Iowa Research Online
The University of Iowa's Institutional Repository

University of Iowa
Iowa Research Online

2014 ASEE North Midwest Section Conference

Design Classroom Innovations

Oct 17th, 10:09 AM - 10:27 AM

The Use of BeagleBone Black Board in Engineering Design and Development

Nannan He
Minnesota State University - Mankato

Han-Way Huang
Minnesota State University - Mankato

Brian David Woltman
Minnesota State University - Mankato

Follow this and additional works at: https://ir.uiowa.edu/aseenmw2014

Part of the Educational Methods Commons, and the Engineering Education Commons

The Use of BeagleBone Black Board in Engineering Design and Development

Nannan He, Han-Way Huang, Brian David Woltman
Department of Electrical, Computer Engineering and Technology
Minnesota State University, Mankato, MN 56001

Abstract

The BeagleBone Black (BBB) board is a low cost, open hardware and expandable computer launched by a community of developers sponsored by Texas Instruments. The size of the board is small enough to fit in a mint tin box. It can be used for a variety of projects from high school fair projects to prototypes of very complex real-world embedded systems. This paper presents our work of employing the BBB board in designing embedded systems, and compares it with Raspberry Pi and Arduino in educating engineering and technology students. Our primary experience demonstrates that this board is an easy-to-use and cost-effective kit which can be employed by college-level students for designing their capstone projects.

Introduction

BBB is the newest product in the Beagle family. This board features a powerful TI Sitara™ ARM Cortex™-A8 processor which runs at 1 GHz. And a 2 GB on-board flash memory acts as the "hard drive" for the board to host a Linux operating system and other software development tools. With a user-friendly, browser-based Bonescript programming environment called Cloud9, a learner can easily program the BBB board to rapidly prototype electronic systems that interface with real-world applications. As the knowledge of users develops, the board provides more complicated interfaces including C/C++ functions to access digital and analog pins aboard the ARM Cortex A8 microprocessor. The full power and capability of the BBB board can be programmed in the underlying onboard Linux operating system, such as Angstrom or Ubuntu. Moreover, the Beagle community provides a useful repository of example projects, forums and hardware/software documentation. In this paper, we first introduce the evolution history of Beagle family and overview of unique features of BBB. Then we present our experience of using BBB board for designing engineering senior projects. Finally, the paper is ended with conclusion and future work.

BeagleBone family

The Beagle family currently consists of BeagleBoard, BeagleBoard–xM, the original BeagleBone and the new BeagleBone Black. All have been designed by Gerald Coley, a Hardware Applications Engineer from TI responsible for hardware design and manufacturing issues. Gerald temporarily named his new designs as "Beagle", as a movie "underdog" was

showed in local theaters during the early Beagle line development period. Recently, an acronym highlights the important Beagle features: **B**ring your own peripherals, **E**ntry level costs, **A**RM Cortex-A8 Superscalar processor, **G**raphics accelerated, **L**inux and open source community, **E**nvironment for innovators. Jason Kridner, another Software Applications Engineer also from TI is responsible for Beagle software. The BeagleBoard community with the support from TI ensures the BeagleBone project remains sustained. Its most important members are the users of the Beagle family. The goal of this community is for the entire community to openly share their application successes and encourage the next generation of STEM practitioners.

The objective of the entire platform line is to provide users a powerful computer at a low cost with few restrictions. It is also expected to provide users the open source hardware details so as to continually improve the product line by keeping the line fresh with new innovations. BeagleBone variants have a powerful 32-bit, super-scalar ARM Cortex-A8 processor. The combination of powerful computing power and small physical size allows them to be used in wide variety of projects. Moreover, a variety of peripheral components called "Caps" are currently available to be easily connected to BeagleBone. The caps can extend the capabilities of processor for the rapid prototyping of complex, expandable embedded systems.

Processor	AM3358; **1GHz** – USB powered or DC powered
Memory	512 MB DDR3 SDRAM; **2GB eMMC Flash**
Power options	USB connection, 5 VDC external jack
Board features	**HDMI with audio;** USB, 10/100 Ethernet; serial debug via external header
Processor Subsystems	176K ROM; 64K RAM; 3D graphics engine; LCD and touchscreen controller; PRU-ICSS; Real Time Clock(RTC); USB ports (2); Ethernet; Controller Area Network; UART (2); McASPs(2); McSPI (2); I2C(2); Analog-to-digital converter; Enhanced Capture Module (3); Pulse width modulation (3); Crypto accelerator;

Table 1. BBB features summary

The original (white) BeagleBone was released in late 2011. It is equipped with a Sitara*TM* AM3359 ARM® processor that operates at a maximum frequency of 720 MHz when powered from 5 VDC source. It also features an SD/MMC micro SD card, which acts as the "hard drive" and hosts the Ångstrom distribution Linux, Cloud9 IDE and Bonescript. BBB was released in early 2013, which is physically similar to the original BeagleBone, but has three major enhancements highlighted with the bold font in Table 1. Firstly, it hosts the SitaraTM AM3358 ARM® processor which operates at a higher frequency - 1 GHz at the maximum, no matter with the USB powered or 5VDC powered. Secondly, a 2 Giga byte eMMC flash is equipped on BBB to provide the non–volatile mass storage in an IC package. The eMMC is used as the "hard drive" for BBB and hosts the Linux OS, Cloud9 IDE and Bonescript, while the micro SD card connector is still available for expanding the storage. Thirdly, BBB has also been enhanced with

an HDMI micro connector to interface with commercial entertainment products such as DVD players and mobile phones. One thing to note is that BBB actually costs approximately half of the original BeagleBone. The basic features of BBB are summarized in Table 1.

Bonescript library and Cloud9 IDE programming environment

Bonescript is an easy–to–use and open source library to harness the power of BBB. This board hosts the Linux operating system (OS); however, because Bonescript programming is accessed over the pre-installed browser based IDE called Cloud9, it could be developed from any OS, MS Windows, Mac OS X, and Linux. Bonescript consists of a library of event driven functions written in JavaScript to use different systems aboard BBB for quickly developing a variety of electronic systems. A set of services also included in Bonescript runs at boot-up to expose many of the functions to a web browser. A summary of functions is provided in Figure 1. Bonescript shipped with BBB is Version 2.0. As Bonescript is rapidly evolving, it is essential to regularly update the eMMC with the latest software image.
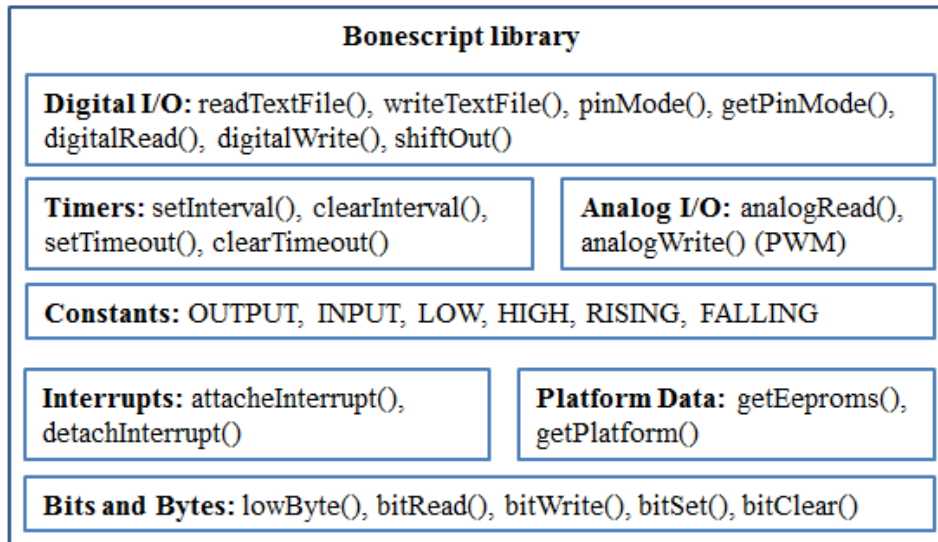

Figure 1. Bonescript library environment

Cloud9 is an open source IDE running on the web platform, which has been developed and maintained by Cloud9 IDE, Inc founded in 2010. This IDE is written almost entirely in JavaScript, and uses Node.js at the back-end. It has different levels of support for more than 40 programming languages, currently supports running application programs written in PHP, Ruby, Python and JavaScript/Node.js by default. Cloud9 IDE builds on *Ace*, which is a free and open source web-based code editor. This editor currently provides over two dozen themes to highlight the syntax of supported languages. It was first released in September 2010, and is actively developed and maintained by both Ajax.org and Mozilla. As the successor of the Mozilla Skywriter, *Ace* can be easily embedded in any web page and JavaScript application. It extends

the characteristics and usability of existing native editors like vim or Eclipse, Any updates like adding new themes and new languages to support in *Ace* can be reflected in Cloud9 IDE.

Collaborative development is another important feature of this IDE. Developers can share their projects by simply copying the project URL in the browser and sending it to others. Other ways to share accesses are also available, such as via Facebook or Twitter. Developers can either grant others to change files in the workspace in real-time with various authorities, or just to view the workspace. When a collaborator changes a file a user is also working on, Cloud9 detects it and asks you whether you would adopt their changes. A three-way-merge algorithm is applied to support an option of merging changes without incurring conflicts. While collaborating, all developers can also discuss with each other using a built-in chat room. For JavaScript code, Cloud9 supports the language analysis by spotting undeclared variables, syntax errors, or other typos at the offending lines of code. Thus, coding in Cloud9 is an interactive process with such language checking taking place during the code entry. Other interesting features of this IDE include the code auto-completion, advanced searching and replacing in files using regular expressions, etc.
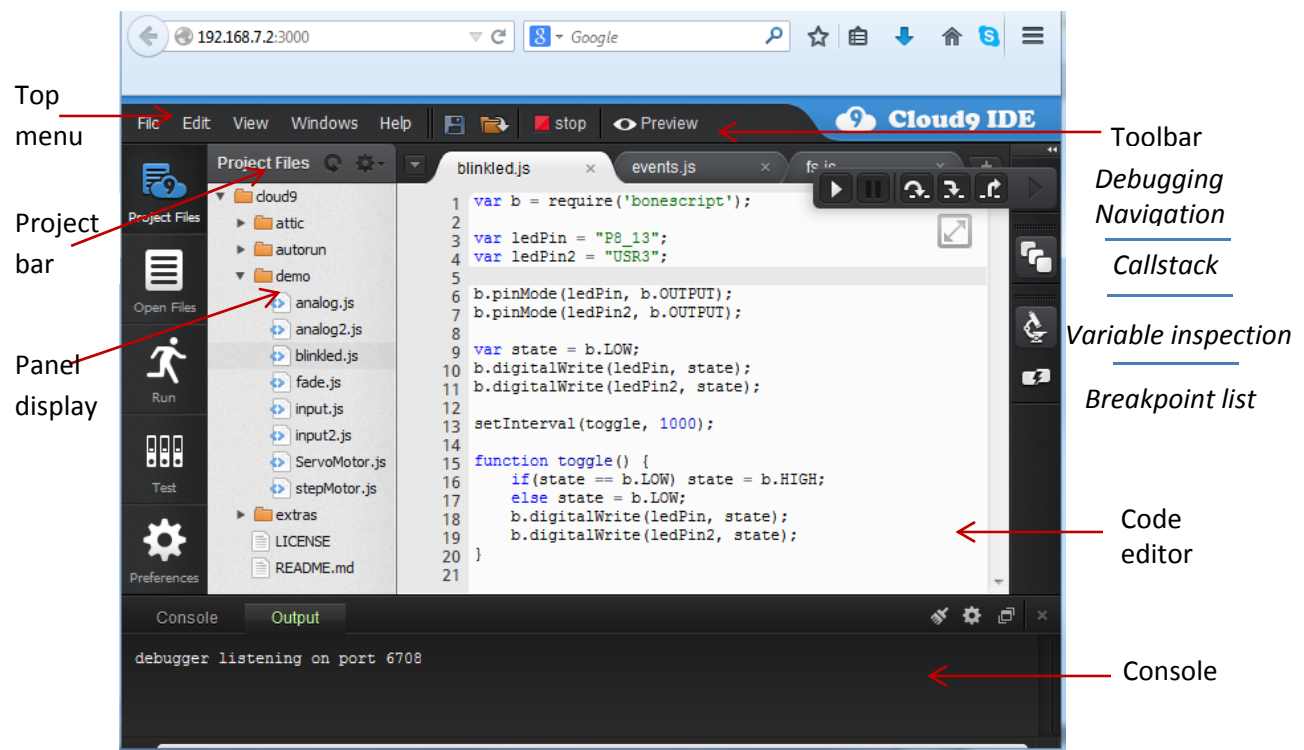

Figure 2. Cloud 9 IDE Window (Debug mode)

To explore the Cloud9 IDE on BBB, users need to navigate to http://192/168.7.2:3000/ after the board is connected and completely set up. Major components of the Cloud9 IDE window are as shown in Figure 2. The Toolbar on the top right of the window shows the available tools when debugging your code. Cloud9 IDE provides two ways to run your code, either through the Run Panel on the left or through the Run button on the Toolbar. Debugging a project is similar to

running the code. The main difference is the expansion of the debugging Toolbar to the right of the editor as shown in Figure 2, which includes *debugging navigation* for moving around the code while debugging, *callstack* which is a structured list of information about the execution sequences of your code, *variable inspection* which shows the type and value of all related variables present in the context, and *breakpoint list* recording all breakpoints applied during debugging.

In summary, Cloud9 IDE, as a browser-based development environment, enables developers to program anywhere in the world and easily collaborate in a team project. Its web-based editor provides the rich set features to facilitate the interactive coding. And it fully supports the breakpoint-based debugging. However, this Cloud9 IDE pre-installed in the eMMC of BBB is a limited version, like not fully supporting Python script. As Bonescript only includes simple functions, this limited version is sufficient for programming BBB. The program *blinked.js* in Figure 2 makes use of three Bonescript library functions *pinMode(), digitalWrite(), and setInterval()* to configure the input/output mode of pins on BBB, assign a logic level on those pins and set a delay time during the program execution respectively. *OUTPUT, LOW, HIGH* are constants defined in the library. The *require()* is a JavaScript() function to load the modules in the file specified by the input parameter, which is the "Bonescript" file.

Use of BBB in teaching the engineering design

Recently, some universities have started creating the teaching materials for the Beagle Bone and BBB. Such education experience is still in the early stage since these boards were initially launched not long ago. Compared with Arduino or Raspberry Pi, the hardware of BBB is not as robust as them, for example BBB needs better power protection than it currently has. Students could easily burn the processor on board when the input current applied to the I/O is higher than the threshold by mistake. And some software programs installed on the board are not stably established or complete. They are still very actively updated with frequent new versions releases. For instance, the Ubuntu Linux image was not available when we conducted the experiments in early 2014. (It has recently became available.)

However, the potential of BBB as the serious competitor of Raspberry Pi or Arduino has been foreseen by educators [2]. On one side, BBB provides an easy-to-use platform to learn not only traditional embedded microcontroller programming using either Bonescript or C but also basic electronics. Thus, BBB can be used as an Arduino board. On the other side, with the 32-bit ARM Cortex A7 processor and 2GB eMMC flash memory, BBB also fully supports teaching high-level computing languages and Operating Systems like Linux computing. Linux support usually offers many advantages over less powerful 8-bit boards, for examples audio and video processing, and advanced peripheral devices control (e.g. touchscreen). In this case, BBB has the potential to compete with Raspberry Pi. There are steadily growing interests in the use of BBB

for the engineering education purpose. As an instance, the embedded Beagle class project has been recently created in Rose-Hulman [3]. Its goal is to create teaching materials for the BeagleBoard that will make it easy for those teaching either 8/16-bit microprocessor classes, or traditional Operating System classes to use the Beagle in their class. In this section, we mainly present our experience of building a robot car project which is a variant of a previous senior project implemented with Raspberry Pi replaced with BBB.

Robot project

This project is about building a robot car carrying BBB and a camera for picture taking. The car moves around to avoid obstacles and sends pictures or live streaming taken by the camera. The on board IR sensors could also send the location sensing data back to the user. The user can control the robot car through the webpage and can watch the picture or live streaming received from the car. The project consists of the hardware and software part. Basic components of each part are listed in Table 2.

Hardware components	Software components
BBB	C++ (processing video stream)
4WD Car	
L298N Dual H Bridge DC Motor Driver IC	Bonescript (DC motor control for car
PlayStation Eye Webcam	movement, IR sensing)
Wi-Fi dongle (Belkin N150)	
Battery bank	Java script (web interface, user
USB Hub	interface)
IR sensor	

Table 2. Hardware and software components

This robot car is a smart car 4WD robot chassis tracing obstacle avoidance car. It has four wheels each of which is connected to a DC motor. When applying a voltage 3v directly to the DC motor, the wheel will rotate with the speed of 100 rpm. (The wheel can reach the maximum speed of 240 rpm while applying 6v.) L298N Dual H-Bridge Motor Driver IC is the driver chip of L298N Dual H-Bridge stepper motor driver controller board module. The output A and B on this module drive the left motor and right motor of the robot car respectively. The input pins are connected to the general purpose I/O pins of BBB for the car movement control. In this project, PlayStation Eye from Sony was adopted as the audio and video source to capture images and video stream. It is capable of capturing 640*480 pixels at 60Hz, and is compatible with the A6 image of Angstrom Linux. Moreover, as this webcam has a built-in microphone which will be useful for future project extension like adding features of the voice recognition and voice control. The WiFi dongle is used to connect to surrounding wireless networks to transmit images back to the center station. Belkin N150 Micro Wireless USB Adapter was applied. It is not natively supported with Angstrom. But the online tutorial is available on how to compile and install the device drivers for this particular USB adapter. A battery bank is used to power BBB. Different with Raspberry pi

board, which has two USB ports, BBB has only one. So a USB hub was needed to connect to battery bank, the WiFi dongle and the USB web camera. Three IR sensors are used to detect the obstacles within certain distance from where the robot car is moving. The total cost is about $140.

The software part of this project makes use of a mix of programming languages for different control tasks. Angstrom A6 Image was installed on BBB in order to use the camera drivers for the playstation eye and drives for the Belkin WiFi dongle. First, Bonescript is adopted to program the GPIO output pins which are connected with the H-bridge motor driver to control the DC motors, and GPIO input pins which are connected with two kinds of sensors. Pin 37, 39 and 40 of Port 9 are configured as input mode to receive signals from right, left and center IR sensors respectively. Pin 13, 14, 15 and 16 of Port 9 are configured as output mode to send control signals to H-bridge driver. The algorithms and implementation details of processing sensor data and control the car moving forward, backward, left and right are omitted in this paper. The source code could be accessed via the research course website [4]. Although this control task could also be realized using the C or C++ programming languages, the Bonescript implementation could be much simpler as the complexity of dealing device tree overlays to configure the usage of GPIO pins is hidden in the Bonescript library functions. The second language used in this project is C/C++. Due to the limitations of the current version of Bonescript environment, the task of processing of video have to be implemented using C/C++ language and openCV libraries supported in the Linux computing. Lastly, Node.js is server side JavaScript environment that uses an asynchronous event-driven model. This enables Node.js to achieve good performance in many Internet applications. In the project, Node.js used as web interface with which user could manually control the movement of car and display the video streaming.

Conclusions

This paper introduces the Beaglebone family, especially BBB. Then its powerful browser-based programming environment Cloud IDE and Bonescript library are described in detail. The comparisons of BBB with Aduino and Raspberry Pi in the engineering education are discussed. A senior project as a case study which demonstrates the representative hardware and programming features of BBB is reported. In the future, we will develop other student projects to explore the potentials of BBB in embedded computing applications.

Bibliography

1.   Cloud9 IDE: https://docs.c9.io/

2. Publitek marketing communications, "BeagleBone Black Brings Arduino-Style Connectivity Simplicity to Embedded Linux", Digi-key Article Library.
3. Beagle class ECE497 in Rose-Hulman: http://elinux.org/Embedded_Beagle_Class
4. Course website: http://mavweb.mnsu.edu/hen/EET310_fall13.html