

Below, you will find the online courses I completed for my continuous improvement. Each course includes a completion certificate, a link to access the course, and its content.

With a total of 157 hours invested in Udemy courses in the embedded software field, it's important to highlight the dedicated time and effort required to successfully complete each course. This demonstrates my commitment, engagement, and proactive attitude towards advancing my skills and expertise in embedded software development.

1. Mastering RTOS: Hands on FreeRTOS and STM32Fx with Debugging	2
2. Bash Mastery: The Complete Guide to Bash Shell Scripting	4
3. Embedded Systems Programming on ARM Cortex-M3 M4 Processor	5
4. Mastering Microcontroller: Timers, PWM, CAN, Low Power(MCU2)	6
5. Embedded Linux Step by Step Using Beaglebone Black.....	8
6. 2021 Complete Python Bootcamp From Zero to Hero in Python	10
7. Git Complete: The definitive, step-by-step guide to Git	11
8. Mastering Microcontroller and Embedded Driver Development.....	12
9. STM32Fx Microcontroller Custom Bootloader Development	14
10. Microcontroller Embedded C Programming: absolute beginners	15

<u>Figure 1. Mastering RTOS: Hands on FreeRTOS and STM32Fx with Debugging.....</u>	<u>3</u>
<u>Figure 2. Bash Mastery: The Complete Guide to Bash Shell Scripting.....</u>	<u>4</u>
<u>Figure 3. Embedded Systems Programming on ARM Cortex-M3 M4 Processor.....</u>	<u>5</u>
<u>Figure 4. Mastering Microcontroller: Timers, PWM, CAN, Low Power(MCU2)</u>	<u>7</u>
<u>Figure 5. Embedded Linux Step by Step Using Beaglebone Black</u>	<u>9</u>
<u>Figure 6. 2021 Complete Python Bootcamp From Zero to Hero in Python</u>	<u>10</u>
<u>Figure 7. Git Complete: The definitive, step-by-step guide to Git.....</u>	<u>11</u>
<u>Figure 8. Mastering Microcontroller and Embedded Driver Development</u>	<u>13</u>
<u>Figure 9. STM32Fx Microcontroller Custom Bootloader Development.....</u>	<u>14</u>
<u>Figure 10. Microcontroller Embedded C Programming: absolute beginners</u>	<u>16</u>

1. Mastering RTOS: Hands on FreeRTOS and STM32Fx with Debugging

Link to the course: ["DIY RTOS Made Easy: Mastering Microcontrollers " | Udemy](#)

- Understanding various RTOS concepts with FreeRTOS Programming and Debugging
- Learn Complete Step by step method to run FreeRTOS on STM32 MCUs using OpenSTM32 System Workbench
- Using STM32 Standard Peripheral Driver APIs to configure peripherals
- FreeRTOS Task Creation , Deletion, Scheduling using with code examples
- Important scheduling policies of FreeRTOS Scheduler
- FreeRTOS Stack and Heap Management
- Right ways of Synchronizing between tasks using Semaphores.
- Right ways of Synchronizing between a task and an interrupt using semaphores
- Synchronization between multiple events and a task
- FreeRTOS Queue management like creation, sending, receiving, blocking, etc
- Implementing mutual exclusion between Tasks using Mutex services and semaphores
- Understanding Context Switching with in detail code explanation
- Understanding Architecture specific codes like SVC_handler, PendSV_handler, SysTick Handler line by line
- You will learn about kernel Tick timer, its importance, the kernel tick frequency and its configuration details.
- Understanding complete ARM Cortex M and FreeRTOS Priority model and its configuration related informations
- FreeRTOS Debugging using SEGGER SystemView Software
- Lots of articles which explains line by line the ARM Cortex M related FreeRTOS architecture specific codes
- Low power instructions use case in FreeRTOS scenario
- Lots of coding exercises
- Debugging with SEGGER software toolkit by taking snapshot and continuous mode recording



N° de certificat : UC-ccfc5a2c-58b8-4cea-bddd-2f209ddee13f
Url du certificat : ude.my/UC-ccfc5a2c-58b8-4cea-bddd-2f209ddee13f
Numéro de référence : 0004

CERTIFICAT DE FIN DE FORMATION

Mastering RTOS: Hands on FreeRTOS and STM32Fx with Debugging

Formateurs **FastBit Embedded Brain Academy, Kiran Nayak**

Vladimir Sirghi

Date **18 février 2024**

Longueur **14 heures au total**

Figure 1. Mastering RTOS: Hands on FreeRTOS and STM32Fx with Debugging

2. Bash Mastery: The Complete Guide to Bash Shell Scripting

Link to the course: [Bash Mastery: The Complete Guide to Bash Shell Scripting | Udemy](#)

- Write bash scripts
- Fluently Read and write bash syntax
- Automate workflows with bash scripts.
- Debug bash scripts using professional debugging tools
- Execute bash scripts on remote servers



Figure 2. Bash Mastery: The Complete Guide to Bash Shell Scripting

3. Embedded Systems Programming on ARM Cortex-M3 M4 Processor

Link to the course: [Embedded Systems Programming on ARM Cortex-M3/M4 Processor | Udem](https://www.udemy.com/course/embedded-systems-programming-on-arm-cortex-m3-m4-processor/)

- Internal architecture of ARM Cortex M3/M4 processor and programming
- Learn Mixed 'C' and Assembly Coding using inline assembly technique
- Demystifying Memory, Bus interfaces, NVIC, Exception handling with lots of animation
- Interrupts and configuration of ARM Cortex Mx based microcontroller
- Low level register Programming for interrupts, System Exceptions, Setting Priorities, Preemption, etc.
- Learn writing IRQ handlers, IRQ numbers, NVIC and mcu more
- Implementation of task scheduler using PENDSV and SYSTICK feature of the processor
- Implementation of context switching
- Learn and write linker script and mcu startup file from scratch
- Bare metal embedded build process
- Processor fault exceptions and fault handler implementation and fault analysis
- Stack and AAPCS standard
- learn inline assembly, naked functions and gcc variable and section attributes.

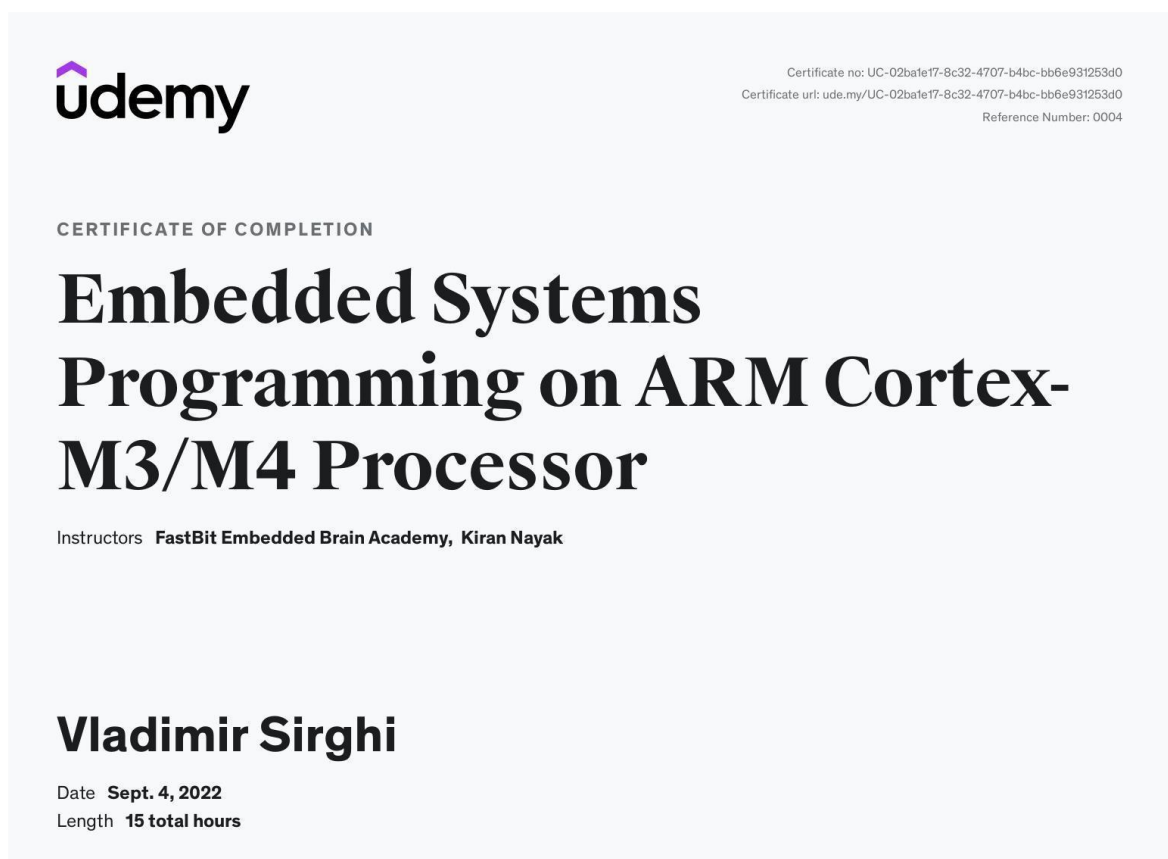


Figure 3. Embedded Systems Programming on ARM Cortex-M3 M4 Processor

4. Mastering Microcontroller: Timers, PWM, CAN, Low Power(MCU2)

Link to the course: [Mastering Microcontroller: Timers, PWM, CAN, Low Power\(MCU2\) | Udemy](#)

- You will learn from scratch about STM32 Timers : Basic and General Purpose Timers
- Understand General purpose timer's Input capture and Output compare unit handling and Exercises
- Handling of Timer interrupts : Time base interrupts, capture interrupts, compare interrupts
- You will learn from scratch CAN Protocol, CAN Signalling, CAN Transceivers , Bus Access procedures
- Understand CAN LOOPBACK mode, SILENT mode and NORMAL mode
- Understand about CAN filtering
- Learn about CAN interrupts
- CAN Peripheral programming using STM32 device HAL drivers
- You will master Low power modes of the MCU : SLEEP, STOP and STANDBY
- You will understand different power domains of the MCU : VDD domain, 1.2V domain, backup domain
- Understand Microcontroller Wakeup Procedures using : RTC, wakeup pins, EXTI, etc.
- You will master RTC Features : CALENDAR, ALARM , TIME STAMP, WAKEUP UNIT
- RTC interrupts and wake up procedures
- Mastering Microcontroller Clocks Handling : HSE, HSI, LSE, LSI, PLL
- Understand phase locked loop (PLL) programming
- Learn PWM mode and Master through step by step code exercises
- You should be able to quickly develop applications which involves STM32 Device HAL layer



Certificate no: UC-0fa4aef2-80e6-4741-95f5-6cc913efd4d4
Certificate url: ude.my/UC-0fa4aef2-80e6-4741-95f5-6cc913efd4d4
Reference Number: 0004

CERTIFICATE OF COMPLETION

Mastering Microcontroller: Timers, PWM, CAN, Low Power(MCU2)

Instructors **FastBit Embedded Brain Academy, Kiran Nayak**

Vladimir Sirghi

Date **May 27, 2022**

Length **21.5 total hours**

Figure 4. Mastering Microcontroller: Timers, PWM, CAN, Low Power(MCU2)

5. Embedded Linux Step by Step Using Beaglebone Black

Link to the course: [Embedded Linux Step by Step using Beaglebone Black Course | Udemy](#)

- Understanding ROM –Uboot-Kernel boot process on Linux-ARM systems and Testing
- Kernel, Bootloaders compilations Step-by-Step and testing on Beaglebone Hardware
- U-Boot, MLO, SPL, explanation and Testing
- Understanding various sub systems of AM335x SOC such as GPIOs, I2C, MMC, boot modes ,etc
- Understanding platform devices and how it works
- Understanding ARM-board configuration files
- Linux device Tree
- Understating various boot modes of AM335x
- eMMC OS updating and booting new Debian OS
- Serial booting (UART)
- TFTP booting
- NFS booting
- Beaglebone Networking
- Busybox compilations and Testing
- Configuring and using Buildroot on Beaglebone
- Writing uEnv dot txt from scratch and Testing
- Using I2C tools on BBB
- Hardware interfacing projects and 'C' application development using Eclipse IDE
- GPIO programming
- Interfacing external LEDs, 7 segment displays and code development
- Interfacing LCD and related projects
- Interfacing MPU6050 sensor and tilt sensing
- EEPROM programming



Certificate no: UC-9bcb39e8-8bbb-4fc2-9349-df48675a8ba0
Certificate url: ude.my/UC-9bcb39e8-8bbb-4fc2-9349-df48675a8ba0
Reference Number: 0004

CERTIFICATE OF COMPLETION

Embedded Linux Step by Step Using Beaglebone Black

Instructors **FastBit Embedded Brain Academy, Kiran Nayak**

Vladimir Sirghi

Date **Feb. 20, 2022**

Length **16 total hours**

Figure 5. Embedded Linux Step by Step Using Beaglebone Black

6. 2021 Complete Python Bootcamp From Zero to Hero in Python

Link to the course: [Course: The Complete Python Bootcamp From Zero to Hero in Python | Udem](#)

- You will learn how to leverage the power of Python to solve tasks.
- You will build games and programs that use Python libraries.
- You will be able to use Python for your own work problems or personal projects.
- You will create a portfolio of Python based projects you can share.
- Learn to use Python professionally, learning both Python 2 and Python 3!
- Create games with Python, like Tic Tac Toe and Blackjack!
- Learn advanced Python features, like the collections module and how to work with times tamps!
- Learn to use Object Oriented Programming with classes!
- Understand complex topics, like decorators.
- Understand how to use both the Jupyter Notebook and create .py files
- Get an understanding of how to create GUIs in the Jupyter Notebook system!
- Build a complete understanding of Python from the ground up!



Figure 6. 2021 Complete Python Bootcamp From Zero to Hero in Python

7. Git Complete: The definitive, step-by-step guide to Git

Link to the course: [Git Complete Full Course: The Definitive Guide to Git | Udemy](#)

- Learn the key concepts of the Git source control system
- Step through the entire Git workflow
- Compare the different states in Git and compare between branches and commits
- Manage files with Git (move, rename, delete) and update files managed outside Git
- Create and fork repositories on GitHub and push changes back after working after working on them locally
- Create branches and resolve merge conflicts like a pro



Figure 7. Git Complete: The definitive, step-by-step guide to Git

8. Mastering Microcontroller and Embedded Driver Development

Link to the course: [Mastering Microcontroller and Embedded Driver Development | Udemy](#)

- Understand Right ways of Handling and programming MCU Peripherals
- Develop Peripheral drivers for your Microcontroller
- Understand complete Driver Development steps right from scratch for GPIO,SPI,I2C and USART.
- Learn Writing peripheral driver headers, prototyping APIs and implementation
- Explore MCU data sheets, Reference manuals, start-up Codes to get things done
- Learn Right ways of handling/configuring Interrupts for various peripherals
- Learn about Peripheral IRQs/Vector table/NVIC interfaces and many
- Learn about Configuration/status/Control registers of various Peripherals
- Demystifying behind the scene working details of SPI,I2C,GPIOs,USART etc.
- Explore hidden secrets of MCU bus interfaces, clock sources, MCU clock configurations, etc.
- Understand right ways of enabling/configuring peripheral clocks/serial clocks/ baud rates of various serial protocols
- Learn about MCUs AHB, APB bus protocols
- Learn about different MCU clocks like HCLK, PCLK, PLL,etc
- Learn to capture/decode/analyze traces of serial protocols on Logic analyzer
- Learn about Quick ways of debugging peripheral issues with case studies



Certificate no: UC-9303126e-5a46-48df-bed3-757a8282c3ed
Certificate url: ude.my/UC-9303126e-5a46-48df-bed3-757a8282c3ed
Reference Number: 0004

CERTIFICATE OF COMPLETION

Mastering Microcontroller and Embedded Driver Development

Instructors **FastBit Embedded Brain Academy, Kiran Nayak**

Vladimir Sirghi

Date **Oct. 17, 2021**

Length **28.5 total hours**

Figure 8. Mastering Microcontroller and Embedded Driver Development

9. STM32Fx Microcontroller Custom Bootloader Development

Link to the course: [Course: STM32Fx Microcontroller Custom Bootloader Development | Udemy](#)

- Write your own custom bootloader for your STM32Fx microcontroller and test
- Host Bootloader Communication implementation
- Custom Bootloader command packets
- Different boot modes of the STM32 Microcontroller
- Boot loader flash handling implementation : Sector Erase/Program/Mass erase
- Boot loader options bytes(OB) Program handling implementation
- Boot loader Flash sector protection status handling implementation
- Boot loader In application programming implementation (IAP)
- Vector table relocation of ARM cortex Mx processor



Figure 9. STM32Fx Microcontroller Custom Bootloader Development

10. Microcontroller Embedded C Programming: absolute beginners

Link to the course: [Microcontroller Embedded C Programming: Absolute Beginners | Udemy](#)

- C programming from scratch
- Data types, variables, declarations,
- Importance of Bit-Wise operators in "Embedded" programming
- Bit manipulation techniques using bit-wise operators and examples
- Bit extraction and working with memory mapped peripheral registers
- Other operators of 'C' (arithmetic, logical, relation, assignment)
- Functions in 'C'
- Implicit and explicit casting and its importance
- Pointers and casting
- Pointers arithmetic and importance of pointer datatypes
- Accessing memory mapped peripheral registers using pointers
- Structures and unions
- Importance of structures in peripheral register access
- Structure and bit fields
- Configuring peripheral register addresses using bit fields
- Structures and pointers
- interfacing LEDs, Keypad
- Const and volatile type qualifiers
- importance of volatile during compiler optimization
- const data, const pointer, const volatile explanation with examples
- Importance of const
- Loops : for, while, do while



Figure 10. Microcontroller Embedded C Programming: absolute beginners