VKRZA1H-SDHI Tool

For micro SD formatting and copying of the VK-RZ/A1H's Boot Loader and command ini file.

USER MANUAL

Vekatech Ltd., Developers

February 18, 2015

Revision Sheet

Release No.	Date	Revision Description
1.0	2/18/2015	Initial Revision

Table of Contents

GENERAL INFORMATION

1.1. Overview

The SD CARD RAM/Flash/I2C Loader application is intended to prepare the micro SD CARD for using the VK-RZ/A1H's Boot loader with the VK-RZ/A1H development board.

The possible scenarios are as follows:

- programming of the onboard NOR FLASH, e.g. user application or boot loader like “U-BOOT”;
- starting the user application copied to the SD CARD in internal SRAM of RZ/A1H (system with NO-Flash, NO-SDRAM);
- starting the user application copied to the SD CARD in external SDRAM of VK-RZ/A1H.

The above allows the development and use of VK-RZ/A1H without a J-TAG debugger.

SYSTEM SUMMARY

2.1. System Configuration

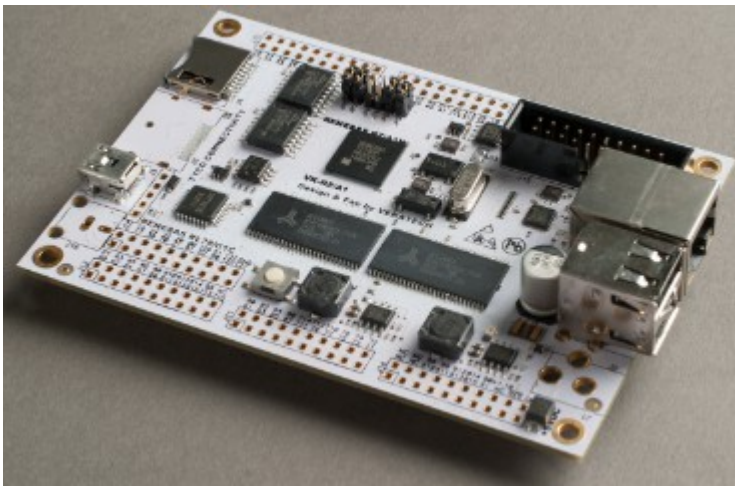
The user must provide the following requirements:

- PC with Windows 7

- VKRZA1H-SDHI Tool for preparing the SD CARD

Can be downloaded from the link below upon free registration:
http://support.vekatech.com/register/?ref=%27VK-RZ+A1H%27

- VK-RZ/A1H Development board



List of distributors:
https://www.rutronik24.com/product/vekatech/vk-rza1h/6475917.html

- Micro SD CARD

- Binary or S37 file for VK-RZ/A1H

The VKRZA1H-SDHI Tool will copy the following sample programs to the micro SD CARD:
- Simple UART application;
- Parallel LCD display demo application;
- Cyclone TCP demo application.

The VK-RZ/A1H's Boot loader is taking control after reset if the boot jumpers are set to “SD CARD” and the ini configuration file is referenced for one of the following actions:
- programing in serial NOR FLASH;
- loading and running in ext. SDRAM;
- loading and running in internal SRAM of RZ/A1H Renesas Cortex A9 ARM microcontroller.

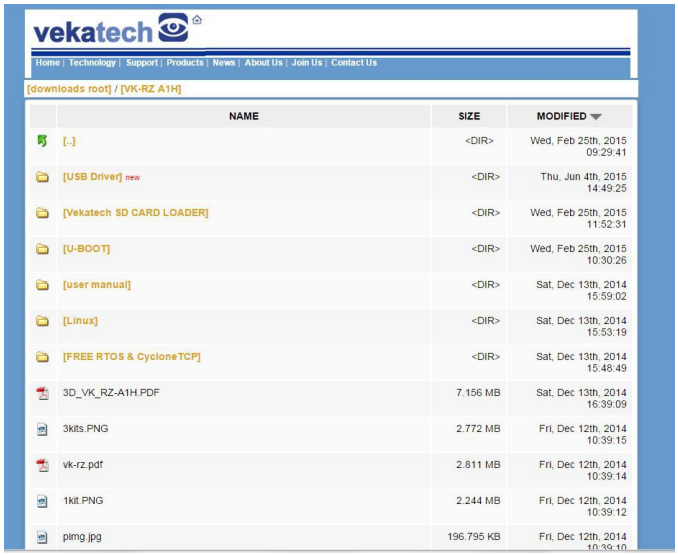
GETTING STARTED

VKRZA1H-SDHI Tool for preparing the micro SD CARD for VK-RZ/A1H development board.

3.1.Installing VKRZA1H-SDHI Tool

3.1.1. Register to receive a download link to your e-mail.

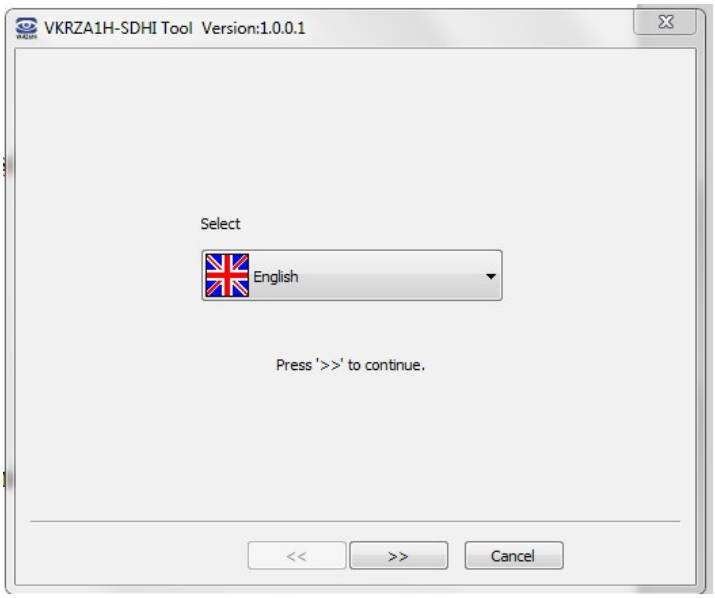
3.1.2. Click on the link in the received e-mail and download the archive from “Vekatech SD CARD LOADER” directory.



3.1.3. Unzip the archive and run as administrator vkrza1h_sdhi-1.0.0.1-install.exe

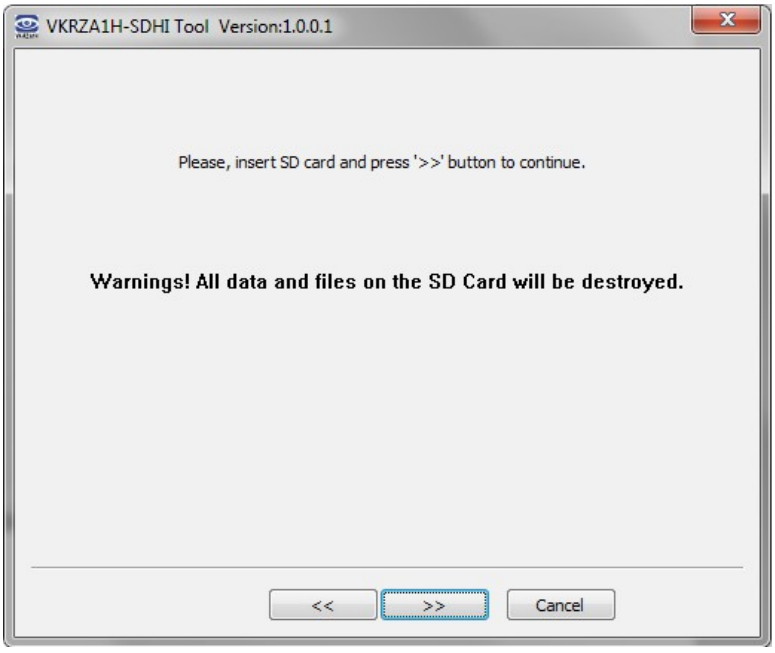
3.2. Micro SD card format and boot loader copying

3.2.1. Run as administrator vkrza1h_sdhi-1.0.0.1.exe



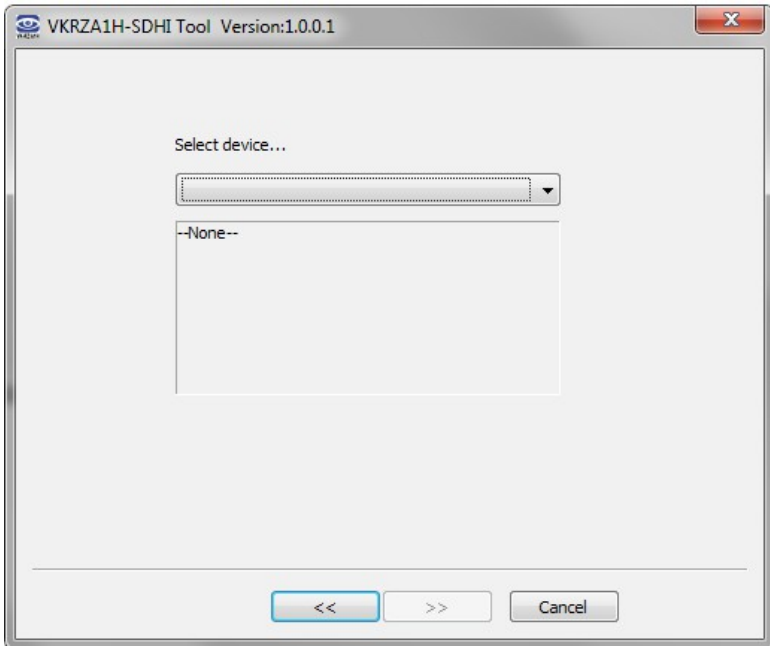
Click the '>>' button.

3.2.2. Insert SD card



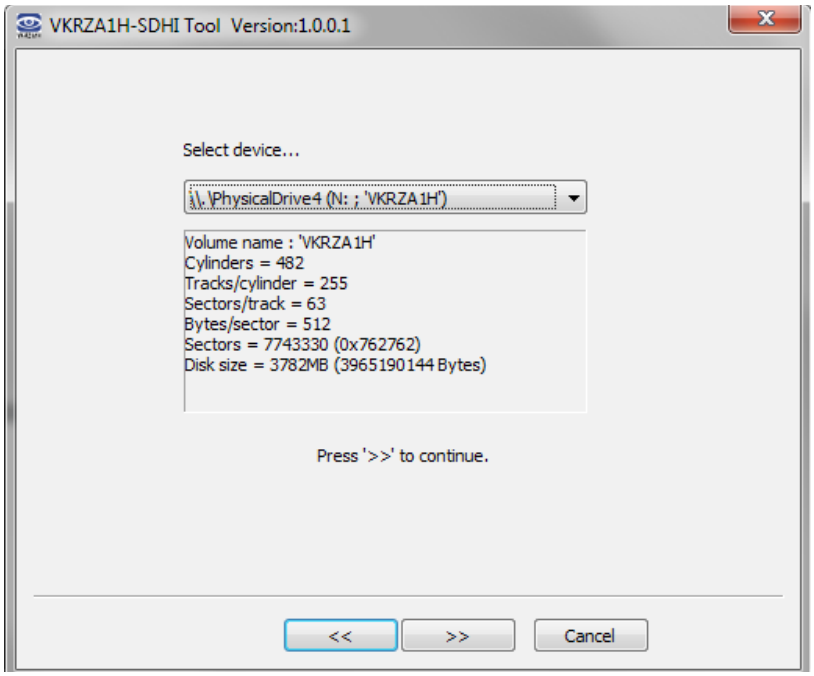
Click the '>>' button.

3.2.3. Select device



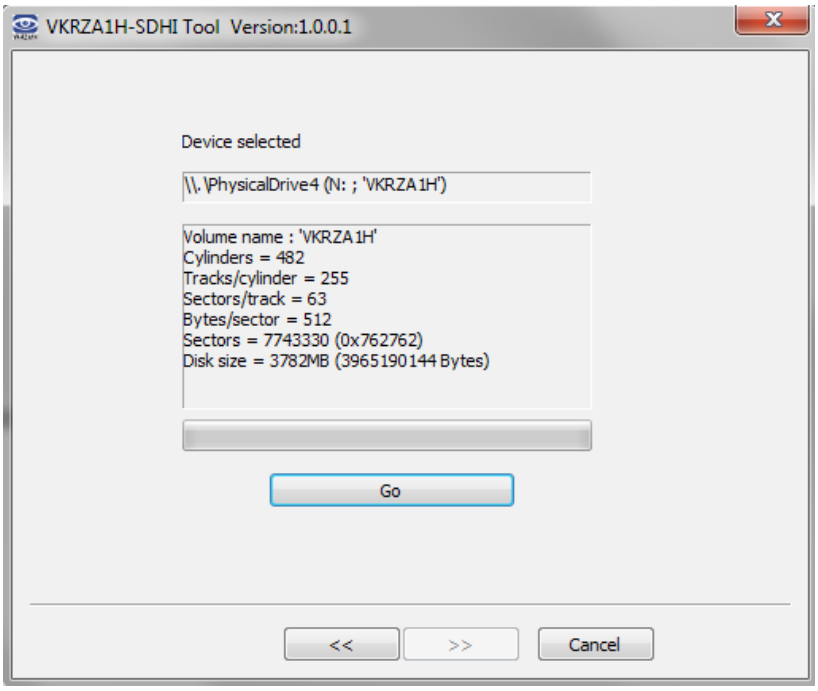
Click on the list box.

3.2.4. Confirm device



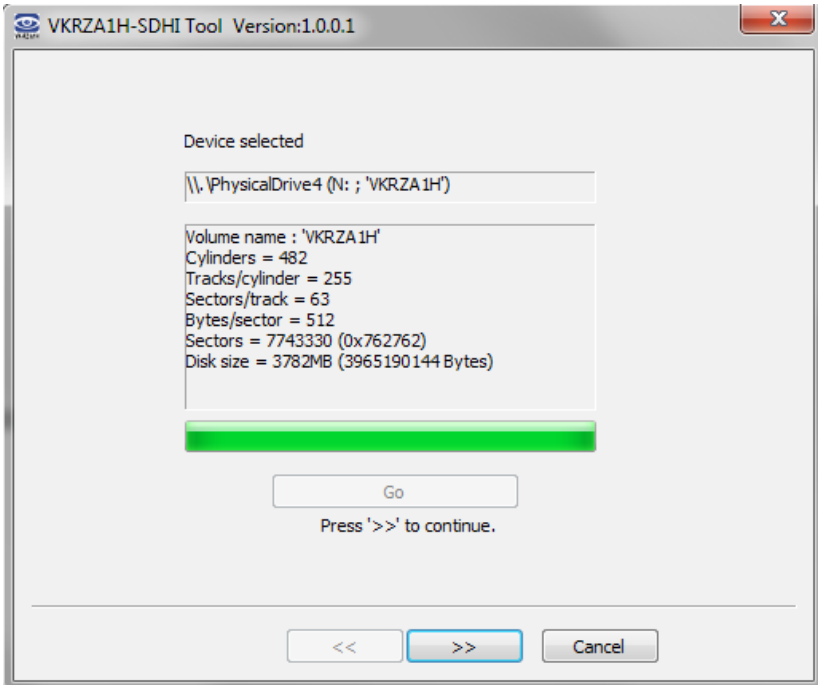
Click the '>>' button

3.2.5. Format drive



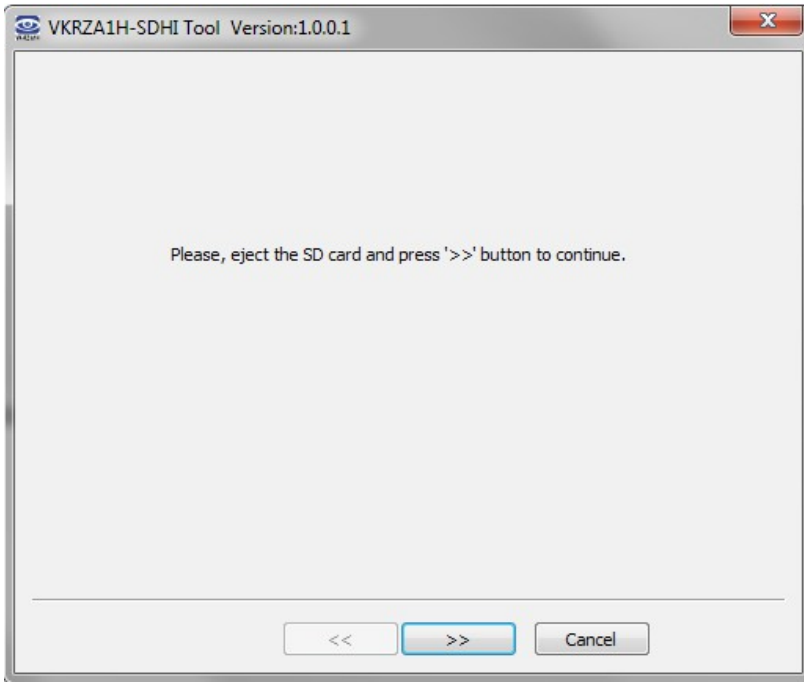
Click the 'Go' button

3.2.6. Drive format complete



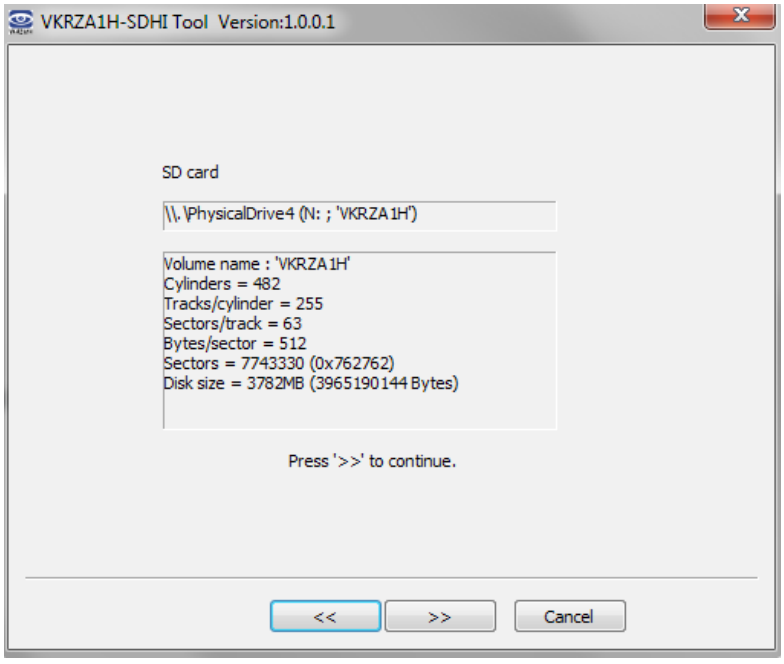
Click the '>>' button

3.2.7. Eject the SD card



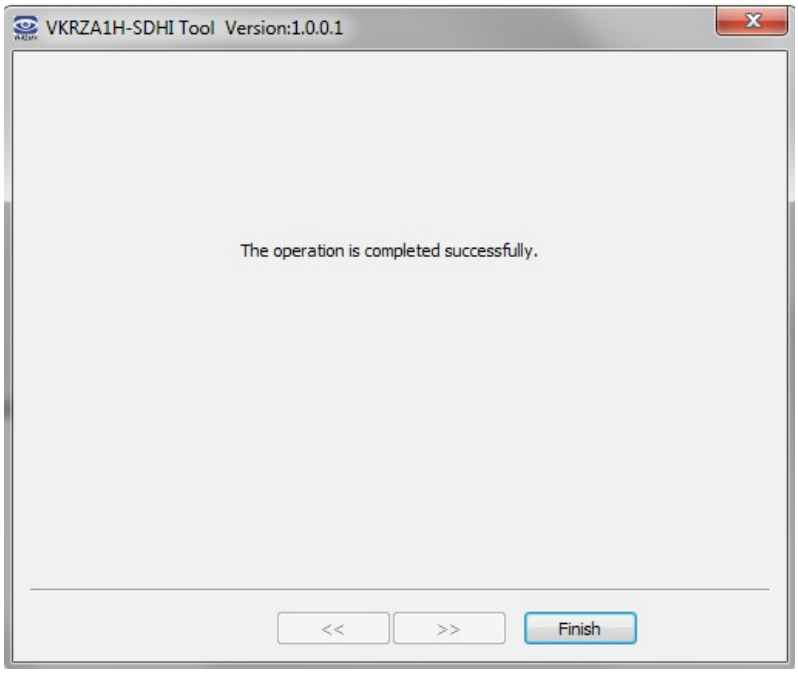
Click the '>>' button

3.2.8. Insert the SD card



Click the '>>' button

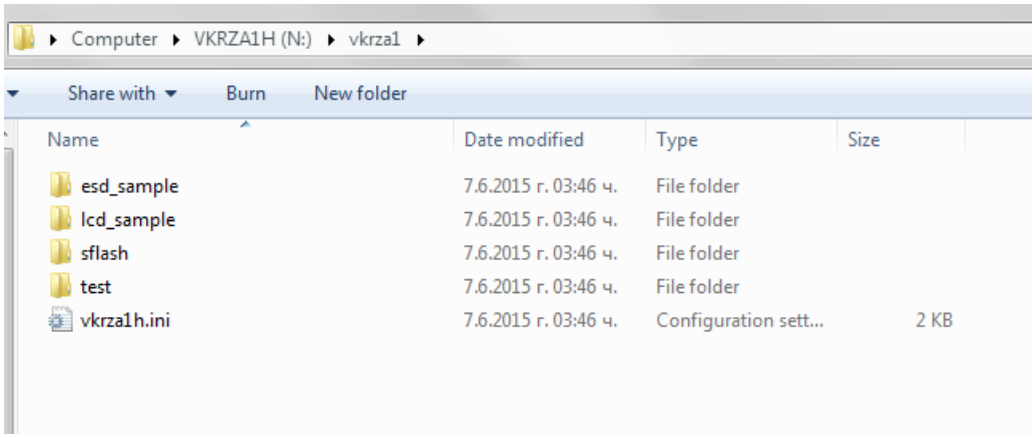
3.2.9. Format and Copy complete



Click the 'Finish' button

3.2.10. Insert the Micro SD card in VK-RZ/A1H

3.3. Micro SD content after preparation with vkrza1h_sdhi-1.0.0.1



The folders contain different demo projects that the user may start directly from the SD card.

The file vkrza1h.ini contains command data for the boot loader program.

If the user needs to start any project from any folder, he has to edit vkrza1h.ini.

3.4. Default content of vkrza1h.ini

#This line cannot be removed! by Vekatech Ltd.(http://www.vekatech.com)

```
[ToDo]
Action=Run
OS=UserSrecord

[SDRAM]
Banks=0
#M12L2561616A-6T
ChipSize=32MB
Refresh=yes
RefMode=Auto
BACTV=no
CASLat=2
WTRP=1
WTRCD=1
TRWL=2
WTRC=5
RefTmrCKS=phy/4
RefreshCount=1
RefTimeConst=128

[S-Flash]
#Auto

[UserBin]
# FileName                          Load        Mirror
#Image0=esd_sample\CONST_DL_TABLE         20100100
#Image1=esd_sample\RESET_HANDLER                200ab200
#Image2=esd_sample\VECTOR_MIRROR_TABLE    200ab100          20100000
#Image3=esd_sample\VECTOR_TABLE           200ab000          2002b000
#EntryPoint=                       200ab000
Image0=lcd_sample\vk-lcd43rta.bin              200a0000
EntryPoint=                       200a0000

[UserSrecord]
Image=test\web_server_demo.s37
EntryPoint=                       200A0000
#Image=test\sdram\async_test.s37
#EntryPoint=                       8000000

[UserSFlashBin]
ImagePath=sflash\u-boot
CommandLine=1 u-boot.bin                  18000000
#ImagePath=sflash\init
#CommandLine=1 VECTOR_TABLE 18000000 CODE_SPIBSC_INIT1 18000200
CODE_SPIBSC_INIT2_24 18000400 RESET_HANDLER 18004000
#ImagePath=sflash\cyclone
#CommandLine=2 web_server_demo.bin        18080000
```

After power up, if the boot jumpers are set to SD card and the card is in the holder, the VK-RZ/A1H's Boot loader is started. The vkrza1h.ini file, indicates to the boot loader program that the application file located at SDCARD\test\web_server_demo.s37 needs to be copied at address 200A0000 and executed

DETAILED DESCRIPTION OF VKRZA1H.INI FILE

4.1. Section TODO

```
*********************************************************************************
Section: [ToDo]
Main command of SD_bootloader
*********************************************************************************
=================================================================================
Parameter:     |  Option:                    |  Meaning:
=================================================================================
Action         |  Run                        |  Performs copying of images
               |  Load                       |  Performs programimg of images
---------------+-----------------------------+---------------------------
OS             |  UserBin        (Run Only)  |  The image for copying is in binary frmat
               |  UserSrecord    (Run Only)  |  The image for copying is in Motorola s37 format
               |  UserSFlashBin  (Load Only) |  The image for programming is in binary format
---------------------------------------------------------------------------------
```

Reads from the specified patch in the OS section.

Copies to hex address specified in the OS section.

Run

Example :

```
Action=Run
OS=UserSrecord
```

4.2. Section UserBin

```
*********************************************************************************
Sections: [UserBin]
"Run" procedure - details about the binary format images:
*********************************************************************************
=================================================================
Parameter:     |  Option:                  |  Meaning:
=================================================================
ImageN         |  Any valid SD card path   |  define the name of the image, its location,
               |                           |  the quantity of copies & its start addresses (Load, Mirror )
               |                           |  N specifies the order number of the image
---------------+---------------------------+---------------------------
EntryPoint=    |  Any valid 32bit address  |  Tells the SD_bootloader where to forward the control after the copying is completed
-----------------------------------------------------------------------------
```

This section is valid when OS=UserBin

Example :

```
Image0=lcd_sample\vk-lcd43rta.bin       200a0000
EntryPoint=200a0000
```

4.3. Section UserSrecord

```
*************************************************************************
Sections: [UserSrecord]
"Run" procedure - details about the Motorola s37 format images:
*************************************************************************
=================================================================
Parameter:     |  Option:                   |  Description:
=================================================================
Image          |  Any valid SD card path    |  Tells the SD_bootloader 2 things:  the name of the image, its location
---------------+----------------------------+---------------------------
EntryPoint=    |  Any valid 32bit address   |  Tells the SD_bootloader where to forward the control after the copying is completed
-------------------------------------------------------------------------
```

This section is valid when OS= UserSrecord

Example :

```
Image=test\web_server_demo.s37
EntryPoint=200A0000
```

S37 file holds the address for copy operation.

4.4. Section UserSFlashBin

```
*************************************************************************
Sections: [UserSFlashBin]
"Load" procedure - details about the binary format images:
*************************************************************************
=================================================================


Parameter:     |  Option:                   |  Meaning:
=================================================================
ImagePath=     |  Any valid SD card path    |  Tells location of the image
---------------+----------------------------+---------------------------
CommandLine=   |  1                         |  The image will be placed in the first SPI flash through 4bit interface
               |  2                         |  The image will be placed in both SPI flashes through 2x4bit interface
               |  image name                |  Specifies the name of the image
               |  start address             |  Specifies the start address for programming
               |  image name N              |  Specifies the name of the N image (if there are more than 1 image for programming)
               |  start address N           |  Specifies the start address of N programming (if there are more than 1 image for programming)
-------------------------------------------------------------------------
!NB! Please place your code after adress 18080000H.
In the address space 18000000H - 1807FFFFH there is a initialization code
responsible for proper init of the multi I/O SPI bus.
This code is activated when   BOOT_MODE 3 is selected. As default the initial setting of the bus is
like a normal SPI (1bit interface). After execution of that code the SPI bus is reinitialized in
(1x4bit or 2x4bit interface) depending of what argument (1 or 2) have been passed when the flash is programmed.
If you accidently wipe out that space, you can quickly restore it by uncommenting the following lines:

ImagePath=sflash\init
CommandLine=1 VECTOR_TABLE 18000000 CODE_SPIBSC_INIT1 18000200 CODE_SPIBSC_INIT2 18000400 RESET_HANDLER 18004000
```

This section is valid when OS= UserSFlashBin

Example how to program onboard flash with U-boot application:

```
ImagePath=sflash\u-boot
CommandLine=1 u-boot.bin 18000000
```

4.5. Section SDRAM

```
*************************************************************************
Section: [SDRAM]
This section specifies the configurations parameters for accessing  SDRAM
*************************************************************************
=========================================================================
Parameter:     |  Option:                          |  Description:
=========================================================================
Banks=         |  0                                |  No SDRAM is installed
               |  1                                |  1 SDRAM chip is isntalled
               |  2                                |  2 SDRAM chips are installed
---------------+-----------------------------------+---------------------------
ChipSize=      |  32MB                             |  Each chip is 32MB
               |  64MB                             |  Each chip is 64MB
---------------+-----------------------------------+---------------------------
Refresh=       |  yes, y, true                     |  Refresh operation of the SDRAM is performed.
               |  no, n, false                     |  No refresh operation of the SDRAM is performed.
---------------+-----------------------------------+---------------------------
RefMode=       |  auto  (only Refresh = yes)       |  Refresh Control -> Auto-refresh is performed
               |  self  (only Refresh = yes)       |  Refresh Control -> Self-refresh is performed
---------------+-----------------------------------+---------------------------
BACTV=         |  yes, y, true                     |  Bank Active Mode -> Bank active mode (using READ and WRIT commands)
               |  no, n, false                     |  Bank Active Mode ->  Auto-precharge mode (using READA and WRITA commands)
---------------+-----------------------------------+---------------------------
CASLat=        |  1                                |  CAS Latency 1 cycle
               |  2                                |  CAS Latency 2 cycle
               |  3                                |  CAS Latency 3 cycle
               |  4                                |  CAS Latency 4 cycle
---------------+-----------------------------------+---------------------------
WTRP=          |  0                                |  0 Auto-Precharge Completion Cycles
               |  1                                |  1 Auto-Precharge Completion Cycles
               |  2                                |  2 Auto-Precharge Completion Cycles
               |  3                                |  3 Auto-Precharge Completion Cycles
---------------+-----------------------------------+---------------------------
WTRCD=         |  0                                |  0 Wait Cycles between ACTV Command and READ(A)/WRIT(A) Command
               |  1                                |  1 Wait Cycles between ACTV Command and READ(A)/WRIT(A) Command
               |  2                                |  2 Wait Cycles between ACTV Command and READ(A)/WRIT(A) Command
               |  3                                |  3 Wait Cycles between ACTV Command and READ(A)/WRIT(A) Command
---------------+-----------------------------------+---------------------------
TRWL=          |  0                                |  0 Auto-Precharge Startup Wait Cycles
               |  1                                |  1 Auto-Precharge Startup Wait Cycles
               |  2                                |  2 Auto-Precharge Startup Wait Cycles
               |  3                                |  3 Auto-Precharge Startup Wait Cycles
---------------+-----------------------------------+---------------------------
WTRC=          |  2                                |  2 Idle Cycles from REF Command/Self-Refresh Release to ACTV/REF/MRS Command
               |  3                                |  3 Idle Cycles from REF Command/Self-Refresh Release to ACTV/REF/MRS Command
               |  5                                |  5 Idle Cycles from REF Command/Self-Refresh Release to ACTV/REF/MRS Command
               |  8                                |  8 Idle Cycles from REF Command/Self-Refresh Release to ACTV/REF/MRS Command
---------------+-----------------------------------+---------------------------
RefTmrCKS=     |  stop                             |  Stop the counting-up
               |  phy/4                            |  Clock input phy/4 to refresh timer counter (RTCNT).
               |  phy/16                           |  Clock input phy/16 to refresh timer counter (RTCNT).
               |  phy/64                           |  Clock input phy/64 to refresh timer counter (RTCNT).
               |  phy/256                          |  Clock input phy/256 to refresh timer counter (RTCNT).
               |  phy/1024                         |  Clock input phy/1024 to refresh timer counter (RTCNT).
               |  phy/2048                         |  Clock input phy/2048 to refresh timer counter (RTCNT).
               |  phy/4096                         |  Clock input phy/4096 to refresh timer counter (RTCNT).
---------------+-----------------------------------+---------------------------
RefreshCount=  |  1                                |  Refresh occurs at every refresh request (RTCNT = RTCOR)
               |  2                                |  Refresh occurs at every 2-nd refresh request (RTCNT = RTCOR)
               |  4                                |  Refresh occurs at every 4-th refresh request (RTCNT = RTCOR)
               |  6                                |  Refresh occurs at every 6-th refresh request (RTCNT = RTCOR)
               |  8                                |  Refresh occurs at every 8-th refresh request (RTCNT = RTCOR)
---------------+-----------------------------------+---------------------------
RefTimeConst=  |  0 - 255                          |  Refresh Time Constant Register (RTCOR)
-------------------------------------------------------------------------
```

This section is compulsory if some of the addresses are located in CS2, CS3 address space.

CS2 = U11

CS3 = U12