TagMaster

North America, Inc.

1 Introduction 4

2 General Information 5
2.1 Virtual Machine 5
2.2 Debian GNU/Linux 5
2.3 GNU Toolchain 5

3 Software Environment Setup 7
3.1 System Requirements 7
3.2 Software Installation 7
3.3 Configuration 8
3.4 File System Integration with Host System 8
3.5 Upgrading to the Latest SDK Version 9
3.6 Backup 10

4 Building Reader Application Software 11
4.1 Target Toolchain 11
4.2 Software Library: taglib 11
4.3 Application Template and Examples 11
4.4 Editors 12
4.5 Installing Application Software on Reader 12

5 Reader Network Boot 14
5.1 File System 14
5.2 Configuring Reader Boot Settings 14

6 Frequently Asked Questions 18

7 Contact 19
7.1 Technical Support 19
7.2 Office 19

8 Glossary 20

9 References 21

10 Online Resources 22

1 Introduction

This document describes the Software Development Kit (SDK) used for developing custom Reader software applications for TagMaster RFID Readers based on the GEN4 platform. That is, LR-series and HD-series Readers, for instance LR-6 and 156640.

This document does not describe how to actually design Reader application software and write the source code; see the taglib Software Library Specification [1] for more information.

The target group for this document is software designers.

It is required to have general knowledge about the TagMaster RFID system to fully comprehend the information in this document. It is recommended to read the GEN4 Reader User's Manual [2] before reading this manual.

General computer knowledge is required and basic networking and Linux knowledge is preferable.

2 General Information

The SDK is a complete software development environment including a virtual machine that runs the Debian GNU/Linux operating systems. The SDK is prepared with compilers, tools, and services for developing Reader application software.

A complete development environment will increase the productivity and lower the start-up time. All tools are already installed and configured and a reliable working method is in place.

Developing software for GEN4 Readers is basically done the same way as if it was developed for a standard Linux host. An application can therefore be developed and debugged on a host before actually being downloaded to the target (in this case a Reader). Debugging on a host is easier than debugging on a target.

2.1 Virtual Machine

The development environment from TagMaster is basically a pre-configured virtual machine with the necessary tools for software development.

Virtualisation is an abstraction layer that decouples the physical hardware from the operating system. It allows multiple virtual machines, with heterogeneous operating systems, to run in isolation side-by-side on the same physical machine. Each virtual machine has its own set of virtual hardware upon which an operating system and applications are loaded. The operating system sees a consistent, normalized set of hardware regardless of the actual physical hardware components.

Virtual machines are encapsulated into files, making it possible to rapidly save, copy and provision a virtual machine. Full systems (fully configured applications, operating systems, BIOS and virtual hardware) can be moved, from one physical machine to another in seconds.

2.2 Debian GNU/Linux

An operating system consists of various fundamental programs which are needed by a computer so that it can communicate and receive instructions from users; read and write data to hard disks; control the use of memory; and run other software.

Debian is a free operating system, which is maintained by an all-volunteer organization dedicated to developing free software and promoting the ideals of the Free Software Foundation.

The most important part of an operating system is the kernel. Linux is the kernel component in a Debian system. The rest of the system consists of other programs, many of which were written by or for the GNU Project. The preferred term for the operating system is Debian GNU/Linux because the Linux kernel alone does not form a working operating system.

2.3 GNU Toolchain

A toolchain is a set of software tools that are used to create software. The tools may be used in a chain, so that the output of each tool becomes the input of the next. The term toolchain is used widely to refer to any set of linked development tools.

A simple software development toolchain consists of a text editor for entering source code, a compiler and linker to transform the source code into executable application software, and libraries to provide interfaces to the operating system.

The GNU toolchain is a term given to the programming tools produced by the GNU project.

The toolchain provided by this SDK consists of:

- GNU make: Used to build applications and automate compilation
- GNU Compiler Collection (GCC): Compilers for several languages
- GNU Binutils: Linker, assembler, and other tools
- GNU Debugger (GDB): An interactive debugger
- GNU C Library: A standard C library

3 Software Environment Setup

This section describes system requirements and how to install and configure the SDK.

3.1 System Requirements

The virtualisation technology provided by VMware is the base of the development environment. The free VMware Player is used to run the virtual machine. It can be installed on Windows and on some Linux distributions.

The host system requirements are:

- 512 MB RAM minimum, 1024 MB or more is recommended
- 3 GB of free disk space minimum, 5 GB or more is recommended
- A network interface is required by some services described in this manual (for instance section 5 Reader Network Boot)

3.2 Software Installation

The installation process is very straightforward: install the virtualisation software and copy the virtual machine files.

3.2.1 Virtualisation Software

Download the free VMware Player from <http://www.vmware.com/download/player>, which is roughly 40 MB, and follow the instructions during the installation process.

For more information about the virtualisation software please see <http://www.vmware.com>.

3.2.2 TagMaster Software Development Environment

The software development environment is a virtual machine to be run by the VMware Player.

The development environment is large, contact TagMaster for more information on how to obtain a copy.

Copy the following files to a suitable working directory:

- vmx-file: Virtual machine configuration file
- vmdk-file: Virtual disk file, which is used as the hard-drive for the virtual machine and will grow as data is saved

3.2.3 Starting the Development Environment

Starting the development environment is a matter of starting the virtual machine. Start the VMware player and open the vmx-file.

The virtual machine will boot Debian GNU/Linux as any stand-alone computer. Click on the virtual machine window to redirect mouse and keyboard actions to it.

Note: Press Ctrl and Alt keys to return mouse and keyboard control to the host system.

3.3 Configuration

This section covers the settings that typically have to be configured. For an extensive documentation, please refer to official Debian GNU/Linux documentation.

3.3.1 Users and Passwords

The default system administrator user account is “root” with password “qwerty”. Only use the administrator account when administrating the Debian system, for instance when installing additional software or configuring system settings.

Note: It is strongly recommended not to use the administrator account for work, for instance developing Reader software.

The default developer user account is “dev” with password “dev”.

Passwords can be changed with the command “passwd” or clicking “Desktop” → “Preferences” → “About Me” → “Change Password”.

3.3.2 IP Address

The virtual machine should be thought of as a server residing on the local network. Using a LAN bridge it accesses the local network of its host system.

By default, this Debian uses DHCP to retrieve network settings. If no DHCP server is available it is possible to use static network settings.

Static settings are also convenient when the IP address must be known, for instance services such as Samba, NFS and tftp.

Configure the network settings as follows:

1. Click menu “Desktop” → “Administration” → “Networking”
2. Enter administrator password when prompted
3. Select Ethernet interface to configure and click “Properties”
4. Enter new settings

The current IP address configuration can also be set and retrieved with the command “ifconfig”. Note that it is necessary to be logged in as administrator to use this command.

3.3.3 Keyboard Layout

The default keyboard layout is U.S. English.

To change the keyboard layout, do the following:

1. Click menu “Desktop” → “Preferences” → “Keyboard”
2. Select “Layout”
3. Click “Add” to add the desired keyboard layout

3.4 File System Integration with Host System

A virtual machine is completely self-contained and its file system is not accessible from the host system. But because it acts as any host on the same local network, common network file services can be used.

The default setup uses Samba to share files over the network. Do the following to connect the home directory of user “dev” as a network drive under Windows:

1. Start Windows Explorer
2. Click menu “Tools” → “Map Network Drive”
3. In the “Drive” box, select a letter to associate the network drive with
4. In the “Directory” box, type the UNC path for the virtual machine and shared resource as: “\\address\dev” (replace address with the IP address of the virtual machine)
5. Type login name “dev” and corresponding password when prompted.
6. A network drive should now appear as a drive under “My Computer”

Please consult <http://support.microsoft.com> for more information about mapping network drives and Windows.

NFS exports are also supported, but the only default export is typically used by networked Readers for mounting a root file system, which is described in section 5 Reader Network Boot.

3.5 Upgrading to the Latest SDK Version

The SDK installed from the DVD must be upgraded to match the system software release in the target Readers. Upgrades are delivered in a zip file that contains new versions of libraries, header files and the Reader’s root file system. Perform an upgrade as follows:

1. Transfer the zip file to the virtual machine. The easiest way to do this is to connect the home directory of user “dev” as described in section 3.4 File System Integration with Host System and then copy the file in Windows Explorer.
2. Login to the SDK virtual machine as user dev and open a terminal window
3. Unzip the file in the SDK virtual machine.
4. Run the INSTALL script.

Steps 2 and 3 may be done as shown below. Bold text should be written by the user. It is necessary to enter the root password to upgrade the Reader’s root file system in /tftpboot/roofts. The old version of the file system, including modifications, will be renamed to “rootfs-YYYY-MM-DD_HH:MM:SS”, where YYYY etc. is the current date and time.

```
dev@tagmaster-sdk:~$ unzip sdk-1.4.0.zip
Archive:  sdk-1.4.0.zip
   creating: sdk-1.4.0/
   ...
dev@tagmaster-sdk:~$ cd sdk-1.4.0
dev@tagmaster-sdk:~/sdk-1.4.0$ ./INSTALL

This will upgrade your TagMaster SDK
Press ENTER to continue...
`tm-sdk/include/taglib.h' -> `/opt/tm-sdk/include/taglib.h'
...

Enter the root password to upgrade /tftpboot/rootfs
(old rootfs will be copied to rootfs-YYYY-MM-DD_HH:MM:SS)
Password: qwerty
bin/
bin/ash
...
```

```
End of installation
dev@tagmaster-sdk:~/sdk-1.4.0$
```

3.6 Backup

Files that are created in the virtual machine will be stored in the virtual disk file. Performing a backup on regular basis is recommended.

A convenient way of performing a snapshot of the complete virtual machine is to copy the virtual disk file but it requires more disk space than just copying the necessary files.

4 Building Reader Application Software

This section describes how to build custom Reader application software, how to copy it to a Reader, and how to execute the software.

4.1 Target Toolchain

A custom GNU toolchain for building target software is installed in the directory “/opt/tm-sdk”. The prefix for the compiler and binary tools is “arm-softfloat-linux-gnu-”. The GCC compiler is consequently invoked with “arm-softfloat-linux-gnu-gcc”.

The paths to tools in the toolchain have been added to the PATH environment variable for the user “dev”.

The target toolchain outputs binary files for the target and these cannot be executed on the host. To build software that is executable on the host, use the native toolchain installed in Debian.

The content of “/opt/tm-sdk” can be copied and moved to a stand-alone Linux host. It is however not recommended because it requires extensive configuration, which is not covered by this document. If the toolchain is copied to another system, the virtual machine is of course not needed.

But as described in this document, the virtual machine release is more than just a tool-chain for cross-compilation. It is a complete development infrastructure.

4.2 Software Library: taglib

The software library taglib is documented in the taglib header file “/opt/tm-sdk/include/taglib.h”. For detailed information about taglib, see the taglib Software Library Specification [1].

Software for the Reader is not required to use taglib, but in order to control the RFID properties of the Reader (for instance set frequency, read tags, and so forth) taglib is required.

The taglib library is by default installed in all Readers and in this development environment.

4.3 Application Template and Examples

An application software template is installed in the home directory of user “dev”, which is “/home/dev/template”.

The template is a good starting point for a custom Reader application that uses taglib. Just copy that directory and change it to a more suitable name. Edit the “template.c” file and use “make” to build application software for target.

A Makefile is used by the make tool. Very briefly it can be described as a definition of how to compile and build a software. The template Makefile does more than building a Reader executable. It also shows how make can be used to install the Reader application software in a simple manner.

Type “make install” at the prompt to install the application to the exported file system. The default directory “/tftpboot/rootfs/tag/bin” can be remotely mounted by Readers over network using NFS (see section 5 Reader Network Boot).

Note: If the name of the source code file is changed, it is also necessary to change the Makefile in the same directory.

Start a terminal and follow the example terminal output below to build and install the template application software (bold text is user input).

```
dev@tagmaster-sdk:~$ cd template/
dev@tagmaster-sdk:~/template$ pwd
/home/dev/template
dev@tagmaster-sdk:~/template$ make
arm-softfloat-linux-gnu-gcc -g -W -Wall -Wstrict-prototypes
-Wdeclaration-after-statement -fsigned-char -I/opt/tm-dk/include
-mlittle-endian -Wno-trigraphs -fno-strict-aliasing
-fno-omit-frame-pointer -march=armv4 -mtune=arm9tdmi -o template
template.c -L/opt/tm-sdk/lib -ltag
dev@tagmaster-sdk:~/template$ ls
Makefile  template  template.c
dev@tagmaster-sdk:~/template$ make install
cp –p template /tftpboot/rootfs/tag/bin
dev@tagmaster-sdk:~/template$
```

Source code examples using taglib are available in the directory “/opt/tm-sdk/examples”. The examples illustrate various software design aspects that are common to most Reader application software.

The examples can be built using “make” and installed by typing “make install” at the command prompt.

4.4 Editors

Because the source code files are plain text files any text editor can be used. Several editors are installed by default.

A common choice for programmers is “xemacs”, which is an advanced editor and probably also one of the most complex to use. There is also a simpler editor call “gedit”. Both editors support C syntax highlighting.

It is not necessary to edit the source code files using Debian GNU/Linux. If the home directory of the developer user has been mapped as a network drive in Windows, it is possible to edit the files with a text editor running in Windows and then build using the virtual machine.

4.5 Installing Application Software on Reader

It is recommended to use the Ethernet interface when installing custom application software to a Reader. It is much faster and more convenient than using the service interface.

In order to install application software to a Reader using its network interface, the Reader must be connected to the same network as the virtual machine. For information

about the different methods on how to install and copy files to and from a Reader, see the GEN4 Reader User's Manual [2].

Developing Reader application software usually implies numerous software builds and tests. During development of application software and when it is not required that the application file is actually stored in a Reader's Flash file system, using the network file system service that Debian provides is recommended (see section 5 Reader Network Boot).

If a file is required to reside in a Reader's Flash file system, the most straightforward approach is to use "scp", which is a secure remote copy utility.

For example, to copy the file "/home/dev/template/template" to directory "/tag/bin" on a Reader with IP address "192.168.0.2", follow the terminal listing below and enter root password when prompted.

```
dev@tagmaster-sdk:~/template$ scp template
root@192.168.0.2:/tag/bin
root@192.168.0.2's password:
template                      100%   11KB  10.9KB/s   00:00
dev@tagmaster-sdk:~/template$
```

Read the GEN4 Reader User's Manual [2] for more information about controlling (starting and stopping) Reader processes, setting the default Reader application, and so forth.

5 Reader Network Boot

This section describes how to setup GEN4 Readers to perform a network boot instead of the default boot from on-board Flash.

Booting a Reader from a network is convenient when developing software for Readers.

When booting a Reader from a network there is no need to move files back and forth as the Reader and the development environment actually share the file system over a network. There are also no disk space constraints, for instance large log files are not an issue.

Having more than one Reader using the same network file system simultaneously is possible. Note however that all Readers access the exact same files.

Note: Using a network file system to run a Reader requires that the exported file system is always available. For critical Reader installations, storing files in each Reader's Flash file system is recommended.

5.1 File System

On the virtual machine there is a directory called "/tftpboot", which is made accessible through a tftp server. The kernel image resides in this directory together with a second directory called "/tftpboot/rootfs", which contains a complete Reader root file system. This second directory is exported through an NFS server that runs in Debian.

Files that are copied or moved to the root file system directory are instantly accessible (executable, readable and writable depending on the file access rights) from a Reader that mounts this directory.

5.2 Configuring Reader Boot Settings

Connect to the service interface of the Reader and abort the autoboot to enter the U-Boot user interface (see GEN4 Reader User's Manual [2] for information about the service interface). In order for a Reader to be able to boot over the network, some U-Boot environment variables must be set using the U-Boot command "setenv".

Type "help" at the U-Boot prompt for a list of available commands, and type "help setenv" for more information about the "setenv" command.

Note: Environment variables set by "setenv" are only stored in volatile memory, to write settings to Flash, the command "saveenv" must be invoked after all necessary variables have been set.

Perform the following steps to configure a Reader to boot from network, that is download kernel and mount network file system provided by the virtual machine:

1. Set server IP address to the IP address of the virtual machine.
2. Set IP address of the Reader, which should be on the same network as the virtual machine.
3. Set name of kernel image to load during boot. Check directory "/tftpboot" for available kernel images, which is typically a file named something similar to "2_6_16_53".

4. Specify the exported NFS directory to mount as root file system during Reader boot, typically “/tftpboot/rootfs”, which is the directory that is exported by default by the virtual machine.
5. Set default boot mode, that specifies if the Reader automatically boots from network when reset or starts from Flash as it does by default. To change the default boot type (network or Flash), change the value of the “bootcmd” variable.
    - Boot from Flash: Set variable to “run boot_flash”
    - Boot from network: Set variable to “run boot_net”

When autoboot is stopped it is also possible to manually select to boot from the network or from Flash by typing “run boot_net” or “run boot_flash” respectively. This starts a small script that initiates the corresponding boot sequence.

Below is a terminal listing that shows how the environment variables are set so that a Reader automatically boots from network. The “boot” command starts the boot sequence as specified by the “bootcmd” variable.

Note: IP addresses used in this example must be set the to the IP addresses of the virtual machine and the Reader.

```
tagmaster-boot 1.1 (Jan 25 2007 - 15:12:29)

Uncompressing image...

Uncompressing...done.


U-Boot 1.1.4 (Sep 21 2007 - 09:05:03)

U-Boot code: 21F00000 -> 21F16054  BSS: -> 21F33004
RAM Configuration:
Bank #0: 20000000 32 MB
Flash: 16 MB
OSR: 00000000, USR: 00000000
In:    serial
Out:   serial
Err:   serial
PHY not connected!!
Hit any key to stop autoboot:  0
U-Boot> setenv serverip 193.15.235.182
U-Boot> setenv ipaddr 193.15.235.136
U-Boot> setenv kernel 2_6_16_53
U-Boot> setenv nfsroot /tftpboot/rootfs
U-Boot> setenv bootcmd run boot_net
U-Boot> saveenv
Saving Environment to Flash...
.
Un-Protected 1 sectors
Erasing Flash...
. done
Erased 1 sectors
Writing to Flash... done
.
```

```
Protected 1 sectors
U-Boot> boot
```

When the Reader boots from network, it starts by downloading the kernel image using tftp. After unpacking and loading, the kernel mounts the root file system and starts the init process. Below is a listing of a successful boot, which has been edited to fit this document.

```
TFTP from server 193.15.235.182; our IP address is
193.15.235.136
Filename '2_6_16_53'.
Load address: 0x21000000
Loading:
#################################################################
#################################################################
#################################################################
         ########################################
done
Bytes transferred = 1205672 (1265a8 hex)
.................................................................
.....................................
Un-Protected 130 sectors
## Booting image at 21000000 ...
   Image Name:   Linux-2.6.16.53-at91-tm
   Image Type:   ARM Linux Kernel Image (uncompressed)
   Data Size:    1205608 Bytes =  1.1 MB
   Load Address: 20008000
   Entry Point:  20008000
   Verifying Checksum ... OK
OK

Starting kernel ...

Uncompressing
Linux.............................................................
.................. done, booting the kernel.
Linux version 2.6.16.53-at91-tm (maed@BlackBox) (gcc version
3.4.4) #1 Tue Sep 25 13:22:17 CEST 2007
CPU: ARM920Tid(wb) [41129200] revision 0 (ARMv4T)
Machine: Atmel AT91RM9200-DK
Memory policy: ECC disabled, Data cache writeback
Clocks: CPU 179 MHz, master 59 MHz, main 18.432 MHz
CPU0: D VIVT write-back cache
~
Looking up port of RPC 100003/2 on 193.15.235.182
Looking up port of RPC 100005/1 on 193.15.235.182
VFS: Mounted root (nfs filesystem).
Freeing init memory: 96K
init started:  BusyBox v1.2.1 (2006.09.28-11:48+0000) multi-call
binary
Starting pid 689, console /dev/ttyS0: '/etc/init.d/rc.sysinit'
~
Welcome to TagMaster Linux
tag login:
```

To set Reader to boot from Flash again, just set environment variable “bootcmd” to refer to the “run boot_flash” script instead:

```
U-Boot> setenv bootcmd run boot_flash
U-Boot> saveenv
```

.

6 Frequently Asked Questions

What if there is no network access?	• Start by verifying that your host system has a network connection. For instance by checking that there is an Ethernet cable connected and that the activity and link status LEDs are active. • Verify that the Network Interface Controller (NIC) is activated in the Debian system, by going to “Desktop” → “Administration” → “Networking”, select your NIC and click “Activate”. • Check that the network settings have been configured as described in section 3.3.2 IP Address.
How do I get out of the virtual machine?	Press Ctrl and Alt keys to return mouse and keyboard control to the host system.
Unable to compile because arm-softfloat-linux-gnu-gcc command is not found?	If you get an error output that says something like “make: arm-softfloat-linux-gnu-gcc: Command not found”, verify that you are not logged in as administrator (root). The PATH variable is only set up for the developer user so building software as the administrator is not possible with the default configuration.

7 Contact

For any further inquiries, please contact TagMaster AB.

7.1 Technical Support

Phone: + 46 8 632 19 50
Fax: +46 8 750 53 62
E-mail: support@tagmaster.com

7.2 Office

Phone: +46 8 632 19 50
Fax: +46 8 750 53 62
E-mail: sales@tagmaster.com
Web: www.tagmaster.com