Embedded C Training aligned with Industrial Automation (240 Hrs)

Company Confidential **ANCIT**

Embedded C Training aligned with Industrial Automation Course Agenda from ANCIT

Delivery Format : This Course is offered in Classroom or Online Format

Duration : 240 Hours (6 Weeks)

Target Group : Embedded Engineers in AUTOMOTIVE, ECU Developer and User, Project Leaders

Prerequisites : Knowledge about C Programming

Outcome : Embedded C Concepts, Communication Protocols, Reading Datasheet, Bootloader, MISRA, IEC Standards, Industry Specific Protocols, Sensors, Actuators, Controllers, Gateway, Platforms, Different Monitoring Solutions, Industrial Case Studies, RTOS and Linux System Programming.

Week 1. Refresher on C Programming along with Assignments

- C Introduction & Datatypes
- Flow Control Statements
- Functions
- Arrays/ Strings
- Structure & Union
- File Handling
- Unit Testing using Open-Source Framework
- Evaluation 1
- Function And Storage Class
- Pointer And Array
- Multi Dimension Array and Advanced Pointer
- User-Defined Data
- Preprocessor
- Evaluation 2

Week 2.3,4 Embedded C Hands-on

GPIO, PWM

1. Control an LED using GPIO. Write a function that implements various LED flashing routines - number of times the LED is flashed, duty cycle (ratio of On to Off time), period (On + Off time), number of flashes etc. Write a file with functions to control and LED and create a header file with function prototypes to be able to call these functions from other files, like main.c.

2. Control an LED using PWM. Write functions to set PWM frequency and duty cycle and create header file to access these functions from other files. In main, call this LED function to control the LED in various patterns like flashing at 1Hz, control brightness of the LED, fade in and fade out. Use the same function to control a buzzer, by configuring frequency and number of beeps.

ADC and Interrupts

3. Measure an analog voltage using ADC. Write a file that has functions to read from ADC and return the value. The function should take channel number as input and return the measured value. Write a header file with configurable values for ADC resolution, reference voltage, scaling factor etc.

4. Combine #2 and #3. Measure an analogue signal and control the brightness of an LED

5. Do #1 using interrupts.

UART & Timer

6. Measure input frequency and duty cycle using timer capture. Write a header file with function prototype of a function that will return the frequency and duty cycle values of an input
signal. Use a function generator to generate a square wave with variable frequency and connect this signal to the development board through its expansion header.

7. Transmit through UART. Write functions that will transmit a debug string on a UART port. Configure printf to print these debug strings.

I2C Memory [EEPROM] Interface with Sensors

8. Write a program to read temperature and humidity through I2C.

9. Write a program to store and read data into an EEPROM using I2C

RTOS, RTC

10. Examples on RTOS and RTC

Additional Concepts covered as part of the sessions

Programming best practices

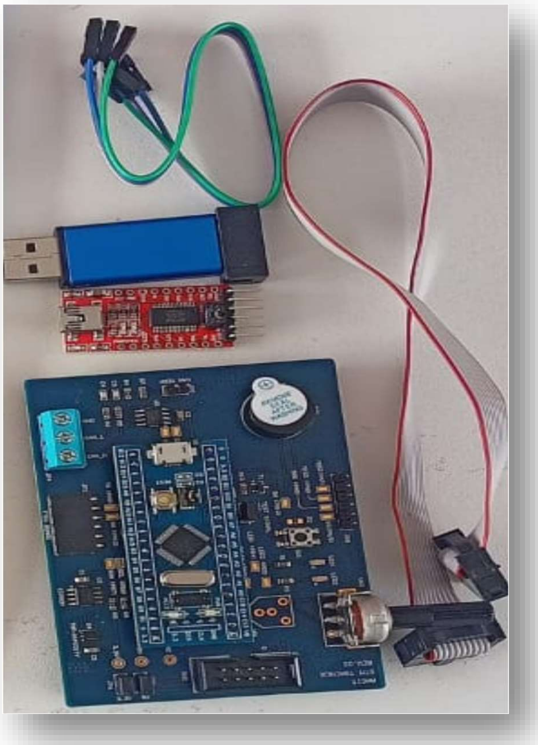
- Initializing variables
- Expose only the required functions to other modules
- Commenting
- Function headers
- File headers with copyright

Debugging Concepts

- Using JTAG or SWD
- Using debug UART

Embedded C Training aligned with Industrial Automation Course Agenda from ANCIT

Hardware Required for the Embedded C Training



Following are the features of the board.

- STM32F103C8T6 Arm Cortex M3 microcontroller
- STLink V2 programmer/debugger
- FTDI USB-UART converter board

Following are the peripherals on the board

- LEDs – 2 Nos.
- Buzzer
- Potentiometer
- Pushbutton switch
- CAN transceiver
- Expansion port to connect USB-UART board
- 24C04 I2C EEPROM (on I2C bus)
- SHT-21 Environment sensor (on I2C bus)
- Expansion header for SPI

Summary Week 2,3,4 :

Associate would be able to understand Microcontroller Peripherals & Communication Interfaces
[GPIO, ADC, Timer, UART, Counter, PWM, I2C, RTOS, RTC, EEPROM]
Associate will understand the process of Code, Test, Build and Flash an Embedded Application
Associate will also be able to DEBUG the Embedded Application

Evaluation Mechanism
MCQ Questionnaire for Theory Sessions
Will be conducted through the Online Quiz Portal

Programming Exercises for Evaluating Embedded C Concepts Will be conducted by the Course Coordinator

Week 5 – Industry-Specific Embedded Training

- Communication protocols - Modbus RTU and Modbus TCP
- Industrial Sensors/Transceivers - Temperature (RTD, thermocouples...), Pressure, Level, Vibration, Flow, Proximity etc.,
- Industrial Actuators - Relays, Solenoids, Valves etc.,
- Industrial Controllers - On/Off Controllers, PID controllers etc.,
- IoT Gateways - Wi-Fi Gateways & GPRS/4G Gateways
- IoT Platforms - Data logging, Visualization and Analytical tools
- Different Monitoring Solutions

  a) Production Monitoring
  b) Process Monitoring
  c) Energy Monitoring
  d) Utility Monitoring

- Different Industrial Case Studies

Week 6 – Linux System Programming and RTOS Concepts

Introduction to Linux systems

- Linux architecture.
- Writing the 'Hello World' program, compiling using GCC and its execution.

What is system call and standard library function calls
Kernel and User mode in Linux
File Operations

- open (), close(), read(), write(), Iseek()
- Blocking and Non-Blocking calls.
- Atomic operations, Race condition.

Memory Management and Virtual Memory

- Stack segment, code segment, heap segment, data segment,
- Virtual memory management.
- Functions related to Memory allocation - Malloc(), calloc(), Realloc(), Auto variables, static variables.

Process Management

- Process creation, Process termination, wait(), Process ID, fork() system call, Exec() family of system call, Parent-Child Process management,
- Command line arguments of Process.

Embedded C Training aligned with Industrial Automation Course Agenda from ANCIT

Signals

- signal(), signal handlers, sending signals to process, ignoring and default signal actions.

Posix Threads

- Thread creation, thread termination, thread ID, joinable and detachable threads.

Thread synchronization.

- Mutex, Condition variables, Thread safe.

Posix - Inter-Process communication

- Pipes, FIFO, Posix message Queue, Posix Semaphores, Posix Shared Memory

RTOS

- **Creating Multiple tasks**
- **Handling Priorities to tasks**
- **Task Operations**

  Create a Task
  Switching Between Tasks
  Suspend and Resume Tasks
  Terminating the Task
  Block the task for some time

- **Binary Semaphore**

  Semaphore Wait
  Semaphore Release

ANCIT

SNO : 37 Gurusamy Nagar,
Codissia Road, Peelamedu,Coimbatore,
Tamil Nadu, India- 641004

+91-9840378602/ 9483541953

info@ancitconsulting.com

www.ancitconsulting.com
www.ancitedutech.com