



EXPERIENCE SUMMARY

EXPERIENCE LEVEL- 1.8 year

Now: Miraфра Technologies Pvt Ltd. March 2025 – Present

Client: Qualcomm, Hyderabad

Past: IQLabs, Bengaluru. April 2024 – November 2024.

ACADEMICS:

B. Tech in ECE from College of Engineering Chengannur, (Govt), 2019 - 2023, KTU University (CGPA : 8.38)

12th from PMGMHSS , Palakkad, Kerala, Board Of Higher Secondary Examination (Govt of Kerala) – 91%

TECHNICAL SKILLS:

- **Programming Languages:** C, C++(OOPS), Data Structure
- **Operating System:** Windows, Linux
- **Tools:** VSCode, Git, STM32cube IDE, Arduino IDE
- **Debugging Tools :** JTAG, GDB, valgrind
- **Microcontrollers :** ESP32, ESP8266, STM32.
- **Protocols:** UART, SPI, I2C, MQTT.
- Good Understanding in **Linux device drivers and Kernel Programming**
 - ✓ **Kernel Programming:** Writing Linux kernel modules and device drivers
 - ✓ Developed and implemented Linux kernel modules for:
 - Character device drivers (registering device numbers, file operations, etc.), platform driver
 - ✓ **Memory Management:** Experience with Kernel memory allocation techniques such as kcalloc and buffer management, and hardware address mapping using ioremap
 - ✓ **Synchronization Mechanisms:** Mutex, Semaphores
 - ✓ **Device Tree:** configuration and integration for device driver
 - ✓ **Interrupt Handling:** Managing interrupt (Top half, Bottom half)
 - ✓ **IPC Mechanisms :** pipes, Shared memory, Message Queue
 - ✓ **RTC Driver Development:** Developed a **custom RTC driver** on the **Raspberry Pi** platform to implement the **basic RTC functionalities** such as setting and getting time.
 - ✓ **ALSA framework**
 - "ALSA (Advanced Linux Sound Architecture), ALSA API, Audio Programming".
 - Understanding of **ALSA** architecture from user space to kernel space (PCM, device drivers,).

PROJECTS:

Company Name: Miraфра Technologies Pvt Ltd, Hyderabad

Role: Software Engineer I

Client: Qualcomm, Hyderabad

Project 1: Asynchronous UART-Driven LED Blinking Kernel Module(ioremap)

Platform: Raspberry Pi 4B, Linux Kernel 6.12+

- Handles UART RX data using asynchronous design.
- Blinks an LED on GPIO based on incoming UART characters.
- Written as a custom Linux kernel module in C.

- Uses gpiolib for safe and abstracted GPIO control.
- Uses ``ioremap()`` to map physical GPIO base address to virtual memory for direct register access
- Designed for Raspberry Pi 4B (BCM2711) using physical GPIO and UART base addresses.
- Designed a Linux character device and integrated ALSA in user space to perform .wav audio playback on trigger events.

Project 2: Clock Consumer Driver Development (Qualcomm SM8150 Platform)

- Developed a clock consumer (client) driver using Linux Common Clock Framework (CCF) APIs for the **Qualcomm Snapdragon 855 platform**
- Enabled dynamic clock management by controlling clock states and frequencies based on subsystem requirements
- Translated API-level requests into hardware-level register configurations, including: RCGs (Root Clock Generators)
- Performed comprehensive testing of all consumer APIs, ensuring correct functionality, stability, and proper clock propagation

Company Name: Indigo Quotient Labs Pvt Ltd, Bengaluru.

Role: Embedded Engineer.

Platform:- Arduino, VS Code, STM32CubeIDE

WATER PURIFIER BOT.

- Developed, and optimized embedded firmware for a smart **Water Purification System**, enhancing operational performance and reliability through real-time diagnostics, iterative testing.
- Integrated **ESP8266** and **ESP32** microcontrollers for wireless control and monitoring, using **MQTT** protocol for reliable communication.
- Interfaced multiple peripherals via **I2C**, ensuring efficient communication between components including flow sensors.

Implemented features such as:

- **Remote pump control using flow-rate calculations** – designed logic to calculate dispensed water from sensor pulse counts; dispensing allowed only when validity is active and total volume is below the flow limit, with automatic pump shut off once the threshold is reached.
- **Validity tracking using EEPROM and RTC** to ensure accurate monitoring of usage duration.
- **OTA (Over-The-Air) firmware updates** for dynamic feature deployment and system maintenance.
- Collaborated directly with clients to gather and implement specifications, ensuring tailored feature sets and seamless system integration.
- Developed a modular codebase using **C/C++**, enhancing maintainability and scalability for future product versions.