

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.

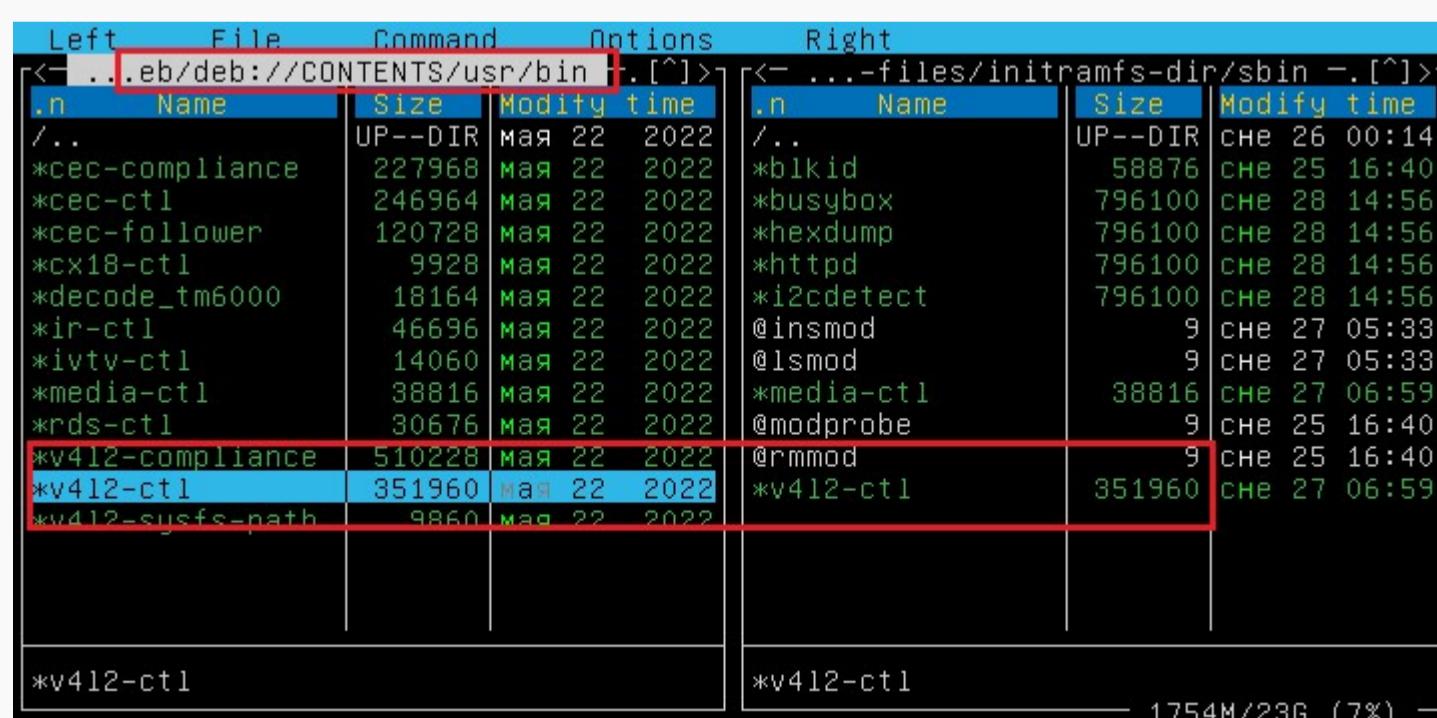
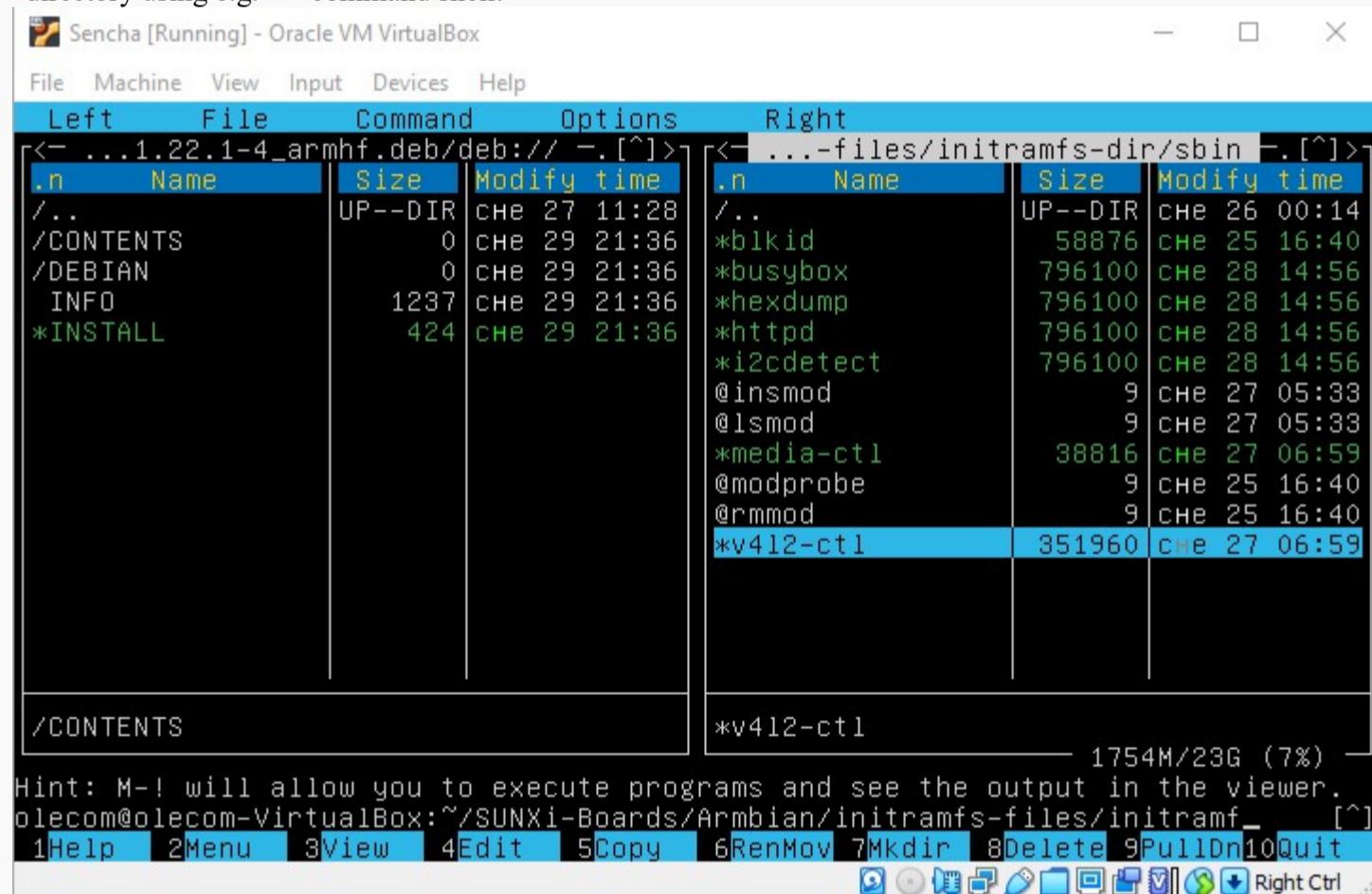
1.3. Manual update of existing initramfs image is possible inside Linux-compatible system where file system supports symbolic / hardlinks used.

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:



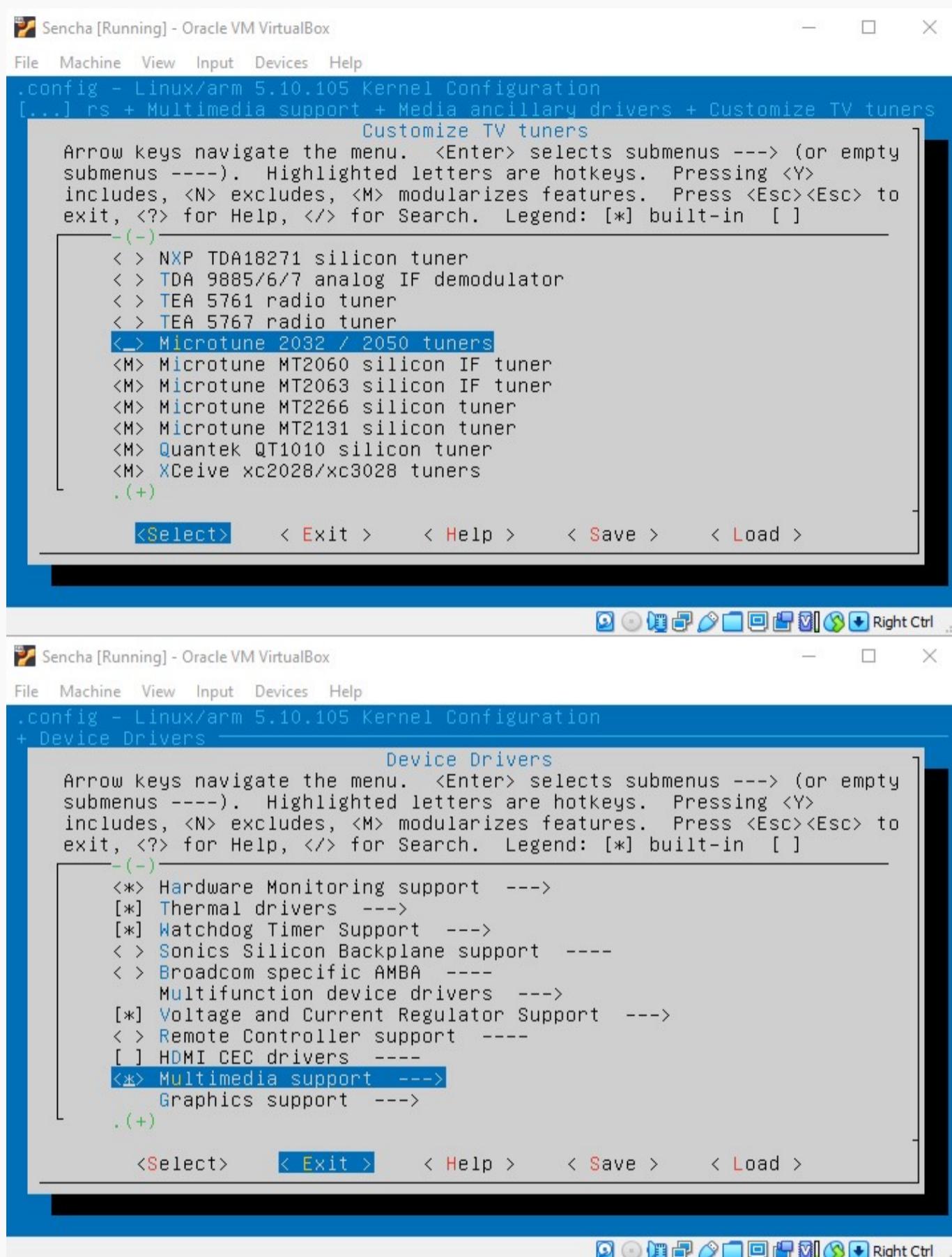
Then generate image using **adani-cu-tools/fs-host-any-gnu-linux/initramfs/mkinitrd.sh** which can be copied into /boot of microFlash card.

2. Compiling Linux kernel with device tree files and Busybox.

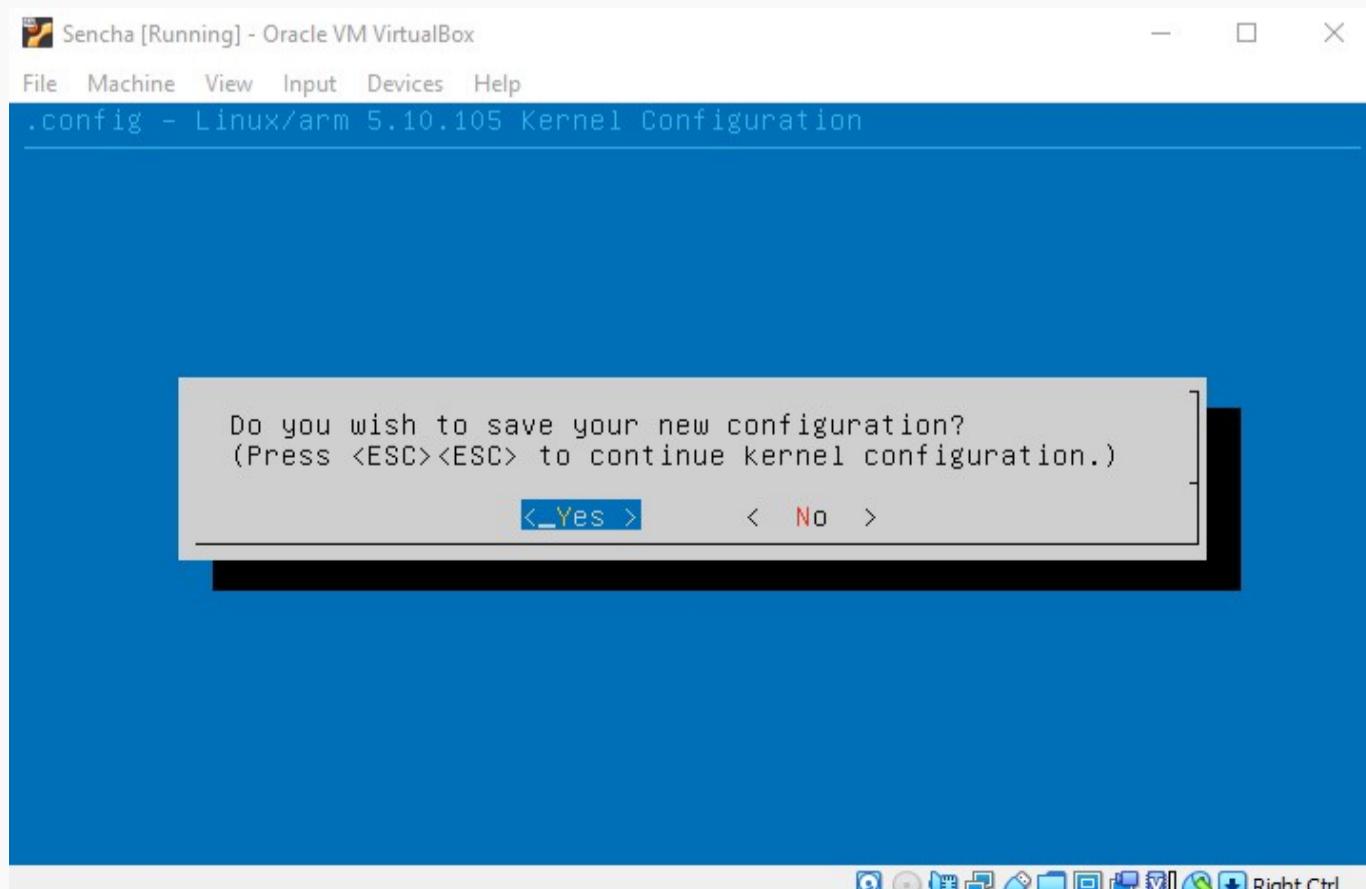
2.1 Get the source: **adani-cu-tools/fs-host-any-gnu-linux/linux-olimex/get-linuX-olimex.sh**

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:

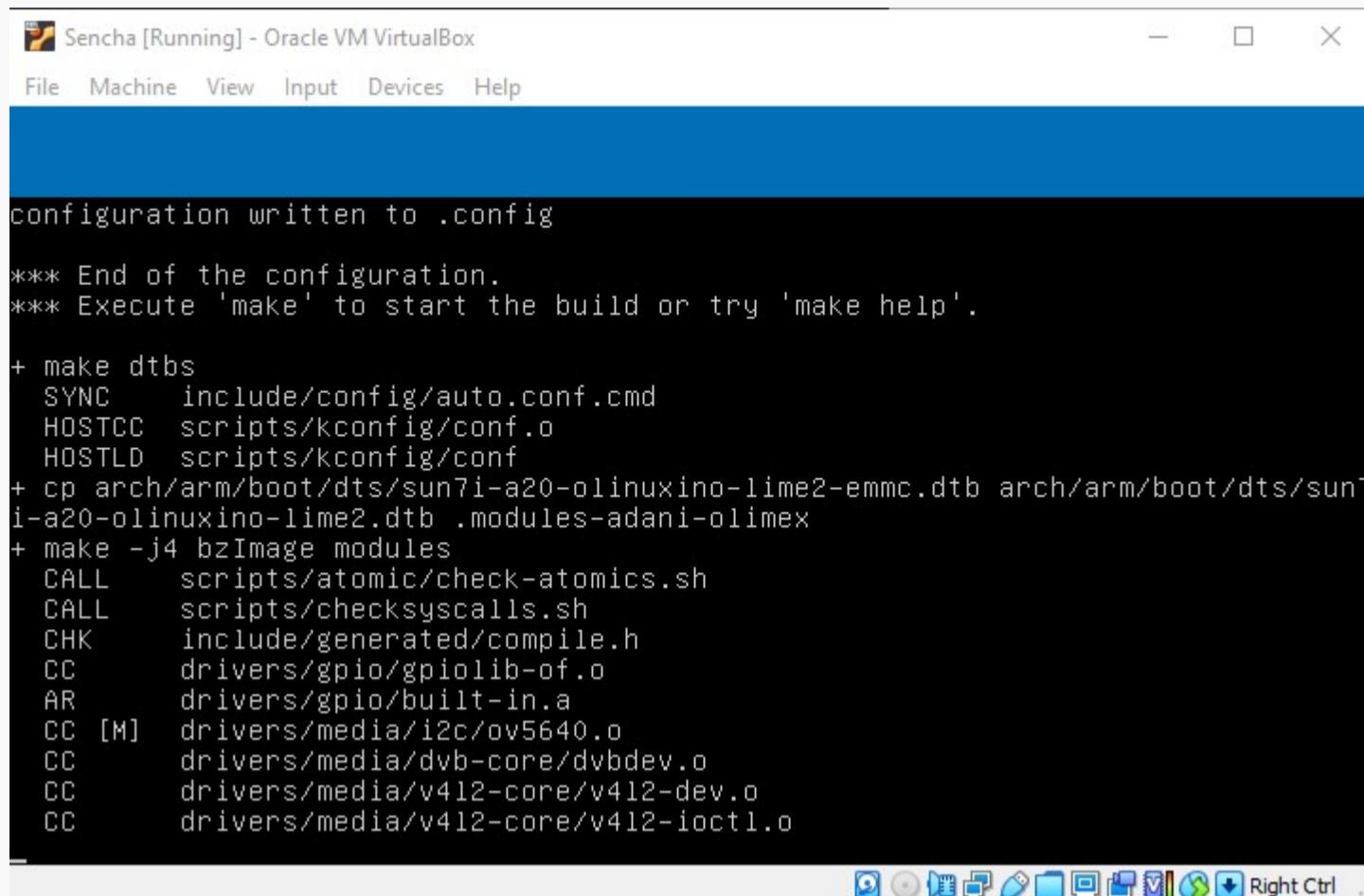


New config file is saved:



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

Start:



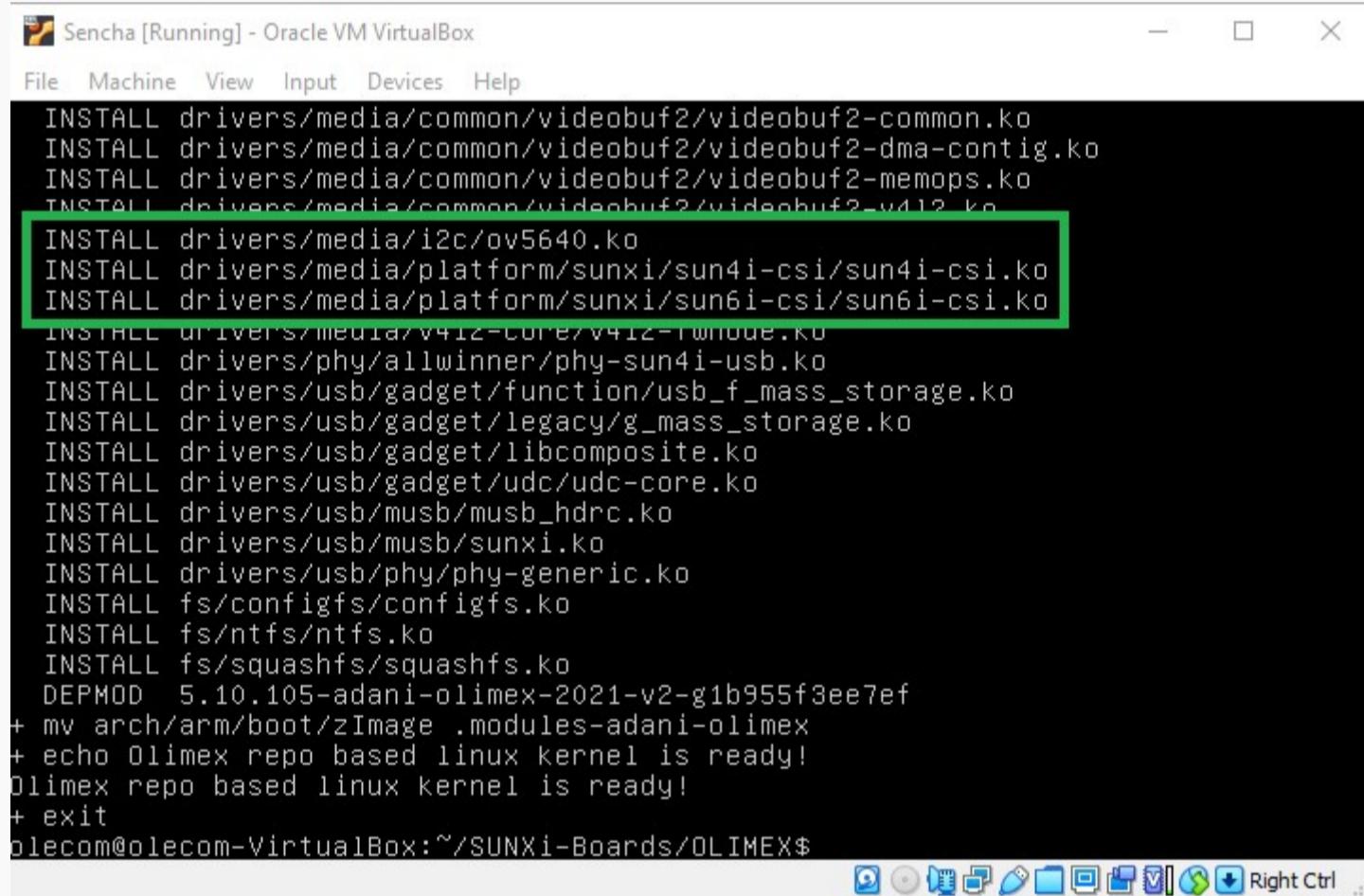
```
configuration written to .config
*** End of the configuration.
*** Execute 'make' to start the build or try 'make help'.

+ 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/v4l2-core/v4l2-dev.o
  CC    drivers/media/v4l2-core/v4l2-iocctl.o
```

Finish. Needed driver files are highlighted. Also there are two device tree files built:

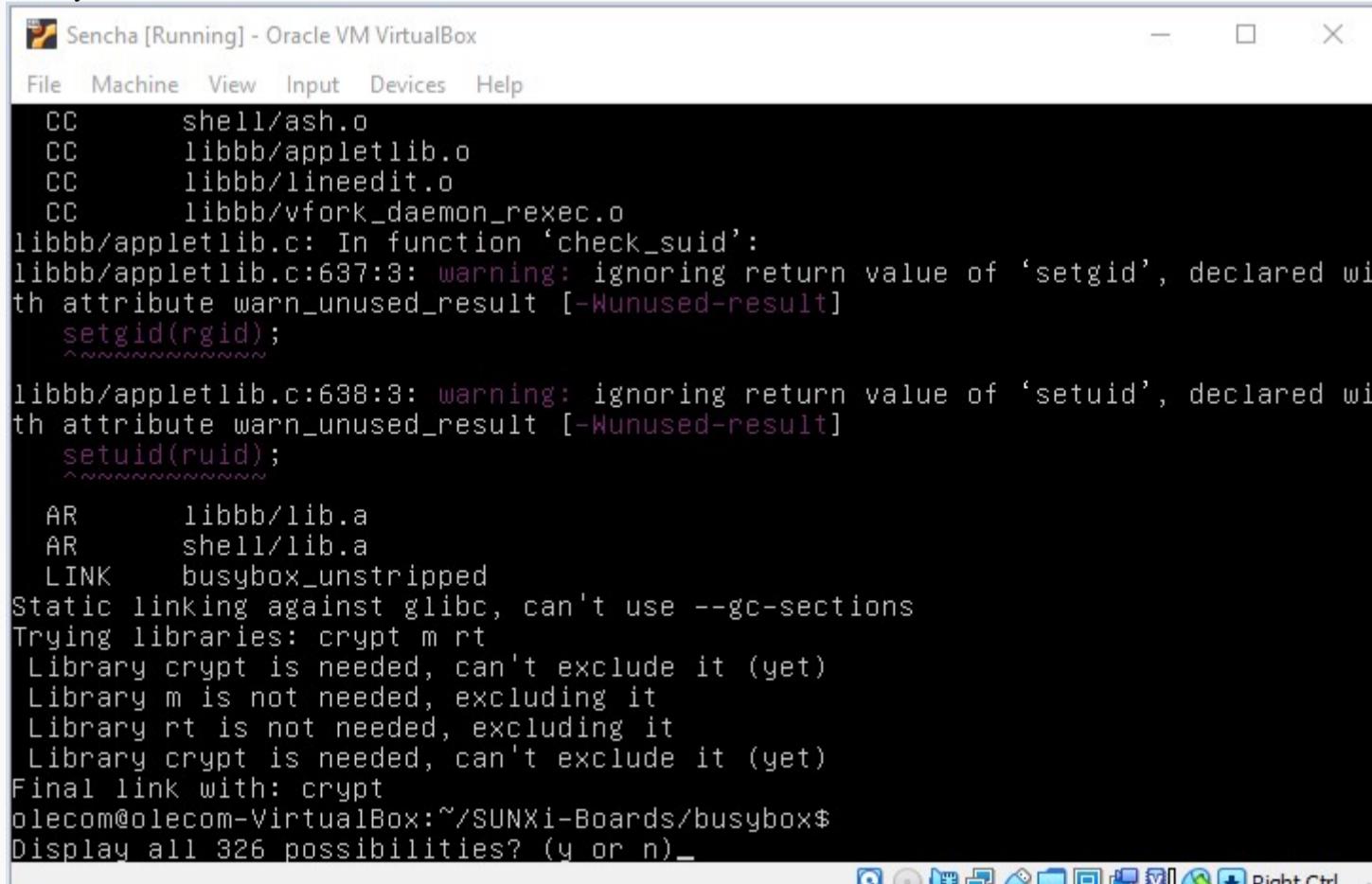
sun7i-a20-olinuxino-lime2.dtb

sun7i-a20-olinuxino-lime2-emmc.dtb



```
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-v4l2.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/media/v4l2-core/v4l2-i2c-new.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>

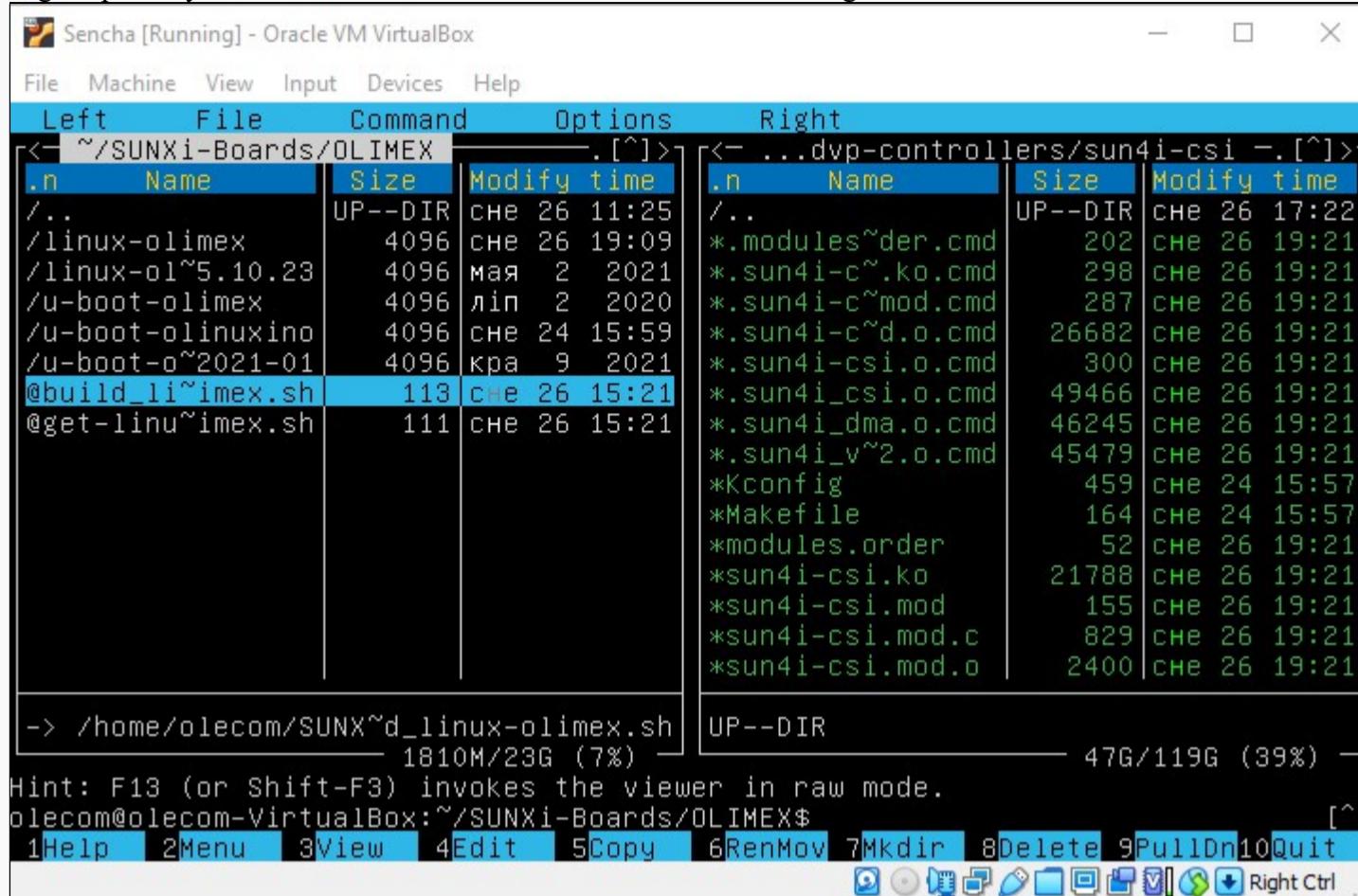


```
CC      shell/ash.o
CC      libbb/appletlib.o
CC      libbb/lineedit.o
CC      libbb/vfork_daemon_reexec.o
libbb/appletlib.c: In function ‘check_suid’:
libbb/appletlib.c:637:3: warning: ignoring return value of ‘setgid’, declared with attribute warn_unused_result [-Wunused-result]
  setgid(rgid);
  ~~~~~
libbb/appletlib.c:638:3: warning: ignoring return value of ‘setuid’, declared with attribute warn_unused_result [-Wunused-result]
  setuid(ruid);
  ~~~~~
AR      libbb/lib.a
AR      shell/lib.a
LINK   busybox_unstripped
Static linking against glibc, can't use --gc-sections
Trying libraries: crypt m rt
  Library crypt is needed, can't exclude it (yet)
  Library m is not needed, excluding it
  Library rt is not needed, excluding it
  Library crypt is needed, can't exclude it (yet)
Final link with: crypt
olecom@olecom-VirtualBox:~/SUNXi-Boards/busybox$
Display all 326 possibilities? (y or n)_
```

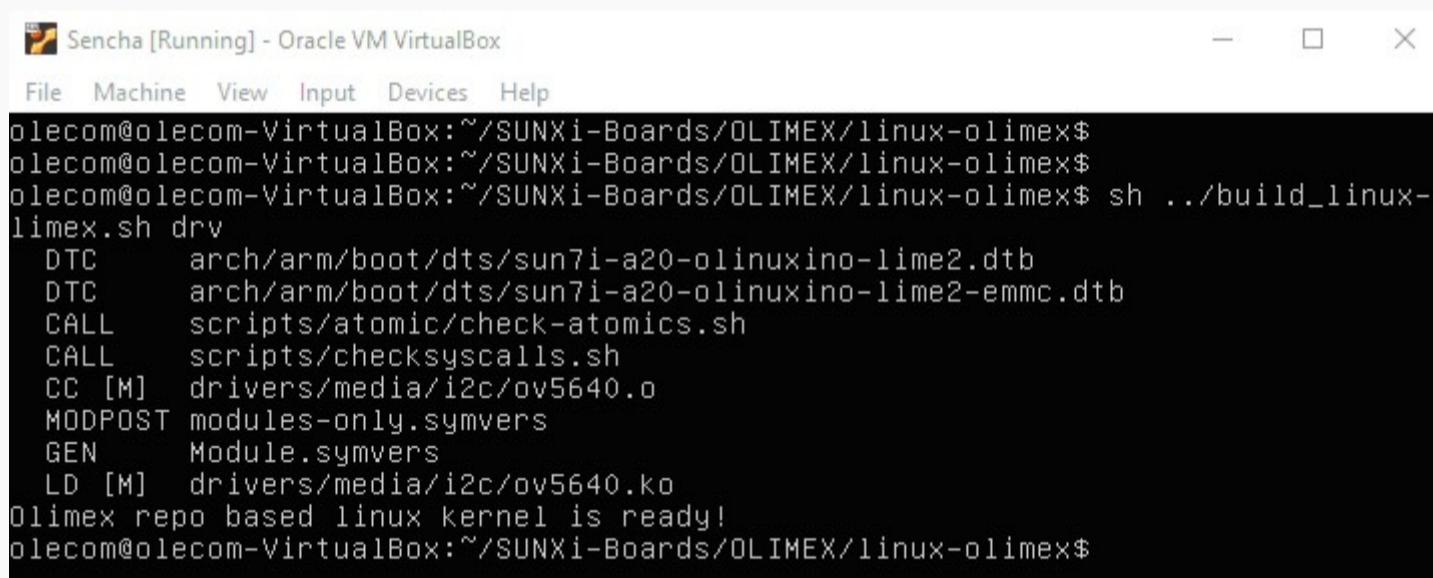
3. Linux kernel drivers development.

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.

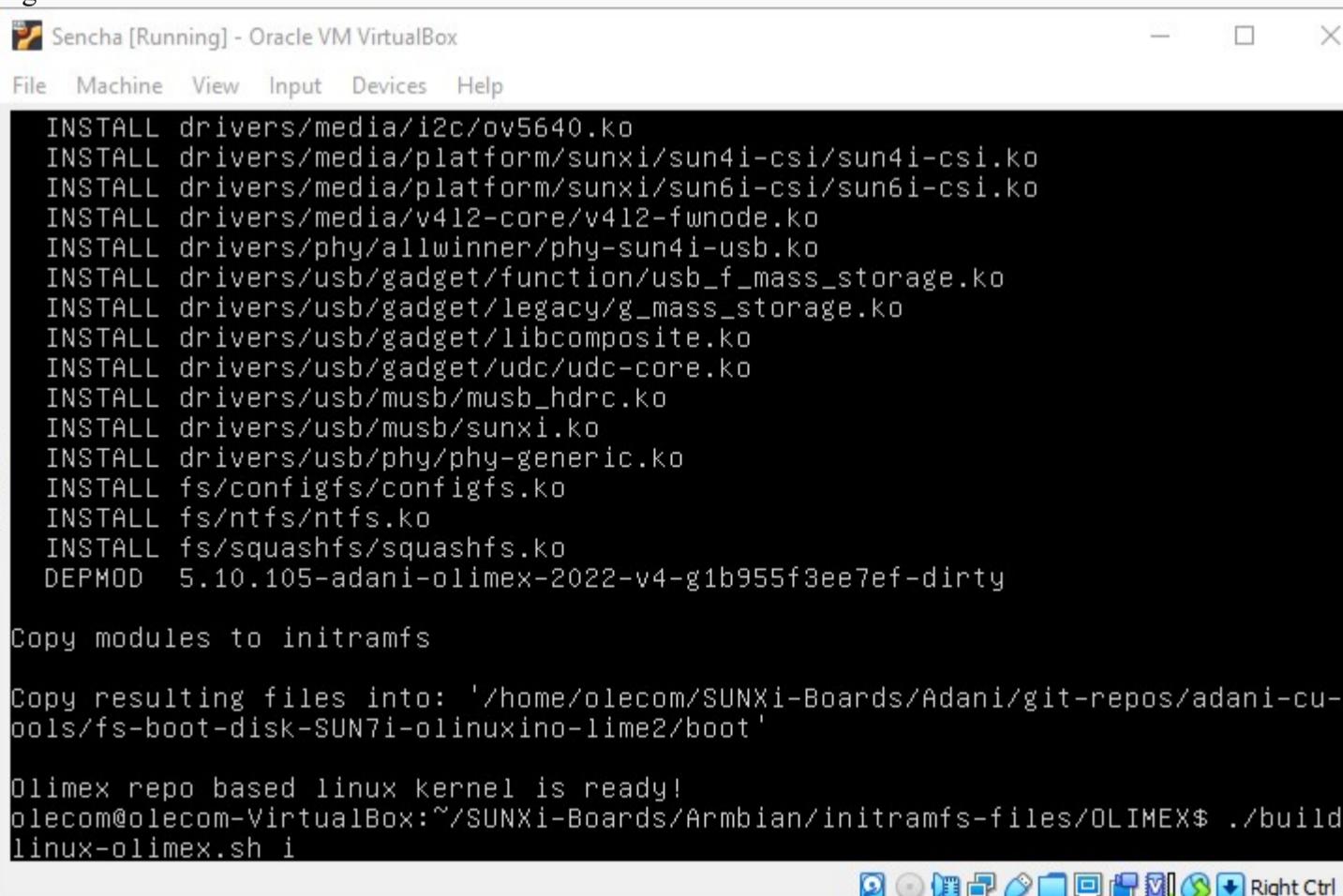
Working directory on the left, git repository **adani-linux-drivers** of CSI drivers on the right:



Compilation process:



Modules installation / copying:



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

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 `sun4i_dma.c`, which includes definitions for various pixel formats like `OV5640_FMT_MUX_RGB` and `OV5640_FMT_MUX_DITHER`. It also defines a structure `ov5640_pixfmt` and a static array `ov5640_formats` containing these formats. The right editor contains the file `SUN7I_LIME_Mammoscan2.c`, which includes headers for media/v4l2-mc.h and media/videobuf2-v4l2.h, and defines constants for `CSI_DEFAULT_WIDTH` and `CSI_DEFAULT_HEIGHT`. It also defines a static const structure `sun4i_csi_format` named `sun4i_csi_formats` with various fields such as `.mbus`, `.fourcc`, `.input`, `.output`, `.num_planes`, `.bpp`, `.hsub`, and `.vsub`. A search results window at the bottom shows two matches for the symbol `CSI_MAX_HEIGHT`, located 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

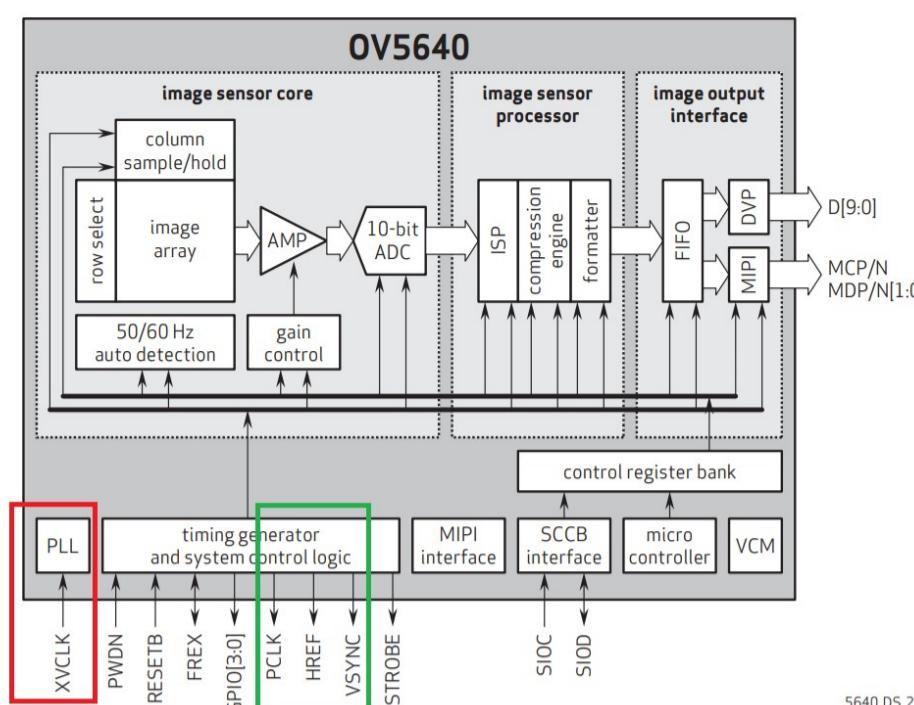


Figure 6-7 DVP timing diagram

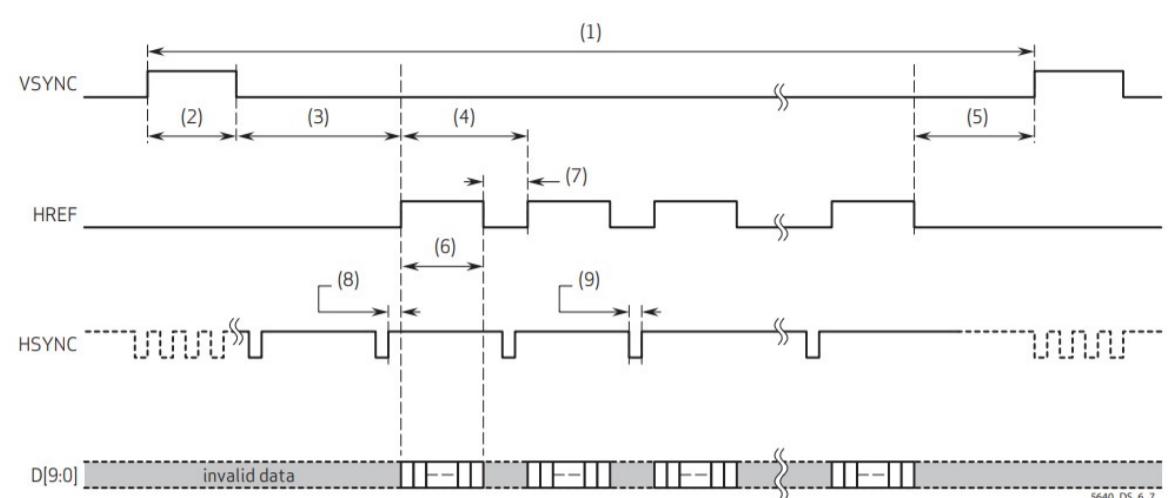


table 6-7

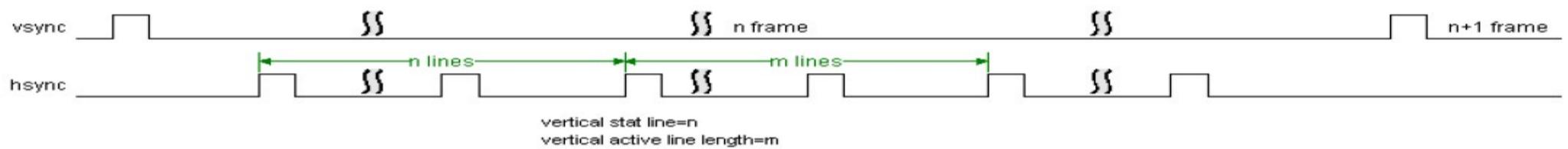
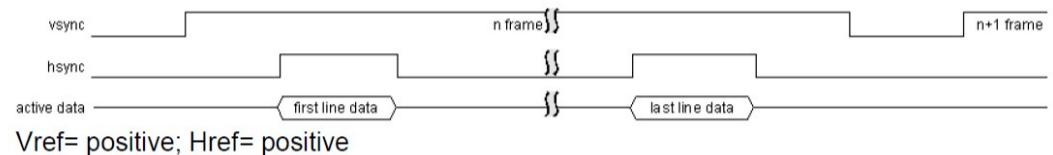
DVP timing specifications (sheet 1 of 2)

GPIO_2

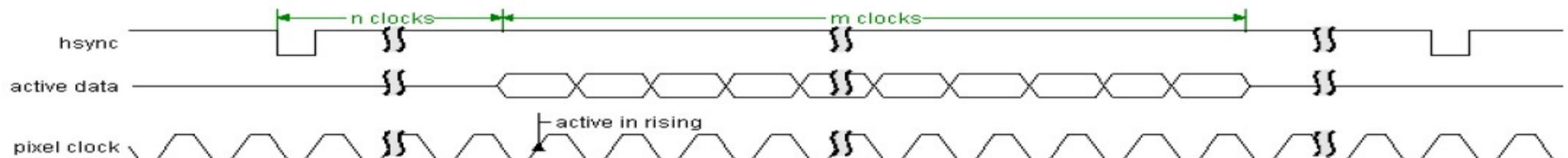
A20 PIN FUNCTIONS	SIGNAL # NAME	#	#	SIGNAL # NAME	A20 PIN FUNCTIONS
-	5V	1	o	2	GND
-	3.3V	3	o	4	LDO3-2.8V
TWI0_SCK/PB0	TWI0-SCK	5	o	6	PE0
TWI0_SDA/PB1	TWI0-SDA	7	o	8	PE1
GPS_CLK/PI0	PI0	9	o	10	PE2
GPS_SIGN/PI1	PI1	11	o	12	PE3
GPS_MAG/PI2	PI2	13	o	14	PE4
PWM1/PI3	PI3	15	o	16	PE5
SDC3_CMD/PI4	PI4	17	o	18	PE6
SDC3_CLK/PI5	PI5	19	o	20	PE7
SDC3_D0/PI6	PI6	21	o	22	PE8
SDC3_D1/PI7	PI7	23	o	24	PE9
SDC3_D2/PI8	PI8	25	o	26	PE10
SDC3_D3/PI9	PI9	27	o	28	PE11
SPI0_CS0/UART5_TX/EINT22/PI10	PI10	29	o	30	PI11
SPI0_CLK/UART5_RX/EINT23/PI11	PI11	31	o	32	PI12
SPI0_MOSI/UART6_TX/EINT24/PI12	PI12	33	o	34	PI13
SPI0_MISO/UART6_RX/EINT25/PI13	PI13	35	o	36	PI14
PS2_SCK1/TCLKIN0/EINT26/SPI0_CS1/PI14	PI14	37	o	38	PI15
PS2_SDA1/TCLKIN1/EINT27/SPI1_CS1/PI15	PI15	39	o	40	PI16

4.1.3.2. TIMING DIAGRAM

CSI timing



vertical size setting



OV7670/OV7171 CMOS VGA (OmniPixel®) CAMERACHIP™

Figure 11 RGB 565 Output Timing Diagram

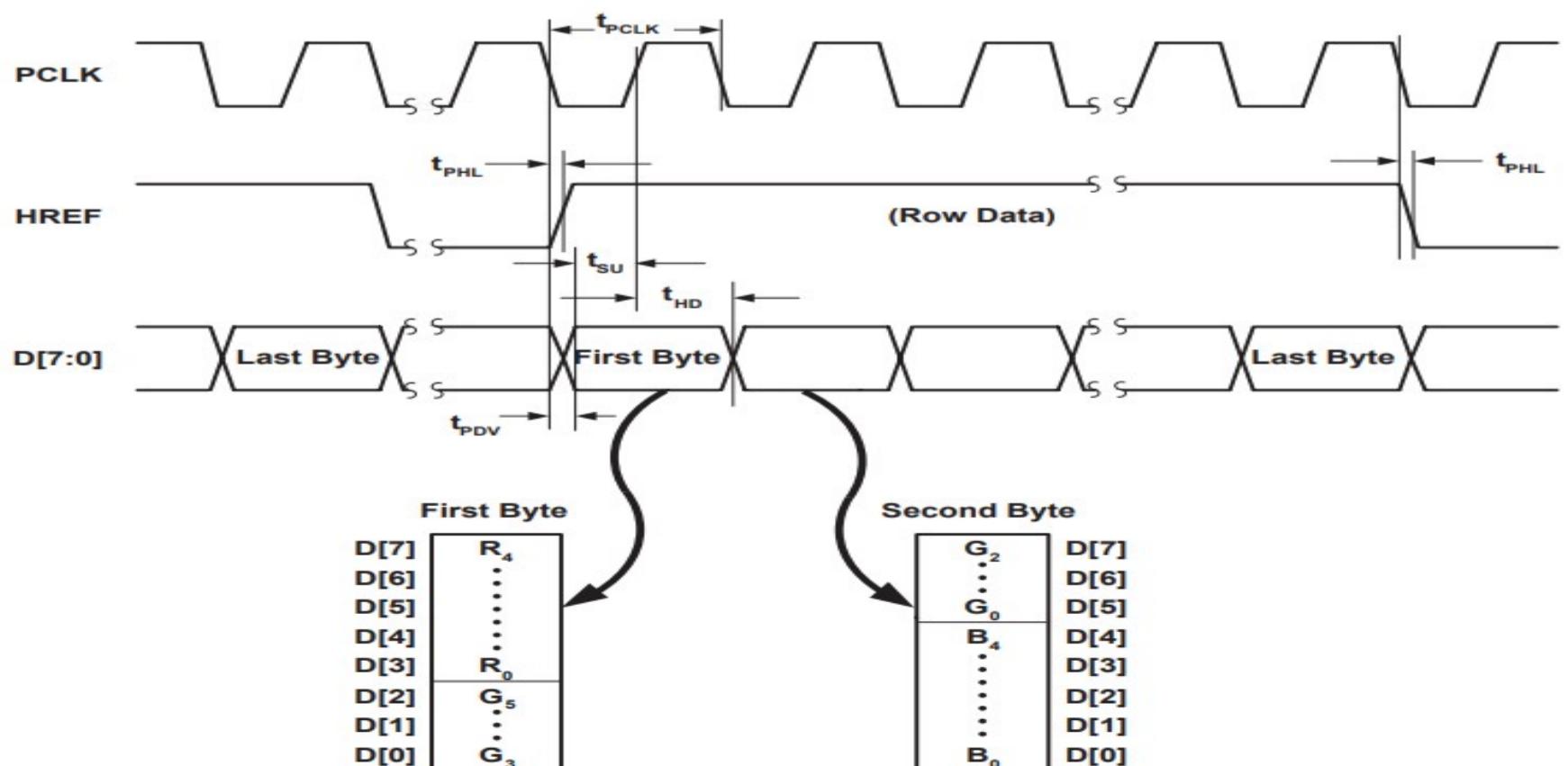


Figure 5 Horizontal Timing

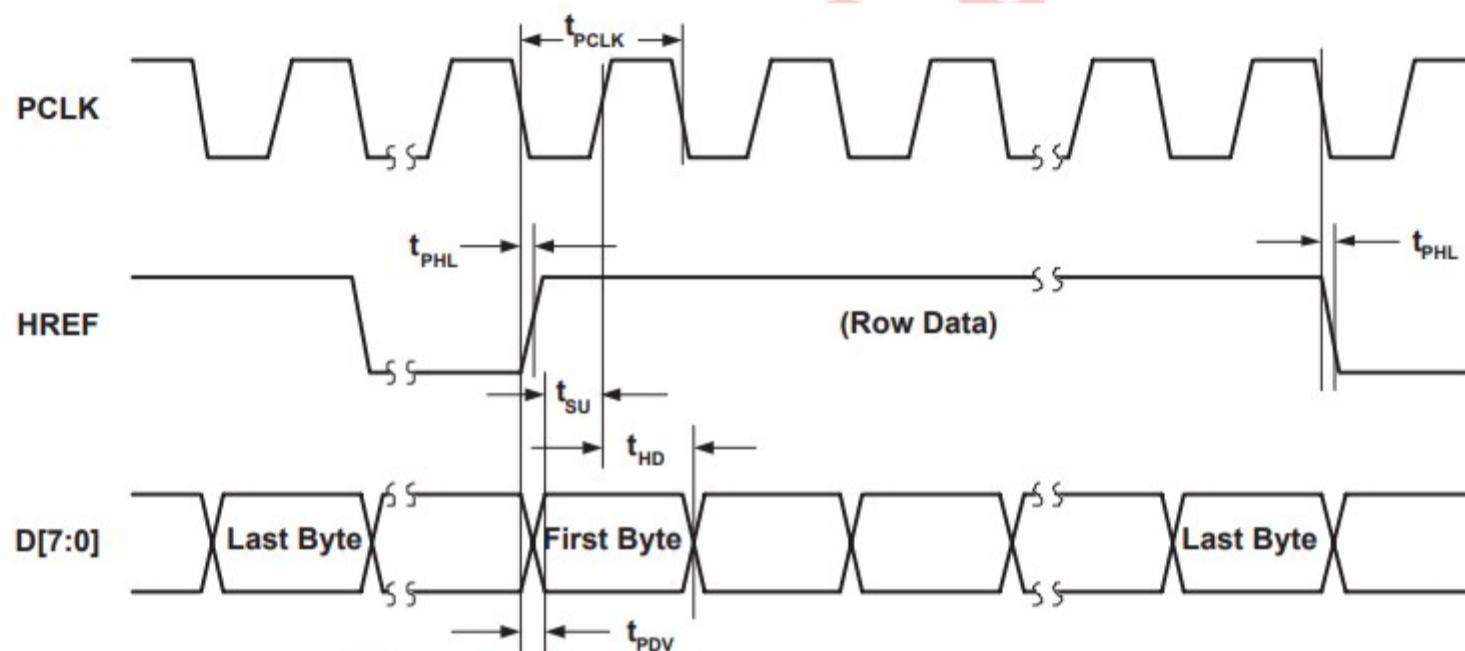
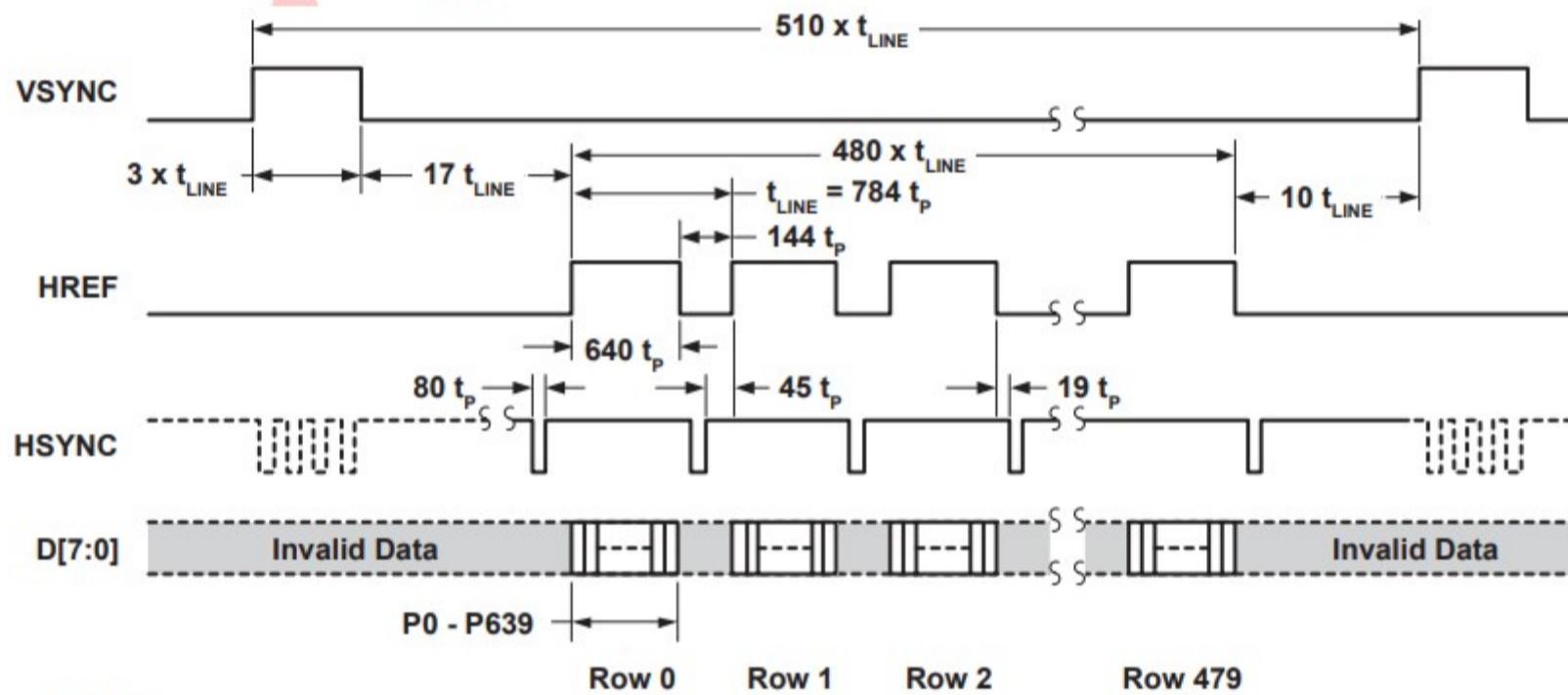


Figure 6 VGA Frame Timing



NOTE:

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

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

Line/Pixel Timing

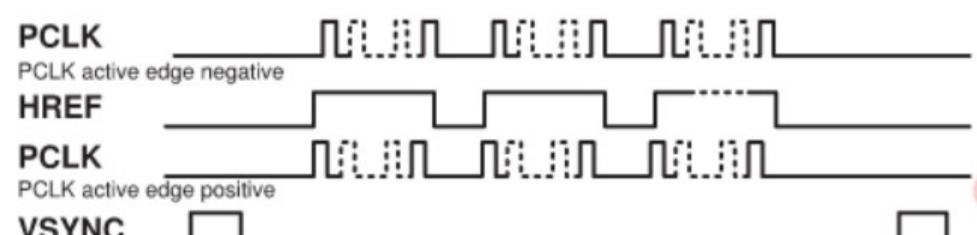
Ov2640 Camera PDF

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, T_A = 25°C, sensor working at 15 fps, external loading = 20 pF.

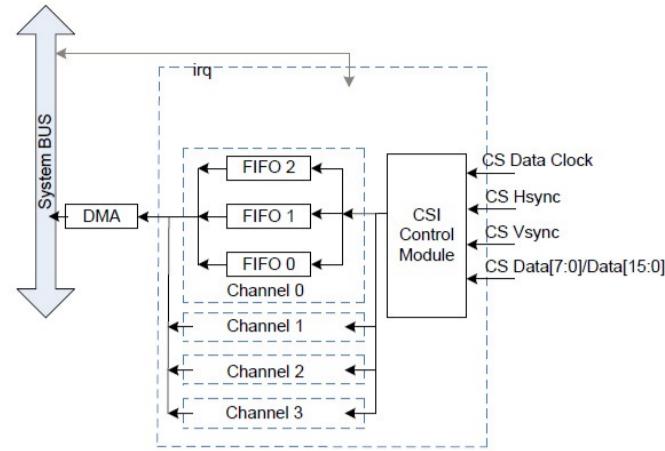
4.1. CSI0

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);
writel(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);
```

4.3 CSI Clock information: default is 24MHz = OSC24M = 0b000

4.3.1 A20 datasheet information.

1.5.3. CCU Register List

Module Name	Base Address	
CCU	0x01C20000	
Register Name	Offset	Description
PLL3_CFG_REG	0x0010	PLL3 CONTROL
PLL4_CFG_REG	0x0018	PLL4 CONTROL
PLL5_CFG_REG	0x0020	PLL5 CONTROL
PLL5_TUN_REG	0x0024	PLL5 TUNING
PLL6_CFG_REG	0x0028	PLL6 CONTROL
PLL6_TUN_REG	0x002C	PLL6 TUNING
PLL7_CFG_REG	0x0030	PLL7 CONTROL
/	0x0034	/
CSI_SCLK_REG	0x0120	CSI SPECIAL CLOCK REGISTER
CSI0_CLK_REG	0x0134	CSI0 CLOCK REGISTER
CSI1_CLK_REG	0x0138	CSI1 CLOCK REGISTER

1.5.4.55. CSI 0 CLOCK(DEFAULT: 0X00000000)

1.5.4.11. PLL7-VIDEO 1(DEFAULT: 0X0010D063)

Offset: 0x30		Register Name: PLL7_CFG_REG	
Bit	Read/Write	Default/Hex	Description
31	R/W	0x0	PLL7_Enable. 0: Disable, 1: Enable. In the integer mode, The PLL7 output=3MHz*M. In the fractional mode, the PLL7 output is select by bit 14. The PLL7 output range is 27MHz~381MHz.
30:16	/	/	/.
15	R/W	0x1	PLL7_MODE_SEL. PLL7 mode select. 0: fractional mode, 1: integer mode.
14	R/W	0x1	PLL7_FRAC_SET. PLL7 fractional setting. 0: 270MHz, 1: 297MHz.
13:7	/	/	/.
6:0	R/W	0x63	PLL7_FACTOR_M. PLL7 Factor M. The range is from 9 to 127.

Offset: 0x134			Register Name: CSI0_CLK_REG
Bit	Read/Write	Default/Hex	Description
31	R/W	0x0	SCLK_GATING. Gating Special Clock 0: Clock is OFF 1: Clock is ON This special clock = Clock Source/Divider M.
30	R/W	0x0	CSI0_RST. CSI0 Reset. 0: reset valid, 1: reset invalid.
29:27	/	/	/
26:24	R/W	0x0	CLK_SRC_SEL. Clock Source Select 000: OSC24M 001: PLL3(1X) 010: PLL7(1X) 011: / 100: / 101: PLL3(2X) 110: PLL7(2X) 111: /
23:5	/	/	/
4:0	/	/	CLK_DIV_RATIO_M. Clock divide ratio (m) The pre-divided clock is divided by (m+1). The divider is from 1 to 32.

4.3.2 Linux kernel / driver programming / setup of PCLK frequency:

```
csi0: csi@1c09000 {
    compatible = "allwinner,sun7i-a20-csi0";
    reg = <0x01c09000 0x1000>;
    interrupts = <GIC_SPI 42 IRQ_TYPE_LEVEL_HIGH>;
    clocks = <&ccu CLK_AHB_CSI0>, <&ccu CLK_CSI_SCLK>, <&ccu CLK_DRAM_CSI0>, <&ccu CLK_CSI0>;
    clock-names = "bus", "isp", "ram", "mclk";
    resets = <&ccu RST_CSI0>;
    status = "disabled";
};

#define CLK_CSI_SCLK      151 [CLK_CSI_SCLK]          = &csi_sclk_clk.common.hw,
#define CLK_CSI0          158 [CLK_CSI0]           = &csi0_clk.common.hw,
#define CLK_CSI1          159 [CLK_CSI1]           = &csi1_clk.common.hw,
Added name for PLL7:
#define CLK_PLL_VIDEO01   17 ]]

diff --git a/include/dt-bindings/clock/sun4i-a10-ccu.h
ck/sun4i-a10-ccu.h
index e4fa61be5..1b149973e 100644
--- a/include/dt-bindings/clock/sun4i-a10-ccu.h
+++ b/include/dt-bindings/clock/sun4i-a10-ccu.h
@@ -44,6 +44,7 @@
#define CLK_HOSC          1
#define CLK_PLL_VIDEO00_2X 9
+#define CLK_PLL7_VIDEO01 17
#define CLK_PLL_VIDEO01_2X 18
#define CLK_CPU            20
```

Not MCLK relevant “isp” clock setup:

```
static const char *const csi_sclk_parents[] = { "pll-video0", "pll-ve",
                                                "pll-ddr-other", "pll-periph" };

static SUNXI_CCU_M_WITH_MUX_GATE(csi_sclk_clk, "csi-sclk",
                                  csi_sclk_parents,
                                  0x120, 0, 4, 24, 2, BIT(31), 0);
```

MCLK relevant:

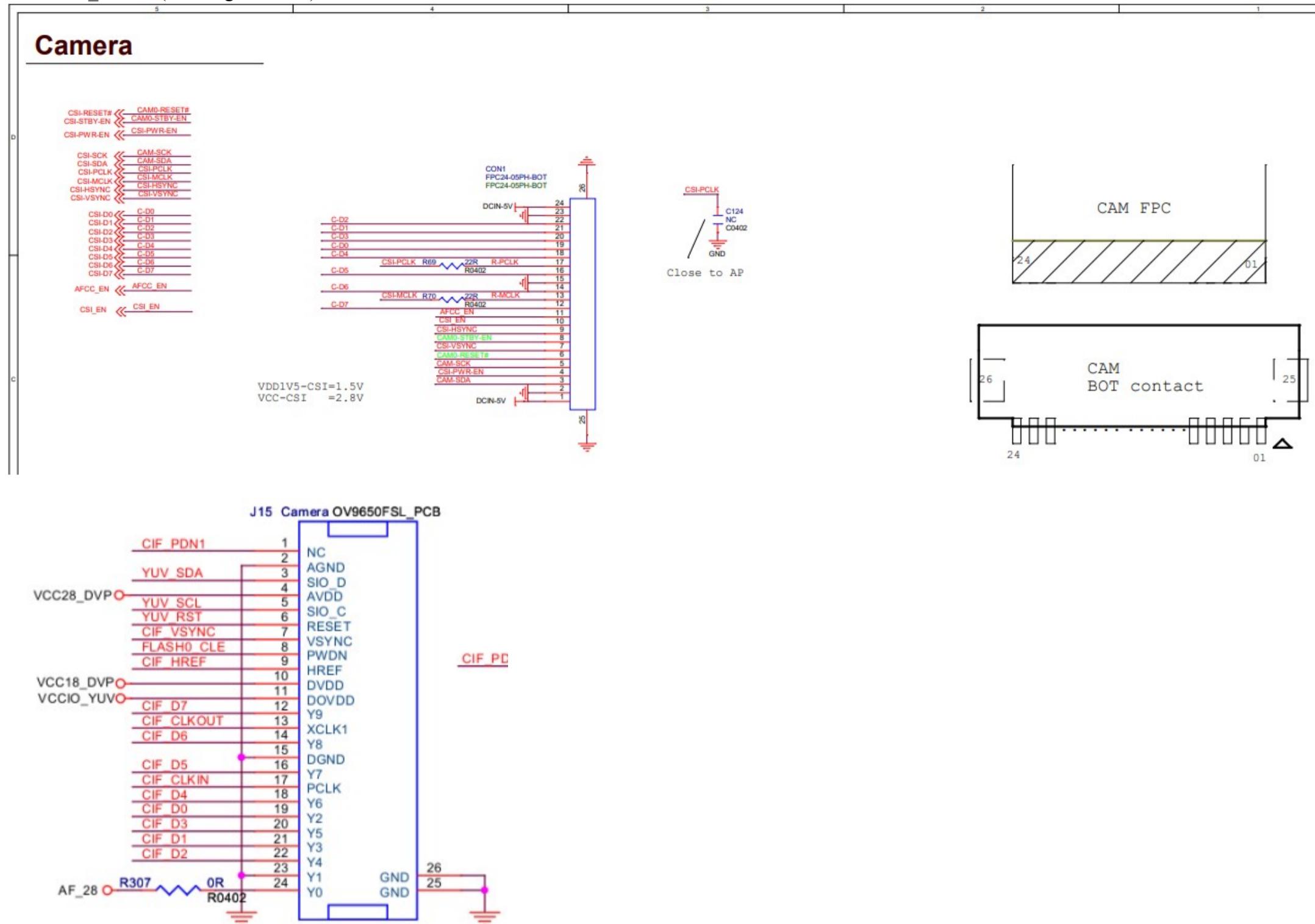
```
static const char *const csi_parents[] = { "hosc", "pll-video0", "pll-video1",
                                           "pll-video0-2x", "pll-video1-2x"};
static const u8 csi_table[] = { 0, 1, 2, 5, 6};
static SUNXI_CCU_M_WITH_MUX_TABLE_GATE(csi0_clk, "csi0",
                                       csi_parents, csi_table,
                                       0x134, 0, 5, 24, 3, BIT(31), 0);
```

This is PLL7 setup:

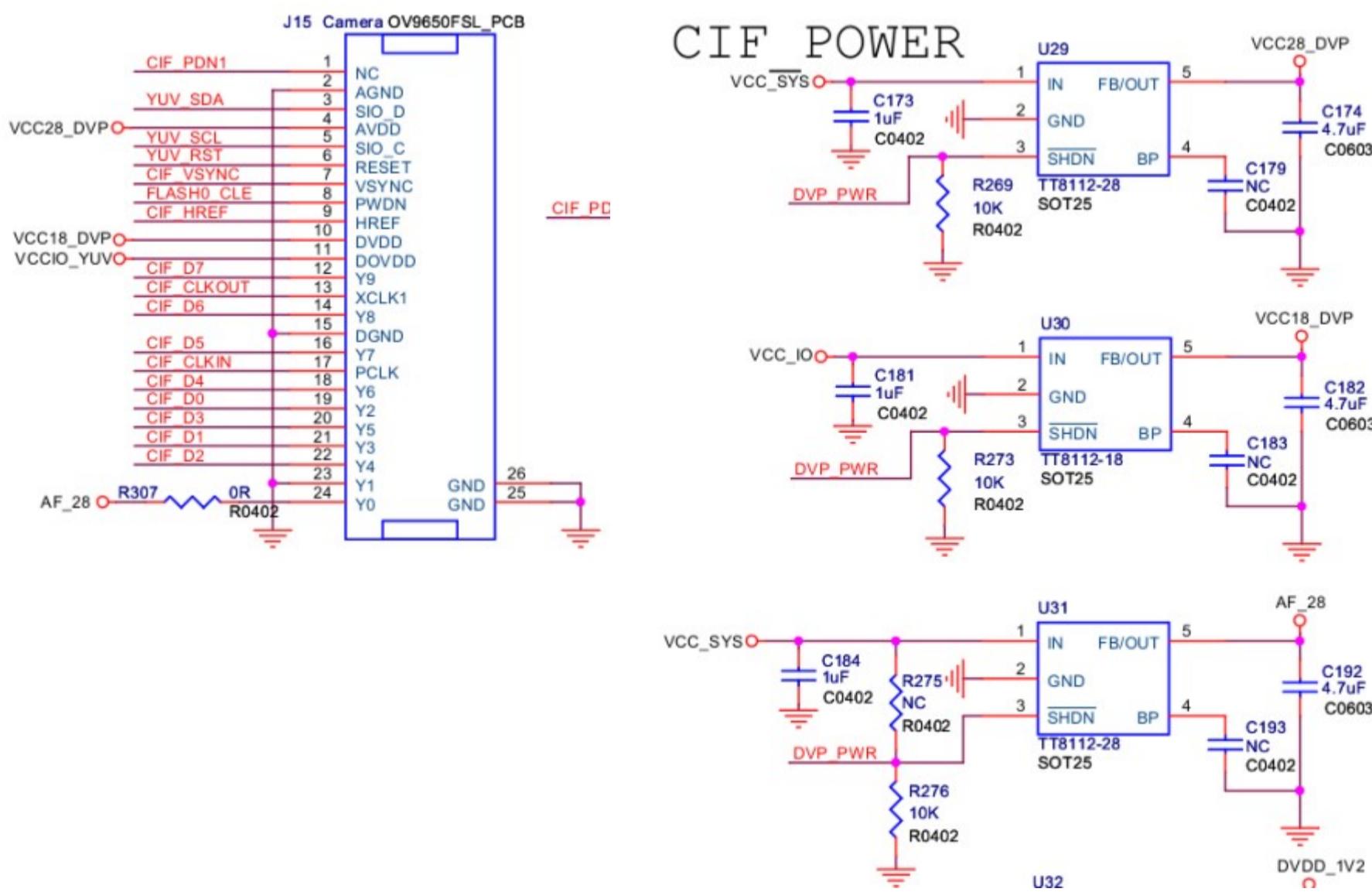
```
static struct ccu_mult pll_video1_clk = {
    .enable     = BIT(31),
    .mult       = _SUNXI_CCU_MULT_OFFSET_MIN_MAX(0, 7, 0, 9, 127),
    .frac       = _SUNXI_CCU_FRAC(BIT(15), BIT(14),
                                270000000, 297000000),
    .common     = {
        .reg       = 0x030,
        .features  = (CCU_FEATURE_FRACTIONAL |
                      CCU_FEATURE_ALL_PREDIV),
        .prediv    = 8,
        .hw.init   = CLK_HW_INIT("pll-video1",
                               "hosc",
                               &ccu_mult_ops,
                               0),
    },
};
```

To change PCLK frequency this way can be used to change file sun7i-a20-olinuxino-lime2.dts file: ov5640: camera@3c {
compatible = "ovti,ov5640";
reg = <0x3c>;
pinctrl-names = "default";
pinctrl-0 = <&csi0_clk_pin>;
clocks = <&ccu CLK_CSI0>;
clock-names = "xclk";
/* modify default clock sources / frequencies */
assigned-clocks = <&ccu CLK_CSI0>, <&ccu CLK_PLL7_VIDEO01>;
/* select CSI0 source clock PLL7 (=17) in CSI0_CLK_REG */
assigned-clock-parents = <&ccu CLK_PLL7_VIDEO01>;
/*
* the integer mode of PLL7: (3 MHz * (9 + 7)) = 48 MHz
* set CSI0 clock in CSI0_CLK_REG as division by (2 + 1) of the
source: 48 MHz / 3 = 16 MHz
*/
assigned-clock-rates = <16000000>, <48000000>;
/* default main oscillator setup: 24000000 Hz
assigned-clocks = <&ccu CLK_CSI0>, <&ccu CLK_HOSC>;
assigned-clock-parents = <&ccu CLK_HOSC>;
*/
assigned-clock-rates = <16500000>;

ORANGE_PI-One (Xunlong Software):



DVP camera interface on the Firefly-RK3288-Reload development board:

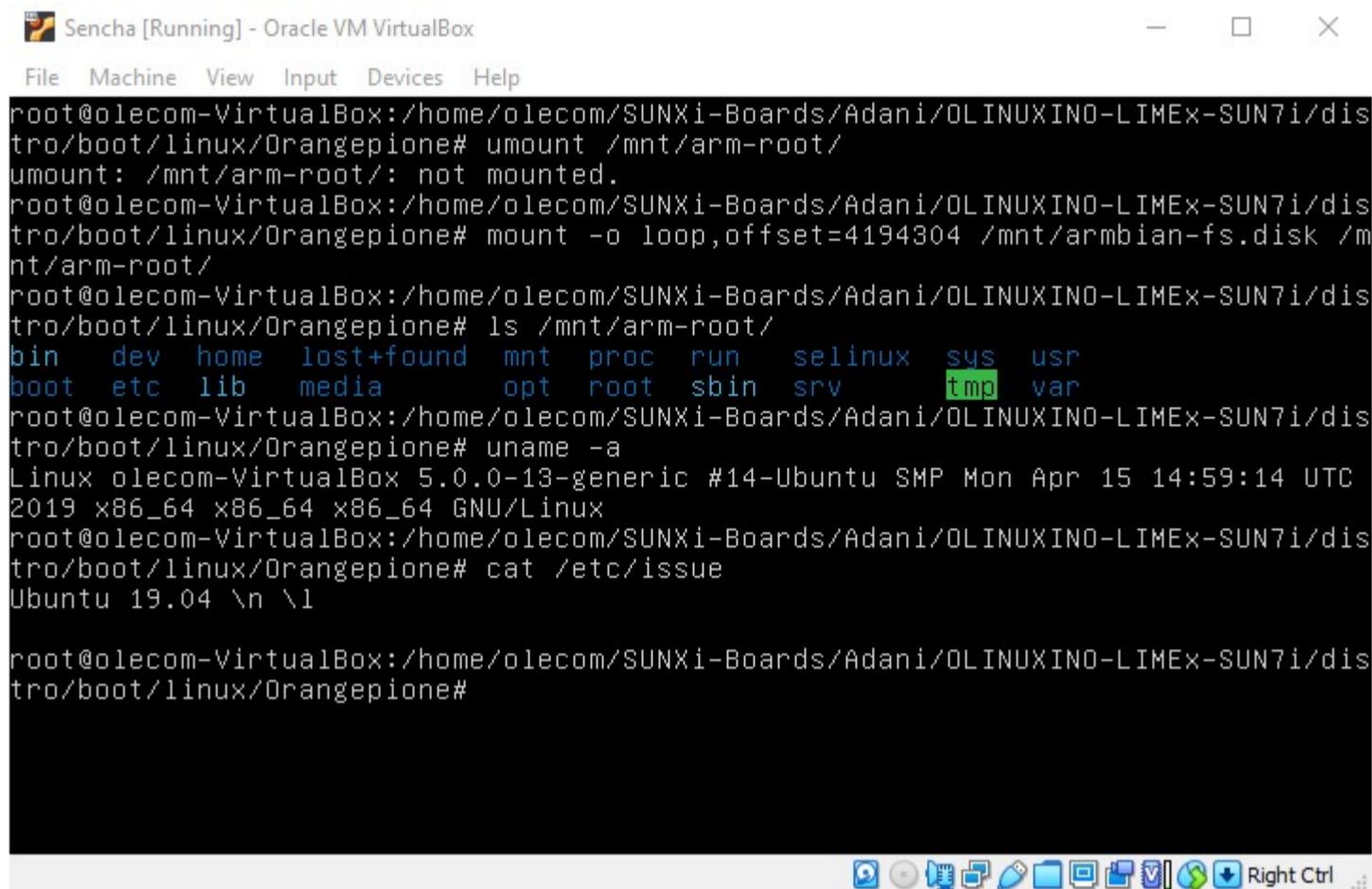


6. qemu

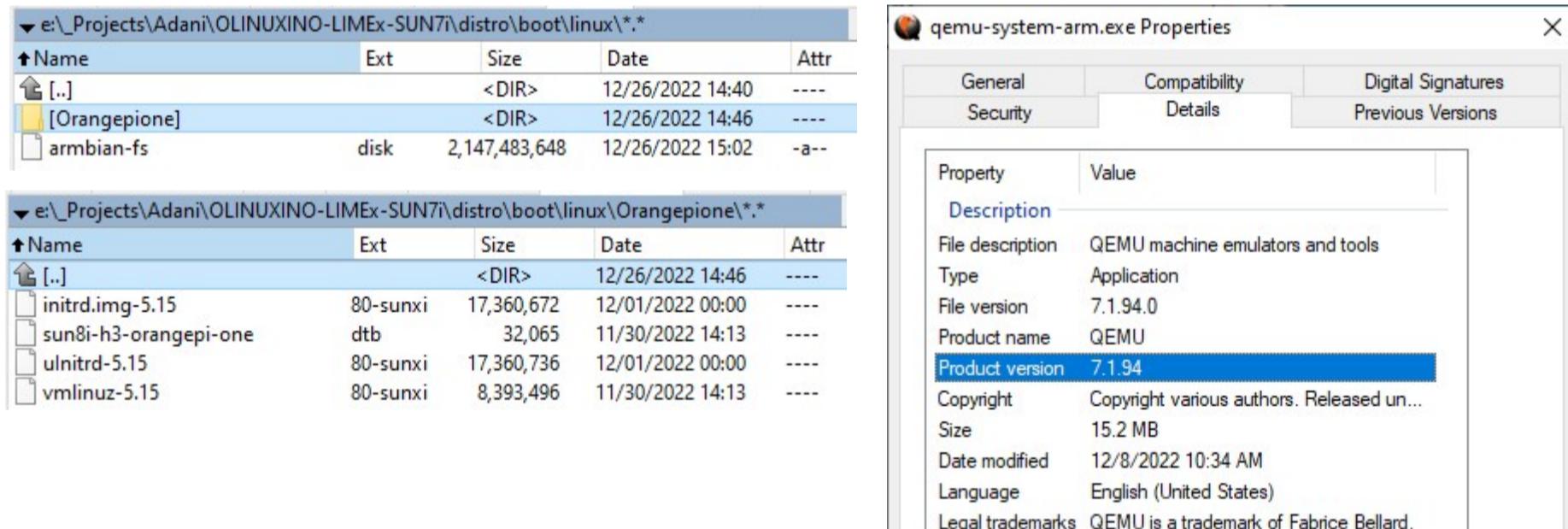
MS Windows 10 + qemu + Armbian for orangepi-pc. Setup description: <https://gist.github.com/wuhanstudio/e9b37b07312a52ceb5973aacf580c453>

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.

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

orangepineo 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@orangepineo:~#
```

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
```

