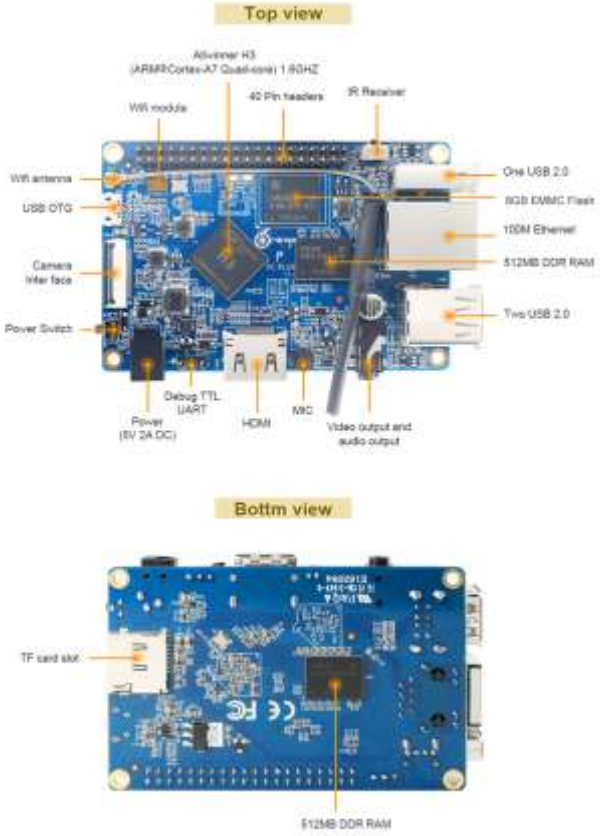
Agenda

1. Hardware description Orange Pi One and PC
2. How Install Linux distribution to Orange PI - **Armbian**
3. OrangePi Build System



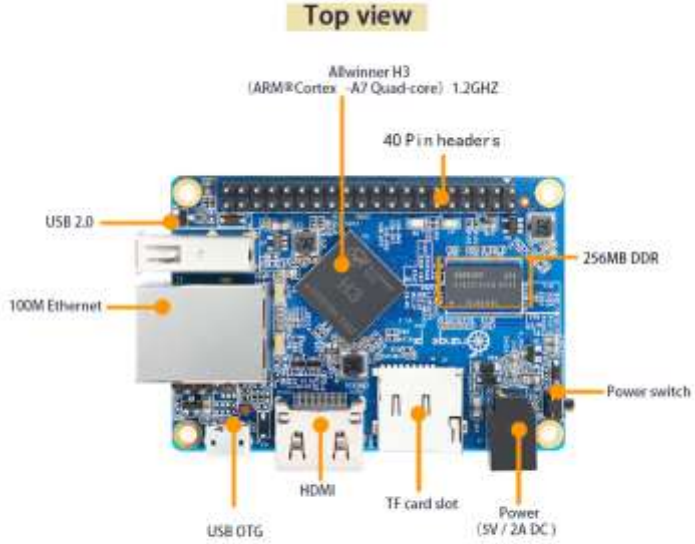
Hardware description
Orange Pi PC

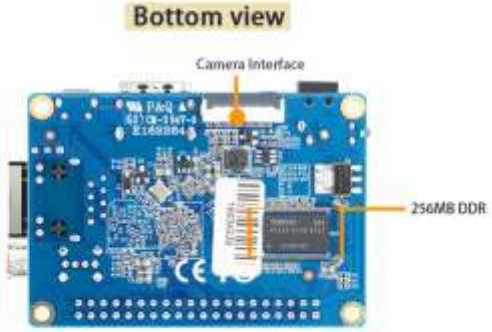
CPU: H3 Quad-core Cortex-A7 H.265/HEVC 4K

Memory (SDRAM): 1GB DDR3 (shared with GPU)

Low-level peripherals: 40 Pins Header,compatible with Raspberry Pi B+

GPIO(1x3) pin:UART, ground.





Hardware description
Orange Pi one

CPU: H3 Quad-core Cortex-A7 H.265/HEVC 4K

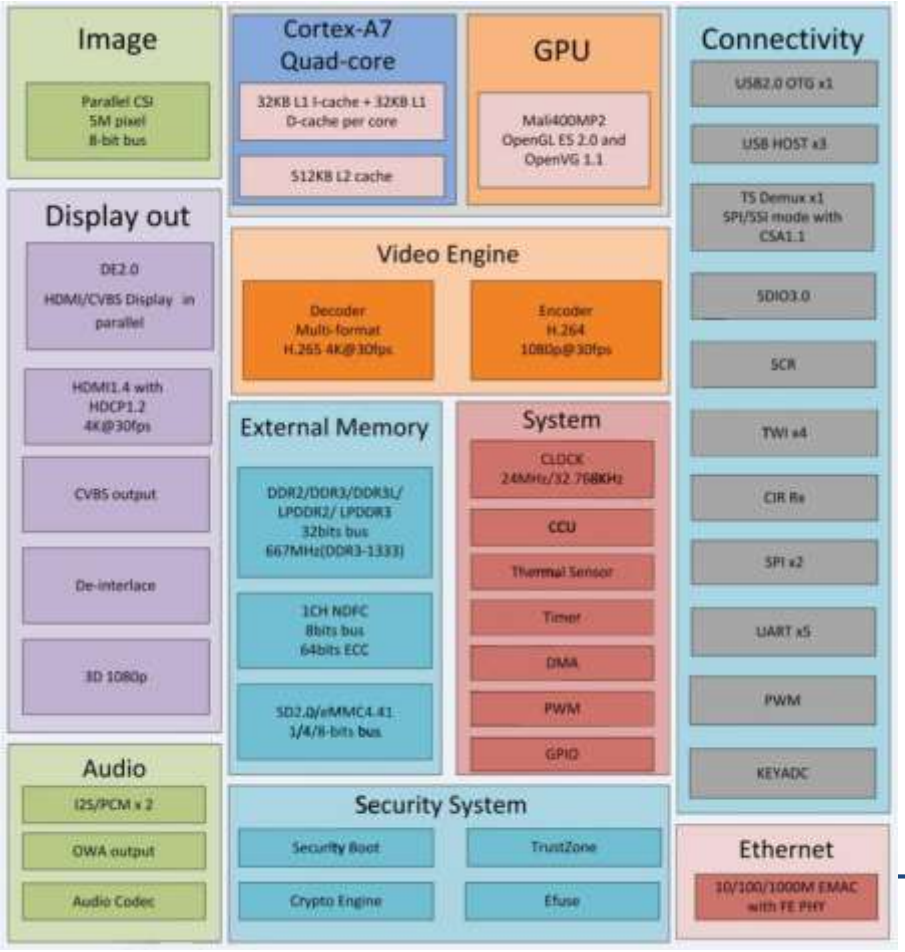
Memory (SDRAM) : 512MB DDR3 (shared with GPU)

Low-level peripherals: 40 Pins Header,compatible with Raspberry Pi B+

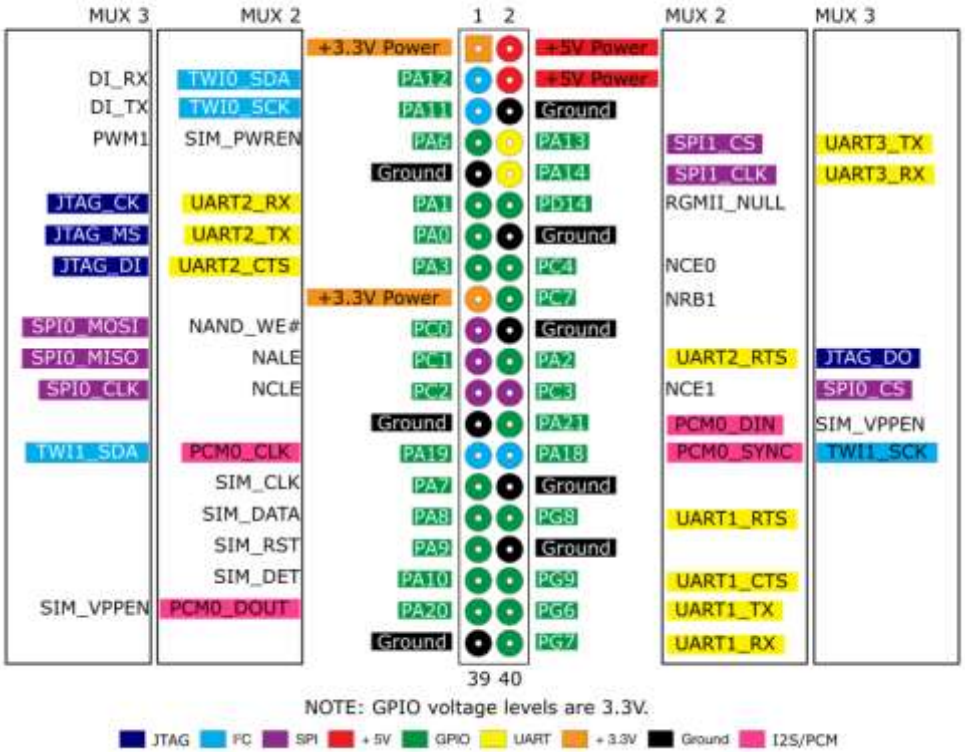
GPIO(1x3) pin: UART, ground.

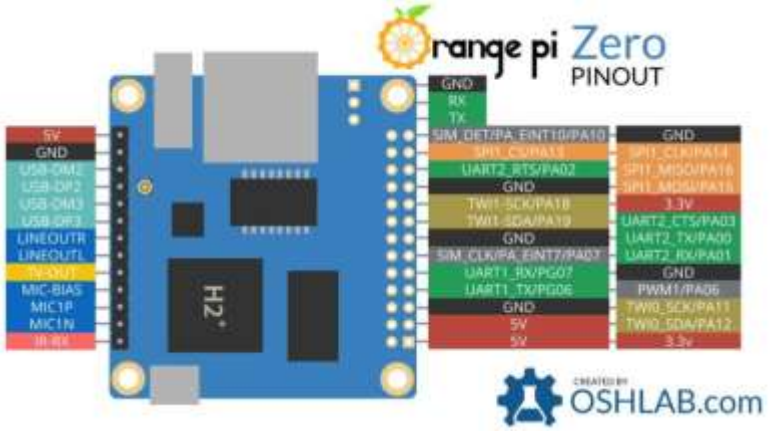
Supported OS: Android Ubuntu, Debian, Raspbian Image

H3 Quad-core Cortex-A7

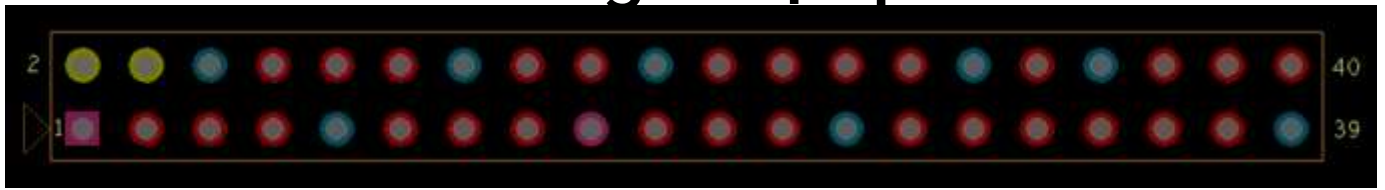


Orange Pi GPIO pinout





Orange Pi 40-pin connector



	*3.3V*
	***VCC-5V !!!!!***
	*GND*
	**Port**

Orange Pi UART debug output

1	*GND*
2	RX
3	TX

Linux distribution for Orange PI one

1. **Armbian Bionic**

   https://dl.armbian.com/orangepione/Ubuntu_bionic_next.7z

2. **Armbian Stretch**

   https://dl.armbian.com/orangepione/Debian_stretch_next.7z

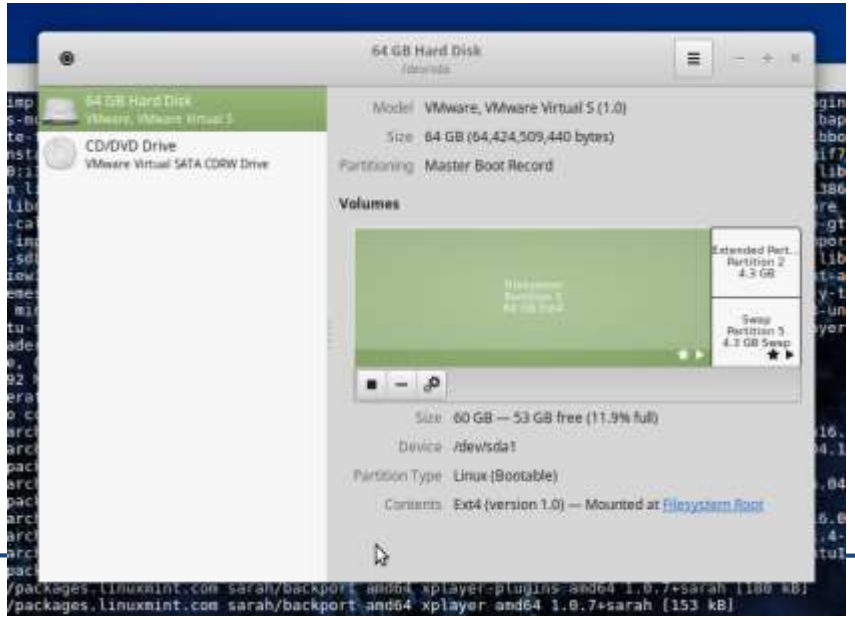
3. **See https://www.armbian.com/download/**

Flash image to target platform

- **Flash it to your SD-card:**
- sudo dd if=image_name.img of=/dev/sdX bs=1M oflag=direct

Username: root
Password: 1234

- disks



Needs for operation with Orange Pi

- Minicom utility
- SD-card reader
- Orange Pi One
- SD-card 8Gb
- Power supply 1A or more

Start Debug and Kernel Debugging with USB-UART converter

1. Connect your Orange Pi to USB-UART converter

2. Make sure the **5V wire is disconnected!!!!!!**

3. Connect your USB-UART converter to PC

4. Start Minicom utility

5. Power on your Orange Pi

Start OS and Debug

1. Connect your Orange Pi to USB-UART converter

2. Check Minicom is a text-based modem control and terminal emulation program for Unix-like operating systems,

3. Check USB

ls /dev/ttyUSB*

Answer like */dev/ttyUSB0 /dev/ttyUSB1 /dev/ttyUSB2 /dev/ttyUSB3*

4.Connect to /dev/ttyUSB2

sudo minicom /dev/ttyUSB2

5.Please check speed

Need 115200

Minicom utility

Setup

- sudo apt-get install minicom

Run

- minicom or sudo minicom

Device list

- dmesg | grep tty
- ls -l /dev/tty*

Minicom running a Windows 2003 EMS prompt

EMS- **Exchange Management Shell**

Minicom utility

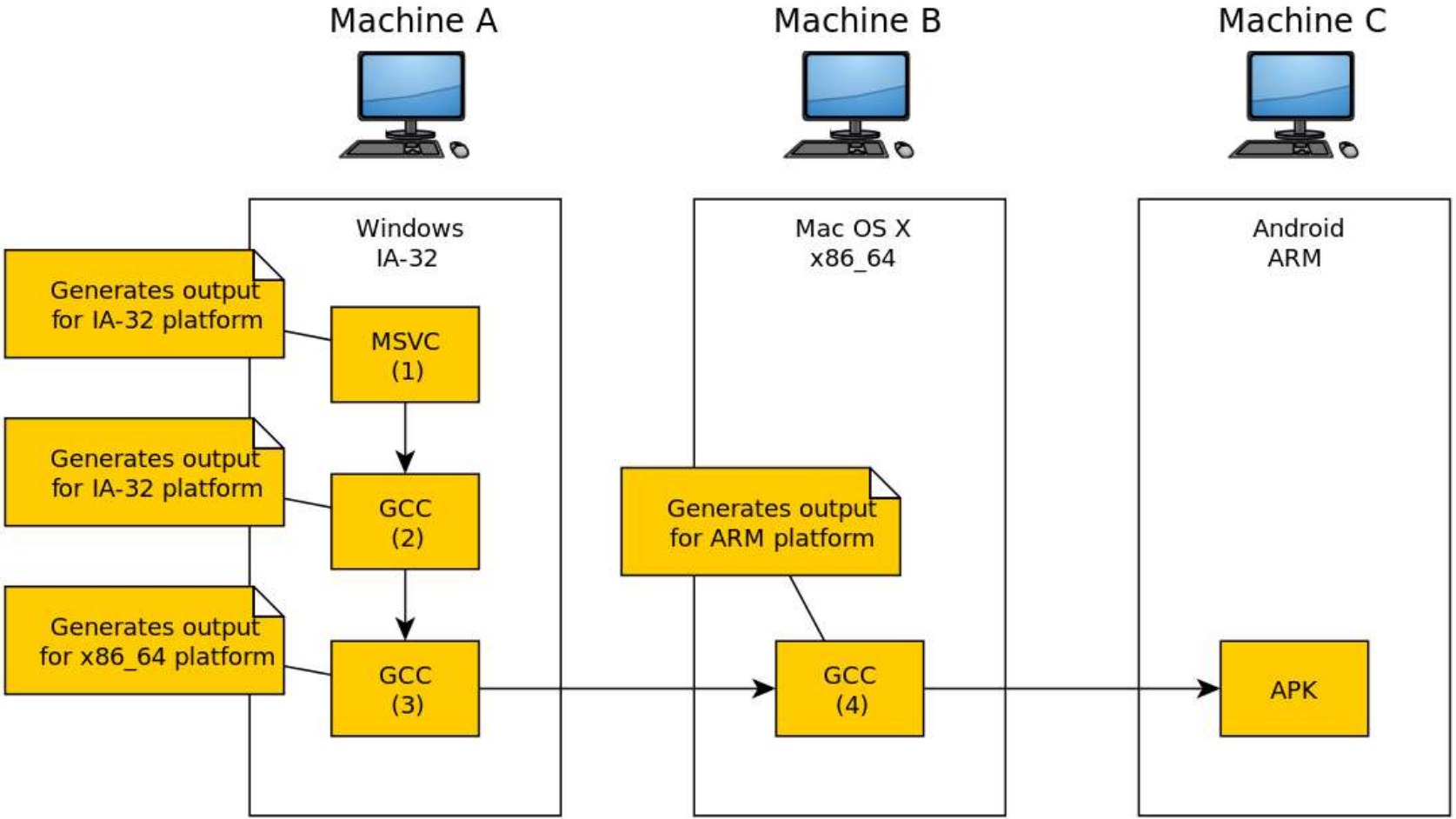
Several key combinations:

- Help: CTRL-A Z
- Configuration: CTRL-A O
- Initializing the modem: CTRL-A M
- Output: CTRL-A Q

Cross-compiling for ARM

- Cross-compiling for ARM requires a toolchain and a platform emulator or a real target platform.
- * -none-eabi is a toolchain for compiling a project working in bare metal.
- * **eabi** is a toolchain for compiling a project running in any OS. In my case, this is Linux.
- * **eabihf** is almost the same as eabi, with a difference in the ABI implementation of floating-point function calls. hf - stands for hard float.

Для кросс-компиляции с GCC необходимо, чтобы была доступна скомпилированная для целевой платформы версия binutils. Особенно важно наличие GNU Assembler. Поэтому binutils должны быть предварительно скомпилированы с ключом --target=some-target, указанным скрипту конфигурирования


Machine A
Machine B
Machine C
Windows
IA-32
Mac OS X
x86_64
Android
ARM
Generates output
for IA-32 platform
MSVC
(1)
Generates output
for IA-32 platform
GCC
(2)
Generates output
for x86_64 platform
GCC
(3)
Generates output
for ARM platform
GCC
(4)
APK

Cross-compiling for ARM

- GCC также должна быть указана опция --target с аналогичным содержанием. После этого, чтобы GCC могла использовать полученные binutils, надо поместить путь к ним в переменную окружения path, например:
- PATH=/path/to/binutils/bin:${PATH} make

Cross-compiling for ARM

```
ifeq ($(findstring x86,$(BUILDMACH)),x86)
#CROSS_COMPILE?=arm-bcm2708hardfp-linux-gnueabi-
    ifeq ($(MAKECMDGOALS), rpi)
    CROSS_COMPILE?=arm-linux-gnueabi-
    else
    CROSS_COMPILE?=arm-linux-gnueabihf-
    endif
else
CROSS_COMPILE?=
endif
CC=$(CROSS_COMPILE)gcc
```

MANTRA LABS

OrangePiH3 toolchain

- https://github.com/OrangePiLibra/OrangePiH3_toolchain

- **cross_comp=arm-linux-gnueabi**
- `apt-get install gcc-4.7-arm-linux-gnueabi`
- `apt-get install gcc-arm-linux-gnueabi`

```
make CROSS_COMPILE=/usr/bin/arm-none-linux-gnueabihf- ARCH=arm all
```

Cross-compiling for ARM

- #guest architecture

ARCH := arm

- **toolchain**

CROSS_COMPILE := arm-linux-gnueabi-
obj-m := $(MODULES)

- **make**

MAKEARCH := $(MAKE) ARCH=$(ARCH) CROSS_COMPILE=$(CROSS_COMPILE)

make ARCH=arm CROSS_COMPILE=arm-linux-gnueabi-

How to compile Linux kernel for Orange Pi

1. Download Toolchain
2. Download kernel source code
3. Download you driver source code
4. Configuration of kernel
5. Compilation of kernel
6. Compilation of you driver
7. Installation
8. Device Tree preparation
9. U-Boot (optional)
10. Make SD card bootable

Get start

Build and replace kernel !

apt-get install gcc-5-arm-linux-gnueabi
apt-get install gcc-arm-linux-gnueabi