

# Embedded System Development Course Syllabus

## Module (1/11)

### Module Name: Advanced / Embedded C Programming

#### Objectives:

Clearly understand concepts of C language  
To obtain good quality and style in programming  
Gear you up for programming in Embedded environment  
To induce confidence in you!

#### Detailed course contents:

Basics of C:  
Operators  
Conditionals  
Arrays  
Functions  
Advanced C programming:  
Structures  
Unions  
Files  
Deep dive into pointers  
Pre-processor directives  
Recursion  
Project environment - Creating & Building a project, Makefiles  
Deep dive - Logic to program translation, Creating your own library, Dry-run  
Data Structures  
Basics of Embedded C programming

## Module (2/11)

### Module Name: Advanced / Embedded C Programming

#### Objectives:

Clearly understand concepts of C++ language  
To obtain good quality and style in programming  
Gear you up for programming in Embedded environment  
To induce confidence in you!

#### Detailed course contents:

## Introduction to Object Oriented Programming

Procedure Oriented Vs Object Oriented

Difference between C and C++

C++ Output/ Input

Keywords in C++

New Style of header files Specification

Comments in C++

Variables in C++

Structures in C

Structure in C++

Access Specifier

Classes

Objects in C++

More on Classes and Objects

Dynamic Memory Management

Constructor and Destructor

Inheritance

Virtual Functions and Inheritance

Operator Overloading

Constructor- Destructor Invocation

Templates • Exception Handling

Working with input and output and files

Basic understanding of standard template library

The string of C++

Miscellaneous concept of C++

## Module (3/11)

### Module Name: Programming with Python

#### Objectives:

Become a professional python programmer

Learn basic to advanced features of Python programming language

Build object oriented programs using python

Work with file streams, access csv files

Create packages, Implement exception handles and learn to debug python

application

Learn basics of regular expressions

#### Detailed course contents:

Overview of Programming with Python

Native Datatypes and Operators

Python Statements and Conditionals

## Functions

Strings  
Object oriented programming with Python  
Errors and Exception Handling  
File handling  
Regular expression  
Modules and Packages

## Module (4/11)

### Module Name: Micro-Controller

#### Objectives:

8051 controller

Arduino

PIC

ARM

AVR,

Cortex MO-M4(8-16-32 Bit) Cypress

STM32,

SAM,

ATMEL

Protocols

Technologies

#### Detailed course contents:

Introduction to 8051 controller

Architecture of 8051

Instructions

Interfaces

1. LED
2. LCD
3. Switches and Keypad
4. Keypad
5. Buzzer
6. RTC and TRIAC
7. Stepper and DC Motor
8. ADC , DAC and PWM
9. Seven Segment Display
10. Sensors

Introduction to Arduino

## Architecture of Arduino

### Interfaces

1. LED
2. LCD
3. Switches and Keypad
4. Buzzer
5. RTC and TRIAC
6. StepperMotor
7. DC Motor
8. ADC ,DAC and PWM
9. Seven Segment
10. Display and Sensors

### Introduction to PIC

### Architecture of PIC

### Interfaces

1. LED
2. LCD
3. Switches and Keypad
4. Buzzer
5. RTC and TRIAC
6. StepperMotor
7. DC Motor
8. ADC ,DAC and PWM
9. 7 Segment
10. Display and Sensors

### Introduction to ARM

### Architecture of ARM

### Interfaces

1. LED
2. LCD
3. Switches and Keypad
4. Buzzer
5. RTC and TRIAC
6. StepperMotor
7. DC Motor
8. ADC ,DAC and PWM
9. 7 Segment
10. Display and
11. Sensors

### Protocols

1. UART
2. USART
3. I2c
4. SPI

## Technologies

1. Bluetooth (BLE)
2. Wi-fi
3. Zigbee
4. IR
5. RF
6. SPI
7. UART
8. I2C
9. USB
10. GSM
11. RFID
12. GPS
13. Finger print
14. Voice Module
15. MODBUS

Introduction to AVR,  
Architecture of AVR  
Interfaces of AVR,

Introduction to Cortex MO-M4(8-16-32 Bit) Cypress  
Architecture of Cortex MO-M4(8-16-32 Bit) Cypress  
Interfaces of Cortex MO-M4(8-16-32 Bit) Cypress

Introduction to STM32  
Architecture of STM32  
Interfaces of STM32

Introduction to SAM  
Architecture of SAM  
Interfaces of SAM

Introduction to ATMEL  
Architecture of ATMEL  
Interfaces of ATMEL

## **Module (5/911**

**Module Name: Linux Operating System and Programming**

### **Objectives:**

Operating System  
Linux Basic Commands  
Linux System Programming

## Linux Device Driver Programming

Linux Network Programming

Real – Time Operating System (RTLinux) Programming

ARM9/ARM11 - Linux Programming

QT Creator-OpenCV-ARM9/ARM11

Kernel Mechanisms

Beagbone black

### **Detailed course contents:**

#### Operating System

Introduction

Processes

Threads

CPU Scheduling

Process Synchronization

Deadlocks

Memory management

Virtual Memory

File management and Disk management

#### Linux Basic Commands

#### Linux System Programming

Program, Process, Process IDs,

Processes Priorities, Process States, CPU Scheduling

Process Management API – fork, vfork, exec, wait and exit.

Zombie and Orphan Process

pthread Programming and Thread Attributes

Process Synchronization Techniques.

Semaphores, Mutex, Spinlock, Memory Barriers

Inter Process Communication Techniques

Pipe, FIFO, Signals, Shared Memory

Timer API – Jiffies, kernel Timers, wait queues, sleeps

Interrupt and Exception API – Task lets, Work queues

Kernel Debugging – kgdb, printk, jprobs, kprobs

#### Linux Device Driver Programming

Introduction to Device Drivers

Device Number, Major and Minor Numbers

Inbuilt and Modular Drivers

User Space and Kernel Space Communication

Dev directory and device files

Character, Block and Network Driver

Advanced Driver API – fcntl,ioctl

Unified Device Model (udev)

Proc File System, sys file system

usbfs file system.

## Character Device Driver Programming

- Block Device Driver Programming
- Serial Port Driver Programming
- Parallel Port Driver Programming
- USB Device Driver Programming
- Network (Ethernet) Driver Programming
- Flash Drive Driver Programming

## Linux Network Programming

- Networking Architecture in Linux
- TCP/IP defector model and layer in kernel.
- Client – Server Programming API.
- TCP, UDP, RAW, UNIX, FTP, TFTP
- VOIP Protocol Programming.
- Arithmetic Server, Concurrent Server.
- Broadcast Server Programming
- Wireless Architecture and Programming

## Real – Time Operating System (RTLinux) Programming

- Real Time Concept, RTLinux Installation.
- Scheduling and Preemption Programming.
- GPOS and RTOS API Programming.
- FIFO and Round Robin Scheduling Programming
- Task Management, synchronization, Intertask Communication
- Timers, Interrupt API in RTOs.
- Signals, Events handling API in RTOs.
- Priority Inversion/Inheritance.

## ARM9/ARM11 - Linux Programming

- ARM Architecture and ARM Processor family.
- ARM Microcontroller ICs in Market.
- ARM Development Boards and features.
- Friendly ARM Development Boards.
- Mini2440, Mini6410, Mini210 Development Boards.
- Raspberry Pi Development Board.
- Beagle Bone Development Board.
- Boot loader Configuration and Compilation for ARM
- Kernel Configuration and Compilation for ARM.
- File System Configuration and Compilation for ARM.
- Porting Linux / Android OS on Development Boards.
- DNW Tools, USB Push, Hyper Terminal Tool.

## QT Creator-OpenCV-ARM9/ARM11

- Installing QT Framework
- Installing Opencv and Pocket Sphinx.
- Application Development Using QT Creator.
- Sensors and modules Programming in QT Creator.

Wireless Protocol Programming in QT Creator.  
Image and Video Processing Programming in Opencv.  
Audio Processing Programming in Pocket Sphinx.  
Interfacing Opencv with QT Creator.  
Interfacing Pocket Sphinx with QT Creator.  
Creating HTML and HTTP pages for ARM Board.  
Setting Cross Compiler in QT for ARM.  
Executing Cross Compiled project file on ARM Board.

#### Kernel Mechanisms

- Kernel Threads
- Kernel Timers
- Delaying Execution
- Tasklets
- Workqueues

#### Beagbone black

- Intro to BBB
- Applications

## **Module (6/11)**

### **Module Name: Processors**

#### **Objectives:**

- Broadcom
- NXP,
- ATSAM
- INTE
- Qualcomm

#### **Detailed course contents:**

Broadcom	Processor's Types Characteristics of Processors Application of Processors
NXP	Processor's Types Characteristics of Processors Application of Processors
ATSAM	Processor's Types Characteristics of Processors Application of Processors
INTEL	Processor's Types Characteristics of Processors



## Application of Processors

Qualcomm

Processor's Types  
Characteristics of Processors  
Application of Processors

### **Module (7/11)**

#### **Module Name: Developing Tools**

#### **Objectives:**

Compilers  
Debugging techniques  
ICSP

#### **Detailed course contents:**

Compilers

Keil-UV,  
Keil MDK,  
Arduino,  
Eclipse,  
STMCUBE CUBIX,  
AVR Studio,  
MPLAB,ATMEL

Debugging techniques

GDB,  
gdb server  
Strace  
/proc& /sys  
Valgrind

ICSP

Introduction of ICSP  
Technical Briefs of ICSP  
Programming Specification of ICSP

### **Module (8/11)**

#### **Module Name: Micro-controller programming using Arduino platform**

#### **Objectives:**

Understand Embedded Systems and its components



Learn how to build embedded applications using Arduino Platform  
Become familiar with hardware interfacing using Arduino  
Should be able to read device data sheets and board schematics  
Learn basic communication protocols and communicate using Arduino Libraries  
Build IoT applications using wired and wireless protocols (ex: Bluetooth, Wifi)  
Debug applications using Arduino IDE

### **Detailed course contents:**

Overview of Embedded Systems  
Components of Embedded Systems  
Micro-controller Architecture and Properties  
Installing and Setting up the Arduino development environment  
Blinky Sketch – A walk through  
Arduino Sketches  
Classes  
Sketch Structure  
Pins  
Arduino Shields  
Hands-on working with GPIOs, Analog I/Os, Memory usage  
Micro controller peripherals usage - Timers, Counters, Interrupts and its sources  
Communication protocols I - UART, SPI, I2C, CAN  
Interfacing IoT sensors and Actuators  
Debug applications using Arduino IDE  
Communication protocols II – Wired and Wireless communication  
Ethernet Client Server Implementation  
Build WiFi Application  
Build Bluetooth Application

### **Module (9/11)**

### **Module Name: Building IoT Applications using Raspberry Pi**

#### **Objectives:**

Become familiar with Raspberry Pi (Rpi) hardware  
Setup and Install Raspbian OS on Rpi  
Understand how Rpi can be leveraged as an IoT gateway  
Become familiar with Linux OS  
Setup Rpi as an IoT gateway  
Using Python Interface with Arduino using Serial Port Interface  
Build Socket applications to communicate to Arduino device using Ethernet, Wifi and Bluetooth interfaces  
Build IoT applications using HTTP and MQTT protocols  
Learn to use Node Red programming tool

## **Detailed course contents:**

Overview of Raspberry Pi (RPi) hardware platform  
Peripherals on Rpi  
Setup and Install Raspbian OS on Rpi  
Overview of Linux OS and its sub-systems  
Process  
Memory Mangement  
Multi-Threading  
IPC  
Linux CLI and important commands  
Linux File System  
Everything is a file concept in Linux  
Device Access using system calls  
Install packages on Raspbian OS  
Setting up Raspbian as an IoT gateway  
Write Python program to interface with Arduino using serial libraries  
IoT Communication Models and Protocols  
Request-Response, Publish-Subscribe, Push-Pull, Exclusive Pair  
Application Protocols: HTTP, CoAP, MQTT, AMQP  
Communication APIs: REST-based, WebSocket-based  
Network Layer: IPv4, IPv6, 6LoWPAN  
Building python based programs to communicate to cloud server using various application protocols  
Develop a complete python based application IoT application  
Rpi as a device  
Interfacing with sensors and actuators using GPIO pins  
Interfacing with camera on Rpi

## **Module (10/11)**

### **Module Name: IoT Cloud Infrastructure**

#### **Objectives:**

To understand IoT cloud infrastructure blocks  
Making right IoT platform choice by understanding various popular platforms  
Configuring IoT platforms to get required analytics  
Learn integration elements (ex: REST APIs) for devices

#### **Detailed course contents:**

IoT cloud building blocks  
Using the platform specific dashboards  
Device configuration and addressing

## IoT Platforms in detail

MQTT Server  
Injection Engine  
Time Series database  
Rules Engine  
Data monitoring, visualization and IoT Analytics  
Rest API interface  
Device Management  
Application Service

## **Module (11/11)**

### **Module Name: Performance and Security in IoT**

#### **Objectives:**

Learn how to benchmark IoT applications and platforms  
Understand the security risks in IoT, counter measures and design consideration  
Get an understanding of using cryptography in IoT

#### **Detailed course contents:**

Benchmarking IoT applications and Platforms  
MQTT vs HTTP performance  
Security considerations  
Firmware updates  
Cryptography basics  
Cryptography in IoT  
Privacy considerations and design guidelines