Embedded Systems Full Road Map

By: Omar Ehab Abdelaziz (Embedded software engineer at valeo)

1- C programming Language

Study any of the following books and focus on the mentioned chapters.

Source	Type	Chapters/Sections	Expected Learning Duration
Pointers On C	Book	2,3,4,5,6,7,8,9,10 11,12,13,14	1 Month
C Programming A modern Approach	Book	2,4,5,6,7,8,9,10,11,12 13,14,16,17,18	1 Month

Reading Chapters only is not enough you should practice writing C programs and for that I recommend the following.

Source	Type	Chapters/Sections	Expected Learning Duration
Hacker Rank	Web-Site	N/A	N/A
San Foundry	Web-Site	N/A	N/A

Solve as much questions as you can over a long period don't stop solving questions.

It is also desirable to practice C MCQ questions which will appear in many pre-interview test and for that I recommend.

Source	Type	Chapters/Sections	Expected Learning Duration
IndiaBix	Web-Site	N/A	N/A

2- Data structure and Algorithms

To be a software engineer you should know at least the basics of data structure and algorithms and be able to use this knowledge to solve problems.

For the basic knowledge I recommend studying the following topics

Concepts
Big O notation
Time Complexity
Space Complexity

Data Structures
Arrays
Dynamic Arrays
Stack
Queues
Linked Lists
Hash Tables

Algorithms
Binary Search
Merge Sort
Quick Sort
Counting Sort

The following are Courses where you can study the previous mentioned topics choose the one you feel comfortable with.

Courses
mycodeschool
UC San Diego data-structures-algorithms
Stanford DSA
Udemy
Udacity Nanodegree

The following are books where you could read from if you prefer reading over courses.

Books
The Algorithm Design Manual.
Grokking Algorithms
Data structures by R.Lafore

You should be practicing solving problems while learning and for that I recommend solving problems on those websites.

Problem solving Web-sites
HackerRank
LeetCode

3- Electronics

If you are in a school where you have studied basic circuits and electronics you can skip this part If you didn’t I recommend studying the following topics from the following resources.

Source	Type	Chapters/Sections	Duration
The Art of Electronics	Book	1,2,3,6,7,10,12,13	1 Month

4- Computer Logic and Architecture

If you are in a school where you have studied Logic design and computer architecture courses you can skip this section if not then I recommend studying the following.

Source	Type	Chapters/Sections	Duration
Computer system Architecture	Book	1,2,3,5,6,7,8	2 Month

5- Embedded Systems and Microcontrollers

This part is the most important one as this is the core of embedded systems so at the beginning I recommend learning the following topics on 8-bit Microcontrollers like Atmega32 or PIC then Apply the topics on 32-bit ARM microcontroller like STM32 or TivaC. I would recommend you start by applying and learning on Atmega32.

Part 1:

For this part you should study each peripheral mentioned directly from the microcontroller data sheet and then implement a Driver for the peripheral and after that you could implement a small project to test the Driver to practice.

Peripheral	Chapter/Section	Driver Implementation	Test Project
I/O	I/O Ports	Required	Write program to Toggle an LED
ADC	Analog to Digital Converter	Required	Write a program to measure temperature using LM35 sensor.
Timer	8-bit Timer/Counter0 with PWM	Required	Write a program to toggle an LED every 100ms
PWM	8-bit Timer/Counter0 with PWM	Required	Write a program to control the speed of a Motor.
UART	USART	Required	Write a program to send and receive data serially with a Computer.

SPI	Serial Peripheral Interface – SPI	Required	Write a program to send and receive data between two microcontrollers
I2C	Two-wire Serial Interface	Required	Write a program to interface with EEPROM.

Part 2:

After you have got a sense of what is embedded systems it is time to gain more theoretical and practical knowledge for that I recommend the following Books.

Source	Type	Chapters/Sections	Durations
Embedded Software Primer	Book	1,2,3,4,9,10	1 month
Programming Embedded Systems	Book	4,6,7,8	2 Weeks

Part3:

At this point you should be familiar with embedded systems and microcontrollers but to enhance your skills I recommend working with 32-bit ARM based Microcontrollers like STM32 or Tiva C and for that I would recommend the following Books and courses.

Source	Type	Chapters/Sections	Durations
Definitive Guide to ARM Cortex	Books	1,4,5,6,7	2:3 Weeks
Embedded-system-programming on arm	Course	N/A	Self-Paced

TIva C datasheet	**Document**	**N/A**	**N/A**
Mastering microcontroller with peripheral driver development	**Course**	**N/A**	**Self-Paced**
Master Microcontroller :Timers, PWM, CAN,RTC,Low Power(MCU2)	**Course**	**N/A**	**Self-Paced**

6- Software Design

At this point you are familiar with lots of concepts related to embedded systems but one important aspect you haven't learned yet is how to design your software and for that I recommend the following Resources (Books and courses) you could choose one of the resources and stick with it.

Source	Type	Chapters/Sections	Durations
Patterns For TIme-Triggered Embedded systems	Book	1,9,13,14,15,16,36	1 Month
Real-time-embedded-systems	Course	N/A	Self-Paced
Real-Time Systems	Books	1,2,4,5,6,7,8	1 Month

7- Real-Time Operating systems

Source	Type	Chapters/Sections	Durations
MicroC OS II: The Real Time Kernel	Book	2,3,4	1 week
Mastering the FreeRtos Real time Kernel	Book	1,2,3,4,5,6,7,8,9	1 Month
Introduction to RTOS	Course	N/A	Self-Paced

8-Projects

At this part you are ready to Implement more advanced and complex projects to do so I recommend choosing projects from this list and Implement it the following projects are from Cornell University and was made as a thesis projects for the ECE department.

Source
Cornell ECE Embedded systems Projects