AI and Mechatronics

AIM / COURSE SYLLABUS / 2025–2026

School of Science and Engineering at Townview
David McLoda | dmcloda@dallasisd.org | robotics.mcloda.com



Build systems that sense and respond

How can a device recognize a gesture, respond to a voice, interpret movement, or make a decision from sensor data? In AIM, you investigate these questions by designing, building, programming, and testing physical computing systems. The course combines electrical and mechanical engineering with computer science, artificial intelligence, and technical communication.

You begin with circuits and embedded programming, then connect sensors, cameras, microphones, and actuators to increasingly capable systems. Tiny machine learning (TinyML) brings AI to microcontrollers; edge computing extends the work to platforms such as Raspberry Pi and NVIDIA Jetson Nano. Guided laboratories prepare you for independent projects in which your team chooses an approach, manages constraints, and explains its results.

What you will learn to do

- Build circuits and explain how inputs, processing, and outputs work together.
- Program microcontrollers and computers using C/C++, Python, libraries, and technical documentation.
- Collect and label data, train or select a model, deploy it, and test its behavior.
- Apply computer vision, audio recognition, motion classification, and language models in projects.
- Troubleshoot hardware and software, revise designs, and explain technical tradeoffs.
- Collaborate, manage milestones, pitch a feasible product, and communicate your individual contribution.

How the course works

Each sequence moves from guided practice to a demonstrated application. You will build and test circuits, maintain an engineering record, research unfamiliar problems, and use feedback to revise your work. Open-ended projects require initiative: a successful prototype must be accompanied by an explanation of how it works and what its limitations are.

Your engineering record

Keep an organized notebook and digital record of designs, code, observations, troubleshooting, and revisions. Save evidence of working prototypes and prepare presentations that explain your decisions. The final capstone brings together a prototype, demonstration, and written project narrative.

Core platforms: Arduino • Raspberry Pi • NVIDIA Jetson Nano and JetBot • XIAO ESP32S3 Sense
Learning tools: Arduino IDE • Python • Google Colab • SenseCraft • sensor and fabrication tools

Course sequence and studio work



Follow the daily agenda for pacing and project choices. Activities adapt to readiness and available equipment; AI applications may be revisited in independent projects.

Phase 1 Electronics and embedded programming

Construct Arduino circuits using inputs, indicators, sensors, and actuators. Practice programming, serial communication, library setup, and systematic debugging. Use guided labs to connect code with physical behavior before proposing an independent Arduino design.

Activities: LED and input/output labs; sensor readings; servo control; circuit troubleshooting; a team-selected Arduino project. **Deliverables:** functioning circuits, lab checks, notebook entries, and a short project presentation.

Phase 2 Fabrication and environmental systems

Develop soldering and repair skills, including headers, prototyping boards, solder-bridge removal, and desoldering. Set up Raspberry Pi systems and integrate external devices. Build LED-matrix displays, incorporate a real-time clock, and connect environmental measurements to displayed output.

Activities: soldering practice and repairs; scrolling displays; particulate sensing or temperature-and-humidity sensing; data collection and a room-condition classifier. **Deliverables:** assembled circuits, an integrated sensor application, and an explanation of the data path.

Phase 3 Robotics and embedded AI

Set up JetBot, explore movement and teleoperation, and investigate image-based collision avoidance and line or road following. Collect FREE and BLOCKED images, train in Google Colab, transfer the model, and test robot behavior. Extend work to XIAO ESP32S3 Sense cameras, microphones, connectivity, and SenseCraft models.

Activities: labeled image collection; model training and deployment; live robot testing; camera and microphone onboarding; vision, audio, motion, and edge-AI projects. **Deliverables:** data and model records, working demonstrations, and explanations of recognition errors and system limitations.

Phase 4 Independent design and product development

Develop an original project proposal, present a Shark Tank-style pitch, and give substantive feedback on classmates' ideas. Form a team, define responsibilities, acquire approved materials, and work through repeated build-and-test cycles. Communicate the intended user, purpose, feasibility, and value of the product.

Activities: brainstorming; proposals and peer review; prototyping; debugging; demonstrations; final presentations and narratives. **Deliverables:** a prototype, engineering record, presentation, and individual explanation of contributions and design decisions.

A recurring engineering process

Define → Design → Build → Test → Revise → Explain

At each stage, connect your decisions to evidence. Record what you expected, what happened, and what you changed. A limitation you can diagnose and explain is part of the engineering record.

AI applications and project pathways

Project work spans four AI areas. Model training creates or adapts a model from data; inference uses a model to make a prediction. Conventional rules, calculations, and control code also remain essential parts of an engineered system.

Application area	Activities and concepts	Demonstration of understanding
Computer vision	Image classification and object detection; recognizing objects, faces, scenes, and hand gestures using camera input.	Explain the image input, model prediction, and resulting action; distinguish classification from locating an object.
Audio processing	Keyword spotting, wake-word detection, and voice commands; on-device microphone input and recognition.	Connect a sound or command to a response; distinguish a target-word detector from broader speech transcription.
Motion classification	Activity and gesture recognition using accelerometer and gyroscope data from an inertial measurement unit (IMU).	Explain how movement is represented by sensor data and classified; distinguish this from camera-based gestures.
Language models at the edge	Language and vision-language models on devices such as Raspberry Pi; reading, generating, and interacting with text.	Explain the input or prompt, the model response, and the computing platform; identify limitations in the output.

From images to robot behavior

The JetBot sequence uses 100 usable images in each of two classes: FREE and BLOCKED. Collect and label images, train the model in Google Colab, transfer it to the robot, and run the live collision-avoidance application. Test the connection between training examples and observed behavior, and record cases that the system handles poorly.

Representative student designs

Camera-based rock-paper-scissors: use a SenseCraft recognition model, game logic, and a servo mechanism to connect a recognized gesture with a physical response. Voice-operated unit converter: combine Raspberry Pi speech processing with extraction of quantities and programmed unit conversion.

Electronic Minesweeper: integrate an LED matrix, controls, displays, and game logic. Color-responsive gripper: combine manual positioning with sensor-triggered servo control. These examples illustrate different design pathways; not every subsystem or final project uses AI.

Evaluating an AI system

Consider how data quality, labels, and operating conditions affect predictions. Discuss bias and limitations, explain errors, and distinguish intended behavior from what testing demonstrates. Select a model and platform that fit the project rather than treating AI as a requirement for every engineering decision.

Learning resources

Assigned resources include Harvard’s Fundamentals of TinyML, Machine Learning Systems hardware labs (mlsysbook.ai), and documentation from Arduino, Raspberry Pi, NVIDIA, and Seeed Studio. Follow the daily agenda for readings, tutorials, and project instructions.

Assessment and course expectations

How your work is assessed



Category	Weight	Typical evidence
Daily work	40%	Classwork, preparation, notebook entries, reflections, and routine checks.
Projects	20%	Laboratories, prototypes, demonstrations, and project documentation.
Major grades	25%	Tests and major assessments of technical understanding.
Six weeks test	15%	Six-weeks assessment.

Check PowerSchool regularly and follow the instructions and assessment criteria provided for each assignment.

Project quality and final narrative

For sensor-AI systems, evaluation addresses model deployment, sensor integration, physical responses, design purpose, and explanation. Be prepared to demonstrate your system and explain your own work. A group's prototype and each student's understanding are both important.

Your final narrative explains what you built and its purpose; inputs, processing, and outputs; changes to the design; challenges and solutions; one or two important code or circuit decisions; what works; and what you would improve. Write clear paragraphs for someone outside the class. The narrative assignment calls for approximately 500–700 words or one to two pages.

Preparation and communication

Bring your laptop, wired headphones, notebook or organized binder, pencils, and pens. Complete assigned readings and tutorials before class. Check the agenda, objective, and success criteria on arrival, and begin the warm-up promptly. Use district email for teacher communication, check email and PowerSchool daily, and follow class announcements. Before-school tutoring is available by advance arrangement.

Safe and responsible laboratory work

Follow the instructor's laboratory and safety procedures, remain with your assigned group, and use tools and equipment only as directed. Keep work areas and aisles clear. Return tools and components to their proper places, clean the workspace, save your work, and complete the required reflection before dismissal. Keep food away from the laboratory and follow classroom rules for personal devices.

Academic integrity and use of AI

Submit your own work and participate in your team's work. Copying, plagiarism, and unauthorized AI assistance are not permitted. AI tools may be used only when explicitly allowed by the instructor. Using an approved AI model as a project component does not replace your responsibility to explain the code, decisions, and results.