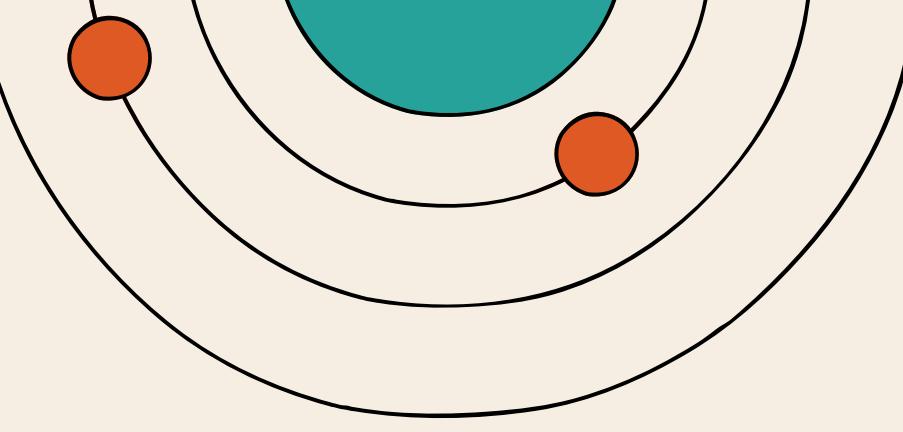
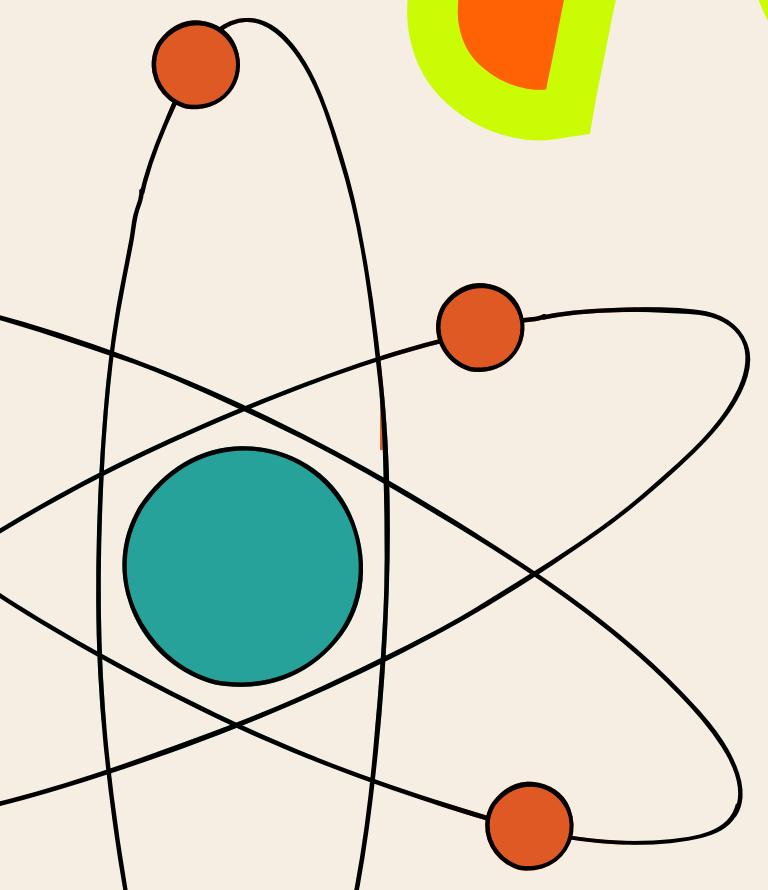
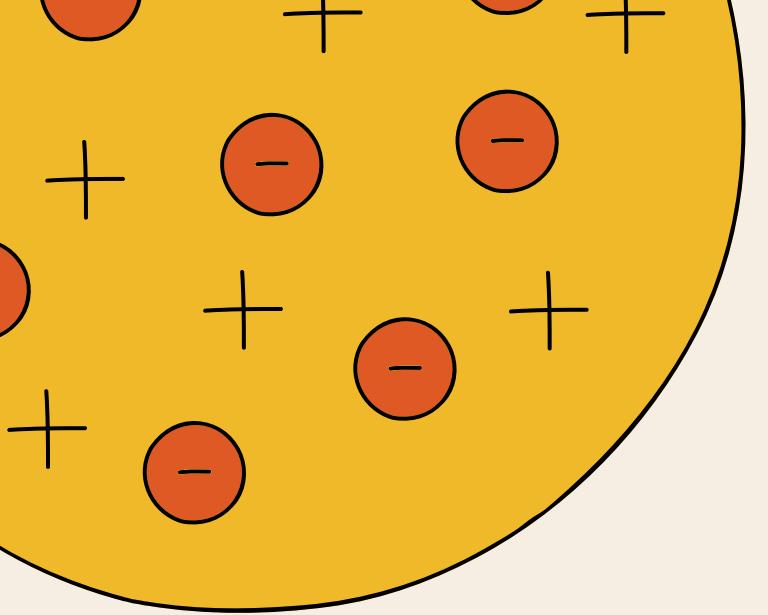
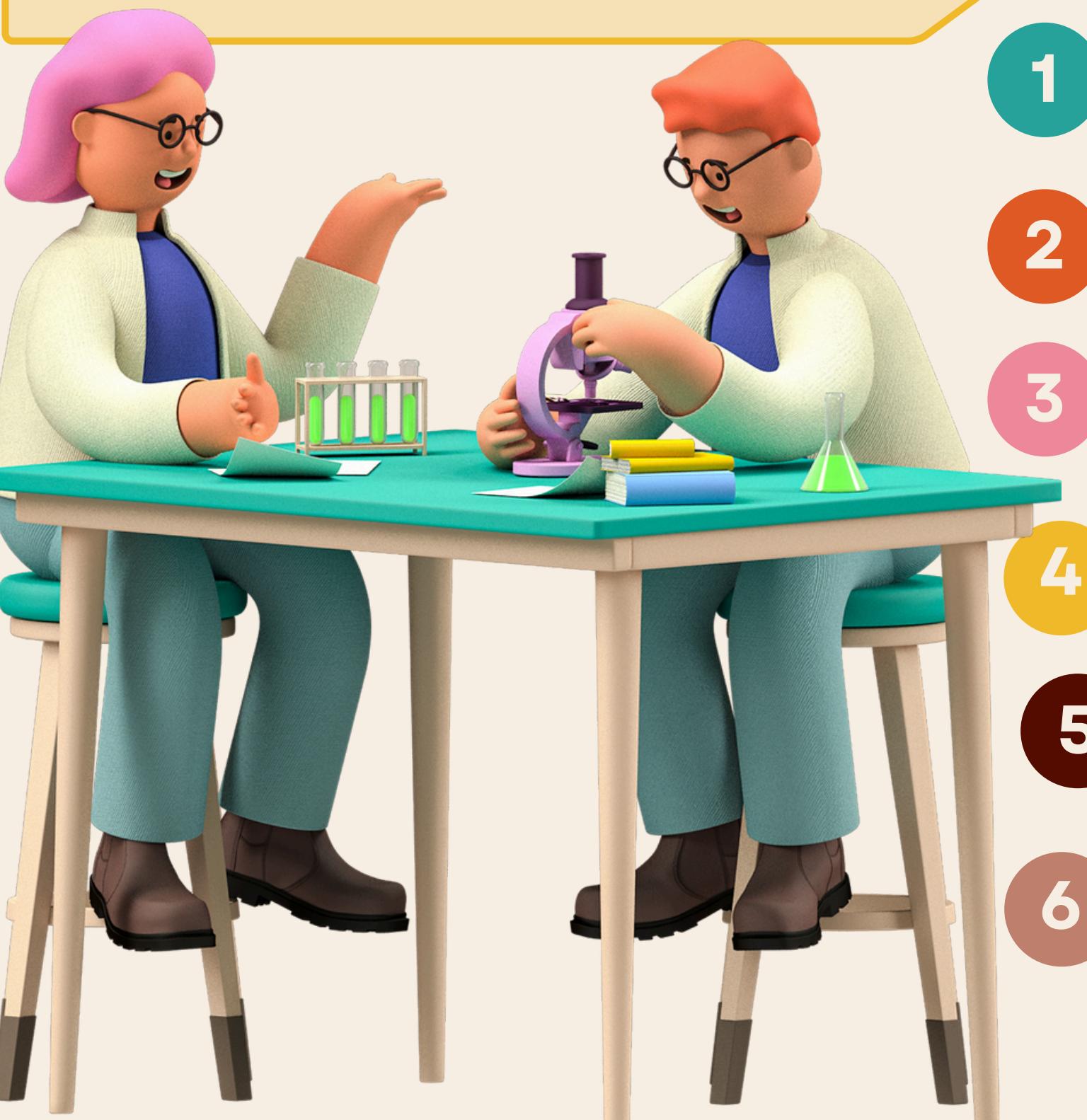


fire alarm

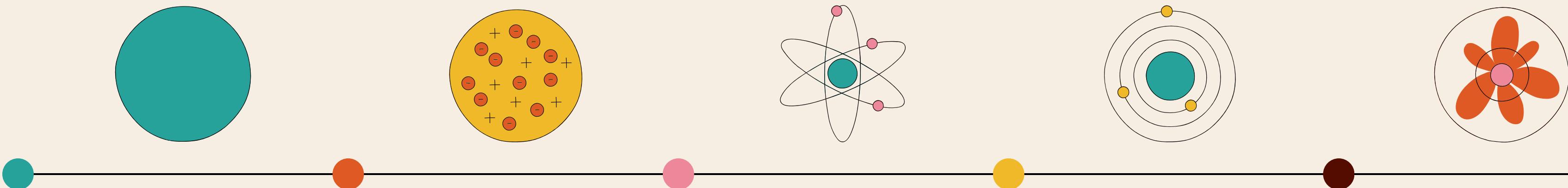


MEMBERS

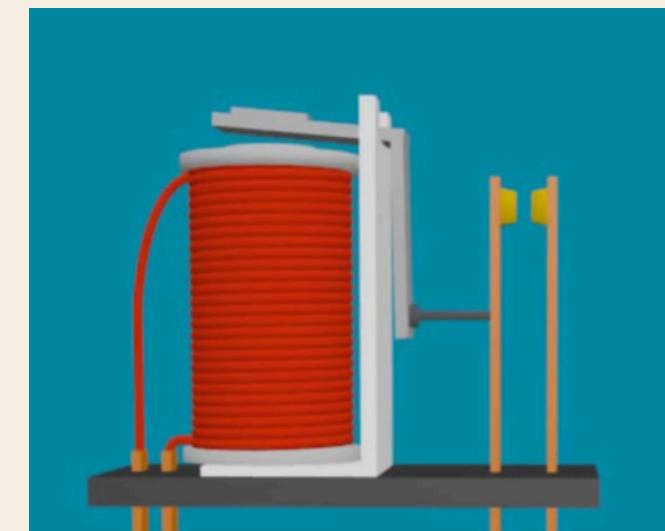
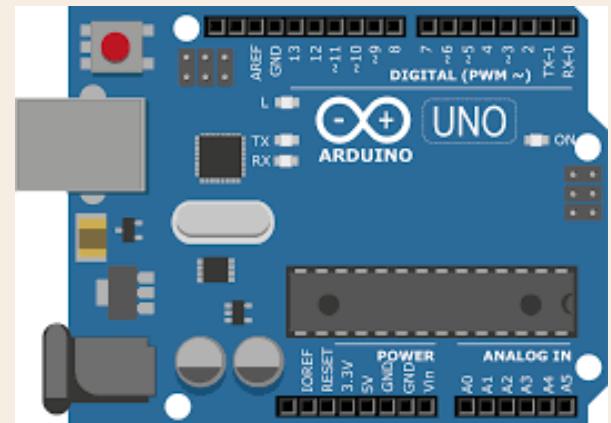
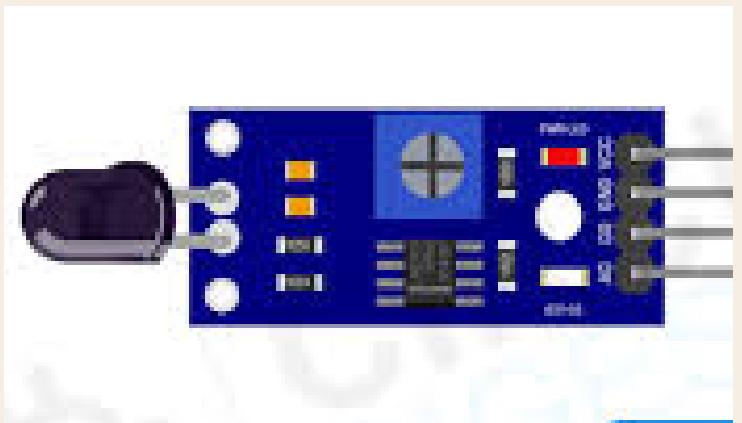


- 1 Aayati Srivastava [RA2411003011184]
- 2 Shambhavi Mishra [RA2411003011211]
- 3 Akshat Sharma [RA2411003011204]
- 4 Akshadha [RA2411003011217]
- 5 Aditi Agarwal [RA2411003011234]
- 6 Arpita Singh [RA24110030112141]

COMPONENTS



Flame Sensor **Arduino Relay UNO** **Water Pump and Switch** **Battery**



FLAME SENSOR

The flame sensor detects the infrared radiation emitted by fire.

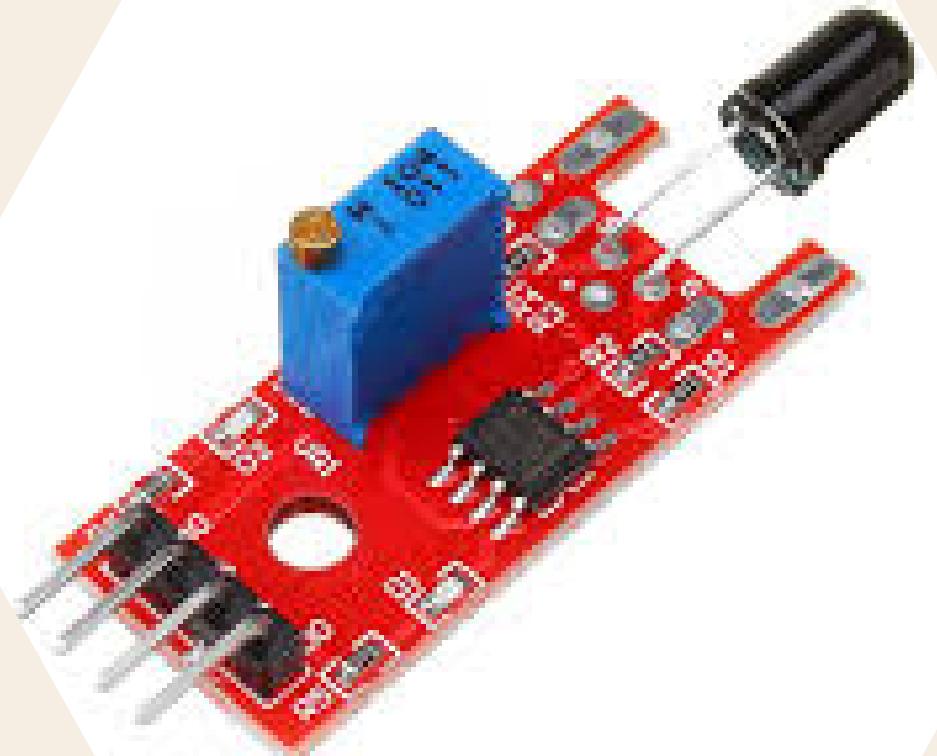
When fire is detected, it gives a digital LOW signal to Arduino.

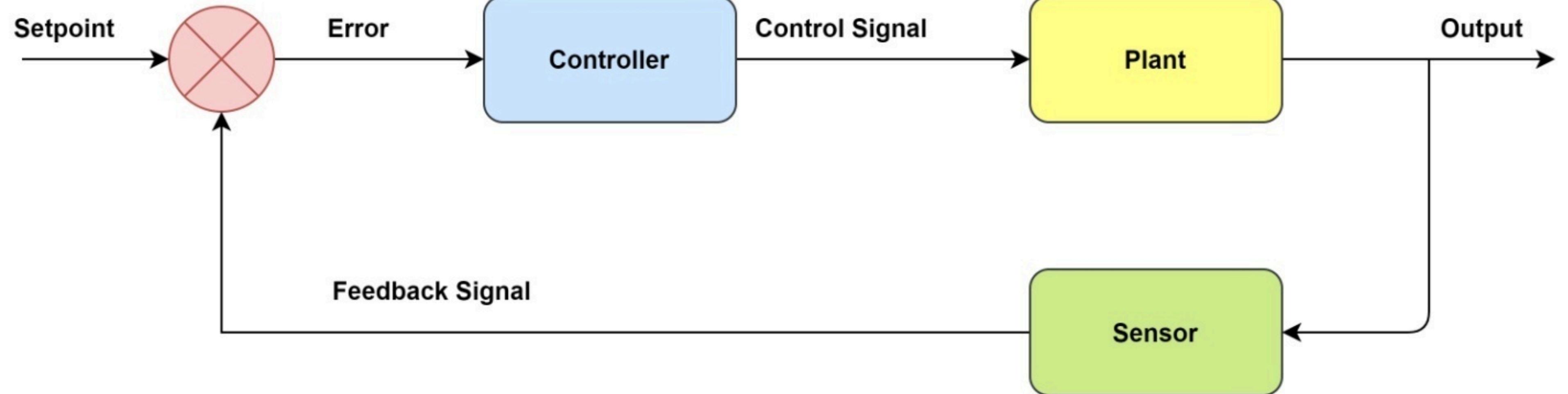
It helps the system respond as soon as a fire is present.

The sensor is placed at the top-right side of the project.

It helps the system respond as soon as a fire is present.

It is fast, sensitive, and has a good detection range.





- Setpoint: Desired output value.
- Error: Difference between setpoint and measured output.
- Controller: Processes error and sends control signal.
- Plant: System or process being controlled.
- Sensor: Measures output and provides feedback.
- Feedback Loop: Continuously adjusts to minimize error.

FLAME SENSOR COMPONENT

Black Component on Top → IR Flame Sensor

This black dome-like thing is actually an Infrared (IR) Sensor.

It detects infrared light (like the light from a flame, fire, etc.).

Flames emit infrared radiation — this sensor catches it.

Three Small Cylindrical Components → Resistors

Those three small components near the top are resistors.

They control the flow of electrical current
to protect the circuit from burning.



FLAME SENSOR COMPONENT

Blue Box with a Screw → Potentiometer (Adjustable Resistor)

This blue box is a variable resistor.

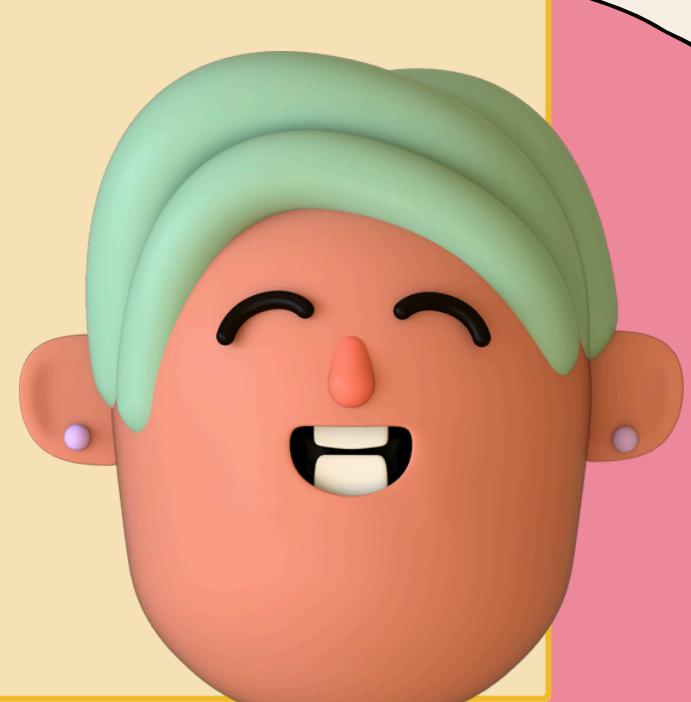
Use a screwdriver to rotate the screw – it adjusts sensitivity.

(Example: How close or far a flame should be for the sensor to detect it.)

Black Square Component → Comparator IC (LM393)

This is an LM393 or similar Comparator IC (small chip).

It compares the amount of infrared
light and decides whether
to give an output signal (HIGH or LOW).



FLAME SENSOR COMPONENT

Three Pins at Bottom → Output Pins

VCC → Connect to +5V (power supply).

GND → Connect to Ground (OV).

DO → Digital Output (gives HIGH or LOW depending on flame detection).

Sometimes modules also have AO (Analog Output), but yours mainly shows DO.

Small Round White Thing → Power LED Indicator

This is a small LED light.

It glows when the module gets power.

It helps you know the sensor is on and working.

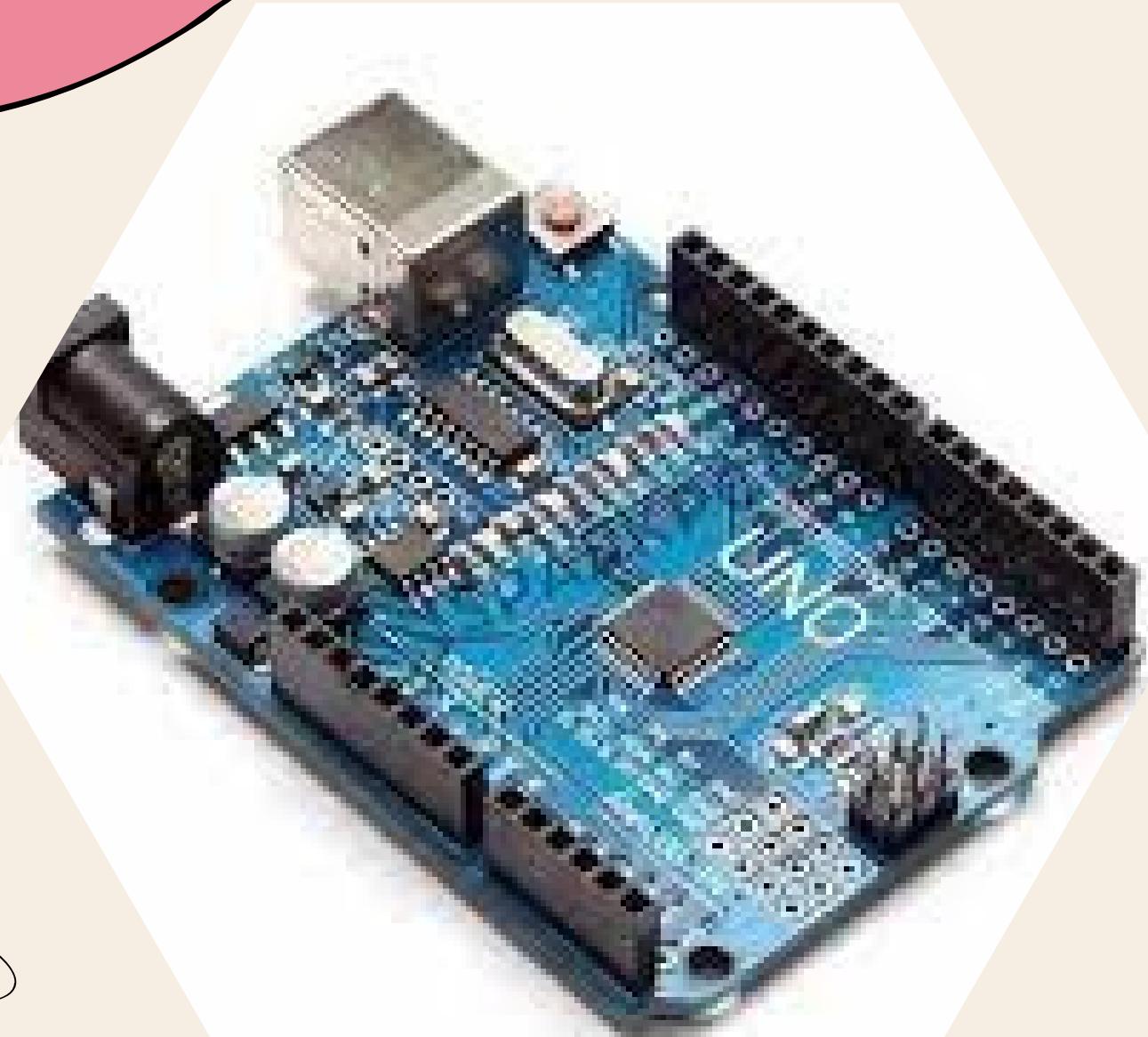
FLAME SENSOR COMPONENT

PCB (Printed Circuit Board)

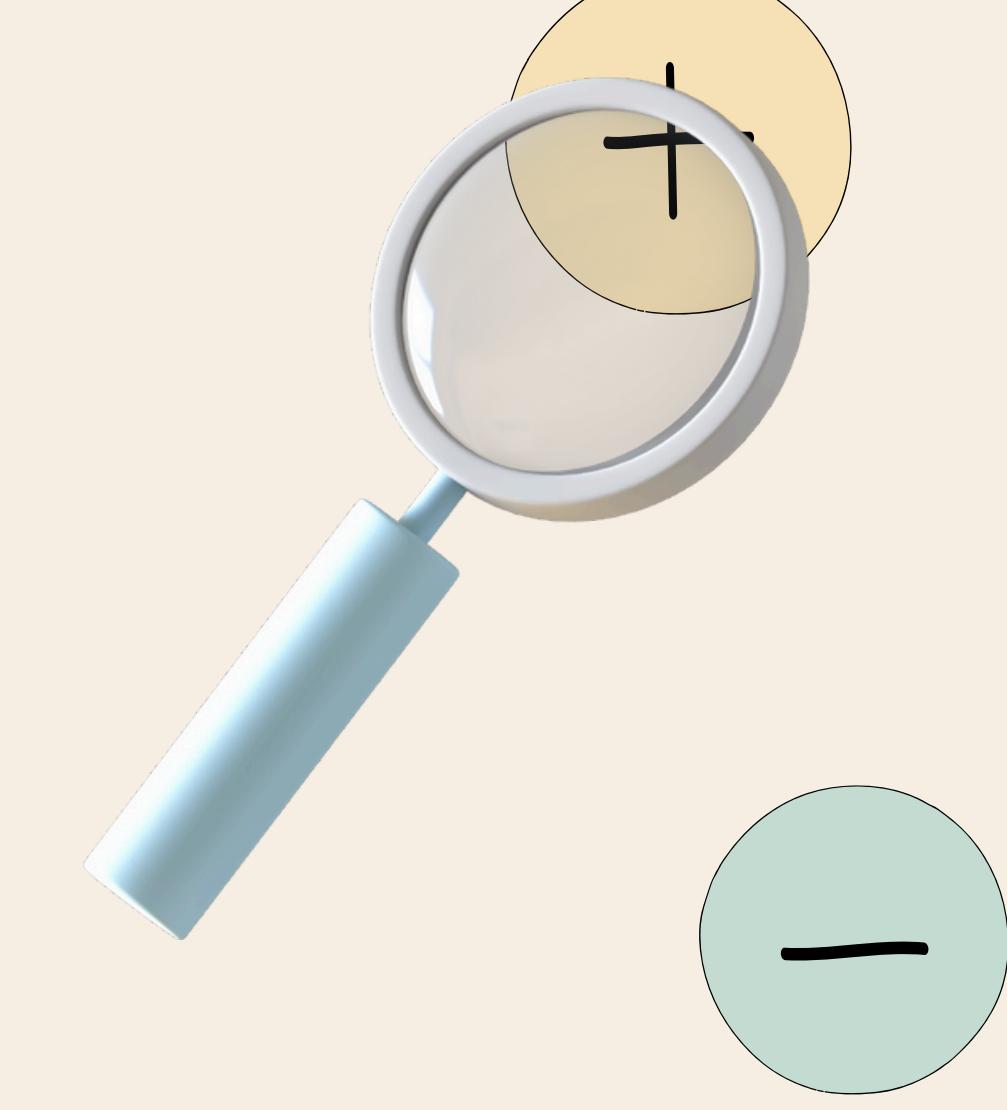
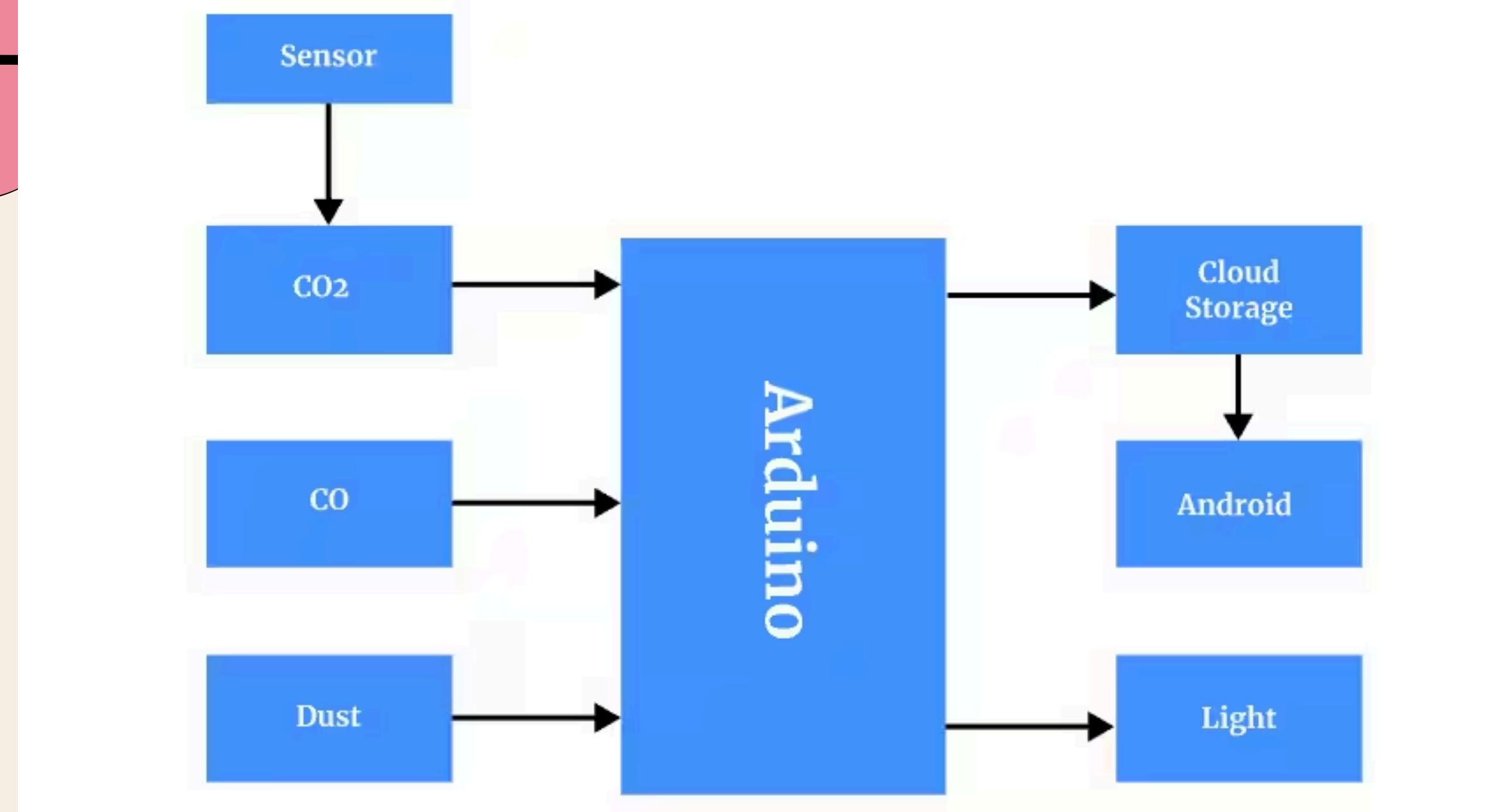
The blue board itself is called a PCB.

It connects all components together using printed copper lines.

ARDUINO UNO



- Arduino UNO is the central controller of this project.
- It takes input from the flame sensor and processes it based on the logic coded.
- If fire is detected, it sends a HIGH signal to the relay module.
- Arduino is powered by a 9V battery and can also be programmed via USB.
- It uses C++-based code written in Arduino IDE.
- Acts as the decision-maker of the whole setup.



- Sensors detect CO₂, CO, and Dust levels.
- Arduino processes sensor data.
- Data is sent to Cloud Storage and Android devices for monitoring.
- Light control activated based on sensor reading



CODE USED IN ARDUINO UNO

```
// Fire Alarm and Extinguisher System using Flame Sensor and Relay (Buzzer Removed)
```

```
// Pin Definitions
```

```
const int flameSensorPin = A0; // Flame sensor analog output
```

```
const int relayPin = 7; // Relay controlling water pump or fan
```

```
const int ledPin = 6; // Optional LED indicator
```

```
// Threshold: Lower value means higher flame intensity
```

```
const int flameThreshold = 200; // You may need to adjust this
```

```
void setup() {
```

```
pinMode(flameSensorPin, INPUT);
```

```
pinMode(relayPin, OUTPUT);
```

```
pinMode(ledPin, OUTPUT);
```

```
digitalWrite(relayPin, LOW);
```

```
digitalWrite(ledPin, LOW);
```

```
Serial.begin(9600);
```

```
Serial.println("Fire Alarm and Extinguisher System Initialized (No Buzzer).");
```

```
}
```

CODE USED IN ARDUINO UNO



```
void loop() {
    int flameValue = analogRead(flameSensorPin);
    Serial.print("Flame Sensor Reading: ");
    Serial.println(flameValue);

    if (flameValue < flameThreshold) {
        // Fire detected
        Serial.println("** FIRE DETECTED! Activating Pump **");

        digitalWrite(relayPin, HIGH); // Turn on water pump via relay
        digitalWrite(ledPin, HIGH); // Turn on LED indicator
    } else {
        // No fire
        digitalWrite(relayPin, LOW); // Pump OFF
        digitalWrite(ledPin, LOW); // LED OFF

        Serial.println("No fire detected. System is safe.");
    }

    delay(500); // Delay for stability
}
```

RELAY MODULE



The relay acts like a switch that can be controlled electrically.

Arduino sends a HIGH signal to turn the relay ON.

This allows the relay to power the water pump with the battery.

This allows the relay to power the water pump with the battery.

Relay ensures safe control of high-current components.

It is placed at the bottom-left side of the setup.

WATER PUMP AND SIDE SWITCH

The water pump sprays water to simulate extinguishing the fire.

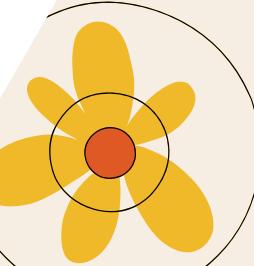
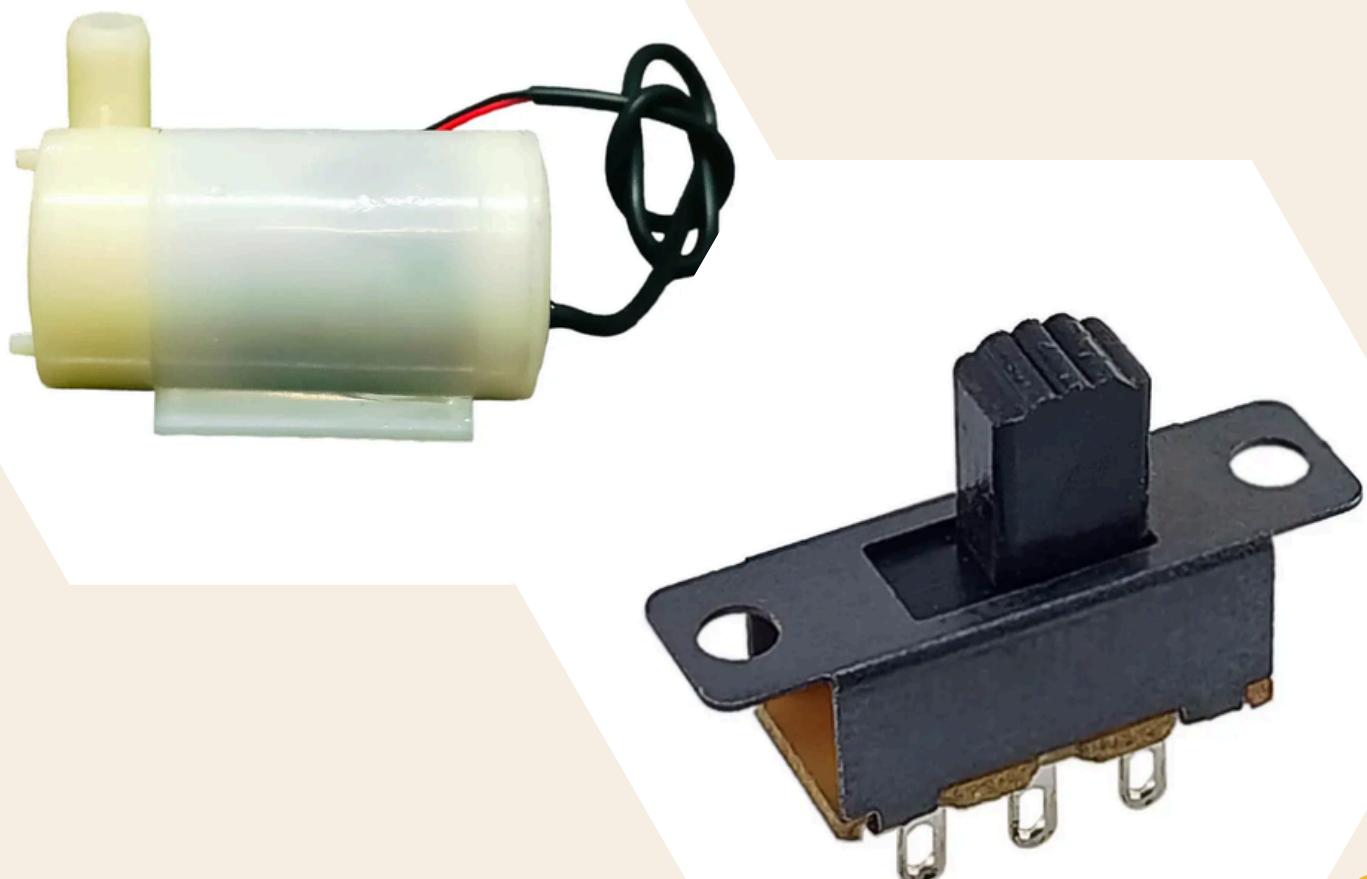
It is connected through the relay and powered by a 9V battery.

Starts automatically when fire is detected.

The slide switch gives users manual control to turn the pump ON/OFF.

It adds an extra safety or override option to the system.

Useful during testing or when manual intervention is needed.



POWER SUPPLY

The system uses two 9V batteries for isolation and safety.

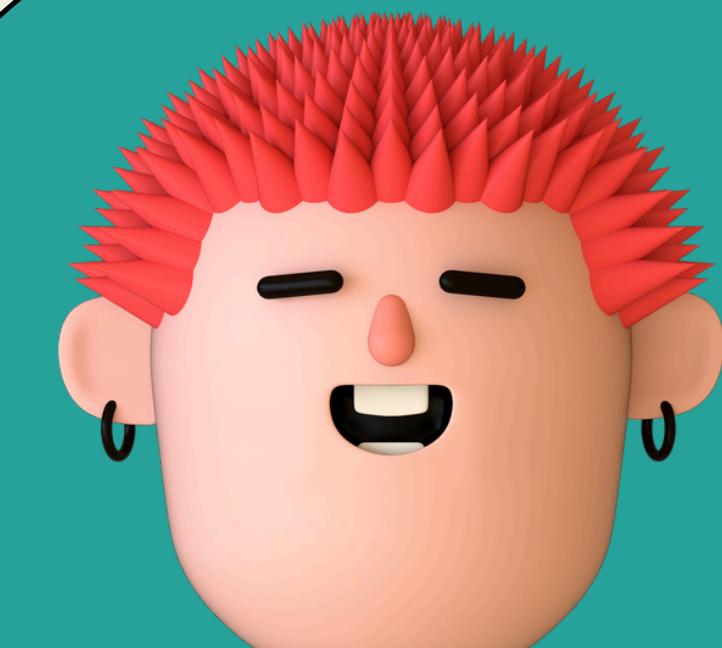
One battery powers the Arduino and sensors.

The other battery powers the motor through the relay.

This setup prevents overload and maintains efficiency.

Breadboard and jumper wires are used for easy connections.

Breadboard and jumper wires are used for easy connections.



CONTACT SWITCH



Homemade contact switch using two metal pins and a non-conductive base

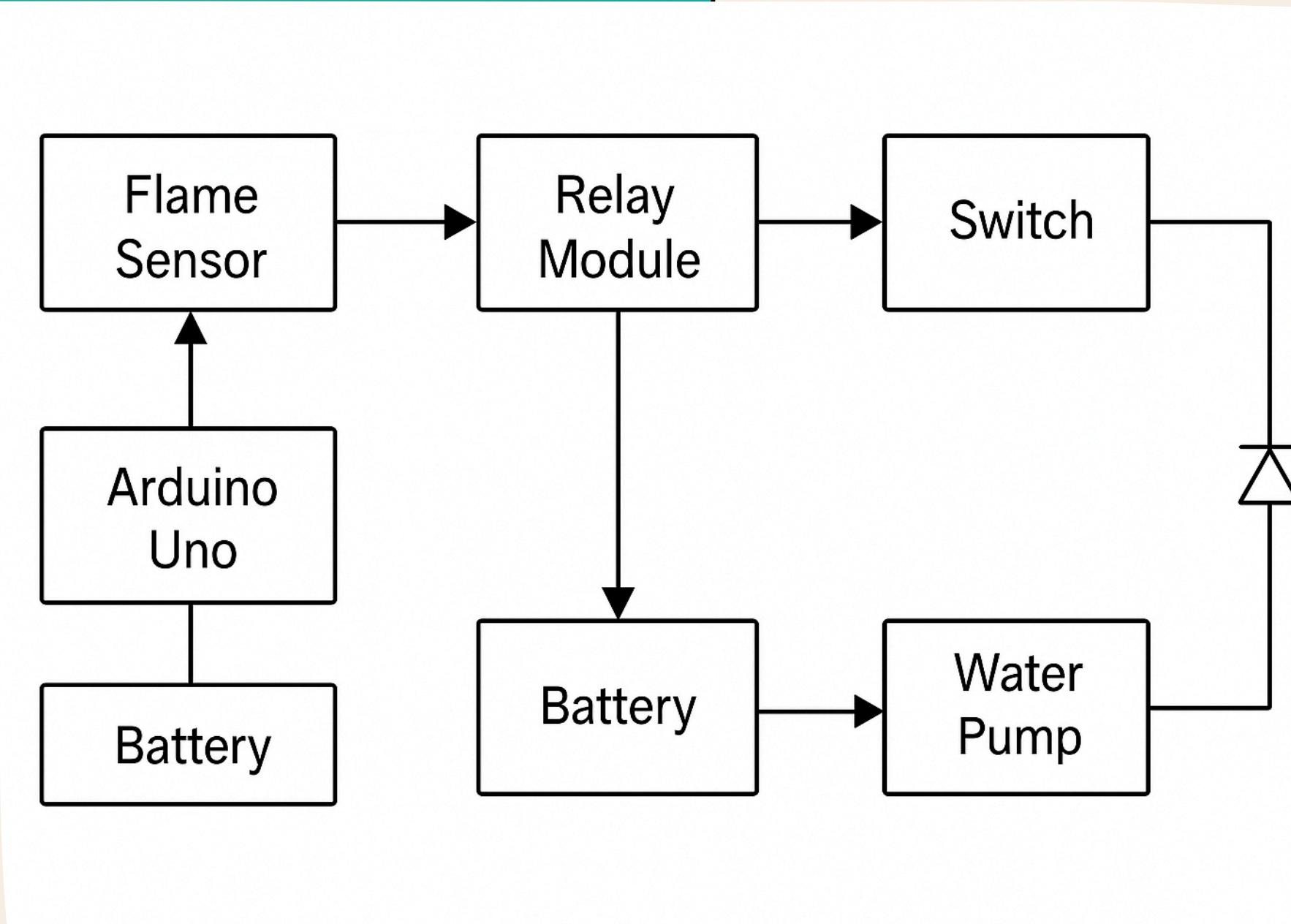
- Pins act as a simple ON/OFF switch when they touch
- Can detect manual press or flame-triggered separation (e.g., thread melts,

Acts as a basic normally open/closed switch

Low-cost, easy to build

Ideal for DIY fire alarms, burglar alarms, or touch triggers

BLOCK DIAGRAM OF FIRE ALARM



- Batteries provide separate power to Arduino and pump for isolation.
- Slide switch controls the pump circuit manually.
- All modules are connected via wires and a breadboard.
- You can add arrows and module icons in the diagram.
- This block diagram helps visualize the working of the system.

APPLICATIONS

- Can be used in kitchen safety systems to prevent accidents.
- Useful in school labs where flammable materials are used.
- Can be placed in small shops or warehouses.
- Good as a prototype for large industrial systems.
- Suitable for robotics or automation competitions.
- Educates students about embedded system safety projects.



thank
you