Exploring The World of Arduino

Date : 07/12/24

Venue : Lab No. 304,3rd Floor

Time : 2:30 pm to 5:30 pm

Organized By Computer Science & Electronics Department

Introduction to Arduino Programming (Workshop)

Subodh S. Kamble

Scan QR code To Get:

1. Code Files
2. Workshop PPT
3. More interesting Projects with code

What is microcontroller?

A microcontroller is a compact, integrated circuit (IC) designed to perform specific control functions within an embedded system.

Used for controlling specific processes, such as in household appliances, cars, medical devices, and industrial automation. Popular microcontrollers include Arduino, ESP32 etc.

Need for microcontroller?

1. Compact Computing Devices
2. Task-Specific Operation
3. Low Power Consumption
4. Widely Used in Automation
5. Cost-Effective and Versatile

Why to Learn Microcontrollers?

1. Hands-On Learning of Embedded Systems
2. Cost-Effective Prototyping
3. Real-World Problem Solving
4. Highly Customizable Applications
5. Build Career-Ready Skills

Table of Context

Why to learn Arduino ?

Interactive Learning: Arduino lets students create cool projects like robots, light sensors, and weather stations. It turns theory into real, interactive devices.

Easy to Use: The hardware and software are beginner-friendly, with a lot of online resources and tutorials to guide them.

Critical Thinking: It helps develop problem-solving skills by experimenting and troubleshooting projects.

Creativity: Students can bring their ideas to life, from controlling lights to making a temperature-controlled fan.

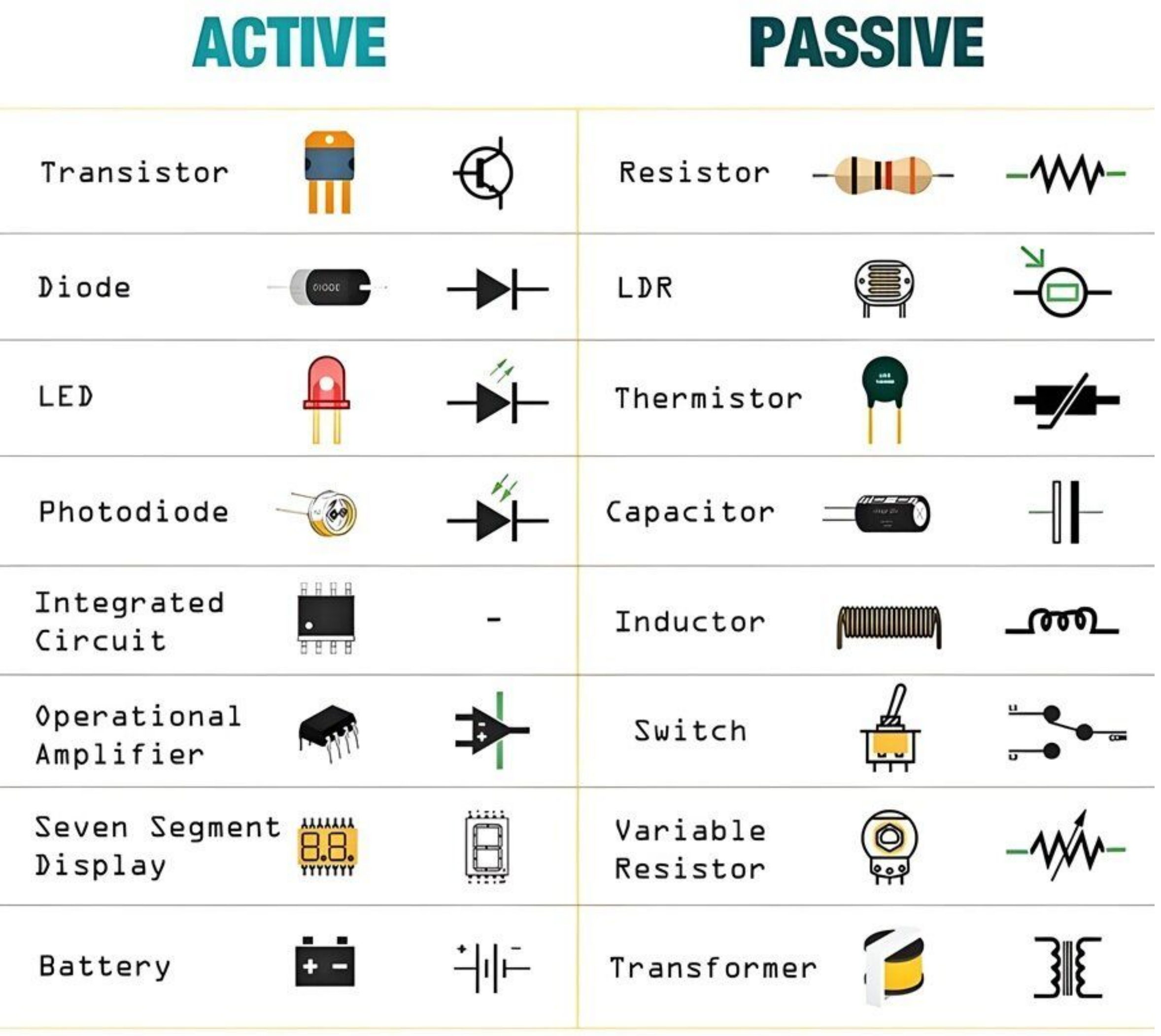
1. Analog Signals

Analog signals are continuous signals that can take any value within a range. For example, a temperature sensor can give you a value that ranges from 0 to 100 degrees Celsius.

2. Digital Signals

Discrete signals that can only take specific values, usually represented as binary (0s and 1s). For example, a button can be either pressed (1) or not pressed (0).

Analog Signals	Digital Signals
Analog signal is continuous and time varying.	Digital signal have two or more states and in binary form.
Troubleshooting of analog signals are difficult.	Troubleshooting of digital signals are easy.
An analog signal is usually in the form of sine wave.	An digital signal is usually in the form of square wave.
Easily affected by the noise.	These are stable and less prone to noise.
Analog signals use continous values to represent the data.	Digital signals use discrete values to represent the data.
Accuracy of the analog signals may be affected by noise.	Accuracy of the digital signals are immune from the noise.
Analog signals may be affected during data transmission.	Digital signals are not affacted during data transmission.
Analog signal use more power.	Digital signal use less power.
Examples: Temperature, Pressure, Flow measurements, etc.	Examples: Valve Feedback, Motor Start, Trip, etc.
Components like resistors, Capacitors, Inductors, Diodes are used in analog circuits.	Components like transistors, logic gates, and microcontrollers are used in Digital circuits.



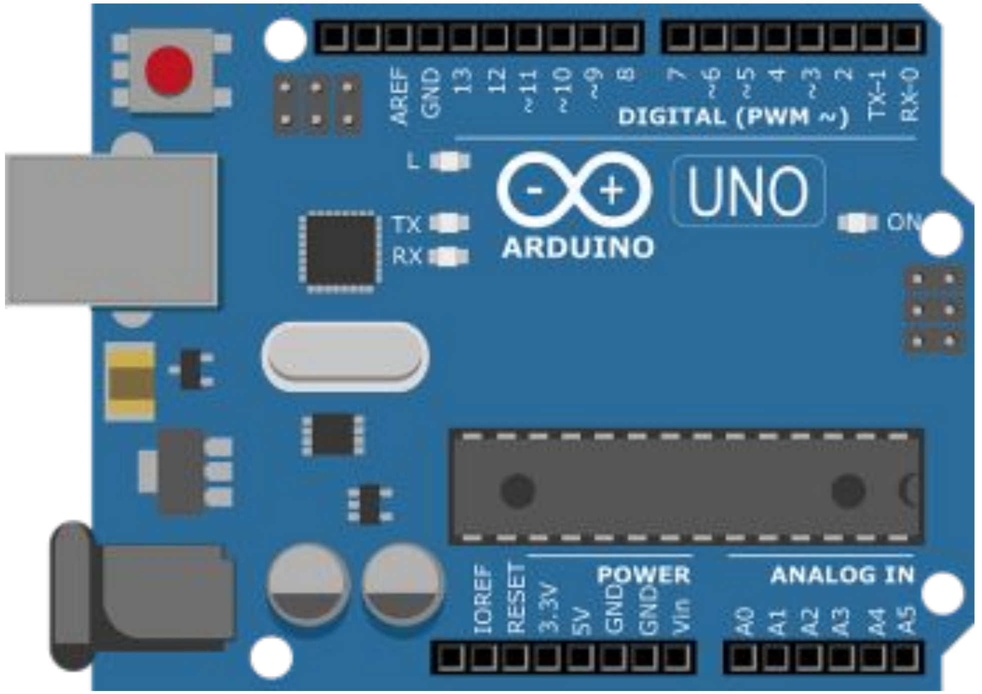
ACTIVE			PASSIVE		
Transistor			Resistor		
Diode			LDR		
LED			Thermistor		
Photodiode			Capacitor		
Integrated Circuit		-	Inductor		
Operational Amplifier			Switch		
Seven Segment Display			Variable Resistor		
Battery			Transformer		

Active Components: Used in signal processing, amplification, and digital circuits (e.g., microcontrollers).

Passive Components: Used in circuit protection, filtering, and energy storage applications.

What is Arduino ?

Arduino is an Italian **open-source hardware** and **software company**, project, and **user community** that designs and manufactures **single-board microcontrollers** and **microcontroller kits** for building digital devices.

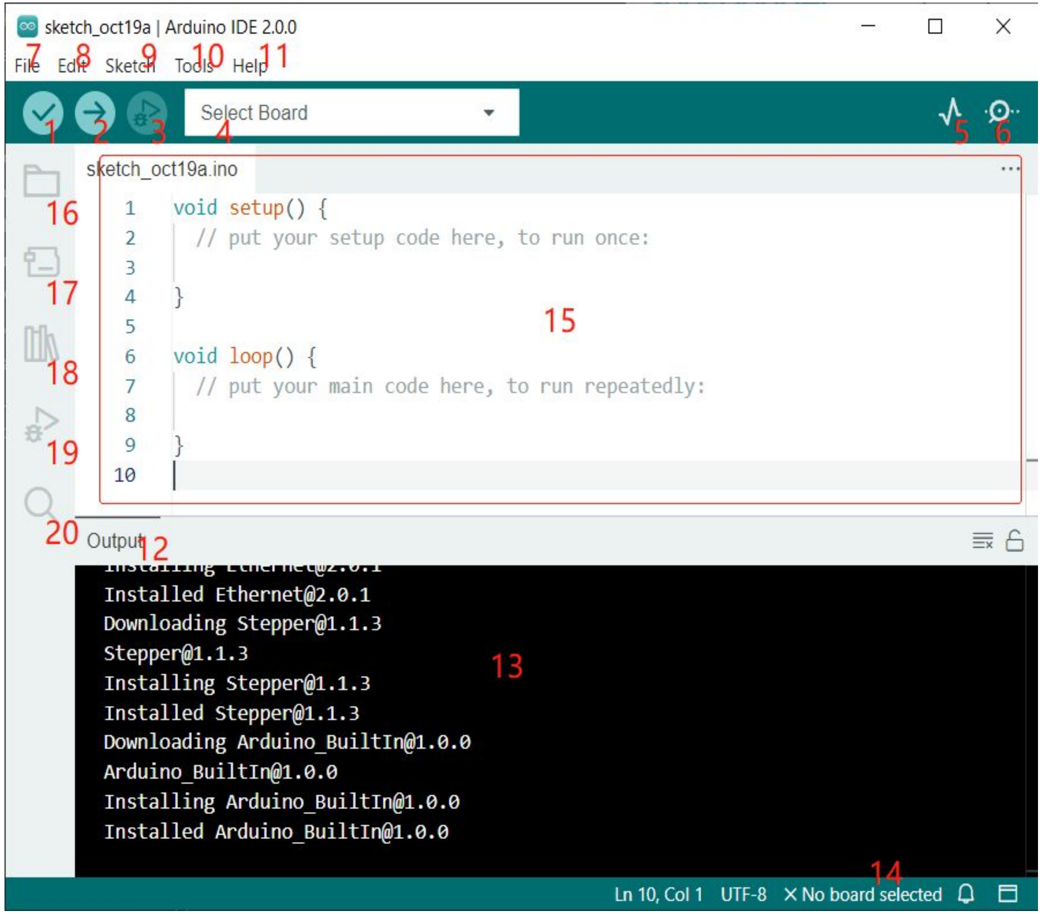


GPIO Pins in Arduino

A standard Arduino Uno board has 20 GPIO pins - 14 digital I/O pins (with 6 of them capable of PWM output) and 6 analog input pins.

Arduino IDE

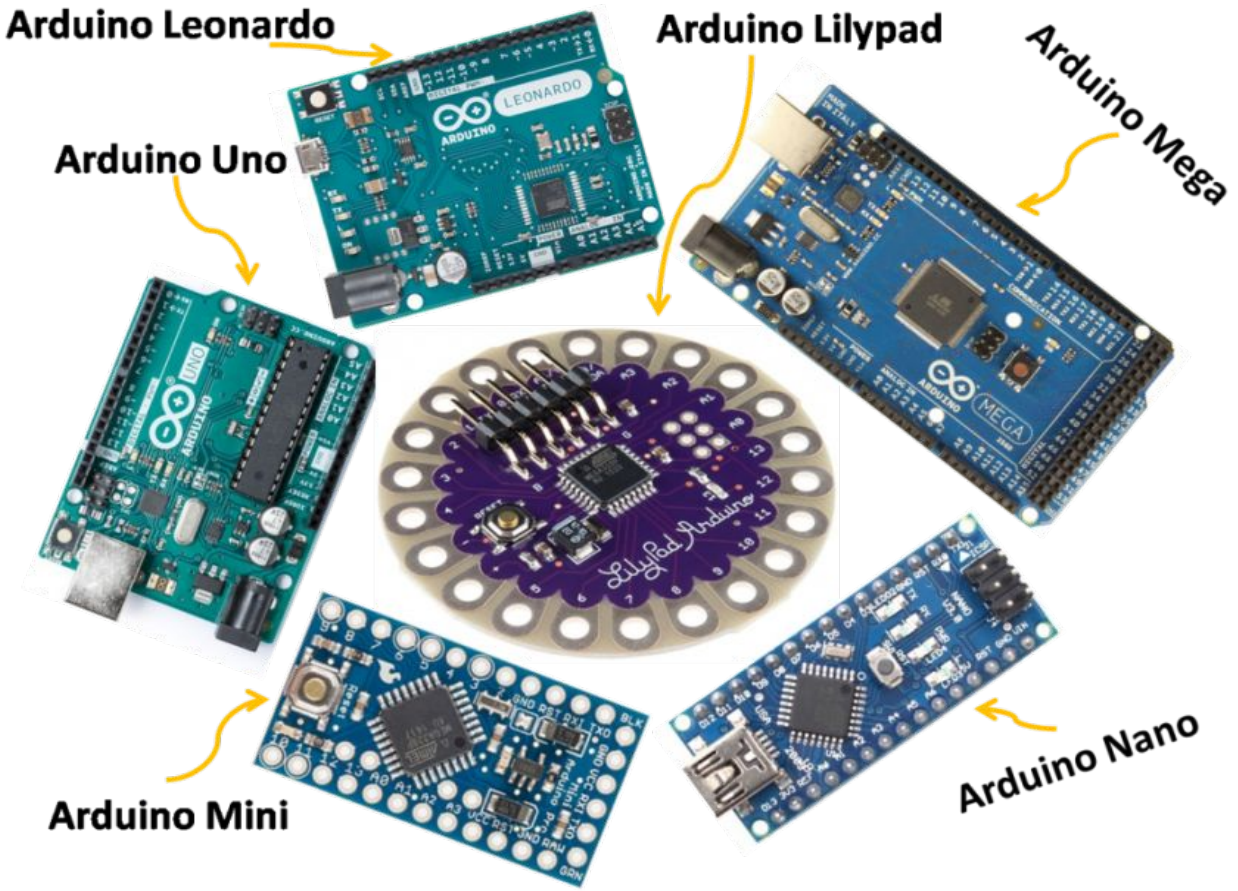
The open-source Arduino Software (IDE) makes it easy to write code and upload it to the board. This software can be used with any Arduino



```
void setup() {
  // put your setup code here, to run once:

}

void loop() {
  // put your main code here, to run repeatedly:

}

```

```
Installed Ethernet@2.0.1
Downloading Stepper@1.1.3
Stepper@1.1.3
Installing Stepper@1.1.3
Installed Stepper@1.1.3
Downloading Arduino_BuiltIn@1.0.0
Arduino_BuiltIn@1.0.0
Installing Arduino_BuiltIn@1.0.0
Installed Arduino_BuiltIn@1.0.0
```

Arduino Board types and I/O pins.



important functions before programming

1. setup() : This function is executed once at the start of the program, typically used to initialize variables, pin modes, or communication protocols like Serial.
2. loop() : This function runs repeatedly after the setup() and contains the main logic of the program.

3. pinMode(pin, mode) : Configures a specified pin to behave either as an input or output.

4. digitalWrite(pin, value) : Writes a HIGH or LOW value to a digital pin, controlling its state.

5. digitalRead(pin) : Reads the value from a specified digital pin (either HIGH or LOW).

6. analogRead(pin) : Reads the analog value from a pin (values between 0 and 1023).

7. analogWrite(pin, value) : Writes an analog value (PWM) to a pin (range: 0 to 255).

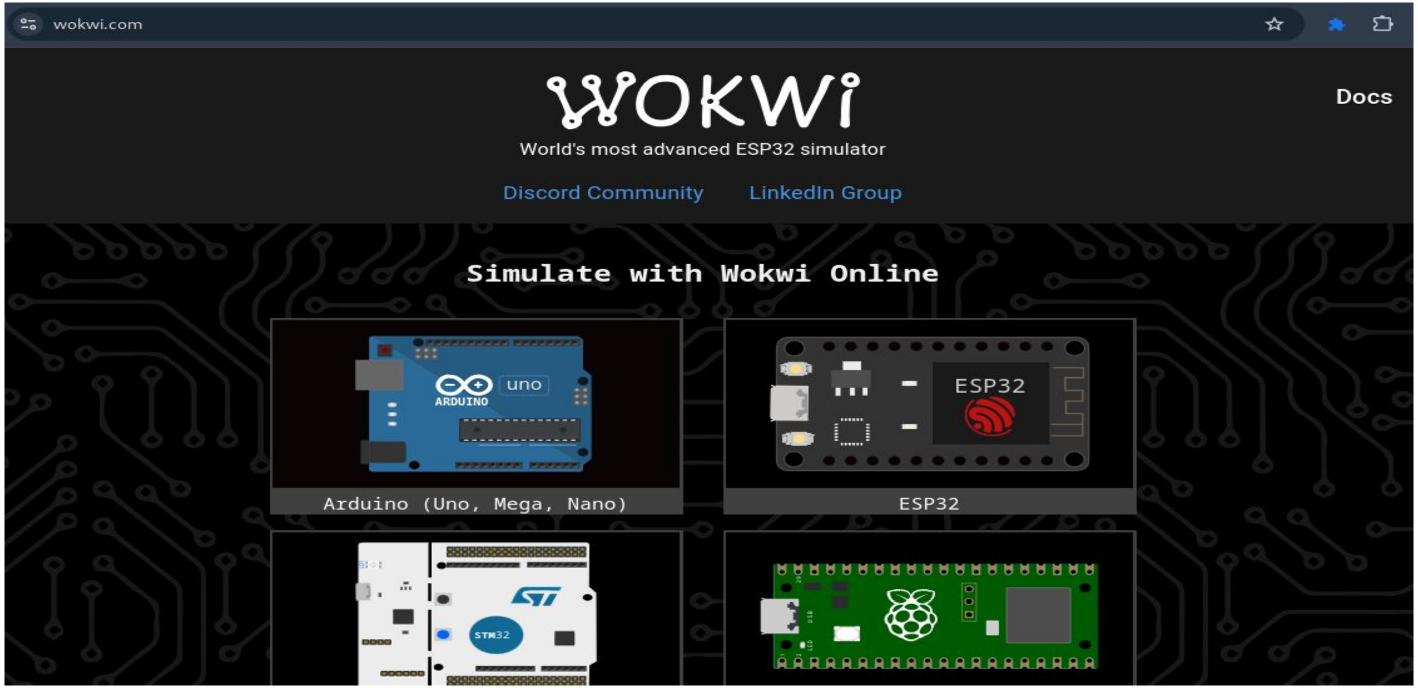
8. delay(ms) : Pauses the program for a specified number of milliseconds.

Simple Blink program Arduino

```
void setup() {
  pinMode(13, OUTPUT);  // Set pin 13 as an output pin
}
void loop() {
  digitalWrite(13, HIGH);  // Turn on the LED
  delay(1000);              // Wait for 1 second
  digitalWrite(13, LOW);   // Turn off the LED
  delay(1000);              // Wait for 1 second
}
```

Wokwi as a simulator

- Wokwi is an online Electronics simulator.
- You can use it to simulate Arduino, ESP32, and many other popular boards, parts, and sensors.


wokwi.com
WOKWI
Docs
World's most advanced ESP32 simulator
Discord Community
LinkedIn Group
Simulate with Wokwi Online
Arduino (Uno, Mega, Nano)
ESP32

Features of Wokwi

1. Start right now
2. No waiting for components or downloading large software.
3. In seconds, your browser has everything you need to start coding your next IoT project.
4. Mistakes are okay
5. You can't destroy the virtual hardware. So don't worry about frying your precious components.
6. Unlike real hardware, you can always undo it.
7. Easy to get help and feedback
8. Sharing a link to your Wokwi project is all you need.

Features of Wokwi

9. Gain confidence in your code.

10. Separate hardware and software issues.

11. Unlimited hardware

12. No need to scavenge parts from old projects.

13. Use as many parts as you need without worrying about project price and stock.

14. Maker-friendly community

15. A place for you to share your projects, ask for help and get inspiration.

Cost of wokwi simulator?

- Wokwi is free to use.
- Professional and frequent users use workwi for prototypeing

Components of wokwi

1. **Interactive Diagram Editor**

The diagram editor provides an interactive way to edit your diagram:

- Add components to the simulation
- Define the connections between them
- It’s a convenient alternative for editing the diagram.json file directly.

Components of wokwi

1. **Adding part**
   a. To add a new part, click on the blue "+" button at the top of the diagram editor.
   b. You'll see a menu with a list of parts you can add.
   c. Choose a part to add it. The part will be added at position (0, 0), and then you can
   d. drag it to the desired position.
   e. Not all parts are currently available through the menu.
   f. and micro-controllers such as the Arduino Nano or the ATtiny85 are missing.
   g. Youcan still add these parts by editing diagram.json directly.

Moving a part

Move a part by clicking on it and then dragging it with your mouse.

Rotating a part

1. Rotate a part by clicking on it (to select it) and then press "R".
2. The part will rotate 90 degrees clockwise. If you need to rotate a part by a different
3. amount (e.g. 45 degrees), you can achieve that by editing diagram.json.

Duplicating a part

Create a new copy of a part by clicking on it (to select it) and pressing "D".

1. You can press "D" several times to create multiple copies of the part.
2. Deleting a part
3. Delete a part by clicking on it (to select it) and then pressing the Delete button.
4. selecting multiple parts
5. Select multiple parts by clicking on the parts with the Shift key pressed. You can
6. then move all the parts together, duplicate them (using the "D" key), or delete them.

Copying and pasting parts

1. You can copy the selected part(s) using the standard Copy keyboard shortcut (Ctrl+C or ⌘+C).
2. If you select multiple parts, all the wires connecting the chosen components are copied.
3. The parts you copied are stored in your system clipboard in a JSON format, similar to the diagram.json format.
4. To paste the parts you copied, click on the diagram and press the standard Paste keyboard (Ctrl+V or ⌘+V).
5. Sometimes, the parts will be pasted outside the currently visible diagram area, so you may have to zoom out to find them. This will be fixed in the future.
6. You can use the copy-paste feature between different project and quickly copy several parts (including all the internal connections) at once.

Editing wires : Creating a wire between two parts

To create a new wire between two parts, click on one of the pins that you'd like to connect.

1. If you want the wire to go in a specific way, you can guide it by clicking where you
2. want it to go after selecting the first pin.
3. To cancel a new wire (delete it without selecting a target pin) click the right mouse
4. button or press Escape.
5. Changing the color of a wire

Editing wires : Editing wires Changing the color of a wire

The function of the pin automatically determines the color of new wires:

1. Wires starting from ground pins are black, 5 V pins are red, and other wires are green.
2. You can change the color of a wire by clicking on it and then selecting a new color for the wire.
3. You can also use the following keyboard shortcuts to set wire colors: [0-Black; 1- Brown; 2-Red; 3-Orange; 4- Gold; 5- Green; 6-Blue; 7-Violet; 8- Gray; 9-White; C-Cyan; L- Lime green; M-Magenta; P-Purple; Y-Yellow]

Deleting a wire

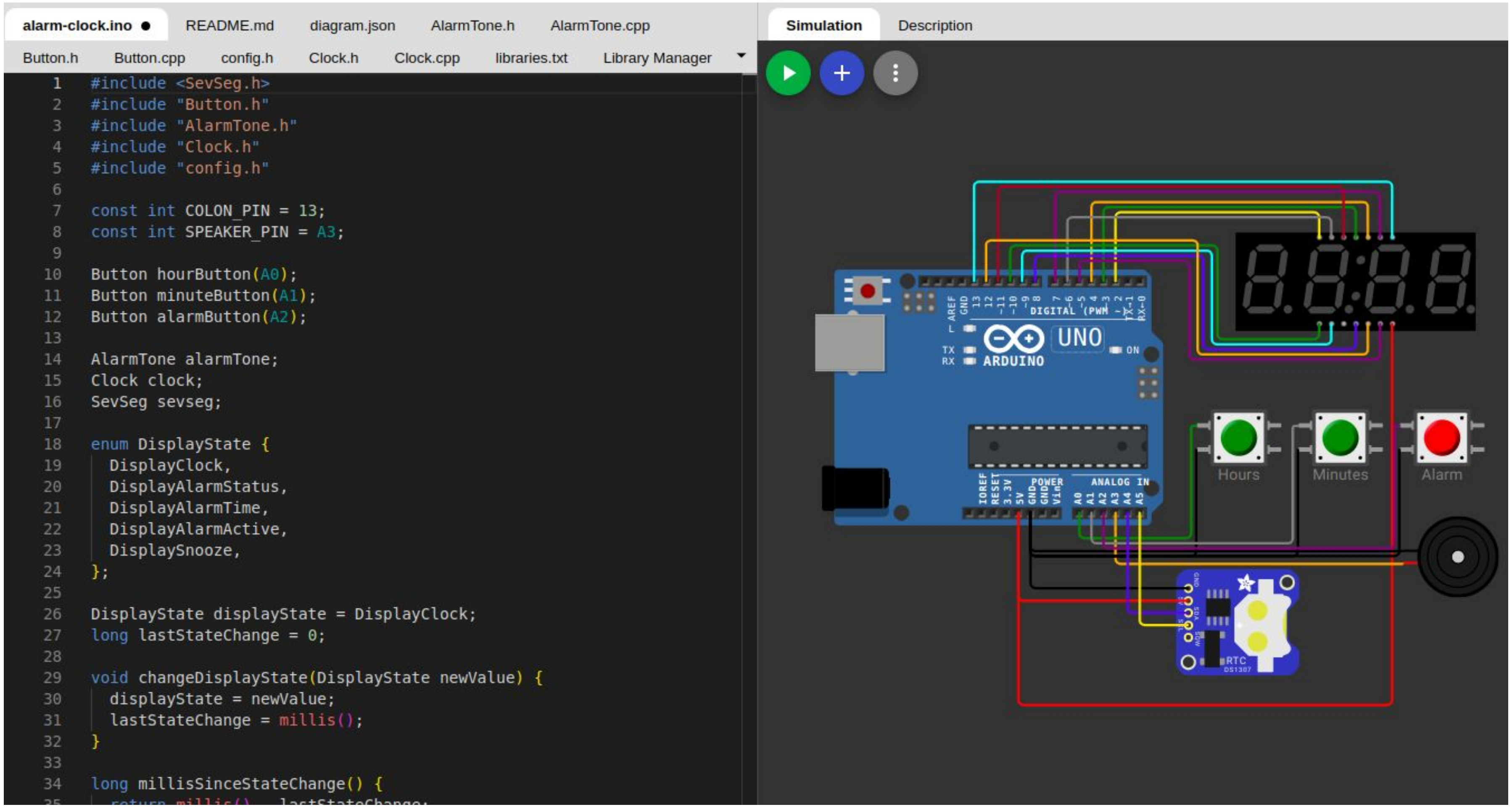
1. Select a wire by clicking on it, then click the trash icon on it (or press the Delete key).
2. You can also delete a wire by double-clicking on it.

Keyboard shortcuts

- The following table summarizes the keyboard shortcuts

Key	**Control**
-	Zoom out
+	Zoom in
F	Fit diagram to window (auto zoom)
D	Duplicate (copy) the selected part
R	Rotate the selected part
Delete	Delete the selected part / wire
?	Open documentation for the selected part

Introduction to wokwi IDE (sample)



```cpp
#include <SevSeg.h>
#include "Button.h"
#include "AlarmTone.h"
#include "Clock.h"
#include "config.h"

const int COLON_PIN = 13;
const int SPEAKER_PIN = A3;

Button hourButton(A0);
Button minuteButton(A1);
Button alarmButton(A2);

AlarmTone alarmTone;
Clock clock;
SevSeg sevseg;

enum DisplayState {
  DisplayClock,
  DisplayAlarmStatus,
  DisplayAlarmTime,
  DisplayAlarmActive,
  DisplaySnooze,
};

DisplayState displayState = DisplayClock;
long lastStateChange = 0;

void changeDisplayState(DisplayState newValue) {
  displayState = newValue;
  lastStateChange = millis();
}

long millisSinceStateChange() {
  return millis() - lastStateChange;
```

Quiz On Arduino

Feedback Form