

1. Setting up a bootable device.

For device to boot there is a need for microSD Flash drive.

Another boot option is to use onboard SPI Flash, available in a variant with 16MB SPI flash memory (suffix **-s16M**) and related u-boot with TFTP kernel load for network. [Howto prepare SPI Flash boot.](#)

Standard microSD Flash has FAT32 file system. It can be accessed from both MS Windows and GNU Linux operating systems. Thus it is not formatted to any linux-specific file system because there are no extra features required to system to work. Files are easily copied to / from the disk.

Getting ARM-based operating system distribution to update binary files of used programs. Or to perform needed setup to create minimal bootable system for CU.

- Allwinner A20/T2 OLInuXino-LIME2 [download image](#) from [Debian distro](#). There is an Ubuntu option also. [Purchase chip&chip, olimex](#).

• ----

Last Armbian used: Armbian_20.02.1_Lime2_bionic_current_5.4.20

ARM-base minimal OS creation with ~ 8M bytes size and microSD Flash with FAT32 filesystem support.

1.1. Need to create **initramfs** inside running ARM GNU/Linux distro **adani-cu-tools/fs-host-armbian/mkinitramfs.sh**. Downloaded image must be installed into Flash, boot, work inside Armbian system. Create file e.g. **adani-cu-tools/fs-boot-disk-SUN7i-olinuxino-lime2/boot/initramfs-***. To run such distro **qemu** can be used because all generated files are ARM-device independent (TODO). If init and other scripts will use only few shell tools and dynamic libraries addition or optimization is not needed then filesystem can be also any (TODO).

1.2. Then all files can be updated on any usual Linux or other system that supports filesystem with hardlinks/symlinks and 'cpio' command. Update examples: binaries like busybox, kernel modules (if used, and newer linux kernel was added). To completely

See:

adani-cu-tools/fs-host-armbian/armbian+usr_share_initramfs-tools/ source and
adani-cu-tools/fs-host-any-gnu-linux/initramfs/ product.

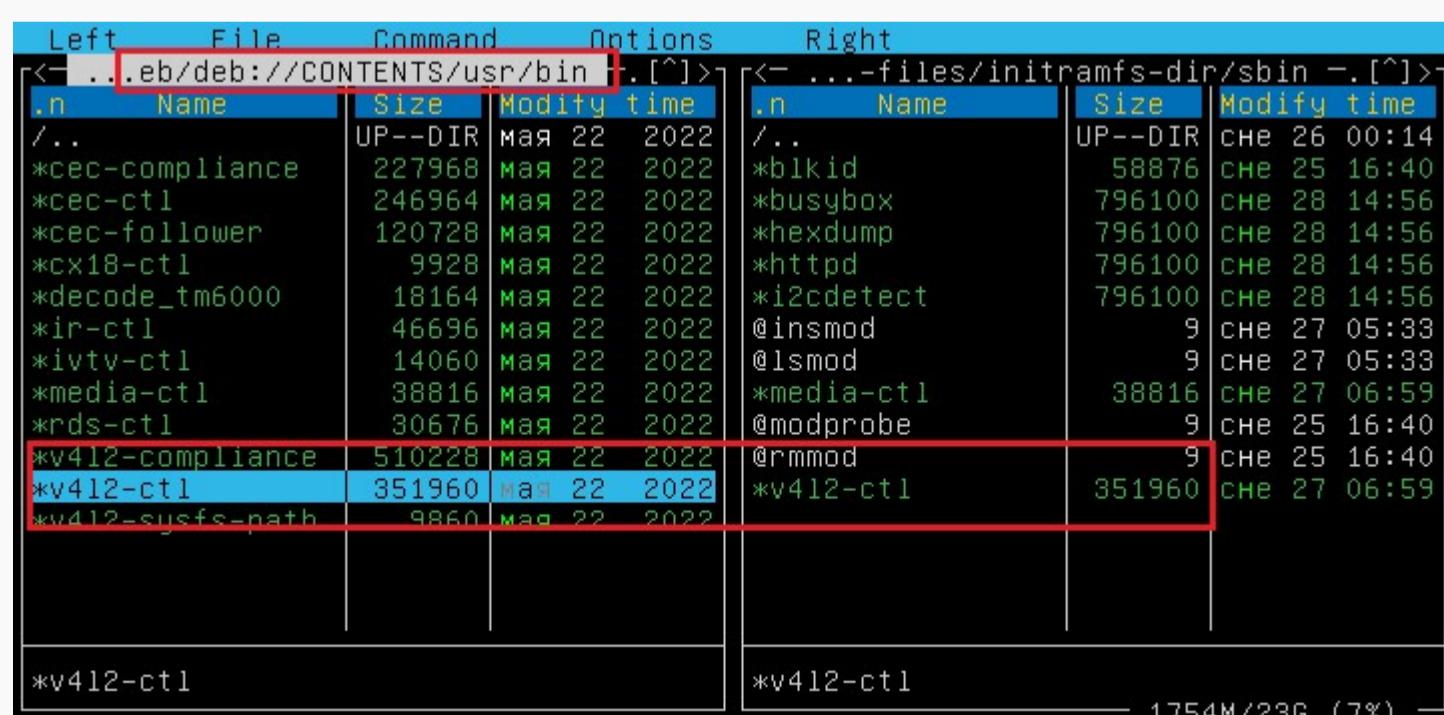
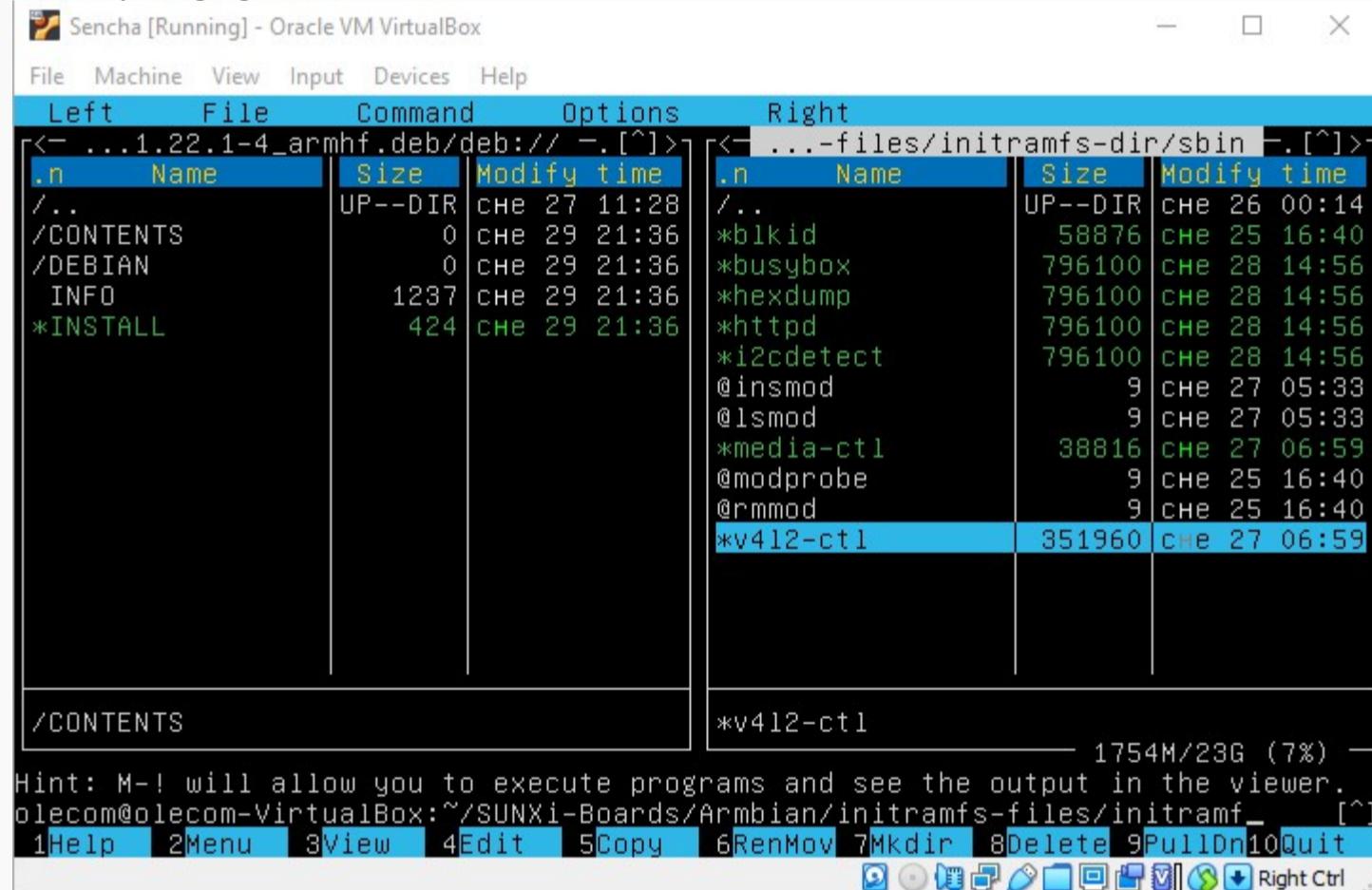
1.3. Manual update of existing initramfs image is possible inside Linux-compatible system where file system supports symbolic / hardlinks used.

1.3.1 unarchive file from **adani-cu-tools/fs-host-any-gnu-linux/initramfs/initramfs-dir.tar.gz**

1.3.2 download and copy files from distribution files, e.g.

<https://packages.ubuntu.com/jammy/v4l-utils> (with all depended libraries, e.g. <https://packages.ubuntu.com/jammy/libgcc-s1>)
http://ports.ubuntu.com/pool/universe/v/v4l-utils/v4l-utils_1.22.1-4_armhf.deb
http://ports.ubuntu.com/pool/main/v/v4l-utils/libv4l0_1.22.1-4_armhf.deb

into **initramfs-dir** directory using e.g. **mc** command shell:



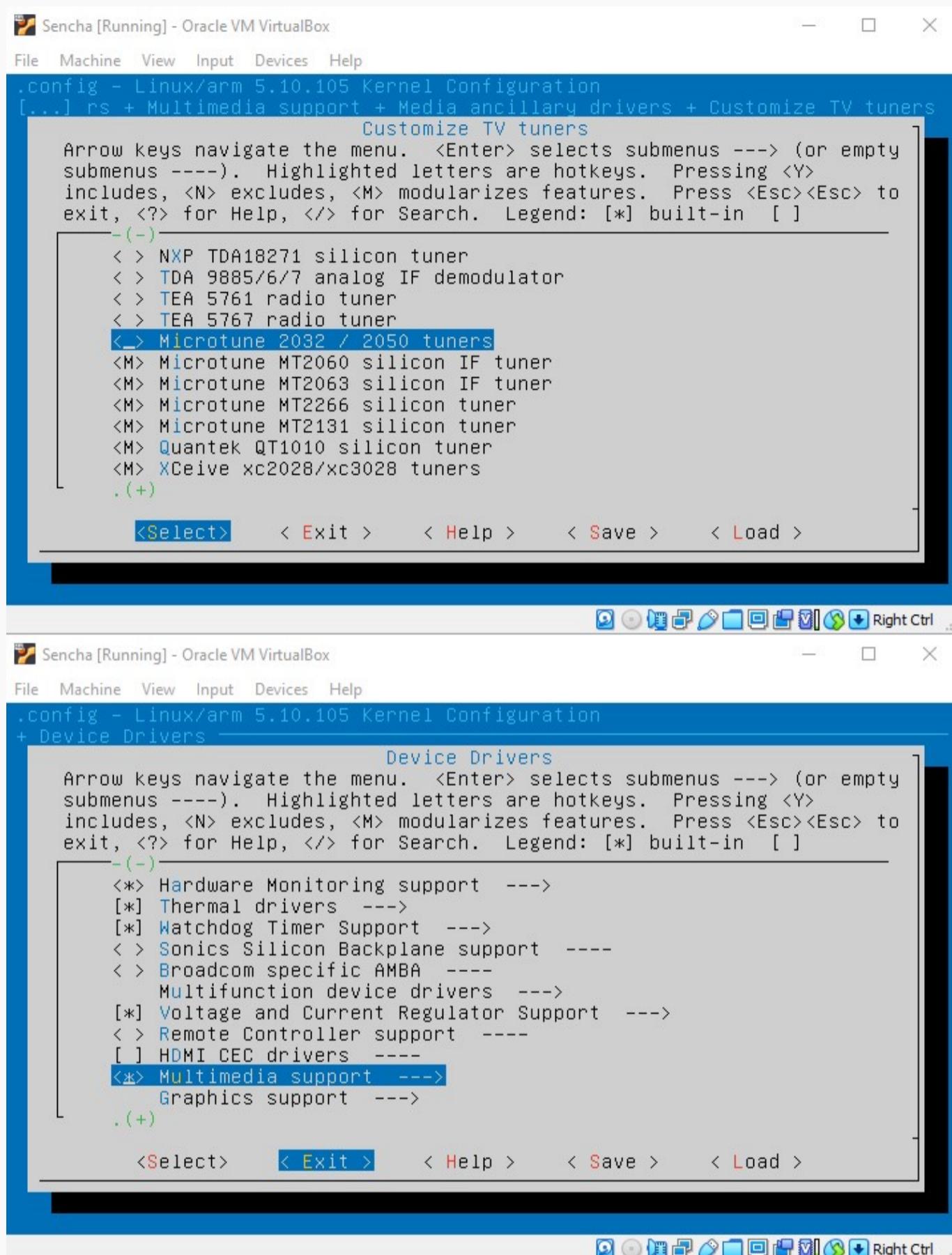
Then generate image using **adani-cu-tools/fs-host-any-gnu-linux/initramfs/mkinitrd.sh** which can be copied into /boot of microFlash card.

2. Compiling Linux kernel and Busybox.

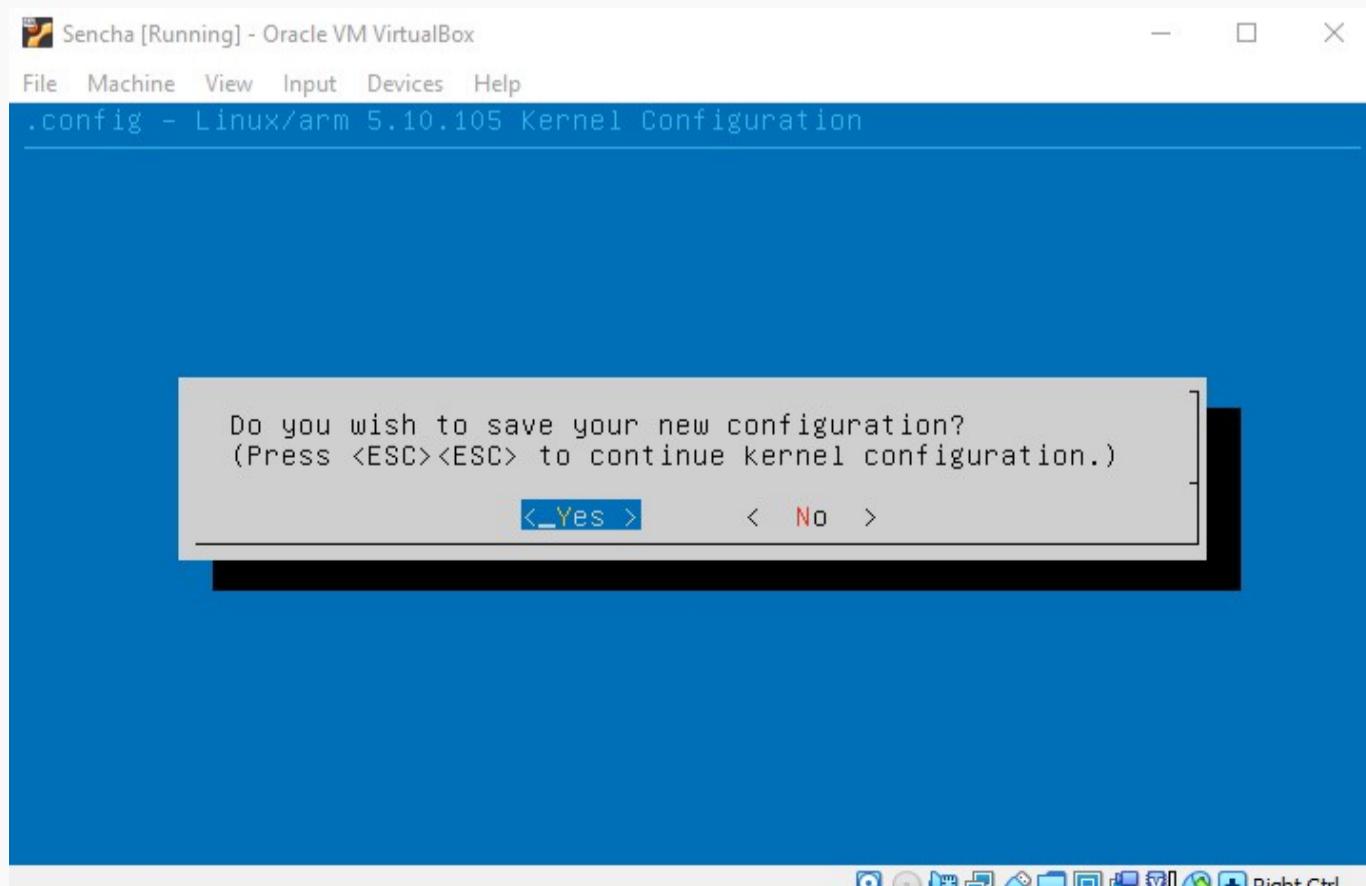
2.1 Get the source: **adani-cu-tools/fs-host-any-gnu-linux/linux-olimex/get-linux-olimex.sh**

2.2 Configure / build script to produce dtb, kernel, modules: **adani-cu-tools/fs-host-any-gnu-linux/linux-olimex/build_linux-olimex.sh**

2.3 Configure. New versions of Linux kernel may have additional features and drivers which are enabled for build. We don't need them. Thus after copying previous **adani-cu-tools/fs-host-any-gnu-linux/linux-olimex/.config_v3_from_v2** config file as new **.config**, all not need item should be removed:

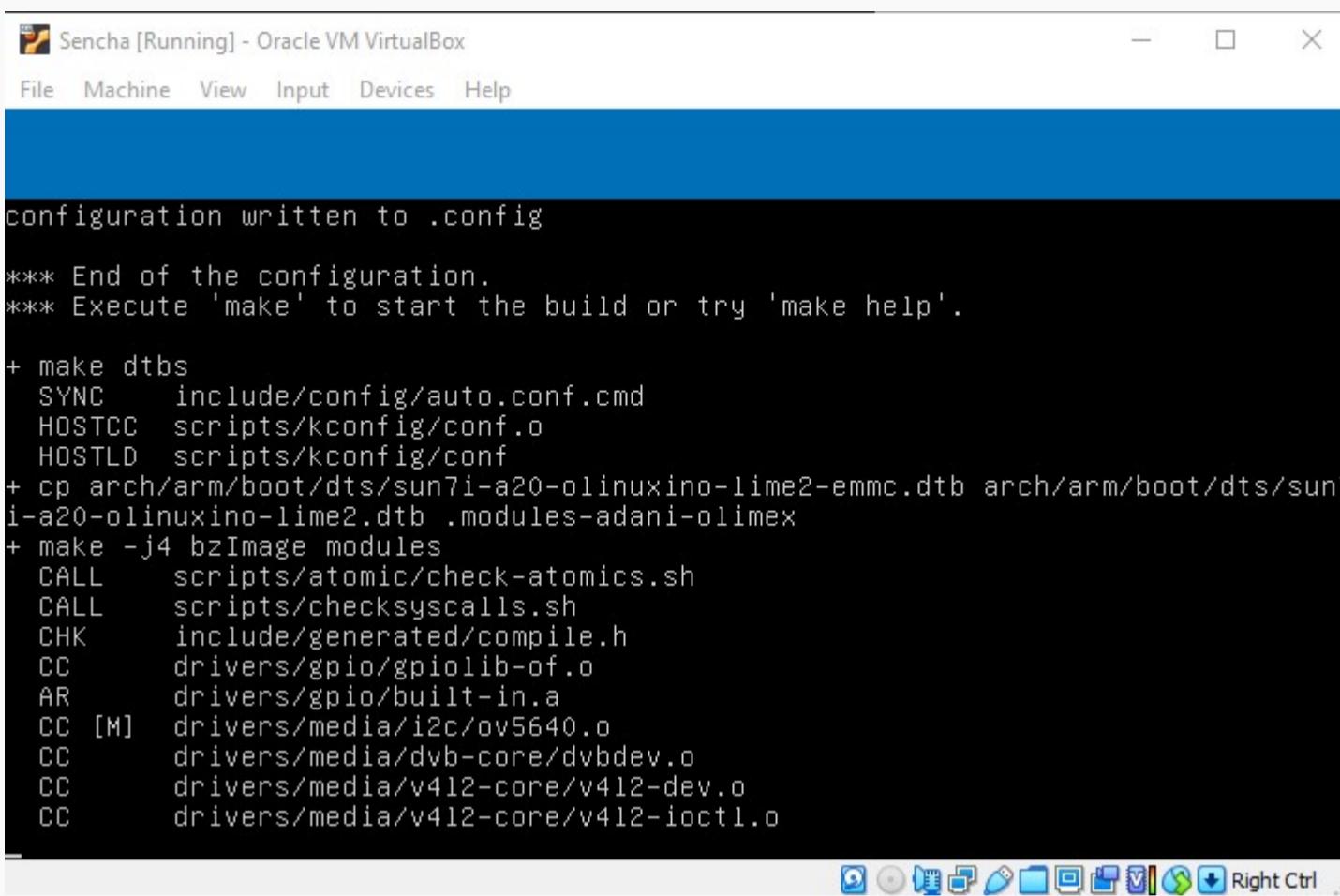


New config file is saved:



2.4 Build happens on usual Linux PC machine (or virtual). Cross compilation into ARM binaries takes place.

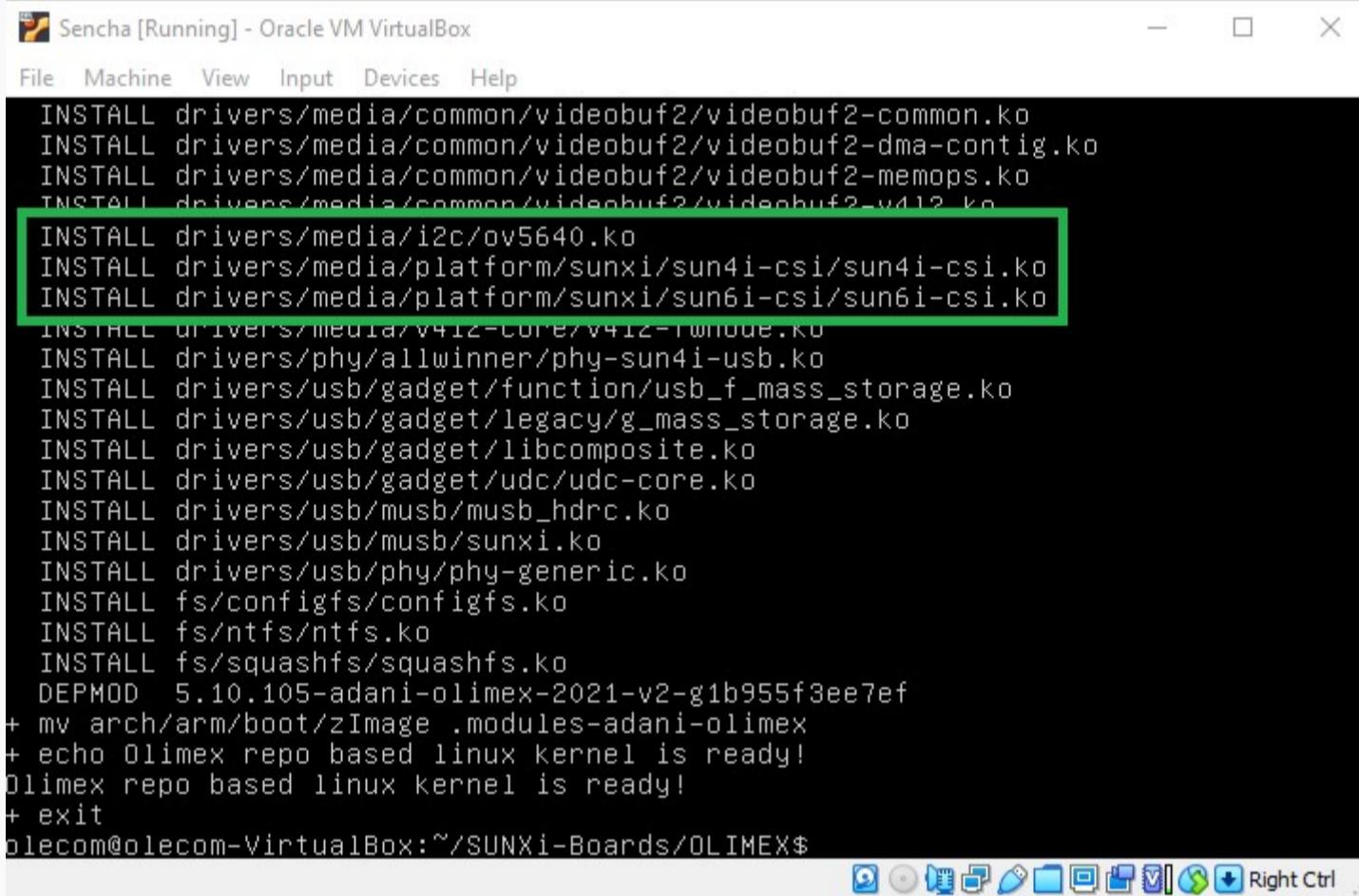
Start:



```
configuration written to .config
*** End of the configuration.
*** Execute 'make' to start the build or try 'make help'.

+ make dtbs
  SYNC  include/config/auto.conf.cmd
  HOSTCC scripts/kconfig/conf.o
  HOSTLD scripts/kconfig/conf
+ cp arch/arm/boot/dts/sun7i-a20-olinuxino-lime2-emmc.dtb arch/arm/boot/dts/sun7i-a20-olinuxino-lime2.dtb .modules-adani-olimex
+ make -j4 bzImage modules
  CALL  scripts/atomic/check-atomics.sh
  CALL  scripts/checksyscalls.sh
  CHK   include/generated/compile.h
  CC    drivers/gpio/gpiolib-of.o
  AR    drivers/gpio/built-in.a
  CC [M] drivers/media/i2c/ov5640.o
  CC    drivers/media/dvb-core/dvbdev.o
  CC    drivers/media/v412-core/v412-dev.o
  CC    drivers/media/v412-core/v412-iocctl.o
```

Finish. Needed driver files are highlighted.



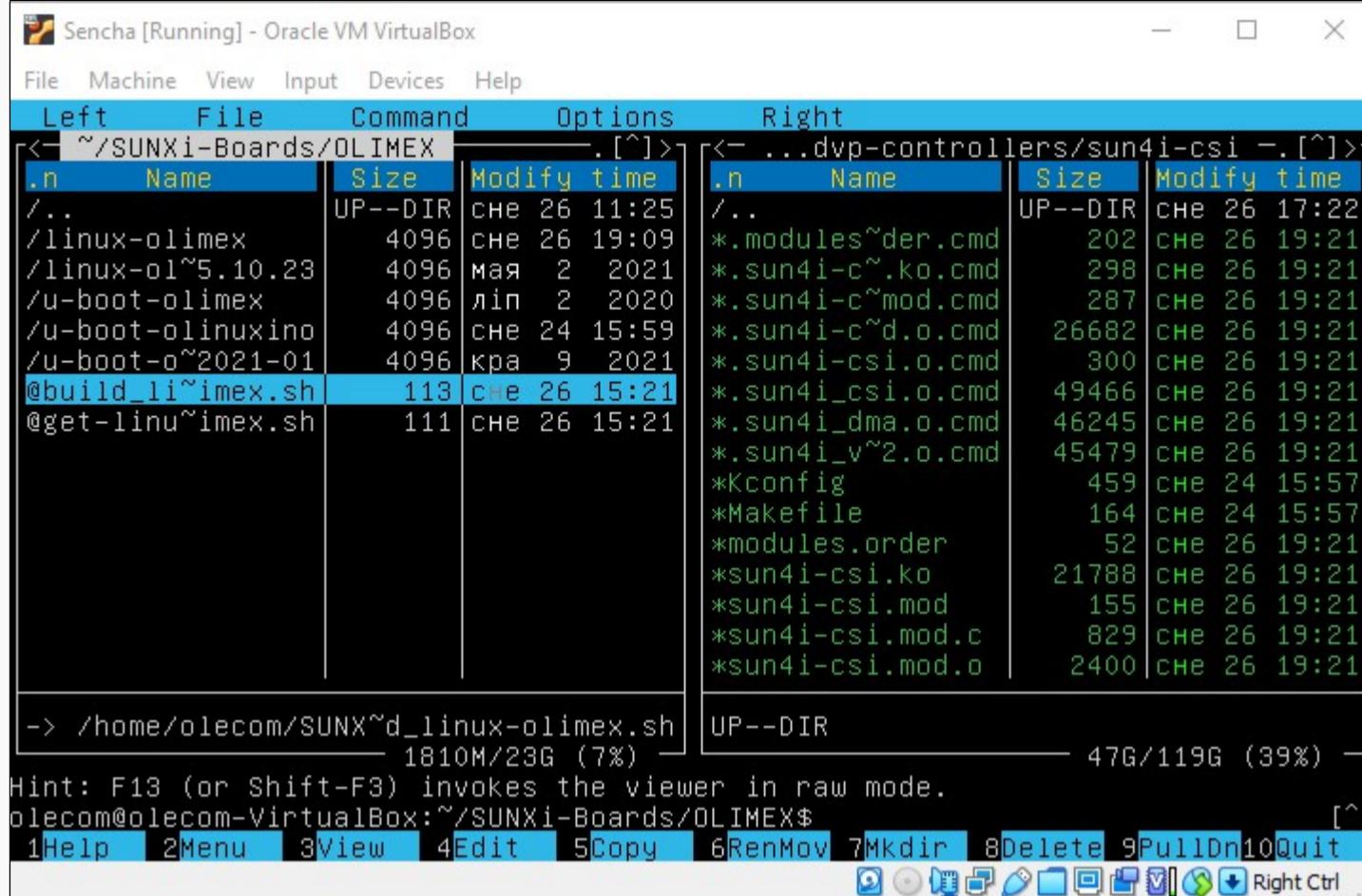
```
INSTALL drivers/media/common/videobuf2/videobuf2-common.ko
INSTALL drivers/media/common/videobuf2/videobuf2-dma-contig.ko
INSTALL drivers/media/common/videobuf2/videobuf2-memops.ko
INSTALL drivers/media/common/videobuf2/videobuf2-v412.ko
INSTALL drivers/media/i2c/ov5640.ko
INSTALL drivers/media/platform/sunxi/sun4i-csi/sun4i-csi.ko
INSTALL drivers/media/platform/sunxi/sun6i-csi/sun6i-csi.ko
INSTALL drivers/media/v412-core/v412-iocnue.ko
INSTALL drivers/phy/allwinner/phy-sun4i-usb.ko
INSTALL drivers/usb/gadget/function/usb_f_mass_storage.ko
INSTALL drivers/usb/gadget/legacy/g_mass_storage.ko
INSTALL drivers/usb/gadget/libcomposite.ko
INSTALL drivers/usb/gadget/udc/udc-core.ko
INSTALL drivers/usb/musb/musb_hdrc.ko
INSTALL drivers/usb/musb/sunxi.ko
INSTALL drivers/usb/phy/phy-generic.ko
INSTALL fs/configfs/configfs.ko
INSTALL fs/ntfs/ntfs.ko
INSTALL fs/squashfs/squashfs.ko
DEPMOD 5.10.105-adani-olimex-2021-v2-g1b955f3ee7ef
+ mv arch/arm/boot/zImage .modules-adani-olimex
+ echo Olimex repo based linux kernel is ready!
Olimex repo based linux kernel is ready!
+ exit
olecom@olecom-VirtualBox:~/SUNXi-Boards/OLIMEX$
```

2.5. Similar build is done for Busybox: <https://github.com/olecom/adani-cu-tools/tree/master/fs-host-any-gnu-linux/busybox>

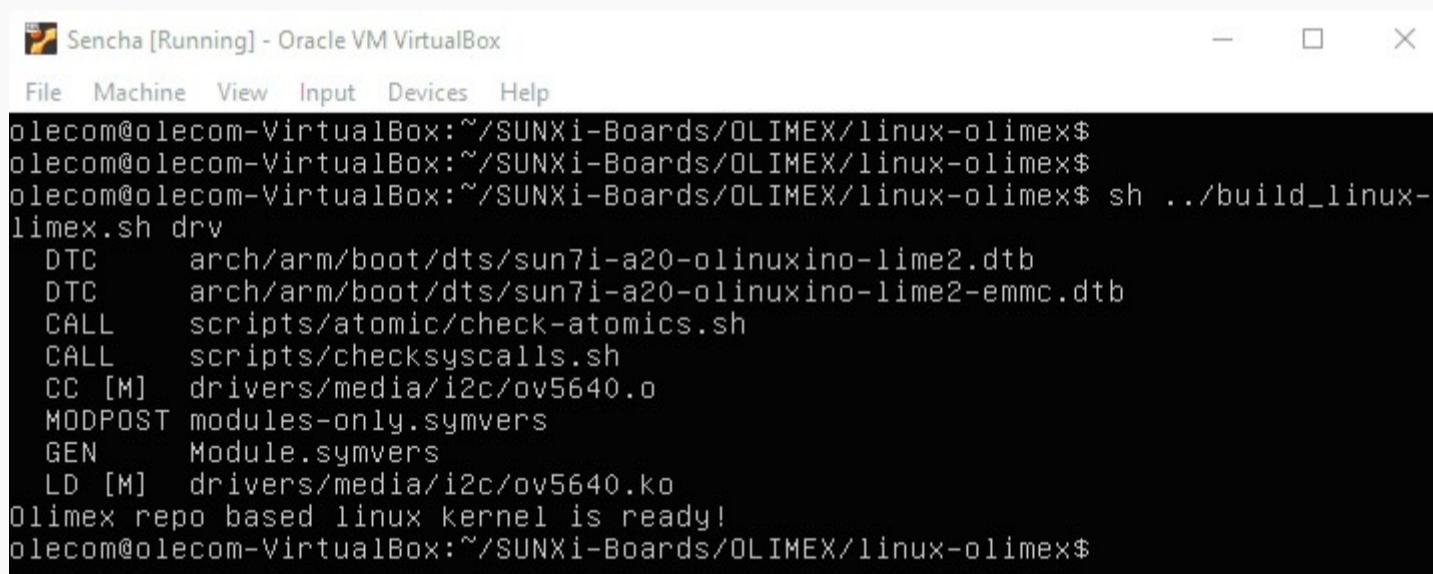
3. Linux kernel drivers development.

Using **adani-cu-tools/fs-host-any-gnu-linux/linux-olimex/build_linux-olimex.sh** source files are connected from git on MS Windows (or any other git repository) using symbolic links into linux source tree and then compiled. [Building External Modules](#) is not used, since there is a full linux source available.

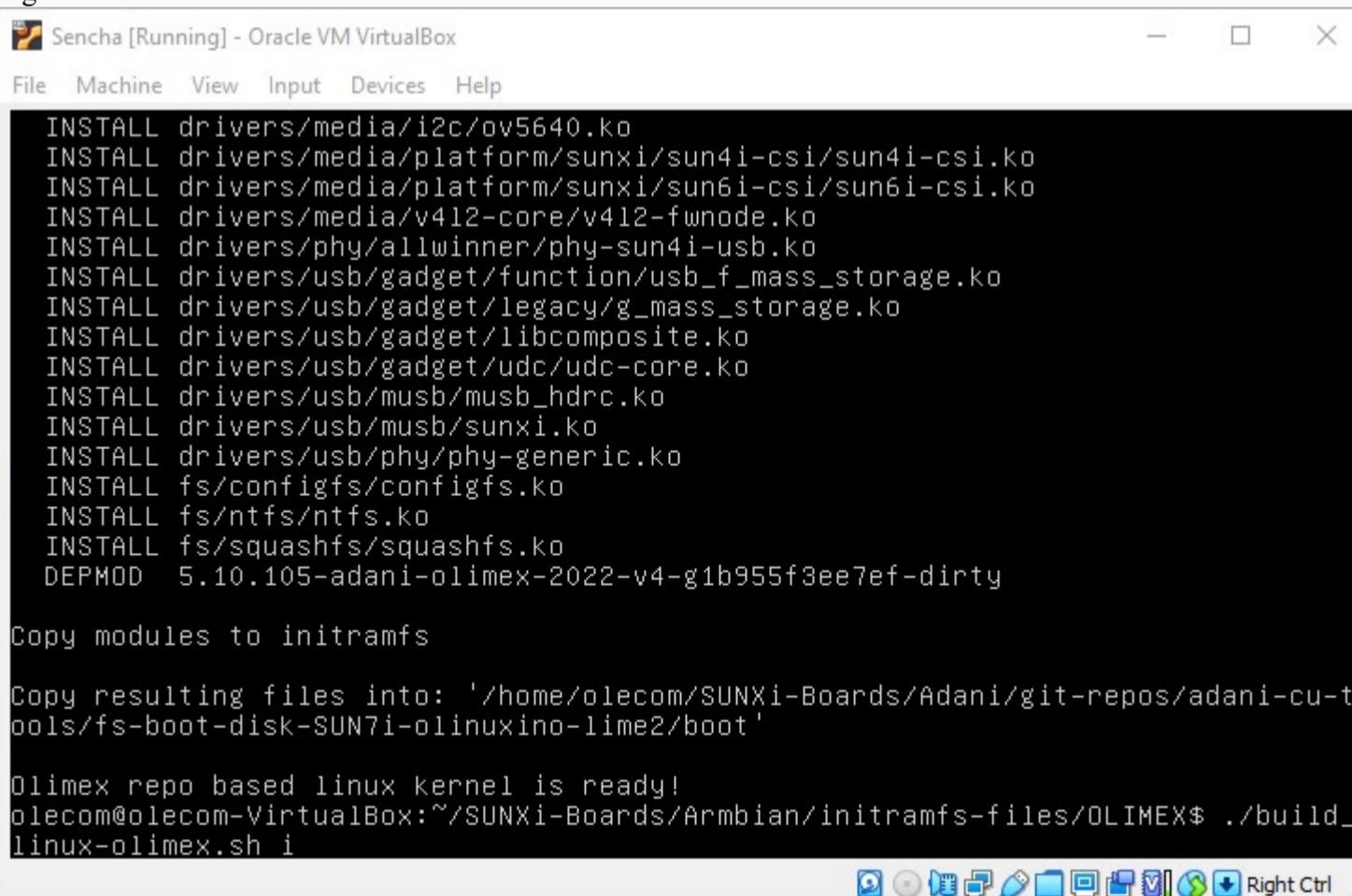
Working directory on the left, git repository **adani-linux-drivers** of CSI drivers on the right:



Compilation process:



Modules installation / copying:



Upload new driver to CU, execute some commands from **adani-cu-arm/exe.sh** with it:

MINGW64:/e/_Projects/Adani/git-repos/adani-cu-arm

```
legion@LAPTOP-SP2SBN7B MINGW64 /e/_Projects/Adani/git-repos/adani-cu-arm (testing-binning2x2_v22)
$ curl -F drivers[]=@./adani-cu-tools/fs-boot-disk-SUN7i-olinuxino-lime2/boot/ov5640.ko 'http://10.0.0.22:2019/updateDriversRAM'
% Total    % Received % Xferd  Average Speed   Time   Time   Time  Current
                                         Dload  Upload Total Spent   Left  Speed
100 23931     0    205  100 23726    7268   821k --:--:-- --:--:--:--:--:--  865k{"err": "", "fields": {}, "files": {"drivers[]": {"size": 23520, "path": "/dev/shm/upload_5d3a4c80afb49512191915257be99483", "name": "ov5640.ko"}, "type": "application/octet-stream", "mtime": "1970-01-01T06:12:29.990Z"}}}

legion@LAPTOP-SP2SBN7B MINGW64 /e/_Projects/Adani/git-repos/adani-cu-arm (testing-binning2x2_v22)
$ curl -F 'drivers[]=@exe.sh' 'http://10.0.0.22/cgi-bin/index.cgi?execute'
% Total    % Received % Xferd  Average Speed   Time   Time   Time  Current
                                         Dload  Upload Total Spent   Left  Speed
100 2604     0   1271  100 1333    2432   2551 --:--:-- --:--:--:--:--:--  4998<html><head><title>CGI
v1</title></head><body>
File /dev/shm/exe.sh has been accepted.<br>
<hr>
<pre>media
Unable to setup formats: Invalid argument (22)
Failed to open /dev/video0: Invalid argument
=====
[ 4668.991365] ov5640 2-003c: remove
[ 4668.992411] i2c-core: driver [ov5640] unregistered
[ 4669.045317] ov5640 2-003c: probe
[ 4669.045718] ov5640 2-003c: supply DVDD not found, using dummy regulator
[ 4669.045911] ov5640 2-003c: supply AVDD not found, using dummy regulator
[ 4669.045993] ov5640 2-003c: supply DVDD not found, using dummy regulator
[ 4669.048373] ov5640 2-003c: ?ov5640_probe: OK
[ 4669.048654] i2c-core: driver [ov5640] registered
[ 4669.059302] sun4i-csi 1c09000.csi: Device registered as video0
[ 4669.059884] sun4i-csi 1c09000.csi: Unbalanced pm_runtime_enable!
Linux 192.168.85.156 5.10.105-adani-olimex-2022-v4-g1b955f3ee7ef-dirty #7 SMP Tue Dec 27 16:53:52 +0
3 2022 armv7l GNU/Linux
Module                  Size  Used by
sun4i_csi              16384  0
ov5640                  20480  1
videobuf2_dma_contig    12288  1 sun4i_csi
videobuf2_memops        12288  1 videobuf2_dma_contig
videobuf2_v412           20480  1 sun4i_csi
```

Video 4 linux commands:

```
sudo apt install v4l-utils libmp3lame-dev libpulse-dev libv4l-dev
$ v4l2-ctl -d /dev/video0 -D
Driver Info (not using libv4l2):
    Driver name      : sun6i-video
    Card type        : sun6i-csi
    Bus info         : platform:camera
    Driver version   : 4.14.0
$ media-ctl --device /dev/media0 --set-v4l2 '"ov5640 0-003c":0[fmt:UYVY8_2X8/1280x720@1/30]'
sudo ffmpeg -f v4l2 -video_size 1280x720 -i /dev/video0 -c:v cedrus264 -pix_fmt nv12 test1.mp4
ffmpeg: + 30 -f v4l2 -channel 0 -video_size 1280x720 -i /dev/video0 -pix_fmt nv12 -r 30 -buv 64k -c:v cedrus264 test.mp4
```

Programming i2c and CSI/DVP drivers using IDE:

The screenshot displays the NetBeans IDE interface with two code editors open. The left editor contains the file `ov5640.c`, which includes definitions for `OV5640_FMT_MUX_RGB`, `OV5640_FMT_MUX_DITHER`, `OV5640_FMT_MUX_RAW_DPC`, `OV5640_FMT_MUX_SNR_RAW`, and `OV5640_FMT_MUX_RAW_CIP`. It also defines a `struct ov5640_pixfmt` and a static const array `ov5640_formats[]` containing various media formats like `MEDIA_BUS_FMT_FIXED`, `V4L2_COLORSPACE_RAW`, etc. The right editor contains the file `SUN7I_LIME_Mammoscan2.c`, which includes `#include <media/v4l2-mc.h>` and `#include <media/videobuf2-v4l2.h>`. It defines `CSI_DEFAULT_WIDTH` as 640 and `CSI_DEFAULT_HEIGHT` as 480. It also defines a static const structure `sun4i_csi_format` named `sun4i_csi_formats[]` with fields like `.mbus`, `.fourcc`, `.input`, `.output`, `.num_planes`, `.bpp`, `.hsub`, and `.vsub`. A search results window at the bottom shows two matches for `CSI_MAX_HEIGHT` in `sun4i_csi.h` and `sun4i_v4l2.c`.

```
125     OV5640_FMT_MUX_RGB,
126     OV5640_FMT_MUX_DITHER,
127     OV5640_FMT_MUX_RAW_DPC,
128     OV5640_FMT_MUX_SNR_RAW,
129     OV5640_FMT_MUX_RAW_CIP,
130   };
131
132   struct ov5640_pixfmt {
133     u32 code;
134     u32 colorspace;
135   };
136
137   static const struct ov5640_pixfmt ov5640_formats[] = {
138     { MEDIA_BUS_FMT_FIXED, V4L2_COLORSPACE_RAW, },
139     { MEDIA_BUS_FMT_JPEG_1X8, V4L2_COLORSPACE_JPEG, },
140     { MEDIA_BUS_FMT_UYVY8_2X8, V4L2_COLORSPACE_SRGB, },
141     { MEDIA_BUS_FMT_YUYV8_2X8, V4L2_COLORSPACE_SRGB, },
142     { MEDIA_BUS_FMT_RGB565_2X8_BE, V4L2_COLORSPACE_SRGB, },
143     { MEDIA_BUS_FMT_RGB565_2X8_BE, V4L2_COLORSPACE_SRGB, },
144     { MEDIA_BUS_FMT_SBGGR8_1X8, V4L2_COLORSPACE_SRGB, },
145     { MEDIA_BUS_FMT_SGBRG8_1X8, V4L2_COLORSPACE_SRGB, },
146     { MEDIA_BUS_FMT_SGRBG8_1X8, V4L2_COLORSPACE_SRGB, },
147     { MEDIA_BUS_FMT_SRGG8B_1X8, V4L2_COLORSPACE_SRGB, },
148   };
149
150   /*
151    * FIXME: remove this when a subdev API becomes available
152    * to set the MIPI CSI-2 virtual channel.
153    */
154   static unsigned int virtual_channel;
155   module_param(virtual_channel, uint, 0444);
```

```
13 #include <media/v4l2-mc.h>
14 #include <media/videobuf2-v4l2.h>
15
16 #include "sun4i_csi.h"
17
18 #define CSI_DEFAULT_WIDTH      640
19 #define CSI_DEFAULT_HEIGHT     480
20
21 static const struct sun4i_csi_format sun4i_csi_formats[] = {
22   /* X-Ray raw in / out */
23   {
24     .mbus      = MEDIA_BUS_FMT_FIXED,
25     .fourcc   = V4L2_PIX_FMT_Y16,
26     .input     = CSI_INPUT_PBM,
27     .output    = CSI_OUTPUT_RAW_PASSTHROUGH,
28     .num_planes = 1, // one FIFO is used for RAW
29     .bpp       = { 8, 8, 8 },
30     .hsub     = 1, // skip chroma subsampling,
31     .vsub     = 1, // --
32   },
33   /* YUV422 inputs */
34   {
35     .mbus      = MEDIA_BUS_FMT_YUYV8_2X8,
36     .fourcc   = V4L2_PIX_FMT_YUV420M,
37     .input     = CSI_INPUT_YUV,
38     .output    = CSI_OUTPUT_YUV_420_PLANAR,
39     .num_planes = 3,
40     .bpp       = { 8, 8, 8 },
41     .hsub     = 2,
42     .vsub     = 2,
43   },
44};
```

Find: CSI_MAX_HEIGHT | Previous | Next | Select | 1 match

Search Results ×

File

Found 2 matches of `CSI_MAX_HEIGHT` in 2 files.

- `sun4i_csi.h`: 55: `#define CSI_MAX_HEIGHT 192U` [column 9]
- `sun4i_v4l2.c`: 135: `pix->height = clamp(height, _fmt->vsub, CSI_MAX_HEIGHT);` [column 42]

4.1 NOTE: Camera signal name HREF but A20 CSI0 controller has HSYNC name in documentation.

Figure 2-1 OV5640 block diagram

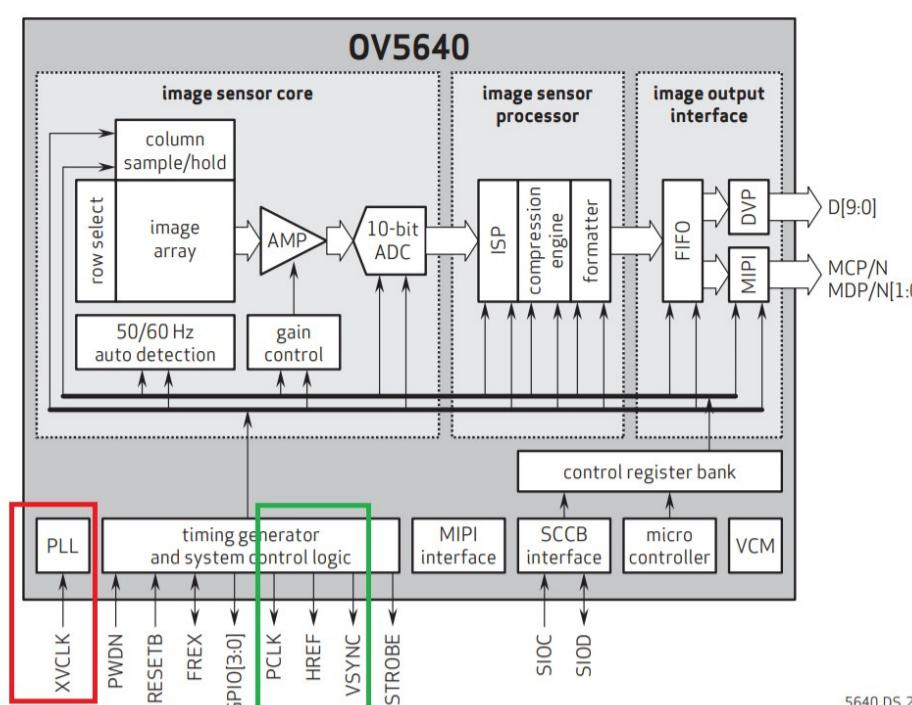


Figure 6-7 DVP timing diagram

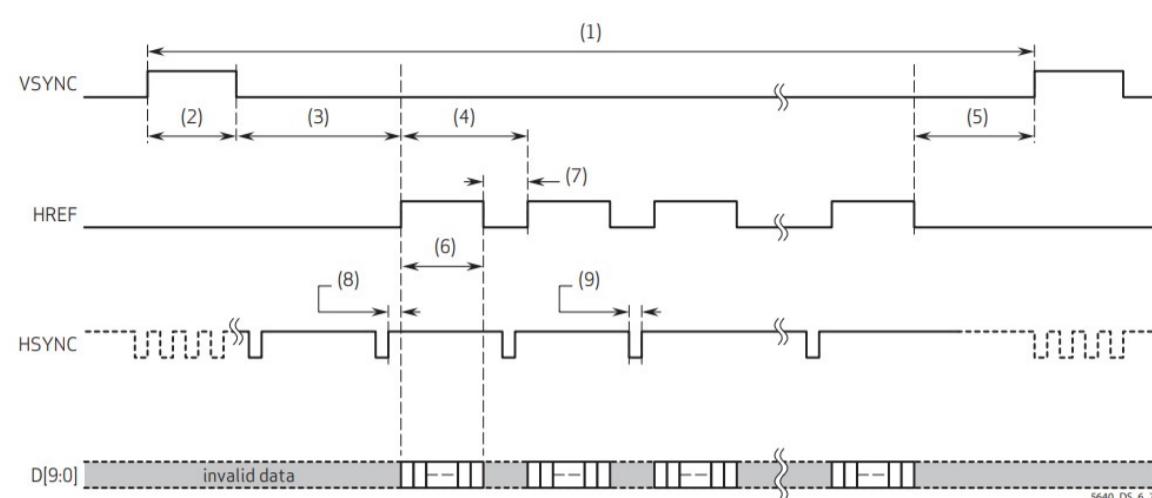


table 6-7

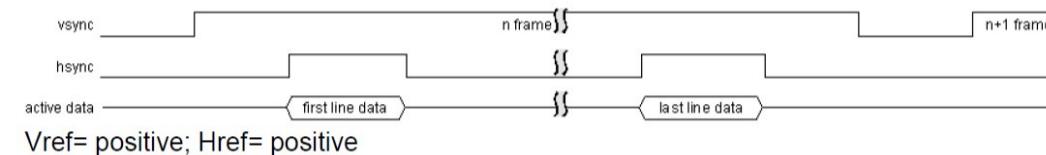
DVP timing specifications (sheet 1 of 2)

GPIO_2

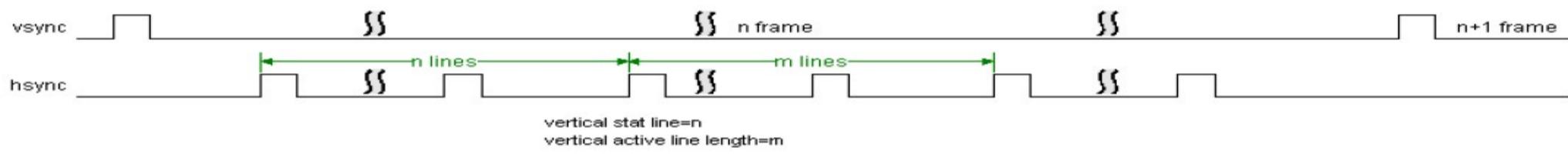
| A20 PIN FUNCTIONS | SIGNAL # | NAME | # | A20 PIN FUNCTIONS | SIGNAL # | NAME | # | A20 PIN FUNCTIONS |
|---------------------------------------|----------|------|----|-------------------|----------|------|----|--------------------------------|
| - | 1 | o | 2 | GND | - | - | - | - |
| - | 3 | o | 4 | LDO3-2.8V | - | - | - | - |
| TWI0_SCK/PB0 | 5 | o | 6 | PE0 | 5 | o | 6 | T50_CLK/CSI0_PCLK/PE0 IN |
| TWI0_SDA/PB1 | 7 | o | 8 | PE1 | 7 | o | 8 | T50_ERR/CSI0_MCLK/PE1 OUT |
| GPS_CLK/PI0 | 9 | o | 10 | PE2 | 9 | o | 10 | T50_SYNC/CSI0_HSYNC/PE2 IN |
| GPS_SIGN/PI1 | 11 | o | 12 | PE3 | 11 | o | 12 | T50_DVLD/CSI0_VSYNC/PE3 IN |
| GPS_MAG/PI2 | 13 | o | 14 | PE4 | 13 | o | 14 | T50_D0/CSI0_D0/PE4 |
| PWM1/PI3 | 15 | o | 16 | PE5 | 15 | o | 16 | T50_D1/CSI0_D1/PE5 |
| SDC3_CMD/PI4 | 17 | o | 18 | PE6 | 17 | o | 18 | T50_D2/CSI0_D2/PE6 |
| SDC3_CLK/PI5 | 19 | o | 20 | PE7 | 19 | o | 20 | T50_D3/CSI0_D3/PE7 |
| SDC3_D0/PI6 | 21 | o | 22 | PE8 | 21 | o | 22 | T50_D4/CSI0_D4/PE8 |
| SDC3_D1/PI7 | 23 | o | 24 | PE9 | 23 | o | 24 | T50_D5/CSI0_D5/PE9 |
| SDC3_D2/PI8 | 25 | o | 26 | PE10 | 25 | o | 26 | T50_D6/CSI0_D6/PE10 |
| SDC3_D3/PI9 | 27 | o | 28 | PE11 | 27 | o | 28 | T50_D7/CSI0_D7/PE11 |
| SPI0_CS0/UART5_TX/EINT22/PI10 | 29 | o | 30 | PI11 | 29 | o | 30 | HSDA/UART7_RX/PS2_SDA0/PI21 |
| SPI0_CLK/UART5_RX/EINT23/PI11 | 31 | o | 32 | PI12 | 31 | o | 32 | HSCL/UART7_TX/PS2_SCK0/PI20 |
| SPI0_MOSI/UART6_TX/EINT24/PI12 | 33 | o | 34 | PI13 | 33 | o | 34 | EINT31/SPI1_MISO/UART2_RX/PI19 |
| SPI0_MISO/UART6_RX/EINT25/PI13 | 35 | o | 36 | PI14 | 35 | o | 36 | EINT30/SPI1_MOSI/UART2_TX/PI18 |
| PS2_SCK1/TCLKIN0/EINT26/SPI0_CS1/PI14 | 37 | o | 38 | PI15 | 37 | o | 38 | EINT29/SPI1_CLK/UART2_CTS/PI17 |
| PS2_SDA1/TCLKIN1/EINT27/SPI1_CS1/PI15 | 39 | o | 40 | PI16 | 39 | o | 40 | EINT28/SPI1_CS0/UART2 RTS/PI15 |

4.1.3.2. TIMING DIAGRAM

CSI timing



Vref= positive; Href= positive



vertical size setting

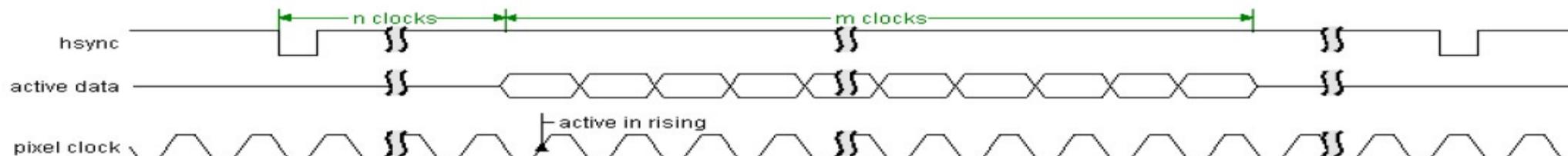
**OV7670/OV7171 CMOS VGA (OmniPixel®) CAMERACHIP™**

Figure 11 RGB 565 Output Timing Diagram

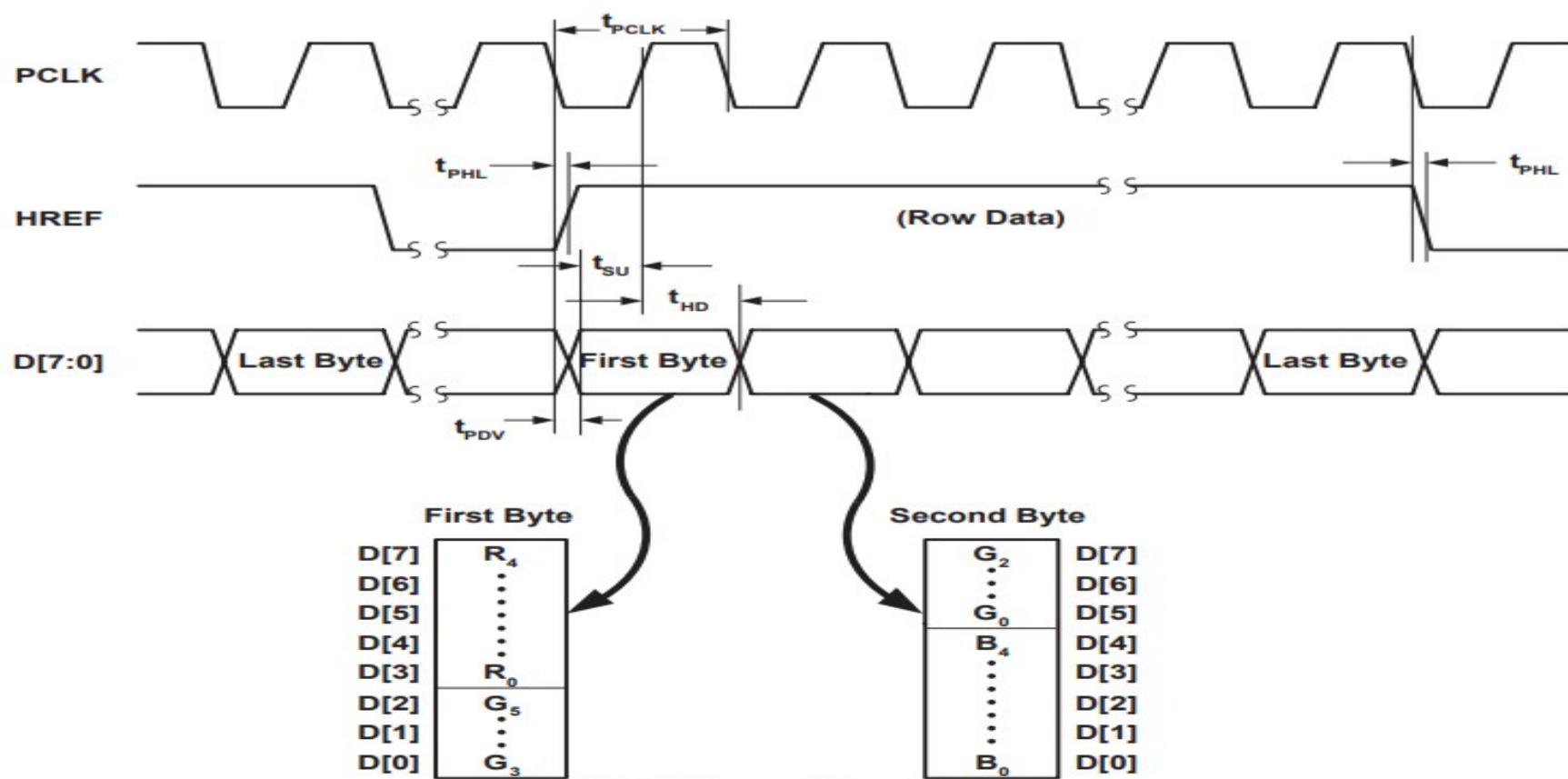


Figure 5 Horizontal Timing

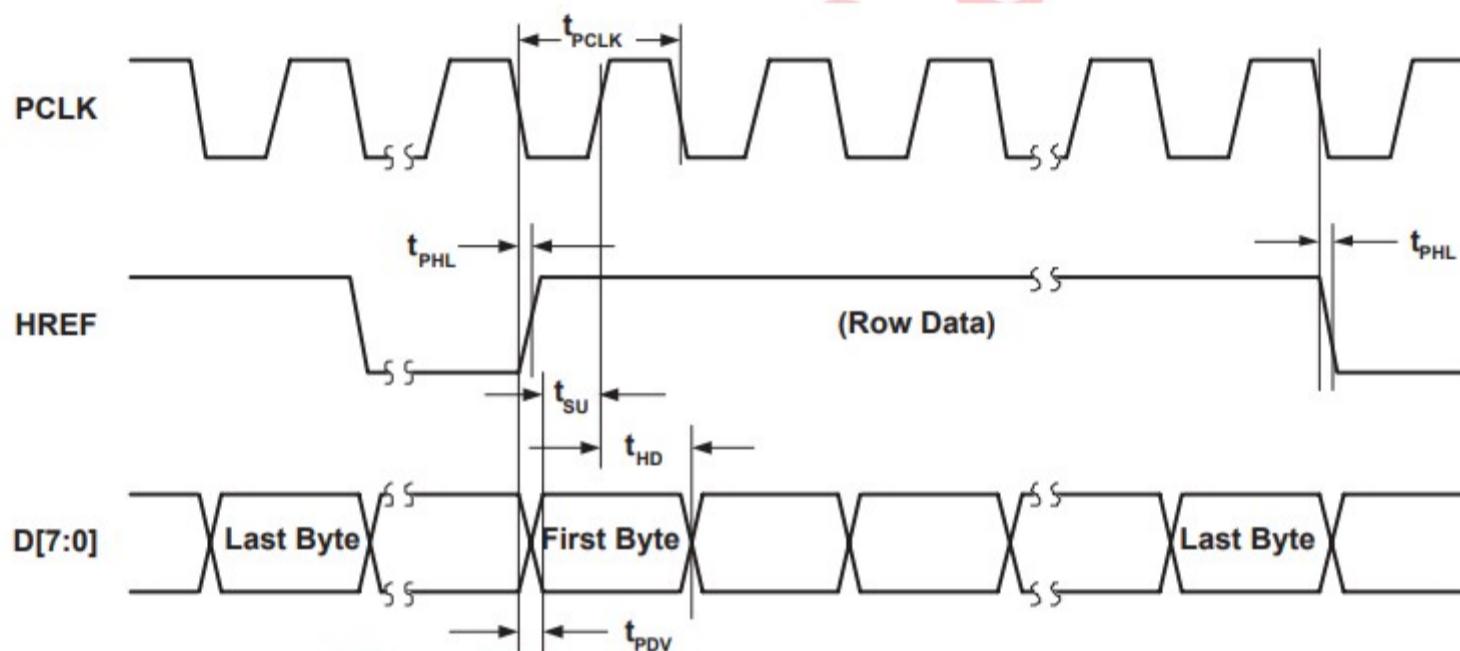
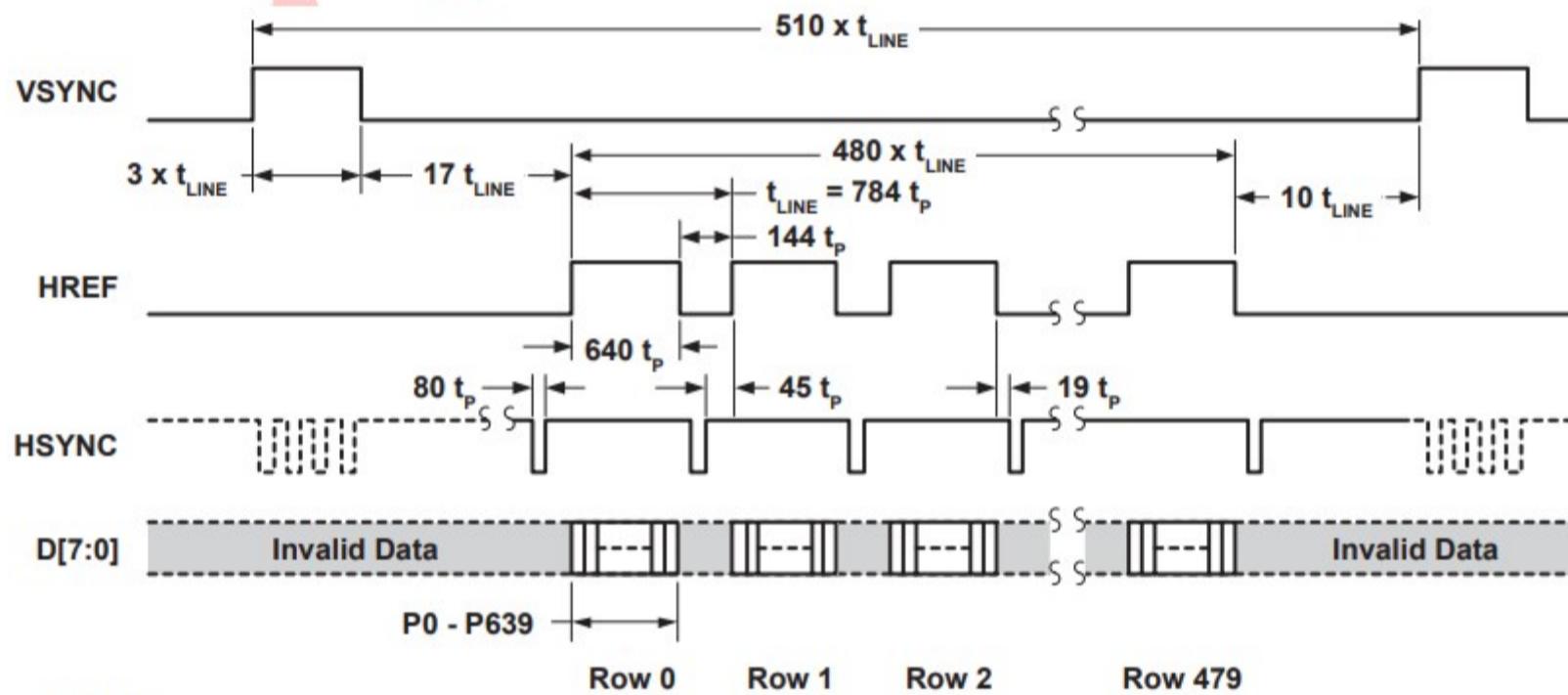


Figure 6 VGA Frame Timing



NOTE:

For Raw data, $t_p = t_{PCLK}$

For YUV/RGB, $t_p = 2 \times t_{PCLK}$

Line/Pixel Timing

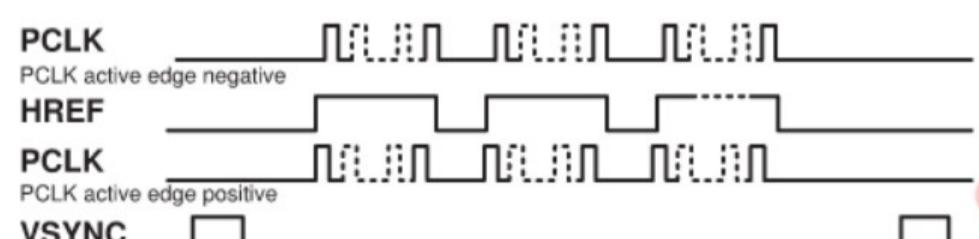
Ov2640 Camera PDF

The OV2640 digital video port can be programmed to work in either master or slave mode.

In both master and slave modes, pixel data output is synchronous with PCLK (or MCLK if port is a slave), HREF, and VSYNC. The default PCLK edge for valid data is the negative edge but may be programmed using register COM10[4] for the positive edge. Basic line/pixel output timing and pixel timing specifications are shown in [Figure 14](#) and [Table 10](#).

Also, using register COM10[5], PCLK output can be gated by the active video period defined by the HREF signal. See [Figure 11](#) for details.

Figure 11 PCLK Output Only at Valid Pixels



The specifications shown in [Table 10](#) apply for DVDD = +1.2 V, DOVDD = +2.8 V, TA = 25°C, sensor working at 15 fps, external loading = 20 pF.

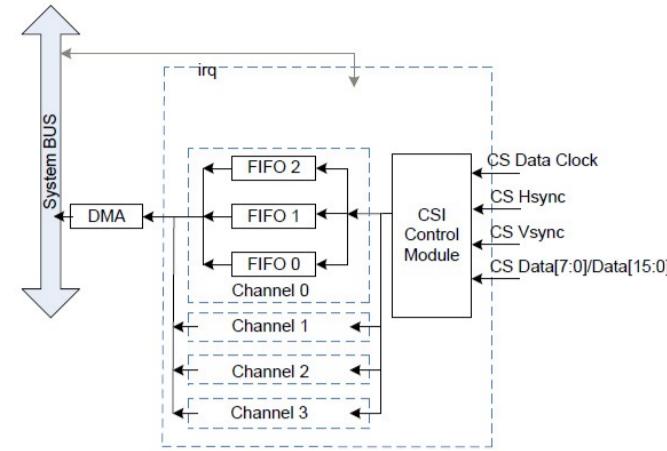
4.1. CSI0

4.1.1. Overview

CSI0 features:

- 8 bits input data
- Support CCIR656 protocol for NTSC and PAL
- 3 parallel data paths for image stream parsing
- Received data double buffer support
- Parsing bayer data into planar R, G, B output to memory
- Parsing interlaced data into planar or tiled Y, Cb, Cr output to memory
- Pass raw data direct to memory
- All data transmit timing can be adjusted by software
- Support multi-channel ITU-R BT656 time-multiplexed format
- Luminance statistical value
- Support 8-bit raw data input
- Support 16-bit YUV422 data input

4.1.2. CSI0 Block Diagram



4.1.3. CSI0 Description

4.1.3.1. CSI DATA PORTS

| | Bayer | YCbCr (YUV) | Interlaced | Pass-through |
|-------|------------------|-------------------|------------------------|----------------|
| FIFO0 | Red pixel data | Y pixel data | All field 1 pixel data | All pixel data |
| FIFO1 | Green pixel data | Cb (U) pixel data | All field 2 pixel data | - |
| FIFO2 | Blue pixel data | Cr (V) pixel data | - | - |

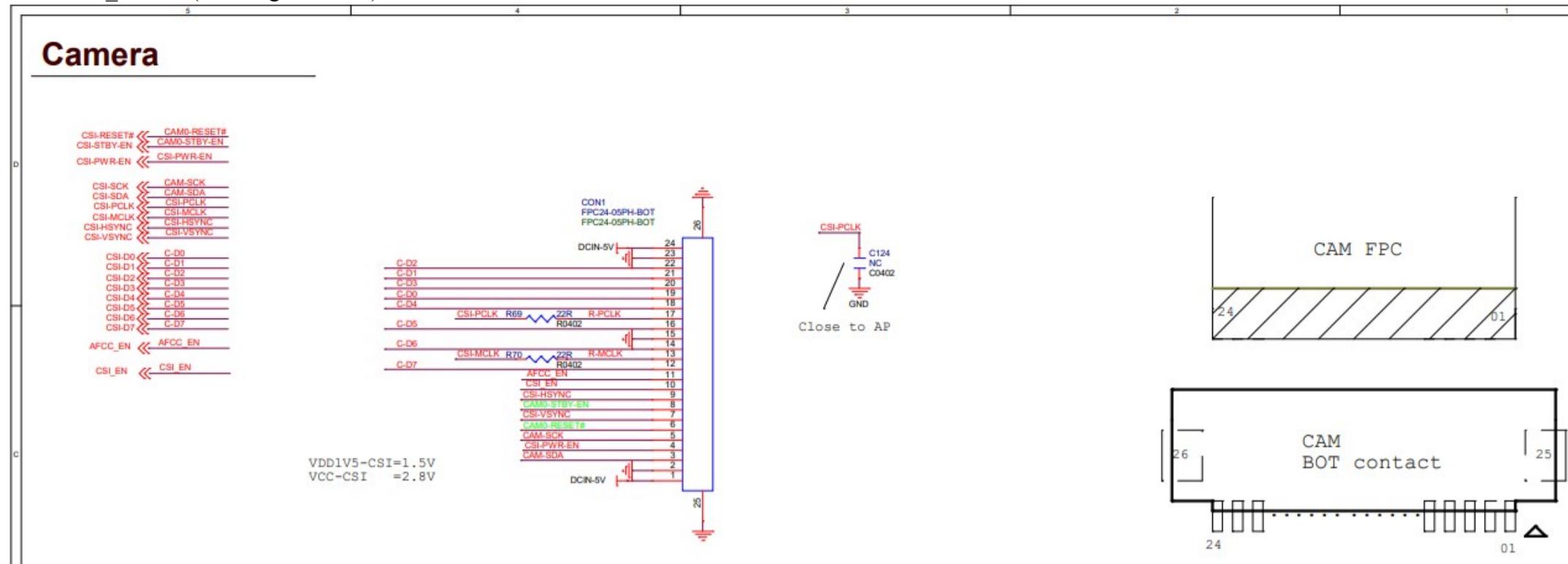
Question about HREF vs HSYNC from linux driver:

```

/*
 * This hardware uses [HV]REF instead of [HV]SYNC. Based on the
 * provided timing diagrams in the manual, positive polarity
 * equals active high [HV]REF.
 *
 * When the back porch is 0, [HV]REF is more or less equivalent
 * to [HV]SYNC inverted.
 */
href_pol = !!(bus->flags & V4L2_MBUS_HSYNC_ACTIVE_LOW);
vref_pol = !!(bus->flags & V4L2_MBUS_VSYNC_ACTIVE_LOW);
pclk_pol = !!(bus->flags & V4L2_MBUS_PCLK_SAMPLE_RISING);
write1(CSI_CFG_INPUT_FMT(csi_fmt->input) |
       CSI_CFG_OUTPUT_FMT(csi_fmt->output) |
       CSI_CFG_VREF_POL(vref_pol) |
       CSI_CFG_HREF_POL(href_pol) |
       CSI_CFG_PCLK_POL(pclk_pol),
       csi->regs + CSI_CFG_REG);

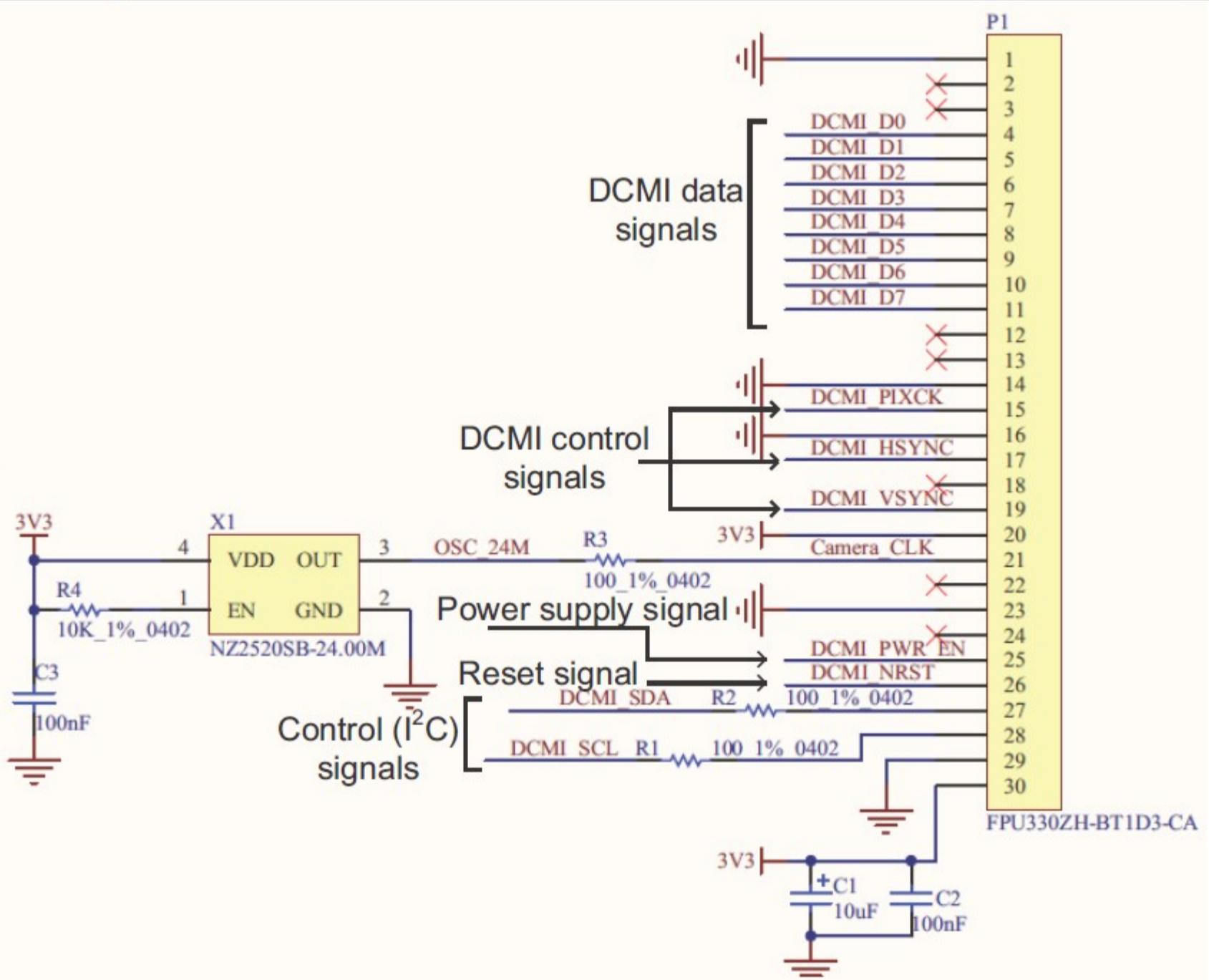
```

ORANGE_PI-One (Xunlong Software):



Digital camera interface (DCMI) for STM32 MCUs AN5020:

Figure 46. Camera connector on the 32F746GDISCOVERY board



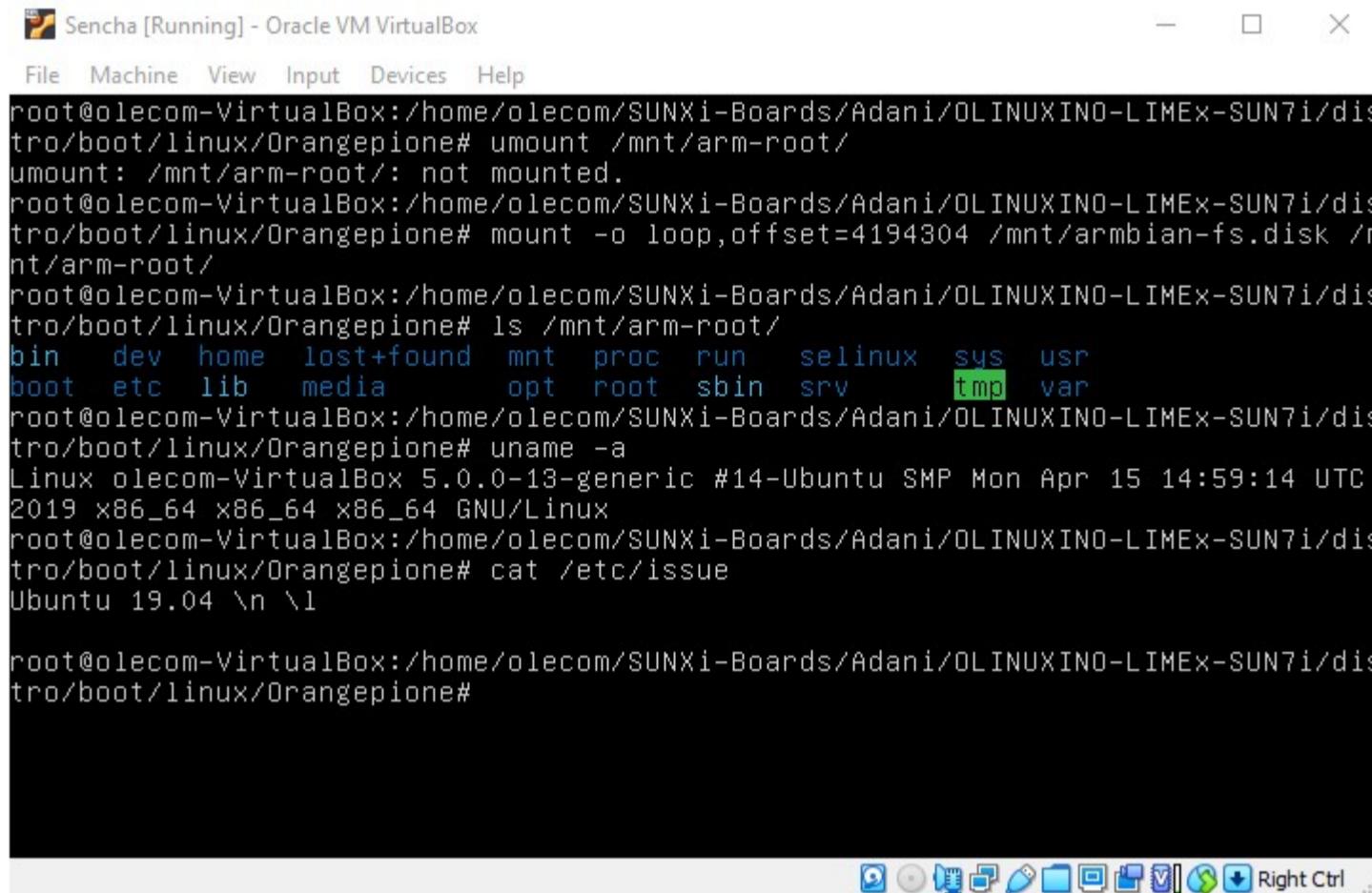
MSv46666V1

6. qemu

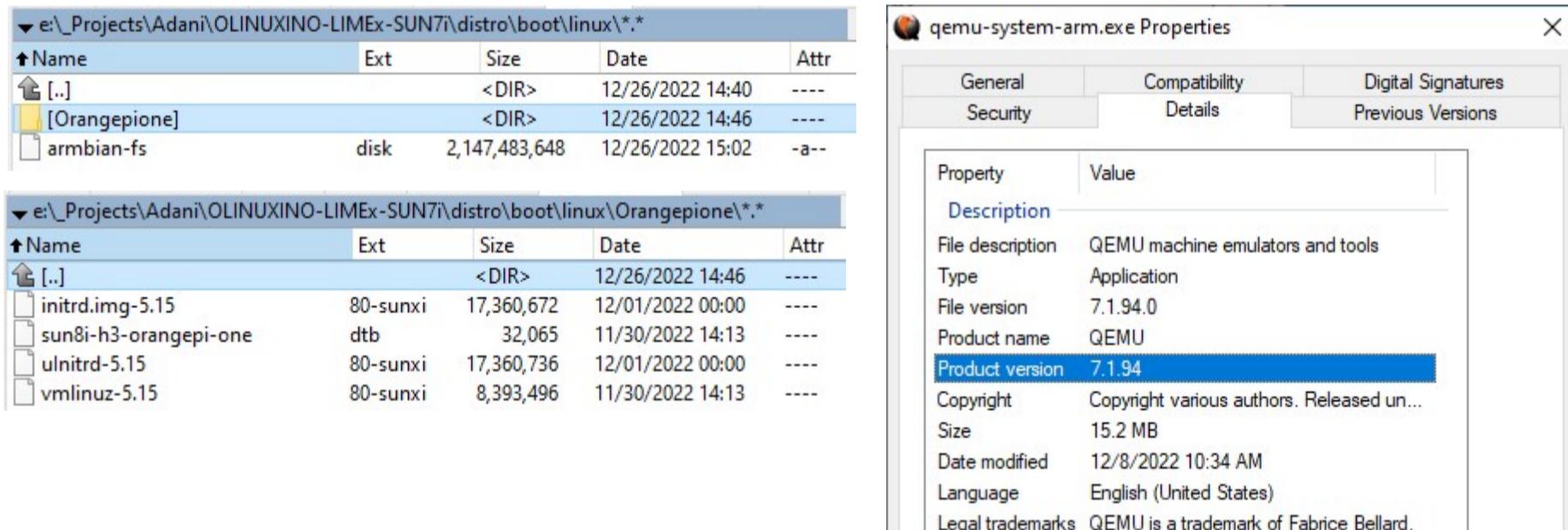
MS Windows 10 + qemu + Armbian for orangepi-pc. Setup description: <https://gist.github.com/wuhanstudio/e9b37b07312a52ceb5973aacf580c453>

Setup is on PC with Linux:

```
xz -d Armbian_22.11.1_Orangepine_jammy_current_5.15.80_minimal.img.xz
fallocate -l 2G /mnt/armbian-fs.disk
dd conv=notrunc of=/mnt/armbian-fs.disk bs=128M if=Armbian_22.11.1_Lime2_bullseye_current_5.15.80.img
mount -o loop,offset=4194304 /mnt/armbian-fs.disk /mnt/arm-root/
# copy kernel, initramfs, dtb files
```



Running on MS Windows (because Linux is too old to install and use **qemu**):



Before that SIGINT handler of current terminal is changed for CTRL+C to CTRL+Q by **stty intr ^Q** thus CTRL+C doesn't kill **qemu** itself. This lets to use CTRL+C as usual process interruption method inside emulation.

```
legion@LAPTOP-SP2SBN7B MINGW64 /e/_Projects/Adani/OLINUXINO-LIMEx-SUN7i/distro/boot/linux/Orangepine
$ stty -a | grep intr
intr = ^Q; quit = ^\; erase = ^?; kill = ^U; eof = ^D; eol = <undef>

legion@LAPTOP-SP2SBN7B MINGW64 /e/_Projects/Adani/OLINUXINO-LIMEx-SUN7i/distro/boot/linux/Orangepine
$ ./bin/qemu/qemu-system-arm -M orangepi-pc -m 1G -smp 4 -kernel vmlinuz-5.15.80-sunxi -dtb sun8i-h3-orangepi-one.dtb -initrd initrd.img-5.15.80-sunxi -sd ./armbian-fs.disk -append 'console=ttyS0,115200 root=/dev/mmcblk0p1' -no-reboot -serial stdio -nographic -monitor none
WARNING: Image format was not specified for '../armbian-fs.disk' and probing guessed raw.
        Automatically detecting the format is dangerous for raw images, write operations on block 0 will be restricted.
        Specify the 'raw' format explicitly to remove the restrictions.
[ 0.000000] Booting Linux on physical CPU 0x0
[ 0.000000] Linux version 5.15.80-sunxi (root@f42af18c1614) (arm-linux-gnueabihf-gcc (GNU Toolchain for the A-profile Architecture 8.3-2019.03 (arm-rel-8.36)) 8.3.0, GNU ld (GNU Toolchain for the A-profile Architecture 8.3-2019.03 (arm-rel-8.36)) 2.32.0.20190321) #22.11.1 SMP Wed Nov 30 11:13:48 UTC 2022
[ 0.000000] CPU: ARMv7 Processor [410fc075] revision 5 (ARMv7), cr=50c5387d
[ 0.000000] CPU: div instructions available: patching division code
[ 0.000000] CPU: PIPT / VIPT nonaliasing data cache, VIPT aliasing instruction cache
[ 0.000000] OF: fdt: Machine model: Xunlong Orange Pi One
[ 0.000000] Memory policy: Data cache writealloc
[ 0.000000] cma: Reserved 128 MiB at 0x78000000
[ 0.000000] Zone ranges:
[ 0.000000]     Normal   [mem 0x0000000040000000-0x000000006fffffff]
```

```
MINGW64:/e/_Projects/Adani/OLINUXINO-LIMEx-SUN7i/distro/boot/linux/Orangepine
[ 31.101922] systemd[1]: Inserted module 'autofs4'
[ 32.357026] systemd[1]: systemd 249.11-0ubuntu3.6 running in system mode (+PAM +AUDIT +SELINUX +APPARMOR +IMA +SMACK +SECCOMP +GCRY
PT +GNUTLS +OPENSSL +ACL +BLKID +CURL +ELFUTILS +FIDO2 +IDN +IPTC +KMOD +LIBCRYPTSETUP +LIBFDISK +PCRE2 -PWQUALITY -P11KIT -QREN
CODE +BZIP2 +LZ4 +XZ +ZLIB -XKBCOMMON +UTMP +SYSVINIT default-hierarchy=unified)
[ 32.412173] systemd[1]: Detected architecture arm.
```

Welcome to Armbian 22.11.1 Jammy!

```
[ 32.483405] systemd[1]: Hostname set to <orangepine>.
[ 38.427670] vcc3v0: disabling
[ 38.428746] vcc5v0: disabling
[ 42.344492] systemd[1]: Queued start job for default target Graphical Interface.
[ 42.366353] random: systemd: uninitialized urandom read (16 bytes read)
[ 42.466020] systemd[1]: Created slice Slice /system/modprobe.
[ OK ] Created slice Slice /system/modprobe.
[ 42.476827] random: systemd: uninitialized urandom read (16 bytes read)
[ 42.494539] systemd[1]: Created slice Slice /system/serial-getty.
[ OK ] Created slice Slice /system/serial-getty.
[ 42.498840] random: systemd: uninitialized urandom read (16 bytes read)
[ 42.514046] systemd[1]: Created slice User and Session Slice.
[ OK ] Created slice User and Session Slice.
[ 42.529818] systemd[1]: Started Forward Password Requests to Wall Directory Watch.
[ OK ] Started Forward Password Requests to Wall Directory Watch.
[ 42.559898] systemd[1]: Set up automount Arbitrary Executable File Formats File System Automount Point.
```

```
MINGW64:/e/_Projects/Adani/OLINUXINO-LIMEx-SUN7i/distro/boot/linux/Orangepine
Armbian 22.11.1 Jammy ttyS0

orangepine login: root
root
Password: 1

Welcome to Armbian 22.11.1 Jammy with Linux 5.15.80-sunxi

System load: 6% Up time: 15 min
Memory usage: 6% of 998M IP: 10.0.2.15
Usage of /: 48% of 1.9G

[ 0 security updates available, 2 updates total: apt upgrade | Kernel and firmware upgrades disabled: armbian-config ]
Last check: 2022-12-26 22:00

[ Menu-driven system configuration (beta): sudo apt update && sudo apt install armbian-config ]

root@orangepine:~#
```

TODO: there are some problems with terminal input on git-windows console. Need to make it **TERM=xterm**?

Using built in console:

```
/e/bin/qemu/qemu-system-arm -M orangepi-pc -m 1G -smp 4 -kernel vmlinuz-5.15.80-sunxi -dtb sun8i-h3-orangepi-one.dtb -initrd initrd.img-5.15.80-sunxi -sd ./armbian-fs.disk -append 'console=ttyS0,115200 root=/dev/mmcblk0p1' -no-reboot
```

