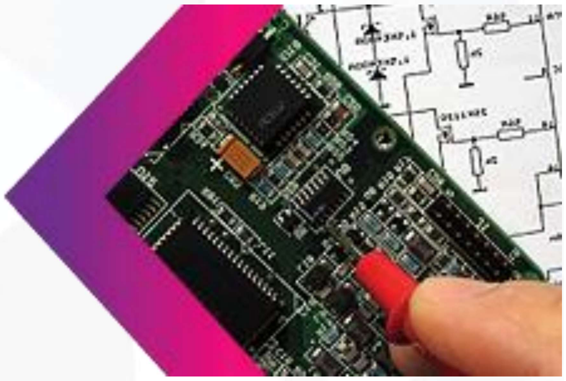
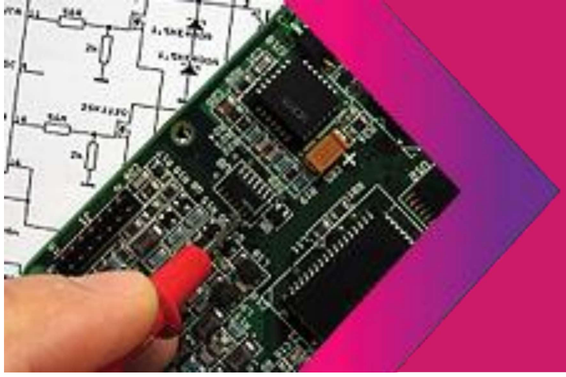
EMERTXE

"Here to take you beyond"

SKILL-SET & PROJECT DETAILS

Trained Embedded Engineers





NOW FOR **SIMPLIFIED** SOLUTIONS

Embedded Systems: Skill-sets

- **Objectives:**

  ✓ Upskill entry level engineers by taking hands-on approach
  ✓ Enable them to work on projects, evaluate them and showcase in profile for hiring
  ✓ Provide exposure to development & process tools
  ✓ Foster teamwork and collaborative approach

- **Skill-sets:**

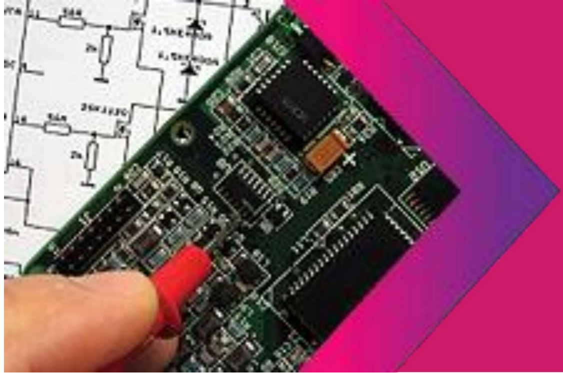
  ✓ **Programming Languages:**
  - Linux Shell scripting
  - Advanced C / Embedded C
  - Data structures & Algorithm design

  ✓ **Linux system programming:**
  - Linux Kernel system calls
  - IPC mechanisms - Pipe, FIFO, Shared memory
  - IPC synchronization - Semaphores, Mutex
  - Networking - TCP and UDP sockets, Protocol development
  - pThreads - Multi thread programming

  ✓ **Embedded Linux:**
  - U-boot, cross compiling, porting Linux kernel
  - Linux bring-up on ARM board
  - Kernel customization and tuning

  ✓ **Embedded controllers:**
  - Hands-on working with GPIOs, Analog I/Os, Memory usage, interfacing, character LCD, Interrupts
  - Peripherals usage - Timers, Counters and ADC
  - Communication protocols - UART, SPI, CAN, I2C etc

  ✓ **Embedded platforms:**
  - Distributions - Linux (Fedora / Ubuntu)
  - ARM based boards - Beaglebone black / Raspberry Pi
  - PIC based boards



NOW FOR **SIMPLIFIED** SOLUTIONS

- ✓ **Software Engineering tools:**
  - Version Control: GIT
  - Code review: Codestriker
  - Defect tracker: Bugzilla
  - Code coverage: Linting tools

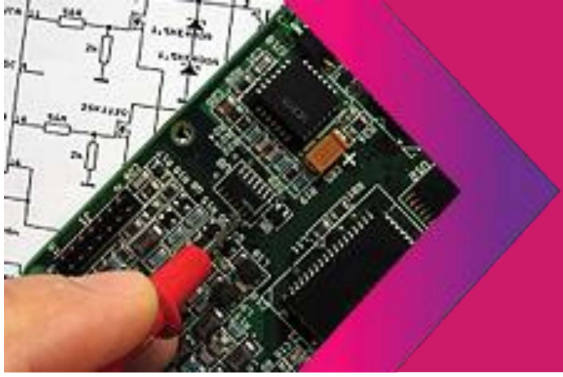
- ✓ **Development tools:**
  - Dev environment: Vim, MPLAB IDE, Makefiles
  - Network packet capture: Wireshark
  - Compilers : GCC, XC8, ARM-Linux-gcc
  - Debuggers: GDB

Embedded Systems: Project list

Linux Systems & Shell programming

Project-1

Title	Command Line Test
Project brief	Command line test is a BASH shell based tool that simulates login based online testing scenario. Initially the user will be provided with sign-in option where pre-defined users will be allowed to login. Upon successful login this tool will display questions for the user from existing data-base. It will also handle error conditions like time-out. This tool will also store answers provided by users for future verification.
Technologies used	▪ Shell programming constructs (ex: loops) ▪ Pattern matching (ex: grep, sed etc...) ▪ File handling (ex: permission, directories etc...)



NOW FOR SIMPLIFIED SOLUTIONS

Project-2

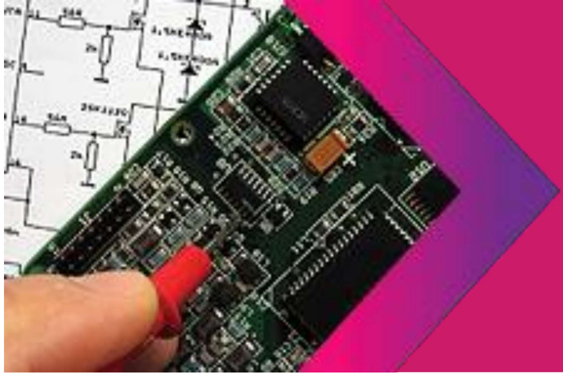
Title	Database tool
Project brief	Database is a BASH shell based tool that is providing various operations to handle a particular database. Initially the data-base need to be pre-populated with some data for which the user will be provided with a set of commands (ex: add / modify / delete / search) using which he should be able to modify them. Along with this user functionality, this tool should record all the actions performed along with time-stamps for future references. Since the idea of this tool is to exposure SHELL programming, CSV file is used as a data-base storing entity.
Technologies used	▪ Basic understanding of data-base operations ▪ Timeout handling ▪ Command parsing using Shell commands

Advanced C programming

Project-1

Title	Image Steganography using LSB mechanism
Project brief	The art and science of hiding information by embedding messages within other, seemingly harmless messages. Bits of unused data are replaced by bits of valuable information using LSB mechanism. Sender and receiver will have individual key / secret based on which they will be able to extract the actual data from the image. This project also gives basic level understanding of image processing methodologies.
Technologies used	▪ Function pointers ▪ File I/O operations ▪ Bitwise operations

ΣMERTXE



NOW FOR SIMPLIFIED SOLUTIONS

Project-2

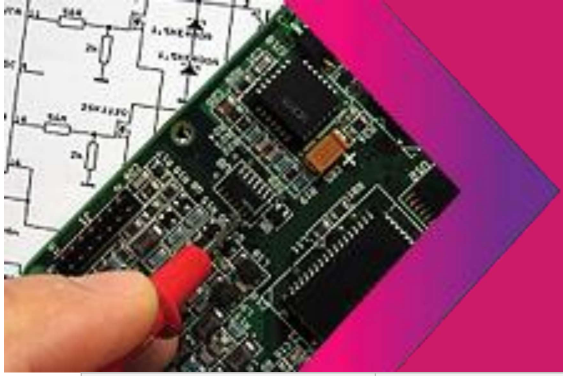
Title	MP3 tag reader
Project brief	MP3 tag reader is a software which will read and display MP3 (ID3) tag information from MP3 files. The software will be desktop based and not web based. This solution will read the given MP3 file, extract various tag information and display them via command line. This project can be extended to implement a tag editor, where-in users can modify mp3 tag information.
Technologies used	▪ Function pointers ▪ String operations ▪ File I/O handling

Project-3

Title	Lexical analyzer
Project brief	Lexical Analyzer is a program which converts the stream of individual characters, normally arranged as lines into the stream of lexical tokens. Tokenization for instance of words and punctuation symbols that make up source code. The main purpose/goal of the project is to take in a C file and produce the sequence of tokens that can be used for the next stage in compilation. This should also take care of necessary error handling conditions that may occur during tokenization.
Technologies used	▪ File I/O operations ▪ File pointers ▪ String operations

Project-4

Title	Code to HTML convertor
Project brief	Given input code (in this project input given as C program) the convertor should make it into an HTML based output that can be read by a browser. In popular syntax highlighter based solutions / tools such mechanisms are internally used, thereby making code walkthrough easier. Reserved key words, preprocessor directives, numerical constants, strings etc must be displayed in pre-defined color.



	This convertor also takes care of coloring scheme to highlight keywords.
Technologies used	▪ Basic understanding of HTML ▪ Usage of all type of pointers ▪ String tokenization

Project-5

Title	Runtime Enforcement of Memory Safety for the C Programming Language
Project brief	Memory access violations are a leading source of unreliability in C programs. Although the low-level features of the C programming language, like unchecked pointer arithmetic and explicit memory management, make it a desirable language for many programming tasks, their use often results in hard-to-detect memory errors. While there are lot of solutions to this problem exists, each of them have their own limitations. This project's idea is to introduce MemSafe, a compiler analysis and transformation for ensuring the memory safety of C programs at runtime while avoiding the above drawbacks.
Technologies used	▪ Compiler design ▪ Memory layout understanding ▪ Advanced pointers usage

Data Structures & Algorithms

Project-1

Title	Arbitrary Precision Calculator (APC)
Project brief	Arbitrary-precision arithmetic, also called bignum arithmetic, multiple precision arithmetic, or sometimes infinite-precision arithmetic, indicates that calculations are performed on numbers whose digits of precision are limited only by the available memory of the host system. This contrasts with the faster fixed-precision arithmetic found in most arithmetic logic unit (ALU) hardware, which typically offers between 8 and 64 bits of precision. A common application is public-key cryptography, whose algorithms commonly employ arithmetic with integers having hundreds of digits. The goal of this project is to implement basic mathematical, modulus and power-of operators of

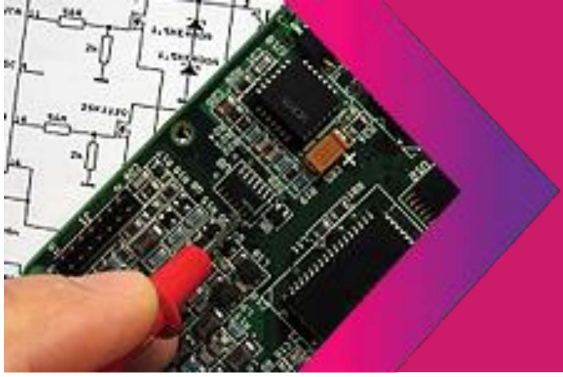
	two large numbers.
Technologies used	▪ Order complexity ▪ Applying Abstract Data Types (ADT) ▪ Self-referential structures using linked list

Project-2

Title	**Expression convertor**
Project brief	Any mathematical expression can be expressed in infix, post-fix and pre-fix formats. A universal expression convertor to be implemented in C using stack data structure. This structure will take input of input expression and conversion option from the user and provide the output in the desired format. The goal of this project is to handle various mathematical operations by converting given expression. This project should also handle appropriate error conditions in the expressions.
Technologies used	▪ Various mathematical expressions and conversion mechanisms ▪ Stack ADT and operations ▪ Dynamic memory allocation

Project-3

Title	**Implementation of line editor**
Project brief	Implementation of line editor using C. Typically line editors allow users to modify contents of a line by offering all possible text related operations (ex: delete). In this project, the line editor to be implemented using linked lists by giving a command line with all possible line editing options. In this project, initially contents are read from the file and appropriate line editor operations modify the text.
Technologies used	▪ File operations ▪ Linked list operations ▪ String tokenization



NOW FOR SIMPLIFIED SOLUTIONS

Project-4

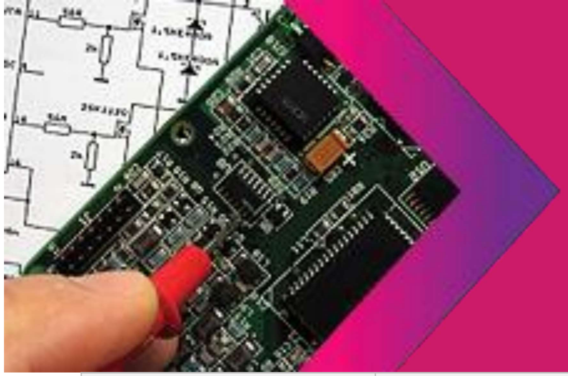
Title	Red Black Tree (RBT)
Project brief	A red-black tree is a kind of self-balancing binary search tree. Each node of the binary tree has an extra bit, and that bit is often interpreted as the color (red or black) of the node. These color bits are used to ensure the tree remains approximately balanced during insertions and deletions. Though it is not perfectly balanced tree, it is good enough to allow it to guarantee searching in O (log n). The goal of the project is to implement all RBT operations and achieve optimal searching for large set of data.
Technologies used	▪ Order complexity ▪ Search and sort algorithms ▪ Tree ADT operations

Project-5

Title	Text indexing using Hash Algorithms
Project brief	Text Indexer is an application that helps to locate a particular text faster in given set of large data by keeping track of words and their locations in files. This console based application uses standard I/O for searching the words in the files. The purpose of storing an index is to optimize speed and performance in finding relevant documents for a search query. Without an index, the search engine would scan every document in the corpus, which would require considerable time and computing power. The goal of the project is to achieve optimal searching by using Hashing.
Technologies used	▪ Hashing algorithms ▪ File I/O ▪ Text parsing

Project-6

Title	Sparse polynomial operator
Project brief	Sparse polynomials are special type of polynomials, with very less items associated with it. An operator program to be written using linked lists to perform operations on it and provide the output sparse polynomial. In this project the goal is to achieve basic mathematical operations using given



	two polynomials.
Technologies used	▪ Linked lists ▪ Sparse polynomials ▪ Mathematical expressions

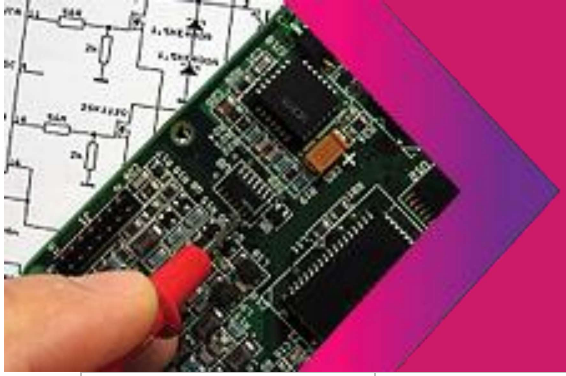
Linux Internals & TCP/IP Networking

Project-1

Title	**Mini Shell implementation - System calls & Signal handling**
Project brief	Mini shell is a command processor, typically run in a text window, allowing the user to type commands which cause actions. BASH can also read commands from a file, called a script. Like all Unix shells, it supports piping and variables as well. The goal of this project is to implement a mini-shell that mimics the BASH shell by using Linux Kernel System calls and IPC mechanisms like signals. It will also handle special keyboard actions (ex: Control C), can be extended for advanced functionalities (ex: Command history) as well.
Technologies used	▪ Linux Kernel System Calls usage ▪ IPC - Signal handling ▪ String pointers & parsing

Project-2

Title	**Message Queue IPC implementation inside Linux Kernel**
Project brief	Message queues are one of the IPC mechanisms that helps two user-space process to communicate and exchange information. They are exposed by the Kernel as system calls. In actual implementation the data-structures are maintained inside the Kernel to facilitate buffer transfer between two processes. The goal of this project is to implement Message Queue IPCs inside the kernel and expose them to user space processes. Along with implementation, these new system calls need to be hooked into Kernel's soft interrupt ecosystem.
Technologies used	▪ Queue & Linked list ADTs ▪ Linux Kernel data structures - process & memory



	▪ System call / Soft-interrupt ecosystem of Linux Kernel

Project-3

Title	Implementation of Trivial File Transfer Protocol (TFTP) - RFC 1350
Project brief	The Trivial File Transfer Protocol (TFTP) is a simple way of transferring file between two systems. This protocol is specified in RFC 1350.This protocol doesn't support advanced available in FTP (ex: User authentication), typically used in Embedded systems for its smaller footprint and simplicity. The goal of this project is to understand RFC and implement the protocol in a LAN environment. Eventually this project can be extended for advanced cases like inter-operating with standard TFTP client / servers.
Technologies used	▪ TCP/IP networking ▪ UDP socket APIs ▪ Time-out and re-transmission used in networks

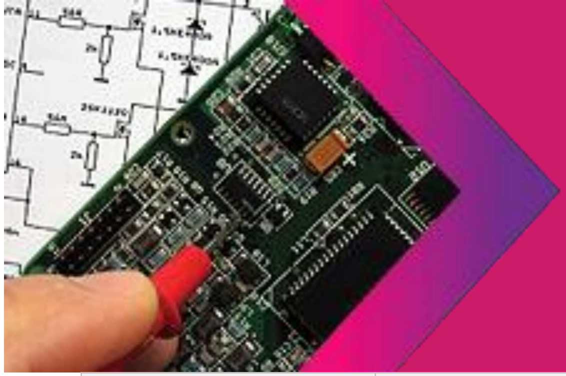
Project-4

Title	TCP based remote management
Project brief	Internet today has become a very complex entity by having different set of devices working together. In a scenario where the network devices are located remotely (ex: Wireless base station) monitoring such devices pro-actively becomes a very critical activity. Any malfunctions happen in remote device (ex: CPU usage) will result in device crash. The goal of this project is to implement a TCP based client and server. The centralized server will connect with multiple clients and monitor various system parameters, thereby enabling remote manageability.
Technologies used	▪ Linux commands used in systems management ▪ TCP socket APIs ▪ IPC Signal handling

Project-5

Title	Network packet injector using RAW sockets

ΣMERTXE



Project brief	One of the key requirements of network is the ability to handle large volume of data. Network should have necessary resilience to handle, which is simulated using Network packet injectors in test environment. The goal of this project is to create a command line based Network packet injector that will generate all major protocol packets (ex: HTTP). For implementation RAW sockets are used.
Technologies used	▪ TCP/IP network protocols - Headers & payload ▪ RAW socket usage ▪ Command line parsing

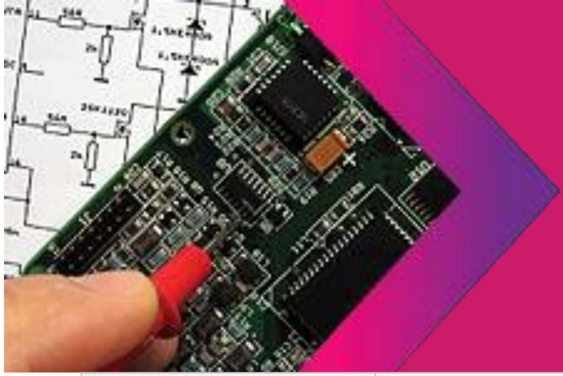
Project-6

Title	TCP/IP chat room
Project brief	Web based chat applications is one of the more commonly used tool for effective two way communication, starting with IRC chat-rooms. Each user need to authenticate himself and join a chat-room after which he should be able to communicate with all in the group. The goal of this project is to simulate a chat room in a LAN by using TCP sockets and demonstrate message exchanges
Technologies used	▪ TCP socket APIs ▪ Server discovery in a LAN environment ▪ User authentication mechanisms

Micro-controller programming

Project-1

Title	Car Black Box (CBB) implementation
Project brief	Black Boxes are typically used in any transportation system (ex: Airplanes) that are used for analysis post-crash and understand the root cause of accidents. Continuous monitoring and logging of events (ex: over-speeding) is critical for effective usage of black box. The goal of this project is to implement core functionalities of a care black-box in a PIC based micro-controller supported by rich peripherals. Events will be logged



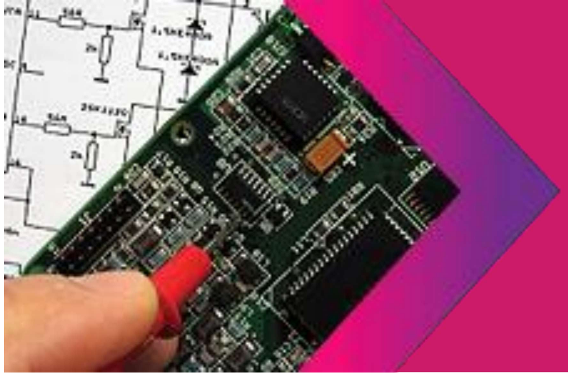
	in EEPROM in this project. This project can be further extended to any vehicle.
Technologies used	▪ PIC micro-controller & schematics ▪ Peripheral (ex: Potentiometer) handling by understanding data-sheets ▪ Interrupt handling

Project-2

Title	Digital Alarm Clock
Project brief	Alarms are used in any Embedded systems to communicate critical event in the system. Timers play critical role in this, which can be understood by implementing applications like personal alarm clock. The goal of this project is to blow alarm depending on the configuration set by the user (ex: daily / weekly / particular day). Eight various options are provided to facilitate this. Focus is also making it user friendly with LCD, I2C RTC, digital keypad and LED's for user interfaces like menus, display time etc.
Technologies used	▪ Timer and interrupt handling ▪ Various peripherals usage (ex: LCD) ▪ Protocols usage (ex: I2C)

Project-3

Title	Remote Controller of Air Conditioner
Project brief	Remote controller is a day-to-day device used by any consumer. Typically they used IR transmitter-receiver for communication using which commands are sent to the A/C. The goal of this project is to demonstrate this functionality between two PIC boards that are acting as transmitter and receiver. Implementation requires an interrupt-driven structure interfaced with matrix keypad, RTC and IR. Upon receiving appropriate commands from the user thru LCD (ex: reduce temperature) the sender will generate appropriate infrared signal padded with user input. This is eventually interpreted by the receiver and appropriate actions are performed.
Technologies used	▪ IR protocol usage ▪ Command / Control protocol definition between transmitter and receiver ▪ Multiple peripheral and interrupt handling



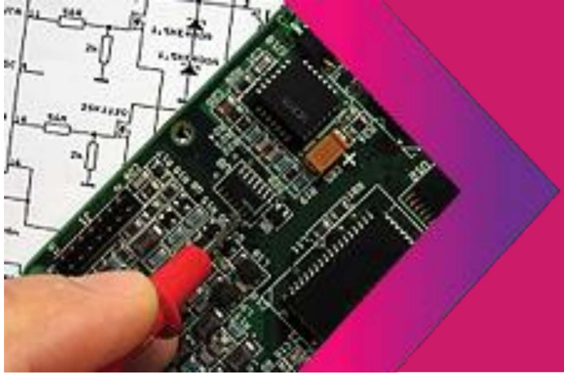
NOW FOR **SIMPLIFIED** SOLUTIONS

Project-4

Title	**CAN based node communication for Automotive**
Project brief	A Controller Area Network (CAN bus) is a vehicle bus standard designed to allow Microcontrollers and devices to communicate with each other in applications without a host computer. It is a message-based protocol, designed originally for multiplex electrical wiring within auto-mobiles, but is also used in many other contexts. The goal of this project is to implement CAN protocol based data exchange between two PIC based boards, which are acting as “nodes”. In this implementation automotive use-case is assumed by exchanging data like engine temperature, speed etc that are displayed using LCD panels.
Technologies used	▪ CAN protocol ▪ Command line based test protocol for node diagnostics ▪ Various peripherals usage (ex: LCD)

Project-5

Title	**WiFi based smart meter**
Project brief	Effective metering (ex: electricity) becomes critical in a connected world powered by Internet-Of-Things (IoT). By connecting heterogeneous devices to a WiFi network such things can be established. The goal of this project is to implement a WiFi based smart metering solution by interfacing a PIC based device with WiFi module. Upon connecting with an access point, device will expose device information (ex: energy usage) which is eventually captured by the user via browser.
Technologies used	▪ WiFi module & sensor interfacing ▪ HTTP protocol (Client & Server) ▪ Internet-Of-Things - Exposure



NOW FOR **SIMPLIFIED** SOLUTIONS

Embedded Linux on ARM

Project-1

Title	Kernel Optimization - Footprint and boot-time reduction on ARM
Project brief	In multi-tasking Embedded systems, having an optimized Kernel is one of the key requirements. With clear understanding of source code organization and various turning methodologies Linux Kernel size can be optimized. In the similar way understanding U-Boot, optimization of boot time can also achieved. The goal of this project is to gain exposure in terms of Kernel optimization by creating a customized Kernel for an ARM based target.
Technologies used	▪ Kernel source code organization & configuration options ▪ U-Boot, Linux Kernel and Init system understanding ▪ ARM target schematics

Project-2

Title	WiFi interface bring-up & Application porting
Project brief	In custom Embedded systems not all the interfaces would be working by default. This requires bringing-up specific interfaces either by writing a device driver from the scratch or compiling pre-existing drive module gets compiled into the custom Kernel, after which applications can be successfully ported. The goal of this project is to compile a WiFi driver into the custom Kernel and port any standard TCP/IP based network applications on to the ARM target. Post porting target should be able to successfully do data exchange with host.
Technologies used	▪ Linux device driver - Basic level understanding ▪ Kernel customization ▪ Application porting

ΣMERTXE

Emertxe Information Technologies Private Ltd
#83, 1st Floor,

Farah Towers,

MG road,

Bangalore - 560001

T: +91 809 555 7 333 (M), +91 80 4128 9576 (L)
E: training@emertxe.com

www.emertxe.com