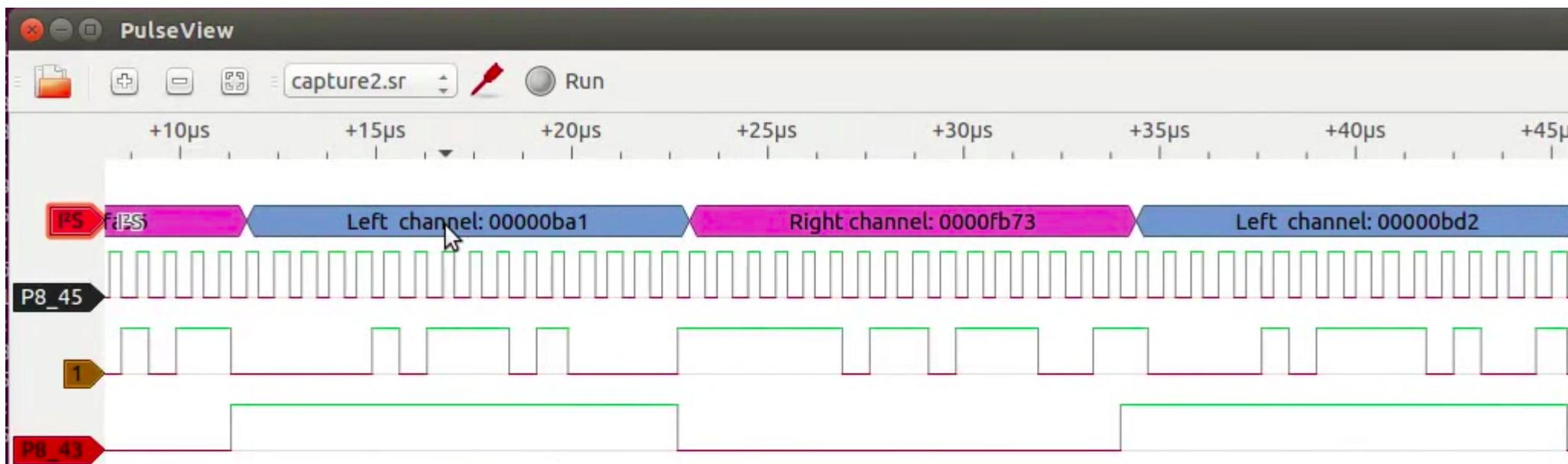


BeagleLogic - Logic Zero to One in 2 minutes



# beaglelogic

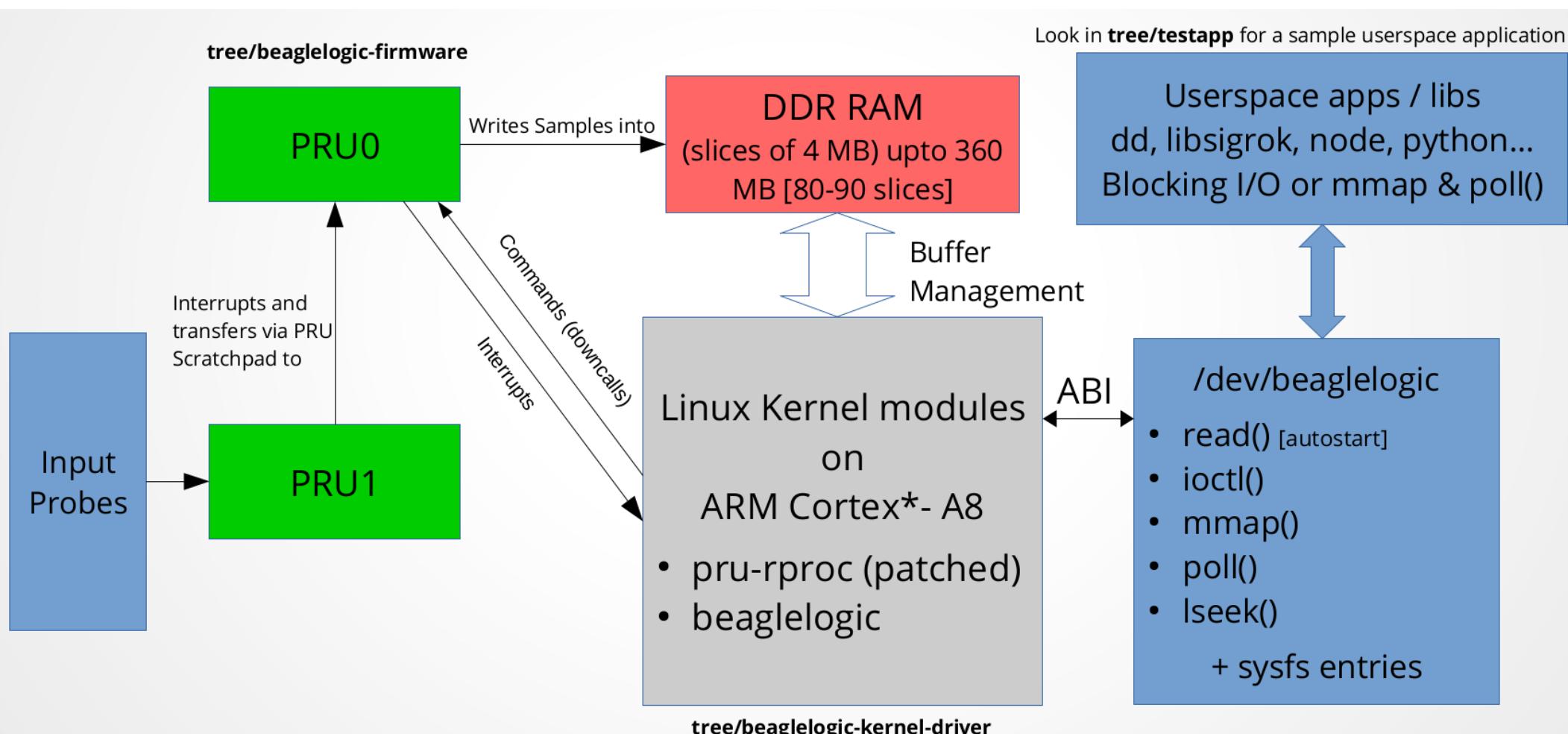
beaglebone logic analyzer



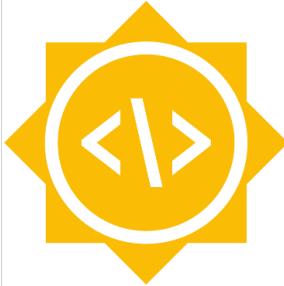


# beaglelogic

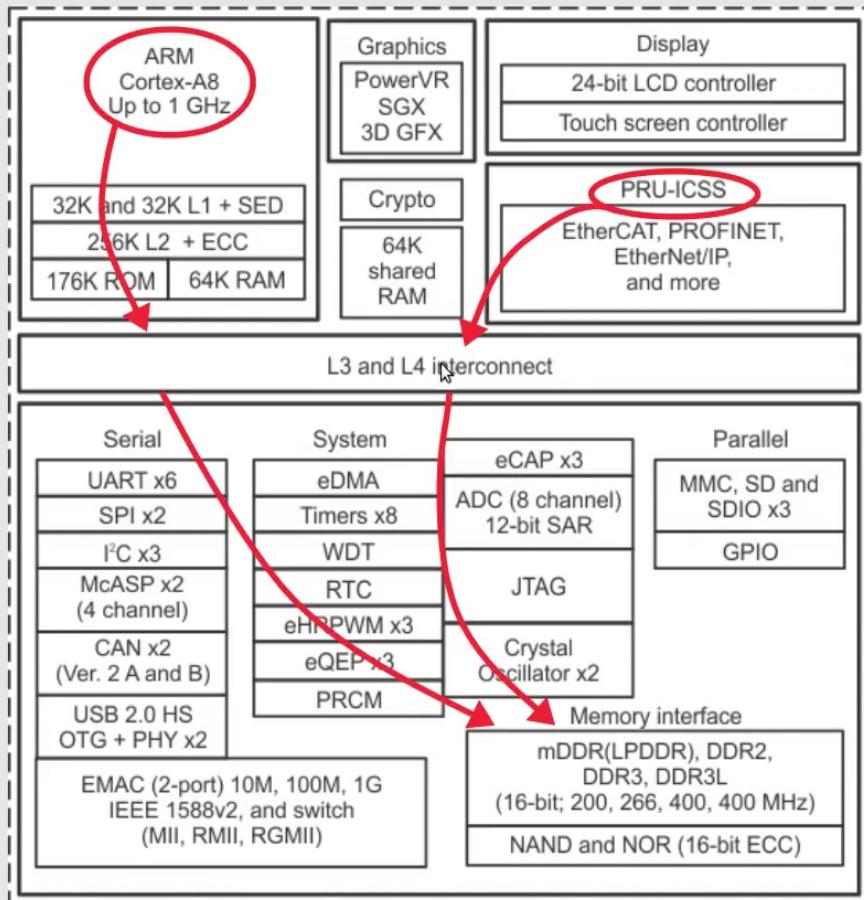
beaglebone logic analyzer



Source: <https://hackaday.io/project/4395-beaglelogic>



## The AM3358 SoC

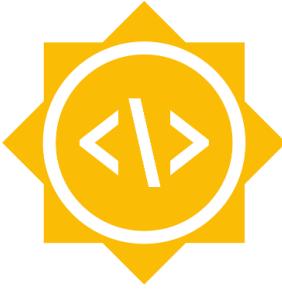


**PRU**  
[Programmable Real-Time Unit]:

Two Programmable 200 MHz Microcontrollers on chip for real-time tasks.

Share the interconnect which connects the ARM core to system memory

Can access DDR3 RAM independent of ARM Core!



## For Developers

- PRU Firmware
- Linux Kernel Driver [beaglelogic kernel module]
- Front-end /dev/beaglelogic **Character Device**
  - Appears as a standard file
  - open( ) to initialize
  - read( ) to sample, block-waits accordingly until data is available
  - ioctl( ) to configure sample rate and other settings
  - Non-blocking and Zero Copy I/O support via Memory Mapping [ mmap( ) ]
- NodeJS server, SocketIO link between Web interface & BeagleBone

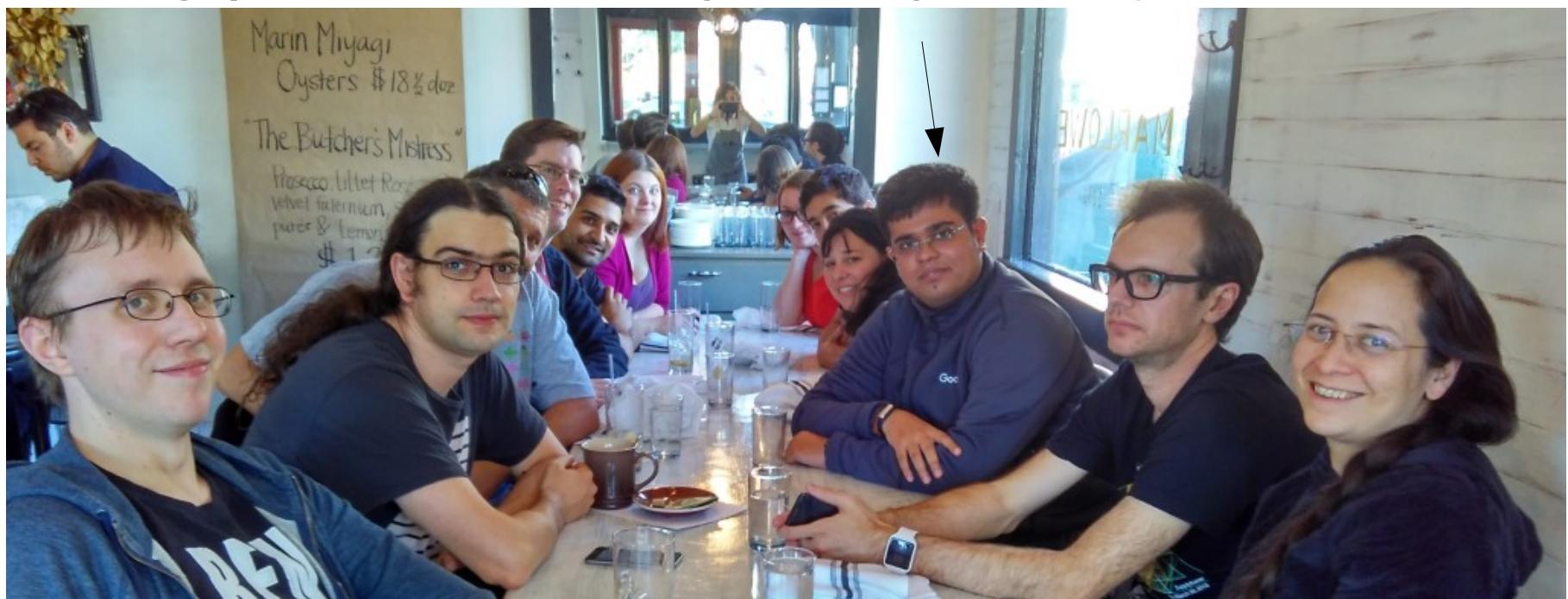


# beaglelogic

beaglebone logic analyzer



- **Best Product finalist** in **2015 Hackaday Prize**
- Traveled to Google Summer of Code Mentor Summit and Hackaday SuperCon in California
- Blog post about his journey: *A day with Hackaday*

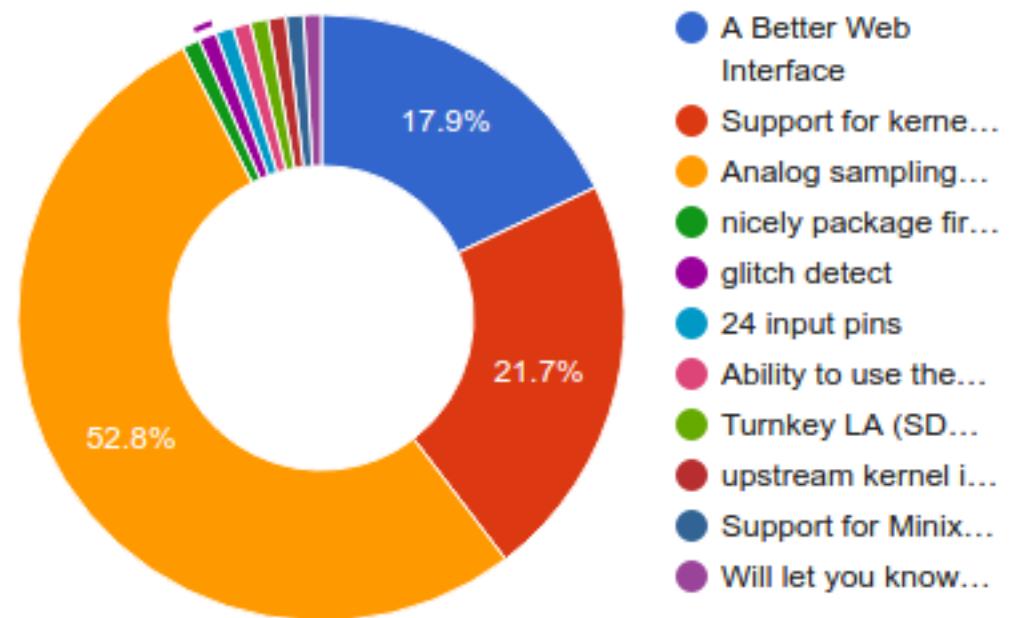


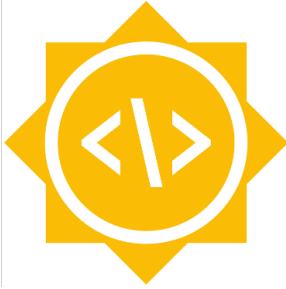


- Summer 2016: Kumar was intern at Google HQ
- July 2016: BeagleLogic: now also analog

“Majority of prospective users wanted to be able to do analog sampling with BeagleLogic”

Which one of these features would you like to see the most in BeagleLogic?





# beaglelogic

beaglebone logic analyzer



- July 2016 - **Google Research** blog announced **PRUDAQ**, an ADC cape for BeagleBone:

Announcing an Open Source ADC board for BeagleBone





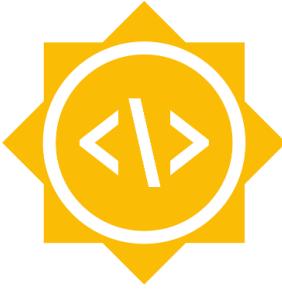
# beaglelogic

beaglebone logic analyzer

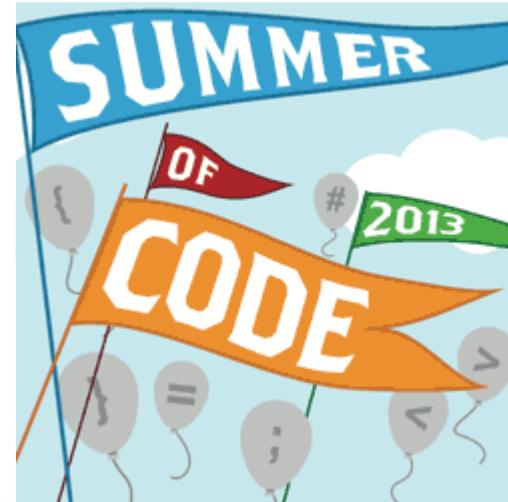


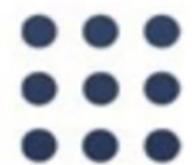
- **Google Research Blog:** “We also were fortunate to have help from Google intern Kumar Abhishek. He added support for PRUDAQ to his Google Summer of Code project BeagleLogic that performs much better than our sample code.”





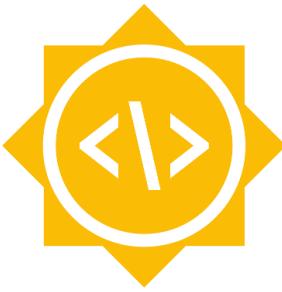
# Beagle-ROS



 ROS

Víctor Mayoral Vilches

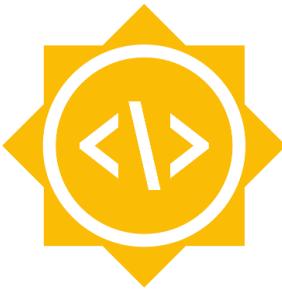
Mentor: Koen Kooi



# Beagle-ROS



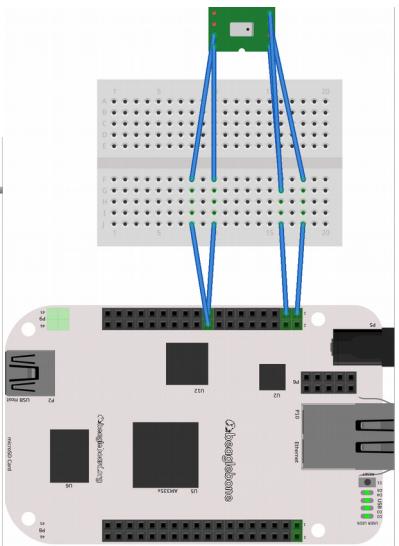
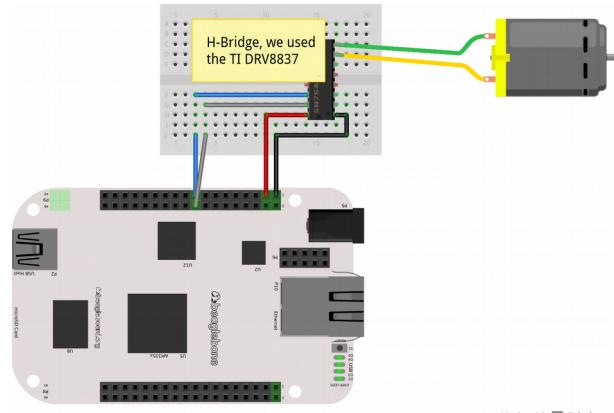
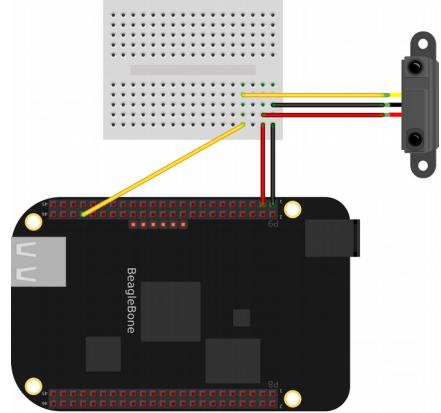
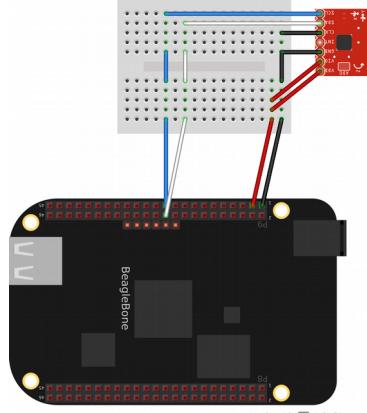
- Víctor Mayoral Vilches for GSoC 2013
- Integration of Robot Operative System (ROS) and BeagleBone through the meta-ros project, a layer for OpenEmbedded Linux
- GitHub: [vmayoral/beagle-ros](https://github.com/vmayoral/beagle-ros)
- Blog: [Beagle-ROS](#)
- Video: [Beagle-ROS Final](#)



# Beagle-ROS



- BeagleBone ROS Packages
  - **bb\_ altimeter**: publishes the altimeter MPL3115A2 values to a Topic
  - **bb\_dc\_motors**: launches a node to control a DC motor connected to the BeagleBone
  - **bb\_mpu9150**: publishes the Invensense MPU-9150 data into a Topic
  - **bb\_sharp\_ir**: Sharp IR sensors





# BeaglePilot



- **Víctor Mayoral Vilches** for GSOC 2014
- Linux-based autopilot for flying robots based on BeagleBone
- Ported ArduPilot to Linux
- ROS integration
- Videos: [Introduction](#) & [Final report](#)
- GitHub: [BeaglePilot](#)





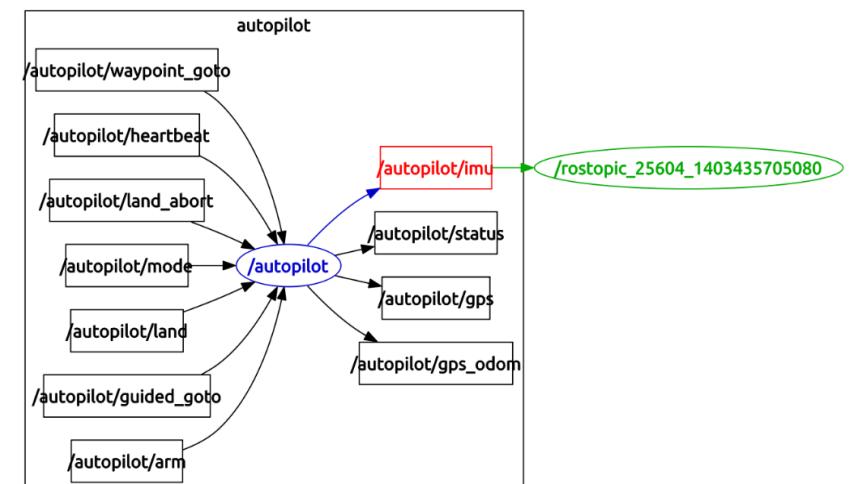
# BeaglePilot



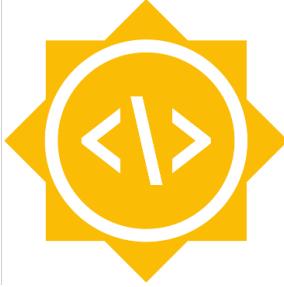
- **Towards an Open Source Linux autopilot for drones**
  - Publication accepted at LibreCon 2014
  - “Linux can perfectly be used to meet the real-time requirements needed by an autopilot requiring only about 25% of the processor in BeagleBone Black.”

**Table 2: Kernel benchmarking results**

Kernel type	Min (us)	Avg (us)	Max (us)
vanilla	14	19	193
PREEMPT	16	21	68
RT_PREEMPT	20	27	91
Xenomai	15	23	630



**Figure 6: autopilot\_bridge topics and nodes pictured with rosgraph**



# BeaglePilot



- Victor co-founded **Erle Robotics** to develop commercial products based on BeaglePilot
- **Erle-Brain**: “An artificial brain for making robots and drones”

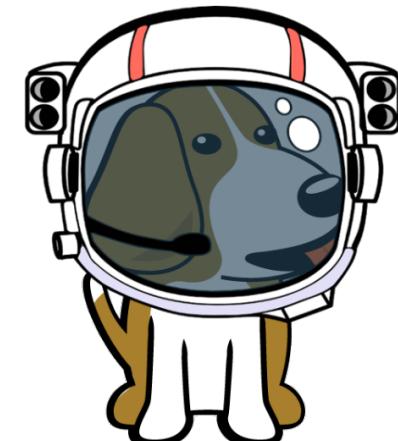




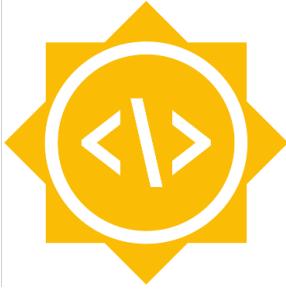
# BeagleSat



- **Niko Visnjic** for GSoC 2015
- Nano satellite platform based on BeagleBone
- Github repo: [nvisnjic/BeagleSat](#)
- [Project video](#)



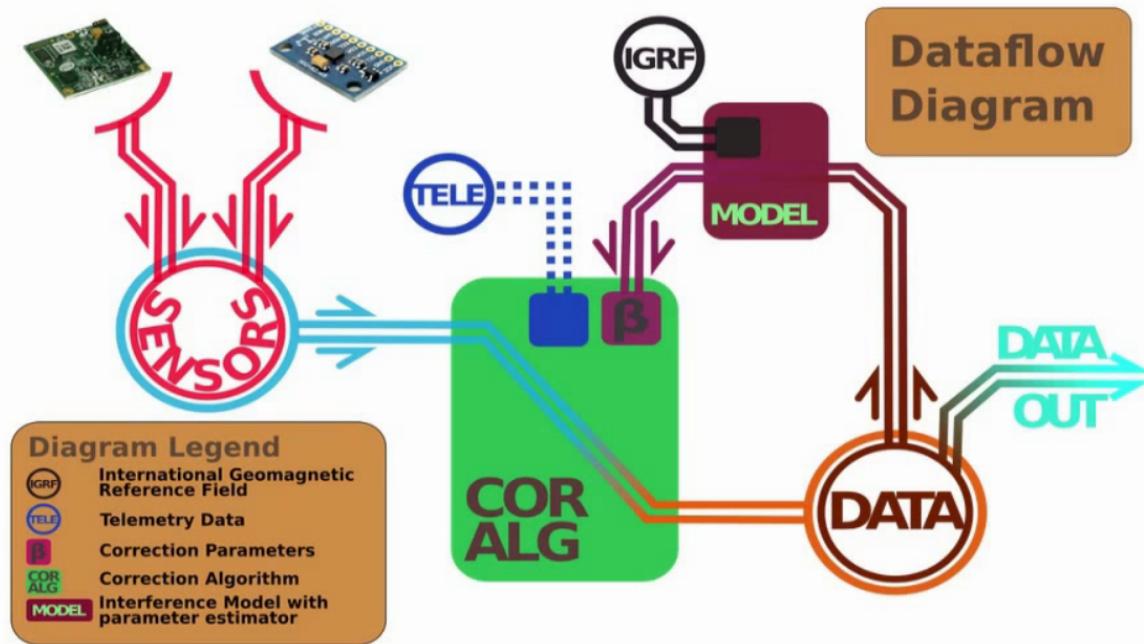
**beaglesat**

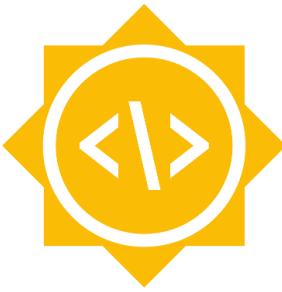


# BeagleSat



- Run data fitting algorithms in real-time on the BeagleBone Black
- Interface with the MPU9250 sensor using the PyBBIO library in Python
- Data visualizer during data collection and correction

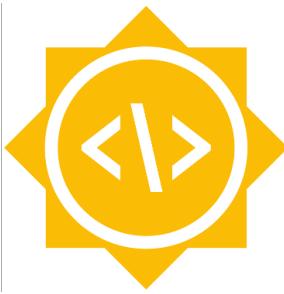




# USB Sniffer



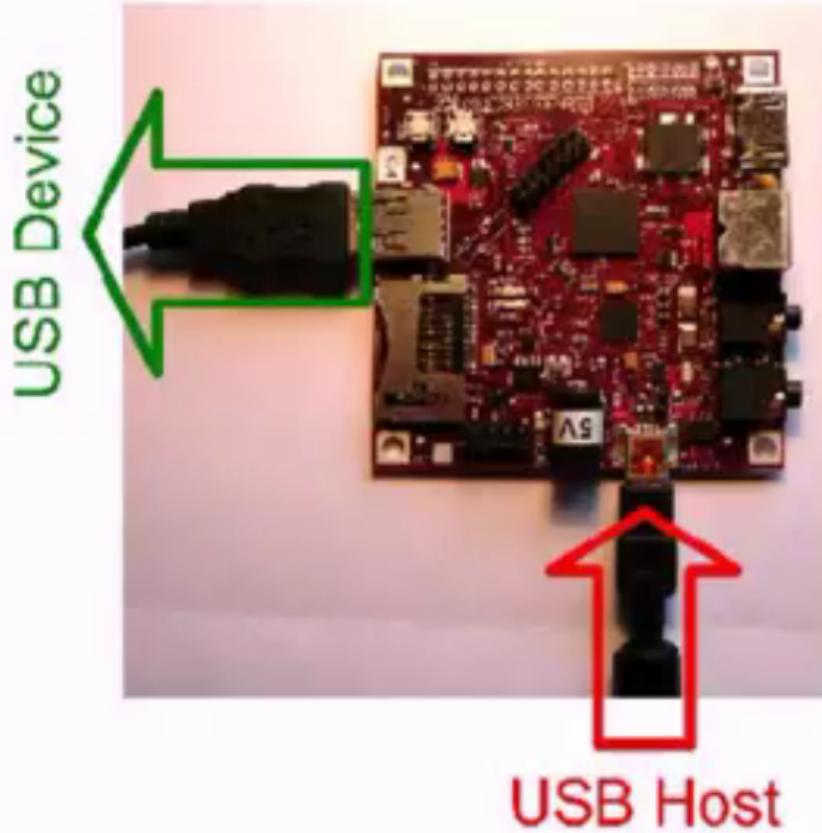
- GSoC 2010 project by **Nicolas Boichat**
- Use the BeagleBoard as an USB sniffer
- Code:
  - Gitorious repo: [beagleboard-usbsniffer-kernel](#)
  - GitHub mirror: [beagleboard-usbsniffer-kernel](#)
- Blog: [beagleboard-usbsniffer.blogspot.com](#)
- Video: [USB sniffer on Beagle Board](#)



# USB Sniffer



## Project goal



- Connect an **USB host** to the slave port,
- and an **USB Device** to the master port.
- **Forward** USB traffic, and **log** it.
- **Transparent** for both the host and the device.

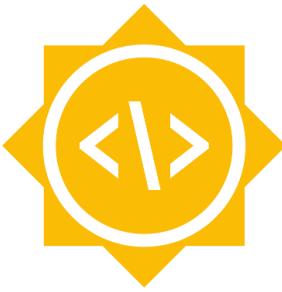


# USB Sniffer



## Implementation

- Linux “proxy” driver:
  - **Gadget** driver (slave side)
  - **Device** driver (host side)
- **Pretend** to be the device, on the gadget side
  - Forward **control** requests
  - According to the USB **descriptors**: connect USB **endpoints**, and forward **data**.

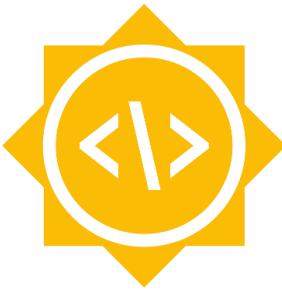


# Userspace Arduino



- GSoC 2013 project by students Anuj Deshpande and Parav Nagarsheth
- Provide a Linux Userspace environment for compiling Arduino style wiring/process sketches to run under Linux





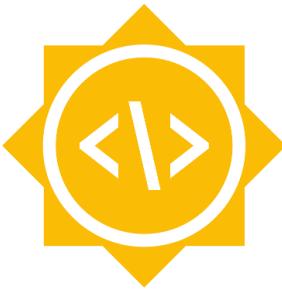
# Userspace Arduino



- eLinux wiki: [Userspace\\_Arduino](#)
- GitHub repo: [Userspace-Arduino](#)
- Example: [Userspace Arduino:BlinkUserspace](#)
- Video: [Using Userspace-Arduino libraries](#)
- Screencast: [Running Arduino Sketches on BBB](#)

BeagleBone Black : Using Userspace-Arduino libraries

```
File Edit View Search Terminal Help
File Options Buffers Tools Help
15
16void loop() {
17
18int i=0;
19
20
21for (i = 0 ; i < 41 ; i++ ) {
22    analogWrite(3,(i*500));
23    delay(50);
24}
25
26for (i = 40 ; i > -1 ; i-- ) {
27    analogWrite(3,(i*500));
28    delay(50);
```



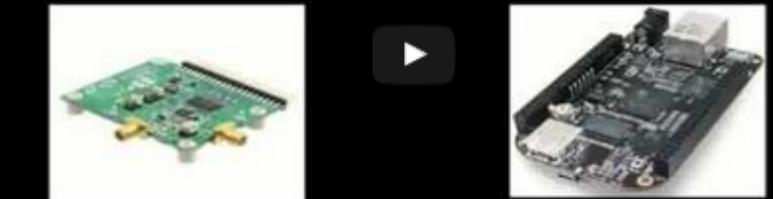
# GSOC 2016 projects



- BeagleScope
- Student: **Zubeen Tolani**
- Mentors: SJLC, Abhishek Kumar, Michael Welling, Hunyue Yau

Introduction BeagleScope project - GSo...

## Hardwares used:



DC782A-P-ND      BeagleBone Black

Introduction BeagleScope



Google Summer of Code

# BEAGLESCOPE

Developer : Zubeen Tolani  
[zeekhuge@gmail.com](mailto:zeekhuge@gmail.com)

**Empower your beaglebone**

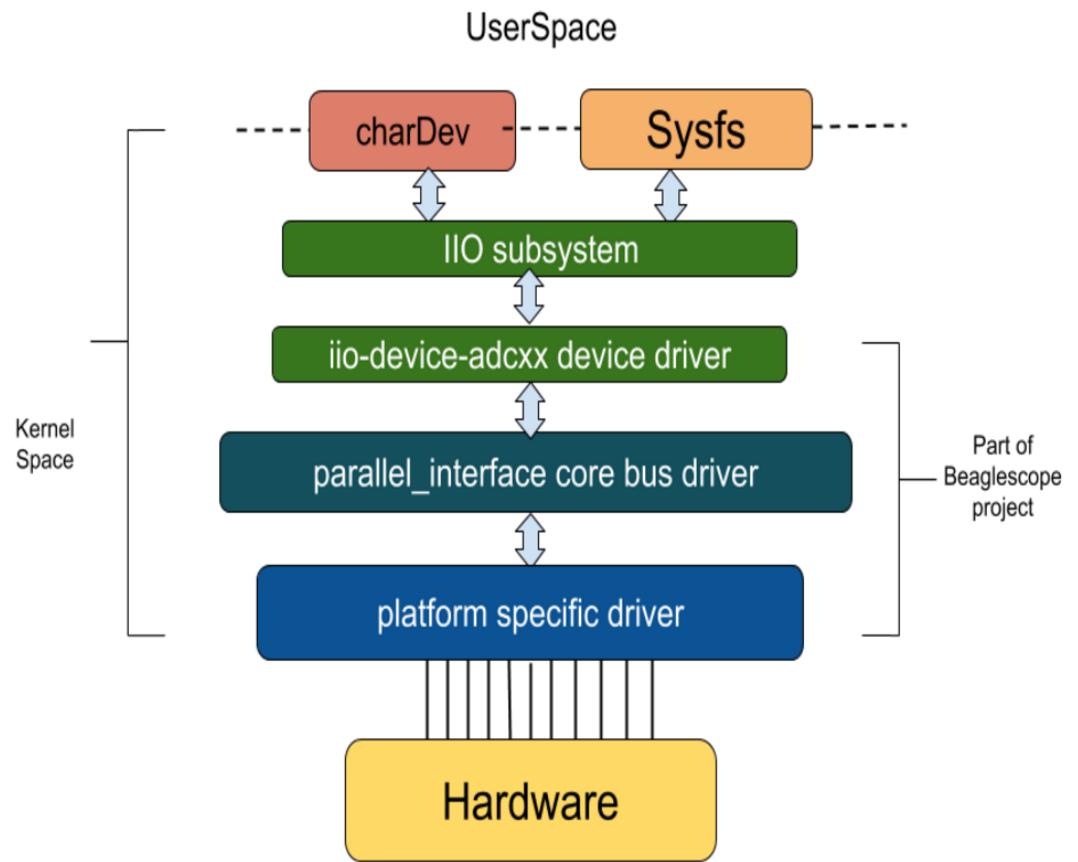
# SUMMARY

- Bootstrapped as a GSoC-2016 project under **BeagleBoard.org**.
- Uses the 2 SoC Programmable realtime units (PRUs) to interface a IIO device using parallel data interface.
- The software adds a sort of 13 bit wide parallel data bus to the board with a few limitations.
- Uses well known IIO subsystem to provide user interface.
- Modular and generic software stack so as to provide highest level of customization and further development specially to kernel hackers.
- The software stack developed can be used for applications like: Oscilloscope, Ultrasound scanners, Software defined radios etc.

# OVERALL WORKING

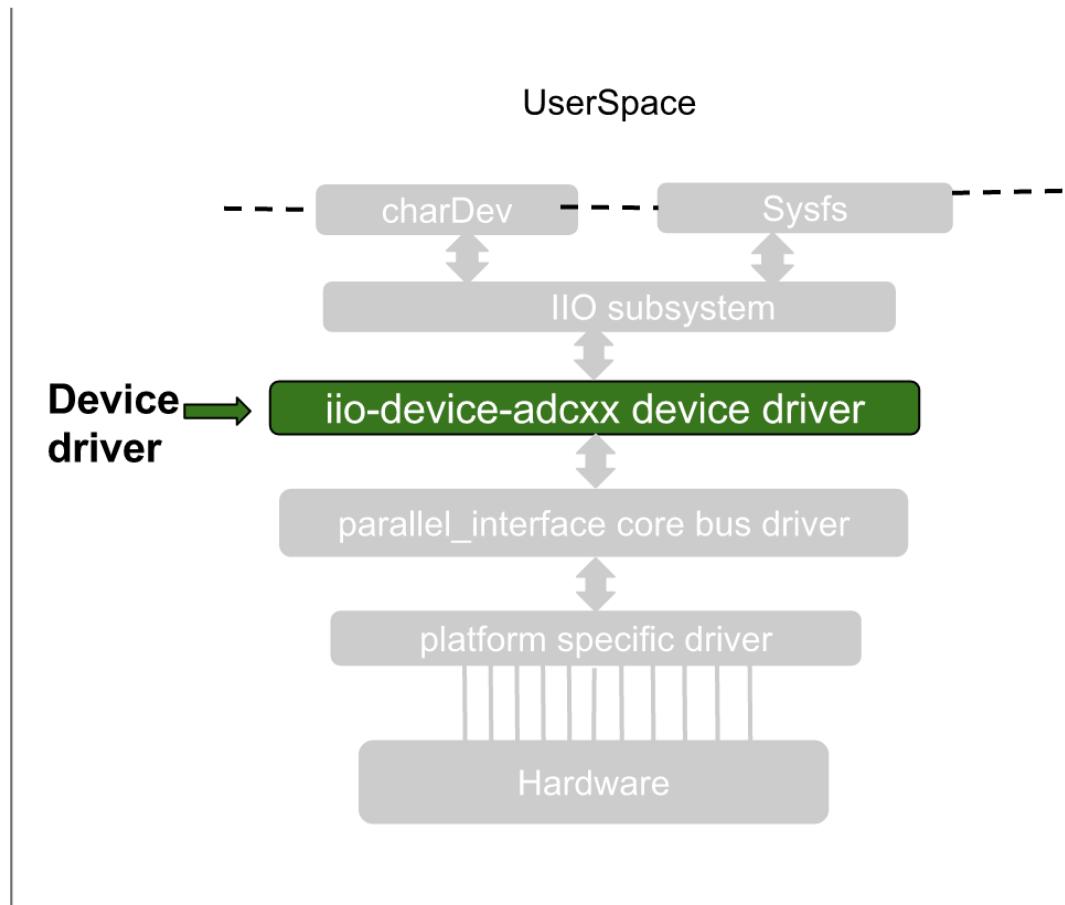
The overall working is divided into 3 parts:

1. The iio-device driver
2. The parallel\_interface bus driver
3. The platform specific driver.



# FOR USERS:

- The users need to worry about just one thing :  
**“If there is a device driver for the device they want to use ?”**
- The beagle-scope project also aims to develop device driver for the DC782A-P adc board.



# HOW TO USE

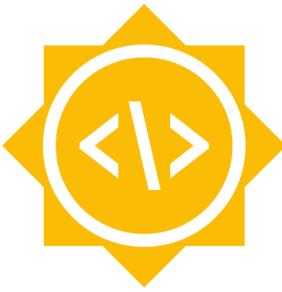
- Just as you would use an IIO device. Some example and docs can be found:
  - [https://kernelnewbies.org/IIO\\_tasks](https://kernelnewbies.org/IIO_tasks)
  - <http://www.at91.com/linux4sam/bin/view/Linux4SAM/IioAdcDriver>
  - <https://wiki.analog.com/software/linux/docs/iio/iio>

# IT SUPPORTS :

- Supports reading raw data.
- Supports buffered capture using IIO buffers.

# RELEVANT HYPERLINKS

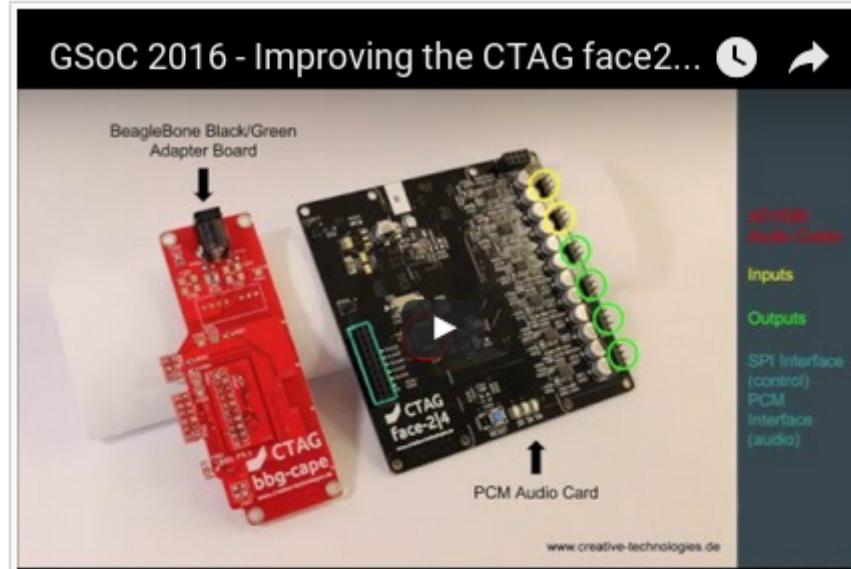
- Project Source : <https://github.com/ZeekHuge/BeagleScope>
- Project Wiki : <https://github.com/ZeekHuge/BeagleScope/wiki>
- Blog : <https://www.zeekhuge.me>
- GSoC project :  
[https://summerofcode.withgoogle.com/projects/#53919754989076  
48](https://summerofcode.withgoogle.com/projects/#5391975498907648)
- Organization : <https://beagleboard.org>
- Developer's Contact : email at zeekhuge@gmail.com



# GSOC 2016 projects



- BeagleBoard X15 multichannel sound driver
- Student: **Henrik Langer**
- Mentors: Robert Manzke, Vladimir Pantelic
- Wiki for libdsp-x15
- Slides from project presentation



# Google Summer of Code

# BeagleBoard.org

...

Henrik Langer  
Kiel University of Applied Sciences

# Overview

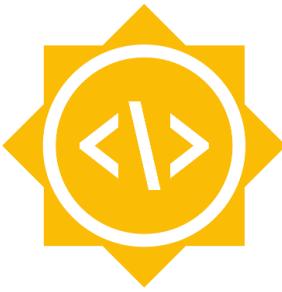
- CTAG face2|4 is a multichannel I2S sound card with 4 stereo inputs and 8 stereo outputs based on AD1938 audio codec and designed for embedded music applications
- Music applications (e.g. audio effects, synthesizers) typically require low latencies and complex signal operations (like Fast-Fourier-Transform)
- Before GSoC 2016 the sound card could only be used with a BeagleBone Black/Green, which doesn't offer these requirements whereas the new BeagleBoard-X15 is based on AM5728 SoC with two cores and two integrated C66x DSPs.

# Project Part 1: Porting Audio Card Drivers to BB-X15

- To use the CTAG face2|4 audio card with the BeagleBoard-X15 the drivers had to be ported:
  - Modifications in ASoC-Machine Driver (glues together digital audio interface of CPU and audio codec)
    - Modifications of clock configurations (new device tree property “bb-device” has been introduced to distinguish between BBB/BBG and BB-X15)
  - New device tree for configuration of required peripherals (e.g. SPI, McASP)
- Problems with remoteproc framework
  - Crash of DSP firmware occurred when sound card is triggered (i.e. audio is played / captured)  
=> Kernel crash
  - Turned out that kernel crash is caused by remoteproc recovery of DSP firmware  
=> Crash could be avoided by disabling recovery of DSP firmware (DSPs still work)

# Project Part 2: Create library for C66x DSPs

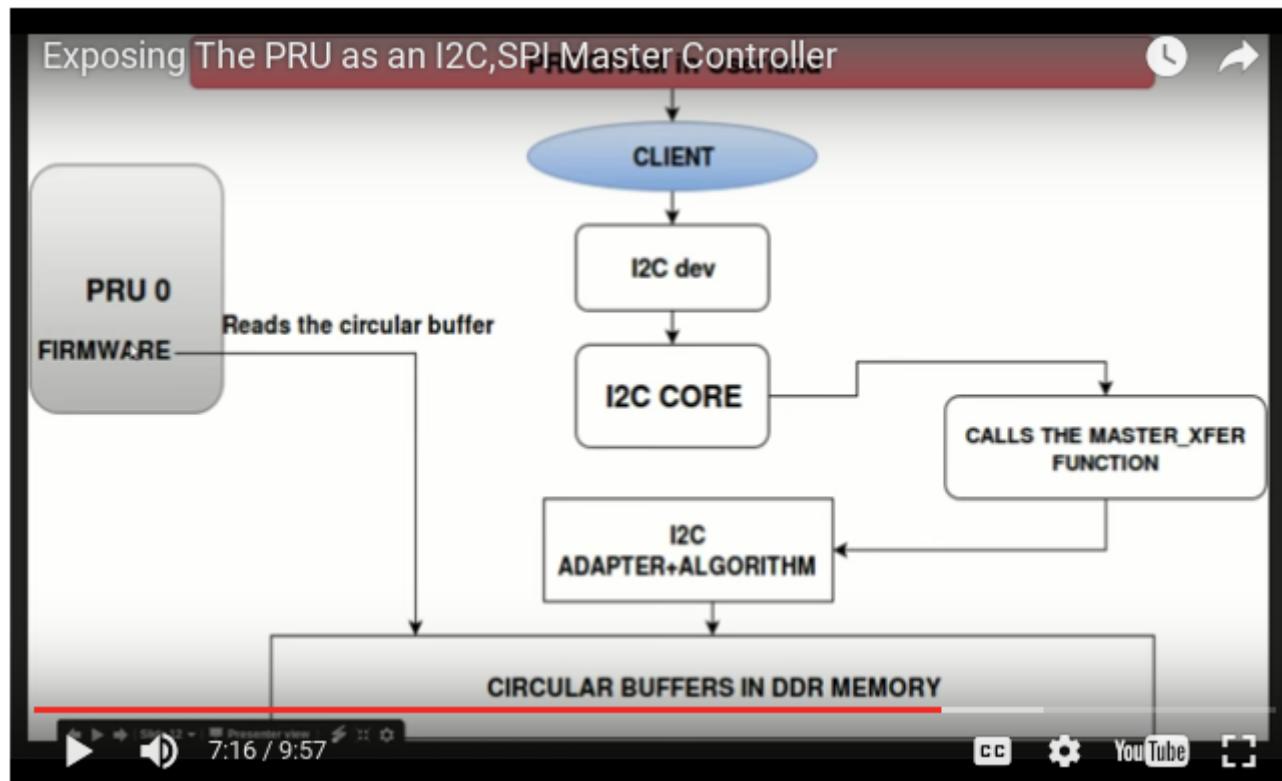
- To simplify usage of C66x DSPs for audio applications a user-space library “libdsp-x15” has been created
- libdsp-x15 offers the following signal operations:
  - Fast Fourier Transformation (FFT)
  - Inverse Fast Fourier Transformation (IFFT)
  - Biquad Filter (often used in audio applications)
- libdsp-x15 uses OpenCL to offload signal operations to DSPs via remoteproc
  - Currently OpenCL is only available in 4.1 kernel, therefore sound card drivers have been ported on 4.1 as well (port to 4.4 is available soon as well)
- Demo
  - For demonstration purposes of signal operations a real time spectrum plot application based on JACK and SDL using FFT has been created

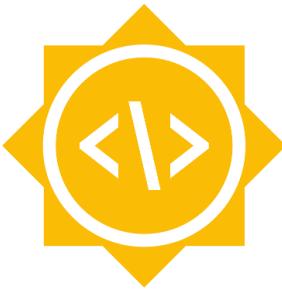


# GSOC 2016 projects



- Exposing the PRU as I2C & SPI master
- Student: **Vaibhav Choudhary**
- Mentors: Andrew Bradford, Matt Porter

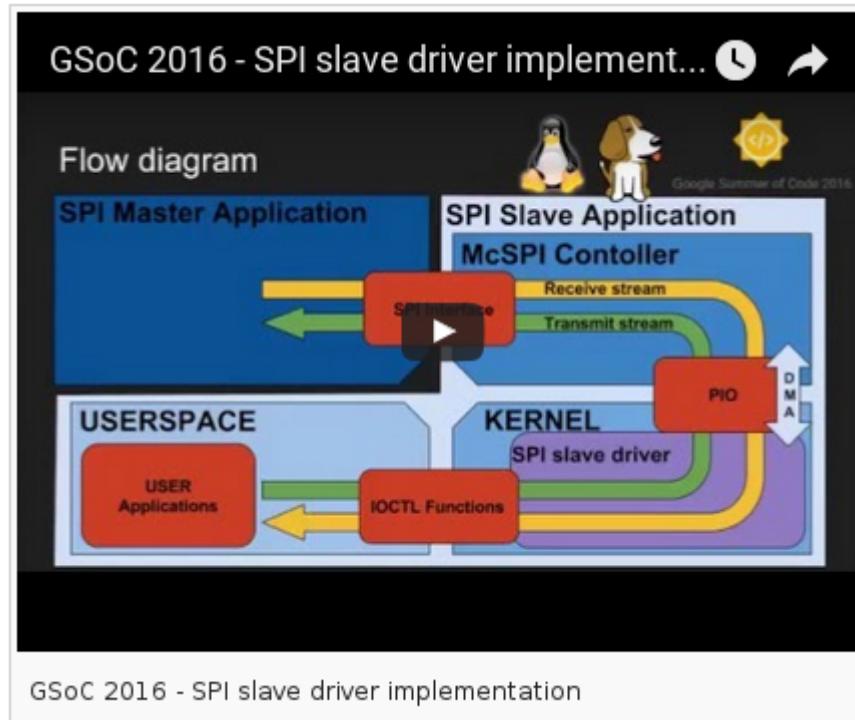


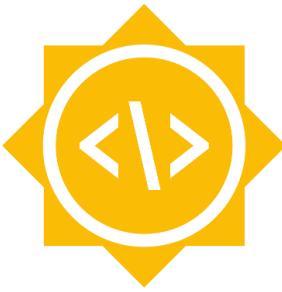


# GSOC 2016 projects



- SPI slave driver implementation
- Student: **Patryk Męzydło**
- Mentors: Michael Welling, Andrew Bradford, Matt Porter





# GSOC 2016 projects



- API support for Beaglebone Blue
- Student: **Kiran Kumar Lekkala**
- Mentors: Alex Hiam, Micheal Welling, Kumar Abhishek, Deepak Karki
- Website / Source Code / Wiki / Documentation



# Students: Apply March 30th!



Google  
Summer of Code

*e-mail:* drew@beagleboard.org  
*twitter:* @pdp7 / @beagleboardorg

 **beagleboard.org**