A20-OLinuXino-MICRO

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

USER'S MANUAL

Document revision P, September 2017
**Designed by OLIMEX Ltd, 2015**

All boards produced by Olimex LTD are ROHS compliant

DISCLAIMER

© 2017 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/.
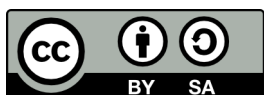
This hardware design by Olimex LTD is licensed under a Creative Commons Attribution-ShareAlike 3.0 Unported License.

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-MICRO. THEY ARE CONSIDERED SUITABLE ONLY FOR A20-OLINUXINO-MICRO.

Table of Contents

CHAPTER 1: OVERVIEW

1. Introduction to the chapter

Thank you for choosing this A20 single board computer from Olimex! This document provides a user's guide for the Olimex A20-OLinuXino-MICRO 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-MICRO 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 that we used to manufacture the board is available to the customer.

1.1 Features

The board has the following set of features (please note the difference between the two versions of the board):

- A20 Cortex-A7 dual-core ARM Cortex-A7 CPU and dual-core Mali 400 GPU
- 1GB DDR3 RAM memory
- **4GB or 8GB NAND FLASH memory (available only on certain variants of the board)**
- **4GB eMMC memory (available only on certain variants of the board)**
- **Industrial temperature range versions available (boards with "-IND" suffix)**
- 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
- VGA output on 6-pin 1.25mm (0.05") step connector
- 100MBit native Ethernet
- Battery connector with battery-charging capabilities
- Audio headphones output on connector
- Microphone input on connector
- 2 x UEXT connectors
- LCD connector compatible with with 4.3", 7.0", 10.1" LCD displays from Olimex
- 160 GPIOs on three GPIO connectors
- MicroSD card connector
- SD/MMC card connector
- DEBUG-UART connector for console debug with USB-SERIAL-CABLE-F
- Battery charge status LED and GPIO LED
- 2KB EEPROM for MAC address storage and more
- 10 BUTTONS with ANDROID functionality + RESET button
- 4 mount holes
- 6-16V input power supply, noise immune design
- PCB dimensions: (5600×3250) mils ~ (142.24×82.55) 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 strong points of the boards are the processor speed, the small form factor and the low price-to-productivity ratio. The A20-OLinuXino-MICRO is a mature product – a lot of experience and projects are already available. The A20 chip is also very well known and well supported.

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

The board has six board variants. The base version of the board, that comes with NO extra flash memory, and works in the commercial temperature range is named **A20-OLinuXino-MICRO.** The base version is usually sufficient for research and development (unless your project specifically requires industrial temperature range and extra flash memory storage).

You might find a comparison table of the different board variants below:

Board variant	Extra flash	Type of flash	Size of flash	Boot Linux from flash memory	Operating temperature range
A20-OLinuXino-MICRO	No	-	-	-	0°C – 70°C
A20-OLinuXino-MICRO-IND	No	-	-	-	-40°C – 100°C
A20-OLinuXino-MICRO-e4GB	Yes	eMMC	4GB	Yes, after install script	0°C – 70°C
A20-OLinuXino-MICRO-e4GB-IND	Yes	eMMC	4GB	Yes, after install script	-40°C – 100°C
A20-OLinuXino-MICRO-n4GB	Yes	NAND	4GB	Yes, after install script	0°C – 70°C
A20-OLinuXino-MICRO-n8GB	Yes	NAND	8GB	Yes, after install script	0°C – 70°C

The different versions of the boards can be determined by the suffixes after the base A20-OLinuXino-MICRO name, different names determine different version. The suffixes indicate three things:

i) type of flash ("e" stands for eMMC; "n" stands for "NAND"; if there is flash memory at all)
ii) size of flash ("4GB" or "8GB")
iii) whether the board uses industrial temperature grade components (when "IND" suffix is present)

Not all combinations are present, for example, at the moment we don't have industrial version of the board with NAND memory. The prices for different board versions are also different. These four versions with extra flash memory are named: **A20-OLinuXino-MICRO-e4GB**, **A20-OLinuXino-**

MICRO-e4GB-IND, A20-OLinuXino-MICRO-n4GB, and **A20-OLinuXino-MICRO-n8GB**. The name consists of the type of the memory and the size of the memory – suffix "**e4GB**" stands for eMMC type of flash with 4GB of size; "**n4GB**" stands for NAND type of flash with 4GB of size; "**IND**" stands for a board with industrial temperature range components.

The other Olimex boards with close characteristics are the ones with A13 and A10 microcontrollers. The A13 boards feature a generation older processor but since they have been longer on the market they have better Linux and Android support. The A10 boards feature single (than A20) processor core but they are more energy efficient making them better choices for handheld devices and devices requiring power efficiency. A more powerful chip than the A20 is used in the design of A64-OLinuXino-eMMC.

For projects and designs that require smaller form factor good alternatives to A20-OLinuXino-MICRO are: A20-OLinuXino-LIME2, A20-OLinuXino-LIME and A20-SOM-EVB. For more powerful and recent designs consider the A64-OLinuXino board.

1.4 Board version used in the manual

Board from revision J was 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.

1.5 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

This product 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 A20 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-MICRO 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)
- 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); or OLIMEX LCD (TS) display + 40-pin CABLE-IDC40-15cm; or A20-VGA-CABLE + VGA monitor/screen/projector
- 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 – 6-16V DC, 5W required (10V @ 0.5A) – for optimal power
- USB-SERIAL-CABLE-F – for Android/Linux debugging on UART0
- CABLE-IDC40-15cm – cable used for LCD_CON ↔ Olimex LCD display
- Adapter cable for the 6-pin VGA connector to standard VGA 15-pin connector

Additional items include:
- Audio device for HEADPHONES jack
- Ethernet cable for wired Ethernet
- A number of extension modules that can add functionality or interface to the board on the UEXT connector; these can be explored here: https://www.olimex.com/Products/Modules/

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

SY0612E – reliable power supply adapter 50Hz (for EU) 12V/0.5A for A20-OLinuXino-MICRO
SY0612E-CHINA – cheaper power supply adapter 50Hz (for EU) 12V/0.5A for A20-OLinuXino-MICRO
A20-Android-SD – a tested class 10 micro SD card with the latest (by the time of leaving the Olimex facilities) official Android release
A20-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 – USB serial console cable female
USB-MINI-CABLE – standard USB type A to USB type mini cable
A20-VGA-CABLE – adapter from 6-pin connector to 15-pin one
CABLE-IDC40-15cm – cable for LCD to LCD_CON connection
LCD-OLinuXino-4.3TS – low-cost 4.3" LCD display with touchscreen component – 480×272
LCD-OLinuXino-7TS – low-cost 7" LCD display with touchscreen component – 800×480
LCD-OLinuXino-10TS – low-cost 10.1" LCD display with touchscreen component – 1024×600
LCD-OLinuXino-15.6 – low-cost 15.6" LCD display with touchscreen component – 1024×600
LCD-OLinuXino-15.6FHD – low-cost 15.6" LCD display with touchscreen component – 1920×1080
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

2.3 Powering the board

There are three possible ways of powering A20-OLinuXino-MICRO – via external supply providing 6-16V 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 consumes around 300mA of current at 12V when there are no peripherals connected to the USB hosts, so make sure the power supply is able to provide at least 500mA before plugging. 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. The best practice is not to count on the mini USB-OTG as a single power source!

The preferred way of powering the board is via the PWR jack with 6-16V DC with a power of 5W (e.g. 6Vx0.8A; 16Vx0.3A). This will 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 power 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-OLinuXino-MICRO is between 100mA and 320mA depending on the current load and the power voltage applied.

If the board has entered power-down state you can bring it back without restart using the PWR_BUT. The PWR_BUT 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_BUT 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 Prebuilt software

The board comes with no software uploaded. The board versions with extra flash memory also come blank.

You can download the latest Debian or Android image from the wiki article of the board and download it to a microSD card.

If your board has extra flash memory and you want to boot Android from it refer chapter *6.1.1 USB-OTG communication (NAND firmware repair/update).*

2.5 Button functions

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

PWR_BUT – 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 couple of seconds to perform each action
RESET – used for hardware reset of the board – it is not recommended
RECOVERY – used to wake up the board from sleep

The following buttons represent functions in the Android (it is possible that not all Android

applications take advantage of the buttons, in such case the button would serve no purpose for that application):

VOL+ – increases the volume
VOL- – lowers the volume
MENU – brings up the main menu
SEARCH – brings up search feature
HOME – shows the home screen; note that HOME is also used to enter bootloader mode for firmware update
ESC – used to navigate away of a menu
ENTER – to select a choice

It is not recommended to disconnect the power supply (either the USB or the power jack) before turning off the Android from the either the menus or by holding PWR_BUT system written on the NAND.

How to restore the Android image might be found in chapter "6.1.1 USB-OTG communication (NAND firmware repair/update)".

2.6 Interacting with the board

The typical and recommended way of interacting with a stand-alone A20-OLinuXino-MICRO board is via a serial cable connected to a personal computer. You would probably need a cable suitable for such a connection due to the fact that most personal computers lack a serial port nowadays. We distribute such a 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:
https://www.olimex.com/Products/Components/Cables/USB-Serial-Cable/USB-Serial-Cable-F/

You need to connect the serial cable lines as follows: RX line to UART0-TX pin; TX line to UART0-RX pin; GND to GND. 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.

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

In addition to the serial communication, you might also use one or more of the following mediums to interact with the board:

1. a monitor via HDMI connector
2. a monitor via the VGA connector and a VGA adapter
3. SSH via the mini USB connector trough a mini USB cable
4. SSH with a remote computer via LAN connector
5. a display via LCD_CON connector

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.

2.6.1 HDMI monitor

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

Make sure to use a tested HMDI 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.

More information about the video output settings and the usage of video settings script might be found in the next chapter "2.7 Changing the default image resolution".

2.6.2 VGA monitor

All official Debian images for A20-OLinuXino-MICRO have the option to for VGA video output via the 6-pin VGA connector. If you wish to transform the custom 6-pin connector to a standard 15-pin VGA connector you can either use wires or get a ready adapter called "A20-VGA-CABLE".

More information about the video output settings and the usage of video settings script might be found in the next chapter "2.7 Changing the default image resolution".

2.6.3 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.

2.6.4 SSH via Ethernet

By default the board requests IP address from the network's DHCP server. You need to find what IP address was assigned to the board before being able to access it over the network. Knowing the IP address assigned allows you to connect to the board using an SSH client (for example "puTTY") then you have to use this address. You can change this policy or the address by editing /etc/network/interface file.

Note that for internet connection you have to set your gateway address in /etc/network/interfaces file and you have to set your DNS server in /etc/resolv.conf (for example "nameserver 192.168.1.1")

2.6.5 LCD display

One of the ways to interact with the board is via an external display (with or without touchscreen component). The 40-pin male connector LCD_CON has the typical 0.1" pin step. All Olimex displays have corresponding 40-pin male connector. You would only need a 0.1" female-female cable for the hardware connection.

In order to get the video output on the LCD display you might need either to run at least once 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

More information about the video output settings and the usage of video settings script might be found in the next chapter "2.7 Changing the default image resolution".

2.7 Changing the default image resolution

Depending on the display or the screen you want to use with the A20-OlinuXino-MICRO, 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-MICRO.

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_OLinuXino.sh
and choose the resolution and the interface (LCD, HDMI or VGA).

The supported resolutions are:

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: (note that the VGA signals are routed to custom 6 pin connector and you need to from adapter to standard VGA connector, Olimex also sells such adapter cables)

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

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 sdcard 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 enabled.

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

2.8 Connecting and calibrating a display

One of the ways to interact with the board is via an external display (with or without touchscreen component). The 40-pin male connector LCD_CON has the typical 0.1" pin step. All Olimex displays have corresponding 40-pin male connector. You would only need a 0.1" female-female cable for the hardware connection.

All LCD displays made by Olimex have at least a 0.1" LCD connector. 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 displays recommended for the board at the moment of writing might be found in the table below:

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	10.1	1024×600	Yes	Yes	Product page
LCD-OLinuXino-10TS	10.1	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.

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 above has two connectors suitable for both cables:
CABLE-IDC40-15cm – 15cm 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

2.8.1 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 only by using the touch component over the display.

2.8.2 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 LXDE 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

2.9 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 to 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 very good document on GPIO usage might be found here:
http://www.py6zgp.com/download/A20-GPIO.pdf – the document was created by Dr. Guido Pelz.

2.10 I2C and SPI under Debian

I2C and SPI are both supported in the latest Debian releases. There is respective kernel support for both. There is a python module called pyA20 might be found here:
https://pypi.python.org/pypi/pyA20

At the same web address you would also find a set of examples on how module is used.

2.11 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-MICRO.

We usually try to provide details on how to build the Linux and the Android images at our wordpress page: http://olimex.wordpress.com/.

Another useful place is the Olimex forums where a lot of people share their experience and advice: https://www.olimex.com/forum/

Additional Android and Linux support and features are added overtime. The Linux support is a work-in-progress and you should not expect full Linux support after the initial volume of such boards have become available on the market. If you are in a hurry consider the older OLinuXino designs (which have almost everything supported, have examples available and so on).

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-MICRO 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-MICRO board itself.

3.1 Layout (top view)

The picture below shows the initial revision of A20-OLinuXino-MICRO. Please note that the NAND memory is present only in the 4GB version of the board.
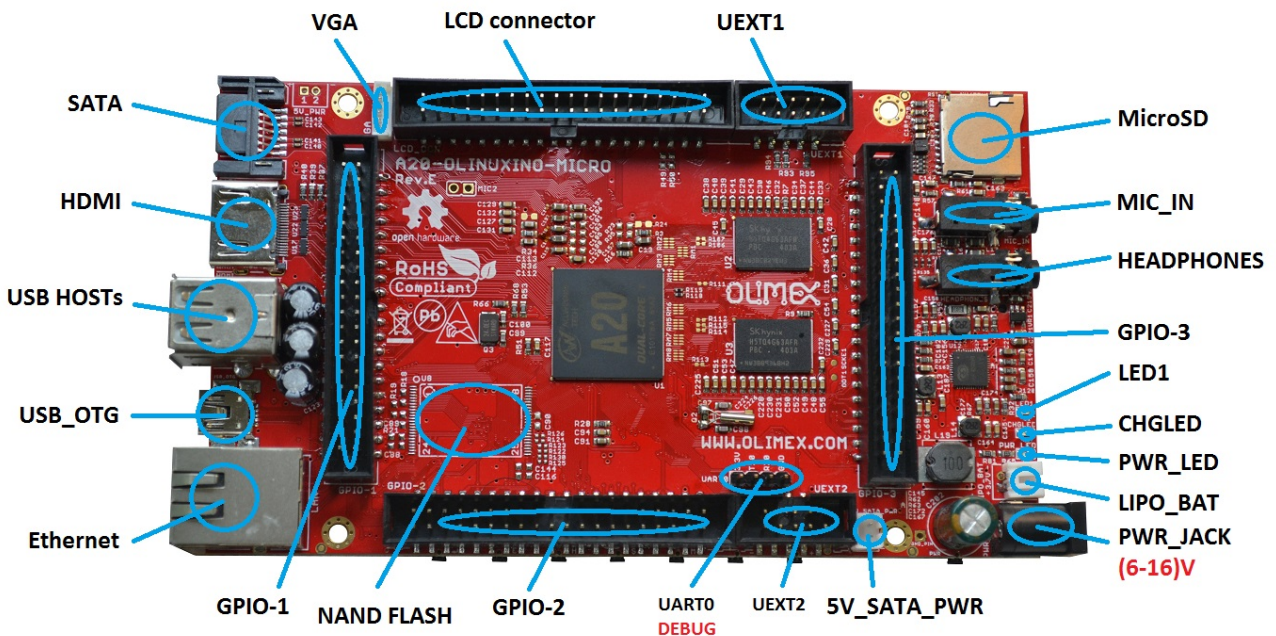
3.2 Layout (bottom view)

At the bottom of the board there are mainly buttons and the large SD/MMC connector.
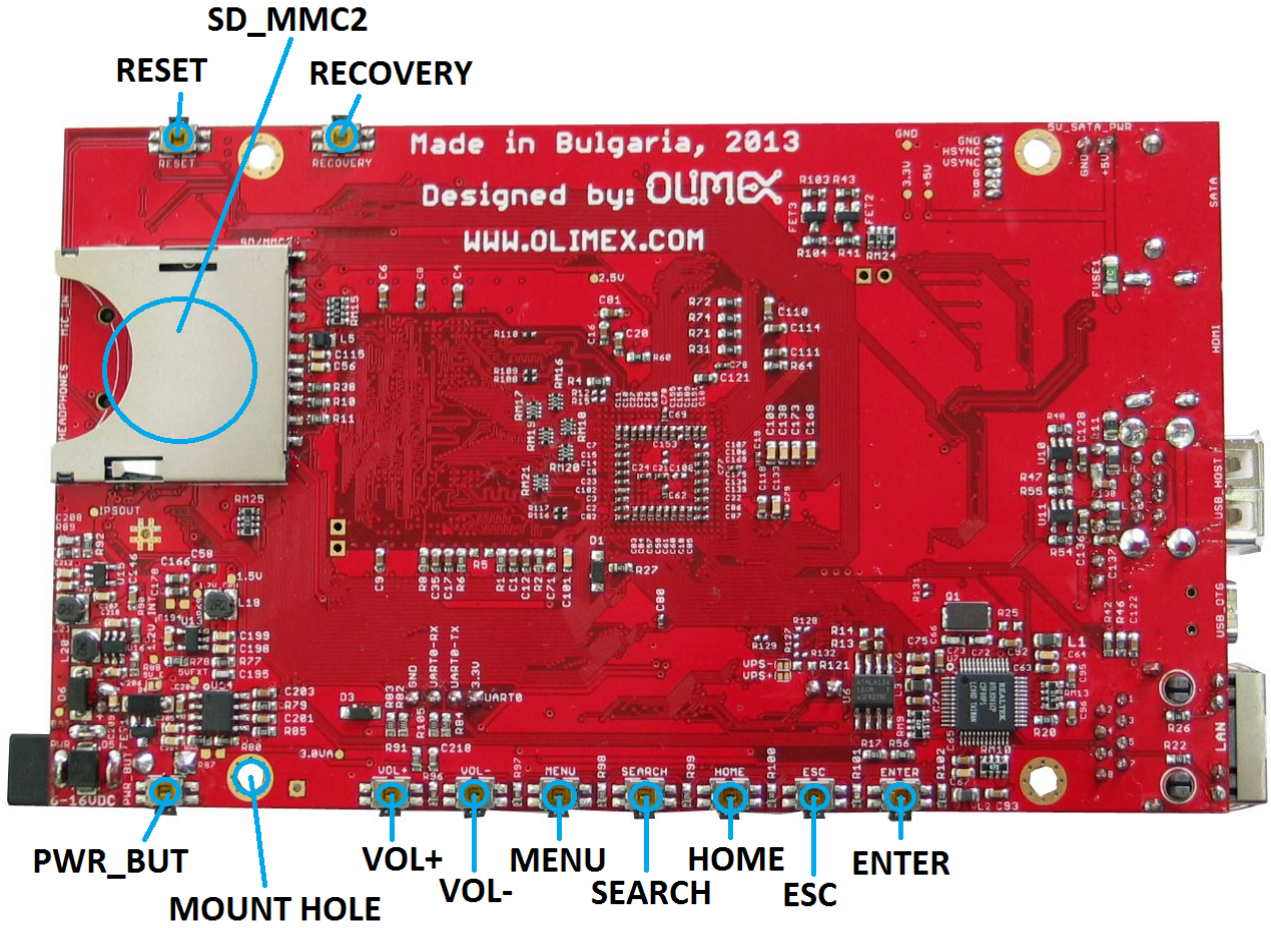
CHAPTER 4: THE ALLWINNER A20 EMBEDDED PROCESSOR

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 main feature of the A20 processor is the sheer computing power that allows FullHD video playback. The graphical processing unit is also pretty powerful and supported by the default software packages that come with the OlinuXino boards. The software support for the features in the processor is at pretty good state thanks to the efforts of the community and Allwinner themselves.

The full list of features might be found below:

- 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

4.2 Block diagram

The block diagram is taken from Allwinner's web-site.
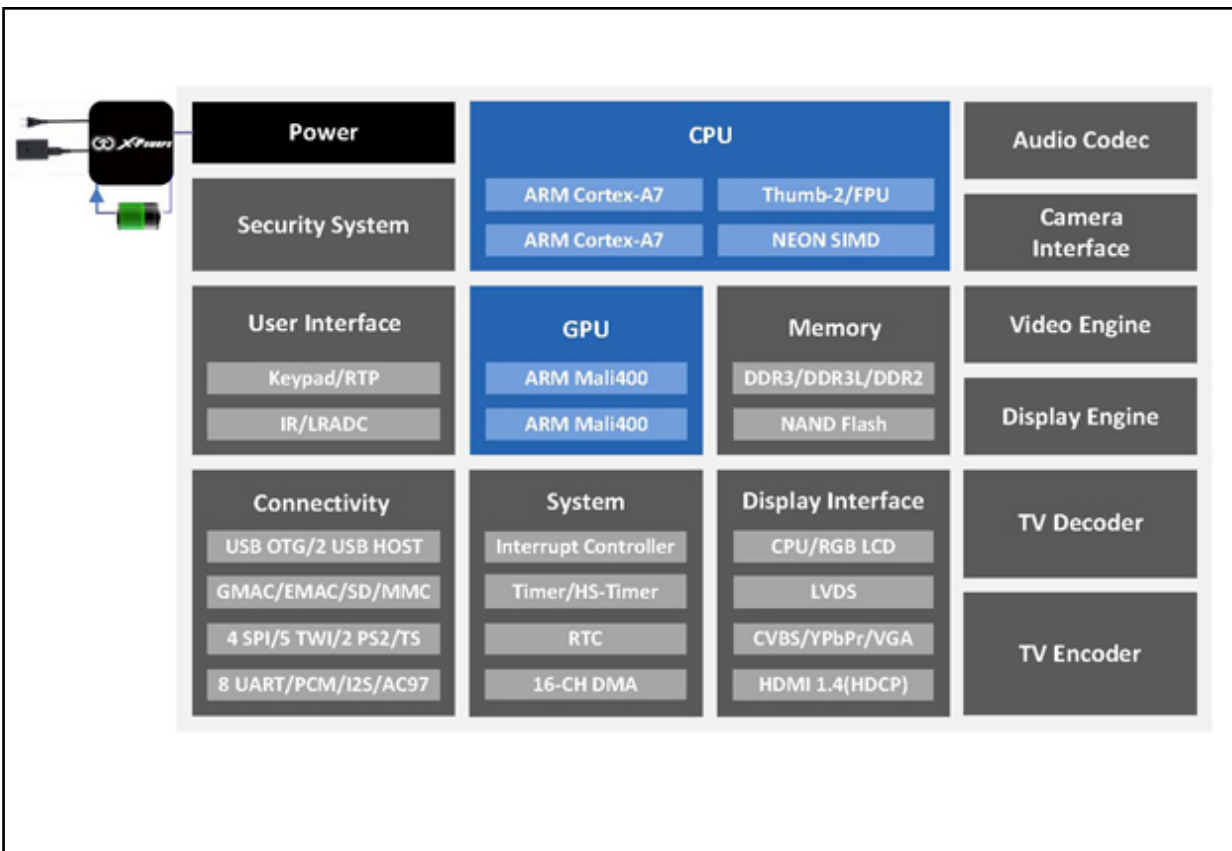
CHAPTER 5: CONTROL CIRCUITY

5. Introduction to the chapter

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

5.1 Reset

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

The board should be turned off the standard OS menu (that might be invoked by holding POWER button or "poweroff" command under Debian) and after the choice is confirmed it is safe to be disconnected from the power supply unit.

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-OLinuXino-MICRO requires input supply of 6-16V. The minimum wattage is 5W, and this threshold may raise if using a lot of devices on the USB-HOST (via external hub), a lot of GPIOs and the LCD_CON.

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 Communication with the A20

The direct communication method is via the serial interface. Such interface is the UART0 connector (five male pins pointing towards the edge of the board) capable of delivering some information on the COM port of your computer – and then use your favorite terminal program (puTTy, teraterm, etc) to receive the data/send commands. You can use USB-SERIAL-CABLE-F to interface UART0 – that allows you to debug the board with a personal computer with a free USB port.

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!

It is highly recommended to own a USB-SERIAL-CABLE-F (or similar product) at hand when debugging – the video output is not always reliable and if you set wrong display settings you might be unable to recover the settings without a proper UART0 connection.

The more indirect ways of communicating with the board are via peripheral devices – mouse and keyboard, via a touch screen LCD that is connected on LCD_CON connector and others.

6.1.1 USB-OTG communication (NAND firmware repair/update)

The main way of changing the firmware image located on the NAND of A20-OLinuXino-MICRO is via the USB-OTG connector.

We have configured an Android image with settings suitable for A20-OLinuXino-MICRO. You can upload it to the board with PhoenixSuit software 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-MICRO.

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 for A20).

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 A20 wiki article).

3. Disconnect the power supply and USB cable from the A20 board.
4. Press and hold RECOVERY button, apply power supply (6-16)V, 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:
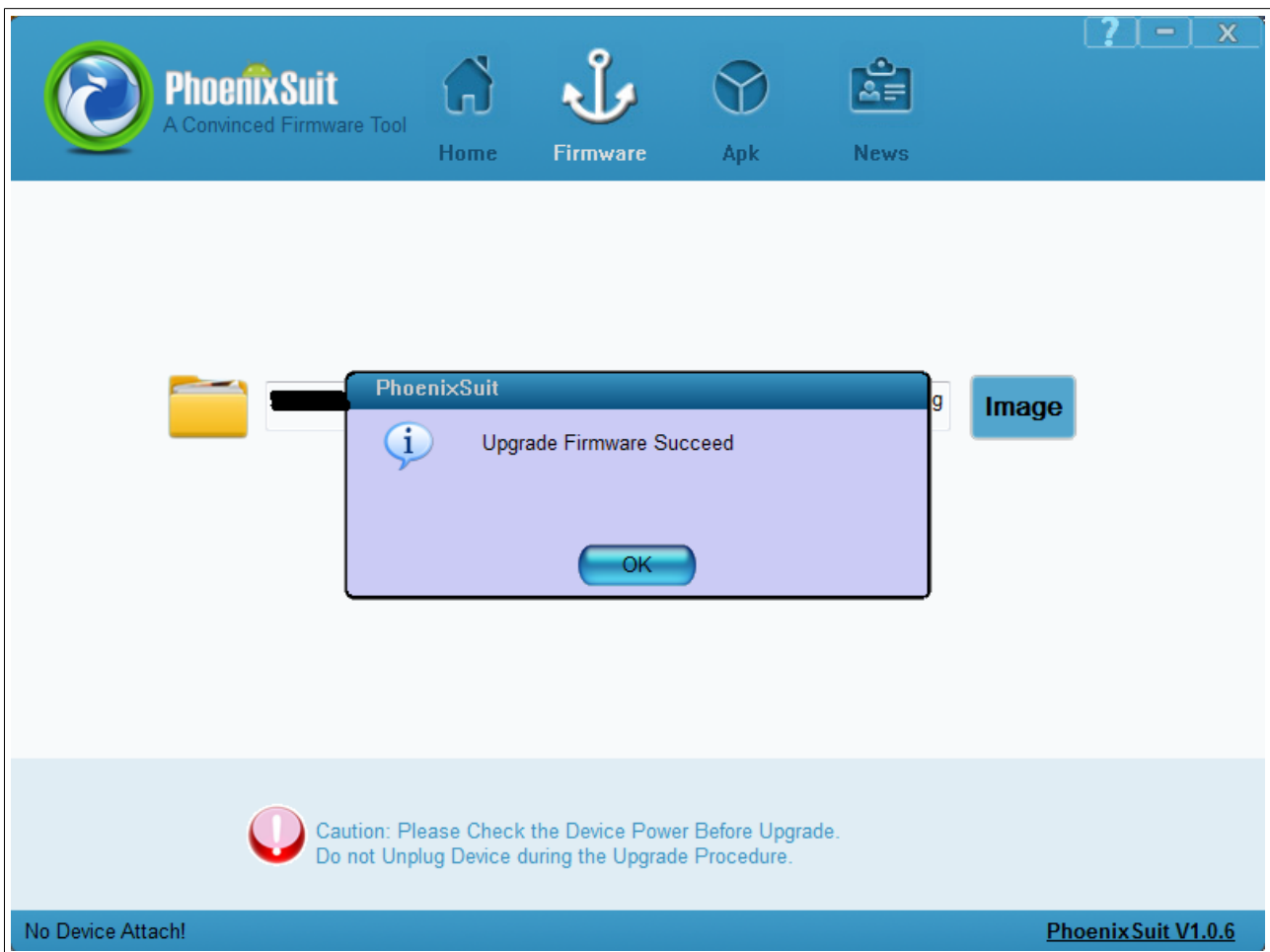
There are different flavors of the Android and Linux distributions depending on whether you want to use 50Hz HDMI or 60Hz HDMI or VGA or 7" display or 10" display etc. You might need to change the configuration files inside (depending on your desirable output medium). How to do it is explained in chapter 2.6 Calibrating a display.

Those images, the software required and further instructions might be found at the wiki page: https://www.olimex.com/wiki/A20-OLinuXino-MICRO.

6.1.2 UART0, UEXT1, UEXT2 interface

The UART0 might be used for UART communication by default. It is typically used for serial/terminal debug. The UEXTs are typically used for connecting additional modules.

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. UEXT1 and UEXT2 are NOT defined by default.

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	3.3V	-
2	UART0-TX	A7
3	UART0-RX	B7
4	GND	-

UEXT1			
Pin #	Signal name	Pin #	Signal name
1	3.3V	2	GND
3	UART6-TX	4	UART6-RX
5	TWI2-SCK	6	TWI2-SDA
7	SPI2-MISO	8	SPI2-MOSI
9	SPI2-CLK	10	SPI2-CS0

UEXT2			
Pin #	Signal name	Pin #	Signal name
1	3.3V	2	GND
3	UART7-TX	4	UART7-RX
5	TWI1-SCK	6	TWI1-SDA
7	SPI1-MISO	8	SPI1-MOSI
9	SPI1-CLK	10	SPI1-CS0

The UEXT connectors are used as an interface for the Olimex's UEXT modules (usually boards with MOD- prefix in their commercial names). For more information on UEXT please visit:
https://www.olimex.com/Products/Modules/UEXT/resources/UEXT.pdf

6.2 SD card connectors

There are two SD card connectors on this A20 board. The now common microSD card connector is located on the top of the board, near the audio in/out connectors.

Note that by default only the micro SD card slot might be used for booting the OS. However, due to popular demand, there is also an SD/MMC card connector located on the bottom of the board.

Both SD slots might be used simultaneously – there is no data multiplexing lines. For instance – one of them might be used for booting the OS, the other – for saving data.

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-MICRO.

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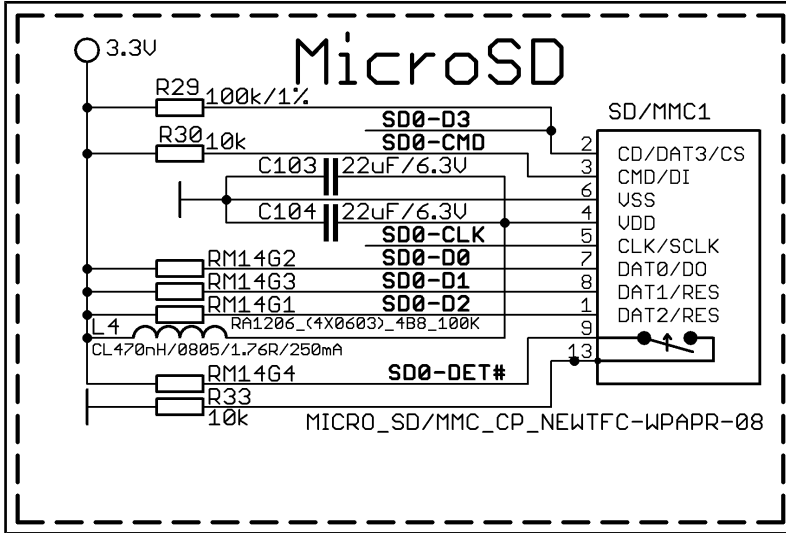
SD/MMC1 slot is the "smaller" microSD card slot, located on the top of the board.

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.

microSD card 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	-
4	VDD	-
5	CLK/SCLK	SD0-CLK(L20)
6	VSS2	-
7	DAT0/DO	SD0-D0(M19)
8	DAT1/RES	SD0-D1(M20)

6.2.2 SD/MMC slot

The schematic related to the SD/MMC1 (SD/MMC connector) is shown below:
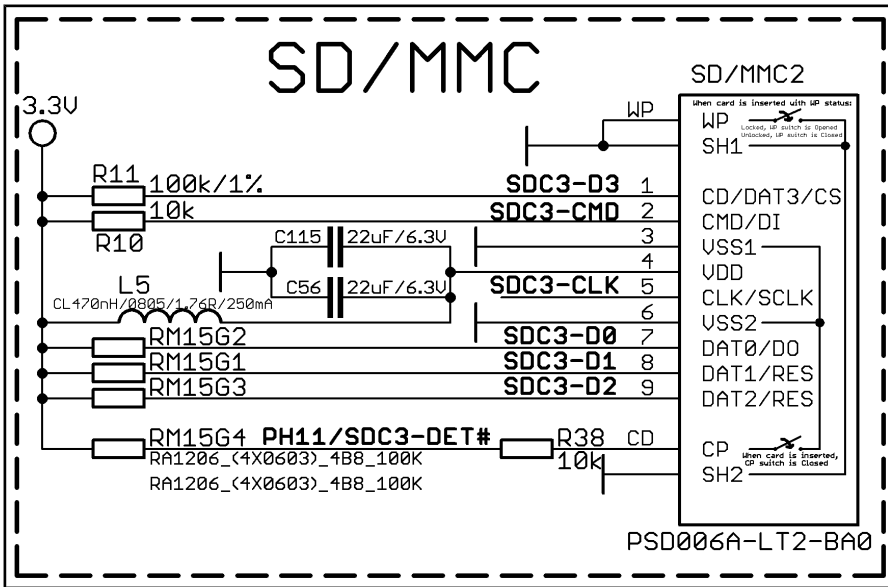
The SD/MMC2 slot is located on the back of the PCB and works with the "larger" SD and MMC cards. The connector was placed by a popular demand.

Note that by default this connector is not suitable for booting OS. Such support is possible but not implemented in the official software releases.

SD/MMC card connector		
Pin #	Connector signal name	Wire name (processor pin)
1	CD/DAT3/CS	SDC3-D3 (B16)
2	CMD/DI	SDC3-CMD (A18)
3	VSS1	-
4	VDD	-
5	CLK/SCLK	SDC3-CLK (B18)
6	VSS2	-
7	DAT0/DO	SDC1-D0 (A17)
8	DAT1/RES	SDC1-D1 (B17)
9	DAT2/RES	SDC1-D2 (A16)

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 one used by Olimex in most of our products – the DC barrel jack has 2.0mm inner pin and 6.3mm hole. More information about the exact component might be found here: https://www.olimex.com/wiki/PWRJACK.

You should provide 6 to 16 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 500mA of current.

Pin #	Signal name
1	Power input
2	GND
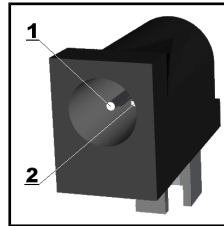
More info about the power supply can be found in chapter 5 of this manual.

6.4 MIC_IN & HEADPHONES connectors

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

Standard MIC_IN jack is mounted for audio input to the board. Note that it is a single channel

MIC_IN. The MIC_IN is a mono input.

MIC_IN connector		
Pin #	Signal name	Processor pin
2	MICIN	AC20
3	MICIN	AC20
5	GND	-

The socket can interface standard 3.5 mm phone connector (also known stereo plug or audio plug).

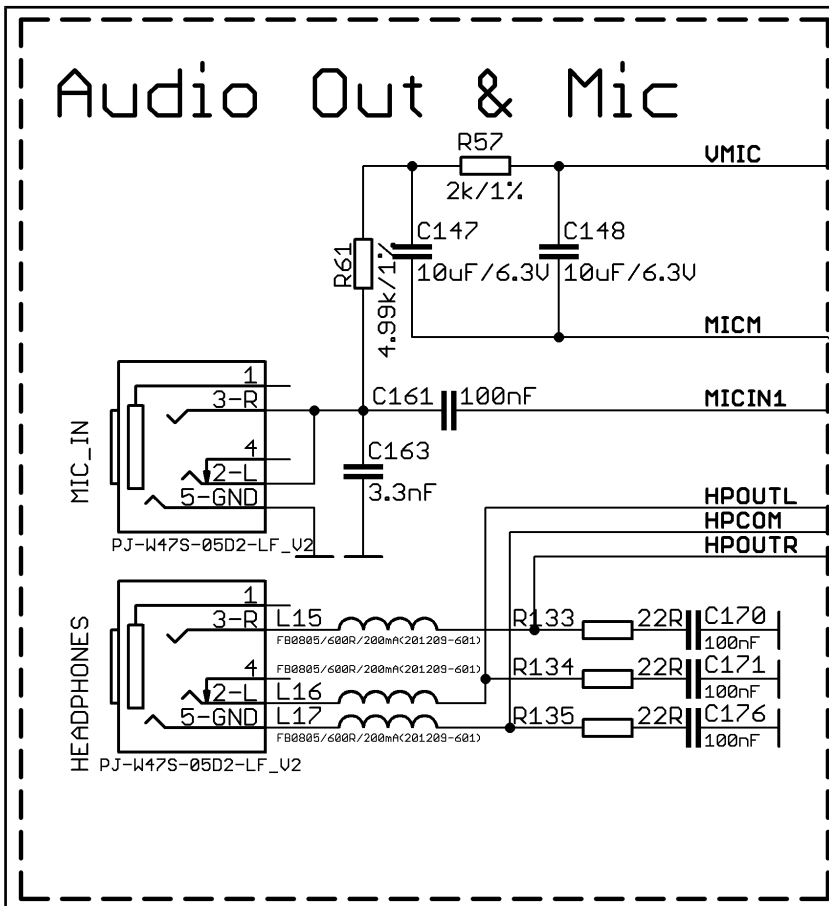| Headphones connector		
Pin #	Signal name	Processor pin
2	HPOUTL	Y19
3	HPOUTR	W19
5	HPCOM (GND)	AA19, AA20
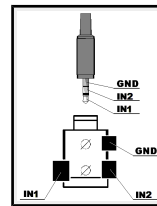
The connector can interface standard 3.5 mm phone connector (also known stereo plug or audio plug).

The default audio output is set to the HDMI connector (for displays and monitors with built-in audio speakers). If you wish to use audio out you need to first disconnect the HDMI, boot the board, connect the audio out device (speakers) and finally connect the HDMI.

6.5 USB_OTG connector

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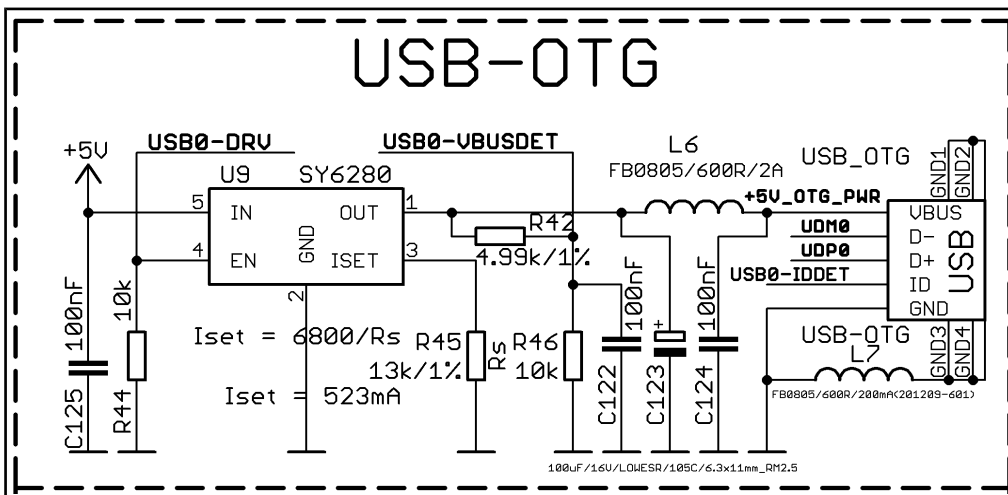
The USB_OTG features Low Loss Power Distribution Switch SY6280 which protects the board in case the devices you have plugged to the USB_OTG attempt to draw more current than 523mA combined. The maximum current available on the 5V USB_OTG is exactly 523mA.

The SY6280 responsible for the USB_OTG is enabled by USB0-DRV (processor pin C12), thus the USB_OTG is also controlled by the same signal.

There are two parts of a successful OTG device/host mode switch and usage – hardware part and software part.

The hardware part is the cable – to use the OTG as a host you would need a mini USB to USB adapter cable. Do not confuse micro and mini USB!

The software part is loading a kernel module responsible for the behavior of the board while in device mode. If such software is not enabled or missing in the official distribution – you would need, respectively, either to load the module or rebuild the kernel to enable the module. There are several modules that you might need depending on what your goals are.

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.

6.6 USB_HOST connector

The part of the schematic related to the USB_HOST connector is listed below:
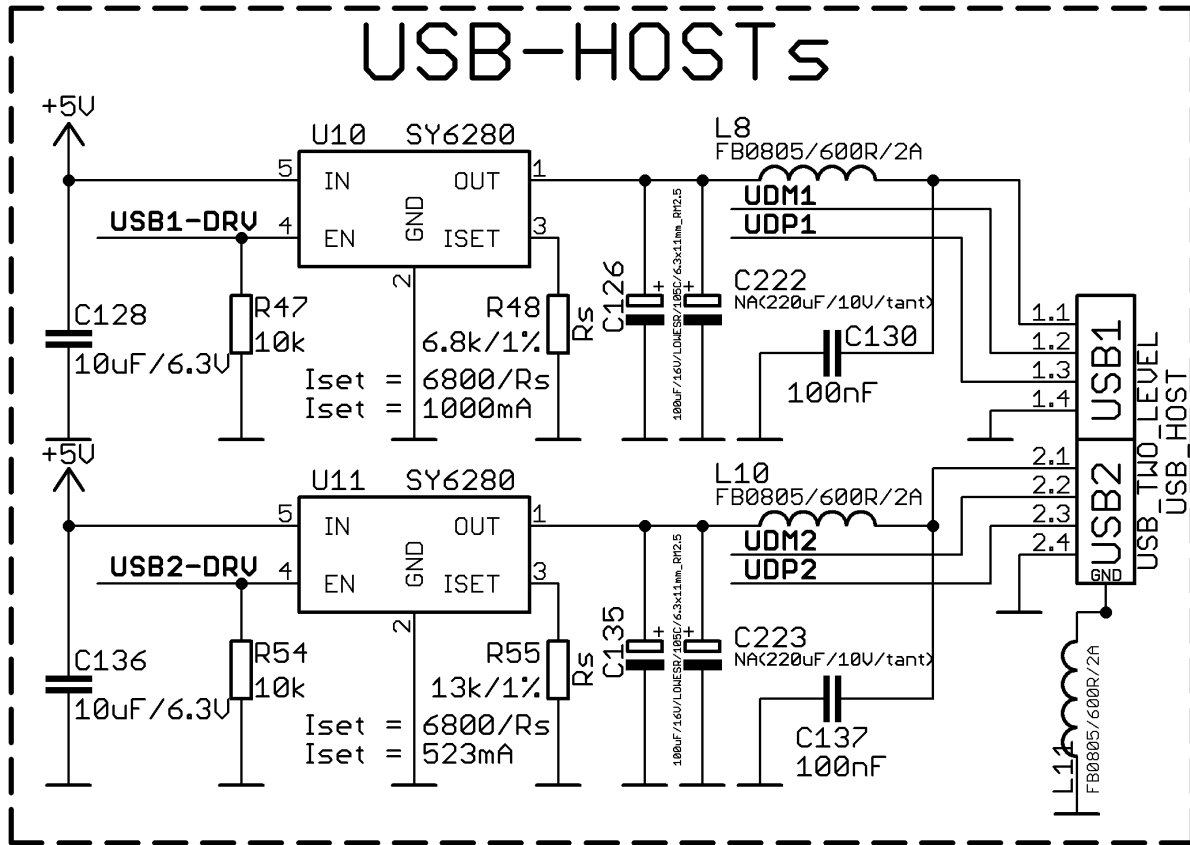
The USB_HOST connector features two levels – called USB1 and USB2 in the schematic. The connector is situated between the USB_OTG and the HDMI connector. Each of slots them features a low loss power distribution switch SY6280 which protects the board in case the devices you have plugged to each level USB_HOST try to draw more than the allowed current.

Please note the pads of capacitors C222 and C223 where you can solder tantalum capacitors (220uF/18V). If you do so you would be able to increase the stability of the powering by the USB ports when the powered device has high current consumption.

The maximum current available on each slot of the 5V USB_HOST is different! USB1 is capable of providing 1000mA of current; while USB2 is capable of providing only 523mA. If you have a device that draws higher current please plug it into USB1.

The SY6280 responsible for the USB1 is enabled by USB0-DRV1 (processor pin A4).

The SY6280 responsible for the USB2 is enabled by USB0-DRV2 (processor pin A5).

USB_HOST two-level connector					
Pin #	Signal name	Processor pin	Pin #	Signal name	Processor pin
1.1	5V	Connected to SY628	2.1	5V	Connected to SY628
1.2	UDM1	P20	2.2	UDM2	R20
1.3	UDP1	P21	2.3	UDP2	R21
1.4	GND	-	2.4	GND	-

The GND is common for both levels of the USB_HOST.

6.7 Ethernet

In different hardware revisions of the board we have used a number of different megabit Ethernet controllers. This is why it is important to test with the latest official image first.

In hardware revision J of the board we switched to Microchip's LAN8710AI-EZK(QFN-32). Note that the industrial version of the board uses industrial version of the chip. Make sure to refer to the hardware schematic for exact information about the chip used.

If you have problems with the software support for the Ethernet in Linux, consider checking our latest build instructions and patches. At the time of writing, a patch to get the Ethernet working can be found here: https://github.com/OLIMEX/OLINUXINO/blob/master/SOFTWARE/A20/A20-build-3.4.103-release-5/a20-phy_1000_100-dram.patch

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:

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

Important: You can configure static/DHCP-given IP addresses in /etc/network/interfaces
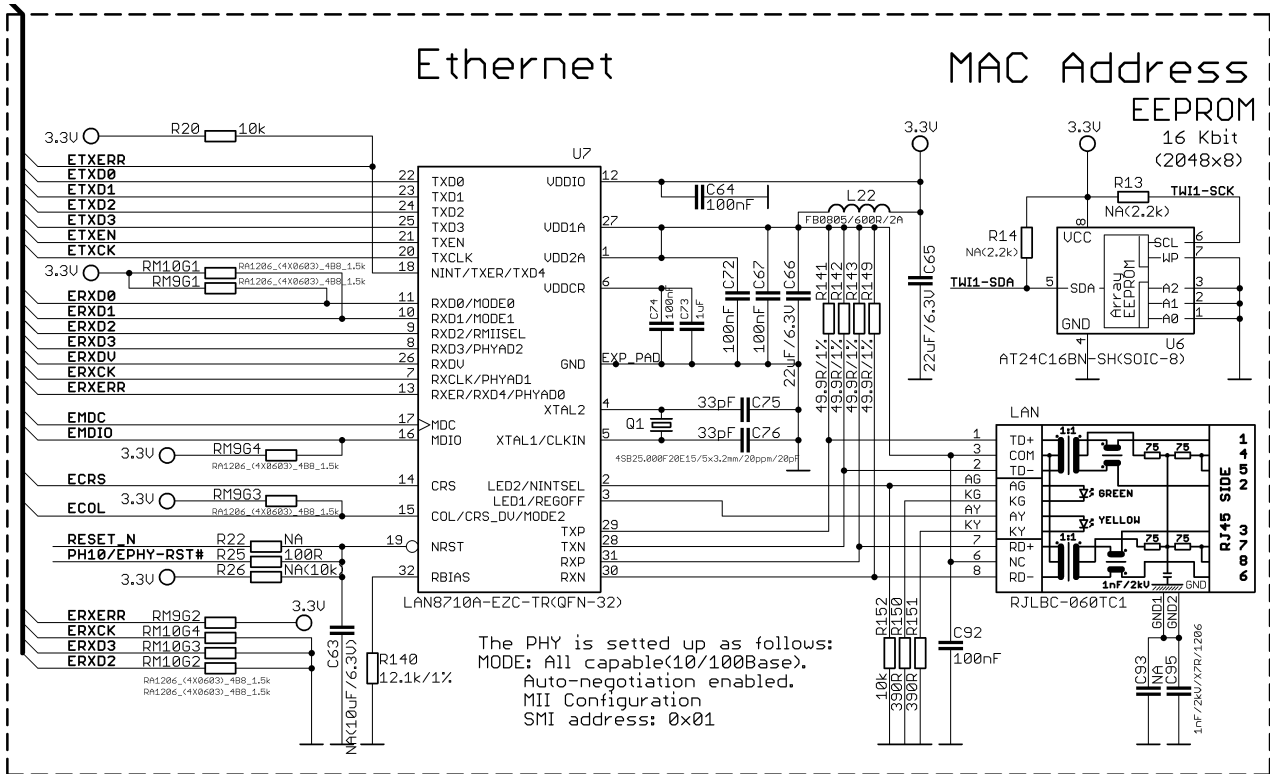
Ethernet	
Pin #	Signal name
1	TD+
2	TD-
3	VDD
4	NC
5	NC
6	VDD
7	RD+
8	RD-
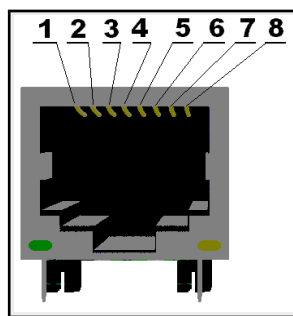
LED	Color	Usage
Right	Green	Link status
Left	Yellow	Activity status

6.8 HDMI connector

The part of the schematic that describes the HDMI module is shown below:
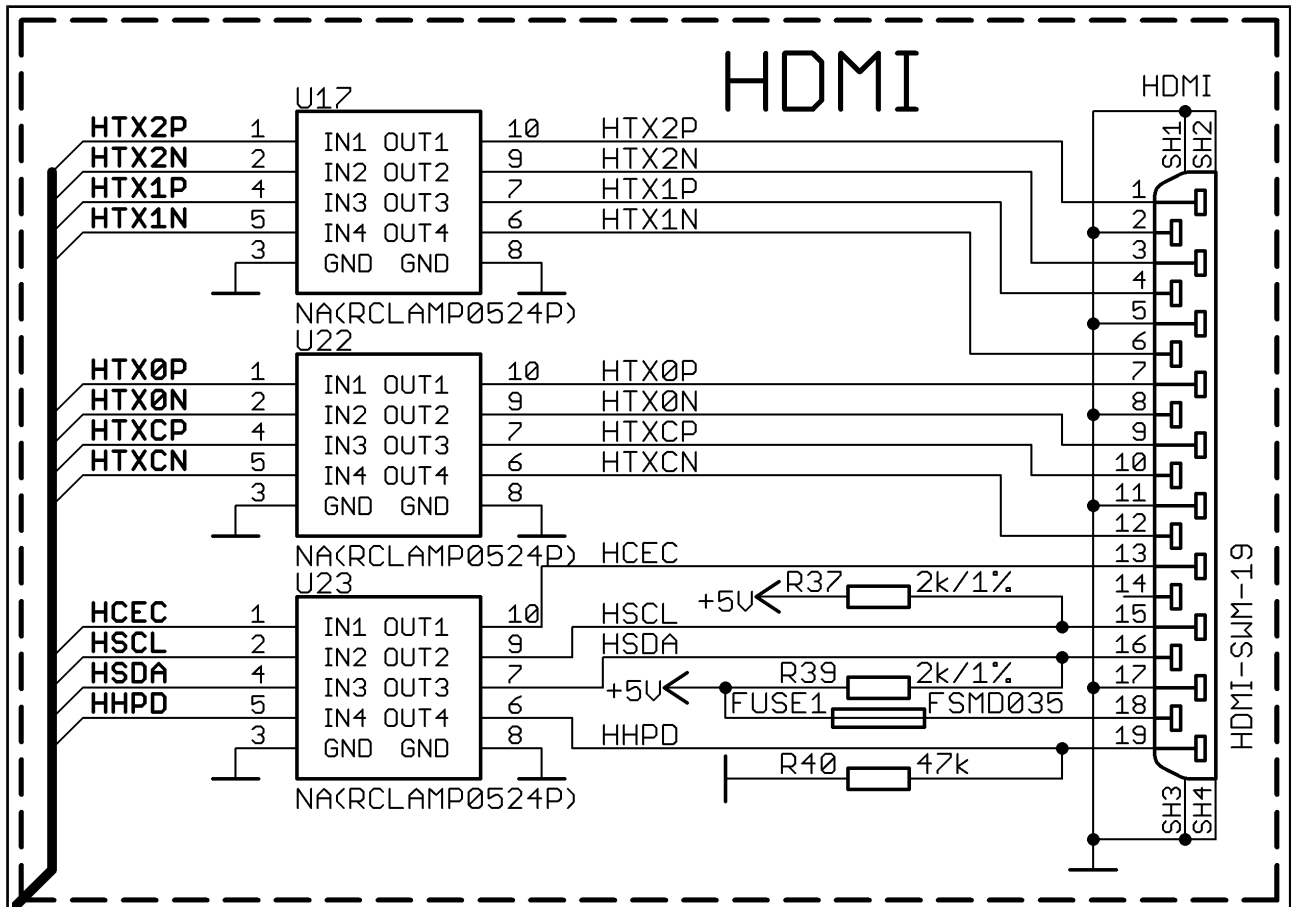
Note that there are different Linux images depending whether the HDMI TV works at 50Hz or 60Hz refresh rate. If you board runs Android there is a specific option to set the appropriate HDMI output.

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.

6.9 VGA connector

You would probably need an adapter cable for the VGA display (6-pin connector to 15-pin female RGB cable). You can make the cable or the connection yourself or you can purchase the Olimex-made cable: https://www.olimex.com/Products/Components/Cables/A20-VGA-CABLE/.Note that you also have to change the script under Debian as explained in chapter 2.6 "Calibrating a display".

Please note that you would need a specific Android image for the VGA output (with proper display and output settings). **We don't provide such image ready at the moment!**

The VGA output has been tested only with the official Debian Linux image so far and it works fine (after selecting the proper mode by executing ./change_display*.sh).

VGA 6-pin connector			VGA 15-pin connector (back numbering)	
Connector pin #	Signal name	Processor pin	Connector pin # (back numbering)	Color code (according to the image below)
1	VGA-R	AC17	1	RED
2	VGA-B	AB16	3	BLUE
3	VGA-G	AC16	2	GREEN
4	VGA_VSYNC	AC9	14	BLACK
5	VGA_HSYNC	AB9	13	BROWN
6	GND	-	6	PINK

Back view of the 15 pin connector
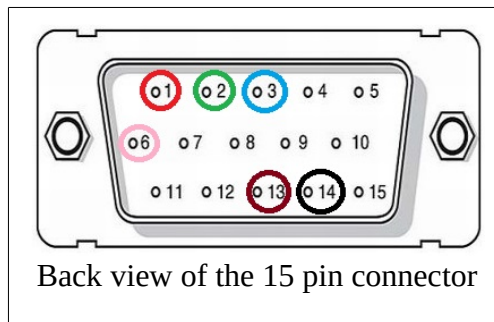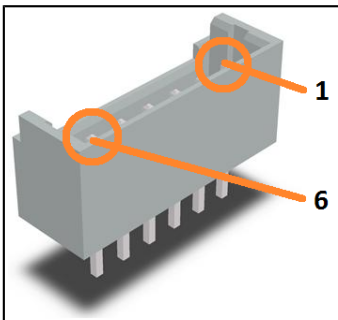
6.10 SATA connector and power

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	-
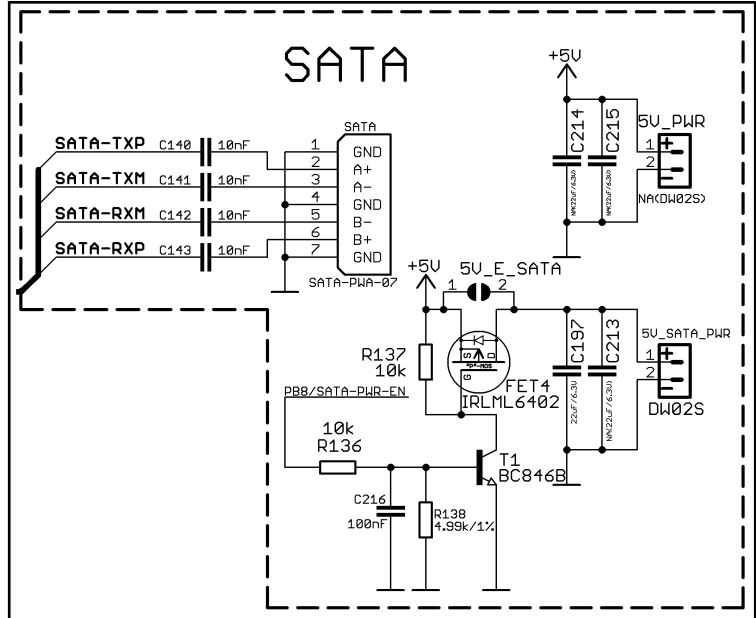
6.11 GPIO ports

There are three 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.

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 "2.8 GPIO under Debian" and "2.9 I2C and SPI under Debian".

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

The GPIO-1 plastic connector has 14 pins. 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	VOL+_BUT	AB23***
7	PG1	E21	8	VOL-_BUT	AB23***
9	PG2	E20	10	MENU_BUT	AB23***
11	PG3	D21	12	SEARCH_BUT	AB23***
13	PG4	D20	14	HOME_BUT	AB23***
15	PG5	C21	16	ESC_BUT	AB23***
17	PG6	E19	18	ENTER_BUT	AB23***
19	PG7	C20	20	TVOUT3	AB17
21	PG8	D19	22	TVIN0	AC18
23	PG9	C19	24	TVIN1	AB18
25	PG10	D18	26	TVIN2	AA17
27	PG11	C18	28	TVIN3	Y17
29	VGA_HSYNC	AB9*	30	LINEINR	AB21
31	VGA_VSYNC	AC9*	32	LINEINL	AB20
33	VGA-G	AC16*	34	FMINR	Y21
35	VGA-B	AB16*	36	FMINL	Y20
37	VGA-R	AC17*	38	MIC1OUTP	AC22
39	LRADC1	AB22*	40	MIC1OUTN	AC23

*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).*
**It was erroneously named GPIO-4 in the first release of the board.*
***All Android buttons are connected to an ADC (LRADC) that uses a divider to determine which button is pressed.*

The yellow part of the table shows all the signals shared with the 6-pin VGA connector.

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

The GPIO pins are led out on a separate 40pin connecter. 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.
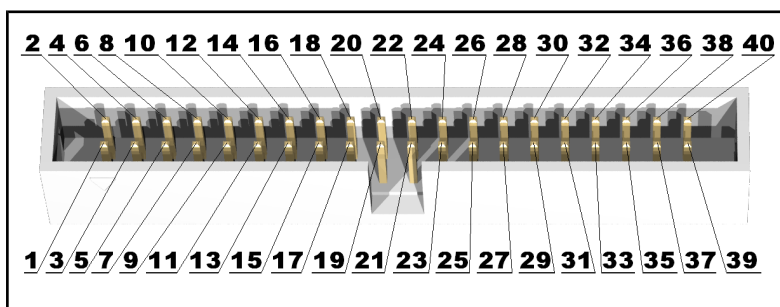
GPIO-2 connector					
GPIO pin#	Signal name	Processor pin#	GPIO pin#	Signal name	Processor pin#
1	5V	-	2	GND	-
3	3.3V	-	4	GND	-
5	TWI0-SCK	A15	6	PE0	E23
7	TWI0-SDA	B15	8	PE1	E22
9	PI0	A20	10	PE2	D23
11	PI1	B20	12	PE3	D22
13	PI2	A19	14	PE4	C23
15	PI3	B19	16	PE5	C22
17	PI10	C17	18	PE6	B23
19	PI11	D17	20	PE7	B22
21	PC3/NCE1	L22*	22	PE8	A23
23	PC7/NRB1	J22*	24	PE9	A22
25	PC16	M21	26	PE10	B21
27	PC17	F23	28	PE11	A21
29	PC18	F22	30	PI14	C15
31	PC23	G19	32	PI15	D15
33	PC24/NQS	F21*	34	UART0-RX	B7*
35	GPIO1	-	36	UART0-TX	A7*
37	GPIO2	-	38	NMI_N	F5
39	GPIO3	-	40	RESET_N	C14

*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).*

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

GPIO-3 connector features the signals of AXP152 on a convenient header. The signals available might be used to implement power controls on the board – for instance – turning off the device at specific voltage level or detecting levels.

The JTAG signals might also be accessed at this header.

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	PH0	A6	6	PB3	B14
7	PH2/LED	C6	8	PB4	A13
9	PH7	B4	10	PB5	B13
11	PH9	D4	12	PB6	A12
13	PH10	A3	14	PB7	B12
15	PH11/SDC3-DET#	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	A10
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

*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).*

The yellow part of the table shows the JTAG signals.

These signals might be conveniently accessed on GPIO-3 connector. These JTAG signals might be found at processor port PB. The pins of the connector are respectively GPIO-3 #26; GPIO-3 #28; GPIO-3 #30; GPIO-4 #32. More specifically:

PB14 – MS0 (TMS) - GPIO-3 #26
PB15 – CK0 (TCK) - GPIO-3 #28
PB16 – DO0 (TDO) - GPIO-3 #30
PB17 – DI0 (TDI) - GPIO-4 #32

You would also need Vdd and GND which might be found on various places on the board. This information should be sufficient to establish a hardware connection. The connection would depend

on the debugger you have and the type of its connector (20-pin JTAG, 10-pin JTAG, etc).

Software-wise – the JTAG should be enabled by default but you might also edit the script.bin/fex file that contains special JTAG part. Please check here: http://linux-sunxi.org/Fex_Guide#.5Bjtag_para.5D; the [jtag_para] part.

The real issue would be finding piece of software that has Allwinner A20 support. The sunxi community has some experimental configuration files for A10 and A13 boards I believe, but they were never tested by an Olimex employee.

Furthermore, we have no experience with the JTAG debug, we have not performed any JTAG debug sessions ourselves.

6.12 LCD_CON 40pin connector

The LCD_CON pins are led out on a separate 40pin connecter 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.
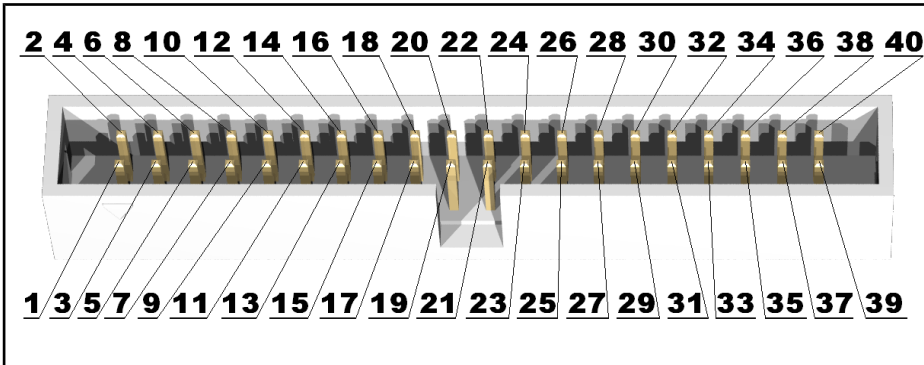
LCD_CON connector					
GPIO pin#	Signal name	Processor pin	GPIO pin#	Signal name	Processor pin
1	+5V	-	2	GND	-
3	+3.3V	-	4	GND	-
5	LCD_D16	Y12	6	LCD_D17	AA12
7	LCD_D18	Y11	8	LCD_D19	AA11
9	LCD_D20	Y10	10	LCD_D21	AA10
11	LCD_D22	AB12	12	LCD_D23	AC10
13	LCD_D8	AB11	14	LCD_D9	AC11
15	LCD_D10	Y15	16	LCD_D11	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).*
**Pins 33 and 34 are disconnected by default. You need to place the two resistors marked as NA to establish connection to processor's PB3 and PB4.

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.**

IMPORTANT:

You need to upload new Android image to the board. These images are available here: https://www.olimex.com/wiki/A20-OLinuXino-MICRO (using an operating system – NAND flash). In Debian Linux you would need to start a start a shell script to change the built-in the image resolution settings. Start it with ./change_display_A20_OLinuXino.sh and follow the instructions. Please refer to chapter 2.5 "Changing the default image resolution" for more details.

6.13 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	PTH	UBOOT	Changes the functionality of the BOOT/REC button – when in UBOOT position the button is used to enter bootloader mode (to connect via USB_OTG). When in REC
VPS-	SMT	Open	Reserved for future use of different types of NAND memories.
VPS+	SMT	Open	Reserved for future use of different types of NAND memories.

6.14 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

1GB = 2×[4Gb(256M x 16b)] DDR3 SDRAM – the exact memories varies between the hardware revisions of the board (also the -IND industrial version uses industrial memories). Refer to the schematic of your board for details on the exact memory.

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 8b)] eMMC FLASH – only available on the 4GB version of the board; the exact memory used varies between hardware revisions of the board. Refer to the schematic of your board for details on the exact memory.

PWR_LED – turns on upon powering the board
LED1 – GPIO button
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-MICRO.

7.1 Eagle schematic

OLinuXino schematics may be found it on the OLinuXino's GitHub repository: https://github.com/OLIMEX/OLINUXINO/tree/master/HARDWARE/A20-OLinuXino-MICRO. You can download the whole repository as .zip without having a GitHub account.

Please note that the schematics of A10 (not to be confused with A10s) and A20 boards are the same since the two Allwinner processors are pin-to-pin compatible (drop-in replaceable). This means that schematics for older hardware revisions of A20-OLinuXino-MICRO can be found in the GitHub folder for A10-OLinuXino-MICRO.

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.
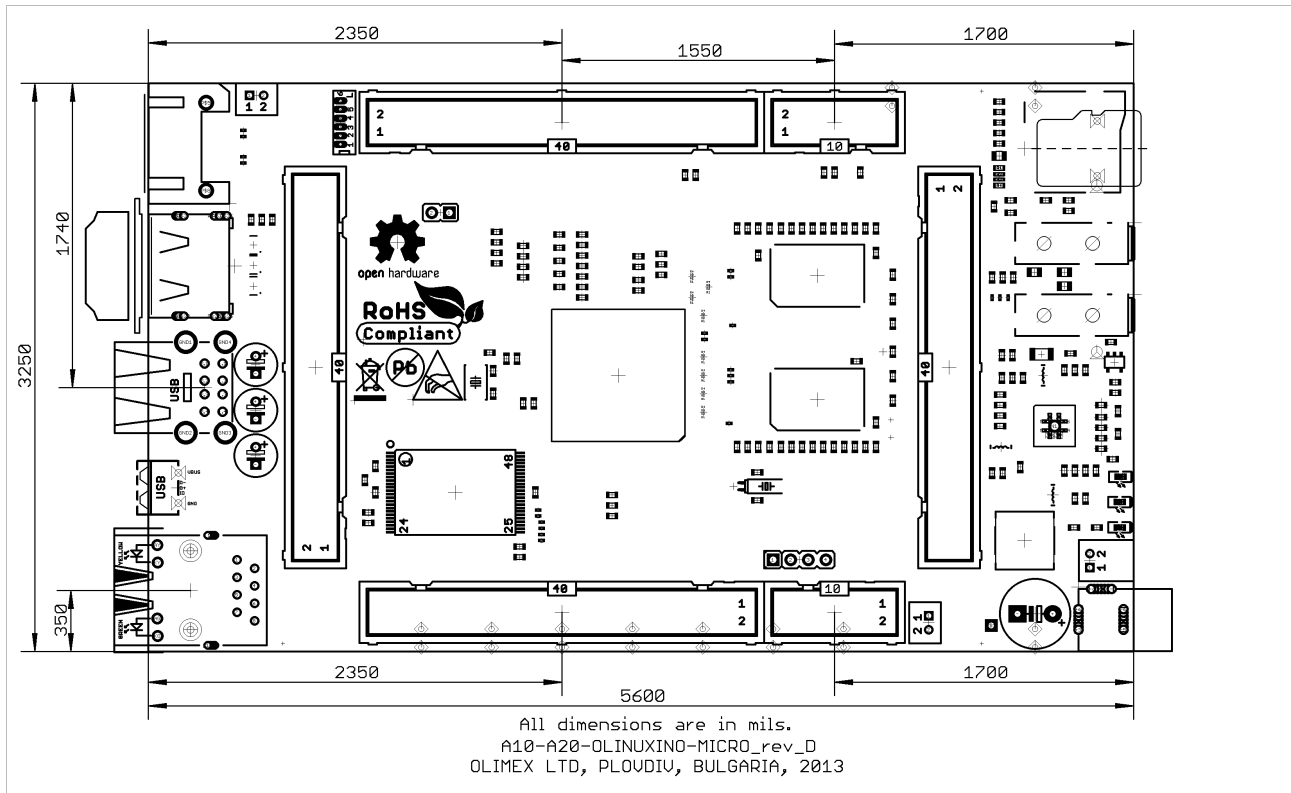
The three highest elements on the board in order from the tallest to the shortest are: USB_HOST connector – 700mils, capacitors C202 – 650mils, the Ethernet connector – 600mils.

The heights listed above include the PCB height.

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

Revision, date	Notable changes	Modified page
A, 30.08.13	Initial manual release	All
B, 16.09.13	1. Quartz values error fixed 2. Updated the board revision changes	15 16
C, 18.09.13	1. Added new additional products 2. Added more information about the SD card booting	9 43
D, 30.09.13	1. Added few suggestions for USB boot 2. Added more information about power saving mode before complete Android booting	9
E, 02.10.13	1. Added more information about display setting 2. Added chapter about the VGA 3. Added FAQ section	11 29 42
F, 16.10.13	Added more information about A20-VGA-CABLE	Multiple pages
G, 24.10.13	1. Added more information about SSH connection 2. Added more information about VGA output requirements 3. Improved the FAQ section	27 29 41
H, 27.05.14	1. Added more information about USB host ports 2. Added new information for latest board revisions	10 41

Revision, date	Notable changes	Modified page
I, 30.05.14	1. Updated pictures with revision E 2. Updated document information to reflect the changes in the board 3. Adjusted several paragraphs and tables	1, 14, 15 26, 27 Multiple pages
J, 03.11.14	1. Expanded GPIO-3 chapter by adding JTAG information 2. Added GPIO, I2C and SPI sections 3. Added information about newer Olimex A20 boards boards 4. Added board revision G changes 5. Fixed few spelling mistakes	3, 13, 14, 34, 35, 36, 37, 44
K, 30.01.2015	1. Updated list of suggested displays 2. Fixed error about the power jack	8, 27, 46
L, 09.02.2015	Fixed information about the CMOS-TTL convertor	21
M, 10.02.2015	Added chapter "2.6 Interacting with the board"	11
N, 16.02.2015	1. Information about the Windows driver for the tethering 2. Clarified how to prepare a microSD card with Android OS	12 52
O, 16.03.2015	Updated HDMI parts to reflect the latest schematic	5, 36
P, 23.12.2015	1. Clarified that we no longer load Android on the 4GB version of the board; 2. Added information about Debian nandinstall	All
Q, 17.05.2017	Added information about new hardware revisions and versions of the board. Mentioned the change to new Ethernet PHY controller.	All
R, 29.09.2017	Added info about new board variants	6, 7

8.2 Board revision

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

Revision	Notable changes
B	Initial release of the board
C	1. Resistor matrix RM23 gets replaced by: R133,R134,R135 2. Adjusted the package of H5TQ2G63BFR/MEM4G16D3EABG-125 3. R12 and R15 values changed to 0R 4. NAND_E jumper is changed from PTH to SMD closed 5. 5V_SATA_PWR connector (altogether with 2×2uF) is changed from WF2S to DW02S and placed near the PWR jack instead of original position 6. GPIO-4 is now correctly named GPIO-1 7. There is a signal for the power of of the SATA – PB8/SATA-PWR-EN (r136, c216, r138, t1, t137, fet4) 8. A jumper between 5V_E_SATA and 5V_PWR_SATA is added 9. The micro SD card connector is changed
D	Resistor optimizations
E	1. Adjusted and optimized the position of PCB names 2. R16 changed from 237R to 330R, for stability purposes 3. R7, R9 also changed from 237R to 330R, for stability purposes 4. R41 and R104 changed from 1k/1% to 390R for better sync signal and to improve the VGA signal (allowing it to work at FullHD)
F	1. Added 10M resistor in parallel to 32768 crystal else there might be hang-ups 2. Changed to 32768 crystal to SMD one 3. Few libraries updated 4. Added pads of two tantalum capacitors C222 and C223 near the USB-host 5. Positions adjusted – A20, NAND, SPIFLASH, LAN, matrix, capacitors, CARDs and other 6. Added tantalum capacitor 220uF/10V to ISPOUT 7. R48(13k/1%) changed to 6.8k/1% so there would be 1A of current on USB1; USB2 still gets 523mA maximum
G	1. Added a 220uF/tant to SATA PWR 2. C216 changed from 100n to 22uF/6.3V, this improves hard-disk switching on 3. Update of all component libraries. HDMI connector is changed to a new package. 4. Now not placing ESD protection buffers due to hardware difficulties with their placement
H (unreleased)	1. Changed the LAN part of the design, added two capactiors 1nF/2kV/X7R/1206, and 1nF/2kV/X7R/1206_or_1M, also the matrix is now 0402 with 49.9 ohm at 1%. 2. Major polygon changes. 3. Removed HDMI protection circuit(RCLAMP0524P becomes NA). 4. Preparation for memory changes from MEM4G16D3EABG-125 to K4B4G1646D-BCK0. 5. Position of C95 and C96 slightly adjusted, so that they are not under GPIO-1 and LAN. USB_OTG polygon changed. USB_HOST polygon adjusted. Removed few viases. 6. Added few more vias for the VCC. 7. Improved white print. Year and revision updated. Web-site

	address updated. 8. Power jack package updated. 9. SATA package updated.
I (unreleased)	1. eMMC was added. 2. RTL8201CP(LQFP-48) Ethernet chip replaced with LAN8710AI-EZK(QFN-32). Both are MbE PHYs. 3. All libraries were updated. 4. L7, L15, L16 and L17 were changed from FB0805/600R/200mA(201209-601) to FB0805/600R/2A. 5. PWR-JAKPWR_JACK_UNI_MILLING_SMALLER_DRILLS changed to PWRJ-2mm(YDJ-1136). 6. R98 optimized from 10k/1% to 10k. 7. C195 and C208 optimized 22pF to 33pF. 8. R43 and R103 optimized from 51R to 100R. 9. C123, C126 and C135 were changed from 100uF/16V/LOWESR/105C/6.3x11mm_RM2.5 to 2x47uF/6.3V/0805, i.e. C123||C240, C126||C222 and C135||C223 10. D6 was renamed to D2 and changed to the D4's package. 11. U13 and U15 were changed from SY8008 to SY8009AAAC(SOT23-5). 12. ODT1, SCKE1, SCS1/SA15 and ZQ1 were connected to the 2 DDR memories with aim to support bigger sizes if one day A40 is released or A10 is used. 13. All resistor matrices: RM1-RM8 and RM16-RM21(RA0805_(4X0402)_0R(Board_Mounted)) were removed and the wires were shortened. The same thing was done for: R106-R117(0R(Board_Mounted)). 14. The viases that were between GPIO-1 pads 12/14 were removed. 15. The eMMC was placed over the NAND flash. 16. The cream layer was improved. 17. BKUP connector(HN1x2) was added for AXP's power management option. 18. RM1 and RM2 were added to enable CTP support for the LCD connector. 19. MP1482DS was exchanged with MP1584EN(SOIC8E). Because of this, were affected D5, C202, C203 and several other components as well. 20. The files were renamded from 4G to nothing, because the schematic is universal for all boms generated! 21. C217's value was changed from 220uF/10V/TANT to 220uF/10V/TANT_D! 22. Added reset control for the new PHY – C63 becomes NA down from 10uF, R26 becomes NA down from 10K and R25 is now 100R up from NA.
J (released)	1. Rotated RM1 and RM2 at 90 degrees for better soldering. 2. Created separate design for each of the variants of the board (1. commercial temperature range and no flash memory – A20-OLinuXino-MICRO; 2. industrial temperature range and no flash memory – A20-OLinuXino-MICRO-IND; 3. commercial temperature range and flash memory – A20-OLinuXino-MICRO-EMMC; 4. industrial temperature range and flash memory – A20-OLinuXino-MICRO-EMMC-IND) - each of these boards would have own hardware revision from now on.

8.3 Useful web links and purchase codes

The web pages you can visit for more information about your device are:
https://www.olimex.com/Products/OLinuXino/A20/A20-OLinuXino-MICRO-4GB/;
https://www.olimex.com/Products/OLinuXino/A20/A20-OLinuXino-MICRO/.

The wiki article for the board: https://www.olimex.com/wiki/A20-OLinuXino-MICRO

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).

ORDER CODES:

A20-OLinuXino-MICRO – the base version of A20-OLinuXino-MICRO, suitable for commercial temperature range and no external NAND

A20-OLinuXino-MICRO-IND – industrial version (with extended operating temperature range) – has industrial versions of RAM memory and Ethernet controller

A20-OLinuXino-MICRO-e4GB – A20-OLinuXino-MICRO with additional 4GB eMMC memory

A20-OLinuXino-MICRO-e4GB-IND – an industrial version of A20-OLinuXino-MICRO with additional 4GB eMMC memory

A20-OLinuXino-MICRO-n4GB – a version of A20-OLinuXino-MICRO with additional 4GB NAND memory

A20-OLinuXino-MICRO-n8GB – a version of A20-OLinuXino-MICRO with additional 4GB NAND memory

USB-SERIAL-CABLE-F – USB serial console cable female
USB-MINI-CABLE – standard USB type A to USB type mini cable
A20-VGA-CABLE – adapter from 6-pin connector to standard 15-pin VGA one
SY0612E – reliable power supply adapter 50Hz (for EU) 12V/0.5A for A20-OLinuXino-MICRO
SY0612E-CHINA – cheaper power supply adapter 50Hz (for EU) 12V/0.5A for A20-OLinuXino-MICRO
A20-Android-SD – a tested class 10 micro SD card with the latest (by the time of leaving the Olimex facilities) official Android release
A20-Debian-SD – a tested class 10 micro SD card with the latest (by the time of leaving Olimex facilities) official Debian Linux release
LCD-OLinuXino-4.3TS – low-cost 4.3" LCD display with touchscreen component – 480×272
LCD-OLinuXino-7TS – low-cost 7" LCD display with touchscreen component – 800×480
LCD-OLinuXino-10TS – low-cost 10.1" LCD display with touchscreen component – 1024×600
LCD-OLinuXino-15.6 – low-cost 15.6" LCD display with touchscreen component – 1024×600
LCD-OLinuXino-15.6FHD – low-cost 15.6" LCD display with touchscreen component – 1920×1080

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; either for SD card or flash memory), and upload it to the flash or 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_BUT for a couple of seconds to wake it up.

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 example, an image with file name "debian_2g.img" would be downloaded to the SD 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 uSD 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 uncommenting #auto eth0 in /etc/network/interfaces in the Linux image.

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

A: Build instructions and required files for the latest Debian images can be found in our GitHub repository.

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.

Q: Is it possible to boot Debian from the extra flash memory? Do you provide such image?

A: Yes, it is possible. Use the latest Debian Jessie image and type "nandinstall". If there are problems try to upload Android first to the NAND and then try "nandinstall" again. More information can be found in this wiki article: installing Debian Jessie to the 4GB NAND memory.

There are other people who are also successful in booting Debian from the NAND. 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 his instructions for NAND installation of Debian: link to his web-site.

Q: Can I use mainline kernel instead of Sunxi kernel?

A: You can. However, most kernels are headless – no audio or video output are supported. If you don't need audio or video – go for it – it had been tested and confirmed working (there are installation reports even) here: https://wiki.debian.org/InstallingDebianOn/Allwinner

Q: I have a 4GB version of the board. The board doesn't seem to have Android loaded.

A: We no longer load Android on the NAND memory of the boards. You can upload Android following the advice above in this document.

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 them 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 (6-16)V, 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 POWER 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.

8.5 Product support

For product support, hardware information and error reports mail to: support@olimex.com. All document or hardware feedback is welcome. Note that we are primarily a hardware company and our software support is limited. Please consider reading the paragraph below about the warranty of Olimex products.

All goods are checked before they are sent out. In the unlikely event that goods are faulty, they must be returned, to OLIMEX at the address listed on your order invoice.

OLIMEX will not accept goods that have clearly been used more than the amount needed to evaluate their functionality.

If the goods are found to be in working condition, and the lack of functionality is a result of lack of knowledge on the customers part, no refund will be made, but the goods will be returned to the user at their expense.

All returns must be authorized by an RMA Number. Email support@olimex.com for authorization number before shipping back any merchandise. Please include your name, phone number and order number in your email request.

Returns for any unaffected development board, programmer, tools, and cables permitted within 7 days from the date of receipt of merchandise. After such time, all sales are considered final.

Returns of incorrect ordered items are allowed subject to a 10% restocking fee. What is unaffected? If you hooked it to power, you affected it. To be clear, this includes items that have been soldered to, or have had their firmware changed. Because of the nature of the products we deal with (prototyping electronic tools) we cannot allow returns of items that have been programmed, powered up, or otherwise changed post shipment from our warehouse.

All returned merchandise must be in its original mint and clean condition. Returns on damaged, scratched, programmed, burnt, or otherwise 'played with' merchandise will not be accepted.

All returns must include all the factory accessories which come with the item. This includes any In-Circuit-Serial-Programming cables, anti-static packing, boxes, etc.

With your return, enclose your PO#. Also include a brief letter of explanation of why the merchandise is being returned and state your request for either a refund or an exchange. Include the authorization number on this letter, and on the outside of the shipping box.

Please note: It is your responsibility to ensure that returned goods reach us. Please use a reliable form of shipping. If we do not receive your package we will not be held liable.

Shipping and handling charges are not refundable. We are not responsible for any shipping charges of merchandise being returned to us or returning working items to you.

The full text might be found at https://www.olimex.com/wiki/GTC#Warranty for future reference.