BeagleLogic Documentation

Release 2.0

Kumar Abhishek

Oct 08, 2017

Contents

beaglelogic

beaglebone logic analyzer

Thank you for choosing BeagleLogic! BeagleLogic is a 100Msps logic analyzer that runs on the BeagleBone including BeagleBone Black, Green and Green Wireless.

The core of the logic analyzer is the 'beaglelogic' kernel module that reserves memory for and drives the two Programmable Real-Time Units (PRU) via the remoteproc interface wherein the PRU directly writes logic samples to the System Memory (DDR RAM) at the configured sample rate one-shot or continuously without intervention from the ARM core.

BeagleLogic can be used stand-alone for doing binary captures without any special client software like this:

```
dd if=/dev/beaglelogic of=mydump bs=1M count=1
```

The above commands grab a binary dump at the specified samplerate using the sysfs attributes.

When used in conjunction with the sigrok library, BeagleLogic supports software triggers and decoding over 70 different digital protocols. Since the BeagleLogic bindings for libsigrok have been merged upstream, the latest built-from-source sigrok libraries and tools support capturing from BeagleLogic.

BeagleLogic also offers a web interface (demo) which, once installed on the BeagleBone, can be accessed from port 4000 and can be used for low-volume captures (upto 3K samples). It makes BeagleLogic a useful tool for beginners as a learning tool about digital protocols.

CHAPTER 1

BeagleLogic System Image

The BeagleLogic system image is the official way to run BeagleLogic on the BeagleBone hardware. **No installation required**, just flash, boot and you are ready to capture samples.

Download the image here [Released on 2017-07-13, sha256sum = *be67e3b8a21c054cd6dcae7c50e9e518492d5d1ddaa83619878afeffe59*
]

Use username:"debian" and password:"temppwd" (without quotes) to log into the image.

Supported Platforms: (tested) BeagleBone Black, Seeed Studio BeagleBone Green and Seeed Studio BeagleBone Green Wireless and (currently untested) BeagleBone Black Wireless and SanCloud BeagleBone Enhanced.

Instructions

- Use Etcher to flash the downloaded image on an SD card.
- Insert the SD Card into the BeagleBone
- Hold down the USER button (usually in the bottom right corner) before applying power to the BeagleBone. This is required so that the BeagleBone boots from the bootloader in the SD Card.
- Verify BeagleLogic is running by doing `ls -l /dev/beaglelogic`. If BeagleLogic did not appear at `/dev/beaglelogic` proceed to *Troubleshooting*.
- Once booted, you can then continue to *make your first capture* using BeagleLogic

Troubleshooting

- Execute on a shell `journalctl | grep beaglelogic`
- For BeagleLogic loading correctly, the expected output should be something like this:

```
Jul 20 04:56:22 beaglebone bash[832]: beaglelogic-startup: Waiting for␣
↪BeagleLogic to show up (timeout in 120 seconds)
Jul 20 04:56:45 beaglebone beaglelogic[862]: Express server listening on port 4000
Jul 20 04:57:05 beaglebone kernel: remoteproc remoteproc1: Booting fw image␣
↪beaglelogic-pru0-fw, size 62576
Jul 20 04:57:05 beaglebone kernel: remoteproc remoteproc2: Booting fw image␣
↪beaglelogic-pru1-fw, size 31996
Jul 20 04:57:05 beaglebone kernel: misc beaglelogic: Valid PRU capture context␣
↪structure found at offset 0000
Jul 20 04:57:05 beaglebone kernel: misc beaglelogic: BeagleLogic PRU Firmware␣
↪version: 0.3
Jul 20 04:57:05 beaglebone kernel: misc beaglelogic: Device supports max 128␣
↪vector transfers
Jul 20 04:57:05 beaglebone kernel: misc beaglelogic: Default sample rate=50000000␣
↪Hz, sampleunit=1, triggerflags=0. Buffer in units of 4194304 bytes each
Jul 20 04:57:05 beaglebone bash[832]: beaglelogic-startup: Configuring LA pins
Jul 20 04:57:06 beaglebone bash[832]: beaglelogic-startup: Allocating 64MiB of␣
↪logic buffer to BeagleLogic
Jul 20 04:57:06 beaglebone kernel: misc beaglelogic: Successfully allocated␣
↪67108864 bytes of memory.
Jul 20 04:57:06 beaglebone bash[832]: beaglelogic-startup: Loaded
```

- If you see `beaglelogic-startup:  timeout. BeagleLogic couldn't load`, do `cat /proc/cmdline`
- If in cmdline you see `bone_capemgr.uboot_capemgr_enabled=1` and BeagleLogic still did not load, please open an issue.
- If in cmdline you do not see `uboot_capemgr_enabled`, then try:

```
sudo -i
echo beaglelogic > /sys/devices/platform/bone_capemgr/slots
echo cape-universalh > /sys/devices/platform/bone_capemgr/slots
```

- This should cause BeagleLogic to show up in the journalctl output as above.
- Then do `sudo service beaglelogic-startup restart` to initialize BeagleLogic
- If these steps do not work for you, please open an issue

Release Notes

This image release represents a major leap for BeagleLogic in general.

- This release is based on Debian 9 [Stretch], which is the latest available stable Debian version.
- The kernel has been upgraded from 3.8.13-bone kernel to the 4.9.36-ti-r46 release. BeagleLogic has been migrated to run on the 4.9 kernels, this needed a firmware and kernel module change. You can read more about it *here*
- Root login is disabled by default for a more secure image. However root permission is now no longer required for editing the BeagleLogic sysfs attributes.
- The underlying sigrok components have been updated to their latest release on the 13th of June 2017. This brings in support for new protocol decoders and faster protocol decoding.
- NodeJS version has been upgraded to v6.11 . This is used by the web interface.

- In order to load BeagleLogic, a overlay needs to be applied to the device tree that is used to boot the BeagleBone device. This is now by default applied at boot time using U-Boot's built in cape manager support. Hence it is essential that the new version of the bootloader bundled in the SD card is used to boot the device. Which means it is important to hold down the USER button on the BeagleBone while booting the device.

CHAPTER 2

Installation

Note: The instructions in this document do not apply if you are running the *BeagleLogic System Image*. You can directly proceed to *Making your first capture*

BeagleLogic has a simple installation script that can be used to install it on a clean BeagleBone Debian Image.

Note: Only BeagleBoard.org provided Debian images are natively supported. If you are using a customized system image build using buildroot or OpenEmbedded, additional steps will be required to complete installation.

Use the following commands:

```
cd <directory-to-install-beaglelogic-in>
git clone https://github.com/abhishek-kakkar/BeagleLogic
cd BeagleLogic
sudo ./install.sh
```

It is recommended to install BeagleLogic in /opt/BeagleLogic but because it is not accessible by the default user, some extra steps are needed:

```
cd /opt
sudo git clone https://github.com/abhishek-kakkar/BeagleLogic
sudo chown -R debian:debian /opt/BeagleLogic
cd BeagleLogic
sudo ./install.sh
```

Replace 'debian' with your own username in case you are not using the default user on the system.

Upgrading an existing installation

Do the following:

```
cd <path-to-installed-BeagleLogic>
git pull
sudo ./install.sh --upgrade
```

Advanced

You can read the install.sh file and the scripts directory for details on how BeagleLogic is installed, or if you want to manually tweak or customize any aspect of the installation.

CHAPTER 3

Differences Between BeagleLogic on 4.9 vs 3.8 kernel versions

BeagleLogic has recently been migrated to run on Linux Kernel Version 4.9 from the earlier kernel version 3.8.13 it used to run on.

For the user, there should be no noticeable difference in the way things function, but this article describes all the under-the-hood changes that have enabled this migration.

Changes:

- No root permissions are needed to modify the sysfs attributes located under `/sys/devices/virtual/misc/beaglelogic`. This is achieved by adding a udev rule to change these attributes' group to beaglelogic and adding the default user under the beaglelogic group so that one can modify the sysfs attributes without requiring root permissions. This will enable applications that need BeagleLogic to also not require running as root which also further enhances security.
- The firmware now has a resource table as needed by the remoteproc framework in order to load the firmware into the PRUs and configure interrupts correctly.
- The kernel driver utilizes the latest remoteproc framework from TI, which requires firmware loading to be handled by the kernel driver itself.
- rpmsg framework for message passing is not used, rather communication between the kernel driver and the PRU happens through a command-response exchange through the PRU0 SRAM. The kernel driver receives interrupts from the PRU whenever data is ready and it makes the data then available to userspace.
- PRU firmware for PRU1 is now a symlink to the actual PRU firmware being used. This can be useful to change the firmware so that applications like PRUDAQ can be enabled.

CHAPTER 4

Making your first capture

Follow the steps below to make your first capture using BeagleLogic.

Assuming you are running the BeagleLogic system image and you have booted up, and BeagleLogic device is available at '/dev/beaglelogic'

Using dd

Simply do:

```
dd if=/dev/beaglelogic of=myfile.bin bs=1M size=20
```

This will capture 20MB worth logic data (20MSamples if you are in the 8-bit mode and 10MSamples if you are in 16-bit mode)

To modify the sample rate, and other aspects of the capture, use the *sysfs attributes*

Using sigrok

Use:

```
sigrok-cli -d beaglelogic:logic_channels=8 -c samplerate=10M --samples 10M -o capture.
↪sr
```

To switch between 8-bit mode and 16-bit mode, replace 8 with 14 (maximum 14 channels are possible)

The other switches are self-explanatory. This command captures 10MSamples at 10MSamples/second and saves it into a file 'capture.sr'. The capture.sr is a sigrok-specific file format that can be downloaded and opened in the PulseView software.