Banana Pro Blueprints

Banana Pro is an open source hardware platform to run different open source Linux-based operating systems. Banana Pro is an upgrade version of the original Banana Pi with dual core processor, Android 4.2, and Wi-Fi on board.

This book follows a tactical plan that will guide you through the implementation of Banana Pro and its configurations. You will then learn the various programming languages used with Banana Pro with the help of examples. In no time at all, you'll be working on a wireless project that implements AirPlay servers, hotspots, and so on. Following this, you'll develop a retro-style arcade kiosk game. Then we'll move on to exploring the multimedia features of Banana Pro by designing and building an enclosure for it. After this, you'll learn to build a remote-controlled smart car and we'll examine how to control a robotic arm. The book will conclude with the creation of a home sensor system that has the ability to expand or shrink to suit any home.

Who this book is written for

This book is for all embedded board enthusiasts who want to use their credit-card-sized computer boards for extraordinary projects. This book will help you get the most out of your embedded board using it in your daily lives or for challenging projects.

What you will learn from this book

- Remotely connect to Banana Pro and program the embedded board
- Use Banana Pro as a hotspot or provide an AirPlay server for wireless audio transmission
- Find out about the different programming languages that can be used with Banana Pro
- Build and program your own multimedia center in order to watch television and movies
- Connect peripherals, such as a camera, LCD, or hard disk, to Banana Pro
- Manage and regulate your Linux system with Banana Pro
- Stream music wirelessly from your mobile phone to Banana Pro

Leverage the capabilities of Banana Pro with exciting real-world projects
In this package, you will find:

- The author’s biography
- A preview chapter from the book, Chapter 1 'Introduction to Banana Pro'
- A synopsis of the book’s content
- More information on **Banana Pro Blueprints**
About the Authors

**Ruediger Follmann** was born in 1968 in Germany. He studied electrical engineering at RWTH Aachen, Germany, and received his PhD from the University of Duisburg-Essen, Germany. He has worked for IMST GmbH, Germany, for more than 20 years, where he heads the RF circuits and system integration department. IMST is a design and development company with more than 170 employees. He uses embedded boards in many different projects, for example, in order to control MMICs or hybrid electronics. Follmann is the author of many technical articles as well as *Das Raspberry Pi Kompendium, Springer*.

**Tony Zhang** was born in 1990 in China. He studied control science and engineering at HIT, China, and received his master's degree in 2015. Since 2014, Tony has been working with LeMaker community, where he is the cofounder and is heading the R&D department.
Preface

The book mainly focuses on some popular applications and projects using Banana Pro. It first introduces the basic usage of Banana Pro, including its hardware and software, and then some applications, such as a multimedia center and laser engraver. The sole purpose of this book will be to show what you can do with Banana Pro through a number of projects, ranging from home automation projects, cameras around the house, and robotics.

This book follows a tactical plan that will guide you through the implementation of Banana Pro and its configurations. You will then learn the various programming languages used with Banana Pi with the help of in-depth examples.

What this book covers

Chapter 1, Introduction to Banana Pro, introduces the Banana Pro single board computer. It explains all its available interfaces as well as the installation of the Linux OS, which will be used in this book.

Chapter 2, Programming Languages, explains several programming languages, such as Python, C/C++, and Scratch. In several examples, the usage of these programming languages is shown through a step-by-step approach.

Chapter 3, Wireless Projects, introduces wireless projects. It shows how Banana Pro can be used as a wireless hotspot or print server. Additionally, the serving of web pages is explained in combination with content management systems. Finally, a measurement server is set up in order to control an electronic circuit.
Chapter 4, *An Arcade Cabinet*, describes how to turn Banana Pro into an arcade cabinet. A step-by-step approach shows you how to install and configure different game emulations. This chapter also describes the steps required to set up a cabinet house with the help of a joystick and an LCD display for Banana Pro.

Chapter 5, *A Multimedia Center*, introduces the usage of Banana Pro as a multimedia center. You will be shown how a DVB receiver is set up through a series of steps. This receiver allows the recording of TV transmissions, watching DVDs, and listening to audio CDs.

Chapter 6, *Remote Controlling a Smart Monitor Car*, teaches you how to make a small mobile car with the remote monitor function. It describes how to realize the IP camera function on Banana Pro. It also shows you how to set up a small mobile car. Finally, you will see how a web page can be used to control the car and view the video of the camera in the car.

Chapter 7, *A Laser Engraver*, shows the laser engraver made by Banana Pro along with a laser CPU. You will learn how to design the frame of the laser engraver and use different kinds of software on the Banana Pro. The laser CPU and PC can be used together to engrave patterns on some special types of material such as cardboard.

Chapter 8, *Scratch – Building a Smart House*, describes how to use Scratch and some other GPIO libraries together to control different sensors. Then, you will learn how to combine all the sensors together in a virtual small house to simulate the smart house.
Introduction to Banana Pro

Embedded boards are part of our lives. We use them daily, and sometimes even without recognizing or knowing about them. They are included in mobile phones, washing machines, or implemented in our televisions. These little computers work in the background and consume very little power.

This chapter introduces Banana Pro. Banana Pro is an updated version of Banana Pi that’s been designed by the LeMaker team (http://www.lemaker.org) from China. LeMaker is a charitable organization and aims to provide educational services by selling its Banana boards and building a community around this technology, similar to what the Raspberry Pi foundation in the UK has done.

Banana Pro is the current flagship of LeMaker. It is compatible with many Linux-based operating systems including Android. In this chapter, Banana Pro is introduced to you along with its connection possibilities. The most popular operating systems in the market today are explained here and the principal installation as well as the boot process is described.

There are already many add-ons available for Banana Pro, such as LCD modules in various sizes, a camera module, or modules that connect to GPIO pins. Sections of this chapter deal with the detailed installation of an LCD module, describing the step-by-step installation and operation of a camera module and provide background information on additional add-ons.

In particular, this chapter includes the following sections:

- Banana Pro
- Getting started
- Operating systems
- Add-ons
Banana Pro

Since Raspberry Pi was born, a lot of new embedded boards have seen the light of day. A very popular one is Banana Pro. It came into the market in late 2014, is affordable ($45 USD), and offers many interfaces (Figure 1).

Banana Pro is the size of a credit card (92 mm x 60 mm and 48 g) and uses an Allwinner A20 system on chip. This includes a dual core CPU (ARM Cortex-A7, 1 GHz), a Mali 400 MP2 GPU, and a 1 GB SDRAM. Additionally, there is also a AP6181 Wi-Fi module onboard. Banana Pro makes use of the AXP209 Power Management Unit (PMU), which is very power efficient and allows power monitoring.

![Figure 1: Banana Pro is a dual core computer featuring many connectors (source: http://en.wikipedia.org/wiki/Banana_Pro).](image)
Specifications of Banana Pro

Table 1 summarizes the Banana Pro specifications. Everything is available to build a complete computer system, including a hard disk or connection to a television or computer screen. Both the Gb LAN and WLAN guarantee easy integration into networks and fast operation, for example, as NAS. USB sockets allow the usage of a keyboard, mouse, or even DVB receivers (refer to Chapter 4, An Arcade Cabinet). The 40-pin General Purpose Input Output (GPIO) header allows arbitrary experiments that feature different bus systems, such as CAN (Controller Area Network) or SPI (Serial Peripheral Interface). Different LEDs provide board statuses; CSI (Camera Serial Interface) and LVDS (Low Voltage Differential Signaling) interfaces are also available. The CSI interface can be used in order to connect a camera and LVDS interface that can be used for display connection purposes. Even an onboard microphone is implemented. The following are the Banana Pro specifications:

<table>
<thead>
<tr>
<th>Components</th>
<th>Specification</th>
</tr>
</thead>
<tbody>
<tr>
<td>System on chip</td>
<td>Allwinner A20, SATA 2, two USBs, and one Micro-USB</td>
</tr>
<tr>
<td>CPU</td>
<td>1 GHz Cortex-A7 dual-core</td>
</tr>
<tr>
<td>GPU</td>
<td>Mali 400 MP2, OpenGL 2.0/1.1 (hardware accelerated)</td>
</tr>
<tr>
<td>Memory</td>
<td>1 GB DDR3 SDRAM</td>
</tr>
<tr>
<td>Power</td>
<td>5V/2A using Micro-USB (DC and/or USB OTG)</td>
</tr>
<tr>
<td>PMU</td>
<td>AXP209</td>
</tr>
<tr>
<td>USB</td>
<td>Two USBs of the 2.0 version and one USB 2.0 OTG</td>
</tr>
<tr>
<td>Low level connectors</td>
<td>A 40-pin GPIO header, including UART, I2C, SPI, PWM, CAN, I2S, and SPDIF</td>
</tr>
<tr>
<td>Storage</td>
<td>A Micro-SD card, SATA 2.0, and a 2.5 inch hard disk power supply onboard</td>
</tr>
<tr>
<td>Network</td>
<td>10/100/1000 Ethernet RJ 45 and 150 Mbps Wi-Fi 802.11 b/g/n</td>
</tr>
<tr>
<td>Display</td>
<td>HDMI-A 1.4, a composite video (PAL and NTSC) (via 3.5 mm TRRS jack shared with an audio output), and LVDS/RGB/CPU display interface (DSI) for raw LCD panels. 11 HDMI resolutions from 640×480 to 1920×1080 plus various PAL and NTSC standards</td>
</tr>
<tr>
<td>Video</td>
<td>An HD H.264 2160p video decoding. A multiformat FHD video decoding, including Mpeg1/2, Mpeg4, H.263, H.264, and so on. An H.264 high-profile 1080p@30fps or 720p@60fps encoding</td>
</tr>
<tr>
<td>Camera</td>
<td>An 8-bit parallel camera interface</td>
</tr>
<tr>
<td>Audio outputs</td>
<td>HDMI, analog audio (via a 3.5 mm TRRS jack that's shared with composite video out), I2S audio (also potentially for audio input), and an electrical SPDIF audio output</td>
</tr>
<tr>
<td>Audio input</td>
<td>An onboard microphone</td>
</tr>
</tbody>
</table>
Introduction to Banana Pro

<table>
<thead>
<tr>
<th>Components</th>
<th>Specification</th>
</tr>
</thead>
<tbody>
<tr>
<td>Buttons</td>
<td>A reset button, power button, and a U-boot button</td>
</tr>
<tr>
<td>LEDs</td>
<td>A power status LED (red), Ethernet status LED (blue), and a user-defined LED (green)</td>
</tr>
<tr>
<td>Other</td>
<td>An onboard IR receiver</td>
</tr>
<tr>
<td>Dimensions</td>
<td>92 mm x 60 mm and 48 gm</td>
</tr>
</tbody>
</table>

Table 1: Banana Pro specifications

Contrary to many other embedded boards, Banana Pro offers a SATA connector for the direct connection to a 2.5 inch hard disk connection. Moreover, for SSDs or a 2.5 inch hard disk, a power connector is onboard.

Note the correct polarity of the SATA/DC power connector. Compared to CubieBoard or Cubietruck, Banana Pro uses the concept of changed polarity. Therefore, when ordering a Banana Pro SATA/DC cable, be sure of the DC polarity (Figure 2).

![Figure 2: The Banana Pro SATA/DC cable (source: Amazon.com)](source: Amazon.com)

All the connectors of Banana Pro are explained in Figure 3. These are as follows:

- **Debug TTL UART**: This interface can be used in order to connect to a serial console PC/laptop via a serial USB cable for debugging purposes.
- **Display interface**: Use this connector to connect to an LCD. LeMaker offers three different sizes of LCDs that are up to 7 inches in size.
- **HDMI**: Using an HDMI (type A) or an HDMI (type A) cable (HDMI 1.4), this socket allows the connection to a monitor or television. In addition to this, digital audio can be transferred through this connector.
- **SATA 2.0 interface and SATA/DC 5V**: Banana Pro has a SATA 2.0 connector onboard. Furthermore, the DC for SSDs or 2.5 inch hard disks can be directly supplied (Figure 2).
• **The micro USB power:** Connect your 5V micro USB power supply to this connector in order to boot Banana Pro. While the 1.2A power supply may be sufficient to boot the board without a hard disk, I recommend at least a 2A power supply when the hard disk is connected to the board. Also, keep in mind that a 7 inch LCD connected to the board will consume an additional 750mA.

![Figure 3: Connectors of Banana Pro](image-url)
• **USB OTG**: This allows for switching between a host and target mode. When in host mode, power will be drawn from Banana Pro to a drive or USB device that’s connected. Many OTG implementations are currently limited and do not enable the powering of all devices.

![USB OTG to USB adapter](https://www.amazon.com)

Figure 4: The USB OTG to USB adapter (source: Amazon.com)

Using an adapter cable (*Figure 4*), USB devices can be connected to this port.

• **Reset button**: Pushing this button will reboot Banana Pro. Note that pushing this button during an operation may result in data corruption. It is recommended that you always shut down Banana Pro using the `sudo halt -p` command when using Linux.

• **Camera interface**: This connector allows the usage of the LeMaker camera.

• **Power button**: This button will boot Banana Pro after shutdown.

• **40-pin GPIO header**: This header provides different signals, such as SPI, CAN, I2S, I2C, UART, or SPDIF.

• **Microphone**: The microphone can be used for recordings (mono).

• **AV output**: This connector combines analog videos and analog audio in a so-called TRRS jack (*Figure 5*):
Figure 5: The TRRS connector for Banana Pro. Pin 1 is audio left, pin 2 is audio right, pin 3 is video out and pin 4 is ground.

Figure 6: The TRRS splitter for audio and videos

There are adapter cables available that separate audio and videos (Figure 6):
Introduction to Banana Pro

- **IR Receiver**: Banana Pro features an onboard IR receiver that receives remote control signals.
- **2 x USB**: Connect your USB devices here.
- **Ethernet interface**: Banana Pro provides a 1 Gb Ethernet interfaces that offers a true GB speed.
- **The microSD card slot**: Banana Pro boots from a microSD card since a direct boot from a hard disk is not possible. However, a root filesystem can be directed to the hard disk once it's booted from the microSD card.
- **FEL button**: FEL is a low-level subroutine that's contained in the BootROM on Allwinner devices. It is used for the initial programming and recovery of devices using a USB.

In addition to the preceding connectors, Banana Pro offers a WLAN module that can also work as an access point. Banana Pro does not supply a **real-time clock** (RTC). However, there are RTC add-on boards available at <http://www.wvshare.com/product/PCF8563-RTC-Board.htm>.

![ RTC add-on board for Banana Pro. Source: http://www.wvshare.com. ](image)
Banana Pro onboard LEDs

Banana Pro features several LEDs onboard. The red LED lights up as soon as DC power is supplied to the board. The blue LED shows Ethernet activity. The green and blue LEDs can be set individually. Chapter 2, Programming Languages, will show you how these LEDs can be programmed.

Getting started

LeMaker offers several predefined images for download. These can be found at http://www.lemaker.org/resources/9-38/image_files.html. Simply download the image file you want and copy it to a microSD card. The minimum required SD card size is 4 GB.

When using the Windows OS, WIN32 Disk Imager (http://sourceforge.net/projects/win32diskimager) can be used in order to transfer an image to an SD card. However, make sure that you select the correct drive for the Win32 Disk Imager. An incorrect drive may result in loss of data.

Using Linux, image files can be transferred to a Micro SD card using the following command:

`sudo dd bs=4M of=/dev/sdX if=/path/to/banana_pro.img`
Introduction to Banana Pro

For uncompressed images, use the following command

```
gzip -dc /home/your_username/image.gz | sudo dd bs=4M of=/dev/sdX
```

For compressed images, replace `/dev/sdX` with your SD card. Note that the block size is set to `4M` and will work most of the time. If it doesn’t, try `1M`, although this will take a considerably longer to complete.

There is one exception for Banana Pro using the Android operating system. Using `Win32 Disk Imager` or the `dd` command of Linux will not work here. Special software called `PhoenixCard` will be required. PhoenixCard for Windows can be downloaded from https://drive.google.com/file/d/0B_VynIqhAcB7NTg2UkRDdHRWX2s/edit. Before you do this, the SD card should be formatted by clicking on **Format to Normal** (Figure 9):

![PhoenixCard Screenshot](https://drive.google.com/file/d/0B_VynIqhAcB7NTg2UkRDdHRWX2s/edit)

**Figure 9:** Copying Android to an SD card for Banana Pro by first formatting the SD card to normal
Chapter 1

It is mandatory to select the **Startup** from the **Write Mode**. After formatting the card, click on the **OK** button in the Information window. In a second step, the Android image can be burned to the SD card (*Figure 10*). This step will take a few minutes to complete.

After writing any image to the microSD card, put the SD card into Banana Pro SD card slot and connect the DC power supply afterwards. This will boot your image. Currently, Banana Pro supports Android 4.4. Version 4.4 is available from this forum at [http://www.bananapi.com/index.php/forum/adroid/117-android-4-4-for-banana-pi-beta-1-release-note](http://www.bananapi.com/index.php/forum/adroid/117-android-4-4-for-banana-pi-beta-1-release-note).

![Figure 10: Copying the Android OS to an SD card for Banana Pro by burning an image to an SD card](image)

Set up your Banana Pro according to Figure 11 after you've copied the image to the SD card.
The first boot

Depending on the chosen image, the first boot will take 20 to 30 seconds. The first boot on Android will take much longer, so be patient.

Some Android versions may require two micro USB power connectors at the same time. One must be supplied to the micro USB DC before boot, and the other one to the micro USB OTG. During my experiments with Android 4.2, one power supply connection was sufficient.

Figure 11: Setting up Banana Pro before the first boot (source: http://www.lemaker.org/cn/product-bananapro-guide.html)
Chapter 1

Available operating systems for Banana Pro

This section presents some of the available operating systems for Banana Pro. Here, Android has briefly described some of the most important Linux distributions:

Figure 12: A screenshot of operating systems available on the LeMaker web page

LeMaker offers image versions for both Banana Pi and Banana Pro. However, be sure to download the correct image.

Figure 13
Android

Although Android is based on Linux, both architectures are quite different. This can be seen even in the example of low-level C language routines, which Google exchanged for Android. In addition to this, Google prohibits all code that's licensed under GPL (GNU Public License). For this reason, Android only implements a rudimentary shell featuring only a few shell commands. Due to the C-library incompatibilities, normal Linux code, even if it's especially compiled for the ARM architecture, will not run under Android. Nevertheless, Android is a very popular operating system (OS) that's used by many phones and tablets. It fully supports graphic hardware in terms of accelerated video and 3D graphics. For this reason, the first Banana Pro version of Kodi (formerly known as XBMC and Xbox Media Centre) made use of an accelerated Android graphics library (refer to Chapter 4, An Arcade Cabinet).

Applications for Android can be developed using a special software development kit (SDK) and Java itself as most Android applications are based in Java. Although this book concentrates on Linux projects rather than Android projects, we will take a short look at Android on Banana Pro.

LeMaker provides three different Android images: one for a 5 inch TFT, the second for a 7 inch TFT, and the third using the HDMI cable as a video output. To install an Android OS, simply download the required version. You most probably want to start with the HDMI enabled version. Use PhoenixCard (as seen in the previous section) in order to write the image to an SD card. Insert this SD card and boot Android. After a while, the start screen of Android will be visible.

Figure 14: This is Android 4.4 running on Banana Pi/Pro (http://www.bananapi-kaufen.de/wp-content/uploads/2014/07/800px-Android_v2.jpg)
Android can be controlled by using a keyboard and mouse. Bluetooth and WLAN will work out of the box and some applications are already preinstalled.

![Android 4.4 running on Banana Pro (source: bananapi.com)](Image)

**Figure 15: Android 4.4 running on Banana Pro (source: bananapi.com)**

**Linux**

There are a variety of Linux distributions available on the LeMaker download site (Figure 13). The most famous operating systems are Raspbian, Bananian, Lubuntu, and Gentoo. LeMedia, a special Linux distribution that provides the Xbox media center, Kodi (formerly known as XBMC), is not available on the official LeMaker download website. We will later on compile and install XBMC for other operating systems such as a Debian distribution. No matter which Linux OS you decide on incorporating, the procedure is always the same: Copy the image to an SD card and boot the card within Banana Pro.

After copying the image, the SD card will contain a directory called /boot. This directory contains the Linux kernel (called uImage; refer to Chapter 2, Programming Languages, for this), a file called uEnv.txt, and a binary file (usually script.bin or bananapro.bin). The uEnv.txt file contains kernel parameters, such as screen resolutions, and the binary file contains the complete settings for Banana Pro with respect to all connectors.
My uEnv.txt file looks like this:

```
console=tty1
root=/dev/mmcblk0p1 rootwait
extraargs=rootfstype=ext4 sunxi_ve_mem_reserve=190 sunxi_g2d_mem_reserve=16
    sunxi_fb_mem_reserve=32 hdmi.audio=EDID:1 disp.screen0_output_mode=1920x1080p50 panic=10 consoleblank=0
script=bananapro.bin
kernel=uImage
```

It defines /dev/tty1 as a console terminal, uses the first partition of an SD card as a root device (/dev/mmcblk0p1), and provides some additional kernel parameters. The video output mode is completely HD progressive (1920x1080p50). The binary file is called bananapro.bin in my case, and the kernel is uImage within the /boot directory. The function of the binary file is described in the next section (the FEX file).

For my experiments, I've used a Banana Pi Linux image of Igor Pečovnik (http://www.igorpecovnik.com/2014/09/07/banana-pi-debian-sd-image/), which I adapted to Banana Pro. In the meantime, the Banana Pi image is compatible with Banana Pro.

I recommend that you download the Debian Wheezy version that comes with kernel 3.4. Although this kernel is not up to date, it enables all the hardware features of Banana Pro as opposed to kernel 4.2, which, for example, neither has an audio driver nor an accelerated video output.

The image itself is based on Debian Wheezy and comes with kernel 3.4. Similar to Bananian, this distribution does not come with any X11 server or display manager. Therefore, I've installed the MATE Windows manager and the slim login manager. Similar to the X11 framebuffer driver, I've used sources from https://github.com/ssvb/xf86-video-fbturbo. This driver requires a number of packages to be installed first as follows:

```
sudo apt-get install git build-essential xorg-dev xutils-dev x11proto-dri2-dev
```

```
sudo apt-get install libldl-dev libtool automake libdrm-dev autoconf
```

Now, acquire the sources of xf86-video-fbturbo, compile them, and then install them as follows:

```
sudo su
```

```
cd /usr/local/src
```
git clone https://github.com/ssvb/xf86-video-fbturbo.git
cd xf86-video-fbturbo
autoreconf -vi
./configure --prefix=/usr
make
make install

Afterwards, copy the default xorg.conf file to the location of your xorg.conf like this:
sudo cp xorg.conf /etc/X11/xorg.conf

Do not forget to back up any existing xorg.conf files first. In Chapter 4, An Arcade Cabinet, we will install accelerated Mali drivers for a desktop in addition to a framebuffer turbo driver.

![Figure 16: The MATE desktop environment (source: mate-desktop.org)](image)

MATE needs to be installed from the Debian Wheezy backports. Add the following to the /etc/apt/sources.list file. This can be done using this command:
sudo nano /etc/apt/sources.list
Afterwards, run the following shell commands:

```
sudo apt-get update
sudo apt-get upgrade
apt-get install mate-desktop-environment mate-desktop-environment-extras
sudo apt-get install slim
```

In order to install the MATE desktop and `slim`, a configuration can be done in the `/etc/slim.conf` file. I've added/changed the following lines:

```
sessionstart_cmd xhost + && xset s noblank && xset s off && xset –dpms
default_user pi
auto_login yes
```

The first line disables access control to the X-windows server. This is needed later on in Chapter 4, An Arcade Cabinet. In addition to this, screensavers and power management for the X-server are switched off. The second line sets the default user as the `pi` user. The last line allows the `pi` user to automatically log in without providing a password.

This user `pi` can be added to the system using this shell command:

```
sudo adduser
```

You need to answer all the questions prompted by the system and use the `pi` as a username. Most Linux distributions provide an already secure shell access (`ssh`). If not, simply install the `openssh` server by typing the following command that will enable a remote login with `ssh` on port 22:

```
sudo apt-get install openssh-server
```

## The FEX file

Banana Pro reads interface settings from a binary file called `script.bin` or `bananapro.bin` that is located in the `/boot` directory. This file defines video output, GPIO selection, and much more. Each value consists of a `key = value` pair combination under a `[sectionheader]`. All three entries are case-sensitive. A complete selection of all possible entries can be found at [http://linux-sunxi.org/Fex_Guide](http://linux-sunxi.org/Fex_Guide).

The FEX files can be compiled into a binary file (and recompiled into an ASCII FEX file) using the `fex2bin` or `bin2fex` commands. These tools are available on GitHub ([sunxi-tools](https://github.com/linux-sunxi/sunxi-tools)). The `sunxi-tools` require `libusb` as a prerequisite.
Additionally, *git* will be installed. Git can be used in order to clone a directory that's available on the Internet on a local SD card or hard disk using this command:

```
sudo apt-get install libusb-1.0-0-dev git
```

Afterwards, clone the *sunxi-tools* repository and then build and install the tools:

```
git clone https://github.com/linux-sunxi/sunxi-tools
cd sunxi-tools
make
sudo make install
```

The syntax of *fex2bin* and *bin2fex* is easy. Simply provide two filenames (the original and the converted one, respectively). Take a look at the following example:

```
sudo fex2bin /boot/script.fex /boot/script.bin
```

This will generate a new *script.bin* binary file that's based on the *script.fex* ASCII file in the */boot* directory.

```
If uEnv.txt redefines, for example, display settings as compared to script.bin, this file will be preferred over script.bin.
```

A typical display section in the FEX file looks like this:

```
[disp_init]
disp_init_enable = 1
disp_mode = 0
screen0_output_type = 3
screen0_output_mode = 4
screen1_output_type = 2
screen1_output_mode = 11
fb0_framebuffer_num = 2
fb0_format = 10
fb0_pixel_sequence = 0
fb0_scaler_mode_enable = 0
fb1_framebuffer_num = 2
fb1_format = 10
fb1_pixel_sequence = 0
fb1_scaler_mode_enable = 0
```
Introduction to Banana Pro

\begin{verbatim}
  lcd0_backlight = 197
  lcd1_backlight = 197
  lcd0_bright = 50
  lcd0_contrast = 50
  lcd0_saturation = 57
  lcd0_hue = 50
  lcd1_bright = 50
  lcd1_contrast = 50
  lcd1_saturation = 57
  lcd1_hue = 50
\end{verbatim}

The section header of the FEX file is `[disp_init].disp_init_enable`, the value of which can be either 1 (enabled display output) or 0 (disabled display output). The display mode is a value between 0 and 4 according to the following table:

<table>
<thead>
<tr>
<th>Mode</th>
<th>Display mode</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>screen0 (screen0, fb0)</td>
</tr>
<tr>
<td>1</td>
<td>screen1 (screen1, fb0)</td>
</tr>
<tr>
<td>2</td>
<td>dualhead (screen0, screen1, fb0, fb1) (two screens and two framebuffers)</td>
</tr>
<tr>
<td>3</td>
<td>xinerama (screen0, screen1, fb0) (two screens and one big framebuffer)</td>
</tr>
<tr>
<td>4</td>
<td>clone (screen0, screen1, fb0) (two screens and one standard framebuffer)</td>
</tr>
</tbody>
</table>

Table 2: The display modes for Banana Pro

The `screen0_out_color_range` mode defines the output color for HDMI. It applies to both screen0 and screen1. Therefore, there is no `screen1_out_color_range`. The output color range can be set according to the following table:

<table>
<thead>
<tr>
<th>Type</th>
<th>Output color range</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>16-255 is the default limited range</td>
</tr>
<tr>
<td>1</td>
<td>0-255 is the full range (PC level)</td>
</tr>
<tr>
<td>2</td>
<td>16-235 is the limited range (video level)</td>
</tr>
</tbody>
</table>

Table 3: The output color range for Banana Pro
The `screen0_output_type` mode sets the output type for `screen0` according to the following table. Therefore, use `screen0_output_type` accordingly:

<table>
<thead>
<tr>
<th>Type</th>
<th>Output type</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>None</td>
</tr>
<tr>
<td>1</td>
<td>LCD</td>
</tr>
<tr>
<td>2</td>
<td>TV</td>
</tr>
<tr>
<td>3</td>
<td>HDMI</td>
</tr>
<tr>
<td>4</td>
<td>Analog video</td>
</tr>
</tbody>
</table>

Table 4: The output types for Banana Pro

The following screen output modes (`screen0_output_mode` and `screen1_output_mode`) can be defined for `screen0` and `screen1`:

<table>
<thead>
<tr>
<th>Mode</th>
<th>Used for the TV/HDMI output</th>
<th>Used for analog video output</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>480i</td>
<td>1680 x 1050</td>
</tr>
<tr>
<td>1</td>
<td>576i</td>
<td>1440 x 900</td>
</tr>
<tr>
<td>2</td>
<td>480p</td>
<td>1360 x 768</td>
</tr>
<tr>
<td>3</td>
<td>576p</td>
<td>1280 x 1024</td>
</tr>
<tr>
<td>4</td>
<td>720p50</td>
<td>1024 x 768</td>
</tr>
<tr>
<td>5</td>
<td>720p60</td>
<td>800 x 600</td>
</tr>
<tr>
<td>6</td>
<td>1080i50</td>
<td>640 x 480</td>
</tr>
<tr>
<td>7</td>
<td>1080160</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>1080p24</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>1080p50</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>1080p60</td>
<td>1920 x 1080</td>
</tr>
<tr>
<td>11</td>
<td>PAL</td>
<td>1280 x 720</td>
</tr>
<tr>
<td>14</td>
<td>NTSC</td>
<td></td>
</tr>
</tbody>
</table>

Table 5: The screen output modes for Banana Pro
The fb0_framebuffer_num value is the number of framebuffers to be used for the 0 framebuffer. It can be set to 2 for double buffering. Use fb1_framebuffer_num accordingly. The pixel format for both framebuffers can be defined as fb0_format or fb1_format according to this table:

<table>
<thead>
<tr>
<th>Format</th>
<th>fb0_format</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>RGB655</td>
</tr>
<tr>
<td>5</td>
<td>RGB565</td>
</tr>
<tr>
<td>6</td>
<td>RGB556</td>
</tr>
<tr>
<td>7</td>
<td>ARGB1555</td>
</tr>
<tr>
<td>8</td>
<td>RGBA5551</td>
</tr>
<tr>
<td>9</td>
<td>RGB888</td>
</tr>
<tr>
<td>10</td>
<td>ARGB8888</td>
</tr>
<tr>
<td>12</td>
<td>ARGB4444</td>
</tr>
</tbody>
</table>

Table 6: The framebuffer format for Banana Pro

The framebuffer pixel sequences (fb0_pixel_sequence and fb1_pixel_sequence) must be set to 0 for Linux and 2 for Android operating systems (refer to this table):

<table>
<thead>
<tr>
<th>Sequence</th>
<th>fb0_pixel_sequence or fb1_pixel_sequence</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>ARGB</td>
</tr>
<tr>
<td>2</td>
<td>BGRA</td>
</tr>
</tbody>
</table>

Table 7: The framebuffer pixel sequence for Banana Pro

The Framebufferscaler mode can be enabled or disabled for both framebuffers (fb0_scaler_mode_enable and fb1_scaler_mode_enable). Use a value of 0 to disable the scaler mode and a value of 1 to enable the scaler mode. TheScaler mode may improve the picture quality of some settings. The lcd0_backlight is a value between 0 and 240 and sets the PWM rate on a selected PWM GPIO. The same is true for lcdn_backlight. All the remaining settings are self-explanatory.

LeMaker provides special configuration files that can be used with LCD screens (refer to The LCD module section).
Transferring an OS to a hard disk

Banana Pro cannot boot directly from a hard disk. This section will show you how the Linux operating system can be copied from a microSD card to a hard disk and how a system can boot into the OS of the hard disk.

First, boot your Banana Pro with a connected hard disk (SATA and DC). The kernel messages will show the SATA device (usually `/dev/sda`). In order to get a filesystem overview, the `lsblk` command can be used. This command will list the complete filesystem including the attached hard disk.

Verify the contents of the hard disk using the following command:

```
fdisk -l /dev/sda
```

Principally, the available hard disk space must be equal to or larger than the size of the SD card in order to host the OS. The `fdisk` command can be used to partition the disk in terms of erasing or adding hard disk partitions. Use this program with care as it can delete everything that exists on your hard disk. The following steps assume an empty hard disk or a hard disk whose content is not needed any more.

The hard disk can be partitioned using the `fdisk` command:

```
sudo fdisk /dev/sda
```

The `p` (print) command will print partition table:

```
Command (m for help): p
```

Existing partitions can be removed using the `d` (delete) command:

```
Command (m for help): d
Partition number (1-6): 1
```

New partitions can be added using the `n` (new) command:

```
Command (m for help): n
Partition type:
  p  primary (0 primary, 0 extended, 4 free)
  e  extended
Select (default p): p
Partition number (1-4, default 1): 1
```
The w (write) command writes all changes to the disk:

```
Command (m for help): w
The partition table has been altered!
```

After this, the `fdisk` write command will automatically quit. If no other partitions (for example, swap) are required, you can press `RETURN` when prompted for the first and last sectors. In its simplest sense, a swap partition serves as an overflow to Banana Pro’s memory (RAM). It might be useful when compiling huge software packages such as Kodi (formerly known as XBMC). As an alternative, a file on a hard disk or SD card can be used for swapping. A swap partition itself has the 82 partition ID, which can be changed using the `t` (toggle) command followed by the ID. Swap partitions can be formatted using the following command:

```
mkswap /dev/sdaX,
```

x represents the swap partition (for example, 6). Swap partitions will be activated as follows:

```
swapon /dev/sdaX
```

New (empty) partitions must be formatted before usage. The following command formats the first partition (`sda1`) with the ext4 filesystem and labels it as `rootfs`. In doing so, all existing data on this partition will be erased. Therefore, use this command with caution and only format empty partitions or partitions whose content is not needed any more:

```
sudo mke2fs -t ext4 -L rootfs /dev/sda1
```

In this step, the complete data from the microSD card will be copied to the hard disk. In our example, the first partition of the hard disk will be mounted into the `/mnt` directory; `rsync` will be used to transfer all data to the directory:

```
sudo mount /dev/sda1 /mnt
sudo rsync -ax / /mnt
```
The complete copy process will take a while. In the last step, the boot parameters of the uEnv.txt file will be adjusted to use the hard drive. Change `root=/dev/mmcblk0p1` in the `/boot/uEnv.txt` file to this:

```
root=/dev/sda1
```

This can be done using the `nano` editor by typing this:

```
sudo nano /boot/uEnv.txt
```

Finally, restart Banana Pro with the following command:

```
sudo reboot
```

The system should be redirected from the microSD card to the hard disk right now. A `df -h` command on my system delivers the following (320 GB of free hard disk space):

```
Filesystem      Size  Used  Avail  Use% Mounted on
rootfs          289G  157G  118G   57%  /
/dev/root       289G  157G  118G   57%  /
devtmpfs        380M     0   380M    0%  /dev
tmpfs           128M  280K   128M    1%  /run
tmpfs           128M  2,3M   126M    2%  /run/shm
tmpfs           1,0G  4,0K   1,0G    1%  /tmp
/dev/mmcblk0p1  14G   11G   2,4G   83%  /media/lime-next
```
If you want to make any changes to uEnv.txt after booting to the hard disk, you will need to mount the SD card first, change into the SD card boot directory, and edit uEnv.txt accordingly:

```
sudo mount /dev/mmcblk0p1 /mnt

cd /mnt/boot

sudo nano uEnv.txt
```

The same applies to the installation of a new kernel (refer to Chapter 2, Programming Languages).

The `gparted` software is a graphical frontend for `fdisk` and other tools. It is very useful to add, delete, or even resize partitions. It may later be required for backups, where it can be used to shrink a partition, for example, on an SD card in order to use `windisk32imager` to copy the contents of the SD card to the backup medium.

![GParted](image)

*Figure 17*
Add-ons
There are a lot of add-ons available for Banana Pro. LeMaker offers LCD modules in three different sizes from 3.5 inches up to 7 inches. Moreover, LeMaker offers a camera board that can be directly connected to Banana Pro. Last but not least, several cases are available that allow the mounting of an WLAN antenna, while others even provide a hard disk housing.

The LCD module
This table shows the resolution of LeMaker LCDs depending on their panel size:

<table>
<thead>
<tr>
<th>LCD panel size</th>
<th>Resolution</th>
</tr>
</thead>
<tbody>
<tr>
<td>3 to 5 inches</td>
<td>320 x 240 pixels</td>
</tr>
<tr>
<td>5 inches</td>
<td>640 x 480 pixels</td>
</tr>
<tr>
<td>7 inches</td>
<td>1024 x 600 pixels</td>
</tr>
</tbody>
</table>

Table 8: The LeMaker LCD panel size resolutions.

In dependence of the LCD panel size, LeMaker offers three different binary files that can be used in order to initialize the LCDs while booting (script.bin). The files are available from GitHub ([https://github.com/LeMaker/fex_configuration](https://github.com/LeMaker/fex_configuration)). This GitHub tree provides both FEX files and compiled FEX files (bin files). Be sure to download the ones for Banana Pro as Banana Pi files are also available. Banana Pro files are named `banana_pro_Xlcd.bin`, whereas, `X` represents the LCD panel size (3 to 5, 5, or 7 inches, respectively). Copy this file to the first partition of your microSD card and rename it as `script.bin` (or whatever your bin file is called).

Do not overwrite bin files without creating a backup first. There are a lot of settings included in these files, and once the files are lost, changes must be applied step by step again manually.

When operating the LCD display, be sure that your power supply can deliver enough current. While a 2A supply may be sufficient to operate Banana Pro and a 2.5 inch hard disk, the display will draw another 750mA. In this case, a 5V/3A supply is required.
The 7 inch LCD module data is summarized in the following table:

<table>
<thead>
<tr>
<th>Specification</th>
<th>Details</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>LCD size</strong></td>
<td>7.0 inch (diagonal)</td>
</tr>
<tr>
<td><strong>Interface</strong></td>
<td>Parallel LVDS of 8-bits</td>
</tr>
<tr>
<td><strong>Resolution</strong></td>
<td>024 x 3 (RGB) x 600</td>
</tr>
<tr>
<td><strong>Driver element</strong></td>
<td>a-Si TFT active matrix</td>
</tr>
<tr>
<td><strong>Dot pitch</strong></td>
<td>0.05(W) x 0.15(H) mm</td>
</tr>
<tr>
<td><strong>Connections to Banana Pro</strong></td>
<td>40-pin FPC to the Display Sensor Interface (DSI)</td>
</tr>
<tr>
<td><strong>Surface treatment</strong></td>
<td>Glare</td>
</tr>
<tr>
<td><strong>Color arrangement</strong></td>
<td>RGB-stripe</td>
</tr>
<tr>
<td><strong>View direction</strong></td>
<td>6 o'clock</td>
</tr>
<tr>
<td><strong>Power</strong></td>
<td>5V/550mA</td>
</tr>
<tr>
<td><strong>Active area</strong></td>
<td>153.6(W) x 90.0(H) mm</td>
</tr>
<tr>
<td><strong>Dimensions</strong></td>
<td>165.75(W) x 105.39(H) x 2.45(D) mm</td>
</tr>
<tr>
<td><strong>Weight</strong></td>
<td>300 g</td>
</tr>
</tbody>
</table>

Table 9: The LeMaker 7 inch LCD module specifications

**The 7-inch LCD step-by-step guide**

In the following section, the installation and operation of a 7 inch LeMaker LCD is described. In the first step, the system will be updated:

```bash
sudo apt-get update
sudo apt-get upgrade
```

Let's look at the software part afterwards. Replace the `/boot/bananapro.bin` directory with your boot directory and `script.bin` with your binary startup file:

```bash
git clone http://github.com/LeMaker/fex_configuration
cd fex_configuration/
sudo cp bin/banana_pro_7lcd.bin /boot/bananapro/script.bin
```

Afterwards, edit the `/etc/modules` file and add the `lcd` module to it. If the `lcd` module is already available and commented out using `#`, simply remove the hash sign as shown in the following screenshot:
After this, power off Banana Pro before installing a cable for the 7 inch LCD. The 7 inch display comes together with a 100 mm and 0.5 mm spacing flexible printed circuit (FPC) cable.
Turn the LCD around carefully and gently pull out the brown cable clamp from both ends:

![Figure 20]

Push the flexible cable carefully. Make sure it is straight and the silver conductive end faces downwards toward the blue circuit board, and blue tape faces upwards (Figure 18). After the cable has been inserted completely, push down on both sides of the cable clamp at the same time.

Installing a cable on Banana Pro is pretty much the same as installing the cable on the preceding LCD module. The cable connects to the CON2 connector of Banana Pro board. Do not use the CON1 camera connector. Again, pull out the cable clamps at both sides and push the cable in from the top. The blue protective tape must face the Ethernet jack, while the conductive fingers at the end of the cable face Banana Pro board (Figure 21):
Figure 21: Connecting the LCD to Banana Pro

After the cable is inserted completely, push down on both sides of the clamp and boot your Banana Pro. You should now see this booting screen on the LCD (refer to Figure 22).

Figure 22: Banana Pi using the 7 inch LCD module
The flex cable that’s provided is quite short (100 mm only). If you want a larger cable, you may search the Internet for FPC cable 40 0.5 mm. The LCD is capable of showing the X11 desktop (refer to Figure 23) as well as high-resolution videos in full HD.

Banana Pro shows an H264 video on a 7 inch LCD using mplayer. This will look similar to what is shown in the following image:
The camera module

Banana Pro camera module (Figure 25) uses an Omnivision 5640 CMOS image sensor in an autofocus module and with an integral IR filter. The camera module connects to Banana Pi board via the CSI (CON1). It fulfills the specifications provided in Table 10.

![Image of Banana Pi/Pro camera module](image_url)

Figure 25: The Banana Pi/Pro camera module.

The following are the specifications for the LeMaker Banana Pro camera module:

<table>
<thead>
<tr>
<th>Specification type</th>
<th>Requirements</th>
</tr>
</thead>
<tbody>
<tr>
<td>Image sensor</td>
<td>An omnivision 5640 CMOS image sensor in an autofocus module with an integral IR filter (650±10nm)</td>
</tr>
<tr>
<td>Resolution</td>
<td>5 megapixels</td>
</tr>
<tr>
<td>Active array size</td>
<td>2592 x 1944</td>
</tr>
<tr>
<td>Maximum frame rate</td>
<td>1080p 30fps@24Mhz</td>
</tr>
<tr>
<td>Picture formats</td>
<td>JPEG PNG YUV420 RGB888</td>
</tr>
<tr>
<td>Video formats</td>
<td>Raw h.264</td>
</tr>
<tr>
<td>Connection to Banana Pro</td>
<td>A 40-pin FPC to the CSI-0 (Camera Sensor Interface)</td>
</tr>
</tbody>
</table>
Introduction to Banana Pro

<table>
<thead>
<tr>
<th>Specification type</th>
<th>Requirements</th>
</tr>
</thead>
<tbody>
<tr>
<td>Image control functions</td>
<td>• Automatic exposure control (AEC)</td>
</tr>
<tr>
<td></td>
<td>• Automatic white balance (AWB)</td>
</tr>
<tr>
<td></td>
<td>• Automatic black-level calibration (ABLC)</td>
</tr>
<tr>
<td></td>
<td>• Automatic band filters</td>
</tr>
<tr>
<td></td>
<td>• Mirror and flip functions</td>
</tr>
<tr>
<td>Temperature range</td>
<td>For an OS: -30 °C to 70 °C</td>
</tr>
<tr>
<td></td>
<td>For a stable image: 0 °C to 50 °C</td>
</tr>
<tr>
<td>Lens size</td>
<td>1/4&quot; (quarter inch)</td>
</tr>
<tr>
<td>Dimensions</td>
<td>36 x 32 x 10 mm</td>
</tr>
<tr>
<td>Weight</td>
<td>5 g</td>
</tr>
</tbody>
</table>

Table 10

A step-by-step guide to the camera module

The installation of a camera module is as easy as the installation of an LCD module. Firstly, required kernel modules will be inserted and are specifically used by this component:

```bash
sudo modprobe ov5640
sudo modprobe sun4i_csi0
```

The preceding commands load the Omnivision camera kernel driver and the CSI driver. In order to load these drivers permanently, add the `ov5640` and `sun4i_csi0` modules to `/etc/modules` using this command:

```bash
sudo nano /etc/modules
```

If your Linux OS does not have a module called `sun4i_csi0`, try `sun4i_csi` instead.

Before installing the hardware to your Banana Pro, make sure it is shut down and disconnected from any power supply. Similar to the LCD module cable, remember to also not touch the silver ends of the camera module flexible cable. The cable, again, is quite short (about 6 cm in length).
The Banana Pro camera module connector is shown in Figure 24. Open the connector and insert the flexible cable carefully (Figure 25).

The connector in Banana Pro is called CON1. Do not use CON2 instead as this is the LCD connector (refer to The LCD module section). Using the wrong connector can cause serious damage to your Banana Pro or the extension board because the pin definitions are completely different.

Open the CON1 connector and put the free end of the camera flexible cable into this connector as shown in Figure 26. Finally, confirm that both ends of the flexible cable are sitting correctly in their sockets. Before using the camera, peel off the plastic foil that covers the lens. The complete arrangement can be seen in the following figure:

![Figure 26](image)

The mplayer software can be used in order to test the camera:

```bash
sudo apt-get install mplayer
sudo mplayer tv://
```
On a television, the result would look like what is shown in the following image:

![Image](image1)

**Figure 27**

### Cases

Although Banana Pro has been available since the end of 2014, there are already a few cases available. Most cases are acrylic cases (refer to this image):

![Image](image2)

**Figure 28: The Banana Pro acrylic case (source: Amazon.com)**
Lenovator offers a case that can integrate both Banana Pro and a 2.5 inch hard disk drive (Figure 29). The case allows the mounting of the original LeMaker Wi-Fi antenna that’s provided together with Banana Pro.

![Figure 29: The Lenovator Banana Pro case with a hard disk bay](image)

Banana Pro uses an onboard U.FL connector (refer to Figure 30). The female U.FL connectors are not designed with reconnection in mind, and they are only rated for a few reconnects before replacement is needed.

![Figure 30: Banana Pro onboard the UFL antenna connector (source: Wikipedia)](image)
For those of you who do not want to integrate a hard disk into a case, the acrylic case of Allnet may be a good choice. The case comes with an R-SMA 5 dBi omnidirectional antenna and adapter cable from U.FL to R-SMA.

![Image of Banana Pro acrylic case with 5 dBi antenna and R-SMA connector](source: the Allnet shop)

**GPIO add-ons**

Lenovator provides a general purpose input/output (GPIO) interface called LN digital. LN digital is designed to plug in Banana Pro's GPIO header.

GPIO can be used for input (such as reading sensor information) and output (such as controlling relays and motors). So, the peripherals you buy are typically either input or output devices for GPIO. This extends the basic functionality that are offer from the GPIO on the Pi. The board offers the following:

- 8 open-collector outputs
- 8 LED indicators
- 8 digital outputs
- 4 tactile switches
- 2 changeover relays
It can be plugged directly into Banana Pro GPIO socket and programmed in Python, Scratch, or C (refer to Chapter 2, Programming Languages).

Figure 32: The LN digital interface board for Banana Pro

An onboard microphone
Banana Pro provides an onboard microphone. Recordings can be made using this command:

```
sox -t alsa default output.wav
```

The preceding command requires `sox` to be installed. If `sox` is not available, it can be installed using this command:

```
sudo apt-get install sox
```

Press `Ctrl + C` to stop the recording. The recording will be saved as `output.wav` in the current directory. Recordings can be listened to using this command:

```
mplayer output.wav
```

The microphone sensitivity can be set using the following command:

```
alsamixer
```
The complete audio installation and settings will be explained in detail in Chapter 4, *An Arcade Cabinet*.

![Figure 33: Setting the microphone sensitivity using alsamixer](image)

**Summary**

This was the first chapter of this book in which Banana Pro was introduced. All the connectors of the board were described in detail. Different operating systems were introduced. You learned how to transfer an operating system from an SD card to a hard disk. Moreover, FEX and bin files were introduced, allowing the configuring of Banana Pro hardware. In addition to this, you learned how add-ons, such as LCDs or a camera module, could be used with Banana Pro.

The next chapter will deal with three different programming languages: Scratch, Python, and C/C++. Additionally, the installation of your own kernel will be explained.
Where to buy this book
You can buy Banana Pro Blueprints 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.