PRACTICAL: 1A

AIM: Introduction to Arduino

Arduino is an open-source electronics platform based on easy-to-use hardware and software. It is widely used for building electronic projects, prototyping, and learning programming. The most popular model, **Arduino Uno**, is based on the **ATmega328P microcontroller** and features both **digital and analog input/output (I/O) pins**.

Input Pins of Arduino Uno

1. Digital Input Pins (0 - 13)

- Pins 0 and 1 (RX/TX) → Used for serial communication (Receiving and Transmitting data).
- Pins 2 to 13 → Can be used as digital input pins to read HIGH (5V) or LOW (0V) signals.

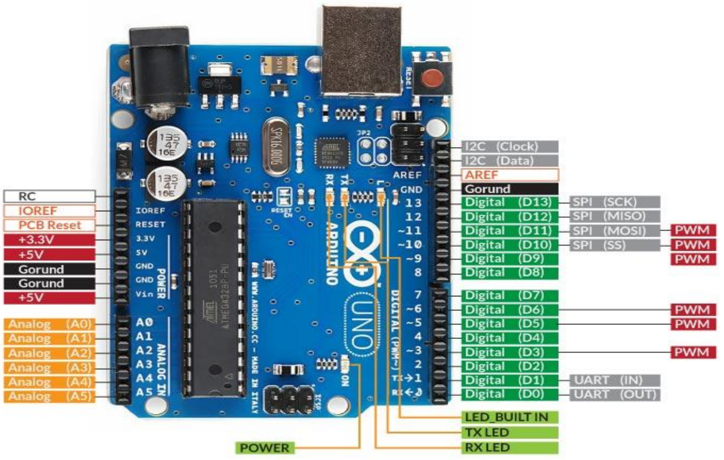
2. Analog Input Pins (A0 - A5)

- Pins A0 to A5 → Used to read analog signals from sensors (values range from 0 to 1023).
- These pins use a 10-bit Analog-to-Digital Converter (ADC) to convert analog voltage into a digital value.

3. Special Input Pins

- AREF (Analog Reference Pin) → Provides a reference voltage for analog input readings.
- RESET Pin → Used to reset the Arduino board when pulled LOW.

ARDUINO IMAGE



PRACTICAL: 1B

AIM: Introduction to Arduino circuits and BreadBoarding.

1. What is Arduino?

Arduino is an open-source microcontroller board designed for electronics prototyping and embedded systems development. It is widely used by beginners and professionals to build automated projects, robotics, and IoT applications.

Key Features of Arduino:

- Easy-to-use hardware and software
- Multiple digital and analog input/output pins
- Compatible with sensors, motors, LEDs, and displays

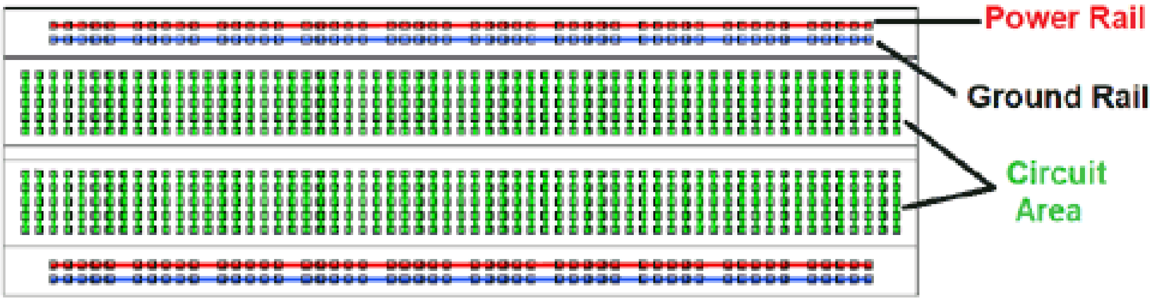
2. What is a Breadboard?

A breadboard is a reusable prototyping tool used to build and test electronic circuits without soldering. It consists of holes and internal metal strips that create electrical connections when components are inserted.

Breadboard Layout:

- Power Rails (Red & Blue/Black) → Used for power distribution (VCC & GND)
- Circuit Area (Green Section) → Used for placing components and wires
- Middle Gap → Separates the two halves, typically for integrated circuits (ICs)

BREADBOARD IMAGE



Key Breadboard Rules

1. Power and ground rails are connected horizontally, making it easier to distribute power.
2. Each vertical column in the circuit area is connected internally, meaning components placed in the same column will share a connection.
3. The middle gap isolates the two halves, allowing ICs to be placed in the center without shorting their pins.

PRACTICAL: 1C

AIM: Blinking of LEDs.

COMPONENTS REQUIRED:

1. **Arduino UNO** (Main microcontroller board)
2. **Breadboard** (For easy prototyping)
3. **LED** (Light-emitting diode, indicates output)
4. **Resistor (220Ω)** (Limits current to protect the LED)
5. **Jumper Wires** (To make electrical connections)

CONNECTION:

Step 1: Gather Components

- Take an **Arduino UNO** and a **breadboard**.
- Ensure you have an **LED, 220Ω resistor, and jumper wires** for connections.

Step 2: Identify LED Terminals

- **Anode (+) → Longer leg →** Connects to the **Arduino's digital pin 13**.
- **Cathode (-) → Shorter leg →** Connects to **GND through a resistor**.

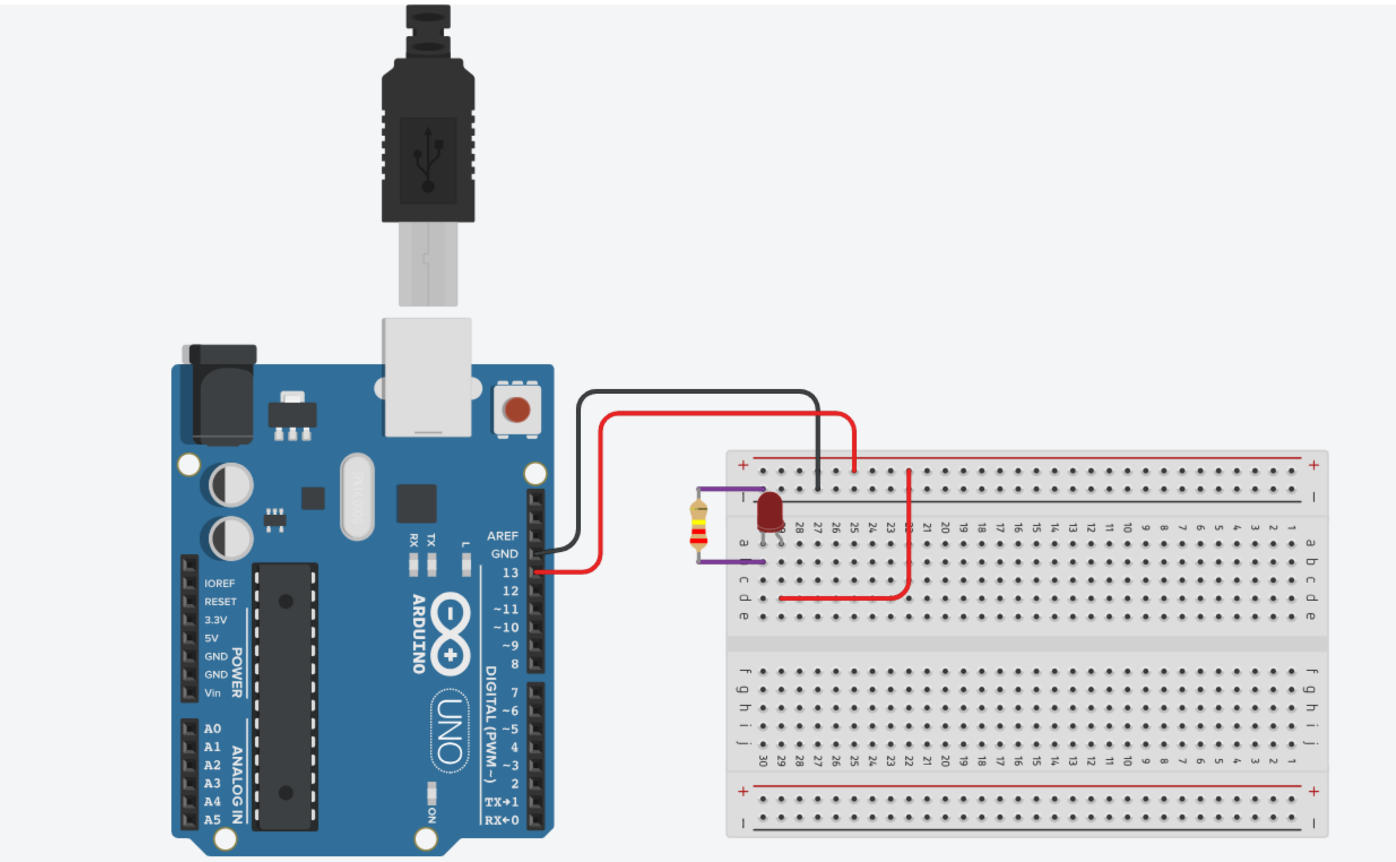
Step 3: Connecting the Resistor

- Insert the **220Ω resistor** into the breadboard.
- One end of the **resistor connects to the LED's cathode (-)**.
- The **other end of the resistor connects to GND** of Arduino (to limit current and protect the LED).

Step 4: Connecting the LED Anode

- The **LED's anode (+) connects to digital pin 13** on the Arduino board using a jumper wire.

CONNECTION DIAGRAM



CODE

```
void setup() {
   pinMode(13, OUTPUT);
}


void loop() {
   digitalWrite(13, HIGH);
   delay(1000);
   digitalWrite(13, LOW);
   delay(1000);
}
```

SUMMERY OF CODE

- **pinMode(13, OUTPUT);** → Sets pin 13 as an output.
- **digitalWrite(13, HIGH);** → Turns the LED ON.
- **delay(1000);** → Waits for 1 second.
- **digitalWrite(13, LOW);** → Turns the LED OFF.
- **delay(1000);** → Waits for 1 second.