MINNESOTA STATE UNIVERSITY, MANKATO
DEPARTMENT OF COMPUTER INFORMATION SCIENCE
COMPUTER SCIENCE PROGRAM
HANDOVER DOCUMENT
THE PENGUIN SURGEONS TEAM

Clients:
NextGen RF

Ross Loven

THE PENGUIN SURGEONS MEMBERS

Cole Schoenbauer
Research Lead

Adam Chase
Documentation Lead

Biruk Anley
Testing Lead

Ahmad Tamer
Team Lead

FACULTY COACH

Ryen Leytem

Subject Matter Expert

Flint Million

Table Of Contents

1) Introduction

1.1) Client Overview

NextGen RF is a U.S.-based engineering services firm specialising in wireless design solutions. We offer a broad range of services from design consulting to comprehensive turnkey product development, leveraging our extensive expertise in wireless technologies to meet diverse product requirements.

1.2) Project Overview

This project will develop and establish a standardised build process for Xilinx Zynq processors utilising the Petalinux embedded operating system and relevant Xilinx tool chains. Our goal is to streamline kernel configuration and build procedures. The project team will produce detailed, wiki-style documentation to guide end-users through setting up the build environment, integrating the Analog Devices' ADRV9002 IIO Driver, and tailoring the device tree for specific target devices. As an advanced objective, the team aims to enhance user experience by enabling the validation of the Linux system and drivers. This will be achieved by developing a cross-platform GUI using GNURadio, which will facilitate issuing IIO commands to the underlying operating system.

1.3) Tools Provided

The following resources and tools are available to support the project:

- **Base Kernel and Build Instructions**: Includes a foundational Linux kernel and detailed guidance on the build process.
- **Xilinx Development Tools**: Access to essential Xilinx tools such as Petalinux, Vivado, and Vitis, which are critical for developing and optimising designs.
- **Linux Repository and Wiki**: Hosted by Analog Devices, this repository contains the necessary Linux configurations and enhancements, as well as the wiki for building their kernel:
  - Analog Devices Linux Repository
  - Analog Devices Wiki for Building on ZynqMP SoC's
- **Project Reference**: For further development resources and support, refer to NextGen RF's proprietary SDK:
  - NextGen RF Design Inc - BytePipe SDK
- **Driver Testing Tool**: An IIO Oscilloscope tool provided by Analog Devices, useful for testing and validating drivers:
  - IIO Oscilloscope Tool

2) Tech Stack

2.1) Setting Up The Build Environment

- **Base Linux Environment:** Utilises a general Linux setup as the foundational operating environment, which we used Ubuntu 18.04.
- **Initial Dependencies Installation:**
    - Tools such as Git, build-essential, u-boot-tools, snapd, and others are installed via apt-get commands to prepare the system.
    - Bash is set as the default shell over dash for script compatibility.

2.2) Xilinx Tool Suite Installation

- **Vivado and Vitis (2021.1 and 2023.2 Versions):**
    - The Vivado Design Suite and Vitis are downloaded and installed from archived and current versions, respectively. This involves setting executable permissions and using the Linux Self-Extracting Web Installer.
- **Petalinux (2021.1 and 2023.2 Versions):**
    - Additional dependencies are installed to support Petalinux installation, followed by the execution of the Petalinux installer. This step also uses the Xilinx Unified Installer for streamlined setup.

2.3) Error Documentation and Support

- A specific GitHub documentation (NextGen RF's GitHub) provides steps and solutions to common issues encountered during the setup and installation processes.

2.4) Development and Documentation Tools

- **GitHub:** Used for hosting the project repository and documentation.
- **RFLAN Documentation and Build Process:**
    - Documentation on building the RFLAN CLI tool which includes setting up the environment, sourcing tools, compiling the HDL, and software for RFLAN using make commands.
    - The application development involves interfacing with ADRV9001 IIO drivers and handling data via DMA streams.

2.5) Deployment

- The compiled and built RFLAN application is packaged and deployed to the BytePipe hardware using an SD card, allowing it to run on boot.

3) Documents Created

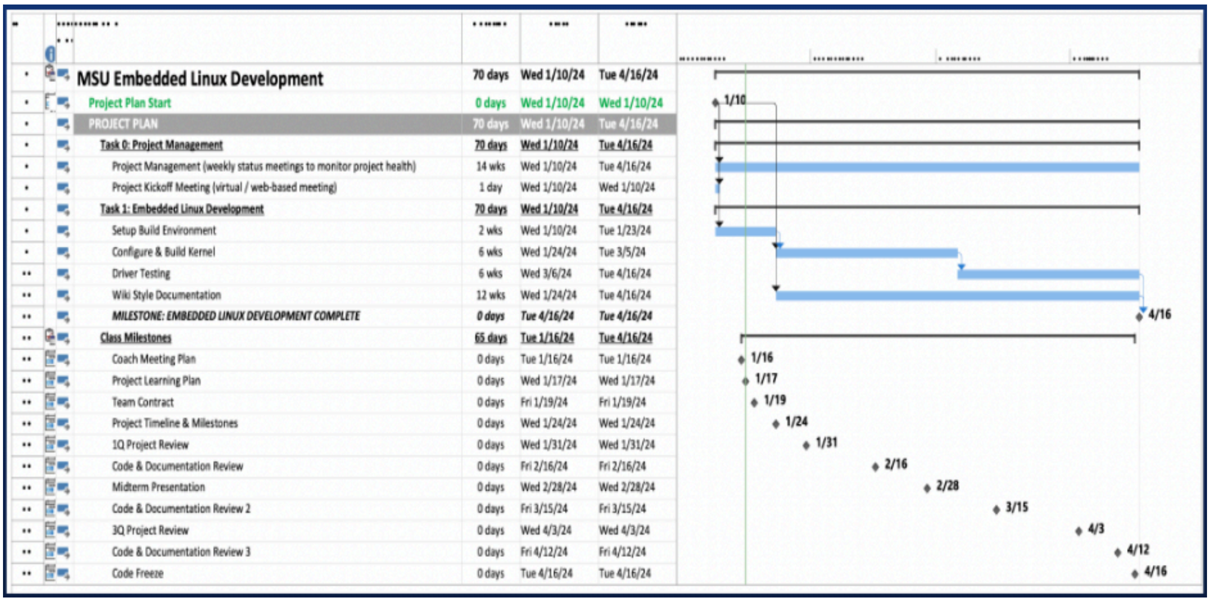
There were a couple of documents we developed to get started with this project.

3.1) Timeline and Milestones

This is a document that uses milestones to divide project work into phases for easy monitoring. This document includes a list of critical dates and actions included in the project.

Our timeline remained consistent throughout the project and aligned with our planned schedule. Following our midpoint assessment, we confirmed that our progress on various project components was on track. We maintained the originally set deliverable dates and continued to follow the established timeline without the need for adjustments.

Below is our project timeline that was discussed and created with our clients NextGen RF Design at the beginning of the project.



3.2) Documentation Outline

This is the general outline of how to follow our documentation for our build process.

3.2.1) Getting Started

This document gives recommended hardware specifications for this project. It also lists the hardware specifications of the system that build the final SD card files for our final image. There is also a guide on how to install Ubuntu 18.04.4 to your system, as this was the recommended operating system and version to be used for this project. Finally, there is also a list summarising the technologies and tools used for this project.

3.2.2) Setting Up the Build Environment

Here we specify the process we used to get all of the necessary dependencies and tools installed to our systems. This generally follows the *BuildEnv.md* document within NextGen RF's GitHub for getting the build environment set up, but we also have some extra steps in our document to give extra detail. We also included steps for installing the newer versions of the Xilinx tool suite (2023.2), which were not specified in NextGen RF's documentation.

3.2.3) RFLAN Documentation

The objective of this document is to replicate the build process for the RFLAN application that came pre-built and deployed on the HDK's provided to our team. Following this process gives an idea of what building files for an embedded device like the BytePipe might look like. Following this process also provides some necessary artefacts for the *Building ADI Kernel* document. The two important artefacts generated from this step are system.dts (our device tree file), and rflan_xczu3eg-sbva484-1-e.xsa (our HDL file).

3.2.4) Building ADI Kernel

This document is where we spent the most time in our project as we attempted all of the methods on ADI's wiki for building the ZynqMP / MPSoC Linux kernel and device trees from source. This includes our experience following the "The script method", "Building using Petlinux", and "On the development host" steps to generate the necessary files to boot their Linux image on the BytePipe. Finally, there is a step-by-step guide for replicating the build process that worked for us to get the modern 6.1.0-xilinxv2023.2 kernel running on the BytePipe.

3.2.5) Additional Documents

We also have two other documents titled *Resources Documentation* and *Error Documentation with Solutions*. *Resources Documentation* is meant to carry the external resources we used for our project as well as a small description for each resource being linked. While following along with our main build documents, these resources can provide useful additional information with what's going on in the project. *Error Documentation with Solutions* is a document that contains a table listing some bigger problems we encountered during our project as well as who had the issue, when the issue occurred, and what the found solution was for the issue. If you have any issues while following along, look to this document as a reference as it may contain a solution.

4) What Was Achieved

During our project timeline, we were able to build a 6.1.0-xilinxv2023.2 analog kernel version, modify the device tree to work with the BytePipe. This kernel contains support for the IIO Oscilloscope drivers. This will pave the way for building the IIO Oscilloscope plug in (as discussed in Section 5: Next Steps). In addition to that, this new updated kernel has support for many other drivers as well as older drivers which means that the RFLAN application should run while using the new kernel as well as any newer application the client wishes to implement on the BytePipe. Additionally, we used updated versions of the AMD Xilinx tool suite (Petalinux, Vivado, Vitis), version 2023.2, instead of using the older 2021.1 version of the tools. To achieve this, we experimented with all the different possibilities to build the new kernel for the BytePipe and modify the device tree to match the BytePipe and documented our learning process, our challenges, and what worked and what didn't with the goal of giving future teams direction to easily build on this project.

5) Next Steps

- Implementing IIO Oscilloscope tool for comprehensively testing the presence of drivers within the ADI Linux image
- Build and use RFLAN application within new Linux image
- Build a GUI interface to send industrial I/O(IIO) commands to the product's underlying(embedded) linux operating system
- GNU Radio integration

6) Deliverables

Below shows a table of the deliverables that were given at project assignment.

Deliverables	Type of work	Activities	Resources	Tech Skills	Priority
Set up a functional build environment on a Linux host with the specific version of build tools required to replicate a released build of ADI Linux from source	Embedded Software Engineering, Computer Engineering, OS	Research, install, and configure build environment	Github reference: https://wiki.analog.com/resources/tools-software/linux-drivers-all#building_the_adi_linux_kernel	Linux, Embedded Systems, Vivado, Petalinux, Cross-compiler toolchains	High
Configure and build the Kernel with BytePipe as the target, including HDL and boot image.	Embedded Software Engineering, Computer Engineering, OS	Research BytePipe SOM, create custom device tree and integrate into the working build process	https://github.com/NextGenRF-Design-Inc/bytepipe_sdk/blob/main/src/petalinux/README.md	Linux, Embedded Systems, Vivado, Petalinux, Cross-compiler toolchains, Bash or Python scripting	High
Driver Testing	Embedded Software Engineering, Computer Engineering, OS, Documentation	Create Software Verification Test Plan Execute testing and report results	IIO Oscilloscope ADRV9002 Control IIO Scope Plugin	Debugging and troubleshooting, Documentation	High
GNURadio Integration (optional)	Embedded Software Engineering, Computer Engineering, OS	Install GNURadio on a windows or linux host, and configure it to use a BytePipe SOM as a target device via IIO commands	https://www.gnuradio.org/	Linux, Digital Signal Processing	Low
Create detailed build process documentation in a Wiki style	Embedded Software Engineering, Computer Engineering	Document the entire build process for inclusion in our Github page.	Reference: https://github.com/NextGenRF-Design-Inc/bytepipe_sdk	Linux, Embedded Systems, Vivado, Petalinux, Cross-compiler toolchains, Bash or Python scripting, Github, Documentation	Medium