ROBOTICS AND AUTOMATION COURSE

A nine-week syllabus for grades 6–9, ending with a maze-solving robot.

The class meets once a week for 2–2.5 hours. Each one opens with a short demo and then turns into lab time. Weeks 1–5 introduce one part at a time. In weeks 6–9 students build a robot that finds its own way through a maze, and they take the kit home when the course ends.

It suits an after-school club or a library program that wants something longer than a one-day event. Students don't need any technical background.

The nine weeks

Week	Topic	Skills
1	Arduino basics and LEDs	Circuits, setting up the editor
2	Inputs and breadboards	Digital and analog signals, wiring
3	Sensors	Reading inputs, if statements
4	Servos	Motor control, PWM
5	Combined project	Using inputs and outputs together
6	Robot chassis	Mounting and safe wiring
7	Bumpers	Prototyping, first drive tests
8	Navigation code	Autonomous programming
9	Showcase	Testing, debugging, presenting

How a class runs

Part	Time	What happens
Demo	20–30 min	A short talk and a live demo of the week's new part
Lab	60–75 min	Students build and code while instructors circulate
Break	15 min	Snack, away from the benches
Project time	30–45 min	Troubleshoot, improve, and save working code

Week by week

Week 1. Arduino basics

Topics	Lab safety, the course plan, the Arduino board and its pins, what's in the kit, setting up the editor.
Lab	Practice on the breadboard, light an LED, and upload and change the Blink example.
Goal	Every student programs a working circuit.
Prep	Test every kind of laptop ahead of time. Bag the kits so each student gets one.

Week 2. Digital and analog inputs

Topics	Digital versus analog signals, series and parallel circuits, reading resistor bands.
Lab	Read a push button. Read a potentiometer and print its values to the Serial Monitor.
Goal	Change Blink so it reacts to a button, then make an LED that dims with a knob.
Prep	Bring a multimeter to show the voltage changing as the knob turns.

Week 3. Sensors and if statements

Topics	What sensors measure, and using if statements to react to readings.
Lab	Test limit switches and a color sensor. Print the readings and pick thresholds.
Goal	An LED or buzzer turns on when a sensor changes state.
Prep	Bring a few colored cards so color readings are consistent.

Week 4. Servos

Topics	How servos work, pulse width modulation, and the difference between positional and continuous servos.
Lab	Control a servo's position with a button or a potentiometer.
Goal	Each student has a working moving part.
Prep	Show both servo types side by side now. Students mix them up later if they don't see the difference early.

Week 5. Putting inputs and outputs together

Topics	Sensors and motors in the same program, and a step-by-step way to troubleshoot.
Lab	Build a security gate. A servo arm lifts when a sensor is triggered, and a buzzer sounds.
Goal	Students demo the gate and explain how they found their bugs.
Prep	This is the halfway point. Check in with anyone who's behind before the robot build starts.

Weeks 6–7. Planning and building the robot

Topics	How maze solving works, bumper logic, and planning a build in steps.
Lab	Assemble the chassis, mount the Arduino and battery, and wire everything safely.
Goal	The robot drives with simple code and both bumpers register a hit.
Prep	Have the practice maze set up by the end of week 6 so students can test against real walls.

Weeks 8–9. Navigation and the showcase

Topics	Autonomous navigation, loops, tuning turn times, and cleaning up code.
Lab	Tune the robot, run timed maze trials, then hold the showcase in week 9.
Goal	A finished maze-solving robot. Every kit goes home with its builder.
Prep	Invite families to week 9 and print certificates ahead of time.

The maze robot

Two continuous-rotation servos drive the robot, and two limit switches on the front act as bumpers. When a bumper hits a wall, the robot backs up and turns away. It's a crude strategy, but it gets through most beginner mazes, and there's plenty of room to improve on it.

Starter code for the maze robot

```
// Bump-and-turn maze robot
#include <Servo.h>

Servo leftWheel, rightWheel;
const int LEFT_BUMP = 2;
const int RIGHT_BUMP = 3;

void drive(int l, int r) { leftWheel.write(l); rightWheel.write(r); }
void forward()   { drive(180, 0); }    // wheels face opposite ways
void reverse()   { drive(0, 180); }
void turnLeft()  { drive(0, 0); }
void turnRight() { drive(180, 180); }

void setup() {
  leftWheel.attach(9);
  rightWheel.attach(10);
  pinMode(LEFT_BUMP, INPUT_PULLUP);
  pinMode(RIGHT_BUMP, INPUT_PULLUP);
}

void loop() {
  if (digitalRead(LEFT_BUMP) == LOW) {        // hit on the left
    reverse();   delay(400);
    turnRight(); delay(350);
  } else if (digitalRead(RIGHT_BUMP) == LOW) { // hit on the right
    reverse();   delay(400);
    turnLeft();  delay(350);
  } else {
    forward();
  }
}
```

Turn times depend on the battery, the floor, and the wheels. Have students time a 90-degree turn on their own robot and replace the 350 with that number.

Suggested grading

Area	Points	Full points
Maze run	40	The robot reaches the end in at least one of three tries
Code	30	Readable, commented, and behaves the same way every run
Build log	20	Notes on what they tried, what broke, and how they fixed it
Showcase	10	Can explain how the robot decides what to do

Materials

Each student kit has an Arduino Uno or compatible board (not a Mega), a breadboard and jumper wires, resistors and LEDs, one positional servo, two continuous-rotation servos, two limit switches, a color sensor, a robot chassis with wheels, and a USB cable.

Instructors need maze walls (cardboard or foam board works), tape, zip ties, spare parts, a few charged battery packs, and a projector for demos.

Students bring a laptop with Google Chrome. School Chromebooks are fine if the board is a genuine Uno R3 or R4, since many clones won't upload from ChromeOS. They should also bring a snack and water.

Staffing and tips

Plan on a lead instructor plus one assistant for every 10 students. Three or four members of an FTC team can cover that by taking turns.

- Have students check their wiring and code after every change. Small steps make bugs easy to find.
- Save a copy of working code with the week in the name (maze_week8_v2) before changing anything.
- Pass new parts around and put a close-up on the projector before anyone wires them.
- Pair a confident student with one who's newer. Both learn more.
- Keep a bin of spare parts. Something breaks every week, and a spare servo can save a lesson.

For students who finish early

- Swap the bumpers for an IR sensor and follow a line instead.
- Wall following. Keep one bumper lightly touching the wall and trace the maze.
- Add an ultrasonic sensor so the robot turns before it hits anything.

Questions? Email **sparkgearsinitiative@gmail.com** or call **571-429-9220**. You can request a mentor or a supply kit at sparkgearsinitiative.org/for-teams.html.