

Arduino

Teachers' Book



Level 1

Rana Dajani

Arduino Teachers' Book - Level 1



Published by **LKD Educational Resources 2023**

Amman - Jordan

Tel: +962 6 5374141

Fax: +962 6 5516404

P.O.Box: 851346

Email: info@lkd.com.jo / info@lkdebooks.com

Websites: www.lkd.com.jo / www.lkdebooks.com

 **Author**

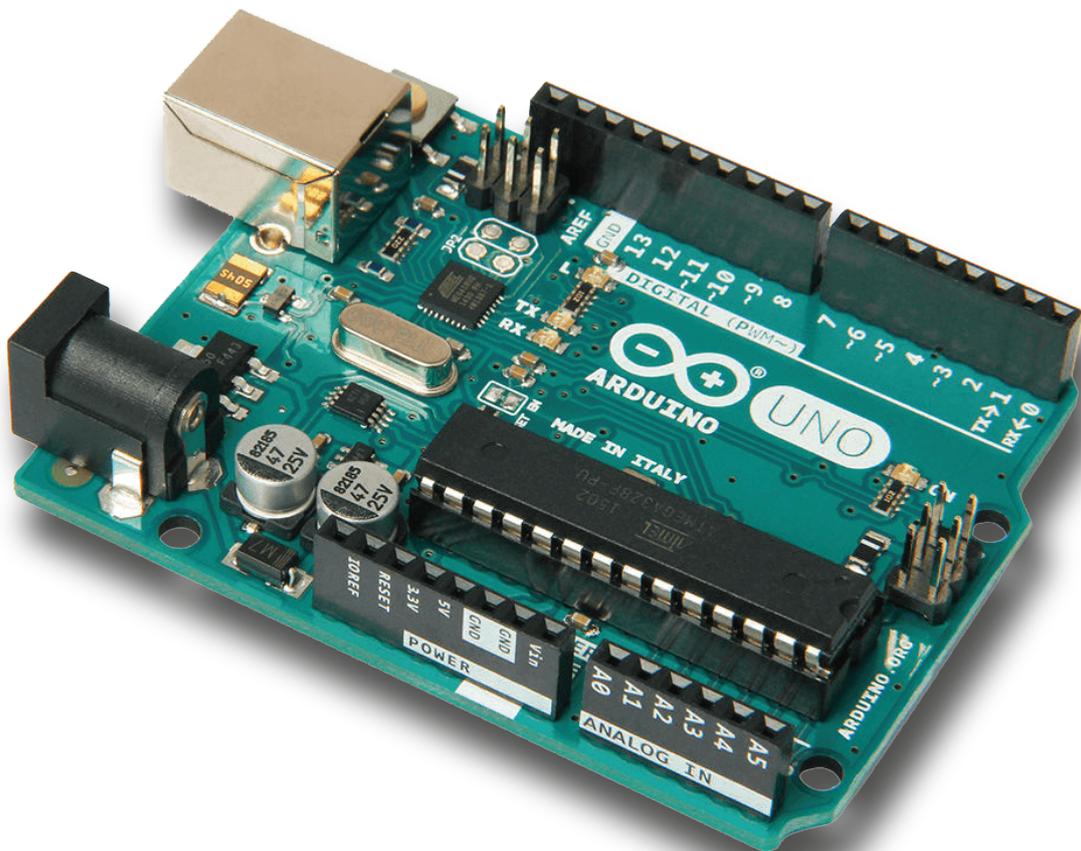
Rana Dajani

ISBN: 2023/9/4930



Arduino

Teachers' Book



Level 1

Rana Dajani

Table of Contents

Lesson 1: Getting Started	8
Lesson 2: Blinking LED	11
Lesson 3: Pattern Waves	13
Lesson 4: Repeat and Resonate	17
Lesson 5: Press On	20
Lesson 6: Key Notes	23
Lesson 7: Stopwatch	26
Lesson 8: Quick Reflex	29
Lesson 9: Counter Shake	32
Lesson 10: Hourglass	37
Lesson 11: Alarm System	42
Lesson 12: Spin Range	46
Lesson 13: LCD Messages	49
Lesson 14: Crystal Luck	52
Lesson 15: Pixel Dodge	56
Lesson 16: Motion Count	62
Lesson 17: Morse Code	64
Lesson 18: Cyclone Spin	66
Lesson 19: Whac-A-Mole	68
Lesson 20: Lightsabers	71

Introduction

Welcome to the teachers' guide for the level one book in our Arduino projects books series, which is designed to teach students how to use the Arduino Uno microcontroller with simple basic components. This guide is intended to provide you with the necessary resources to support your students as they learn to program and build electronic devices using the Arduino system.

The Arduino system is an incredibly versatile tool that can inspire creativity and problem-solving skills in students of all ages. With its wide range of input and output devices, sensors, indicators, displays, motors, and more. Students can learn to program the exact interactions required to create a functional device. Whether they want to build a robot, control the lights in their room, or create an interactive art installation, the Arduino system can help them turn their ideas into reality.

As a teacher, you play a critical role in helping your students develop the skills and knowledge they need to succeed in the 21st century. By teaching them to use the Arduino system, you can help them build a foundation in electronics, coding, and problem-solving that will serve them well in their future careers and personal projects.

This level one book in our series covers the basics of electronics and programming and introduces the components of the Arduino board. It guides students through a series of 20 projects that will help them build their skills with confidence. The last 5 projects are arcade games themed comprehensive projects, unguided and without providing the program code, but with step-by-step pseudocode instructions for students to follow. These projects are designed to allow students to use all the built-up knowledge in full-fledged projects and to assess their understanding.

In this guide, we provide tips and strategies for teaching Arduino to students, as well as resources and activities that you can use to support their learning. We encourage you to adapt the materials to your own teaching style and the needs of your students.

Troubleshooting Tips

"It is not working!" is a common exclamation made when working with electronics. But, have no fear. There are numerous troubleshooting tips and possible solutions to problems that frequently come up that make diagnosing problems much simpler than it may seem.



Hardware Checks

Check Your Connections

It is best to double-check wired connections just in case a wire was connected incorrectly or is loose. Certain components have a certain polarity and will only function when it is connected in the correct position. If everything appears to be wired correctly, you might want to try going back to the beginning and looking at the basic hookup. Also, disconnecting the parts and rewiring each circuit can help troubleshoot. This will assist in narrowing down the problem.

- Is there continuity between the pins, and is the connection secure?
- Is the wire or cable bad?
- Was the component connected using the correct polarity?
- Did you use the basic hookup?
- Is there a problem with the USB Hub or USB port that you are using?

Check Your Logic Levels

- Can you connect a 3.3V sensor to a 5V board?

Check Your Power Supply

- Is your power supply sufficient?



Software Checks

Board Selection

- Did you select the correct board definition?

COM Port Selection

- Did you select the correct COM port?
- Remove any connections to the 0(RX) and 1(TX) pins when uploading code.
- COM port number changes when connecting board on different USB ports.

Installed Arduino Libraries

- Does your example code require libraries to be installed?



Program Checks

If you get a compilation error when trying to compile or upload code to your board, check for errors in syntax, typos and more. When compilation fails, the IDE (Integrated Development Environment) will present you with the errors on its bottom part. However, the error messages generated by the Arduino IDE are limited in their description and therefore not always very helpful, in which case it is a good idea to use Google to search for solutions to the problem.

Best Practices

- Write code in small chunks and test each of them.
- Provide meaningful names for variables and functions.
- Write comments to explain coding choices for future reference.
- Make sure your code has a proper indentation and remains readable at all times. you can use Alt-t to auto-indent the whole sketch.

Common syntax errors are:

- missing semicolon ; at the end of each line
- misplaced or missing parenthesis { }
- typo of misspelled words or case sensitivity
- undefined variables and functions

Serial Monitor

Use the serial monitor to print an overview of the current state of the program. As such:

- variables
- inputs – sensor readings
- prints that indicate the programs flow, like inside an 'if' statement to see whether the condition was met
- outputs – e.g. PWM values before writing them to the pin
- anything else you find important to print to screen

Check, Test and Recheck



Online Research for Resources

Try using an online search engine, chances are, there is someone in the world who has done the project (or something similar) already, and they have provided some documentation online.

Lesson 1: Getting Started

Overview

In this lesson, students will be introduced to the basic components of the Arduino Uno microcontroller board and basic circuit components. They will learn how to identify and use these components to build simple circuits and write basic programs. By the end of this lesson, students will have a fundamental understanding of how the Arduino board works and the role of basic components in building circuits.

Vocabulary

- **Microcontroller:** a small computer on a single integrated circuit that can be programmed to control various devices.
- **Arduino:** an open-source electronics platform based on easy-to-use hardware and software.
- **IDE (Integrated Development Environment):** a software application used to write, compile, and run code.
- **Sketch:** a program written in the Arduino IDE.
- **Serial Port:** a type of communication interface used to transfer data between a computer and a device.
- **USB Driver:** software that enables communication between a computer and a USB device.
- **Tinkercad:** an online open-source platform that allows users to design and simulate 3D objects and circuits.
- **Circuit:** a closed loop through which electricity flows.
- **Breadboard:** a platform used for prototyping electronic circuits.
- **LED (Light Emitting Diode):** a device that emits light when a current passes through it.
- **Anode:** the positive (+) leg of an LED (light-emitting diode).
- **Cathode:** the negative (-) leg of an LED
- **Resistor:** a component that limits the flow of current in a circuit.
- **Ground:** the reference point in an electrical circuit.
- **Polarity:** the property of having a positive or negative electrical charge.

Before the Lesson

- It is important to ensure that each student has the necessary tools and components. This includes an Arduino Uno board, a breadboard, jump wires, a battery snap, a USB cable, LEDs, and resistors.



The Lesson Plan

Introduction to the Microcontroller Board

The first part of the lesson introduces students to the Arduino Uno microcontroller board. Students will learn about the ATmega microcontroller, which is the “brain” of the board, and the different pins and components on the board. This includes analog input pins, digital pins, ground and 5V pins, the reset button, USB port, and power connector. Students will also learn about the different LEDs on the board and their functions.

Basic Circuit Components

The second part of the lesson focuses on basic circuit components, including the breadboard, jump wires, battery snap, USB cable, LEDs, and resistors. Students will learn how to use the breadboard to build circuits and connect different components to each other. They will also learn about the function of each component, such as how resistors resist the flow of electrical energy in a circuit.

Throughout the lesson, students will be engage in hands-on activities, building simple circuits and writing basic programs to control the LEDs. By the end of the lesson, they will have gained practical experience in using the Arduino board and basic circuit components.

It is important to make sure that students have a basic understanding of electricity and how it works. It might be helpful to review some basic concepts of electricity and how electricity flows through conductive materials from the positive point of higher potential energy to the negative point where potential energy is lowest in the circuit.

It is important to emphasize the relationship between voltage, current, and resistance, by introducing the concept of Ohm’s Law. Students should also understand that the value of resistance can be affected by the type of material and the length of the wire used in the circuit.

Software Setup

In this part of the lesson, students will learn how to set up their programming environment using the Arduino IDE software. The Arduino IDE is an essential tool for programming the microcontroller on the Arduino board. By the end of this lesson, students will have the ability to write programs, also known as “sketches,” and upload them to the board.

Explain to students that they may encounter issues when setting up their programming environment. Encourage them to troubleshoot by checking that the correct board and serial port are selected, that the USB cable is properly connected, and that the USB driver is installed correctly.

Another programming environment students can also use is Tinkercad, which provides a simulated Arduino Board that allows students to visualize what a program they write in the “Code Editor” does in the Workplane.

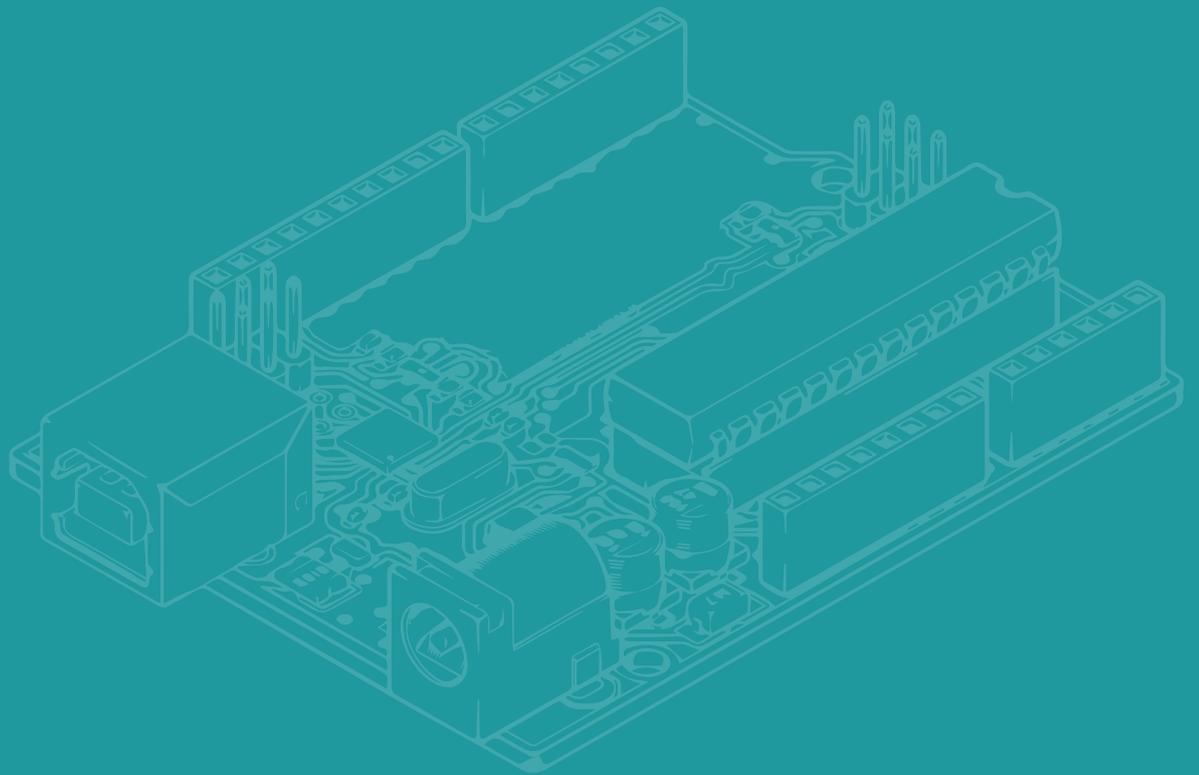
One of the benefits of using Tinkercad is that students can make mistakes in their code without damaging any physical components. While writing an Arduino sketch, students may make a mistake and the program may not compile (save) and upload. This happens to professional programmers all the time! The error message provided by Tinkercad can help students understand what went wrong and how to fix it.

First Circuit - Let there be Light!

In this section, students will learn how to light up an LED in three different ways using a 9V power source, breadboard, and Arduino board. They will also learn how to connect an LED by its “legs” to the positive and negative terminals and the importance of a 220Ω resistor in regulating voltage flow through the LED and protecting it from damage.

Challenge Assessment:

To assess student understanding, have students draw a simple circuit diagram with the components they have learned about. Ask them to label the components and explain how they work together to create a functional circuit.



ISBN: 2023/9/4930

ARDUINO