

OLinuXino-MINI
Open-source single-board Linux computer
USER'S MANUAL

Revision F, January 2013
Designed by OLIMEX Ltd, 2012



All boards produced by Olimex LTD are ROHS compliant

DISCLAIMER

© 2012 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.

The Hardware project is released under the Creative Commons Attribution-Share Alike 3.0 United States License. You may reproduce it for both your own personal use, and for commercial use. You will have to provide a link to the original creator of the project <http://www.olimex.com> on any documentation or website. You may also modify the files, but you must then release them as well under the same terms. Credit can be attributed through a link to the creator website: <http://www.olimex.com>

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 OLINUXINO. THEY ARE CONSIDERED SUITABLE ONLY FOR OLINUXINO.

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 The OLinuXino family.....</u>	<u>6</u>
<u>1.2 Target market and purpose of the board.....</u>	<u>6</u>
<u>1.3 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 warning.....</u>	<u>8</u>
<u>2.3 Requirements.....</u>	<u>8</u>
<u>2.4 Powering the board.....</u>	<u>9</u>
<u>2.5 Prebuilt software.....</u>	<u>9</u>
<u>2.6 Using BitBurner.....</u>	<u>10</u>
<u>2.7 Building the Linux image.....</u>	<u>11</u>
<u>2.8 How to blink the LED.....</u>	<u>13</u>
<u>CHAPTER 3 OLINUXINO BOARD DESCRIPTION.....</u>	<u>15</u>
<u>3. Introduction to the chapter.....</u>	<u>15</u>
<u>3.1 Layout (top view).....</u>	<u>15</u>
<u>CHAPTER 4 THE iMX233 MICROCONTROLLER.....</u>	<u>16</u>
<u>4. Introduction to the chapter.....</u>	<u>16</u>
<u>4.1 The microcontroller.....</u>	<u>16</u>
<u>CHAPTER 5 CONTROL CIRCUITY AND HARDWARE MODULES.....</u>	<u>19</u>
<u>5. Introduction to the chapter.....</u>	<u>19</u>
<u>5.1 Reset.....</u>	<u>19</u>
<u>5.2 Clocks.....</u>	<u>19</u>
<u>5.3 Power supply circuit.....</u>	<u>20</u>
<u>CHAPTER 6 CONNECTORS AND PINOUT.....</u>	<u>22</u>
<u>6. Introduction to the chapter.....</u>	<u>22</u>
<u>6.1 Debugging interfaces.....</u>	<u>22</u>
<u>6.1.1 UART debug.....</u>	<u>23</u>
<u>6.1.2 SJTAG debug.....</u>	<u>23</u>
<u>6.1.3 Classic JTAG debug.....</u>	<u>24</u>
<u>6.2 SD/MMC slot.....</u>	<u>26</u>
<u>6.3 UEXT module.....</u>	<u>27</u>
<u>6.4 GPIO (General Purpose Input/Output) 40pin connector</u>	<u>28</u>

<u>6.5 USB HOSTs.....</u>	<u>31</u>
<u>6.6 PWR Jack.....</u>	<u>32</u>
<u>6.7 Headphones and line-in connector.....</u>	<u>33</u>
<u>6.8 Battery connector.....</u>	<u>34</u>
<u>6.9 Composite video connector.....</u>	<u>34</u>
<u>6.10 Boot mode positions.....</u>	<u>34</u>
<u>6.11 Jumper description.....</u>	<u>36</u>
<u>6.12.1 SCL SW/SCL HW and SDA SW/SDA HW.....</u>	<u>36</u>
<u>6.12.2 5V E.....</u>	<u>36</u>
<u>6.12.3 3.3VIO E and 3.3V E jumpers.....</u>	<u>36</u>
<u>6.12.4 Boot mode selecting jumpers.....</u>	<u>37</u>
<u>6.13 Additional hardware components.....</u>	<u>37</u>
<u>6.14 Accessories.....</u>	<u>37</u>
<u>6.14.1 USB-SERIAL-CABLE-F.....</u>	<u>37</u>
<u>CHAPTER 7 BLOCK DIAGRAM AND MEMORY.....</u>	<u>38</u>
<u>7. Introduction to the chapter.....</u>	<u>38</u>
<u>7.1 Memory addresses.....</u>	<u>38</u>
<u>7.2 Processor block diagram.....</u>	<u>39</u>
<u>7.3 Physical memory map.....</u>	<u>40</u>
<u>CHAPTER 8 SCHEMATICS.....</u>	<u>41</u>
<u>8. Introduction to the chapter.....</u>	<u>41</u>
<u>8.1 Eagle schematic.....</u>	<u>41</u>
<u>8.2 Physical dimensions.....</u>	<u>43</u>
<u>CHAPTER 9 REVISION HISTORY AND SUPPORT.....</u>	<u>44</u>
<u>9. Introduction to the chapter.....</u>	<u>44</u>
<u>9.1 Document revision.....</u>	<u>44</u>
<u>9.2 Board revision.....</u>	<u>45</u>
<u>9.3 Useful web links and purchase codes.....</u>	<u>47</u>
<u>9.3 Product support.....</u>	<u>48</u>

CHAPTER 1 OVERVIEW

1. Introduction to the chapter

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

The OLinuXino development board enables code development of applications running on the microcontroller i.MX233, manufactured by FreeScale Semiconductor.

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

1.1 Features

- iMX233 ARM926J processor at 454Mhz
- 64 MB RAM
- SD-card connector for booting the Linux image
- TV PAL/NTSC video output
- 3 USB High Speed Hosts
- Stereo Audio Input
- Stereo Headphones Audio Output
- Two Buttons
- UEXT connector for connection of different peripheral modules
- 40 pin GPIO for connection to external hardware
- Board is in the proper shape and dimensions to fit inside Pactec JM42 plastic box http://www.pactecenclosures.com/pdfs/drw_JM-42.pdf
- Power supply input 6-16VDC
- optional WIFI RTL8188CU module
- PCB dimensions: 3.70" x 2.15" (94.0mm x 54.6mm)
- Nominal dimensions: 3.70" x 2.65" (94.0mm x 67.3mm)

1.2 The OLinuXino family

Table of comparison			
	OLinuXino-MICRO	OLinuXino-MINI	OLinuXino-MAXI
Processor	iMX233 @ 454Mhz	iMX233 @ 454Mhz	iMX233 @ 454Mhz
Ram [MB]	64	64	64
# USB hosts	1	3	2
100/150 Mbit Ethernet*	No/WIFI option**	No/WIFI option***	Yes/WIFI option**
GPIO connector	60pins	40pins	40pins
# Buttons	3	2	2
Reset button	Yes	Yes	Yes
DC power supply	5V	5V-16V	5V-16V
Dimensions	3.40'' x 1.70''	3.70'' x 2.65''	3.70'' x 2.65''
Breadboarding	Yes	No	No
Audio IN connector	No	Yes	Yes
Audio OUT connector	No	Yes	Yes
UEXT connector	No	Yes	Yes

* 100Mbit Ethernet for the wired network of OLinuXino-MAXI. 150Mbit for the WIFI following 811.02n standard.

** All three boards have the option to work with MOD-WIFI_RTL8188, which is USB WIFI modem with RTL8188CU chip and can be purchased separately. MOD-WIFI_RTL8188 can be connected to any of the OLinuXino boards via the USB.

*** OLinuXino-MINI has additional option of having RTL8188CU hardware mounted! If you wish RTL8188CU embedded in the device you should purchase OLinuXino-MINI-WIFI. Choosing the embedded WIFI option will leave your USB-HOSTs available for use.

1.2 Target market and purpose of the board

The boards from the OLinuXino family are ready to use, easy to setup and are suitable for embedded programming enthusiasts, Linux hobbyists, gadget fans 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 mobility of the board and the low price.

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.

1.3 Organization

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

- Chapter 1 is an overview of the board usage and features
- Chapter 2 provides a guide for quickly setting up the board
- Chapter 3 contains the general board diagram and layout
- Chapter 4 describes the component that is the heart of the board: the iMX233 microcontroller
- Chapter 5 is an explanation of the control circuitry associated with the microcontroller to reset. Also shows the clocks on the board
- Chapter 6 covers the connector pinout, peripherals and jumper description
- Chapter 7 shows the memory map
- Chapter 8 provides the schematics
- Chapter 9 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 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.

2.3 Requirements

In order to set up the OLinuXino optimally, the following items are required:

- 6V to 16V source of power with 1A maximum amperage.
- SJTAG interface programmer
- USB keyboard
- Monitor with composite interface or Personal Computer + **USB-SERIAL-CABLE-F**
- SD card with Linux image

Note that the board arrives without SD card or Linux image. You can purchase a card with Linux separately. It is recommended that the user has basic Linux experience.

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

iMX233-OLinuXino-SD - SD card with the Linux image

USB-SERIAL-CABLE-F - USB serial console cable female (check “6.1.1 UART Debug” for info how to connect it to the board)

SY0612E - power supply adapter 12V/0.5A for iMX233-OLinuXino-MINI

2.4 Powering the board

The board is powered either via the PWR jack or via a battery. It should be supplied from a 6V to 16V source with maximum current of 1A from the power jack.

All measures below are taken at 10V.

If measuring the current consumption it should be around 0.06A before initializing all the peripherals. The consumption raises to 0.12A without WIFI and USB hosts initialized and Linux running. The consumption goes up 0.15A with the Linux running when WIFI and both USB hosts initialized.

If you have a standard USB flash drive attached to a USB host, Linux and WIFI running the typical consumption is around 0.20A.

When powered by the typical 3.7V battery the WIFI and USB-hosts will be disabled if you use external 3.3 DC-DC (not mounted by default).

For the European customers we sell a power supply adapter 12V/0.5A - **SY0612E**.

2.5 Prebuilt software

Note that the boards arrive without Linux or SD card. The Linux image can be purchased separately on a SD card or you can built and write it yourself.

When we program the boards we change the default position of the following HW_OCOTP_ROM0 fuses of the processor:

SD_MBR_BOOT(3) - Blown

SD_POWER_GATE_GPIO(21:20) – 10-PWM3

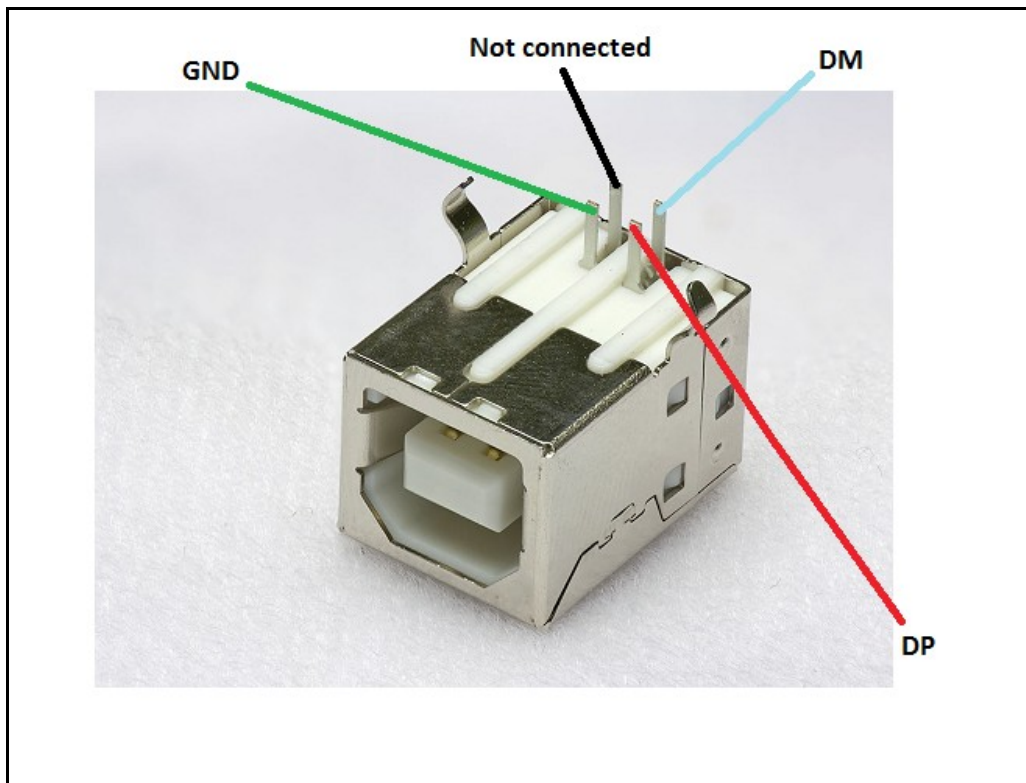
For burning the fuse position we use the BitBurner software. This operation is discussed in details before. Proceed with great caution when burning fuses since it is irreversible operation.

2.6 Using BitBurner

IMPORTANT! MODIFYING THE FUSES IS IRREVERSIBLE PROCESS! BURNING THE WRONG FUSES MIGHT DAMAGE OLINUXINO IRREVERSIBLY! BURNING WRONG FUSES MIGHT CAUSE BOOT PROBLEMS!

BURN FUSES AT OWN RISK!

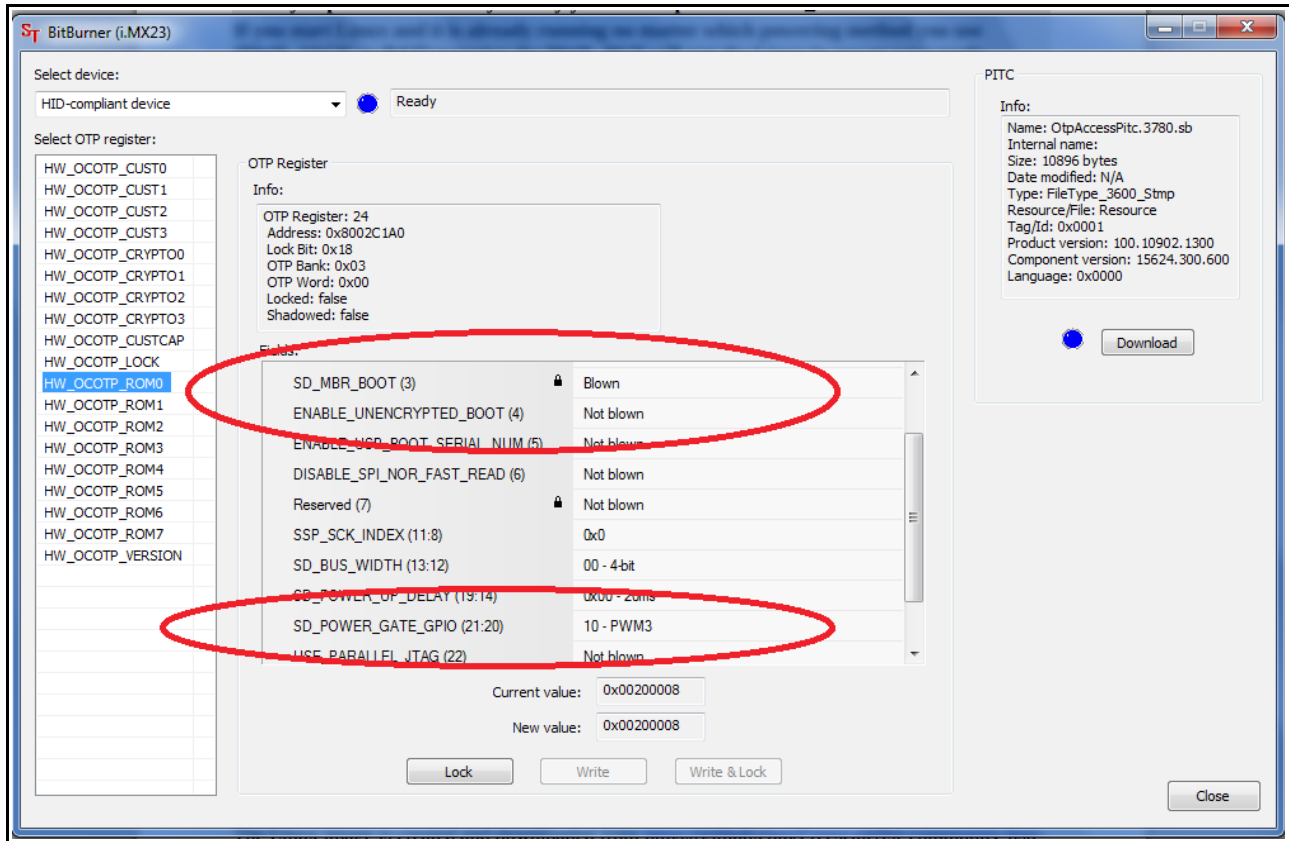
The bit burning is done via the USB of the computer connected to the OLINUXINO board and the BitBurner software. To be able to burn the fuses you will need to make a custom cable that connects a USB with the 3 pin holes found at the bottom of the board named “GND”, “DP”, “DM” (check the picture below for a better view how the three wires must be connected).



After soldering the three wires you can place a marker on each of them or use colored wires to be able to distinguish them. You can also use some gel to keep them tight on the USB connector. On the opposite side of the cables you might place 50mil (1.27mm) male connector following the order of the signals. Please also restrain from using wires longer than 20 cm since that might make the connection unreliable.

Download BitBurner from <https://www.olimex.com/dev/OLINUXINO/iMX233-OLINUXINO/BitBurner.v1.0.4.6.zip>. Extract it and start the .exe. If you connect everything you should see and choose HID-compliant device from the “Select device” drop-down menu. If you

check the fuses we have already burned the application should show an image as the one on the next page.



2.7 Building the Linux image

Note that building the Linux image from scratch is a time-consuming task. Even with powerful machine and fast internet connection it might take few hours compiling. Some Linux distributions might lack the tools required to compile/build/execute scripts/download from repository – how to get those is not discussed below.

The Linux image is created and downloaded from <https://github.com/Freescale/fsl-community-bsp-platform>. For the test here we used Debian 6.0 with GNOME visual libraries. The steps we did:

1) From the terminal created folder “bin” in home folder:

```
user@dist$: mkdir bin
user@dist$: cd bin
```

Add bin directory to PATH in order to do the next steps easier. Else navigate to the right folders.

2) Installed `repo` utility needed for the bitbake file fetching from the repository:

```
user@dist$: curl https://dl-ssl.google.com/dl/googlesource/git-repo/repo > ~/bin/repo
user@dist$: chmod a+x ~/bin/repo
```

3) Created directory for the project and download the BSP source from the git repository:

```
user@dist$: mkdir fsl-community-bsp
user@dist$: cd fsl-community-bsp

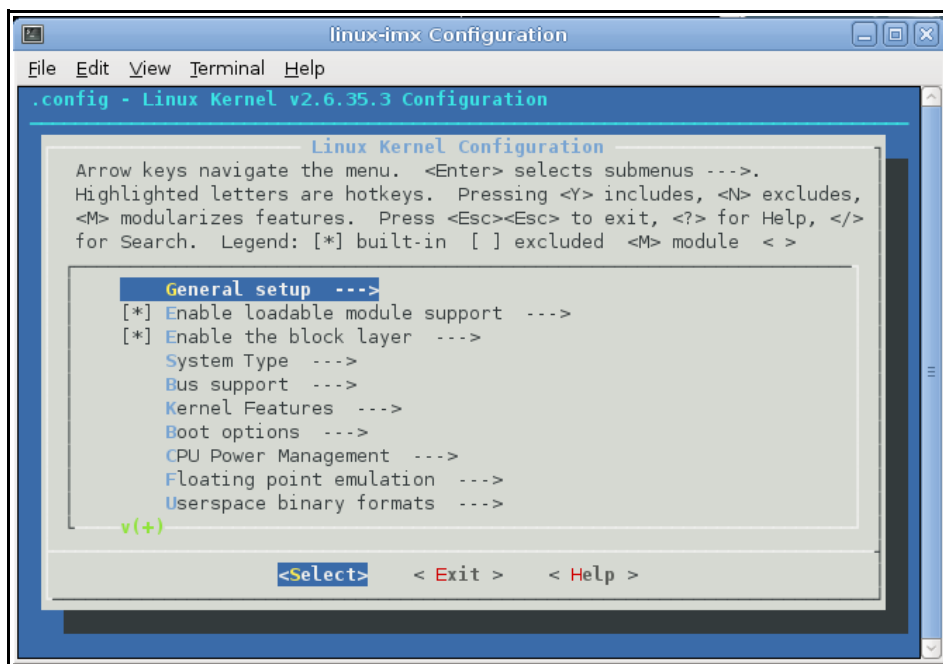
~/fsl-community-bsp$: repo init -u https://github.com/Freescale/fsl-community-bsp-platform -b
denzil
~/fsl-community-bsp$: repo sync
```

4) You can change the settings for the build if you want at fsl-community-bsp/build/conf/local.conf. I changed the “machine” name to “imx233-olinuxino-maxi”.

For Linux kernel configurations and settings you can do (of course you can use also the default settings):

```
~/fsl-community-bsp$: ./setup-environment build
~/fsl-community-bsp/build$: bitbake linux-imx -c menuconfig
```

Check the image below:



5) Now to start building the image:

```
~fsl-community-bsp$:. ./setup-environment build
~fsl-community-bsp/build $: bitbake core-image-minimal
```

Note: on different Linux distributions you might have different tools installed and you will probably need to install dependencies needed for the compile/build scripts. Here are some (but not all) of the mandatory ones: G++; diffstat; texi2html; chrpath; gawk; texinfo; some git client.

To ensure you have the latest version supported with all the updates visit <https://github.com/OLIMEX/OLINUXINO> and <https://github.com/Freescale/fsl-community-bsp-platform>.

2.8 How to blink the LED

In this chapter you will find a way to achieve the most basic task in electronics – the “Hello World” of electronics - blinking the LED.

First we set the pin responsible for the LED as an output and we can set its value manually to high or low position – make it blink manually. The LED mounted on the board uses GPIO65. You can use external diode instead of the one mounted - you have to look at the table “The Linux implementation of pins” in the hardware section to get the correct linux name for the GPIO pin.

```
echo out > /sys/class/gpio/gpio65/direction
echo 1 > /sys/class/gpio/gpio65/value
```

If you want to set the blink off you should change the value on the second line to:

```
echo 0 > /sys/class/gpio/gpio65/value
```

To show the info for all GPIOs:

```
ls /sys/class/gpio
```

To make it turn on – turn off automatically (e.g. blink) we use the text redactor VI to write the Linux script:

```
echo out > /sys/class/gpio/gpio65/direction
while true
do
echo 1 > /sys/class/gpio/gpio65/value
```

```
sleep 1
echo 0 > /sys/class/gpio/gpio65/value
sleep 1
done
```

We save it as as “gpio” and we make it executable with

```
chmod +x gpio
```

then we execut the script with:

```
./gpio
```

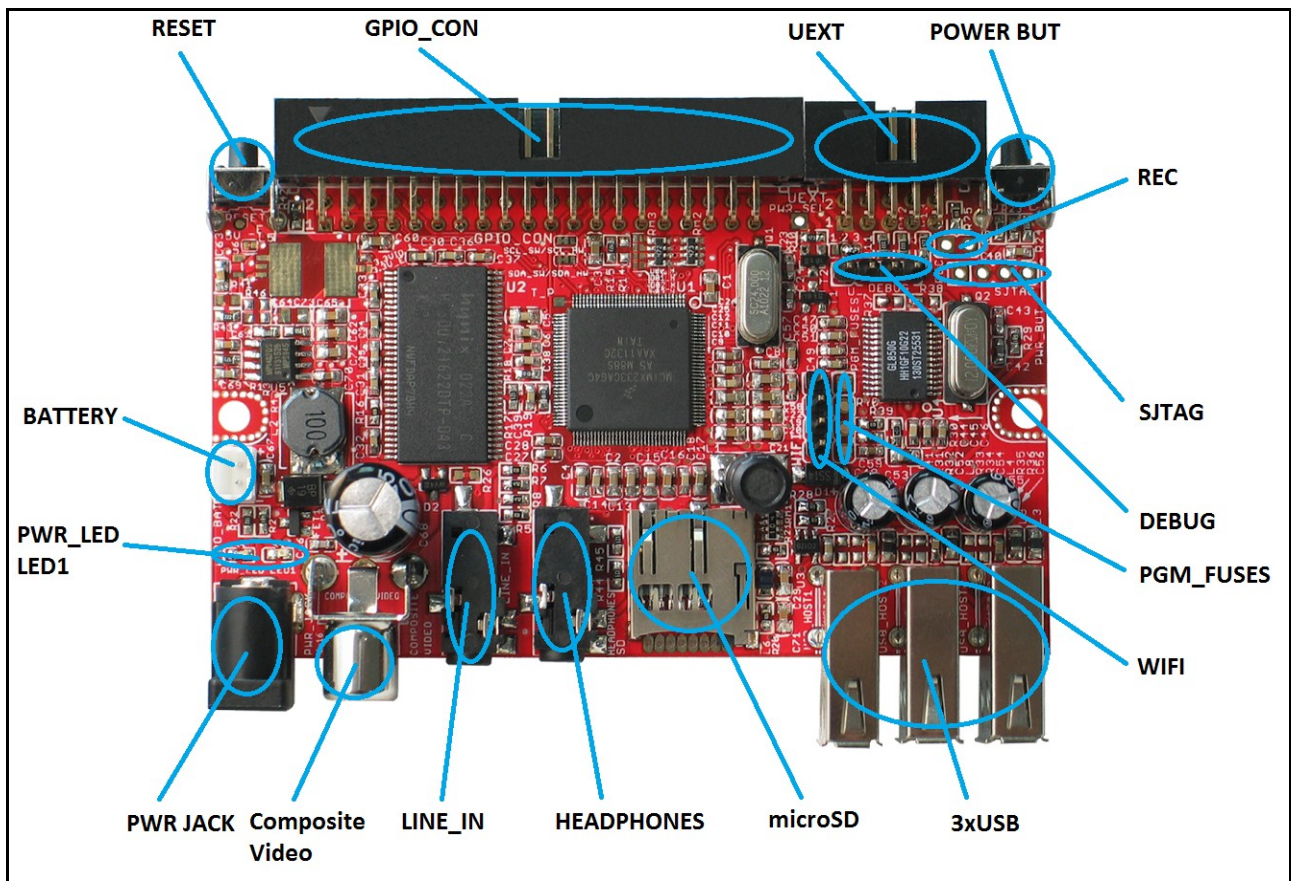
The LED should start blinking with 0.5Hz.

CHAPTER 3 OLINUXINO 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 differ from the names used to describe them. For the actual names check the OLinuXino board itself.

3.1 Layout (top view)



CHAPTER 4 THE iMX233 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 microcontroller

- ARM926 CPU Running at 454 MHz
- Integrated ARM926EJ-S CP
- 16-Kbyte data cache and 16-Kbyte instruction cache
 - One-wire JTAG interface
 - Resistor-less boot mode selection using integrated OTP values
- 32Kbytes of Integrated Low-Power On-Chip RAM
- 64 Kbytes of Integrated Mask-Programmable On-Chip ROM
- 1 Kbit of On-Chip One-Time-Programmable (OCOTP) ROM
- Universal Serial Bus (USB) High-Speed (Up to 480 Mb/s), Full-Speed (Up to 12 Mb/s)
 - Full-speed/high-speed USB device and host functions
 - Fully integrated full-speed/high-speed Physical Layer Protocol (PHY)
 - Mass storage host-capable (uncertified by USB-IF)
- Power Management Unit
 - Single inductor DC-DC switched converter with multi-channel output supporting Li-Ion batteries.
 - Features multi-channel outputs for VDDIO (3.3 V), VDDD (1.2 V), VDDA (1.8 V), VDDM (2.5V) and regulated 4.2V source.
 - Direct power from 5-V source (USB, wall power, or other source), with programmable current limits for load and battery charge circuits.
 - Silicon speed and temperature sensors enable adaptive power management over temperature and silicon process.
- Audio Codec
 - Stereo headphone DAC with 99 dB SNR
 - Stereo ADC with 85 dB SNR
 - Stereo headphone amplifier with short-circuit protection and direct drive to eliminate bulky capacitors
 - Amplifiers are designed for click/pop free operation.

- Two stereo line inputs
- Microphone input
- SPDIF digital out
- 16-Channel Low-Resolution ADC
 - 6 independent channels and 10 dedicated channels
 - Resistive touchscreen controller
 - Temperature sensor controller
 - Absolute accuracy of 1.3%
- Security Features
 - Read-only unique ID for digital rights management algorithms
 - Secure boot using 128-bit AES hardware decryption
 - SHA-1 hashing hardware
 - Customer-programmed (OTP) 128 bit AES key is never visible to software.
- External Memory Interface (EMI)
 - Provides memory-mapped (load/store) access to external memories
 - Supports the following types DRAM:
 - 1.8V Mobile DDR
 - Standard 2.5V DDR1
- Wide Assortment of External Media Interfaces
 - High-speed MMC, secure digital (SD)
 - Hardware Reed-Solomon Error Correction Code (ECC) engine offers industry-leading protection and performance for NANDs.
 - Hardware BCH ECC engine allowing for up to 20-bit correction and programmable redundant area.
- Dual Peripheral Bus Bridges with 18 DMA Channels
 - Multiple peripheral clock domains save power while optimizing performance.
 - Direct Memory Access (DMA) with sophisticated linked DMA command architecture saves power and off-loads the CPU.
- Highly Flexible Display Controller
 - 8-bit data ITU-R BT.656 D1 digital video stream output mode (PAL/NTSC), with on-the-fly RGB to YCbCr color-space-conversion.
 - Flexible input formats
- Pixel Processing Pipeline (PXP)
 - Provides full path from color-space conversion, scaling, alpha-blending to rotation without intermediate memory access
 - Bi-linear scaling algorithm with cropping and letterboxing
 - Alpha-blend, BITBLT, color-keying
 - Memory efficient block-based rotation engine

- **Integrated TV-Out Support**
 - Integrated PAL/NTSC TV-encoder fully pipelined to display controller's D1 resolution output stream
 - Integrated low-power 10-bit Video DAC (VDAC) for composite analog video output.
- **Data Co-Processor (DCP)**
 - AES 128-bit encryption/decryption
 - SHA-1 hashing
 - High-speed memory copy
- **Three Universal Asynchronous Receiver-Transmitters (UARTs)**
 - Two high-speed application UARTs operating up to 3.25 Mb/s with hardware flow control and dual DMA.
 - Debug UART operates at up to 115Kb/s using programmed I/O.
- **I2C Master/Slave**
 - DMA control of an entire EEPROM or other device read/write transaction without CPU intervention
- **Dual Synchronous Serial Ports (for SPI, MMC, SDIO, Triflash)**
 - 1-bit, 4-bit and 8-bit MMC/SD/SDIO modes
 - Compliant with SDIO Rev. 2.0
 - SPI with single, dual and quad modes.
- **Four-Channel 16-Bit Timer with Rotary Decoder**
- **Five-Channel Pulse Width Modulator (PWM)**
- **Real-Time Clock**
 - Alarm clock can turn the system on.
 - Uses the existing 24-MHz XTAL for low cost or optional low power crystal (32.768 kHz or 32.0 kHz), customer-selectable via OTP.
- **Customer-Programmable One-Time-Programmable (OTP) ROM via Integrated eFuse Block**
 - Resistor-less boot mode selection
 - 128-bit boot mode crypto key
 - Boot mode specification of NAND characteristics for device that the customer is soldering to the board. This means no more costly delays waiting for new device support in the boot ROM.
 - Fully software-programmable and accessible
- **Flexible I/O Pins**
 - All digital pins have drive-strength controls
 - Most non-EMI digital pins have general-purpose input/output (GPIO) mode.

For comprehensive information on the microcontroller visit the Freescale's web page for a datasheet.

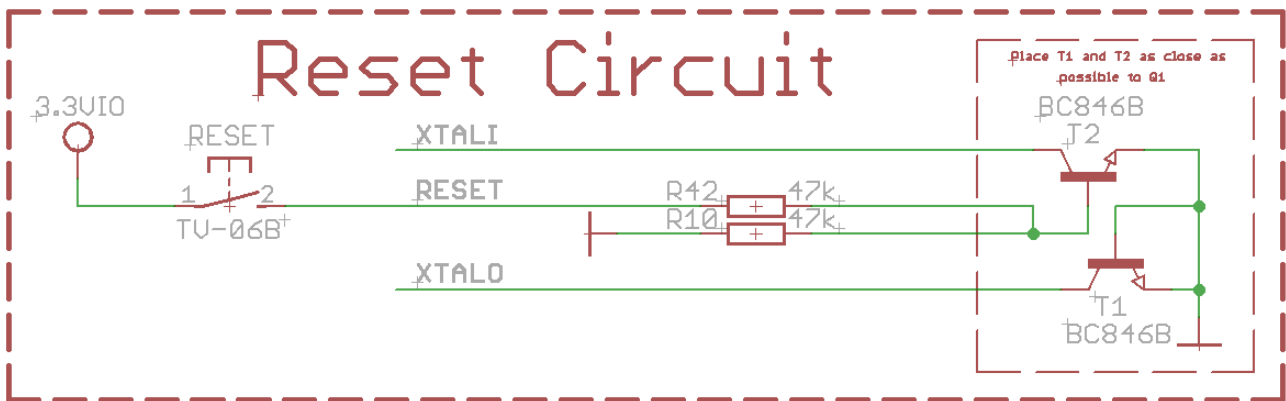
CHAPTER 5 CONTROL CIRCUITRY AND HARDWARE MODULES

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

OLinuXino's reset circuit includes R42 (47KΩ), R10 (47 Ω), T1, T2, Q1 and a RESET button. The RESET is specific for the fact that it is accomplished when the quartz is disconnected using 3.3V and the transistors T1 and T2.



5.2 Clocks

24 MHz quartz crystal Q1 is connected to pins 121 and 122 of the iMX233 processor.

12 MHz quartz crystal Q2 is found at pins 6 and 7 of the LAN and WIFI controller – GL850G.

5.3 Power supply circuit

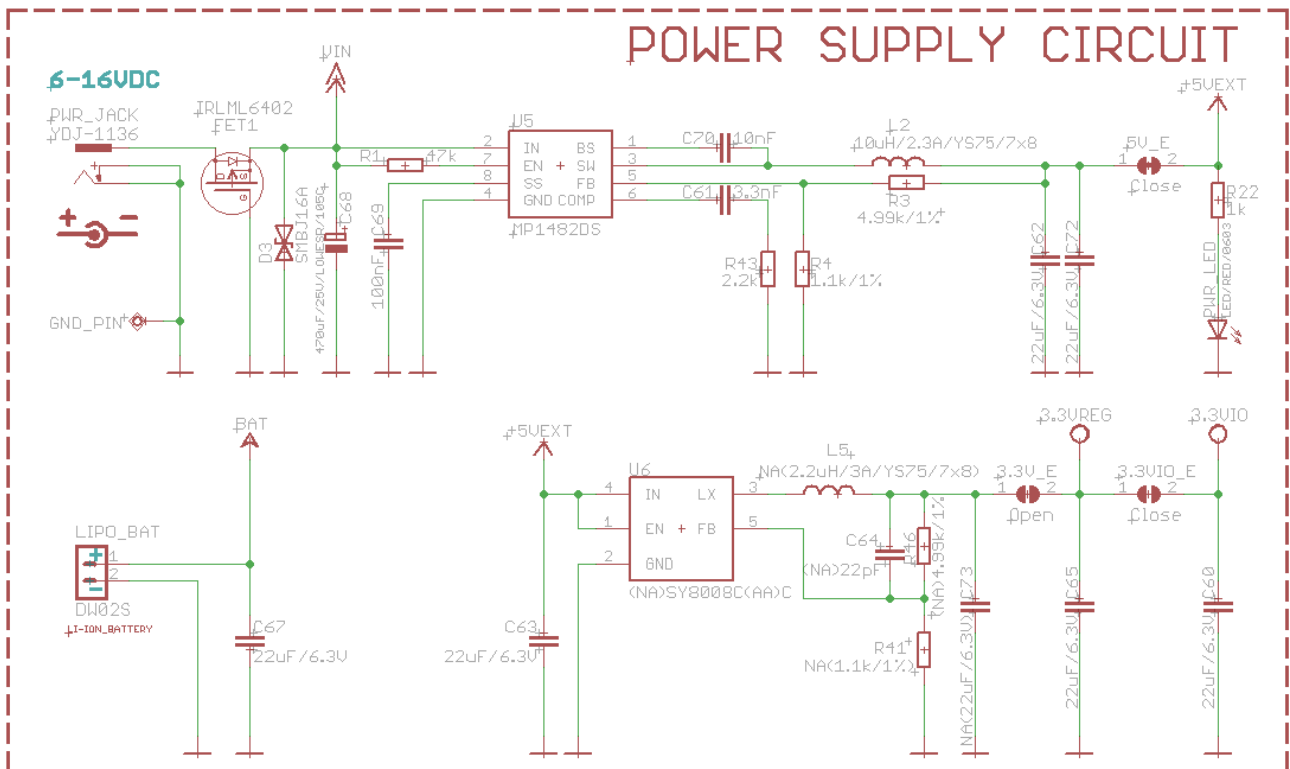
The power supply circuit of OLinuXino-MINI allows flexible input supply from 6V to 16V direct current. This means a wide range of power supplies, adapters, converters are applicable. The maximum amperage the board can draw is 1A.

The board with the prebuilt software and SD card equipped consumes between 0.12A and 0.16A while booting the Linux image. When no other peripherals are connected the board (except the SD card) the typical consumption is 0.12A. With one flash memory stick the consumption raises to 0.16A.

When powered by the typical 3.7V battery the USB-hosts are disabled. The consumption when Linux is running is around 0.75A.

The board can be powered either from the power jack or from the LIPO_BAT connector. Note that since standard batteries can't provide the voltage needed for the board it is advisable to use the power from the power jack. When you use the LIPO_BAT connector and a battery the chip handling the Ethernet and the USB hosts is disabled. Also the power button has alternative function when the board is powered by a battery – check 6.12.

If you have successfully powered the board the red PWR LED will turn on. Note that it is possible to have the PWR LED on even if there isn't enough power for proper operation of the board and all the peripherals currently connected.



Note that the whole 3.3V part is not mounted (elements marked as NA) by default because we use the internal 3.3V DC-DC convertor in the iMX233. However, the internal DC-DC in the processor is not very reliable when a lot of current is required. It is better to mount the elements of the 3.3V part circuit if you wish to use 3.3V at heavier loads.

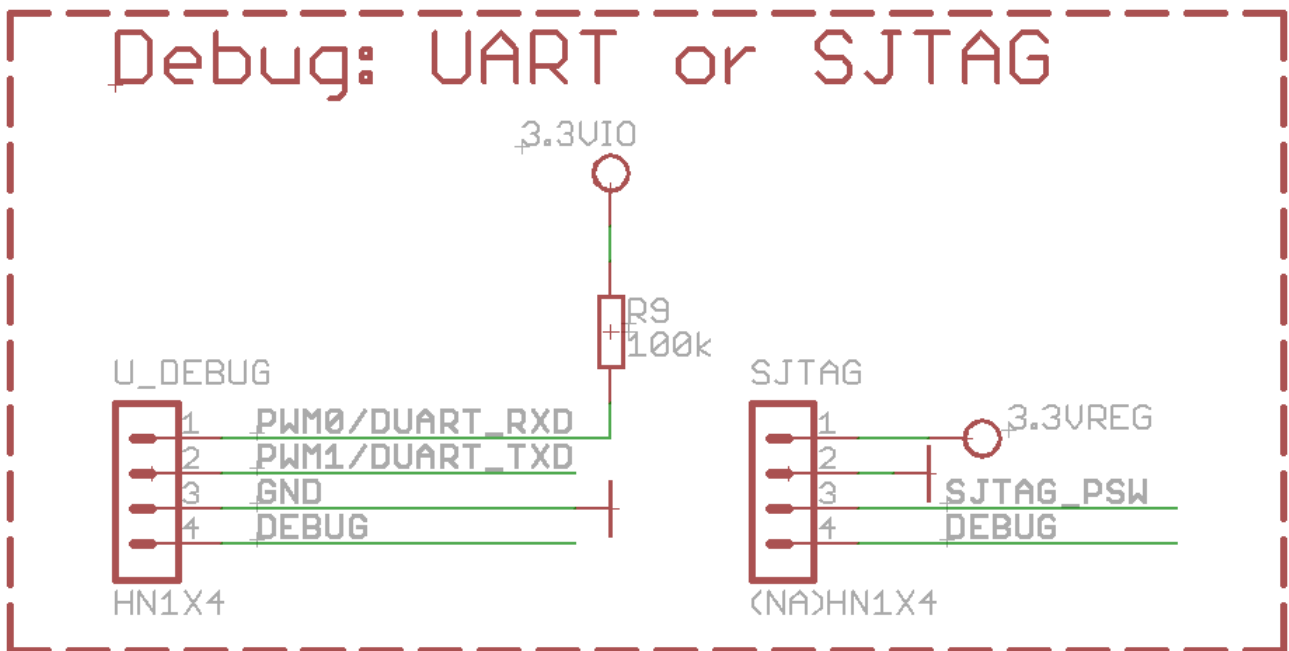
CHAPTER 6 CONNECTORS AND PINOUT

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 Debugging interfaces

If you don't have a separate monitor or display around, and you don't wish to swap cables constantly with your personal computer using the debug interfaces provide better option for you OLinuXino experience.



There are three debugging options available on the OLinuXino. It is preferable to use one of the first two - UART or SJTAG but in case you want to use the classic 6-wire parallel JTAG there is an option explained below.

6.1.1 UART debug

The first one is a debug UART interface – U_DEBUG. You can use our USB-SERIAL-CABLE for debugging via the UART.

Note on how to use the U_DEBUG with USB-SERIAL-CABLE-F which has RED GREEN BLUE wires GND=BLUE, RX(INPUT)=GREEN, TX(OUTPUT)=RED. You have to connect to OLinuXino-MAXI U_DEBUG connector in this sequence:

BLUE wire to pin.3 (GND)
GREEN wire to pin.2 (TX)
RED wire to pin.1 (RX)

The baud rate for serial communication is 115 200.

U_DEBUG		
Pin #	Signal Name	Processor Pin #
1	PWM0/DUART_RXD	125
2	PWM1/DUART_TXD	126
3	GND	30, 35, 98, 105, 112, 118
4	DEBUG	84

You can also check the pin names at the bottom of the board under the U_DEBUG header.

6.1.2 SJTAG debug

The second interface is the Serial JTAG (SJTAG) one-wire interface. It works with various external JTAG debugger dongles through a Freescale-defined FPGA/CPLD. SJTAG supports the Green Hills Slingshot and ETM probe debugger dongles, as well as those made by ARM, Abatron, and Lauterbach.

Note that the SJTAG interface comes without plastic header mounted.

The HW_DIGCTL_CTRL_USE_SERIAL_JTAG bit in the digital control block selects whether the one-wire serial JTAG interface or the alternative six-wire parallel JTAG interface is used. There are

other options in the digital control block which might interest you and our advice is to check the iMX233's datasheet released by Freescale.

SJTAG		
Pin #	Signal Name	Processor Pin #
1	3.3VREG	-
2	GND	30, 35, 98, 105, 112, 118
3	SJTAG_PSW	119,12
4	DEBUG	84

The pin names are also written at the bottom of the board for your convenience.

6.1.3 Classic JTAG debug

The third option is to use the classic 6-pin parallel JTAG (not including GND and VCC). There are pads left for this option at the bottom of the board under the SD card holder. They are named individually and framed near a "JTAG" name.

The important thing is that these pins are multiplexed with the SD card signals. In order to use the 6-pin JTAG you will have to stop using the SD card. Note also that the classic JTAG interface comes without plastic header mounted so in order to use it you will need to use wires for connection.

Consider the the software settings required to switch from SJTAG to 6-pin JTAG and vice verse referring to the iMX233 manual. For instance:

The HW_DIGCTL_CTRL_USE_SERIAL_JTAG bit in the digital control block selects whether the serial JTAG interface or the alternative six-wire parallel JTAG interface is used.

HW_DIGCTL_CTRL_USE_SERIAL_JTAG - 0x0 - Selects whether the one-wire serial JTAG interface or the alternative six-wire parallel JTAG interface is used. 0 = Parallel six-wire JTAG is enabled and is mapped to a collection of module pins that must be enabled by programming their MUXSEL bits in the pin control block. 1 = Serial JTAG is enabled and uses the dedicated DEBUG pin. The ROM bootcode writes this field prior to enabling JTAG, selecting which type of JTAG pin signaling to use. OLD_JTAG = 0x0 Use six-wire parallel JTAG mode. SERIAL_JTAG = 0x1 Use one-wire serial JTAG mode.

It is good idea to check the datasheet of the iMX233 processor for all the options in the digital control block. The processor's datasheet should be always the first piece of paper you consult when dealing with electronics.

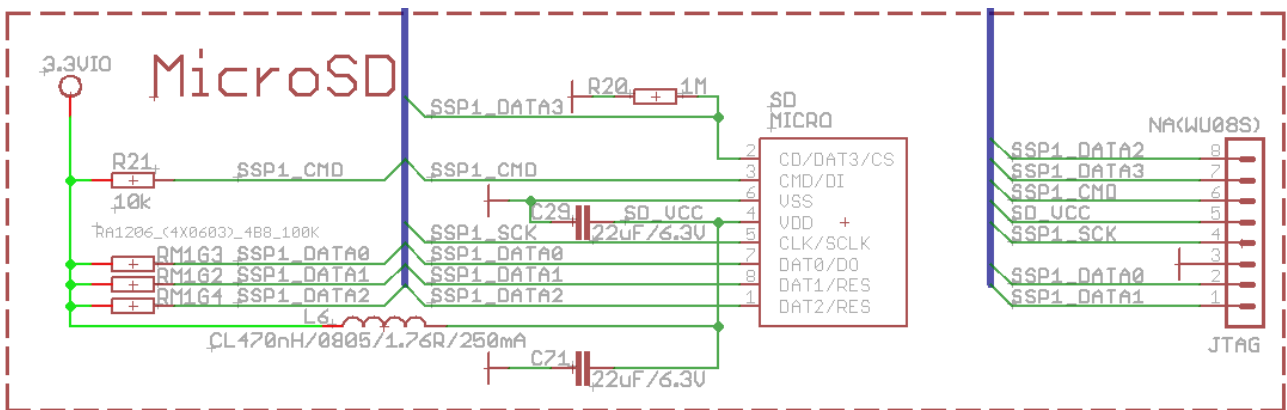
Classic JTAG (multiplexed with SD card)		
Pin #	Signal Name	Processor Pin #
1	SSP1_DATA1	85
2	SSP1_DATA0	84
3	GND	30, 35, 98, 105, 112, 118
4	SSP1_SCK	90
5	SD_VCC	-
6	SSP1_CMD	83
7	SSP1_DATA3	87
8	SSP1_DATA2	86

Notice that the pad numeration is written at the bottom of OLinuXino-MINI under the microSD card connector. Please check the manual part for microSD card for a schematic of the pins.

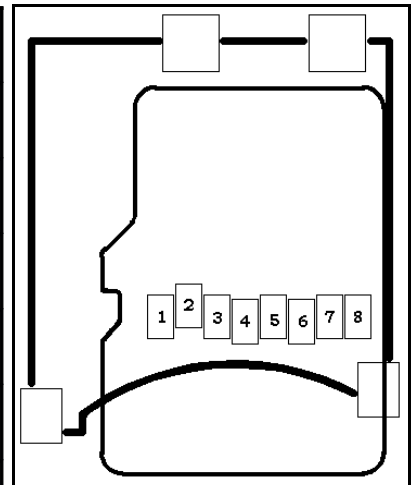
6.2 SD/MMC slot

The microSD card slot is a standard 8pin connector.

We have tested a number of microSD cards on the OLinuXino boards and all of them worked fine regardless manufacturer or capacity. However, keep in mind that some of the lower quality microSD cards might draw too much current from the slot which might cause power-state problems. If you suspect the microSD card is causing problems please try using another one of better quality for better results.



microSD card connector		
Pin #	Signal Name	Processor Pin #
1	SSP1_DATA2	86
2	SSP1_DATA3	87
3	SSP1_CMD	83
4	SD_VCC	-
5	SSP1_SCK	90
6	GND	30, 35, 98, 105, 112, 118
7	SSP1_DATA0	84
8	SSP1_DATA1	85



Notice that the pad numeration is written at the bottom of OLinuXino-MINI under the microSD card connector.

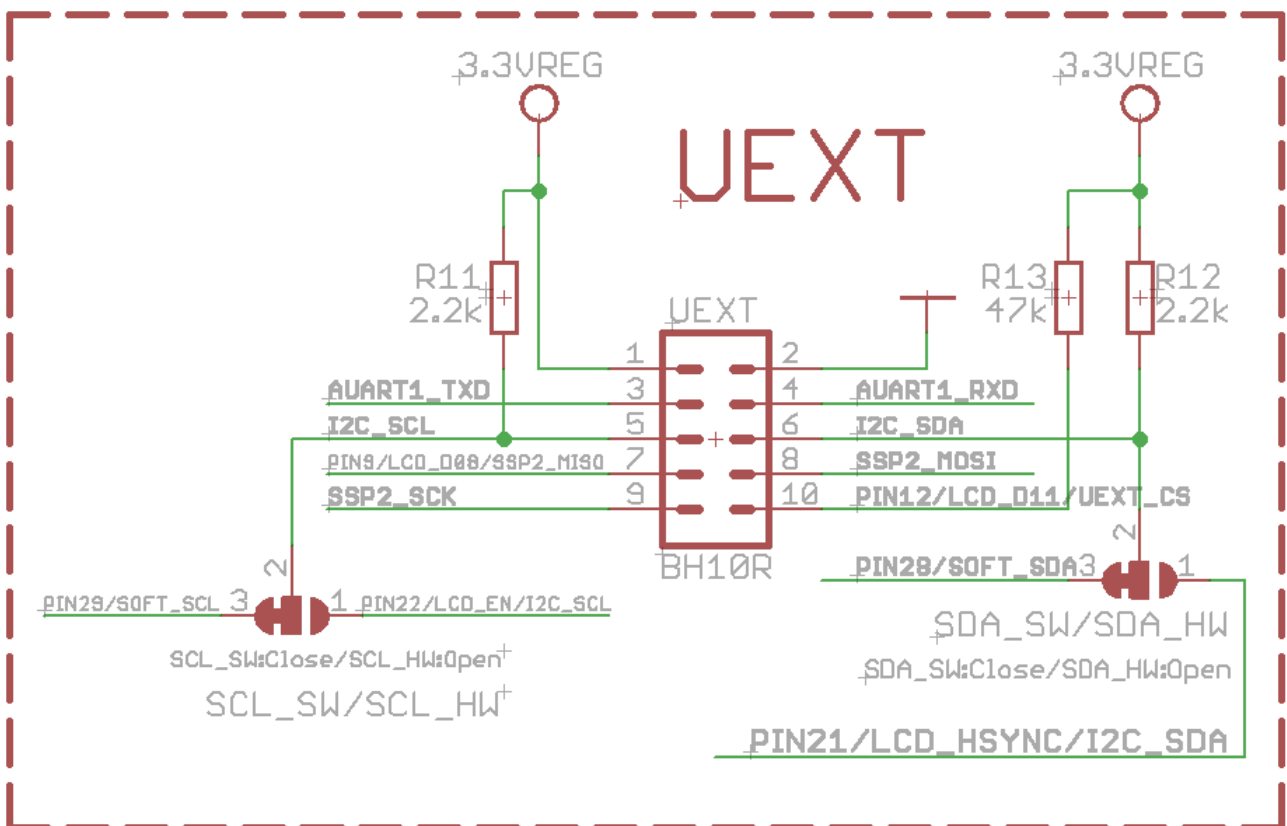
When removing the card, please make sure that you release it from the connector by pushing and NOT by pulling the card directly (this can damage both the connector and the microSD card).

6.3 UEXT module

OLinuXino board has UEXT connector and can interface Olimex's UEXT modules.

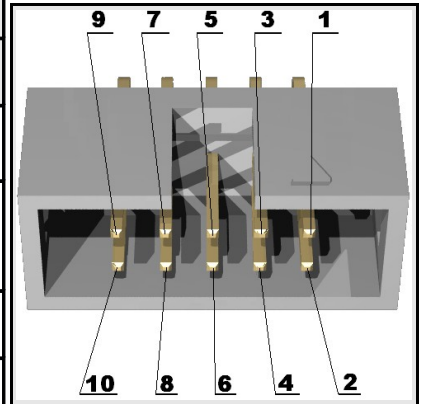
For more information on UEXT please visit:

<http://www.olimex.com/dev/OTHER/UEXT.pdf>



*Note the two jumpers PIN29/SOFT_SCL – PIN22/LCD_EN/I2C_SCL, and PIN28/SOFT_SDA3 – PIN21/LCD_HSYNC/I2C_SDA which by default are set to a software SPI. They can be used to set the whole UEXT to a hardware SPI.

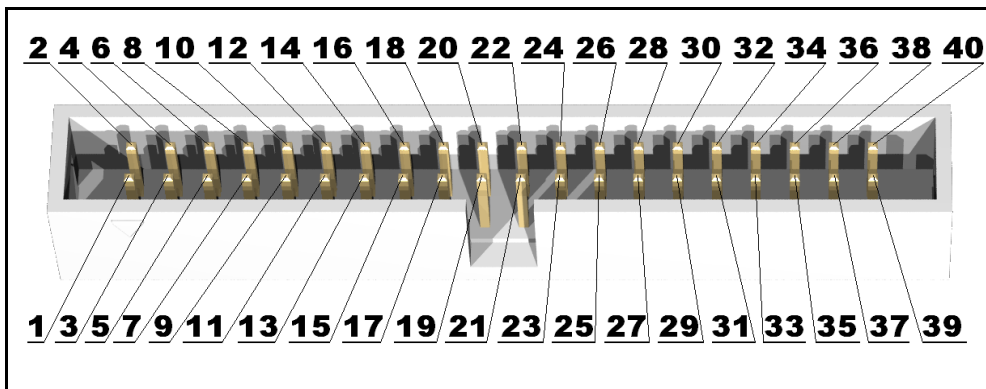
UEXT connector		
Pin #	Signal Name	Processor Pin #
1	+3.3VREG	-
2	GND	30, 35, 98, 105, 112, 118
3	AUART1_TXD	127
4	AUART1_RXD	128
5	I2C_SCL	34(default) OR 11*
6	I2C_SDA	31(default) OR 15*
7	PIN9/LCD_D08/SSP2_MISO	22
8	SSP2_MOSI	21
9	SSP2_SCK	33
10	PIN12/LCD_D11/UEXT_CS	25



The UEXT pinout is also printed at the bottom of the board under the connector.

6.4 GPIO (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 Pin#” column shows connector number and does NOT represent the naming on the bottom of the board.



GPIO connector hardware					
GPIO Pin#	Signal Name	Processor pin#	GPIO Pin#	Signal Name	Processor pin#
1	BAT	103	21	PIN8/LCD_D07	9
2	GND	30, 35, 98, 105, 112, 118	22	PIN27/PWM2	91
3	PIN17/LCD_D16	19	23	PIN7/LCD_D06	8
4	PIN18/LCD_D17/LAN_RES	20	24	PIN28/SOFT_SDA	31
5	PIN16/LCD_D15	28	25	PIN6/LCD_D05	7
6	PIN19/LCD_DOTCLK	17	26	PIN29/SOFT_SCL	34
7	PIN15/LCD_D14	29	27	PIN5/LCD_D04	6
8	PIN20/LCD_VSYNC	16	28	PIN30	81
9	PIN14/LCD_D13	26	29	PIN4/LCD_D03	5
10	PIN21/LCD_HSYNC/ I2C_SDA	15	30	PIN31	82
GPIO Pin#	Name	Processor pin#	GPIO Pin#	Name	Processor pin#
11	PIN13/LCD_D12	27	31	PIN3/LCD_D02	4
12	PIN22/LCD_EN/I2C_SCL	11	32	PIN32/LRADC1	107
13	PIN12/LCD_D11/UEXT_CS	25	33	PIN2/LCD_D01	3
14	PIN23/LCD_DISP	12	34	PIN33/LRADC0	108
15	PIN11/LCD_D10	24	35	PIN1/LCD_D00	2
16	PIN24/LCD_WR	13	36	PIN34/MIC	116
17	PIN10/LCD_D09	23	37	+5VUEXT	102
18	PIN25/LCD_RS	14	38	3.3VREG	-
19	PIN9/LCD_D08/ SSP2_MISO	22	39	VIN	-
20	PIN26/LCD_CS	10	40	GND	30, 35, 98, 105, 112, 118

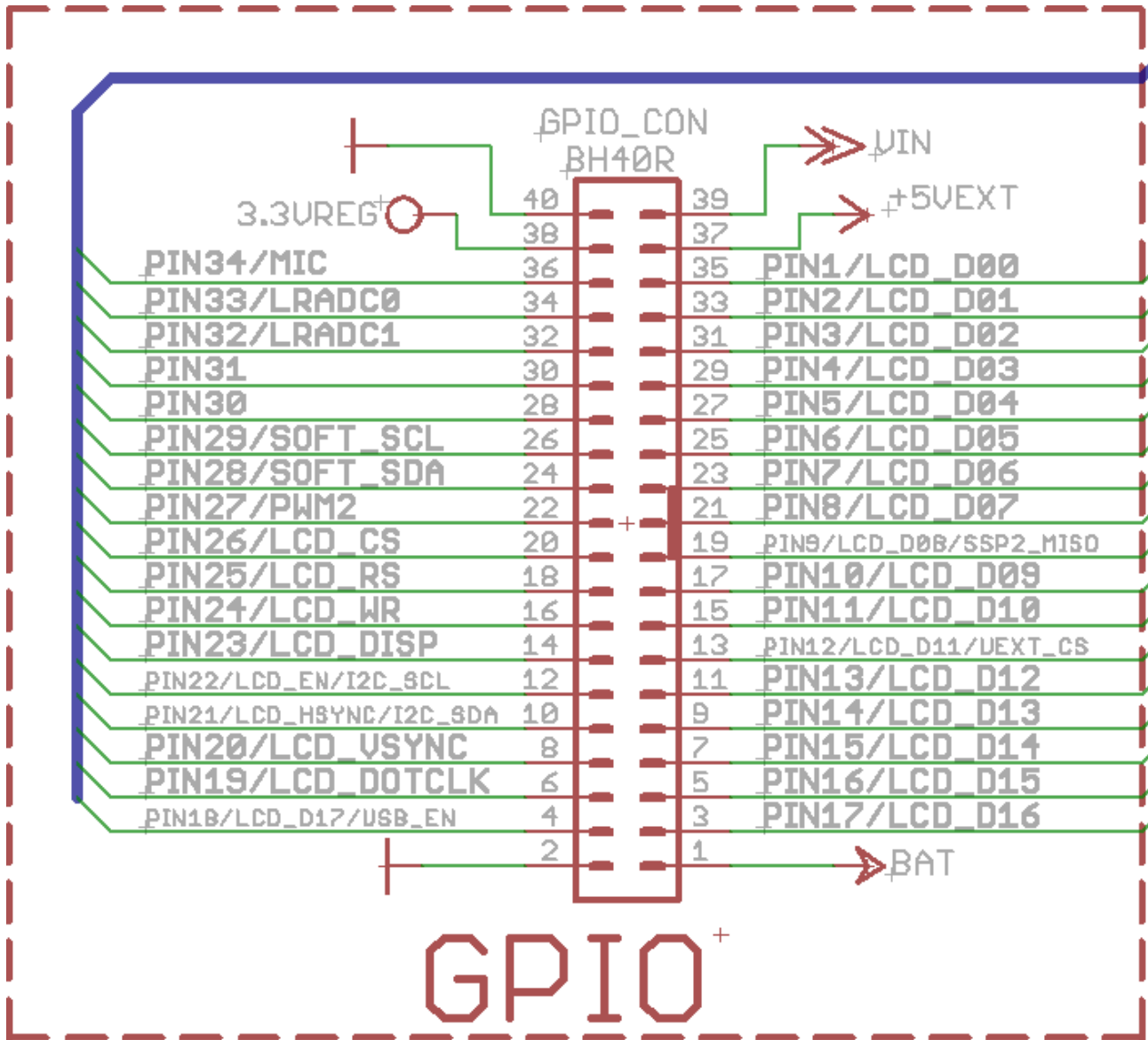
The hardware is associated differently in the Linux following the GPIO naming conventions suggested in the iMX233 datasheet. You can check the connection between Linux naming of the pin, Olimex naming of the pin and the consecutive connector pin number in the table below. The

ones filled with “Not implemented” doesn't have Linux support by the time of writing and will be updated overtime. “Linux GPIO” is the one you should use in Linux (the one in the datasheet); “OLinuXino name” is the pin as written on the bottom of the board. ”OLinuXino GPIO Connector #” is the consecutive number of pins with BAT being Pin#1 and GND#40.

Note that if “OLinuXino Name” starts with P and then is followed by a number X, the PX is the name written on the bottom of the board (under the connector) with white ink. If “OLinuXino Name” is other text it is a signal used for the hardware of the board but still can be controller by the Linux (it can't be found at the GPIO connector though).

The Linux implementation of pins					
Linux GPIO/iMX 233 GPIO	OLinuXino Name	OLinuXino GPIO Connector #	Linux GPIO/iMX 233 GPIO	OLinuXino Name	OLinuXino GPIO Connector #
0	PIN9	19	32 to 39	Not implemented	Not implemented
1	PIN10	17	51	PIN23	18
2	PIN11	15	52	PIN25	16
3	PIN12	13	53	PIN24	20
4	PIN13	11	55	PIN19	12
5	PIN14	9	56	PIN22	10
6	PIN15	7	64	JTAG_TD01 SSP1_CMD	Not implemented
7	PIN16	5	65	LED1	Not implemented
16	PIN17	3	91	PIN30	28
17	PIN18	4	92	PIN31	30
19	TEST_PAD	Not implemented			
20	UEXT_SPI2_MOSI	9			
23	PIN29	26			
24	UEXT_SPI2_SCK	Not implemented			
25	PIN28	24			
30	UEXT_TX1	Not implemented			
31	UEXT_RX1	Not implemented			

Below you can find the GPIO_CON as seen in the schematic:



When looking at the bottom of OLinuXino-MINI near the GPIO connector there is also an additional GND pad named GND_PIN which is a fast way to have access to a ground signal.

6.5 USB HOSTS

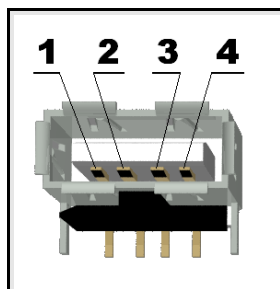
The USB hosts (and wireless connectivity option) are controlled by a single chip (GL850G).

The big advantage of having USB hosts available over USB devices is that you can as well use

them as masters.

The signals follow the familiar and standard USB host pattern:

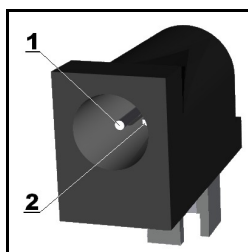
USB 2-level host	
PIN#	SIGNAL NAME
1	USB_PWR_A
2	USB_HOST_D-
3	USB_HOST_D+
4	GND



6.6 PWR Jack

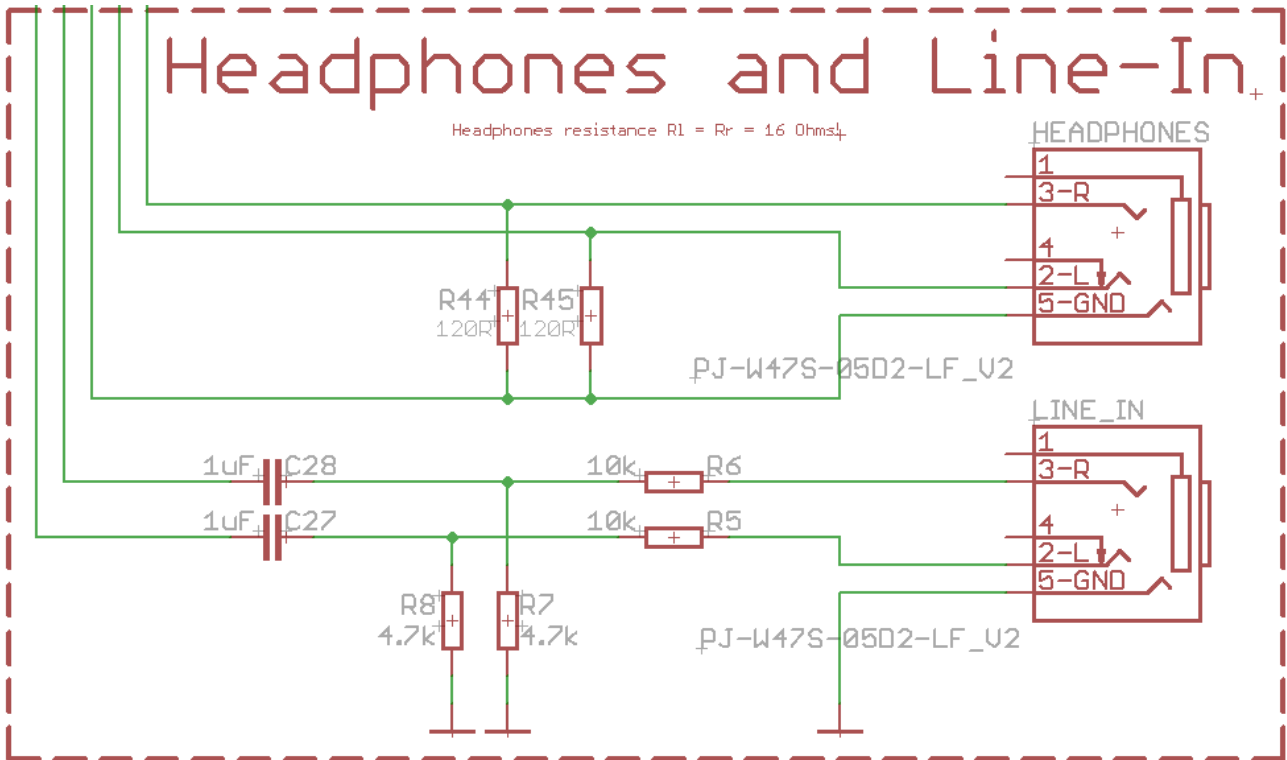
The power jack used is the typical 2.5mm one used by Olimex in most of our products. You should provide between 6 and 16 volts @ 1A to the board.

Pin #	Signal Name
1	Power Input
2	GND



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

6.7 Headphones and line-in connector



Standard audio jack and phone jack are mounted for the audio interfacing.

Headphones/Audio out connector		
Pin#	SIGNAL NAME	Processor Pin#
2	L channel	113
3	R channel	109
5	GND	GND pins

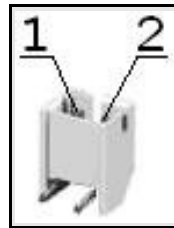
The headphones resistance is 16 Ohms!

Line in/Audio in connector		
Pin#	SIGNAL NAME	Processor Pin#
2	L channel	115
3	R channel	114
5	GND	GND pins

6.8 Battery connector

When using the battery connector keep in mind that it is an energy solution that wouldn't be able to power the board and all the peripherals. The voltage of a 3.7V LIPO battery would be enough to power the processor and the memory but probably won't be enough to power all the devices you mount on the USB hosts.

Pin #	Signal Name
1	VBAT
2	GND



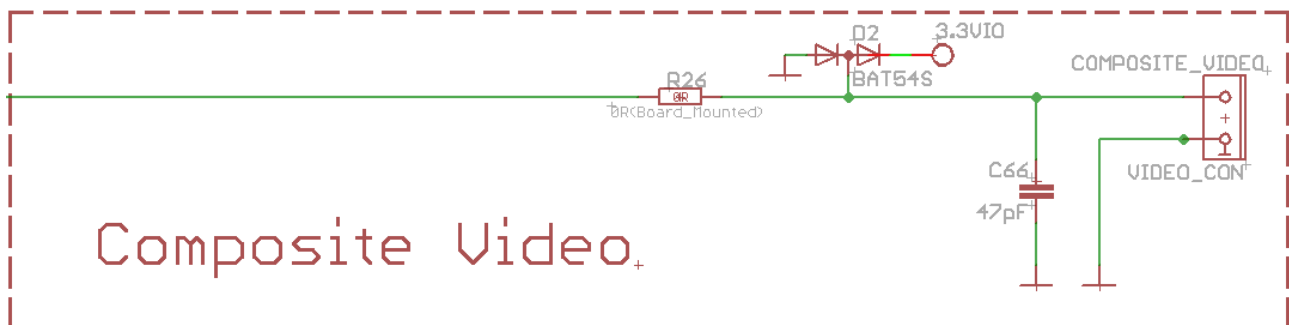
The pins are also written on the top of the board in the base of the connector.

6.9 Composite video connector

The composite video is the connector you should use if you wish OLinuXino-MINI video output on a monitor.

The whole signal is controlled by pin #104 from the i.MX233 processor.

The + signal is lead to the middle of the connector. The outside part is the GND.

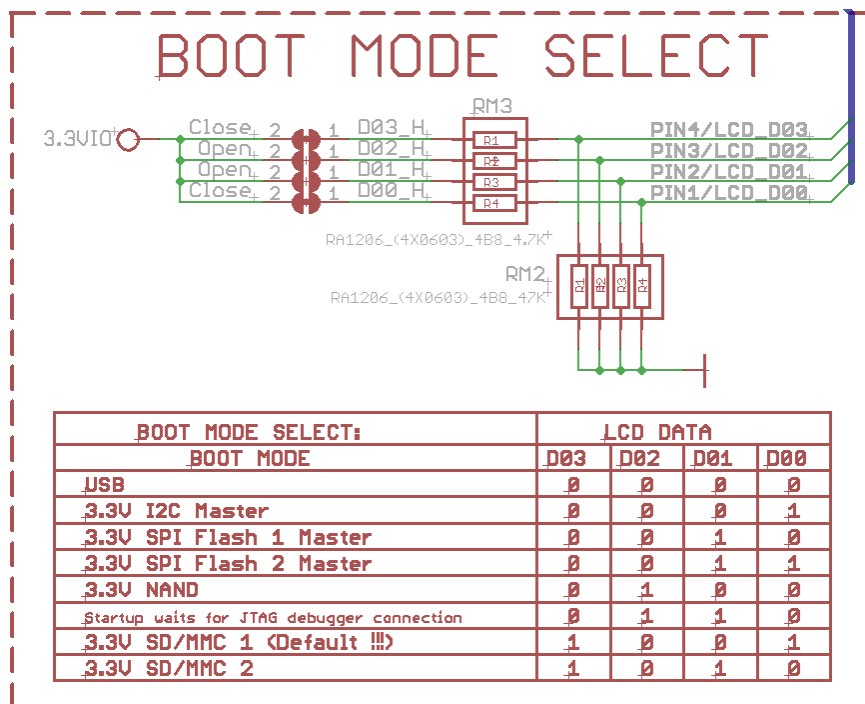


6.10 Boot mode positions

The iMX233 can boot the operating system from different locations. The default location for the Linux files we used is the microSD card. There are 4(four) jumpers responsible for the boot

location: D03, D02, D01 and D00. They are located on the top of the board between the processor and 40pin GPIO connector. Note that the jumpers are SMD type and opening a jumper would require cutting, closing a jumper would require soldering. To be able to do the quoted operations you will need basic engineering skills and experience. You can check below the table or the schematic for the correct positions. Value of “1” means the jumper is closed.

BOOT MODE	D03	d02	d01	d00
USB	0	0	0	0
3.3V I2C Master	0	0	0	1
3.3V SPI Flash 1 Master	0	0	1	0
3.3V SPI Flash 2 Master	0	0	1	1
3.3V NAND	0	1	0	0
Start up waits for JTAG debugger connection	0	1	1	0
3.3V SD/MMC 1 (Default !!!)	1	0	0	1
3.3V SD/MMC 2	1	0	1	0



6.11 Jumper description

Please note that all the jumpers on the board are SMD type. If you feel insecure in your soldering/cutting technique it is better not to try to adjust the jumpers.

6.12.1 SCL_SW/SCL_HW and SDA_SW/SDA_HW

Those two jumpers must be moved together – there are two available options – configuring software I2C interface (SCL_SW, SDA_SW) or hardware I2C interface (SCL_HW, SDA_HW positions).

The default position is SCL_HW/SDA_HW (hardware SCL).

6.12.2 5V_E

The 5V_E jumper allows control over the powering line. If you want to disable the 5V powering line open 5V_E jumper.

The default position is closed.

6.12.3 3.3VIO_E and 3.3V_E jumpers

Important these jumpers must be moved together! Closing both of 3.3V_E and 3.3VIO_E at the same time will damage the processor. One of them has to be open(unsoldered)Important these jumpers must be moved together! Closing both of 3.3V_E and 3.3VIO_E at the same time will damage the processor. One of them has to be open (unsoldered/cut). There are two possible positions:

- 1) 3.3V_E closed (soldered), 3.3VIO_E open (unsoldered/cut)
- 2) 3.3V_E open (unsoldered/cut), 3.3VIO_E closed (soldered)

In the default variant 1) the board uses the mounted by OLIMEX DC-DC 3.3V convertor which when the board is powered by external supply is the better alternative. However, if you use battery it will not power the chip handling the USB and the LAN functionality. Basically the board will lack USB-HOSTs and LAN when jumpers are set in variant 1) and the board is powered by a battery via the BAT connector.

If you set the jumpers in 2) state the board's 3.3V will be handled by the built-in iMX233 DC-DC but the problem is it isn't configured to be used with another chip so it doesn't provide enough current for all the USB-HOSTs and the LAN which might cause some of the devices plugged in the

USB to lack power and also might burn the chip if too much current is drawn. However, in this 2) mode of jumpers it is possible to power the board from a battery and still have some current on the USB-HOSTs and the LAN. The maximum current the built-in DC-DC can provide safely is 200mA.

6.12.4 Boot mode selecting jumpers

The boot mode is discussed in chapter 6.11 of this manual.

6.13 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 – when Linux is running pressing PWR_BUT will put the board in low power mode; when powered by battery PWR_BUT is used to initially power up the board – and pressing the board again after it is powered will put it low power mode

Recovery pads (REC) - these can be used for attaching a recovery button; when powered by battery the recovery button is used to bring the processor to normal power mode – you can short-circuit the pads for this feature without having to mount a button

512 (32M x 16) MBit DDR SDRAM - the exact memory used in the first revisions of the board is Xylinx HY5DU121622D

LED1 + Power LED

6.14 Accessories

Here you will find additional information for Olimex products you can use with OLinuXino-MINI purchase

6.14.1 USB-SERIAL-CABLE-F

The cable for the U_DEBUG interface that can be purchased for additional cost has three cables. It is important to specify in your purchase order whether you want the USB-SERIAL-CABLE-F variant with male or female connectors.

CHAPTER 7 BLOCK DIAGRAM AND MEMORY

7. Introduction to the chapter

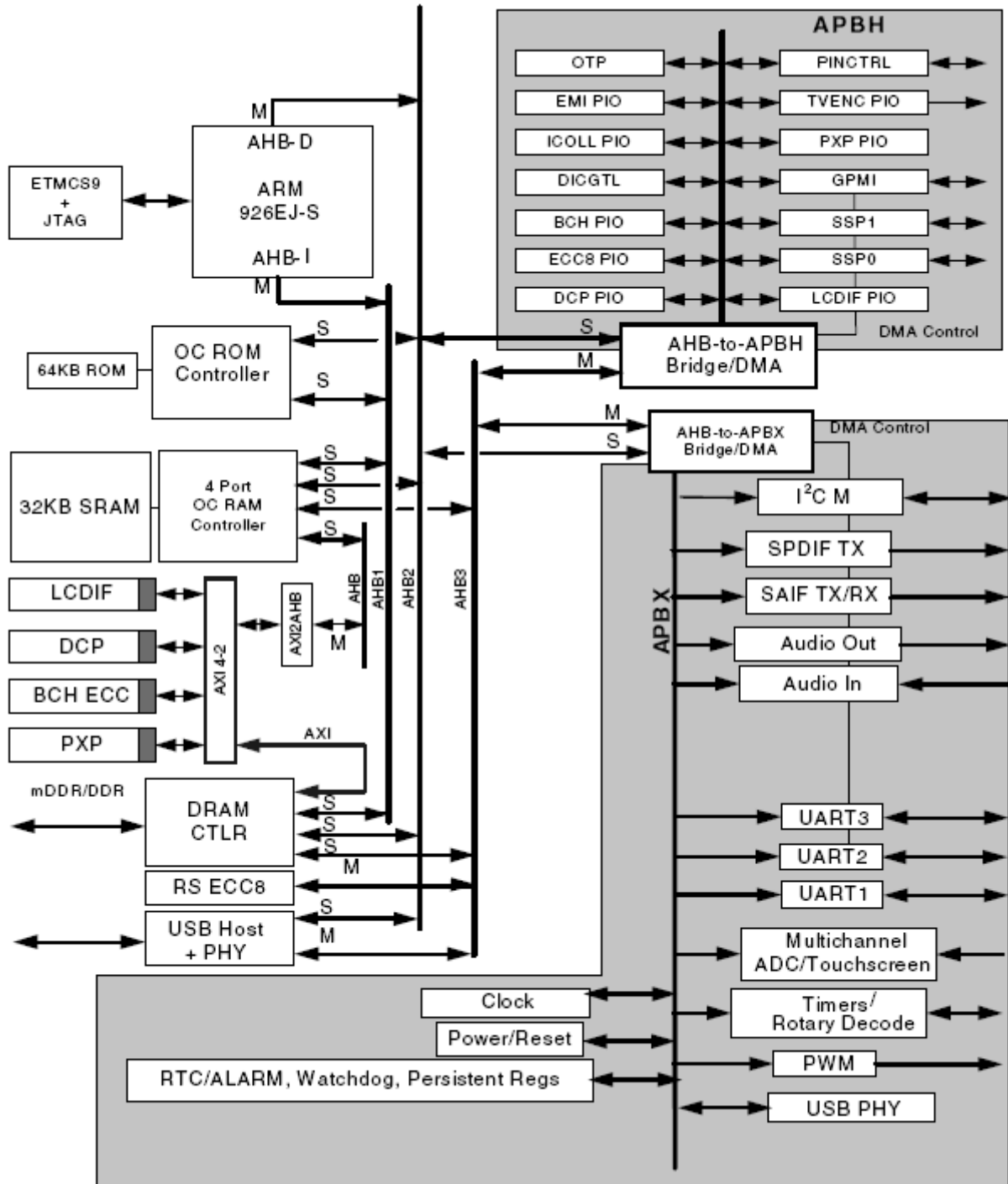
On the next page you can find a memory map for this family of processors. It is strongly recommended to refer to the original datasheet released by Freescale for one of higher quality.

7.1 Memory addresses

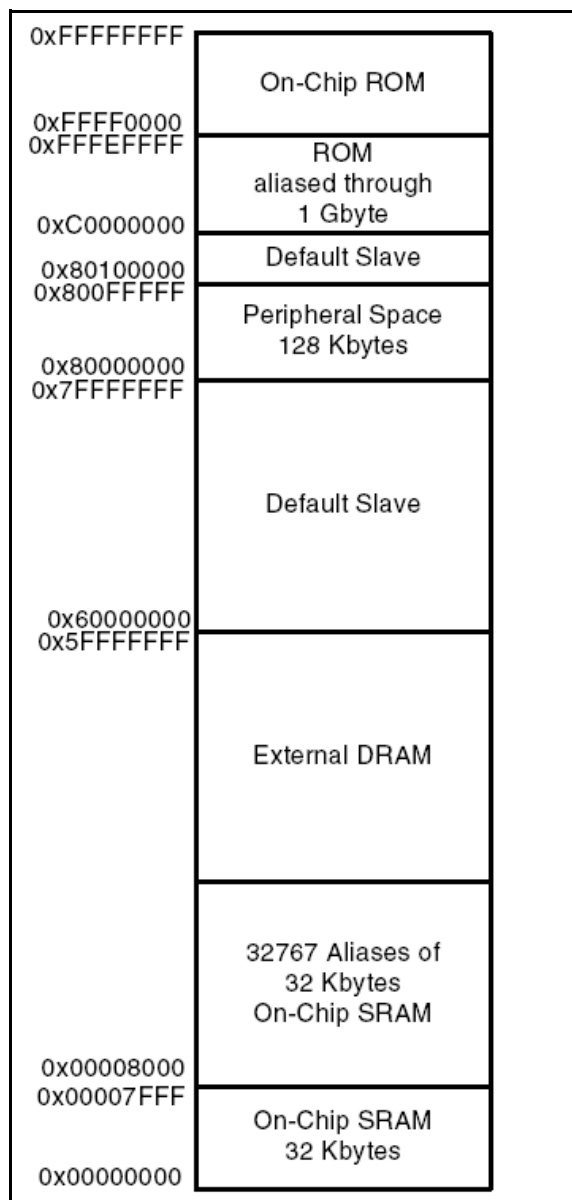
Below is the table with some of the most frequently used addresses. For full list of addresses check the manual released by Freescale (Chapter Memory Map).

Decode block	Device	Mnemonic	Start address	End address	Size
AHB	On-chip RAM	OCRAM	0x00000000	0x00007FFF	32KB
	On-chip RAM alias	OCRAM	0x00008000	0x3FFFFFFF	
	External memory		0x40000000	0x5FFFFFFF	512MB
	Default Slave		0x60000000	0x7FFFFFFF	512M

7.2 Processor block diagram



7.3 Physical memory map



CHAPTER 8 SCHEMATICS

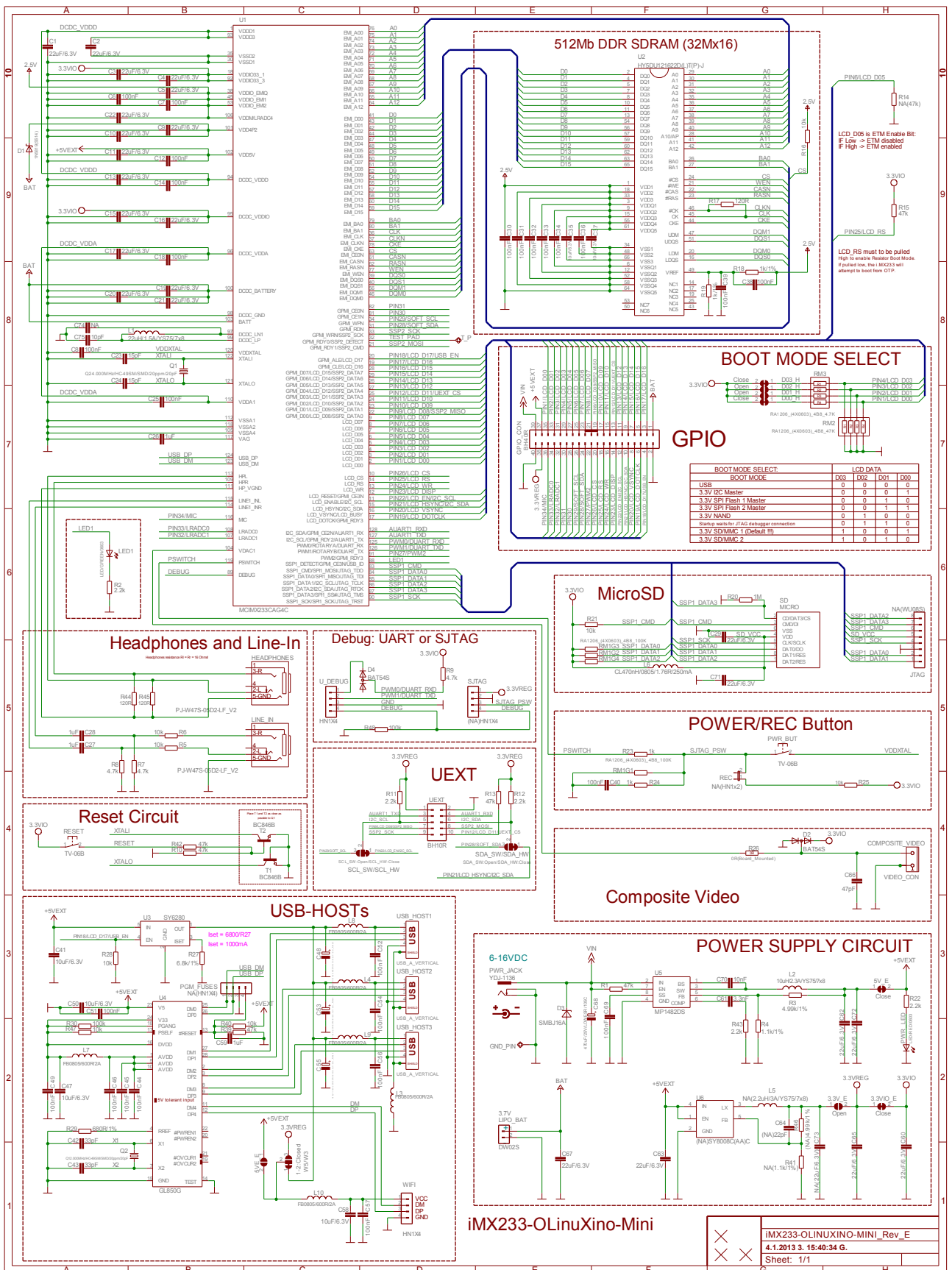
8. Introduction to the chapter

In this chapter are located the schematics describing logically and physically OLinuXino.

8.1 Eagle schematic

OLinuXino schematic is visible for reference here. You can also find them on the web page for OLinuXino at our site: <http://www.olimex.com/dev/imx233-OLinuXino-MINI.html>. They are located in HARDWARE section.

The EAGLE schematic is situated on the next page for quicker reference.



BOOT MODE SELECT

BOOT MODE SELECT	D03	D02	D01	D00
USB	0	0	0	0
3.3V I2C Master	0	0	1	0
3.3V SPI Flash 1 Master	0	0	1	1
3.3V SPI Flash 2 Master	0	1	0	1
3.3V NAND	0	1	1	0
Startup waits for JTAG debugger connection	1	0	1	0
3.3V SD/MMC 1 (Default em)	1	0	0	1
3.3V SD/MMC 2	1	0	1	1

Startup waits for JTAG debugger connection
3.3V SD/MMC 1 (Default em)
3.3V SD/MMC 2

GPIO

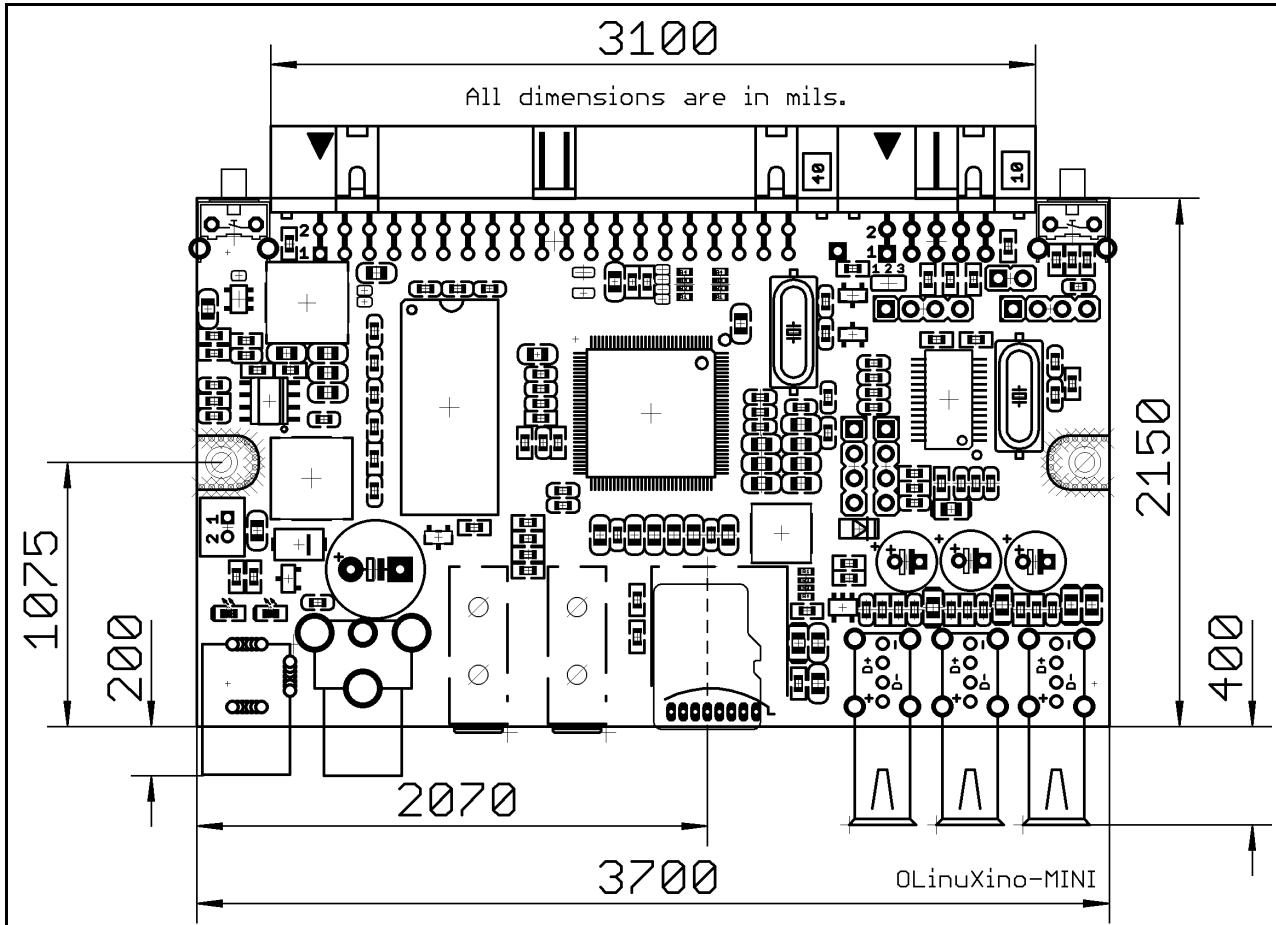
GPIO	Pin	Function
GPIO_00	1	GPIO_00
GPIO_01	2	GPIO_01
GPIO_02	3	GPIO_02
GPIO_03	4	GPIO_03
GPIO_04	5	GPIO_04
GPIO_05	6	GPIO_05
GPIO_06	7	GPIO_06
GPIO_07	8	GPIO_07
GPIO_08	9	GPIO_08
GPIO_09	10	GPIO_09
GPIO_10	11	GPIO_10
GPIO_11	12	GPIO_11
GPIO_12	13	GPIO_12
GPIO_13	14	GPIO_13
GPIO_14	15	GPIO_14
GPIO_15	16	GPIO_15
GPIO_16	17	GPIO_16
GPIO_17	18	GPIO_17
GPIO_18	19	GPIO_18
GPIO_19	20	GPIO_19
GPIO_20	21	GPIO_20
GPIO_21	22	GPIO_21
GPIO_22	23	GPIO_22
GPIO_23	24	GPIO_23
GPIO_24	25	GPIO_24
GPIO_25	26	GPIO_25
GPIO_26	27	GPIO_26
GPIO_27	28	GPIO_27
GPIO_28	29	GPIO_28
GPIO_29	30	GPIO_29
GPIO_30	31	GPIO_30
GPIO_31	32	GPIO_31
GPIO_32	33	GPIO_32
GPIO_33	34	GPIO_33
GPIO_34	35	GPIO_34
GPIO_35	36	GPIO_35
GPIO_36	37	GPIO_36
GPIO_37	38	GPIO_37
GPIO_38	39	GPIO_38
GPIO_39	40	GPIO_39
GPIO_40	41	GPIO_40
GPIO_41	42	GPIO_41
GPIO_42	43	GPIO_42
GPIO_43	44	GPIO_43
GPIO_44	45	GPIO_44
GPIO_45	46	GPIO_45
GPIO_46	47	GPIO_46
GPIO_47	48	GPIO_47
GPIO_48	49	GPIO_48
GPIO_49	50	GPIO_49
GPIO_50	51	GPIO_50
GPIO_51	52	GPIO_51
GPIO_52	53	GPIO_52
GPIO_53	54	GPIO_53
GPIO_54	55	GPIO_54
GPIO_55	56	GPIO_55
GPIO_56	57	GPIO_56
GPIO_57	58	GPIO_57
GPIO_58	59	GPIO_58
GPIO_59	60	GPIO_59
GPIO_60	61	GPIO_60
GPIO_61	62	GPIO_61
GPIO_62	63	GPIO_62
GPIO_63	64	GPIO_63
GPIO_64	65	GPIO_64
GPIO_65	66	GPIO_65
GPIO_66	67	GPIO_66
GPIO_67	68	GPIO_67
GPIO_68	69	GPIO_68
GPIO_69	70	GPIO_69
GPIO_70	71	GPIO_70
GPIO_71	72	GPIO_71
GPIO_72	73	GPIO_72
GPIO_73	74	GPIO_73
GPIO_74	75	GPIO_74
GPIO_75	76	GPIO_75
GPIO_76	77	GPIO_76
GPIO_77	78	GPIO_77
GPIO_78	79	GPIO_78
GPIO_79	80	GPIO_79
GPIO_80	81	GPIO_80
GPIO_81	82	GPIO_81
GPIO_82	83	GPIO_82
GPIO_83	84	GPIO_83
GPIO_84	85	GPIO_84
GPIO_85	86	GPIO_85
GPIO_86	87	GPIO_86
GPIO_87	88	GPIO_87
GPIO_88	89	GPIO_88
GPIO_89	90	GPIO_89
GPIO_90	91	GPIO_90
GPIO_91	92	GPIO_91
GPIO_92	93	GPIO_92
GPIO_93	94	GPIO_93
GPIO_94	95	GPIO_94
GPIO_95	96	GPIO_95
GPIO_96	97	GPIO_96
GPIO_97	98	GPIO_97
GPIO_98	99	GPIO_98
GPIO_99	100	GPIO_99

MicroSD

MicroSD	Pin	Function
SSP1_CMD	1	SSP1_CMD
SSP1_DATA0	2	SSP1_DATA0
SSP1_DATA1	3	SSP1_DATA1
SSP1_DATA2	4	SSP1_DATA2
SSP1_DATA3	5	SSP1_DATA3
SSP1_DATA4	6	SSP1_DATA4
SSP1_DATA5	7	SSP1_DATA5
SSP1_DATA6	8	SSP1_DATA6
SSP1_DATA7	9	SSP1_DATA7
SSP1_DATA8	10	SSP1_DATA8
SSP1_DATA9	11	SSP1_DATA9
SSP1_DATA10	12	SSP1_DATA10
SSP1_DATA11	13	SSP1_DATA11
SSP1_DATA12	14	SSP1_DATA12
SSP1_DATA13	15	SSP1_DATA13
SSP1_DATA14	16	SSP1_DATA14
SSP1_DATA15	17	SSP1_DATA15
SSP1_CMD	18	SSP1_CMD
SSP1_DATA0	19	SSP1_DATA0
SSP1_DATA1	20	SSP1_DATA1
SSP1_DATA2	21	SSP1_DATA2
SSP1_DATA3	22	SSP1_DATA3
SSP1_DATA4	23	SSP1_DATA4
SSP1_DATA5	24	SSP1_DATA5
SSP1_DATA6	25	SSP1_DATA6
SSP1_DATA7	26	SSP1_DATA7
SSP1_DATA8	27	SSP1_DATA8
SSP1_DATA9	28	SSP1_DATA9
SSP1_DATA10	29	SSP1_DATA10
SSP1_DATA11	30	SSP1_DATA11
SSP1_DATA12	31	SSP1_DATA12
SSP1_DATA13	32	SSP1_DATA13
SSP1_DATA14	33	SSP1_DATA14
SSP1_DATA15	34	SSP1_DATA15
SSP1_CMD	35	SSP1_CMD
SSP1_DATA0	36	SSP1_DATA0
SSP1_DATA1	37	SSP1_DATA1
SSP1_DATA2	38	SSP1_DATA2
SSP1_DATA3	39	SSP1_DATA3
SSP1_DATA4	40	SSP1_DATA4
SSP1_DATA5	41	SSP1_DATA5
SSP1_DATA6	42	SSP1_DATA6
SSP1_DATA7	43	SSP1_DATA7
SSP1_DATA8	44	SSP1_DATA8
SSP1_DATA9	45	SSP1_DATA9
SSP1_DATA10	46	SSP1_DATA10
SSP1_DATA11	47	SSP1_DATA11
SSP1_DATA12	48	SSP1_DATA12
SSP1_DATA13	49	SSP1_DATA13
SSP1_DATA14	50	SSP1_DATA14
SSP1_DATA15	51	SSP1_DATA15
SSP1_CMD	52	SSP1_CMD
SSP1_DATA0	53	SSP1_DATA0
SSP1_DATA1	54	SSP1_DATA1
SSP1_DATA2	55	SSP1_DATA2
SSP1_DATA3	56	SSP1_DATA3
SSP1_DATA4	57	SSP1_DATA4
SSP1_DATA5	58	SSP1_DATA5
SSP1_DATA6	59	SSP1_DATA6
SSP1_DATA7	60	SSP1_DATA7
SSP1_DATA8	61	SSP1_DATA8
SSP1_DATA9	62	SSP1_DATA9
SSP1_DATA10	63	SSP1_DATA10
SSP1_DATA11	64	SSP1_DATA11
SSP1_DATA12	65	SSP1_DATA12
SSP1_DATA13	66	SSP1_DATA13
SSP1_DATA14	67	SSP1_DATA14
SSP1_DATA15	68	SSP1_DATA15
SSP1_CMD	69	SSP1_CMD
SSP1_DATA0	70	SSP1_DATA0
SSP1_DATA1	71	SSP1_DATA1
SSP1_DATA2	72	SSP1_DATA2
SSP1_DATA3	73	SSP1_DATA3
SSP1_DATA4	74	SSP1_DATA4
SSP1_DATA5	75	SSP1_DATA5
SSP1_DATA6	76	SSP1_DATA6
SSP1_DATA7	77	SSP1_DATA7
SSP1_DATA8	78	SSP1_DATA8
SSP1_DATA9	79	SSP1_DATA9
SSP1_DATA10	80	SSP1_DATA10
SSP1_DATA11	81	SSP1_DATA11
SSP1_DATA12	82	SSP1_DATA12
SSP1_DATA13	83	SSP1_DATA13
SSP1_DATA14	84	SSP1_DATA14
SSP1_DATA15	85	SSP1_DATA15
SSP1_CMD	86	SSP1_CMD
SSP1_DATA0	87	SSP1_DATA0
SSP1_DATA1	88	SSP1_DATA1
SSP1_DATA2	89	SSP1_DATA2
SSP1_DATA3	90	SSP1_DATA3
SSP1_DATA4	91	SSP1_DATA4
SSP1_DATA5	92	SSP1_DATA5
SSP1_DATA6	93	SSP1_DATA6
SSP1_DATA7	94	SSP1_DATA7
SSP1_DATA8	95	SSP1_DATA8
SSP1_DATA9	96	SSP1_DATA9
SSP1_DATA10	97	SSP1_DATA10
SSP1_DATA11	98	SSP1_DATA11
SSP1_DATA12	99	SSP1_DATA12
SSP1_DATA13	100	SSP1_DATA13
SSP1_DATA14	101	SSP1_DATA14
SSP1_DATA15	102	SSP1_DATA15
SSP1_CMD	103	SSP1_CMD
SSP1_DATA0	104	SSP1_DATA0
SSP1_DATA1	105	SSP1_DATA1
SSP1_DATA2	106	SSP1_DATA2
SSP1_DATA3	107	SSP1_DATA3
SSP1_DATA4	108	SSP1_DATA4
SSP1_DATA5	109	SSP1_DATA5
SSP1_DATA6	110	SSP1_DATA6
SSP1_DATA7	111	SSP1_DATA7
SSP1_DATA8	112	SSP1_DATA8
SSP1_DATA9	113	SSP1_DATA9
SSP1_DATA10	114	SSP1_DATA10
SSP1_DATA11	115	SSP1_DATA11
SSP1_DATA12	116	SSP1_DATA12
SSP1_DATA13	117	SSP1_DATA13
SSP1_DATA14	118	SSP1_DATA14
SSP1_DATA15	119	SSP1_DATA15
SSP1_CMD	120	SSP1_CMD
SSP1_DATA0	121	SSP1_DATA0
SSP1_DATA1	122	SSP1_DATA1
SSP1_DATA2	123	SSP1_DATA2
SSP1_DATA3	124	SSP1_DATA3
SSP1_DATA4	125	SSP1_DATA4
SSP1_DATA5	126	SSP1_DATA5
SSP1_DATA6	127	SSP1_DATA6
SSP1_DATA7	128	SSP1_DATA7
SSP1_DATA8	129	SSP1_DATA8
SSP1_DATA9	130	SSP1_DATA9
SSP1_DATA10	131	SSP1_DATA10
SSP1_DATA11	132	SSP1_DATA11
SSP1_DATA12	133	SSP1_DATA12
SSP1_DATA13	134	SSP1_DATA13
SSP1_DATA14	135	SSP1_DATA14
SSP1_DATA15	136	SSP1_DATA15
SSP1_CMD	137	SSP1_CMD
SSP1_DATA0	138	SSP1_DATA0
SSP1_DATA1	139	SSP1_DATA1
SSP1_DATA2	140	SSP1_DATA2
SSP1_DATA3	141	SSP1_DATA3
SSP1_DATA4	142	SSP1_DATA4
SSP1_DATA5	143	SSP1_DATA5
SSP1_DATA6	144	SSP1_DATA6
SSP1_DATA7	145	SSP1_DATA7
SSP1_DATA8	146	SSP1_DATA8
SSP1_DATA9	147	SSP1_DATA9
SSP1_DATA10	148	SSP1_DATA10
SSP1_DATA11	149	SSP1_DATA11
SSP1_DATA12	150	SSP1_DATA12
SSP1_DATA13	151	SSP1_DATA13
SSP1_DATA14	152	SSP1_DATA14
SSP1_DATA15	153	SSP1_DATA15
SSP1_CMD	154	SSP1_CMD
SSP1_DATA0	155	SSP1_DATA0
SSP1_DATA1	156	SSP1_DATA1
SSP1_DATA2	157	SSP1_DATA2
SSP1_DATA3	158	SSP1_DATA3
SSP1_DATA4	159	SSP1_DATA4
SSP1_DATA5	160	SSP1_DATA5
SSP1_DATA6	161	SSP1_DATA6
SSP1_DATA7	162	SSP1_DATA7
SSP1_DATA8	163	SSP1_DATA8
SSP1_DATA9	164	SSP1_DATA9
SSP1_DATA10	165	SSP1_DATA10
SSP1_DATA11	166	SSP1_DATA11
SSP1_DATA12	167	SSP1_DATA12
SSP1_DATA13	168	SSP1_DATA13
SSP1_DATA14	169	SSP1_DATA14
SSP1_DATA15	170	SSP1_DATA15
SSP1_CMD	171	SSP1_CMD
SSP1_DATA0	172	SSP1_DATA0
SSP1_DATA1	173	SSP1_DATA1
SSP1_DATA2	174	SSP1_DATA2
SSP1_DATA3	175	SSP1_DATA3
SSP1_DATA4	176	SSP1_DATA4
SSP1_DATA5	177	SSP1_DATA5
SSP1_DATA6	178	SSP1_DATA6
SSP1_DATA7	179	SSP1_DATA7
SSP1_DATA8	180	SSP1_DATA8
SSP1_DATA9	181	SSP1_DATA9
SSP1_DATA10	182	SSP1_DATA10
SSP1_DATA11	183	SSP1_DATA11
SSP1_DATA12	184	SSP1_DATA12
SSP1_DATA13	185	SSP1_DATA13
SSP1_DATA14	186	SSP1_DATA14
SSP1_DATA15	187	SSP1_DATA15
SSP1_CMD	188	SSP1_CMD
SSP1_DATA0	189	SSP1_DATA0
SSP1_DATA1	190	SSP1_DATA1
SSP1_DATA2	191	SSP1_DATA2
SSP1_DATA3	192	SSP1_DATA3
SSP1_DATA4	193	SSP1_DATA4
SSP1_DATA5	194	SSP1_DATA5
SSP1_DATA6	195	SSP1_DATA6
SSP1_DATA7	196	SSP1_DATA7
SSP1_DATA8	197	SSP1_DATA8
SSP1_DATA9	198	SSP1_DATA9
SSP1_DATA10	199	SSP1_DATA10
SSP1_DATA11	200	SSP1_DATA11
SSP1_DATA12	201	SSP1_DATA12

8.2 Physical dimensions

Note that all dimensions are in inches.



The three highest elements on the board in order from the tallest to the shortest are: capacitor C68 – 16.2mm (0.640") over the pcb; USB host connectors – 13.6mm (0.525"); composite connector – 12.0mm(0.464").

CHAPTER 9 REVISION HISTORY AND SUPPORT

9. 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.

9.1 Document revision

Revision	Changes	Modified Page#
A, 02.07.12	Initial Creation	All
B, 03.07.12	Page 21 - added info about the U_DEBUG Page 34 - info about internal and external DC-DC adjusted	21,34
C, 05.07.12	Page 6 - corrected info about different WIFI options Page 20 - adjusted information about using U_DEBUG Various - adjust various small mistakes	6, 20, various
D, 18.07.12	Pages 10, 11 - Added info about fuse programming	10, 11

Revision	Changes	Modified Page#
E, 29.10.12	Page 16, 17, 18 - Fixed wrong info about the LQFP128 package of iMX233	16, 17, 18
F, 04.01.12	Page 10 - Wrong wiring for fuse burning fixed Page 43 - Changed schematic with latest one Page 46 - Added info about the latest hardware revision	10, 43, 46

9.2 Board revision

Revision	Notable Changes
C	- Initial release of the board
D	<ul style="list-style-type: none"> - R31 to R38 are changed to NA, because into the newest GL850G datasheet(schematic) they are removed - In addition PSELF is connected to V33 pin through 10k resistor with the aim of GL850G to be set in self-powered mode - Altered inductors: L1,L2 i L5 to 7x8mm package. - All capacitors 22uF/0805 are now changed to 22uF/0603 - Added prints of the names of WIFI and PGM_FUSES connectors - New powering scheme with jumpers for the WIFI - Removed FET1 due to resistance and because TVS works also as a protection - PWR_LED and LED1 were too bright, the resistor is now 2.2k - WIFI connector has been moved with 1mm near the quartz so it is far enough from the capacitors - Added 100k pull-down resistor on the debug line. According to the IMX23RM, page 1368: If the DEBUG pin is unused, Freescale recommends pulling the

	<p>DEBUG pin to ground through a 100K resistor.</p> <ul style="list-style-type: none">- Changed the text DEBUD to DEBUG into the SJTAG's connector bottom silk.- Added Schottky diode and increased R9 to 4.7K because of bug that might appear if using- Added C75(10pF)- Added ferrite bead 2A on the 4th USB of the hub also (wifi only)
E	<ul style="list-style-type: none">- Moved the print of the names of PWR_JACK and 6-16VDC since they were overlapping; The Composite Video print is also adjusted. LINE_IN and HEADPHONES are also moved a bit. R44 and R55 too.- The jumpers SW_SCL/HW_SCL and SW_SDA/HW_SDA position is changed to default HW (hardware SCL)- PGM_FUSES line is changed 10mils to the upper side, C48 is moved 5mils to the bottom edge.

9.3 Useful web links and purchase codes

The web page you can visit for more info on your device is <http://www.olimex.com/dev/imx233-OLinuXino-MINI.html>, <https://www.olimex.com/wiki/IMX233>

You can get the latest updates on the software at: <https://github.com/OLIMEX/OLINUXINO>.

The OLinuXino Linux images sources: <https://github.com/Freescale/fsl-community-bsp-platform>.

ORDER CODES:

iMX233-OLinuXino-MAXI – the best version of OLinuXino featuring Ethernet controller

iMX233-OLinuXino-MINI – the mini version of the single-board computer

iMX233-OLinuXino-MICRO – tiny in size, tiny in price and perfect for breadboarding

iMX233-OLinuXino-MINI-WIFI – the MINI version of OLinuXino + embedded RTL8128CU WIFI module

iMX233-OLinuXino-SD - SD card with the Linux image which can be used with every board from the OLinuXino family

MOD-WIFI_RTL8188 – external USB WIFI modem with RTL8188 chip

USB-SERIAL-CABLE - USB serial console cable for U_DEBUG

SY0612E - power supply adapter 12V/0.5A for iMX233-OLinuXino-MAXI and iMX233-OLinuXino-MINI (Will not work with OLinuXino-MICRO)

SY0605E - power supply adapter 5V/1A for iMX233-OLinuXino-MICRO (Will not work with iMX233-OLinuXino-MAXI and iMX233-OLinuXino-MINI)

The latest price list can be found at <https://www.olimex.com/prices>.

How to order?

You can order from us directly or from any of our distributors.

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

The full list of distributors can be found here: <https://www.olimex.com/Distributors/>.

9.3 Product support

For product support, hardware information and error reports mail to: support@olimex.com. 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.

Warranty and returns:

Our boards have lifetime warranty against manufacturing defects and components.

During development work it is not unlikely that you can burn your programmer or development board. This is normal, we also do development work and we have damaged A LOT of programmers and boards during our daily job so we know how it works. If our board/programmer has worked fine then stopped, please check if you didn't apply over voltage by mistake, or shorted something in your target board where the programmer was connected etc. Sometimes boards might get damaged by ESD shock voltage or if you spill coffee on them during your work when they are powered.

Please note that warranty do not cover problems caused by improper use, shorts, over-voltages, ESD shock etc.

If the board has warranty label it should be not broken. Broken labels void the warranty, same applies for boards modified by the customer, for instance soldering additional components or removing components - such boards will be not be a subject of our warranty.

If you are positive that the problem is due to manufacturing defect or component you can return the board back to us for inspection.

When we receive the board we will check and if the problem is caused due to our fault and we will repair/replace the faulty hardware free of charge, otherwise we can quote price of the repair.

Note that all shippings back and forth have to be covered by the customer. Before you ship anything back you need to ask for RMA. When you ship back please attach to it your shipping address, phone, e-mail, RMA# and brief description of the problem. All boards should be sent back in antistatic package and well packed to prevent damages during the transport.