Product Documentation
DAT096-M1

Siddharthan Azhaguvel, Amjad Shujaa Aldeen, Boyue Sun, Zhili Xia

May 2025

1 Introduction

The purpose of this project is to develop a functional User Equipment (UE) prototype capable of interfacing with a 6G testbed. The system facilitates signal transmission and reception between MATLAB and hardware peripherals using the Digilent Eclypse Z7 development board, along with the associated Zmod modules.

The design integrates both hardware and software components to support real-time signal processing and communication via Ethernet. This documentation provides a comprehensive guide for developers and researchers, outlining the hardware setup, software configuration, development environment, and debugging procedures. It aims to minimize onboarding time and ensure smooth integration for future enhancements.

Key functionalities of the system include:

- Transmitting user-defined signals from MATLAB to an oscilloscope through the Zmod AWG.
- Receiving signals in MATLAB from an external waveform generator connected via the Zmod Scope.

By following the instructions outlined in this document, users can efficiently configure the complete system and begin experimenting with signal generation, acquisition, and processing in a real-time environment.

2 Hardware Setup

This section lists the hardware tools needed to realize the product, along with the connection setup.

2.1 Required Tools

- Digilent Eclypse Z7
- Digilent Zmod Scope and Zmod AWG
- Digilent Analog Discovery 3
- Cables:
  - Power adapter for the Eclypse Z7 board

- MicroUSB cable
- Ethernet cable
- USB-C cable
- BNC-to-SMA cables
- Power cable

2.2 Connection Setup

1. Connect the **Zmod Scope** and **Zmod AWG** to the **Zmod A** and **Zmod B** ports of the Eclypse Z7, respectively as shown in the Figure 1.

   Important: This order is required as the synthesized hardware design assumes this configuration. Ensure the Zmod boards are connected only when the Eclypse Z7 is **powered off**.

2. Connect the power adapter to the Eclypse Z7 and plug it into a power socket.

3. Connect the MicroUSB cable from the **PROG port** on the Eclypse Z7 to your PC. This enables the UART interface for programming the board and printing debug messages.

4. Connect the Ethernet cable between the Eclypse Z7 and your PC (or network switch) to enable data transfer with MATLAB.

5. Connect the Analog Discovery 3 Adapter to the **J2 port** on the Discovery BNC as shown in the Figure 2.

6. Connect the BNC-to-SMA cables as follows:

   - `W1` on the Discovery BNC → `CH1` on the Zmod Scope
   - `W2` on the Discovery BNC → `CH2` on the Zmod Scope
   - `CH1` on the Discovery BNC → `CH1` on the Zmod AWG
   - `CH2` on the Discovery BNC → `CH2` on the Zmod AWG

7. Connect the USB-C cable between the Analog Discovery 3 and the host PC.

8. turn on the power switch on the board.

Now the whole setup is complete as shown in Figure 3. After that, you can proceed with the software setup in the next section.

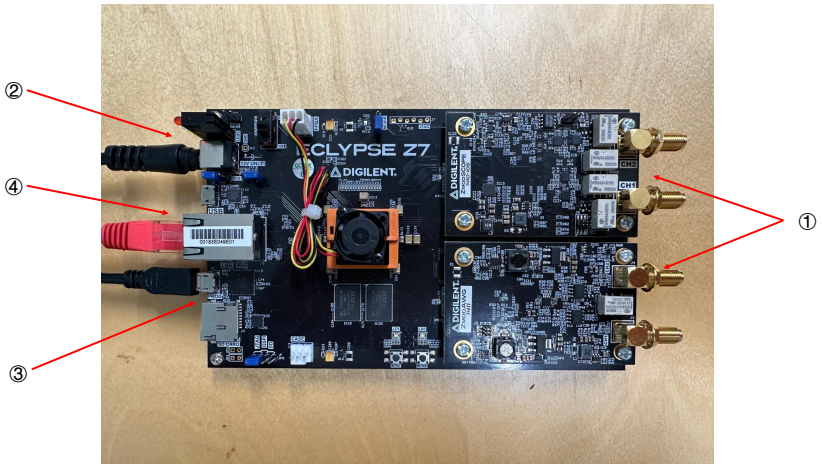

Figure 1: Ports of Eclypse Z7

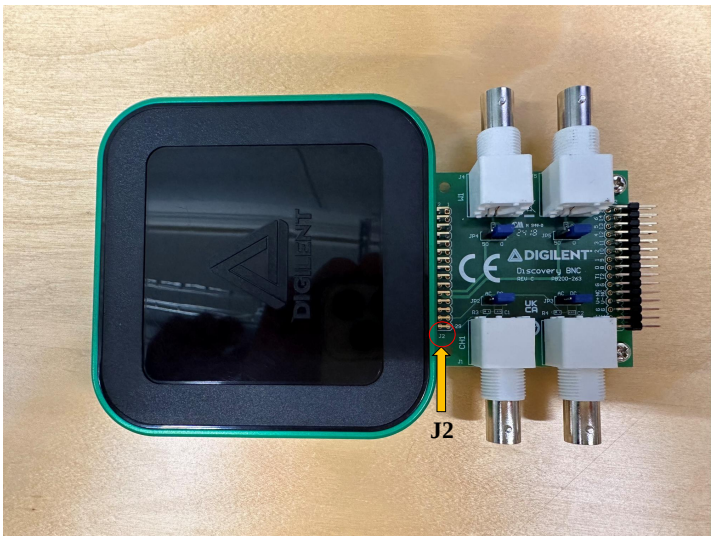

Figure 2: Overview of the Analog Discovery hardware.

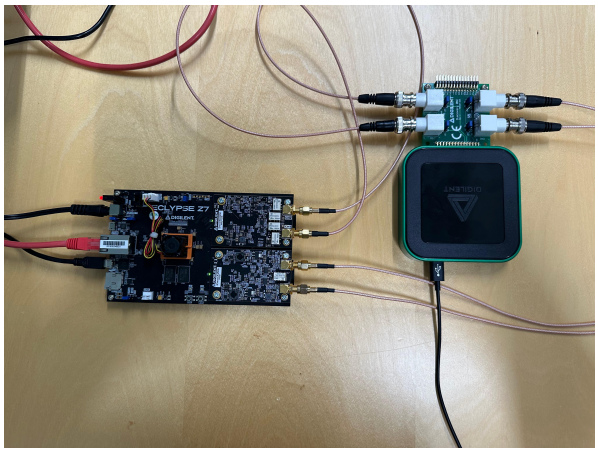

Figure 3: Overview of the Analog Discovery hardware.

3 Software Configuration

This section describes the software required to realize the working of the project and the steps needed to set up the working environment. The software tools are listed below:

- MATLAB
- Vitis Classic 2024.1
- Waverforms application from Digilent
- Any serial terminal

To set up the working environment, the following steps are needed after the hardware is set up:

1. Open Vitis Classic 2024.1 and import the project archive mentioned in Section 6.
2. Open MATLAB and open the MATLAB script referenced in Section 6.
3. Open Waveforms with the workspace archive mentioned in Section 6. The Scope tab is used to view the outputs from the Zmod AWG, while the Wavegen tab is used to generate waveforms that are fed to the Zmod Scope.
4. Open any serial terminal and connect to the correct COM port with the following settings:
    - Baud rate: 115200
    - Flow Control: None
    - Parity: None
5. Program the board with the software by running the application project as shown in Figure 4. Once the board is correctly programmed, prints should appear on the serial terminal that shows a successful TCP connection. In case Vitis is being run from a remote server, follow the steps mentioned in Appendix A.
6. Run the MATLAB script. Once it connects to the board via Ethernet, the menu shown in Figure 5 should be visible in the command window of MATLAB.

7. In case you select the option to transmit a signal, make sure to run the Scope in Waveforms to view the transmitted signal. In case you want to receive a signal, make sure to run the Wavegen in Waveforms. It is up to the user to select the type of waveform to run and the channels to run.

8. There is no option to stop reception in the MATLAB menu. This is by design since a timeout will occur automatically if the board does not receive a signal within 10 seconds.

9. It is necessary to exit the program in MATLAB by pressing the appropriate key as listed in the menu to view the variables.

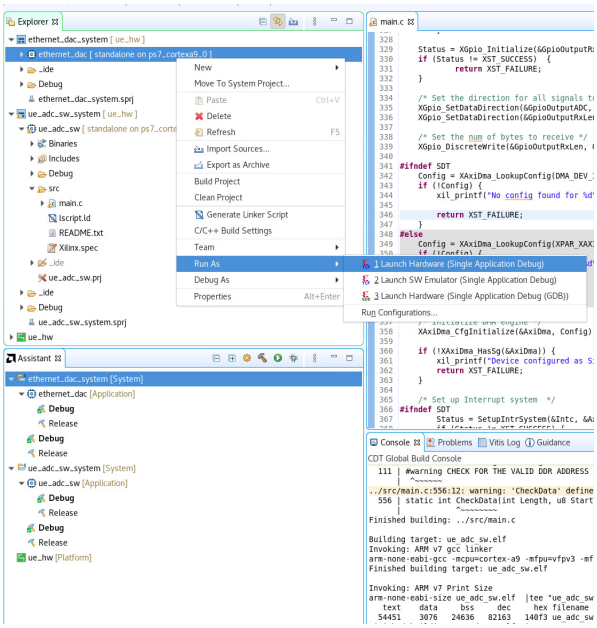

Figure 4: Running the application project in Vitis

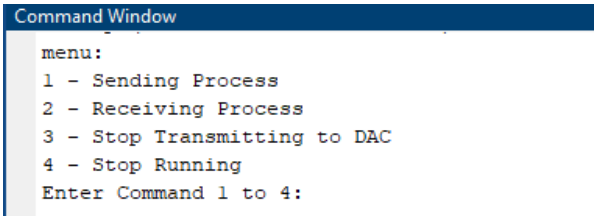

Figure 5: MATLAB application menu

4 Development environment

This section describes the development environment for developing the hardware or software needed for the project.

4.1 Hardware development

Vivado 2024.1 is used for the development. The archive project provided in Section 6 runs only on this version of Vivado. The development was carried out using a mix of IP libraries and self-developed HDL modules. The connections and settings of different IP blocks were carried out using block design on Vivado. The PL clock is sourced from the system clock of the PS, which runs at 100 MHz. The sampling clocks of the Zmods are also fixed at 100 MHz using a clock wizard IP. Refer to the Zmod controller IP user guide in Section 7 to understand the Zmod controller clocking requirements. ILAs have been provided to debug key signals, especially AXI stream signals.

Validate the block design and generate the bitstream. The hardware specification can then be exported, which can then be imported into Vitis to support the software development. Ensure there are no timing violations in the implemented design before exporting the hardware.

Apart from the main Vivado project, a Vivado project with a standalone pilot sequence detector and its testbench has also been provided in section 6. This can be used to functionally test the pilot sequence detector on its own without the rest of the design.

4.2 Software development

Vitis Classic 2024.1 was used for the development. Note that the provided archive project cannot be opened using another version or the new Vitis Unified IDE. The classic Vitis IDE encapsulates a project into a system project, an application project, and a platform project. The platform project contains the hardware specification that was exported from Vivado. Application projects are the actual software applications that run on the board. Any application project is always part of a system project. In the provided Vitis project archive, there are two system projects, each with a single application project. The system projects and their uses are listed below:

1. **ethernet_dac_system** contains the main application that can be used for the working environment described in section 3. You can transmit and receive using this project.
2. **ue_adc_sw_system** contains a standalone application for only the sample reception. This does not require the use of MATLAB or an Ethernet connection. This is used to loop back the received waveform in the Zmod Scope to the Zmod AWG, and can be viewed in the Waveforms application to ensure correct reception of data.

Ensure to **build the application project** whenever updating the software before running it. Also, ensure to update the hardware specification in the platform project and build it when a new hardware specification is exported from Vivado. Ensure that you use the same hardware specification path when exporting from Vivado and when updating the specification in Vitis.

5 Debugging and error recovery

It is possible to encounter issues while working with the project. Common issues and their recoveries are given below.

1. **MATLAB shows error stating no Ethernet connection detected**
   Ensure that the application project is running on the board **before** the MATLAB script is run. Ensure that the Ethernet cable is connected. Check that the IP address in the MATLAB script matches the one in the application project in Vitis. This can be checked in main.c in the ethernet_dac application project.

2. **Certain order of commands issued in the menu in MATLAB results in an error in MATLAB**
   Run the MATLAB script again. If an Ethernet connection is not detected, program the board once more.

3. **The plot of the received signal does not match the input signal to the Zmod Scope**
   This is a known issue. Our interpretation of the ADC 14-bit value does not match the observation. The ue_adc_sw application project can be used instead to view the waveform directly on the Waveforms Scope

6 List of program files

This section lists the development and working program files needed by various software tools to run the project.

1. **Vitis project**
2. **Main application Vivado project**
3. **Pliot sequence detector Vivado project**
4. **MATLAB script**
5. **Waveforms workspace template**

The above files are available in the following GitHub repository: `https://github.com/SidTesla/DAT096-M1.git`.

7 Reference documentation

This section lists the necessary reference documentation needed to continue development of the project as clickable links.

1. **Zynq 7000 SoC technical reference manual**
2. **Zmod Scope controller IP user guide**
3. **Zmod AWG controller IP user guide**
4. **AXI DMA IP product guide**
5. **AXI4-Stream infrastructure IP Suite product guide**

A Software setup when running Vitis on a remote server

1. In Vitis (after exporting the .xsa file from Vivado and creating the platform and application component, of course), make sure that the BSP settings for the standalone configuration for the OS are set to ps7_uart_0 for the stdin and stdout. See Figure 6.

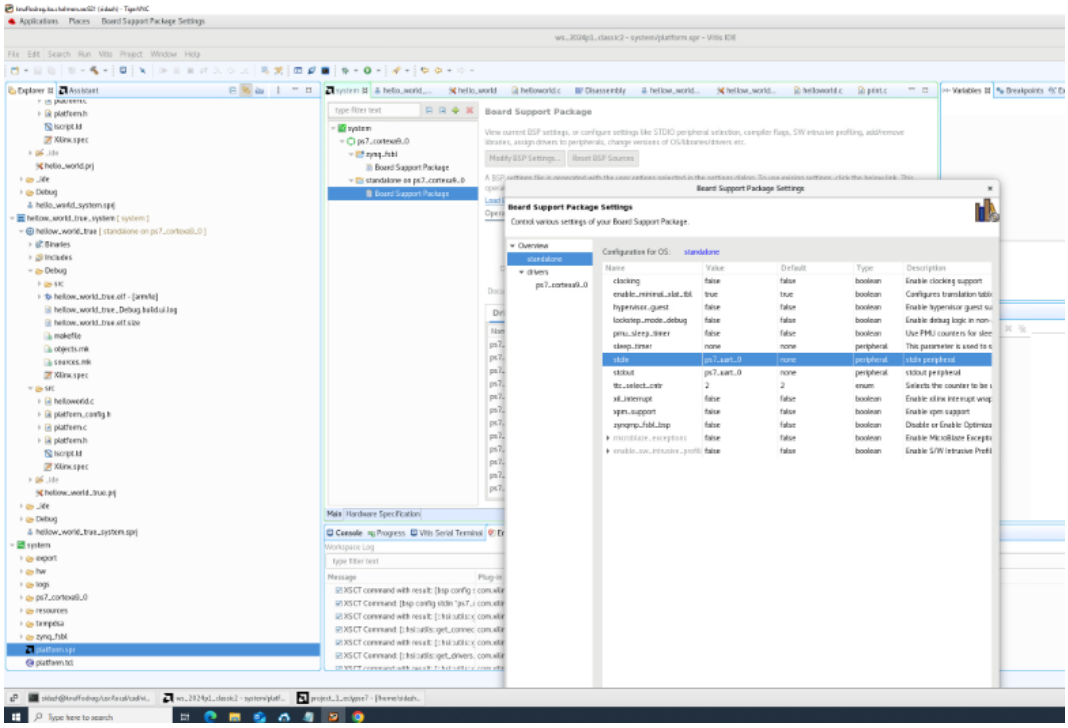

Figure 6: stdin and stdout configuration in Vitis

2. On the local machine, run the hw_server from the Vivado directory, NOT from the Vitis directory (it doesn't work with Vitis for some reason). Path on the lab computer:

```
C:\Progs\Xilinx\Vivado\2024.2\bin\hw_server.bat
```

3. Build the project. Set up a target connection for the hardware server running on the local machine and set it as the default. See Figure 7.

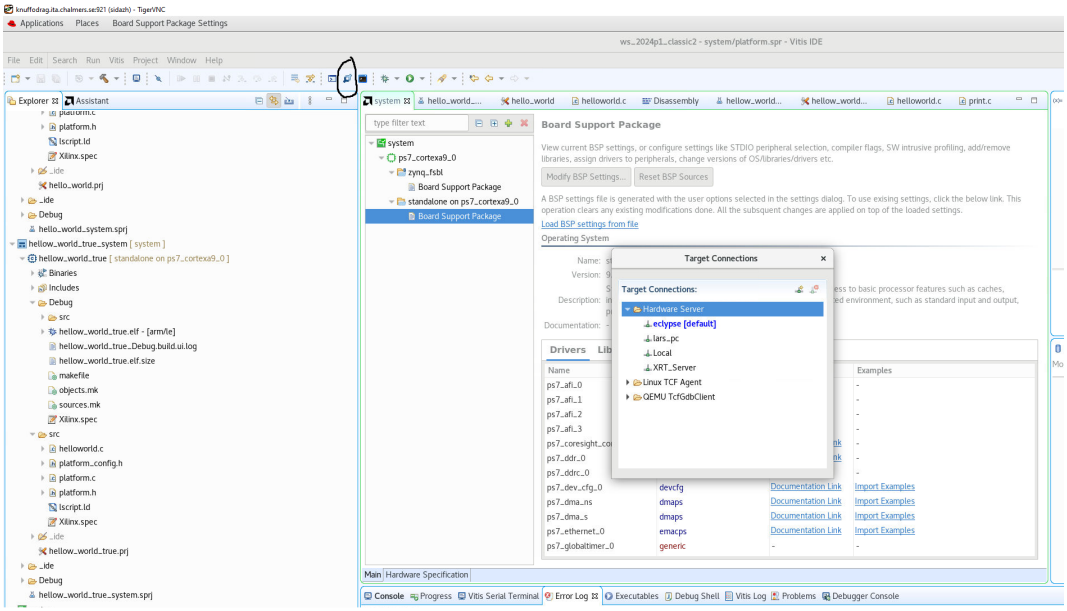

Figure 7: Target connection setup in Vitis

4. On the local computer, open a serial terminal with the correct COM port and baud rate 115200 (Eclypse7 uses this as default).

5. Make sure to select the correct connection for the application project run configuration (see Figure 8) and click run. We use Vitis Classic, but we assume it should be similar to the Vitis Unified IDE.

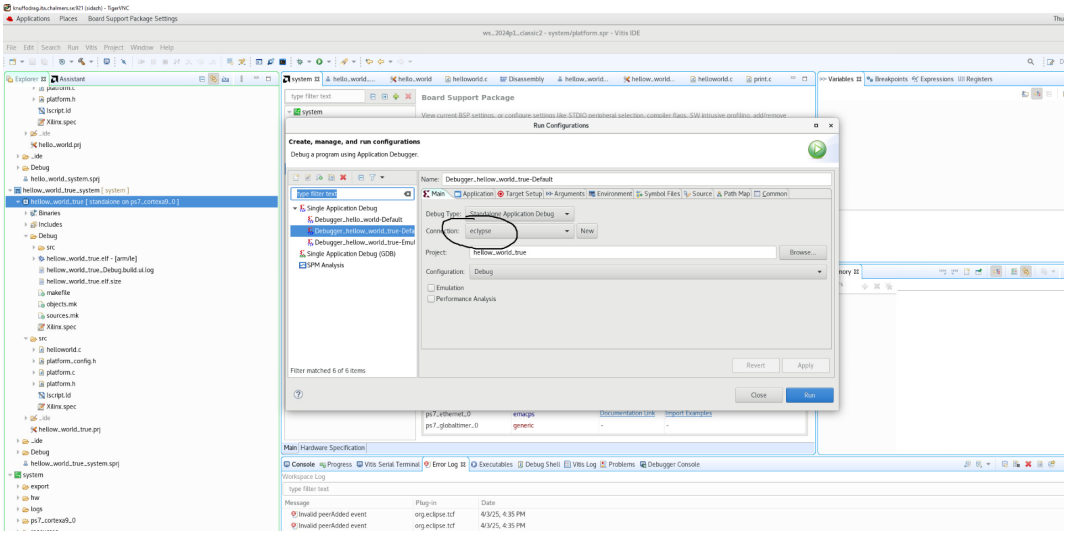

Figure 8: Run configuration selection in Vitis

6. Once the program downloads to the board, you should be able to see the output on the serial terminal.