Learning Banana Pi

Banana Pi is a small but powerful ARM board that can run on multiple operating systems. Banana Pi has been adopted by tech-savvy enthusiasts and hobbyists to develop their projects. This book will take you from opening your Banana Pi’s box for the first time all the way to working with hardware and code. We will explore the different configuration options and operating system choices. You’ll proceed to working with Banana Pi hooked up to a breadboard and connecting hardware components such as LEDs and motors to the board. Further, you’ll build an Internet radio working with an LCD display, interacting with user input, and streaming audio over the Internet. This book also acquaints you with another really popular use of Banana Pi, home servers. Finally, you’ll realize the power of the robust hardware of Banana Pi by playing games on Android and Linux.

By the end of this book, you should be familiar with the basics of development using Banana Pi, from installation to building basic projects.

Who this book is written for
This book is for anyone who wants to explore Banana Pi and get started building projects on their own using it. No prior experience of working with single board computers is required.

What you will learn from this book
- Get started with single-board computing by diving deep into the operating system and hardware of Banana Pi
- Interact with the GPIO pins from scripts and code to start programming on the board
- Connect hardware components to the GPIO pins and control them directly from Banana Pi
- Connect Banana Pi devices with your home network to download packages from the Internet
- Link your Banana Pi remotely with other computers via SSH to run as a headless device
- Work with the onboard IR receiver of Banana Pi as well as connect external sensors and interact with them through their code
- Build an Internet radio powered by Banana Pi to stream podcasts and music
- Build and run a home server using Banana Pi to sync files, calendars, and contacts

Unleash the power of Banana Pi and use it for home automation, games, and various practical applications

Daniel Blair

In this package, you will find:

- The author biography
- A preview chapter from the book, Chapter 1 'An Overview of the Banana Pi'
- A synopsis of the book’s content
- More information on Learning Banana Pi
Daniel Blair is a tech entrepreneur and technologist from Winnipeg, Manitoba. He is heavily involved in start-up and maker communities in Winnipeg and regularly attends and organizes community events that are centered around development and open hardware.

When the Banana Pi board was first released to the community, Daniel got involved in the Banana Pi due to his experience with similar boards, such as the Raspberry Pi. He has worked on various projects, which includes bringing the Tor browser to the Banana Pi.

He has used nearly every incarnation of the Banana Pi and the boards that it has inspired. He has been a part of all kinds of projects related to software and hardware. Although Daniel has written a lot on the Banana Pi on his various websites, this is his first book on the subject.

Bit Space Development, a tech start-up founded by Daniel, offers various solutions to small businesses that need technology such as menu boards, Internet access points, and company dashboards. The power of open hardware has enabled them to offer top-quality support and robust hardware solutions as a small company.
This book covers the topics of the Banana Pi and single-board computing. Single-board computing generally refers to the computers that reside entirely on one chip. The computer that we are dealing with is the Banana Pi, which is based on the Raspberry Pi, the open source computer.

Single-board computing has become extremely popular as of late for its use in prototyping, education, and gaming. This book will introduce you to all of these topics. We will go from the beginner to the intermediate level in just a few chapters.

The communities that developed these devices have come up with many cool use cases, such as tiny low power servers, media devices, and arcade machines. We will explore a few popular uses of the Banana Pi, which will show you just how versatile your little computer is.

Also, there are a lot of different operating systems that are available for the Banana Pi. We will explore the set up and use of two of the most popular operating systems, Android and Linux. You will be an intermediate user by the time we are done here, which will set us up to explore complex projects, such as robotics, further down the road.

**What this book covers**

*Chapter 1, An Overview of the Banana Pi*, gives an introduction to the Banana Pi board. It also covers the basic components of the Banana Pi.

*Chapter 2, Preparing the SD Card*, covers one of the most important steps to owning your first Banana Pi. We will learn how to prepare the SD card on multiple operating systems.
Preface

Chapter 3, *Linux and the Command Line*, talks about the Linux operating system and some useful commands. This is important when you are using the Banana Pi because almost all of the operating systems for the Banana Pi are based on Linux.

Chapter 4, *Programming on the Pi*, explores the different programming languages that are supported by the Banana Pi. This will introduce the user to some code and different use cases.

Chapter 5, *Hardware for Your Pi*, explores some of the hardware that you can use with the Banana Pi. We will explore electrical components such as the LED and capacitor.

Chapter 6, *Interacting with Sensors*, deals with sensors, which are very useful in circuits. We will have a look at a few different sensors and how they can be used.

Chapter 7, *Building an Internet Radio*, is built upon the discussions from the previous chapters and shows you how to construct a media device. The Internet radio will be explored in a couple of ways, which will allow for it to be used more easily.

Chapter 8, *Building a Home Server*, takes a look at how you can build a low-power, low-cost Banana Pi home server. We will delve into the Google Apps replacements and shared calendars. All of this will be powered by your own hardware.

Chapter 9, *Gaming on Your Pi*, is the most fun chapter of this book. We will explore video games on the Banana Pi. We will look at building a retro game computer and explore game streaming on Android.
This book is going to help you on your journey into single-board computing. We are going to cover a lot of topics that are popular in the community and build some skills that will help us when we are working on projects later on. The world of single-board computing is new and fast-paced but it opens up so many doors when it comes to education and technology. The Raspberry Pi (pictured here) is the inspiration for the Banana Pi.
An Overview of the Banana Pi

Single-board computing is a relatively new technology that was popularized by the Raspberry Pi and really made it easy to prototype and learn. The Raspberry Pi brought cheap computing power to people around the world and empowered kids and hobbyists to learn about Linux and start to play with electronics. It brought out the maker in all of us and we started to build and see what we could do.

The Banana Pi allows you to take rapid prototyping to the next level. With an operating system directly on the board and access to the general purpose input output (GPIO) pins, you can not only set up a home theatre PC (HTPC), you can actually build the next great product using one.

Learning electronics is easier than ever when you attach the GPIO output to a breadboard and build directly from that. The Banana Pi, like the Raspberry Pi, is so small you can even incorporate it into a finished product and forego the need to implement another micro controller. The GPIO pins allow you to write code on the device that interacts with physical hardware such as LEDs, resistors, and switches. This means you can mock-up your idea, and get a working version of it before making a commitment to manufacturing something.
Coding is something that is becoming very important as a skill now, and with coding tutorials, and software that you can install directly on the Banana Pi, you can build skills that will be valuable to you for a long time. The Google Coder interface is pictured in the preceding screenshot.

The Banana Pi is usable as a web server, which means that you can learn languages such as Ruby, PHP, and Python directly on the board, or push code from your other computer to the Pi, which can act as your local development server. I have found projects such as Google Coder to be incredibly useful tools for learning these skills.

Popular development tools are available for the Banana Pi on Linux as long as there is an ARM-compatible build for it. This includes software such as gedit. You can also run web-based solutions (for instance, Adafruit’s WebIDE, which runs as a web service directly on the Pi).

One of the most popular uses of the Raspberry Pi is as an HTPC or Media Centre PC. This means that the device is hooked up to a TV and used for the displaying of media. With tools such as XBMC (see the preceding screenshot) and Samba for file sharing, this is quite easy. The Banana Pi makes this even more accessible. What used to be a slower GUI is now sleek and fast and, with the built-in IR receiver, you can even use a remote to control the Pi from the sofa without the need for a keyboard and mouse. The SATA port on the M1 makes it faster than ever to get media to your TV due to the read/write speed of SATA being a lot faster than USB. The M2’s built-in Wi-Fi is also very useful for this task.
Popular software such as Octoprint (used to control a 3D Printer—for example, the Printrbot) is also compatible with the Banana Pi. This allows you to directly connect your Banana Pi to your 3D printer and use that as a print server. You can send your 3D models to the Pi and have it slice up your model and start the print. This is one of my favorite uses for the Pi, something that also translates into rapid prototyping.

Differences between the Raspberry Pi and Banana Pi

The Raspberry Pi took open source software and added a hardware aspect to it. Now for less than $100 you could have a cheap computer and that really got people excited. A huge community has grown from the Raspberry Pi, which has helped with the success of the board and has been the inspiration for newer boards with better specs such as the Banana Pi. The Banana Pi is a board that builds on the success of the Raspberry Pi. It is a single-board computer, which means that everything needed to operate on the system, is built-in, all you need to do is add your peripherals. The Banana Pi takes it a step further with increased stats and performance, which allow you to really take your projects to the next level.

The community for the Banana Pi is growing and you can already find projects and tutorials specific to it. The additional hardware such as the SATA port and Gigabit LAN has made it very popular for file servers or ownCloud. The nice thing is, you can still do everything you wanted to do with the Raspberry Pi, but with a little extra hardware. The following screenshot shows some hardware used to breadboard a Banana Pi.
The M1
There are two versions of the Banana Pi: the M1 and the M2. The M1 is more common and currently sports a 1.2 GHz dual core processor, while the M2 is newer and has a 1.4 GHz quad core processor. Both boards operate similarly, but the M2 is really where things are headed. You can identify the board you are using by counting the GPIO pins in the bottom-left corner of the board. If you have 40 (like the Raspberry Pi B+) then you have the M2. If you have 26, like the classic Raspberry Pi, then you have the Banana Pi M1.

Getting familiar with the hardware on the Banana Pi is important; this will help you determine what can be done with the Pi, and will help you decide which model to use in your future projects at the planning phase. The M1 board has a very similar layout to the Raspberry Pi model B (classic). On the left side of the board you have your 26 GPIO pins, AV video out, and 3.5 mm audio out. The front of the board has your LED indicators, IR receiver, and Ethernet port. The ports for the M1 are outlined in the following diagrams:
Both the M1 and M2 boards have nearly identical mounting holes around the four corners of the board.

The M2
Down the right side of the board you have the UBOOT key, LVDS connector for video displays, HDMI out, SATA, 5 v micro USB power port, and SATA power out. The bottom of the board has the power and reset switches, as well as a micro USB OTG (host) port. The back side of the board shows the CPU and the SD card slot for the OS.

The M2 board is similar in layout to the Raspberry Pi model B+, with 40 GPIO pins in the bottom-left corner of the board. The layout does differ though. On the left-hand side of the board you will find the GPIO pins, 3.5 mm audio jack, and two USB 2.0 ports. On the top of the board you will find the IR receiver, two more USB 2.0 ports, and the Ethernet port. The right-hand side of the board has the boot button, LVDS/RGB display interface, HDMI out, and 5V, 2A power jack (not USB). The bottom edge of the board has the camera interface, power and reset buttons, and the micro USB OTG host port.
These ports are outlined in the following diagram:

The bottom side of the board has the CPU, Wi-Fi antenna connector, RealTek 802.11 b/g/n Wi-Fi adapter, and the micro SD card slot for the OS. The biggest difference between the M1 and the M2 is the lack of SATA when upgrading the M2. This may determine which model you plan to use, or it may not affect your decision. The M2 supports Micro SD and Wi-Fi, unlike the M1, which does impact the final project if those are features you are looking for.
An Overview of the Banana Pi

The next image shows the bottom side of the M2:

```
Overall, the boards are similar in terms of architecture and ports with several exceptions for each board. You could compare the Banana Pi with the Model B versions of the Raspberry Pi. The Raspberry Pi Model B+ builds upon and improves some aspects of the Model B classic, however both are still useful. The M1 is not necessarily worse than the M2, since it offers different hardware and even some different software to the M2. The following chart gives a direct comparison of features between both versions of the Banana Pi. The Banana Pi M1 uses the A20 processor, which is currently very widespread and widely supported. The A31 boasted by the M2 is still very new, and the lack of SATA on the board is a shortcoming of the board’s design.

The following table outlines the differences in hardware between the Banana Pi M1 and M2. The boards are comparable; although the M2 is the upgrade, the M1 has features that the M2 does not.
The following table visualizes the differences between both the M1 and M2 Banana Pi boards:

<table>
<thead>
<tr>
<th></th>
<th>Banana Pi M1</th>
<th>Banana Pi M2</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>CPU</strong></td>
<td>A20 ARM® Cortex™-A7 Dual-Core</td>
<td>A31S ARM Cortex-A7 quad-core, 256 KB L1 cache 1 MB L2 cache</td>
</tr>
<tr>
<td><strong>GPU</strong></td>
<td>ARM Mali-400MP2 complies with OpenGL ES 2.0/1.1</td>
<td>PowerVR SGX544MP2</td>
</tr>
<tr>
<td><strong>Memory (SDRAM)</strong></td>
<td>1 GB DDR3 (shared with GPU)</td>
<td>1 GB DDR3 (shared with GPU)</td>
</tr>
<tr>
<td><strong>Storage</strong></td>
<td>SD max. 64 GB</td>
<td>Micro SD max. 64 GB</td>
</tr>
<tr>
<td><strong>Ethernet</strong></td>
<td>10/100/1000 Ethernet RJ45</td>
<td>10/100/1000 Ethernet RJ45</td>
</tr>
<tr>
<td><strong>Wi-Fi</strong></td>
<td>N/A</td>
<td>802.11 b/g/n</td>
</tr>
<tr>
<td><strong>Video Outputs</strong></td>
<td>HDMI, CVBS, LVDS/RGB</td>
<td>HDMI, CVBS, LVDS/RGB</td>
</tr>
<tr>
<td><strong>Audio Output</strong></td>
<td>3.5 mm Jack and HDMI</td>
<td>3.5 mm Jack and HDMI</td>
</tr>
<tr>
<td><strong>Power Source</strong></td>
<td>5 volt via MicroUSB (DC In Only) and/or MicroUSB (OTG)</td>
<td>5 volt via MicroUSB (DC In Only) and/or MicroUSB (OTG)</td>
</tr>
<tr>
<td><strong>USB 2.0 Ports</strong></td>
<td>2x</td>
<td>4x</td>
</tr>
<tr>
<td><strong>IR Receiver</strong></td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td><strong>Buttons</strong></td>
<td>Power, Reset, UBoot</td>
<td>Power, UBoot</td>
</tr>
</tbody>
</table>

**GPIO pins**

The general purpose input output (GPIO) pins are one of the best features of both the Raspberry and Banana Pi boards. These pins are what allow your software to interact with the physical world much like working with an Arduino. These pins can be interfaced with, via code right from within the operating system running on the board. The pins can be used to interface with sensors, displays, touch screens, and the layout is exactly the same as the Raspberry Pi; thus, basically all the accessories are compatible with the Banana Pi.
CON1 is the extensible on-board CSI connector. It is a 40-pin FPC connector that can be used to connect the external camera module. CON2 is an LVDS display connector. CON3 contains the CAN bus, SPI bus, I2C bus, PWM, serial port, and so on. It can be configured to be used for all kinds of peripherals. The table viewable from the link contains the CON3 pin definitions. This is a reference for future use as you will need to know which pin is used for what.

**CON3 GPIO pin definitions**

The definitions for the GPIO pins can be located on the official website. These are quick and easy references and will benefit you when working with the Banana Pi. This following link will get you to the tables: [http://www.bananapi.com/index.php/component/content/article?layout=edit&id=24](http://www.bananapi.com/index.php/component/content/article?layout=edit&id=24).
J11 contains the default serial port. J12 also contains a serial port UART. UART0 (UART0-RX, UART0-TX) is configured for console input/output. This is useful if you want to log in using the serial port, which makes it the most commonly used PIN.

UART stands for Universal Asynchronous Receiver/Transmitter. This is hardware that translates data between serial and parallel sources. This means that those pins are directly connected to a piece of hardware that translates this data. This is used for directly connecting to the board and diagnostics, among other things.

The GPIO pins represent all of the interfaces you may need while working with the Banana Pi. This includes connecting over serial, and even powering the board. These pins allow you to interact with physical components through code you have written on the board. This means you can, for example, signal an alarm when a moisture sensor detects that your plants are dry, or write a row in your database when your beer has reached a certain stage while brewing.

GPIO is not specific to the Raspberry or Banana Pi; this represents any general pins that are on a board. This means that these pins are merely input and output and are there for the sake of allowing you to work with them. The following image shows the GPIO pins in relation to the actual device:
Operating systems

The Banana Pi supports many different operating systems. Because of its ARM v7 processor, it also runs operating systems such as Android. The list is actually quite long but currently there are several ported Raspberry Pi builds of Linux, which includes Kano and Raspbian as well as an optimized build called Bananian. There are also several ARM-based Linux builds targeted at the Banana Pi to choose from. These builds include OpenSUSE, Arch, Fedora, Lubuntu, and even Kali Linux. There are also two versions of Android available: 4.2.2 Jelly Bean and 4.4 KitKat.

The choice in operating systems is part of the reason the Banana Pi is so versatile. The operating system you choose may impact your project; something like Android would make a great media PC, whereas Arch or Debian would make a better file server. The choice is there though and all are available for free from the official website at http://www.bananapi.com/index.php/download.

The most common choice in operating systems for the Banana Pi is the Bananian port of Raspbian. This is due to the popularity of Raspbian for the Raspberry Pi. Basically every Raspbian-based project for the Raspberry Pi is directly compatible with the Banana Pi. This means that, by sticking with Bananian or Raspbian, you immediately have access to loads of tutorials and projects. Raspbian Linux is based on Debian Linux; this is one of the most popular distributions in the world.
Raspberry Pi accessories
The Banana Pi is made to be compatible with Raspberry Pi accessories. This is of course in relation to boards that are attached directly.

The GPIO layout of the M1 Banana Pi is identical to the Raspberry Pi Model A and B (classic). That means that boards such as the PiFace command and control will directly fit on the board and function just as if it was connected to a Raspberry Pi. The Banana Pi boards have a different layout than the Raspberry Pi boards, which means that you can't use your Raspberry Pi cases for the Banana Pi and vice versa.

Getting started
Now that you know about the hardware aspects of the different Banana Pi boards and their differences from, and compatibilities with, Raspberry Pi boards, you can start to plan out your projects.
Setting up your environment

There are a number of ways that you can set up your actual work environment for working with the Banana Pi. Depending on the board you are using, there are various cables that you will need to get started, some of which can later be removed depending on the project you are working on.

As you read earlier, the different boards have different layouts. There are some common cables that you will need to get started. Both versions of the Banana Pi have an HDMI out interface. Generally, no matter what the project, you are going to connect it via HDMI even if it is just to get the project started.

Both boards can be powered over USB, although the M2 does have a 5V power jack slot that can be used to power the board as well. The USB cable needed to power the Banana Pi is a standard Micro USB cable that is similar to, if not exactly the same as, the one you may use to charge your cell phone. You can connect a USB keyboard and mouse to either board, and of course you will need an SD card to hold the operating system. The essential hardware you are going to need to work with the Banana Pi is:

- HDMI cable
- Ethernet cable
- Wi-Fi dongle for the M1 (the M2 has built-in Wi-Fi)
- USB keyboard and mouse
- SD card (Micro for M2)
Once you have set up the initial board, you will be able to downsize some of this hardware. If you plan on using SSH to connect to the board, you won’t need a display, keyboard, or mouse, which leaves you with a very small footprint.

**Powering the boards**

There is more than one way to power the boards. In fact, there are three ways:

- Micro USB through the OTG Port
- 5V connector for the M2
- Power through the GPIO

The most common way to power the board is through the micro USB port. This is usually because we all seem to have a surplus of cables lying around and they generally tend to be USB from hardware such as cell phones and controllers. The M2 also comes equipped with a 5V DC connector that is new to the Banana Pi.

One way you can power your project that may be right for you is over GPIO. The pins are connected to the 5V rail, which means you can supply power to the pins. This means you can wire up whatever sensors and electronics you need for your project, as well as a direct power line. This is useful if you are planning on doing something battery- or solar-powered. The +5V supplied from the USB connector is filtered to give a stable 5V supply to the 5V0 Rail.
There are implications to powering the board this way, though. Although you can supply 5V of power though the GPIO pins, there is no protection for the board. This is also true if you use the 3.3V pins. This means that the board is not protected from spikes and you run the risk of a fried board. It is really recommended that you stick to micro USB or the 5V connector unless you are prepared to build in that protection before supplying power. Never power the device through GPIO pins AND micro USB at the same time. Always use only one power source.

Summary

With the information we have covered so far you should be able to start doing some projects. From here you will need to prepare your SD card with the operating system of choice and gather the supplies you will need to get started.

You also have the tools necessary to start working with the GPIO pins as well as a layout of the actual pins. This will help you later on when we start to do some interesting stuff with sensors and electronics.

In the next chapter, we are going to cover some of the software, options, and the process of installing the operating system onto the SD card for our Banana Pi. We will cover Windows, Linux, and Mac OS X as the process varies depending on your computer's operating system.
Where to buy this book
You can buy Learning Banana Pi 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.