## Revision History

<table>
<thead>
<tr>
<th>Revision</th>
<th>Data</th>
<th>Author</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.0</td>
<td>2015-04-17</td>
<td>Tony</td>
<td>Initial version</td>
</tr>
</tbody>
</table>


# Table of Contents

1 Introduction............................................................................................................................................. 1  
  1.1 What is the Banana Pi™? .................................................................................................................. 1  
    1.1.1 Banana Pi Layout .......................................................................................................................... 2  
    1.1.2 Hardware Specifications .............................................................................................................. 3  
  1.2 What is the Banana Pro™? ............................................................................................................... 4  
    1.2.1 Banana Pro Layout ...................................................................................................................... 5  
    1.2.2 Hardware Specifications .............................................................................................................. 6  
  1.3 What can I do with Banana Pro/Pi? ................................................................................................. 7  
  1.4 OS introduction .................................................................................................................................. 7  
    1.4.1 Bananian ..................................................................................................................................... 7  
    1.4.2 Raspbian .................................................................................................................................... 8  
    1.4.3 Lubuntu ...................................................................................................................................... 8  
2 Quick start ............................................................................................................................................... 9  
  2.1 Get what you need. ............................................................................................................................ 9  
  2.2 Prepare your MicroSD or SD card for the Banana Pro/Pi ................................................................ 11  
    2.2.1 Installing the general Linux OS image ....................................................................................... 12  
    2.2.2 Installing the Android OS image ............................................................................................... 15  
  2.3 Set up your Banana Pro/Pi .............................................................................................................. 19  
  2.4 Shut down your Banana Pro/Pi ....................................................................................................... 22  
3 Basic Usages .......................................................................................................................................... 23  
  3.1 Log in to Banana Pro/Pi .................................................................................................................. 23  
    3.1.1 Using the HDMI ......................................................................................................................... 23  
    3.1.2 Using the RCA interface .......................................................................................................... 24  
    3.1.3 Using SSH ............................................................................................................................... 26  
    3.1.4 Using VNC .............................................................................................................................. 28  
    3.1.5 Using the TTL serial port ....................................................................................................... 31  
  3.2 Connect hard drive to SATA interface ............................................................................................ 36  
  3.3 OTG .................................................................................................................................................. 37
4 Advance Usages ........................................................................................................ 39
  4.1 Pin definition ........................................................................................................ 39
  4.2 Extension Interface ............................................................................................. 42
    4.2.1 LVDS/RGB/CPU display interface ................................................................. 42
    4.2.2 Camera Module ............................................................................................ 45
    4.2.3 Uart port ...................................................................................................... 47
    4.2.4 IR .................................................................................................................. 48
  4.3 Resize SD card ...................................................................................................... 55
    4.3.1 Raspbian ...................................................................................................... 56
    4.3.2 Other linux OS ............................................................................................ 56
5 Configure your Banana Pro/Pi .............................................................................. 58
  5.1 How to Modify the script.bin file ....................................................................... 58
    5.1.1 Download sunxi-tools tool: ......................................................................... 58
    5.1.2 How to get FEX file ..................................................................................... 58
    5.1.3 Compile FEX file. ........................................................................................ 59
    5.1.4 Generate scrip.bin file. ................................................................................ 60
  5.2 Auto Log in to system ......................................................................................... 60
  5.3 Network Configuration ....................................................................................... 60
    5.3.1 Use WiFi as the station mode. ....................................................................... 60
    5.3.2 WiFi AP mode .............................................................................................. 63
  5.4 Turn off the screen saver ................................................................................... 66
  5.5 Auto mount storage device. ................................................................................. 66
  5.6 Add desktop shortcut .......................................................................................... 67
6 Applications ........................................................................................................... 68
  6.1 Using Banana Pro to build NAS home server .................................................... 68
    6.1.1 The main function of the NAS home server ................................................. 68
    6.1.2 Preparation work .......................................................................................... 68
    6.1.3 Create a file server ....................................................................................... 70
    6.1.4 Download machine ...................................................................................... 72
6.1.5 Install the DLNA streaming media server ........................................ 76

6.2 XBMC ........................................................................................................... 79
   6.2.1 Modprobe needed modules: ................................................................. 79
   6.2.2 Install UMP library ............................................................................. 79
   6.2.3 Install Xorg() ...................................................................................... 80
   6.2.4 Install libvdpau .................................................................................... 80
   6.2.5 Check the hardware acceleration ....................................................... 81
   6.2.6 Building XBMC for A10 and A20 SoC series ..................................... 82
   6.2.7 Start XBMC .......................................................................................... 83

7 Appendix ........................................................................................................ 84

Appendix A: Banana Pro pin definition ...................................................... 84
Appendix B: Banana Pi pin definition ......................................................... 89
1 Introduction

1.1 What is the Banana Pi™?

It's an open-source single-board computer (SBC). It can run Android 4.2, Android 4.4*, Ubuntu, Debian, Fedora, Raspbian, ArchLinux, openSUSE, OpenWrt, Bananian, and other Linux OS systems (also known as distros) such as Gentoo and Fedora. It uses the AllWinner A20 SoC, and has 1GB DDR3 SDRAM, with wired Gigabit ethernet (RJ45 socket) and a SATA port.

*Please note that the Android_for_bananapi_4.4_beta1 image is not a LeMaker product and therefore only available from an external website with which we are not affiliated or connected to in any way. We therefore provide no support for this image. However, discussions about it in our forums are freely encouraged.
1.1.1 Banana Pi Layout

Front side

Back side
## 1.1.2 Hardware Specifications

### Hardware specification of Banana Pi™

<table>
<thead>
<tr>
<th>Spec</th>
<th>Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>Soc</td>
<td>Allwinner® A20(sun 7i)</td>
</tr>
<tr>
<td>CPU</td>
<td>ARM® Cortex™-A7 Dual-Core1GHz (ARM v7 instruction set)</td>
</tr>
<tr>
<td>GPU</td>
<td>Mali400MP2 Complies with OpenGL ES 2.0/1.1 (hardware acceleration support)</td>
</tr>
<tr>
<td>SDRAM</td>
<td>1GB DDR3 (shared with GPU)</td>
</tr>
<tr>
<td>Power</td>
<td>5V @ 2A via MicroUSB (DC in Only)</td>
</tr>
<tr>
<td>PMU</td>
<td>AXP209</td>
</tr>
</tbody>
</table>

### Features

<table>
<thead>
<tr>
<th>On board Network</th>
<th>10/100/1000 Ethernet RJ45 (optional USB WiFi Dongle)</th>
</tr>
</thead>
<tbody>
<tr>
<td>On board Storage</td>
<td>SD (Max. 64GB) / MMC card slot UP to 4T on SATA disk drive</td>
</tr>
<tr>
<td>Sound Input</td>
<td>Mic</td>
</tr>
<tr>
<td>Display</td>
<td>Supports multi-channel HD display:</td>
</tr>
<tr>
<td></td>
<td>HDMI 1.4 (Type A - full)</td>
</tr>
<tr>
<td></td>
<td>LVDS/RGB/CPU display interface (DSI) for raw LCD panels</td>
</tr>
<tr>
<td></td>
<td>Composite video (PAL and NTSC)</td>
</tr>
<tr>
<td></td>
<td>11 HDMI resolutions from 640×480 to 1920×1080 plus various PAL and NTSC standards</td>
</tr>
<tr>
<td>Video</td>
<td>HD H.264 2160p video decoding</td>
</tr>
<tr>
<td></td>
<td>Multi-format FHD video decoding, including Mpeg1/2, Mpeg4, H.263, H.264, etc H.264 high profile 1080p@30fps or 720p@60fps encoding</td>
</tr>
<tr>
<td>Camera Input</td>
<td>A CSI input connector (Parallel 8-bit camera interface) allows for the connection of a designed camera module (available now from LeMaker)</td>
</tr>
<tr>
<td>USB</td>
<td>2 USB 2.0 host, 1 USB 2.0 OTG (all direct from A20 chip)</td>
</tr>
<tr>
<td>Audio Output</td>
<td>3.5 mm Jack and HDMI</td>
</tr>
<tr>
<td>Buttons</td>
<td>Reset button: Next to MicroUSB connector</td>
</tr>
<tr>
<td></td>
<td>Power button: Next to Reset button</td>
</tr>
<tr>
<td></td>
<td>UBoot button (optional): Behind HDMI connector</td>
</tr>
<tr>
<td>GPIO(2X13) pin</td>
<td>GPIO, UART, I2C bus, SPI bus with two chip selects, CAN bus, ADC, PWM, +3.3V, +5V, ground</td>
</tr>
<tr>
<td>LED</td>
<td>Power Status LED (Red)</td>
</tr>
<tr>
<td></td>
<td>Ethernet Status LED (Blue)</td>
</tr>
<tr>
<td></td>
<td>User Define LED (Green)</td>
</tr>
<tr>
<td>Remote</td>
<td>IR</td>
</tr>
</tbody>
</table>

### Interface definition

| Product size     | 92 mm × 60mm                                                          |
|------------------|                                                                      |
| Weight           | 48g                                                                   |
1.2 What is the Banana Pro™?

Banana Pro™ is an updated version of Banana Pi™ designed by the LeMaker Team. Try Banana Pro™ today and take advantage of the many enhanced features. Banana Pro™ is compatible with many Linux-based operating system and has many distributions specially developed for Banana Pi™ Hardware. Some of these distribution include Lubuntu, Android, Debian, Bananian, Berryboot, OpenSuse, Scratch, Fedora, Gentoo, Open Media Vault, OpenWRT. Banana Pro™ also supports the BSD system.

Banana Pro™ has a wide selection of home applications including: Building a low-cost computer, Servers (for Multimedia, Minecraft or other home servers), Video Game Emulators, Home Security Cameras and more.

Banana Pro™ is an excellent educational learning tool that can be used for many projects including: building Multimedia projects, Robots, Arduino applications or Computer Programming with many popular programming languages available including Scratch (a drag and drop programming language for people beginning to learn how to code).
1.2.1 Banana Pro Layout
## 1.2.2 Hardware Specifications

<table>
<thead>
<tr>
<th>Feature</th>
<th>Specification</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Soc</strong></td>
<td>Allwinner® A20 (sun 7i)</td>
</tr>
<tr>
<td><strong>CPU</strong></td>
<td>ARM® Cortex™-A7 Dual-Core 1GHz (ARM v7 instruction set)</td>
</tr>
<tr>
<td><strong>GPU</strong></td>
<td>Mali400MP2 Complies with OpenGL ES 2.0/1.1 (hardware acceleration support)</td>
</tr>
<tr>
<td><strong>SDRAM</strong></td>
<td>1GB DDR3 (shared with GPU)</td>
</tr>
<tr>
<td><strong>Power</strong></td>
<td>5V @ 2A via MicroUSB (DC in Only)</td>
</tr>
<tr>
<td><strong>PMU</strong></td>
<td>AXP209</td>
</tr>
<tr>
<td><strong>Low-level peripherals</strong></td>
<td>40 Pins Header, 28×GPIO, some of which can be used for specific functions including UART, I2C, SPI, PWM, CAN, I2S, SPDIF, LRA DC, ADC, LINE-IN, FM-IN, HP-IN.</td>
</tr>
<tr>
<td><strong>On board Network</strong></td>
<td>10/100/1000Mbps ethernet (Realtek RTL8211E/D)</td>
</tr>
<tr>
<td><strong>WiFi Module</strong></td>
<td>WiFi 802.11 b/g/n</td>
</tr>
<tr>
<td><strong>Bluetooth</strong></td>
<td>Optional</td>
</tr>
<tr>
<td><strong>On board Storage</strong></td>
<td>MicroSD (TF) card, SATA 2.0</td>
</tr>
<tr>
<td><strong>Sound Input</strong></td>
<td>Mic</td>
</tr>
<tr>
<td><strong>Display</strong></td>
<td>Supports multi-channel HD display:</td>
</tr>
<tr>
<td></td>
<td>HDMI 1.4 (Type A - full)</td>
</tr>
<tr>
<td></td>
<td>LVDS/RGB/CPU display interface (DSI) for raw LCD panels</td>
</tr>
<tr>
<td></td>
<td>Composite video (PAL and NTSC) (via 3.5 mm TRRS jack shared with audio out)</td>
</tr>
<tr>
<td></td>
<td>11 HDMI resolutions from 640×480 to 1920×1080 plus various PAL and NTSC standards</td>
</tr>
<tr>
<td><strong>Video</strong></td>
<td>HD H.264 2160p video decoding</td>
</tr>
<tr>
<td></td>
<td>Mutli-format FHD video decoding, including Mpeg1/2, Mpeg4, H.263, H.264, etc H.264 high profile 1080p@30fps or 720p@60fps encoding</td>
</tr>
<tr>
<td><strong>Audio outputs</strong></td>
<td>HDMI, analog audio (via 3.5 mm TRRS jack shared with composite video out), I2S audio (also potentially for audio input)</td>
</tr>
<tr>
<td><strong>Camera</strong></td>
<td>Parallel 8-bit camera interface</td>
</tr>
<tr>
<td><strong>Audio input</strong></td>
<td>On board microphone</td>
</tr>
<tr>
<td><strong>USB</strong></td>
<td>2 USB 2.0 host, 1 USB 2.0 OTG (all direct from A20 chip)</td>
</tr>
<tr>
<td><strong>Buttons</strong></td>
<td>Reset button</td>
</tr>
<tr>
<td></td>
<td>Power button</td>
</tr>
<tr>
<td></td>
<td>U-boot button</td>
</tr>
<tr>
<td><strong>Leds</strong></td>
<td>Power status led (red)</td>
</tr>
<tr>
<td></td>
<td>User defined led1 (green)</td>
</tr>
<tr>
<td></td>
<td>User defined led2 (blue)</td>
</tr>
<tr>
<td><strong>Other</strong></td>
<td>IR receiver</td>
</tr>
<tr>
<td><strong>Interface definition</strong></td>
<td></td>
</tr>
<tr>
<td><strong>Sizes</strong></td>
<td>92 mm × 60mm</td>
</tr>
<tr>
<td><strong>Weight</strong></td>
<td>45g</td>
</tr>
</tbody>
</table>
1.3 What can I do with Banana Pro/Pi?

Build…

- A computer
- A wireless server
- Games
- Music and sounds
- HD video
- A speaker
- Android
- Scratch
- Pretty much anything else, because Banana Pi is open source

1.4 OS introduction

Linux is a Unix-like and mostly POSIX-compliant computer operating system assembled under the model of free and open-source software development and distribution. The defining component of Linux is the Linux kernel, an operating system kernel first released on 5 October 1991 by Linus Torvalds. The Free Software Foundation uses the name GNU/Linux to describe the operating system, which has led to some controversy.

Typically, Linux is packaged in a form known as a Linux distribution, for both desktop and server use. Some popular mainstream Linux distributions include Debian, Ubuntu, Linux Mint, Fedora, openSUSE, Arch Linux, and the commercial Red Hat Enterprise Linux and SUSE Linux Enterprise Server. Linux distributions include the Linux kernel, supporting utilities and libraries and usually a large amount of application software to fulfill the distribution's intended use.

1.4.1 Bananian

Bananian Linux is a pre-installed Debian 7 image optimized for Banana Pi/Pro. It uses
the official Debian wheezy armhf repositories with a kernel and bootmanager (u-boot),
customized for Banana Pi/Pro.
The main focus is to provide a lightweight headless platform for home servers, small
webservers, ownCloud hosting, Linux based wifi access points, router, NAS systems,
monitoring devices, etc.
For more information please refer to https://www.bananian.org/

1.4.2 Raspbian

A Raspbian image is a file that you can download onto an SD card which in turn can
be used to boot your single-board computer. Using a Raspbian image is the easiest
way for a new user to get started with Raspbian.
For more information please refer to http://www.raspbian.org/

1.4.3 Lubuntu

Lubuntu is a fast and lightweight operating system. The core of the system is based on
Linux and Ubuntu. Lubuntu uses the minimal desktop LXDE, and a selection of
light applications. It focus on speed and energy-efficiency. Because of this, Lubuntu
has very low hardware requirements.
For more information please refer to http://lubuntu.net/
2 Quick start

By following this short quick start guide, you can use your Banana Pro/Pi in just a few minutes. There are three steps to booting your Banana Pro/Pi.

2.1 Get what you need

To enjoy the use of your Banana Pro/Pi, you will need at the very minimum these accessories in the table below.

<table>
<thead>
<tr>
<th>No.</th>
<th>Item</th>
<th>Minimu recommended specification &amp; notes</th>
</tr>
</thead>
</table>
| 1   | SD card for Pi  
     MicroSD card for Pro | ● Minimum size 4Gb; class 4 (the class indicates how fast the card is).  
                               ● We recommend using branded SD or MicroSD cards as they are more reliable. |
| 2a  | HDMI (Full sized – Type A)  
     to  
     HDMI / DVI(VGA) cable | ● HDMI (Type A) to HDMI lead (for HD TVs and monitors with HDMI input).  
                             OR  
                             HDMI (Type A) to DVI adapter cable (for monitors with only a DVI input).  
                             Type A HDMI is 13.9 mm wide (check Wikipedia or Google for more info or look at the photos) |
| 2b  | AV video lead for Pi  
     Composite AV cable for Pro | ● A standard AV video lead or Composite AV cable to connect to your analogue display (eg a TV) if you |
### 3 Keyboard and mouse
- Any standard USB keyboard and mouse should work.
- However, keyboards or mice that take a lot of power from the USB ports may need a powered USB hub. This may include some wireless devices.

### 4 Ethernet cable
- Networking is optional, although it makes updating and getting new software for your Banana Pro/Pi much easier. On Banana Pro, it has onboard WiFi module. On Banana Pi, you can connect an USB WiFi adapter.

### 5 Micro USB power adapter
- A good quality, micro USB power supply that can provide at least 2A at 5V is essential.
- However, most mobile phone chargers are NOT suitable — check the label on the plug. It's possible they can deliver 2 amps and 5 volts, but maybe not at the same time!

### 6 Audio lead (Optional)
- You can use a 3.5mm jack audio cable to connect the audio port to external speakers to get stereo audio. (Pro’s 3.5 mm jack shared with composite video out)

### 7 Mobile hard disk (optional)
- You can choose to connect a mobile hard disk to the SATA port to store more files. Special cables are required for this – look on Amazon – but they are not expensive, under 10 dollars/Euros/pounds.

### 8 A case for your B-Pro/B-Pi (optional but highly recommended)
- A suitable acrylic or similar case, which should cost less than 10 dollars/Euros/pounds and will protect your B-Pro/B-Pi from dust, moisture and most
importantly short circuits and static.

- Please be aware that ALL of the Raspberry Pi cases are NOT compatible (the board dimensions and also the layout of the B-Pi’s inputs/outputs are different).

## 2.2 Prepare your MicriSD or SD card for the Banana Pro/Pi

In order to enjoy your Banana Pro/Pi, you will need to install an Operating System (OS) onto an MicroSD/SD card. **You need to pay attention to that the Android image cannot use the dd command in linux or the Win32Diskimage in Windows, so you need to use the PhoenixCard tool to burn the image to the MicroSD/SD card (you can refer to the 2.2.2 section).** The instructions below will teach you how to write an OS image to your MicroSD/SD card either in Windows or Linux.
2.2.1 Installing the general Linux OS image

1. Insert your MicroSD/SD card into your computer or card reader. The size of the MicroSD/SD should be larger than the OS image size, generally 4GB or greater.

2. Format the MicroSD/SD card.

**In Windows:**

i. Download an MicroSD/SD card format tool such as **SD Formatter** from [https://www.sdcard.org/downloads/formatter_4/eula_windows/](https://www.sdcard.org/downloads/formatter_4/eula_windows/)

ii. Unzip the download file and run the setup.exe (Run as Administrator) to install the tool on your machine.

iii. After installation, start the tool (again, Run as Administrator). In the "Options" menu, set "FORMAT TYPE" option to QUICK, "FORMAT SIZE ADJUSTMENT" option to "ON".

![SD Formatter Tool]

iv. Check that the drive letter of the MicroSD/SD card you inserted matches the one selected by the software. Otherwise it will format and delete all data on another drive or card. To be safe, only have your main hard drive...
connected and only the MicroSD/SD card that you want to use in your B-Pi. LeMaker is not responsible for any loss of data.

v. Click the “Format” button.

**In Linux:**

i. In a terminal, run the

```
sudo fdisk -l
```

command to check the MicroSD/SD card node.

ii. Run the

```
umount /dev/sdxx
```

to unmount all the partitions of the MicroSD/SD card.

iii. Run the

```
sudo fdisk /dev/sdx
```

command. Use the o command to delete all partition of MicroSD/SD card and use the n command to add one new partition. Use the w command to save change.

iv. Run the

```
sudo mkfs.vfat /dev/sdx1
```

command to format the new created partition of MicroSD/SD card as FAT32.

(x should be replaced according to your MicroSD/SD card node as discovered in point vi above)

You can also jump this step under Linux, because write image command `dd` under Linux will format the MicroSD/SD card automatically.

3. Download the OS image from the Downloads webpage.

4. Unzip the download file to get the OS image (should have the extension .img). To do this -
In **Windows**: Right click on the file and choose “Extract all”.

In **Linux**: Run the

```
unzip [path]/[downloaded filename]
```

command. If the filename extension is .tgz, run the

```
tar zvxf [path]/[downloaded filename]
```

command.

Ensure that neither the file name of the image you’re using or the path contain any spaces (or other odd characters, for that matter).

5. Write the image file to the MicroSD/SD card.

**In Windows:**

i. Download a tool that can write images to MicroSD/SD card, such as **Win32 DiskImager** from:


ii. Open the unzipped image file.

iii. Click the Write button. Wait patiently to successfully complete the writing. Do not disturb or disconnect/remove the card or shut down the computer during this process. When it has finished, soft-eject the card using the 'Safely Remove Hardware' icon in the System Tray/Notification area (bottom right of your screen), then physically remove the card from the card reader.
In Linux:

i. Run the

```
sudo fdisk -l
```

command to check the MicroSD/SD card node.

ii. Verify if the hash key of the zip file is the same as shown on the downloads page (optional).

```
shalsum [path]/[imagename]
```

This will print out a long hex number which should match the "SHA-1" line for the MicroSD/SD image you have downloaded.

iii. Run the

```
sync && sudo umount /dev/sdxx
```

to unmount all the partitions of the MicroSD/SD card.

iv. Run the

```
sudo dd bs=4M if=[path]/[imagename] of=/dev/sdxx
```

command to write image file to MicroSD/SD card. Wait patiently to successfully complete writing, it’s a very long time. Please note that block size set to 4M will work most of the time, if not, please try 1M, although 1M will take considerably longer. You can use the

```
sudo pkill –USR1 –n –x dd
```

command to check progress.

2.2.2 Installing the Android OS image

The Android images (4.2 v2.0 & 4.4 beta 1) cannot unfortunately use the dd command in Linux or the Win32Diskimager in Windows, so you need to use the PhoenixCard tool to write the image the MicroSD/SD card.
Download the Android image and PhoenixCard.

The Android 4.2 image (a LeMaker product) from our Downloads page:

http://www.lemaker.org/resources/9-38/image_files.html

The Android 4.4 image (not a LeMaker product) from the external website.

PhoenixCard:

International download address:

https://drive.google.com/file/d/0B_VynIqhAcB7NTg2UkRDdHRWX2s/edit?usp=sharing

China domestic download address:

http://pan.baidu.com/s/1qW9dFLq?qq-pf-to=pcqq.c2c

Install the PhoenixCard software (Run as Administrator), then start the program from its icon (also using Run as Administrator).

First, check the disk letter is EXACTLY the same as your MicroSD/SD card with the button 'Disk Check'. (Better safe than sorry – disconnect all other USB devices and secondary hard drives before continuing. See this forum link for help if you do this.)

Then load the disk file by clicking on the button 'Img File'. Next step – click the radio button 'Startup' in the Write Mode box, then press the 'Format to Normal' button.

Please note: although the text in the window of the above screenshot says "Device OK, the size of the device is XXXXX M", depending on your system and if it is an internal/external or single/multi-card reader, you may instead get only a message such as "Find 4 device, Please select correct the one"
Burn the Android image to the MicroSD/SD card (by pressing 'Burn' of course).

This can take quite a while so be patient. As mentioned previously, do not disturb the MicroSD/SD card or remove it or turn off the computer during this process. Serious damage to the card and/or computer hardware can occur.
When the final 'Success' message appears, soft-eject the card using the 'Safely Remove Hardware' function in the Notification Area/System Tray (bottom right hand corner of the Windows screen). Now you can physically remove it and then insert it into the Banana Pro/Pi, plug in the mouse (a physical keyboard is optional as you can use Android's in-built on-screen version instead), the display and finally the power supply to experience the Android system on your Banana Pro/Pi.

The Android system start-up GUI.

(GUI = Graphical User Interface)

NB. The first time the system starts will take a long time, so please be patient - subsequent boots will be quicker.

The picture below shows the start screen of the 'Android For Banana Pro v1503' image.
2.3 Set up your Banana Pro/Pi

According to the set up diagram below, you can easily set up your Banana Pro/Pi.

1. Insert an SD card
   If you have a free SSD/HDD, you can plug it into the SATA port.

2a. Connect the Display
   Connect to your HDMI digital TV or monitor (HDMI-DVI adapter may be required).

2b. Connect the Display
   If you don’t have an HDMI-equipped display, simply plug it into your analogue TV or monitor.

3. Connect the Inputs
   Plug in a USB keyboard and mouse.

4. Connect the Network
   Plug in an RJ-45 Ethernet cable.

5. Power up
   Plug in the micro USB power supply.

6. (Optional) Connect the hard drive
   Step 2 above taught you how to write an image to the SD card.

   If you have a free SSD/HDD, you can plug it into the SATA port.
1. Insert an MicroSD card
   Stop 2 above taught you how to write an image to the MicroSD card

2a. Connect the Display
   Connect to your HDMI digital TV or monitor (HDMI-DVI adapter may be required)

2b. Connect the Display
   If you don’t have an HDMI-equipped display, simply plug it into your analogue TV or monitor

3. Connect the Inputs
   Plug in a USB keyboard and mouse

4. Connect the Network
   Plug in an RJ-45 Ethernet cable or use the WiFi

5. Power up
   Plug in the micro USB power supply

6. (Optional) Connect the hard drive
   If you have a free SSD/HDD, you can plug it into the SATA port
1. Insert the newly written MicroSD/SD card into the MicroSD/SD card socket on the left side edge of the underside of the board.

2. On the bottom edge in the middle of the board is the HDMI Type A (full sized – 13.9mm wide) port, just to the right of the SATA port. Just connect any HDMI Type A cable from the B-Pi or B-Pro to your TV or HDMI Monitor.
   
   *If you don't have a TV/Monitor with a HDMI or DVI-D port:*
   
   *On the B-Pi* you can use the yellow AV jack located in the middle of the top edge and the 3.5 mm stereo headphone jack to the right of it.
   
   *On the B-Pro* you can use the AV jack located in the top edge.

3. Plug a USB keyboard and mouse into the USB slots located on the right hand edge.

4. Just under the USB ports on the right hand edge is the Ethernet connector if you want to plug the Banana Pro/Pi into a wired network.

5. Finally, at the very left of the bottom edge is the micro-USB power connector. Plug in a regulated power supply that is rated at 5V ±5% and at least 2A. Any value bigger than 2A (like 2.5A) will also work. Avoid using the smaller chargers used for small GSM phones, as these are often unregulated - even if they claim "5V 2A", they may do "5V" and may do "2A", but not at the simultaneously!

   Make sure you have the correct USB plug. In the photo below, the mini-USB (on the left) is the wrong one. It’s thicker and looks like a trapezoid with its sides pinched in. The micro-USB (on the right) is the correct one. It is thinner and also looks like a trapezoid except its sides are rounded outward.
6. (This step is optional) If you have a free SATA 2.5 inch or 3.5 inch hard drive (SSD or HDD), you can use it on the Banana Pro/Pi. Connect the SATA cable to the SATA port just between the micro-USB and HDMI ports. Remember to put the power cable with the 2 male 2.54mm headers into the SATA power. Then you can plug your hard drive into the other side of the SATA cable. Be careful with the connection of the different color cables.

If all goes well, the Banana Pro/Pi will boot in a few minutes. The screen will display the OS's GUI (Graphical User Interface). The first boot of a new OS can sometimes take a long time. Be patient! Subsequent boots are usually much quicker.

2.4 Shut down your Banana Pro/Pi

You can use the GUI to shut down the Banana Pro/Pi safely.

Also you can run the command in the terminal:

```bash
sudo halt
```

or

```bash
sudo shutdown –h now
```

This will shut down the Pro/Pi safely, (just use the power key to turn off might damage the MicroSD/SD-cards file system).
3 Basic Usages

3.1 Log in to Banana Pro/Pi

There are five methods for you to log in to Banana Pro or Banana Pi OS.

3.1.1 Using the HDMI

(1). First, get the basic things you need: a Banana Pro/Pi with a prepared MicroSD/SD card containing an OS, an HDMI cable with Type A plug (13.9mm wide) for the B-Pi or B-Pro end, an HD-ready monitor, a micro USB power adapter, a keyboard and a mouse.

We'll be using the HDMI to VGA lead in the photo below.

(2). Connect the Banana Pro/Pi as shown here:
(3) Power on the Banana Pro/Pi by plugging in the mains adapter. You will then see the boot screen and eventually get to the desktop of the Banana Pro/Pi. (Remember, the first boot with a new OS on a card takes longer than usual - subsequent boots are quicker.) The photos below show first the Linux code scrolling up as it boots, and then the next screenshot shows the final main screen of the Raspbian OS (a variant of Debian 7 'Wheezy') for the Raspberry Pi which has been adapted for the Banana Pro/Pi.

3.1.2 Using the RCA interface

The AV cable is usually yellow and of the RCA connector/phono type – it normally comes bundled with similar red and white versions. Plug the yellow* one into the AV port (also yellow) of the Banana Pi, and the other end into the corresponding socket on your TV. But Pro via 3.5 mm TRRS jack shared with composite video out so it need a Composite AV cable to connect device.
Power on the Banana Pro/Pi. If there is no display in the monitor, you may need check the script.bin file.

script.bin is a binary configuration file used by Allwinner SOC core driver or LiveSuit. And it contains the information that how to set the peripherals, port, and I/O pins of A10/A20 target board.

Because we can’t compile the script.bin file directly, we have to modify the corresponding text file whose format is FEX, so we must change script.bin file into script.fex file. We can use the sunxi-tools to change binary file into text file or change it back. Please refer to the steps in the 5.1 section.

AV driver has been included in the kernel. Just Configure the script.fex directly:

```plaintext
[disp_init]
disp_init_enable = 1
disp_mode = 0
screen0_output_type = 2
screen0_output_mode = 11
screen1_output_type = 0
screen1_output_mode = 4
fb0_width = 1024
fb0_height = 768
fb0_framebuffer_num = 2
fb0_format = 10
fb0_pixel_sequence = 0
fb0_scaler_mode_enable = 0
fb1_width = 1024
fb1_height = 768
fb1_framebuffer_num = 2
fb1_format = 10
fb1_pixel_sequence = 0
```

Banana Pro&Pi User Manual

Copyright © 2015 LeMaker. All Rights Reserved.
fb1_scaler_mode_enable = 0
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
[tvout_para]
tvout_used=1
tvout_channel_num=1

Then convert it into script.bin format by using fex2bin, and replace the earlier script.bin on the board.

Reboot the system, now the AV port can be used for displaying.

[* For pi only use the cable with the yellow plugs as it has a 75 ohm resistor built in – the white and red ones don't. Please, someone correct or update this if it is inaccurate.]

### 3.1.3 Using SSH

Using SSH to log in to the Banana Pro/Pi for remote operation is very convenient, safe and highly efficient. In addition, it is not necessary to even use a monitor linked to the Banana Pro/Pi via HDMI cable in some situations, for example, if the Banana Pro/Pi is acting as a home server. The SSH server is installed by default and starts during boot up on the 'Raspbian for Banana Pro/Pi' and 'Lubuntu for Banana Pro/Pi' operating systems. So in general, you don't need to install SSH on your Banana Pro/Pi.

1. If the SSH is not installed, you can install it using this command

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

2. Check whether the SSH has started.
ps -e | grep ssh

If ssdh is in the output, the SSH server has started. If not, you should start it with your own command:

```
sudo /etc/init.d/ssh start
```

To stop the SSH server:

```
sudo /etc/init.d/ssh stop
```

Restart the SSH server:

```
sudo /etc/init.d/ssh restart
```

(3). Configure the rc.local file so that you can set the SSH server to start during boot:

```
sudo nano /etc/rc.local
```

Add

```
/etc/init.d ssh start
```

before exit 0.

Now you need to make sure your Banana Pro/Pi and your computer are connected to the same local internet.

(4) Log in to your Banana Pro/Pi.

In Windows, download a free SSH client such as PuTTY for remote login to the Banana Pro/Pi. Start PuTTY on your computer and then enter the IP address of your Banana Pro/Pi. Then click Open to connect to your Banana Pro/Pi. Finally, enter the user name and password to complete verification.
In Ubuntu, it is easier to log in to your Banana Pro/Pi using the ssh command only:

```
ssh remote_username@remote_host
```

The remote username is the same user name that you use to log in to the Banana Pro/Pi such as pi. The remote_host is the Banana Pro/Pi's IP address.

### 3.1.4 Using VNC

In the previous section, we saw how SSH can be used to control remotely your Banana Pro/Pi without an HDMI display, as well as being safe, convenient and efficient. Another way you can try this is by using VNC to display the Banana Pro/Pi's desktop on your PC through its IP address.

When the VNC service is on, a .vnc file will be generated. This file contains the information about the VNC service. The location and path of .vnc is generally to be found at either [/home/username] or [root] according to the user's permissions. The following steps will guide you in configuring VNC if you are the root user.

1. Install the VNC Server

   ```
sudo apt-get install tightvncserver
   ```

2. Start the VNC Server and set the password

   ```
vncserver
   ```
This will require you to enter a VNC password (at least 6 characters) for the first time, and then it asks you if you would like to enter a view-only password (y/n), enter n to skip this step. You can check whether the VNC service is set up successfully.

The default port is **5901**.

(3). Configure the `.vnc/xstartup` script

You should configure the xstartup script to display the desk in VNC client. You can choose which desktop system session to use.

Edit the xstartup script to enable different desktop sessions

```bash
sudo nano /root/.vnc/xstartup
```

**Gnome. The most powerful desktop session.**

```
/bin/sh
xrdb $HOME/.Xresources
xsetroot -solid grey
x-terminal-emulator -geometry 80x24+10+10 -ls -title "$VNCDESKTOP Desktop" &
x-window-manager &
$xfce4-session &
```

**X-Window. The simplest desktop session.**

```
/bin/sh
xrdb $HOME/.Xresources
xsetroot -solid grey
x-terminal-emulator -geometry 80x24+10+10 -ls -title "$VNCDESKTOP Desktop" &
x-window-manager &
$xfce4-session &
```

**Xfce 4. Linux like desktop session.**
After modifying the xstartup script you should restart the VNC service to make the modification work. First kill the current VNC service.

```bash
vncserver -kill
```

And restart the modified vnc service.

```bash
vncserver :1
```

(4). Use VNC-View on your computer to log in to your Banana Pro/Pi. Enter the Banana Pro/Pi's IP and port. The port of desktop 1 is 5901, desk 2 is 5902 and so on. You can use `ifconfig` command to get the IP address.
This is an X-Window display.

On X-Window, if using the Chromium web browser, you should start the VNC service under a normal user and then use VNC-View to log in to the Banana Pro/Pi. Don't use the root user to start the VNC service.

Summary of Commands
The commands to start the VNC service:

```
vncserver
vncserver :1
tightvncserver
```

The commands to stop the VNC service:

```
vncserver -kill
tightvncserver -kill
```

The command to change the password:

```
ps -axjf | grep vnc
```

### 3.1.5 Using the TTL serial port

This section will introduce you to using the TTL serial port to log in to the Banana Pro/Pi.
(1). Familiarise yourself with the pin assignments of the UART interface on the Banana Pro/Pi.

(2). Use the PL2303 to connect the Banana Pro/Pi and the computer.

The PL2303 operates as a bridge between a USB port and a standard RS232 serial port. There are pins for 3.3V, TXD, RXD, GND and 5V on the PL2303 as shown here.
The table below shows the connections between the Banana Pro/Pi and the PL2303.

<table>
<thead>
<tr>
<th>Pin on Banana Pro/Pi</th>
<th>Pin on PL2303</th>
</tr>
</thead>
<tbody>
<tr>
<td>GND port</td>
<td>GND</td>
</tr>
<tr>
<td>TX port</td>
<td>RXD</td>
</tr>
<tr>
<td>RX port</td>
<td>TXD</td>
</tr>
</tbody>
</table>

Attention:

A. TX on one device is connected to RX on the other and vice versa.
B. The power line (red one, 5V) is NOT connected. The connection between Banana Pro/Pi and PL2303 is shown below.
(3). Software on the computer

A. In Linux, the driver for PL2303 is already in the system.
Install the minicom software.

```
sudo apt-get install minicom
```

When the installation has finished, setup the minicom:

```
sudo minicom -s
```

Select the "Serial port setup" option
Modify the parameters:

| A | Serial Device: /dev/ttyUSB0 |
| F | Hardware Flow Control: No |

And save and then select the "save setup as dfl" option.

Save the setting and select "Exit from Minicom" to exit.

B. In Windows, the driver may already have been automatically installed. If not, you can install it yourself. You can try TeraTerm or Putty to use the TTL serial port.

(With thanks to native speaker “roses” for checking and upgrading this document)
3.2 Connect hard drive to SATA interface

The SATA port on the Banana Pro/Pi can be used with any SATA hard drive (in terms of data transfer).

However, in terms of supplying power to your drive, the Pro/Pi can ONLY supply 5 Volts to a 2.5 inch SSD.

In certain cases, connecting a 2.5 inch HDD is also possible, but some brands and models require too much current, especially during spin up, and this can damage the power rails of your SBC. Furthermore, it also depends on what rating PSU you are using, which and how many USB devices you have plugged in, whether your USB hub is powered or passive, and possibly a few other factors as well. If in doubt, power any kind of HDD externally.

Powering a 3.5 inch HDD MUST be done via an external power supply because it needs both +5V and +12V via a Molex connector. (An old PSU from a desktop computer is possible, but you need to physically connect two specific pins in the ATX connector block to simulate the Power button being pressed. Again, if you don't know what you're doing, buy a dedicated PSU - they're not expensive and are more efficient and therefore 'greener' anyway.)

The two pin SATA power output of the Pro/Pi is situated between the micro-USB DC In and the corner of the board. The 5V pin is nearest the DC In (on the right as you look at it from the side) and the other is ground (GND).
For the connecting cables, you can get these easily and cheaply. Look on Amazon for "dedicated SATA cable HDD connectors with power supply". In the UK, they are currently 3.70 GBP, in Europe 5.96 EUR (and you get 2 pieces!!) with postage also being very low. Delivery times are not quick however. Try and obtain them locally if possible (if the price is right) and 'try before you buy'.

Beware: it's been reported external link that some of these cables have the polarity of +5V & GND the wrong way round!! (Unfortunately, the plug and its socket are 'keyed' and can only meet each other in one orientation, so it's not just a simple case of turning it upside down - pity!!)

If you look closely at the photo in this blog, you can see insulating tape where he's chopped both cables and rejoined red to black and vice versa. (If you have the correct tool(s), it is maybe possible to push out the internal metal contacts of the plug like in a Molex connector and then swap them around but it's VERY fiddly. Personally, rather than doing such a 'bodge job' as that blogger did, I would use some insulated bullet connectors.)

### 3.3 OTG

The OTG port (OTG = On the Go) allows you to connect any USB device using a special (but usually inexpensive) micro-USB to female Type A connector cable. This therefore gives you a total of 3 USB ports - more if you have a passive or powered USB hub.

The socket for this port is located on the edge of the board where the SD card slot is,
just to the right of the Reset switch and around the corner from the SATA power supply socket. (Confusingly, it's in the same place as where the Raspberry Pi's Power In socket would be - that SBC doesn't have an OTG socket of course - so if you are a regular R-Pi user just coming over to the B-Pi side of things, have your wits about you when connecting your PSU. Personally, I have my PSU plugged into a switched wall socket or switched extension cable with the micro-USB permanently connected to the Pi. This reduces wear and tear on the socket's metal housing and also the solder joints and anchor points around it on the PCB.)

In certain cases, the OTG port can also be used for supplying extra power to the SBC by plugging into it what will become for the Pi a second Power Supply, for example when first booting the Android 4.4 image. However, it is NOT recommended to plug in any power source into the OTG for Linux and/or Android 4.2 images at ANY time, and certainly NEVER on its own if there is nothing plugged into the DC In socket.
4 Advance Usages

4.1 Pin definition

The Banana Pi's CON3, J11 & J12 connectors
Banana Pro CON6

<table>
<thead>
<tr>
<th>Pin#</th>
<th>NAME</th>
<th>Pin#</th>
<th>NAME</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>VCC-3.3V</td>
<td>2</td>
<td>VCC-5V</td>
</tr>
<tr>
<td>3</td>
<td>TWI2-SDA</td>
<td>4</td>
<td>VCC-5V</td>
</tr>
<tr>
<td>5</td>
<td>TWI2-SCK</td>
<td>6</td>
<td>GND</td>
</tr>
<tr>
<td>7</td>
<td>IO-1</td>
<td>8</td>
<td>UART4_TX</td>
</tr>
<tr>
<td>9</td>
<td>GND</td>
<td>10</td>
<td>UART4_RX</td>
</tr>
<tr>
<td>11</td>
<td>IO-0 (UART2_RX)</td>
<td>12</td>
<td>PWM1</td>
</tr>
<tr>
<td>13</td>
<td>IO-2 (UART2_TX)</td>
<td>14</td>
<td>GND</td>
</tr>
<tr>
<td>15</td>
<td>IO-3 (UART2_CTS)</td>
<td>16</td>
<td>IO-4(CAN_TX)</td>
</tr>
<tr>
<td>17</td>
<td>VCC-3.3V</td>
<td>18</td>
<td>IO-5(CAN_RX)</td>
</tr>
<tr>
<td>19</td>
<td>SPI0_MOSI</td>
<td>20</td>
<td>GND</td>
</tr>
<tr>
<td>21</td>
<td>SPI0_MISO</td>
<td>22</td>
<td>IO-6(UART2_RTS)</td>
</tr>
<tr>
<td>23</td>
<td>SPI0_CLK</td>
<td>24</td>
<td>SPI0_CS0</td>
</tr>
<tr>
<td>25</td>
<td>GND</td>
<td>26</td>
<td>SPI0_CS1</td>
</tr>
<tr>
<td>27</td>
<td>TWI3-SDA</td>
<td>28</td>
<td>TWI3-SCK</td>
</tr>
<tr>
<td>29</td>
<td>IO-7(IR0_TX/SPDIF_MCLK)</td>
<td>30</td>
<td>GND</td>
</tr>
<tr>
<td>31</td>
<td>UART7_RX</td>
<td>32</td>
<td>UART7_TX</td>
</tr>
<tr>
<td>33</td>
<td>IO-8(SPDIF_DO)</td>
<td>34</td>
<td>GND</td>
</tr>
<tr>
<td>35</td>
<td>I2S0_LRCK</td>
<td>36</td>
<td>I2S0_BCLK</td>
</tr>
<tr>
<td>37</td>
<td>I2S0_MCLK</td>
<td>38</td>
<td>I2S0_DI</td>
</tr>
<tr>
<td>39</td>
<td>GND</td>
<td>40</td>
<td>I2S0_DO0</td>
</tr>
</tbody>
</table>

The Banana Pro's CON6 connector
The detailed pin definition please refer to the appendix.
4.2 Extension Interface

4.2.1 LVDS/RGB/CPU display interface

LeMaker has designed three different size LCD modules, include 3.5 inch, 5.0 inch and 7.0 inch.
3.5 inch LCD module is RGB interface with 320*240 resolution.
5.0 inch LCD module is RGB interface with 800*480 resolution.
7.0 inch LCD module is LVDS interface with 1024*600 resolution.

Replace the configuration file

In order to use the LCD module, you need modify the script.bin file in your OS. You can download the modified file for each size LCD module from LeMaker github:

https://github.com/LeMaker/fex_configuration

```
git clone http://github.com/LeMaker/fex_configuration
```

Enter into the fex_configuration:

```
cd fex_configuration
```

You will find two directories, bin and fex. In the bin directories there are compiled bin files that you can use it directly.

Enter the bin directory:

```
cd bin
```

You will see 6 bin files, 3 for Banana Pro and 3 for Banana Pi. On Banana Pro, you need use the files named banana_pro_Xlcd.bin. On Banana Pi, you need use the files named banana_pi_Xlcd.bin.(X should be 35, 5 or 7, choose the right file according to what size LCD you use) Rename the corresponded bin file that you use to script.bin, and replace the new script.bin file with the old one in your OS.
The script.bin file is located at the first partition of your memory card with OS.

Examples are as follows:

3.5 inch RGB LCD FOR Banana Pro/Pi
Picture below is the 3.5" display with 320x240 24-bit color pixels, and the module connect the the Banana Pro via the Parallel RGB interface. Provide distinct display effect with lightweight and portable, we can use the 3.5" display as a console, X window port, displaying images or video etc.

3.5 inch LCD specifications shown in the following table

<table>
<thead>
<tr>
<th>LCD size</th>
<th>3.5 inch (Diagonal)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Interface</td>
<td>Parallel RGB</td>
</tr>
<tr>
<td>Resolution</td>
<td>320 x 3(RGB) x 240</td>
</tr>
<tr>
<td>Driver element</td>
<td>a-Si TFT active matrix</td>
</tr>
<tr>
<td>Dot pitch</td>
<td>73(W) x 219(H) um</td>
</tr>
<tr>
<td>Connections to Banana Pi</td>
<td>40 Pin FPC to the Display Sensor Interface (DSI)</td>
</tr>
<tr>
<td>Surface treatment</td>
<td>Glare</td>
</tr>
<tr>
<td>Color arrangement</td>
<td>RGB-stripe</td>
</tr>
<tr>
<td>View direction</td>
<td>6 O’Clock</td>
</tr>
<tr>
<td>Power</td>
<td>5V/250mA</td>
</tr>
<tr>
<td>Active area</td>
<td>70.08(W) x 52.56(H) mm</td>
</tr>
<tr>
<td>Dimension</td>
<td>76.9(W) x 63.9(H) x 3.26(D) mm</td>
</tr>
<tr>
<td>Weight</td>
<td>TBD</td>
</tr>
</tbody>
</table>

This module consists of two parts: 3.5 inch LCD display and 40 pin FPC. The LCD connect to Banana Pro board by use the 40pin FPC as shown in the picture below:
To use the 3.5 inch lcd module, you need change the script.bin file in the MicroSD card. Power on the Banana Pro, and mount the first partition of the MicroSD card:

```
mount /dev/mmcblk0p1 /mnt
```

Then replace the script.bin file with the one for 3.5 inch lcd:

```
git clone http://github.com/LeMaker/fex_configuration
cd fex_configuration/bin
sudo cp banana_pro_35lcd.bin /mnt/script.bin
sync
sudo umount /mnt
```

(For Raspbian: sudo cp banana_pro_35lcd.bin /mnt/bananapi/script.bin)

Edit the `/etc/modules` with your favorite editor (with sudo, ie sudo vim `/etc/modules`), uncomment ‘lcd’ (remove the ‘#’ in front of it).

Reboot the system, and you will see the display on the 3.5 inch LCD module.

**5 inch LCD FOR Banana Pro/Pi**
4.2.2 Camera Module

The Banana Pro/Pi camera module is available right now and is a high definition camera module for the Banana Pro/Pi using an Omnivision 5640 CMOS image sensor in an auto-focus module and with an integral IR filter. The camera module connects to the Banana Pro/Pi board via the CSI connector designed specifically for interfacing to cameras. Providing high sensitivity, low crosstalk and low noise image capture in a small and lightweight design.

(1) Connecting the camera module
(2) Setting up the B-Pro/B-Pi and running the camera

In the graphical interface, open the command line terminal LXterminal.

First you have to load the necessary drivers needed by the camera module:

```bash
sudo modprobe ov5640
sudo modprobe sun4i_csi
```

If `modprobe sun4i_csi` does not work try:

```bash
sudo modprobe sun4i_csi0
```

Then you can use MPlayer to test the camera:

```
mplayer tv://
```
In lubuntu, you also can use the application GUVCvideo.

4.2.3 Uart port

Enable UART

On Banana Pi, there are UART0, UART2, UART3, UART7 available. On Banana Pro, there are UART0, UART2, UART4, UART7 available. You can enable them or disable them in script.bin file. For example, if you want to enable UART3 on Banana Pi, so you need first use bin2fex tool to transform script.bin to script.fex, and then you can edit the [uart3_para] in script.fex. Please refer to the steps in the 5.1 section. Set the uart_used value to 1 to enable it:

uart_used = 1

Then transform script.fex back to script.bin using fex2bin tool.

Identify UART

When we enable uart3, we need to know uart3 is known as ttySX (X should be 0, 1, 2, 3,...).

ls sys/bus/platform/devices/sunxi_uart.3/tty

this displays ttySX.

Test UART

By default, the serial baud rate is 9600, so we need set the serial debug tool baud rate to 115200 on PC. Connect the serial cable between Banana Pro/Pi and PC:
echo hello > /dev/ttyS2

If the UART3 is well, you will see hello on your PC. After that you can use the UART in your program.

4.2.4 IR

(1) Load IR kernel module:

modprobe sunxi-ir

(Now, sunxi_ir was built into kernel, you needn’t to do this step anymore. If you are not sure whether sunxi-ir is in kernel, please follow this step, you can add it in /etc/modules.)

Check if you have the correct device (should be dev/input/event0 )

cat /proc/bus/input/devices

and/or check with dmesg (but no evidence about event*)

dmesg|grep sunxi-ir

So, Device is /dev/input/event0

(2) Find a working remote

apt-get install evtest
Run `evtest` and press button on your remote facing the IR receiver and if you see some output ... bingo! You have found a working remote!

Remember: only few remotes work.

```
evtest /dev/input/event0
```

```
root@bananapi:/etc/network# evtest /dev/input/event0
Input driver version is 1.0.1
Input device ID: bus 0x19 vendor 0x1 product 0x1 version 0x100
input device name: "sunxi-ir"
Supported events:
  Event type 0 (EV_SYN)
  Event type 1 (EV_KEY)
    Event code 1 (KEY_ESC)
    Event code 2 (KEY_1)
    Event code 3 (KEY_2)
    Event code 4 (KEY_3)
    Event code 5 (KEY_4)
    Event code 6 (KEY_5)
    Event code 7 (KEY_6)
    Event code 8 (KEY_7)
    Event code 9 (KEY_8)

Properties:
Testing ... (interrupt to exit)
Event: time 162309680.219942, type 1 (EV_KEY), code 28 (KEY_ENTER), value 1
Event: time 162309680.219952, --------- SYN_REPORT  ---------
Event: time 162309680.464036, type 1 (EV_KEY), code 28 (KEY_ENTER), value 0
Event: time 162309680.464044, --------- SYN_REPORT  ---------
Event: time 162309670.555974, type 1 (EV_KEY), code 13 (KEY_EQUAL), value 1
Event: time 162309670.555984, --------- SYN_REPORT  ---------
Event: time 162309670.804063, type 1 (EV_KEY), code 13 (KEY_EQUAL), value 0
Event: time 162309670.804074, --------- SYN_REPORT  ---------
Event: time 162309668.443753, type 1 (EV_KEY), code 21 (KEY_U), value 1
Event: time 162309668.443766, --------- SYN_REPORT  ---------
Event: time 162309668.694032, type 1 (EV_KEY), code 21 (KEY_U), value 0
Event: time 162309669.694033, --------- SYN_REPORT  ---------
Event: time 162309670.816394, type 1 (EV_KEY), code 22 (KEY_U), value 1
Event: time 162309670.816405, --------- SYN_REPORT  ---------
```

(3) Now install and configure LIRC

```
apt-get install lirc
```

```
edit /etc/lirc/hardware.conf as below:
```
```
# /etc/lirc/hardware.conf
#
# Arguments which will be used when launching lircd
LIRCD_ARGS=""
#
#Don't start lircmd even if there seems to be a good config file
```
START_LIRCMD=false

# Don't start irexec, even if a good config file seems to exist.
START_IREXEC=false

# Try to load appropriate kernel modules
# if LOAD_MODULES=false , modules must be preloaded, i.e. during boot (/etc/modules)
LOAD_MODULES=true

#MODULES="sun4i_ir"
# newer kernel
MODULES="sunxi-ir"

# Run "lircd --driver=help" for a list of supported drivers.
REMOTE_DRIVER="devinput"

# usually /dev/lirc0 is the correct setting for systems using udev
REMOTE_DEVICE="/dev/input/event0"

START_LIRCD="true"

# Default configuration files for your hardware if any
LIRCD_CONF=""
LIRCMD_CONF=""

Now you must configure LIRC with your remote.

Take a look at http://lirc.sourceforge.net/remotes/generic/ and download a generic configuration file, i.e. NEC.conf. Run following command at /etc/lirc/.
wget http://lirc.sourceforge.net/remotes/generic/NEC.conf

Run `irrecord` to record the code of your remote.

```
irrecord -H devinput -d /dev/input/event0 NEC.conf
```

```
irrecord: initializing '/dev/input/event0'
This program will record the signals from your remote control and create a config file for lircd.

Usually it's not necessary to create a new config file for devinput devices. A generic config file can be found at:
http://www.lirc.org/remotes/devinput/
You should try this config file before creating your own config file.

A proper config file for lircd is maybe the most vital part of this package, so you should invest some time to create a working config file. Although I put a good deal of effort in this program it is often not possible to automatically recognize all features of a remote control. Often short-comings of the receiver hardware make it nearly impossible. If you have problems to create a config file READ THE DOCUMENTATION of this package, especially section "Adding new remote controls" for how to get help.

If there already is a remote control of the same brand available at http://www.lirc.org/remotes/ you might also want to try using such a remote as a template. The config files already contain all parameters of the protocol used by remotes of a certain brand and knowing these parameters makes the job of this program much
```

Now, press the “Enter” key to start record your remote key code. First, you need to input a name for a key, you can’t define the key name by yourself, just run command like these to look at the legal key names.

```
irrecord -l
```
For example, input "KEY_0" after ENTER, there is a reminder tell you to press the corresponding remote key. It will begin the next record after record a key successfully, then repeat the process until all of your keys are recorded. ENTER to exit and save the records.

New record file will be saved as NEC.conf with new hardware parameters and key codes.

If your key codes are doubled like
Edit `NEC.conf.conf` and manually remove the second code 0x0000000000000000

Result

```
begin codes
  KEY_0 0x010052000000001 0x0000000000000000
  KEY_1 0x010016000000001 0x0000000000000000
end codes
```

Edit `NEC.conf.conf` and change the 'name' from NEC.conf.conf to something better, i.e. BananaPi-IR.

```
begin remote
  name BananaPi-IR
  bits 56
  flags SPACE ENC|CONST LENGTH
```

Finally copy `NEC.conf.conf` to `/etc/lirc/lircd.conf`

Now test lircd. Start (or restart) it with

```
/etc/init.d/lirc start
```

and run `irw` and press some keys on your remote:

```
irw
```

SUCCESS : LIRC is working fine!

Provided as example, this is my full `/etc/lirc/lircd.conf` working with a remote used for LeMedia.
# Please make this file available to others
# by sending it to <lirc@bartelmus.de>
#
# this config file was automatically generated
# using lirc-0.9.0-pre1(devinput) on Thu Sep 25 09:45:21 2014
#
# contributed by
#
# brand:                       NEC.conf.conf
# model no. of remote control:
# devices being controlled by this remote:
#
begin remote
  name  BananaPi-IR
  bits           56
  flags SPACE_ENC|CONST_LENGTH
  eps            30
  aeps          100
  header       9000  4500
  one           563  1687
  zero          563   562
  ptrail        563
  pre_data_bits   8
  pre_data       0x0
  gap          108000
  toggle_bit_mask 0x0
  frequency    38000
  duty_cycle   33

4.3 Resize SD card

Sometimes the image file is just 4GB or smaller. But the SD card is 8GB or 16GB. Then we want to use the left space of SD card. So we can resize the SD card.

<table>
<thead>
<tr>
<th>Code</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>KEY_0</td>
<td>$0x01001600000001$</td>
</tr>
<tr>
<td>KEY_1</td>
<td>$0x01000C00000001$</td>
</tr>
<tr>
<td>KEY_2</td>
<td>$0x01001800000001$</td>
</tr>
<tr>
<td>KEY_3</td>
<td>$0x01005E00000001$</td>
</tr>
<tr>
<td>KEY_4</td>
<td>$0x01000800000001$</td>
</tr>
<tr>
<td>KEY_5</td>
<td>$0x01001C00000001$</td>
</tr>
<tr>
<td>KEY_6</td>
<td>$0x01005A00000001$</td>
</tr>
<tr>
<td>KEY_7</td>
<td>$0x01004200000001$</td>
</tr>
<tr>
<td>KEY_8</td>
<td>$0x01005200000001$</td>
</tr>
<tr>
<td>KEY_9</td>
<td>$0x01004A00000001$</td>
</tr>
<tr>
<td>KEY_PLAYPAUSE</td>
<td>$0x01004300000001$</td>
</tr>
<tr>
<td>KEY_VOLUMEUP</td>
<td>$0x01001500000001$</td>
</tr>
<tr>
<td>KEY_VOLUMEDOWN</td>
<td>$0x01000700000001$</td>
</tr>
<tr>
<td>KEY_BACK</td>
<td>$0x01004400000001$</td>
</tr>
<tr>
<td>KEY_ENTER</td>
<td>$0x01000900000001$</td>
</tr>
<tr>
<td>KEY_FORWARD</td>
<td>$0x01004000000001$</td>
</tr>
<tr>
<td>KEY_KP0</td>
<td>$0x01004500000001$</td>
</tr>
<tr>
<td>KEY_KP1</td>
<td>$0x01004600000001$</td>
</tr>
<tr>
<td>KEY_KP2</td>
<td>$0x01004700000001$</td>
</tr>
<tr>
<td>KEY_KP3</td>
<td>$0x01001900000001$</td>
</tr>
<tr>
<td>KEY_KP4</td>
<td>$0x01000D00000001$</td>
</tr>
</tbody>
</table>
4.3.1 Raspbian

In Raspbian, you can use bpi-config command to easy resize sd card.

```
sudo bpi-config
```

Then select `expand_rootfs` to resize your SD card.

4.3.2 Other linux OS

From the command line or a terminal window enter the following:

```
sudo fdisk /dev/mmcblk0
```

Then type `p` to list the partition table

you should see two partitions. if you look in the last column labeled System you should have:

Linux

make a note of the start number for partition 2, you will need this later. though it will likely still be on the screen (just in case).

Next type `d` to delete a partition.

You will then be prompted for the number of the partition you want to delete. In the case above you want to delete both the Linux and Linux swap partitions.

So type `2`

Now you can resize the main partition.

**type n** to create a new partition.

This new partition needs to be a primary partition so **type p**.

Next enter **2** when prompted for a partition number.

You will now be prompted for the first sector for the new partition. **Enter the start number from the earlier step (the Linux partition)**

Next you will be prompted for the last sector you can **just hit enter** to accept the default which will utilize the remaining disk space.

**Type w** to save the changes you have made.
Next reboot the system with the following command:

```
sudo reboot
```

once the system has reboot and you are **back at the commandline** enter the following command:

```
sudo resize2fs /dev/mmcblk0p2
```
5 Configure your Banana Pro/Pi

5.1 How to Modify the script.bin file

script.bin is a binary configuration file (lubuntu) used by Allwinner SOC core driver or LiveSuit. And it contains the information that how to set the peripherals, port, and I/O pins of A10/A20 target board. Because we cannot compile the script.bin file directly, we have to modify the corresponding text file whose format is FEX. We can use the sunxi-tools to change binary file into text file or change it back. You can refer to more information as below:

http://linux-sunxi.org/Fex_Guide

5.1.1 Download sunxi-tools tool:

git clone git://github.com/linux-sunxi/sunxi-tools.git

Then go into sunxi-tools menu, and generate fex2bin and bin2fex file through “make” command.

```
bananapi@lemaker:~$ cd sunxi-tools/
bananapi@lemaker:~/sunxi-tools$ make
```

There may be make failure in the course, so we enter the following command

```
sudo apt-get update
dsudo apt-get install gcc
dsudo apt-get install pkg-config libusb-1.0
```

Then we enter the “make” command again.

5.1.2 How to get FEX file.

A. Get from AllWinner

Over there, it contains FEX file for different kinds of chipset such as A10 and A20, but currently banana Pi does not put their own FEX file in the Allwinner git, so the FEX file from Allwinner is just for study. So the FEX file from Allwinner cannot be used at this stage.
B. Get from image file.
Take the Raspbian_For_BananaPro_v1412 image file as an example. When we burn the image into MicroSD card, we can get the script.bin file from the /dev/mmcblk0p1 directory after the system booting, then we can use bin2fex tool to change it into compiled fex file.

```
bananapi@lemaker:~ $ sudo mount /dev/mmcblk0p1 /mnt/
bananapi@lemaker:~ $ cd /mnt/
bananapi@lemaker:~ $ ls
script.bin  System Volume Information  uEnv.txt  uImage
bananapi@lemaker:~ $ cd
bananapi@lemaker:~ $ ls
bash  Downloads  Libmp3  Music  text  Videos
Desktop  fex_configuration  lircd.conf  Pictures  sunxi-tools
Documents  hostap  melekbing  Public  Templates
bananapi@lemaker:~ $ sunxi-tools/bin2fex script.bin bananapro.fex
fexc-bin: script.bin: version: 0.1.2
fexc-bin: script.bin: size: 50564 (86 sections)
bananapi@lemaker:~ $ ls
bananapro.fex  Documents  hostap  melekbing  Public  Templates
bash  Downloads  Libmp3  Music  text  Videos
Desktop  fex_configuration  lircd.conf  Pictures  sunxi-tools
```

5.1.3 Compile FEX file.
As shown above, we have used bin2fex tool to get the fex file. We can open this fex file with text tool(such as nano). And then we can edit the fex file.

```
GNU nano 2.2.6    File: bananapi.fex
[product]
version = "100"
machine = "bananapro"

[target]
eraseflag = 0

[boot_clock]
boot_clock = 912
```
5.1.4 Generate scrip.bin file.

Use fex2bin tool to re-compile the modified fex file into script.bin file, then copy the new scrip.bin file to the first folder.

So far, we can insert the MicroSD card to Banana Pro, and reboot.

5.2 Auto Log in to system

In Lubuntu

Modify the 20-lubuntu.conf file.

```
sudo vim /etc/lightdm/lightdm.conf.d/20-lubuntu.conf
```

Insert the below lines after [SeatDefaults]:

```
autologin-user=yourloginname
autologin-user-timeout=0
```

5.3 Network Configuration

On Banana Pro, it has onboard WiFi adapter. On Banana Pi, you can connect a USB wifi adapter. Sometimes we hope connect wifi on boot, it is simple to achieve this by modifying network interface file.

5.3.1 Use WiFi as the station mode

Most people will use the on board WiFi to connect the router, so it is station mode.

(1) Load the WiFi driver

On Banana Pro, you need load the WiFi driver. Although we use AP6181, but the wifi driver is the same of the AP6210, so the driver name is called AP6210. You can use the command below to enable the WiFi:
sudo modprobe ap6210

You can see if the driver has been loaded by:

lsmod

If you want to auto load the WiFi driver when you boot the system. You can add "ap6210" into the /etc/modules file:

sudo nano /etc/modules

And then add the "ap6210" below the file, save and exit.

(2) Use the WiFi

Raspbian

On the Raspbian system, you can connect to the WiFi hot spot by using the WiFi Config application, as follows:

Lubuntu

On the Lubuntu system, you can connect to the WiFi hot spot by clicking the button which is at the right bottom corner of the desktop, as follows:
Linux without Desktop

If you use the Linux-like system without desktop, and recommend you to use wpa_supplicant tool to connect the WiFi hot spot, An example of the Debian as follows. Install wpa_supplicant

```bash
sudo apt-get install wpa_supplicant
```

Edit the wpa_supplicant.conf

```bash
vim /etc/wpa_supplicant/wpa_supplicant.conf
```

Add the content below into the wpa_supplicant.conf:

```bash
ap_scan=1  # use the wpa_supplicant to scan and choose the AP
network={
    ssid="your_wifi_name"
    psk="your_wifi_password"
}
```

If the wpa_supplicant.conf is not exist, you need create one. ssid: the WiFi hot spot id name, psk: the password of the WiFi hot spot. Edit the /etc/network/interfaces

```bash
vim /etc/network/interfaces
```

Edit the wlan part of the interfaces file:

```bash
auto wlanx
```
iface wlanx inet dhcp
pre-up wpa_supplicant -B -i wlanx -c/etc/wpa_supplicant
/wpa_supplicant.conf
pre-down killall -q wpa_supplicant

Note: wlanx represent wlan0,wlan1 ...you can enter the command `ifconfig` to see what the x should be.

5.3.2 WiFi AP mode

If you want to use the AP mode of the AP6181 WiFi module on the Banana Pro, you can refer to the instructions below.

(1) Edit the `/etc/modules`

Edit the `/etc/modules` file, and add the content below into the file:

```
ap6210 op_mode=2
```

(2) Download and compile the hostapd

```
git clone git://w1.fi/srv/git/hostap.git
sudo apt-get install libnl-dev
sudo apt-get install openssl
sudo apt-get install libssl-dev
cd hostap/hostapd
cp defconfig .config
make
cp hostapd /usr/local/bin
```

(3) Create and edit the hostapd.conf

```
mkdir -p /etc/hostapd
vim /etc/hostapd/hostapd.conf
```

For example add the content below into the hostapd.conf file:

```
interface=wlanx
```
You can change the ssid and wpa_passphrase.

(4) Edit the network interfaces

vim /etc/network/interfaces

For example replace the content below:

auto lo
iface lo inet loopback
iface eth0 inet dhcp
allow-hotplug wlanx
iface wlanx inet static
address 192.168.100.1
netmask 255.255.255.0

(5) Enable the DHCP server

1) Install the DHCP server

sudo apt-get install udhcpd

2) Edit the udhcpd.conf
vim /etc/udhcpd.conf

For example add the content below into the file:

```
#The start and end of the IP lease block
start 192.168.100.101   #default: 192.168.0.20
end 192.168.100.254     #default: 192.168.0.254
#The interface that udhcpd will use
interface wlanx          #default: eth0
#Examples
option subnet 255.255.255.0
opt router 192.168.100.1
opt wins 192.168.100.1
option dns 192.168.100.1
option domain local
option lease 864000
```

3) **Reboot and restart the server**

Run the DHCP server:

udhcpd /etc/udhcpd.conf

Run the hostapd:

hostapd -B /etc/hostapd/hostapd.conf

And then you can check the result:

iwconfig

(6) **Bridge**

1). **Install bridge-utils**

sudo apt-get install bridge-utils

2). **Add a new bridge**

sudo brctl addbr br0
3) Close STP (not necessary)

```bash
sudo brctl stp br0 off
```

4) Set an IP address for br0, for example 192.168.1.100

```bash
sudo ifconfig br0 192.168.1.100
```

5) Add eth0&wlanx to br0

```bash
sudo brctl addif br0 eth0
sudo brctl addif br0 wlanx
```

6) Start br0

```bash
sudo ifconfig br0 up
```

When you Setup completed you can use the pro as the wireless switch to connect the internet.

5.4 Turn off the screen saver

Edit the file /etc/lightdm/lightdm.conf, in the **SeatDefaults** section, make it looks like below:

```
[SeatDefaults]
xserver-command=X -s 0 -dpms
```

5.5 Auto mount storage device

Edit /etc/udev/rules.d/10-usbstorage.rules:

```
[SeatDefaults]
xserver-command=X -s 0 -dpms
KERNEL!="sd*", GOTO="media_by_label_auto_mount_end"
SUBSYSTEM!="block", GOTO="media_by_label_auto_mount_end"
IMPORT{program}="/sbin/blkid -o udev -p %N"
ENV{ID_FS_TYPE}"", GOTO="media_by_label_auto_mount_end"
ENV{ID_FS_LABEL}"", ENV{dir_name}="%E{ID_FS_LABEL}"
```
ENV{ID_FS_LABEL}=="", ENV{dir_name}="Untitled-%k"
ACTION=="add", ENV{mount_options}="relatime.sync"
ACTION=="add", ENV{ID_FS_TYPE}=="vfat", ENV{mount_options}="iocharset=utf8,umask=000"
ACTION=="add", ENV{ID_FS_TYPE}=="ntfs", ENV{mount_options}="iocharset=utf8,umask=000"
ACTION=="add", RUN+="/bin/mkdir -p /media/%E{dir_name}", RUN="/bin/mount -o $env{mount_options} /dev/%k /media/%E{dir_name}"
ACTION=="remove", ENV{dir_name}=="", RUN="/bin/umount -l /media/%E{dir_name}",
RUN="/bin/rmdir /media/%E{dir_name}"
LABEL="media_by_label_auto_mount_end"

5.6 Add desktop shortcut

Raspbian
Add a shortcut in desktop, for example add Arduino-1.5.4 application.
Create a file called arduino.desktop:

```
[Desktop Entry]
Type=Application
Encoding=UTF-8
Name=Arduino-1.5.4
Comment=Programming system and content development tools
Terminal=false
Categories=Programming
Exec=/opt/arduino-1.5.4/run.sh
Icon=/usr/share/pixmaps/arduino.xpm
```

And then

```
cp arduino.desktop $HOME/Desktop
```
6 Applications

6.1 Using Banana Pro to build NAS home server

I have an old computer not to be used that just have the computer case, hard drive and a power supply. If you don’t use those devices they will be wasted. So I use the surplus equipment and Banana Pro (Using Banana Pi can achieve the same effect) set up my NAS home server.

6.1.1 The main function of the NAS home server

(1) **File server**: Use samba to realize the file sharing.

(2) **Download machine**: Use aria2c to realize the automatic download.

(3) **Streaming media server**: DLNA realize the use of computer or mobile phone playing the media files that were placed in the sharing directory.

The steps are as follows.

6.1.2 Preparation work

(1) **Write the OS image file Raspbian for BananaPro to MicroSD card**
(2) The preparation of hardware

1) Use the power supply to supply power for the hard drive
2) Connect the hard drive to the SATA port on the B-Pro
3) Plug in an RJ-45 Ethernet
4) Plugging in the mains adapter power on the Banana Pro.

(3) Format the hard drive to create a new partition and mount the root directory to the hard drive

1) Format the hard drive to create a new partition

Run the `sudo fdisk /dev/sda` command. Use the `o` command to delete all partition of hard drive and use the `n` command to add one new partition. Use the `w` command to save change.

Run the `sudo mkfs.ext4 /dev/sda1` command to format the new created partition of hard drive as ext4.

2) Mount the root directory to your hard drive

When we mount the root directory to the hard drive, the file system will start in the hard drive. This setting is a good way to protect our MicroSD card, when we need a long time to use the system. Specific methods are as follows.

Create a temporary directory in the root directory

```
mkdir temp
sudo fdisk -l
```

Mount the `/dev/mmcblk0p2` to the temp directory

```
sudo mount /dev/mmcblk0p2 /temp
```

Copy the file system to the hard drive partition

```
sudo mount /dev/sda1 /mnt
cp -aP /temp /mnt
```

Edit the `/boot/uEnv.txt`,

```
sudo nano uEnv.txt
```
Change the content `root=/dev/mmcblk0p2` as follow

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

(4) Close the GUI

```bash
sudo nano /etc/X11/default-display-manager
```

Add `false` in the file

```
/usr/sbin/lightdm
false
```

Reboot the system then graphical interface is no longer present. Automatically run in the CUI mode when the system reboot. The machine does not need to use the GUI. The graphical interface for the B-Pro was closed to save computational resources.

### 6.1.3 Create a file server

Use the file server can achieve LAN file sharing. Through this function we can upload files to the shared directory or download files from the shared directory. Here we use the samba software to realize our LAN sharing.

(1) Set up static IP

Edit `/etc/network/interfaces`

```bash
sudo nano /etc/network/interfaces
```

Change the `iface eth0 inet dhcp` as follow

```bash
iface eth0 inet static
```

Add the content as follow

```bash
address 192.168.1.15
netmask 255.255.255.0
gateway 192.168.1.1
```

And then delete `iface default inet dhcp`

Here the static IP is set to 192.168.1.15. Restart the network or reboot the system to
(2) Installation of samba services

sudo apt-get update
sudo apt-get install samba
sudo apt-get install samba-common-bin

(3) Add user and create the download directory

sudo useradd maker

To set a password for the new user, this is set to 123

sudo passwd maker
sudo mkdir /aria2download/download
sudo chown –R maker /aria2download/download

(4) Edit the samba configuration file

sudo nano /etc/samba/smb.conf

To find out the security = user and delete the symbol # in front of the sentence.

Add the content at the end of the file.

[aria2share]
comment=my family’s share
path=/aria2download/download
valid users=maker
public=no
writable=yes
browseable=yes
create mask=0777
directory mask=0777
Aria2share is the sharing file’s name that we can see on the computer

(5) Add Samba user

Here we set Samba password is 321

```
sudo smbpasswd -a maker
sudo service samba restart
```

So far we have established the file server based on samba. Type the Winkey+R and input our IP address of the server. Then we will see our sharing file on the computer. On the mobile phone use software ES file explorer to view the sharing file. In my network page of the ES file explorer click the new option and input the path of our sharing file like IP/sharing filename then input user name with password. Here I input 192.168.1.15/aria2share. User name is maker and password is 321. We can manage our shared file in our phone now.

6.1.4 Download machine

(1) Set up aria2

1) Install aria2

```
sudo apt-get install aria2
```

Create a new directory and file

```
sudo mkdir /etc/aria2
sudo touch /etc/aria2/aria2.session
```

2) To configure the aria2

```
sudo nano /etc/aria2/aria2.conf
```

# Basic Options

- dir=/aria2download/download
- input-file=/etc/aria2/aria2.session
- max-concurrent-downloads=20
- continue=true
- lowest-speed-limit=0
- max-connection-per-server=5
- min-split-size=10M
- split=5

# Bt Options

- bt-enable-lpd=true
- bt-max-open-files=100
- bt-max-peers=55
- bt-min-crypto-level=plain
- bt-require-crypto=true
- follow-torrent=true
- listen-port=6881-6999
- max-overall-upload-limit=0
- max-upload-limit=0
- seed-ratio=1
- seed-time=120

# RPC Options

- enable-rpc=true
- rpc-allow-origin-all=true
- rpc-listen-all=true

# Advanced Options

- disable-ipv6=true
- file-allocation=falloc
max-overall-download-limit=0
max-download-limit=0
save-session=/etc/aria2/aria2.session
save-session-interval=300

Save configuration and test the function

```bash
aria2c --conf-path=/etc/aria2/aria2.conf -D
```

To see whether aria2 is running

```bash
ps -ef|grep aria2
```

3) Enable aria2 boot as the system start

Edit /etc/rc.local, and insert the following contents before `exit0`.

```bash
sudo aria2c --conf-path=/etc/aria2/aria2.conf -D
```

The next time you start Pro, aria2 will automatically run

(2) **To build web server**

1) Install nginx

```bash
sudo apt-get install nginx
```

Create a new directory

```bash
sudo mkdir /aria2download/html
sudo chmod 777 /aria2download/html
```

2) To configure the nginx

```bash
sudo nano /etc/nginx/sites-available/default
```

find out sentence `#listen 80;` and delete the symbol #. Underneath change the sentence

```
root /usr/share/nginx/html to root /aria2download/html
```

```
server {
listen 80; ## listen for ipv4; this line is default and implied
#listen [::]:80 default_server ipv6only=on; ## listen for ipv6
root /aria2download/html;
index index.html index.htm;
```
3) Reload and restart the nginx

```bash
sudo /etc/init.d/nginx reload
sudo /etc/init.d/nginx restart
```

4) Install yaaw

Download the yaaw from [https://github.com/binux/yaaw](https://github.com/binux/yaaw)

After download, extract the compressed file and copy all of the contents under the directory `yaaw-master` to the directory `/aria2download/html`.

Now we use Google Chrome or Firefox browse to open the webpage address `192.168.1.15` on our computer then you can see the page as follow.
6.1.5 Install the DLNA steaming media server

DLNA is mainly oriented to media resources (such as video, music) to realize the network sharing. The setting steps as follows:

(1) Install minidlna

```
sudo apt-get install minidlna
```

(2) Configure the minidlna

Create the directory

```
sudo mkdir /aria2download/minidlna
sudo mkdir /aria2download/minidlna/dd
sudo mkdir /aria2download/minidlna/log
sudo chmod –r 777 /aria2download/minidlna
```

Edit /etc/minidlna.conf

```
sudo nano /etc/minidlna.conf
```
Find out `media_dir=/var/lib/minidlna` and replace it

```
media_dir=/aria2download/download
```

Find out `#db_dir=/var/lib/minidlna` and replace it

```
db_dir=/aria2download/minidlna/db
```

Find out `#log_dir=/var/log` and replace it

```
log_dir=/aria2download/minidlna/log
```

Last find out the default minidlna database and log files then delete it to avoid confusion

```
rm -r /var/lib/minidlna
rm -r /var/log
```

(3) Restart minidlna

```
sudo /etc/init.d/minidlna restart
```

To see whether aria2 is running

```
/etc/init.d/minidlna status
```

We have successfully established a streaming media server. But we need to turn off the firewall or allow the Windows Media Player through the firewall before we use it. Now we can see the sharing streaming media files by the Windows Media Player in the LAN.
We also can see it on our mobile phone via the streaming media player software. Here I use moliplayer.

NAS home server finally completed! We can put the server on a corner of home working for us, now.
6.2 XBMC

Note that the 4&5 step can be used to support the hardware acceleration for Mplayer to compared the playback with XBMC, so it is not necessary to compile the XBMC and you can skip.

To install the XBMC, you should use 8G or larger SD card and resize SD card, please reference 4.3 section first.

6.2.1 Modprobe needed modules:

```text
modprobe sunxi_cedar_mod
modprobe mali_drm
modprobe drm
modprobe mali
modprobe ump
```

6.2.2 Install UMP library

http://linux-sunxi.org/Mali_binary_driver

The main steps are follows:

(1) Prequisites

```text
sudo apt-get install git build-essential autoconf libtool
```

(2) Clone the repository

```text
git clone https://github.com/linux-sunxi/libump.git
cd libump
```

(3) Manual Build

```text
sudo apt-get update
sudo apt-get install debhelper dh-autoreconf pkg-config
autoreconf -i
```
6.2.3 Install Xorg()

http://linux-sunxi.org/Xorg#fbturbo_driver

The main steps are follows:

1) Prerequisites

```
sudo apt-get install xorg-dev xutils-dev x11proto-dri2-dev xorg libdrm-dev libltdl-dev automake
```

2) Clone the repository

```
git clone -b 0.4.0 https://github.com/ssvb/xf86-video-fbturbo.git
cd xf86-video-fbturbo
```

3) Build

```
autoreconf -vi
./configure --prefix=/usr
make
```

4) Installation

```
make install
```

5) Configuration

```
sudo rm /usr/share/X11/xorg.conf.d/99-sunxifb.conf
cp xorg.conf /etc/X11/xorg.conf
```

6.2.4 Install libvdpau

```
sudo apt-get install libvdpau-dev vdpauinfo
git clone https://github.com/linux-sunxi/libvdpau-sunxi
cd libvdpau-sunxi
```
6.2.5 Check the hardware acceleration

```
export DISPLAY=:0
export VDPAU_DRIVER=sunxi
vdpauinfo
```

(1) Make sure you have /dev/disp /dev/g2d /dev/mali /dev/fb*

```
sudo apt-get install mplayer
```

(2) Edit config file (~/.mplayer/config)

```
vo=vdpau
vc=ffmpeg12vdpau,ffh264vdpau
fullscreen=yes
quiet=yes
ao=pulse
framedrop=yes
cache=8192
lavdopts=threads=2
ass=no
ass-font-scale=1.4
ass-color=FFFFFF00
ass-border-color=00000000
ass-use-margins=yes
ass-bottom-margin=50
spualign=2
subalign=2
subfont=/usr/share/fonts/truetype/ttf-dejavu/DejaVuSans.ttf
subcp=cp1250
```
6.2.6 Building XBMC for A10 and A20 SoC series

(1) Install the dependencies for XBMC (needs update!)

```
sudo apt-get build-dep xbmc
sudo apt-get update
sudo apt-get install swig default-jre libgtk2.0-bin libssh2-4 libssh-dev libxml2-dev
```

(2) Ensure you use hardware acceleration: (to set it permanently)

```
echo -e "\nA10HWR=1" >> /etc/environment
```

(3) Prerequisites for native compile

Create a swap-file, because otherwise the compiler runs out of memory during compiling and aborts

```
dd if=/dev/zero of=/swap bs=1M count=384
mkswap -c /swap
swapon /swap
```

(4) Create your workspace directory:

```
sudo mkdir melehacking
cd melehacking
```

(5) Checkout the source code

```
git clone https://github.com/warped-rudi/xbmc.git
cd xbmc
git checkout origin/Gotham-A10
```

(6) Build (network is needed!)

Build dependencies

```
cd tools/a10/depends
sudo mkdir -p /opt/a10hacking/xbmctmp/tarballs
make # (here will be download 5 library: taglib, cedarx, libmad, mali, mali-dev)
```

Build xbmc itself
6.2.7 Start XBMC

```bash
echo A10HWR=1 (ensure to have this set if not rebooting!)
cd /allwinner/xbmc-pvr-bin/lib/xbmc
./xbmc.bin
```

Dear reader, it may still exist many imperfect places, readers can tell us, thank you for your support. We are doing all we can.
# Appendix A: Banana Pro pin definition

<table>
<thead>
<tr>
<th>PIN</th>
<th>PIN define</th>
<th>GPIO</th>
</tr>
</thead>
<tbody>
<tr>
<td>CON5-P01</td>
<td>LINEINL</td>
<td></td>
</tr>
<tr>
<td>CON5-P02</td>
<td>LINEINR</td>
<td></td>
</tr>
<tr>
<td>CON5-P37</td>
<td>HPL</td>
<td></td>
</tr>
<tr>
<td>CON5-P36</td>
<td>HPR</td>
<td></td>
</tr>
<tr>
<td>CON5-P07</td>
<td>FMINL</td>
<td></td>
</tr>
<tr>
<td>CON5-P09</td>
<td>FMINR</td>
<td></td>
</tr>
<tr>
<td>CON5-P04</td>
<td>ADC_X1</td>
<td></td>
</tr>
<tr>
<td>CON5-P06</td>
<td>ADC_X2</td>
<td></td>
</tr>
<tr>
<td>CON5-P08</td>
<td>ADC_Y1</td>
<td></td>
</tr>
<tr>
<td>CON5-P10</td>
<td>ADC_Y2</td>
<td></td>
</tr>
<tr>
<td>CON5-P13</td>
<td>LRADC0</td>
<td></td>
</tr>
<tr>
<td>CON5-P15</td>
<td>LRADC1</td>
<td></td>
</tr>
<tr>
<td>CON5-P33</td>
<td>RESET#</td>
<td></td>
</tr>
<tr>
<td>CON5-P17</td>
<td>CSI-D0</td>
<td>PE4</td>
</tr>
<tr>
<td>CON5-P19</td>
<td>CSI-D1</td>
<td>PE5</td>
</tr>
<tr>
<td>CON5-P21</td>
<td>CSI-D2</td>
<td>PE6</td>
</tr>
<tr>
<td>CON5-P23</td>
<td>CSI-D3</td>
<td>PE7</td>
</tr>
<tr>
<td>CON5-P25</td>
<td>CSI-D4</td>
<td>PE8</td>
</tr>
<tr>
<td>CON5-P27</td>
<td>CSI-D5</td>
<td>PE9</td>
</tr>
<tr>
<td>CON5-P29</td>
<td>CSI-D6</td>
<td>PE10</td>
</tr>
<tr>
<td>CON5-P31</td>
<td>CSI-D7</td>
<td>PE11</td>
</tr>
<tr>
<td>CON5-P24</td>
<td>CSI-PCLK</td>
<td>PE0</td>
</tr>
<tr>
<td></td>
<td>CSI-MCLK</td>
<td>PE1</td>
</tr>
<tr>
<td>CON5-P28</td>
<td>CSI-VSYNC</td>
<td>PE3</td>
</tr>
<tr>
<td>----------</td>
<td>-----------</td>
<td>------</td>
</tr>
<tr>
<td>CON5-P30</td>
<td>CSI-HSYNC</td>
<td>PE2</td>
</tr>
<tr>
<td>CON5-P18</td>
<td>CSI0-STBY-EN</td>
<td>PH19</td>
</tr>
<tr>
<td>CON5-P26</td>
<td>CSI0-RESET#</td>
<td>PH14</td>
</tr>
<tr>
<td>CON5-P32</td>
<td>CSI1-STBY-EN</td>
<td>PH18</td>
</tr>
<tr>
<td>CON5-P34</td>
<td>CSI1-RESET#</td>
<td>PH13</td>
</tr>
<tr>
<td>CON5-P14</td>
<td>TWI1-SDA</td>
<td>PB19</td>
</tr>
<tr>
<td>CON5-P16</td>
<td>TWI1-SCK</td>
<td>PB18</td>
</tr>
<tr>
<td>CON5-P12</td>
<td>CSI-FLASH</td>
<td>PH17</td>
</tr>
<tr>
<td>CON5-P22</td>
<td>CSI0-PWR-EN</td>
<td>PH16</td>
</tr>
<tr>
<td>CON5-P35</td>
<td>CSI-IO0</td>
<td>PH11</td>
</tr>
<tr>
<td>CON5-P38</td>
<td>IPSOUT</td>
<td></td>
</tr>
<tr>
<td>CON5-P40</td>
<td>IPSOUT</td>
<td></td>
</tr>
<tr>
<td>CON5-P05</td>
<td>GND</td>
<td></td>
</tr>
<tr>
<td>CON5-P11</td>
<td>GND</td>
<td></td>
</tr>
<tr>
<td>CON5-P39</td>
<td>GND</td>
<td></td>
</tr>
<tr>
<td>CON5-P03</td>
<td>VCC-CSI</td>
<td></td>
</tr>
<tr>
<td>CON2-P09</td>
<td>LCD0-D00</td>
<td>PD0</td>
</tr>
<tr>
<td>CON2-P11</td>
<td>LCD0-D01</td>
<td>PD1</td>
</tr>
<tr>
<td>CON2-P13</td>
<td>LCD0-D02</td>
<td>PD2</td>
</tr>
<tr>
<td>CON2-P15</td>
<td>LCD0-D03</td>
<td>PD3</td>
</tr>
<tr>
<td>CON2-P17</td>
<td>LCD0-D04</td>
<td>PD4</td>
</tr>
<tr>
<td>CON2-P19</td>
<td>LCD0-D05</td>
<td>PD5</td>
</tr>
<tr>
<td>CON2-P21</td>
<td>LCD0-D06</td>
<td>PD6</td>
</tr>
<tr>
<td>CON2-P23</td>
<td>LCD0-D07</td>
<td>PD7</td>
</tr>
<tr>
<td>CON2-P25</td>
<td>LCD0-D08</td>
<td>PD8</td>
</tr>
<tr>
<td>CON2-P27</td>
<td>LCD0-D09</td>
<td>PD9</td>
</tr>
<tr>
<td>CON2-P29</td>
<td>LCD0-D10</td>
<td>PD10</td>
</tr>
<tr>
<td>CON2-P31</td>
<td>LCD0-D11</td>
<td>PD11</td>
</tr>
<tr>
<td>----------</td>
<td>----------</td>
<td>------</td>
</tr>
<tr>
<td>CON2-P33</td>
<td>LCD0-D12</td>
<td>PD12</td>
</tr>
<tr>
<td>CON2-P35</td>
<td>LCD0-D13</td>
<td>PD13</td>
</tr>
<tr>
<td>CON2-P37</td>
<td>LCD0-D14</td>
<td>PD14</td>
</tr>
<tr>
<td>CON2-P39</td>
<td>LCD0-D15</td>
<td>PD15</td>
</tr>
<tr>
<td>CON2-P40</td>
<td>LCD0-D16</td>
<td>PD16</td>
</tr>
<tr>
<td>CON2-P38</td>
<td>LCD0-D17</td>
<td>PD17</td>
</tr>
<tr>
<td>CON2-P36</td>
<td>LCD0-D18</td>
<td>PD18</td>
</tr>
<tr>
<td>CON2-P34</td>
<td>LCD0-D19</td>
<td>PD19</td>
</tr>
<tr>
<td>CON2-P32</td>
<td>LCD0-D20</td>
<td>PD20</td>
</tr>
<tr>
<td>CON2-P30</td>
<td>LCD0-D21</td>
<td>PD21</td>
</tr>
<tr>
<td>CON2-P28</td>
<td>LCD0-D22</td>
<td>PD22</td>
</tr>
<tr>
<td>CON2-P26</td>
<td>LCD0-D23</td>
<td>PD23</td>
</tr>
<tr>
<td>CON2-P22</td>
<td>LCD0-CLK</td>
<td>PD24</td>
</tr>
<tr>
<td>CON2-P20</td>
<td>LCD0-CS</td>
<td>PH6</td>
</tr>
<tr>
<td>CON2-P18</td>
<td>LCD0-HSYNC</td>
<td>PD26</td>
</tr>
<tr>
<td>CON2-P16</td>
<td>LCD0-VSYNC</td>
<td>PD27</td>
</tr>
<tr>
<td>CON2-P14</td>
<td>LCD0-DE</td>
<td>PD25</td>
</tr>
<tr>
<td>CON2-P12</td>
<td>LCD0-IO2</td>
<td>PH9</td>
</tr>
<tr>
<td>CON2-P10</td>
<td>PWM0</td>
<td>PB2</td>
</tr>
<tr>
<td>CON2-P08</td>
<td>LCD0-IO1</td>
<td>PH8</td>
</tr>
<tr>
<td>CON2-P06</td>
<td>LCD0-IO0</td>
<td>PH7</td>
</tr>
<tr>
<td>CON2-P04</td>
<td>TWI3-SCK</td>
<td>PI0</td>
</tr>
<tr>
<td>CON2-P02</td>
<td>TWI3-SDA</td>
<td>PI1</td>
</tr>
<tr>
<td>CON2-P07</td>
<td>LCDIO-03</td>
<td>PH12</td>
</tr>
<tr>
<td>CON2-P01</td>
<td>IPSOUT</td>
<td></td>
</tr>
<tr>
<td>CON2-P03</td>
<td>IPSOUT</td>
<td></td>
</tr>
<tr>
<td>CON2-P05</td>
<td>GND</td>
<td></td>
</tr>
<tr>
<td>Component</td>
<td>Description</td>
<td>Pin</td>
</tr>
<tr>
<td>-----------</td>
<td>---------------------------</td>
<td>------</td>
</tr>
<tr>
<td>CON2-P24</td>
<td>GND</td>
<td>PH21</td>
</tr>
<tr>
<td>CON6-P18</td>
<td>IO-5(CAN_RX)</td>
<td>PH20</td>
</tr>
<tr>
<td>CON6-P16</td>
<td>IO-4(CAN_TX)</td>
<td>PH20</td>
</tr>
<tr>
<td>CON6-P23</td>
<td>SPI0_CLK</td>
<td>PI11</td>
</tr>
<tr>
<td>CON6-P21</td>
<td>SPI0_MISO</td>
<td>PI13</td>
</tr>
<tr>
<td>CON6-P19</td>
<td>SPI0_MOSI</td>
<td>PI12</td>
</tr>
<tr>
<td>CON6-P24</td>
<td>SPI0_CS0</td>
<td>PI10</td>
</tr>
<tr>
<td>CON6-P26</td>
<td>SPI0_CS1</td>
<td>PI14</td>
</tr>
<tr>
<td>CON6-P05</td>
<td>TWI2-SCK</td>
<td>PB20</td>
</tr>
<tr>
<td>CON6-P03</td>
<td>TWI2-SDA</td>
<td>PB21</td>
</tr>
<tr>
<td>CON6-P28</td>
<td>TWI3-SCK</td>
<td>PI10</td>
</tr>
<tr>
<td>CON6-P27</td>
<td>TWI3-SDA</td>
<td>PI11</td>
</tr>
<tr>
<td>CON6-P15</td>
<td>IO-3(UART2_CTS)</td>
<td>PI17</td>
</tr>
<tr>
<td>CON6-P22</td>
<td>IO-6(UART2_RTS)</td>
<td>PI16</td>
</tr>
<tr>
<td>CON6-P11</td>
<td>IO-0(UART2_RX)</td>
<td>PI19</td>
</tr>
<tr>
<td>CON6-P13</td>
<td>IO-2(UART2_TX)</td>
<td>PI18</td>
</tr>
<tr>
<td>CON6-P10</td>
<td>UART4_RX</td>
<td>PH5</td>
</tr>
<tr>
<td>CON6-P08</td>
<td>UART4_TX</td>
<td>PH4</td>
</tr>
<tr>
<td>CON6-P31</td>
<td>UART7_RX</td>
<td>PI21</td>
</tr>
<tr>
<td>CON6-P32</td>
<td>UART7_TX</td>
<td>PI20</td>
</tr>
<tr>
<td>CON6-P07</td>
<td>IO-1</td>
<td>PH2</td>
</tr>
<tr>
<td>CON6-P29</td>
<td>IO-7(IR0_TX/SPDIF_MCLK)</td>
<td>PB3</td>
</tr>
<tr>
<td>CON6-P33</td>
<td>IO-8(SPDIF_DO)</td>
<td>PB13</td>
</tr>
<tr>
<td>CON6-P12</td>
<td>PWM1</td>
<td>PI3</td>
</tr>
<tr>
<td>CON6-P35</td>
<td>I2S0_LRCK</td>
<td>PB07</td>
</tr>
<tr>
<td>CON6-P36</td>
<td>I2S0_BCLK</td>
<td>PB06</td>
</tr>
<tr>
<td>CON6-P37</td>
<td>I2S0_MCLK</td>
<td>PB05</td>
</tr>
<tr>
<td>CON6-P38</td>
<td>I2S0_DI</td>
<td>PB12</td>
</tr>
<tr>
<td>CON6-P40</td>
<td>I2S0_D00</td>
<td>PB08</td>
</tr>
<tr>
<td>----------</td>
<td>----------</td>
<td>------</td>
</tr>
<tr>
<td>CON6-P01</td>
<td>VCC-3V3</td>
<td></td>
</tr>
<tr>
<td>CON6-P17</td>
<td>VCC-3V3</td>
<td></td>
</tr>
<tr>
<td>CON6-P02</td>
<td>VCC-5V</td>
<td></td>
</tr>
<tr>
<td>CON6-P04</td>
<td>VCC-5V</td>
<td></td>
</tr>
<tr>
<td>CON6-P09</td>
<td>GND</td>
<td></td>
</tr>
<tr>
<td>CON6-P25</td>
<td>GND</td>
<td></td>
</tr>
<tr>
<td>CON6-P06</td>
<td>GND</td>
<td></td>
</tr>
<tr>
<td>CON6-P14</td>
<td>GND</td>
<td></td>
</tr>
<tr>
<td>CON6-P20</td>
<td>GND</td>
<td></td>
</tr>
<tr>
<td>CON6-P30</td>
<td>GND</td>
<td></td>
</tr>
<tr>
<td>CON6-P34</td>
<td>GND</td>
<td></td>
</tr>
<tr>
<td>CON6-P39</td>
<td>GND</td>
<td></td>
</tr>
<tr>
<td>LED1</td>
<td>GREEN LED</td>
<td>PH24</td>
</tr>
<tr>
<td>LED2</td>
<td>BLUE LED</td>
<td>PG2</td>
</tr>
</tbody>
</table>
## Appendix B : Banana Pi pin definition

<table>
<thead>
<tr>
<th>PIN</th>
<th>PIN define</th>
<th>GPIO</th>
</tr>
</thead>
<tbody>
<tr>
<td>CON1-P01</td>
<td>LINEINL</td>
<td></td>
</tr>
<tr>
<td>CON1-P02</td>
<td>LINEINR</td>
<td></td>
</tr>
<tr>
<td>CON1-P37</td>
<td>HPL</td>
<td></td>
</tr>
<tr>
<td>CON1-P36</td>
<td>HPR</td>
<td></td>
</tr>
<tr>
<td>CON1-P07</td>
<td>FMINL</td>
<td></td>
</tr>
<tr>
<td>CON1-P09</td>
<td>FMINR</td>
<td></td>
</tr>
<tr>
<td>CON1-P04</td>
<td>ADC_X1</td>
<td></td>
</tr>
<tr>
<td>CON1-P06</td>
<td>ADC_X2</td>
<td></td>
</tr>
<tr>
<td>CON1-P08</td>
<td>ADC_Y1</td>
<td></td>
</tr>
<tr>
<td>CON1-P10</td>
<td>ADC_Y2</td>
<td></td>
</tr>
<tr>
<td>CON1-P13</td>
<td>LRADC0</td>
<td></td>
</tr>
<tr>
<td>CON1-P15</td>
<td>LRADC1</td>
<td></td>
</tr>
<tr>
<td>CON1-P33</td>
<td>RESET#</td>
<td></td>
</tr>
<tr>
<td>CON1-P17</td>
<td>CSI-D0</td>
<td>PE4</td>
</tr>
<tr>
<td>CON1-P19</td>
<td>CSI-D1</td>
<td>PE5</td>
</tr>
<tr>
<td>CON1-P21</td>
<td>CSI-D2</td>
<td>PE6</td>
</tr>
<tr>
<td>CON1-P23</td>
<td>CSI-D3</td>
<td>PE7</td>
</tr>
<tr>
<td>CON1-P25</td>
<td>CSI-D4</td>
<td>PE8</td>
</tr>
<tr>
<td>CON1-P27</td>
<td>CSI-D5</td>
<td>PE9</td>
</tr>
<tr>
<td>CON1-P29</td>
<td>CSI-D6</td>
<td>PE10</td>
</tr>
<tr>
<td>CON1-P31</td>
<td>CSI-D7</td>
<td>PE11</td>
</tr>
<tr>
<td>CON1-P20</td>
<td>CSI-PCLK</td>
<td>PE0</td>
</tr>
<tr>
<td>CON1-P24</td>
<td>CSI-MCLK</td>
<td>PE1</td>
</tr>
<tr>
<td>CON1-P28</td>
<td>CSI-VSYNC</td>
<td>PE3</td>
</tr>
<tr>
<td>CON1-P30</td>
<td>CSI-HSYNC</td>
<td>PE2</td>
</tr>
<tr>
<td>CON1-P18</td>
<td>CSI0-STBY-EN</td>
<td>PH19</td>
</tr>
<tr>
<td>---------</td>
<td>-------------</td>
<td>------</td>
</tr>
<tr>
<td>CON1-P26</td>
<td>CSI0-RESET#</td>
<td>PH14</td>
</tr>
<tr>
<td>CON1-P32</td>
<td>CSI1-STBY-EN</td>
<td>PH18</td>
</tr>
<tr>
<td>CON1-P34</td>
<td>CSI1-RESET#</td>
<td>PH13</td>
</tr>
<tr>
<td>CON1-P14</td>
<td>TWI1-SDA</td>
<td>PB19</td>
</tr>
<tr>
<td>CON1-P16</td>
<td>TWI1-SCK</td>
<td>PB18</td>
</tr>
<tr>
<td>CON1-P12</td>
<td>CSI-FLASH</td>
<td>PH17</td>
</tr>
<tr>
<td>CON1-P22</td>
<td>CSI0-PWR-EN</td>
<td>PH16</td>
</tr>
<tr>
<td>CON1-P35</td>
<td>CSI-I00</td>
<td>PH11</td>
</tr>
<tr>
<td>CON1-P38</td>
<td>IPSOUT</td>
<td></td>
</tr>
<tr>
<td>CON1-P40</td>
<td>IPSOUT</td>
<td></td>
</tr>
<tr>
<td>CON1-P05</td>
<td>GND</td>
<td></td>
</tr>
<tr>
<td>CON1-P11</td>
<td>GND</td>
<td></td>
</tr>
<tr>
<td>CON1-P39</td>
<td>GND</td>
<td></td>
</tr>
<tr>
<td>CON1-P03</td>
<td>VCC-CSI</td>
<td></td>
</tr>
<tr>
<td>CON2-P09</td>
<td>LCD0-D00</td>
<td>PD0</td>
</tr>
<tr>
<td>CON2-P11</td>
<td>LCD0-D01</td>
<td>PD1</td>
</tr>
<tr>
<td>CON2-P13</td>
<td>LCD0-D02</td>
<td>PD2</td>
</tr>
<tr>
<td>CON2-P15</td>
<td>LCD0-D03</td>
<td>PD3</td>
</tr>
<tr>
<td>CON2-P17</td>
<td>LCD0-D04</td>
<td>PD4</td>
</tr>
<tr>
<td>CON2-P19</td>
<td>LCD0-D05</td>
<td>PD5</td>
</tr>
<tr>
<td>CON2-P21</td>
<td>LCD0-D06</td>
<td>PD6</td>
</tr>
<tr>
<td>CON2-P23</td>
<td>LCD0-D07</td>
<td>PD7</td>
</tr>
<tr>
<td>CON2-P25</td>
<td>LCD0-D08</td>
<td>PD8</td>
</tr>
<tr>
<td>CON2-P27</td>
<td>LCD0-D09</td>
<td>PD9</td>
</tr>
<tr>
<td>CON2-P29</td>
<td>LCD0-D10</td>
<td>PD10</td>
</tr>
<tr>
<td>CON2-P31</td>
<td>LCD0-D11</td>
<td>PD11</td>
</tr>
<tr>
<td>CON2-P33</td>
<td>LCD0-D12</td>
<td>PD12</td>
</tr>
<tr>
<td>CON2-P35</td>
<td>LCD0-D13</td>
<td>PD13</td>
</tr>
<tr>
<td>---------</td>
<td>---------</td>
<td>-------</td>
</tr>
<tr>
<td>CON2-P37</td>
<td>LCD0-D14</td>
<td>PD14</td>
</tr>
<tr>
<td>CON2-P39</td>
<td>LCD0-D15</td>
<td>PD15</td>
</tr>
<tr>
<td>CON2-P40</td>
<td>LCD0-D16</td>
<td>PD16</td>
</tr>
<tr>
<td>CON2-P38</td>
<td>LCD0-D17</td>
<td>PD17</td>
</tr>
<tr>
<td>CON2-P36</td>
<td>LCD0-D18</td>
<td>PD18</td>
</tr>
<tr>
<td>CON2-P34</td>
<td>LCD0-D19</td>
<td>PD19</td>
</tr>
<tr>
<td>CON2-P32</td>
<td>LCD0-D20</td>
<td>PD20</td>
</tr>
<tr>
<td>CON2-P30</td>
<td>LCD0-D21</td>
<td>PD21</td>
</tr>
<tr>
<td>CON2-P28</td>
<td>LCD0-D22</td>
<td>PD22</td>
</tr>
<tr>
<td>CON2-P26</td>
<td>LCD0-D23</td>
<td>PD23</td>
</tr>
<tr>
<td>CON2-P22</td>
<td>LCD0-CLK</td>
<td>PD24</td>
</tr>
<tr>
<td>CON2-P20</td>
<td>LCD0-CS</td>
<td>PH6</td>
</tr>
<tr>
<td>CON2-P18</td>
<td>LCD0-HSYNC</td>
<td>PD26</td>
</tr>
<tr>
<td>CON2-P16</td>
<td>LCD0-VSYNC</td>
<td>PD27</td>
</tr>
<tr>
<td>CON2-P14</td>
<td>LCD0-DE</td>
<td>PD25</td>
</tr>
<tr>
<td>CON2-P12</td>
<td>LCD0-IO2</td>
<td>PH9</td>
</tr>
<tr>
<td>CON2-P10</td>
<td>PWM0</td>
<td>PB2</td>
</tr>
<tr>
<td>CON2-P08</td>
<td>LCD0-IO1</td>
<td>PH8</td>
</tr>
<tr>
<td>CON2-P06</td>
<td>LCD0-IO0</td>
<td>PH7</td>
</tr>
<tr>
<td>CON2-P04</td>
<td>TWI3-SCK</td>
<td>PI0</td>
</tr>
<tr>
<td>CON2-P02</td>
<td>TWI3-SDA</td>
<td>PI1</td>
</tr>
<tr>
<td>CON2-P07</td>
<td>LCDIO-03</td>
<td>PH12</td>
</tr>
<tr>
<td>CON2-P01</td>
<td>IPSOUT</td>
<td></td>
</tr>
<tr>
<td>CON2-P03</td>
<td>IPSOUT</td>
<td></td>
</tr>
<tr>
<td>CON2-P05</td>
<td>GND</td>
<td></td>
</tr>
<tr>
<td>CON2-P24</td>
<td>GND</td>
<td></td>
</tr>
<tr>
<td>CON3-P18</td>
<td>IO-5(CAN_RX)</td>
<td>PH21</td>
</tr>
<tr>
<td>CON3-P16</td>
<td>IO-4(CAN_TX)</td>
<td>PH20</td>
</tr>
<tr>
<td>----------</td>
<td>--------------</td>
<td>------</td>
</tr>
<tr>
<td>CON3-P23</td>
<td>SPI0_CLK</td>
<td>P111</td>
</tr>
<tr>
<td>CON3-P21</td>
<td>SPI0_MISO</td>
<td>P113</td>
</tr>
<tr>
<td>CON3-P19</td>
<td>SPI0_MOSI</td>
<td>P112</td>
</tr>
<tr>
<td>CON3-P24</td>
<td>SPI0_CS0</td>
<td>P110</td>
</tr>
<tr>
<td>CON3-P26</td>
<td>SPI0_CS1</td>
<td>P114</td>
</tr>
<tr>
<td>CON3-P05</td>
<td>TWI2-SCK</td>
<td>PB20</td>
</tr>
<tr>
<td>CON3-P03</td>
<td>TWI2-SDA</td>
<td>PB21</td>
</tr>
<tr>
<td>CON3-P15</td>
<td>IO-3(UART2_CTS)</td>
<td>P117</td>
</tr>
<tr>
<td>CON3-P22</td>
<td>IO-6(UART2_RTS)</td>
<td>P116</td>
</tr>
<tr>
<td>CON3-P11</td>
<td>IO-0(UART2_RX)</td>
<td>P119</td>
</tr>
<tr>
<td>CON3-P13</td>
<td>IO-2(UART2_TX)</td>
<td>P118</td>
</tr>
<tr>
<td>CON3-P10</td>
<td>UART3_RX</td>
<td>PH1</td>
</tr>
<tr>
<td>CON3-P08</td>
<td>UART3_TX</td>
<td>PH0</td>
</tr>
<tr>
<td>CON3-P12</td>
<td>IO-1</td>
<td>PH2</td>
</tr>
<tr>
<td>CON3-P07</td>
<td>GPCLK</td>
<td>P13</td>
</tr>
<tr>
<td>CON3-P01</td>
<td>VCC-3V3</td>
<td></td>
</tr>
<tr>
<td>CON3-P17</td>
<td>VCC-3V3</td>
<td></td>
</tr>
<tr>
<td>CON3-P02</td>
<td>VCC-5V</td>
<td></td>
</tr>
<tr>
<td>CON3-P04</td>
<td>VCC-5V</td>
<td></td>
</tr>
<tr>
<td>CON3-P09</td>
<td>GND</td>
<td></td>
</tr>
<tr>
<td>CON3-P25</td>
<td>GND</td>
<td></td>
</tr>
<tr>
<td>CON3-P06</td>
<td>GND</td>
<td></td>
</tr>
<tr>
<td>CON3-P14</td>
<td>GND</td>
<td></td>
</tr>
<tr>
<td>CON3-P20</td>
<td>GND</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>J12-P03</th>
<th>IO-7</th>
<th>PH5</th>
</tr>
</thead>
<tbody>
<tr>
<td>J12-P05</td>
<td>IO-8</td>
<td>PH3</td>
</tr>
<tr>
<td>J12-P04</td>
<td>UART7_RX</td>
<td>PI21</td>
</tr>
<tr>
<td>--------</td>
<td>----------</td>
<td>------</td>
</tr>
<tr>
<td>J12-P06</td>
<td>UART7_TX</td>
<td>PI20</td>
</tr>
<tr>
<td>J12-P01</td>
<td>VCC-5V</td>
<td></td>
</tr>
<tr>
<td>J12-P02</td>
<td>VCC-3V3</td>
<td></td>
</tr>
<tr>
<td>J12-P07</td>
<td>GND</td>
<td></td>
</tr>
<tr>
<td>J12-P08</td>
<td>GND</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>J11-P01</th>
<th>UART0-TX</th>
<th>PB22</th>
</tr>
</thead>
<tbody>
<tr>
<td>J11-P02</td>
<td>UART0-RX</td>
<td>PB23</td>
</tr>
</tbody>
</table>