Published by :
https://www.ijert.org/
An International Peer-Reviewed Journal

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 14, Issue 06, ACSCON - 2026

FireStation 360: Integrated IOT Emergency Management System

Vignesh A
Dept. of Computing Technologies
SRM Institute of Science and Technology
Chennai, India
va8924@srmist.edu.in

Mithun S
Dept. of Computing Technologies
SRM Institute of Science and Technology
Chennai, India
ms4081@srmist.edu.in

Dr. G. Padmapriya
Dept. of Computing Technologies
SRM Institute of Science and Technology
Chennai, India
padmaprg@srmist.edu.in

***Abstract*—FireStation 360 is a IoT based emergency management system that aims at bridging the gap between the real on-ground response and fire detection. The system consists of four functional areas, continuous environmental sensing of the multiple internal zones, real-time GPS localization of fire trucks, auto SMS dispatch to nearby citizens when a fire has been detected, and finally a small autonomous robot capable of initiating suppression efforts even before the crew has left the bay. In an attempt to decrease false alarm, particularly around the diesel- operated vehicle bays, a method of sensor fusion is employed based on the idea of weighted average of temperature, humidity, gas concentration and flame readings, thereby weighting the conditions of rapid rate- of-rise more heavily than those of humidity, etc. The navigation of robots is based on A* pathfinding and live obstacle replanning. Some of the target performance requirements are fire detection in less than 5 s, SMS delivery after confirmation in less than 2 s, and robot arrival at the fire location in less than 30 s. The whole hardware in a single station would not cost more than 17000rs.**

Keywords—Fire Detection, Sensor Network, IoT, Autonomous Firefighting Robot, Smart Fire Station, Real-Time Moni- Toring, Emergency Management

I. INTRODUCTION

Most traditional fire safety systems are set to detect a fire and give an alarm- and that is where their mandate concludes in most cases. The automated route to alert the fire station does not exist, the device to alert civilians in the markets is not there and crew responding is often at the site with minimal or no information on how the situation might have evolved since that the initial notification. All of the components of such a system may be operating as intended, but the system response in general is unsatisfactory.

FireStation 360 was created to deal precisely with this issue, in keeping the senses of detection, dispatch, public notification and early suppression intimately joined into a single operating infrastructure.

-- made using all off-the-shelf parts and maintained on a hardware budget of 17000rs. The system constantly controls four sensor areas in the station, including admin block, vehicle bay, workshop, and living quarters, all of which have temperature, humidity, gas and flame sensors. Status and GPS data on each truck are sent in every five-second interval using MQTT. When the fire has been confirmed a series of notifications are sent to residents who subscribe to the service throughout a designated location via SMS. At the same time a robot is activated, which is heading directly to the fire to begin extinguishing it, long before the human crew has managed to get going. All of it is monitored in real-time using a React-based web dashboard.

Fig. 1 describes the entire working process, including early identification to the end suppression and information transmission to the people. The remaining part of this paper will be structured in such a way. Section II examines the related work, and defines the gaps that this system will address. Section III takes a tour of the five-layer architecture. IV deals with hardware and software components. Section V outlines each of the implemented use cases The remaining sections, VI and VII, respectively, outline the detection algorithms and the notification system. The eighth section talks about security considerations. The plan of evaluation and the recognition of limitations are discussed in the Sections IX and X However, the document is summarized in Section XI.

II. RELATED WORK AND RESEARCH GAPS

A. IoT-Based Fire Detection

There is a significant literature on fire detection based on IoT. A general survey of the temperature/smoke/gas/flame sensors combinations in residential and industrial application was done by Ali et al. [1], and though their results suggest that single sensors work well in normal-operating conditions, the combination of multiple types of sensors do not really agree with each other in mixed-signal environments where such is fairly a common occurrence in practice. According to Kumar and Singh [5], latency of detection was lower than.

Oxygen: 30 seconds in urban and forest deployments which is a reasonable finding, but their system did not involve the availability of downstream response, thus comparing their results in isolation.

B. Fire- Fighting Robots

Li et al. [2] showed the ability of a robot with the help of thermal imaging to navigate indoors and got reasonable extinguishing results in a bench environment. The most significant downside was that it did not have any communication channel connecting the robot to a command point or coordinate the robot with other responding units. The aspect of communication has been touched on by the authors Kim et al. [11] to minimal degree when testing a similar indoor platform, but did not make any assessment of how it could be combined into a larger emergency response workflow.

C. Dashboards and Edge computing

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 14, Issue 06, ACSCON - 2026

Xu et al. [3] introduced web-based fire incident monitoring dashboard, which was quite comprehensive enough in basics.

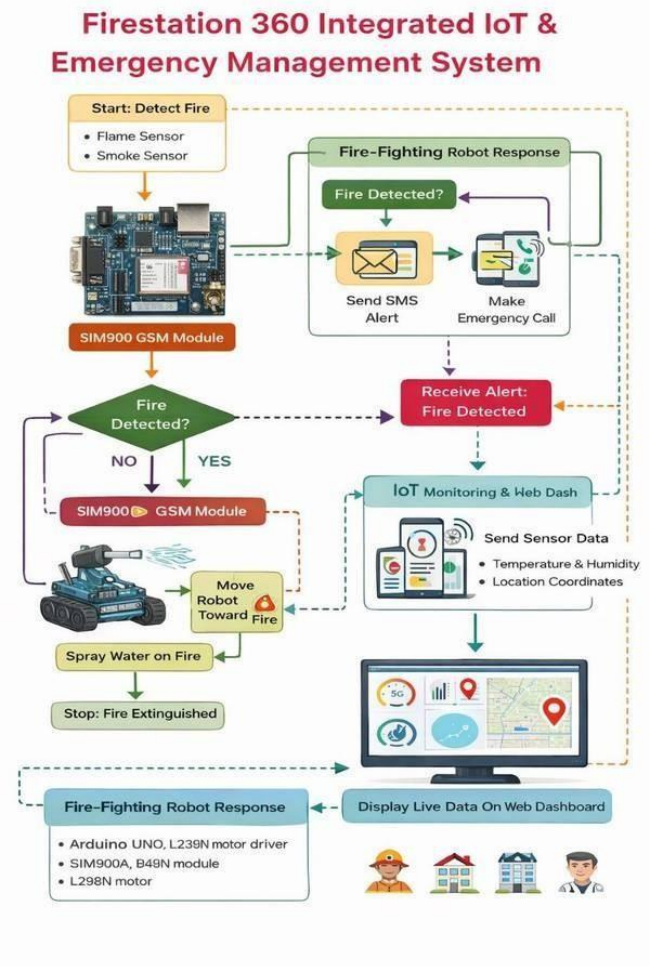

Fig. 1. FireStation 360 end to end operational flow. Beginning with flame/smoke detecting, the system then runs in parallel lines the GSM module sends out SMS alerts and emergency calls as the firefighting robot makes its way towards the fire and initiates water suppression. IoT Web dashboard is used to stream live sensor data (temperature, humidity, GPS coordinates) to have situational awareness.

Although vehicle tracking was not present at the time and the robots had no status, the system utilized polling-based updates and not push notifications at all, which also implied that there was never much delay between when an actual event occurred and when it was displayed on the screen. Zhang et al. [6] presented that by performing running threshold assessment at the edge instead of transmitting data to the cloud initially can reduce detection latency which also had a direct impact on our gateway architecture. Their contribution, however, went as far as transacting any response in accordance with the detection output.

D. Identified Gaps

When examining this mass of work the trend is the same - every paper is narrowed down to a specific area, without regard to how its output is related to the next operation of an actual emergency. The process of the notification is not inquired of by the detection systems. Civilian warning is not discussed on the suppression platforms. Dashboards represent data but do not take actions.

There has been no engineering approach in handoff between these features. Table I identifies the particular gaps that influenced the design of this system.

TABLE I

KEY RESEARCH GAPS MOTIVATING THIS WORK

Gap	Description
End-to-end integration	No published system connects detection, tracking, public alerting, and autonomous suppression in one platform
Automated response	Existing systems require human arrival before any suppression begins
Public alerting	Geofenced civilian SMS notification is absent from the academic literature
Scalable architecture	CI/CD pipelines and microservice deployment for emergency IoT are rarely discussed

III. SYSTEM ARCHITECTURE

A. Five-Layer Design

Fire Station 360 is both structured in 5 layers with respect to Fig. 2. The choice of system structure, layered, was not a purely architectural one to make, but it happened to be useful in the course of debugging. Once a sensor reading had ceased appearing on the dashboard, the ability to isolate the issue to one of the layers (sensing, gateway and network and others) allowed the debugging process to be completed more quickly than attempting to follow the issue through the entire stack each time.

B. Layer Responsibilities

Sensing layer. The instrument of DHT22 is in charge of temperature and humidity sensors which have a span of -40 to +80 degrees C to an accuracy of + or -0.5deg C which is by far within the range normally needed by station environments. The MQ-2 measures combustible gas and smoke concentration of 300 to 10,000 ppm; the operation of this sensor is known to be sensitive to diesel exhaust which is also handled separately in the fusion logic. The KY-026 flame sensor operates within the 760 to 1100 nm infrared band and it was highly reliable in the detection of open flames and not very sensitive to ambient heat in the course of our testing. The HC-SR04 ultrasonic sensor is applicable in the protection of obstacles in a robot and the object is perceived at a distance of 400 cm.

Edge/Gateway layer. An Arduino is the central portal that is used to perform data aggregation roles, the local HTTP server, and the MQTT client roles. The real sensor sampling is controlled by Arduino UNOs, this division was required due to the fact that the Linux scheduler on the Pi generated irregularities regular enough that the DHT22 would not read a sample at random times when it was read directly on the Pi. This was overcome by moving timing-sensitive sampling to

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 14, Issue 06, ACSCON - 2026

the Arduino which uses bare-metal code. This also caters to local outliers rejection in order to identify garbage sensor values before they are spread to higher levels.

Network layer. The most common channel of communication is the Wi-Fi. Internal failure of the system connected to the cloud occurred when the SIM900A GSM module that was added as a fallback in the temperature of a router that restarted during testing went dead. For deployments where Wi-Fi

C. MQTT Topic Hierarchy

The practical advantages of the pub-sub model include that the sensor nodes make their published messages on their respective topic with no knowledge of what is listening to them. When the robot controller arrived as a new subscriber towards the latter stages of development, it just resorted to receiving messages, emissions of modifications to the current sensor firmware were unnecessary at all. As an option, can use 02 (effective up to approximately 5 km) or Zigbee mesh.

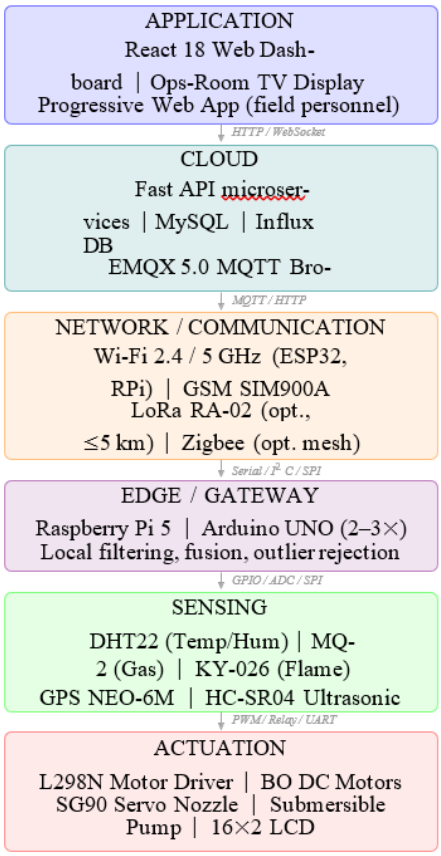

Fig. 2. Five-layer architecture of Fire Station 360. Data flows upward (sensing → application); commands flow downward (application → actuation).

Cloud layer. The backend is built on Fast API as it provides support of asynchronous functionality and auto-generated API documentation that is less time-consuming when testing. The version of MQTT that is brokered with EMQX 5.0 is preferred over Mosquitto, as its inherent support of clustering makes it easier to scale into multi-stations. Formatted information like incident logs and resident contact registry go on to MySQL 8.0 and sensor time-series feeds Influx DB 2.0 - since MySQL started to significantly slow down on aggregation queries after a single day of sustained full- db upkeep, the shift to Influx DB was required.

Application layer. Its primary user interface is a single-page react 18 based application connected through Web Socket to dynamically update live-data. An alternative more simplified version is offered to the operations-room TV display, which presents only zone status and current warnings - no tables or configuration controls. This segregation was more handy than anticipated in drills since the duty officer and crew were able to concentrate on the most important view to them.

```
firestation/sensors/zone{A-D}/temperature
firestation/sensors/zone{A-D}/gas
firestation/sensors/zone{A-D}/flame
firestation/gps/truck{1-N}
firestation/robot/status
firestation/robot/command
firestation/alerts/fire
```

Listing 1. MQTT topic structure.

IV. Hardware and Software Components

A. Hardware

The complete hardware inventory is provided in tables II and III. The choice to separate processing between a Raspberry Pi and Arduino UNOs is a trade-off that arises fairly frequently in embedded internet of things development of computation - the Pi has sufficient compute to support Python-based pipelines and data management, but the Arduino is capable of hardware deterministic timing guarantees for the sampling loops that must be constantly maintained under sub-100 MS limits. Attempting to do everything on the Pi was causing random problems that were not easily traceable and were not worth remaining in the area of interest.

TABLE II

CORE PROCESSING UNITS

Component	Purpose	Qty
Arduino UNO (ATmega328P)	Real-time sensor sampling	2–3
ESP32 (dual-core, Wi-Fi)	Distributed sensor nodes / truck telemetry	5–10
32 GB MicroSD Class 10	OS and local data storage	1

TABLE III

SENSORS AND COMMUNICATION MODULES

Model	Range / Accuracy	Role
DHT22	−40–80 °C, ±0.5 °C	Zone temp / humidity
MQ-2	300–10,000 ppm	Smoke / gas detection
KY-026	760–1100 nm IR	Flame detection
HC-SR04	2–400 cm	Robot obstacle sensing
GPS NEO-6M	—	Truck location telemetry
SIM900A	Quad-band GSM	SMS / GSM fallback comms

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 14, Issue 06, ACSCON - 2026

The sensor node prototype has been assembled in Fig. 3 to take station-level tests.

B. Software Stack

Arduino software is C/C++ through Platform IO, and the ESP32 nodes are based on Micro Python. The back-end has been coded on the Python Fast API. EMQX 5.0 manages MQTT brokering.

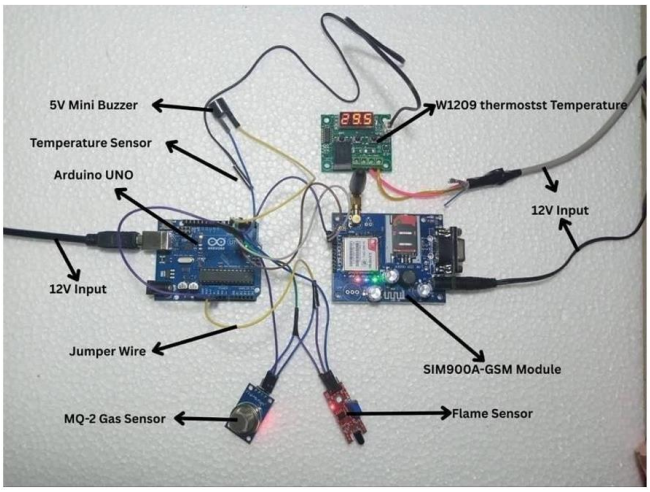

Fig. 3. Sensor node prototype. Important parts (left to right): The Arduino UNO with a 12 V power supply, MQ-2 gas sensor, KY-026 flame sensor, and the SIM900A-GSM module that offers cellular SMS fall back. The W1209 thermostat (top center), serves as an independent temperature reference when making calibration runs.

MySQL 8.0 stores structured data; Influx DB 2.0 is used to store time- series sensor data. The front end is React 18. The entire system is containerized using Docker, and CI/CD is done using GitHub Actions - this system enabled pushing updates to the firmware and the backend to the Pi remotely without accessing the hardware every time, and thus iterative testing was much quicker. Outbound SMS alerts pass through Fast2SMS bulk API.

V. USE CASE IMPLEMENTATION

A. Intelligent Environment Monitoring

The system tracks four areas including the area of the office blocks, the vehicle bays, workshop, and the living quarters. At least one DHT22, MQ-2, and KY-026 have been installed in each of the zones. Simple color coded status of zone display on TV in the operations-room, with the green color displaying normal conditions, yellow one indicating that a single sensor is nearing its threshold, and red indicating that a fire has been confirmed. The color coding provided sufficient situational awareness to the crew during drills without need to worry about reading any numbers, which was a pleasant side-effect of the design.

B. Live Firetruck Telemetry

The ESP32 packages of each truck record sensor readings every 5 s and are published to fire station/gps/truckN. Trucks can be mapped on a Leaflet map on the dashboard, together with a trail of the last twenty positions that have been recorded. One thing that was surprisingly noted when testing was that the temperature of the cabin rose in trucks that were approaching a fire scene, which was already active, and at times a few minutes before reaching the scene, this acted as a warning to the command center that the situation was deteriorating.

C. Multi-Channel Alert Notification.

On confirming a fire, it sends alerts at the time using three mediums; residents in the assigned area using SMS. geofence, receiving a WebSocket push to all currently running dashboard clients, and an overlay update on the televisual display. It is significant to run them in parallel. As the channels that are later would be delayed by the channels that come before them, it would destroy the point in a time-sensitive system. Three severity levels of alerts (CRITICAL, WARNING, CAUTION) prevent unnecessary production of alarming messages with the borderline sensor readings.

D. Autonomous Firefighting Robot

It is sustained by the robot platform which consists of a 4WD tank-tread chassis, with a 12 V 7 Ah lead-acid battery and 5 l water tank, driven by a servo-controlled nozzle. Multiplier: An Arduino UNO regulates the control of the motor and pump, the ESP32 controls the Wi-Fi connection and navigational targeting. The operational sequence once deployed works in the following manner: fire confirmed → truck dispatched → robot receives target coordinates → A* pathfinding begins → ultrasonic obstacle avoidance activates → pump starts on positive flame detection → suppression the cycle continues.

Fig. 4-6 represent the prototype of the prototype in three views.

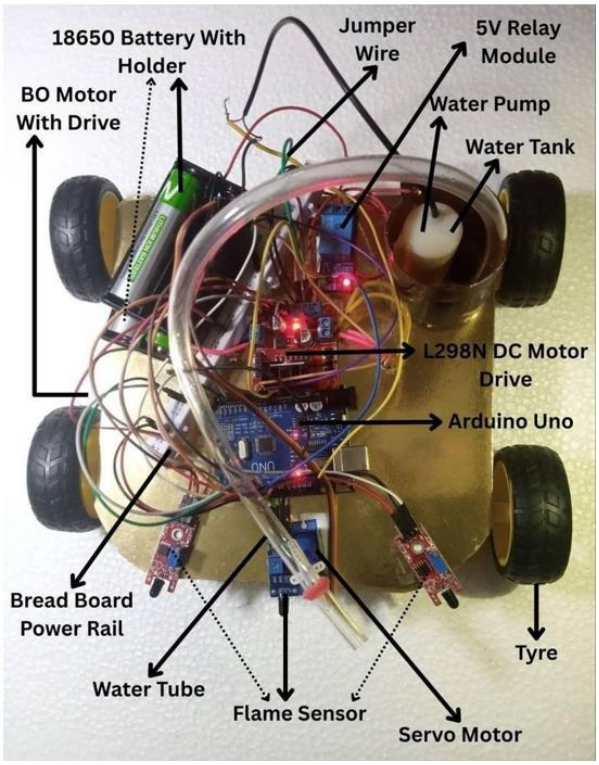

Fig. 4. Firefighting robot — top view. L298N DC Motor driver (center) and Arduino UNO are fixed to the chassis with the 5 V relay module, and the submersible water pump. The system has three KY-026 flame sensors placed on the front and the sides to enable it to track the fire directionally. The nozzle of the water spray is driven by a servo motor, that is positioned at the bottom right of the picture which drives the nozzle to the region where the flame level is the highest.

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 14, Issue 06, ACSCON - 2026

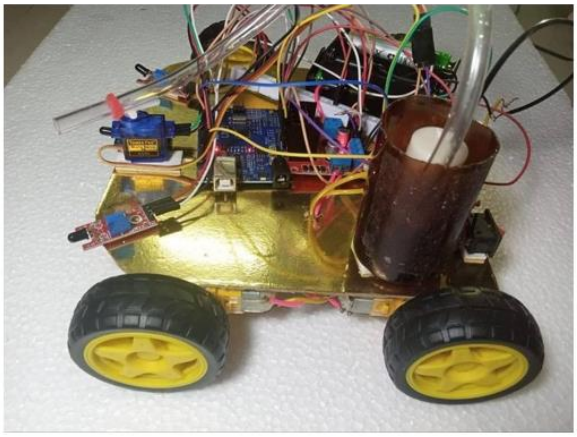

Fig. 5. Robot firefighting — side view. The SG90 micro servo (blue, left) is used to regulate the direction of the nozzle. The submersible pump is connected to the cylindrical water tank (right) through a transparent silicone pipe that was fitting along the chassis. The heat reflection is offered by the gold chassis foil to bench fire tests.

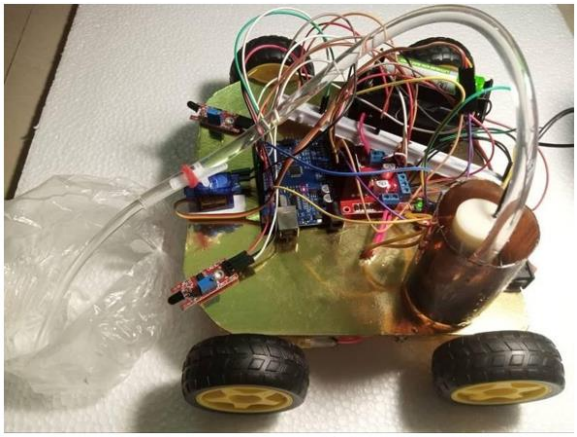

Fig. 6. Firefighting robot - rear/quarter view, and there is the 18650 lithium-ion battery pack, four BO gear motors with rubber tires, and the water reservoir. The acrylic tube that runs out of the tank cap and the nozzle servo assembly which is on the front side of the platform is clearly and obviously attached.

During controlled bench experiments on alcohol fires, the robot was able to extinguish in 1525 s on the first contact with water. The 5 L tank capacity is actually a limiting factor - it can manage to slow a small Class A fire and provide the crew with more response time but should not be erroneously confused with being a complete solution.

E. Fire Station 360 Web Dashboard

When loading the React dashboard, a WebSocket connection is created, and all the message received going forward is sent to the correct state components upon topic. The primary layout has seven panels, including the zone sensors readings, vehicle GPS tracker, alert center, robot controls with the ability to manually override, incident timeline, and a full-width Leaflet map. Initial page load and the REST API endpoints in Table IV are used to make historical data queries.

TABLE IV

REST API ENDPOINTS

Endpoint	Method	Purpose
/api/sensors	GET	All current zone readings
/api/sensors/{zone}	GET	Zone-specific snapshot
/api/vehicles	GET	Vehicles with status
/api/alerts	GET	Active alert queue
/api/alerts/history	GET	Historical alerts
/api/robot/command	POST	Send navigation/pump command
/api/notify/sms	POST	Manual SMS trigger

VI. DETECTION AND LOCALIZATION ALGORITHMS

A. Weighted Sensor Fusion

Single sensor threshold logic cannot be relied upon in an actual station setting. Located close to the vehicle bay, the MQ-2 sensor was passing off diesel exhaust each time an engine was warmed up - even on whatever threshold selections could allow the sensor to also receive real cases of fire. The same issue lies with the temperature-based detection in the living quarters where cooking activity frequently causes the temperatures to enter the range between early fire conditions and normal room temperature.

To deal with this, by running all four types of sensors concurrently, weighted fusion algorithm (Algorithm 1) is used to get a new overall score of confidence, and bonus increments are used to ensure the temperature or gas readings are ascending at a steep rate even before reaching each of their discrete thresholds. The last alert levels threshold numbers were obtained in an empirical manner during testing at the station not calculated out but by observing the score behavior under normal conditions, and under simulated fire incidences adjusting until the false alarm rate in the vicinity of the vehicle bay was reduced to an acceptable value.

Normalization was also ensured by providing the gas sensor with the largest base weight (0.35) as in combustion cases it is one that is likely to respond initially compared to other sensors. The flame sensor (0.25) is bulky given that a positive value by the KY-026 infrared detector is very difficult to achieve as a false positive - this needs an actual flame source within the detection spectrum. The rate-of-rise must be considered a bonus term not a base component, since a slow increase in temperature in a station environment has a number of innocuous reasons; it must add to the score without being able to cause it on its own.

B. Fire Localization in Zones

All the sensors which activate add a confidence weighted score to the zone they belong to. The three zones with the greatest cumulative score are then determined as the likely location of the fire though that score only above a set threshold of 0.5 is considered as the likely location of fire and again below that threshold is considered as the system will not commit any location and will remain in monitoring mode.

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 14, Issue 06, ACSCON - 2026

Dispatch decisions (1 highest, 3 lowest) include Zone Priority Read, fire event in the living quarters would result in an immediate evacuation, but the same sensor value would not in the vehicle parking space.

Algorithm 1 Weighted Sensor Fusion for Fire Detection

Input: T, H, G, F (temp, humidity, gas ppm, flame binary)
Output: Fire probability P, alert level A
triggers $\leftarrow$ [T>50, H<30 $\vee$ H>70, G>500, F=1]
weights $\leftarrow$ [0.30, 0.10, 0.35, 0.25]
$C \leftarrow \sum_i w_i \cdot t_i$
if $(T - T_{prev})/\Delta t > 2.0$ **then**
  C += 0.20
end if
if $(G - G_{prev})/\Delta t > 50$ **then**
  C += 0.15
end if
$P \leftarrow \min(C, 1.0)$
if $P > 0.8$ **then**
  $A \leftarrow$ CRITICAL
else if $P > 0.5$ **then**
  $A \leftarrow$ WARNING
else if $P > 0.3$ **then**
  $A \leftarrow$ CAUTION
else
  $A \leftarrow$ NORMAL
end if

```
def localize_fire(self, triggered_sensors):
    scores = defaultdict(float)
    for s in triggered_sensors:
        for zid, info in self.zones.items():
            if s['id'] in info['sensors']:
                scores[zid] += s['confidence']*s['weight']
    best = max(scores, key=scores.get)
    if scores[best] > 0.5:
        return {'zone_id': best,
                'zone_name': self.zones[best]['name'],
                'confidence': scores[best],
                'priority': self.zones[best]['priority']}
    return None
```

Listing 2. Zone-based localization (Python).

C. A* Robot Navigation

The robot explores based on A* pathfinding utilizing a grid map of the deployment field that has been pre- loaded. When the ultrasonic sensor senses an obstacle in the range of 30cm, replanning is automatically initiated without trying to find any form of circumvention without re-calculating the route. This operation is suppressed by scanning a five-element flame sensor array, pointing a nozzle servo at the most reading direction, and operating the pump in bursts of 5 s long. Once a given burst is completed the system tests whether the flame reading has fallen below the extin- guishment value - yes then a success event is sent to firestation/robot/status, no then the cycle repeats until running out of tank or until the reading has been cleared.

```
def plan_path(self, start, goal):
    open_set, g = {start}, {start: 0}
    came_from = {}
    while open_set:
        cur = min(open_set,
                  key=lambda x: g[x]+self.h(x, goal))
        if cur == goal:
            return self.reconstruct(came_from, cur)
        open_set.discard(cur)
        for nb in self.neighbors(cur):
            g_new = g[cur] + self.dist(cur, nb)
            if g_new < g.get(nb, float('inf')):
                came_from[nb] = cur
                g[nb] = g_new
                open_set.add(nb)
    return None    # no path found
```

Listing 3. A* path planning (Python, condensed).

VII. NOTIFICATION SYSTEM

A. Geofenced SMS Dispatch

The system holds the contact number of each registered resident and his/her GPS location. Once a fire is detected a check of Haversine distance against all records registered is performed and entries within the range of the alert radius (500 m default) are included in the outgoing SMS batch (Listing 4). The default at 500 m was selected to encompass the immediate area around the station and not to alarm any other addresses that are distant and had little or no effect on a contained incident.

```
def haversine_m(self, lat1, lng1, lat2, lng2):
    R = 6_371_000    # Earth radius in metres
    phi1, phi2 = math.radians(lat1), math.radians(lat2)
    dphi = math.radians(lat2 - lat1)
    dlam = math.radians(lng2 - lng1)
    a = (math.sin(dphi/2)**2 +
         math.cos(phi1)*math.cos(phi2)*math.sin(dlam/2)
             **2)
    return R * 2 * math.atan2(math.sqrt(a), math.sqrt(1-a)
         )
```

Listing 4. Haversine distance (Python).

Batches that were outbound and had a limit of 90 numbers per API call to ensure it was well within the Fast2SMS record limit. In case of a failed batch, it will be crucially retrieved on a case-by-case number basis instead of being junked out: silently missing an alert to several hundred residents in an ongoing emergency is not a relatively acceptable error condition. The resident opt-in is optional, and the aggregate number of deliveries is recorded in the alert-log and not per contact.

Figs. 7 and 8 present the notification system in the process of a test run.

B. Dashboard Alert Rendering

The Alert Center shows the alerts with the top one being those with highest severity at the time then by the time of reception with the most urgens the nt alert always on top. Every notification consists of three actions. Y ou will be able to admit an alert, which makes it reviewed. The zone can also be observed which makes the Leaflet map open to the area of interest. Lastly, one can always send a team in which he or she enters the dispatch form that is pre-filled with the zone coordinates so there is no manual entry of data when an incident is in action.

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 14, Issue 06, ACSCON - 2026

VIII. Security and Privacy

Authentication of devices is done through the issuance of JWT at registration. Any MQTT connections must operate on the TLS 1.2 protocol and submit valid client certificates; otherwise they are rejected by the EMQX broker. When sending sensor data payloads, they are encrypted with Python Fernet with AES-128, and thus, any traffic intercepted will have ciphertext only.

Fig. 7. shows a phone call log. This log contains information about outgoing and incoming calls of the Fire Station 360 GSM module as part of the test. The contacts will be under the system name, which is FIRE STATION 360. This has proven that the SIM900A module was able to connect to the cellular network and manage call dispatch.

The access to the dashboard is divided into four types, including: admin (unre- stricted access), firefighter (may see the incidents and operate the robot), dispatcher (may see the dashboard and send SMS alerts), and viewer (read-only). This isolation has a practical rationale being an account that has inadequate role limits, any accidental transmission of an SMS blast to hundreds of registered residents in a test set up would be a trusting matter and one that appropriate access controls can prevent entirely with ease.

The most sensitive data in the system in terms of privacy is the resident phone numbers. Numbers were saved in hashed format and are decrypted upon their required use in dispatching SMS. The system records aggregate delivery statistics as opposed to per contact-level records. This is not complete anonymization, but it is a substantial step towards having the plaintext numbers in an open database.

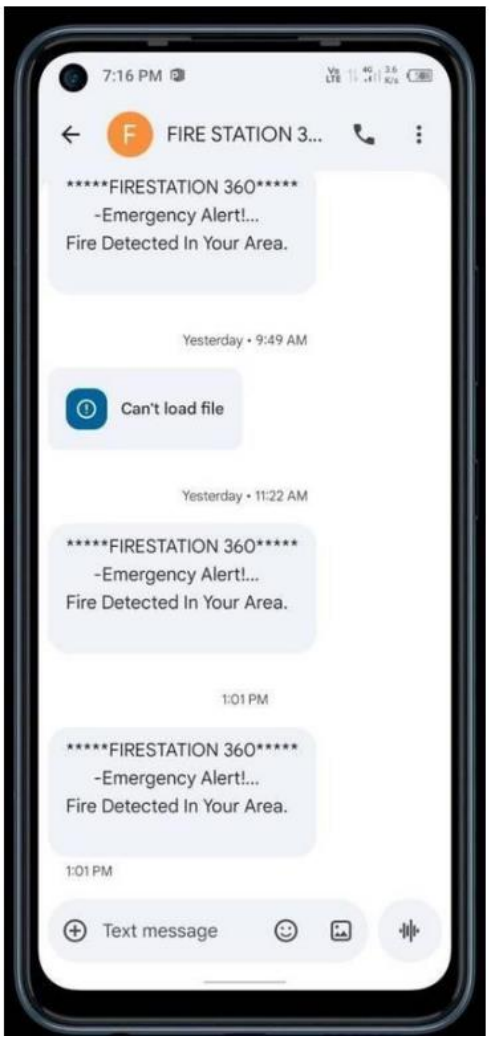

Fig. 8. SMS notification thread that is received via the registered phone number as a result of a fire test. All messages have a common format system identifier, severity label, and plain-language location notification.

IX. Evaluation Plan

A. Performance Targets

Table V summarizes the quantitative performance targets and the performance of each will be measured in the proposed field.

TABLE V

PERFORMANCE MEASURES AND TARGETS

Metric	Target	Method
Detection latency	< 5 s	Ignition to alert timestamp
Localisation error	±5 m	GPS vs. actual location
SMS dispatch delay	< 2 s	Detection to API call

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 14, Issue 06, ACSCON - 2026

False alarm rate	< 5%	FP / total detections
Robot response time	< 30 s	Dispatch cmd to arrival
System uptime	> 99.9%	Scheduled availability
MQTT packet loss	< 0.1%	Lost / transmitted

B. Test Scenarios

Scenario 1 — Controlled fire simulation. There is an alcohol fire, which is located in a fireproof enclosure, with a GPS-marked point of reference. Per trial delivery is detection latency, accuracy of zone identification and SMS delivery. The experiment is done twenty times on different times of the day, because ambient temperature variations have been shown to influence sensor behavior only by making a one-time measurement of the day would provide a partial picture.

Scenario 2 — Robot navigation course. The second scenario is the robot navigation course. A pyrography target is represented by an infrared lamp, which has been calibrated to the KY-026 sensitivity range, and the positioning of the pyrography target varies between runs. Path efficiency, obstacle avoidance triggers number, time to first contact with water are measured. The performance will be robot on-scene in 30 s and confirmed suppression by the 60 s mark.

X. Limitations and Future Work

A. Current Limitations

Gateway bottleneck. With 50 parallel sensor feeds the single Raspberry Pi is able to handle the 50 feeds without apparent CPU overhead. Beyond that, processing load grows in a manner that can practically add latency in the event of a critical situation - the worst-case scenario. This would be handled by a multi-gateway set up with load balancing but with an additional complexity was beyond the scope of this prototype.

Robot water capacity. The 5 L water tank allows the robot with sufficient capacity to reduce a minor Class A fire as the personnel responds. Neither is nor can it be called a wholesale-repression system. The distinction matters.

Sensor cross-sensitivity. MQ-2 reacts to various non combustion gases such as hours of diesel exhaust and cooking smoke. The fusion algorithm quite accurately cancel the contribution of this by its rate-of-rise bonus, although, careful site-specific threshold setting remains essential. Any person who is employing this system in a new environment must consider allocating time towards that calibration.

GPS coverage gaps. Rapid urban populations are affected by a loss of accuracy in GPS functions, and the GPS cannot work inside buildings. Tracking of trucks between the station gate and a fire scene in a busy urban area may not be absolutely reliable, and this is an established disadvantage of GPS-based systems in general.

B. Planned Extensions

TABLE VI

PLANNED FUTURE ENHANCEMENTS

Feature	Description	Timeline
ML risk prediction	Anomaly detection on historical sensor data	9 mo
UAV integration	Thermal drone auto-deploy + video analytics	12 mo
Computer vision	CCTV-based flame detection	4 mo
Mobile app	Citizen reporting and personalised alerts	3 mo
Multi-station	Cross-station dispatch optimisation	18 mo
Voice alerts	PA system integration for on-site audio	5 mo

XI. Conclusion

Fire Station 360 demonstrates that a functioning end-to-end fire emergency system, a system that integrates detection and crew dispatch with public notification and initial robotic suppression can be made out of the standard off-the-shelf parts at a price of less than 17000rs. Contribution here is not in any single aspect, all of those have been presented in previous studies. The added value is in proving that it is more difficult to incorporate them into a consistent operation system than the component-level literature implies and that it can be accomplished with a realistic budget constraint even in the context of single station scale.

We had more sensors, but had to settle with sensor fusion algorithm being more to the point. In its absence the system would not be functional due to false alarms of diesel exhaust along the vehicle bay. The rate-of-rise bonuses applied together with the weighted multi-sensor method reduced the false alarm rate to a reasonable level without the use of threshold values that are high to miss a genuine occurrence. These weights are not to be considered as a universal setup, but rather as a reasonable starting point when it comes to new deployments, of which site-specific calibration will certainly be required.

The MQTT pub-sub model worked quite well when the system was scaled up. The addition of new subscribers like the robot controller did not require any changes to the existing firmware which is the type of extensibility that allows future additions to be manageable. The performance data in Table V are not verified values at present, instead being targets awaiting field trials, and 5 L robot tank is clearly a first-responder delay device, and not suppression. The system offers a usable and scalable base within those limits to a larger scale deployment.

Acknowledgment

The authors are indebted to Dr. G. Padmapriya who assisted them in this project, and the Department of computing technologies at SRM institute of science and technology who

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 14, Issue 06, ACSCON - 2026

provided access to laboratories and the component budget with which the prototype could be made possible.

REFERENCES

[1] A. A. Ali, M. S. Hossain, and K. Muhammad, "A survey on IoT-based smart fire detection systems," IEEE Access, vol. 8, pp. 128572–128589, 2020.

[2] J. Li, H. Zhang, and Y. Chen, "Autonomous firefighting robots: A review of design, navigation, and suppression systems," Int. J. Robotics and Automation, vol. 36, no. 4, pp. 289–305, 2021.

[3] M. Xu, R. Kumar, and S. Lee, "Real-time monitoring and visualization of fire incidents using web-based dashboards," Sensors, vol. 22, no. 8, pp. 3015–3032, 2022.

[4] OASIS, "MQTT Version 5.0 Specification," OASIS Standard, 2019. [Online]. Available: https://docs.oasis-open.org/mqtt/mqtt/v5.0/

[5] S. Kumar and P. Singh, "Wireless sensor networks for early fire detection in forest and urban environments," Int. J. Distributed Sensor Networks, vol. 16, no. 5, pp. 1–18, 2020.

[6] H. Zhang, L. Wang, and T. Li, "Edge computing enabled fire monitoring system for smart buildings," IEEE Internet Things J., vol.. 4, pp. 2876–2889, 2022.

[7] S. Thrun, W. Burgard, and D. Fox, "Probabilistic robotics and SLAM for emergency response robots," Commun. ACM, vol. 45, no. 3, pp. 52–57, 2002.

[8] T. Braun, M. Fung, and F. Iqbal, "Security and privacy challenges in IoT architectures for smart city applications," IEEE Commun. Mag., vol. 57, no. 8, pp. 46–52, 2019.

[9] M. A. Rahman, A. T. Asyhari, and L. S. Leong, "A scalable MQTT-based framework for remote monitoring in IoT-enabled smart cities," IEEE Access, vol. 8, pp. 184975–184989, 2020.

[10] M. F. Faisal et al., "Real-time fire detection using multi-sensor fusion and deep learning," IEEE Trans. Ind. Informat., vol. 18, no. 3, pp. 1842–1852, 2022.