

A20-OLINUXINO-LIME
and
A20-OLINUXINO-LIME-4GB

Open-source single-board Android/Linux mini-computer

USER'S MANUAL

Revision I, May 2015
Designed by OLIMEX Ltd, 2015



All boards produced by Olimex LTD are ROHS compliant

DISCLAIMER

© 2015 Olimex Ltd. Olimex®, logo and combinations thereof, are registered trademarks of Olimex Ltd. Other product names may be trademarks of others and the rights belong to their respective owners.

The information in this document is provided in connection with Olimex products. No license, express or implied or otherwise, to any intellectual property right is granted by this document or in connection with the sale of Olimex products.

This work is licensed under the Creative Commons Attribution-ShareAlike 3.0 Unported License. To view a copy of this license, visit <http://www.creativecommons.org/licenses/by-sa/3.0/>.



This hardware design by Olimex LTD is licensed under a Creative Commons Attribution-ShareAlike 3.0 Unported License.

The software is released under GPL.

It is possible that the pictures in this manual differ from the latest revision of the board.

The product described in this document is subject to continuous development and improvements. All particulars of the product and its use contained in this document are given by OLIMEX in good faith. However all warranties implied or expressed including but not limited to implied warranties of merchantability or fitness for purpose are excluded. This document is intended only to assist the reader in the use of the product. OLIMEX Ltd. shall not be liable for any loss or damage arising from the use of any information in this document or any error or omission in such information or any incorrect use of the product.

This evaluation board/kit is intended for use for engineering development, demonstration, or evaluation purposes only and is not considered by OLIMEX to be a finished end-product fit for general consumer use. Persons handling the product must have electronics training and observe good engineering practice standards. As such, the goods being provided are not intended to be complete in terms of required design-, marketing-, and/or manufacturing-related protective considerations, including product safety and environmental measures typically found in end products that incorporate such semiconductor components or circuit boards.

Olimex currently deals with a variety of customers for products, and therefore our arrangement with the user is not exclusive. Olimex assumes no liability for applications assistance, customer product design, software performance, or infringement of patents or services described herein.

THERE IS NO WARRANTY FOR THE DESIGN MATERIALS AND THE COMPONENTS USED TO CREATE A20-OLINUXINO-LIME AND A20-OLINUXINO-LIME-4GB. THEY ARE CONSIDERED SUITABLE ONLY FOR, RESPECTIVELY, A20-OLINUXINO-LIME OR A20-OLINUXINO-LIME-4GB.

Table of Contents

<u>DISCLAIMER.....</u>	<u>2</u>
<u>CHAPTER 1: OVERVIEW.....</u>	<u>5</u>
<u>1. Introduction to the chapter.....</u>	<u>5</u>
<u>1.1 Features.....</u>	<u>5</u>
<u>1.2 Target market and purpose of the board.....</u>	<u>6</u>
<u>1.3 Board variants.....</u>	<u>6</u>
<u>1.4 Board version used in the manual.....</u>	<u>7</u>
<u>1.5 Document organization.....</u>	<u>7</u>
<u>CHAPTER 2: SETTING UP THE OLINUXINO BOARD.....</u>	<u>8</u>
<u>2. Introduction to the chapter.....</u>	<u>8</u>
<u>2.1 Electrostatic and electrical polarity warning.....</u>	<u>8</u>
<u>2.2 Requirements.....</u>	<u>9</u>
<u>2.3 Powering the board.....</u>	<u>10</u>
<u>2.4 Button functions.....</u>	<u>11</u>
<u>2.5 Interacting with the board.....</u>	<u>11</u>
<u>2.5.1 Serial connection at UART0.....</u>	<u>12</u>
<u>2.5.2 HDMI monitor.....</u>	<u>12</u>
<u>2.5.3 LCD display.....</u>	<u>13</u>
<u>2.5.4 SSH via mini USB cable in Debian.....</u>	<u>13</u>
<u>2.5.5 SSH via Ethernet connector.....</u>	<u>13</u>
<u>2.6 Changing the default image resolution.....</u>	<u>13</u>
<u>2.7 Connecting and calibrating a display.....</u>	<u>16</u>
<u>2.7.1 Android calibration.....</u>	<u>16</u>
<u>2.7.2 Debian calibration.....</u>	<u>16</u>
<u>2.8 GPIO under Debian.....</u>	<u>17</u>
<u>2.9 I2C and SPI under Debian.....</u>	<u>18</u>
<u>2.10 Software support.....</u>	<u>18</u>
<u>CHAPTER 3: A20-OLINUXINO-LIME BOARD DESCRIPTION.....</u>	<u>20</u>
<u>3. Introduction to the chapter.....</u>	<u>20</u>
<u>3.1 Layout (top view).....</u>	<u>20</u>
<u>3.2 Layout (bottom view).....</u>	<u>21</u>
<u>CHAPTER 4: THE ALLWINNER A20 MICROCONTROLLER.....</u>	<u>22</u>
<u>4. Introduction to the chapter.....</u>	<u>22</u>
<u>4.1 The processor.....</u>	<u>22</u>
<u>4.2 Block diagram.....</u>	<u>23</u>
<u>CHAPTER 5: CONTROL CIRCUITY.....</u>	<u>24</u>
<u>5. Introduction to the chapter.....</u>	<u>24</u>
<u>5.1 Reset.....</u>	<u>24</u>
<u>5.2 Clocks.....</u>	<u>24</u>
<u>5.3 Power supply circuit.....</u>	<u>24</u>

CHAPTER 6: CONNECTORS AND PINOUT.....	25
6. Introduction to the chapter.....	25
6.1 UART0 interface.....	25
6.2 MicroSD card connector.....	25
6.2.1 SD/MMC1 slot.....	26
6.3 PWR jack.....	27
6.4 USB OTG.....	27
6.5 USB HOST connectors.....	30
6.6 Ethernet.....	30
6.7 HDMI connector.....	33
6.8 SATA connector and power.....	34
6.9 GPIO connectors.....	35
6.9.1 GPIO-1 (General Purpose Input/Output) 40pin connector.....	35
6.9.2 GPIO-2 (General Purpose Input/Output) 40pin connector.....	37
6.9.3 GPIO-3 (General Purpose Input/Output) 40pin connector.....	38
6.9.4 GPIO-4 (General Purpose Input/Output) 20pin connector.....	38
6.10 LCD CON 40pin connector.....	39
6.11 Jumper description.....	40
6.12 Additional hardware components.....	41
CHAPTER 7: SCHEMATICS.....	42
7. Introduction to the chapter.....	42
7.1 Eagle schematic.....	42
7.2 Physical dimensions.....	43
CHAPTER 8: REVISION HISTORY AND SUPPORT.....	44
8. Introduction to the chapter.....	44
8.1 Document revision.....	44
8.2 Board revision.....	44
8.3 Useful web links and purchase codes.....	45
8.4 Frequently asked questions.....	46
8.5 Product support.....	49

CHAPTER 1: OVERVIEW

1. Introduction to the chapter

Thank you for choosing this OLinuXino single board computer from Olimex! This document provides a user's guide for the Olimex A20 LIME board. As an overview, this chapter gives the scope of this document and lists the board's features. The document's organization is then detailed.

The A20-OLinuXino-LIME development board enables code development of applications running on the A20 microcontroller, manufactured by Allwinner Technology from China.

OLinuXino is an open-source, open-hardware project and all documentation is available to the customer.

1.1 Features

The board has the following set of features:

- Allwinner A20 dual core Cortex-A7 processor, each core typically running at 1GHz and dual-core Mali 400 GPU
- 512MB DDR3 RAM memory
- **4GB NAND FLASH memory (available only on the 4GB version of the board)**
- **Android already loaded on the NAND (available only on the 4GB version of the board)**
- SATA connector with 5V SATA power jack
- Capable of FullHD (1080p) video playback
- Native HDMI connector
- 2 x USB High-speed host with power control and current limiter
- USB-OTG with power control and current limiter
- 100MBit native Ethernet with connector
- Battery connector with battery-charging capabilities
- LCD row of pins (0.05" step)
- 160 GPIOs on four GPIO rows of pins (0.05" step)
- MicroSD card connector
- DEBUG-UART connector for console debug with USB-SERIAL-CABLE-F
- GPIO LED
- Battery charge status LED
- 5V input power supply, noise immune design
- Power LED
- 2KB EEPROM for MAC address storage and more
- 3 BUTTONS with ANDROID functionality + RESET button
- 4 mount holes
- PCB dimensions: (3005×2345) mils ~ (76×59) mm

1.2 Target market and purpose of the board

The boards from the OLinuXino family are easy to setup and powerful. It is possible to use them in almost any application as a host board. They are suitable for embedded programming enthusiasts, Linux and Android gadget fans (they can just use the board as a media center or fully functional Linux-PC, for instance) and also professionals (since its low cost makes it very good solution for application-orientated embedded systems). The main usage of the board is software embedded development without the urge of understanding perfectly the hardware.

The LIME boards are a special kind of OLinuXino boards – they are as small as possible while maintaining a remarkable stand-alone functionality. The strong points of these boards are the processor speed, the small form factor and the low price-to-productivity ratio.

Customers have full access to the technical documentation of the board. The software is released under General Purpose License and the board is considered open-hardware – all schematics and board design files are available to the customer under the Creative Commons Attribution-ShareAlike 3.0 Unported License.

1.3 Board variants

There are two major board variants named: A20-OLinuXino-LIME and A20-OLinuXino-LIME-4GB. The 4GB version has a built-in NAND memory that allows the storage of an operating system without the need of a SD card. The 4GB version comes with already programmed Android image.

There is a more expensive board variant called A20-OLinuXino-LIME2 – it looks similar to A20-OLinuXino-LIME but there are four major differences:

1. LIME2 has double the amount of DDR3 RAM memory (1GB vs 512MB)
2. LIME2 has a native GbE interface (1000BASE-T vs 100BASE-TX)
3. LIME2 has a different LCD_CON layout
4. LIME2 is more expensive

The A10-OLinuXino-LIME and A10-OLinuXino-LIME-4GB boards are almost identical to, respectively, A20-OLinuXino-LIME and A20-OLinuXino-LIME-4GB with the only difference being the processor mounted. This brings different processor speed and productivity. The A10 has a single Cortex A8 core. The A20 has a dual-core Cortex A7. This makes the A20 boards more powerful and suitable for heavier computational tasks, but more power-hungry. The processor swap is possible since the A10 and A20 processors are pin-to-pin compatible.

1.4 Board version used in the manual

Revision C boards and resources were used while writing this document. It is possible that they are outdated so it is always recommended to download the latest sources from the GitHub page of the board (<https://github.com/OLIMEX/OLINUXINO/tree/master/HARDWARE>).

Please note that A20-OLinuXino-LIME design is almost identical to the design of its predecessor A10-OLinuXino-LIME – the only difference being the processor. Even then the processors are pin-to-pin compatible. Some peripherals, documents or software may be suitable for both the A20 and the A10 LIME boards but it is generally advised to use resources for the specific board.

1.5 Document organization

Each section in this document covers a separate topic, organized as follows:

- Chapter 1 is an overview of the board usage and features
 - Chapter 2 provides a guide for quickly setting up the board and software notes
 - Chapter 3 contains the general board diagram and layout
 - Chapter 4 describes the component that is the heart of the board: the A20 – Allwinner processor
 - Chapter 5 is an explanation of the control circuitry associated with the microcontroller
 - Chapter 6 covers the connector pinout, peripherals and jumper description
 - Chapter 7 provides the schematics and the dimensions of the board
 - Chapter 8 contains the revision history, useful links and support information
-
-

CHAPTER 2: SETTING UP THE OLINUXINO BOARD

2. Introduction to the chapter

This section helps you set up the OLinuXino development board for the first time. Please consider first the electrostatic warning to avoid damaging the board, then discover the hardware and software required to operate the board.

The procedure to power up the board is given, and a description of the default board behavior is detailed.

2.1 Electrostatic and electrical polarity warning

OLinuXino is shipped in a protective anti-static package. The board must not be exposed to high electrostatic potentials. A grounding strap or similar protective device should be worn when handling the board. Avoid touching the component pins or any other metallic element.

Ensure that your development board gets attached to properly working hardware. For example, it is common for cheap HDMI monitors to lack grounding. Avoid TVs which have no grounding on their power supply cable! If you can't avoid them try to add the grounding yourself, if this is not possible please use USB-ISO to save your development board from potential over voltage.

If you connect other electrical devices to the LIME board make sure that they have equal electrical polarity. For example, when you connect an HDMI cable between a TV and the board it is a good idea to have them both connected to the same electrical source (to the same utility power socket). This might be said for a serial cable connected between a PC and the board's DEBUG port. In rare cases different polarity might cause hardware damage to the board.

2.2 Requirements

In order to set up the A20-OLinuXino-LIME optimally one or more additional items may be used. They might be generally placed in three categories:

Required – items that are needed in order to achieve minimum functionality;

Recommended – items that is good to have in order to be able to interact with the most important of the features of the board;

Additional – items that provide access to additional features or expand the features of the board.

Required items:

- USB type A to USB mini cable – to connect to a personal computer; used for powering the board and uploading new Android image to the NAND memory (if your board has 4GB NAND), can also be used to access the Linux of the board
- Input device – either a mouse/keyboard or touchscreen LCD
- Output device – either HDMI cable + native HDMI monitor/screen/projector; or USB-SERIAL-CABLE-F + personal computer (for Linux and/or Android debugging)
- SD card with compatible image – if you have the board version with NO additional NAND memory you will need it to use one of the images available.

Recommended items:

- External USB hub – to split the USB_HOST mounted on the board; you need that to connect more USB devices
- External power supply unit – 5V DC, 5W required – for optimal power
- A10-OLinuXino-LIME-BOX – an aesthetic box specially made for the board that protects it from dust and accidental short-circuits
- USB-SERIAL-CABLE-F – for Android/Linux debugging on UART0

Additional items include:

- Small heatsink for the processor – during long video playbacks the A20 processor might heat up
- External 2.5" SATA hard disk drive (proper cables sold separately)
- Ethernet cable for wired Ethernet

Important note on requirements:

The board works with 2.5" SATA hard disks that can be powered by 5V source from the board! For the 3.5" hard disks you would probably need to provide external 12V supply.

It is a very good idea to have a USB-SERIAL-CABLE or similar cable since it allows you to:

1. Change display output settings for Debian easily
2. Debug the board via any personal computer even if there are problems with the video output

Some of the above-suggested items can be purchased by Olimex, for instance:

SY0605E – reliable power supply adapter 50Hz (for EU) 5V/1A for A20-OLinuXino-LIME

SY0605E-CHINA – cheaper power supply adapter 50Hz (for EU) 5V/1A for A20-OLinuXino-LIME

A20-LIME-ANDROID-SD – a tested class 10 micro SD card with the latest (by the time of leaving the Olimex facilities) official Android release

A20-LIME-DEBIAN-SD – a tested class 10 micro SD card with the latest (by the time of leaving Olimex facilities) official Debian Linux release

USB-SERIAL-CABLE-F – female USB serial console cable

USB-MINI-CABLE – standard USB type A to USB type mini cable

A10-OLINUXINO-LIME-UEXT – GPIO adapter which makes LIME GPIOs available in breadboard friendly 0.1" step connectors, needed if you want to connect an older Olimex-made display to the board (newer displays listed below have 0.05" step connector)

CABLE-40-40-10CM – 0.05" step cable for connecting a LCD_CON to A10-OLINUXINO-LIME-UEXT

LCD-OLINUXINO-4.3TS – low-cost 4.3" LCD display with touchscreen component – 480×272

LCD-OLINUXINO-7TS – low-cost 7" LCD display with optional touchscreen component – 800×480

LCD-OLINUXINO-10TS – low-cost 10" LCD display with optional touchscreen component – 1024×600

LCD-OLINUXINO-15.6 – low-cost 15.6" LCD display – 1366×768

LCD-OLINUXINO-15.6FHD – low-cost Full HD 15.6" LCD display – 1920×1080

CABLE-IDC40-15cm – 0.1" step cable for connecting an LCD to LCD_CON

SATA-HDD-2.5-500GB – 5GB 2.5" SATA hard disk

SATA-CABLE-SET – cables that allow the connection of a 2.5" hard disk to the board

ALUMINIUM-HEATSINK-20×20×6MM – heatsink radiator for better processor heat dissipation

Note that some of the items above carry the A10 name but are suitable for use with the A20 design due to the boards similarity.

2.3 Powering the board

There are three possible ways of powering A20-OLinuCino-LIME – via external supply providing 5V DC at the power jack, from 5V USB port via USB_OTG connector or from 3.7V Li-Po battery via the LIPO_BAT. Note that the board might consume up to 650mA of current at 5V when there are no peripherals connected to the USB hosts. Make sure your power supply is able to provide at least 1A before. Depending on your preferred way of powering you might need additional hardware.

Important: Not all USB ports would be able to provide enough power for the board. Try using another USB port/USB hub or a cable of higher quality.

If you have an LCD display connected to the board or external 2.5" hard disk then the USB will NOT be sufficient source of power.

The best practice is to never count on the mini USB-OTG as a single source of power for the board!

The preferred way of powering the board is via the PWR jack with 5V DC. This would make the board fully powered and able to power all the peripherals connected to it.

Note that when powering the board from the USB_OTG, the current provided might be insufficient to also power a bigger LCD connected to the LCD_con. However, this power option is capable of driving the board when using external display connected to the HDMI connector.

The typical consumption of A20-OLinuCino-LIME is between 400mA and 750mA depending on the current load.

If the board has entered power-down state you can bring it back without restart using the PWR_BUTTON. The PWR_BUTTON is also used to start the board when powered from a Li-Po battery on the battery connector.

Sometimes when starting Android it is possible the board to enter battery save mode even before booting fully. Especially, if you have turned off the board without quick boot mode enabled. In this case you should press the PWR button for at least 5 seconds which would allow the board to start.

For the European customers we sell two power supply adapters, please check chapter 2.2. We also sell USB OTG to USB type A cables if you lack such.

The default username/password combination for the default Linux image on the SD card (if purchased) is: root/olimex.

Note that it is normal that when the board is powered some integrated circuits might appear hotter than others. This is perfectly normal for some chips – for instance – voltage regulators and the main processor.

2.4 Button functions

The bellow three buttons usually are supported under both Android and Debian:

PWR – used to perform software turn off, software turn on; used to turn on board when powered by battery – has to be held down for at least 5 seconds to perform each action

RESET – used for hardware reset of the board

RECOVERY – used to wake up the board from sleep

It is recommended to always make a soft “turn off” of the board. If that is not possible then please hold PWR button down for a few seconds to “turn off the board”. Then you are free to remove the power supply. If you disconnect the power supply (either the USB, the battery or the power jack) before turning off the board you may corrupt your SD card. If you have placed NAND memory you can corrupt the image on the NAND memory.

2.5 Interacting with the board

There are five typical ways to interact with A20-OLinuXino-LIME. You can interact with A20-OLinuXino-LIME and:

- a personal computer or another board via UART0 and a serial debug cable
- an HDMI monitor via the HDMI connector and an HDMI cable
- an LCD display via LCD_CON connector and 0.05" step 40-pin cable
- a personal computer or another board via the mini USB connector and a mini USB cable, using SSH protocol
- via Ethernet via Ethernet connector and LAN cable, using SSH protocol

More details on each of the connections might be found in the consequent sub-chapters.

Note that not all interface options are available for all images. Furthermore, some of the ways of interaction are (obviously) not suitable for Android OS. The official Debian image should give you

the most possible options of interfacing the board!

Using HDMI, LCD_CON or LAN might require additional configurations. Furthermore, it is possible to corrupt the output settings over those interfaces and, thus, lose the output. In such cases, you can always use the serial cable USB-SERIAL-CABLE-F as a reliable way to establish connection to the board.

2.5.1 Serial connection at UART0

The default and recommended way to communicate with the board is using the serial connection available at UART0 connector. You would probably need an USB<->SERIAL cable for such a connection. Even if you intend to use HDMI or LCD it is a good idea to have it as a basis – the video output is not always reliable and if you set wrong display settings you would be unable to recover the software without an alternative connection.

We distribute a ready-to-use cable. Even if you already have such a cable or you decide to purchase it elsewhere it is advisable to check this product page for a reference: [Product page](#)

If you decide to make your own cable you would need to consider that the levels at board's UART0 are in CMOS level (3.3V) and you would need a convertor to bring them to the TTL level of your computer or cable! That is true for the RX and TX also!

The cable probably needs drivers – after successful driver installation a COM port device would show up in the hardware manager of your computer. Then you would need to connect the serial cable wires to the board as follows: RX line to UART1-TX pin; TX line to UART1-RX pin; GND to GND. The last pin of UART1 connector is named 3.3V and it can be used to power the board, if you have other ways of power the board leave 3.3V pin unconnected! Be careful to avoid the connection to the serial cable else there would be a short-circuit. Make sure that the serial cable is connected to your personal computer and recognized properly after driver installation.

Then open a terminal program on the serial (COM) port which the cable is associated with. The settings for the connection are 115200 baud, 8-N-1.

After everything else is set, you would need to power the board as explained in “2.3 Powering the board”.

2.5.2 HDMI monitor

All official Debian and Android images for A20-OLinuCino-LIME have HDMI output enabled by default. The board would work out-of-the-box with a native HDMI monitor.

Make sure to use a tested HDMI cable.

The default HDMI resolution in the official images is 720p60 (1280×720p at 60Hz). In order to change that setting the video output on the LCD display you would need to run a configuration script (if you use Debian Linux) or download a suitable image (if you use Android). Video output settings are hard-coded in Android images but there are ways to edit the images. You might more information about editing pre-built Android images at this wiki article: [how to edit prebuilt Android images](#).

More information about the video output settings and the usage of video settings script might be

found in chapter “2.6 Changing the default image resolution”.

2.5.3 LCD display

Various LCD displays can be attached to A20-OLinuXino-LIME and used to interact with the board.

Depending on the display chosen and whether you attempt to run Debian Linux or Android additional video settings configuration may be required. It is always a good idea to have one of the other connection options available as back-up (in case the video output settings need modification).

For more information refer to chapter “2.7 Connecting and calibrating a display”.

2.5.4 SSH via mini USB cable in Debian

The latest official Debian Linux image allows the use the USB_OTG connector for SSH connection without the need of a LAN cable or a serial cable. You can use a mini USB cable connected between your host PC and the on-board mini USB connector. For connection convenience there is a DHCP server running specifically for USB0 interface. The DHCP server should give IP address to the new USB0 interface of your host PC so you can make SSH connection from your PC to the default board IP address of the USB0 interface – 192.168.2.1.

You can connect to the board using a mini USB cable and an SSH client (if you use Windows you might use "puTTY", for example) at address 192.168.2.1.

For Windows operating system – upon connection, the board should show up in "Windows Device Manager" as "RNDIS Ethernet Gadget". You might be asked to install a driver. The drivers can be found online as "RNDIS driver" (Remote Network Driver Interface Specification). The drivers are provided by Microsoft and they should be available for every Windows distribution – refer to the respective files and articles provided by Microsoft on how to install the required drivers.

2.5.5 SSH via Ethernet connector

SSH protocol allows you to login remotely to the command-line interface of the A20-OLinuXino-LIME board. You would need an active Ethernet connection to the board.

The newest images have a configuration that allows SSH via the Ethernet connector. The static IP address to access the board is 192.168.1.254

2.6 Changing the default image resolution

Depending on the display or the screen you want to use with the A20-OLinuXino-LIME, you might need to apply software changes to the prebuilt Android or Linux image.

The typical OLinuXino user would not need to edit the files, however.

To ease the process of changing the resolution we have compiled a number of Android images for the Android users (with hard-coded video output settings). Alternatively, for Debian Linux users, we have provided a shell script that can be executed in order to set preferred video output and resolution.

For Android that you boot from the NAND memory you would need an image suitable for the specific resolution. Download locations to such images might be found at the wiki article for the A20 board here: <https://www.olimex.com/wiki/A20-OLinuXino-LIME>.

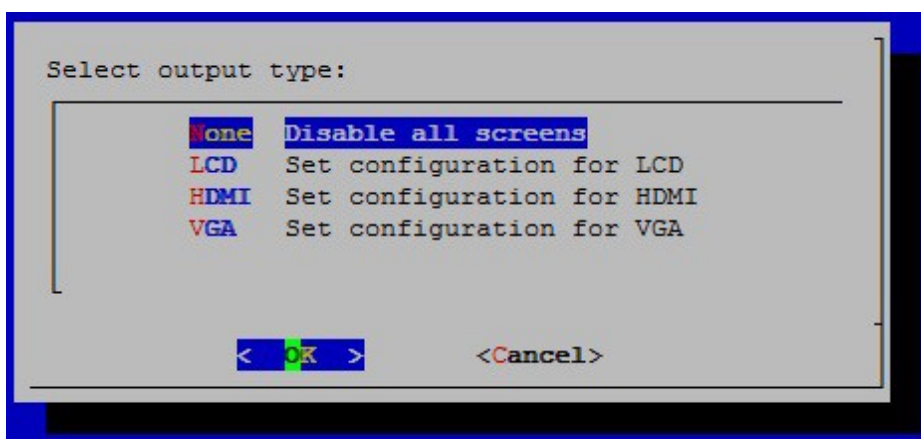
For Linux Debian you would need to execute a shell script to be able to change the resolution. It is very good idea to use a serial cable for connection to the board from a personal computer since in this case you are dependent on a video resolution (a cable like USB-SERIAL-CABLE-F). When the board boots type:

```
./change_display*
```

or

```
./change_display_A20_lime.sh
```

Then the main menu of the video configuration script shows up and you can select the desired interface (LCD, HDMI or VGA):



The supported resolutions are listed below.

For LCD:

1. 4.3" (480×272)
2. 7" (800×480)
3. 10" (1024×600)

For HDMI:

0. 480i
1. 576i
2. 480p
3. 576p
4. 720p50
5. 720p60
6. 1080i50
7. 1080i60
8. 1080p24

9. 1080p50
10. 1080p60

For VGA: (please note that there is no VGA connector and you might need to install additional hardware - more instructions on the VGA signals might be found in the chapter 6):

0. 1680×1050
1. 1440×900
2. 1360×768
3. 1280×1024
4. 1024×768
5. 800×600
6. 640×480
7. 1920×1080
8. 1280×720

Chose the resolution that suits you and reboot the board.

If you decide to edit the configurations yourself the easiest way would be to do it on the board. This can be done offline too (manipulating the image located on the microSD card via a microSD card reader).

The tools for script.bin changing are located in /opt/sunxi-tools directory:

```
#cd /opt/sunxi-tools
#./chscr.sh
```

This will convert script.bin file from sdcad to script.fex file and the file will be opened using nano editor. Now you can change the board modules and parameters, save the changes ("CTRL"+"X"; confirm with "Y") and exit ("CTRL"+"X" again) from nano editor.

```
#./wrscr.sh
```

this will convert script.fex to script.bin and the script.bin file will be written to the microSD card.

```
reboot
```

Reboot the board and the new settings would be applied.

Alternatively, you can do the changes on the microSD card off the board. You would need to remove the microSD card and explore it in a microSD card reader. You would need to edit the configuration file script.bin and edit the settings inside. This file is usually located in Script.bin can't be opened in the binary format so you would need to convert it to .fex file format first. There are ready-to-use tools that convert script.bin <-> script.fex. Note that script.bin/fex contains configuration settings and definitions not only for the video output but also for the pin descriptions and names; power setting and much more. If you really want to modify and customize the default images (to change port functions, port names, to disable specific peripherals) you would need to be able to edit the script files. Please refer to the following web page for more information:

http://linux-sunxi.org/Fex_Guide

2.7 Connecting and calibrating a display

One of the ways to interact with the board is via an external display (with or without touchscreen component). If you want to use a display for video output from the LIME board the best way is to use the LCD_CON row of pins. The step between the pins of the connector is the smaller one – only 0.05" (compared to 0.1" of bigger boards).

Newer displays made by Olimex have both 0.1" and 0.05" step connectors. Going for an LCD output you would also need need and a cable to attach the display to the board. The cable is sold separately.

The cable used for connection depends on the specific board you are using and more specifically it depends on the pitch of the LCD connector of the board. We have two cables – both 40-pins ones but one for the bigger pitch (0.1") and the other for the smaller one (0.05"). Each of the displays listed in the table further below has two connectors suitable for both cables:

CABLE-IDC40-6cm – 6cm long cable suitable for 0.1" step connectors – [Product page](#)

CABLE-40-40-10CM – 10cm long cable suitable for 0.05" step connectors – [Product page](#)

The displays recommended for the board at the moment of writing might be found in the following table:

Display name	Size of display in inches	Native resolution in pixels	Official Debian image support	Official Android image support	Link to product page
LCD-OLinuXino-4.3TS	4.3	480×272	Yes	No	Product page
LCD-OLinuXino-7	7	800×480	Yes	Yes	Product page
LCD-OLinuXino-7TS	7	800×480	Yes	Yes	Product page
LCD-OLinuXino-10	10	1024×600	Yes	Yes	Product page
LCD-OLinuXino-10TS	10	1024×600	Yes	Yes	Product page
LCD-OLinuXino-15.6	15.6	1366×768	Yes	No	Product page
LCD-OLinuXino-15.6FHD	15.6	1920×1080	Yes	No	Product page

The displays whose names contain “TS” - include a resistive touch screen component.

2.7.1 Android calibration

Calibrating a display under Android is pretty straightforward from the Android application.

Important: initially the boards are calibrated for a specific display and resolution. If you re-write the image (no matter whether the SD card or the NAND memory) you might need to use a mouse to calibrate the display initially. It might be impossible to calibrate it via touching the display.

2.7.2 Debian calibration

The command that allows calibrating in Debian Linux is:

```
ts_calibrate
```


The default Debian setup is made with settings for HDMI 720p/60Hz. If you want to change some other LCD, VGA or HDMI resolution then you have to start script file in /root directory.

If the problem is under Debian Linux make sure you are properly logged in the XFCE interface! Else applying calibration would not happen for the current user – if you are calibrating from the X graphical interface make sure that you are logged as user “olimex” (if calibrating without the X, the user is “root”).

```
#su olimex
```

```
enter the password: olimex
```

```
calibrate the touch screen and reboot the board
```

```
#sudo reboot
```

2.8 GPIO under Debian

You can read data from a given GPIO port. The logical ranges are usually as follows:

0V-1V for LOW (or 0)

2.4V-3.3V for HIGH (or 1)

All voltages are measured against ground (GND).

If the input signal is too high, you will at least destroy the port!

The algorithms for writing a value to a GPIO port and reading such a value are pretty similar. The usage of GPIO ports follows the algorithm (we would use GPIO #49 for demonstration purposes):

1. Export GPIO 49:

```
echo 49 > /sys/class/gpio/export
```

Note that you can export GPIOs in range with:

```
for i in `seq 1 1 230`; do echo $i > /sys/class/gpio/export; done
```

2. Set input/output GPIO 49

2.1 Set input:

```
echo "in" > /sys/class/gpio/gpio49_ph9/direction
```

2.2 Set output:

```
echo "out" > /sys/class/gpio/gpio49_ph9/direction
```

3. Set value or read value GPIO 49

3.1 Set value:

```
echo 0 > /sys/class/gpio/gpio49_ph9/value
echo 1 > /sys/class/gpio/gpio49_ph9/value
```

3.2 Read input:

```
cat /sys/class/gpio/gpio49_ph9/value
```

4. Unexport GPIO 49 when finished

```
echo 49 > /sys/class/gpio/unexport
```

A helpful document on A20's GPIO usage might be found here:

<http://www.py6zgp.com/download/A20-GPIO.pdf> – the document was created by Dr. Guido Pelz.

Alternatively, you might also use the Python script mentioned in the next chapter.

2.9 I2C and SPI under Debian

I2C and SPI are both supported in the latest Debian releases. There is a python module called pyA20Lime, which also might be found here: <https://pypi.python.org/pypi/pyA20Lime>

At the same web address you would also find information on how the library is used.

2.10 Software support

We maintain Linux and Android images for SD card which might be downloaded for free and modified as the user wishes. The latest images and updates are featured at the wiki article of the device: <https://www.olimex.com/wiki/A20-OLinuXino-LIME>.

We usually try to provide details on how to build the Linux and the Android images at our wordpress page: <http://olimex.wordpress.com/>. The required files details on building the latest Debian images for Olimex A20 boards might be found at the following GitHub address: <https://github.com/OLIMEX/OLINUXINO/tree/master/SOFTWARE/A20/A20-build>
Another useful place is the Olimex forums where a lot of people share their experience and advice: <https://www.olimex.com/forum/>

The official images are a constant work-in-progress – newer releases are packed with better hardware support, newer kernels and extra features.

You are more than welcome to send or share your suggestions and ideas at our e-mail, the public forums or irc channel. We would attempt to help in almost every case. We listen to the feedback and if the majority of users suggest a software change or update we try to implement such. Customer feedback is very important for the overall state of the software support. However, do not expect full Linux or Android software support.

We can share our experience. We can give you full details for things we have tried. We can point you to a resource or a guide. We can give you general directions to solving a specific problem or

places to look for more information. However, we won't install a piece of software for you or write custom program for you. We won't provide a specific software solution to a specific software problem.

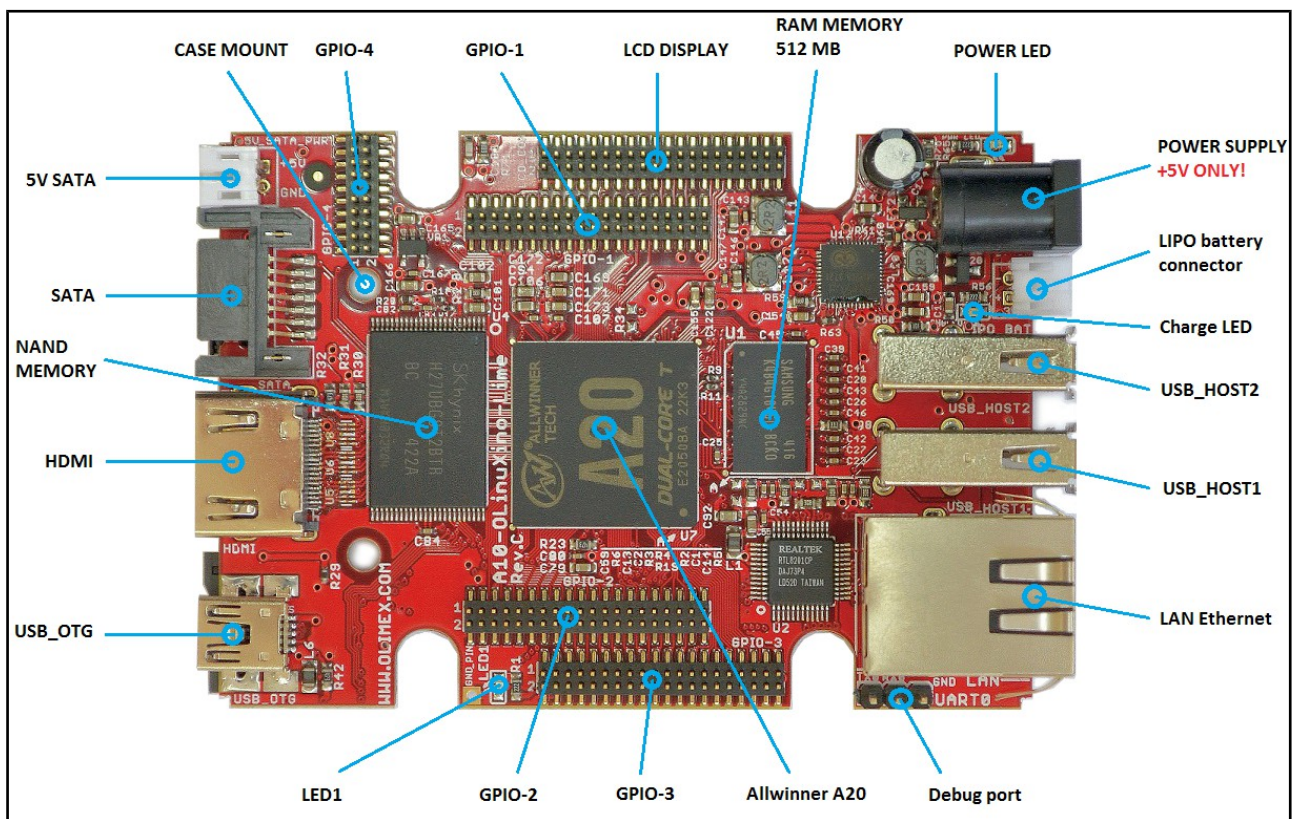
CHAPTER 3: A20-OLINUXINO-LIME BOARD DESCRIPTION

3. Introduction to the chapter

Here you get acquainted with the main parts of the board. Note the names used on the board might differ from the names used below to describe them. For the actual names check the A20-OLinuXino-LIME board itself.

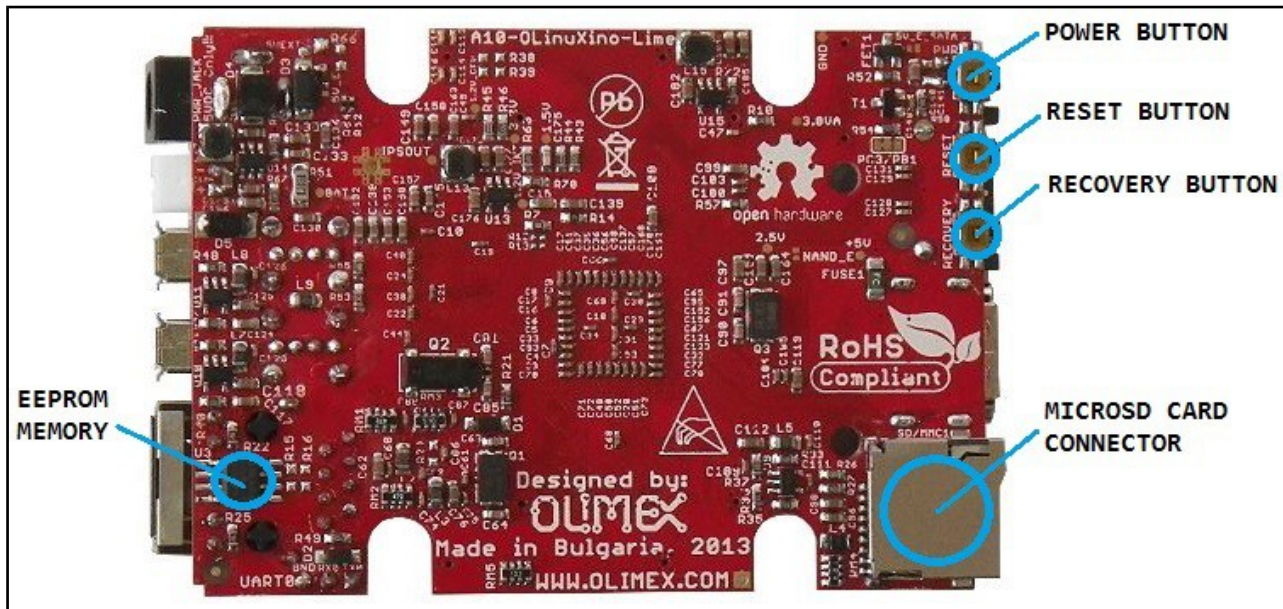
3.1 Layout (top view)

The picture below shows the top side of the board and highlights the most important parts:



3.2 Layout (bottom view)

At the bottom are located mainly the buttons and microSD card connector.



CHAPTER 4: THE ALLWINNER A20 MICROCONTROLLER

4. Introduction to the chapter

In this chapter is located the information about the heart of OLinuXino – its microcontroller. The information is a modified version of the datasheet provided by its manufacturers.

4.1 The processor

The features of the A20 processor according to the manufacturer Allwinner:

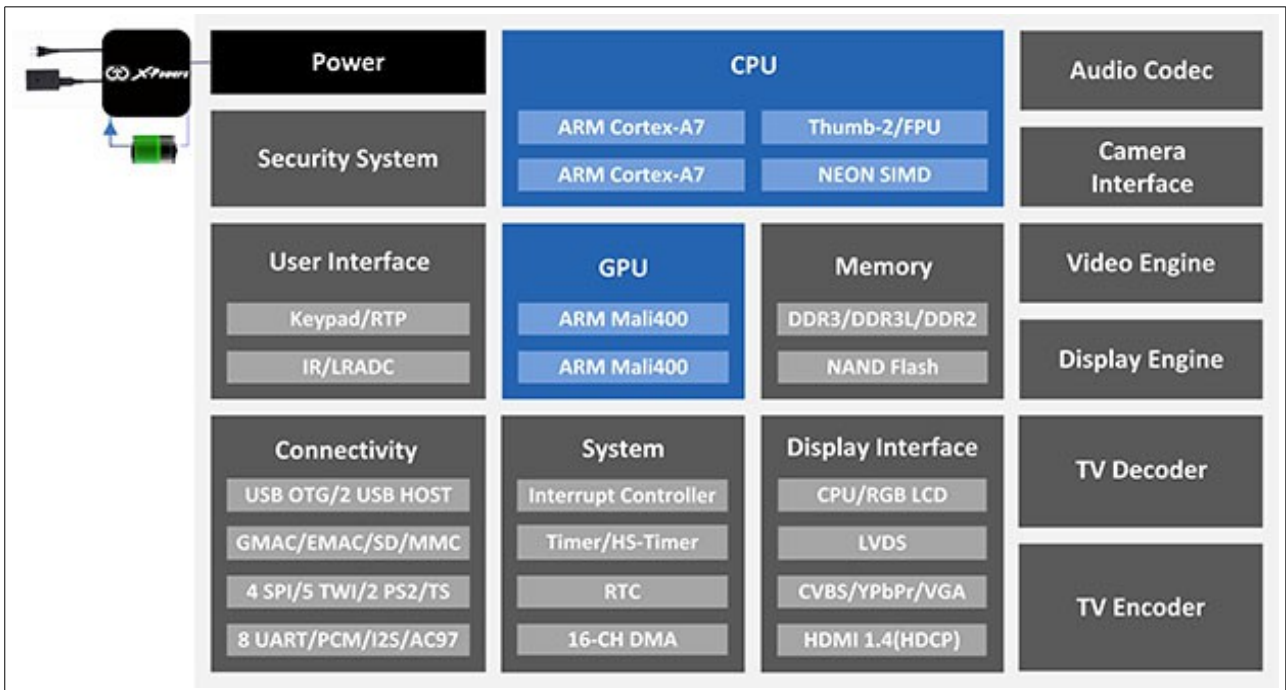
- CPU
 - ARM® Cortex™-A7 Dual-Core
- GPU
 - ARM® Mali400MP2
 - Complies with OpenGL ES 2.0/1.1
- VIDEO
 - HD H.264 2160p video decoding
 - Multi-format FHD video decoding, including Mpeg1/2, Mpeg4 SP/ASP GMC, H.263, H.264,
 - VP6/8, AVS jizun, Jpeg/Mjpeg, etc.
 - H.264 High Profile 1080p@30fps or 720p@60fps encoding
 - 3840×1080@30fps 3D decoding, BD/SBS/TAB/FP supported
 - Complies with RTSP, HTTP, HLS, RTMP, MMS streaming media protocols
- DISPLAY
 - Supports multi-channel HD display
 - Integrated HDMI 1.4 transmitter with HDCP support
 - CPU/RGB/LVDS LCD interface
 - Supports CVBS/YPbPr/VGA
 - Integrated TV decoder
- CAMERA
 - Integrated parallel 8-bit I/F YUV sensor
 - Integrated 24-bit parallel YUV 444 I/F
 - Supports 5M CMOS sensor
 - Supports dual sensors
- MEMORY
 - DDR2/DDR3/DDR3L controller
 - NAND Flash controller with 64-bit ECC
- AUDIO
 - Integrated HI-FI 100dB Audio Codec
 - Dual analog mic amplifiers

More information can be found on Allwinner's web site at the following web-address:
<http://www.allwinnertech.com/en/product/A20.html>

A good source of information on the processor is the linux sunxi wiki article: <http://linux-sunxi.org/A20>.

4.2 Block diagram

The block diagram is taken from Allwinner's web-site.



CHAPTER 5: CONTROL CIRCUITY

5. Introduction to the chapter

Here you can find information about reset circuit and quartz crystals locations, the power supply circuit is also briefly discussed.

5.1 Reset

The board has hardware reset controlled by the AXP209 power system management IC.

It is a good practice to perform software reset of the board. Performing reset by disconnecting the power supply might lead to software corruption of the operating system of choice.

5.2 Clocks

25 MHz quartz crystal Q1 is connected to pins X1 and X2 of the RTL8201CP Ethernet controller.

32 768 Hz (RTC) quartz crystal Q2 is found at pins F1 and F2 of the A20 microcontroller.

24 MHz quartz crystal Q3 is found at pins N22 and N23 of the A20 microcontroller.

5.3 Power supply circuit

The power supply is handled mainly by AXP209 power management system, an Allwinner chip that goes together with the A20 processor. It is mounted on the board but since it is relatively hard to find we also sell it separately.

The power supply circuit of A20-OLinuCino-LIME requires input supply of 5V. The minimum wattage is 2.5W, and this threshold may raise if using a lot of devices on the USB-HOST (via external hub), a lot of GPIOs or you have a display connected to the LCD_CON or a hard disk powered from the 5V_SATA_PWR connector. If a discharged Li-Po battery is connected to the respective connector the power consumption of the whole LIME board might rise enormously, since the built-in battery charger would attempt to recharge it.

For more info on how to power the board refer to chapter “2.3 Powering the board”.

CHAPTER 6: CONNECTORS AND PINOUT

6. Introduction to the chapter

In this chapter are presented the connectors that can be found on the board all together with their pinout and notes about them. Jumpers functions are described. Notes and info on specific peripherals are presented. Notes regarding the interfaces are given.

6.1 UART0 interface

The UART0 interface might be used for serial communication between the board and a personal computer by default. In case of video output problem a cable might provide needed feedback and greatly reduce the efforts needed to repair the board or to adjust the software setting.

Note that by default only UART0 is defined as a port suitable for serial debug. You can use our USB-SERIAL-CABLE-F for debugging.

Consider table below when connecting the USB-SERIAL-CABLE-F according to the wire color code. The RX line of the cable (GREEN wire) should go to TX line of the target board; the TX line of the cable (RED wire) should go to the RX line of the target board. The BLUE wire should go to the target's GND line.

UART0		
Pin #	Signal name	Processor pin
1	UART0-TX	A7
2	UART0-RX	B7
3	GND	-

6.2 MicroSD card connector

The micro SD card slot is primarily used for booting the operating system.

The board works with micro SDHC cards up to 32GB of storage.

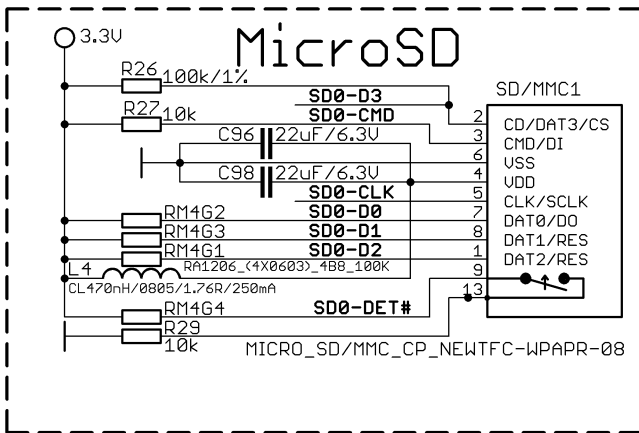
As a general precaution be careful with the SD cards you purchase. There is a big percentage of fake cards due to the low effort required to counterfeit popular brands and the big demand for SD cards worldwide. When in doubt – try the same operation with another card from another brand.

Olimex sells microSD cards with Linux or Android images, that have been tested – please refer to chapter “2.2 Requirements”. Of course, if you already have a large enough microSD card you can download the official Linux image from the wiki pages: <https://www.olimex.com/wiki/A20-OLinuXino-LIME>.

When removing the card, please make sure that you release it from the connector by pushing and NOT by pulling the card directly (this can damage both the connector and the microSD card).

6.2.1 SD/MMC1 slot

The schematic related to the SD/MMC1 (microSD connector) is shown below:



SD/MMC1 slot is the microSD card slot, located on the top of the board.

This slot is typically used for booting the OS, due to the larger capacities of the microSD cards (compared to SD or MMC cards). It is suggested to have an SD card with a proper Linux/Android image especially if you have ordered a version of the board without NAND memory. It is also recommended to use Class 10 (10MByte/sec) card for faster read/write operations, lower class cards (especially higher capacity ones) might slow down the whole system.

SD/MMC1 connector		
Pin #	Connector signal name	Wire name (processor pin)
1	DAT2/RES	SD0-D2 (K19)
2	CD/DAT3/CS	SD0-D3 (K20)
3	CMD/DI	SD0-CMD (L19)
4	VDD	-
5	CLK/SCLK	SD0-CLK(L20)
6	VSS2	-
7	DAT0/DO	SD0-D0(M19)
8	DAT1/RES	SD0-D1(M20)

Additionally, there are the WP and CP switches that are responsible, respectively, for sensing whether the card is locked for reading and whether there is a card inserted.

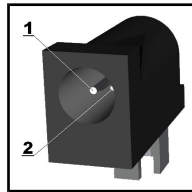
6.3 PWR jack

The power jack used is the typical DC barrel jack one used by Olimex (2.1×6.3×9.2mm) in most of our products. More information about the exact component might be found here:

<https://www.olimex.com/wiki/PWRJACK>

You should provide 5 volts direct current and the required current may vary depending on the peripherals connected to the board. The power supply you use should be capable of providing at least 1A of current.

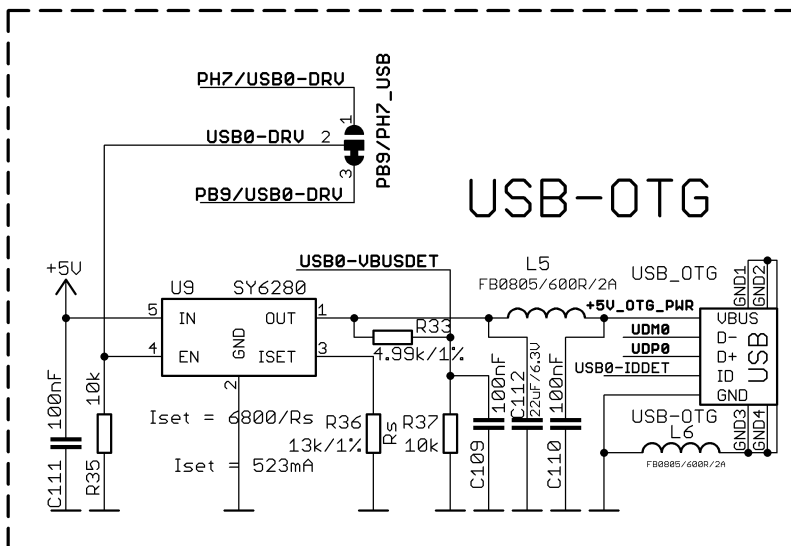
Pin #	Signal name
1	Power input
2	GND



More info about the power supply can be found in chapter 5 of this manual.

6.4 USB_OTG

The main way of changing the firmware image located on the NAND of A20-OLinuXino-LIME-4GB is via the USB-OTG connector – the update of the Android image is explained at the bottom of this sub-chapter. The connector can also be used for establishing SSH connection to the default Debian Linux of LIME boards (for more information about the tethering please refer to “2.5.4 SSH via mini USB cable in Debian”). The part of the schematic related to the USB_OTG is shown below:



The USB_OTG features Low Loss Power Distribution Switch SY6280 which protects the board in case the devices you have plugged to the USB_OTG attempt to draw more current than 523mA combined. The maximum current available on the 5V USB_OTG is exactly 523mA.

The SY6280 responsible for the USB_OTG is enabled by USB0-DRV (processor pin C12, port B9),

thus the USB_OTG is also controlled by the same signal.

Please note that the USB0-DRV (pin C12, port B9) is multiplexed with the I2S_DO1 signal! If you are going to use the I2S audio interface then you would probably need to change the position of the SMT jumper PB9/PH7_USB which by default connects USB_OTG to the PB9. This operation would require cutting between the pads of the default position with a very sharp object and then soldering the pads of the other position together. Doing so you would be able to use both the USB_OTG (via port PH7 this time) and the I2S interface.

Additionally, PB9/PH7_GPIO jumper controls which of the two ports (PB9 or PH7) should be lead out to pin 9 of GPIO3 connector for easier access. By default this jumper is in PH7_GPIO position and PH7 is lead to pin 9 of GPIO3.

USB_OTG connector		
Pin #	Signal name	Processor pin
1	+5V_OTG_PWR	-
2	UDM0	N20
3	UDP0	N21
4	USB0-IDDET	B5
5	GND	-

The connector case is also grounded.

We have configured an Android image with settings suitable for A20-OLinuXino-LIME. Then using PhoenixSuit tools we uploaded the image to the board via the USB OTG. The image is available for users to try and tweak the settings. The images can be downloaded from the wiki article at: <https://www.olimex.com/wiki/A20-OLinuXino-LIME>.

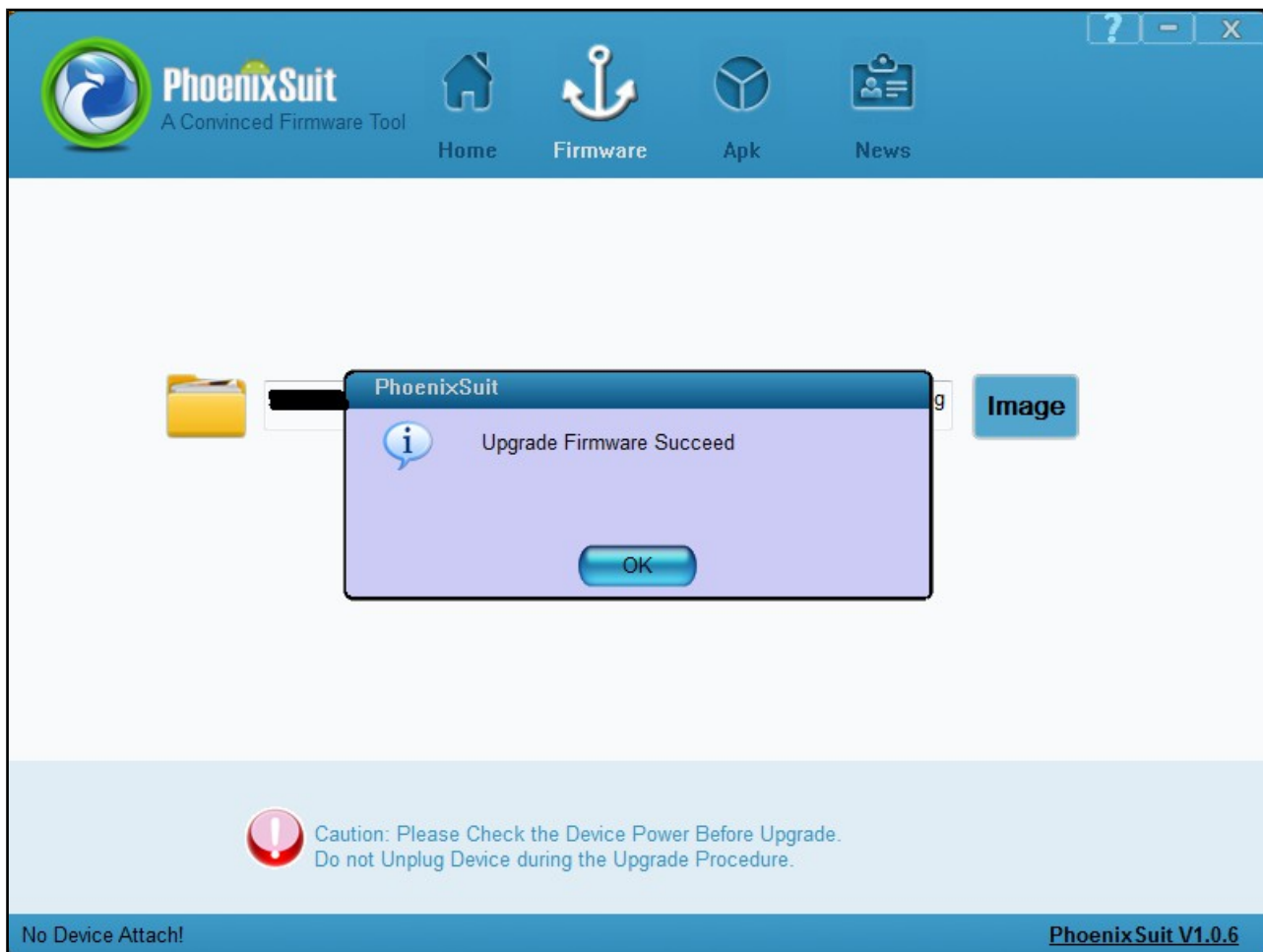
The board variant without NAND needs an SD card with bootable OS – Android or Linux. There are ready images available for download at the above-linked A20 wiki article.

To repair the image on NAND re-upload it following these easy steps:

1. Install and run PhoenixSuit (can be found in the wiki article of the board).
2. Go to firmware tab of the program and point to a valid Android image (the latest official one may also be downloaded from the wiki article).
3. Disconnect the power supply and USB cable from the A20 board.
4. Press and hold RECOVERY button, apply power supply 5V, release RECOVERY button.
5. Connect USB cable to the mini USB connector
6. You will be asked for drivers for the bootloader. Navigate to the folder where you extracted the PhoenixSuit and install the drivers from the respective executables (or manually point the installer to the drivers folder in the PhoenixSuit installation path).
7. PhoenixSuit will detect the board and would ask for the method of writing the image. Choose

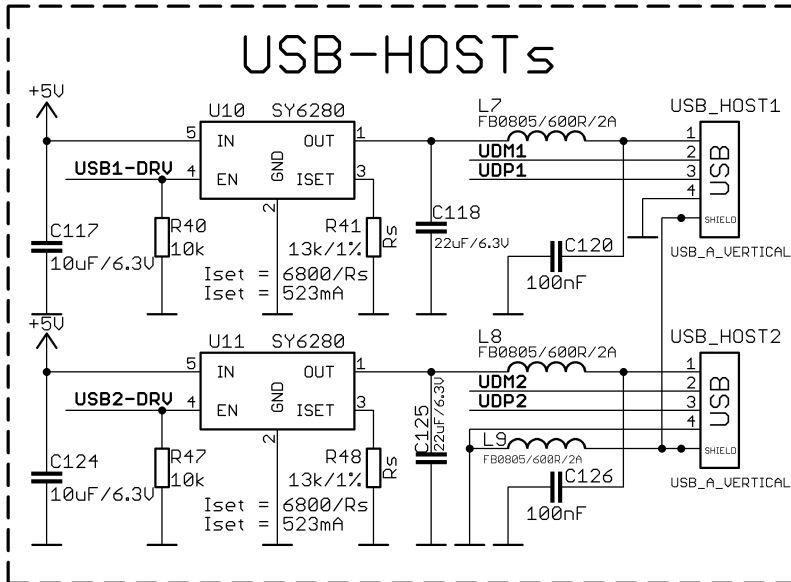
method of writing the image and confirm your wish to write the image.

8. Wait till upgrade succeeds as shown below:



6.5 USB_HOST connectors

The part of the schematic related to the USB_HOST connectors is listed below:



There are two USB host connector featured on the board. They are called USB_HOST1 and USB_HOST2. Each of them has own connector, both situated near the Ethernet connector. Each of connector has own low loss power distribution switch SY6280 which protects the board in case the devices you have plugged to the board try to draw more current than 523mA combined. The maximum current available on each of the USB hosts is exactly 523mA.

The SY6280 responsible for the USB_HOST1 is enabled by USB0-DRV1 (processor pin A4).

The SY6280 responsible for the USB_HOST2 is enabled by USB0-DRV2 (processor pin A5).

USB_HOST connectors					
USB_HOST1	Signal name	Processor pin	USB_HOST2	Signal name	Processor pin
Pin #1	5V	Connected to SY628	Pin #1	5V	Connected to SY628
Pin #2	UDM1	P20	Pin #2	UDM2	R20
Pin #3	UDP1	P21	Pin #3	UDP2	R21
Pin #4	GND	-	Pin #4	GND	-

The GND is common for both levels of the USB_HOST.

6.6 Ethernet

The LIME board is equipped with a standard RJ45 Ethernet connector at the top of the board that allows you to access local network via an Ethernet cable.

The LAN connectivity is handled by Realtek's RTL8201CP. Some of the features of this 10/100Mbit controller are:

- 10/100Mbps operation
 - Full/half duplex operation
 - Supports auto crossover detection
 - Adaptive equalization
 - IEEE 802.3/802.3u compliant
 - Supports IEEE 802.3u clause 28; 1.8V
 - Operation with 3.3V IO signal tolerance
- and much more

The Ethernet connector is grounded according to the standard (check GND1 and GND2 lines on the schematics).

If you use Linux then the Ethernet connector might be used for SSH connection to the board. If you lack any other option for debugging it might be hard to guess the correct IP of the board since it has DHCP enabled by default (especially if you are in a large network). It is good idea to check the default settings by exploring the Linux image settings. Those are usually stored in:

```
/etc/network/interfaces
```

For DHCP you need to enable auto detection and dhcp as shown below:

```
auto eth0
iface eth0 inet dhcp
```

For setting a static address please follow the pattern below:

```
auto eth0
iface eth0 inet static
address 192.168.1.5
netmask 255.255.255.0
gateway 192.168.1.254
```

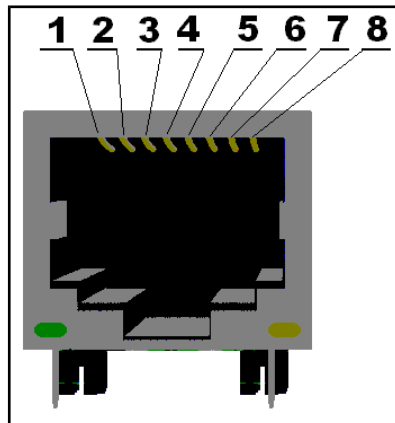
The EEPROM memory may be used to store the MAC address.

Important: In some Debian images it is possible that the Ethernet doesn't get auto-detected during boot-up. This is done on purpose because if there is auto-detection enabled and you don't want to use the Ethernet connector or you have forgotten to plug a cable the start-up would be greatly delayed. This might be problem in the first start-up to users counting only on SSH connection. You can enable the Ethernet after a successful boot-up with:

```
ifconfig -a
dhclient eth#(as seen after after the ifconfig command)
```

Important: You can configure static/DHCP-given IP addresses in /etc/network/interfaces

Ethernet	
Pin #	Signal name
1	TD+
2	TD-
3	VDD
4	NC
5	NC
6	VDD
7	RD+
8	RD-

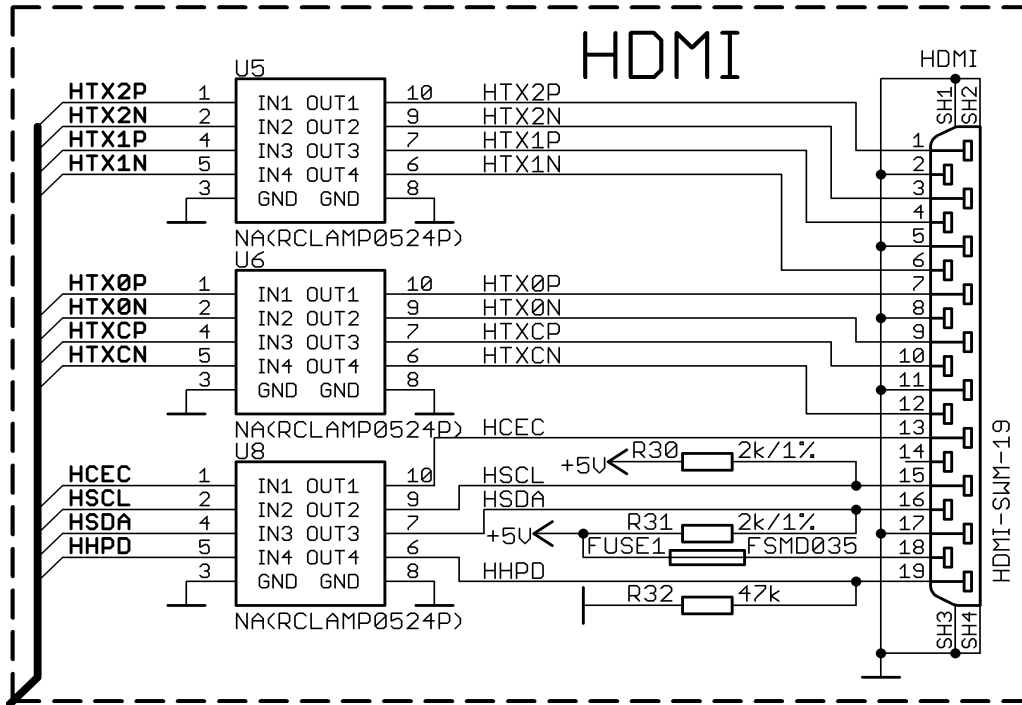


The RJ45 connector has a small yellow and a small built-in LEDS and they are described below:

LED	Color	Usage
Right	Green	Link status
Left	Yellow	Activity status

6.7 HDMI connector

The part of the schematic that describes the HDMI module is shown below:



Note that there are different Linux images depending whether the HDMI TV works at 50Hz or 60Hz refresh rate. If you board runs Android there is a specific option to set the appropriate HDMI output.

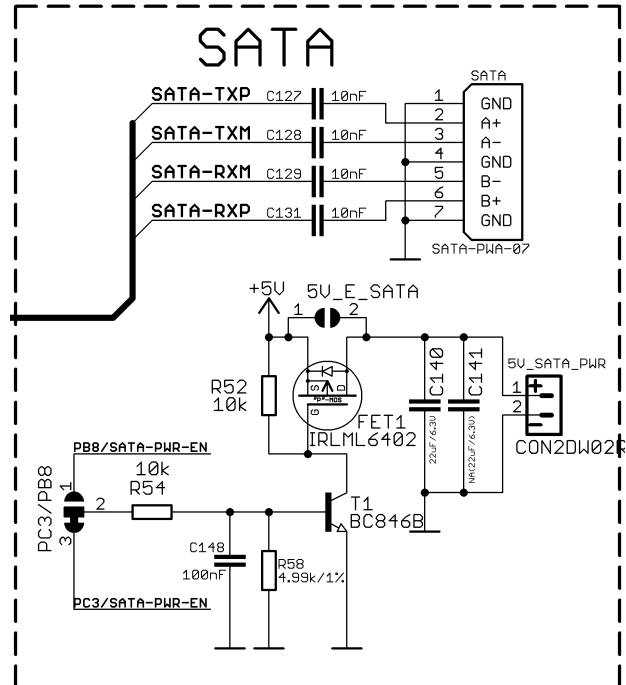
HDMI connector					
Pin #	Signal name	Processor pin	Pin #	Signal name	Processor pin
1	HTX2P	T23	11	GND	-
2	GND	-	12	HTXCN	W22
3	HTX2N	T22	13	HCEC	P23
4	HTX1P	U23	14	NA*	NA
5	GND	-	15	HSCL	R23
6	HTX1N	U22	16	HSDA	R22
7	HTX0P	U23	17	GND	-
8	GND	-	18	+5V	-
9	HTX0N	U22	19	HHPD	P22
10	HTXCP	W23			

*Pin 14 of the HDMI is not mandatory. In HDMI 1.0-1.3c it is reserved pin; in HDMI 1.4+ it is optional.

6.8 SATA connector and power

The part of the schematic describing the SATA module is shown below:

SATA connector		
Pin #	Signal name	Processor pin
1	GND	-
2	SATA-TXP	T20
3	SATA-TXM	T21
4	GND	-
5	SATA-RXM	U21
6	SATA-RXP	U20
7	GND	-



There is a 5V_SATA_PWR connector that is suitable for powering 5V hard-disks through the board. To keep the low form factor we have placed the small JST connector.

You can make an adapter cable accordingly. The other option is to check the SATA-CABLE-SET that we have available.

For more information please visit: <https://www.olimex.com/Products/Components/Cables/SATA-CABLE-SET/>

Pay note the two jumpers 5V_E_SATA and PC3/PB located at the bottom of the board (under the HDMI connector). The first one 5V_E_SATA is open by default because the board has software control of the powering for the hard disk. If you close 5V_E_SATA there would always be 5V at the 5V_SATA_PWR connector.

Which pin does the software control of the SATA powering is determined by the position of the PC3/PB8 jumper. By default PC3 is used.

6.9 GPIO connectors

There are four GPIO ports which are used generally to access unused by the board's peripherals pins. However, there are exceptions – some of the pins might be used to easily peripherals or their levels.

Three of the GPIO connectors have 40 pins and the last one has 20 pins. Note that all GPIO connectors have 0.05" (50 mil) step between pins. Olimex sells a compatible extension cable for these connectors named CABLE-40-40-10CM.

IMPORTANT: the cables and the 0.05" connectors are very fragile – if you attempt to disconnect the setup by pulling the cable it might break! Furthermore – you might bend the board's or the cable's pins! Use pliers or other suitable object to disconnect the connectors carefully!

Most of the pins are already defined in default operating system images. Some of them can be used as GPIOs, I2C or SPI without much of a problem. Information on the software usage of GPIO ports might be found in chapters “2.7 GPIO under Debian” and “2.8 I2C and SPI under Debian”

6.9.1 GPIO-1 (General Purpose Input/Output) 40pin connector

This connector features the processor signals needed for audio headphones output and microphone input. Please note that not only the connectors but also the rest of the schematic is missing! You would need to add resistors and capacitors. The connector also has the signals needed for a VGA output. The official Linux Debian images include a script that can be used to change the video output to VGA. We don't provide Android images for the VGA video output.

Microphone signals are: MICIN1 and VMIC

Headphone signals are: HPOUTL, HPCOM and HPOUTR

Please note that part of the VGA and composite video signals are multiplexed (VGA-R, VGA-B, VGA-G share the same pins with the TVOUT signals).

Composite video signals are located on GPIO-1 connector: TVOUT0 (GPIO-1 pin 20), TVOUT1 (GPIO-1 pin 22), TVOUT2 (GPIO-1 pin 24); you would also need AGND (GPIO-1 pin 4)

VGA signals are located at GPIO-1 connector and LCD_CON connector: The total of 5 signals you need for VGA output can be split into two categories:

1) Data signals:

VGA-R (AC17); VGA-B (AB16); VGA-G (AC16)

2) Synchronization signals:

LCD_VSYNC (AC9) and LCD_HSYNC (AB9)

If you inspect the schematics of a board with VGA connector (like A20-OlinuXino-MICRO) carefully you would notice that there are two extra transistors on the synchronization lines. Those are transistors meant to shift the levels to 5V. The 3.3V levels from the processor might work but the standard requires 5V levels of VSYNC and HSYNC. However, if you lead the signals to a 6 pin

header (like the one on the A20 board) you might use the cable for the A20-OLinuXino-MICRO (called A20-VGA-CABLE). The 6 signals needed are the 5 signals mentioned about plus GND.

A general advice for easier use of the microphone, headphone and VGA interfaces of A20-OLinuXino-LIME is to take a look at the schematics of another board with those interfaces already installed. For example, those are present in A20-OLinuXino-MICRO. Inspect the schematic carefully.

The signal layout is described below:

GPIO-1 connector					
Pin #	Signal name	Processor pin	Pin #	Signal name	Processor pin
1	+5V	-	2	GND	-
3	+3.3V	-	4	GND	-
5	PG0	F20	6	LRADC0	AB23
7	PG1	E21	8	LRADC1	AB22
9	PG2	E20	10	MICIN1	AC20
11	PG3	D21	12	VMIC	AA21
13	PG4	D20	14	HPOUTL	Y19
15	PG5	C21	16	HPCOM	AA19; AA20
17	PG6	E19	18	HPOUTR	W19
19	PG7	C20	20	VGA-G	AC16
21	PG8	D19	22	VGA-B	AB16
23	PG9	C19	24	VGA-R	AC17
25	PG10	D18	26	UART0-RX	B7*
27	PG11	C18	28	UART0-TX	A7*
29	PC3/SATA-PWR-EN	L22**	30	PB21	C7
31	PC18	F22	32	PB20	C8
33	PC19	L21	34	PB19	B8
35	PC20	K21	36	PB18	A8
37	PC21	J21	38	PC24	F21
39	PC22	J20	40	PC23	G19

*Signals marked with * are multiplexed with either a peripheral device or another connector. Ensure that the multiplexing is handled properly on software and/or hardware level (else there would probably occur wrong levels, loss of data, etc).

**Pin #29 is used to control the power supply available at the 5V_SATA_PWR by default

6.9.2 GPIO-2 (General Purpose Input/Output) 40pin connector

The GPIO pins are led out on a separate 40pin connector. They allow the user to attach additional hardware, check readings or perform hardware debug. The GPIO-2 connector numbers are printed at the bottom of the board for your convenience.

GPIO-2 connector					
GPIO pin#	Signal name	Processor pin#	GPIO pin#	Signal name	Processor pin#
1	5V	-	2	GND	-
3	3.3V	-	4	LD03_2.8V	Goes to AXP209
5	TWI0-SCK	A15	6	PE0	E23
7	TWI0-SDA	B15	8	PE1	E22
9	PI0	A20	10	PE2	D23
11	PI1	B20	12	PE3	D22
13	PI2	A19	14	PE4	C23
15	PI3	B19	16	PE5	C22
17	PI4	A18	18	PE6	B23
19	PI5	B18	20	PE7	B22
21	PI6	A17	22	PE8	A23
23	PI7	B17	24	PE9	A22
25	PI8	A16	26	PE10	B21
27	PI9	B16	28	PE11	A21
29	PI10	A15	30	PI21	E13
31	PI11	B15	32	PI20	E14
33	PI12	A14	34	PI19	D14
35	PI13	B14	36	PI18	E15
37	PI14	A13	38	PI17	E16
39	PI15	B13	40	PI16	E17

Signals marked with * are multiplexed with either a peripheral device or another connector. Ensure that the multiplexing is handled properly on software and/or hardware level (else there would probably occur wrong levels, loss of data, etc).

6.9.3 GPIO-3 (General Purpose Input/Output) 40pin connector

GPIO-3 connector					
GPIO pin#	Signal name	Processor pin#	GPIO pin#	Signal name	Processor pin#
1	5V	-	2	GND	-
3	3.3V	-	4	GND	-
5	RESET_N	C14	6	PB3	B14
7	PH0	C6	8	PB4	A13
9	PH7/PB9	B4/C12	10	PB5	B13
11	PH9	D4	12	PB6	A12
13	PH10	A3	14	PB7	B12
15	PH11	B3	16	PB8/SATA-PWR-EN	A11
17	PH12	C3	18	PB10	C11
19	PH13	A2	20	PB11	C10
21	PH14	B2	22	PB12	C9
23	PH15	A1	24	PB13	B11
25	PH16	B1	26	PB14	A20
27	PH17	C1	28	PB15	B10
29	PH18	C2	30	PB16	A9
31	PH19	D1	32	PB17	B9
33	PH20	D2	34	PH24	E3
35	PH21	D3	36	PH25	E4
37	PH22	E1	38	PH26	F3
39	PH23	E2	40	PH27	F4

6.9.4 GPIO-4 (General Purpose Input/Output) 20pin connector

GPIO-4 connector features the rest of the signals available and few additional signals that might need testing.

GPIO-4 connector					
GPIO pin#	Signal name	Processor pin#	GPIO pin#	Signal name	Processor pin#
1	3.3V	-	2	GND	-
3	TVOUT3	AB17	4	NMI_N	F5
5	TVIN0	AC18	6	MIC1OUTP	AC22
7	TVIN1	AB18	8	MIC1OUTN	AC23
9	TVIN2	AA17	10	MICIN2	AC21
11	TVIN3	Y17	12	GPIO2	Goes to AXP209
13	LINEINR	AB21	14	GPIO3	Goes to AXP209
15	LINEINL	AB20	16	PC7	J22
17	FMINR	Y21	18	PC16/NWP	M21*
19	FMINL	Y20	20	PC17	F23

Signals marked with * are multiplexed with either a peripheral device or another connector. Ensure that the multiplexing is handled properly on software and/or hardware level (else there would probably occur wrong levels, loss of data, etc).

6.10 LCD_CON 40pin connector

The LCD_CON pins are led out on a separate 40pin connector for the ease of connecting an LCD. We have tested the ability of the board to interact with such a display. They allow the user to attach additional hardware, check readings or perform hardware debug.

Important: you need additional 40PIN ribbon cable to connect an Olimex display and the LCD_CON.

The LCD connector is suitable for a number of Olimex displays and touchscreen panels with different native resolution – the smallest available is the 4.3" one called **LCD-OLinuXino-4.3TS** with native screen resolution of **480×272**, through the 7" one named **LCD-OLinuXino-7TS** with **800×480**, to the **1024×600 10.1" LCD-OLinuXino-10TS**.

LCD_CON connector					
GPIO pin#	Signal name	Processor pin	GPIO pin#	Signal name	Processor pin
1	+5V	-	2	GND	-
3	+3.3V	-	4	GND	-
5	LCD_D16	Y12	6	LCD_D17	AA12
7	LCD_D18	Y11	8	LCD_D19	AA11
9	LCD_D20	Y10	10	LCD_D21	AA20
11	LCD_D22	AB12	12	LCD_D23	AC10
13	LCD_D8	AB11	14	LCD_D9	AC11
15	LCD_D10	Y15	16	LCD_D11	AA15
17	LCD_D12	Y14	18	LCD_D13	AA14
19	LCD_D14	Y13	20	LCD_D15	AA13
21	LCD_D0	AB15	22	LCD_D1	AC15
23	LCD_D2	AB14	24	LCD_D3	AC14
25	LCD_D4	AB13	26	LCD_D5	AC13
27	LCD_D6	AB12	28	LCD_D7	AC12
29	LCD_HSYNC	AB9	30	LCD_VSYNC	AC9
31	LCD_CLK	Y9	32	LCD_DE	AA9
33	PB3	B14	34	PB4	A13
35	LCD_PWR	C4	36	PB2/PWM0	A14*
37	TPX1	Y22	38	TPX2	AA22
39	TPY1	Y23	40	TPY2	AA23

Signals marked with * are multiplexed with either a peripheral device or another connector. Ensure that the multiplexing is handled properly on software and/or hardware level (else there would probably occur wrong levels, loss of data, etc).

IMPORTANT:

To use Android with display or resolution different than HDMI and 1024×600 (default settings) you need to upload new Android image to the board. These images are available here: <https://www.olimex.com/wiki/A20-OLinuXino-LIME> (using an operating system – NAND flash).

In Debian Linux you would need to start a shell script to change the built-in the image resolution settings. Start it with `./change_display_a20_lime.sh` and follow the instructions. Please refer to chapter 2.5 “Changing the default image resolution” for more information.

6.11 Jumper description

Please note that most the jumpers on the board are SMT type. If you feel insecure of your soldering/cutting technique it is better not to try to adjust the jumpers since it is possible to damage the board.

Board jumpers			
Jumper name	Type	Default position	Function
NAND_E	SMT	Either CLOSED or OPEN, depending whether the board has 4GB NAND MEMORY MODULE	Controls whether the NAND memory module is enabled or disabled. If the jumper is closed than the NAND memory is operational.
PC3/PB8	SMT	PC3	Controls which processor pin should be used for SATA power enabling signal. By default the SATA powering control uses processor pin L22 which is port PC3. Changing this jumper also changes the availability of PC3 and PB8 at GPIO-1 pin 29 and GPIO-3 pin 16. Please note that if you change the position of the jumper from PC3 to PB8 then you might block one of the processor pins responsible for I2S and AC97 audio output.
PB9/PH7_USB	SMT	PB9	Controls which processor pin should be used for the USB_OTG. By default the USB_OTG uses processor pin C12 which is port PB9. The same pin/port is used for I2S audio. In case you want to access the I2S capabilities of the board you would need to change the default position.
PB9/PH7_GPIO	SMT	PH7_GPIO	PB9/PH7_GPIO jumper controls which of the two ports (PB9 or PH7) should be lead out to pin 9 of GPIO3 connector for easier access. Changing this jumper also changes the signal available at GPIO-3 connector's pin 9. By default this jumper is in PH7_GPIO position and PH7 is lead to pin 9 of GPIO3.

6.12 Additional hardware components

The components below are mounted on OLinuXino but are not discussed above. They are listed here for completeness:

Reset button – used to reset the board

Power button – used to reset the board

Recovery button – used to reset the board

512MB = 1× [4Gb (256M x 16 bit)] DDR3 SDRAM – the exact memory used currently in the board is SAMSUNG K4B4G1646D-BCK0

The DDR3 memory part name in the schematic might be outdated. We have used a number of different but fully compatible DDR3 memories due to supply unavailability. It is always recommended to check the exact memory name printed on the component itself.

4GB = 1× [32Gb (4096M x 8 bit)] NAND FLASH – only available on the 4GB version of the board; the exact memory used in the board revision mentioned is Hynix H27UBG8T2A

PWR_LED – turns on upon powering the board

CHGLED – lights when charging a battery connected to the Li-Po battery connector.

CHAPTER 7: SCHEMATICS

7. Introduction to the chapter

In this chapter is located information about the schematics describing logically and physically A20-OLinuXino-LIME.

7.1 Eagle schematic

OLinuXino schematics may be found it on the OLinuXino's GitHub repository: <https://github.com/OLIMEX/OLINUXINO/tree/master/HARDWARE>. You can download the whole repository as .zip without having a GitHub account.

Please note that the schematics of A20-OlinuXino-LIME and A10-OlinuXino-LIME are similar and it is possible that a single board file and schematic for both boards exist.

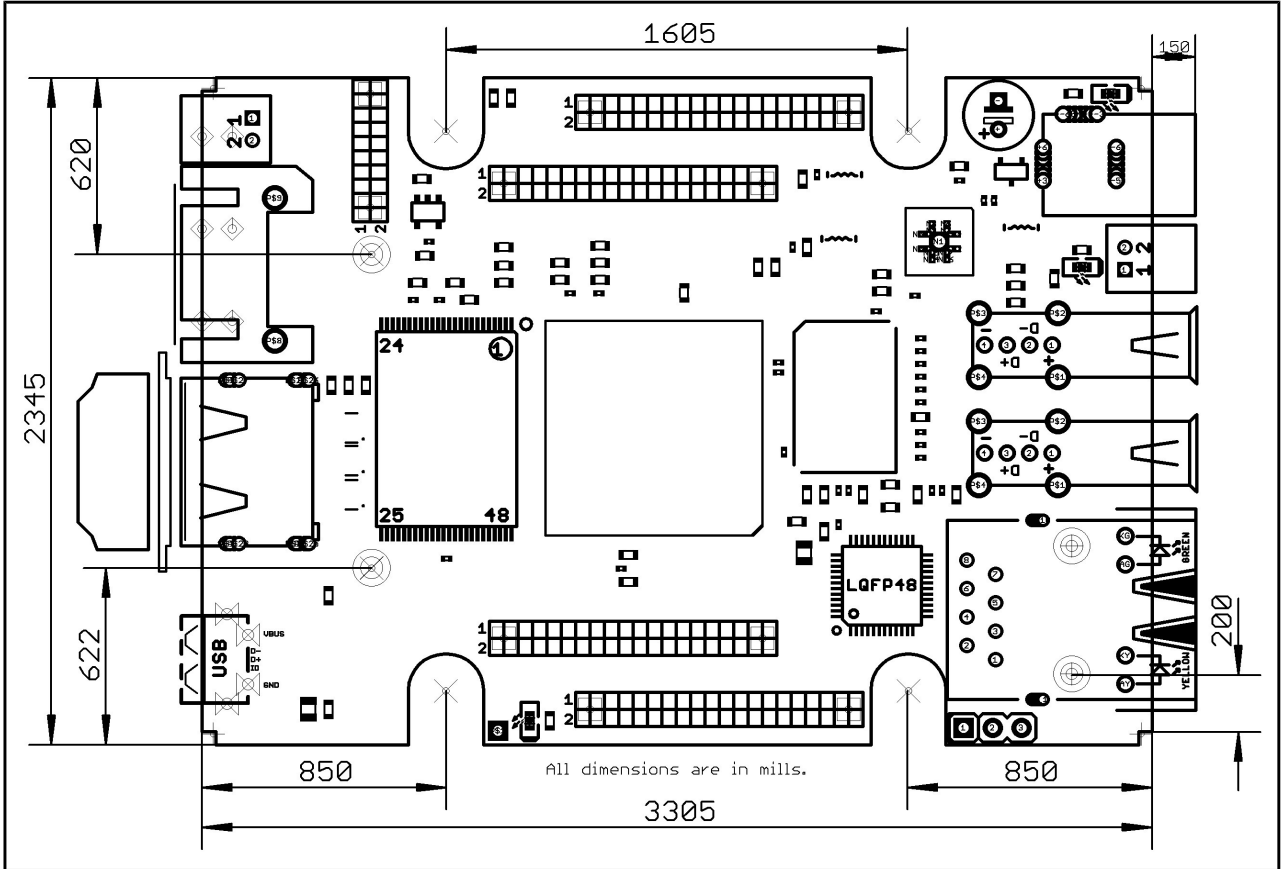
We mostly use Eagle by Cad Soft 4.16r2 for designing. However, the files should be compatible with the latest Eagle available. Cad Soft offers a trial version of their software that allows you to inspect schematics and board files (without being able to modify them).

This work is licensed under the Creative Commons Attribution-ShareAlike 3.0 Unported License. To view a copy of this license, visit <http://creativecommons.org/licenses/by-sa/3.0/>.

If you are looking for a schematic of an older revision of the board and it isn't available at our web site you may request it by the support e-mail.

7.2 Physical dimensions

Note that all dimensions are in mils.



CHAPTER 8: REVISION HISTORY AND SUPPORT

8. Introduction to the chapter

In this chapter you will find the current and the previous version of the document you are reading. Also the web-page for your device is listed. Be sure to check it after a purchase for the latest available updates and examples.

8.1 Document revision

Document revision	Changes	Modified pages
A, 28.08.14	Initial manual release	All
B, 27.10.14	Adjusted information about signals required for VGA output	12, 30
C, 03.11.14	Added information about new displays Added information about GPIO, I2C, SPI New section about the initial connection	12, 13, 14, 31, 32
D, 30.01.15	Fixed information about power jack	25
E, 09.02.15	Fixed information about the CMOS-TTL convertor	11
F, 10.02.15	Added information about the SSH via miniUSB in Debian	12
G, 16.02.15	Information about the Windows driver for the tethering Clarified how to prepare a microSD card with Android OS	12 47
H, 17.03.15	Updated HDMI part according to the latest schematic Updated DDR3 memory according to the latest schematic	33 41
I, 13.05.15	Fixed the table of GPIO-2 according to the latest schematic	37

8.2 Board revision

Remember to check the schematics and the board design files to compare the differences.

Board revision	Notable changes
C	Initial release of the board

8.3 Useful web links and purchase codes

The web page you can visit for more information about your device are:

- *Product page: <https://www.olimex.com/Products/OLinuXino/A20/A20-OLinuXino-LIME/>
- *Wiki article of the board: <https://www.olimex.com/wiki/A20-OLinuXino-LIME>
- *A place for general questions, FAQ or friendly talk: <https://www.olimex.com/forum/>.
- *You can get the latest updates on the software at: <https://github.com/OLIMEX/OLINUXINO>.
- *You may may join our IRC channel #olimex @ freenode.net (<http://webchat.freenode.net/?channels=olimex>).
- *The sunxi community is behind the Allwinner Linux support: http://linux-sunxi.org/Main_Page.

ORDER CODES:

A20-OLinuXino-LIME – the product which the document manual is about

A10-OLinuXino-LIME-BOX – an aesthetic box specially made for the board that protects it from dust and accidental short-circuits

USB-SERIAL-CABLE-F – USB serial console cable female

USB-MINI-CABLE – standard USB type A to USB type mini cable

SY0605E – reliable power supply adapter 50Hz (EU) 5V/1A for A20-OLinuXino-LIME

SY0605E-CHINA – cheaper power supply adapter 50Hz (EU) 5V/1A for A20-OLinuXino-LIME

A20-LIME-ANDROID-SD – a tested class 10 micro SD card with the latest (by the time of leaving the Olimex facilities) official Android release

A20-LIME-DEBIAN-SD – a tested class 10 micro SD card with the latest (by the time of leaving Olimex facilities) official Debian Linux release

A20-OLinuXino-LIME-UEXT – an adapter from 0.05" step connector to 0.1" step one; also suitable for LCD adapting

LCD-OLINUXINO-4.3TS – low-cost 4.3" LCD display with touchscreen component – 480×272

LCD-OLINUXINO-7TS – low-cost 7" LCD display with optional touchscreen component – 800×480

LCD-OLINUXINO-10TS – low-cost 10" LCD display with optional touchscreen component – 1024×600

LCD-OLINUXINO-15.6 – low-cost 15.6" LCD display – 1366×768

LCD-OLINUXINO-15.6FHD – low-cost Full HD 15.6" LCD display – 1920×1080

CABLE-40-40-10CM – 0.05" step cable for connecting a new display to the board

CABLE-IDC40-15cm – 0.1" step cable for connecting an LCD to A20-OLinuXino-LIME2-UEXT

SATA-HDD-2.5-500GB – 5GB 2.5" SATA hard disk

SATA-CABLE-SET – cables that allow the connection of a 2.5" hard disk to the board

ALUMINIUM-HEATSINK-20×20×6MM – heatsink radiator for better processor heat dissipation

How to purchase?

You can purchase directly from our online shop or from any of our distributors. Note that usually it is faster and cheaper to purchase Olimex products from our distributors. List of confirmed Olimex LTD distributors and resellers: <https://www.olimex.com/Distributors>.

Please visit <https://www.olimex.com/> for more info.

8.4 Frequently asked questions

Q: I power my board, it shows a logo and then nothing happens.

A: This might be due to a number of reasons but it is recommended to try the following:

1. Download latest official image from our wiki (either Android or Debian for SD card), and upload it to an SD card again. There are instructions how to do it in the other questions below. Try if the board works now.
2. Check if your power supply provides enough current, try with different/better power supply.
3. Check the USB hub you are using, plug the USB cable directly to the back of your personal computer, check the USB cable.
4. The board might enter sleep mode very fast (especially if the Android was turned off without the "Quick Boot" mode being ticked on). Try pressing or holding down the PWR button for a couple of seconds to wake it up.

Q: I use LCD_CON for video output. There is some screen flickering under Debian. What could be the reason?

A: This is usually caused by either a loose cable connection or badly configured script.bin. Double check if all cables are properly connected. Also check if there is a change in the behavior if you either twist the cables carefully or press different components at the back.

If the above doesn't fix the flickering problem, edit the script bin parameter `lcd_io_cfg0` – make sure that it is set to `= 0x10000000 (1x1 and 7x0)`.

Q: How do I write the Linux image to a micro SD card to use with my A20 board?

A: To write a Linux image to an SD card under Windows we use Win32 Disk Imager:

Download Win32 Disk Imager software
Insert card
Start program
Select file
Click "write"

To write a Linux image to an SD card under Linux:

For instance, an image with the file name of "debian_2g.img" would be downloaded to the microSD card connected to a Linux machine using one of the following commands:

```
# dd bs=4M oflag=sync if=debian_2g.img of=/dev/sdX  
or
```

```
# cp debian_2g.img /dev/sdX
```

where X is the microSD card.

The tips above can also be used to prepare a microSD card with an Android image, suitable for a microSD card.

Q: How to detect and enable the Ethernet controller (if it is disabled by default)?

A: You can enable it by following these two steps:

1. To check under what name the LAN is associated write "ifconfig -a"
2. If, for example, it is under eth0 name, then write: "dhclient eth0"

This should enable the Ethernet and then SSH would also be available.

You can also enable auto detection of Ethernet on power-up by removing the comment #auto eth0 in /etc/network/interfaces in the Linux image.

Q: How to install Android on an SD-card?

A: First download one of the official Android images, which might be found in the Android section above.

Make sure that the download link you visit clearly indicates that the image is suitable for the microSD card since there are images suitable for NAND memory also. The images suitable for the microSD memory and those suitable for NAND card are different. However, the upload method is almost identical – using PhoenixSuit.

There are two types of Android images for microSD card that we usually provide and each of them has to be downloaded to a microSD card using a different method. The image provided for microSD card is either the native Android image that can be downloaded to the card via a software tool like PhoenixSuit (through the board) or an image taken from an already prepared microSD card that requires to simply write the image (through a microSD card reader).

It is more likely that you have an Android image that requires a simple copy to a card. If that is the case you can follow the exact steps as for Linux (e.g. using "Win32 Disk Imager" or "dd" command).

In order to prepare a microSD card with a native Android you will need a software tool called PhoenixSuit and then:

- Install and run PhoenixSuit.
- Go to the firmware tab of the program and point to a valid Android image (note that the images on Gdrive are compressed and you have to extract the archives to .img files to be able write them with PhoenixSuit)
- Disconnect the power supply and USB cable from the A20 board. Put an SD card in micro SD holder. We recommend 4GB class 10 card.
- Press and hold RECOVERY button, apply power supply 5V, release RECOVERY button.
- Connect USB cable to the mini USB connector.
- You will be asked for drivers for the boot-loader. Navigate to the folder where you extracted the PhoenixSuit and install the drivers from the respective executables (or manually point the installer to the drivers folder in the PhoenixSuit installation path).

- PhoenixSuit will detect the board and would ask for the method of writing the image. Choose method of writing the image and confirm your wish to write the image.
- Wait till upgrade succeeds

Note that it is not recommended to have your mini USB connected to an external USB hub. This might cause delays and might distort the signal levels. Always test with the USB connected straight to the USB ports of your computer.

Important: When Android runs for very first time it takes several minutes to initialize all files and buffers please do not cut the power supply during this process! Also when fresh image is installed fast boot may be disabled, which means that when you apply power supply after few seconds Android will go in sleep mode and you have to press PWR button to start it, you can change to fast boot when you power off there is dialog box asking you if you want next boot to be fast boot, you have to check this box before you power off. Also note that you must do touch screen calibration when you run Android for very first time which might require a mouse.

Q: How to generate boot-able SD-card Debian Linux image for A20-OLinuXino-LIME?

A: Build instructions and required files for the latest Debian images: [GitHub location](#)

Note that Linux-Sunxi Kernel is a work-in-progress, this means you can try the current stage/sunxi-x.x branch but if something is broken and doesn't work just revert to the git tags we give in the blog and they would work for sure.

[Sunxi u-boot loader](#) - the linux-sunxi git page contains a lot of sources for all Olimex Allwinner boards.

Q: Is it possible to boot Debian from NAND? Do you provide such image?

A: It is possible but we don't provide such image. There are people who were successful in booting Debian from the NAND, however. Make sure to check on the forum. Make sure to check the number of very good and optimized A20-OLinuXino Debian images by Igor Pečovnik. There are also instructions for NAND installation of Debian: [link to his web-site](#).

Q: How to connect a display and adjust the output resolution?

A: It is explained above in the user's manual. For extra reading there is a separate article on the subject: [wiki article](#)

Q: How to edit board configurations and definitions in the official Debian Linux?

A: Do you want a custom video resolution output? Do you need a different port definition? Do you need to change the hardware definitions of the board?

It is explained above in the manual. But you can also visit this wiki article: [wiki article](#).

Q: How to edit board configurations and definitions in the official Android images?

A: There is a wiki article about that, please visit: [wiki article](#).