

# Parth Rajeshkumar Thakkar

Boulder, CO | (303)-548-9344 | [Portfolio](#) | [Linkedin](#) | [Mail](#) | [Github](#)

## EDUCATION

### Masters of Science in Electrical Engineering

University of Colorado, Boulder

Coursework: Principle of Embedded systems, Embedded Systems Design, Internet of Things(IoT)  
PCB, Real Time Embedded Systems

August 2023-May 2025

GPA : 3.85/4

### Bachelor of Engineering in Electronics and Communication

L.D. College of Engineering, Ahmedabad

June 2019-May 2023

GPA : 8.67/10

## WORK EXPERIENCE

### Embedded Developer - Griden Power - An EV Charger Company (September 2022 - August 2023)

- Responsible for designing the IoT part of the charger 100% from the ground.
- Designed the printed circuit boards (PCBs) and contributed to the manufacturing process of the chargers.
- Created, managed, and hosted a portal within the AWS (Amazon Web Services) environment. Integrated freeRTOS with Charger
- Established communication between the chargers and the AWS server using the OCPP and MQTT protocols.
- Developed drivers to integrate with other peripherals like touch LCD, Neoway GSM/GPS module.
- From portal deployment on AWS, integrating embedded components, the charger went live within **four months**.

### Firmware Developer - Scan point geomatics - Indian Geomatics Company (October 2022 - July 2023)

- Elevated the capability of a custom UHF-RFID board by developing a driver that enabled it to effectively read and write RFIDs at an extended distance of **10 feet**, a significant improvement from the initial 5 feet range.
- Worked on Custom UHF reader, ESP32, Touch screen. Enhanced the driver by incorporating algorithms, hash maps that improved its overall performance.
- Designed and Manufactured a full-fledged product, including PCB and 3D printing.

### Hardware Team - Club Robocon (March 2019 - September 2019)

- Deployed bare metal drivers for a robot that represented LD college in DD robocon India.
- Developed drivers for atmega328p including I2C, UART, SPI, Rotary Encoder, PID/Motor Driver, USB, LCD, worked on different boards and pieces of software like stm32 and Raspberry Pi and simulated path that will robot take in the competition in MATLAB, calculated PID for motor in MATLAB.

## SKILLS

**Software :** C, C++, Assembly, VHDL, Python, Docker, AWS, OpenCV, Django Rest Framework, Tensorflow, JavaScript

**Hardware Architectures :** ARM cortex M0/M4 Architecture, AVR Architecture, x86 Architecture, RISC

**Boards :** 6052, 8051, MSP430, KL25Z, LPC2148, Stm32f0/f4, Raspberry pi4, ESP32c3, ATmega328p, ATtiny85

**Communication Protocols :** USB, UART, I2C, SPI, Websockets, TCP/UDP, IoT, RF, HTTPS

**Tools and IDE :** uVision, Git, Eagle, KiCad, MATLAB, Postman, MCUXpresso, Microchip Studio, Stm32cubeIDE, Proteus, Logisim, Digital(Simulation Software)

**OS :** Linux(debian, arch, fedora), Windows, WindowsCE, freeRTOS, Zephyr RTOS

**Skills :** Logic Analyzer, Oscilloscope, Datasheet/Reference Manual, Function Generator, Digital Design, DSA, PCB

## PROJECTS

### Pong in 8051 using Assembly

- Using assembly code, I made a Pong game for the 8051 microcontroller, Integrated a custom-designed PS2 keyboard hardware controller for paddle control and an LED matrix serving as the display.
- Key pushes translate into on-screen action as the software controls paddle motions, ball trajectory, and scorekeeping. The whole code was written in Assembly and compiled in the AS31 assembler.
- Designed BMS to make it battery operated.

### 8-Bit CPU

- Engineered a CPU composed of basic transistor-transistor-logic and logic gates on breadboard, and is capable of executing basic programs such as Fibonacci series, addition, subtraction.
- As an embedded system it is equipped with all necessary core blocks such as Program Counter, ALU, Control Unit, LCD for output register, Register, Instruction Register, Control unit, Clock, Instruction decoder, RAM(16 byte)Address Decoder.

### Simulator in C++ for 6052

- Simulated 6052 in the C++ that can run programs which can be hard coded into virtual memory by the user. Also, it uses the same opcodes and machine cycle to run the instruction.
- Implemented instructions LDA, LDX/Y, STA, STX/Y, JMP, RTS, JSR, BEQ, CMP, TAX, TAY, NOP to run the given code in the simulator. Used Google test suit for C++ to test the Simulation.

### Bare Metal RTOS in Stm32f070

- Developed a Bare Metal RTOS For an ARM Cortex M0+ core (stm32f070). Included a number of features, such as a Scheduler, RTOS with Round Robin, Cooperative, and Periodic Scheduler methods. Semaphores and Mutex were also Implemented.