

PAVAN KUMAR AS

Email: pavangowda63613@gmail.com | Phone Number: 9110808770

LinkedIn : www.linkedin.com/in/pavankumar63 | GitHub : <https://github.com/Pavangowda9731>

CAREER OBJECTIVE

To leverage my strong foundation in C, C++, and Data Structures to contribute effectively to innovative software development and embedded system projects. I aim to apply my technical expertise, analytical thinking, and problem-solving skills in a dynamic environment while continuously learning and growing as a software engineer

WORK EXPERIENCE

- Currently undergoing technical training program – **Advanced Embedded Systems Course** at Emertxe Information Technologies (<http://www.emertxe.com>) Bangalore
- This course is Government of India certified program, aligned with **Skill India** / NSDC under Electronics Sector Skill Council of India (<http://www.essc-india.org>) - **Embedded Software Engineer QP ELE /Q1501**

Embedded Systems Intern - Mindset IT Solutions

- Developed and programmed microcontroller-based embedded systems using Embedded C.
- Interfaced and tested **peripheral modules** such as **LCD, LEDs, buzzer, and keypad** using GPIO and timer concepts.
- Worked on **hardware-software integration**, register-level programming basics, and system debugging.

TECHNICAL SKILLS

- Programming Languages :
 - Advanced C programming
 - OOPS using C++
 - Standard Template Library (STL) in C++
 - Data structures and Algorithm
- Embedded Controllers : PIC18F4580
 - Hands-on working with GPIOs, interfacing, character LCD, Interrupts.
 - Peripherals usage - Timers, ADC, Switches - Digital keypads, matrix keypad, Seven Segment delay(SSD), Pulse Width Modulation(PWM),
 - Communication protocols - UART, SPI, CAN, I2C.

EDUCATION

- B.E (ECE), Rajeev Institute of Technology, Hassan, VTU, **8.73(CGPA)**, 2021-2025
- Class – XII, STATE, **88.67%**, 2021
- Class – X, STATE, **87.36%**, 2019

PROJECTS AT EMERTXE

Project number:1

Title	CAN based Automotive Dashboard
Project brief	This project implements a CAN-based automotive dashboard system using three interconnected ECUs. ECU-2 generates and transmits indicator status and RPM data, while ECU-1 handles vehicle speed and gear position information. ECU-3 receives all CAN messages and displays the consolidated vehicle data on a character LCD. The system demonstrates reliable in-vehicle communication, real-time data acquisition.
Technologies used	Microcontroller - PIC18F4580. Communication protocol – CAN Protocol. Peripherals used – CLCD, ADC, Switches(DKP), LED's.
Key challenges & Learnings	<ul style="list-style-type: none"> ✓ Adjusting the delay was a key challenge to ensure a proper gap after each data transmission, preventing collisions that could result in nothing being displayed on the CLCD. Proper understanding of delay concept. ✓ Assigning different CAN message IDs to each ECU was a challenge. I learned how to give higher priority to important data like speed and RPM so that it is transmitted correctly and displayed without any data loss.

Project number:2

Title	Huffman Based ZIP like File Compress and Extract Tool In C
Project brief	The objective of this project is to design and implement a file compression and decompression tool similar to ZIP utilities using the Huffman coding algorithm. The tool reduces file size by assigning shorter binary codes to frequently occurring characters and longer codes to less frequent ones, thereby achieving efficient lossless compression.
Technologies used	C Programming, File Handling, Data Structures (Binary Trees, Queues), Algorithms, Bit-level Operations
Key challenges & Learnings	<ul style="list-style-type: none"> ✓ Implementing correct decoding was challenging because Huffman codes do not have fixed lengths. I learned how traversing the Huffman tree bit-by-bit ensures accurate reconstruction of the original file without data loss. ✓ Managing dynamic memory for tree nodes and frequency tables required careful handling to avoid memory leaks, strengthening my understanding of pointers and dynamic allocation in C.

Project Number:3

Title	Inverted search Engine
Project brief	An inverted search is an indexing data structure. The purpose of storing an index is to optimize speed and performance in finding relevant documents while searching. Without an index, the search engine would have to scan every document in its entire collection of files or text data, which would require lot of time and computing power.
Technologies used	Advanced C and Data Structure – Hashing, Single linked lists, File concepts, Pointer operations, Self-referential structure, Dynamic Memory Allocation(DMA), Command line Arguments.
Key challenges & Learnings	<ul style="list-style-type: none"> ✓ Since I used four self-referential structures, handling the main-node and sub-node pointers was a key challenge. Deciding how these nodes link to each other and ensuring correct pointer relationships helped me understand multi-level linked lists. ✓ While implementing the update database function, handling duplicates from saved file was bit challenging.

Project number:4

Title	Arbitrary Precision Calculator (APC)
Project brief	The aim of this project is to create an Arbitrary Precision Calculator (APC) that functions like a normal calculator but provides results for complex calculations with higher precision. The goal of this project is to implement basic arithmetic operations such as addition, subtraction, multiplication, and division with two large numbers.
Technologies used	Advanced C and Data Structure – Doubly linked lists, Pointers, Command Line arguments.
Key challenges & Learnings	<ul style="list-style-type: none"> ✓ During the implementation of the Division function, I encountered a segmentation fault due to an incorrect implementation of the function that removes leading zeros after subtraction. ✓ While fixing this error, I learned that since we are using a doubly linked list, we must update the previous pointer of the new head node to NULL before freeing the old node. This ensures that the new head node's previous pointer does not point to the deleted node (which would otherwise create a Dangling pointer). ✓ Multiplication of two big number was challenging as it requires new result list every time for each digit. This problem was solved using instant adding the result.

Project number:5

Title	Image Steganography using LSB (Encoding and Decoding)
Project brief	The objective of this project is to securely transmit secret data Encoded inside a BMP image using an encoding process. The process includes encoding the length of the secret text as well as the text itself into the LSB (Least Significant Bits) of the image bytes. This ensures that the image remains visually unchanged while carrying hidden information. The decoding process involves extracting the encoded length first and then retrieving the hidden text bit by bit.
Technologies used	Advanced C – File operations, Pointers, Structures, bitwise operations, Strings, Command Line arguments.
Key challenges & Learnings	<ul style="list-style-type: none"> ✓ Ensuring correct bitwise manipulation of each byte while encoding the data into LSB was bit challenging. Errors here can corrupt hidden data or produce wrong results. Understanding bitwise manipulation by self understanding. ✓ As multiple files are opened simultaneously(source image, secret file, stego image) Ensuring all files are successfully opened and properly closed to avoid memory leaks or file corruption.

Project number:6

Title	MP3 Tag Reader and Editor
Project brief	MP3 tag reader is a software which will read and display MP3(ID3) tag information from MP3 files. Every MP3 file contains information such as Title, Artist, Album, Year, Genre, Composer, and Comments, stored in a special header region called the ID3 tag. This project focuses on reading that data directly from the binary file and displaying it in a user-friendly format.
Technologies used	Advanced C – File operations, Strings, Structure concept, Pointers, Bitwise operations, Command Line Arguments
Key challenges & Learnings	<ul style="list-style-type: none"> ✓ Handling binary file reading using fread() and fseek() correctly was crucial for me. Even a 1-byte misalignment could cause logical errors. I learned how binary files store data and how to navigate through them using file-handling concepts. ✓ Calculating the frame size correctly was a bit challenging. Incorrect bit shifting or improper handling of unsigned values can lead to huge sizes and buffer overflows. I rectified this problem by gaining a clear understanding of bitwise operations.

Project number:7

Title	Lexical Analyzer In C
Project brief	Developed a foundational Lexical Analyzer in C to process C source code , tokenizing and classifying input into fundamental types like Keywords, Operators, Identifiers, and various Constants (integer, float, octal, binary, hexadecimal, string, and character). The project includes integrated basic error detection for malformed tokens and invalid syntax constructs
Technologies used	C Programming, File Handling (I/O), String Manipulation.
Key challenges & Learnings	<ul style="list-style-type: none"> ✓ Identifying and differentiating between integer, floating-point, binary, octal, and hexadecimal constants required careful character validation and base detection logic, improving my attention to edge cases. ✓ Properly ignoring single-line and multi-line comments while maintaining line and column counts helped improve error reporting and scanner reliability. ✓ Careful pointer handling during token construction prevented buffer overflows and segmentation faults, improving defensive programming practices in C.

Project number:8

Title	Addressbook management system
Project brief	The objective of Address Book Management System is to efficiently store and organize personal or professional contact information. The project provides essential operations such as adding, viewing, searching, editing, and deleting contact records using a simple menu-driven interface. Each record contains details like name, phone number, email, and address, which are stored permanently in a file using file handling functions.
Technologies used	Advanced C – File operations, Pointers, Structures, Strings.
Key challenges & Learnings	<ul style="list-style-type: none"> ✓ As this was my first project preventing data duplication and maintaining data consistency was bit complex. I learnt Proper traversal of loops and comparing the data was done through self understanding ✓ In the initial stage Managing file operations for storing and retrieving data from file was challenging .understood the file operations concepts from self understanding.

ACADEMIC PROJECTS**Project number:1**

Title	Paralysis Patient Health Care Monitoring System
Project brief	Developed an assistive healthcare monitoring system for paralysis patients that continuously monitors vital parameters such as heart rate and body temperature, while also enabling gesture-based communication to convey emergencies or basic needs. The system ensures timely caregiver notification through GSM-based alerts, improving patient safety and response efficiency.
Technologies used	Arduino, Flex Sensor, ADXL (Accelerometer) Sensor, Heartbeat Sensor, Temperature Sensor, GSM Module, Embedded C Programming
Key challenges & Learnings	<ul style="list-style-type: none"> ✓ Integrating biometric sensors (heartbeat, temperature) with motion-based sensors (flex and ADXL) required careful synchronization and calibration, strengthening my understanding of mixed-signal sensor interfacing. ✓ Designing reliable gesture detection using flex and accelerometer data was challenging due to noise and varying hand movements. This improved my skills in analog signal interpretation and threshold-based decision logic. ✓ Balancing power, performance, and responsiveness gave me practical exposure to real-world embedded system constraints.