Linux Device Driver
for
BlueNRG

Project Synopsis
of Major Project

Bachelors of Technology
Computer Science & Engineering

Govind Sharma
D4CSE 2013-17
1311053
135026
9779490183

Vasdev Flour Mills
V.P.O. Kishanpura Kalan
Moga, Punjab
142058
dnivogamrahs@gmail.com

Guru Nanak Dev Engineering College
Ludhiana 141006

CONTENTS

1 Introduction **1**

1.1 Technologies Used 1

1.1.1 Bluetooth Low Energy 1

1.1.2 BeagleBone Black 1

1.1.3 Raspberry Pi 2

1.1.4 Buildroot 3

2 Feasibility Study **4**

2.1 BlueNRG compatibility with Bluez 4

2.2 BlueNRG compatibility with Linux 4

3 Planning of Work **5**

4 Facilities required for proposed work **6**

4.1 Hardware Requirements 6

4.2 Software Requirements 6

LIST OF FIGURES

1.1 BeagleBone Black 2
1.2 Raspberry Pi 3 2
1.3 Buildroot 3

CHAPTER 1

INTRODUCTION

The topic of my project is a device driver for BlueNRG in the Linux kernel.
BlueNRG is a Bluetooth Low Energy network coprocessor developed by STMicroelectronics. Bluetooth is a wireless technology standard for exchanging data over short distances. While exchanging data is the most accurate and popular use case of Bluetooth, the technology has transcended this simple initial purpose. Bluetooth is managed by the Bluetooth Special Interest Group (SIG), which has more than 25,000 member companies in the areas of telecommunication, computing, networking and consumer electronics. Bluetooth technology has found itself to be a perfect fit for many and many short distance communication problems, and has also re invented and improved itself over the years in order to keep up with the market trends. One such monumental change or addition to Bluetooth that is going to feature heavily in this report is Bluetooth Low Energy, a smarter and edgier version of Bluetooth.

1.1 Technologies Used

1.1.1 Bluetooth Low Energy

Marketed as **Bluetooth Smart**) is a wireless personal area network technology designed and marketed by the **Bluetooth Special Interest Group aimed** at novel applications in the healthcare, fitness, beacons, security, and home entertainment industries.
It is part of the Bluetooth **4.0 specification**. It aims at being a low energy, low cost, low bandwidth, low power and low complexity extensible framework for exchanging data pertaining to various applications fields.

1.1.2 BeagleBone Black

The BeagleBoard is a low-power open-source hardware single-board computer produced by Texas Instruments in association with Digi-Key and Newark element14. The BeagleBoard was also designed with open source software development in mind, and as a way of demonstrating the Texas Instrument's OMAP3530 system-on-a-chip. The board was developed by a small team of engineers as an educational board that could be used in colleges around the world to teach open source hardware and software capabilities. It is also sold to the public under the Creative Commons share-alike license. The board was designed using Cadence OrCAD for schematics and Cadence Allegro for PCB manufacturing; no simulation software was used.

Figure 1.1: BeagleBone Black

1.1.3 Raspberry Pi

Raspberry Pi is a series of credit card-sized single-board computers developed in the United Kingdom by the Raspberry Pi Foundation to promote the teaching of basic computer science in schools and developing countries. The original model got way more popular than anticipated, outside of the target market; enthusiasts use it, and the later models, for various uses, such as robotics. Accessories, such as keyboard and mice (not needed for some uses) and even a case, are not included, while planned for; they've frequently been included in unofficial bundles, and later in an official bundle.

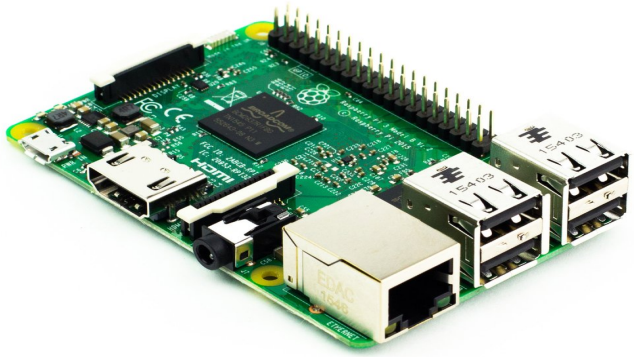

Figure 1.2: Raspberry Pi 3

1.1.4 Buildroot

Buildroot is a set of Makefiles and patches that simplifies and automates the process of building a complete and bootable Linux environment for an embedded system, while using cross-compilation to allow building for multiple target platforms on a single Linux-based development system. Buildroot can automatically build the required cross-compilation toolchain, create a root file system, compile a Linux kernel image, and generate a boot loader for the targeted embedded system, or it can perform any independent combination of these steps. For example, an already installed cross-compilation toolchain can be used independently, while Buildroot only creates the root file system.

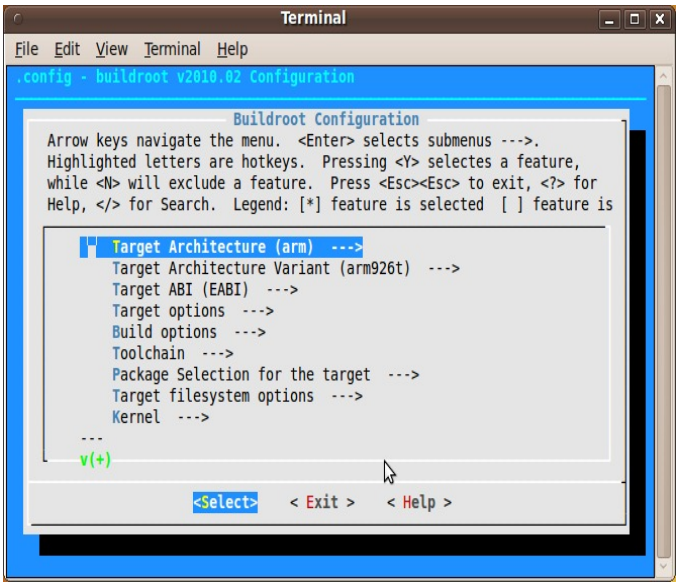

Figure 1.3: Buildroot

CHAPTER 2

FEASIBILITY STUDY

The feasibility of a Linux device driver for BlueNRG is contingent upon the following two aspects

2.1 BlueNRG compatibility with Bluez

The answer is yes. Bluez interfaces with any device or adapter using the Host Controller Interface which is essentially a protocol standard defined in the Bluetooth specification itself as a means to communicate between the host and the controller. Since BlueNRG is responsive to HCI commands, all seems to be set here. The communication between the host and the controller are to be had through HCI commands (send from the host) and HCI events (generated by the controller in response to the commands send by the host). According to the ACI programing manual, HCI is indeed the subset of ACI. So BlueNRG should be able to respond to HCI commands offered by Bluez of Linux.

2.2 BlueNRG compatibility with Linux

BlueNRG is based on SPI. Any form of communication between the host and BlueNRG is to be had through the Serial Peripheral Interface. While there isnt a SPI based Bluetooth driver in Linux at the time of this writing, but there certainly is a rich SPI support in Linux. There are many network drivers that allow the communication between the host and the controller through SPI. So following the same footsteps it should be possible to send commands (HCI) from the host to the controller using SPI.
Considering the above things, it seems feasible to write a driver for BlueNRG in Linux. The driver shall essentially act as a bridge between Bluez protocol stack residing partly in the user space and partly in the kernel space of Linux and BlueNRG. Every Bluetooth action that is needed to be performed will be in the form of an HCI command, which is what the driver should expect from Bluez and interface it through SPI to BlueNRG.

CHAPTER 3

PLANNING OF WORK

Following are the steps needed to complete the project and then test it:

1. Choosing a host system upon which the development is to be had e.g. **BeagleBone Black** or **Raspberry Pi 3**. Raspberry Pi 3 seems like the more obvious choice because of its builtin Bluetooth capabilities and configurations.

2. Preparing a Linux environment consisting of a kernel, device tree binaries, bootloader and a root file system consisting of all the necessary software support for proper working of Bluetooth. **Buildroot** is to be used for this step.

3. Enabling SPI support in the host.

4. Configuring the device tree binaries of the board in the Linux kernel so as to recognise BlueNRG as an SPI device.

5. Coding the driver. This is a big step as it includes many substeps and considerations from making the driver communicate with the host and then from there making it respond to commands sent from the driver. Once the hardware starts responding then comes the turn of the main coding activity i.e. extending all the features offered by BlueNRG in the driver and mapping them to the command line utilities offered by Bluez.

6. Documenting the driver.

7. Embedding the whole code into a local copy of the kernel and booting the host from it.

8. Testing the new environment with BlueNRG to see if it works like it should.

9. Preparing patches for the final product and sending them to appropriate Linux maintainers for consideration.

CHAPTER 4

FACILITIES REQUIRED FOR PROPOSED WORK

4.1 Hardware Requirements

The following hardware components are to be used in the development of this project.

- **BlueNRG**: a Bluetooth Low Energy network coprocessor by **STMicroelectronics**
- A **host** running on Linux e.g. **BeagleBone Black** and **Raspberry Pi 3** are to be used for this.

4.2 Software Requirements

The following softwares may be used while developing and testing the software.

- **Linux** kernel based operating system.
- **Buildroot** for building Linux environments.
- **Cscope** used for browsing and exploring source files in C.
- **Git** used for managing source code.
- **GDB** a graphical debugger.
- **LaTeX** for all the report work.

BIBLIOGRAPHY

[1] **Getting started with Bluetooth low energy**
a book by *Kevin Townsend* and *Robert Davidson*.

[2] **Linux Device Drivers**, 4th edition
a book by *Jonathan Corbet, Alessandro Rubini*, and *Greg Kroah-Hartman*.

[3] **Linux Kerenl Documentation for device drivers**.

[4] **BlueNRG network processor Datasheet**
http://www.st.com/en/wireless-connectivity/bluenrg.html