



# **Mars ZX3 Android manual**

**Antmicro**

**Sep 27, 2017**

<b>1</b>	<b>Introduction</b>	<b>1</b>
1.1	Acknowledgements . . . . .	1
1.2	Version information . . . . .	1
<b>2</b>	<b>Compiling the system</b>	<b>2</b>
2.1	Prerequisites . . . . .	2
2.2	Building the Linux kernel . . . . .	3
2.3	Getting the Android sources . . . . .	4
2.4	Building Android . . . . .	4
2.5	Additional boot files . . . . .	4
<b>3</b>	<b>Creating an SD Card with the system</b>	<b>5</b>
3.1	Preparing the card . . . . .	5
3.2	Copying files . . . . .	6
<b>4</b>	<b>Booting</b>	<b>7</b>
4.1	U-Boot . . . . .	7
4.2	Setting up the U-Boot environment . . . . .	7
<b>5</b>	<b>Using a USB WiFi dongle</b>	<b>9</b>
5.1	Hints on adding support for other interfaces . . . . .	9

# CHAPTER 1

---

## Introduction

---

This is a compilation and usage manual for the Android 4.3.1 operating system for **Enclustra's Mars ZX3 Zynq module**.

## Acknowledgements

We would like to thank Enclustra GmbH (<http://enclustra.com>) for co-sponsoring the port.

## Version information

Author	Content	Date	Version
Peter Gielda	Draft version	07.07.2013	0.1.0
Michael Gielda	Updated for Sphinx	08.07.2013	0.2.0
Sebastian Kramer	Typos; required packages; improvements	08.07.2013	0.2.1
Peter Gielda	Minor improvements	06.06.2014	0.2.2
Mariusz Glebocki	Updated to Android 4.3.1	03.09.2014	0.2.3
Michael Gielda	Corrections	12.09.2014	0.2.4
Michael Gielda	More info on U-Boot	19.09.2014	0.2.5
Peter Gielda	Changed list of packages and default OS to Ubuntu 12.04	26.09.2014	0.2.6
Peter Gielda	Git hashtag updated	26.09.2014	0.2.7

---

### Compiling the system

---

The port was prepared using Gentoo and Debian Linux environments. The procedures described here should also work on other systems, but if you detect any errors or omissions please e-mail us at [contact@antmicro.com](mailto:contact@antmicro.com).

### Prerequisites

#### Toolchain

The toolchain used to compile the kernel is Sourcery G++ Lite 2011.03-42, and can be obtained from [the Mentor Graphics website](#).

When this was written, Sourcery G++ Lite was directly available without the need for logging in using a direct link:

```
wget 'https://sourcery.mentor.com/sgpp/lite/arm/portal/package8736/public'\
'/arm-none-eabi/arm-2011.03-42-arm-none-eabi.bin'
chmod +x arm-2011.03-42-arm-none-eabi.bin
./arm-2011.03-42-arm-none-eabi.bin
```

**Warning:** The installer for the toolchain does not cope with dash as the system shell, however instructions how to get around that should be printed out when the installer is run.

The toolchain should be decompressed and its `bin` directory included in the `PATH` variable. The proper availability of the toolchain can be checked by finding out if `arm-none-eabi-gcc` is available from the shell.

## 2.2. Building the Linux kernel

### repo

The `repo` tool, used to manipulate the Android git repositories, can be downloaded from Google's repositories and made available in your system as follows:

```
mkdir ~/bin
PATH=~/.bin:$PATH
curl https://storage.googleapis.com/git-repo-downloads/repo > ~/bin/repo
chmod a+x ~/bin/repo
```

### Java

The Java JDK 1.6 is required for Android compilation. Oracle JDK 1.6 installer might not be available in your system's package manager. In this case, get the latest version of the Java SE Development Kit from the [Oracle website](#). In the file list accept the license and download the `.bin` file for your system (Linux x86 or Linux x86). This step requires logging in.

Instructions to install the package named `jdk-6u45-linux-x64.bin` downloaded to the `~/downloads` directory are:

```
cd ~
chmod +x ~/downloads/jdk-6u45-linux-x64.bin
~/downloads/jdk-6u45-linux-x64.bin
```

The package is unpacked to the `~/jdk1.6.0_45` directory.

### Other packages

A number of other dependencies are also required, but these should be available from package managers on most systems without a problem - e.g. on 64-bit Ubuntu 12.04 you can get them by typing:

```
sudo apt-get install git gnupg flex bison gperf build-essential \
zip curl libc6-dev libncurses5-dev:i386 x11proto-core-dev \
libx11-dev:i386 libreadline6-dev:i386 libgl1-mesa-glx:i386 \
libgl1-mesa-dev g++-multilib mingw32 tofrodos \
python-markdown libxml2-utils xsltproc zlib1g-dev:i386 \
u-boot-tools lzop
```

## Building the Linux kernel

You will need to compile your Android-enabled kernel separately. You can base your configuration on [https://github.com/antmicro/linux-xlnx/raw/enclustra/config\\_enclustra\\_android](https://github.com/antmicro/linux-xlnx/raw/enclustra/config_enclustra_android).

To compile the 3.13 kernel from our repository:

## 2.3. Getting the Android sources

```
git clone https://github.com/antmicro/linux-xlnx.git
git checkout enclustra
git checkout d66a0ec2f397a285365b339b1386f166c5285b2d .
cp config_enclustra_android .config
CROSS_COMPILE="arm-none-eabi-" LOADADDR=0x8000 ARCH=arm make uImage -j$(nproc)
```

## Getting the Android sources

The sources are fetched using the repo tool:

```
repo init -u git://github.com/antmicro/android_mars_zx3 -b master
repo sync -f # use -f to ignore fetch errors
```

**Warning:** Before starting this procedure, be aware that it may take a long time, especially if you are running on a slow Internet connection!

## Building Android

Android can now be compiled using your Java installation. Be sure to supply the correct path to the JDK.

```
source ./build/envsetup.sh
export JAVA_HOME=$HOME/jdk1.6.0_45 # path to the JDK installed before
export ANDROID_JAVA_HOME=$JAVA_HOME
export PATH=$JAVA_HOME/bin:$PATH
lunch mars_zx3-userdebug
make -j$(nproc)
```

## Additional boot files

To boot Android on the device, you will also need some additional files:

- devicetree.dtb
- system\_top.bit

To download them, use the following command:

```
git clone --depth 1 https://github.com/antmicro/boot_files_mars_zx3.git
```

---

## Creating an SD Card with the system

---

To boot Android on the device you have to use at least a 512 MB SD Card. The system needs four partitions:

- for the kernel image and related files (vfat, 32 MB)
- root partition (ext4, 32 MB)
- system (ext4, 256 MB)
- data (ext4, 100 MB or more)

The last partition, used to store user data and additional applications, will be formatted to take up the remaining space on the SD card.

### Preparing the card

**Warning:** All data on the card will be lost. `/dev/sdX` below is used as the card device node.

Insert the card into reader and create partitions with the following commands (lines beginning with a colon are typed inside the `fdisk` command prompt, without the colon):

```
sudo fdisk /dev/sdX
: o [enter]
: n [enter] [enter] [enter] [enter] +32M [enter]
: n [enter] [enter] [enter] [enter] +32M [enter]
: n [enter] [enter] [enter] [enter] +256M [enter]
: n [enter] p [enter] [enter] [enter]
: w [enter]
```

## 3.2. Copying files

```
mkfs.vfat -n BOOT /dev/sdX1
mkfs.ext4 -L root /dev/sdX2
mkfs.ext4 -L system /dev/sdX3
mkfs.ext4 -L data /dev/sdX4
```

## Copying files

**Note:** `$KERNEL`, `$ANDROID`, and `$BOOTFILES` used below are respectively: the kernel and Android sources main directories paths, and path to additional boot files (`system_top.bin` and `devicetree.dtb`)

If the compilation was successful, the rootfs CPIO image is located at `$ANDROID/out/target/product/mars_zx3/ramdisk.img`, and the system partition at `$ANDROID/out/target/product/mars_zx3/system.img`. The compiled kernel image is at `$KERNEL/arch/arm/boot/uImage`.

To install files on the card, run the following commands as root:

```
mkdir -p /mnt/android/{img,boot,root,system}
mount /dev/sdX1 /mnt/android/boot
mount /dev/sdX2 /mnt/android/root
mount /dev/sdX3 /mnt/android/system
mount -o loop $ANDROID/out/target/product/mars_zx3/system.img /mnt/android/img

cp $KERNEL/arch/arm/boot/uImage /mnt/android/boot
cp $BOOTFILES/{devicetree.dtb,system_top.bit} /mnt/android/boot

rsync -av /mnt/android/img/* /mnt/android/system
cd /mnt/android/root
gunzip -c $ANDROID/out/target/product/mars_zx3/ramdisk.img | cpio -i
chmod +x *.sh

cd /
umount /mnt/android/{img,boot,root,system}
```



### U-Boot

To boot Android on Mars ZX3 you have to use a matching U-Boot bootloader.

The sources and compilation instructions can be found in the [ZX3 GitHub repository](#)

Additional information about the bootloader itself can be found at the [U-Boot website](#).

---

**Note:** How to flash the Mars ZX3 module with a proper `boot.bin` file with the correct U-Boot is beyond the scope of this manual, but this only needs to be done once.

---

### Setting up the U-Boot environment

Connect the USB cable to the micro USB port and run a serial terminal program, for example `picocom`:

```
picocom -b 115200 /dev/ttyUSB0
```

In the U-Boot command prompt type the following commands to set environment variables:

```
setenv bootargs console=ttyPS0,115200 root=/dev/mmcblk0p2 rw rootwait
↳earlyprintk
setenv bootcmd mmcinfo && fatload mmc 0 0x3000000 uImage && \
fatload mmc 0 0x2A00000 devicetree.dtb && \
fatload mmc 0 0x200000 system_top.bit && fpga loadb 0 0x200000 ${filesize} &&
↳\
```

## 4.2. Setting up the U-Boot environment

---

```
bootm 0x3000000 - 0x2A00000  
saveenv
```

And to boot:

```
boot
```

---

### Using a USB WiFi dongle

---

By default, only WiFi interfaces based on Atheros AR9271 are supported. Simply connect the dongle to the USB port and go to the Android settings, where you can turn on WiFi.

#### Hints on adding support for other interfaces

To use other interfaces, you have to turn on the required interface's driver in the kernel config, rebuild it, and optionally put its firmware in the `etc/firmware` directory on the system partition. For detailed information which driver and firmware to use, google for its name or ID, which can be obtained with the `lsusb` command. The [Linux Wireless](#) page is a good place to start.