

## Internal Project

### **Multi-Client Socket LED Control System with Audio Alert and OLED Interface**

Developed an embedded alert coordination system for industrial and hospital applications.

Raspberry Pi acts as the central TCP server, dynamically assigning roles to two x86 clients.

Client 1: LED blink (visual alert) Client 2: Audio beep (audio alert). Integrated OLED display (I2C) for real-time monitoring. Measured Round-Trip Time (RTT) for alert acknowledgment to ensure timely and reliable alert delivery.

**Debugging Tools:** dmesg, printk (for logs), Raspberry Pi 4, x86 Systems, OLED (I2C), TCP/IP Socket Programming, C, GCC, Linux (Raspberry Pi OS / Ubuntu).

## LDK Training Completion Work Summary

Here is the summary of progress I have made after completion of LDK Training (02-09-25) to present day.

### **1. Platform Drivers**

Gained clear understanding of the platform bus and platform drivers. Implemented and tested basic drivers for GPIO, I2C, UART, RTC, and PCI. Utilized debugging tools like for driver testing and analysis.

**Debugging Tools:** printk, ftrace, ltrace, dmesg, sysfs

### **2. Crash Dump Analysis**

Explored the differences between OOPS and Kernel Panic. Intentionally triggered OOPS and panic conditions in a virtual environment using driver code. Collected and analyzed crash logs with kdump, kexec, and crash tools to understand kernel fault handling.

**Debugging Tools:** crash utility, kernel panic logs, oops reports

### **3. Memory Leak Analysis**

Created memory leak scenarios in both user space and kernel space. Analyzed logs and reports using kmemleak, valgrind, and AddressSanitizer. Understood best practices for leak detection and prevention in system programming.

**Debugging Tools:** Valgrind reports(user\_space), kmemleak logs(kernel\_space)

### **4. U-Boot Exploration**

Studied the booting process of Raspberry Pi using U-Boot. Understood key environment variables like bootdelay, bootcmd, bootargs, loadaddr, fdtaddr . Explored U-Boot commands including printenv, setenv, bootm, bootz and their role in kernel and DTB loading.

**Debugging Tools:** U-Boot console logs, serial monitor

## 5. Yocto Project

Understood the need for Yocto and its key terminology: bitbake, recipes, poky, layers. Built a minimal image for Raspberry Pi 4 with customized configuration. Successfully flashed and booted the image on the Raspberry Pi board.

**Debugging Tools:** dmesg, journalctl, dd command

## 6. Displaying Images on ILI9225 TFT LCD

Displayed images on a 176x220 ILI9225 TFT LCD using a Raspberry Pi. I converted images to RGB565 format and sent the data via SPI, while controlling the display with GPIO pins.

**Debugging Tools:** logic analyzer, SPI communication logs

## 7. Device Tree & BSP Log Analysis

Explored Board Support Package (BSP) components and integrated custom driver code. Implemented a Device Tree Source (DTS) file for hardware customization. Used pstore, dmesg, and journalctl tools for kernel crash log analysis and debugging.

**Debugging Tools:** dmesg, journalctl, pstore

## Additional Practice

- Practiced **LDD topics MCQs** and **general critical thinking questions**.
- Performed **R&D on JTAG and SWD debugging techniques** and hands-on work with **GDB, dmesg, pstore, journalctl**.