SAIKIRAN GALIPELLY

**Mobile: +919700004249 Email:** [saikirangalipelly23@gmail.com](mailto:saikirangalipelly23@gmail.com)

* **2+ years** of experience as a Embedded Software Engineer
* Having good work experience in programming with **C, Embedded C, Autosar.**
* Working knowledge of Serial Protocols: **I2C, SPI, UART** and **CAN**
* Worked on Embedded peripherals: **GPIO, GPT, PWM, ADC, EEPROM** and **DAC**.
* Knowledge of reading schematics and data sheets for components.
* Ability to develop the low-level firmware drivers referring chip manuals/ datasheets.
* Quick learner with the ability to grasp new technologies and software.

# Academic Profile:

* **B. Tech** EEE graduate from Matrusri engineering college, Hyderabad in 2018 with 60%.
* **Diploma** in Jawaharlal Nehru Government Polytechnic College, Hyderabad in 2015 with 74.6%.
* **S.S.C** in Krishnaveni Talent High school with 9.2 cgpa.

# Experience:

* Working as an Embedded Software Engineer at **TeamBits Embedded Systems & Solutions**, Hyderabad since April 2022.

# Technical Skills:

* Language Proficiency : C, Embedded C.
* OS Platforms : Windows, Linux (Ubuntu)
* Micro Controllers : MCU 8051 , NXP LPC11C24 (32-bit Arm Cortex M0)
* Dev Tools : Keil ARM, ULINK2 Debugger, VsCode, FlashMagic, hTerm, Logic Analyzer, CANalyzer.
* Unit test Tools : Ceedling Framework (GCC, Unity, GCOVr), Notepad++

# Personal Details:

* Name : Saikiran Galipelly
* Father’s Name : Mohan
* Date of Birth : 23-07-1997
* Native Place : Karimnagar, Telangana

# Project # 1: Multi IO test through CAN bus

Microcontroller : LPC11C24

Software Layer(s) : Chip Level, Board Level and Application

OS : SysTick driven Task Scheduler (10ms, 20ms and 50ms)

Peripherals : UserKeys, LCD, Rotary, POT

Tools : Keil IDE, Flash Magic, ULINK2 and Canalyzer

# Description*:*

There are 2 nodes connected on a CAN bus: ECUA and ECUB.

ECUA tracks all the inputs at every 10ms and constructs a CAN message and trasmists at every 50ms. ECUB receives the CAN message with acceptance filter, processes the data to display on the LCD.

# Roles and Responsibilities

* + Develop an application program for both the ECUs with the provided MCAL drivers.
  + Flashed the hex files in both the ECUs using Flash Magic tool or ULINK2 debugger.
  + Connect the CANalyzer tool on the CAN bus to PC to monitor the CAN messages.
  + Test the application in both the ECUs.

# Project # 2: BSOC Indication on RG LED

Microcontroller : LPC11C24

Software Layer(s) : MCAL, Task scheduler and Application Peripherals : Battery terminal (0 to 3.3v), ADC, LCD, RG LED Protocol(s) : UART

Tools : Keil IDE, Flash Magic, ULINK2

# Description:

In this project, voltage across battery terminals is fed to on-chip 10-bit ADC. Analog to digital count is continuous tracked in 20ms task and calculated battery state of charge (BSOC) percentage. Instantaneous BSOC % is displayed on LCD and also sends to PC serial terminal through UART. The current BSOC level is indicated on RG LED.

# Roles and Responsibilities

* + Develop an application program with the ADC, UART and LCD drivers.
  + Flashed the hex file to the ECU using FlashMagic tool or ULINK2 debugger.
  + Test the on-board application.
  + Launch hTerm serial terminal at PC side to monitor the BSOC level.

# Project # 3: AUTOSAR Basic Software (BSW) Development for the Braking Control Module

# (BCM) Electronic Control Unit based on CAN bus communication protocols

# Responsibilities:

* Configuration of AUTOSAR COM stack, DCM and DEM Modules.
* Analyzing and understanding the system requirements, discussing with system team where software requirements are concerned.
* Configuration of CAN and all CAN BSW modules according to customer
* requirements and DBC files with Davinci configurator and developer.
* Performing testing against requirements using CANalyzer.
* Involved in providing lesson learnt, Knowledge sharing session to team members.

# Project #4: Automated Unit Testing

Unit testing is a technique of breaking the code in small units of the entire code. These units can be verified to check the behavior of a specific aspect of the software. Keeping hardware interaction as thin as possible helps in testing most of the code by dividing it into small pieces. These pieces can then be independently tested without hardware interaction. Ceedling is one of the best automation frameworks available for Embedded C software unit testing. Ceedling build system is made up of Rakefiles in Ruby language. Ceedling contains three main utilities (Unity, CMock and CException), which individually contribute to Ceedling functionality.