

Each project below maps to one or more academic ro

Robotics Project

Intro to Robotics: What Makes a Robot?

Input–Process–Output: The Universal Systems Model

Precision Measurement & Coordinate Systems

Circuit Basics: AC vs DC, Voltage, Current, Resistance

Design-a-Circuit Challenge: Night Light

Soldering Lab: Build Your Own LED Badge

Mechanical Systems: Gears, Levers & Linkages

Fluid Power Primer: Hydraulic Arm Build

Sensor Safari: What Can a Robot See, Hear & Feel?

Analog vs Digital: Logic Gate Challenge

Microcontroller Foundations: Your First Micro:bit Program

Program a Robot: SPIKE Prime Line Follower

Robot Interfaces & Data Transmission

Simulate First: Virtual Robotics in CoderZ / VEXcode VR

3D Design & Additive Manufacturing: Print a Robot Part

Jitterbug Soldering Kit

Arduino Battle Bots (Summer Camp)

LED Drone Gate Demo

Interactive Mirror Project

Capstone: Design, Build & Reengineer a Mission Robot

Innovation Academy — Robotics

botics competencies. See 'Competency Key' tab for code definitions.

Materials	Grade Range
Thames & Kosmos Robotics Smart Machines kit, demo videos, career profile cards, engineering notebooks	3-5
Micro:bit v2, USB cables, tactile buttons, LEDs, breadboards, laminated systems-model worksheets	4-6
Digital calipers, metric rulers, dial indicators, LEGO Technic parts, graph paper, coordinate-grid mats	5-8
Snap Circuits Pro kit, 9V batteries, multimeters, resistors (assorted), LEDs, safety posters	4-7
Breadboards, LEDs, photoresistors, transistors, resistors, 9V batteries, multimeters, circuit simulator (Tinkercad)	6-8

Through-hole LED badge kits, soldering irons (25W), solder, helping hands, safety glasses, fume extractors	7-8
LEGO SPIKE Prime sets, gear ratio worksheets, pulley kits, torque-vs-speed demo rigs	4-7
LEGO Technic Pneumatics set OR syringe-and-tubing hydraulic arm kits, water-based fluid, measuring cups	5-7
LEGO SPIKE Prime sets (ultrasonic, color, force sensors), Databot 2.0 sensor pucks, data-logging worksheets	5-8
Snap Circuits Logic Gate set OR breadboards with 74-series logic ICs, LEDs, switches, truth-table worksheets	7-8
Micro:bit v2, MakeCode editor (block-based), USB cables, Chromebooks/laptops	3-6

LEGO SPIKE Prime sets, SPIKE App (Word Blocks/Python), black electrical tape on whiteboard mats	5-8
LEGO SPIKE Prime + Micro:bit radio pairs, USB vs Bluetooth demo, wired vs wireless comparison chart	6-8
Chromebooks/laptops, CoderZ or VEXcode VR accounts, headphones, mission worksheets	4-8
Tinkercad accounts, FDM 3D printers (Bambu A1 or similar), PLA filament, calipers, sample broken parts	5-8
Jitterbug PCB kit (vibrating motor, coin-cell battery holder, LED eyes, SPST switch), soldering irons (25W), lead-free solder, helping hands, safety glasses, fume extractors, googly eyes + decoration supplies	6-8
Arduino Uno R3, motor driver shields (L298N), DC gear motors, caster wheels, chassis plates, LiPo or AA battery packs, jumper wires, RC receivers/transmitters OR Bluetooth modules (HC-05), combat arena barriers, safety shields	7-8

LED drone gate (demo unit) — details TBD pending Irfan's write-up	5-8
Interactive mirror (demo unit) — details TBD pending Irfan's write-up	5-8
LEGO SPIKE Prime sets, 3D-printed custom parts, Micro:bit add-ons, mission mat, scoring rubrics, engineering notebooks	6-8

Robotics Projects & Curriculum Map

Description

A hands-on introduction to the world of robotics. Students define what robotics, automation, and control systems are by comparing real-world examples (manufacturing arms, Mars rovers, household appliances). They build 2–3 simple Thames & Kosmos models to identify core robot components: power, sensors, controller, actuators. Class closes with a careers discussion and student-written 'Where will robots be in 2040?' predictions.

Students learn the universal systems model by building a Micro:bit 'reaction timer' — a clear Input (button) → Process (code) → Output (LED/display) chain. They then modify the system to include feedback (closed loop) and compare it to an open-loop version with no feedback. Concludes with students diagramming household objects as systems (toaster, thermostat, automatic door).

Students use digital calipers and rulers to measure LEGO Technic components to the nearest 0.1 mm, then plot robot chassis layouts on coordinate grids. They learn why precision matters in robotics (gear meshing, wheel alignment, sensor placement) and apply both direct (caliper) and indirect (coordinate-based) measurement to design a chassis footprint.

Students build simple series and parallel circuits, measure voltage/current/resistance with a digital multimeter, and see Ohm's Law in action. Lesson opens with an AC vs DC safety briefing (household outlets vs batteries) and closes with students diagnosing a 'broken' circuit using the multimeter.

Students first simulate the circuit in Tinkercad Circuits, then build a working night light on a breadboard that turns on automatically in darkness. Introduces the photoresistor as a sensor and the transistor as a switch. Students measure key voltages, troubleshoot, and document their design with a schematic.

A supervised introduction to soldering technique. Students review iron safety, tin their tips, and solder a simple through-hole LED badge PCB. They measure final voltages with a multimeter and take home a working wearable. Strong emphasis on shop safety, PPE, and proper joint inspection.

Students build a series of mechanisms — gear trains, lever systems, and four-bar linkages — to explore torque vs speed trade-offs. They calculate gear ratios, predict outcomes, then test. Concludes with a design challenge: build a geared mechanism that lifts a 500g weight using a single SPIKE motor.

Students build a working hydraulic robotic arm using syringes and tubing. They learn the difference between pneumatic (air) and hydraulic (liquid) power, why incompressible fluids transmit force efficiently, and calculate mechanical advantage. The arm is then used to move small objects in a timed challenge.

A rotation-style lab where students spend 15 minutes at each of 4 sensor stations: ultrasonic (distance), color, force/pressure, and temperature/humidity (Databot). They log readings, identify which real-world robots use each sensor type (self-driving cars, warehouse bots, medical devices), and discuss open vs proprietary hardware.

Students compare analog signals (a dimmer knob) with digital signals (a light switch), then build AND, OR, and NOT gates on a breadboard or Snap Circuits board. They fill in truth tables and design a simple 'two-key safe' that only opens when two conditions are met (AND gate).

Students learn what a microcontroller is and why it's the 'brain' of most hobby robots. Using the MakeCode block editor, they program their first three applications: name badge, dice simulator, and step counter. Introduces the concept of an operating system vs firmware in plain language.

Students build a two-motor driving base with a color sensor, then program it in Word Blocks (or Python for advanced students) to follow a black line. They iterate through three versions: on/off control, proportional control, and obstacle-aware follower (adds ultrasonic). Closes with a timed run scored by the class.

Students explore how robots communicate — USB, Bluetooth, Wi-Fi, and radio. They pair two Micro:bits over radio to send 'commands' from a handheld controller to a SPIKE Prime driving base. Discusses latency, range, and when to pick wired vs wireless. Includes a short segment on program/data security (why you wouldn't want an open radio channel on a real robot).

Before touching hardware, students complete 4–5 missions in a virtual robotics simulator. They debug code, test sensor thresholds, and see a full control loop run end-to-end. This reinforces that simulation is a real engineering step, not 'just a game,' and saves hardware wear-and-tear.

Students learn additive manufacturing fundamentals, design a replacement robot part (sensor bracket, wheel hub, or claw tip) in Tinkercad, and print it. They measure the printed part with calipers against their design, discuss shrinkage/tolerances, and iterate. Introduces G-code and CNC fundamentals at an age-appropriate level.

A student-favorite intro to electronics manufacturing. Each student solders together a Jitterbug — a small vibrating 'bug' robot powered by an offset-weight motor and coin cell. Covers iron safety, proper joint technique, polarity (LED, battery, motor direction), and post-build troubleshooting with a multimeter. Students finish by decorating their Jitterbug and racing them on a tilted track. Great for building shop-floor confidence before larger builds.

A week-long summer intensive. Teams build a remote-controlled combat robot from an Arduino chassis kit, wire the motor driver to the Arduino, write the drive-control sketch, and tune response curves. Students iterate across three test-and-reengineer cycles before the final tournament. Introduces the Arduino IDE, C/C++ syntax basics, PWM motor control, and serial debugging. Emphasis on mechanical durability — students learn that the best code can't save a bad chassis.

Live demonstration of an LED drone gate system. [Placeholder — Irfan to share project details, materials list, and exact learning objectives. Likely covers sensor-triggered lighting, microcontroller timing, and input/output loops in a visually exciting format suitable for school visits, open houses, and demo days.]

Live demonstration of an interactive mirror build. [Placeholder — Irfan to share project details, materials list, and exact learning objectives. Likely covers sensor input, display/output systems, and microcontroller logic in an engaging, relatable format for classroom and recruitment demos.]

The culminating project. Teams receive a mission (e.g., 'retrieve the sample from Zone C and return it in under 60 seconds'). They design, build, program, test, score, and then reengineer their robot based on failure data across three official runs. Students present their design decisions, cite which sensors/mechanisms they chose and why, and reflect on what they'd change next.



Core Competencies

CS1, CS2, CS3, CS4, RC1, RC3

CS1, CS4, CS5, CS9, RC1, MC2, MC4

CS6, CS7, CS8, MF1, CR1

EL1, EL2, EL3, EL4, EL8, EL9

EL4, EL5, EL6, EL8, EL9, EL10, RC2

EL3, EL4, EL7, EL8, EL9

MF1, MF3, MF4, CS6, CS8

MF1, MF2, MF3, MF4, RC1

EL5, RC1, RC2, RC3, RC4, MC5

EC1, EC2, EL4, EL8

**MC1, MC2, MC3, MC4, MC6, PR1, PR2,
PR3**

PR1, PR2, PR3, PR5, RC1, RC2, CS5,
MC6

EC3, MC5, PR6, RC4

CS9, PR1, PR2, PR5, EL6, MC3, MC4

CR1, CR2, PR4, CS6, CS7, CS8

EL3, EL4, EL7, EL8, EL9, MF1, RC1, CR3

PR1, PR2, PR3, PR5, MC1, MC2, MC4,
MC6, EL2, EL4, EL8, MF1, MF3, MF4,
RC1, RC4, CR1, CR3, CR4, CS1

**CS1, CS5, CS9, EL4, EL5, RC1, RC2,
MC4, MC6, PR5**

**CS1, CS4, CS5, EL4, EL5, RC1, RC2,
MC3, MC4, MC5, PR1, PR5**

**CR1, CR3, CR4, CS1, CS5, CS9, RC1,
RC2, MF3, PR1, PR5**

Academic Robotics

Codes used in the Projects sheet to map each project to academic robotics competencies.

Code	Category
CS1	Exploring Control Systems and Robotic Systems
CS2	Exploring Control Systems and Robotic Systems
CS3	Exploring Control Systems and Robotic Systems
CS4	Exploring Control Systems and Robotic Systems
CS5	Exploring Control Systems and Robotic Systems
CS6	Exploring Control Systems and Robotic Systems
CS7	Exploring Control Systems and Robotic Systems
CS8	Exploring Control Systems and Robotic Systems
CS9	Exploring Control Systems and Robotic Systems
EL1	Exploring Electrical Concepts
EL2	Exploring Electrical Concepts
EL3	Exploring Electrical Concepts
EL4	Exploring Electrical Concepts
EL5	Exploring Electrical Concepts
EL6	Exploring Electrical Concepts
EL7	Exploring Electrical Concepts
EL8	Exploring Electrical Concepts
EL9	Exploring Electrical Concepts
EL10	Exploring Electrical Concepts
MF1	Exploring Mechanical and Fluid Concepts
MF2	Exploring Mechanical and Fluid Concepts
MF3	Exploring Mechanical and Fluid Concepts
MF4	Exploring Mechanical and Fluid Concepts
RC1	Exploring the Components of Robotic Systems
RC2	Exploring the Components of Robotic Systems
RC3	Exploring the Components of Robotic Systems
RC4	Exploring the Components of Robotic Systems
EC1	Explaining Control Systems
EC2	Explaining Control Systems
EC3	Explaining Control Systems
MC1	Exploring Microprocessor/Microcontroller System Basics
MC2	Exploring Microprocessor/Microcontroller System Basics
MC3	Exploring Microprocessor/Microcontroller System Basics
MC4	Exploring Microprocessor/Microcontroller System Basics
MC5	Exploring Microprocessor/Microcontroller System Basics
MC6	Exploring Microprocessor/Microcontroller System Basics
PR1	Programming Control Systems
PR2	Programming Control Systems
PR3	Programming Control Systems
PR4	Programming Control Systems

PR5	Programming Control Systems
PR6	Programming Control Systems
CR1	Creating a Robotic System
CR2	Creating a Robotic System
CR3	Creating a Robotic System
CR4	Creating a Robotic System

cs Competency Reference Key

Competency
Define robotics, automation, and control systems.
Investigate careers in robotics.
Research the development and future of robotics.
Explain the universal systems model.
Identify open and closed loops in control systems.
Demonstrate precision measurement equipment and techniques.
Describe components or processes that typically require precision measurement.
Apply direct and indirect measurement systems and coordinate systems.
Simulate control and automation systems.
Describe the difference between alternating current (AC) and direct current (DC).
Describe the concepts of voltage, current, and resistance in electricity.
Identify safety precautions and information for electricity (AC and DC).
Explain the primary functions of electronic systems components.
Identify sensors.
Simulate electronic circuits.
Demonstrate soldering technique.
Create circuits.
Measure circuit values with a multimeter.
Design a circuit for a given purpose.
Identify the primary concepts and components of mechanical systems.
Explain primary concepts and components of a fluid power system.
Design a mechanical system.
Demonstrate a mechanical system.
Identify components of robotic systems.
Demonstrate how sensors are used to control robotic technology.
Research robotic hardware used in various industries.
Compare open and proprietary hardware components.
Compare analog electronics and digital electronics for control systems.
Describe the operation of basic logic circuits.
Identify the primary types of data transmission hardware.
Describe the function of an operating system.
Describe the essential components of a computing system.
Describe software applications within robotic systems.
Describe how computers are used to control automated systems.
Describe robotic systems interfaces.
Describe the purpose of a microcontroller/logic controller.
Design code for a given purpose.
Implement basic programming procedures.
Select the most appropriate programming language or platform for an application.
Describe the fundamentals of CNC.

Program a robotic system.

Ensure the security of programs.

Design a robot.

Use additive manufacturing.

Build a robot.

Reengineer the design of an existing robotic system.