

ECE 471/571 Real-time Operating Systems – 3CR

Instructor: Prof. Aleksander Malinowski <http://olek.matthewm.com.pl/>

Course Description

Advanced programming of small microprocessor-based systems using high-level programming languages applied to real situations; data acquisition, control, communication, small real-time operating systems. Software development for devices from a family of microcontrollers that is relevant to industrial applications.

Prerequisites by topics

Computer programming in C or C++ (ECE103) and microprocessors (ECE205)

Textbooks and other required material

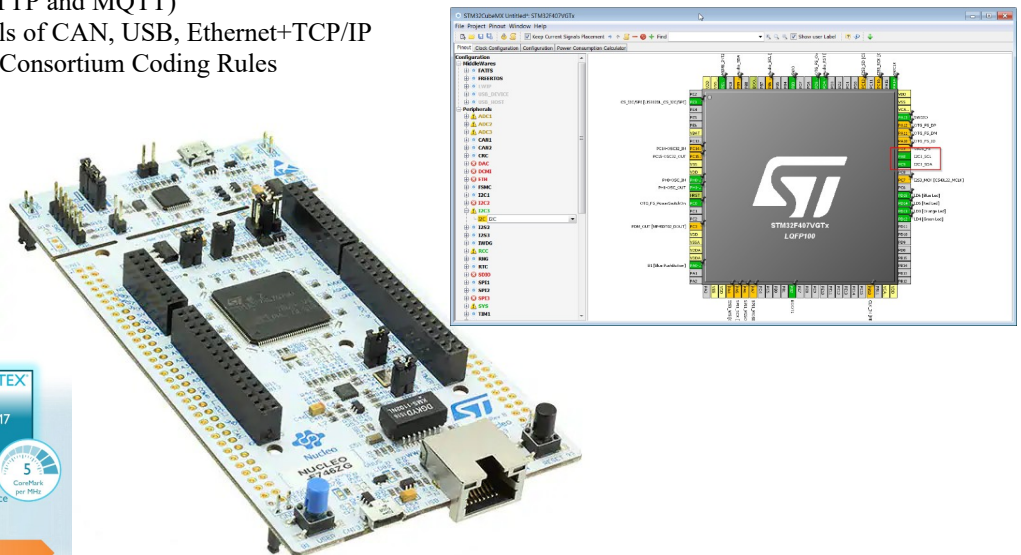
1. Mastering STM32 by Carmine Noviello, eBook - <https://leanpub.com/mastering-stm32> (\$30, donation-ware)
2. FreeRTOS (free documentation at <http://www.freertos.org/>)
3. **NUCLEO-F746ZG version of the Nucleo-144 board with Ethernet** (\$25 at DigiKey) <https://www.digikey.com/en/products/detail/stmicroelectronics/NUCLEO-F746ZG/5806779>
4. Additional materials posted on the course Web site and on the Internet
5. Your own computer running Windows, MacOS or Linux (preferably Debian or Red Hat) with HDD space

Course Objectives

1. Improve programming skills in a high-level programming language for embedded applications
2. Learn the hardware of the ARM Cortex-M family of microcontrollers (RISC)
3. Learn the core features of a small RTOS including setup, task management, and task communication
4. Learn how to use a popular embedded TCP/IP communication stack – LwIP – for Web interface and MQTT for IoT
5. Develop the mindset of multitasking, interrupt-driven, and energy efficient programming
6. Develop the mindset of event-driven data communication using UART/RS232, CAN bus, and USB
7. Develop an appreciation for efficiency of use of computing resources

Topics Covered

1. C90 and C99: Variables, Operators, Branching and Looping, Functions, Arrays, Structures and Pointers
2. Modular Programming: C Modules, Scope of Variables and Functions
3. RTOS: Multitasking, Basic and Commercial RTOS, Scheduler, co-routines, cooperative and preemptive task switching, inter-task communication using queues and semaphores, priorities and deadlocks
4. Hardware: Ports, Programmable Clock Circuitry, Interrupts, Timers, UART, ADC, DAC, DMA, Watchdog, Sleep
5. Algorithms: User Input/Output, Packet Communication, Lookup tables, Buffers (Single-use, Circular FIFO), Data Interpolation, Client-Server (HTTP and MQTT)
6. Networks: UART, Fundamentals of CAN, USB, Ethernet+TCP/IP
7. MISRA - Automotive Industry Consortium Coding Rules



Taking the Cortex-M Series to the Next Level

