COLLEGE OF ENGINEERING, UIC

CS494 Edge Computing Systems

(Credit Hours: 3 undergraduate, 4 graduate)

Spring 2026

Edge computing is where real systems meet real constraints. In this course, we build end-to-end pipelines that run where data is generated, not where it is convenient. You will work with sensing, on-device processing, networking, and lightweight inference to make systems that respond in the time available, with the power available, over the network you actually have.
By the end of the term, you should be able to take a concrete application scenario (for example: environmental sensing, innovative infrastructure, scientific instruments, or mobile platforms), choose an edge architecture that fits constraints, implement it, and justify the tradeoffs you made. The course is project-forward and deployment-forward: you will prototype early, test in a realistic setting, and iterate based on measurements.

Instructor and Course Details

Course:	CS 494
Section:	1
Credit Hours:	3 undergraduate, 4 graduate
Days/Times:	Tuesday and Thursday 02:00 PM to 03:15 PM
Location:	CDRLC 1406
Course website:	Edge Computing Systems
Grades:	Posted in Canvas

Instructor:	Michael E. Papka
Email:	papka@uic.edu
Office:	CDRLC 5407
Drop-In Office Hours:	Tuesday 03:30 PM to 04:30 PM; Thursday 12:30 PM to 01:30 PM
Drop-In Hours Location:	CDRLC 5407
Ad hoc Office Hours:	Slack and Zoom as needed (Slack is a great way to reach me)
Phone:	(312) 996-2994 (rarely answered)
Teaching Assistant:	Zhong Zheng
Email:	zzheng33@uic.edu
Drop-In Office Hours:	TBD
Drop-In Hours Location:	TBD
Ad hoc Office Hours:	TBD
Phone:	TBD

Course Information

2.1 Catalog Course Description

Edge computing brings computation out of the cloud and onto small, distributed devices physically located where data is generated. This shift enables latency-sensitive, privacy-aware, and resilient systems across various applications, including smart cities, environmental monitoring, and scientific instrumentation.

Students will design and implement full-edge pipelines that integrate embedded processors, sensors, communication protocols, and on-device machine learning. While cloud computing focuses on elastic resources and HPC emphasizes centralized performance at scale, edge systems demand localized decision-making under power, bandwidth, and size constraints.

Prerequisite(s): Grade of B or better in *CS 361 Systems Programming* or consent of the instructor. Recommended background includes: data structures, systems programming, networking basics, and Python.

2.2 Course Goals and Learning Outcomes

2.2.1 Course objectives:

The primary goal of this course is to provide a comprehensive introduction to edge computing systems and their role in enabling low-latency, privacy-aware, and resilient computation close to where data is generated. Students will learn how edge devices, such as sensors, microcontrollers, and single-board computers, support real-time decision making without relying solely on cloud infrastructure.

This course emphasizes the design, deployment, and evaluation of complete edge pipelines that integrate embedded hardware, networking protocols, local data processing, and lightweight on-device machine learning. Through hands-on labs and a semester-long team project, students will develop the practical skills necessary to build functional edge systems that operate under real-world constraints, including power, bandwidth, environmental variability, and limited compute resources.

By the end of the course, students will be positioned to apply edge computing principles to applications in smart cities, environmental sensing, scientific instrumentation, and other domains where localized intelligence is essential.

2.2.2 Learning Outcomes:

- **Understand edge computing and its applications:** Describe the role of edge computing in modern systems. Explain why latency, privacy, bandwidth limitations, and resiliency motivate computation at the network edge.
- **Comprehend edge hardware and embedded systems:** Identify and explain the fundamental components of edge devices, including microcontrollers, single-board computers, and sensors. Understand how resource constraints shape system design.
- **Network communication and edge-cloud interaction:** Explain and utilize low-power communication protocols such as MQTT, BLE, and LoRa. Understand how edge devices interface with cloud services when needed, and when fully local processing is required.

- **On-device machine learning and inference:** Deploy lightweight ML models (for example, TinyML, TensorFlow Lite) on edge hardware. Evaluate trade-offs among accuracy, latency, memory, and power consumption.
- **Systems design, integration, and deployment:** Develop complete edge computing systems that integrate sensing, data processing, communication, and local decision-making. Address constraints such as power efficiency, network limitations, and operational robustness.
- **Project-based engineering and collaboration:** Apply software engineering best practices, version control, documentation, testing, and structured project management to design, prototype, and evaluate a fully functional edge system.
- **Critical thinking and real-world impact:** Analyze and justify design decisions, including system architecture, model selection, and deployment strategies. Understand the ethical, societal, and environmental implications of edge AI technologies.

2.3 Growth Mindset Statement

Course materials and assignments can be complex and challenging, but they are crucial to your intellectual and personal growth and development. There are times you may need extra help. Students who attend class consistently, complete all assignments, thoughtfully engage with feedback on work, develop good study strategies, use office hours, and seek tutoring or other support when needed can develop a thorough understanding of the course material and ultimately succeed in the course.

2.4 Required and Recommended Course Materials

2.4.1 Required Materials:

- Core readings, lecture notes, and lab instructions will be distributed through the course website and form the basis for in-class discussion, assignments, and assessments.

2.4.2 Recommended Materials:

- Supplementary references and tool documentation may be suggested throughout the term as we select hardware, protocols, and inference stacks for labs and projects. These materials are intended for enrichment and are optional unless explicitly assigned.

2.5 Required Technology

Access to the needed computational resources will be provided to succeed in the course. Students may develop code anywhere, but grading of projects and assignments will be done on class resources.

2.6 Respect for Copyright

Please protect the copyright integrity of all course materials and content. Please do not upload course materials not created by you onto third-party websites or share content with anyone not enrolled in our course.

Course Policies and Classroom Expectations

3.1 General

- You are responsible for all material presented in class and in assigned readings. If you miss a class meeting, you are expected to obtain notes from a classmate before the next session.
- You are responsible for maintaining your own notes and records needed to succeed in the course. Lecture notes and other course materials are typically provided on the course website, but this is not guaranteed. Sample code may be shared in the course GitHub repository on a best-effort basis.

3.2 Major Projects and Deliverables

This course is organized around a single end-to-end edge system that you build as part of a project team iteratively. Most weeks contribute directly to the project. The intent is to make design decisions measurable and defensible.
This course treats edge computing as a systems engineering problem shaped by constraints. You are expected to design and build an end-to-end system that runs on real hardware and interacts with real data. Success is defined by what you can demonstrate, measure, and explain with evidence.

Core project artifacts:

- **Repository:** reproducible builds, clear structure, and minimal setup steps.
- **System design note:** short architecture description (what runs where and why), including constraints and tradeoffs.
- **Measurements:** latency, resource use, and reliability metrics appropriate for your system.
- **Deployment evidence:** photos, logs, or data demonstrating the system running in a realistic or representative setting.
- **Documentation:** a README that an external reviewer can follow to run the pipeline and reproduce results.

Milestone 1 (Prototype): a partially working (best case) or well-defined (worst case) end-to-end pipeline on target hardware with baseline measurements (best case) or clearly defined measurement expectations (worst case).

Milestone 2 (Field Test): evidence of execution in a realistic or representative setting with measured behavior under real conditions (best case), or a clearly documented field attempt with observed failures, bottlenecks, and incomplete components supported by logs, measurements, or traces (worst case). In both cases, teams must provide a concrete iteration plan grounded in evidence.

Final Project: a stable demo plus complete documentation, measurements, and a clear explanation of design choices, including what worked, what failed, and what would be done differently with additional time or resources.

3.3 Grading Policy and Point Breakdown

3.3.1 Undergraduate (3 credit hours):

Component	Weight	Notes
Homework / Prep Assignments	35%	no late work accepted
Milestone 1 (Prototype)	10%	missing milestones impact final grade
Milestone 2 (Field Test)	20%	must show realistic or representative deployment progress with evidence
Final Project	30%	demo, repository management, documentation, measurements
Participation	5%	discussion, peer review, team contribution

3.3.2 Graduate (4 credit hours):

Component	Weight	Notes
Homework / Prep Assignments	25%	no late work accepted
Milestone 1 (Prototype)	10%	missing milestones impact final grade
Milestone 2 (Field Test)	20%	must show realistic or representative deployment progress with evidence
Final Project	30%	demo, repository management, documentation, measurements
Graduate Project	10%	TBD
Participation	5%	discussion, peer review, team contribution

3.3.3 Letter Grade Mapping:

Points	Percentage	Grade
900	≥ 90	A
800	≥ 80	B
700	≥ 70	C
600	≥ 60	D
–	< 60	F

3.3.4 Additional Grade Information:

- Course grades will be maintained in Canvas. Students are responsible for regularly reviewing their grades and notifying the instructor promptly if a discrepancy or error is observed.

- The instructor will make a good-faith effort to keep the gradebook current. Grades are typically posted within two weeks of the due date for an exam, quiz, assignment, or project. Please do not inquire about the status of grading during this period.

3.4 Project Rubric for Milestones and Final Project

Project milestones and the final project are evaluated using a common rubric. The rubric applies across Milestone 1, Milestone 2, and the Final Project, with expectations increasing over time.

Rubric categories (used for Milestone 1, Milestone 2, and the Final Project):

- **System design and architecture (20%):** Clarity of the end-to-end system, including what runs where and why, and how design choices respond to constraints such as data size, computation, privacy, and power.
- **Implementation and engineering quality (20%):** correctness, reliability, and maintainability, including reproducible builds and execution on target or representative hardware.
- **Measurement and evaluation (25%):** appropriate measurements of performance, resource use, data movement, and power or energy, where feasible, with interpretation that informs design decisions.
- **Field testing and iteration (20%):** evidence of execution in realistic or representative settings, including observed failures, bottlenecks, and an iteration plan grounded in logs, traces, or measurements.
- **Documentation and communication (15%):** documentation sufficient for a third party to reproduce results and a clear explanation of assumptions, constraints, and design tradeoffs.

3.5 Qualitative Grading Guidelines for Project Work

Grades reflect correctness, completeness, and engineering quality under realistic constraints. The rubric categories above are the primary evaluation lens. A system that fails in operation but is supported by evidence, measurements, and analysis is evaluable and may receive a high grade. A submission that cannot be evaluated due to missing artifacts, evidence, or measurements will not.

Qualitative signals used across milestones and the final:

- **A-level:** strong performance across all rubric categories; runs reliably on class resources; measurements are well-motivated and interpreted; design choices are defensible under stated constraints; documentation enables reproduction without instructor intervention; code is organized and maintainable.
- **B-level:** largely correct and complete, with one or more weaknesses in rubric execution (for example: limited measurement quality, unclear tradeoff justification, incomplete reproducibility details, or reduced robustness); documentation may require some instructor intervention to run.
- **C-level:** partially correct or incomplete pipeline; limited evidence of realistic testing; unclear ownership of design decisions; documentation insufficient for reproduction; weaknesses span multiple rubric categories.
- **D/F-level:** no evaluable system submission, missing core artifacts, absence of meaningful measurements or evidence of execution, or academic integrity violations.

3.6 Assignment Policy

- Coding projects and assignments must be tested on class resources. You may develop anywhere, but grading will be done on class resources.
- Any project or assignment submitted for grading containing compile-time errors will receive zero points.
- The following statement must be included in any file you modify. It is your commitment that the work submitted is your own. Failure to include it will result in a lower grade.

Listing 1: Required DocBox for Any Files You Write or Modify

```
/*
 * Name: <Your Name>
 * Assignment: <Assignment>
 * Date: <Date>
 *
 * I certify that this is my own work and, where appropriate,
 * an extension of the starter code provided for the assignment.
 */
```

3.7 Extra Credit

Extra credit opportunities may be announced periodically in class or via Slack. Deadlines for extra credit are strict, and submissions must be made on time to receive credit.

3.8 Missed or Late Work Policy

Deadlines matter in this course. Late work is penalized, missing work carries additional consequences, and there is a single end-of-semester deadline for resolving issues. Students are responsible for notifying the instructor when something needs to be reviewed.

3.8.1 Late Work:

- There are no makeup assignments.
- Assignments submitted after the deadline lose 10% per day late, including weekends and university holidays.
- Work submitted more than three days late will receive a score of 0.

3.8.2 Missing Work and Grade Penalties:

- Missing assignments initially appear as -50 points in the Canvas to clearly flag incomplete work.
- Each missing assignment or project may result in a half-letter grade reduction.

3.8.3 Final Deadline and Required Notification:

- Any late or missing assignment or project may be submitted for a grade of 0 up to the last day of regularly scheduled class meetings (week 15), before finals week.

- Submitting work by this deadline removes the -50 placeholder and avoids the half-letter grade reduction associated with missing work.
- To have a submission, correction, or grading issue reviewed, you must submit the course regrade request form. This form is the official way to notify the instructor that something needs attention. Work submitted without a regrade request may not be reviewed.

3.9 Attendance and Participation Policy

Regular attendance and active participation are expected in all class sessions. While attendance itself does not earn points, this course uses a **penalty-credit model** in which attendance and participation may influence borderline final grade decisions.

- One unexcused absence is allowed per fifteen class meetings.
- Each unexcused absence beyond the grace allowance may result in a 1% deduction from the final course grade.
- Participation includes engagement in class discussions, peer feedback, teamwork contributions, and professional conduct during lectures and project activities.
- Consistently failing to participate meaningfully may negatively affect final grade determinations in borderline cases.

Students should notify the instructor as soon as possible if unexpected circumstances interfere with attendance, participation, or timely completion of required work.

3.10 Academic Integrity

Good academic work is founded on honesty. Presenting work not personally produced is a grave offense. Cheating includes copying from others during exams or submitting work partly or entirely written by another. Plagiarism, whether intentional or not, occurs when students fail to acknowledge the sources of copied or paraphrased materials.

All students should review the UIC Community Standards and Academic Integrity Policy:

- Community Standards
- Academic Integrity
- Everything you do in this course must reflect your work. Copying all or part of someone else's work, including code from a website, is considered cheating.
- Discussion among classmates is allowed and encouraged, but it should not cross into collaboration. Discussion should occur in an open and inclusive space, such as Slack.

A student found responsible for academic dishonesty is subject to disciplinary action as outlined by the Student Disciplinary Policy.

3.11 Email Expectations and Electronic Communication

This course uses Slack for most course communication. Students are responsible for monitoring messages sent to their `@uic.edu` email account, communications posted in Canvas, and messages sent through the course Slack workspace. Important announcements may occasionally be sent to students' official university email accounts and must be read in a timely manner.

The instructor will attempt to respond promptly to course-related messages sent to either the instructor's `@uic.edu` address or Slack.

- The course website is the primary location for course materials. Canvas is used for posting grades and other course-related information.
- Project and assignment handouts will be distributed as a README in the corresponding GitHub repository.
- Additional clarifications or updates will be posted in the assignments Slack channel. Students are responsible for monitoring Slack channels regularly and setting up appropriate notifications.

Students are expected to check the course website, Slack, and Canvas regularly for course updates, deadlines, and grading information. The primary course website is Edge Computing Systems.

For technical issues related to Canvas or other supported instructional technologies (excluding the course website, Slack, and resources specific to the course), students should contact the Learning Technology Solutions team at LTS@uic.edu.

3.12 Generative AI Use Policy

Generative AI tools may be used in this course to support learning, exploration, and development. However, their use must be transparent, ethical, and consistent with the goals of the course. Students are responsible for understanding when and how such tools may be used appropriately.

3.12.1 Permitted Use:

- Generative AI may be used to assist with brainstorming, debugging, exploring alternative approaches, and generating small code snippets.
- AI tools may be used as a learning aid to help understand concepts, APIs, or system behavior, but not as a replacement for your own problem-solving or design work.
- AI may be used to help interpret error messages, suggest test cases, draft small unit tests, and identify potential performance bottlenecks, provided you validate the results and can explain the outcome.

3.12.2 Required Disclosure:

- Use of generative AI must be clearly disclosed in your submission.
- You must describe what tool was used and how it contributed to your work.
- Failure to disclose AI use may be treated as a violation of academic integrity.
- Disclosure must include:
  1. the tool name and version if available,
  2. the specific task (e.g., debugging MQTT reconnect logic), and
  3. what you accepted, modified, or rejected.
- If AI materially influenced code, design, or analysis, include an *AI Use Log* section in your README or report with short excerpts of prompts and outputs sufficient to support review.

3.12.3 Prohibited Use:

- Generative AI may **NOT** be used during quizzes, exams, or other assessments intended to evaluate individual understanding.
- Submitting AI-generated content as if it were entirely your own work is not permitted.
- Generative AI may not be used to fabricate measurements, logs, deployment evidence, citations, or experimental results.
- Generative AI may not be used to produce a complete assignment solution where the primary intellectual contribution is expected to be yours, including full project writeups and end-to-end codebases.

3.12.4 Relationship to Academic Integrity:

- All submitted work must reflect your own understanding and effort, even when AI tools are used.
- Improper use of generative AI, including undisclosed use or reliance that replaces independent work, may be treated as academic dishonesty.
- You should assume you may be asked to explain and defend any part of your submission in a short oral check, including design choices, measurements, and specific code paths. If you cannot explain it, you should not submit it.

3.12.5 Examples of Use:

- **Allowed:** *Suggest possible causes of an intermittent bug and propose tests to isolate it.* You run the tests, report outcomes, and document what you changed.
- **Allowed:** *Explain an unfamiliar API or error message and suggest a minimal fix.* You validate the fix, can explain why it works, and disclose the assistance.
- **Allowed:** *Propose alternative designs and outline tradeoffs.* You choose an approach, justify it, and clearly distinguish your reasoning from the tool output.
- **Not allowed:** *Produce a complete solution or full writeup for an assignment intended to assess your own work.* That replaces the primary intellectual contribution expected from you.
- **Not allowed:** *Invent or embellish results, measurements, citations, logs, quotes, or other evidence.* Any reported outcomes must be real, reproducible, and attributable.
- **Not allowed:** *Use AI during quizzes, exams, or other closed-book assessments unless explicitly authorized.* If permitted in a specific assessment, the allowed scope will be stated in writing.

This policy applies only to this course. Policies governing generative AI use may differ in other classes, departments, or university contexts.

Accommodations

4.1 Disability Accommodation Procedures

UIC is committed to the full inclusion and participation of people with disabilities in all aspects of university life. If you face or anticipate disability-related barriers while at UIC, please connect with the Disability Resource Center (DRC) at `https://drc.uic.edu`, via email at drc@uic.edu, or call

(312) 413-2183 to create a plan for reasonable accommodations. To receive accommodations, you must disclose the disability to the DRC, complete an interactive registration process with the DRC, and provide me with a Letter of Accommodation (LOA). Upon receipt of an LOA, I will gladly work with you and the DRC to implement approved accommodations.

4.2 Religious Accommodations

I will avoid scheduling exams or requiring student projects to be submitted on religious holidays. If you wish to observe your religious holidays, please notify me by the tenth day of the semester of the date when you will be absent, unless the religious holiday is observed on or before the tenth day of the semester. In such cases, please notify me at least five days in advance of the date when you will be absent. I will try to honor your request and not penalize you for missing class. If an examination or project is due during your absence, you will be given an exam or assignment equivalent to the one completed by those students in attendance. Students may appeal through campus grievance procedures for religious accommodations. Reference: religious calendar.

Classroom Environment

5.1 Inclusive Community

UIC values diversity and inclusion. Regardless of age, disability, ethnicity, race, gender, gender identity, sexual orientation, socioeconomic status, geographic background, religion, political ideology, language, or culture, we expect all members of this class to contribute to a respectful, welcoming, and inclusive environment for every other member of our class. If there are aspects of the instruction or design of this course that result in barriers to your inclusion, engagement, accurate assessment, or achievement, please notify me as soon as possible.

5.2 Name and Pronoun Use

If your name does not match the name on my class roster, please let me know as soon as possible. I am committed to using your preferred name and pronouns in this course.

My pronouns are [he/him]. You are welcome to share your pronouns with me if you would like. For more information about pronouns, see: `https://www.mypronouns.org/what-and-why`.

5.3 Community Agreement / Classroom Conduct Policy

- Be present and attentive by minimizing distractions such as unrelated device use.
- Be respectful of the learning space and community, including avoiding side conversations.
- Use preferred names and gender pronouns.
- Assume goodwill in all interactions, even during disagreement.
- Engage in dialogue that values a free and respectful exchange of ideas.
- Debate ideas and concepts, not individuals.
- Be willing to collaborate constructively and share helpful learning strategies.
- Be mindful of one another's privacy and do not invite outsiders into our classroom or online spaces.

5.4 Content Notices

This course provides an open space for critical and civil exchange of ideas, inclusive of a variety of perspectives and positions. Some readings, discussions, or course content may expose you to ideas or subjects that may challenge you, cause discomfort, or recall past negative experiences.

All topics will be approached with dignity, humanity, and respect for scholarly inquiry. If you would like me to be aware of a specific topic of concern, please get in touch with me during office hours or via email.

Resources: Academic Success, Wellness, and Safety

We all need the help and the support of our UIC community. Please visit my drop-in hours for course consultation and other academic or research topics. For additional assistance, please get in touch with your assigned college advisor and see the support services available to all UIC students.

6.1 Academic Success

- UIC Tutoring Resources: https://tutoring.uic.edu/
- College of Engineering Learning Center: https://uofi.app.box.com/s/m4gmbsu9ed41wul35qwkmru6xaizu7f8
- Equity and Inclusion in Engineering Program: https://engineeringequity.uic.edu/
- UIC Library: http://library.uic.edu/ and UIC Library Research Guides: http://researchguides.uic.edu/
- Offices supporting the UIC Undergraduate Experience and Academic Programs: https://osb.uic.edu/offices/
- Student Guide for Information Technology: https://uofi.app.box.com/s/0nyho51r24yo0pivafibb1nvqogjm401
- First-at-LAS Academic Success Program: https://first-at-las.uic.edu/

6.2 Wellness

- Counseling Services: https://counseling.uic.edu/
- Access UandI Care Program: https://dos.uic.edu/student-assistance/uicare/
- Campus Advocacy Network (Title IX): TitleIX@uic.edu and http://can.uic.edu/

6.3 Safety

- UIC Safe App: https://ready.uic.edu/toolkit/uic-safe-app/ (please download for your safety)
- UIC Safety Resources: https://ready.uic.edu/safetyresources/
- Night Ride: https://transportation.uic.edu/night-ride/
- Emergency: Dial 5-5555 from a campus phone, or (312) 355-5555 from a cell phone

Course Schedule

Week	Date(s)	Topic(s) / In-Class Activities	Readings / Work Due
00	01/13, 01/15	Syllabus and Edge Video	HW00 Assigned
01	01/20, 01/22	Linux and Data Visualization	HW01 Assigned
02	01/27, 01/29	Edge Computing	HW00 and HW01 Due

Disclaimer: While every attempt is made to provide an accurate overview of the course, unanticipated circumstances or events may make it necessary to modify the syllabus during the semester. Adjustments may be made to account for student progress, experiences, needs, or other unforeseen outside factors. Any changes will be made with as much advance notice as possible to not disadvantage students. Please refer to the current version on the course website for the most up-to-date information.