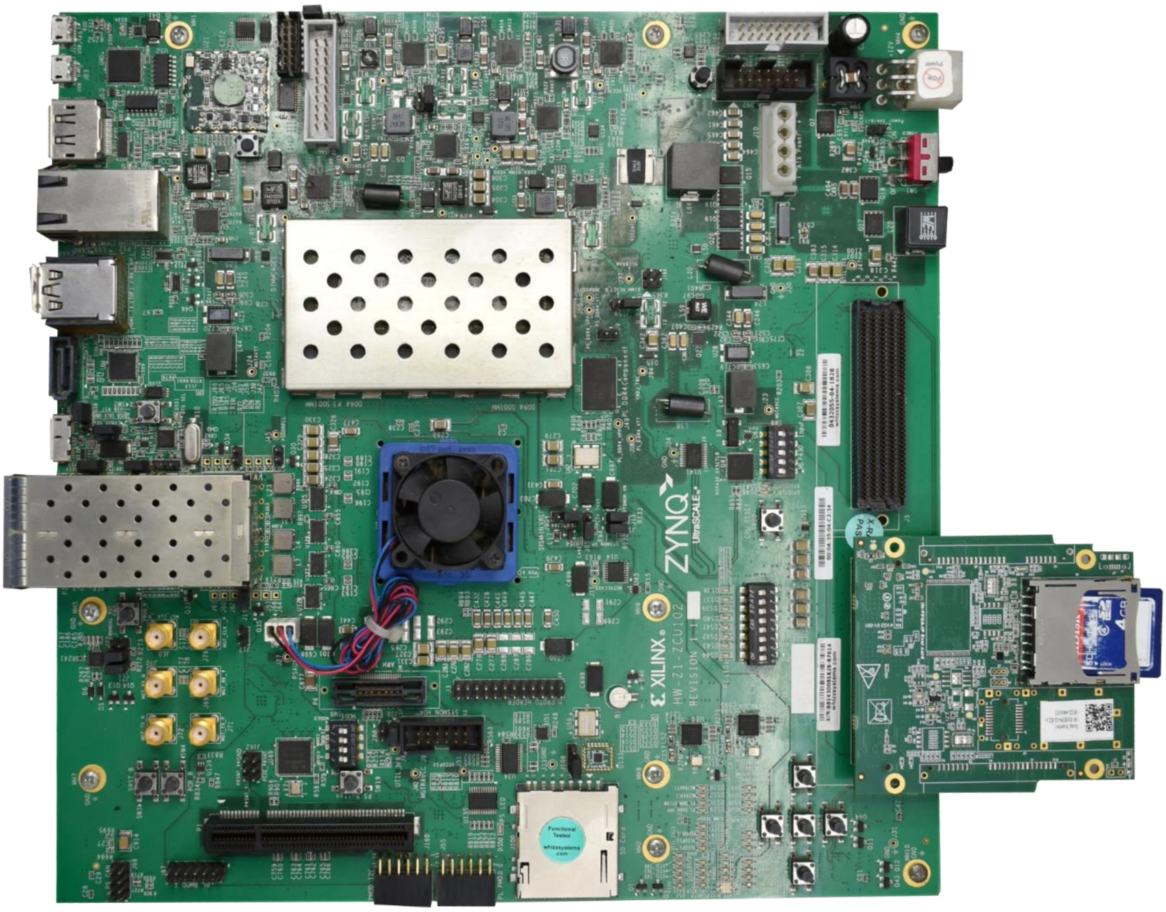
SD 3.0 Host

Software User Guide



iWave

iWave Systems Technologies Pvt. Ltd.

Disclaimer

iWave Systems reserves the right to change details in this publication including but not limited to any Product specification without notice.

No warranty of accuracy is given concerning the contents of the information contained in this publication. To the extent permitted by law no liability (including the liability to any person by reason of negligence) will be accepted by iWave Systems, its subsidiaries or employees for any direct or indirect loss or damage caused by omissions from or inaccuracies in this document.

CPU and other major components used in this product may have several silicon errata associated with it. Under no circumstances, iWave Systems shall be liable for the silicon errata and associated issues.

Trademarks

All registered trademarks, product names mentioned in this publication are the property of their respective owners and used for identification purposes only. **Certification** iWave Systems Technologies Pvt. Ltd. is an ISO 9001:2015 Certified Company.

Warranty & RMA

Warranty support for Hardware: 1 Year from iWave or iWave's EMS partner. For warranty terms, go through the below web link, http://www.iwavesystems.com/support/warranty.html For Return Merchandise Authorization (RMA), go through the below web link, http://www.iwavesystems.com/support/rma.html

Technical Support

iWave Systems technical support team is committed to provide the best possible support for our customers so that our Hardware and Software can be easily migrated and used.

For assistance, contact our Technical Support team at,

Email	: emfga@iwavesystems.com
Website	: www.iwavesystems.com
Address	: iWave Systems Technologies Pvt. Ltd. # 7/B, 29th Main, BTM Layout 2nd Stage, Bangalore, Karnataka, India – 560076

Table of Contents

List of Figures

List of Tables

1. INTRODUCTION

1.1 Purpose and Scope

The purpose of this document is to help the user to apply kernel patch file in PetaLinux to build the project for any platform and testing the SD Host Controller 3.0 features.

1.2 List of Acronyms

The following acronyms will be used throughout this document.

Acronyms	Abbreviations
SD	Secure Digital
UART	Universal Asynchronous Receiver/Transmitter

Table 1: Acronyms & Abbreviations

2. Procedure to apply kernel patch in PetaLinux

- Change into the directory of your PetaLinux Project

```
host@host~$ cd <path-to-Petalinux-Project>/Petalinux_project
```

- The Kernel patch from deliverables is located in the below path

  ./ iW-EMFGA-SC-01-R1.0-REL1.1/PATCH000-iW-EMFGA-SC-01-R1.0-REL1.1- Kernel_basic_customization.patch

- The Kernel configuration file from deliverables is located in the below path

  iW-EMFGA-DF-01-R1.0-REL1.0-SD_Deliverables/SourceCode/Kernel/iwave_sd.cfg

- Copy the PetaLinux patch file and configuration file, by executing the below command

```
host@host//Petalinux_project~$ cp <path_to_Patch>/PATCH000-iW-EMFGA-SC-01-R1.0-REL1.1-Kernel_basic_customization.patch project-spec/meta-user/recipes-kernel/linux/linux-xlnx/
host@host//Petalinux_project~$ cp <path_to_config_file>/ iwave_sd.cfg project-spec/metauser/recipes-kernel/linux/linux-xlnx/
```

- Add the below lines marked in red to bb file:

```
host@host//Petalinux_project~$ vim project-spec/meta-user/recipes-kernel/linux/linux-xlnx_%.bbappend
SRC_URI_append = " file://bsp.cfg \
                   file://iwave_sd.cfg \
                   file://PATCH000-iW-EMFGA-SC-01-R1.0-REL1.0-Kernel_basic_customization.patch \
                    "
```

  Note: Lines marked in blue are the lines present in demo.hpp file.

  Source code:

```
LINUX_VERSION = "5.10"

KBRANCH="xlnx_rebase_v5.10"
SRCREV = "568989d44176ae0a38ea78c16d0590c726d3b60a"
```

- Add the below lines marked in red to device-tree file:

```
host@host//Petalinux_project~$ vim project-spec/meta-user/recipes-bsp/device-tree/files/system-user.dtsi
&sd_host_controller_0 {
    compatible = "iw,usdhci-1.0";
};
```

- Now compile the PetaLinux project

3. BINARY PROGRAMMING

This section explains the procedure and detailed information about programming the binaries into boot device of ZCU102 platform. The programming steps are remains same for both pre-built binaries and user compiled binaries.

- The pre-built binaries are available in our deliverables in the below path.

  iW-EMFGA-BN-01-R1.0-REL1.1

3.1 Requirements

To program the binaries into ZCU102 platform, following items are required.

- SD card
- Host PC(Linux) for manual binary programming

3.2 Linux Binary Programming to SD card

Preparing the USB

- The prebuilt binaries are available in the deliverables in the below path.

  ```
  host@host:~<path_to_deliverables> $ cd iW-EMFGA-BN-01-R1.0-REL1.1/
  ```

- Refer section **SD card Partition** for partitioning SD card. If the partitions already exist, then remove all the folders/files present inside the SD partitions before copying the binaries.
- Copy the binaries from deliverables to SD partition.

  ```
  host@host:~<path_to_Binaries>/Binaries $ cp BOOT.bin image.ub boot.scr system.bit /media/<path-to-sd-bootpartition>
  ```

- Execute the below command to update the binary's to Ext4 partition.

  ```
  host@host:~/Binaries $ sudo tar -xvf rootfs.tar.gz -C /media/<path-to-sd-boot-ext4-partition>/
  host@host:~/Binaries $ sync
  ```

4. Test setup

4.1 Test requirements

- Xilinx ZCU102 dev kit
- iWave FMC daughter card
- UBUNTU PC
- USB cables for JTAG (optional) and UART (mandatory)
- SD card for testing and booting
- Power Supply 12 V,5A

NOTE: iWave FMC daughter card requires V_{adj} =1.8V for its operation and the V_{adj} value depends on the custom board.

4.2 Connections

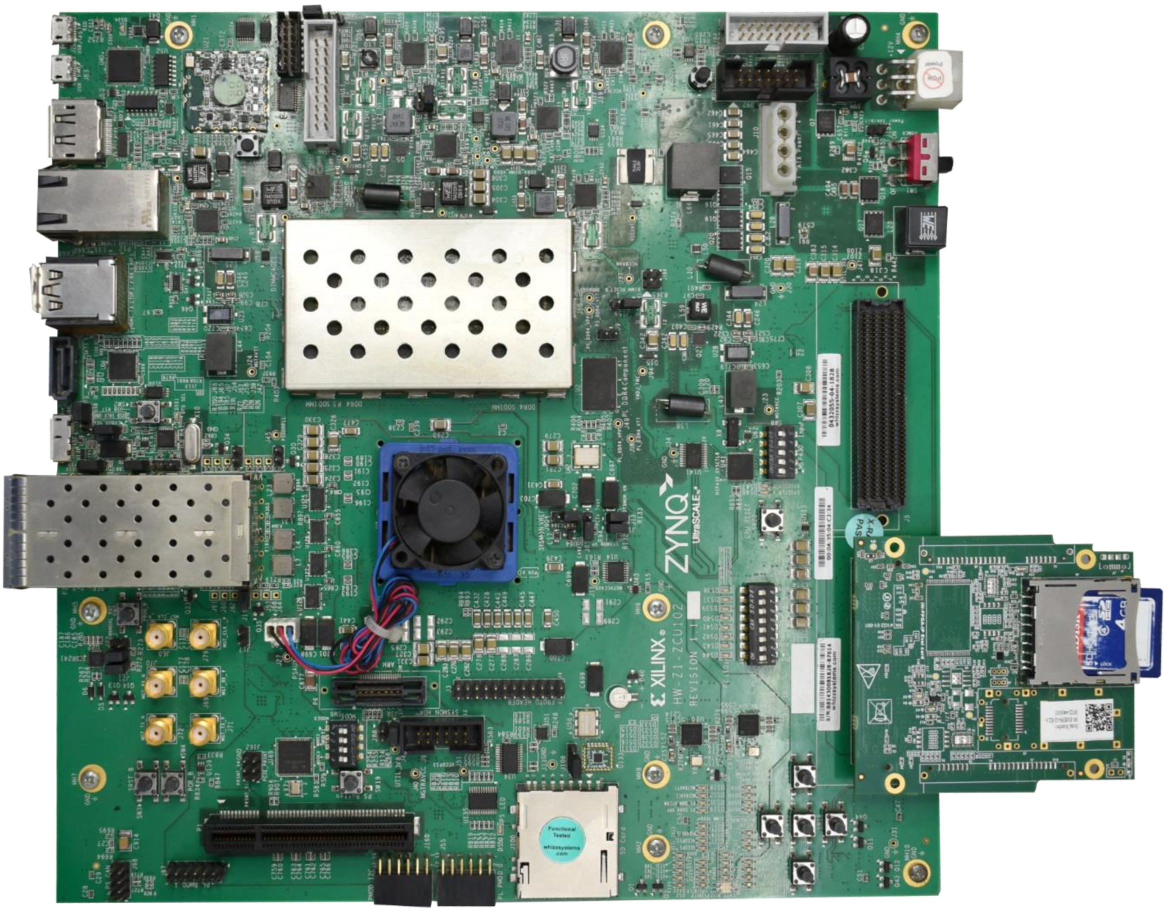


Figure 1:Test Setup of ZCU102 board with iW_FMC card

1. Connect the 12V,5A power supply
2. Please make sure that iWave FMC daughter card with SD connector interface is connected to HPC1 FMC of Xilinx ZCU102 board.
3. Power on the board
4. When powered on, you will see boot sequence messages on the Minicom/TeraTerm console as shown in figure below. Observe SD card specific messages of detection and size.

5. Make sure that iWave FMC daughter card is set to 1.8V and 2 Blue LED in FMC card is glowing as shown in the Figure Below

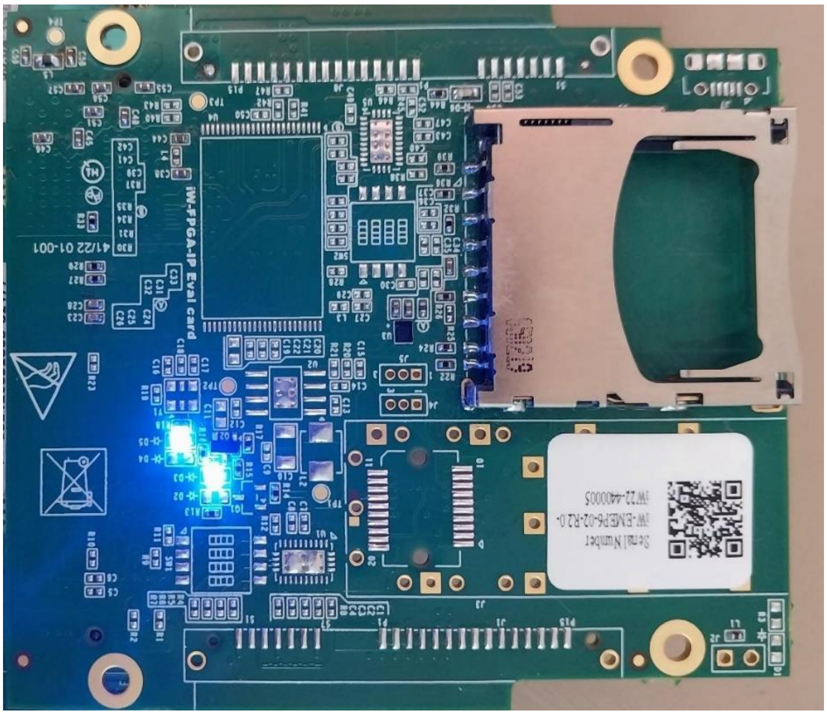


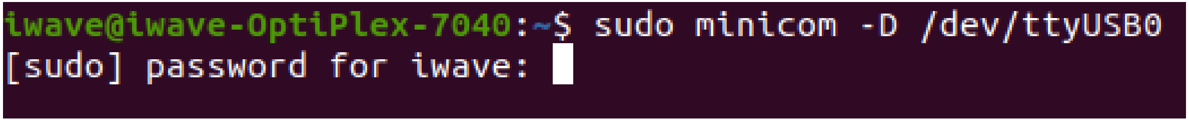
Figure 2: iWave FMC daughter card

- Note: The iWave FMC daughter card cannot be hot-plugged.

4.3 Minicom/Tera Term configuration

- Connect a serial cable between the UART connector on board and PC.
- Run Minicom application on the PC and configure it with the following parameters.

*COM number may change



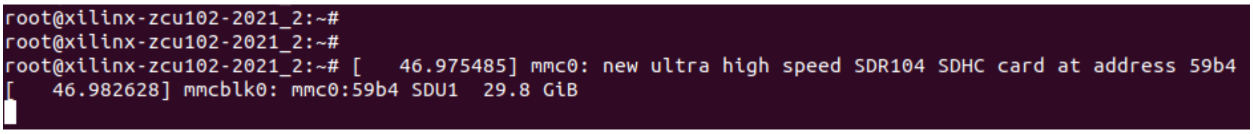
```
iwave@iwave-OptiPlex-7040:~$ sudo minicom -D /dev/ttyUSB0
[sudo] password for iwave: 
```

Figure 3: Minicom setting

5. SD Testing

5.1 Card Detection

- Insert SD 3.0 card in J1 connector
- Once the card is detected below messages appears on the Linux console (HyperTerminal/ /Minicom/TeraTerm)



```
root@xilinx-zcu102-2021_2:~#
root@xilinx-zcu102-2021_2:~#
root@xilinx-zcu102-2021_2:~# [   46.975485] mmc0: new ultra high speed SDR104 SDHC card at address 59b4
[   46.982628] mmcblk0: mmc0:59b4 SDU1  29.8 GiB
```

Figure 4: Card Detection

5.2 File write and read by using the Linux cp command.

- Mount the card using the mount command as shown below. Please make sure the SD card is partitioned before mounting. Check APPENDIX chapter for SD card Partition

  mount /dev/mmcblk0p1 /mnt/

  Write:

  cp /home/root/sd.txt /mnt

  Read:

  cp/mnt/sd.txt /home/root/sd1.txt

  Note: sd.txt and sd1.txt are just example files. Any files can be copied using cp command.
- Check whether the file differs.

  diff /home/root/sd.txt /home/root/sd1.txt

  command should execute without any error messages.

5.3 File write and read by using the Linux dd command.

- Use **dd** command to read/write a file from card.

  dd if=/dev/urandom of=/tmp/data bs=1M count=10

  dd if=/tmp/data of=/dev/mmcblk0(p1) bs=1M count=10

  dd if=/dev/mmcblk0(p1) of=/tmp/data1 bs=1M count=10
- Use **md5sum** command to verify that both files match.

  md5sum /tmp/data /tmp/data1

  sha values reported by md5sum should be equal for data and data1 files

6. APPENDIX

6.1 SD card Partition

This section describes the steps to partition the SD card to program the binaries.

- Connect the new SD to the Linux Host system using SD Card Reader.
- Execute the mount command to see the attached nodes and mount points.
- SD may attach to dev nodes either sdb/sdc/sdd in Host PC. Assume the SD is attached to /dev/sdb node.

```
$ umount /dev/sdb
```

- Start partitioning using fdisk command.

```
$ sudo fdisk /dev/sdb
```

- After running fdisk, it will change shell prompt to.

```
Command (m for help):
```

- Press 'p' to view already existing partitions.

```
Disk /dev/sdb: 14.6 GiB, 15664676864 bytes, 30595072 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disklabel type: dos
Disk identifier: 0xe937a7bf
Device    Boot Start    End Sectors Size Id Type
/dev/sdb1        32 30595071 30595040 14.6G c W95 FAT32 (LBA)
```

- Delete all existing partitions using command ‘d’. Enter individual partitions like 1, 2, 3, etc. until all the partitions are deleted.
- Once all the partitions are deleted, the below message gets displayed.

```
Command (m for help):
Selected partition 1
Partition 1 has been deleted.
Command (m for help): d
No partition is defined yet!
```

- Press 'n' to create new partition (going to create first partition).

```
Command (m for help): n Partition
type
  p   primary (0 primary, 0 extended, 4 free)
e   extended (container for logical partitions)
```

```
Select (default p): p
Partition number (1-4, default 1):
First sector (2048-30595071, default 2048):
Last sector, +sectors or +size {K, M, G, T, P} (2048-30595071, default 30595071): 6293504 Created
a new partition 1 of type 'Linux' and of size 3 GiB.
```

- Now, set the partition IDs

```
Command (m for help): t
Selected partition 1
Hex code (type L to list all codes): b
Changed type of partition 'Linux' to 'W95 FAT32'.
```

- Check the new partition table and write the changes.

```
Command (m for help): p
Disk /dev/sdb: 14.6 GiB, 15664676864 bytes, 30595072 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disklabel type: dos
Disk identifier: 0xe937a7bf
Device     Boot Start    End Sectors Size Id Type
/dev/sdb1       2048 6293504 6291457   3G b W95 FAT32
```

- Now the partitions are created as above. Save these changes by pressing ‘w’.

```
Command (m for help): w
The partition table has been altered.
Calling ioctl () to re-read partition table.
Syncing disks.
```

- Format both the partitions, first partition as VFAT (windows) and second partition as EXT4 (Linux).

```
$ sudo mkfs.vfat /dev/sdb1
```

- Now SD card is ready to use.
- Remove the SD card and insert again, then the respective partitions can be viewed by the below command. `$ mount`