

Lesson 4 Teacher Lesson Plan

Level 1 C++ Foundations Hello ESP32

Lesson 4 is the first hardware lesson. Students connect an ESP32, upload a simple Arduino sketch, and use the Serial Monitor to observe setup and loop. No motors, sensors, or external wiring are used.

Lesson Snapshot

Item	Plan
Audience	Middle and high school robotics students
Length	60 to 90 minutes
Prerequisite	Lessons 1 to 3 and a laptop with Arduino IDE plus ESP32 board support
Big idea	ESP32 programs still use C++ ideas, but Arduino sketches start with setup and loop instead of main.
Student product	A working ESP32 sketch that prints messages to the Serial Monitor

Learning Objectives

- Identify the ESP32 as a microcontroller board.
- Select an ESP32 board and port in Arduino IDE.
- Explain the difference between setup and loop.
- Upload a sketch to the ESP32.
- Use `Serial.begin` and `Serial.println` to view board output.
- Modify and re-upload a safe no-motion sketch.

Materials Per Student Pair

- One ESP32 DevKit-style board.
- One USB data cable, not charge-only.
- Laptop with Arduino IDE installed.
- ESP32 board support installed through Boards Manager.
- Student workbook and starter sketch.

Setup Reference

Item	Recommended Setting
------	---------------------

Board Manager URL	https://espressif.github.io/arduino-esp32/package_esp32_index.json
Board package	esp32 by Espressif Systems
Common board choice	ESP32 Dev Module, unless your exact board is listed
Serial Monitor speed	115200 baud
Safety rule	No external wiring, motors, or robot battery in this lesson

Timing Guide

Time	Activity
0 to 10 min	Review desktop C++ and introduce the ESP32 board.
10 to 25 min	Check Arduino IDE, board package, board selection, and port.
25 to 40 min	Teacher demo: upload Hello ESP32.
40 to 60 min	Students upload, open Serial Monitor, and observe output.
60 to 75 min	Students modify messages and re-upload.
75 to 90 min	Troubleshooting, skill check, and exit response.

Starter Sketch

```

void setup() {
  Serial.begin(115200);
  delay(1000);

  Serial.println("Hello, ESP32!");
  Serial.println("Setup runs once.");
}

void loop() {
  Serial.println("Loop is running.");
  delay(1000);
}

```

Teaching Notes

- Keep the first hardware win small: board connected, sketch uploaded, Serial Monitor showing output.

- Explain that Arduino hides main, but the board still runs C++ code.
- Say setup runs once after reset; loop repeats until power is removed or a new sketch is uploaded.
- If upload fails, check cable, port, board choice, and whether the BOOT button must be held during upload.