

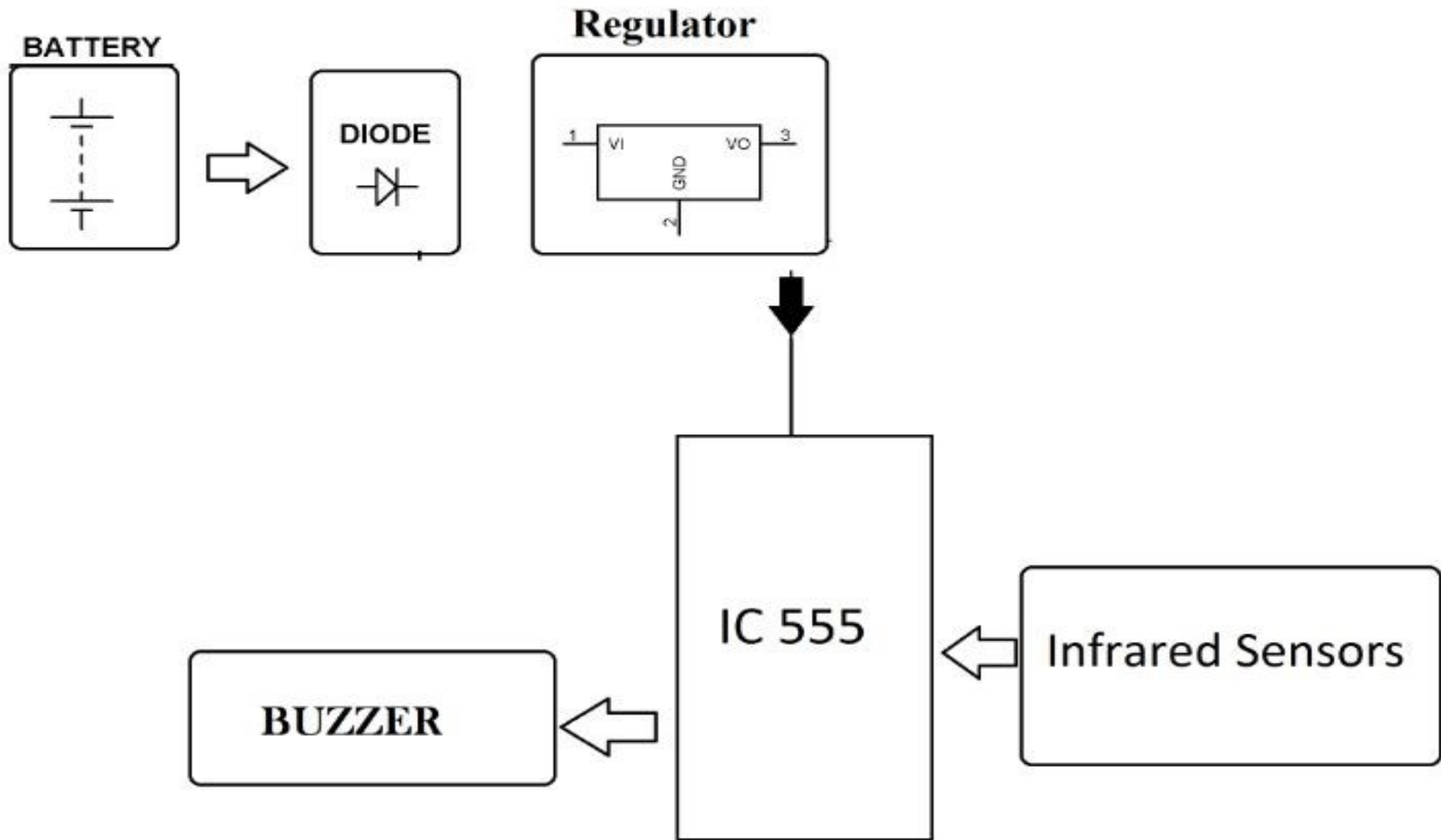
Automatic smoke
detector alarm

INTRODUCTION:

- In case of fire at home/office the internal parts of the premises gets filled with smoke. This smoke layer progressively becomes so thick that it lowers the visibility considerably. We can use this fact to construct our simple smoke detector alarm that can notify the residents in around the premises about the fire breakout.
- The Automatic Smoke Detector Alarm is very compact as it doesn't employs the costly smoke detecting sensors. It has pair of infrared sensors which are continuously monitoring the visibility in the premises where they are attached. In case a fire erupts, smoke gets generated and spreads in and around the premises. Once this smoke reaches the smoke detector module the visibility of the infrared sensors get obstructed. This triggers a fire alarm in the detector module. Hearing this alarm the resident in and around the premises can get alert about the fire and vacate the premises. The alarm goes ringing till there is smoke in the premises and goes off once the smoke recedes.

- The principle behind working of this project is of the opaqueness property of smoke. In case when there is no smoke the infrared sensors are continuously in view of each other. Once smoke comes in the view, the sensor signal is obstructed and the output of the sensor changes. This change is sensed by the system to ring an alarm to notify the residents in and around the premises about the fire breakout.

BLOCK DIAGRAM:



HARDWARE SPECIFICATION:

- I R Transmitter Receiver
- 555 Timer IC
- Buzzer
- Resistors
- Capacitors
- Transistors
- Cables and Connectors
- Diodes
- PCB and Breadboards
- LED
- Push Buttons
- Switch
- IC
- IC Sockets
- Battery

CODING:

- `// Define pin numbers`
- `const int smokeSensorPin = A0; // Analog pin connected to the smoke sensor`
- `const int buzzerPin = 3; // Digital pin connected to the buzzer`
- `const int ledPin = 4; // Digital pin connected to the LED`
- `// Define threshold value for smoke level`
- `const int smokeThreshold = 300; // Adjust according to your sensor and environment`
- `void setup() {`
- `pinMode(smokeSensorPin, INPUT); // Smoke sensor input`
- `pinMode(buzzerPin, OUTPUT); // Buzzer output`
- `pinMode(ledPin, OUTPUT); // LED output`
- `Serial.begin(9600); // Initialize serial communication for debugging`
- `}`

- `void loop() {`
- `int smokeLevel = analogRead(smokeSensorPin); // Read the smoke sensor value`
- `Serial.print("Smoke Level: ");`
- `Serial.println(smokeLevel); // Print smoke level for debugging`
- `// Check if smoke level is above the threshold`
- `if (smokeLevel > smokeThreshold) {`
- `digitalWrite(buzzerPin, HIGH); // Turn on the buzzer`
- `digitalWrite(ledPin, HIGH); // Turn on the LED`
- `Serial.println("Smoke detected! Alarm activated.");`
- `} else {`
- `digitalWrite(buzzerPin, LOW); // Turn off the buzzer`
- `digitalWrite(ledPin, LOW); // Turn off the LED`
- `}`
- `delay(500); // Wait for half a second before the next reading`
- `}`