

# Project 2 - Module 2

Team Members: Nimish Sabnis, Nimisha Madapura Rajashekhar, Hannah Caldwell Meurer

## 1. Booting RPi with QNX

The Images below show what the QNX OS looks like upon boot and after a successful login.

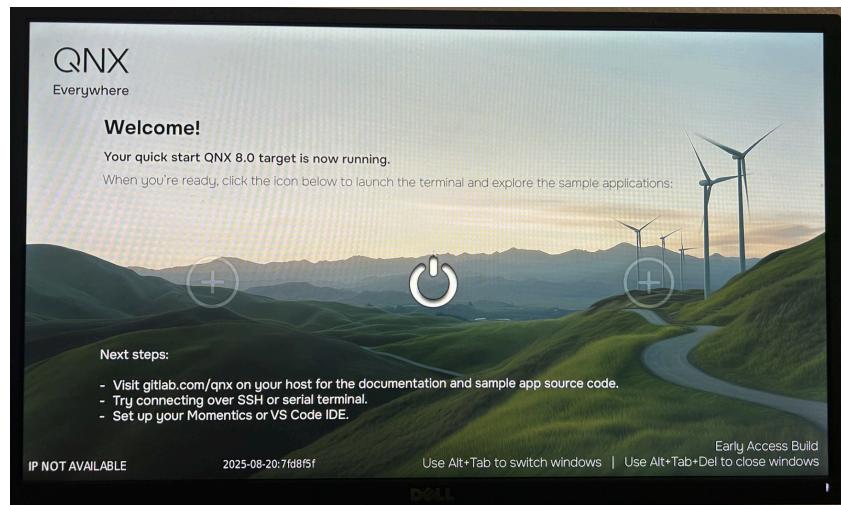


Figure 1: QNX Welcome Screen

A screenshot of a terminal window showing a user attempting to log in. The user enters "nisa8867@colorado.edu" and "Password:", but receives multiple "Login Incorrect" messages. The user then tries "root" as the password, which also fails. Finally, the user logs in successfully as "qnxuser". The terminal then displays the QNX 8.0 welcome message and information about bundled samples. At the bottom, it shows a command history and a note about using ALT-TAB to switch between windows. A footer at the bottom of the terminal window reads: "Have questions? Find the community on Reddit at /r/QNX, ask on StackOverflow, or log an issue at https://gitlab.com/qnx. qnxuser@qnx1:~\$".

Figure 2: QNX User Sign In

## 2. Serial Cable communication to serial tools

November 16, 2025

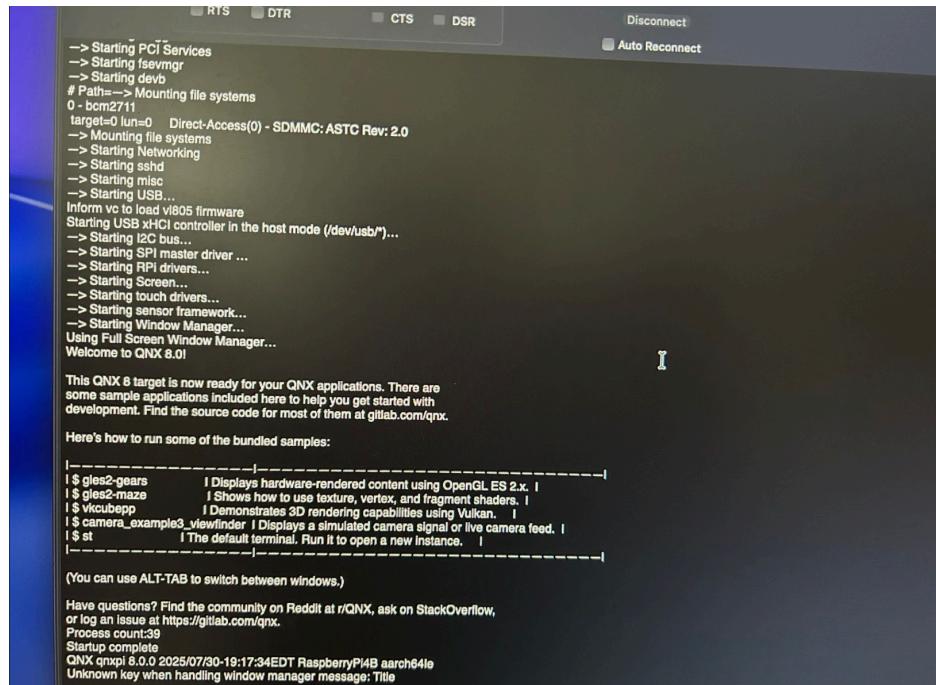


Figure 3: QNX Screen with Putty Connection via Serial Connection

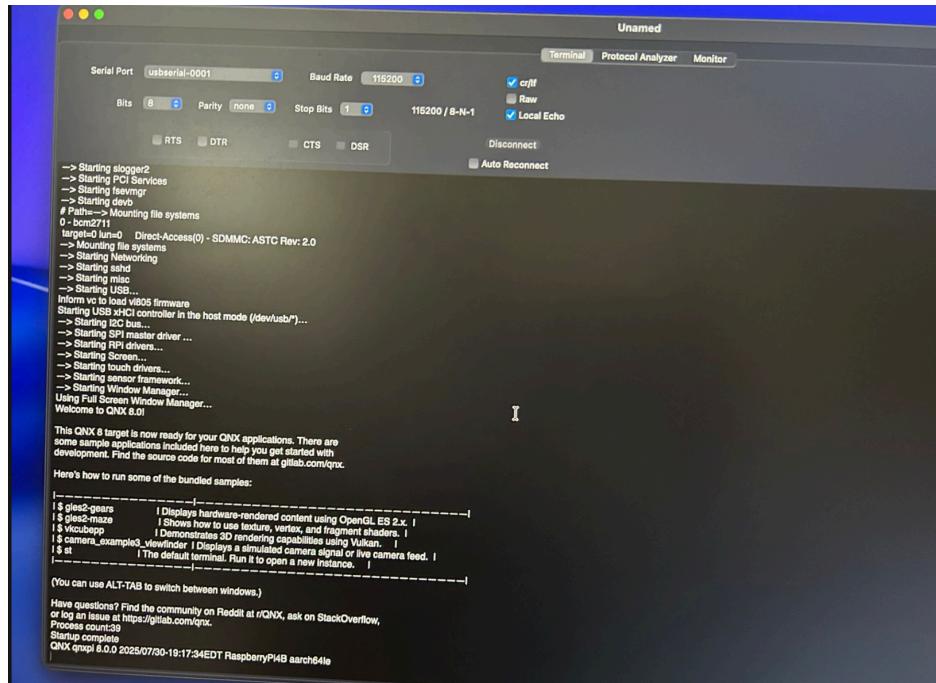


Figure 4: QNX Screen with Serial Cable Screen Capture #2

### 3. How much memory is used by the code? (What is the image size?)

The size of the image is 11Mb (ifs) + 2.4 Gb(System)

November 16, 2025

QNX qnxpi 8.0.0 2025/07/30-19:17:34EDT RaspberryPi4B aarch64le  
login: qnxuser  
Password:  
Welcome to QNX 8.0!

This QNX 8 target is now ready for your QNX applications. There are some sample applications included here to help you get started with development. Find the source code for most of them at [gitlab.com/qnx](https://gitlab.com/qnx).

Here's how to run some of the bundled samples:

\$ gles2-gears	Displays hardware-rendered content using OpenGL ES 2.x.
\$ gles2-maze	Shows how to use texture, vertex, and fragment shaders.
\$ vkcubepp	Demonstrates 3D rendering capabilities using Vulkan.
\$ camera_example3_viewfinder	Displays a simulated camera signal or live camera feed.
\$ st	The default terminal. Run it to open a new instance.

(You can use ALT-TAB to switch between windows.)

Have questions? Find the community on Reddit at [r/QNX](https://r/QNX), ask on StackOverflow, or log an issue at <https://gitlab.com/qnx>.

```
qnxuser@qnxpi:~$ df -h
ufs              11M      11M      0    100% /
/dev/hd0t179     31G     1.1G    30G     4% /data/
/dev/hd0t12      70M     16M    54M    23% /boot/
/dev/hd0t178     2.9G    594M    2.4G    20% /system/
/dev/hd0          58G     58G      0    100%
/dev/shmem        0       0       0    100% (/dev/shmem)
qnxuser@qnxpi:~$
```

Figure 5: Screen Capture of Memory Usage

#### 4. Ethernet communication

If one is using QNX while there is an Ethernet connection, the IP address can be seen in the bottom left-hand corner of the welcome screen. Then, on a personal computer using either Putty or a similar application, the UI for QNX can be seen.

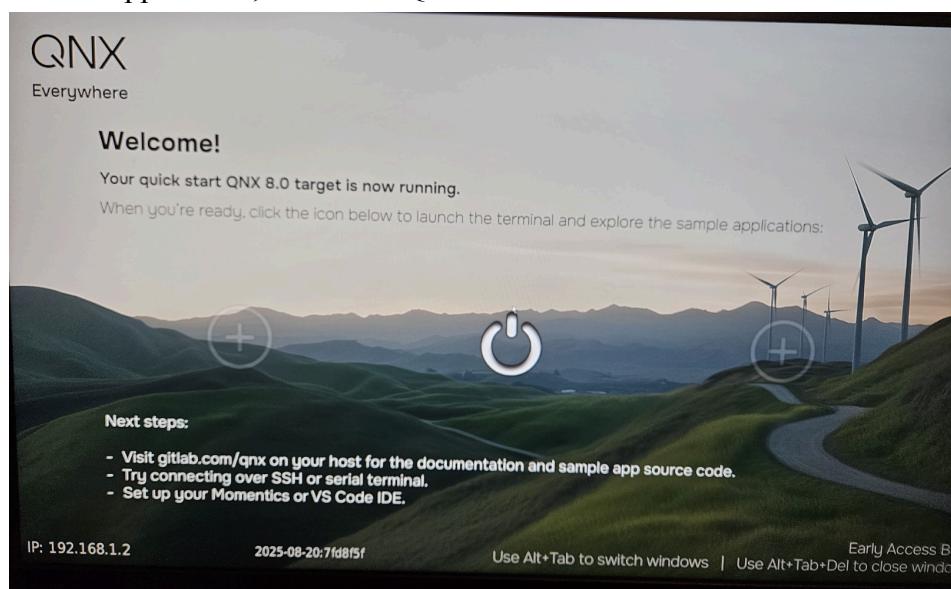


Figure 6: QNX - Ethernet Connection with IP Address Bottom Left

November 16, 2025

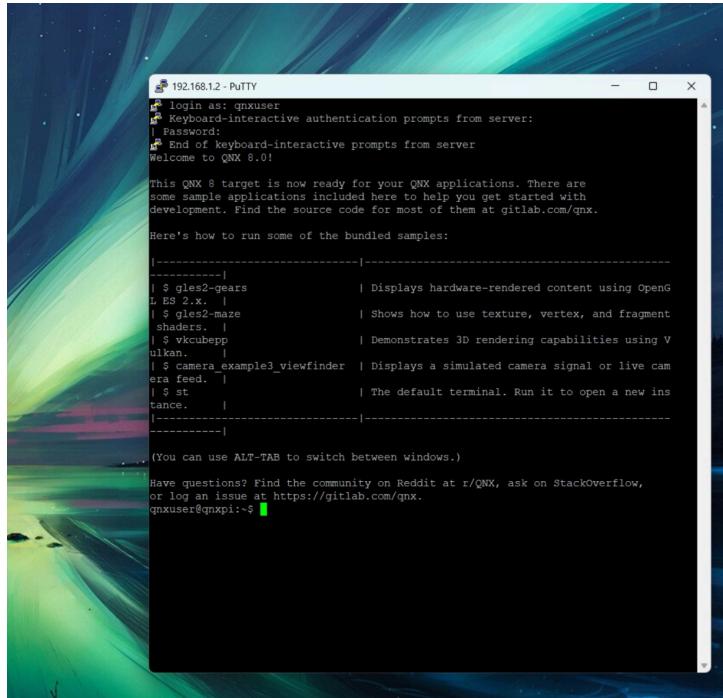


Figure 7: QNX Screen with Ethernet Cable Connection

## 5. Thermal temperature:

The Raspberry Pi 4 thermal temperature can be viewed in the thermal document under the dev folder. Cat the document and there one can view the devices.

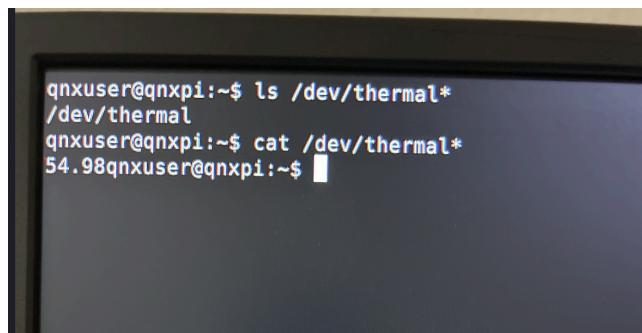


Figure 8: Thermal Screen Capture

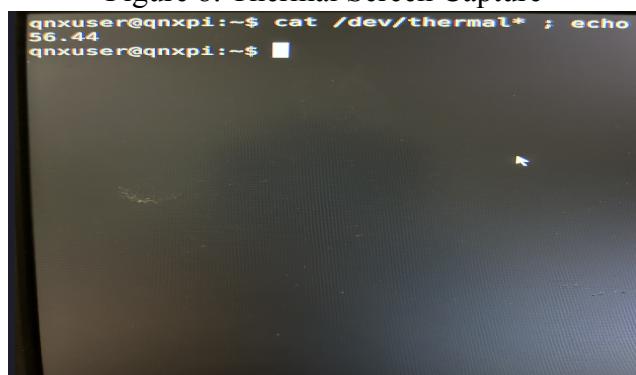


Figure 9: Thermal Screen Capture Using Echo

## 6. G.711 coder/decoder:

The C code for the decoder application is attached to this document along with the generated decoded .wav files.

The decoded .wav file says:

*The ship was torn apart on a sharp reef.  
Sickness kept him home the 3rd week.  
The box will hold 7 gifts at once.  
Jazz and swing fans like fast music*

## 7. How is the behavior of QNX different from Linux?

Linux has a monolithic kernel architecture, while QNX has a Microkernel architecture. QNX provides hard real-time guarantees with bounded interrupt latencies, while Linux provides soft real-time at best when the Linux kernel is patched with the PREEMPT\_RT patch. QNX systems are statically composed with an image filesystem (IFS) and specific startup scripts. Linux boots with an initramfs, dynamic drivers, and systemd.

## **References:**

1. M. Sweetgall, “Using the G711 standard”, Code Project.com  
<https://www.codeproject.com/articles/Using-the-G711-standard#comments-section>  
(accessed Nov. 13, 2025)
2. “PCM A-law and u-law Companding Algorithms in ANSI C”.  
dystopiancode.blogspot.com.  
<https://dystopiancode.blogspot.com/2012/02/pcm-law-and-u-law-companding-algorithms.html>  
(accessed Nov. 13, 2025)