MANIPAL
ACADEMY of HIGHER EDUCATION
(Institution of Eminence Deemed to be University)

Firmware Development and Testing using QEMU

Mini-Project Final Report

submitted by

Reg. Number	Name	Branch
221039016	VARSHA V D	Embedded Systems
221039024	ARJUN M KRISHNA	Embedded Systems

Under the guidance of

Dr. Mohan Kumar J

Associate Professor,

Manipal School of Information Sciences,

MAHE, MANIPAL

21/12/2022

MANIPAL SCHOOL OF INFORMATION SCIENCES
MANIPAL
(A constituent unit of MAHE, Manipal)

MANIPAL SCHOOL OF INFORMATION SCIENCES
MANIPAL
(A constituent unit of MAHE, Manipal)

CERTIFICATE

This is to certify that the project titled **Firmware Development and Testing using QEMU** is a record of the bonafide project work done by **VARSHA V D** (*Reg. No. 221039016*), **ARJUN M KRISHNA** (*Reg. No. 221039024*) submitted for learning as a mini project for 1st Semester ME (Embedded Systems) of Manipal School of Information Sciences. This is an original piece /emulated work, done as per the regulations of the Manipal Academy of Higher Education, Manipal for the purpose of project learning.

ACKNOWLEDGEMENT

We would like to express our sincere gratitude to everyone who helped and guided us in completing this project work. We would also like to express our sincere gratitude to our project guide, Dr. Mohan Kumar J, for his valuable advice, guidance and feedback for the project. We are also thankful to all the panel members for their assistance and encouragement.

ABSTRACT

Firmware Emulation can serve a number of different purposes such as analyzing the firmware in a better way, performing exploitation, performing remote debugging, speed up testing and so on. With this technique, one can emulate a Firmware originally meant to be run on a different architecture, and interact with it even without having a physical IoT device. It makes it possible to replicate the behaviour of a physical device or program virtually.

Physical devices can be difficult to test thoroughly for faults and moreover, hardware is required for testing. In order to overcome this challenge emulation plays an important role as it can provide better debugging capabilities as compared to running the device on actual hardware. Currently, there are different solutions for device emulation. Some of the most popular are QEMU, GHIDRA, Firmadyne, ARMX Firmware Emulation Framework and many more.

QEMU is widely used in cloud environments. Basically it is a fast and portable dynamic translator and an embedded machine emulator that emulates multiple CPUs and many board models. But QEMU only supports common hardware, so new virtual device module should be developed for QEMU in order to emulate new hardware. Research result shows that the whole running, testing and debugging environment are built, and user level applications can be developed for the virtual hardware without the physical device become available.

In this project, we are demonstrating few of the hardware devices that can be emulated using the QEMU, a popular open source emulator and ARMX Firmware Emulation Framework and propose detailed steps to the emulated hardware devices.

LIST OF FIGURES

Contents

Chapter 1

INTRODUCTION

Industrial Automation and Power technologies are characterized by extremely complex software systems which are in many cases very tightly coupled with mechanical and electrical sub-systems. These systems tend to be safety and mission critical. Development of such systems is considerably time consuming. However, they are also characterized by long if not very long product lifecycles sometimes in the order of fifteen years. The scope and complexity of software components in these industrial products has gone up significantly over the years. Continuous remodelling and development of software for the enduring industrial automation systems, places considerable challenges of efficiency on the development environment and processes followed. Here we look at the acceleration of embedded software development by decoupling it from the underlying hardware.

Traditionally, embedded software development is inherently dependent on hardware availability. Hardware development includes design, simulation and testing of hardware architecture, logic, circuit schematics and finally the PCB. This is a very time consuming process, often iterative and unpredictable due to dependency on factors like component availability and vendor support. Software developers usually cannot afford to wait for the entire hardware design to be completed and the board to arrive on their desks before starting on application development. This would increase the product development time substantially which is unacceptable. Application development as well as verification(testing) should be well underway by the time the hardware is finalized and delivered to the software developers. This is where system emulation and virtual environments step in and save the day. With the various advanced hardware emulators and virtual machines available today, embedded software developers can get a head start on their applications. There are few emulation products. SkyEye is an embedded emulator and mainly for ARM processors. Bochs is a platform emulator but only focus on X86. QEMU is a fast, portable and dynamic binary translator(DBT) that supports a wide range of processors(X86,

ARM, PowerPC, MIPS etc), it can also emulate the whole platform not only PC but also embedded development board. In this work, we choose QEMU as target emulator.

1.1 QEMU (Quick EMUlator)

QEMU is a general-purpose, open-source machine emulator and virtualizer that can run operating systems and applications designed for one machine (such as an ARM board) on another when used as a machine emulator (e.g. your own PC). It runs on several host operating systems such as Linux, Windows and Mac OS X. The host and target CPUs can be different. QEMU is capable of emulating a complete machine in software without having actual hardware. It provides excellent system-level compatibility and support, making it ideal and lightweight virtual machine environment.

1.2 Why QEMU?

QEMU is used for the emulation of the Raspberry Pi board and Tenda AC15 Wi-fi Router. It was based on the following considerations.

- QEMU supports ARM processor emulation.
- QEMU supports both x86-Windows and x86- Linux as host machines.
- QEMU is an active open source project with a good developer community and support.

1.3 Firmware

Firmware is a program on a hardware device that provides a control for a specific hardware device and gives instructions for how that hardware device communicates with other computer hardware. Firmware Development is the process of writing code that runs one embedded hardware rather than on a full-fledged computer. It is done on a microcontroller or a

microprocessor. Firmware is not entirely considered to be software, but code that manages the sensors, peripherals, motors, and timers in devices.

Embedded firmware is responsible for controlling various peripherals of the embedded hardware and generating responses in accordance with the functional requirements for the particular product. Firmware is considered as the master brain of the embedded systems. Imparting intelligence to an embedded system is a one-time process and it can happen at any stage of the design. Once the intelligence is imparted to the embedded product, by embedding the firmware in the hardware, the product start functioning properly and will continue serving the assigned task till hardware breakdown occurs or a corruption in embedded firmware occurs. Designing an embedded firmware requires understanding of embedded product hardware like, various component interfacing, memory map details I/O port details, configuration and register details of various hardware chips used and some programming language.

1.4 Docker and Container

Docker is an open source tool that allows developers to easily deploy their applications in a sandbox (called containers) to run on the host operating system i.e. Linux. The key benefit of Docker is that it allows users to package an application with all of its dependencies into a standardized unit for software development. The whole idea of Docker is for developers to easily develop applications, ship them into containers which can then be deployed anywhere.

Chapter 2

LITERATURE SURVEY

Paper 1

Pradyumna Sampath, Rachana Rao proposed “Efficient embedded software development using QEMU”

This paper looks at employing virtualization for in-house real-time application software development through QEMU. It also details the process of establishing such an environment and looks at some of the advantages of such a development model for embedded systems. QEMU is briefly presented and explained its advantages over other emulating tools. Integrating QEMU with Eclipse, it makes a fairly complete software development environment reducing the time to market for embedded products. Virtualization provides the ability to simulate components like load generators, performance monitors and other resources beyond the scope of a usual test setup.

Paper 2

XioXio Bian proposed “Implement a virtual development platform based on QEMU”

This article gives a detailed study of architecture and internals of QEMU and explains the steps to create user defined virtual hardware devices and also develop the Linux kernel for the new device. Here they built convenient environment and user level applications which can be used to develop, test and debug target code without access to actual hardware. Software development takes place before the hardware is ready.

Chapter 3

OBJECTIVES & SPECIFICATIONS

3.1 Objectives

- To learn and understand the working of QEMU emulator.
- To emulate Raspberry PI with QEMU emulator.
- To emulate the available Firmware (Router/Any boards) on the QEMU emulator.

3.2 Specifications

- To accomplish this, the target runs on the latest LTS version of Ubuntu Operating System 22.04.1 LTS.

Chapter 4

PROPOSED BLOCK DIAGRAM

4.1 Block Diagram

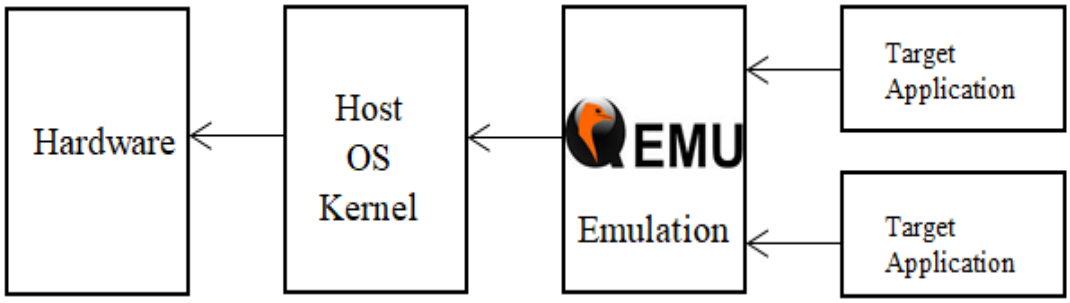

Fig 1: Proposed Blocked Diagram

When emulating, there are some basic terms that we refer to the systems that is involved in the process. The two main components are the host machine and the target machine. The target is the system that we wish to emulate. The host is the machine that is used to build the target software. In Raspberry Pi emulation the target will be Raspbian OS and the host will be Ubuntu. In Tenda AC15 Wi-fi router emulation target is the router and the host is ARMX.

Chapter 5

DESIGN AND IMPLEMENTATION

In this section, detailed steps to emulate Raspberry pi and Tenda AC15 router is explained.

5.1 Emulating Raspberry Pi with QEMU:

Here the host machine is Linux kernel and the target machine is Raspbian OS. The Raspbian OS needs to be setup and emulated on QEMU using certain set of linux commands.
Ubuntu will be used to imitate the desired ARM versions. We will require the following for the QEMU simulation:

- Raspbian OS image file.
- The corresponding latest qemu kernel for the Raspbian OS.

Make a new folder and insert the raspbian OS image file and the corresponding kernel in the folder and emulate raspberry pi on QEMU as explained in the below section.

5.1.1 Steps to emulate the Raspbian OS on QEMU:

Step 1: First, install the latest Ubuntu version and run it on a VM

Step2: To perform QEMU emulation the following are required:

- A Raspbian Image
- Latest qemu kernel

Step 3: Inside the Ubuntu VM, create a new folder and place the Raspbian Jessie image and qemu-kernel into the folder created.

Step 4: Emulate Raspbian OS on QEMU by running few linux commands. After this, GUI of Raspbian OS (Target application) appears on the host machine(Ubuntu).

5.2 Emulating the Tenda AC-15 Wi-fi Router with EMUX (formerly known as ARMX):

The Tenda AC15 Wi-Fi Router is a popular dual-band Gigabit WiFi router. Multiple devices can gain Internet access with this device. This Tenda AC15 router is successfully emulated with ARMX emulated device.

ARMX is a collection of scripts, kernels and filesystems to be used with QEMU residing in the /armx directory. It uses qemu-system-arm to boot up a virtual ARM/Linux environment. The /armx directory is exported over NFS to also make the contents available within the QEMU guest. ARMX is packaged as a Docker image. ARMX is aimed to facilitate IoT research by virtualising as much of the physical device as possible.

5.2.2 Steps to emulate the Tenda AC-15 Wi-fi Router with EMUX:

There are eight steps in running an emulated device:

Step 1: Build the docker volume and image.

Step 2: Run EMUX.

Step 3: Start the EMUX launcher.

Step 4: Launcher - choose from a list of available emulated devices

Step 5: Select a device and boot its kernel and its hostfs

Step 6: Userspace - choose from a list of available userspace actions

Step 7: Start the devices' userspace processes

Step 8: Device is booted up and ready.

Chapter 6

RESULT ANALYSIS

6.1 Raspberry Pi Emulation Result:

- fdisk is used to prepare and partition a brand new hardware. fdisk -l lists out all the options available for fdisk. We need the boot start address of 2017-03-02-raspbian-jessie.img2 and that corresponds to 92160 as shown in fig 2.

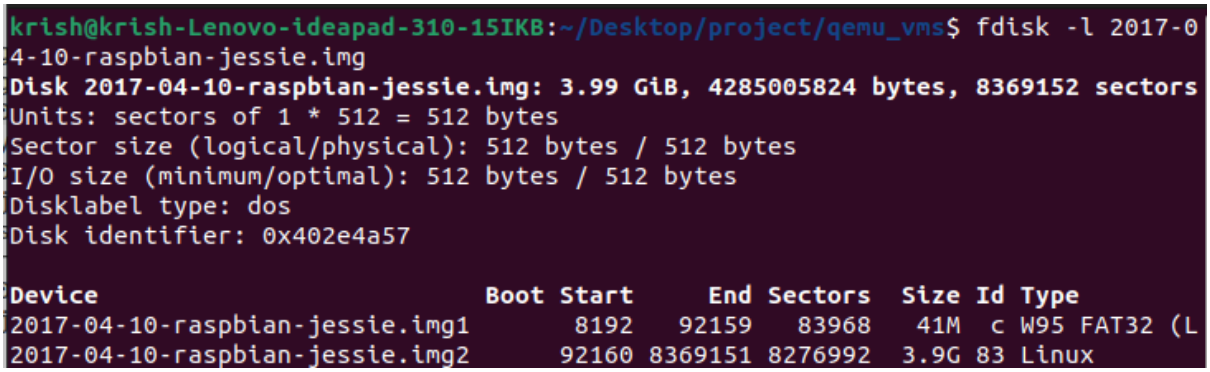

Fig 2: Snap of file system displaying boot start value.

- Next we need to mount the rasbian os image file with an offset which is equal to 512 x the boot start value obtained from above. That is 92160 x 512 = 47185920. We can mount the raspbian OS as shown in fig 3.

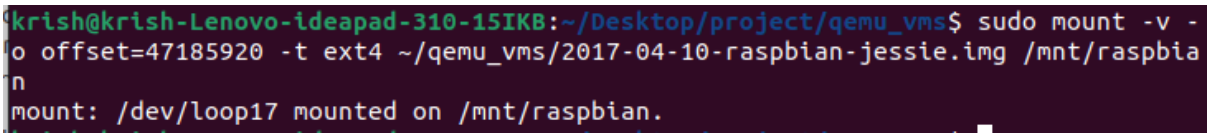

Fig 3: Mounting Raspbian OS.

- After mounting the raspbian image file, we will then emulate it using the "qemu-system-arm" command following with the kernel file – "kernel-qemu-4.4.34-jessie" along with specifying the Raspbian image file name as shown in fig 4.

```
krish@krish-Lenovo-ideapad-310-15IKB:~/Desktop/project/qemu_vms$ qemu-system-arm
 -kernel ~/qemu_vms/kernel-qemu-4.4.34-jessie -cpu arm1176 -m 256 -M versatilepb
 -serial stdio -append "root=/dev/sda2 rootfstype=ext4 rw" -hda ~/qemu_vms/2017-
04-10-raspbian-jessie.img
WARNING: Image format was not specified for '/home/krish/qemu_vms/2017-04-10-ras
pbian-jessie.img' and probing guessed raw.
         Automatically detecting the format is dangerous for raw images, write o
perations on block 0 will be restricted.
         Specify the 'raw' format explicitly to remove the restrictions.
pulseaudio: set_sink_input_volume() failed
pulseaudio: Reason: Invalid argument
pulseaudio: set_sink_input_mute() failed
pulseaudio: Reason: Invalid argument
Uncompressing Linux... done, booting the kernel.
```

Fig 4: Emulation of mounted Raspbian OS.

- With these steps we will be able to emulate raspbian OS using QEMU. The below fig 5. shows the emulation of the Raspbian OS running on QEMU.

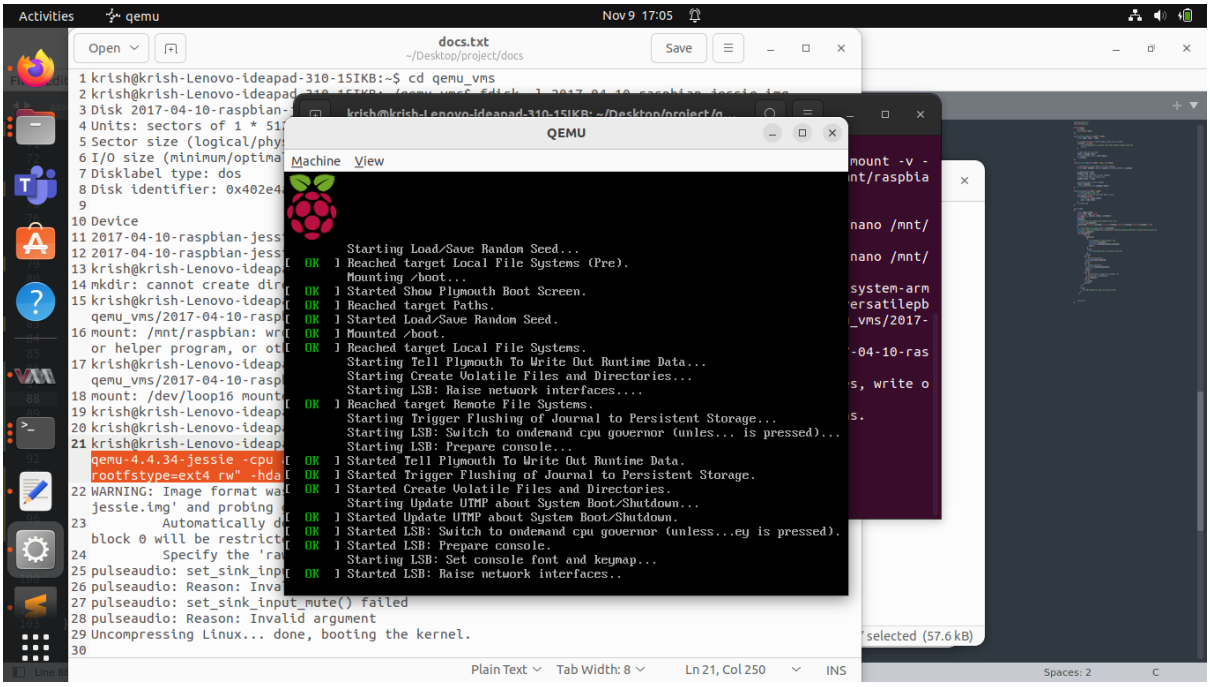

Fig. 5: Emulation of the Raspbian OS running on QEMU.

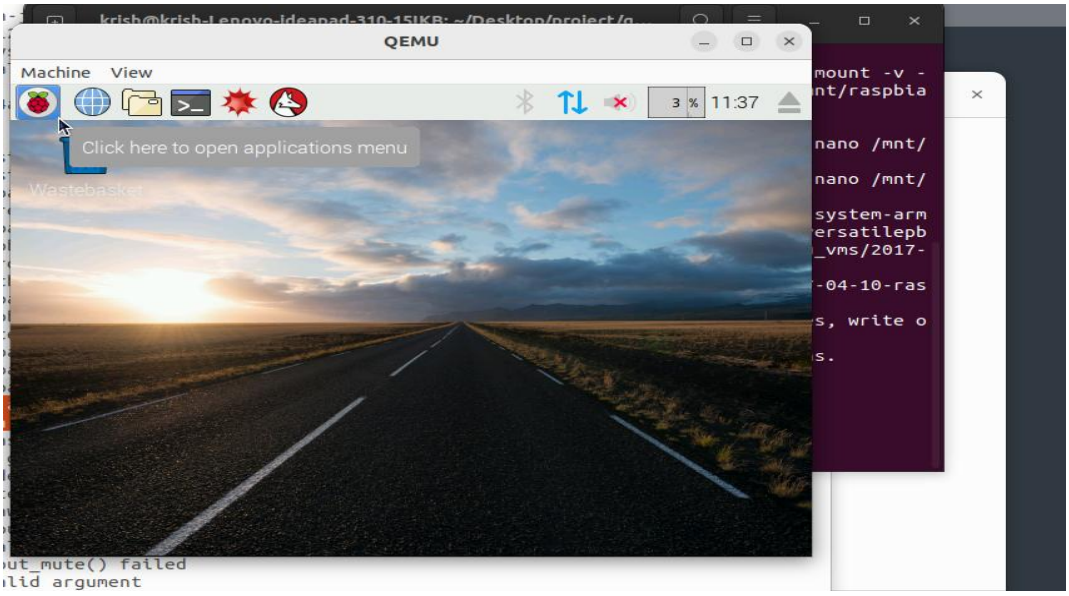

Fig.6: Emulated Raspbian OS.

- Construct a shared network interface (tap0), enabling access to all open ports (if those ports are not bound to 127.0.0.1). This can be done with the following commands on the HOST (Ubuntu) system:

  $ sudo apt-get install uml-utilities

  $ sudo tunctl -t tap0 -u <host-name>

  $ sudo ifconfig tap0 172.16.0.1/24

  $ ifconfig tap0

  Start QEMU VM, when the QEMU VM starts, assign an IP to its eth0 interface with the following command:

  $ sudo ifconfig eth0 172.16.0.2/24.

  We were able to reach open ports on the GUEST (Raspbian) from the HOST (Ubuntu) system as shown in the fig 7.

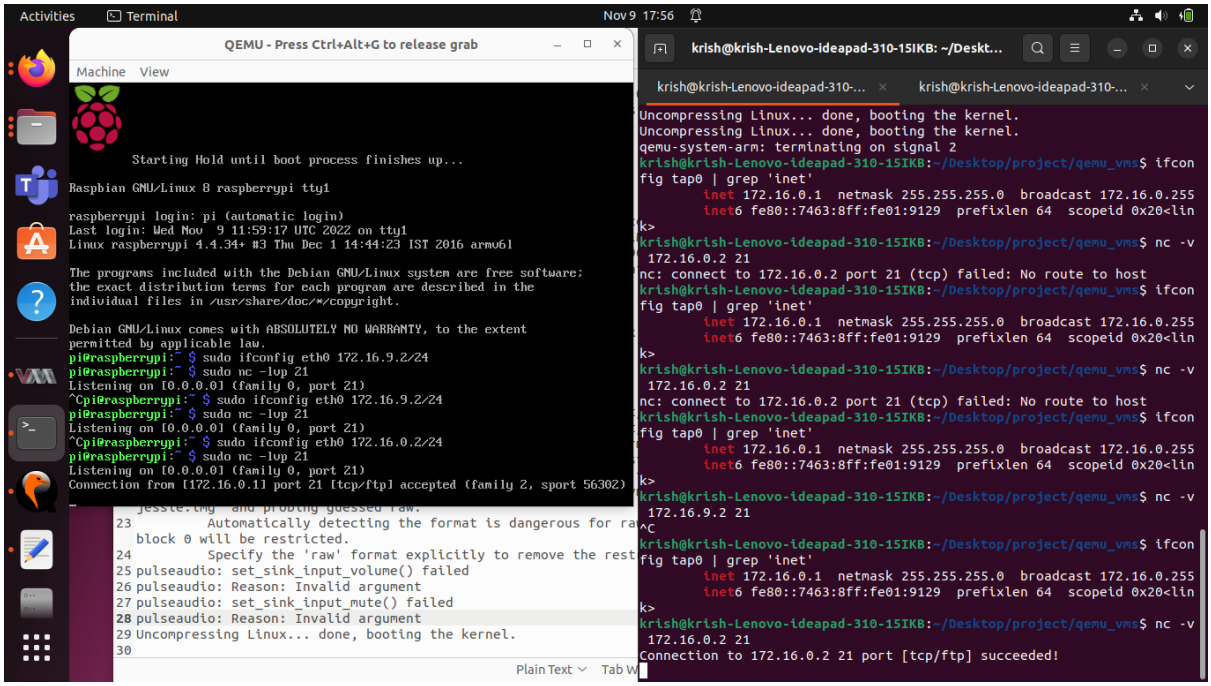

Fig 7: Established connection between host and target.

6.2 Tenda AC15 Wi-fi Router Emulation Result:

- Open a terminal, and start the emux-docker container.
- Invoke launcher.

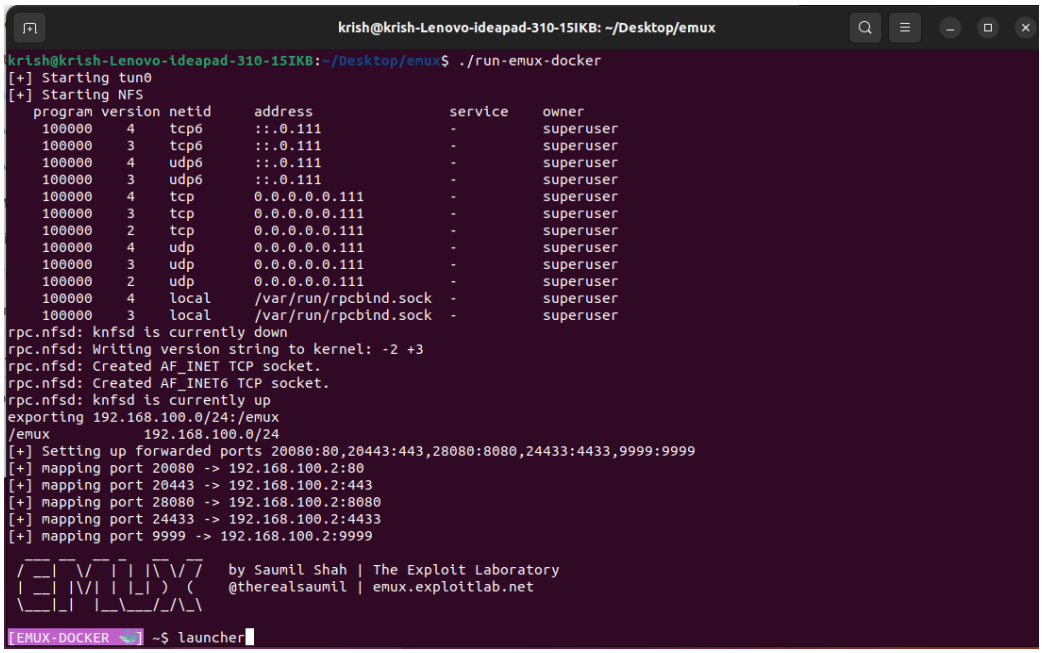

Fig 8: Invoke Launcher.

- This will display a menu as shown below. Select the Tenda AC15 Wi-fi Router.

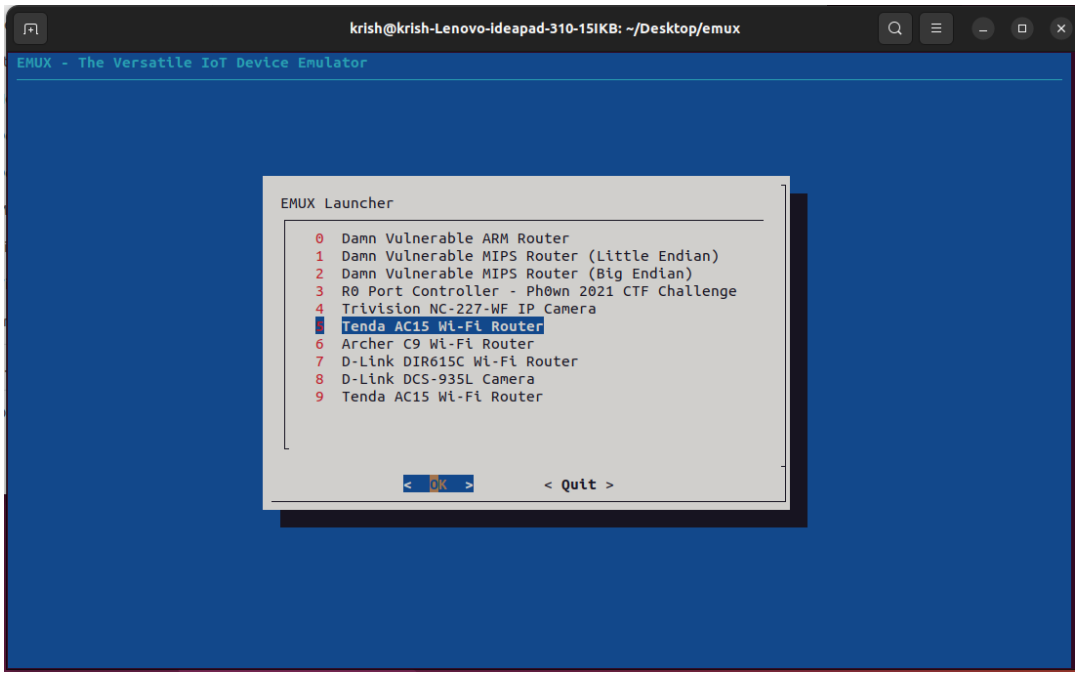

Fig 9: Display Menu of devices to be emulated.

- Selecting one of the devices will launch it under QEMU. The kernel which is included in the kernel/ directory of the Tenda AC15 Wi-fi Router's device configuration is booted in qemu-system-arm.

- Invoke the userspace command at the shell prompt.

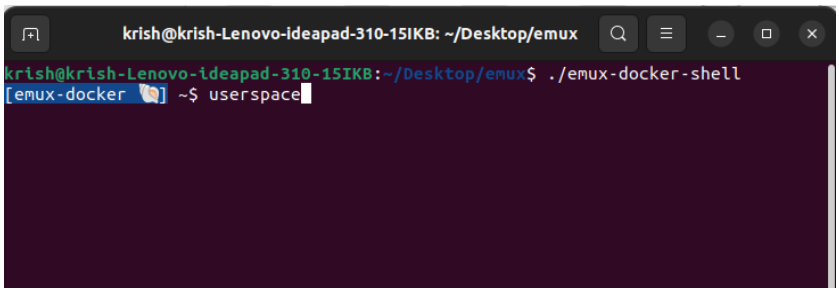

Fig 10: Invoke userspace at shell prompt.

- Internally the userspace command simply connects to the QEMU guest using SSH. This brings up a menu as shown below:

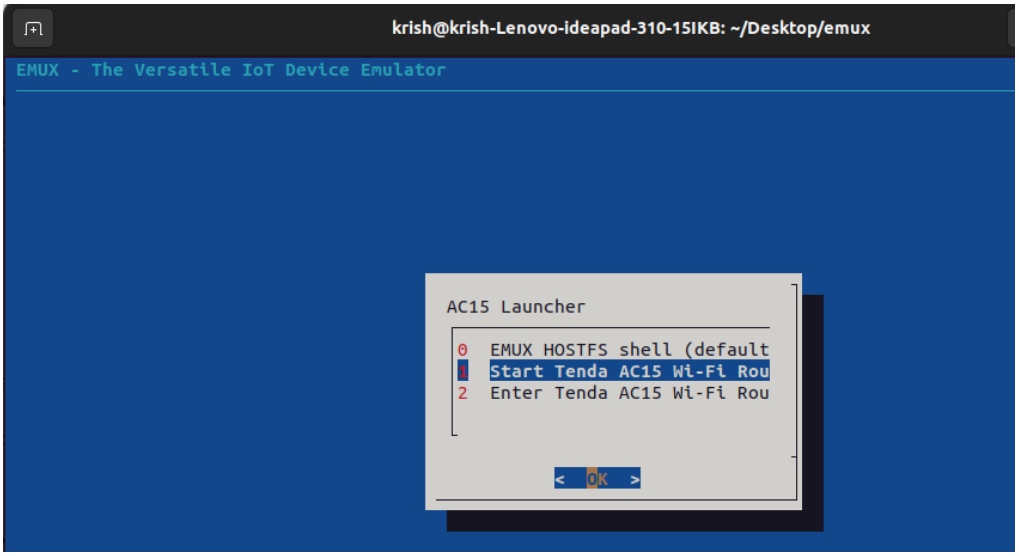

Fig 11: Display menu of Tenda AC15 router.

- Selecting the option to launch the userspace processes of the device results in run-init being invoked from the corresponding device configuration directory within /emux.

- Once the device has fully "booted up" in EMUX, it is available for testing and analysis. The image below shows the administration interface of the Tenda AC15 Router loaded in a browser.

- To access the web administration interface for the booted up device, open a browser and navigate to localhost:28000. This in turn will forward your request to 192.168.100.2:80 inside the emux-docker container. We are greeted by the Tenda Quick Setup Wizard.

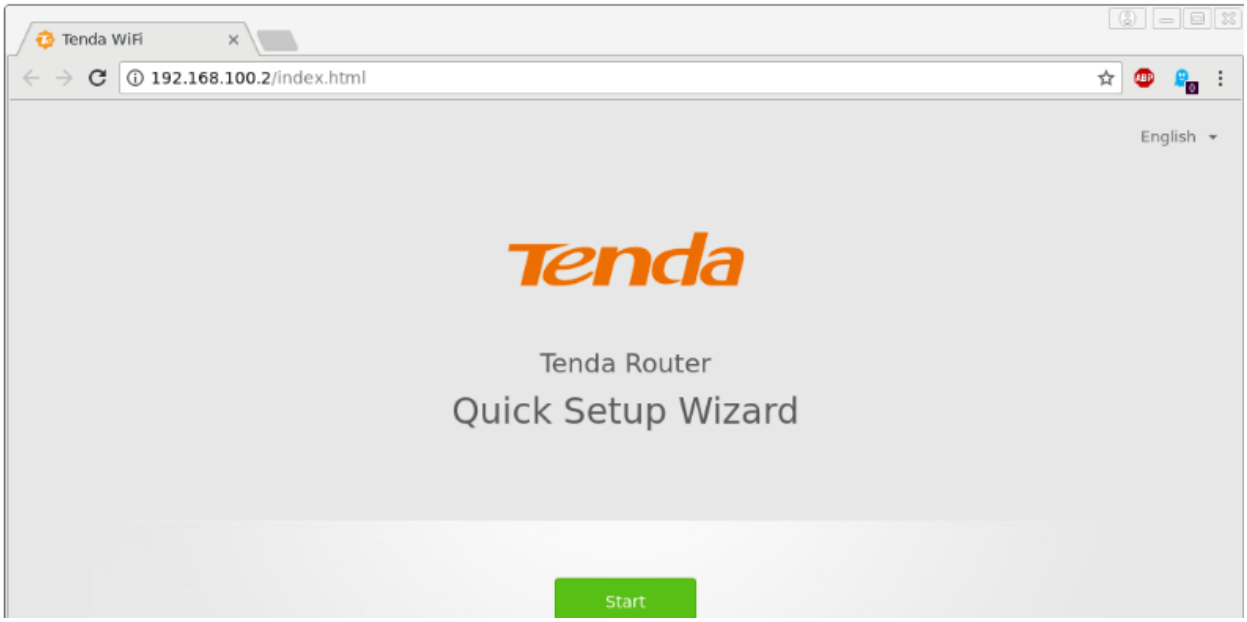

Fig 12: Interacting with the Tenda's web interface.

- Login to the Web Administration Interface.

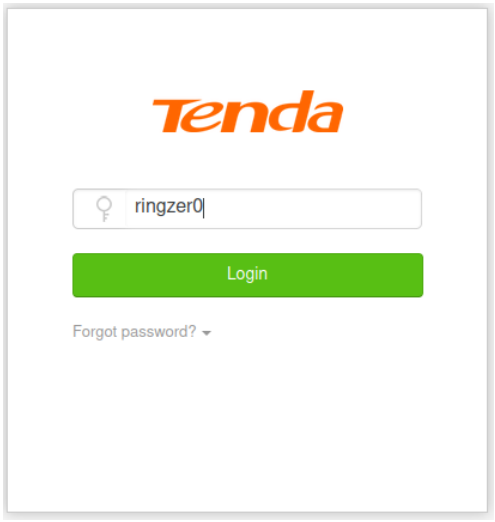

Fig 13: Tenda's Login Page.

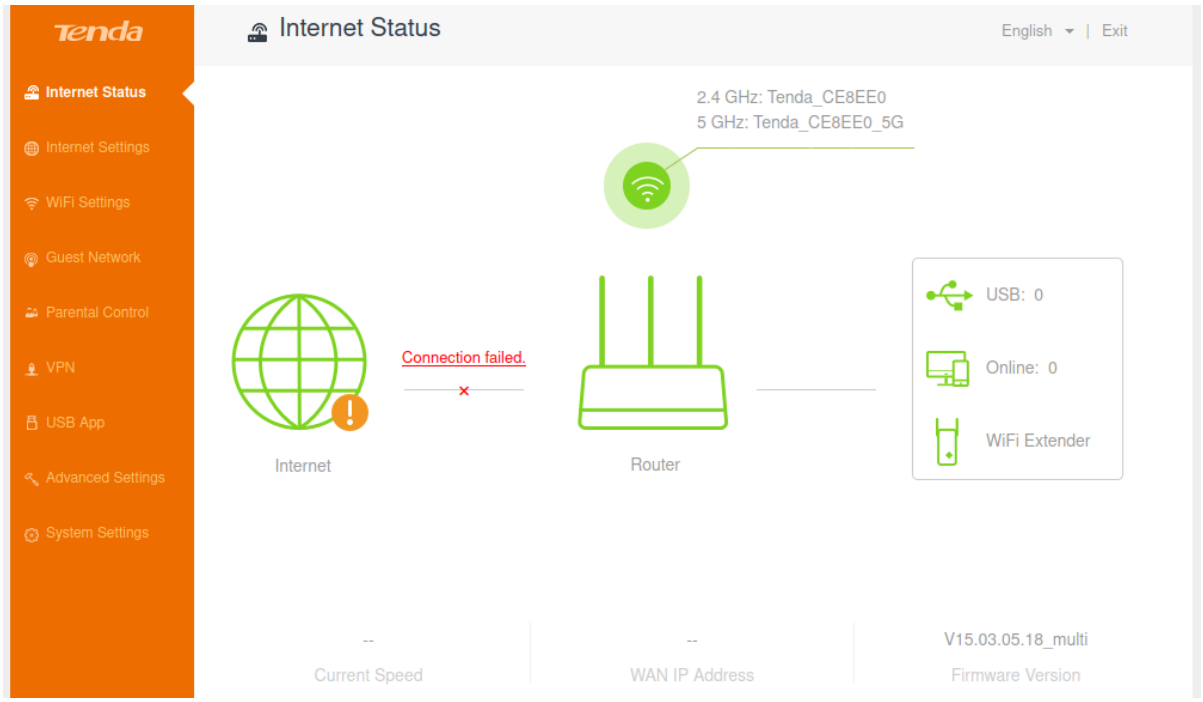

Fig 14: Emulated Tenda AC15 Router.

Chapter 7

CONCLUSION AND FUTURE WORK

7.1 Conclusion

Qemu plays an important role in the emulation of different kind of firmware and it is important in the field of Embedded application development. Since it is very hard to procure the necessary hardware required to build applications, emulating the same hardware, Qemu is very efficient.

Docker is a tool designed to make it easier for developers to develop, ship, and run applications by using containers. Containers allow developers to package an application with all of its requirements and configurations, such as libraries and other dependencies and deploy it as a single package.

7.2 Scope for future work

There are numerous tools such as QEMU , KVM and VMware server that help us to perform these emulation processes. However, new, and more mature alternatives are emerging every day that will probably help to speed up and facilitate this type of research work.

TIMELINE CHART

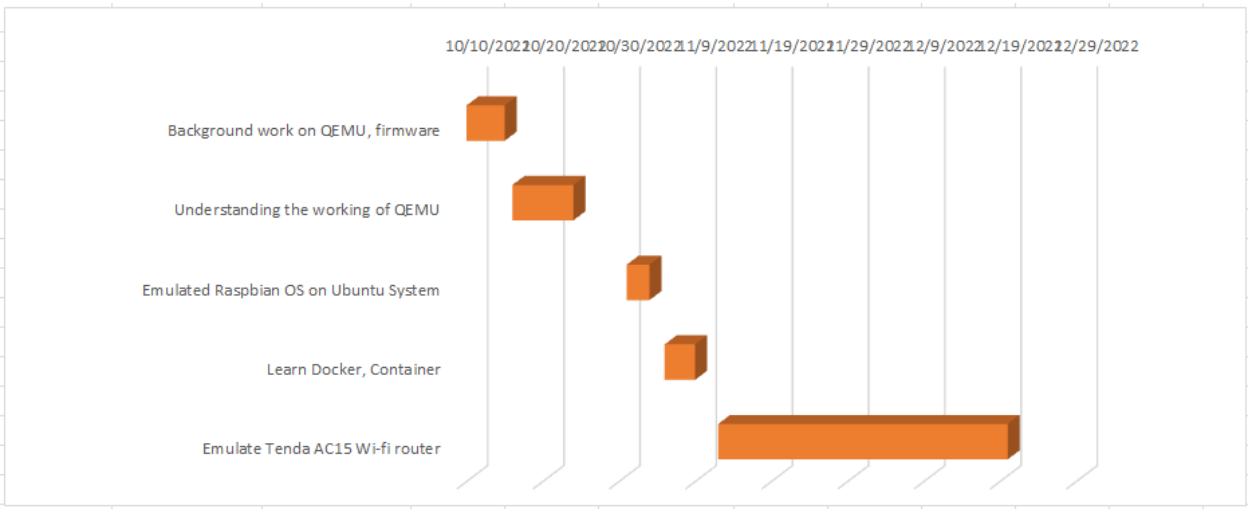

Table 1: Timeline Table

Tasks	**Background work on QEMU, firmware**	**Understanding the working of QEMU**	**Emulated Raspbian OS on Ubuntu System**	**Learn Docker, Container**	**Emulate Tenda AC15 Wi-fi router**
Start Date	10-10-2022	16-10-2022	31-10-2022	05-11-2022	12-11-2022
End Date	15-10-2022	24-10-2022	03-11-2022	09-11-2022	20-12-2022
Duration (No. of days)	5	8	3	4	38

REFERENCES

[1] XiaoXiao Bian, “Implement a Virtual Development Platform Based on QEMU”, *2017 International Conference on Green Informatics (ICGI).*

[2] Pradyumna Sampath, Rachana Rao, “Efficient embedded software development using QEMU”

[3] F. Bellard, "QEMU a Fast and Portable Dynamic Translator", *Proceedings of USENIX Annual Technical Conference*, pp. 41-46, June 2005.

[4] Raspberry Pi On Qemu
Available: https://azeria-labs.com/emulate-raspberry-pi-with-qemu/

[5] ARMX Firmware Emulation Framework
Available: https://armx.exploitlab.net

[6] Virtualization

Available: http://en.wikipedia.org/wiki/Virtualization

[7] QEMU

Available: http://www.qemu.org/

[8] Emulator

Available: http://en.wikipedia.org/wiki/Emulator