./

Learning Report

Embedded Linux

|  |  |  |
| --- | --- | --- |
| **Name** | **PS Number** | **Email ID** |
| Mohammed Sadiq Afreed Meeran | 99002449 | mohammed.meeran@ltts.com |

# 

**Table of Contents**

[ACTIVITY 1:SETUP ACTIVITY 4](#__RefHeading___Toc630_3000186962)

[Activity 2: Differences between Raspberry pie , Dragon, imx7 Sabre, BBB 6](#__RefHeading___Toc632_3000186962)

[Activity 3:Evolution and Changes of Beagle back Bone Board 8](#__RefHeading___Toc634_3000186962)

[Activity 4: Pin expansion header of BBB and locate the various peripherals of Bone 10](#__RefHeading___Toc636_3000186962)

[Activity 5: Testing MLO image and U-boot image on BBB 11](#__RefHeading___Toc638_3000186962)

# ACTIVITY 1:SETUP ACTIVITY

**Step by Step Configuration of the boards and set up in Windows OS:**

Step 1: Plug in the Ethernet chord to Beagle Bone Black Ethernet port (to establish communication)

(or) use any other bootalbe options such as SD card, MMC.

Plug in the USB chord to the host machine and the other end to the micro USB port in the

Beagle Bone Black (To supply power to the Beagle Bone Black)

Step 2: Installing Drivers in the Host machine

Visit https://beagleboard.org/getting-started

Depending on the configuration of the host machine download and install the respective USB

Driver Installer. Once installed Reboot the host machine.

Step 3: Connecting to Beagle Bone Black via Ethernet.

Open any Browser (preferably Chrome or Firefox) type the below IP address in the URL

IP address: 192.168.7.2 .

Beagleboadr.org Web page gets loaded which is already present in the Beagle Bone Black

Now the connection to Beagle Bone Black is successful.

Step 4: Obtaining the unique IP address

Click on the Cloud9 IDE a web page loads and select once again Cloud9 IDE this takes to the

Cloud9 IDE which is running on the Beagle Bone Black.

(If any error occurs then make sure don’t use the Internet Explorer Browser)

Open new terminal in the Cloud9 IDE and type the command “ifconfig”.

Note down the IP address in the eth0 section.

Step 5: Connect to the Beagle Bone Black using simple SSH client i.e., PuTTY

Open the URL : https://www.chiark.greenend.org.uk/~sgtatham/putty/latest.html

Download the putty.exe depending on the host machine configuration.

Step 6: Launch the SSH client i.e., PuTTY. Enter the IP address (Noted in the Step 4) in the Host name *IP*

*address text field. Click on Open. This launches a Linux Terminal window in the host machine*

*which asks for the login credential login as: root (which is default).*

Step 7: Remotely connecting to the Beagle Bone Black using TightVNC viewer.

TightVNC server is already installed in the Beagle Bone Black. But TightVNC viewer is

required for the remote PC in order to access remotely. Download the TightVNC viewer using url

link : <https://www.tightvnc.com/download.php>

Download and install the TightVNC viewer based on the configuration on windows.

Step 8: Fire up VNC sever before running the TightVNC viewer in the remote machine.

Launch putty.exe in the Host machine and login. Then install tightvncserver.

Type the “sudo apt-get install tightvncserver” command.

(Note: The below steps are to be performed only once)

Then type command “typevncserver” (Press enter)

Set Password and Verify Password.

Now type command “vncserver :1 -geometry 1280x800 -depth 24 -dpi 96” (Press enter)

Step 9: Launch TightVNC viewer in the remote machine and enter the IP address (Noted in Step 4) along

with :1 Example: 10.1.15.25 : 1

Click on connect and enter VNC password which was set in the Step 8.

A graphical user interface window pops up.

**Step by Step Configuration of the boards and set up in Linux OS:**

Step 1: sudo minicom -s

Step 2: serial port setup (know the TTL cable name)

(In other terminal)

Step 3: dmesg (Search for Prolific Technology) port ttyUSB0

Step 4: Press a and enter /dev/ttyUSB0

Step 5: Press e check for Standard Bod rate : 115200 8N1

8-bit

N-no parity

1-Stop bit

Step 6: Press f, Check for Hardware flow control set it to NO

Step 7: Press g, Check for software flow control set it to NO

Step 8: Save the settings as dfl (default)

Step 9: Exit

Step 10: connect usb

ANGSTROM

Step 11: beaglebone login: root

root@beaglebone:~#

Step 12: Shutdownnow

# Activity 2: Differences between Raspberry pie , Dragon, imx7 Sabre, BBB

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Parameter** | **RASPBERRY PIE** | **BEAGLEBONE BLACK** | **Dragon** | **imx7 Sabre** |
| Processor Type | It uses ARM11 processor. | It uses ARM Cortex-A8 processor. | Quad-core ARM® Cortex® A53 at up to 1.2 GHz per core with both 32-bit and 64-bit supporT | Two Arm Cortex-A7 core OS upto 1 GHz, Single Arm Cortex- M4 CORE os |
| RAM | For the functioning of raspberry pi, 512 MB SDRAM is used. | For the functioning of beaglebone black, 512 MB DDR3L is used. | 1GB LPDDR3 533MHz / 8GB eMMC 4.5 / SD 3.0 (UHS-I) | 1 GB DDR3, 533 MHz  eMMC expansion footprint. |
| Processor Speed | It uses 700 MHz for processing. | It uses 1 GHz for its processing. | 1.2 GHz per core with both 32-bit and 64-bit support | 1 GHz :Arm Cortex-A7  200 MHz :Arm Cortex -M4 |
| Min Power | It requires a power supply of 700mA (3.5W). | It requires min power of 210mA (1.05W) for its functioning. | It requires a power supply of 8-18V 2A. | It requires 5V/5A universal power supply. |
| GPIO Pins | It has 12 GPIO pins. | It has 69 GPIO pins. | It has 40 GPIO pins | It has 138 GPIO pins |
| Dev IDE | It uses IDLE, Scratch, Squeak/Linux to perform tasks. | It uses Python, Scratch, Squeak, Cloud9/Linux to perform a particular task. | Android 5.1 (Lollipop) on Linux Kernel 3.10 | Vivante Tool v6.2.4.p4.1.7.9 and linux based systems |
| USB Master | It has 2 USB 2.0 on board. | It has 1 USB 2.0 on its board. | one micro USB (device mode only), two USB 2.0 (host mode only) | * 1 USB host connector * 1 micro USB OTG connector |
| Audio Output | Supports HDMI, Analog audio output | It uses Analog output for audio. | It calls for a minimum of single channel audio through two interfaces, BT and HDMI/MHL/DisplayPort | i.MX7 has multiple audio interfaces and one is fully available on the SODIMM connector of the Colibri iMX7. |
| Video Output | It supports HDMI, Composite output for video. | No such specific video output. | 1080p@30fps HD video playback and capture with H.264 (AVC), and 720p playback with H.265 (HEVC) | Its supports HDMI, composite output for video. |
| UART | It uses 1 UART to transmit and receive serial data. | It uses 5 UART to transmit and receive serial data. | support for one SoC UART and an optional second UART both to be routed to the Low Speed Expansion Connector. | UART via USB port |
| No. of I/O pins | It has 8 Digital, 0 Analog pins. | It has 65 Digital, 7 Analog pins. | It has 11 Digital ,0 Analog pins. | It has total 138 pins |

# Activity 3:Evolution and Changes of Beagle back Bone Board

|  |  |
| --- | --- |
| **Revision** | **Additions(differences)** |
| A4 | Preliminary |
| A4A | Incorporated the capacitors to fix the noise issue on the display |
| A4B | Added a 100K pull down resistor between pins 1 and 4 of J1 to fix the serial port issue |
| A5.1 | 1.Added information on Power button and the battery access  points  2.Final production released version. |
| A5.2 | 1) Updated the PCB to incorporate the modification that was being done on Rev A5A. There is NO difference at all in functionality between REV A5A and REV A5B. 2) Made the LEDs dimmer for those that could not sleep due to the brightness of the LEDs. |
| A5.3 | 1. Updated serial number locations.  2. Corrected the feature table for 4 UARTS  3. Corrected eMMC pin table to match other tables in the manual. |
| A5.4 | 1. Corrected revision listed in section 2. Rev A5A is the initial  production release.  2. Added all the locations of the serial numbers  3. Made additions to the compatibility list.  4. Corrected Table 7 for LED GPIO pins.  5. Fixed several typos.  6. Added some additional information about LDOs and Step-Down converters.  7. Added short section on HDMI. |
| A5.5 | 1. Release of the A5B version.  2. The LEDS were dimmed by changing the resistors.  3. The serial termination mode was incorporated into the PCB. |
| A5.6 | 1. Added information on Rev A5C  2. Added PRU/ICSS options to tables for P8 and P9.  3. Added section on USB Host  4. Correct modes on Table 15.  5. Fixed a few typos |
| A5.7 | 1. Updated assembly revision to A6.  2. PCB change to add buffer to the reset line and ground the oscillator GND pin.  3. Added resistor on PCB for connection of OSC\_GND to board GND. |
| A6 | 1. Added changes for rev A6 that covered fixing of the link LED, JTAG Reset, and DHCP issue.  2. Added PRU information and two additional signals for the PRU.  3. Added write protection to EEPROM.  4. Fixed numbering of subsections in Section 7.0  5.Fixed error in Table 9 pin 23Mode 1 should be MMC1\_DAT4.  6. Updated Table 7 to show the revision number in the EEPROM matches the revision of the board.  7. Corrected various typos.  8. Updated Battery Interface section to accurately document the LDO dropout at 200mV.  9. Added SW Support section. |
| A6A | 1) Added optional zero ohm resistor to tie GND\_OSC1 to system ground. 2) Changed C106 to a 1uF capacitor. 3) Changed C24 to a 2.2uF capacitor. |
| B | 1.Changed the processor to the AM3358BZCZ  2.No changes in features or operation of the board resulted from this change. |
| C | 1.This revision increased the eMMC from 2GB to 4GB |

# Activity 4: Pin expansion header of BBB and locate the various peripherals of Bone

<https://github.com/99002449/Embedded_Linux/tree/main/Pin_Extension_header_of_BBB>

the above link has been updated with the Pin expansion header of BeagleBone Black. Which has been P8 and ref excel datasheet.

Types of Pins:

1. 23 – Reconfigurable Digital pins
2. 7 -- Analog Inputs Pins
3. 2 -- Shared I2C Pins
4. 7 -- Pulse width modulation
5. 25 – Digital Pins
6. 32 – Power management Pins. Testing MLO image on BBB and Testing U-boot image on BBB. Explain each and every section of stage in the learning report.

# Activity 5: **Testing MLO image and U-boot image on BBB**

Step 1: Update Linux:

- sudo apt update

- sudo apt upgrade

Step 2: Install GParted

- sudo apt-get install gparted

Step 3: Making partition in the SD Card

check SD card name in the terminal using - dmesg (Eg: Sdb)

Step 4: Select SD card name (Sdb) in the gparted application.

Delete all the folders in the SD card and make it unallocated area

Right click on unallocated area and select new option

Step 5: Select File System and select FAT 16

In the Label area name it as BOOT

Click on Add

Step 6: Right Click on unallocated area and select new

Select File System and select ext3

In the Label area name it as ROOTFS

Click on Add

Now the SD card has been divided into two parts

(fat16) BOOT

(ext3) ROOTFS

Step 7: After partition Format it (Select tick mark in the menu bar).

Right click on BOOT partition and Select Managed Flags

Select option BOOT

Step 8: In the File Manager check whether the SD card is partitioned into

BOOT and ROOTFS

Note down the location of BOOT (eg. usr/admin/....)

Copy and paste the MLO file in the BOOT partition of SD Card

Eject the SD card and mount it in the BBB board

**Testing MLO image**

Step 9: Open terminal type sudo minicom

Connect serial port

While Pressing the S2 (Boot button on BBB) give power supply to

BBB using micro USB cable

In the Terminal Below message appears:

U-Boot SPL 2017.05-rc2 (May .............)

Trying to boot from MMC1

\*\*\* Warning bad CRC, using default environment (it means it

didn't boot from eMMC)

reading u-boot.img

spl\_load\_image\_fat: error reading in u-boot.img err --1

\*\*\* Warning bad CRC, using default environment

Failed to mount ext2 file system.......

spl\_load\_image\_ext: ext4fs mount err – 0

**Testing U-Boot image**

Step 10: Now copy U-Boot.img and paste it into the BOOT partition of SD Card

Insert the card into BBB

In **the** Terminal Below message appears:

...

...

reading u-boot.img

reading u-boot.img

U-Boot 2013.04-dirty (jun .......)

MMC: OMAP SD/MMC

\*\*\* Warning - readenv() failed default environment

...

...

# **Activity 5: Commands and Environmental Variables in U-Boot area**

1. => mmcinfo

Device: OMAP SD/MMC

Manufacturer ID: 0

OEM: 3000

Name: APPSD

Tran Speed: 50000000

Rd Block Len: 512

SD version 2.0

High Capacity: Yes

Capacity: 15 GiB

Bus Width: 4-bit

Erase Group Size: 512 Bytes

2. => base

Base Address: 0x00000000

3. =>version

U-Boot 2017.05-rc2 (May 02 2017 - 08:53:40 +0530)

arm-linux-gnueabihf-gcc (Linaro GCC 6.3-2017.02) 6.3.1 20170109

GNU ld (Linaro\_Binutils-2017.02) 2.27.0.20161019

4. => fatinfo mmc 0:1

Interface: MMC

Device 0: Vendor: Man 000000 Snr 80000601 Rev: 1.2 Prod: APPSD

Type: Removable Hard Disk

Capacity: 15360.0 MB = 15.0 GB (31457280 x 512)

Filesystem: FAT16 "BOOT "

5. => fatls mmc 0:1

72320 mlo

367496 u-boot.img

370 uenv.txt

3 file(s), 0 dir(s)

6. => echo "message"

message

=> echo message2

message2

7. => sleep 5

sleeps for 5 seconds

8. => ls <interface> [<dev[:part]> [directory]]

- List files in directory 'directory' of partition 'part' on

device type 'interface' instance 'dev'.

=> ls mmc 0

72320 mlo

367496 u-boot.img

370 uenv.txt

3 file(s), 0 dir(s)