

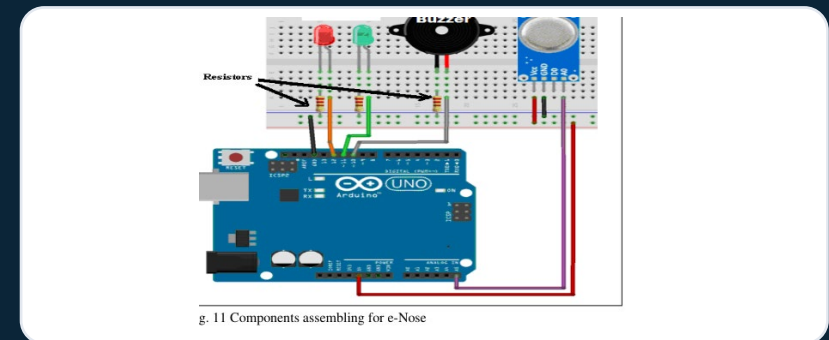
Cloud-Assisted IoT-Enabled Smoke Monitoring System

(e-Nose) using Machine Learning Techniques

An intelligent warning prototype using Arduino, MQ-2 gas sensor, LEDs and buzzer for privacy-preserving cigarette smoke detection in public spaces.

Presented at SSIC 2019 · Springer Proceedings

Anubha Parashar · Manipal University Jaipur



Why this problem matters

Public no-smoking rules often fail without immediate, non-invasive enforcement feedback.

- Active and passive smoking both create serious health risks.
- Camera-based detection can be costly and privacy-sensitive.
- A compact sensor-based warning device can alert smokers in real time.

Sensor-based

No camera required

Low cost

Easy to assemble

Real time

Immediate warning

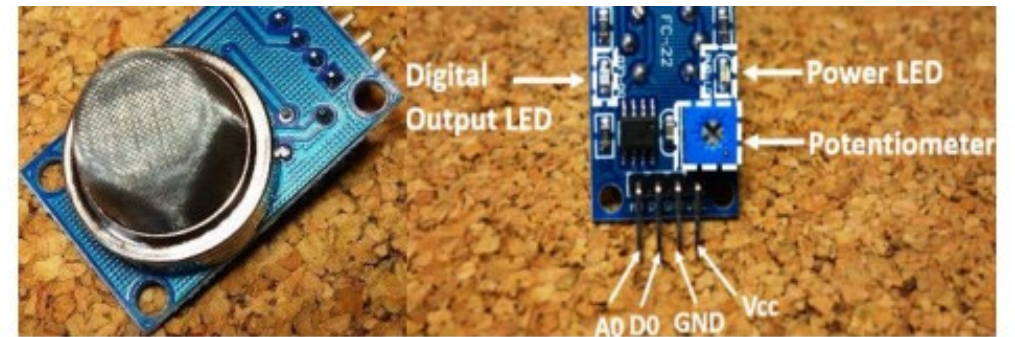
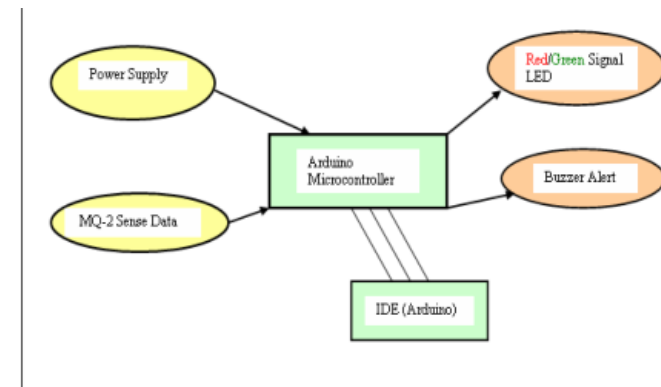
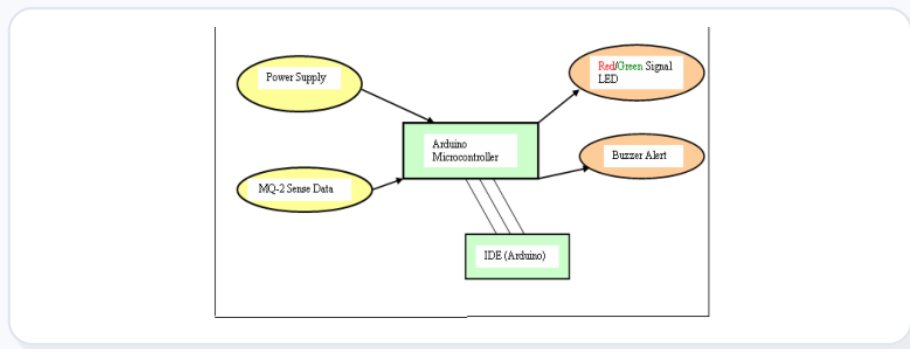
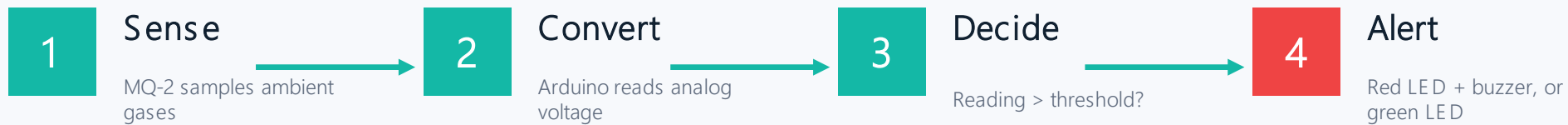


Fig 5. Smoke Sensor MQ-2 [15]



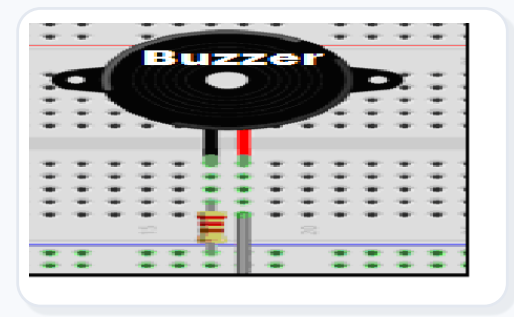
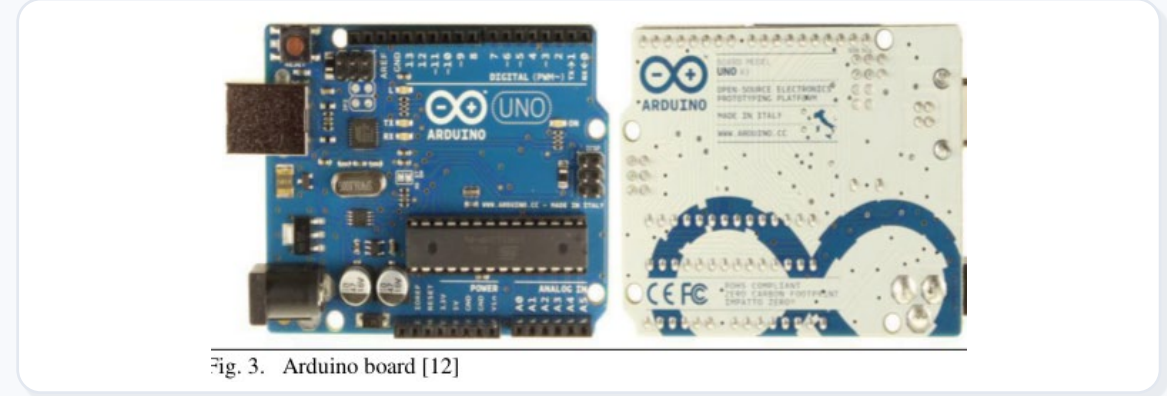
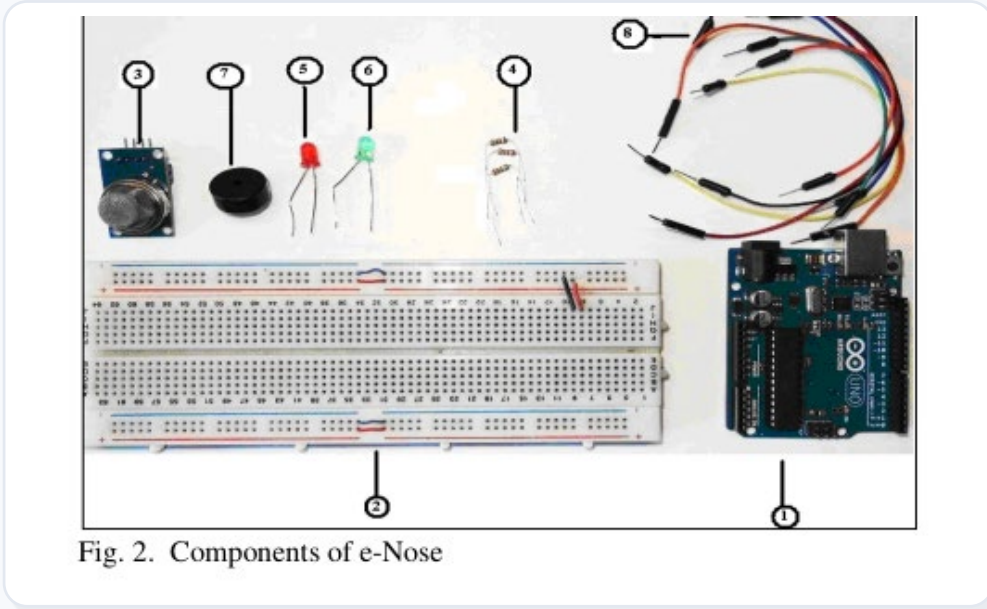
e-Nose solution overview

A simple closed-loop device: sense smoke particles, compare against a threshold, and trigger human-readable alerts.



Core design from the paper: sensor input + Arduino microcontroller + LED/buzzer outputs.

Hardware stack and signal path



Bill of materials

- Arduino UNO board
- MQ-2 smoke detection sensor
- Breadboard + jumper wires
- Red/Green LEDs, buzzer, resistors

Signal path

MQ-2 output voltage → Arduino analog read → threshold decision → LED/buzzer action

Sensing logic: smoke thresholding

Algorithm 1 : Algorithm for E-Nose

```
begin Algorithm_1;
```

Step :1. Make Global Definitions for variables and digital interfaces-pin positions for sensors

```
1. int smokeThreshold = 400;
2. int redLEDSignal = 12;
3. int greenLEDSignal = 11;
4. int buzzer_Warning = 10;
5. int smoke_Read = A5;
```

Step :2. Do Initialization: Assign mode for sensors and set baud-rate to the serial transmission

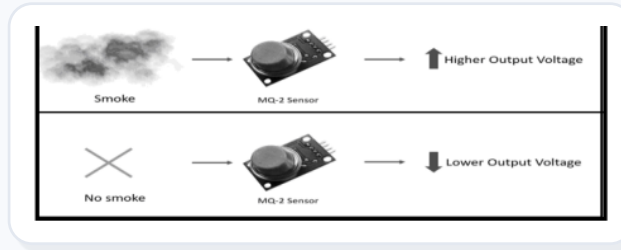
```
6. void setup()
  begin setup;
7.   pinMode(redLEDSignal, OUTPUT);
8.   pinMode(greenLEDSignal, OUTPUT);
9.   pinMode(buzzer_Warning, OUTPUT);
10.  pinMode(smoke_Read, INPUT);
11.  Serial.begin(9600);
  end setup;
```

Step :3. Define body of loop

```
12 void loop()
  begin loop;
13.  int analog_smoke = analogRead(smoke_Read);
14.  Serial.print("Pin A0: ");
15.  Serial.println(analog_smoke);
16.  if (analog_smoke > smokeThreshold)
  begin if;
17.    digitalWrite(redLEDSignal, HIGH);
18.    digitalWrite(greenLEDSignal, LOW);
19.    tone(buzzer_Warning, 1000, 200);
  end if;
20.  else
  begin else;
21.    digitalWrite(redLEDSignal, LOW);
22.    digitalWrite(greenLEDSignal, HIGH);
23.    noTone(buzzer_Warning);
  end else;
24.  delay(100);
  end loop;
```

```
end Algorithm_1;
```

Fig. 12 Algorithm for e-Nose (Steps for working)



400

Configured smoke threshold

A5

Arduino analog input

100 ms

Loop delay between reads

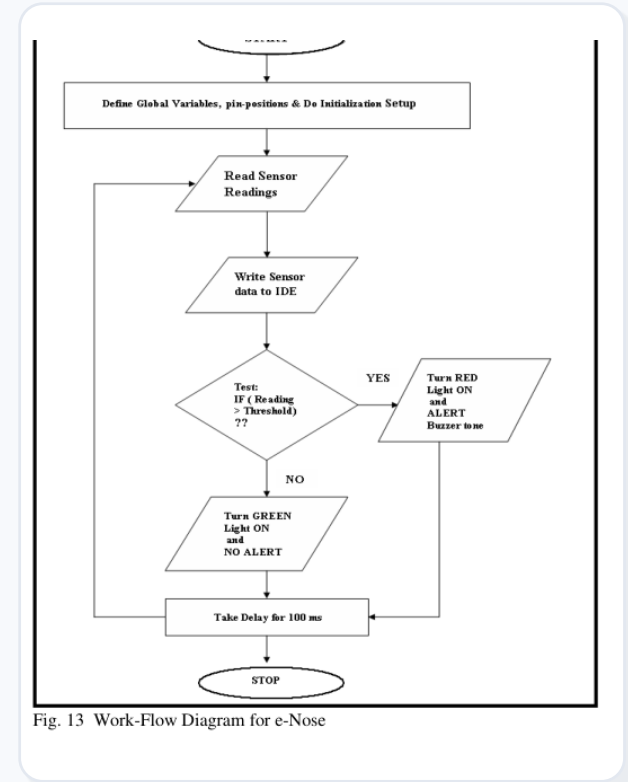
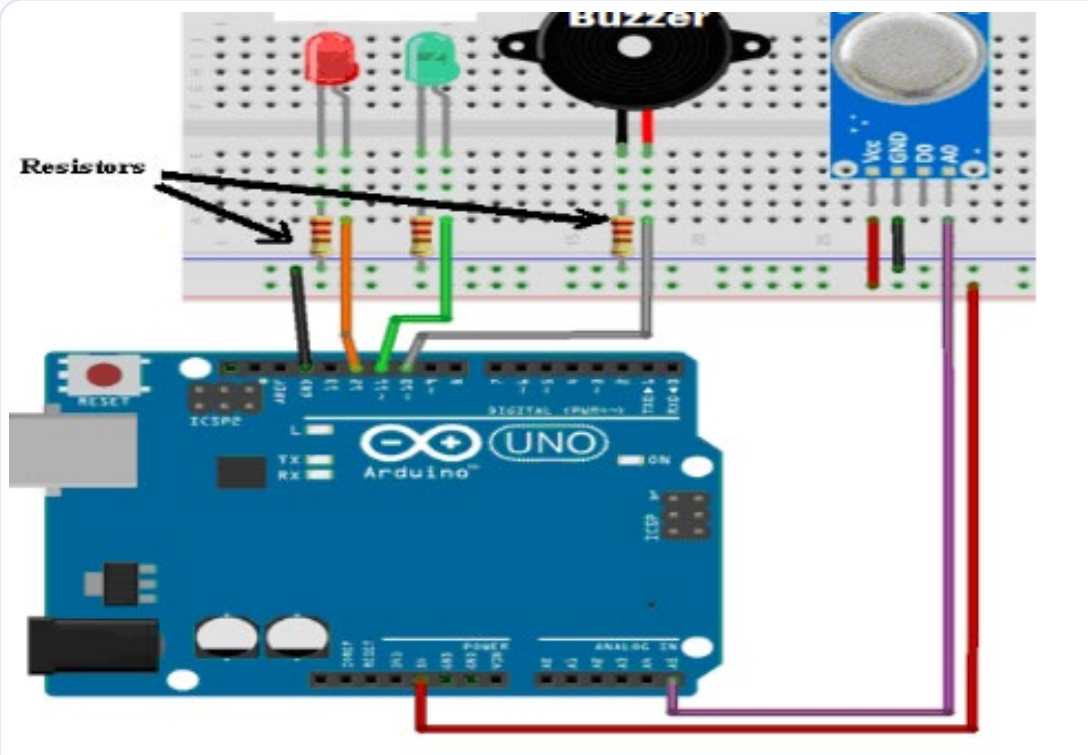


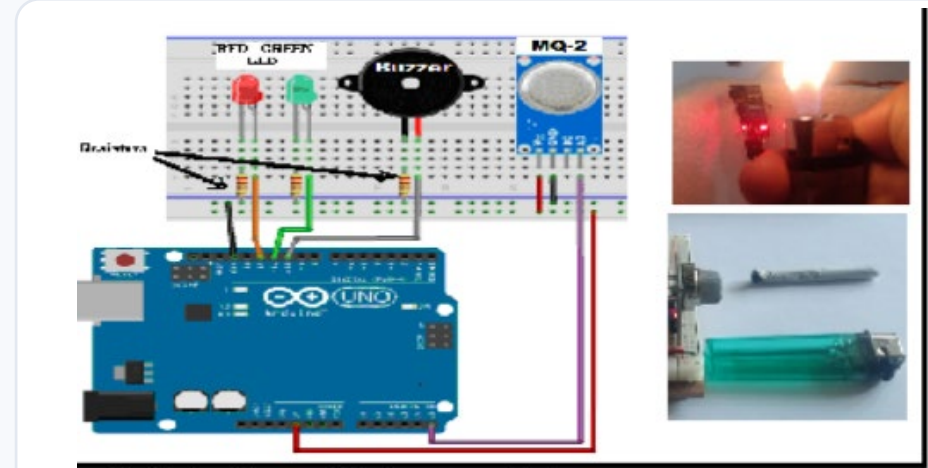
Fig. 13 Work-Flow Diagram for e-Nose

When sensor reading exceeds the threshold, the prototype turns the red LED on and plays a buzzer tone; otherwise, the green LED remains on.

Prototype assembly and test setup



g. 11 Components assembling for e-Nose




g. 14 Testing Set-up with cigarette and lighter

Experiment structure

- Initial environment: no smoke, green LED on.
- Introduce burning cigarette smoke near sensor.
- Observe alert state and recovery as smoke dissipates.

Three-phase behavior during testing

Phase 1 · no smoke



```

in A0: 473
in A0: 293
in A0: 292
in A0: 291
in A0: 291
in A0: 290
in A0: 290
in A0: 289
in A0: 288
in A0: 288
in A0: 287
in A0: 287
in A0: 286
in A0: 286
in A0: 285
in A0: 285

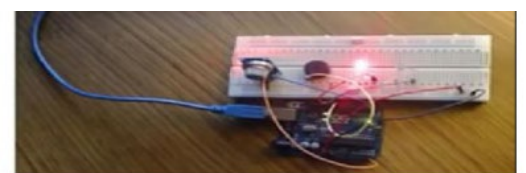
```

Test_Phase 1: Initial time instant
Absence of smoke
Green-LED as ON

Readings at Pin A0 < 400 (Threshold)

g. 15 Test_Phase 1: NO Smoke with GREEN LED

Phase 2 · smoke alert



```

in A0: 499
in A0: 497
in A0: 495
in A0: 492
in A0: 490
in A0: 487
in A0: 485
in A0: 481
in A0: 478
in A0: 476
in A0: 472
in A0: 464
in A0: 466
in A0: 464
in A0: 464
in A0: 461
in A0: 458
in A0: 455
in A0: 452
in A0: 449
in A0: 447
in A0: 444

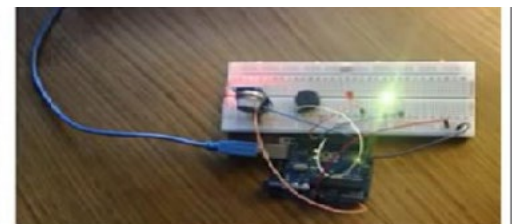
```

Test_Phase 2: After the introduction of burning cigarette to e-Nose
Presence of Smoke
RED LED as ON

Readings at Pin A0 > 400 (Threshold)

g. 16 Test_Phase 2: Smoke ALERT with RED LED

Phase 3 · smoke removed



```

Pin A0: 364
Pin A0: 362
Pin A0: 361
Pin A0: 360
Pin A0: 357
Pin A0: 355
Pin A0: 353
Pin A0: 352
Pin A0: 351
Pin A0: 349
Pin A0: 348
Pin A0: 346
Pin A0: 345
Pin A0: 343
Pin A0: 342
Pin A0: 341
Pin A0: 340
Pin A0: 338
Pin A0: 337
Pin A0: 335
Pin A0: 335
Pin A0: 334

```

Test_Phase 3: At later time instant
No Smoke particles in air
GREEN LED as ON

Readings at Pin A0 < 400 (Threshold)

Continuously decreasing as the source of smoke is removed from the e-Nose

Fig. 17 Test_Phase 3: NO Smoke with GREEN LED

Observed transition

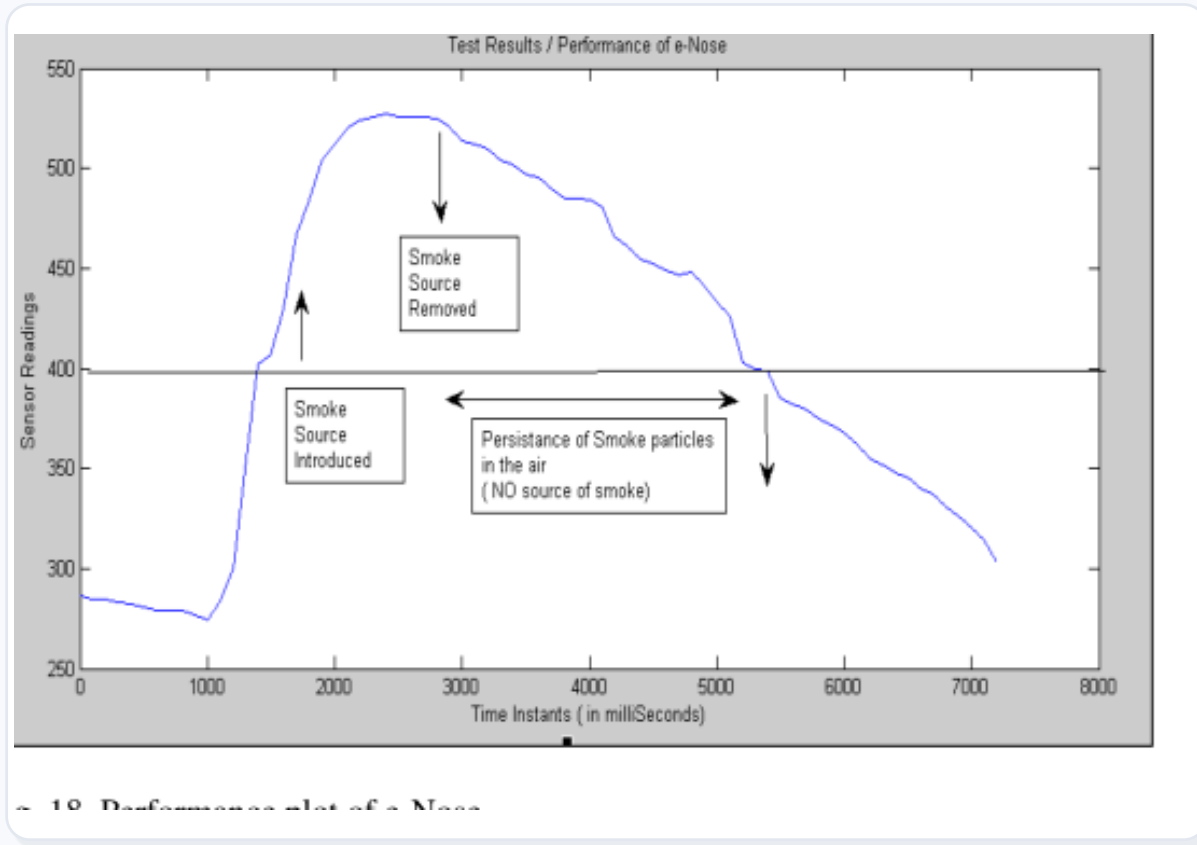
The readings rise above the 400 threshold when smoke is introduced, then fall as smoke particles dissipate. The user-facing alert changes accordingly.

Green
Normal / safe indication

Red + buzzer
Smoke warning state

Green
Return to baseline

Experimental performance



Pin A0 :	Time Instant (mSec.)	Sensor Readings
Pin A0 :	0	286
Pin A0 :	100	285
Pin A0 :	200	285
.....		
Pin A0 :	1200	300
Pin A0 :	1300	355
Pin A0 :	1400	402
Pin A0 :	1500	407
Pin A0 :	1600	430
.....		
Pin A0 :	2100	520
Pin A0 :	2200	524
Pin A0 :	2300	526
Pin A0 :	2400	527
Pin A0 :	2500	526
Pin A0 :	2600	526
Pin A0 :	2700	526
Pin A0 :	2800	525
Pin A0 :	2900	521
Pin A0 :	3000	514
Pin A0 :	3100	512
Pin A0 :	3200	510
Pin A0 :	3300	504
Pin A0 :	3400	502
.....		
Pin A0 :	4200	466
Pin A0 :	4300	461
Pin A0 :	4400	455
Pin A0 :	4500	452
Pin A0 :	4600	449

Key interpretation

- Baseline readings stay below threshold.
- Smoke source pushes readings above 400.
- Peak readings reach around 520+ in the recorded samples.
- Reported overall accuracy is 96.5%.

96.5%

Reported prototype accuracy

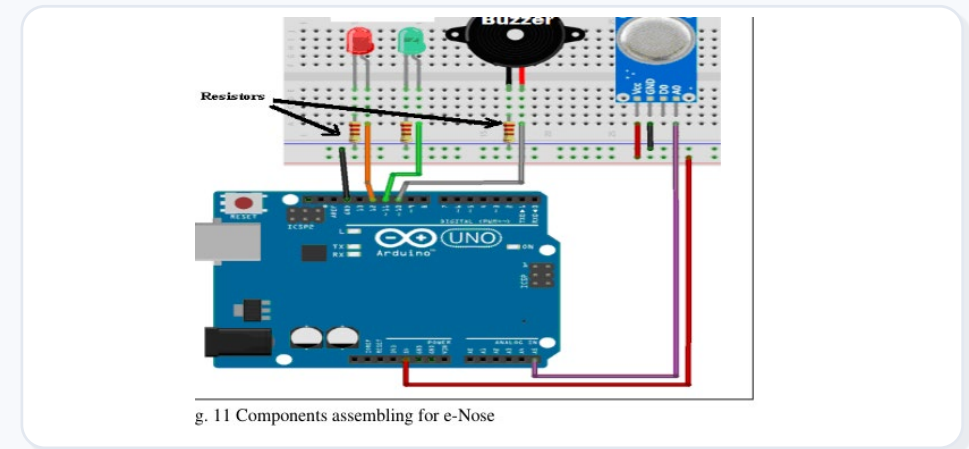
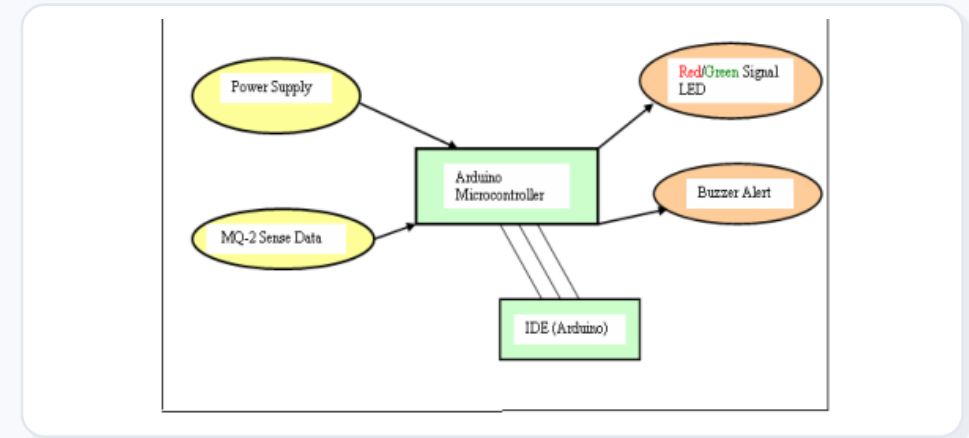
≈ 520+

Peak sample readings

What this work contributes

A practical IoT prototype that converts invisible smoke particles into immediate, visible and audible feedback.

- Privacy-preserving alternative to video-based smoke detection.
- Low-cost components make the system deployable in public places.
- Arduino-based logic is easy to replicate and tune for different thresholds.
- The design can be extended with multiple sensors and machine learning.



Conference presentation & next steps

2nd International Conference on Smart IoT Systems: Innovations in Computing (SSIC 2019)

Future enhancement pathway

- Add multiple gas/environment sensors for richer context.
- Use machine learning to distinguish cigarette smoke from other gases.
- Connect alerts to a cloud dashboard for monitoring and analytics.
- Package the system for reliable deployment in public areas.

The project demonstrates how simple embedded sensing can become a scalable, smart public-health intervention.