» User Guide «

HMITR-104

uEFI BIOS

Doc. ID: 1044-1629, Rev. 1.0
March 29, 2011

Revision History

Publication Title:	HMITR-104 uEFI BIOS User Guide	
Doc. ID:	1044-1629	
Rev.	**Brief Description of Changes**	**Date of Issue**
1.0	Initial issue based on the uEFI BIOS version R10	29-Mar-2011

Imprint

Kontron Modular Computers GmbH may be contacted via the following:

MAILING ADDRESS

Kontron Modular Computers GmbH

Sudetenstraße 7

D - 87600 Kaufbeuren Germany

TELEPHONE AND E-MAIL

+49 (0) 800-SALESKONTRON

sales@kontron.com

For further information about other Kontron products, please visit our Internet web site: www.kontron.com.

Disclaimer

Table of Contents

Chapter 1

Starting uEFI BIOS Setup

This page has been intentionally left blank.

1. Starting uEFI BIOS Setup

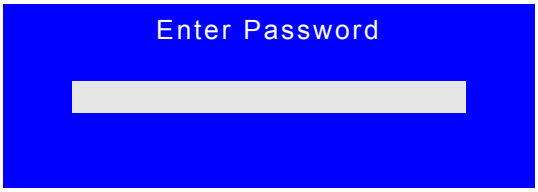
The HMITR-104 is provided with a Kontron-customized, pre-installed and configured version of Aptio® (referred to as uEFI BIOS in this manual), AMI's next generation BIOS firmware based on the Unified Extensible Firmware Interface (uEFI) specification and the Intel® Platform Innovation Framework for EFI. This uEFI BIOS provides a variety of new and enhanced functions specifically tailored to the hardware features of the HMITR-104. This user guide reflects the uEFI BIOS version R10.

To take advantage of these functions, the uEFI BIOS comes with a Setup program which provides quick and easy access to the individual function settings for control or modification of the uEFI BIOS configuration.

The Setup program allows the accessing of various menus which provide functions or access to sub-menus with more specific functions of their own. The individual menus and the configurable functions are described in this guide.

To start the uEFI BIOS Setup program, follow the steps below:

1. Power up the HMITR-104.
2. Wait until the first characters appear on the screen.
3. Press the <DEL> key.
4. If the uEFI BIOS is password-protected, a window such as the one below will appear:



   Enter either the User password or the Administrator password (refer to Chapter 4, Security Setup, for further information), press <RETURN>, and proceed with step 2.

5. A Setup menu with the following token attributes will appear.
   The currently active menu and the currently active uEFI BIOS Setup item are highlighted in white.

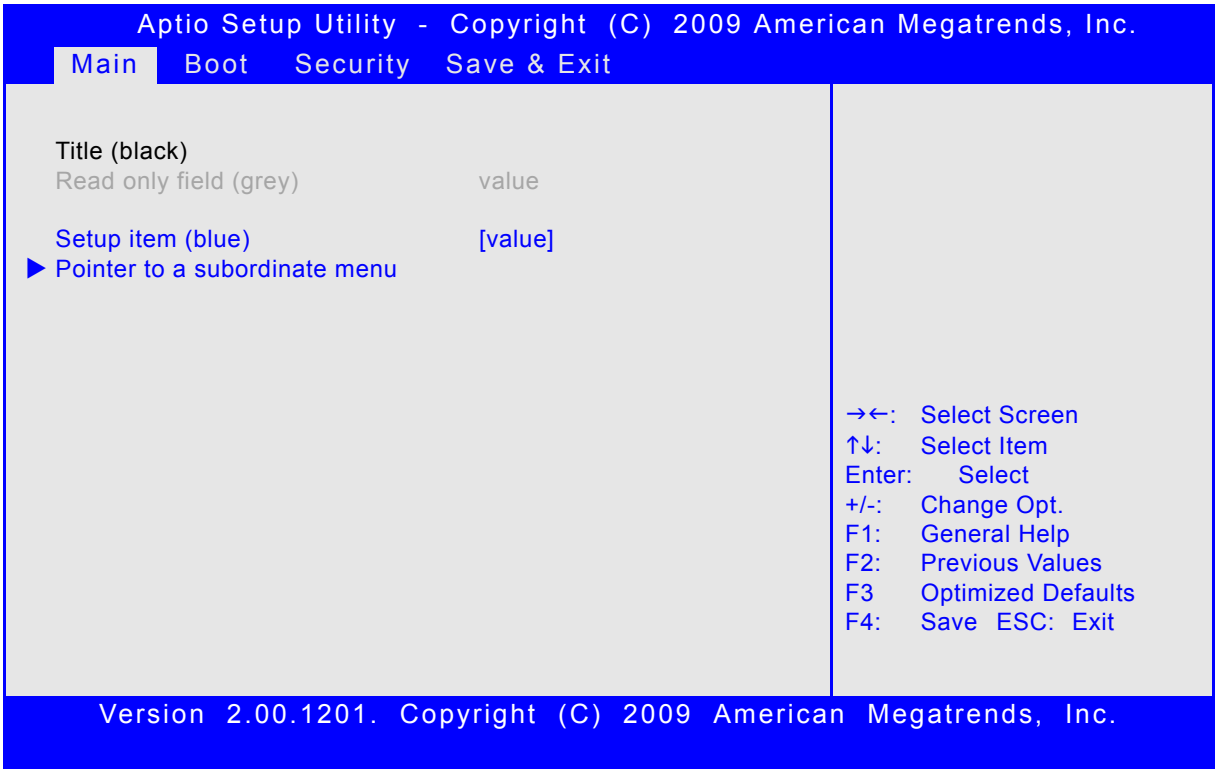
1.1 Main Setup Menu

The Main setup menu is the first screen that appears after starting the Setup program.

At the top of this screen and all of the other major screens, there is a setup menu selection bar, which permits access to all of the other major setup menus. These menus are selected via the left-right arrow keys.

All setup menu screens have two main frames. The left frame displays all the functions that can be configured. They are displayed in blue. Functions displayed in gray provide information about the status or the operational configuration.

The right frame displays the key legend. Above the key legend there is an area reserved for a text message. When a function is selected in the left frame, it is displayed in white. Often a text message will accompany it.



1.2 Navigation

The HMITR-104 uEFI BIOS setup program uses a hot key-based navigation system. A hot key legend is located in the right frame on most setup screens.The following table provides information concerning the usage of these hot keys.

HOT KEY	DESCRIPTION
<F1>	The <F1> key is used to invoke the General Help window.
<F2>	The <F2> key is used to restore the previous values.
<F3>	The <F3> key is used to load the defaults.
<F4>	The <F4> key is used to save the current settings and exit the uEFI BIOS Setup.
→ ← Left/Right	The *Left and Right* <Arrow> keys are used to select a major Setup screen. For example: Main Screen, Advanced Screen, Chipset Screen, etc.
↑ ↓ Up/Down	The *Up and Down* <Arrow> keys are used to select a Setup function or a sub-screen.
+ - Plus/Minus	The *Plus and Minus* <Arrow> keys are used to change the field value of a particular Setup function, for example, system date and time.
<ESC>	The <ESC> key is used to exit a menu or the uEFI BIOS Setup. Pressing the <ESC> key in a sub-menu causes the next higher menu level to be displayed. When the <ESC> key is pressed in a major Setup menu, the uEFI BIOS Setup is terminated without saving any changes made.
<Enter>	The <Enter> key is used to execute a command or select a menu.

This page has been intentionally left blank.

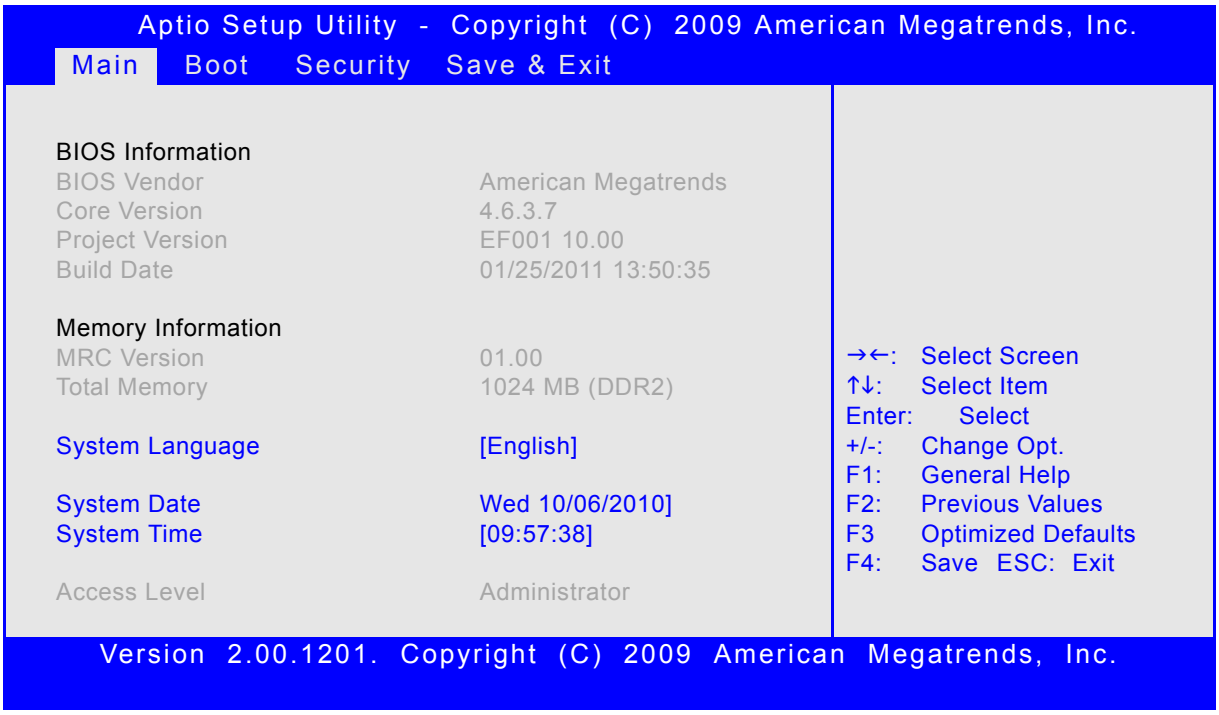
Chapter 2

Main Setup

This page has been intentionally left blank.

2. Main Setup

Upon entering the uEFI BIOS Setup program, the Main setup screen is displayed. This screen lists the main setup sub-screens and provides very basic system information as well as functions for setting the system time and date. In addition, the remaining major setup menus can be accessed from this screen. This screen can also be selected from any other major setup screen by using the Main tab.



2.1 BIOS Information

This function provides display only information concerning the uEFI BIOS.

2.2 System Language

SETTING	DESCRIPTION
English	Use this function to select the system language. Currently, only English is supported.

2.3 System Date

SETTING	DESCRIPTION
<MM/DD/YYYY>	Use this function to change the system date. Select System Date using the Up and Down <Arrow> keys. Enter the new values through the keyboard. Press the Left and Right <Arrow> keys to move between fields.

2.4 System Time

SETTING	DESCRIPTION
<HH:MM:SS>	Use this function to change the system time. Select System Time using the Up and Down <Arrow> keys. Enter the new values through the keyboard. Press the Left and Right <Arrow> keys to move between fields.

Note: The time is in 24-hour format. For example, 5:30 A.M. appears as 05:30:00, and 5:30 P.M. as 17:30:00.

2.5 Access Level

This function provides display-only information concerning the uEFI BIOS Setup accessibility for the current Setup session. Depending on the type of password protection used, one of the following settings is displayed:

SETTING	DESCRIPTION
Administrator	This setting indicates that read/write access to all setup options is available.
User	This setting indicates that only a limited subset of all setup options is modifiable.

Note: If no password is set, the access setup is Administrator.

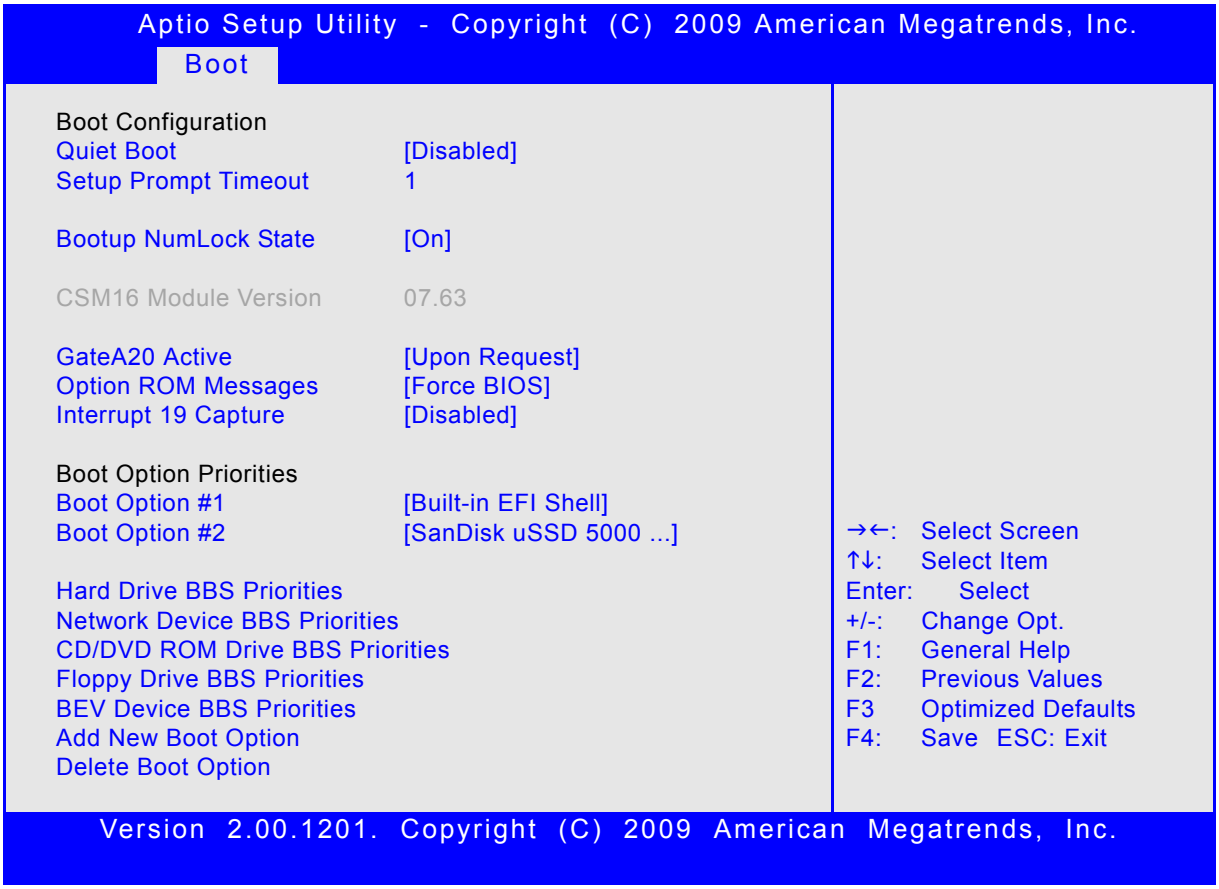
Chapter 3

Boot Setup

This page has been intentionally left blank.

3. Boot Setup

Select the Boot tab to enter the Boot Setup screen. This screen lists the sub-screens for boot configuration and boot device priority.



3.1 Boot Configuration

3.1.1 Quiet Boot

This function is used to display either POST output messages or a splash screen during boot-up.

SETTING	DESCRIPTION
Disabled	Use this setting to display POST output messages during boot-up.
Enabled	Use this setting to display a splash screen during boot-up.

Default setting: Disabled

3.1.2 Setup Prompt Timeout

This integer function is used to set an additional time the POST should wait for the operator to press the key to enter setup. The time is entered in seconds.

SETTING	DESCRIPTION
1 ⋮ 65535	Use one of these settings to specify the setup prompt timeout.

Default setting: 2

3.1.3 Bootup NumLock State

This function is used to set the state of the keyboard's numlock function after POST.

SETTING	DESCRIPTION
On	Use this setting to switch on the keyboard's numlock function after POST.
Off	Use this setting to switch off the keyboard's numlock function after POST.

Default setting: On

3.1.4 CSM16 Module Version

This function provides display-only information concerning the CSM Module and is intended for internal use only.

3.1.5 GateA20 Active

This function is used to enable or disable GateA20.

SETTING	DESCRIPTION
Upon Request	Use this setting to disable GA20 in the uEFI BIOS.
Always	Use this setting to prevent the system from disabling GA20.

Default setting: Upon Request

3.1.6 Option ROM Messages

This function is used to control the messages of the loaded PCI option ROMs.

SETTING	DESCRIPTION
Force BIOS	Use this setting to force to a BIOS-compatible output. This will show the option ROM messages.
Keep Current	Use this setting to keep the current video mode. This will suppress option ROM messages. Option ROMs requiring interactive inputs may not work properly in this mode.

Default setting: Force BIOS

3.1.7 Interrupt 19 Capture

This function is used to specify if legacy PCI option ROMs are allowed to capture software interrupt 19h.

SETTING	DESCRIPTION
Disabled	Use this setting to prevent legacy PCI option ROMs from capturing software interrupt 19h.
Enabled	Use this setting to allow legacy PCI option ROMs to capture software interrupt 19h.

Default setting: Disabled

3.2 Boot Option Priorities

3.2.1 Boot Option #1..2

These functions are used to form the boot order and are dynamically generated. They represent either a legacy BBS (BIOS Boot Specification) class of devices or a native EFI boot entry. Press Return on each option to select the BBS class / EFI boot entry desired.

3.2.2 Hard Drive/Network Device/CD/DVD ROM Drive/Floppy Drive/BEV Device BBS Priorities

These functions lead to sub-menus that allow configuring the boot order for a specific device class. These options are only visible if at least one device for this class is present. These functions are dynamically generated.

3.2.3 Add New Boot Option

This function is used to create a native uEFI boot option. Refer to the user manual for the respective native uEFI operating system further information about creating a boot option.

3.2.4 Delete Boot Option

This function is used to delete a native uEFI boot option. Refer to the user manual for the respective native uEFI operating system further information about deleting a boot option.

Note: Do not delete the “Built-in EFI Shell” boot option as this would remove the uEFI Shell from the boot order. In case the uEFI Shell got removed, use “Save & Exit” / “Boot Override” / “Built-in EFI Shell” to recover.

This page has been intentionally left blank.

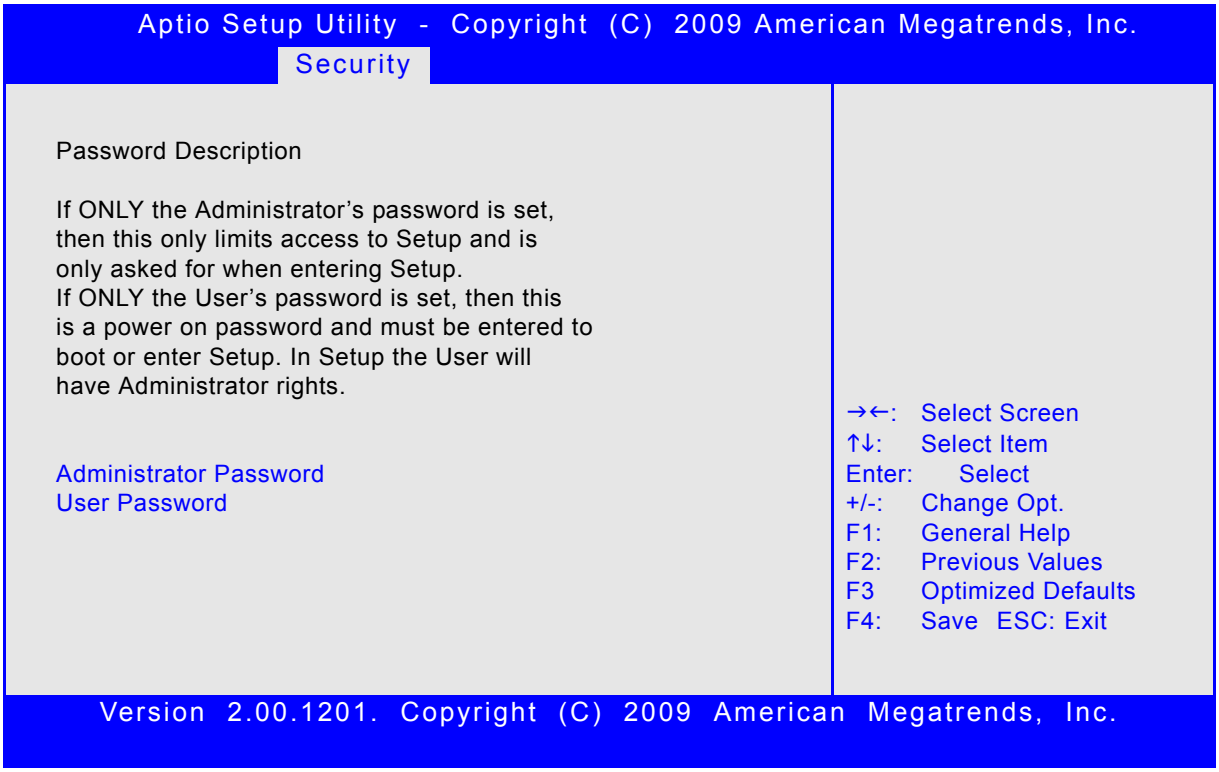
Chapter 4

Security Setup

This page has been intentionally left blank.

4. Security Setup

Select the Security tab to enter the Security Setup screen. This screen provides information about the passwords and functions for specifying the security settings.



The following modes of security are provided:.

SETTING	DESCRIPTION
No password is set	Booting the system as well as entering the Setup is unsecured.
Only Administrator password is set	Booting the system is unsecured. For entering the Setup, the Administrator password is required.
Only User password is set	The password is required for booting the system as well as for entering the Setup menu. On every startup, the user will be asked for the password.
Both User and Administrator passwords are set	Booting the system is unsecured. For entering the Setup, a password is required. If the User password is entered here, most of the Setup entries are read only; only entries related to the boot sequence can be modified. Entering the Administrator password provides full access to all Setup entries.

4.1 Administrator Password

This function is used to set, change or delete the Administrator password. If there is already a password installed, the system asks for this first. To clear a password, simply enter nothing and acknowledge by pressing Return. To set a password, enter it twice and acknowledge by pressing Return.

Note: The password is case sensitive.

4.2 User Password

This function is used to set, change or delete the User password. If there is already a password installed, the system asks for this first. To clear a password, simply enter nothing and acknowledge by pressing Return. To set a password, enter it twice and acknowledge by pressing Return.

Note: The password is case sensitive.

4.3 Remember the Password

It is highly recommended to keep a record of all passwords in a safe place. Forgotten passwords may lead to being completely locked out of the system. Booting may not be possible, and in worst case the uEFI BIOS Setup program will also not be accessible.

If the system cannot be booted because neither the User password nor the Administrator password are known, contact Kontron for further assistance.

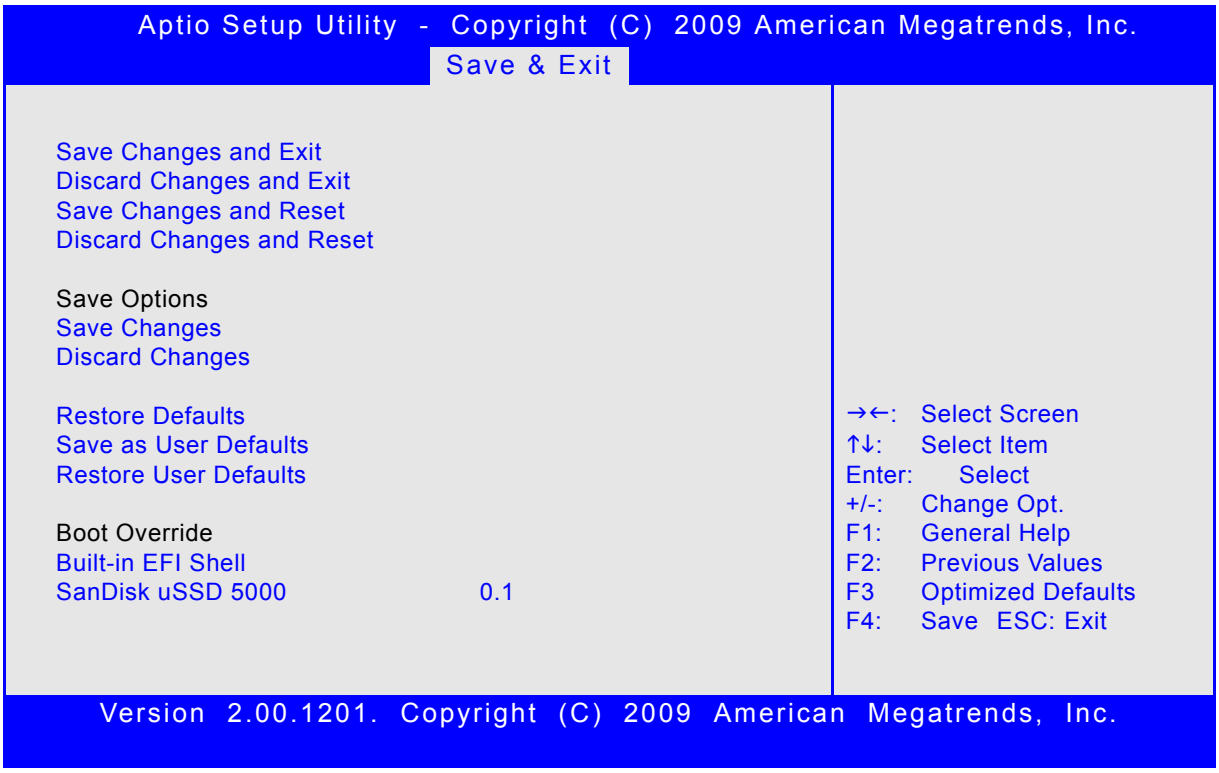
Chapter 5

Save & Exit

This page has been intentionally left blank.

5. Save & Exit

Select the Save & Exit tab to enter the Save & Exit menu screen. This screen provides functions for handling changes made to the uEFI BIOS settings and the exiting of the Setup program.



5.1 Save Changes and Exit

This function is used to save all changes made within the Setup to Flash. This function continues the boot process as long as no option was altered that requires a reboot.

Note: The Setup will ask for confirmation prior to executing this command.

5.2 Discard Changes and Exit

This function is used to discard all changes made within the Setup. This function continues the boot process.

Note: The Setup will ask for confirmation prior to executing this command.

5.3 Save Changes and Reset

This function is used to save all changes made within the Setup to Flash. This function performs a reboot afterwards.

Note: The Setup will ask for confirmation prior to executing this command.

5.4 Discard Changes and Reset

This function is used to discard all changes made within the Setup. This function performs a reboot afterwards.

Note: The Setup will ask for confirmation prior to executing this command.

5.5 Save Changes (Save Options)

This function is used to save all changes made within the Setup to Flash. This function returns to Setup.

Note: The Setup will ask for confirmation prior to executing this command.

5.6 Discard Changes (Save Options)

This function is used to discard all changes made within the Setup. This function returns to Setup.

Note: The Setup will ask for confirmation prior to executing this command.

5.7 Restore Defaults (Save Options)

This function is used to restore all tokens to factory default.

Note: The Setup will ask for confirmation prior to executing this command.

5.8 Save as User Defaults (Save Options)

This function is used to save all current settings as user default. The current setup state can later be restored using Restore User Defaults.

Note: The Setup will ask for confirmation prior to executing this command.

5.9 Restore User Defaults (Save Options)

This function is used to restore all tokens to settings previously stored by Save as User Defaults.

Note: The Setup will ask for confirmation prior to executing this command.

5.10 Boot Override

This group of functions includes a list of tokens, each of them corresponding to one device within the boot order. Select a drive to immediately boot that device regardless of the current boot order. If booting to EFI Shell this way, an exit from the shell returns to Setup.

Chapter 6

The uEFI Shell

This page has been intentionally left blank.

6. The uEFI Shell

The Kontron uEFI BIOS features a built-in and enhanced version of the uEFI Shell. For a detailed description of the available standard shell scripting refer to the EFI Shell User's Guide. For a detailed description of the available standard shell commands, refer to the Shell Command Manual 1.0. Both documents can be downloaded from the EFI and Framework Open Source Community homepage (https://efi-shell.tianocore.org) under the "Documents and Files" section.

Please note that not all shell commands described in the Shell Command Manual 1.0 are provided by the Kontron uEFI BIOS.

6.1 Introduction, Basic Operation

The uEFI Shell forms an entry into the uEFI boot order and is the first boot option by default. It is simply started by putting the uEFI Shell first in boot and running the board as usual.

6.1.1 Shell Startup

If the shell is executed, it displays its signon message followed by a list of detected devices. The output produced by the device mapping table can vary depending on the board's configuration.

```
EFI Shell version 2.00 [4.631]
Current running mode 1.1.2
Device mapping table
  fs0      :Removable HardDisk - Alias hd33b0b0b blk0
            Acpi(PNP0A03,0)/Pci(1D|7)/Usb(1, 0)/Usb(1, 0)/HD(Part1,Sig17731773)
  fs1      :Removable BlockDevice - Alias f33b0c0 blk1
            Acpi(PNP0A03,0)/Pci(1D|7)/Usb(1, 0)/Usb(2, 0)
  blk0     :Removable HardDisk - Alias hd33b0b0b fs0
            Acpi(PNP0A03,0)/Pci(1D|7)/Usb(1, 0)/Usb(1, 0)/HD(Part1,Sig17731773)
  blk1     :Removable BlockDevice - Alias f33b0c0 fs1
            Acpi(PNP0A03,0)/Pci(1D|7)/Usb(1, 0)/Usb(2, 0)
  blk2     :HardDisk - Alias (null)
            Acpi(PNP0A03,0)/Pci(1F|2)/Ata(Primary,Master)/HD(Part1,SigC811D18D)
  blk3     :BlockDevice - Alias (null)
            Acpi(PNP0A03,0)/Pci(1F|2)/Ata(Primary,Master)
  blk4     :Removable BlockDevice - Alias (null)
            Acpi(PNP0A03,0)/Pci(1D|7)/Usb(1, 0)/Usb(1, 0)

Press the ESC key within 5 seconds to skip startup.nsh, and any other key to
continue.
```

If the ESC key is pressed before the 5-second timeout has elapsed, the shell prompt is shown:

```
Shell>
```

6.2 Kontron Shell Commands

The Kontron uEFI implementation provides the following additional commands related to the specific HW features of the Kontron system:

- **kboardconfig**
- **kboardinfo**
- **kboot**
- **kbootnsh**
- **kclearnvram**
- **kclsp**
- **kmkramdisk**
- **kpassword**
- **kwdt**

The following tables provide information concerning these Kontron-specific commands. Where “RESPONSE” information is provided in “USAGE”, the value indicated in brackets is the currently selected setting. Where “SETTINGS” information is provided, the value indicated in brackets is the default setting.

kboardconfig

FUNCTION:	Configure the non-volatile board settings
SYNTAX:	**kboardconfig** **kboardconfig [-?|<device>|<setting>]** where: ? Show online help <device> Specify device from list <setting> Select configuration type
DESCRIPTION:	The **kboardconfig** command enables the PXE feature or sets the front/ rear I/O configuration of the dedicated device.

USAGE:	Show all possible configurations
	COMMAND / RESPONSE:
	`Shell> kboardconfig`
	`Control nonvolatile board settings`
	`Example: kboardconfig`
	`pxe: Select PXE boot network adapter ([disabled] all gbe_a gbe_b service)`
	`speedstep: Intel Speedstep(disabled [enabled])`
	`htd: Hyper-Threading (disabled [enabled])`

	Show allowed settings e.g. for “pxe”:
	`Shell> kboardconfig pxe`
	`pxe: Select pxe boot network adapter`
	`pxe == disabled`
	`Allowed options: disabled, all, gbe_a, gbe_b, service`

SETTINGS:	**pxe:** Select PXE boot network adapter **disabled:** No PXE boot available **[all]:** Try all Ethernet devices round robin for PXE boot **gbe_a:** Try only Ethernet port A for PXE boot **gbe_a:** Try only Ethernet port B for PXE boot **service:** Try only Ethernet port service port
	speedstep: Enable/disable Intel SpeedStep technology

kboardinfo

FUNCTION:	Show board identification data
SYNTAX:	`kboardinfo`
DESCRIPTION:	The **kboardinfo** command shows a summary of board-specific identification data. It is especially useful for support queries because it contains this data in a concentrated form.

USAGE:	Show board identification data
	COMMAND / RESPONSE:
	`Shell> kboardinfo`
	`KOMaOEMF rev.: 3`
	`Board ID: 0xEF00`
	`Hardware rev.: 0x0`
	`Logic rev.: 0xF`
	`Boot flash: Boot flash 0`
	`Material number:`
	`Hardware index:`
	`Serial number:`
	`EFI article name: SK-EFI-EF001`
	`EFI material number: 1042-9756`
	`EFI index: 10, standard`
	`EFI build time: 13:50:35`
	`EFI build date: 01/25/2011`
	`NorthBridge rev.: 0x3`
	`SouthBridge rev.: 0x1`
	`Microcode: 0x104`
	`CPU ID: 0x20661`
	`CPU Branding: GenuineIntel(R) CPU @ 1.50GHz`

kboardinfo

KOMaOEMF rev.:	Revision of KOMaOEMF protocol
Board ID:	Kontron board identification value (should be for the HMITR-104)
Hardware rev.:	Hardware revision of this board
Logic rev.:	Logic revision of this board
Boot flash:	Current boot Flash: either “Boot flash 0” or “Boot flash 1”
Material number:	Kontron hardware reference number
Hardware index:	Kontron hardware index
Serial number:	This board’s unique serial number
EFI article name:	Kontron uEFI reference name
EFI material number:	Kontron uEFI reference number
EFI index:	Version of this uEFI BIOS
NorthBridge rev.:	Chip revision of the NorthBridge (GMCH die of the Intel® Core™ i7 processor)
SouthBridge rev.:	Chip revision of the SouthBridge (Intel ® QM57)
Microcode:	Currently loaded microcode
CPU ID:	CPUID
CPU Branding:	CPU identification string

kboot

FUNCTION:

Boot a legacy OS
Not to be used for uEFI BootLoaders!

SYNTAX:

```
kboot [-?|-d|-p|-p <path>|-n <name>|-t <type>]
```

where:

?	Show online help
-d	Boot default order
-p <path>	Specify the path to the device to boot from
-n <name>	Specify the device name to boot from
-t <type>	Specify the device type to boot from

Available types are:

floppy
harddrive
cdrom
network
usb-floppy
usb-harddrive
usb-cdrom

DESCRIPTION:

The **kboot** command boots a legacy OS. Boot device can be selected in a very flexible way. If the requested device is not present, boot returns to shell. The **kboot** command cannot boot native uEFI operating systems. But since these are bootable from shell by calling their bootloader, this is not necessary either. If a requested device is present but not bootable, uEFI continues to boot with the next bootable device in the boot order.

USAGE:

Show all connected devices:

COMMAND / RESPONSE:

```
fs0:\> kboot
____BBS_TABLE____
00000 usb-harddrive "USB DISK 26xPMAP"
Device path: Acpi(PNP0A03,0)/Pci(17|0)/Pci(0|0)/
Pci(2|1)/Usb(1,0)/Usb(2,0)
00002 network "Intel UNDI,PXE-1.0 Gold-1 (build Dev)"
00003 network "IBA GE Slot 0600 v1327"
00004 network "IBA GE Slot 0700 v1327"
00001 harddrive "P0: KONTRON MMSATA01"
Device path: Acpi(PNP0A03,0)/Pci(17|0)/Pci(0|0)/
Pci(6|0)/?
```

Boot from device containing the string “Kingston”:

```
fs0:\> kboot -n KONTRON
```

Boot from the first device found that is of type floppy:

```
fs0:\> kboot -t floppy
```

kbootnsh

FUNCTION:	Manage the startup script stored in the Flash
SYNTAX:	**`kbootnsh [-b][-?|-g <filename>|-p <filename>|-d]`** where: -b Display output page by page -? Show online help -g <filename> Store the current boot script to disk. If there is no physical disk drive present, the **kmkramdisk** command may be used. -p <filename> Store the shell script pointed to by filename to Flash. Note: The shell script cannot be larger then 400 bytes. -d Delete the current startup script from Flash.
DESCRIPTION:	The **kbootnsh** command manages the Flash stored startup script. If the shell is launched by the boot process, it executes a shell script stored in the Flash. If the shell script terminates, the shell executes a **kboot -d** command to continue the boot process. However, the shell script can of course contain any other boot command.
USAGE:	Get current startup script to file named boot.nsh **`kbootnsh -g boot.nsh`** Store file named boot.nsh to Flash: **`kbootnsh -p boot.nsh`** Delete startup script: **`kbootnsh -d`**

kclearnvram

FUNCTION:	Clear the NVRAM to restore the system's default settings
SYNTAX:	**`kclearnvram`** No parameters required. For safety reasons this command must be confirmed by pressing "c".
DESCRIPTION:	The **kclearnvram** command allows to clear the system NVRAM. Since all EFI settings are stored inside the NVRAM, the default settings are loaded afterwards.

kclsp

FUNCTION:	Configure clock spreading
SYNTAX:	**`kclsp [-?|-d|-e]`** where: -? show help -d disable clock spreading -e enable clock spreading
DESCRIPTION:	The **kclsp** command enables or disables clock spreading on the onboard core clock generator. Clock spreading can be used to reduce system EMI.
USAGE:	Get help: COMMAND / RESPONSE: **`Shell> kclsp -?`** **`Kontron Clock Spreading Configuration for ICS9LPRS365`** **`-d disable clock spreading`** **`-e enable clock spreading`**

kmkramdisk

FUNCTION:	Create RAMdisk drives
SYNTAX:	**kmkramdisk [-?|-s <size> <name>]** where: -? show help -s <size> <name> create a RAMdisk of given size in Megabytes with the mount point name <name>
DESCRIPTION:	Creates a RAMdisk of variable size. Can be very useful to perform file operations when no real filesystem is connected to the system. Note: The RAMdisk loses its mount point name after all drives are remapped by the **map -r** command. The RAMdisk will then be enumerated as any other connected drive and gain a mount point name like “fs0”. This is not a bug of the **kmkramdisk** command but a normal function of the uEFI framework.
USAGE:	Create RAMdisk: COMMAND / RESPONSE: **rd:\> kmkramdisk -s 5 myramdisk** **Device mapping table** **myramdisk :BlockDevice - Alias (null)** **VenMsg’(93B5F448-127A-4B29-B306-5BE8AAC4826E)** **Success - Force file system to mount** **rd:\> myramdisk:** **myramdisk:\> echo testfile > testfile** **myramdisk:\> ls** **Directory of: myramdisk:** **05/24/08 04:39a 22 testfile** **1 File(s) 22 bytes** **0 Dir(s)**

kpassword

FUNCTION:	Control EFI setup and shell passwords
SYNTAX:	**kpassword [-u|-s]** Control EFI setup and shell passwords
DESCRIPTION:	The **kpassword** command is used to get and set the EFI shell and setup passwords. Both user and superuser (Administrator) passwords can be controlled.
USAGE:	**kpassword [-u|-s]** Control EFI setup and shell passwords Parameters: -u Install or change user password -s Install or change superuser password Call without parameters to get current password status Note: Old passwords must be verified if set. Entering an empty password disables the password.

kwdt

FUNCTION:	Configure the Kontron onboard Watchdog
SYNTAX:	**kwdt [-?|-t <timeindex>]** where: -? Show help -t <timeindex> Configure the Watchdog with the time related to timeindex and activate it with reset routing Call kwdt -h to obtain a list of timeindex values and related times
DESCRIPTION:	The **kwdt** command allows to enable the Kontron onboard Watchdog with reset target before OS boot. This can be used to detect if the OS fails to boot and react by reset. The OS Watchdog driver is required for this functionality to operate.
USAGE:	Get help: COMMAND / RESPONSE: **Shell> kwdt -?** **-t [time] - set Timer** **value 0 = 125ms** **value 1 = 250ms** **value 2 = 500ms** **value 3 = 1s** **value 4 = 2s** **value 5 = 4s** **value 6 = 8s** **value 7 = 16s** **value 8 = 32s** **value 9 = 64s** **value 10 = 128s** **value 11 = 256s** **value 12 = 512s** **value 13 = 1024s** **value 14 = 2048s** **value 15 = 4096s**
	Set Watchdog to 16 seconds and activate it COMMAND / RESPONSE (none): **Shell> kwdt -t 7** Note: Because there is no application which triggers the Watchdog, the system will be reset after 16 seconds in this case. This command should be invoked from a script, followed by an operating system boot, and the OS then has to start triggering the Watchdog.

6.3 uEFI Shell Scripting

6.3.1 Startup Scripting

If the ESC key is not pressed and the timeout is run out, either the Kontron Flash-stored startup is executed, if present, or the uEFI specified **`startup.nsh`** script located under **`\efi\boot\`** on any of the attached drives is executed. If none of the startup scripts is present, or the startup script terminates, the default boot order is continued.

If the shell is started with no interaction, it tries to execute some startup scripts automatically. It searches for scripts in the following order:

1. Kontron Flash-stored startup script

2. If there is no Kontron Flash-stored startup script present, the uEFI specified **`startup.nsh`** script is used. This script must be located on any of the attached FAT formatted disk drives under **`\efi\boot\startup.nsh.`**

If both startup scripts are absent, the shell terminates and the default boot order is continued.

6.3.2 Create a Startup Script

Startup scripts can be created using the uEFI Shell built-in editor **edit** or under any OS with a plain text editor of your choice. To create a **`startup.nsh`** type shell script, simply save the script on any FAT-formatted drive attached to the system under **`\efi\boot\startup.nsh.`** To create a Kontron Flash-stored startup script, the script is to be saved anywhere on a FAT-formatted drive attached to the system and stored to Flash using the built-in uEFI Shell command **kbootnsh**.

6.3.3 Examples of Startup Scripts

6.3.3.1 Automatic Booting from USB Memory Stick

Automatic booting is made from a USB memory stick, if present, otherwise the boot is made from the harddrive.

```
kboot -t usb-harddrive
kboot -t harddrive
```

If neither a USB memory stick nor a harddrive is present, the boot order is continued.

6.3.3.2 Switch On Clock Spreading Prior to Booting from Harddrive

```
kclsp -e
kboot -t harddrive
```

If no harddrive is present, the default order is continued.

6.3.3.3 Execute Shell Script on Other Harddrive

This example executes the shell script named **`bootme.nsh`** located in the root of the first detected disc drive (**`fs0`**).

```
fs0:
bootme.nsh
```

Chapter 7

Updating the uEFI BIOS

This page has been intentionally left blank.

7. Updating the uEFI BIOS

BIOS updates are typically delivered as an update CD ISO image. This ISO image needs just to be burned to a CD and booted. Follow the menu for updating the uEFI BIOS. For further information refer to the update CD documentation.

7.1 BIOS Redundancy Strategy

The HMITR-104 has two sets of EFI Flash chips to form an EFI redundancy strategy. Basic idea behind that is to always have at least one working EFI available regardless if there have been any flashing errors or not.

7.2 Fallback Mechanism

In case of one Flash being corrupted and therefore the board not starting up, the second Flash can be selected via the onboard 2-position DIP switch position 2. For further information refer to Chapter 2.3 in the HMITR-104 User Guide or contact Kontron for further assistance.

7.3 Determining the Active Flash

Sometimes it may be necessary to check which Flash is active. On the AMI Aptio-based uEFI BIOS, the information is available using the EFI shell command “kboardinfo”. For further information, refer to the “kboardinfo” section in the uEFI Shell chapter of this document.

This page has been intentionally left blank.