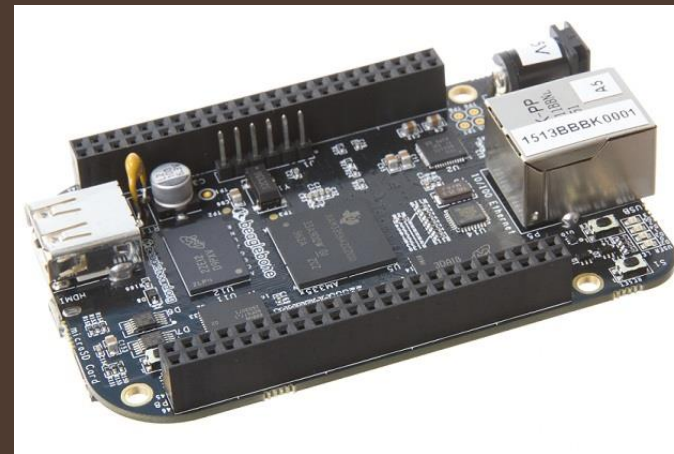


# Introduction to BeagleBone

Intern Sebastian Arrazate Martinez

## Software

- The Beaglebone Black is a free hardware and software microcomputer compatible with most Linux-based operating systems. It began with the Ångström Linux operating system pre-installed on its 2 GB of eMMC flash memory, although as of revision C this changed. This revision (the most current at the date of creation of this document) incorporates 4 GB of eMMC in which, as a novelty, it has the S.O. Debian Linux, due to user requests for this board.



# BoneScript Library

BoneScript is a Node.js library specifically optimized for the Beagle family and featuring familiar Arduino function calls, exported to the browser.

It contains some very useful functions to interact with the hardware of the board in a very simple way, quite similar to how it would be done in the Arduino world.

It can be used directly due to the Node interpreter that Cloud9 invokes or through calls to a remote procedure using the web server installed at the BBB.

# JavaScript Library

Performing physical computing tasks in JavaScript is a rather different than C on microcontrollers. JavaScript and the Node.js interpreter like to do everything asynchronously using callbacks. An event loop runs waiting on whatever the next system-blocking event is, such as waiting for a keypress or a file load to complete. The callbacks are then executed to completion before other event handlers are run.

# Libraries

- The BoneScript Library runs in Node.JS. You can run it directly on the board using the 'node' interpreter or the Cloud9 IDE that invokes the 'node' interpreter. You can also run it using the [bonescript.js](#) script within your browser via remote procedure calls using Socket.io and served up by the web server running on your Beagle.
- Access to the library functions is provided through the "require('bonescript')" function call. The call returns an object containing all of the functions and constants exported by the library. The Node.JS API documentation on [modules](#) provides more information on the usage of 'require' within the 'node' interpreter.

# Functions BoneScript

- `getPlatform()`
- `pinMode()`
- `getPinMode()`
- `digitalWrite()`
- `digitalRead()`
- `shiftOut()`
- `analogWrite()`
- `analogRead()`
- `attachInterrupt()`
- `detachInterrupt()`
- `readTextFile()`
- `writeTextFile()`

# Functions JavaScript

- `console()`
- `setTimeout()`
- `clearTimeout()`
- `setInterval()`
- `clearInterval()`
- `typeof` operator
- `require()`

<https://github.com/yigityuce/BlackLib>

# BeagleBone BlackLib

- BlackLib library is written for controlling Beaglebone Black's feature. It takes power from C++ language. It is created for reading analog input, generating pwm signal, using gpio pins, and communicating with other devices over uart, spi and i2c. In addition to them, it includes debugging feature. So, you can check errors after call any function in the library. It also takes parallel programming, mutex usability, realization of directory operation and realization of time operation ability with the last update(BlackLib v3.0).



# BLACK LIB

Beaglebone Black C++ Library

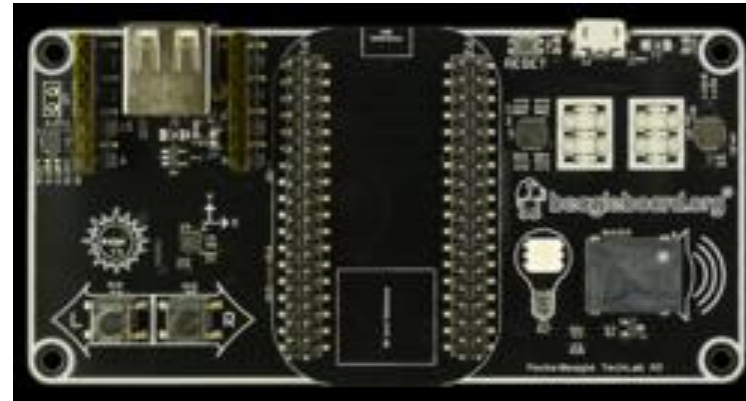


# Capes

Capes are daughter-board add-on products for BeagleBone<sup>®</sup>-family and PocketBeagle<sup>®</sup>-family products. Each extends the functionality of your BeagleBone or PocketBeagle for new exciting capabilities. Hundreds of cape designs are available in the community and now there is also a line of BeagleBoard.org<sup>®</sup> capes readily available from numerous world-wide distributors to help you achieve your learning and design goals.

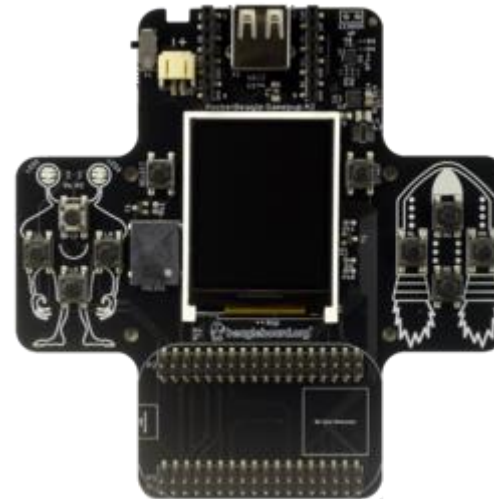
# PocketBeagle TechLab Cape

- For getting your first introduction to programming, Linux, or hacking the Linux kernel itself.



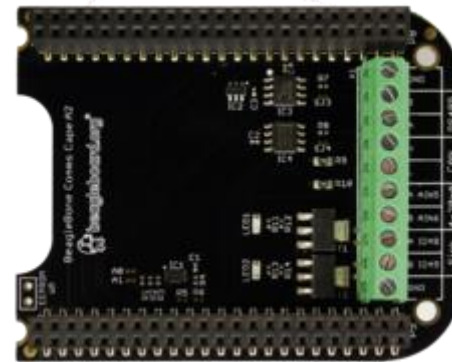
# PocketBeagle GamePup Cape

- For making a handheld arcade emulator, your own games or otherwise taking your PocketBeagle on-the-go.



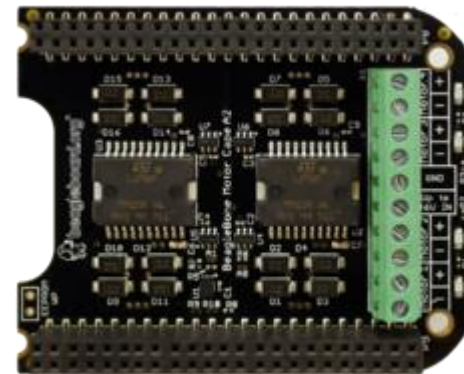
# BeagleBone Comms Cape

- For industrial communication applications. Provides an array of communication protocols including one RS485, one CAN, two analog 4–20 mA current loops, and two 3A 50V interfaces allowing the control of high current loads.



# BeagleBone Motor Cape

- Drive DC motors with direction and PWM control.



# BeagleBone Relay Cape

- Quick and easy relay solution for your home or other automation needs.



## BeagleBone Robotics Cape

- Everything needed for mobile robotics. Loaded with innovative features and a comprehensive software library designed to effortlessly take your robotics concepts from design to reality. A huge array of on-board sensors and controllers and even more expansion options gives you everything you need for your robotics project.

