Getting Started with Cubieboard

Embedded platforms are interesting because they combine two fun fields. On one side, there is open source software, and on the other side, there is open source hardware. But a little further from that, we have electronics (AVR-microcontrollers), which can be very interesting and fun to work with. Cubieboard is a powerful single board computer, similar to Raspberry Pi, that supports multiple operating systems, such as Ubuntu and Debian.

This book will teach you everything you need to know about project development using Cubieboard, even if you are not an embedded platform expert.

The book starts by going over the most well-known Allwinner development boards, helps you choose a board, and recommends additional required hardware. Next, the book briefly explains how to "talk" to the board. Then, things start to get interesting with the installation of a desktop OS onto an SD card and booting into a fully graphical desktop system. Concluding this work, the last chapter gives you an example of how to connect external peripherals such as an LED.

Who this book is written for

If you are anywhere from a beginner to an advanced user of ARM, who wishes to get into the rapidly advancing world of development boards, such as Cubieboard, this is the book for you. Whether you are a hobbyist or a professional, you will learn from this book as it teaches you in an easy-to-follow manner. No previous ARM experience is required.

What you will learn from this book

- Differentiate between the numerous ARM development boards based on the Allwinner A-series of chips
- Connect and communicate with a development board using a UART interface
- Install Fedora to create a desktop system
- Create a custom rootfs based on Debian or Ubuntu
- Set up a server that runs various services, such as a file and a web server
- Compile the bootloader and the kernel from scratch using a board support package (BSP), creating your own hardware support package
- Familiarize yourself with some basic electronic concepts using Cubieboard, as you move on to toggling GPIO pins and making LEDs blink

Cover Design: Purnima Nair


Free Sample
In this package, you will find:

- The author biography
- A preview chapter from the book, Chapter 1 “Choosing the Right Board”
- A synopsis of the book’s content
- More information on Getting Started with Cubieboard

About the Author

Olliver M. Schinagl is Austrian-born and a software developer at heart with a strong interest in electronic engineering. Embedded software is where both his passions come together. Having lived in the Netherlands for most of his life, Olliver is currently working at Ultimaker, a 3D printer manufacturer, where his love for Linux, free and open source software, and embedded development is satisfied. Having worked on open source projects, and as a longtime member of the linux-sunxi community, Olliver has in-depth and hands-on experience with Allwinner-based hardware.

He always had a desire to teach but a stronger desire to work on open source projects and embedded hardware. Thus, when offered the chance to write a book in his spare time, he decided to listen to his inner voice and took the chance to use the printed form to teach.

Having never done any writing except for academic work, this was both a challenge and a great experience. Hopefully, you will appreciate the effort and not only learn from the things brought via this book, but also gain the appetite to work out creative ideas, put the knowledge to good use, and share it with others so they can then benefit from it.

Writing a book costs time, and to understand and support this, I would like to thank my partner in life, Anshariah, who encouraged and cursed those late night writing sessions. Additionally, I would like to thank my parents and all my friends for always being there for me, supporting me, and being proud as parents and friends would be.

Finally, a pledge of gratitude goes out to all the free and open source software and hardware developers and advocates for all the things they make and create, all the things they share, and all the things I have learned from. It is because of them that I am able to write code and text using all the source tools. It is people like you who, in the end, make the world a better place.
Getting Started with Cubieboard

Over the last few years, ARM chips have become trendy and ubiquitous, ranging from the phone and tablet market to power-efficient server farms. The low cost associated with the chips in conjunction with their powerful features makes them an apt choice for hobbyists and enthusiasts. In addition to offering a ton of connectivity, these chips have been used by several manufacturers on their development boards. The Cubieboard is a type of board with built-in networking and various input and output ports, making it an awesome utility for myriad purposes, such as media centers, robotic projects, home automation, web servers, and home security systems, to mention a few. The Cubieboard is a microcontroller, which provides a whole new set of capabilities with the extensibility of desktop machines but without the bulk or noise.

Low cost, highly expandable, and high performing with a massive, diverse range of uses and applications, the Cubieboard will revolutionize the way we think about computing and programming. With its power-packed attributes and versatility, you can create fun things. There is absolutely no fixed way to develop complex projects; however, this book will give you enough basics of the Cubieboard in a few different realms so that you can dig deeper on your own.

What This Book Covers

*Chapter 1, Choosing the Right Board*, starts with an overview of various development boards and compares a few popular ones to help you choose a board tailored to your requirements. You will also take a look at the additional hardware and a few extra peripherals that will help you understand the stuff you require for your projects.

*Chapter 2, Getting Started with the Hardware*, helps you with the initial settings before you can try out things with it. After unwrapping the Cubieboard, you will learn the procedure to connect a serial port to the development board and move on to booting up the preinstalled software.

*Chapter 3, Installing an Operating System*, explains the procedure of installing an operating system in addition to installing a fully-functional graphical desktop environment onto a microSD card. It also points out the difference between an OS image and a clean installation, thereafter moving on to installing Fedora in addition to writing the OS image to a microSD card.

*Chapter 4, Manually Installing an Alternative Operating System*, helps you with the process of installing a customized OS on an alternative medium (SATA SSD) in addition to making the destination medium bootable using the command line.
Chapter 5, Setting Up a Home Server, explains how the Cubieboard can be used as a home server efficiently in addition to setting up different services to be used in a home environment. You can learn about the procedure of setting up a web server, file server, torrent server, and then summing it up with setting up a personal cloud.

Chapter 6, Updating the Bootloader and Kernel, helps you to understand the difference between the various bootloader and kernel types while also assisting you with the process of obtaining and installing a new bootloader or kernel onto an SD card, which will be used as a boot device. Kernels often get updated to newer versions with security fixes or support for new hardware, thereby making it mandatory to know about them when working with many ARM boards, such as the Cubieboard.

Chapter 7, Compiling the Bootloader and Kernel Using a BSP, deals with the board support package (BSP), thereby helping you compile the bootloader and kernel from source when some changes are to be made to the source code of the bootloader and kernel. You will learn to use BSP in conjunction with Git and create an easy-to-use, device-specific hardware pack.

Chapter 8, Blinking Lights and Sensing the World, starts with explaining basic electronic concepts and moves on to toggling GPIO pins and then make LEDs blink, thereby encouraging you to try out new things as you make your foray into the world of possibilities with the Cubieboard.

Appendix A, Getting Help and Finding Other Helpful Online Resources, educates you on the online resources at your disposal due to the vibrant communities and also how to obtain these resources and get help from the community in general.

Appendix B, Basic Linux Commands Cheatsheet, is a collection of various Linux commands that form a major part of your workload while using the Cubieboard, thereby helping you get to grips with the technology.

Appendix C, The FEX Configuration File, helps you understand the FEX files that are imperative due to the fact that they are used to configure the drivers.

Appendix D, Troubleshooting the Common Pitfalls, is a small guide that will be quite handy when faced with errors and hurdles, such as boot failures, stability issues, and errors that pop up while executing commands.
Choosing the Right Board

It is that time of the year again when there are a few days to spare, and you are anxious to play with one of these new ARM development boards everybody keeps talking about. There are, however, a lot of boards available. With so many choices available, which board do you pick? Choosing the board to start working with can make a difference later on, so this chapter provides an introduction to the various boards and states the major differences between them. While the focus of this book does indeed lie on the Cubieboard family from Cubietech, it might still be prudent to give this chapter some attention for a potential second board. Additionally, this book does apply just as easily to the boards mentioned here.

In this first chapter, we will cover the following topics:

- Why are there so many boards to choose from?
- An overview of various boards
- Highlighting the most popular boards
- Ideas regarding what additional hardware is required

Wading through the forest of available chips and boards

There are many chips and even more boards to choose from when going into ARM development. This chapter also provides a short introduction to various chips and compares them.
A short overview of chips

In the last few years, ARM-based Systems on Chips (SoCs) have become immensely popular. Compared to the regular x86 Intel-based or AMD-based CPUs, they are much more energy efficient and still perform adequately. They also incorporate a lot of peripherals, such as a Graphics Processor Unit (GPU), a Video Accelerator (VPU), an audio controller, various storage controllers, and various buses (I2C and SPI), to name a few things. This immensely reduces the required components on a board. With the reduction in the required components, there are a few obvious advantages, such as reduction in the cost and, consequentially, a much easier design of boards. Thus, many companies with electronic engineers are able to design and manufacture these boards cheaply.

So, there are many boards; does that mean there are also many SoCs? Quite a few actually, but to keep the following list short, only the most popular ones are listed:

- Allwinner's A-series
- Broadcom's BCM-series
- Freescale's i.MX-series
- MediaTek's MT-series
- Rockchip's RK-series
- Samsung's Exynos-series
- NVIDIA's Tegra-series
- Texas Instruments' AM-series and OMAP-series
- Qualcomm's APQ-series and MSM-series

While many of the potential chips are interesting, Allwinner's A-series of SoCs will be the focus of this book. Due to their low price and decent availability, quite a few companies design development boards around these chips and sell them at a low cost. Additionally, the A-series is presently the most open source friendly series of chips available. There is a fully open source bootloader, and nearly all the hardware is supported by open source drivers. Among the A-series of chips, there are a few choices. The following is a list of the most common and most interesting devices:

- **A10**: This is the first chip of the A-series and the best supported one as it has been around for a long time. It is able to communicate with the outside world over I2C, SPI, MMC, NAND, digital and analog video out, analog audio out, SPDIF, I2S, Ethernet MAC, USB, SATA, and HDMI. This chip initially targeted everything, such as phones, tablets, set-top boxes, and mini PC sticks. For its GPU, it features the MALI-400.
• **A10S**: This chip followed the A10; it focused mainly on the PC stick market and left out several parts, such as SATA and analog video in/out, and it has no LCD interface. These parts were left out to reduce the cost of the chip, making it interesting for cheap TV sticks.

• **A13**: This chip was introduced more or less simultaneously with the A10S for primary use in tablets. It lacked SATA, Ethernet MAC, and also HDMI, which reduced the chip's cost even more.

• **A20**: This chip was introduced way after the others, and even was pin-compatible to the A10 with the intend to replace it. As the name hints, the A20 is a dual-core variant of the A10. The ARM cores are slightly different; Cortex-A7 has been used in the A10 instead of Cortex-A8 used previously.

• **A23**: This chip was introduced after the A31 and A31S and is reasonably similar to the A31 in its design. It features a dual-core Cortex-A7 design and is intended to replace the A13. It is mainly intended to be used in tablets.

• **A31**: This chip features four Cortex-A7 cores and generally has all the connections that the A10 has. It is, however, not popular within the community because it features a PowerVR GPU that, until now, has seen no community support at all. Additionally, there are no development boards commonly available for this chip.

• **A31S**: This chip was released slightly after the A31 to solve some issues with the A31. There are no common development boards available.

### Choosing the right development board

Allwinner's A-series of SoCs was produced and sold so cheaply that many companies used these chips in their products, such as tablets, set-top boxes, and eventually, development boards. Before the availability of development boards, people worked on and with tablets and set-top boxes. The most common and popular boards are from Cubietech and Olimex, in part because both companies handed out development boards to community developers for free.
Olimex

Olimex has released a fair amount of different development boards and peripherals. A lot of its boards are open source hardware with schematics and layout files available, and Olimex is also very open source friendly. You can see the Olimex board in the following image:

Olimex offers the A10-OLinuXino-LIME, an A10-based micro board that is marketed to compete with the famous Raspberry Pi price-wise. Due to its small size, it uses less standard 1.27 mm pitch headers for the pins, but it has nearly all of these pins exposed for use.
You can see the A10-OLinuXino-LIME board in the following image:

The Olimex OLinuXino series of boards is available in the A10, A13, and A20 flavors and has more standard 2.54 mm pitch headers that are compatible with the old IDE and serial connectors. Olimex has various sensors, displays, and other peripherals that are also compatible with these headers.

Olimex recently announced that it will be releasing a **System on a Module (SoM)**. It is identical in concept to the Itead board, which will be mentioned in a later section.
Cubietech

Cubietech was formed by previous Allwinner employees and was one of the first development boards available using the Allwinner SoC. While it is not open source hardware, it does offer the schematics for download. Cubietech released three boards: the Cubieboard1, the Cubieboard2, and the Cubieboard3—also known as the Cubietruck. Interfacing with these boards can be quite tricky as they use 2 mm pitch headers that might be hard to find in Europe or America. You can see the Cubietech board in the following image:
Cubieboard1 and Cubieboard2 use identical boards; the only difference is that A20 is used instead of A10 in Cubieboard2. These boards only have a subset of the pins exposed. You can see the Cubietruck board in the following image:

Cubietruck is quite different but is a well-designed A20 board. It features everything that the previous boards offer, along with Gigabit Ethernet, VGA, Bluetooth, Wi-Fi, and an optical audio out. This does come at a cost as there are fewer pins to keep the size reasonably small. Compared to Raspberry Pi or LIME, it is almost double the size.
Lemaker

Lemaker made a smart design choice when releasing its Banana Pi board. It is an Allwinner A20-based board but uses the same board size and connector placement as Raspberry Pi, hence the name Banana Pi. Because of this, many of those Raspberry Pi cases could fit the Banana Pi and even shields will fit it. Software-wise, it is quite different and does not work when using Raspberry Pi image files. Nevertheless, it features composite video out, stereo audio out, HDMI out Gigabit Ethernet, two USB ports, one USB OtG port, CSI out and LVDS out, and a handful of pins. Also available are a LiPo battery connector, a SATA connector, and two buttons, but those might not be accessible on a lot of standard cases. See the following image for the topside of the Banana Pi:
Itead and Olimex

Itead and Olimex both offer interesting boards, which are worth mentioning separately. The Iteaduino Plus and the Olimex A20-SoM are quite interesting concepts; the computing module, which is a board with the SoC, memory, and flash, which are plugin modules, and a separate baseboard. Both of them sell a very complete baseboard as open source hardware, but anybody can design their own baseboard and buy the computing module. You can see the following board by Itead:
Choosing the Right Board

Refer to the following board by Olimex:

Additional hardware
While a development board is a key ingredient, there are several other items that are also required. A power supply, for example, is not always supplied and does have some considerations. Also, additional hardware is required for the initial communication and to debug.

Serially interfacing with the board
With headless systems such as these, things don't always just work. Sometimes, debugging at a lower level is required. This is also certainly true with development boards such as these. An error occurs, and there's no output; what could have possibly gone wrong? So spending hours on trial and error can be avoided; the old and trusty serial port exists on many types of hardware. With Allwinner's SoCs, they are implemented in two ways.
Universal asynchronous receiver/transmitter

On all of the developer boards discussed throughout this book, there are dedicated pins to connect to the serial port of the chip. If the PC that is used to connect to the developer board has such a serial port, be wary of how this is connected. While both speak the same protocol, they operate at different voltages to do this; thus, a level translator is required at the least. Most PCs are without a serial port these days anyway and have to rely on a USB to universal asynchronous receiver/transmitter (UART) adapter. When getting a USB to UART adapter, it is important that they are 3.3V TTL. Cubieboards are usually shipped with a USB to UART adapter, as shown in the following image:
The microSD adapter
Sometimes, the UART simply isn’t available for use as in the case of most tablets. In such cases, a second UART is made available through the microSD card slot. A specific adapter is required to connect to the previously mentioned UART. In the following image, a microSD to UART adapter can be seen (this specific variant also has the ability to grant access to the JTAG pins):

The microSD card
Usually, a microSD card is the boot medium for these boards. It can be thought of as a bootable CD or USB drive used on a PC. When creating microSD cards to be used as a medium, it is advisable to use a class 10 or faster microSD.
Power supply

Believe it or not, but most developer boards actually come without a power supply; this is usually due to the following reasons:

- Without the power supply, developer boards do not have to pass the FCC regulations
- Importing a power supply might require certain local certification and might be forbidden
- It can give rise to the complexity of the developer not knowing which power supply needs to be sent to a country
- It has reduced the cost, as not bundling the power supply has brought about a reduction in the cost

Most boards will take 5 volts for their input, but 700 milliamp of current is the least that they should supply when not using anything power hungry, such as an HDD or an LCD. If extra peripherals are attached, this requirement also goes up. A proper 5-volt, 2-amp power supply will be enough to power a board under full load with an LCD attached. Depending on the power requirements of the hard drive, even that should be able to work quite nicely. When in doubt, always check the power drain or power supply stability to exclude that from causing strange issues. Cheaply made power supplies available from various shops are often overrated and thus they might not supply the required current; however, they might not supply the required current and cause everything to run unstably.

Summary

After working your way through this chapter, you should now have an idea of the available, popular development boards, and which one might be a good choice for you. Finally, it should be noted that extra hardware is sometimes required or, at least, extremely helpful to work with these boards.

The next chapter will take this newly acquired hardware and explain how you can interact with it. If the selected board comes with preinstalled software, booting it will be covered as well.
Where to buy this book

You can buy Getting Started with Cubieboard from the Packt Publishing website. Alternatively, you can buy the book from Amazon, BN.com, Computer Manuals and most internet book retailers. Click here for ordering and shipping details.