# System Documentation

## Overview

This documentation provides a detailed description of the system setup for a Raspberry Pi, including the hardware and software components, directory structure, and usage instructions. The setup includes updating the system, installing necessary dependencies, configuring an OLED display, enabling various interfaces, and setting up remote access and monitoring tools.

### Hardware Components

1. **Raspberry Pi**: The main computing device.
2. **OLED Display**: Used for displaying system information.
3. **NVMe SSD**: Used for storage and booting the system.
4. **PCIe Interface**: Used for connecting the NVMe SSD.
5. **Hailo AI Processor**: Optional hardware for AI processing tasks.

### Software Components

1. **Operating System**: Raspberry Pi OS (based on Debian).
2. **Python 3.12.3**: Installed for running Python scripts.
3. **Git**: Version control system.
4. **luma.oled**: Python library for controlling OLED displays.
5. **Hailo Software**: Includes Hailo drivers and tools for AI processing.
6. **dhcpcd**: DHCP client daemon for network configuration.
7. **xrdp**: Remote Desktop Protocol server.

### Directory Structure

The following directory structure is used in the setup:

/usr/src/

Python-3.12.3/ # Directory for Python source code

/opt/luma/

luma.examples/ # Cloned repository for OLED examples

/home/pi/toshiba/ # Mount point for NVMe SSD

/home/\_GitHub/External/hailo-rpi5-examples/ # Directory for Hailo examples

## System Setup

### 1. System Update and Upgrade

The system is updated and upgraded to ensure all packages are up-to-date:

sudo apt update && sudo apt full-upgrade -y

### 2. Python Installation

Build tools and dependencies for Python are installed, followed by downloading and installing Python 3.12.3:

sudo apt install -y build-essential libssl-dev libffi-dev zlib1g-dev libsqlite3-dev libbz2-dev libreadline-dev libncurses5-dev libgdbm-dev libnss3-dev liblzma-dev uuid-dev wget

cd /usr/src

sudo wget https://www.python.org/ftp/python/3.12.3/Python-3.12.3.tgz

sudo tar xzf Python-3.12.3.tgz

cd Python-3.12.3

sudo ./configure --enable-optimizations

sudo make -j$(nproc)

sudo make altinstall

### 3. Git Installation

Git is installed for version control:

sudo apt install -y git

### 4. OLED Display Setup

Necessary system packages for the OLED display are installed, and the luma.examples repository is cloned:

sudo apt install -y python3-dev python3-pip python3-numpy libfreetype6-dev libjpeg-dev build-essential

sudo apt install -y libsdl2-dev libsdl2-image-dev libsdl2-mixer-dev libsdl2-ttf-dev libportmidi-dev

sudo apt install -y python3-luma.oled

sudo mkdir -p /opt/luma

sudo chown $USER:$USER /opt/luma

cd /opt/luma

git clone https://github.com/rm-hull/luma.examples.git

A systemd service is created for the OLED display script:

SERVICE\_FILE="/etc/systemd/system/oled\_display.service"

sudo bash -c "cat > $SERVICE\_FILE <<EOF

[Unit]

Description=OLED Display Service

After=multi-user.target

[Service]

ExecStart=/usr/bin/python3 /opt/luma/luma.examples/examples/sys\_info\_extended.py

WorkingDirectory=/opt/luma/luma.examples/examples

Restart=always

User=$USER

Group=$USER

[Install]

WantedBy=multi-user.target

EOF"

sudo systemctl daemon-reload

sudo systemctl enable oled\_display.service

sudo systemctl start oled\_display.service

### 5. Interface Enabling

Various interfaces are enabled using raspi-config:

sudo systemctl enable ssh

sudo systemctl start ssh

sudo raspi-config nonint do\_vnc 0

sudo raspi-config nonint do\_spi 0

sudo raspi-config nonint do\_i2c 0

sudo raspi-config nonint do\_serial 0

sudo raspi-config nonint do\_serial 1

sudo raspi-config nonint do\_onewire 0

**6. PCIe and NVMe Setup**

Functions are provided to enable the PCIe interface and set up NVMe power monitoring:

# Enable PCIe Interface

echo "dtparam=pciex1" | sudo tee -a /boot/firmware/config.txt

sudo reboot

# Set up NVMe Power Monitoring

wget https://files.waveshare.com/upload/0/06/PCIE\_HAT\_INA219.zip

unzip -o PCIE\_HAT\_INA219.zip -d ./PCIE\_HAT\_INA219

cd PCIE\_HAT\_INA219

sudo python INA219.py

The NVMe SSD is partitioned, formatted, and mounted:

sudo fdisk /dev/nvme0n1 <<EOF

n

p

1

w

EOF

sudo mkfs.ext4 /dev/nvme0n1p1

sudo mkdir -p /home/pi/toshiba

sudo mount /dev/nvme0n1p1 /home/pi/toshiba

Auto-mounting and booting from the NVMe SSD are set up:

echo "/dev/nvme0n1p1 /home/pi/toshiba ext4 defaults 0 0" | sudo tee -a /etc/fstab

sudo mount -a

sudo rpi-eeprom-config --edit <<EOF

NVME\_CONTROLLER=1

BOOT\_ORDER=0xf416

EOF

sudo reboot

**7. Hailo Software Installation**

Hailo software is installed and verified:

sudo apt install -y hailo-all

sudo reboot

hailortcli fw-control identify

gst-inspect-1.0 hailotools

gst-inspect-1.0 hailo

Functions are provided for upgrading, downgrading, and troubleshooting Hailo software.

### 8. Network Configuration

dhcpcd is checked and installed if necessary, and static IP addresses are set for eth0 and wlan0:

sudo apt install -y dhcpcd5

sudo systemctl enable dhcpcd

sudo systemctl start dhcpcd

# Static IP configuration for eth0 and wlan0

echo "interface eth0" | sudo tee -a /etc/dhcpcd.conf

echo "static ip\_address=$eth0\_ip/24" | sudo tee -a /etc/dhcpcd.conf

echo "static routers=$router\_ip" | sudo tee -a /etc/dhcpcd.conf

echo "static domain\_name\_servers=$dns\_ip" | sudo tee -a /etc/dhcpcd.conf

echo "interface wlan0" | sudo tee -a /etc/dhcpcd.conf

echo "static ip\_address=$wlan0\_ip/24" | sudo tee -a /etc/dhcpcd.conf

echo "static routers=$router\_ip" | sudo tee -a /etc/dhcpcd.conf

echo "static domain\_name\_servers=$dns\_ip" | sudo tee -a /etc/dhcpcd.conf

Port forwarding, Dynamic DNS, VPN, and Remote Desktop are set up:

# Port forwarding

read -p "Enter the external port for SSH (e.g., 2222): " external\_port

internal\_ip=$(hostname -I | awk '{print $1}')

echo "Please configure your router to forward port $external\_port to $internal\_ip:22"

# Dynamic DNS

echo "Please sign up for a Dynamic DNS service and follow their instructions."

# VPN

echo "Please follow the instructions to set up OpenVPN on your Raspberry Pi: https://pimylifeup.com/raspberry-pi-vpn-server/"

# Remote Desktop

sudo apt install -y xrdp

sudo systemctl enable xrdp

sudo systemctl start xrdp

### Usage Instructions

1. **Running the Script**: Execute the script with appropriate options to perform specific tasks.
2. **Monitoring Services**: Use systemctl status to check the status of services.
3. **Updating the System**: Regularly update the system using sudo apt update && sudo apt full-upgrade -y.
4. **Troubleshooting**: Use provided functions for troubleshooting hardware and software issues.

# Rack Setup Description

This documentation provides a detailed description of the rack setup, including the hardware components and their respective positions within the rack.

## Rack Layout

The rack is organized into several units (U), each containing specific hardware components. Below is the detailed layout:

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| *11U* | 11U |  | | | | | | | | | | | | | | | | | | | | | |
|  |
| *10U* | 10U |  | | | | | | | | | | | | | | | | | | | | | |
|  |
| *9U* | 9U | Function Generator | | | | | |  | | | | | | | | | | | | | | | |
|  |
| *8U* | JDS6600 | | | | | | **EXT IN** | | | | | | **CH1** | | | | | **CH2** | | | | |
|  | 7.5U | Network Switch | | | | | | SW1 | SW2 | | | SW3 | | | SW4 | | SW5 | | SW6 | | SW7 | | SW8 |
| *7U* | SW9 | SW10 | | | SW11 | | | SW12 | | SW13 | | SW14 | | SW15 | | SW16 |
|  | 6.5U |  | | | | | | | | | | | | | | | | | | | | | |
| *6U* | 6U | **SW1** | **SW2** | **SW3** | **SW4** | **SW9** | **SW10** | | | **SW11** | | | **SW12** | | | **SW5** | | **SW6** | | **SW7** | | **SW8** | |
|  |
| *5U* | 5U | **PI5.1** | **PI5.2** | **PI5.3** | **PI5.4** | **PI4.1** | **PI4.2** | | | **PI4.3** | | | **PI4.4** | | | **OSC** | | **DAQ** | |  | | **NWIN** | |
|  |
| *4U* | 4U | **PI5.1** | | | | | | | | | **PI5.2** | | | | | | | | | | | | |
|  |
| *3U* | 3U | **PI5.3** | | | | | | | | | **PI5.4** | | | | | | | | | | | | |
|  |
| *2U* | 2U | **PI4.1** | | | | | | | | | **PI4.2** | | | | | | | | | | | | |
|  |
| *1U* | 1U |  | | | | | | | | | | | | | | | | | | | | | |
|  |

### Unit Positions

* **11U**: Empty
* **10U**: Empty
* **9U**: Function Generator
  + JDS6600
  + EXT IN
  + CH1
  + CH2
* **7.5U**: Network Switch
  + SW1, SW2, SW3, SW4, SW5, SW6, SW7, SW8
  + SW9, SW10, SW11, SW12, SW13, SW14, SW15, SW16
* **6.5U**: Empty
* **6U**: Patch panel – Network switch connections

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **SW1** | **SW2** | **SW3** | **SW4** | **SW9** | **SW10** | **SW11** | **SW12** | **SW5** | **SW6** | **SW7** | **SW8** |
|

* **5U**: Patch panel – Device connections

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **PI5.1** | **PI5.2** | **PI5.3** | **PI5.4** | **PI4.1** | **PI4.2** | **PI4.3** | **PI4.4** | **OSC** | **DAQ** |  | **NWIN** |
|

* **4U**: Raspberry Pi
  + PI5.1, PI5.2
* **3U**: Raspberry Pi
  + PI5.3, PI5.4
* **2U**: Raspberry Pi
  + PI4.1, PI4.2
* **1U**: Empty

## Network Switch: Zyxel GS1100-16v3

The Zyxel GS1100-16v3 is a 16-port Gigabit Ethernet unmanaged switch designed for high-bandwidth network applications. It is suitable for office environments where silent operation and energy efficiency are important.

### Technical Specifications

* **Ports:** The switch has 16 Gigabit Ethernet (GbE) RJ-45 ports. These ports support auto-negotiation for speed and duplex mode, and auto MDI/MDI-X for automatic crossover detection.
* **Switching Capacity:** The switch has a switching capacity of 32 Gbps, which allows for full wire-speed forwarding on all ports.
* **Forwarding Rate:** The forwarding rate is 23.8 million packets per second (Mpps), ensuring efficient handling of network traffic.
* **Packet Buffer:** The switch includes a 525 K byte packet buffer to manage data traffic and prevent packet loss during high traffic periods.
* **MAC Address Table:** The switch supports up to 8,000 MAC addresses, which is sufficient for most small to medium-sized networks.
* **Jumbo Frame Support:** The switch supports jumbo frames up to 9 K bytes, which can improve performance for large data transfers.

### Environmental Specifications

* **Operating Temperature:** The switch operates within a temperature range of 0°C to 40°C (32°F to 104°F).
* **Humidity:** It can operate in environments with humidity levels between 10% and 90% (non-condensing).

### Power Specifications

* **Power Supply:** The switch has an internal power supply that operates on 100-240 V AC, 50/60 Hz.
* **Max Power Consumption:** The switch consumes a maximum of 10 watts.

## Patch Panel: HB-DIGITAL 12-Port

### Specifications

* **Brand:** HB-DIGITAL
* **Size:** 10 inches
* **Material:** Metal
* **Ports:** 12 keystone module slots
* **Grounding:** Integrated grounding cable

**Technical Details**

* **Port Configuration:** The patch panel supports up to 12 keystone modules
* **Compatibility:** Compatible with keystone modules, Ethernet cables, Cat cables (Category cables), twisted pair cables, RJ45 cables, installation cables, patch cables, network cables, LAN cables, copper laying cables, home networks, NAS devices, and switches.
* **Strain Relief:** The patch panel includes cable tie points for securing cables, ensuring effective strain relief and preventing cable damage.
* **Grounding:** The integrated grounding cable ensures safe operation by providing a reliable grounding path, reducing the risk of electrical interference and improving overall network stability.