Lab Report for

CE6023 – Computer Vision Systems

Student Name : **Dylan Rodrigues**

Student ID : **24121479**

Revision Timestamp: 16/09/2024 15:42:19

Contents

Introduction

In this lab, we focused on setting up and configuring a Raspberry Pi 4 for network connectivity. The tasks included unboxing and setting up the Raspberry Pi with a keyboard and display, extracting the device's MAC addresses, and configuring it for Ethernet. Each step was essential to ensure the Raspberry Pi was properly initialized and ready for network use. The following sections detail these tasks.

Setting up a Raspberry Pi 4

In this lab, we worked with the **Raspberry Pi 4 [4GB Starter Kit] Model B**. The kit came with several components, each serving a specific function, and we explored how these parts fit together to create a fully functioning system.

1. **Raspberry Pi 4 Model B Board**: This is the central component, essentially a mini-computer. It houses the processor, memory, and various ports needed for connectivity. The model we used comes with 4GB of RAM, sufficient for moderate computing tasks.
2. **MicroSD Card**: The microSD card acts as the primary storage device, holding the operating system and any files required. We inserted the card into the designated slot on the Raspberry Pi 4.
3. **3 x Heat Sinks**: The heat sinks are small metal plates designed to dissipate heat generated by key components, such as the CPU, RAM, and voltage regulators. We applied adhesive heat sinks directly to these components to prevent overheating, especially during intensive tasks. The heat sinks enhance thermal management by increasing the surface area for heat dissipation.
4. **Cooling Fan**: The kit also included a fan to further enhance cooling. We attached the fan to the case’s designated slot and connected it to the appropriate GPIO pins for power. The

fan helps maintain optimal operating temperatures, preventing thermal throttling that can reduce performance, particularly during prolonged or resource-intensive usage.

5. **Power Supply**: The power supply is a USB-C adapter that provides the necessary power to the Raspberry Pi 4. We connected this directly to the board's USB-C port.
6. **HDMI Cables**: The kit includes a micro-HDMI to standard HDMI cable. This is used to connect the Raspberry Pi 4 to a display (monitor or TV) for visual output. The Raspberry Pi 4 features two micro-HDMI ports, supporting dual displays. We used one HDMI cable to connect it to our monitor for displaying the output.

After understanding the function of each component, we followed the assembly steps outlined in the YouTube video "*OKDO Raspberry Pi 4 Kit (8GB Ram) - Unboxing and Step by Step on how to assemble*." We first placed the heat sinks on the specified components and installed the cooling fan inside the case. Then, we inserted the Raspberry Pi board into the case and connected all external peripherals, including the power supply, HDMI cable, and input devices.

Once everything was assembled and powered on, the Raspberry Pi booted into the operating system, confirming that the setup was successful and ready for the next stages of configuration.

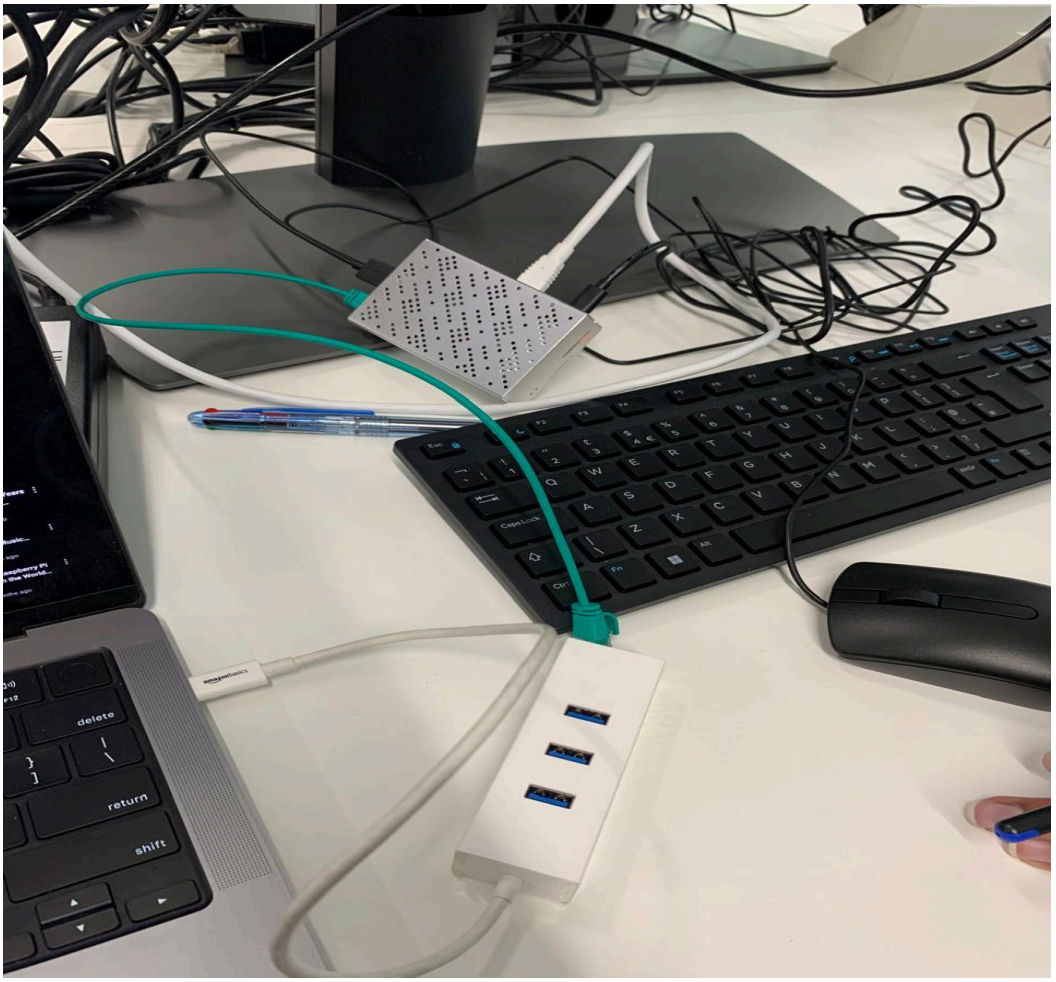

Figure 1: Built Raspberry Pi 4 Kit

Setup using Keyboard and Display

Once the Raspberry Pi 4 was properly assembled, we connected it to a monitor, keyboard, and mouse to begin the setup process. This allowed us to interact directly with the Raspberry Pi's operating system and configure it as needed.

1. **Connecting to the Monitor**:
   The Raspberry Pi 4 includes two micro-HDMI ports, allowing for dual-display output. We used a micro-HDMI to HDMI cable from the kit to connect the Raspberry Pi to a standard monitor. This connection enabled us to view the output from the Raspberry Pi's operating system on the screen.
2. **Keyboard and Mouse**:
   To interact with the Raspberry Pi, we connected a USB keyboard and mouse to its available USB ports. The Raspberry Pi 4 has both USB 2.0 and USB 3.0 ports, providing flexibility for peripheral connections. The keyboard and mouse allowed us to navigate through the operating system's menus and input necessary commands during the setup process.
3. **Powering On and Initial Setup**:
   After connecting the necessary peripherals, we plugged in the USB-C power supply to power on the Raspberry Pi. The device booted up, displaying the Raspberry Pi OS desktop environment on the monitor. The initial boot prompted us to go through a series of setup steps, including:
   - Installation of the Raspberry Pi Operating System
   - Reconfiguring the password
   - Searching for a Wifi network (optional)
   - Software update prompt (optional)
4. **Rebooting the Raspberry Pi**:
   After completing the initial setup and configuration, the system required a reboot to apply changes. The Raspberry Pi rebooted, and once it powered back on, we were able to access the fully configured system via the monitor. At this point, the Raspberry Pi was ready for further use, and we could now begin additional tasks or run applications directly from the connected display.

Extract MAC addresses from the Raspberry Pi

After setting up the Raspberry Pi, we needed to find the MAC addresses for both the Ethernet and Wi-Fi connections.

1. **Open Terminal**:
   We started by opening a terminal window on the Raspberry Pi. This can be done by clicking the terminal icon in the top-left corner of the desktop.
2. **Run the Command**:
   In the terminal, we typed `ifconfig` and pressed Enter. This command showed us the network details of the Raspberry Pi.

3. **Find MAC Addresses**:
   We looked for the sections labeled `eth0` (for Ethernet) and `wlan0` (for Wi-Fi). Under each section, we found the MAC addresses next to the word "ether." Both addresses started with `d8:3a:dd` and were followed by different numbers. We wrote down both MAC addresses on the template paper provided during the lab.
4. **Enter into the Shared File**:
   Lastly, we went to the shared document from the link that Prof. Patrick shared with us and found our student number. We entered our MAC addresses in the columns for “RPi4 Ethernet MAC address” and “RPi4 Wifi MAC address.”

By following these steps, we successfully recorded the MAC addresses of our Raspberry Pi.

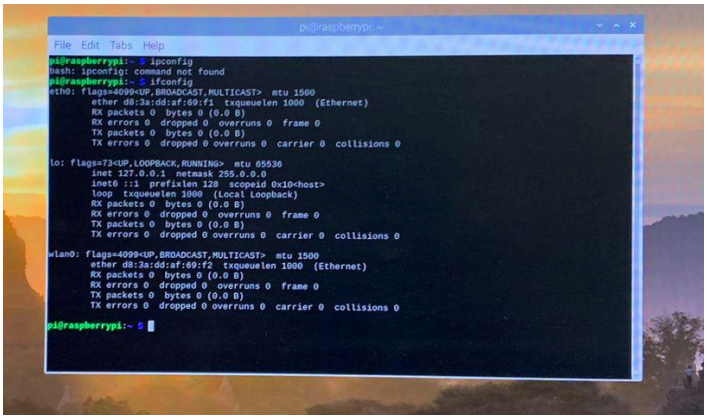

Figure 2: Screenshot of mac address for Ethernet and mac address for Lan

Configure Raspberry Pi Settings

To access the Raspberry Pi Configuration window, we clicked on the Raspberry Pi icon in the top-left corner of the desktop. From the dropdown menu, we selected “Preferences” and then clicked on “Raspberry Pi Configuration.” In the configuration window, we navigated to the “Interfaces” tab, where we enabled both the Camera and SSH options by selecting the corresponding radio buttons to “Enabled.” After making these changes, we clicked “OK” to save the settings and close the window. This allowed us to use the camera and remotely access the Raspberry Pi via SSH.

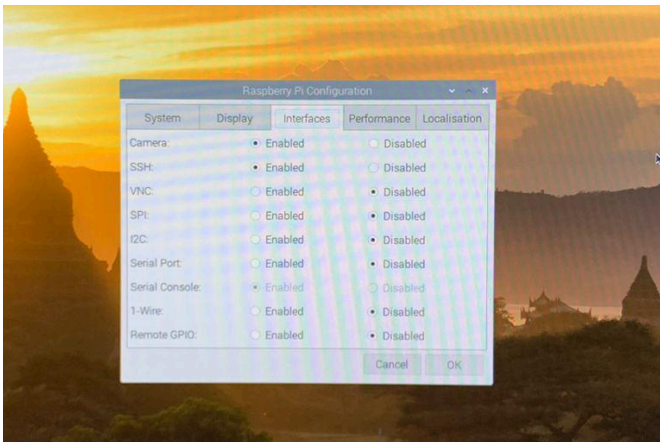

Figure 3: Raspberry Pi Configuration

Ssh Connection to Raspberry Pi

After connecting our Raspberry Pi 4 to the Ethernet and displaying the interface on the monitor, we successfully initiated an SSH session from our local machine to the Raspberry Pi 4's operating system, allowing for remote access and control.

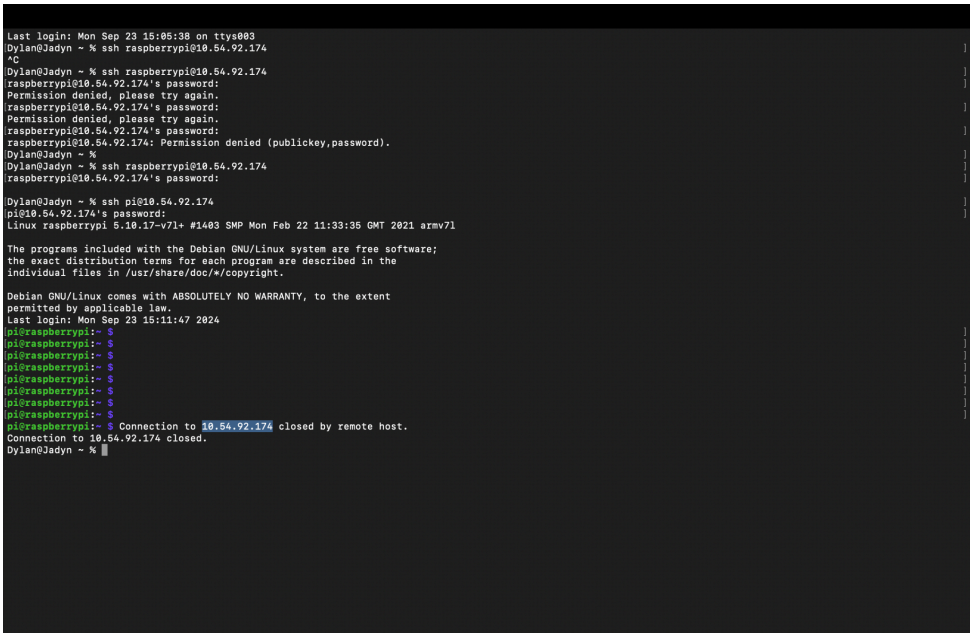

Figure 4: Successful ssh connection to Raspberry Pi 4 from a remote machine

Conclusion

In this lab, we successfully set up the Raspberry Pi 4 Model B by assembling its components, including the heat sinks and fan, and connecting it to a monitor, keyboard, and mouse. We extracted the MAC addresses for both Ethernet and Wi-Fi connections, which are crucial for network identification. Additionally, we configured essential settings by enabling the camera and SSH for remote access. Overall, this experience enhanced our understanding of Raspberry Pi hardware and software integration, laying the foundation for future projects and applications.