

# ARMSDK-VM Virtual Appliance

## A preconfigured Linux system

## Disclaimer

The information contained within this specification, including but not limited to any product specification, is subject to change without notice.

Ka-Ro electronics provides no warranty with regard to this specification or any other information contained herein and hereby expressly disclaims any implied warranties of merchantability or fitness for any particular purpose with regard to any of the foregoing.

## Copyright Notice

Copyright © 2009-2011, Ka-Ro electronics GmbH. All rights reserved. All text, pictures and graphics are protected by copyrights. No copying is permitted without written permission.

We have made every attempt to ensure that the information in this document is accurate yet the information contained within is supplied "as-is".

## Trademarks

ARM is a registered trademark of ARM Limited.

VMware is a registered trademark of VMware, Inc. in the United States.

VirtualBox is a registered trademark of Oracle Corporation.

Microsoft, Windows, Windows NT and VirtualPC are registered trademarks of Microsoft Corporation.

Linux is a registered trademark of Linus Torvalds.

All other brand and product names, trademarks, and copyrights are the property of their respective owners.

## Table of contents

1 System-Setup.....	4
1.1 Ready to use virtual appliance.....	4
1.2 System requirements.....	4
1.3 VMware Player.....	4
1.4 KARO-ARM virtual appliance.....	5
1.4.1 Login.....	5
1.4.2 Mount the CDROM.....	5
1.4.3 Change the keyboard layout.....	5
2 Setting up the tool-chain.....	6
2.1 Getting the source.....	6
2.2 Preparing.....	6
2.3 Building.....	7
3 Linux kernel.....	8
3.1 Getting the source.....	8
3.2 Preparing.....	8
3.3 Building .....	8
4 Root-File-System.....	9
4.1 Getting the source.....	9
4.2 Preparing.....	9
4.3 Building.....	9
5 Redboot .....	10
5.1 Getting the source.....	10
5.2 Preparing.....	10
5.3 Building .....	10
6 Prerequisites on other systems.....	11
6.1 List of required Debian 5 (Lenny) utilities.....	11
7 Document revision history.....	12

## 1 System-Setup

### 1.1 Ready to use virtual appliance

A **virtual appliance** is a [virtual machine](#) image designed to run on a virtualisation platform (e.g., [VMware Workstation](#), [Xen](#), [VirtualBox](#)). Ka-Ro provides a complete preconfigured Debian 6 (Squeeze) VMware virtual machine, called *ARMSDK-VM*, which can be used to customise RedBoot, U-Boot, Linux kernel and root-file-system.

### 1.2 System requirements

The minimum host system requirements for installing and using the *ARMSDK-VM* virtual appliance are:

- Standard x86-compatible or x86-64-compatible PC
- Processor speed 733MHz or faster
- Memory 512 MB minimum (2GB recommended). You must have enough memory to run the host operating system, the virtual machine, and applications on the host and guest operating systems.
- Hard disk - At least 10GB free disk space is recommended
- VMware Player version 3.1.4 or later

### 1.3 VMware Player

To be able to use the *ARMSDK-VM* virtual appliance a VMware Player has to be installed and properly configured on the host machine. VMware Player is the easiest way to run multiple operating systems at the same time on your PC. With its user-friendly interface, VMware Player makes it effortless for anyone to try out Windows 7 or the latest Linux releases, or create isolated virtual machines to safely test new software and surf the Web.

The *ARMSDK-VM* virtual appliance has been tested with VMware Player version 3.1.4. Please download and install this from here (*account needed*):

[http://downloads.vmware.com/d/info/desktop\\_downloads/vmware\\_player/3\\_0](http://downloads.vmware.com/d/info/desktop_downloads/vmware_player/3_0)

Refer to the VMware Player documentation for more details:

[http://www.vmware.com/support/pubs/player\\_pubs.html](http://www.vmware.com/support/pubs/player_pubs.html)

## 1.4 KARO-ARM virtual appliance

The *ARMSDK-VM* virtual appliance can be downloaded from the Ka-Ro website:

<http://www.karo-electronics.com/download.htm>

To execute the VM copy first the delivered Version on your local hard disk C:\. Start the installed VM Player by double click on the desktop icon and click the "open" button on the Virtual Machine window. Browse into the folder of the VM, place the mouse cursor over in the *armsdk.vmx* and right click the mouse button to pass control to VMware Player. After the Linux kernel is booted the VM windows shows the Debian desktop environment. Enter the login name and user password as root.

The VM is not designed to build a particular version of the tool-chain or rootfs and therefore it comes without any pre-installed source code.

### 1.4.1 Login

Three user accounts are configured on the *ARMSDK-VM* virtual appliance:

```
user: root      password: user
user: user      password: user
user: armsdk    password: armsdk
```

Always login as *armsdk* for doing all the tasks described in this document.

### 1.4.2 Mount the CDROM

Login as *armsdk* and put the CD into the cdrom drive.

Mount the cdrom device:

```
sudo mount -t iso9660 /dev/hda /cdrom
```

Verify mounted device:

```
ls /cdrom
```

### 1.4.3 Change the keyboard layout

The default keyboard mapping is setup for a german keyboard. Use this command to change the layout:

```
sudo dpkg-reconfigure keyboard-configuration
```

## 2 Setting up the tool-chain

### 2.1 Getting the source

The gcc and tool-chain sources can be found on the Starterkit CD in '/Linux/tool-chain'

**Note:** Please refer to the README file for additional information. Also check for updates on the Ka-Ro homepage:

<http://www.karo-electronics.com/download.htm>

### 2.2 Preparing

Login as armsdk

Create the folder /user/local/arm:

```
sudo mkdir -p /usr/local/arm
```

Change the permissions to be able to use it with your armsdk user account:

```
sudo chown armsdk /usr/local/arm
```

Create the tool-chain and arm-linux directories

```
mkdir -p ~/starterkit/tool-chain
```

```
mkdir -p ~/starterkit/arm-linux
```

Extract the archives

```
cd ~/starterkit/tool-chain
```

```
tar -xjf /cdrom/Linux/tool-chain/tool-chain-src.tar.bz2
```

```
tar -C ../arm-linux -xjf /cdrom/Linux/arm-linux_src.tar.bz2
```

## 2.3 Building

Use the build-tool-chain script to build the ARM EABI tool-chain for compiling the Linux kernel and applications:

```
bash build-tool-chain -m tx25 -pgi
```

Use the build-tool-chain script to build a OABI tool-chain for compiling RedBoot

```
bash build-tool-chain -m tx25 -op
```

Call

```
bash build-tool-chain -h
```

for further information and available options.

Modify the PATH settings:

for TX25/TX27/TX28:

```
PATH="$PATH:/usr/local/arm/cross-gcc-4.4.1-armv5te-soft/`uname -m`-pc-linux-gnu/bin"
```

```
PATH="$PATH:/usr/local/arm/cross-gcc-4.4.1-armv5te/`uname -m`-pc-linux-gnu/bin"
```

for TX51/TX53:

```
PATH="$PATH:/usr/local/arm/cross-gcc-4.4.1-armv7a-soft/`uname -m`-pc-linux-gnu/bin"
```

```
PATH="$PATH:/usr/local/arm/cross-gcc-4.4.1-armv7a/`uname -m`-pc-linux-gnu/bin"
```

## 3 Linux kernel

### 3.1 Getting the source

The Linux kernel sources can be found on the Starterkit CD in /Linux/TX25/ or /Linux/TX27/, etc. depending on the module type you are using.

**Note:** Please refer to the README file for additional information. Also check for updates on the Ka-Ro homepage:

<http://www.karo-electronics.com/download.htm>

### 3.2 Preparing

Login as armsdk

Create the build directory:

```
mkdir -p ~/starterkit/arm-linux  
cd ~/starterkit/arm-linux
```

Extract the archive:

```
tar -xjf /cdrom/Linux/arm-linux_src.tar.bz2
```

### 3.3 Building

Refer to

```
~/starterkit/arm-linux/Documentation/arm/README
```

and

```
~/starterkit/arm-linux/Documentation/kbuild/makefiles.txt
```

for details.



## 4 Root-File-System

### 4.1 Getting the source

The root-file-system source code can be found on the Starterkit CD in '/Linux/rootfs'.

**Note:** Please refer to the README file for additional information. Also check for updates on the Ka-Ro homepage:

<http://www.karo-electronics.com/download.htm>

### 4.2 Preparing

Login as armsdk

Create the build directory:

```
mkdir -p ~/starterkit/rootfs-build  
cd ~/starterkit/rootfs-build
```

Extract the archive:

```
tar -xjf /cdrom/Linux/rootfs/rootfs_src.tar.bz2
```

### 4.3 Building

To compile the rootfs issue the command:

```
bash build_all.sh
```

This will create a tar archive file that contains the rootfs binaries.

Refer to the README file in the rootfs source archive for more information about the build process.

To create a bootable root file system or an image file for flash programming, this archive has to be extracted as user 'root':

```
mkdir -p /tftpboot/rootfs  
tar -C /tftpboot/rootfs -xzf build-arm-926ejs-linux-gnueabi-soft/rootfs-arm-926ejs-linux-gnueabi-soft.tgz  
cd ../arm-linux/  
make modules_install INSTALL_MOD_PATH=/tftpboot/rootfs
```

The contents of '/tftpboot/rootfs' can now be used as mount point for an NFS-mounted rootfs or as source for creating a flash image with the 'mkfs.jffs2' utility:

```
mkfs.jffs2 -n -e 0x20000 -p 0x20000 -d /tftpboot/rootfs -o rootfs.image  
mkfs.jffs2 can be installed with the command:  
sudo aptitude install mtd-utils
```

## 5 Redboot

### 5.1 Getting the source

The RedBoot sources can be found on the Starterkit CD in ' /RedBoot '

**Note:** Please refer to the `README` file for additional information. Also check for updates on the Ka-Ro homepage:

<http://www.karo-electronics.com/download.htm>

### 5.2 Preparing

Login as armsdk

Create the build directory:

```
mkdir -p ~/starterkit/ecos
cd ~/starterkit/ecos
```

Extract the archive:

```
tar -xjf /cdrom/RedBoot/ecos_src.tar.bz2
```

### 5.3 Building

The RedBoot source code contains support for TX25 as well as TX27 with either 64MiB or 128 MiB SDRAM. The target for which to build is selected via configuration files (\*.ecc) in the 'config' subdirectory of the RedBoot source tree.

Run the build script 'build.sh' giving the name of the appropriate configuration file from the 'config' directory as parameter. E.g. for TX25-4021:

```
bash build.sh TX25-40x1
```

The binary file is located in the folder 'build/current/install/bin/redboot.bin'

## 6 Prerequisites on other systems

Ka-Ro provides a complete preconfigured Debian 6 (Squeeze) VMware virtual machine, called *ARMSDK-VM*. To setup your own system the following steps can be used as a guideline. The version numbers indicate the configuration that was used to build the rootfs for the starterkit CD, which does not necessarily mean, that other versions of those tools cannot be used.

If needed, update the Debian 6 (Squeeze) GNU/Linux distribution first:

```
aptitude update
```

Use aptitude to install all the packages listed below, e.g.:

```
aptitude install gcc bison ...
```

## 7 Document revision history

Revision	Changes
2010-06-08	New layout
2011-02-22	New version
2011-09-09	Updated to release change (Lenny → Squeeze), changes in layout