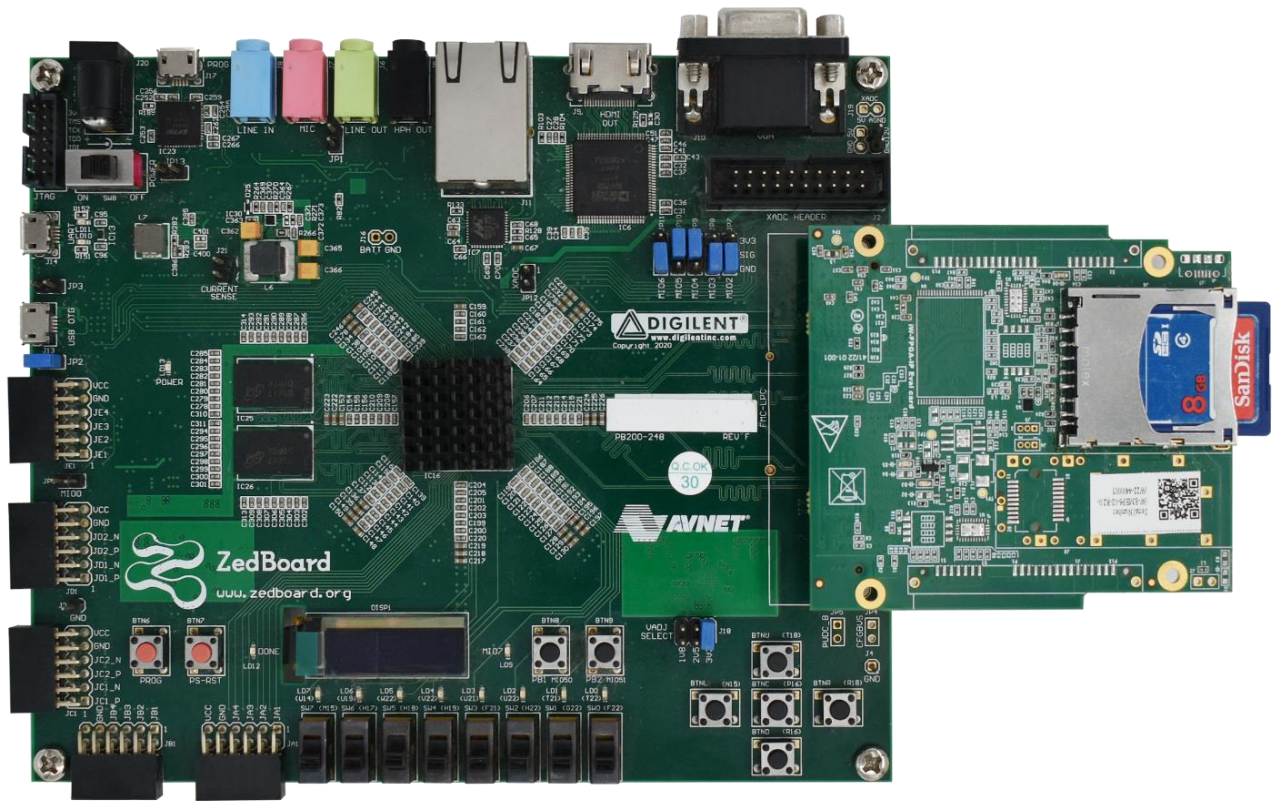


# SD 3.0 Host Software User Guide



**iWave**

## 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 : [ascdi@iwavesystems.com](mailto:ascdi@iwavesystems.com)

Website : [www.iwavesystems.com](http://www.iwavesystems.com)

Address : iWave Systems Technologies Pvt. Ltd.  
# 7/B, 29th Main, BTM Layout 2nd Stage,  
Bangalore, Karnataka, India – 560076

## Table of Contents

<b>1. INTRODUCTION</b> .....	<b>4</b>
<b>1.1 Purpose and Scope</b> .....	<b>4</b>
<b>1.2 List of Acronyms</b> .....	<b>4</b>
<b>2. BSP PETALINUX COMPILATION</b> .....	<b>5</b>
<b>2.1 Host PC Setup</b> .....	<b>5</b>
<b>2.1.1 Host Requirements</b> .....	<b>5</b>
<b>2.1.2 Host Package Installation</b> .....	<b>5</b>
<b>2.1.3 Petalinux Tool Installation</b> .....	<b>6</b>
<b>2.1.4 Petalinux Build</b> .....	<b>6</b>
<b>3. BINARY PROGRAMMING</b> .....	<b>9</b>
<b>3.1 Manual Programming</b> .....	<b>9</b>
<b>3.1.1 Requirements</b> .....	<b>9</b>
<b>3.1.2 Linux Binary Programming to SD card</b> .....	<b>9</b>
<b>4. APPENDIX</b> .....	<b>10</b>
<b>4.1 SD card Partition</b> .....	<b>10</b>

## List of Figures

Figure 1: Petalinux system configuration.....	7
---	---

## List of Tables

Table 1: Acronyms & Abbreviations.....	4
--	---

## 1. INTRODUCTION

### 1.1 Purpose and Scope

The purpose of this document is to help the user to build the software binaries for testing the SD Host Controller 3.0 features on the ZedBoard platform. The document describes the installation of Xilinx tools and compilation of Petalinux Project.

### 1.2 List of Acronyms

The following acronyms will be used throughout this document.

**Table 1: Acronyms & Abbreviations**

Acronyms	Abbreviations
SD	Secure Digital
UART	Universal Asynchronous Receiver/Transmitter

## 2. BSP Petalinux Compilation

### 2.1 Host PC Setup

This section explains procedure and detailed information about compiling PetaLinux for ZedBoard platform.

#### 2.1.1 Host Requirements

- 32 GB RAM.
- 2 GHz CPU clock or equivalent.
- 800 GB free HDD space.
- Ubuntu 18.04.
- Root permission on the Development Host. The Petalinux tools need to be installed as a non-root user.

#### 2.1.2 Host Package Installation

Xilinx packages requires a number of standard development tools and libraries to be installed on your Linux host PC.

- Open a terminal window and install the below packages in Host System

```
$ sudo apt-get update
```

```
$ sudo apt-get install iproute2 tftpd net-tools flex
```

```
$ sudo apt-get install git bison libselinux1 zlib1g-dev
```

```
$ sudo apt-get install gnupg pax gawk python3
```

```
$ sudo apt-get install diffstat python3-pexpect python3-pip
```

```
$ sudo apt-get install python3-git python3-jinja2
```

```
$ sudo apt-get install gcc tar gzip cvs subversion
```

```
$ sudo apt-get install cpio vim git-core coreutils unzip
```

```
$ sudo apt-get install texinfo zlib1g:i386 make screen
```

```
$ sudo apt-get install chrpath libgl1-mesa-dev g++
```

```
$ sudo apt-get install wget python-pysqlite2 autoconf
```

```
$ sudo apt-get install xz-utils debianutils iputils-ping
```

```
$ sudo apt-get install build-essential socat tftpd
```

```
$ sudo apt-get install automake libtool xterm
```

```
$ sudo apt-get install libglu1-mesa-dev mercurial libssl-dev
```

```
$ sudo apt-get install gcc-multilib libstd1.2-dev libglib2.0-dev libncurses5-dev dkms
```

### 2.1.3 Petalinux Tool Installation

- Download Petalinux 2021.2 Tool Installer from the below link

<https://www.xilinx.com/member/forms/download/xef.html?filename=petalinux-v2021.2-final-installer.run>

- Petalinux Tools installation is straight-forward. Without any options, the Petalinux Tools are installed into the current working directory. Alternatively, an installation path may be specified

- Once the tool is downloaded, Navigate to the Downloads folder.

```
cd /<path_to_downloads>/Downloads
```

**Example**

```
cd /home/iwave/Downloads
```

- Copy the installer file from downloads folder to petalinux installation directory.

```
host@host:~/<path_to_downloads>/Downloads ~$ cp petalinux-v2021.2-final-installer.run /<target_directory>/
```

**Example**

```
iwave@iwave:~/Downloads ~$ cp petalinux-v2021.2-final-installer.run /home/iwave/tools/petalinux/
```

- Navigate to petalinux installation directory.

```
host@host:~/<path_to_downloads>/Downloads ~$ cd /<target_directory>/
```

**Example**

```
iwave@iwave:~/Downloads ~$ cd /home/iwave/tools/petalinux/
```

- Execute the below commands to install the petalinux tool.

```
host@host:~/<target_directory> ~$ chmod 755 petalinux-v2021.2-final-installer.run  
host@host:~/<target_directory> ~$ ./petalinux-v2021.2-final-installer.run
```

**Example**

```
iwave@iwave:~/tools/petalinux ~$ chmod 755 petalinux-v2021.2-final-installer.run  
iwave@iwave:~/tools/petalinux ~$ ./petalinux-v2021.2-final-installer.run
```

- The above command installs the Petalinux Tool into the current working directory
- Read and agree to the Petalinux End User License Agreement to install Petalinux

### 2.1.4 Petalinux Build

- To setup the Petalinux working environment, source the appropriate settings script using below command.

```
host@host~$ source <path-to-installed-petalinux>/settings.sh
```

```
host@host~$ export LANG=en_US.UTF-8
```

- Download the BSP file for the ZedBoard from the below link

<https://www.xilinx.com/member/forms/download/xef.html?filename=avnet-digilent-zedboard-v2021.2-final.bsp>

- Once the BSP is downloaded, execute the below command to create a Petalinux project.

## SD 3.0 Host Software User Guide

```
host@host $ petalinux-create -t project -s avnet-diligent-zedboard-v2021.2-final.bsp
```

- Change into the directory of your PetaLinux Project.

```
host@host~$ cd <path-to- avnet-diligent-zedboard-2021.2>/ avnet-diligent-zedboard-2021.2
```

- The PetaLinux project-spec patch from deliverables is located in the below path.

```
iW-ASCDI-DF-02-R1.0-REL1.0-SD_Deliverables/SourceCode/Petalinux/PATCH000-iW-ASCDI-SC-02-R1.0-REL1.0-SD_3.0_Host_Petalinux_basic_customization.patch
```

- Copy the PetaLinux patch file, by executing the below command.

```
host@host/<Directory>/avnet-diligent-zedboard-2021.2~$ cp <path_to_Patch>/PATCH000-iW-ASCDI-SC-02-R1.0-REL1.0-SD_3.0_Host_Petalinux_basic_customization.patch ../
```

- To apply the patch file, execute the below command.

```
host@host/<Directory>/avnet-diligent-zedboard-2021.2~$ patch -Np1 < ../PATCH000-iW-ASCDI-SC-02-R1.0-REL1.0-SD_3.0_Host_Petalinux_basic_customization.patch
```

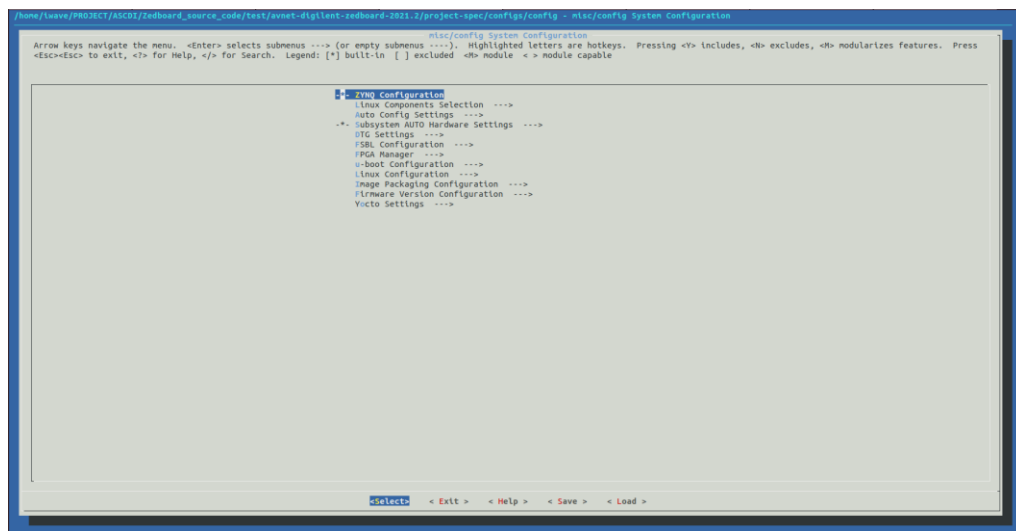
- The FPGA XSA file from deliverables is located in the below path.

```
iW-ASCDI-DF-02-R1.0-REL1.0-SD_Deliverables/SourceCode/FPGA/design_1_wrapper.xsa
```

- Import the hardware description with petalinux-config command, by giving the path of the directory containing design\_1\_wrapper.xsa file as follows:

```
host@host/ avnet-diligent-zedboard-2021.2$ petalinux-config --get-hw-description=<path-to-parent-directory-of-xsa-file>
```

- The above command launches the top system configuration menu as shown below;



**Figure 1: Petalinux system configuration**

- Click on save and exit to save the configuration.
- To build the system image, execute the below command.

```
host@host/<Directory>/ avnet-diligent-zedboard-2021.2~$ petalinux-build
```

- After the successful compilation the binaries will be placed in below path.

```
~/<path to avnet-diligent-zedboard-2021.2>/avnet-diligent-zedboard-2021.2/images/linux/  
The binary files are listed below
```

## SD 3.0 Host Software User Guide

*zynq\_fsbl.elf*

*u-boot.elf*

*system.bit*

*image.ub*

- To create BOOT.bin, execute the below command. The BOOT.bin will be created in images/linux folder.

```
host@host/<Directory>/avnet-digilent-zedboard-2021.2~$petalinux-package --boot --format BIN --fsbl  
images/linux/zynq_fsbl.elf --u-boot images/linux/u-boot.elf -o images/linux/BOOT.bin --force
```

- Refer the “**BINARY PROGRAMMING**” section to update the Linux kernel binary.



## 3. BINARY PROGRAMMING

This section explains the procedure and detailed information about programming the binaries into boot device of ZedBoard 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-ASCDI-DF-02-R1.0-REL1.0-SD\\_Deliverables/Binaries](#)

### 3.1 Manual Programming

This section explains the step-by-step procedure to program the BOOT.bin, bitstream, kernel image and Linux binaries into ZedBoard platform using a SD card.

#### 3.1.1 Requirements

To program the binaries into Corazon-AI platform, following items are required.

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

#### 3.1.2 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-ASCDI-DF-02-R1.0-REL1.0-SD\_Deliverables/Binaries`

- The compiled binaries are available in the below path.

`host@host:~$ cd /<target_directory>/<path to avnet-digilent-zedboard-2021.2>/avnet-digilent-zedboard-2021.2/images/linux/`

- 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-boot-partition>`

**Note:** For SD 3.0 Host Feature testing refer the “[iW-ASCDI-TS-02-R1.0-REL1.2.pdf](#)” document.

## 4. APPENDIX

### 4.1 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
```

## SD 3.0 Host Software User Guide

*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*