Cyber-Physical Systems

TELE 7374: IoT High End Device Development

Spring 2025 Saturdays, 9:00 AM - 12:00 PM

Instructor: Rolando Herrero, PhD (e-mail: r.herrero@northeastern.edu)

Course Description: This class explores the technologies and techniques behind the ever-growing field of design and development of high end embedded smart devices in Cyber-Physical Systems. Specifically, it focuses on a hands-on approach based on Raspberry Pi embedded software development. The content revolves around Rust as a modern, Post-OOP, memory-safe alternative to traditional C, C++, and C++11 development and debugging in embedded devices. Applies memory architecture, and input and output interfaces (GPIO, I2C, UART, SPI, I2S) to high end devices. It covers the fundamentals of signal processing, including filter design, as well as TinyML. This enables interaction with analog sensors through sampling and quantization over *Analog-to-Digital Coverters* (ADCs). The course also discusses Yocto-based Linux as an operating system for high-end smart devices. Discusses multitasking, scheduling, interrupts, threads, processes, tasks, inter process communication, contention, drivers, semaphores, mutexes and shared memory in Linux. Additionally, it explores several related special topics, including the role of *Field Programmable Gate Arrays* (FPGAs) in smart device development, drivers in Linux, and the *Robot Operating System* (ROS/ROS2).

Prerequisite(s): None

Credit Hours: 4

Text(s): *No Required Text*

Recommended Text(s):

1. *"Datasheet: Raspberry Pi 4 Model B"*, `https://datasheets.raspberrypi.com/rpi4/raspberry-pi-4-datasheet.pdf`; 2024
2. *"The Rust Standard Library"*, `https://doc.rust-lang.org/std/index.html`; 2024
3. *"Yocto Project Reference Manual"*, `https://docs.yoctoproject.org/ref-manual/index.html`; 2024
4. *"Vivado Design Suite User Guide"*, `https://www.xilinx.com/support/documents/sw_manuals/xilinx2022_1/ug908-vivado-programming-debugging.pdf`; 2022
5. *"Verilog Reference Guide"*, `https://in.ncu.edu.tw/ncume_ee/digilogi/vhdl/Verilog_Reference_Guide.pdf`; 1999
6. *"Cyber-Physical Systems"*, R. Rajkumar, D. de Niz and M. Klein, 1st Edition, Addison-Wesley; 2017
7. *"Real-time Operating Systems: Book 1"*, J. Cooling, 2nd Edition, Lindentree Assoc; 2019
8. *"Linux Device Drivers Development"*, John Madieu, 2nd Edition, Packt Publishing; 2017
9. *"Embedded Software for the IoT: The Basics, Best Practices and Technologies"*, K. Elk, 2nd Edition, CreateSpace Independent Publishing Platform; 2017
10. *"Embedded System Design: Embedded Systems Foundations of Cyber-Physical Systems, and the Internet of Things"*, P.Marwedel, 3rd Edition,Springer; 2018

11. *"Fundamentals of IoT Communication Technologies"*, R. Herrero, 1th Edition, Springer-Nature; 2021

Course Objectives:

At the completion of this course, students will be able to:

1. Develop Software for High End Embedded Devices focussing on modern C++11 and Post OOP Rust over embedded Linux-driven multitasking
2. Integrate inputs and outputs with advanced embedded devices for sensing and actuation
3. Combine IoT connectivity to build Cyber-Physical Systems with high end embedded devices

Grade Distribution:

Labs	50%
Project	25%
Quizzes	5%
Midterm Exam	10%
Final Exam	10%

Course Policies:

- **General**
  - Laptops/smartphones are not to be used unless instructed to do so.
  - Exams are closed book, closed notes.
  - **No makeup exams will be given.**
- **Assignments**
  - Students are expected to work independently. **Offering** and **accepting** solutions from others is an act of **plagiarism**, which is a serious offense and **all involved parties will be penalized according to the Academic Integrity and Plagiarism Policies**. A student who is found cheating in any exam or test **will receive an automatic F for this course**. Discussion among students is encouraged, but when in doubt, direct your questions to the professor.
  - **No late assignments will be accepted under any circumstances**.

Academic Integrity:

A commitment to the principles of academic integrity is essential to the mission of Northeastern University. The promotion of independent and original scholarship ensures that students derive the most from their educational experience and their pursuit of knowledge. Academic dishonesty violates the most fundamental values of an intellectual community and undermines the achievements of the entire University.

As members of the academic community, students must become familiar with their rights and

responsibilities. In each course, they are responsible for knowing the requirements and restrictions regarding research and writing, examinations of whatever kind, collaborative work, the use of study aids, the appropriateness of assistance, and other issues. Students are responsible for learning the conventions of documentation and acknowledgment of sources in their fields. Northeastern University expects students to complete all examinations, tests, papers, creative projects, and assignments of any kind according to the highest ethical standards, as set forth either explicitly or implicitly in this Code or by the direction of instructors. Go to `http://www.northeastern.edu/osccr/academic-integrity-policy` to access the full academic integrity policy.

Student Accommodations:

Northeastern University and the *Disability Resource Center* (DRC) are committed to providing disability services that enable students who qualify under Section 504 of the Rehabilitation Act and the *Americans with Disabilities Act Amendments Act* (ADAAA) to participate fully in the activities of the university. To receive accommodations through the DRC, students must provide appropriate documentation that demonstrates a current substantially limiting disability. For more information, visit `http://www.northeastern.edu/drc/getting-started-with-the-drc`.

Diversity and Inclusion:

Northeastern University is committed to equal opportunity, affirmative action, diversity and social justice while building a climate of inclusion on and beyond campus. In the classroom, members of the University community work to cultivate an inclusive environment that denounces discrimination through innovation, collaboration and an awareness of global perspectives on social justice. It is my intention that students from all backgrounds and perspectives will be well served by this course, and that the diversity that students bring to this class will be viewed as an asset. I welcome individuals of all ages, backgrounds, beliefs, ethnicities, genders, gender identities, gender expressions, national origins, religious affiliations, sexual orientations, socioeconomic background, family education level, ability – and other visible and nonvisible differences. All members of this class are expected to contribute to a respectful, welcoming and inclusive environment for every other member of the class. Your suggestions are encouraged and appreciated.

Please visit `http://www.northeastern.edu/oidi` for complete information on Diversity and Inclusion.

TITLE IX:

Title IX of the Education Amendments of 1972 protects individuals from sex or gender-based discrimination, including discrimination based on gender-identity, in educational programs and activities that receive federal financial assistance. Northeaster's Title IX Policy prohibits Prohibited Offenses, which are defined as sexual harassment, sexual assault, relationship or domestic violence, and stalking. The Title IX Policy applies to the entire community, including male, female, transgender students, faculty and staff. In case of an emergency, please call 911. Please visit `www.northeastern.edu/titleix` for a complete list of reporting options and resources both on- and off-campus

Tentative Course Outline: The weekly coverage might change as it depends on the progress of the class.

Week	Content
Week 1 (1/11)	• RPI Architecture – Processor – Memory Architecture – I/Os – RTOS vs GPOS • C Programming on RPI
Week 2 (1/18)	• Multitasking on Linux – Threads and Processes – Interprocess Communication – Mutexes, Semaphores, Deadlocks, – Queues – Mailboxes • C++98 Programming on RPI
Week 3 (1/25)	• Linux Kernel – Process and Memory Management – File System – Device Drivers – Networking • C++23 Programming on RPI
Week 4 (2/1)	• Linux Device Drivers – Device types: character, network, block – Functions: probe, remove, open, close, read, write, and ioctl • Rust Programming – Basic: Functions, data types, control flows
Week 5 (2/8)	• Linux Device Drivers – Memory Management – I/O Operations • Rust Programming – Basic: ownership and borrowing, memory safety, vectors and strings

Week	Content
Week 6 (2/15)	• Yocto on RPI – Layers – Recipes – Bitbake • Rust Programming – Advanced: traits, generics, lifetimes
Week 7 (2/22)	• Yocto on RPI – Packages – Configuration – Build Environment • Rust Programming – Advanced: macros, smart pointers
Week 8 (3/1)	• RPI Processor Archicture [1] – 64-bit ARMv8 Programming – Instructions, Registers – Floating point operations • Rust Programming – Advanced: threads, channels, async/await
Week 9 (3/8)	• SPRING BREAK
Week 10 (3/15)	• Signal Processing and Machine Learning – Sampling and Quantization Revisited – Real-Time digital filters • Rust Programming – Advanced: error handling
Week 11 (3/22)	• Signal Processing and Machine Learning – Real-Time frequency analysis – From filters to perceptrons: Classification and Clustering • Rust Ecosystem and Tools

Week	Content
Week 12 (3/29)	• Advanced Topics: FPGAs – Digital Logic Design – Hardware Description Languages (HDLs) – FPGA Architecture • Rust, C, C++ integration
Week 13 (4/5)	• Advanced Topics: FPGAs – Xilinx Artix 7 FPGA Architecture – Vivado Design Suite – Intro to Verilog
Week 14 (4/12)	• Advanced Topics: FPGAs – Verilog Development
Week 15 (4/19)	• Advanced Topics: Robot Operating System – nodes, topics, services, and actions – messages and services – networking and communication