

PROJECT: HOME AUTOMATION

Chapter Notes

FIRST TEN CORE CONCEPTS



1 HOME AUTOMATION

- Home automation means using electronic circuits, sensors, microcontrollers and programmes to make devices operate **automatically** with little or no continuous human control.

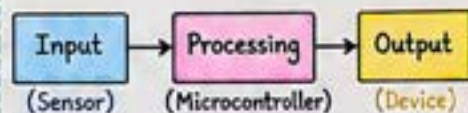
Example:

Automatic light that turns ON when it becomes dark.



2 AUTOMATION CYCLE (INPUT → PROCESSING → OUTPUT)

- An automation system works in three stages:
- Input (sensing) → Processing (decision making) → Output (action).
- A sensor gives input to the microcontroller, it processes the information and controls an output device.



3 SENSOR

- A sensor is a device that detects a physical change in the surroundings and converts it into an electrical signal.
- Sensors help the system to **receive** input from the environment.

Example:

- Motion sensor
- LDR (light sensor)
- Temperature sensor
- Sound sensor



4 MICROCONTROLLER (ARDUINO UNO)

- A microcontroller is a small programmable device that receives input, processes it according to a programme and controls output.
- Arduino Uno is a popular and easy-to-use microcontroller board.

Example: Arduino Uno



5 PROGRAMME

- A programme is a set of instructions given to the microcontroller to tell it what to do.
- Arduino programmes are written in Arduino IDE using C/C++.

Example:

```

void setup() {
  pinMode(13, OUTPUT);
}
void loop() {
  digitalWrite(13, HIGH);
  delay(1000);
}
  
```

6 OUTPUT DEVICE

- An output device performs an action after receiving instructions from the microcontroller.
- Examples:** LED, Buzzer, DC Motor, Servo Motor, LCD, Relay, Speaker.

Examples:



7 ELECTRONIC CIRCUIT

- An electronic circuit is a connection of electronic components through which electrical current flows to perform a task.
- A circuit diagram shows the components and their connections.

Example: LED Circuit



8 BREADBOARD

- A breadboard is used to connect and test electronic components without soldering.
- It is useful for prototyping and experimenting with circuits.

Example:



9 SIMULATION (TINKERCAD)

- Simulation means creating a virtual version of a circuit on a computer.
- Tinkercad** is an online platform used to build and test electronic circuits before making the physical circuit.

Example: Tinkercad Circuit



10 SAFETY

- Safety is important while working with electronic circuits.
- Avoid short circuits, check proper connections, and do not directly connect beginner circuits to 230 V AC mains supply.
- Use DC power (battery or adapter) and handle tools carefully.

Safety Tips:



- Check connections before power ON
- Avoid loose wires and short circuits
- Use correct voltage (DC)
- Handle tools (soldering iron, cutters) safely.

PROJECT: HOME AUTOMATION

Chapter Notes

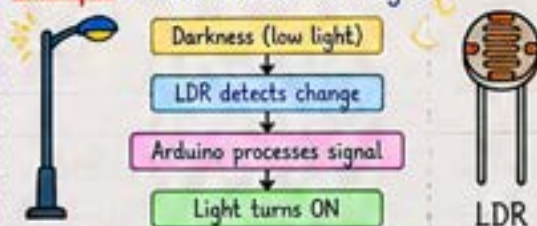
NEXT SIX CORE CONCEPTS (11 - 16)



11 LDR (LIGHT DEPENDENT RESISTOR)

- LDR is a light sensor whose resistance changes according to the amount of light falling on it.
- In bright light: resistance is low.
- In darkness: resistance is high.
- It is used to detect **light intensity** in automation projects.

Example: Automatic Street Light



12 BUZZER

- A buzzer is an output device that produces **sound** when electrical energy is supplied.
- It is used for alarms, notifications and warning systems.
- Types: **Active buzzer** (built-in oscillator) and **Passive buzzer** (requires signal).
- It can be controlled using digital output pins of Arduino.

Example: Security Alarm



13 LCD (LIQUID CRYSTAL DISPLAY)

- An LCD is an output device used to **display messages, values and sensor readings**.
- A 16 x 2 LCD can display 16 characters per row and 2 rows.
- A potentiometer is often used to adjust the contrast (brightness).
- It is useful in many home automation projects.

Example: Displaying Temperature



14 MOTOR (DC, SERVO, STEPPER)

- A motor converts electrical energy into mechanical movement (rotation).
- Types:
 - DC Motor** - provides continuous rotation. (used in robot cars, fans, pumps)
 - Servo Motor** - provides controlled angular movement. (used in gates, robotic arms)
 - Stepper Motor** - rotates in fixed steps. (used where precise control is required)

Example: Types of Motors



15 RELAY

- A relay is an electrically controlled switch.
- It allows a low-power control circuit (like Arduino) to control a higher-power electrical device.
- It is used to switch appliances such as lights, fans and pumps.

Example: Appliance Control using Relay



16 POWER SUPPLY (BATTERY / ADAPTER)

- A power supply provides electrical energy to electronic circuits.
- Battery provides DC power and is portable.
- An adapter converts high-voltage AC from wall socket to low-voltage DC (e.g. 9 V or 12 V) suitable for Arduino.
- Choose the correct voltage and current rating for your project.

Examples:



PROJECT: HOME AUTOMATION

Chapter Notes

MORE CORE CONCEPTS (17 - 23)



17 DIODE

- A diode is an electronic component that allows current to flow mainly in one direction.
- It is used for controlling current direction and protecting circuits.
- Symbol:

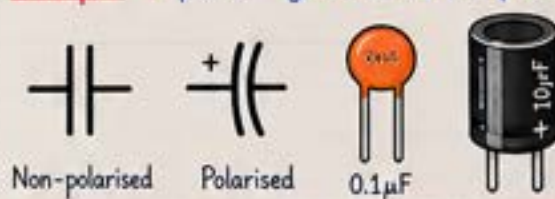
Example: Diode symbol and component



18 CAPACITOR

- A capacitor is an electronic component that temporarily stores electrical energy.
- It is used in filtering and timing applications in electronic circuits.
- Unit: Farad (F) (commonly μF , nF, pF)

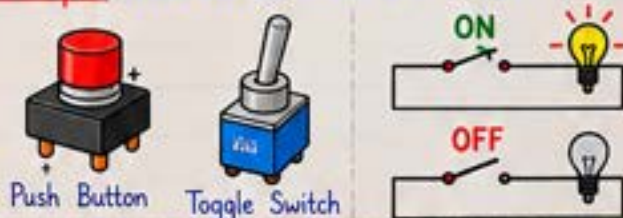
Example: Capacitor symbols and component



19 SWITCH

- A switch is used to manually open or close an electrical circuit.
- It controls the flow of current.
- Types:** Push button switch, Toggle switch, Slide switch, etc.

Example: Push button switch and circuit



20 GROUND (GND)

- Ground is a common reference point in an electronic circuit.
- It provides a return path for current.
- All components in a circuit usually share the same ground.

Example: Ground symbol and Arduino connection



21 JUMPER WIRE

- A jumper wire is an insulated wire used to connect components on a breadboard.
- It helps in making temporary circuit connections without soldering.

Example: Jumper wires and breadboard



22 PCB (PRINTED CIRCUIT BOARD)

- A PCB is a board with conductive copper tracks used to connect electronic components.
- It is more durable than a breadboard and is used for making permanent circuits.

Example: Printed Circuit Board (PCB)



23 SOLDERING

- Soldering is the process of joining electronic components using solder.
- It is commonly used when making permanent circuits on a PCB.
- Tools:** Soldering iron, solder wire, flux, etc.

Example: Soldering iron, solder wire and soldered joint



Safety Tip: Do not touch the hot soldering iron.



PROJECT: HOME AUTOMATION

Chapter Notes

MORE CORE CONCEPTS (24 - 30)



24 DIGITAL MULTIMETER (DMM)

- A Digital Multimeter (DMM) is an electronic measuring instrument used to measure voltage, current and resistance.
- It is useful for testing and troubleshooting circuits.

Example: Digital Multimeter (DMM)



Measures:

- Voltage (V)
- Current (A)
- Resistance (Ω)

25 VOLTAGE

- Voltage is the electrical potential difference that drives current through a circuit.
- Unit:** Volt (V)
- Common electronic values: 5 V, 9 V, 12 V.

Example: Voltage measurement in a circuit

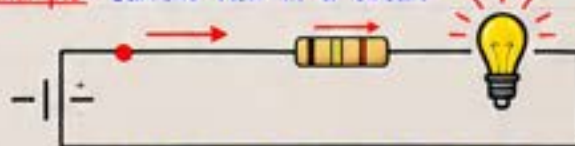


Voltage supplies electrical energy to the circuit.

26 CURRENT

- Current is the flow of electric charge through a circuit.
- Unit:** Ampere (A)
- In electronic circuits, current is usually small (e.g. in milliampere, mA).

Example: Current flow in a circuit



Current flows in the circuit.

27 RESISTANCE

- Resistance is the opposition to the flow of electric current.
- Unit:** Ohm (Ω)
- A resistor is used to provide resistance and limit current.

Example: Resistor and its value



220 Ω

Resistor limits the amount of current in a circuit.

28 SHORT CIRCUIT

- A short circuit occurs when an unintended low-resistance path is created in a circuit.
- It can cause high current and damage components.
- Always check connections to avoid short circuits.

Example: Short Circuit



Very low resistance path (short circuit)



Can cause high current, heat and damage.

29 AC (ALTERNATING CURRENT)

- In AC, the direction of current changes periodically.
- Household electrical supply is AC.
- Common household supply in India: 230 V AC.

Example: AC waveform

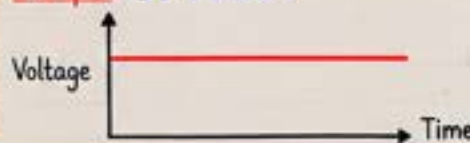


Commonly used for household supply:
230 V AC (in India)

30 DC (DIRECT CURRENT)

- In DC, current flows mainly in one direction.
- Batteries provide DC power.
- Electronic circuits such as Arduino commonly use low-voltage DC (e.g. 5 V, 9 V, 12 V).

Example: DC waveform



AC vs DC (Comparison)

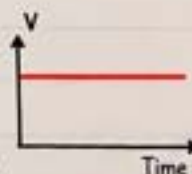
AC (Alternating Current)

- Direction of current changes periodically.
- Waveform is sinusoidal.
- Used in household electrical supply.
- Common supply in India: 230 V AC.



DC (Direct Current)

- Current flows in one direction.
- Waveform is constant (straight line).
- Provided by batteries, adapters, etc.
- Common electronic values: 5V, 9V, 12V.



PROJECT: HOME AUTOMATION

Chapter Notes

MORE CORE CONCEPTS (31 - 37)



31 ADAPTER

- An adapter converts high-voltage AC from a wall socket into low-voltage DC suitable for electronic devices.
- Common Arduino adapters may provide 9 V or 12 V DC.

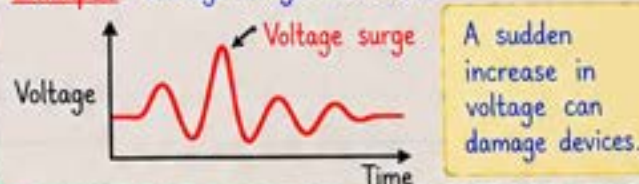
Example: AC to DC Adapter



32 VOLTAGE SURGE

- A voltage surge is a sudden increase in voltage.
- It can damage electrical/electronic devices.
- Protection systems can detect abnormal voltage and disconnect equipment.

Example: Voltage surge waveform



33 ELECTRICAL SAFETY

- Check connections before switching power ON.
- Avoid short circuits and loose connections.
- Use correct voltage.
- Do not directly connect beginner electronic circuits to 230 V AC mains.
- Prefer suitable low-voltage DC supplies for learning circuits.



Important:

- Use correct voltage.
- Check connections.
- Avoid short circuits.
- Do not touch live wires.
- Use low-voltage DC for learning circuits.

34 INTERNET SAFETY

- Use trusted websites for circuit tutorials.
- Do not download unsafe files.
- Do not share personal information online.
- Take teacher/parent guidance when required.

Stay Safe Online:



- Use reliable sources.
- Be careful while downloading files.
- Do not share personal information.
- Take guidance from your teacher or parents.

35 PROTOTYPING

- Prototyping means making an initial working version of a circuit for testing.
- Breadboards are useful for prototyping.
- Components can be easily rearranged before making a permanent circuit.

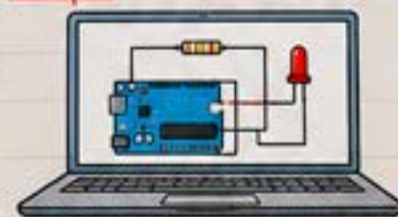
Example: Prototyping process



36 SIMULATION

- Simulation means creating and testing a virtual version of a circuit on a computer.
- It allows students to test circuits before building them physically.

Example: Virtual circuit simulation

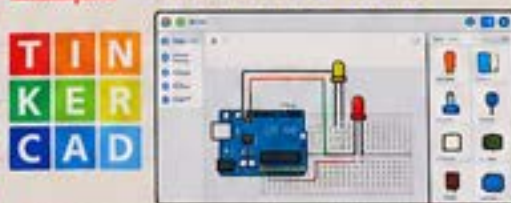


Helps to check connections and programme before building the real circuit.

37 TINKERCAD

- Tinkercad is an online platform used to build and simulate electronic circuits.
- It is useful for learning circuit design and Arduino programming.

Example: Tinkercad circuit interface



Useful for learning and practising circuit design and coding.

PROJECT: HOME AUTOMATION

Chapter Notes

DETAILED NOTES : SIMULATION (Traffic Light)



36 SIMULATION

What is Simulation?

- Simulation means creating and testing a virtual version of a circuit on a computer.
- It allows us to check whether the circuit works correctly before building it physically.
- We can test different components, connections and code without using real hardware.
- It helps save time, cost and components.
- If there is a mistake, we can easily correct it and test again.
- Simulation is very useful for learning electronics and Arduino programming.

Advantages of Simulation

- No physical components required.
- Safe and easy to learn.
- Quickly test and modify the circuit.
- Helps in understanding concepts.
- Useful for projects and experiments.



Simulation Process

Design Circuit
(virtual)



Write Code
and Upload
(simulate)

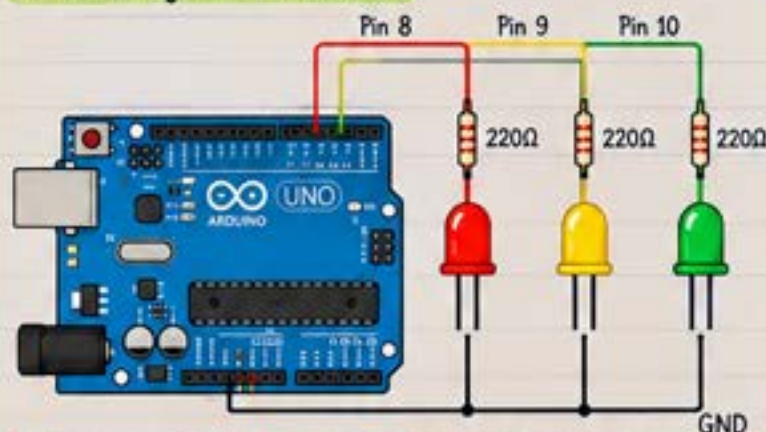


Observe Output
(Check if it
works)



Example : Traffic Light Simulation (LED)

Circuit Diagram in Tinkercad



How the Code Works ?

- 1** `digitalWrite(redPin, HIGH);`
 - Red LED turns ON.
 - Yellow and Green LEDs remain OFF.
 - Waits for 5 seconds.
- 2** `digitalWrite(yellowPin, HIGH);`
 - Yellow LED turns ON.
 - Red and Green LEDs remain OFF.
 - Waits for 2 seconds.
- 3** `digitalWrite(greenPin, HIGH);`
 - Green LED turns ON.
 - Red and Yellow LEDs remain OFF.
 - Waits for 5 seconds.

Arduino Code (Sequential Traffic Light)

```
const int redPin = 8; // Red LED -> Pin 8
const int yellowPin = 9; // Yellow LED -> Pin 9
const int greenPin = 10; // Green LED -> Pin 10

void setup() {
  pinMode(redPin, OUTPUT); // set as output
  pinMode(yellowPin, OUTPUT);
  pinMode(greenPin, OUTPUT);
}

void loop() {
  // 1. Red ON, Yellow OFF, Green OFF
  digitalWrite(redPin, HIGH);
  digitalWrite(yellowPin, LOW);
  digitalWrite(greenPin, LOW);
  delay(5000); // wait 5 seconds

  // 2. Yellow ON, Red OFF, Green OFF
  digitalWrite(redPin, LOW);
  digitalWrite(yellowPin, HIGH);
  digitalWrite(greenPin, LOW);
  delay(2000); // wait 2 seconds

  // 3. Green ON, Red OFF, Yellow OFF
  digitalWrite(redPin, LOW);
  digitalWrite(yellowPin, LOW);
  digitalWrite(greenPin, HIGH);
  delay(5000); // wait 5 seconds
}
```

Output Sequence (Timing)

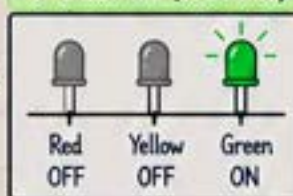
1. Red ON (5 seconds)



2. Yellow ON (2 seconds)



3. Green ON (5 seconds)



The sequence repeats again and again (Red → Yellow → Green).



37

TINKERCAD

- Tinkercad is a free, online platform by Autodesk used to design and simulate electronic circuits.
- It provides a simple and easy-to-use interface for beginners.
- We can create circuits using components like Arduino, sensors, LEDs, motors, etc.
- We can also write Arduino code and run the simulation directly in the browser.
- No installation is required, it works online.
- It is very useful for learning circuit design, Arduino programming and testing ideas for home automation projects.

Features of Tinkercad

- Free to use (with Autodesk account).
- Simple and beginner friendly interface.
- Drag and drop electronic components.
- Virtual breadboard and Arduino.
- Write and test Arduino code in built-in editor.
- Observe the output in real time (simulation).
- Useful for learning, experimentation and school projects.

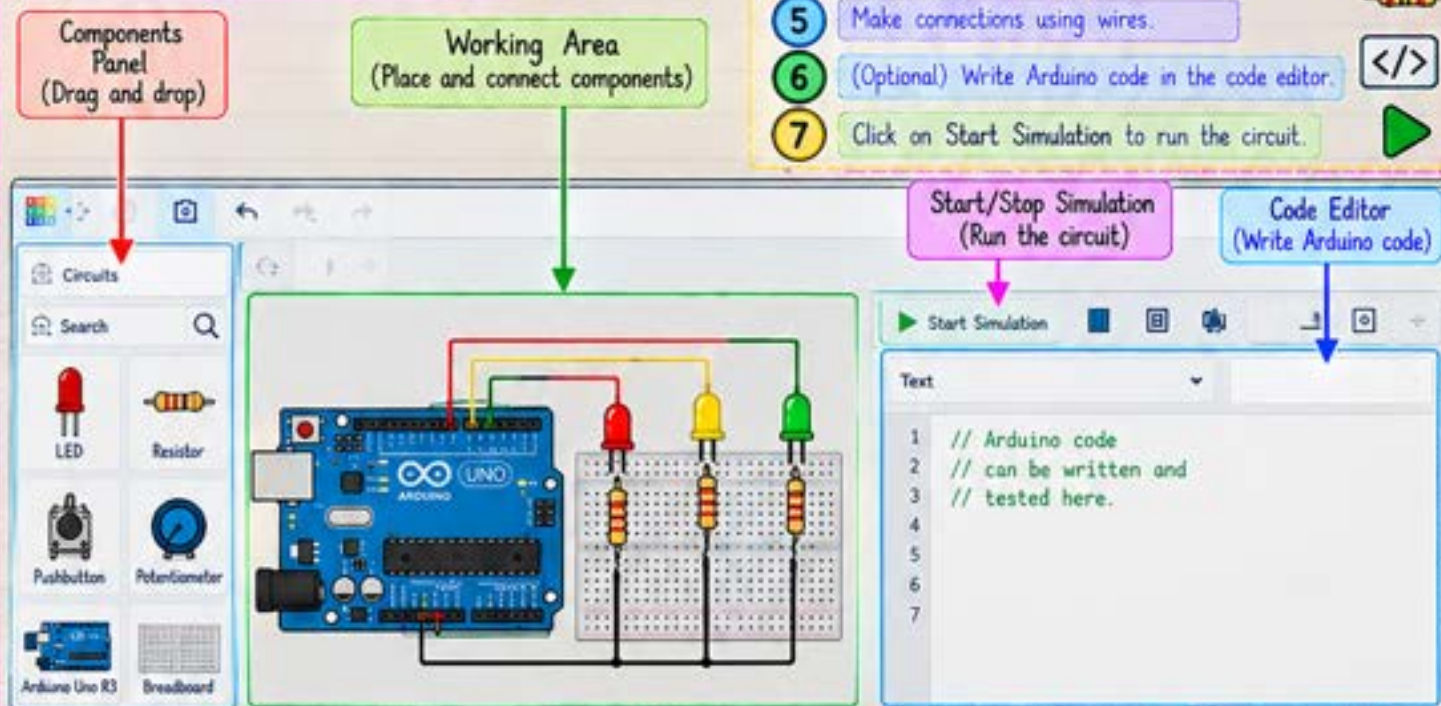


How to Use Tinkercad (Circuit Simulation)

- 1 Open website : <https://www.tinkercad.com>
- 2 Sign in or create a free Autodesk account.
- 3 Click on Circuits → Create new Circuit.
- 4 Drag and drop components from the components panel.
- 5 Make connections using wires.
- 6 (Optional) Write Arduino code in the code editor.
- 7 Click on Start Simulation to run the circuit.



Tinkercad Interface Overview



Components Panel (Common Components)



Uses of Tinkercad

- Design and test electronic circuits virtually.
- Learn how components work together.
- Test Arduino code before building the physical circuit.
- Try different ideas without using real components.
- Useful for home automation, robotics and IoT projects.

Benefits for Students

- Easy to learn and use.
- Helps in understanding circuit design and coding.
- Save time, cost and components.
- Correct mistakes easily and test again.
- Useful for school projects, competitions and learning.



Key Points :

- Tinkercad is a free, online platform to design and simulate circuits.
- We can build circuits, write Arduino code and run simulation in the browser.
- It is very useful for learning electronics and making home automation projects.

PROJECT: HOME AUTOMATION

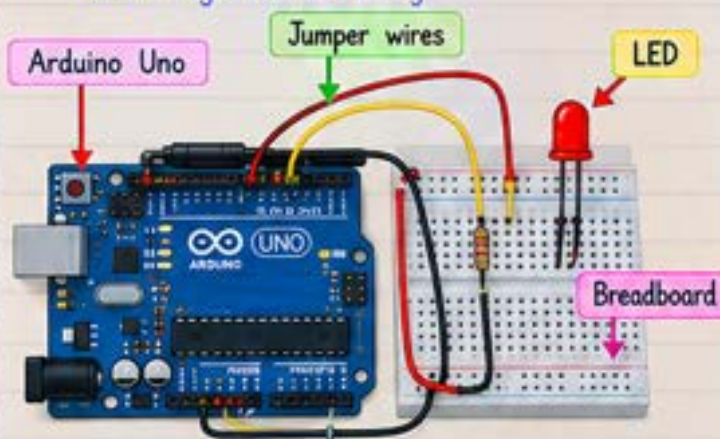
Chapter Notes

MORE CORE CONCEPTS (38-44)



38 PHYSICAL CIRCUIT

- A physical circuit is a real circuit built using actual electronic components.
- It is built and tested using real hardware.
- It helps us to learn how components work together in reality.



39 CIRCUIT DIAGRAM

- A circuit diagram is a graphical representation of an electronic circuit.
- It shows components and their connections using standard symbols.
- It helps us to understand and design circuits before building them.

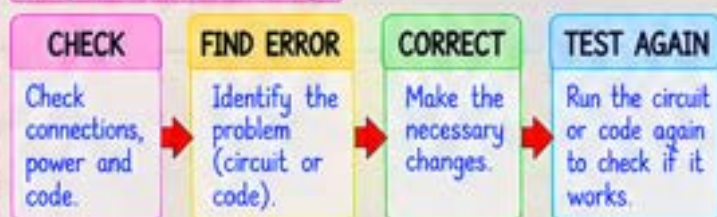
Example : LED Circuit Diagram



40 TROUBLESHOOTING

- Troubleshooting means finding and correcting problems in a circuit or programme.
- Common problems: wrong connections, loose wires, wrong pin numbers, incorrect code and missing power or GND.

Steps in Troubleshooting



43 LITHIUM-ION BATTERY

- A lithium-ion battery is a rechargeable battery used as a portable power source.
- It is lightweight, has high energy density and can be used for suitable electronic and robotics projects.
- Common type: 18650 Li-ion cell (3.7 V).
- It requires a proper charging and protection circuit.



Important :

- Use with proper protection circuit.
- Do not overcharge.
- Handle with care.

41 BATTERY

- A battery is a portable source of electrical energy.
- It provides DC power to electronic circuits.
- Common batteries: 9 V, AA cells etc.



42 DC CONNECTOR CABLE

- A DC connector cable connects a suitable DC power source to an electronic device such as Arduino.
- It is commonly a barrel connector (5.5 mm x 2.1 mm).
- It is used with an adapter to provide DC power.



44 CASING

- A casing protects electronic components.
- It makes a project safer, stronger and easier to handle.
- It can be made using plastic, acrylic, wood, cardboard or 3D printing.



Electronic Circuit inside a Casing

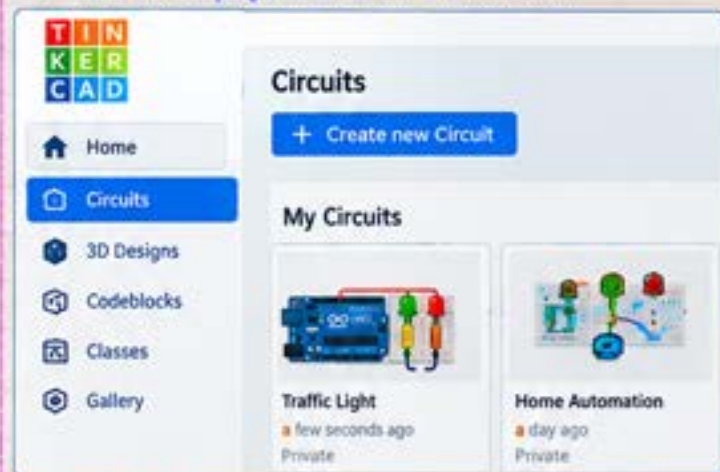


Closed Casing (Project Box)



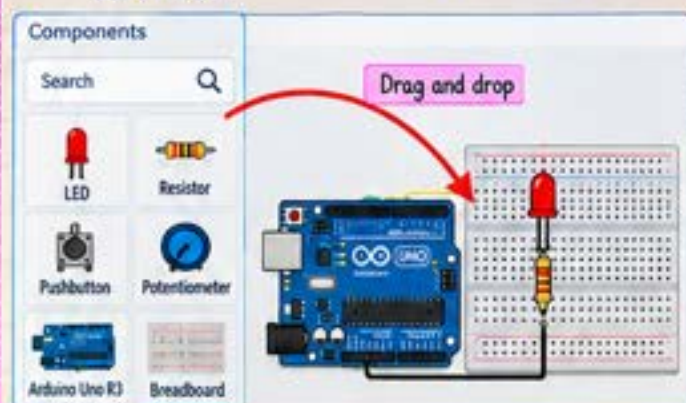
45 TINKERCAD DASHBOARD

- After login, we see the Tinkercad dashboard.
- From here we can create new circuits, 3D designs and code projects.
- Our saved projects are also shown here.



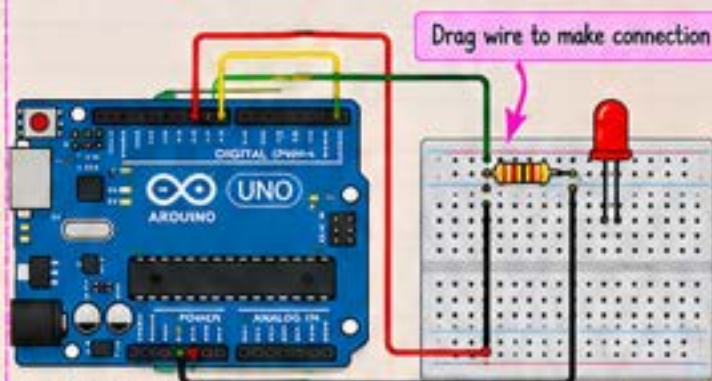
46 ADDING COMPONENTS

- Components are added from the components panel on the left side.
- We can drag and drop components (LED, resistor, sensor, motor, etc.) to the work area.
- We can also search for a component using the search box.



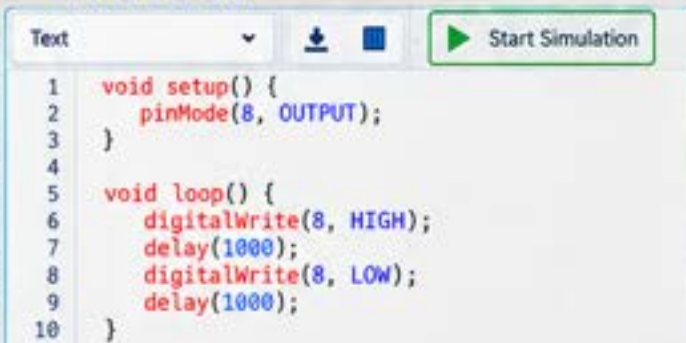
47 MAKING CONNECTIONS

- We can make connections by dragging wires between component pins.
- Different colored wires can be used.
- Connections in Tinkercad work like real hardware.



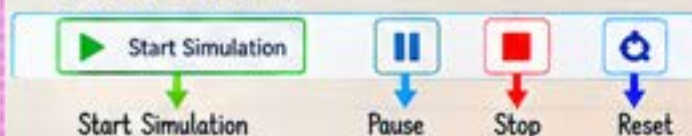
48 CODE EDITOR

- We can write Arduino code in the code editor.
- Tinkercad supports block coding, block + text and text coding (C++).
- After writing the code, click on "Start Simulation" to run the circuit.



49 RUNNING THE SIMULATION

- Click on Start Simulation to see the working of the circuit.
- We can pause, stop or reset the simulation.
- The output (LED, motor, sensor value, etc.) can be observed in real time.



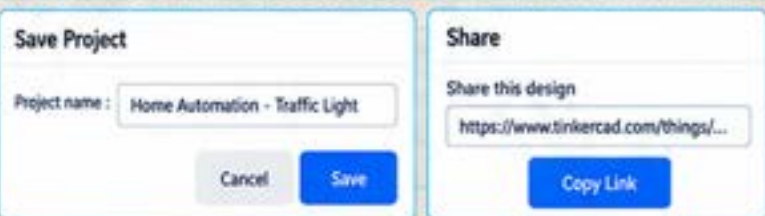
50 DEBUGGING AND MODIFYING

- If the circuit does not work, we can check for:
 - Wrong connections
 - Wrong pin numbers in code
 - Missing components
 - Incorrect values (e.g., resistor)
- We can change the code or connection and run the simulation again.
- This helps us learn by trial and error.



51 SAVING AND SHARING PROJECTS

- We can save our project by giving a name.
- Saved projects appear in the dashboard.
- We can edit the project later.
- Projects can also be shared with others using a link.



PROJECT: HOME AUTOMATION

Chapter Notes

MORE CORE CONCEPTS (59-65)



59 SENSORS

- A sensor is a device that detects changes in the environment and gives an electrical signal.
- Sensors help in making our project smart and automatic.
- Examples: Light sensor, motion sensor, temperature sensor, ultrasonic sensor, etc.

Examples of Sensors in Tinkercad



60 ACTUATORS

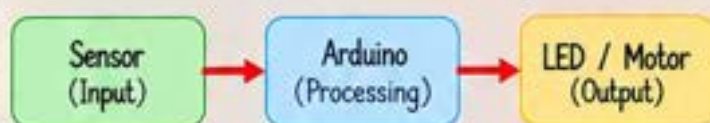
- An actuator is a device that performs a physical action based on a signal from Arduino.
- Actuators help in controlling real world devices.
- Examples: LED, buzzer, DC motor, servo motor, relay, etc.

Examples of Actuators in Tinkercad



61 INPUT AND OUTPUT

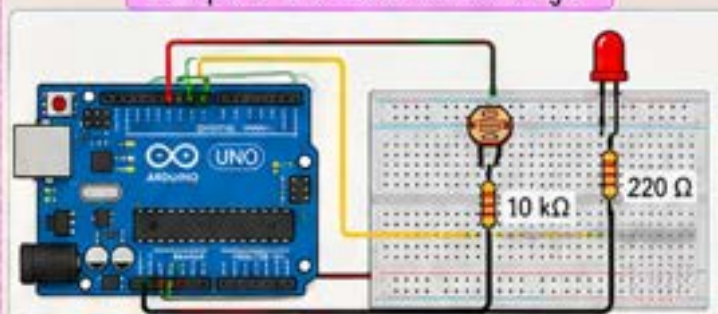
- Sensors are input devices (they send information to Arduino).
- Actuators are output devices (they perform an action).
- Arduino reads input, processes it and gives output.



63 USING SENSORS AND ACTUATORS

- In Tinkercad, we can connect sensors and actuators to Arduino and write code to control them.
- For example, an LDR can detect light and turn ON an LED automatically.

Example : LDR based Automatic Light



65 USING DELAY AND TIMING

- The `delay()` function pauses the program for a specified time (in milliseconds).
- It is used to control the speed of actions in a project.
- Example: `delay(1000);` // wait for 1 second

62 DIGITAL AND ANALOG PINS

- Arduino has two types of pins:
- Digital Pins (0-13): give HIGH (5V) or LOW (0V) signal.
- Analog Pins (A0-A5): used to read varying values from sensors (e.g., LDR, temperature sensor).



64 CONDITIONAL LOGIC (IF-ELSE)

- We use if-else statements in Arduino to take decisions based on sensor values.
- If the condition is true, one action is performed, otherwise another action is performed.

Example Code (LDR controlled LED)

```

const int ldrPin = A0; // LDR connected to A0
const int ledPin = 8;  // LED connected to D8
int sensorValue;

void setup() {
  pinMode(ledPin, OUTPUT);
  Serial.begin(9600);
}

void loop() {
  sensorValue = analogRead(ldrPin);
  if (sensorValue < 500) { // dark
    digitalWrite(ledPin, HIGH);
  } else { // bright
    digitalWrite(ledPin, LOW);
  }
  delay(500);
}
  
```



Key Points :

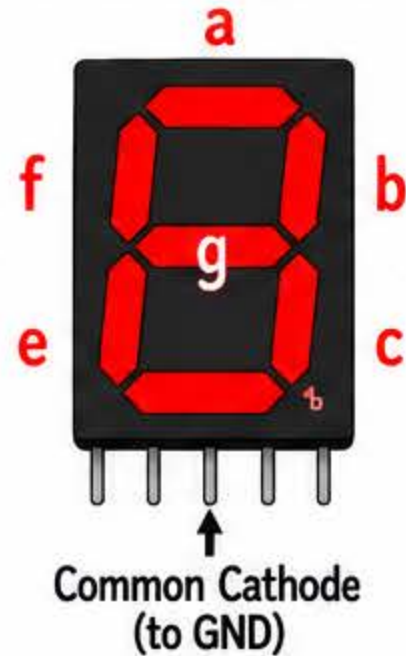
- Sensors give input and actuators give output.
- Arduino uses digital and analog pins to read and control devices.
- We use if-else conditions to make decisions.
- Delay helps to control the timing of actions.
- In Tinkercad, we can test all these components and ideas before making the real project.

7-SEGMENT DISPLAY

1 Introduction

- A 7-segment display is an electronic component used to display numbers (0–9) and some letters.
- It has 7 LED segments arranged in the shape of 8
- Each segment is labeled as **a**, **b**, **c**, **d**, **e**, **f**, **g**.
- In this project we are using Common Cathode type.
- The common cathode pin is connected to GND, and each segment is connected to an Arduino digital pin through a **220 Ω** resistor.

2 7-Segment Display (Common Cathode)



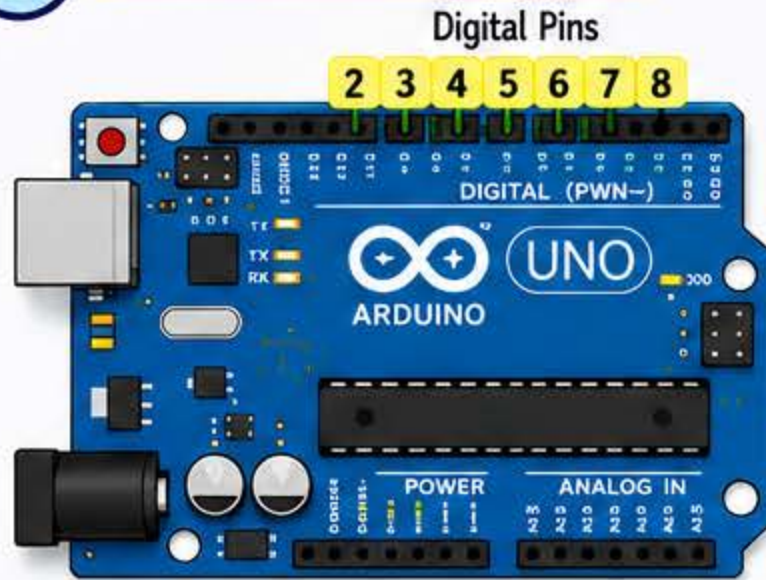
Segments:

- a** – top
- b** – top right
- c** – bottom right
- d** – bottom
- e** – bottom left
- f** – top left
- g** – middle

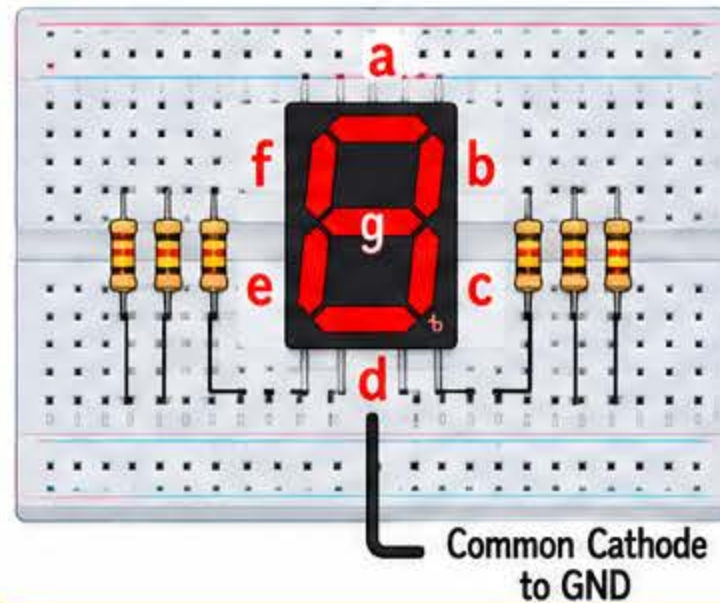
3 Pin Mapping (Arduino Uno)

Segment	Arduino Pin	Resistor Value
a	2	220 Ω
b	3	220 Ω
c	4	220 Ω
d	5	220 Ω
e	6	220 Ω
f	7	220 Ω
g	8	220 Ω
Common Cathode	–	GND

4 Arduino Uno Board



5 7-Segment Display on Breadboard



Each segment is connected to the respective Arduino pin through a **220 Ω** resistor.

6 Digit 0 – 9 (Segment ON/OFF)

Digit	a	b	c	d	e	f	g
0	ON	ON	ON	ON	ON	ON	OFF
1	OFF	ON	ON	OFF	OFF	OFF	OFF
2	ON	ON	OFF	ON	ON	OFF	ON
3	ON	ON	ON	ON	OFF	OFF	ON
4	OFF	ON	ON	OFF	OFF	ON	ON
5	ON	OFF	ON	ON	OFF	ON	ON
6	ON	OFF	ON	ON	ON	ON	ON
7	ON	ON	ON	OFF	OFF	OFF	OFF
8	ON	ON	ON	ON	ON	ON	ON
9	ON	ON	ON	ON	OFF	ON	ON

7 Key Points

- Use 220 Ω resistor with each segment to limit current.
- We are using Common Cathode type.
- Common cathode pin is connected to GND.
- Segments are connected to Arduino digital pins (a $\rightarrow$ 2, b $\rightarrow$ 3, c $\rightarrow$ 4, d $\rightarrow$ 5, e $\rightarrow$ 6, f $\rightarrow$ 7, g $\rightarrow$ 8).
- A segment glows when the corresponding Arduino pin is set HIGH.
- We can display digits 0 to 9 and some letters (e.g., A, b, C, etc.).
- This circuit can be easily made in Tinkercad or on a breadboard.

1 What is Arduino Uno?

- Arduino Uno is a programmable microcontroller board used to build electronic, robotics and automation projects.
- It can read inputs from sensors, process them and control outputs like LEDs, motors, etc.
- It is easy to use and widely used in education and projects.

4 Power Pins

Pin	Description
IOREF	Reference voltage for shields
RESET	Resets the microcontroller
3.3V	3.3 V regulated output
5V	5 V regulated output
GND	Ground (0 V)
GND	Ground (0 V)
VIN	External input voltage (7 - 12 V)

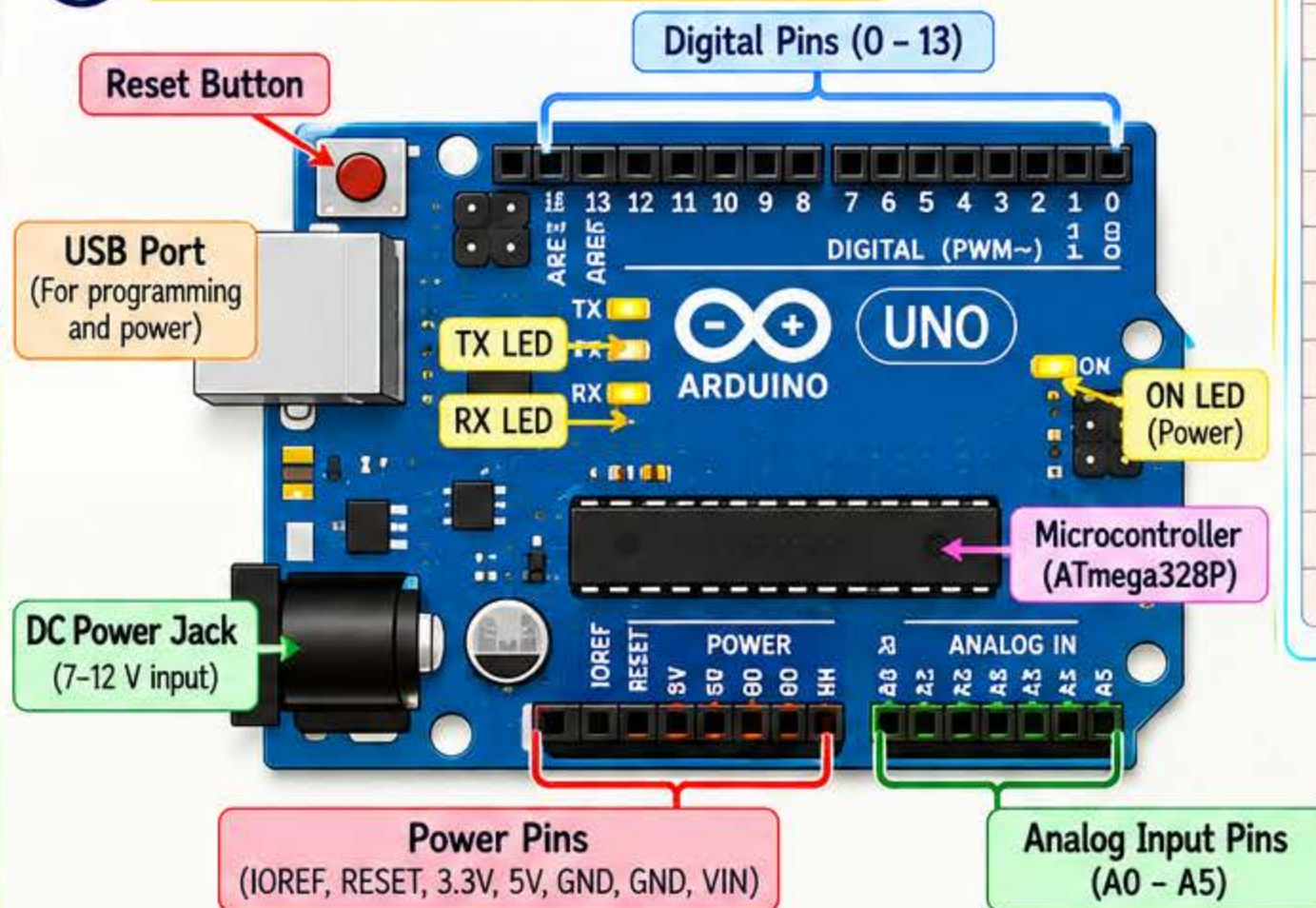
5 Analog Input Pins (A0 - A5)

Pin	Function	Special Use
A0	Analog input	Can also be used as digital I/O
A1	Analog input	Can also be used as digital I/O
A2	Analog input	Can also be used as digital I/O
A3	Analog input	Can also be used as digital I/O
A4	Analog input	I ² C SDA (also digital I/O)
A5	Analog input	I ² C SCL (also digital I/O)

ARDUINO UNO

COMPLETE EXPLANATION WITH PINS

2 Parts of Arduino Uno Board



3 Digital Pins (0 - 13)

Pin No.	Function	PWM	Special Use
0	Digital I/O	-	RX (Serial Receive)
1	Digital I/O	-	TX (Serial Transmit)
2	Digital I/O	-	General Purpose
3	Digital I/O	✓	PWM Output
4	Digital I/O	-	General Purpose
5	Digital I/O	✓	PWM Output
6	Digital I/O	✓	PWM Output
7	Digital I/O	-	General Purpose
8	Digital I/O	-	General Purpose
9	Digital I/O	✓	PWM Output
10	Digital I/O	✓	PWM Output
11	Digital I/O	✓	PWM Output
12	Digital I/O	-	General Purpose
13	Digital I/O	-	Built-in LED

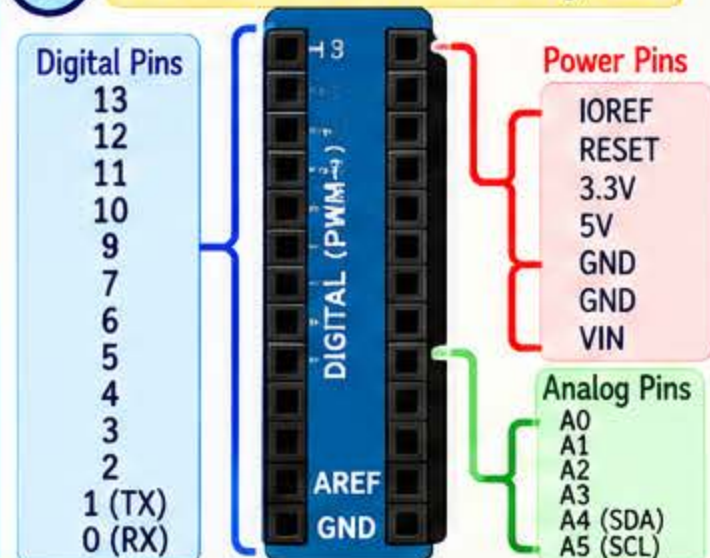
8 Built-in LED (Pin 13)



7 Important Points

- Digital pins give HIGH (5 V) or LOW (0 V).
- PWM pins: 3, 5, 6, 9, 10, 11.
- Pins 0 (RX) and 1 (TX) are used for serial communication.
- A0 - A5 are analog input pins (can also be used as digital).
- A4 = SDA and A5 = SCL for I²C devices.
- Use 5V or 3.3V as per your module requirement.
- GND is the common ground.
- Pin 13 has a built-in LED (ON when HIGH).
- Arduino Uno uses ATmega328P microcontroller.

6 Arduino Uno Pin Layout

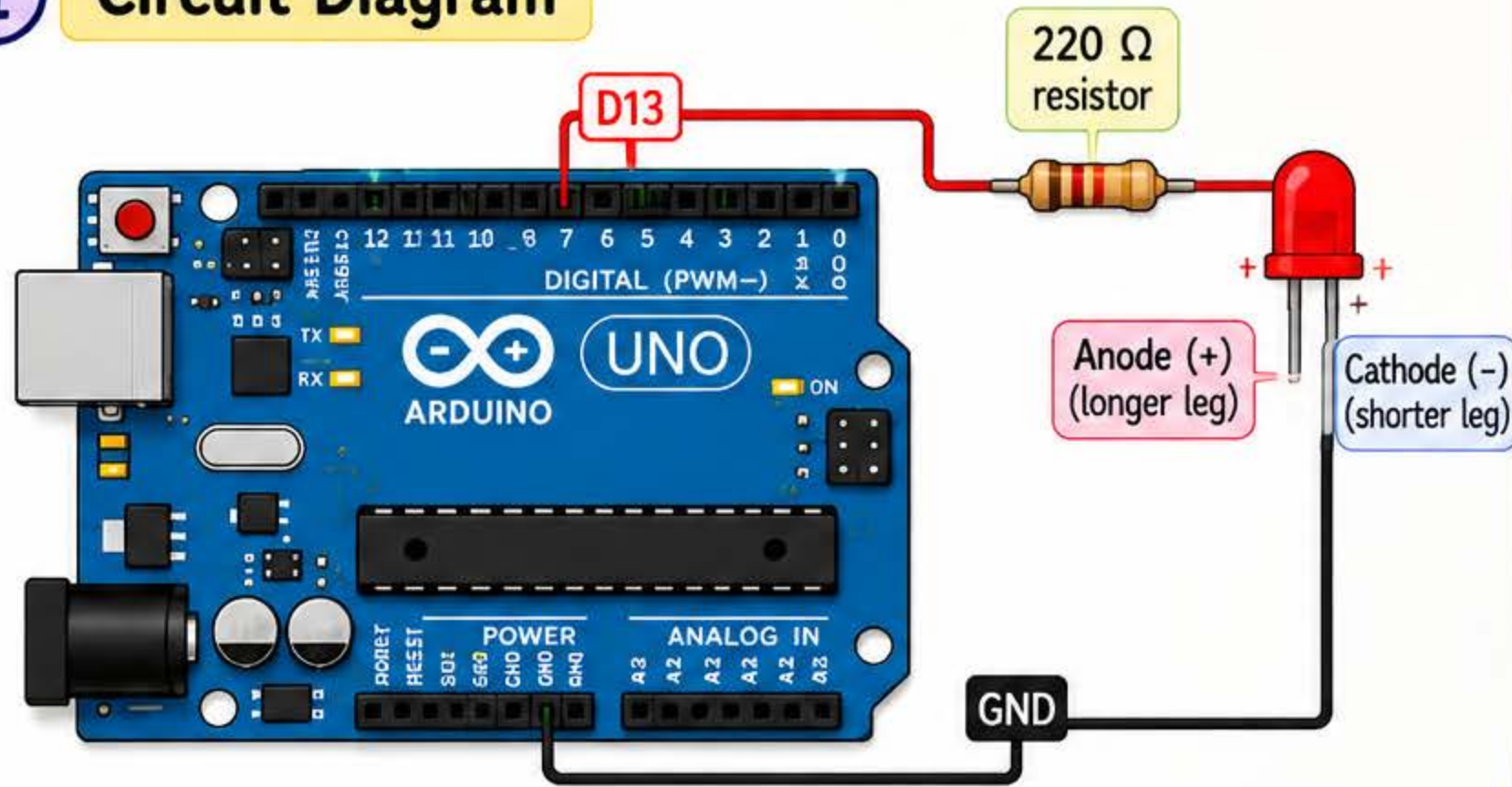


9 USB and Power Supply

- **USB Port:** Used for programming and also provides 5 V power.
- **DC Power Jack:** Can be used with 7 - 12 V adapter.
- 5V and 3.3V pins can be used to power other components (within limits).

BLINKING A SINGLE LED

1 Circuit Diagram



2 Arduino Code

```
void setup() {  
  pinMode(13, OUTPUT); // set pin 13 as output  
}  
  
void loop() {  
  digitalWrite(13, HIGH); // turn LED ON  
  delay(1000); // wait for 1 second  
  
  digitalWrite(13, LOW); // turn LED OFF  
  delay(1000); // wait for 1 second  
}
```

3 Components Used



LED
(5 mm)



220 Ω
Resistor
(1/4 W)



Arduino Uno

4 Working

- Pin 13 is set as OUTPUT.
- When pin 13 is HIGH (5 V), current flows through the 220 Ω resistor, and the LED glows.
- The LED remains ON for 1 second (delay 1000 ms).
- Then pin 13 is set to LOW (0 V), so the LED turns OFF.
- This repeats continuously in the `loop()`.

5 Key Points

- Always use a 220 Ω resistor with an LED to limit current.
- Connect the longer leg (anode) of the LED to Arduino pin 13 through the resistor.
- Connect the shorter leg (cathode) to GND.
- Pin 13 has a built-in LED on the board (you will also see the onboard LED blinking).
- You can use any other digital pin instead of 13.