

Neesu Bala Rajendra

Email: rajendraneesu5501@gmail.com

Mobile: +91-8247303988

PROFESSIONAL EXPERIENCE

Having nearly 3 year of Experience in the Embedded Software and Testing of Embedded Applications, Design and Implement Software for Embedded Devices, Debugs and test the code in **Argus Embedded Systems**.

- Hands of experience in Firmware development of **STM32, ESP32, nRF5340, nRF5340 Audio and Raspberry Pi**
- Hands on Experience in Programming languages: **C**
- Knowledge in Software development life cycle processes.
- Hands on Experience in Different IDE: **Simplicity studio, nRF connect with VS code, ESP-IDF and ARDUINO**
- Good Knowledge On-Board Communication Protocols **UART, I2C, SPI, ADC, BLE**.
- Good in handling **Debugging tools, Logic Analyzer** and testing equipment.
- Hands on experience on **Free RTOS, Zephyr RTOS and Bare Metal**.

Educational Qualification

- ❖ Master of Technology in Embedded systems from RGM college of Engineering and Technology.
- ❖ Bachelor of Technology (B.Tech) in ECE from AVR and SVR Engineering college
- ❖ Intermediate from GSR and PSR Junior College with 82%.
- ❖ SSC from GOVT High School with 77%.

Projects

- **Project Name:** B.Tech IoT

Software IDE: Arduino IDE

Hardware: ESP32, Current sensor (HSTS21), Fingerprint sensor (GT-511C3), TFT LCD display, EG95

Description: When a user places a finger on the fingerprint sensor, it displays user details on the LCD and sends this information, along with current sensor data, to an MQTT server over Wi-Fi.

Responsibility:

- Implanted HSTS21 current sensor data acquisition.
- Developed ESP32 to PC communication** over Wi-Fi to facilitate adding new Wi-Fi credentials dynamically.

- **Project Name: Kitchenery**

Software IDE: Arduino IDE

Hardware: Two ESP32 boards (TX and RX), communicating via BLE, Current sensor(ACS712) and temperature.

Description: This project is a kitchen appliance control system where the TX board controls the RX board, handling both induction base control and wireless power transmission via BLE, UART, and voice commands. It also sends current sensor and temperature data to AWS. If the Wi-Fi not available sensor data will in SPI-Flash. Once the Wi-Fi is connected those data transmit to AWS.

Responsibility:

- Implemented AT commands, BLE commands, and voice control (Alexa) for communication between the transmitter and receiver devices.
- Implemented and integrated the ACS71240 current sensor and a temperature sensor on both transmitter and receiver devices
- Developed frequency tuning mechanisms for handling wireless power transmission between the transmitter and receiver devices.
- Implemented RTC and the MAX6070 I/O expander for handling peripheral controls.
- Implemented and integrated SPI-Flash (W25Q32JV) to store sensor data locally when Wi-Fi is unavailable. Upon reconnection, the stored data is transmitted to AWS.

- **Project Name: nRF5340 Mini CPS**

Software IDE: nRF Connect extension with VS Code.

Hardware: nRF5340 (TX), nRF5340 DK (RX), IMU sensor (LSM6DSV16X), laser sensor (VL53L0X V2), ultrasonic sensor (CH101), digital mic (STEVAL-MIC003V1)

Description: Transmitting sensor data and audio data from the nRF5340 to the nRF5340 DK over BLE, based on user commands. The sensor data is displayed in a Python window, and the audio is heard on a headset.

Responsibility: I implemented the laser sensor, IMU sensor, digital mic, and BLE communication between the nRF5340 DK and nRF5340 DK.

- **Project Name: nRF5340 Audio Mini CPS**

Software IDE: nRF Connect with VS Code.

Hardware: nRF5340 (TX), nRF5340 Audio DK (RX), IMU sensor (LSM6DSV16X), laser sensor (VL53L0X V2), ultrasonic sensor (CH101) and digital mic (STEVAL-MIC003V1)

Description: Transmitting audio data and sensor data from the nRF5340 Audio to the nRF5340 Audio DK over BLE, The sensor data is displayed in a Python window, and the audio is heard on a headset.

Responsibility: I implemented custom device trees for the laser sensor, IMU sensor, and BLE communication between the nRF5340 DK and nRF5340 Audio DK.

- **Project Name: Smart-Hub**

Software IDE: VS Code with ESP-IDF extension

Hardware: ESP32, Raspberry Pi Pico, TFT LCD display, Push buttons, RGB LEDs, Current sensor (ACS7140), BLE OTA, USB OTA, Debug port

Description: Transmit RGB sensor data from the ESP32 to Raspberry Pi Pico over RS232 and update the color sensor data on the ESP32 via USB OTA and BLE OTA.

Responsibility: I was responsible for smoothing the RGB LEDs and managing BLE OTA firmware updates.

- **Project Name: Pico-Hardmat**

Software IDE: VS Code with ESP-IDF extension

Hardware: ESP32, Raspberry Pi Pico, Temperature sensor, LEDs with hard mat, Current sensor (ACS7140), Debug port

Description: Transmit RGB LED data from the ESP32 to Raspberry Pi Pico over RS232, ensuring data integrity and safe temperature levels during transmission to two Raspberry Pi Pico boards.

Responsibility: I was responsible for communication between the ESP32 and Raspberry Pi, and retrieving sensor data from the ESP32 over RS232.

- **Project Name: Silicon**

Software IDE: Simplicity Studio

Hardware: EFM8UB2 Universal Busy Bee, ESP32, TFT LCD, Joystick, Temperature sensor, BMP280, and MQTT

Description: Display sensor data on the LCD and send it to the ESP32 via UART. The ESP32 then transmits the data to an MQTT server over Wi-Fi.

Responsibilities:

- Implemented BMP280 sensor data and TFT LCD.
- Established communication between the EFM8UB2 Universal Busy Bee and ESP32 for transmitting sensor data, which is then forwarded to AWS.

Hobbies

- Web browsing
- Learning new technologies

Declaration

I hereby declare that the above information is true to the best of my knowledge.