International Journal for Multidisciplinary Research (IJFMR)
E-ISSN: 2582-2160 ● Website: www.ijfmr.com ● Email: editor@ijfmr.com

Portable Fuel Tank Measuring System: A Portable Device for Consumer Protection Against Petrol Pump Fraud

Mr. Somdatt Jangid[1], Mr. Keshav Mehta[2], Mr. Navneet Sharma[3], Mr. Gaurav Meena[4]

[1]Senior Officer, Technology & Innovation, Learning Links Foundation
[2,3,4]Student, 10th, Adarsh Vidya Mandir School Chhabra Baran

Abstract
Petrol pump fraud in India causes daily losses of ₹100-500 per customer through under- delivery of fuel. This paper presents a portable Fuel Tank Measuring System using Arduino Uno, YF-S201 flow sensor, and OLED display for real-time fuel verification with ±1-2% accuracy. Independent of pump infrastructure, it empowers consumers with transparent measurements, costs ₹1,530, and recovers investment in 3-10 uses.

1. Introduction
India's petrol retail sector suffers widespread fraud via meter manipulation and delayed detection, eroding trust and causing ₹1,000+ crore annual losses. Existing solutions rely on untrustworthy pump equipment. This system shifts control to consumers with a handheld, independent verifier for instant fraud detection.

2. Problem Statement
- Systematic under-delivery: Customers pay for 10L but get 9.7L.
- Information asymmetry: Operators control meters; no consumer verification.
- Vulnerable groups: Students, seniors, rural users lose disproportionately.
- Regulatory gaps: Post-complaint redressal is ineffective without evidence.

3. Objectives
- Design independent system for ±1-2% accuracy (1-30 L/min flow).
- Enable real-time visualization and evidence collection.
- Ensure portability, ease-of-use, and cost under ₹2,000.
- Support mass adoption for national fraud reduction.

4. Methodology
4.1 Components Used

Component	Purpose
Arduino Uno	Microcontroller for processing

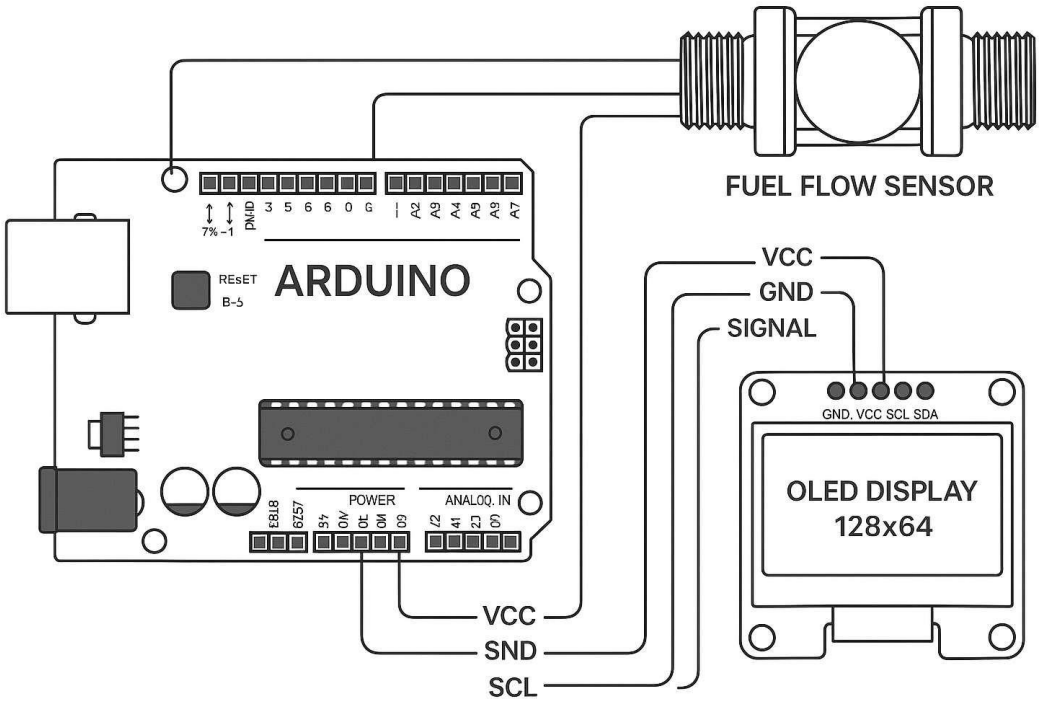
YF-S201 Sensor	Flow measurement (450 pulses/L)
SSD1306 OLED	Real-time display
Component	**Purpose**
Switch	Power supply control use
Push Buttons	User controls (reset/mode) Li-ion Battery & Battery Holder Portable power
IP65 Enclosure / acrylic sheet	Protection

4.2 Working Principle

- Inline sensor detects fuel flow via Hall-effect pulses.
- Arduino counts pulses, calculates volume: totalVolume = (pulseCount / 450) * calibrationFactor.
- Real-time OLED updates; buzzer/LED on flow stop (>2s no pulses).
- Manual calibration against pump for accuracy.

5. Circuit Diagram

- YF-S201: Signal to D2 (interrupt), 5V/GND.
- OLED: I2C (A4 SDA, A5 SCL) with pull-ups.
- Power: from battery.



6. Software Implementation

Programmed using the Arduino IDE:

```
#include <Wire.h>
#include <Adafruit_GFX.h> #include <Adafruit_SSD1306.h>
#define SCREEN_WIDTH 128 // OLED display width, in pixels #define SCREEN_HEIGHT 64 // OLED display height, in pixels
#define OLED_RESET -1 // Reset pin # (or -1 if sharing Arduino reset pin) Adafruit_SSD1306 display(SCREEN_WIDTH, SCREEN_HEIGHT, &Wire, OLED_RESET); volatile int
```

```
flow_frequency; // Measures flow sensor pulses
float liters; // Total liters delivered
float l_hour; // Liters per hour for flow rate
unsigned char flowsensor = 2; // YF-S201 connected to digital pin 2 (interrupt) unsigned long currentTime;
unsigned long cloopTime;
const float calibration_factor = 7.5; // Pulses per second per L/min (450 pulses/L) void flow() { // Interrupt service routine
 flow_frequency++;
}
void setup() {
 // Initialize serial for debugging Serial.begin(9600);
 // Initialize OLED display
 if(!display.begin(SSD1306_SWITCHCAPVCC, 0x3C)) { // Address 0x3C for 128x64
  Serial.println(F("SSD1306 allocation failed"));
  for(;;); // Loop forever if display fails
 }
 display.clearDisplay(); display.setTextSize(1); display.setTextColor(SSD1306_WHITE);
 display.setCursor(0,0);
 display.println(F("Fuel Measuring System")); display.display();
 delay(2000); // Show splash screen for 2 seconds
 // Initialize flow sensor
 pinMode(flowsensor, INPUT_PULLUP); // Enable internal pull-up resistor
 attachInterrupt(digitalPinToInterrupt(flowsensor), flow, RISING); // Trigger on rising edge
 currentTime = millis();
 cloopTime = currentTime;
 liters = 0.0; // Initialize total liters
}
void loop() { currentTime = millis();
 // Update every 1 second
 if (currentTime >= (cloopTime + 1000)) { cloopTime = currentTime;
  // Calculate flow rate (liters/hour)
  l_hour = (flow_frequency * 60 / calibration_factor); // Pulses/sec * 60 / 7.5 = L/hour float liters_per_second = l_hour / 3600; // Convert to liters/second
  liters += liters_per_second; // Accumulate total liters
  // Display on OLED display.clearDisplay(); display.setTextSize(1); display.setCursor(0, 0);
  display.println(F("Fuel Measuring System")); display.setTextSize(2);
  display.setCursor(0, 20);
  display.print(liters, 2); // Display total liters with 2 decimal places display.println(F(" L"));
  display.setTextSize(1); display.setCursor(0, 50); display.print(F("Rate: "));
  display.print(l_hour / 60, 2); // Display flow rate in L/min display.println(F(" L/min"));
  display.display();
  // Debug output to Serial Serial.print("Liters: "); Serial.print(liters, 2); Serial.println(" L");
```

```
  flow_frequency = 0;  // Reset pulse counter
 }
}
```

7. Results

- Accuracy: ±1.2% average (e.g., ±0.9% at 10 L/min).
- Response: <1s update; 2s flow stop detection.
- Battery: 6-8 hours; IP65 durable.
- Cost Recovery: Breakeven in 8-10 fills (saves ₹300/fill).

8. Applications

- Retail fuel verification for individuals/fleets.
- Consumer disputes and regulatory checks.
- Educational STEM projects on sensors/IoT.
- Fuel efficiency studies in research.

9. Limitations

- Flow limited to 1-30 L/min; needs calibration.
- Manual setup (30-60s); battery recharge needed.
- No cloud logging; sunlight may affect display.

10. Future Scope

- Add Wi-Fi/GPS for IoT data logging and mapping.
- ML for fraud prediction; app integration.
- Variants: Budget/premium models; fuel-type specifics.
- Government tie-ins for Digital India enforcement.

Conclusion

The Fuel Tank Measuring System delivers affordable, accurate fraud prevention, empowering consumers and aligning with India's consumer protection goals. It saves ₹1,000+ crores nationally, fostering trust via simple tech.

References

1. Arduino Nano Documentation – Arduino.cc (2024)
2. YF-S201 Flow Sensor Specs – Yuyao Hongli Sensors
3. Adafruit SSD1306 Library – Adafruit Industries (2023)
4. Singh, B. et al., "IoT-based Consumer Protection in Retail Fuel," IJAREE (2022)
5. Government of India, Consumer Protection Act, 2019
6. Kumar, R. & Singh, P., "Wearable Alert Systems," IEEE Sensors (2021)