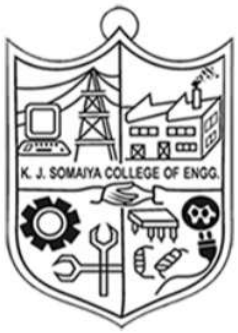
Fire-Fighting Robot Using Arduino

Flame Detection, Approach and Water Suppression

A Project Report
Submitted in partial fulfillment of
the requirements for the degree of
Bachelor of Technology
by

Sahil Ramesh Undale - 16014024034
Navya Mathur - 16014024038
Abhishek Khedekar - 16014024029

Supervisor:
Kirti Sawlani



Department of Electronics and Computer Engineering
K. J. Somaiya School of Engineering
Somaiya Vidyavihar University
Mumbai 400077 (India)

November 2026

CERTIFICATION

This is to certify that the project report entitled
"Fire-Fighting Robot Using Arduino: Flame Detection, Approach and Water Suppression" has been completed and submitted by the students under the guidance and supervision of the undersigned.

Project Guide
Name: Kirti Sawlani
Date: 3/11/2026

External Examiner
Name: KIRAN THOLE
Date: 03/11/26

DECLARATION

We hereby declare that this project report represents our collective work and ideas expressed in our own words. Wherever the ideas, words, or contributions of others have been included, we have appropriately cited and referenced the original sources. We confirm that all sources used in the preparation of this report have been properly and accurately acknowledged.

We further declare that all three members of the project group have adhered to the principles of academic honesty and integrity. We have not misrepresented, fabricated, or falsified any idea, data, fact, or source presented in this submission. We understand that any violation of these declarations may result in disciplinary action by the Institute and may also lead to legal or other action by the original sources or rights holders if proper citation or permission has not been obtained where required.

Member 1
Name: Sahil Ramesh Undale
Date: 11/3/26

Member 2
Name: Navya Mathur
Date: 11/3/26

Member 3
Name: Abhishek Khedekar
Date: 11/3/26

Abstract

Small fires are easiest to put out in their first few minutes, but that is exactly when nobody may be nearby. This project presents a low-cost mobile robot that finds a flame on its own, drives up to it and sprays water on it, completing the full detect–approach–extinguish cycle without human input. The primary objective is to respond to small, localized flames quickly and at their source, using commonly available hobby components.

The robot detects fire with three infrared (IR) flame-sensor modules mounted in a divergent array that faces left, centre and right. Each module is built around a YG1006 phototransistor and an LM393 comparator, giving a digital output that goes LOW when a flame is seen. An Arduino Uno reads the three outputs many times a second and decides which way to move: if the left sensor sees the flame the robot turns left, if the right sensor sees it the robot turns right, and if the centre sensor sees it the robot drives straight.

Locomotion uses differential drive. An L298N dual H-bridge drives the left and right pairs of four BO gear motors at different directions, so the chassis can turn on the spot without any steering mechanism. When the flame is close enough, the robot stops at a safe distance, switches on a 3–6 V mini submersible pump through a transistor driver and sweeps the water nozzle across 180° with an SG90 servo. Two 18650 lithium-ion cells (7.4 V nominal) power the system.

In indoor trials with small open flames, the robot detected flames from 10 cm to 65 cm away, covered approximately 120° in front of the chassis, reacted to a change in flame direction in under 50 ms, stopped about 22 cm from the source and extinguished the flame in 4.5 s to 6.0 s. The main limitation found in testing was false triggering by strong sunlight, which emits in the same near-infrared band as the sensors.

The project demonstrates that inexpensive, widely available components are sufficient for early-stage, localized fire suppression in indoor spaces. It is intended as a demonstrator and a supplement to, not a replacement for, certified fire-detection and suppression systems. Smoke and temperature sensing, flicker analysis and a more efficient motor driver are proposed as improvements.

Table of Contents

Abstract **iv**

1 Introduction **1**
1.1 Background 1
1.2 Problem Statement 1
1.3 Objectives 2
1.4 Scope and Limitations 2
1.5 Overall System Architecture 3

2 Technical Background **4**
2.1 Physics of Flame Detection 4
2.2 IR Flame Sensor Modules 4
2.3 Inverse-Square Law and Detection Range 5
2.4 Differential-Drive Locomotion 5
2.5 H-Bridge Motor Control 5
2.6 Servo Position Control 6

3 System Design and Workflow **7**
3.1 Sense–Decide–Act Workflow 7
3.2 Flame Direction Logic 7
3.3 Approach and Stop Logic 8
3.4 Suppression Response 9

4 Hardware Design and Implementation **10**
4.1 Hardware Components 10
4.2 Arduino Uno 12
4.3 Flame Sensor Array 12
4.4 L298N Motor Driver and BO Motors 13
4.5 Transistor Driver and Mini Submersible Pump 13
4.6 SG90 Servo and Nozzle 13

4.7 Power Subsystem 14
4.8 Hardware Schematic 14

5 Control Logic and Calibration 17
5.1 Pin Assignment 17
5.2 Drive Commands 18
5.3 Sensor Threshold Calibration 19
5.4 Stop Threshold 19
5.5 Servo Sweep Routine 19

6 Experimental Results and Analysis 20
6.1 Evaluation Setup 20
6.2 Measured Performance 20
6.3 Detection Range 21
6.4 Angular Coverage and Reaction Time 21
6.5 Stop Distance and Extinguishing Time 22
6.6 Power Path Analysis 22
6.7 False-Trigger Analysis 22

7 Software Implementation and System Integration 24
7.1 Software Modules 24
7.2 Development and Upload 24
7.3 Control Loop 24
7.4 Operating Sequence 25

8 Conclusion and Future Scope 26
8.1 Conclusion 26
8.2 Limitations 26
8.3 Future Scope 27

References 28

A Key System Parameters 30

B User Manual and Outreach Material 31

Acknowledgements 33

Chapter 1

Introduction

1.1 Background

Fire incidents in homes, shops, laboratories and warehouses cause loss of life and heavy damage to property every year. Most large fires begin as small, localized flames. If such a flame is attacked within the first few minutes, it can often be put out with very little water. The difficulty is that people are not always present when a fire starts.

This project builds a small mobile robot that searches for a flame, moves to it and applies water directly at the source. The robot combines infrared flame sensing, decision making on a microcontroller, differential-drive locomotion, a pump switched by a transistor and a nozzle swept by a servo into one compact unit.

The approach focuses on simple, reactive behaviour that runs entirely on an 8-bit microcontroller. The robot does not build a map or identify the type of fire. It responds to the presence and direction of a flame with a predefined approach and suppression routine.

1.2 Problem Statement

Fixed fire-protection systems are designed for whole rooms or buildings. Ceiling sprinklers react to heat that has risen to the ceiling, by which time a fire has already grown, and they spray an entire zone rather than the actual source. Industrial fire-fighting robots and networked smart-sprinkler systems exist, but they are expensive and too large for homes, labs or small commercial spaces.

The problem addressed in this work is therefore the response gap between ignition and human intervention. A small mobile robot that can find the exact point of ignition, drive to it and apply water directly would reduce both the response time and the risk to people who would otherwise have to approach the flame.

1.3 Objectives

The main objective of this project is to develop a low-cost robot that detects a small flame, moves toward it and extinguishes it with water, with no human input during the detect–approach–extinguish cycle.

The specific objectives are:

1. **Flame Detection:** To detect a flame in front of the robot using infrared flame sensors and to determine whether it lies to the left, centre or right.

2. **Steering Toward the Source:** To turn the robot toward the detected flame and drive toward it using differential drive.

3. **Safe Stand-off Distance:** To stop the robot at a safe distance from the flame so that the chassis, wiring and sensors are protected from radiant heat.

4. **Water Suppression:** To spray water on the flame through a transistor-switched pump and to sweep the nozzle across the flame area using a servo.

5. **Re-check and Repeat:** To check the sensors again after spraying and repeat the suppression routine if the flame is still present.

6. **Integrated Hardware Control:** To integrate the Arduino Uno, flame sensors, L298N motor driver, BO motors, transistor driver, pump, servo and battery pack into a single coordinated system.

7. **Affordable and Practical Solution:** To build a compact, low-cost prototype using commonly available hobby components.

Note: The robot is intended as a demonstrator for early-stage, small-flame suppression. It does not replace smoke alarms, fire extinguishers, certified suppression systems or human supervision.

1.4 Scope and Limitations

The project covers the design, construction and indoor testing of a four-wheeled robot that detects small open flames, such as a candle or lighter flame, using three IR flame-sensor modules. An Arduino Uno runs a reactive control loop that steers the robot through an L298N motor driver and four BO motors. When the flame is close enough, the robot stops and sprays water from an on-board tank through a mini submersible pump and an SG90-swept nozzle. The design, control logic, power path and measured performance are documented in this report.

The robot's performance depends on flame size, sensor calibration, ambient lighting and the surface it drives on. Strong sunlight and hot lamps emit in the same near-infrared band as the sensors and can cause false detections. The robot has no obstacle avoidance and carries

only a small volume of water. It is suitable only for small flames that can be put out with water; it must never be used on electrical, oil, grease or gas fires. The prototype is a student demonstrator and is not a certified fire-safety device.

1.5 Overall System Architecture

The robot follows a sense–decide–act loop. The three flame sensors are the inputs, the Arduino Uno makes the decision, and two actuator chains carry it out: locomotion (L298N and four BO motors) and suppression (transistor driver, pump and servo). Two 18650 cells, switched by a rocker switch, supply the motor driver and the Arduino, and the Arduino's on-board 5 V regulator powers the sensors and the servo.

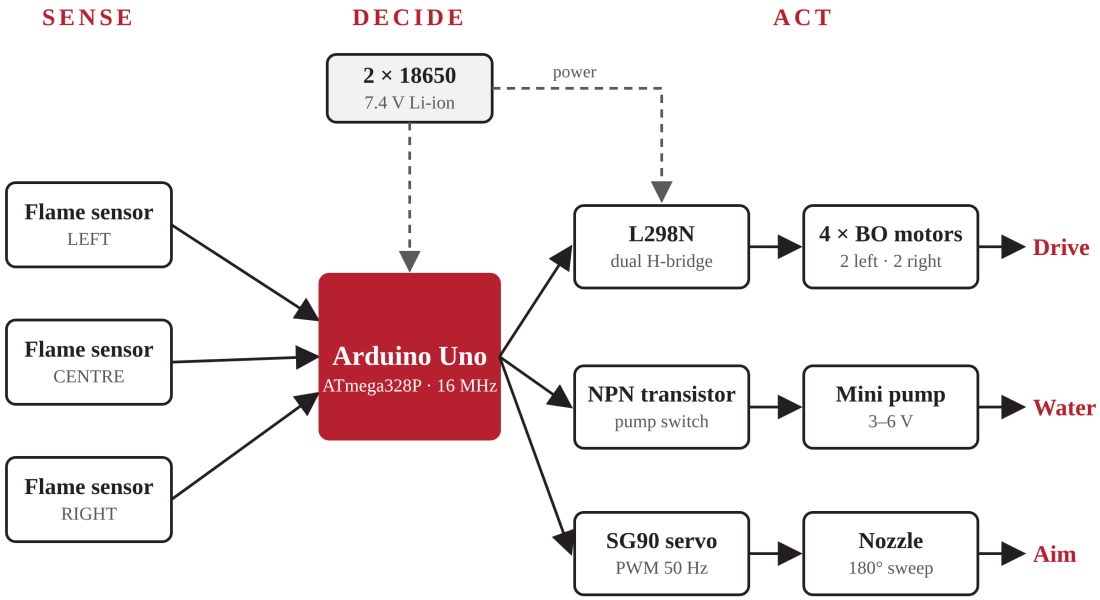


Figure 1.1: System Architecture

The architecture separates sensing, decision and actuation into clearly defined stages. The sensor modules perform their own thresholding in hardware, so the Arduino only has to read digital pins. This keeps the control loop short and allows the robot to react quickly when the flame moves relative to the chassis.

Chapter 2

Technical Background

2.1 Physics of Flame Detection

A flame emits radiation from the ultraviolet to the long-wave infrared. Commercial flame detectors exploit different bands. UV detectors respond within milliseconds but are falsely triggered by welding arcs and lightning. Wideband IR detectors look for the strong 4.3 µm emission of hot CO_2, which cold atmospheric CO_2 filters out of sunlight [1].

Near-infrared detectors working between 0.7 µm and 1.1 µm are cheap, but they share this band with sunlight and incandescent lamps. They therefore need thresholding or extra checks such as flicker detection, since real flames flicker at roughly 1 Hz to 20 Hz [1]. The sensors used in this project fall in this low-cost near-infrared category.

2.2 IR Flame Sensor Modules

The hobby flame-sensor modules used here are built around the YG1006 NPN silicon phototransistor. Its black epoxy package makes it sensitive mainly to infrared, with a spectral range of 760 nm to 1100 nm and a viewing angle of about 60° [2, 3]. An LM393 comparator with an adjustable potentiometer compares the phototransistor signal with a reference and produces a digital output (D0) that goes LOW when a flame is detected. An indicator LED on the module lights at the same time.

This hardware threshold means that the microcontroller only reads a digital level. The decision is fast, and noise in the analogue signal is largely rejected before it reaches the Arduino. A lighter flame is typically detected at up to about 80 cm [2], but the range depends strongly on flame size and on the potentiometer setting.

2.3 Inverse-Square Law and Detection Range

The radiation received from a small flame falls with distance roughly as an inverse square [1]. If E is the received irradiance and D is the distance from the flame,

$$E \propto \frac{1}{D^2}, \tag{2.1}$$

so doubling the distance cuts the signal to about one quarter. Treating the flame as a source of radius R and normalizing to the signal at 10 cm gives the model used later in this report:

$$\frac{E}{E_{10}} = \frac{10^2 + R^2}{D^2 + R^2}, \qquad D \text{ and } R \text{ in cm.} \tag{2.2}$$

Two practical consequences follow. First, the potentiometer setting strongly affects detection range. Second, because the IR level rises steeply as the robot approaches, the same threshold can be used as a crude distance sensor to decide when to stop.

2.4 Differential-Drive Locomotion

Differential (two-sided) drive is the most common configuration for small robots because steering needs no extra mechanism [6, 7]. With left and right wheel ground speeds v_L and v_R and track width W, the robot's forward speed v and turning rate ω are

$$v = \frac{v_R + v_L}{2}, \qquad \omega = \frac{v_R - v_L}{W}. \tag{2.3}$$

Equal wheel speeds give straight motion ($\omega = 0$). Opposite wheel speeds give $v = 0$ and rotation on the spot, which is how the robot re-aims at an off-axis flame. Unequal speeds produce an arc.

With four wheels and no casters, the chassis actually skid-steers, so some wheel slip occurs during turns. Equation (2.3) is therefore an idealization, but it correctly predicts the direction of each manoeuvre.

2.5 H-Bridge Motor Control

An H-bridge is four switches arranged around a motor. Closing one diagonal pair drives current through the motor in one direction, and closing the other pair reverses it. Each side of the robot can therefore go forward, go backward or stop. The L298N module contains two H-bridges, accepts a motor supply of 5 V to 35 V and supplies up to 2 A continuous per channel [9].

2.6 Servo Position Control

A hobby servo is positioned by a pulse repeated every 20 ms (50 Hz). The pulse width, about 1 ms to 2 ms, sets the output angle, as shown in Figure 2.1. Exact end-points vary between SG90 units [10].

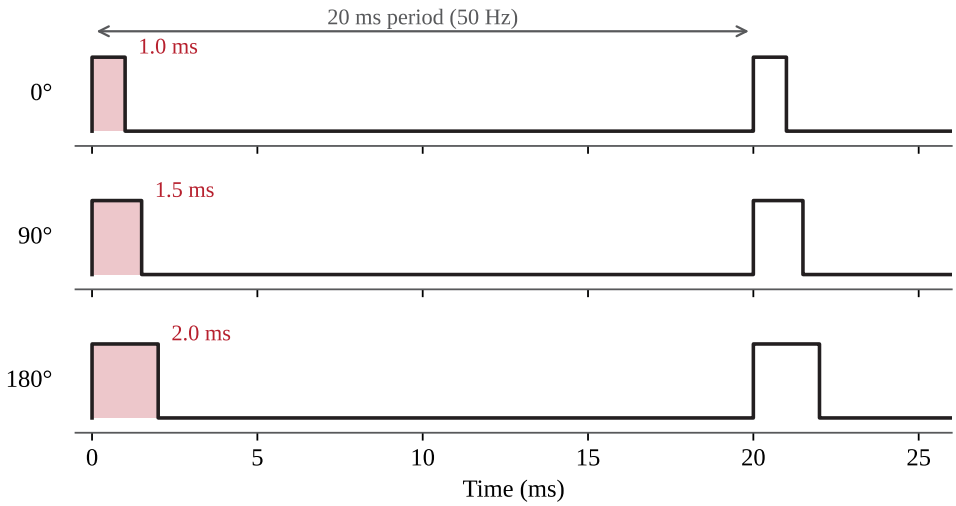


Figure 2.1: Servo control signal. The pulse width sets the nozzle angle; the period stays at 20 ms.

Chapter 3

System Design and Workflow

3.1 Sense–Decide–Act Workflow

In every loop, the Arduino reads the three flame-sensor outputs, decides on one action and applies it to the motors or to the suppression hardware. The logic is purely reactive: it does not build a map, it simply turns toward whichever side senses IR and advances when the flame is centred.

Because the sensor outputs are digital and the loop contains no long delays, the robot re-orients almost immediately when the flame moves relative to the chassis. The loop starts again after every action, so the robot keeps correcting its heading as it approaches.

3.2 Flame Direction Logic

Each sensor module sees only about 60° [3], so three modules are mounted in a divergent array: one facing forward and one angled to each side. With adjacent cones overlapping slightly, the array covers approximately 120° to 150° in front of the robot with no central blind zone (Figure 3.1). The pattern of active sensors tells the controller which way the flame lies.

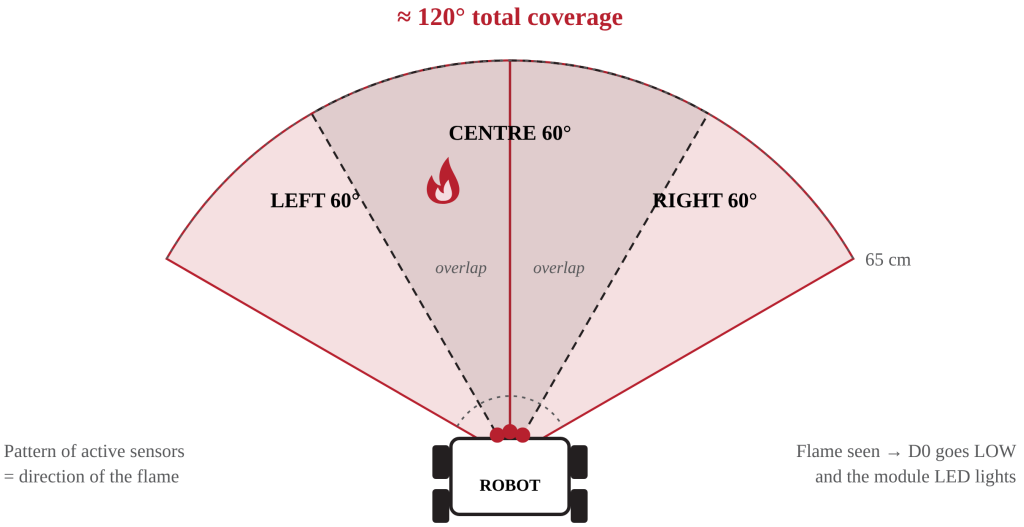


Figure 3.1: Divergent three-sensor array and its angular coverage.

The overlap is important. When the flame lies between two cones, both sensors fire and the robot continues turning smoothly instead of losing the target. Table 3.1 lists the action taken for each sensor pattern, following the order of checks in the firmware flowchart (Figure 3.2).

Table 3.1: Sensor pattern and resulting action (• = sensor sees flame).

Left	Centre	Right	Action
			Stop (idle)
•			Turn left on the spot
•	•		Turn left on the spot
	•		Move forward (far) / stop and spray (close)
	•	•	Turn right on the spot
		•	Turn right on the spot

3.3 Approach and Stop Logic

When only the centre sensor sees the flame, the robot drives forward. As it approaches, the received IR rises steeply, following Equation (2.2). When the centre sensor reports the flame strongly enough, that is, when the stop threshold is reached, the robot halts. In testing, this happened at about 22 cm from the flame.

The stand-off distance protects the chassis, jumper wires and sensor housings from radiant heat while staying within the reach of the low-pressure pump. Because the IR level rises steeply at short range, a fixed comparator setting corresponds to a fairly consistent distance for a given flame size.

3.4 Suppression Response

After stopping, the Arduino switches the pump on through the transistor driver and steps the servo-mounted nozzle across 0° to 180°. It then checks the sensors again and repeats the spray if the flame is still present. When no sensor sees a flame, the robot stops and waits.

The robot never stops perfectly centred on the flame. Sweeping the nozzle trades some water per point for coverage of both the flame and the area around it, which proved more reliable than a fixed jet.

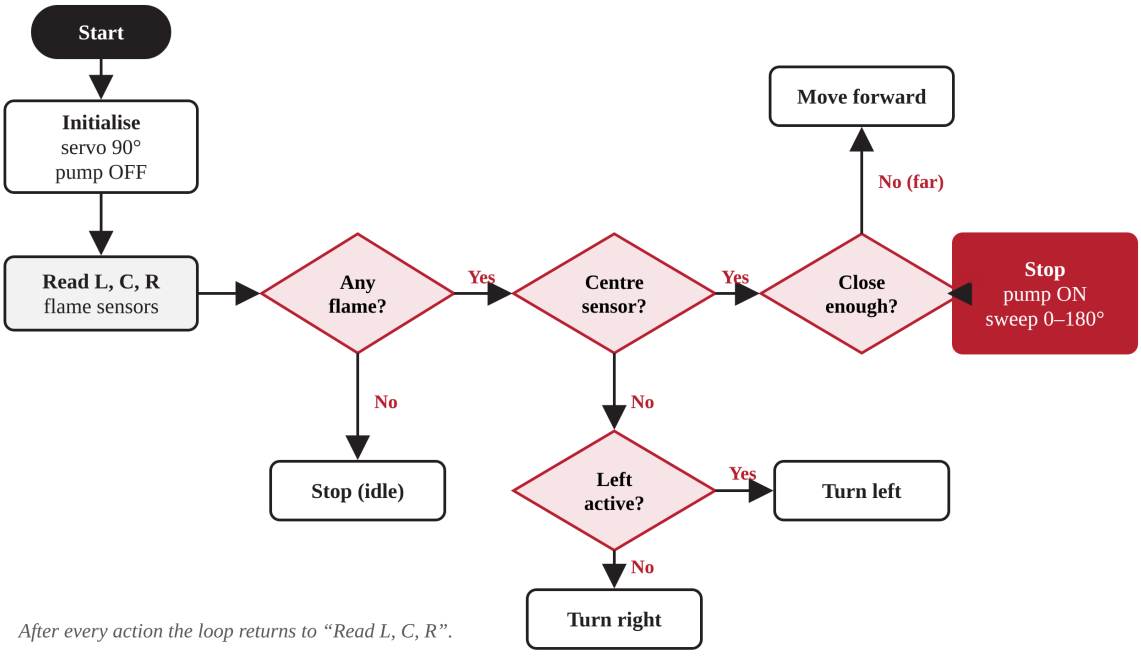


Figure 3.2: Work Flow

Chapter 4

Hardware Design and Implementation

4.1 Hardware Components

The prototype consists of an Arduino Uno, three IR flame-sensor modules, an L298N motor-driver module, four BO gear motors with wheels, an NPN power transistor with a base resistor, flyback diode and capacitor, a 3–6 V mini submersible pump with a water tank and tube, an SG90 micro servo carrying the nozzle, and two 18650 lithium-ion cells in a holder with a rocker power switch. The components are mounted on a flat deck above the motors.

Table 4.1: Hardware components.

Component	Qty	Role
Arduino Uno (ATmega328P)	1	Reads sensors, runs control logic, drives outputs
IR flame sensor (YG1006 + LM393)	3	Detects flame left, centre and right
L298N dual H-bridge	1	Drives the two motor pairs; speed set by PWM on ENA, ENB
BO gear motor + wheel	4	Differential-drive locomotion
NPN power transistor (TO-220)	1	Switches the pump; driven from A5 through a base resistor
Flyback diode, capacitor	1 each	Protect the transistor and suppress pump-motor noise
Mini submersible pump (3–6 V)	1	Pumps water from the tank to the nozzle
SG90 micro servo	1	Sweeps the nozzle across 180°
18650 Li-ion cell	2	7.4 V nominal power supply
Rocker switch	1	Main power on/off
Water tank, tube, nozzle, chassis	–	Mechanical assembly

The components are arranged into sensing, processing, actuation and power groups. The sensor modules and the Arduino form the sensing and decision path, the L298N and motors form the locomotion path, and the transistor driver, pump and servo form the suppression path.

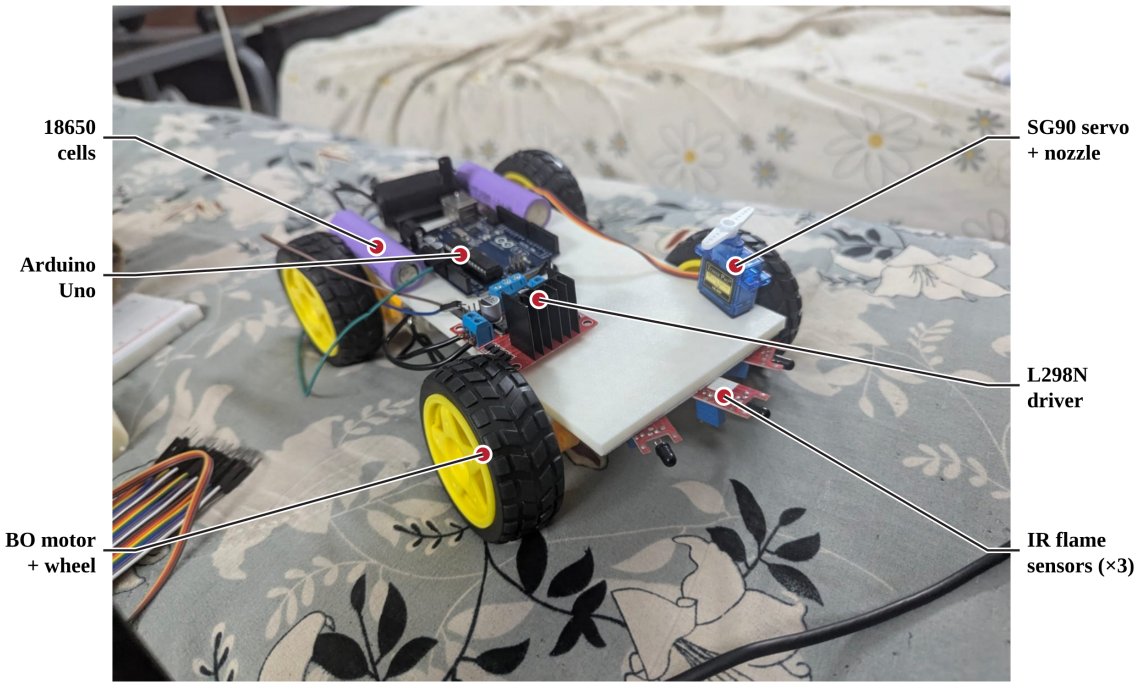


Figure 4.1: Assembled prototype with its main components labelled.

4.2 Arduino Uno

The Arduino Uno is based on the ATmega328P running at 16 MHz, with 14 digital I/O pins, six of which provide hardware PWM, and six analog inputs [8]. It reads the three sensor outputs, runs the decision logic, generates the direction and PWM speed signals for the L298N, generates the 50 Hz servo signal and switches the pump transistor.

An Arduino pin can source only about 40 mA [8]. All loads that need more current, the motors, pump and servo, are therefore driven through the L298N, the transistor driver or a supply rail rather than directly from a pin.

4.3 Flame Sensor Array

The three flame-sensor modules are mounted at the front of the deck, one facing forward and one angled to each side. Each module's D0 output is connected to one of the Arduino's analog pins A0 to A2, which are read as digital inputs, and each module is powered from the Arduino's 5 V pin. The on-board potentiometer of each module sets its detection threshold and is adjusted during calibration (Section 5.3).

The sensors should be mounted rigidly so that their viewing directions do not change while the robot moves. A change in angle shifts the coverage cones and can open a blind zone straight ahead.

4.4 L298N Motor Driver and BO Motors

The L298N receives four direction signals (IN1 to IN4) and two enable signals (ENA and ENB) from the Arduino. Two BO motors are wired in parallel on each output channel, so the two sides of the robot are controlled independently. The direction inputs select each manoeuvre (Table 5.2), and PWM on the enable inputs, from pins D10 and D5, sets the speed of each side.

An important practical detail is that the L298N uses bipolar transistors, so roughly 1.8 V to 3 V is lost across each bridge as heat [9]. With two 18650 cells (7.4 V nominal, 8.4 V fully charged), the motors therefore receive only about 4.5 V to 6.5 V. This is adequate for 6 V BO motors, but the robot slows noticeably as the battery discharges.

4.5 Transistor Driver and Mini Submersible Pump

The mini submersible pump runs from 3 V to 6 V, draws 130 mA to 220 mA and delivers 80 L/h to 120 L/h at zero head [11]. This current is far above what an Arduino pin can source, so the pump is switched by an NPN power transistor. Pin A5 drives the transistor's base through a resistor; the pump's positive terminal is connected to the switched battery supply and its negative terminal to the transistor's collector, with the emitter to ground.

A flyback diode across the pump gives the motor's stored inductive energy a safe path when the transistor turns off, protecting the transistor from a voltage spike. A small capacitor across the pump suppresses electrical noise from the motor brushes.

The flow rate corresponds to about 22 mL/s to 33 mL/s. A 5 s spray therefore delivers at most roughly 110 mL to 165 mL, which is enough for a candle- or paper-sized flame. In practice the flow is lower because the water is pushed up a tube to the nozzle.

4.6 SG90 Servo and Nozzle

The nozzle is carried by an SG90 micro servo. The SG90 is rated at 1.8 kg cm stall torque and 0.1 s per 60° at 4.8 V [10], so an unloaded 180° sweep takes about 0.3 s. The firmware deliberately steps the angle slowly so that each part of the arc receives water.

Sweeping the nozzle widens the wetted area to about 35 cm and compensates for the robot stopping slightly off-centre.

4.7 Power Subsystem

Two 18650 lithium-ion cells in series supply 7.4 V nominal (8.4 V full, about 6 V near empty). A rocker switch on the battery's positive lead turns the whole robot on and off. After the switch, the battery feeds the L298N motor input (+12 V terminal), the Arduino's Vin pin and the pump supply. The Arduino's on-board regulator produces 5 V, which powers the three flame sensors and the servo.

Motor start-up and pump switching cause short voltage dips on the shared battery. A common ground between all modules is essential for the logic signals to be read correctly.

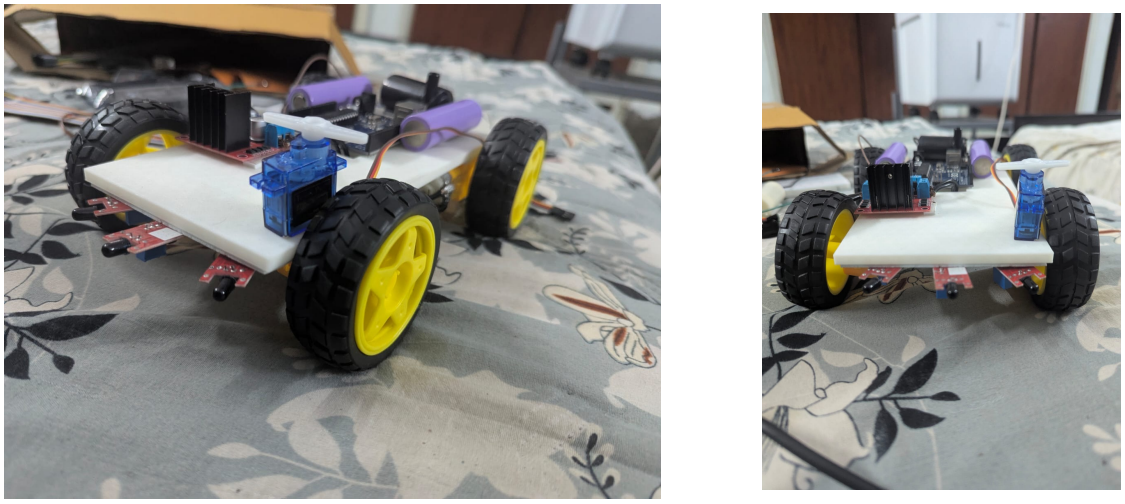


Figure 4.2: Prototype: side view (left) and front view showing the servo-mounted nozzle (right).

4.8 Hardware Schematic

Figure 4.3 shows the wiring diagram of the prototype, and Figure 4.4 redraws the same connections as a net-label schematic, in which pins carrying the same label are connected. The pin numbers follow Table 5.1.

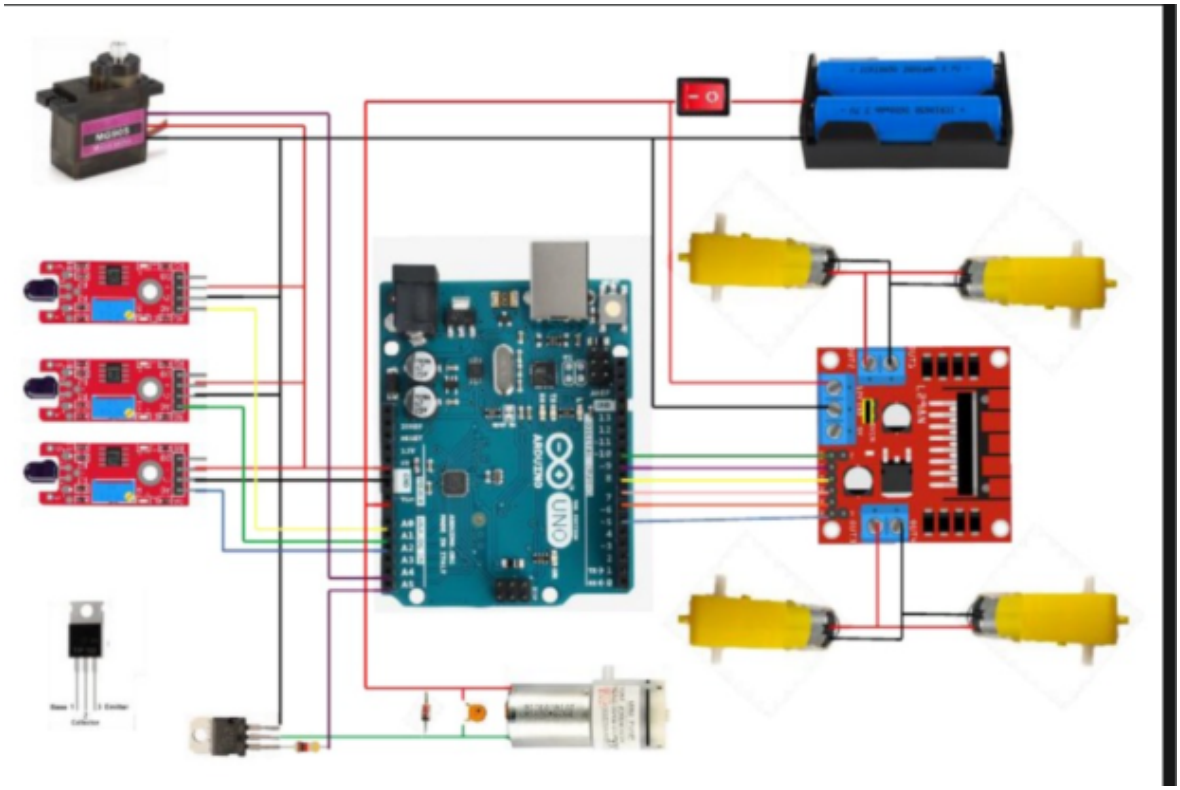
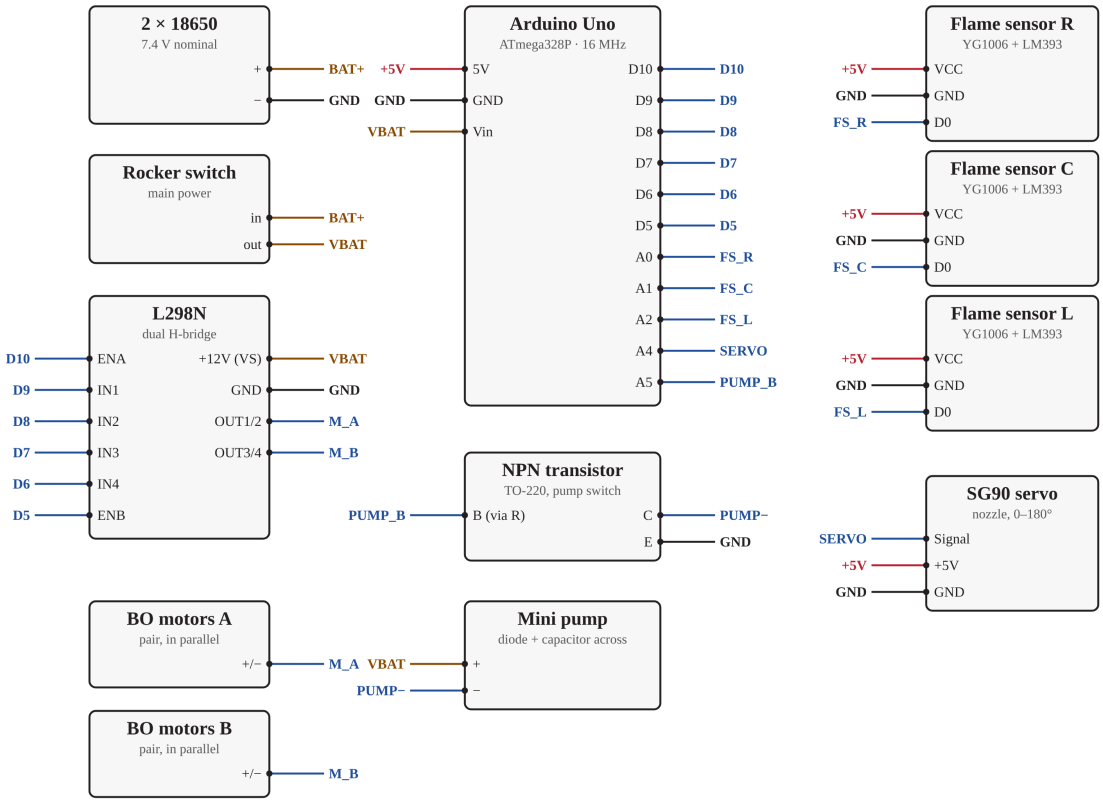


Figure 4.3: Wiring diagram



Pins carrying the same net label are connected. Red: +5 V from the Arduino; brown: battery (switched as VBAT); black: common ground.

Figure 4.4: Schematic

Chapter 5

Control Logic and Calibration

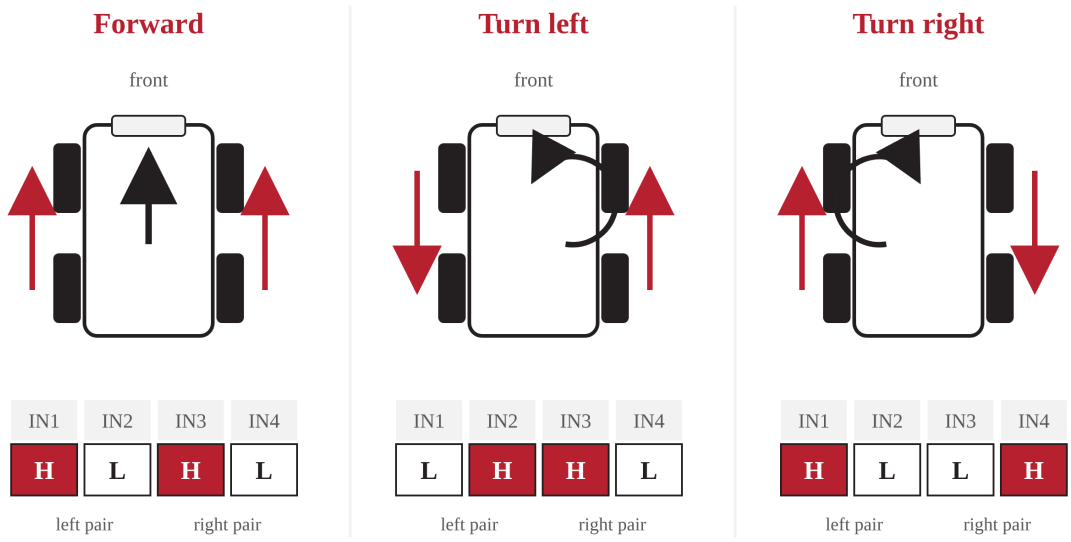


Figure 5.1: Drive modes and the corresponding L298N input levels.

5.1 Pin Assignment

Table 5.1 gives the pin assignment of the prototype. The three sensor outputs are connected to analog pins A0 to A2, which are read as digital inputs that go LOW when a flame is seen. All other signals are outputs from the Arduino. Pins D10 and D5 provide hardware PWM for the motor enables.

Table 5.1: Arduino Uno pin assignment.

Pin	Connected to	Signal
A0	Right flame sensor D0	Input, LOW = flame
A1	Centre flame sensor D0	Input, LOW = flame
A2	Left flame sensor D0	Input, LOW = flame
D10	L298N ENA	PWM, motor A speed
D9, D8	L298N IN1, IN2	Motor A direction
D7, D6	L298N IN3, IN4	Motor B direction
D5	L298N ENB	PWM, motor B speed
A4	Servo signal	50 Hz servo pulses
A5	Pump transistor base (via resistor)	Digital output
5 V, GND	Flame sensors, servo	5 V supply
Vin, GND	Battery via rocker switch	7.4 V input

Keeping all pin numbers in one place in the firmware, as named constants, makes it easy to check the wiring against the code.

5.2 Drive Commands

Each manoeuvre corresponds to one combination of the four L298N direction inputs (Table 5.2 and Figure 5.1), with both enable inputs driven by PWM. Driving both sides forward gives straight motion; driving one side backward and the other forward rotates the robot on the spot.

Table 5.2: Drive commands sent to the L298N.

Action	IN1	IN2	IN3	IN4
Forward	H	L	H	L
Turn left (spot)	L	H	H	L
Turn right (spot)	H	L	L	H
Stop	L	L	L	L

Note: swap the H/L pair of a side if its motors are wired in reverse.

5.3 Sensor Threshold Calibration

Each flame sensor is calibrated with its potentiometer. A lit candle is placed on a metal tray about 50 cm in front of one sensor, at the same height. The potentiometer is turned slowly until the module LED just turns on. The flame is then moved out of view, and the LED must turn off; if it stays on, the potentiometer is turned back slightly. The procedure is repeated for the left, centre and right sensors.

The setting is a trade-off. A very sensitive setting increases detection range but also causes false detections from room lighting and sunlight; a less sensitive setting reduces false detections but shortens the range. Calibration should be done in the room where the robot will be used. In this project, the potentiometers were set conservatively to reduce false triggering from room lighting.

5.4 Stop Threshold

The robot uses the centre sensor's response as its stop condition. Because the received IR rises steeply below about 20 cm (Figure 6.1), the stop threshold corresponds to a fairly repeatable distance for a given flame size. In testing, the robot stopped about 22 cm from the flame on average.

The stop distance depends on flame size: a larger flame is seen more strongly at the same distance and therefore triggers the stop earlier. The threshold should be rechecked whenever the sensor potentiometers are adjusted.

5.5 Servo Sweep Routine

At start-up the servo is set to 90°, pointing the nozzle straight ahead, and the pump is off. During suppression, the pump is switched on and the servo angle is stepped from 0° to 180°. The steps are deliberately slow compared with the servo's 0.3 s full-travel time, so that each part of the arc receives water.

After the sweep, the pump is switched off and the sensors are read again. If any sensor still reports a flame, the approach and suppression routine repeats.

Chapter 6

Experimental Results and Analysis

6.1 Evaluation Setup

The prototype was tested indoors with small open-flame targets (candle and lighter) placed at different distances and angles in front of the robot.

A trial is counted as successful when the robot reaches the flame, stops and extinguishes it without human input. Number of trials: _____.

6.2 Measured Performance

Table 6.1 summarizes the observed performance.

Table 6.1: Measured performance (indoor trials).

Parameter	Observed value
Reliable detection range	10–65 cm
Angular coverage (3 sensors)	≈120°
Drive reaction to heading change	<50 ms
Stop distance from flame	≈22 cm (average)
Nozzle sweep	180°, water spread ≈35 cm
Time to extinguish	4.5–6.0 s
Number of trials	_____
Successful extinguishments	_____

The robot completed the full detect–approach–extinguish cycle without human input, and the swept spray extinguished small flames within a few seconds.

6.3 Detection Range

The measured detection range of up to 65 cm is below the 80 cm to 100 cm quoted for these modules [2]. This is expected: the potentiometers were set conservatively to reduce false triggering from room lighting, and a small candle flame emits less infrared than a lighter flame.

Figure 6.1 relates the observed operating window to the sensor physics. It plots the relative IR signal predicted by Equation (2.2) for a small flame. The signal rises very steeply below about 20 cm, which is why the stop threshold is repeatable, and falls to a few percent of its close-range value by 65 cm, which is where reliable detection ended in the trials.

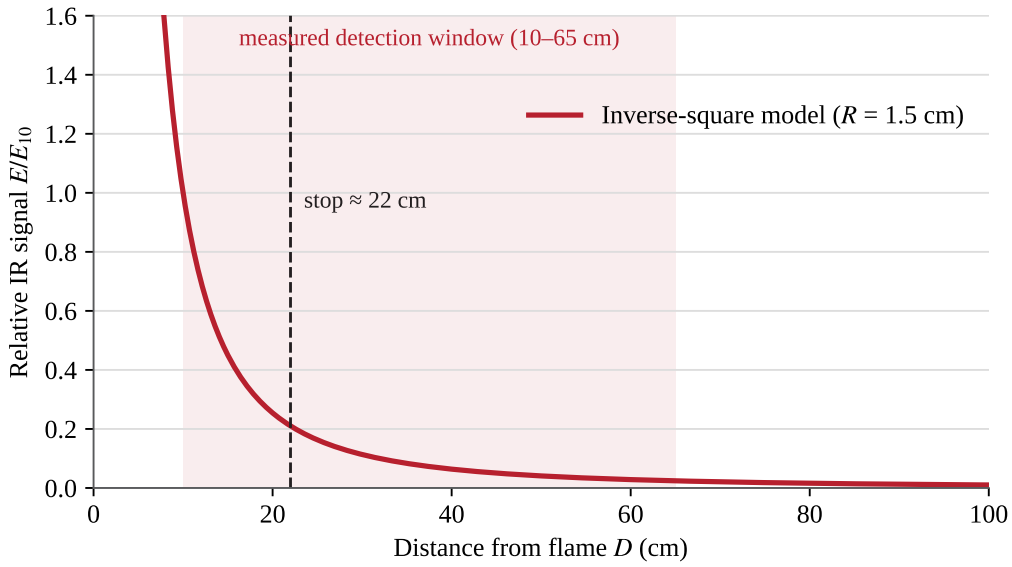


Figure 6.1: Relative near-IR signal versus distance from the inverse-square model (normalized to $D = 10$ cm; flame radius assumed 1.5 cm). Shaded: measured detection window; dashed: measured stop distance.

6.4 Angular Coverage and Reaction Time

The measured coverage of about 120° is consistent with three 60° sensors placed with slight overlap. The pattern of active sensors encodes the direction of the flame, and overlapping the cones avoids a dead zone straight ahead. In single-sensor designs, such a dead zone causes the robot to oscillate or overshoot.

The robot re-oriented almost immediately when a flame moved off-axis, with a drive reaction under 50 ms. This is because the sensor outputs are digital and the control loop contains no long delays.

6.5 Stop Distance and Extinguishing Time

The stop threshold reliably halted the robot about 22 cm from the flame. At this distance, radiant heat on the chassis and wiring is limited, and the flame is still within reach of the low-pressure pump.

The swept spray extinguished the targets in 4.5 s to 6.0 s. At the pump's rated flow, this corresponds to at most roughly 100 mL to 200 mL of water per flame; the actual volume is lower because the water is lifted to the nozzle through a tube.

6.6 Power Path Analysis

The L298N's bipolar output stage loses about 1.8 V to 3 V as heat [9]. With the battery between 8.4 V (full) and about 6 V (near empty), the motor voltage therefore varies widely, and the robot slows noticeably as the cells discharge. This affects approach speed but not the direction logic, which depends only on the sensor pattern.

The pump's current of 130 mA to 220 mA is switched by the transistor from the battery side, so pin A5 supplies only the small base current. Motor start-up and pump switching cause short voltage dips on the shared battery, which is why all modules share a common ground and why the flyback diode and noise capacitor are fitted across the pump.

6.7 False-Trigger Analysis

The main problem observed in testing was false detection under strong sunlight. This is a direct consequence of the sensor physics: sunlight and hot lamps contain plenty of radiation in the same 760 nm to 1100 nm band that the YG1006 responds to (Figure 6.2). Industrial detectors avoid this by working near 4.3 µm or by combining several bands [1].

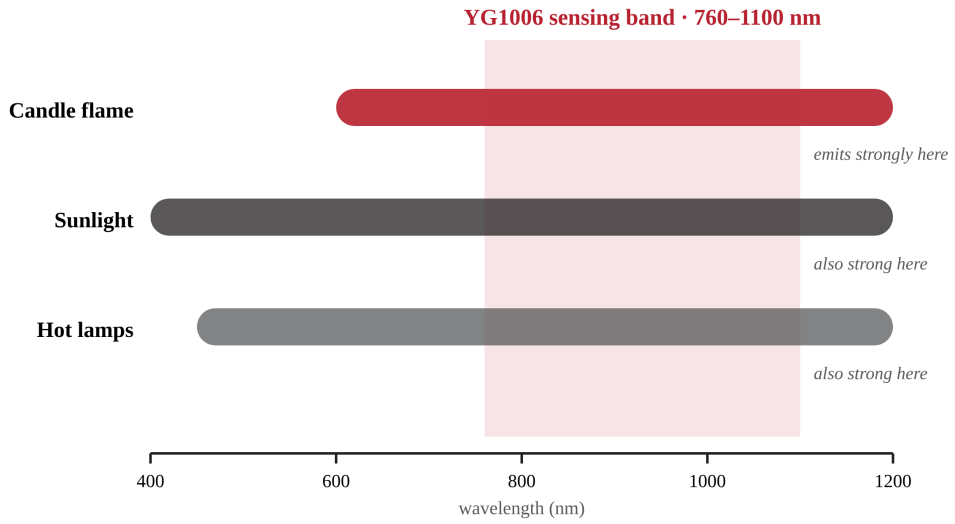


Figure 6.2: Candle flame, sunlight and hot lamps all emit within the YG1006 sensing band.

Raising the comparator threshold reduced false positives but also shortened the detection range. The robot should therefore be calibrated and used indoors, away from windows and hot lamps. A second, independent signal, such as smoke, temperature or flame flicker, is the most direct way to remove this limitation.

Chapter 7

Software Implementation and System Integration

7.1 Software Modules

The firmware can be divided into sensor reading, decision logic, motor control, pump control and servo sweep modules. These modules coordinate the transition between searching, approaching and suppressing.

Separating the firmware into small functions makes each one easy to test independently. For example, the motor directions can be verified with the wheels off the ground before the sensors are connected, and the pump and servo can be tested with explicit calls before full integration.

7.2 Development and Upload

The firmware is written in the Arduino language (C/C++) and uploaded to the Arduino Uno over USB. The servo signal is a 50 Hz pulse-width signal on pin A4, the motor speeds are set with PWM on D10 and D5, and the remaining pins are used as plain digital inputs and outputs. Analog pins A0 to A2, A4 and A5 are used here as digital pins.

During development, the serial monitor can be used to print the three sensor states and the chosen action in each loop. This makes it easy to confirm the sensor pattern table (Table 3.1) before the robot is allowed to drive.

7.3 Control Loop

Listing 7.1 summarizes the control loop as pseudocode. It follows the flowchart in Figure 3.2.

Listing 7.1: Control loop (pseudocode).

```
setup:
    configure sensor pins as inputs, motor/pump pins as outputs
    servo -> 90 degrees, pump OFF, motors STOP

loop:
    L, C, R = read sensors            (LOW = flame seen)
    if no sensor sees a flame:
        stop motors                   (idle)
    else if centre only:
        if stop threshold reached:
            stop motors
            pump ON
            sweep servo 0 -> 180 degrees in slow steps
            pump OFF, servo -> 90 degrees
        else:
            move forward
    else if left sees flame:
        turn left on the spot
    else:
        turn right on the spot
    repeat
```

7.4 Operating Sequence

At start-up, the robot initializes its pins, centres the nozzle and switches the pump and motors off. It then repeatedly reads the sensors and applies one action per loop. When the stop condition is met, it runs the suppression routine and then returns to reading the sensors, so that a flame that is still burning is sprayed again.

A known safe state is important. On start-up, or whenever no flame is seen, the motors and pump are off. The pump is switched off at the end of every sweep, so it cannot be left running if the flame goes out during spraying.

Chapter 8

Conclusion and Future Scope

8.1 Conclusion

This project designed, built and tested a low-cost fire-fighting robot based on the Arduino Uno. A three-sensor near-infrared array gives the robot both detection and direction information; a differential-drive chassis driven by an L298N H-bridge turns it toward the flame; a distance threshold stops it at a safe stand-off; and a transistor-switched pump with a servo-swept nozzle extinguishes the fire.

Indoor trials confirmed the complete detect–approach–extinguish cycle without human input, with detection up to 65 cm, about 120° of coverage, a stop distance of about 22 cm and extinguishing times of 4.5 s to 6.0 s. The results show that inexpensive, widely available components are sufficient for early-stage, localized fire suppression in indoor spaces. The prototype is a demonstrator and a supplement to, not a replacement for, certified fire-detection and suppression systems.

8.2 Limitations

The main limitation is false detection under strong sunlight and hot lamps, which emit in the same near-infrared band as the sensors. Raising the threshold reduces false positives but also shortens the detection range. Other limitations are the voltage lost in the L298N, which wastes battery energy as heat and slows the robot as the cells discharge, the absence of obstacle avoidance, and the small water tank.

The evaluation was carried out indoors with small candle and lighter flames. It does not establish performance for larger fires, other fuels, outdoor conditions or cluttered rooms. Water is unsuitable for electrical, oil, grease and gas fires, and the robot must never be used on them.

8.3 Future Scope

Future work includes adding a smoke (MQ-2) or temperature sensor so that two independent signals must agree before spraying, and sampling the sensors' analog output fast enough to detect the 1 Hz to 20 Hz flicker that is unique to flames [1]. Replacing the L298N with a MOSFET driver such as the TB6612FNG would cut the 2 V to 3 V driver loss and extend running time.

Ultrasonic obstacle avoidance and a larger tank would move the design toward real rooms. Longer trials with different flame sizes, distances, angles and lighting conditions would quantify the false-trigger rate and the success rate more precisely.

References

[1] "Flame detector," Wikipedia. [Online]. Available: https://en.wikipedia.org/wiki/Flame_detector

[2] Robu.in, "Flame sensor module (YG1006, LM393)," product documentation. [Online]. Available: https://robu.in/product/flame-sensor-module/

[3] MakerLab Electronics, "Flame sensor module — specifications." [Online]. Available: https://www.makerlab-electronics.com/products/flame-sensor-module

[4] K. Chenchireddy *et al.*, "Design of a prototype firefighting robot based on an Arduino microcontroller," *IAES Int. J. Robotics and Automation (IJRA)*. [Online]. Available: https://ijra.iaescore.com/index.php/IJRA/article/download/20891/13132

[5] University of Central Florida Senior Design Group 18, "Automatic fire fighting robot," conference paper, 2022. [Online]. Available: https://ece.ucf.edu/seniordesign/su2022fa2022/g18/8_Page_Conference_Paper.pdf

[6] "Differential wheeled robot," Wikipedia. [Online]. Available: https://en.wikipedia.org/wiki/Differential_wheeled_robot

[7] G. Klančar, A. Zdešar, S. Blažič, and I. Škrjanc, *Wheeled Mobile Robotics: From Fundamentals Towards Autonomous Systems*. Oxford, U.K.: Butterworth-Heinemann, 2017.

[8] Arduino, "Arduino Uno Rev3 — technical specifications." [Online]. Available: https://docs.arduino.cc/hardware/uno-rev3/

[9] STMicroelectronics, "L298: Dual full-bridge driver," datasheet; module notes at https://docs.circuit.rocks/motors/l298n/

[10] Tower Pro, "SG90 9 g micro servo," datasheet. [Online]. Available: https://tams.informatik.uni-hamburg.de/lehre/2023ss/vorlesung/es/doc/sg90.pdf

[11] Electrokit, “Submersible water pump 3–6 V 120 l/h,” product specifications. [Online]. Available: https://www.electrokit.com/en/submersible-waterpump-3-6v-120l/h

Appendix A

Key System Parameters

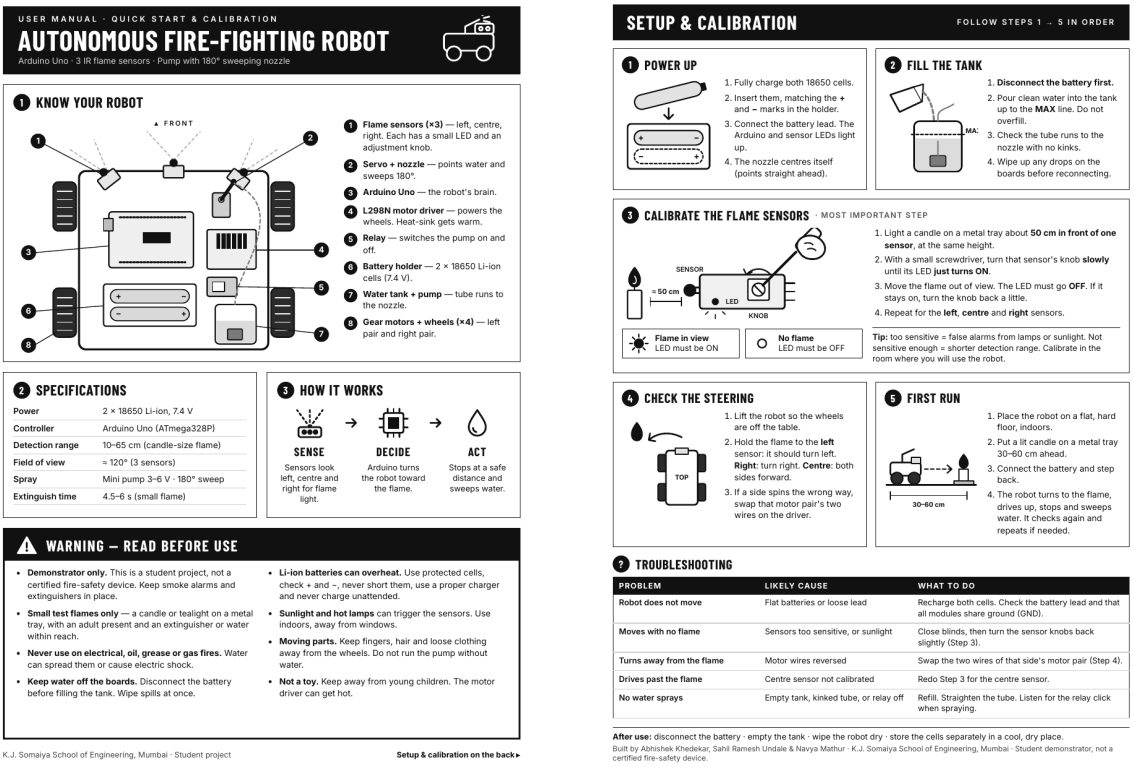
Table A.1: Hardware and control configuration.

Parameter	Configuration
Controller	Arduino Uno (ATmega328P, 16 MHz)
Flame sensors	3 × YG1006 + LM393 on A0–A2
Sensor band	760–1100 nm
Viewing angle per sensor	≈60°
Array coverage	≈120°
Detection range (calibrated)	10–65 cm
Motor driver	L298N; IN1–IN4 on D9–D6, ENA/ENB PWM on D10/D5
Drive motors	4 × BO gear motors (2 per side)
Stop distance	≈22 cm
Pump	3–6 V submersible, 130–220 mA, 80–120 L/h
Pump switching	NPN transistor, base driven from A5
Nozzle servo	SG90 on A4, 50 Hz, 0–180° sweep
Power	2 × 18650 Li-ion, 7.4 V nominal
Logic supply	Arduino 5 V (Vin from battery)
Power switch	Rocker switch on battery positive
Extinguishing time	4.5–6.0 s (small flame)

Appendix B

User Manual and Outreach Material

A two-page user manual was prepared for the prototype. It covers the parts of the robot, specifications, safety warnings, set-up, sensor calibration, a first test run and troubleshooting. A comic-style poster was also produced to present the project.



USER MANUAL · QUICK START & CALIBRATION

AUTONOMOUS FIRE-FIGHTING ROBOT

Arduino Uno · 3 IR flame sensors · Pump with 180° sweeping nozzle

1 KNOW YOUR ROBOT

▲ FRONT

1. **Flame sensors (×3)** — left, centre, right. Each has a small LED and an adjustment knob.
2. **Servo + nozzle** — points water and sweeps 180°.
3. **Arduino Uno** — the robot's brain.
4. **L298N motor driver** — powers the wheels. Heat-sink gets warm.
5. **Relay** — switches the pump on and off.
6. **Battery holder** — 2 × 18650 Li-ion cells (7.4 V).
7. **Water tank + pump** — tube runs to the nozzle.
8. **Gear motors + wheels (×4)** — left pair and right pair.

2 SPECIFICATIONS

Power	2 × 18650 Li-ion, 7.4 V
Controller	Arduino Uno (ATmega328P)
Detection range	10–65 cm (candle-size flame)
Field of view	≈ 120° (3 sensors)
Spray	Mini pump 3–6 V · 180° sweep
Extinguish time	4.5–6 s (small flame)

3 HOW IT WORKS

SENSE — Sensors look left, centre and right for flame light.

DECIDE — Arduino turns the robot toward the flame.

ACT — Stops at a safe distance and sweeps water.

⚠ WARNING — READ BEFORE USE

- **Demonstrator only.** This is a student project, not a certified fire-safety device. Keep smoke alarms and extinguishers in place.
- **Small test flames only** — a candle or tealight on a metal tray, with an adult present and an extinguisher or water within reach.
- **Never use on electrical, oil, grease or gas fires.** Water can spread them or cause electric shock.
- **Keep water off the boards.** Disconnect the battery before filling the tank. Wipe spills at once.
- **Li-ion batteries can overheat.** Use protected cells, check + and –, never short them, use a proper charger and never charge unattended.
- **Sunlight and hot lamps** can trigger the sensors. Use indoors, away from windows.
- **Moving parts.** Keep fingers, hair and loose clothing away from the wheels. Do not run the pump without water.
- **Not a toy.** Keep away from young children. The motor driver can get hot.

K.J. Somaiya School of Engineering, Mumbai · Student project

Setup & calibration on the back ▸

SETUP & CALIBRATION — FOLLOW STEPS 1 → 5 IN ORDER

1 POWER UP

1. Fully charge both 18650 cells.
2. Insert them, matching the **+** and **–** marks in the holder.
3. Connect the battery lead. The Arduino and sensor LEDs light up.
4. The nozzle centres itself (points straight ahead).

2 FILL THE TANK

1. **Disconnect the battery first.**
2. Pour clean water into the tank up to the **MAX** line. Do not overfill.
3. Check the tube runs to the nozzle with no kinks.
4. Wipe up any drops on the boards before reconnecting.

3 CALIBRATE THE FLAME SENSORS · MOST IMPORTANT STEP

SENSOR · ≈ 50 cm · LED · KNOB

1. Light a candle on a metal tray about **50 cm in front of one sensor**, at the same height.
2. With a small screwdriver, turn that sensor's knob **slowly** until its LED **just turns ON**.
3. Move the flame out of view. The LED must go **OFF**. If it stays on, turn the knob back a little.
4. Repeat for the **left**, **centre** and **right** sensors.

Flame in view — LED must be ON

No flame — LED must be OFF

Tip: too sensitive = false alarms from lamps or sunlight. Not sensitive enough = shorter detection range. Calibrate in the room where you will use the robot.

4 CHECK THE STEERING

TOP

1. Lift the robot so the wheels are off the table.
2. Hold the flame to the **left** sensor: it should turn left. **Right:** turn right. **Centre:** both sides forward.
3. If a side spins the wrong way, swap that motor pair's two wires on the driver.

5 FIRST RUN

30–60 cm

1. Place the robot on a flat, hard floor, indoors.
2. Put a lit candle on a metal tray 30–60 cm ahead.
3. Connect the battery and step back.
4. The robot turns to the flame, drives up, stops and sweeps water. It checks again and repeats if needed.

? TROUBLESHOOTING

PROBLEM	LIKELY CAUSE	WHAT TO DO
Robot does not move	Flat batteries or loose lead	Recharge both cells. Check the battery lead and that all modules share ground (GND).
Moves with no flame	Sensors too sensitive, or sunlight	Close blinds, then turn the sensor knobs back slightly (Step 3).
Turns away from the flame	Motor wires reversed	Swap the two wires of that side's motor pair (Step 4).
Drives past the flame	Centre sensor not calibrated	Redo Step 3 for the centre sensor.
No water sprays	Empty tank, kinked tube, or relay off	Refill. Straighten the tube. Listen for the relay click when spraying.

After use: disconnect the battery · empty the tank · wipe the robot dry · store the cells separately in a cool, dry place.

Built by Abhishek Khedekar, Sahil Ramesh Undale & Navya Mathur · K.J. Somaiya School of Engineering, Mumbai · Student demonstrator, not a certified fire-safety device.

Figure B.1: User manual: front (left) and back (right).



Figure B.2: Comic-style project poster.

Acknowledgements

We would like to express our sincere gratitude to everyone who supported and guided us throughout the successful completion of our project, “Fire-Fighting Robot Using Arduino.”

We are especially thankful to our project guide, Kirti Sawlani, for valuable guidance, continuous encouragement and constructive suggestions throughout the development of this project. This support helped us overcome technical challenges and improve our understanding of sensors, embedded systems, motor control and hardware integration.

We extend our appreciation to all the faculty members, technical staff and our classmates who provided valuable suggestions and assistance during the implementation and testing phases.

Sahil Ramesh Undale - 16014024034
Navya Mathur - 16014024038
Abhishek Khedekar - 16014024029
KJSSE