Gandla Bhargavi

Bangalore, India 560089 ◆ +91 8778731437 ◆ gandlabhargavi6@gmail.com

Websites, Portfolios, Profiles

- github.com/bhargavi-gudur
- Bhargavi Gandla - gandlabhargavi6 | HackerRank

Professional Summary

Aspiring Embedded Software Engineer with 4 years of industry experience in automotive and telematics domains. Gained hands-on exposure to bare-metal programming, bootloader development, and basic unit testing using G-Test. Currently learning and practicing Embedded C, C++, and Python with STM32 (ARM Cortex-M4) microcontrollers. Familiar with CAN protocol, Agile methodology, and simulation-level testing for Battery Management Systems (BMS). Proactively building skills through personal GitHub projects using Keil, STM32CubeIDE, deepen embedded systems knowledge.

Work History

Senior Software Engineer, 05/2023 - 10/2024
TATA ELXSI – Bangalore

- Developed unit tests using G Test for GNSS and V2X modules in telematics systems.
- Conducted static analysis and ensured line coverage in compliance with automotive coding standards.
- Performed geofence feasibility analysis using GeographicLib-2.3 and simulation-based testing.
- Used Linux virtual machine with VS Code for feature implementation and debugging.
- Followed Agile development practices and tracked tasks/bugs using Jira.
- Created and maintained requirement documents using Tata Elxsi internal templates.
- Delivered real-time telematics alerts such as overspeed, low battery, and refuel detection.
- Contributed to improved documentation and mentored team members to enhance team productivity.

Firmware Engineer - Embedded Systems, 10/2018 - 03/2021
GREEN CUBES TECHNOLOGY – Bangalore

- Developed firmware for Battery Management Systems (BMS) using Embedded C on STM32F series microcontrollers.
- Ported and implemented bootloader features including memory mapping and flash handling.
- Conducted simulation and frontend-based testing to monitor battery parameters (voltage, current, SOC) using internal software after DFU file uploads.
- Used Bus Master tool for CAN protocol validation and implemented basic hardcoded CAN message testing.
- Created and executed test plans using MS Word, performed real-time monitoring and verified system behavior under various conditions.
- Managed source code using Git Extensions and later migrated to Plastic SCM for version control.
- Followed Agile methodology, tracked issues and progress using Jira.
- Contributed to battery reliability improvements through debugging, cell balancing, and safety checks.

Skills

- **Languages** Embedded C, C++, Python (Basic)
- **Protocols** CAN, UART, SPI, I2C
- **Microcontrollers** STM32 (ARM Cortex-M4)
- **Tools** Keil uVision, STM32CubeIDE, VS Code, Git, BusMaster, Terminal (Bash), Code::Blocks

- **Testing** GTest, Manual Testing, Static Analysis, Line Coverage
- **Development Practices** Agile Methodology, Jira, Git, Plastic SCM
- **Platforms** Linux Virtual Machine, Windows
- **AI Tools** GitHub Copilot, ChatGPT

Education

M.Tech: Automotive Electronics, 06/2018
Vellore Institute of Technology (VIT) - Vellore
GPA: CGPA: 8.15/10

B.E: Electrical and Electronics Engineering, 06/2014
SCSVMV Deemed University - Kanchipuram
GPA: CGPA: 8.67/10

Certifications

- Embedded Systems & Automotive - Udemy
- C Programming - Great Learning
- Python Programming - Great Learning
- The Complete Git Guide - Udemy
- GitHub Copilot / ChatGPT for Developers - Great Learning
- Problem Solving (Basic) - Hacker Rank, 04/01/25
- Number System & Intro to Programming - Log2Base2

Github Projects

- **STM32 Bare-Metal Programming – Keil IDE**
  https://github.com/bhargavi-gudur/STM32_Embedded_C.git
- **STM32CubeIDE + Mbed C Integration (Beginner-level)**
  https://github.com/bhargavi-gudur/STM32CubeIDE_MbedC_C.git
- **C Programming Fundamentals**
  https://github.com/bhargavi-gudur/C_BasicCode.git
- **C++ Fundamentals with CMake + GTest**
  https://github.com/bhargavi-gudur/CPP_CMAKE__GTEST.git
- **Python Basics**
  https://github.com/bhargavi-gudur/BasicPythonConcepts.git

Ai Tools Integration

Assisted in auto-generating clean code, comments, and commit messages, Used for design validation, logic clarification, and bug diagnosis