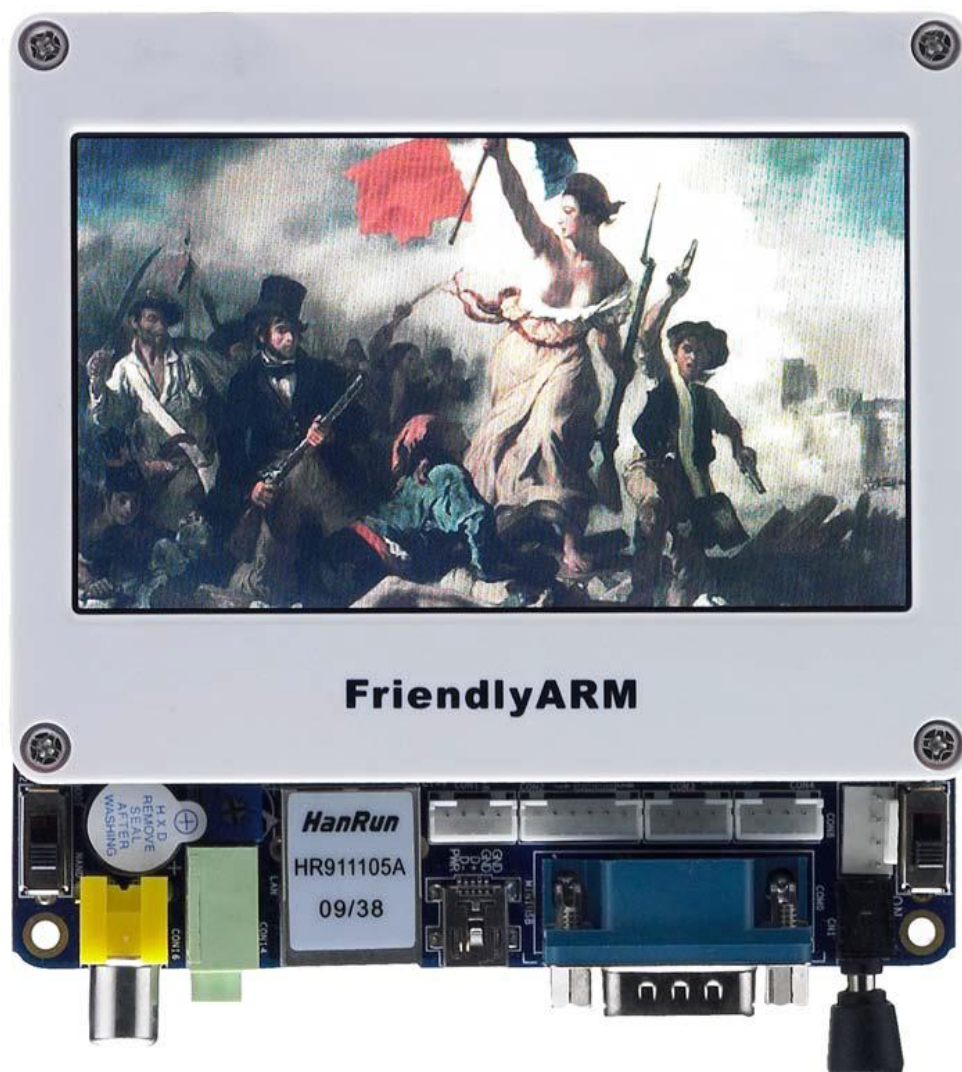




User's Guide to Mini6410 Ubuntu



REVISION	ORIGINATOR	SCR	REV DATE
0.1.0	FriendlyARM Co., Ltd		March 28th, 2011
FriendlyARM Co., Ltd Confidential: This document and information contained in it shall not be reproduced by, used by, or disclosed to others except as expressly authorized in writing by FriendlyARM Co., Ltd.			FriendlyARM Co., Ltd Guangzhou, China

copyright@2010



COPYRIGHT STATEMENT

The content (content being images, text, programs and scripts) of this English manual is copyright © FriendlyARM Co., Ltd. All rights expressly reserved.

Any content of the manual printed or downloaded may not be sold, licensed, transferred, copied or reproduced in whole or in part in any manner or in or on any media to any person without the prior written consent of FriendlyARM Co., Ltd including but not limited to:

- transmission by any method
- storage in any medium, system or program
- display in any form
- performance
- hire, lease, rental or loan

Requests for permission to reproduce material from this manual should be addressed to FriendlyARM Co., Ltd.



Index

1 Install and Run Ubuntu.....	5
1.1 Burn Ubuntu to NAND Flash	5
1.2 Run Ubuntu from SD Card.....	5
2 Set up Ubuntu Development Environment	8
2.1 Ubuntu Development and Compiler	8
2.2 Uncompress and Install Kernel	8
3 Install and Run File System	11
3.1 Make UBIFS Image	11
3.2 Make EXT3 Image	11



The Mini6410 system supports Ubuntu-0910 which is based on ARMV6 command sets. All Ubuntu versions released after 2010 are based on ARMV7 command sets and therefore cannot be run on this system. Usually an Ubuntu system is too large (more than 600M) for the NAND flash and therefore has to be run from the SD card.

A common way is to format the SD card to ext2/3 and then copy system files to it and run. This process is not easy for beginners. What we do is to make a ubuntu system a ext3 image and copy it to a FAT32/FAT SD card.

Since Ubuntu is for desktop systems and wasn't designed for embedded systems we mainly focus on how to run it on the Mini6410 system in this section and will not go any deeper

Since Superboot-20100917, Superboot has supported burning Ubuntu to the 1GB Nand Flash via the SD card. Running Ubuntu from the SD card for 256M (DDR RAM) systems is OK but a little bit slow for 128M systems



1 Install and Run Ubuntu

1.1 Burn Ubuntu to NAND Flash

Please refer to <<User's Guide to Mini6410 System Installation>>

1.2 Run Ubuntu from SD Card

With our dedicated tools users can easily run Ubuntu from the SD card. Please follow the steps below:

Step1: burn a Superboot to the SD card with the SD-Flasher.exe utility

Step2: copy the whole image directory in the shipped CD to the SD card

Step3: open “\images\FriendlyARM.ini” in the SD card, make these changes: “Action=Run” and “OS=Ubuntu”

Step4: toggle the S2 switch to “SDBOOT” insert your SD card, power on and you can play with Ubuntu.

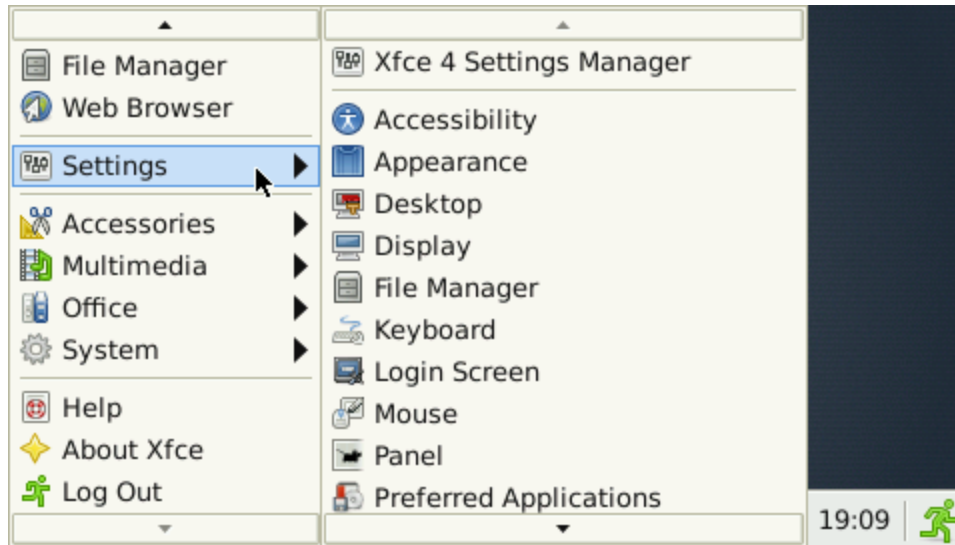
On your first system boot you will be prompted to calibrate your screen. Please follow the “+” to calibrate and Ubuntu will be loaded as follows



Our Ubuntu uses XFCE rather than GNOME which is too big for this system. The whole system doesn't run very elegantly. In addition you can notice that it can detect your SD card and list it on the desktop



Below is a screenshot of its settings menu



Note: running Ubuntu from the SD card only needs two files:

- uzImage: the kernel image
- rootfs_ubuntu.ext3: the EXT3 image

If you want to drive your serial port control touch screen you need a rootfs_ubuntu-s.ext3 image



2 Set up Ubuntu Development Environment

Note: here the “development environment” refers to the environment needed to compile the kernel and the bootloader.

2.1 Ubuntu Development and Compiler

The development environment for Ubuntu is very similar to that for Linux. The main component is the Fedora9 platform, a cross compiler and mktools tools. Please refer to section 4.3

2.2 Uncompress and Install Kernel

Create a working directory “/opt/FriendlyARM/mini6410/ubuntu”

Run the command below

```
#mkdir -p /opt/FriendlyARM/mini6410/ubuntu
```

All the files that will be uncompressed in later steps will be put into this directory

(1) Get a copy of complete **Ubuntu** source code

Create a temporary directory **/tmp/ubuntu** in Fedora9

```
#mkdir /tmp/ubuntu
```

Copy the whole “Ubuntu” directory in the shipped CD to **/tmp/ubuntu**



(2) Uncompress **u-boot** source code

Run the commands below in the “/opt/FriendlyARM/mini6410/ubuntu” directory

```
#cd /opt/FriendlyARM/mini6410/ubuntu
```

```
#tar xvzf /tmp/ubuntu/ubuntu-kernel-2.6.28-20100622.tar.gz
```

This will create a “u-boot-mini6410” directory which contains complete source code

Note: “20100622” is the date when we released it

(3) Uncompress Kernel Source Code

Execute the commands below in the “/opt/FriendlyARM/mini6410/ubuntu” directory

```
#cd /opt/FriendlyARM/mini6410/ubuntu
```

```
#tar xvzf /tmp/ubuntu/ ubuntu-kernel-2.6.28-20100622.tar.gz
```

This will create a “linux-2.6.28.6-ubuntu” directory which contains complete source code

Note: “20100622” is the date when we released it

(4) Uncompress Ubuntu File System Source Code for ARM

Execute the commands below in “/opt/FriendlyARM/mini6410/ubuntu”

```
#cd /opt/FriendlyARM/mini6410/ubuntu
```

```
#tar xvzf /tmp/ubuntu/rootfs-ubuntu-0910-20100917.tar.gz
```

```
#tar xvzf /tmp/ubuntu/rootfs-ubuntu-s-0910-20100917.tar.gz
```

This will create a “rootfs-ubuntu-0910” directory and a “rootfs-ubuntu-0910-s” directory

Note: 20100917 is the date when we released it. This source code contains a copy of Ubuntu source code and compiling scripts. “rootfs-ubuntu-0910-s” is for serial port control touch



Complete ARM Solutions
Design, Development and Manufacturing
Expertise on Embedded Linux, Android, WindowsCE

screens

3 Install and Run File System

3.1 Make UBIFS Image

With the **mkubimage-slc** utility, you can make a UBIFS image. The Ubuntu kernel by default

supports this file system:

```
#cd /opt/FriendlyARM/mini6410/ubuntu1
```

```
#mkubimage-slc rootfs-ubuntu-0910 rootfs-ubuntu-0910.ubi
```

This will generate a rootfs-ubuntu-0910.ubi file in the current directory.

Note: burning a UBIFS image is faster than burning a YAFFS2 image since a UBIFS image has smaller size. If you want to drive your serial port control touch screen you need a **rootfs_ubuntu-0910-s** image

3.2 Make EXT3 Image

With the **mkext3image** utility, you can make an EXT3 image. You can copy it to the SD card and run it directly. The Android kernel by default supports this file system. The default FriendlyARM.ini supports this file system too:

```
#cd /opt/FriendlyARM/mini6410/ubuntu
```

```
#mkext3image rootfs-ubuntu-0910 rootfs-ubuntu-0910.ext3
```



This will generate a rootfs_ubuntu-0910.ext3 file. You can copy it to your SD card's "images/Ubuntu/" directory. Also you need to make sure to define "Ubuntu-RootFs-RunImage =" to this file in the FriendlyARM.ini file.

Note: the size of an EXT3 file image usually is 30% bigger than that of other images. For a file system that is less than 64M it will be treated as a 64M system. That is the minimum size of an ext3 image is $64M \times 1.3 = 83.2M$.

Note: If you want to drive your serial port control touch screen you need a **rootfs_ubuntu-0910-s** image