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

Revision K, March 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

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-MICRO 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
- 1 USB High Speed Host
- three Buttons
- 2x30 pin GPIO for connection of other hardware
- PCB dimensions: 3.00" x 1.70" (76.2mm x 43.2mm)
- Nominal dimensions: 3.40" x 1.70" (86.4mm x 43.2mm)

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	6V-16V	6V-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
Built-in USB hub	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 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 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-MICRO optimally, the following items are required:

- 5V source of power with 1A maximum amperage.
- SJTAG interface programmer
- USB hub (USB splitter)
- USB keyboard
- Monitor with composite interface or Personal Computer + USB-SERIAL-CABLE
- 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)
SY0605E - power supply adapter 5V/1A for iMX233-OLinuXino-MICRO

2.4 Powering the board

The board is powered either via the PWR jack or via a battery. It should be supplied by 5V source with maximum current of 1A from the power jack.

All measures below are taken at 5V.

If measuring the current consumption it should be around 0.06A before initializing all the peripherals.

IMPORTANT! We discovered a situation which might leave some of the SD cards (iMX233-OLinuXino-SD) in unrecoverable state when powering OLinuXino-MICRO. The problem might occur if two specific conditions are met simultaneously:

1)Plugged iMX233-OLinuXino-SD micro SD card with holographic sticker on its back side (some of the cards we have distributed are from a brand that places holographic sticker on their backs, the other half lack such a sticker)
2)Plugged USB-SERIAL-CABLE-F at the moment when powering the board

If you happen to have received SD card with holographic sticker on its back side and you use it with OLinuXino-MICRO and you plug USB-SERIAL-CABLE and then you power the board there is a chance of malfunction of the SD card.

There are two possible workarounds to protect the SD card. The first one is simpler and the second one requires some soldering experience.

Workaround 1: First insert the iMX233-OLinuXino-SD card and then power the board (and if powering the board from a battery also press the PWR button). Wait 4-5 seconds and then connect the USB-SERIAL-CABLE-F. After the initial power-up it is safe to use the reset button.

Workaround 2: You will need a Shottky diode. The Shottky should be soldered on the USB-SERIAL-CABLE-F TX line/wire (RED cable) with anode towards the board.

When you power the board by battery you have to press the PWR_BUT to start the board. If you start Linux and it is already running no matter which powering method you use (PWR_JACK or BAT) pressing the PWR_BUT will put the Linux in power-save mode.

For the European customers we sell a power supply adapter 5V/1A – SY0605E.

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 adjust 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.

The first batches of the board and the SD-card used the Debian Linux image. After that we switched over to ArchLinux for the ease of the package manager. Instructions how to build the ArchLinux can be found at the gitHub address of OLinuXino.

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 connect a USB-A to USB-A cable between a computer and the board's USB-HOST connector.

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-compilant device from the "Select device" drop-down menu.
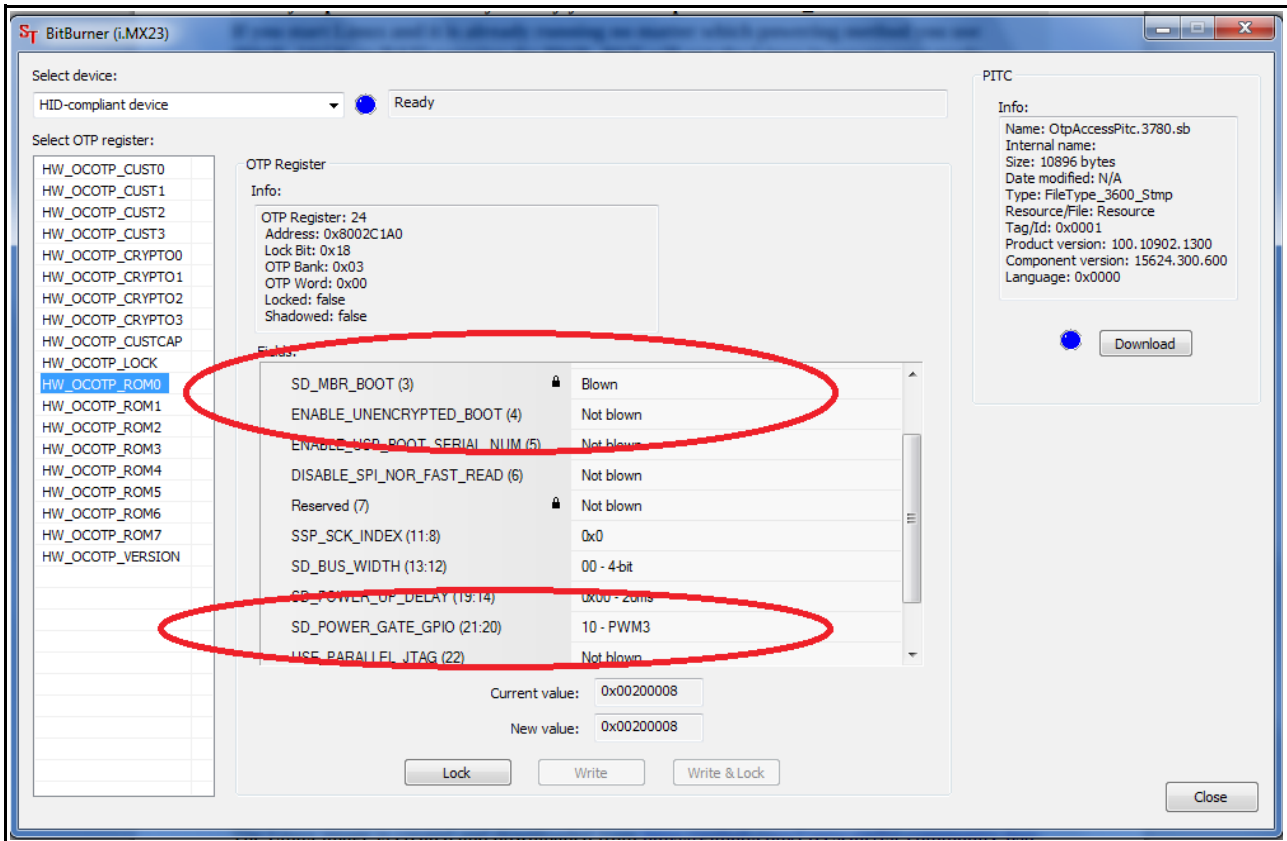
2.7 Building the Debian Linux image

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:
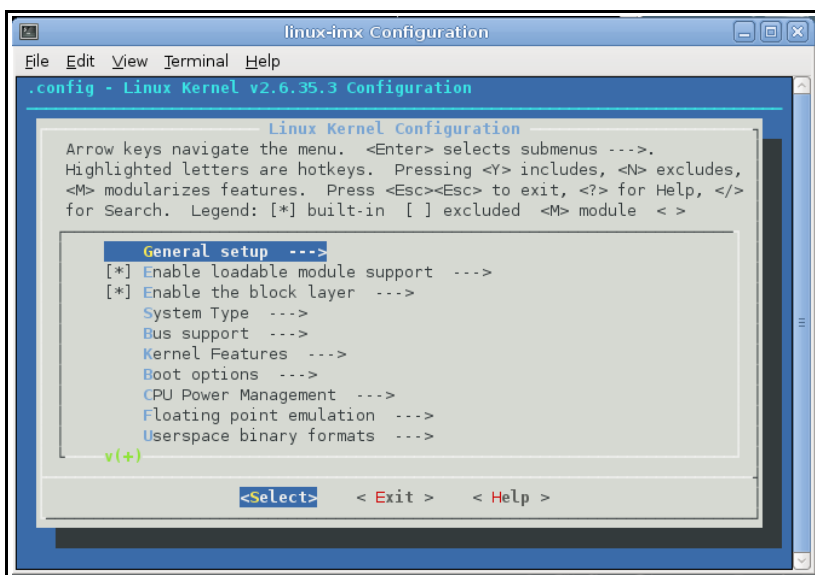
5) Now to start building the image:

```
~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 sub-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.

2.9 How to setup Arch Linux distribution

You can refer to the manual of OLinuXino-MAXI for general build instructions but mind we haven't tested the algorithm listed there on the MICRO. There shouldn't be significant differences. If you meet such head over to the OLIMEX forums.

2.10 How setup the I2C, SPI, UART

There are number of examples with our extension module board to achieve those connections on the UEXT. The examples might be used as an example for I2C, SPI or UART communication. You can find them at our GitHub page:
https://github.com/OLIMEX/OLINUXINO/tree/master/SOFTWARE/iMX233

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 might differ from the names used below to describe them. For the actual names check the OLinuXino board itself.

The top and the bottom view can be viewed on the next two pages of the manual. There is explanation of the different peripherals in the following chapters of the manual.

3.1 Layout (top view)

CON2

SJTAG

SD card connector

Composite video

SMD jumpers

USB_HOST

PWR jack

LED1   CON1   Reset   Power button   REC button

U_DEBUG
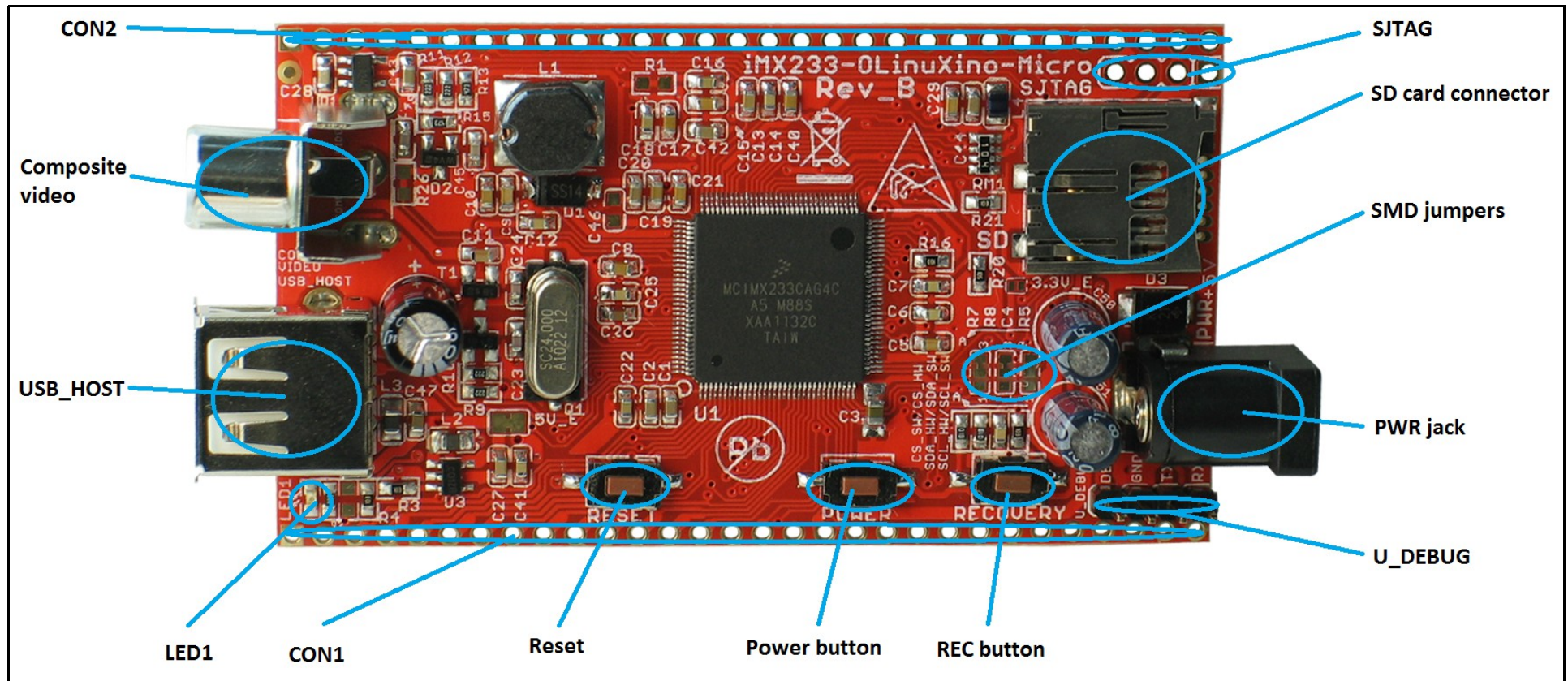
3.2 Layout (bottom view)
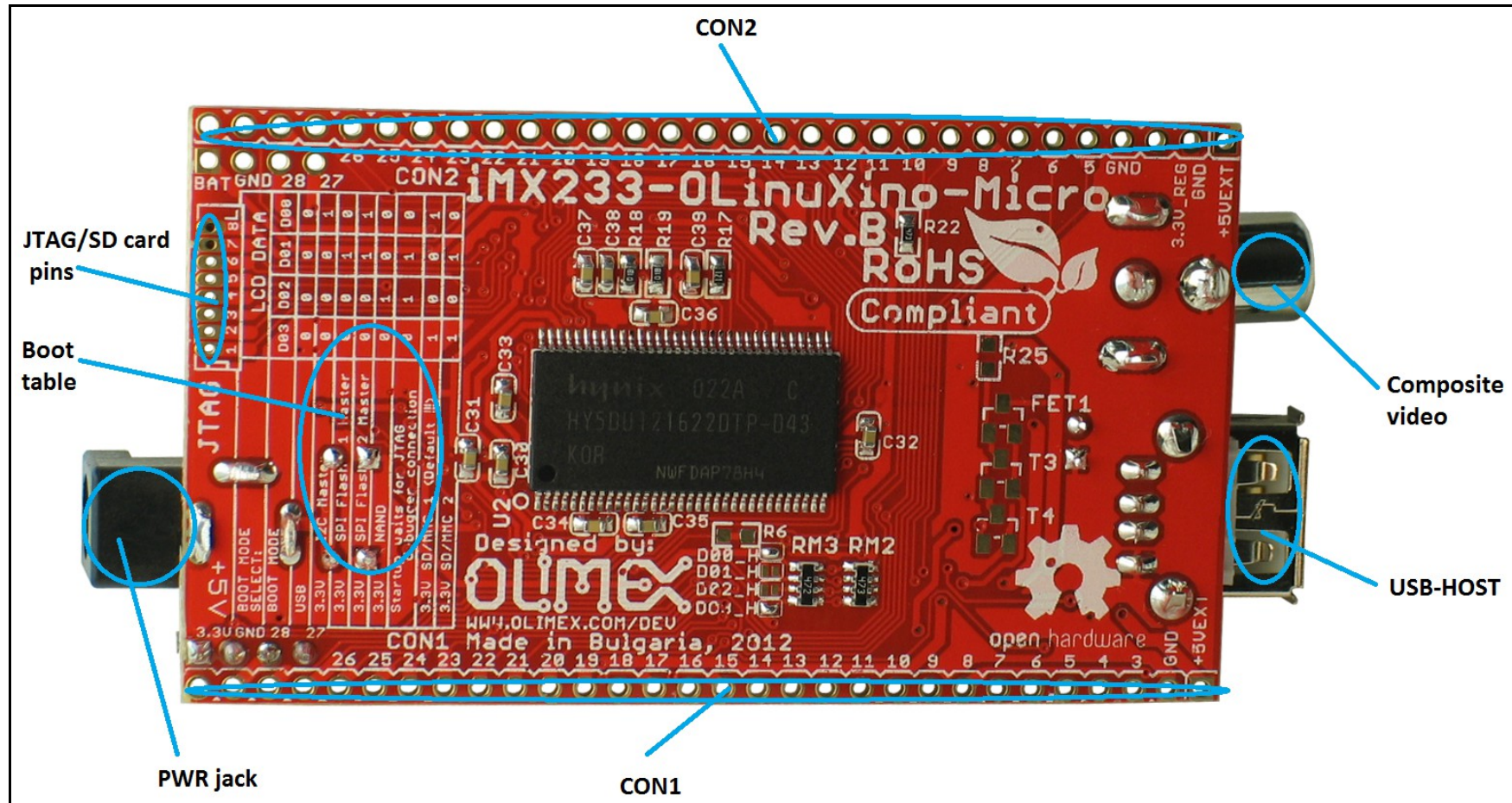
CHAPTER 4: THE iMX233 MICROCONTROLLER

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 onthe-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 t he 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.

At the moment of writing the microcontroller datasheet can be found at the following link: http://www.freescale.com/files/dsp/doc/ref_manual/IMX23RM.pdf

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

OLinuXino's reset circuit includes R9 (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.
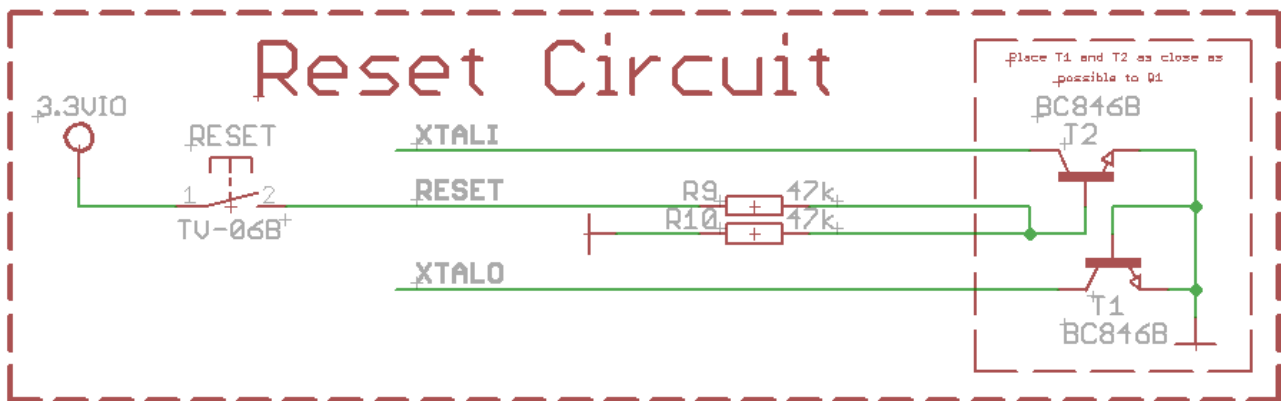
5.2 Clock

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

5.3 Power supply circuit

The board is powered by 5V supply and all measures listed below are taken at 5V external supply.

If measuring the current consumption it should be around 0.06A before initializing all the peripherals.

If you have successfully booted and powered the board the green LED1 will start blinking (you have to put a SD card with Linux image in the SD card holder).

POWER SUPPLY CIRCUIT

Only 5V power supply!!!!!
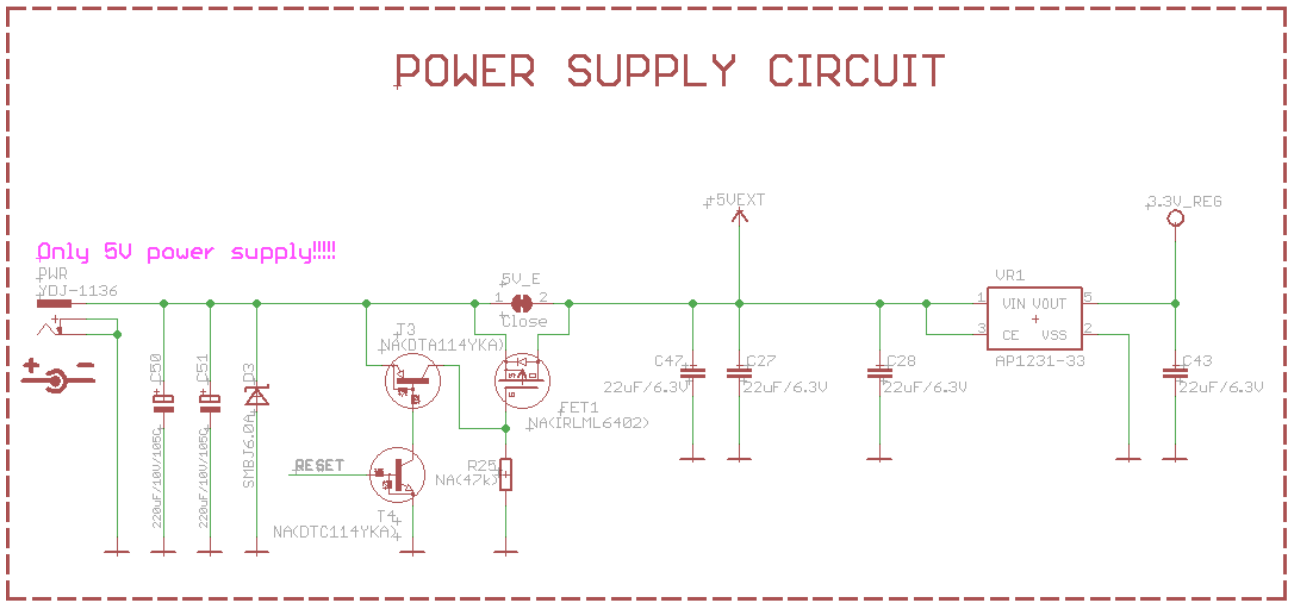
CHAPTER 6: CONNECTORS AND PINOUT

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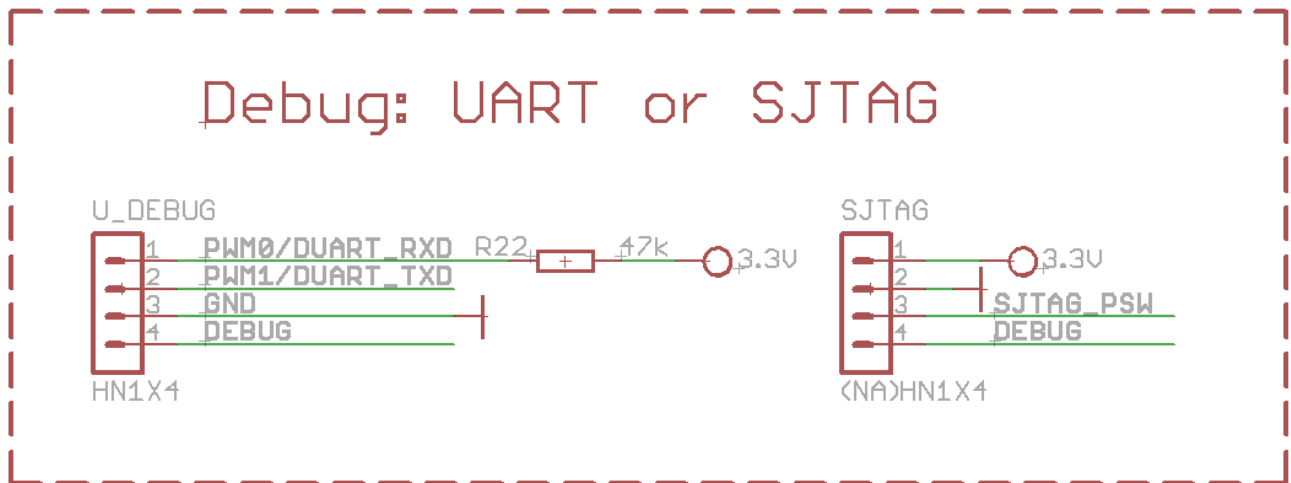
There are three debugging options available on the OLinuXino. It is preferable to use one of the first two - UART or SJTAG but in case you want to use the classic 6-wire parallel JTAG there is an option explained below.

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.

In order to avoid SD card malfunction, when using the USB-SERIAL-CABLE-F with OLinuXino-MICROit is advisable to first insert the iMX233-OLinuXino-SD card and then power the board (if using battery also press the PWR button). Wait 4-5 seconds and then connect the USB-SERIAL-CABLE-F.

Please also check "2.4 Powering the board" for detailed info.

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-MICROU_DEBUG connector in this sequence:

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

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
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-MICROunder 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.
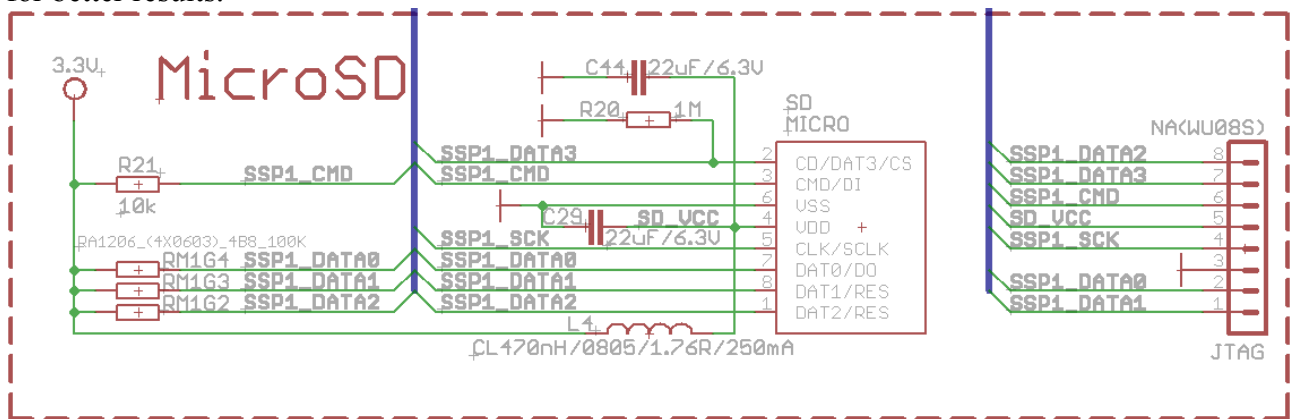
microSD card connector		
Pin #	**Signal Name**	**Processor Pin #**
1	SSP1_DATA2	86
2	SSP1_DATA3	87
3	SSP1_CMD	83
4	SD_VCC	-
5	SSP1_SCK	90
6	GND	30, 35, 98, 105, 112, 118
7	SSP1_DATA0	84
8	SSP1_DATA1	85
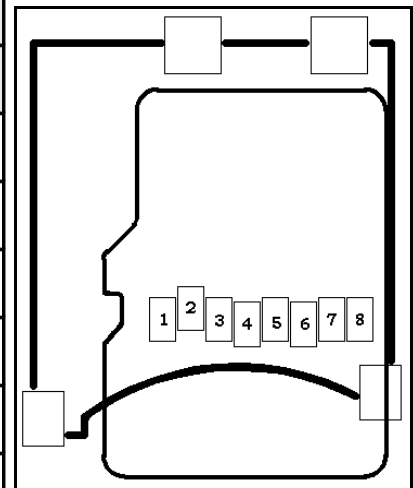
Notice that the pad numeration is written at the bottom of OlinuXino-MICRO under the microSD card connector.

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 connector

OLIMEX offers board extensions which can be connected via UEXT interface. OLinuXino-MICRO board lacks UEXT connector but it can be formed using the signals on the CON2 pad line. You can check which signals you need in the CON1 and CON2 chapter.

For more information on UEXT please visit:

http://www.olimex.com/dev/OTHER/UEXT.pdf
*Note the two jumpers PIN29/SOFT_CL – PIN22/LCD_EN/I2C_SCL, and PIN28/SOFT_SDA3 – PIN21/LCD_HSYNC/I2C_SDA which by default are set to a software I2C. They can be used to set the whole UEXT to a hardware I2C.

Note that the correct way of connecting modules to the UEXT is:
1. Plug module to host board via UEXT
2. Power host board
If you first power the board and then establish the UEXT it is possible host-board restarting to occur due to the fact that some modules have large capacitors on the power line.
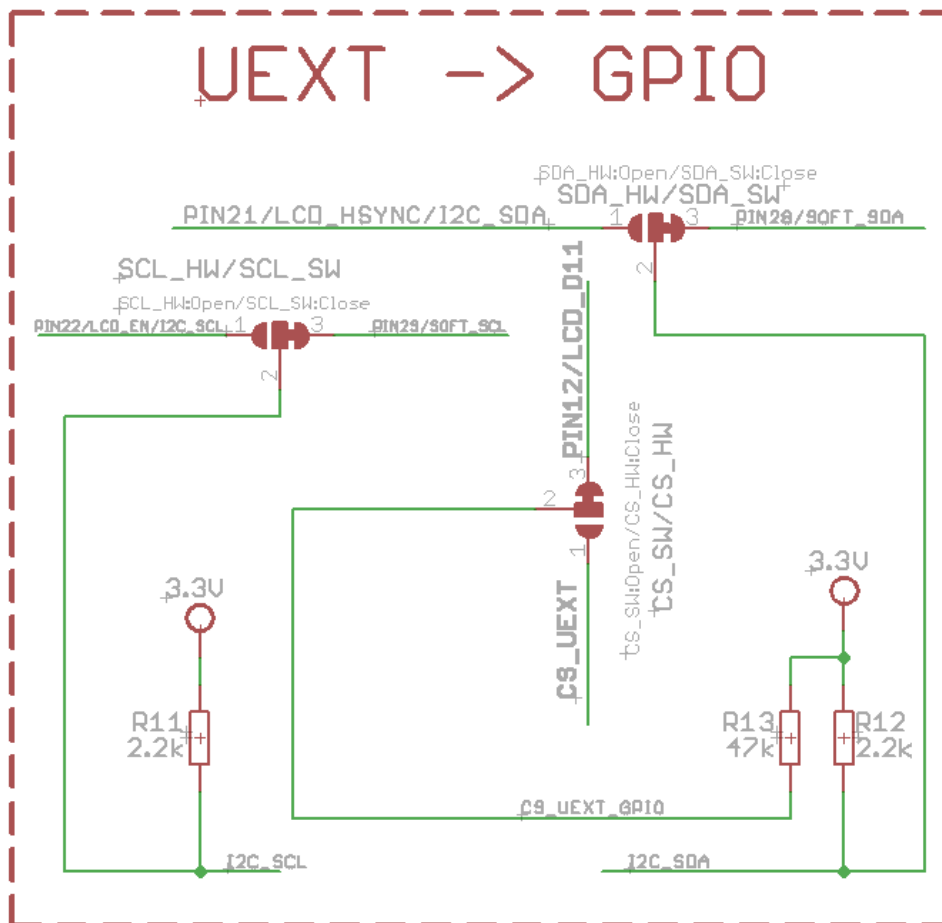
UEXT connector		
Pin #	**Signal Name**	**Processor Pin #**
1	+3.3VREG	-
2	GND	30, 35, 98, 105, 112, 118
3	AUART1_TXD	127
4	AUART1_RXD	128
5	I2C_SCL	34(default) OR 11*
6	I2C_SDA	31(default) OR 15*
7	PIN9/LCD_D08/SSP2_MISO	22
8	SSP2_MOSI	21
9	SSP2_SCK	33
10	PIN12/LCD_D11/UEXT_CS	25

6.4 CON1 and CON2 pads

The GPIO pins are led out to CON1 and CON2 pads situated on the both sides of the board. They allow the user to attach additional hardware, check readings or perform hardware debug. The "Pin#" column shows connector number and does NOT represent the naming on the bottom of the board. The table is divided into two big columns and explains the pin of both the connectors CON1 and CON2.

There are three color sections in the table below. Each color represents signals needed to establish a specific connection and functionality like in the MAXI and the MINI versions of the board. The GREEN section (CON2 – Pin#3 to Pin#11) marks the signals needed to make UEXT (check the schematics of OLinuXino-MAXI or OLinuXino-MINI or the UEXT section above for further help). The YELLOW section (CON2 - Pin#12 to Pin#15) marks the signals needed to build SJTAG. The BLUE section (CON2- Pin#16 to Pin#21) shows the signals needed for audio in and audio out.

GPIO connector hardware					
CON1			**CON2**		
Pin#	**Signal Name**	**Processor pin#**	**Pin#**	**Signal Name**	**Processor pin#**
1	+5VEXT	102	1	+5VEXT	102
2	GND	-	2	GND	-
3	PIN1/LCD_D00	2	3	3.3V_REG	-
4	PIN2/LCD_D01	3	4	GND	30, 35, 98, 105, 112, 118
5	PIN3/LCD_D02	4	5	AUART1_TXD	127
6	PIN4/LCD_D03	5	6	AUART1_RXD	128
7	PIN5/LCD_D04	6	7	I2C_SCL	11 or 34
8	PIN6/LCD_D05	7	8	I2C_SDA	15 or 31
9	PIN7/LCD_D06	8	9	PIN9/LCD_D08/ SSP2_MISO	22
10	PIN8/LCD_D07	9	10	SSP2_MOSI	21
11	PIN9/LCD_D08	22	11	SSP2_SCK	33
Pin#	**Signal Name**	**Processor pin#**	**Pin#**	**Signal Name**	**Processor pin#**
12	PIN10/LCD_D09	23	12	CS_UEXT_GPIO	25 or 32
13	PIN11/LCD_D10	24	13	PWM0_DUART_RXD	125
14	PIN12/LCD_D11	25	14	PWM1_DUART_TXD	126
15	PIN13/LCD_D12	27	15	SJTAG_PSW	-
16	PIN14/LCD_D13	26	16	HPL	113
17	PIN15/LCD_D14	29	17	HPR	109
18	PIN16/LCD_D15	28	18	HPVGND	111
19	PIN17/LCD_D16	19	19	LIN1_INL	115
20	PIN18/LCD_D17	20	20	LIN1_INR	114
21	PIN19/LCD_D18	17	21	PIN34/MIC	116
22	PIN20/LCD_D19	16	22	PIN33/LRADC0	108
23	PIN21/LCD_D20	15	23	PIN32/LRADC10	107
24	PIN22/LCD_D21	11	24	PIN31	82

| 25	PIN23/LCD_D22	12	25	PIN30	81
Pin#	Name	Processor pin#	Pin#	Name	Processor pin#
26	PIN24/LCD_D23	13	26	PIN29	34
27	PIN25/LCD_D23	14	27	PIN28	31
28	PIN26/LCD_D23	10	28	PIN27	91
29	GND	-	29	GND	-
30	3.3V	-	30	BAT	100; 103

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 CON pad #	Linux GPIO/iMX 233 GPIO	OLinuXino Name	OLinuXino CON pad #
0	PIN9	CON1#11	32 to 39	Not implemented	Not implemented
1	PIN10	CON1#12	51	PIN25	CON1#27
2	PIN11	CON1#13	52	PIN24	CON1#26
3	PIN12	CON1#14	53	PIN26	CON1#28
4	PIN13	CON1#15	55	PIN22	CON1#24
5	PIN14	CON1#16	56	PIN21	CON1#23
6	PIN15	CON1#17	64	JTAG_TDO1 SSP1_CMD	Not implemented
7	PIN16	CON1#18	65	LED1	Not implemented
16	PIN17	CON1#19	91	PIN30	CON2#25
17	PIN18	CON1#20	92	PIN31	CON2#24

19	TEST_PAD	Not implemented			
20	UEXT_SPI2_MOSI	CON1#11, CON2#9			
23	PIN29	CON2#26			
24	UEXT_SPI2_SCK	Not implemented			
25	PIN28	CON2#27			
30	UEXT_TX1	Not implemented			
31	UEXT_RX1	Not implemented			

Below is the schematic of GPIO_CON:
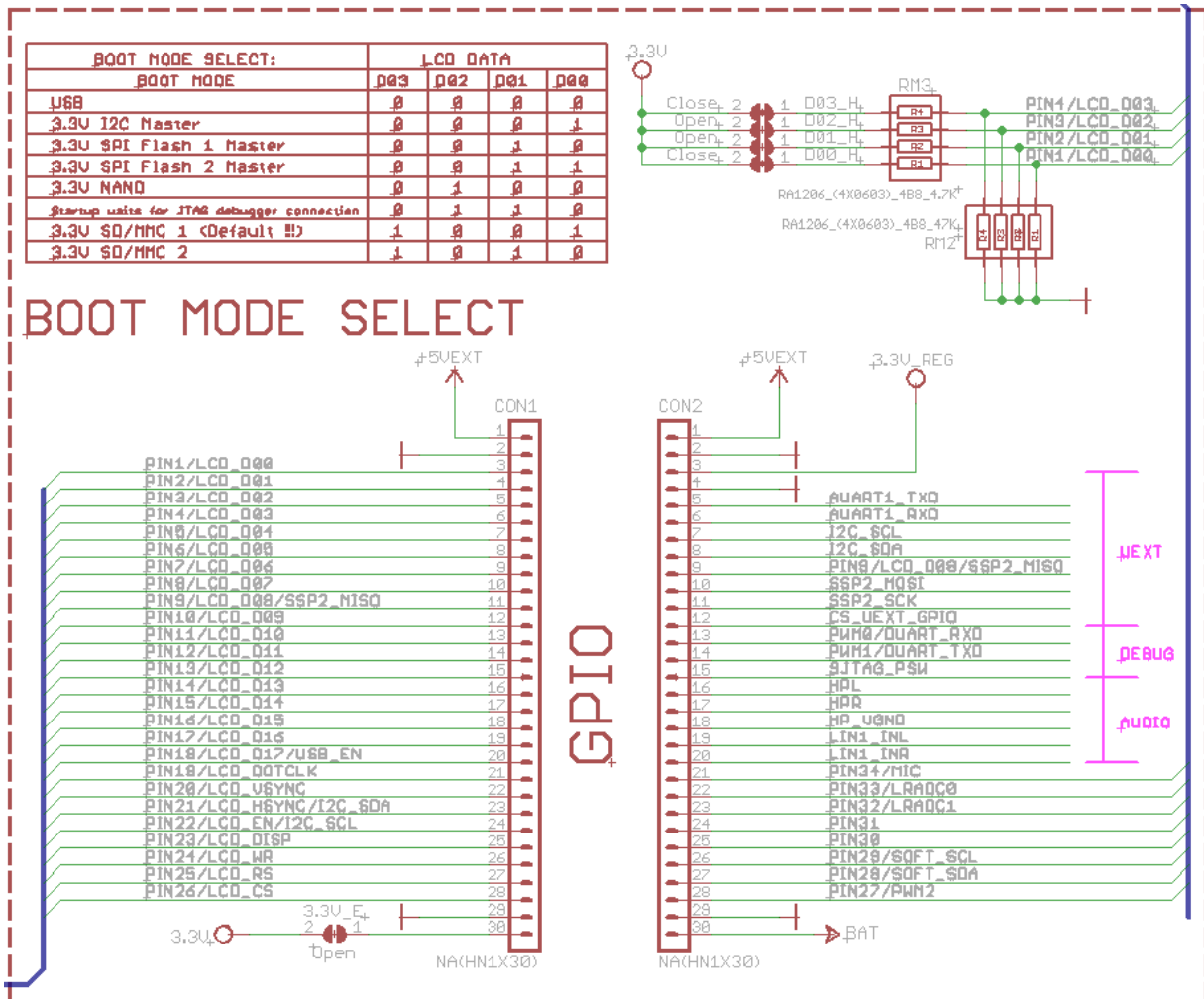
You can see the signals needed to form UEXT connector, SJTAG debug interface and audio interface marked with purple.

Thanks to David Whittaker some of the above information can be found in compact variant at, a table he compiled: https://docs.google.com/spreadsheet/ccc?key=0AspkrcYcY5bWdFB6WC1xRlp5bFRjc1hwVnlQZDVmeUE

6.5 USB HOST

The USB hosts and the Ethernet are controller by the iMX233 controller.

Important! The USB of OLinuXino-Micro can communicate directly only with High-speed or Full-speed USB devices. The majority of keyboards and mice are low-speed devices.

A good solution for plugging a keyboard or a mouste to the Micro is to use a USB hub between your device and OLinuXino-Micro - using such a hub has the additional benefit of providing additional USB slot (which can be used for example to mount flash memories etc.).

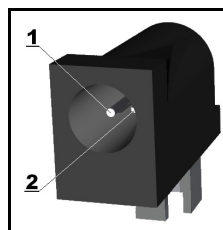The signals follow the familiar and standard USB host pattern:

USB 2-level host	
PIN#	SIGNAL NAME
1	+5V_HOST_PWR
2	USB_DM
3	USB_DP
4	GND
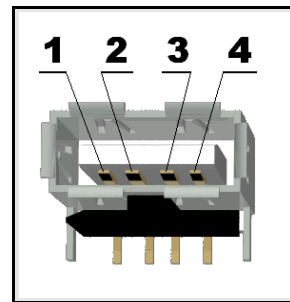
6.6 PWR Jack

The power jack used is the typical 2.5mm one used by Olimex in most of our products. You should provide 5 volts at 1A maximum 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

The audio connectors are not mounted on OLinuXino-MICRO, but the signals needed can be found on CON2 connector. Check the CON1 and CON2 section for further info.

6.8 Composite video connector

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

The 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.
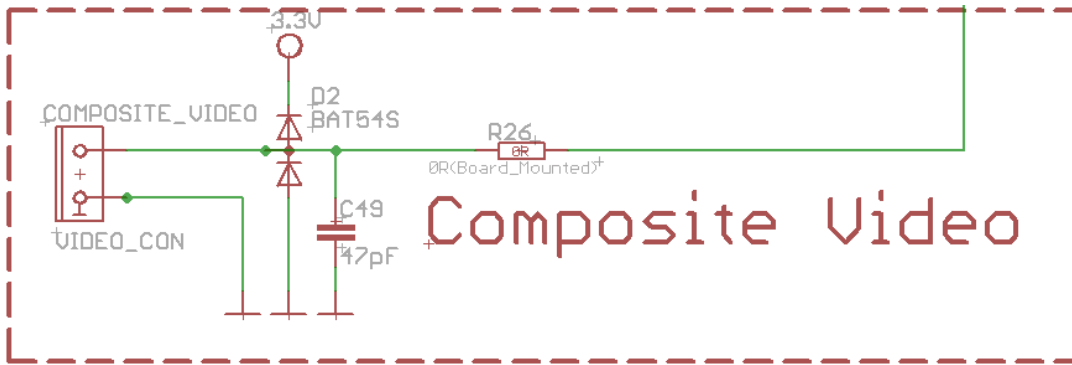
6.9 Boot mode positions

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
33.3V 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

6.10 Jumper description

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

6.10.1 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 (both SCL_SW, SDA_SW closed) or hardware I2C interface (both SCL_HW, SDA_HW closed).

The default position is SCL_HW/SDA_HW closed (soldered).

6.10.2 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.

6.10.3 3.3V_E jumper

3.3V_E jumper is open by default. If closed (soldered) provides 3.3V at the pin 30 of CON1.

The default position is open.

6.10.4 Boot mode selecting jumpers

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

6.11 Additional hardware components

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

Reset button – used for hardware reset of 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 button (REC) – when the board is put to sleep REC can be used to "wake-up" the board

512 (32M x 16) MBit DDR SDRAM - the exact memory used at the moment of writing is Xylinx HY5DUxxxyyy

LED1

6.12 Accessories

Here you will find additional information for Olimex products you can use with OlinuXino-MICRO

6.12.1 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 of female connectors. The wires follow the following color pattern GND=BLUE, RX(INPUT)=GREEN, TX(OUTPUT)=RED.

You will need a drivers that can be downloaded from the page of USB-SERIAL-CABLE: https://www.olimex.com/Products/Components/Cables/USB-Serial-Cable/USB-Serial-Cable-F/.

CHAPTER 7: BLOCK DIAGRAM AND MEMORY
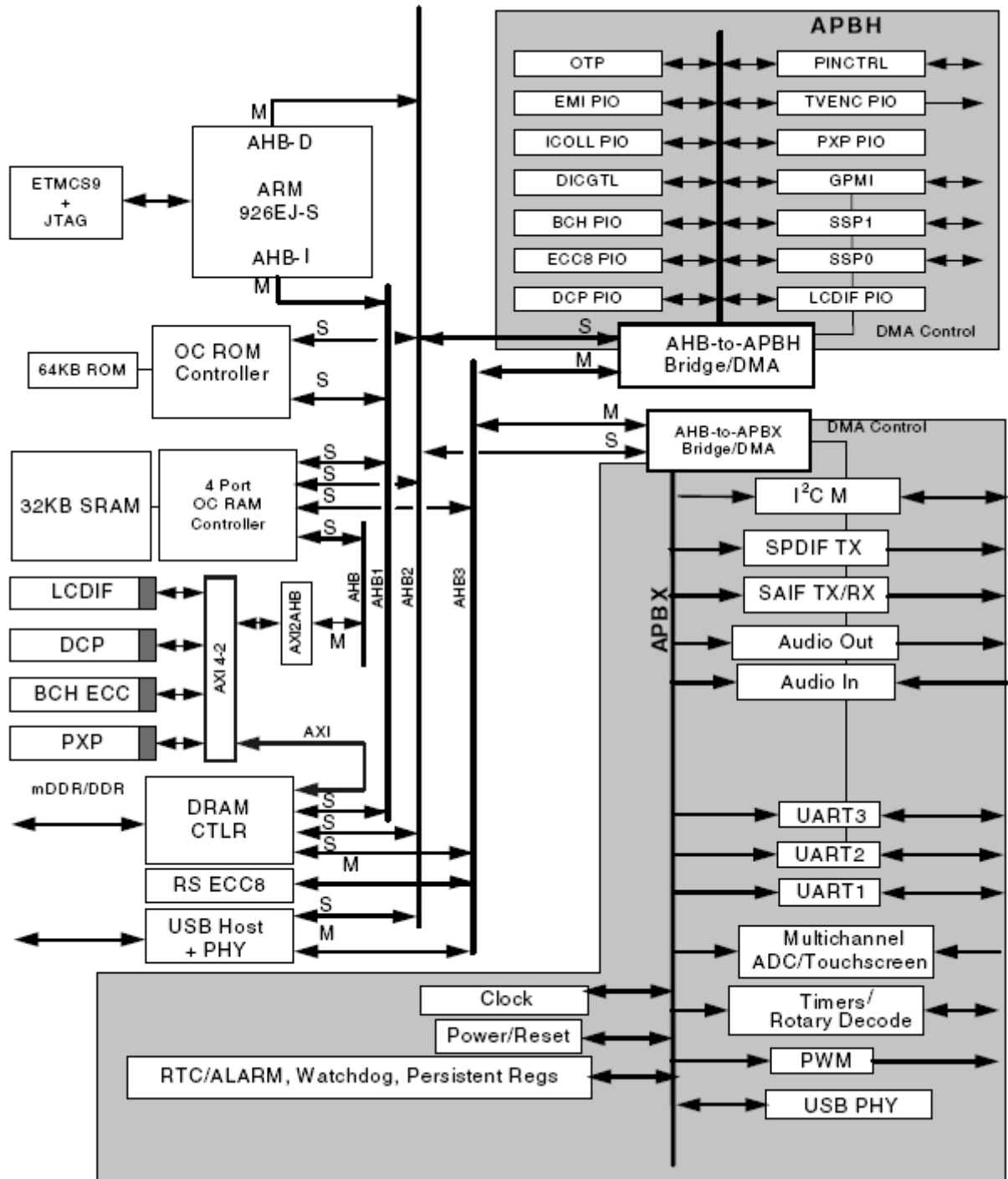
7. Introduction to the chapter

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
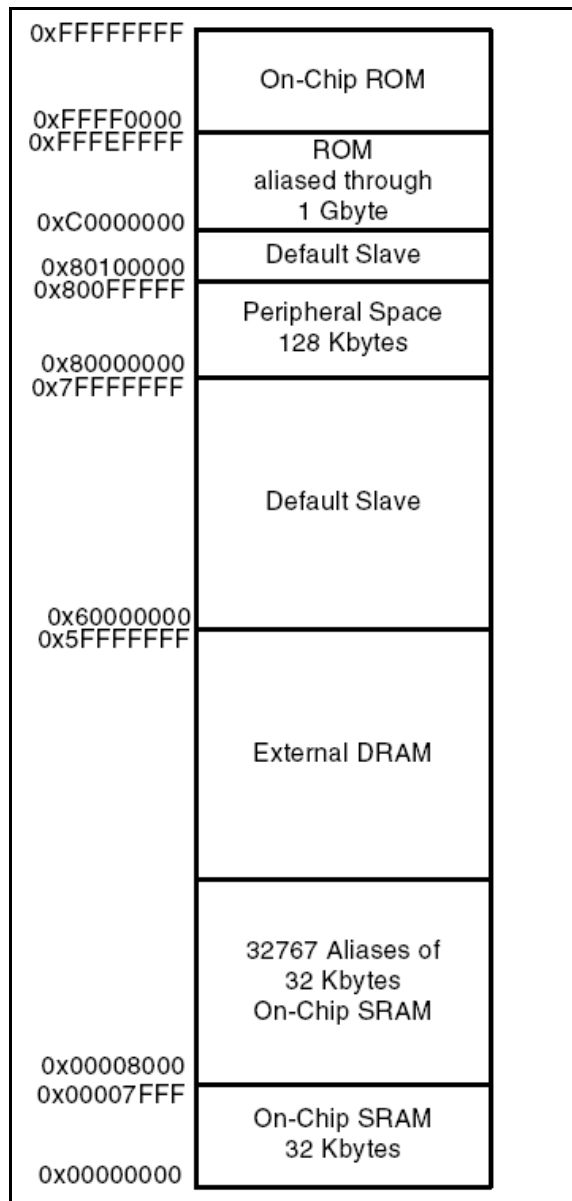
CHAPTER 8: SCHEMATICS

8. Introduction to the chapter

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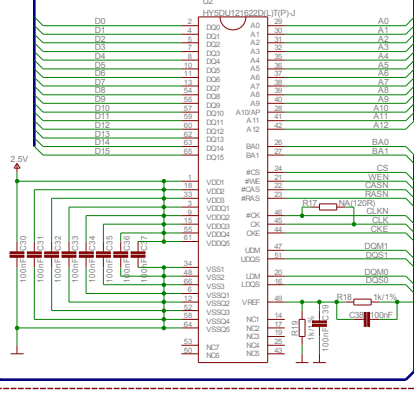
8.1 Eagle schematic

OLinuXino schematic is visible for reference here. You can also find them on GitHub page for OLinuXino at our site: https://www.olimex.com/Products/OLinuXino/iMX233/iMX233-OLinuXino-MICRO/. The link to the GitHub is located in HARDWARE section.

The EAGLE schematic is situated on the next page for quicker reference.
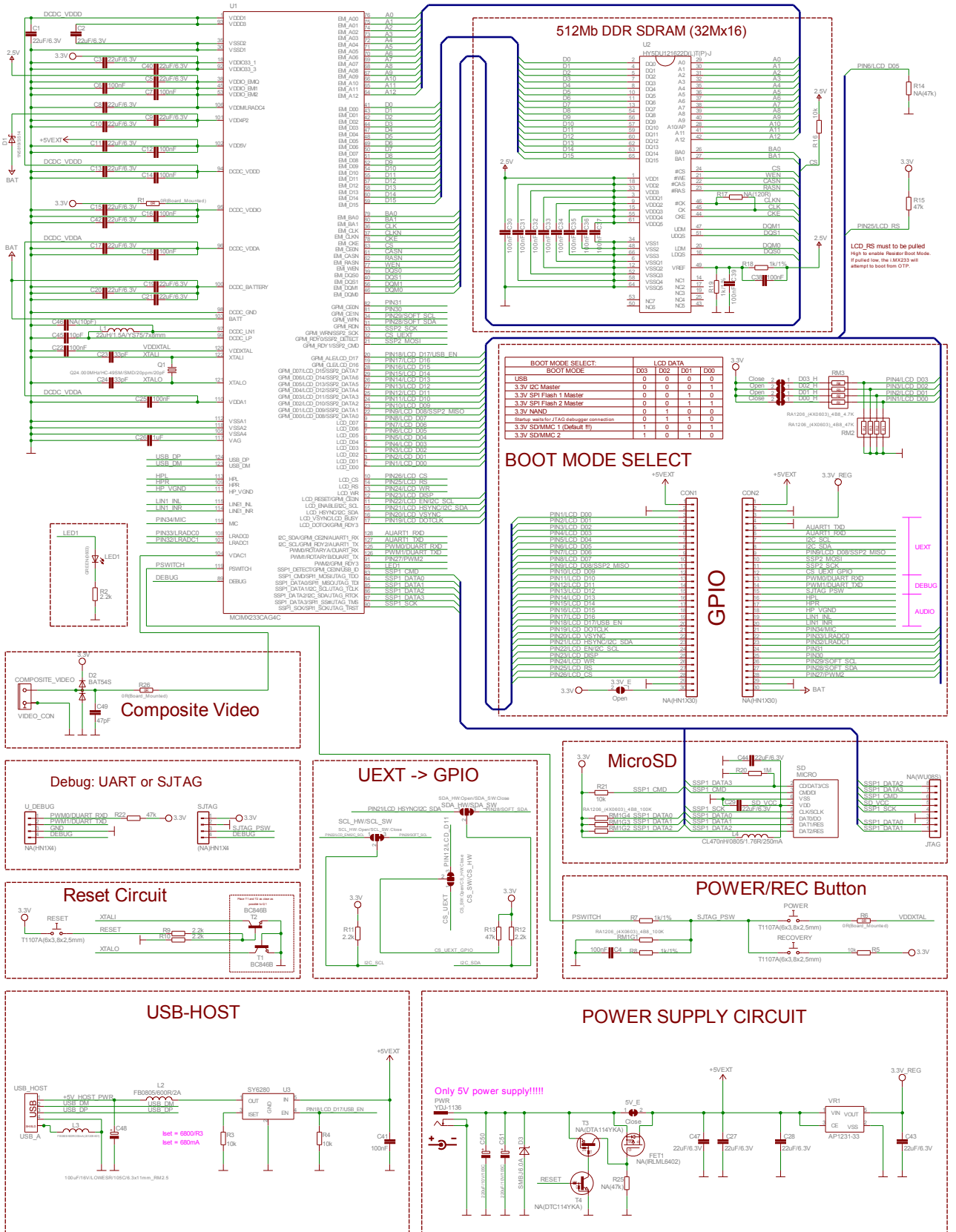
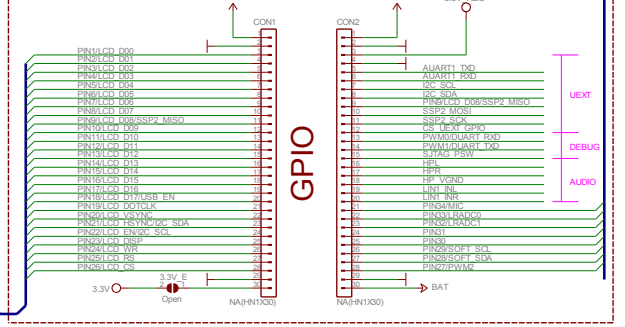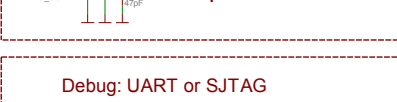512Mb DDR SDRAM (32Mx16)

BOOT MODE SELECT

BOOT MODE SELECT:	LCD DATA			
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
Startup 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

LCD_RS must be pulled
High to enable Resistor Boot Mode.
If pulled low, the i.MX233 will
attempt to boot from OTP.

GPIO

Composite Video

Debug: UART or SJTAG

UEXT -> GPIO

MicroSD

Reset Circuit
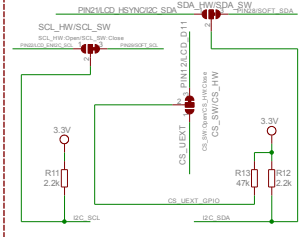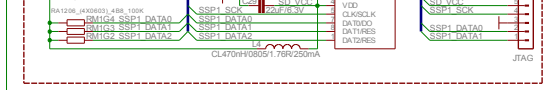
POWER/REC Button
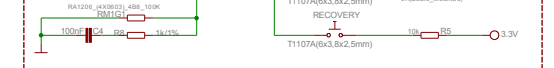
USB-HOST

POWER SUPPLY CIRCUIT
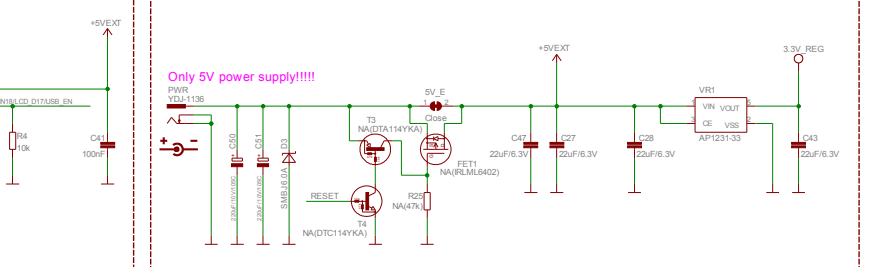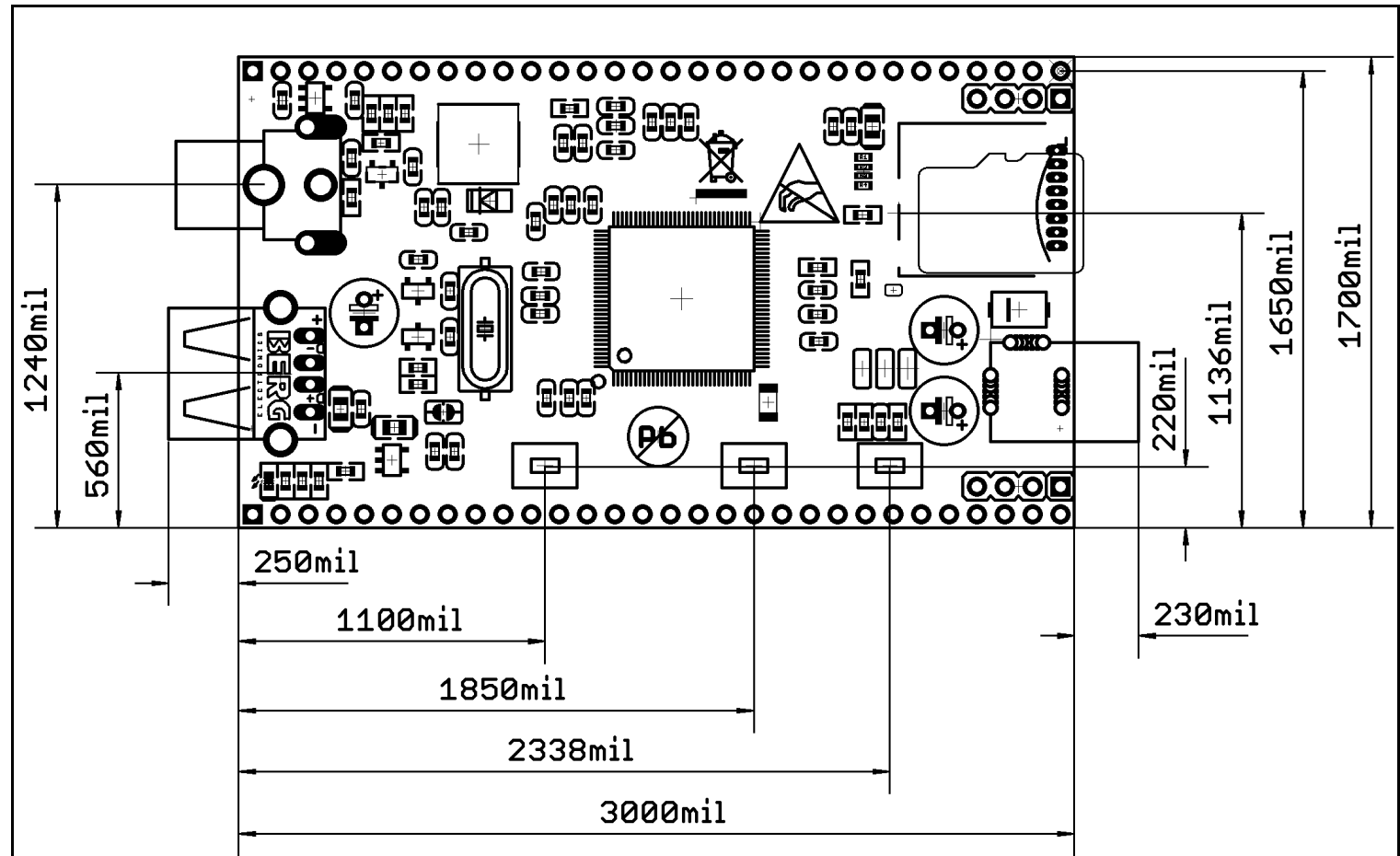
Only 5V power supply!!!!!
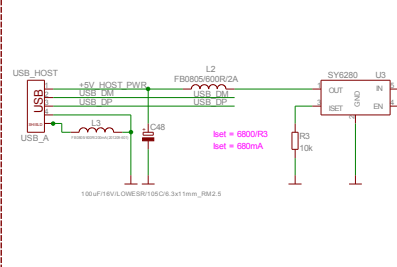
8.2 Physical dimensions

Note that all dimensions are in mills.

Additionally, the three highest elements on the board in order from the tallest to the shortest are (dimensions without the PCB thickness):

1. Composite video connector: 490 mil

2. C50/C51: 480 mil

3. C48: 470mil

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, 05.07.12	Initial Creation	All
B, 17.07.12	Added notes on USB compatibility	8, 33
C, 18.07.12	Added info about fuse programming	10, 11
D, 20.07.12	4 – Added links to all indexes 40 – Added info about USB-SERIAL-CABLE 48 – Fixed typo about the power supplies	4, 40, 48
E, 24.07.12	9 – wrong powering information fixed 22,23 – wrong power supply information fixed 37 – REC, Reset fixed	9, 22,23, 37
F, 12.09.12	3,4 – Edited the index 10 – Added info about ArchLinux 33 – Added link to a cheat sheet 42 – Updated the schematic	3, 4, 10, 33, 42
Continues on next page		

Revision	Changes	Modified Page#
	Continued from previous page	
G, 01.10.12	29 – Fixed error about GPIO connectors 46 – Added info about a random hang-up problem caused by R17 47 – Link updates	29, 46, 47
H, 04.10.12	14 – Added short redirection info about Arch-Linux setup 28 – UEXT connection important info added 32 – Fixed error in pinout table (PIN 29 from CON1 is GND)	14, 28, 32
I, 29.10.12	18, 19, 20 – Fixed wrong info about the LQFP128 package of iMX233	18, 19 , 20
J, 04.01.12	10 – Wrong picture and description of the fuse burning algorithm	10
K, 21.03.12	6 – fixed the voltages for -MINI and -MAXI in the comparison table 34 – removed left-over battery connector explanation 35 – jumpers positions adjusted based on design 36 – memories names updated Various – updated a couple of links	6, 34, 35, 36, 45

9.2 Board revision

Revision	Notable Changes
B	Initial release of the board
B1	Removed termination R17 (120R) after first batch of boards which was causing memory lock-ups
B2	Capacitors C23 and C24 changed from 33pF to 15pf due to occasional initial power hang-ups.
C	1. Changed pull-up resistor of the RX line to 4.7k and BAT54C is added on the same line. 2. SW_SCL/HW_SCL i SW_SDA/HW_SDA jumpers got changed to default hardware. 3. SD card is changed from TFR-MEM-COM to TFC-WXCII00-08-LF and the connector is moved towards the middle of the board

Important note about owners of revision B of the board: if you are one of the first owners of iMX233-OLinuXino-Micro and you experience random hang-ups (Kernel oops, Kernel panic) it is recommended to unsolder/remove R17 (check the schematic or the board file to locate it easier). Removing R17 fixes the random lock-up.

This problem has been fixed in revision B1 of the board.

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-maxi.html and https://www.olimex.com/wiki/IMX233
The forum for general questions and FAQ: https://www.olimex.com/forum/
You can get the latest updates on the software at: https://github.com/OLIMEX/OLINUXINO
(please check the readme at the bottom).
The OLinuXino Linux images sources: https://github.com/Freescale/fsl-community-bsp-platform.

ORDER CODES:

iMX233-OLinuXino-MICRO– tiny in size, tiny in price and perfect for breadbording
iMX233-OLinuXino-MINI – the mini version of the single-board computer
iMX233-OLinuXino-MAXI – the best version of OLinuXino featuring Ethernet controllert

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-MICRO 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)

How to order?
You can order directly from the web-site or from any of our distributors.

Please 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 warrany do not cover problems caused by unproper 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.
```