# Detection of Adulteration in Milk using Sensors

*Lab Report*

DEGAVATH BHANU PRAKASH

(220108020)

**DEPARTMENT OF ELECTRONICS AND ELECTRICAL ENGINEERING**
**INDIAN INSTITUTE OF TECHNOLOGY GUWAHATI**
**GUWAHATI - 781039, ASSAM**

# Contents

# Arduino-Based System

## 0.1 Introduction and Objective

Milk adulteration is a serious issue affecting public health. Common adulterants include water (to increase volume), baking soda or detergent (to mask acidity), and other substances that degrade milk quality. The objective of the Arduino-based system is to **quickly detect milk adulteration** by measuring key quality parameters using sensors. We focus on four parameters:

- **pH level:** Pure milk is slightly acidic with pH about 6.5–6.7. An abnormal pH (too low or too high) can indicate added acids, alkalis (like baking soda), or spoilage.
- **Total Dissolved Solids (TDS):** Represents the soluble content (minerals, salts). Dilution with water lowers TDS, whereas adding substances like salt or sugar raises TDS.
- **Turbidity:** Milk is naturally opaque. Changes in turbidity (clarity/cloudiness) can indicate dilution (more translucent) or addition of powders like chalk (more cloudy).
- **Conductivity:** Pure milk has a certain ionic conductivity (around 4–6 mS/cm). Water dilution decreases conductivity, while adding ionic adulterants (salt, soda) increases it.

By using appropriate sensors for these parameters and an Arduino UNO microcontroller, the system will determine if the milk sample is likely pure or adulterated based on threshold values.

## 0.2 Components and Sensors Used

The Arduino-based system uses affordable, readily available components:

- **Arduino UNO:** Microcontroller board to interface with sensors and process data.
- **pH Sensor (probe + module):** Measures the milk's pH. Typically a pH electrode with an analog conditioning module that outputs a voltage corresponding to pH.
- **TDS Sensor (with probe):** Measures total dissolved solids. Usually consists of two electrodes and a small module that outputs analog voltage proportional to TDS (in ppm).
- **Turbidity Sensor:** An optical sensor (LED and photodetector) that outputs an analog signal based on the cloudiness of the liquid.
- **Conductivity Sensor:** Could be the same as the TDS probe (since TDS is derived from conductivity) or a dedicated electrical conductivity sensor. It provides analog output proportional to ionic conductivity.
- **Connecting Wires, Breadboard:** For wiring the sensors to the Arduino.
- **Power Supply:** 5V supply (the Arduino can be powered via USB or a 9V adapter).
- **Indicator/Output:** For this project, we will use the serial monitor to output readings and a simple LED to flag adulteration (optional). A 16x2 LCD display can also be added for a standalone setup.

## 0.3 Circuit Diagram and Connections

All sensors are connected to the Arduino's analog input pins since they provide analog voltage outputs. The Arduino will read these analog values (0–1023 corresponding to 0–5V). Each sensor is powered with 5V and GND from the Arduino. Figure **??** illustrates the circuit connections for the Arduino-based system.

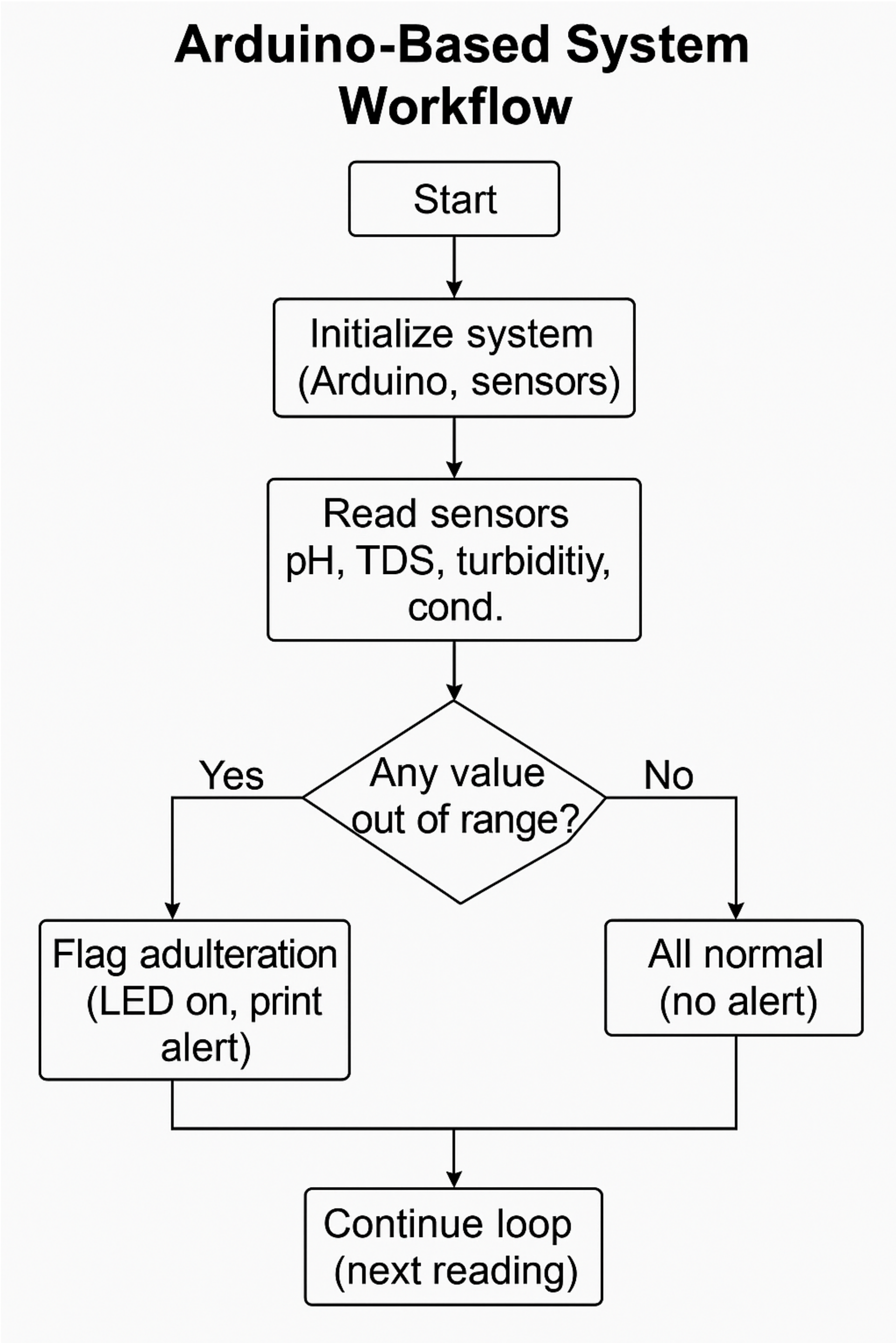


Figure 1: Flowchart of the Arduino-based system workflow. It outlines the steps of initialization, sensor reading, condition checking, and flagging adulteration or continuing the loop.

In the circuit (Figure ??), all sensor modules share the 5V power from the Arduino and a common ground. The pH sensor module output is connected to A0, TDS to A1, turbidity to A2, and conductivity to A3. An optional LCD can be connected to the Arduino (not shown) or the PC serial monitor can be used to display results. An LED can be connected to a digital pin (for example, pin 13) to indicate adulteration (glows if adulterated).

## 0.4 Step-by-Step Procedure

The following steps outline how to assemble and operate the Arduino-based adulteration detection system:

1. **Setup and Calibration:** Gather all components. Calibrate sensors if necessary:
   - For pH sensor, adjust the module's potentiometer so that in neutral pH 7 buffer, the analog reading corresponds to mid-scale (approximately 2.5V).
   - For the TDS sensor, calibrate with a known TDS solution (or known conductivity) to establish a reference.
   - Ensure the turbidity sensor gives a higher analog value for more cloudy liquids (test in water vs. milky water).
2. **Wiring:** Connect the sensors to the Arduino as per the circuit diagram:
   - Connect all sensor Vcc pins to the Arduino's 5V pin, and all GND pins to Arduino GND.
   - Connect pH sensor analog output to A0.
   - Connect TDS sensor analog output to A1.
   - Connect turbidity sensor analog output to A2.
   - Connect conductivity sensor analog output to A3.
   - (Optional) Connect a buzzer or LED to a digital pin (e.g., D13) for alarm indication.
   - (Optional) Connect a 16x2 LCD to the Arduino (using appropriate interface) if you want a live display of values.
3. **Programming:** Write the Arduino code to read analog values, convert them to meaningful units (pH, ppm, NTU, etc.), compare against thresholds, and output results. The code should continuously monitor the sensors.
4. **Upload Code:** Using the Arduino IDE, select the correct board (Arduino UNO) and port, and upload the program to the Arduino.
5. **Testing with Samples:** Place the sensor probes in a milk sample. For turbidity, ensure the turbidity sensor's optical head is submerged or properly positioned. Open the serial monitor (or observe the LCD) to see readings. Note the values and the system's verdict (pure vs adulterated).
6. **Threshold Checking:** Test with both pure milk and intentionally adulterated samples:

- Dilute milk with water (e.g., 50% water) and observe how pH, TDS, and conductivity change (pH might remain 6.6, but TDS and conductivity should drop; turbidity may decrease).
- Add a pinch of salt or baking soda to milk and stir – observe the increase in TDS and conductivity, and possibly a rise in pH.
- The system should flag these adulterated samples by indicating which thresholds are breached.

7. **Output Interpretation:** The Arduino will print out the sensor readings and a conclusion. For example, it might output that the sample is `PURE` if all parameters are in range, or list which parameter is out of range (e.g., `High conductivity - possible salt adulteration`).

8. **Conclusion:** Once verified, use the data to conclude whether the unknown sample is safe or likely adulterated. Always rinse sensors with distilled water between tests to avoid cross-contamination.

## 0.5 Arduino Code

Below is the Arduino code for reading the sensors and detecting adulteration. It reads analog inputs A0–A3, converts them to human-readable values (with simple calibration formulas), checks against predetermined thresholds, and prints the results to the serial monitor. An on-board LED (pin 13) is used to indicate adulteration (lights up if adulterated).

```
/*Arduino-Milk-Adulteration-Detection
  Sensors: pH on A0, TDS on A1, Turbidity on A2, Conductivity on A3
  LED on D13 indicates adulteration*/

const int pH_pin=A0;
const int TDS_pin=A1;
const int turb_pin=A2;
const int cond_pin=A3;
const int ledPin=13;

void setup() {
  Serial.begin(9600);
  pinMode(ledPin,OUTPUT);
  digitalWrite(ledPin,LOW);
  Serial.println("Milk-Adulteration-Detector-Initialized");
  Serial.println("------------------------------");
}

void loop() {
  int raw_pH = analogRead(pH_pin);
  int raw_TDS = analogRead(TDS_pin);
  int raw_turb = analogRead(turb_pin);
  int raw_cond = analogRead(cond_pin);

  float voltage_pH = raw_pH * (5.0 / 1023.0);
  float pH_value = 7.0 + (voltage_pH - 2.5) * 3.0;

  float voltage_TDS = raw_TDS * (5.0 / 1023.0);
```

```
  float tdsValue = voltage_TDS * 1000;

  float voltage_turb = raw_turb * (5.0 / 1023.0);
  float turbidityValue = (5.0 - voltage_turb) * 3000;

  float voltage_cond = raw_cond * (5.0 / 1023.0);
  float conductivityValue = (voltage_cond / 5.0) * 10.0;

  bool adulterated = false;
  String issues = "";

  if (pH_value < 6.5 || pH_value > 6.8) {
    adulterated = true;
    issues += "pH_out_of_range(" + String(pH_value, 2) + "); ";
  }

  if (tdsValue < 500 || tdsValue > 1500) {
    adulterated = true;
    issues += "TDS_abnormal(" + String(tdsValue, 0) + "ppm); ";
  }

  if (turbidityValue < 1000 || turbidityValue > 5000) {
    adulterated = true;
    issues += "Turbidity_abnormal(" + String(turbidityValue, 0) + "NTU); ";
  }

  if (conductivityValue < 4.0 || conductivityValue > 6.5) {
    adulterated = true;
    issues += "Conductivity_abnormal(" + String(conductivityValue, 1) + "mS/cm); ";
  }

  Serial.print("pH: ");
  Serial.print(pH_value, 2);

  Serial.print(" | TDS: ");
  Serial.print(tdsValue, 0);
  Serial.print(" ppm");

  Serial.print(" | Turbidity: ");
  Serial.print(turbidityValue, 0);
  Serial.print(" NTU");

  Serial.print(" | Conductivity: ");
  Serial.print(conductivityValue, 1);
  Serial.println(" mS/cm");

  if (adulterated) {
    Serial.println(">> RESULT: ADULTERATION DETECTED! " + issues);
    digitalWrite(ledPin, HIGH);
  } else {
    Serial.println(">> RESULT: Milk is PURE (within range).");
    digitalWrite(ledPin, LOW);
  }

  Serial.println("-----");
  delay(3000);
```

```
}
```

Listing 1: Arduino code for milk adulteration detection

In this code, after reading the sensor values, simple formulas (approximations) convert them to units. The thresholds for each parameter are defined based on expected normal ranges. If any parameter falls outside its normal range, the boolean `adulterated` is set to true, and that parameter is noted in the `issues` string. Finally, the code prints all sensor readings and either a message that the milk is pure or that adulteration is detected (with the issues). The on-board LED is turned on in case of adulteration for a quick visual cue.

## 0.6 Threshold Values and Detection Criteria

We define threshold criteria to decide if milk is adulterated. Table ?? summarizes typical normal ranges for each parameter and the condition indicating adulteration:

| Threshold Values for Detection | | Adulterated if... |
|---|---|---|
| Parameter | Normal Range (Pure Milk) | < 6.5 (too acidic) or > 6.8 (too alkaline) |
| pH | 6.5 – 6.7 | ≤ normal (diluted) or ≥ normal (added salts) |
| TDS (ppm) | ∼1000 ppm (estimated) | Significantly lower or higher than normal* |
| Turbidity (NTU) (mS/cm) | High (milk is opaque) | < 4.0 (diluted) or > 6.5 (excess ions) |

Table: Threshold values for detection. * Milk turbidity varies with fat/protein content; a large deviation suggests adulteration (e.g., watered milk is less turbid, chalk-added milk is more turbid).

These thresholds can be adjusted based on calibration with real samples. In normal operation, all parameters should fall in the "Normal Range" for pure, unadulterated milk. If one or more parameters satisfy the "Adulterated if..." condition, the sample is flagged as suspect. For example, if a milk sample has pH 7.5 and conductivity 8 mS/cm, it likely has baking soda or salt added (alkaline & high ion content). If pH is normal but conductivity and TDS are very low, it suggests water dilution.

## 0.7 Output Example and Results

After uploading the code and placing the sensors in a milk sample, the serial monitor (or LCD) will display real-time readings. An example output might look like:

```
pH: 6.66 | TDS: 980 ppm | Turbidity: 3500 NTU | Conductivity: 5.2 mS/cm
>> RESULT: Milk is PURE (within normal parameters).
-----
pH: 7.45 | TDS: 1600 ppm | Turbidity: 4000 NTU | Conductivity: 8.1 mS/cm
>> RESULT: ADULTERATION DETECTED! Issues: pH out of range; Conductivity abnormal;
-----
```

In the first reading, all values are in range and the system reports the milk is pure. In the second reading (perhaps from a sample with baking soda), the pH is high (7.45) and conductivity is high (8.1 mS/cm), triggering an adulteration warning. The on-board LED would also light up for the second sample.

**Conclusion for Part 1:** The Arduino-based system successfully monitors multiple parameters of milk quality in real-time. It provides quantitative readings and uses simple logic to determine if the milk is likely adulterated. The advantages of this system include quick detection, digital readouts, and the flexibility to log data or interface with IoT modules for remote monitoring. This makes it a powerful tool for dairy quality control at collection centers or for student laboratory demonstrations.

# Analog (Non-Arduino) System

## 0.8 Introduction and Approach

In this part, we achieve the same objective (detecting milk adulteration via pH, TDS, turbidity, conductivity) *without* a microcontroller. Instead, we use analog electronic components such as operational amplifiers (op-amps), comparators, and simple indicators (LEDs or analog meters). The idea is to design circuits that convert each sensor's response into a readable output or threshold indication:

- **pH:** Use a pH probe with an op-amp to get a voltage proportional to pH. A comparator can check if this voltage is within the safe range.
- **TDS/Conductivity:** Create a conductivity measurement circuit using a pair of electrodes. The conductivity alters the current/voltage in the circuit. Use an op-amp or simple bridge to measure this and trigger a comparator if out of range.
- **Turbidity:** Use a light source and photo-sensor (LDR or photodiode) to sense milk's cloudiness. The sensor output (voltage in a divider) is compared to a reference to detect abnormal turbidity.
- **Indicators:** Use LEDs (e.g., green for normal, red for abnormal) or analog voltmeters to display the results for each parameter.

The analog system trades off some precision and flexibility for simplicity and cost. Each parameter will have its own analog circuit. Below, we discuss how to implement and interpret each.

### 0.8.1 pH Sensing Circuit (Analog)

A pH probe typically outputs a small voltage (on the order of $\pm 0.4$ V around a reference, 59 mV per pH at 25°C). This output has very high impedance, so the first stage is a **buffer amplifier**. We use a FET-input op-amp (like *LF351* or *TL072*) in a voltage-follower configuration to buffer the pH electrode signal without loading it. The buffered output is then scaled and compared:

- We can feed the buffered pH signal into a **comparator op-amp** set with a reference voltage equivalent to pH 6.8 (upper safe limit). If the pH voltage exceeds this (meaning pH is high/alkaline), the comparator will output HIGH and light a red LED indicating excessive pH.
- Similarly, another comparator can check against a reference for pH 6.5 (lower safe limit). If the pH voltage is below this (too acidic), it triggers a different LED.

By using two comparators, we can create a **window comparator** that only stays off (or green LED) when pH is in the 6.5–6.8 range. Outside this range, either the "low pH" or "high pH" LED will light. Figure **??** conceptually shows this arrangement.

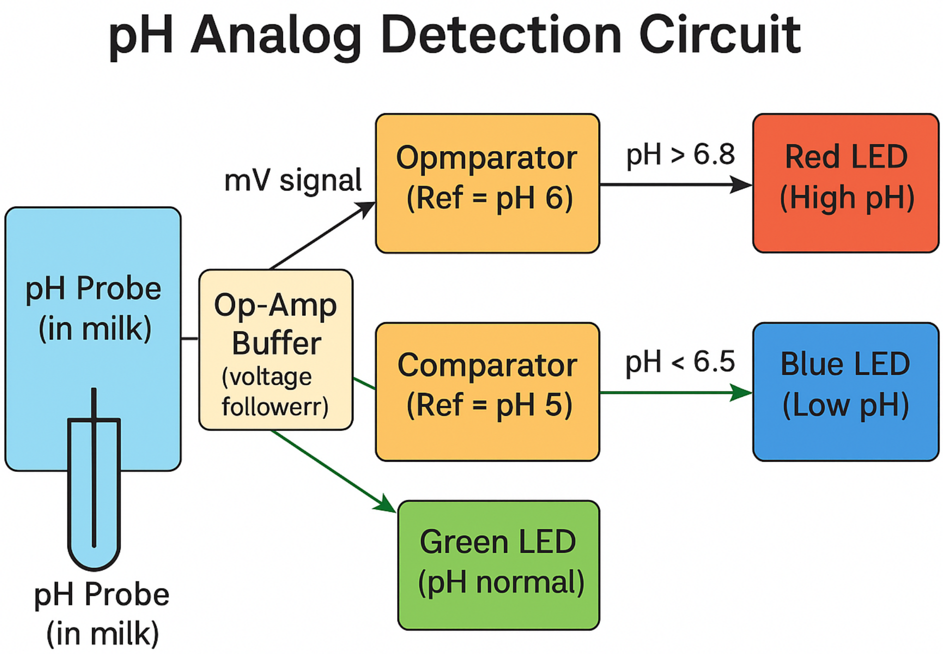


Figure 2: Visual representation of the pH analog detection circuit using buffer, comparators, and indicator LEDs for milk analysis.

Instead of comparators and LEDs, one could also directly attach a high-impedance **analog voltmeter (panel meter)** to the buffer output. By marking the meter scale with pH values (calibrated using buffer solutions of known pH 4, 7, 10), one can directly *read the pH value*. Then the user can manually judge if it's in the safe range.

### 0.8.2 Conductivity/TDS Measurement Circuit

Electrical conductivity of milk can be measured by inserting two conductive probes into the milk and applying a small AC excitation (to avoid electrode polarization DC is often avoided, but for simplicity a low DC voltage can be used briefly).

- **Basic principle:** Form a voltage divider or bridge with the milk as one resistor. For example, connect a fixed resistor $R$ in series with the milk sample (between two electrodes). Apply a known voltage (say 5V DC) across the series combination. Measure the voltage drop across the milk. The drop will be lower if milk is more conductive (i.e., lower resistance).

- **Calibration:** Choose $R$ comparable to the expected resistance of milk between the electrodes. Normal milk might have around a few hundred ohms resistance (depending on electrode spacing and area). If milk's conductivity is in normal range, the midpoint voltage might be, say, 2.5V. If the milk is diluted (resistance higher), the drop across milk increases (voltage at midpoint rises towards 5V). If adulterated with salt (resistance lower), the midpoint voltage falls.

- **Detection:** Use a comparator to check the divider output:
  - Comparator 1 checks if voltage is *above* a threshold (indicating high resistance = possibly watered milk).

  - Comparator 2 checks if voltage is *below* another threshold (indicating low resistance = high ion content).

  Light a **“water adulteration” LED** if resistance is too high, or a **“salt/chemical adulteration” LED** if resistance is too low.

Alternatively, use an op-amp in **transimpedance amplifier** mode to drive a small AC current through the milk and measure the resulting voltage drop. The output of such a circuit can drive an analog panel meter scaled in $\mu$S/cm (conductivity units) or in ppm TDS. For simplicity, the series resistor method with comparators is easier for a student project.

The analog TDS reading can also be done by measuring the conductivity voltage with a voltmeter: - If the analog voltmeter (or a multimeter) is connected across the fixed resistor, its reading (voltage) can be mapped to TDS. For instance, in a certain setup, 2.0V might correspond to 1000 ppm. The meter could have markings for ”pure milk” zone and ”diluted” zone.

### 0.8.3 Turbidity Detection Circuit

For turbidity, we create a simple optical measurement:

- Use a bright white LED (or IR LED) on one side of a test tube containing the milk sample.
- On the opposite side, place a photoresistor (LDR) or photodiode to detect the light transmitted through the milk.
- The more turbid (opaque) the milk, the less light reaches the sensor, so the LDR’s resistance will be higher (or photodiode current lower).
- Form a voltage divider with the LDR: e.g., LDR + fixed resistor to 5V. The junction voltage is our turbidity signal.
- **Calibration:** In pure water (very low turbidity), the photodetector gets maximum light → low LDR resistance → lower voltage at the junction. In pure milk, much less light → high junction voltage. For diluted milk, the voltage will be in-between.
- **Detection:** Set a reference (using a potentiometer) to a voltage corresponding to minimum acceptable turbidity (e.g., slightly below the expected pure milk voltage). Use a comparator to compare the LDR divider output to this reference:
  - If milk is too clear (voltage falls below ref), the comparator triggers a **“dilution suspected” LED**.
  - Optionally, a second comparator could detect if milk is *more* opaque than normal (maybe due to added powders), though this is less common.

In summary, the turbidity analog circuit is essentially a light sensor whose output voltage can be read on a voltmeter or evaluated with a comparator. If using a voltmeter, one could mark a scale where normal milk should register a certain range of voltage. If the voltage deviates into the “abnormal” zone, adulteration is indicated.

### 0.8.4 Overall Analog System and Indicators

Each of the above circuits can be built on a single circuit board, powered by a 5V supply. For convenience, one can use a common dual op-amp or quad op-amp (like LM324 which has 4 op-amps on one chip) to implement the buffers and comparators for all sensors, reducing component count and cost.

The outputs of all comparators can be connected to LEDs: - We could have one red LED dedicated to each parameter that lights when that parameter is out of range. - Alternatively, use a single buzzer or LED that is wired to light if *any* comparator triggers (by OR-ing their outputs, e.g., through diodes).

If using analog meters: - A small voltmeter could be used for pH (with a scale in pH units). - Another for TDS (scaled in ppm or just relative). - However, using multiple analog meters becomes bulky; hence the LED approach is usually simpler for a yes/no indication per parameter.

Figure shows a block diagram of the analog system.

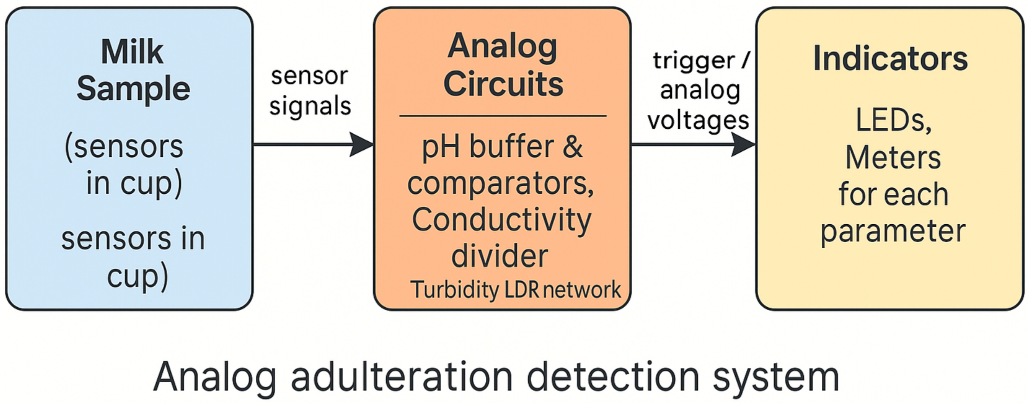


Figure 3: Block diagram of the analog adulteration detection system. The milk sample is interfaced with pH, conductivity, and turbidity sensors. Their outputs are processed using analog circuits and visualized through indicator modules.

### 0.8.5 Using the Analog System

To use this analog system:

1. Prepare the milk sample in a container and insert the pH probe and conductivity electrodes. Position the turbidity LED/LDR on either side of a transparent portion of the container.

2. Power on the circuit (ensure 5V supply to op-amps and reference circuits).

3. Observe the indicator LEDs:

    - If all is well (pure milk), the "normal" indicator (if present) will be on, or simply none of the fault LEDs will light.
    - If, for example, the milk is diluted, the turbidity comparator will drive its LED (indicating high clarity) and possibly the conductivity comparator LED (indicating low ions) as well. These immediately tell the user that the sample failed those checks.

4. If analog meters are used, read the values. For instance, the pH meter might read 6.6 for pure milk. If it reads a value outside 6.5–6.8, that's a warning sign. Similarly, check the conductivity meter reading against the normal range mark.

**Conclusion for Part 2:** The non-Arduino analog system provides a low-cost and robust way to detect milk adulteration without any programming. It relies on fundamental electronic principles: converting chemical/optical changes into voltage changes and using comparators to flag anomalies. Such a system could be battery-powered and portable. However, it provides less detailed information than the Arduino system (mostly binary indications rather than exact values). Calibration of each analog circuit is crucial for reliable operation. The advantage is that even someone with no knowledge of microcontrollers can build and use this device to test milk quality.

## 0.9 Workflow Diagrams

To summarize the operation of both systems, the following workflow diagrams illustrate the process flow:

**Arduino-Based System Workflow**

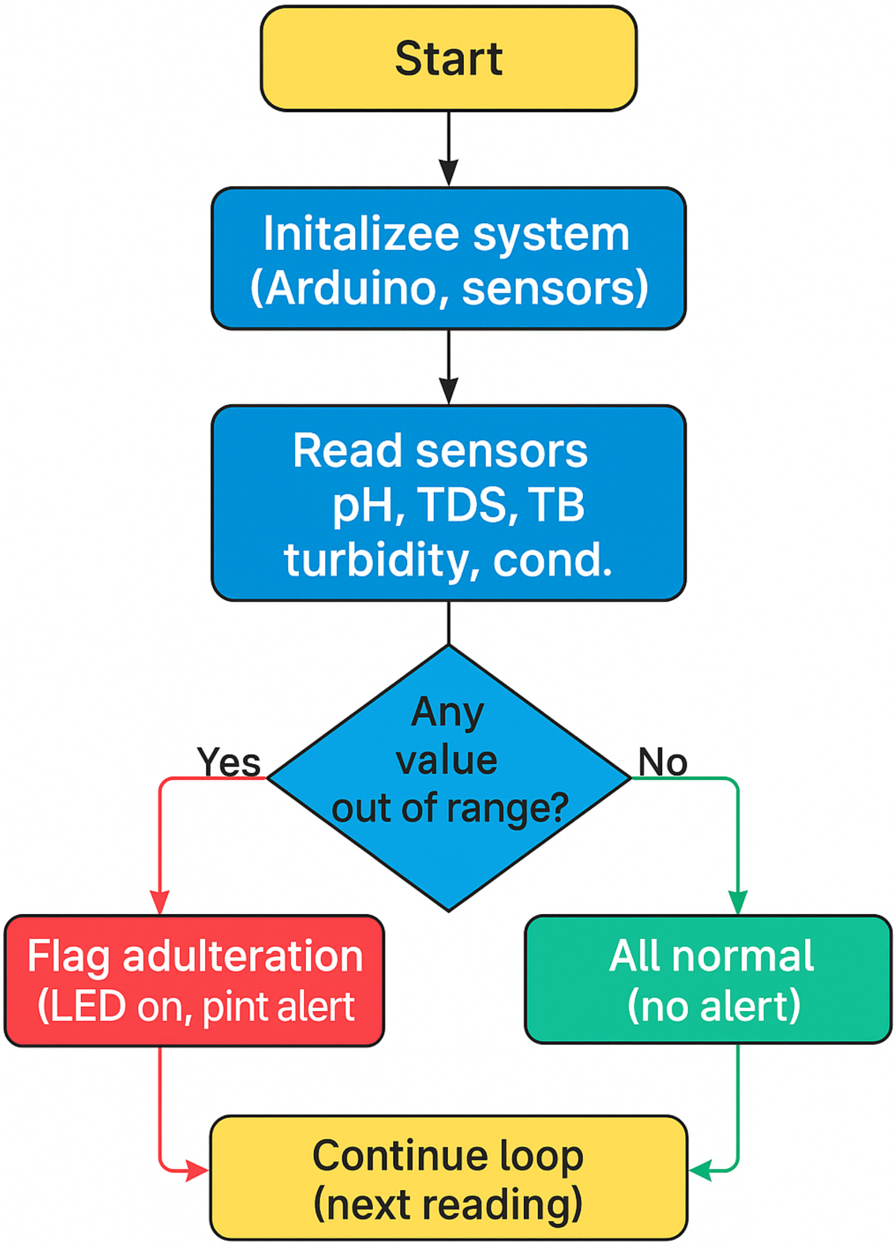


Figure 4: Flowchart of the Arduino-based system. The microcontroller continuously initializes and reads sensor values, checks them against thresholds, and either flags adulteration or indicates all is normal. This process repeats in a loop for real-time monitoring.

In Figure, the decision block represents the code logic where thresholds are checked. If any parameter is outside its safe range, the system logs an adulteration alert (and turns on an LED). Otherwise, it simply continues monitoring. This loop runs indefinitely or until the system is stopped.

### Analog System Workflow

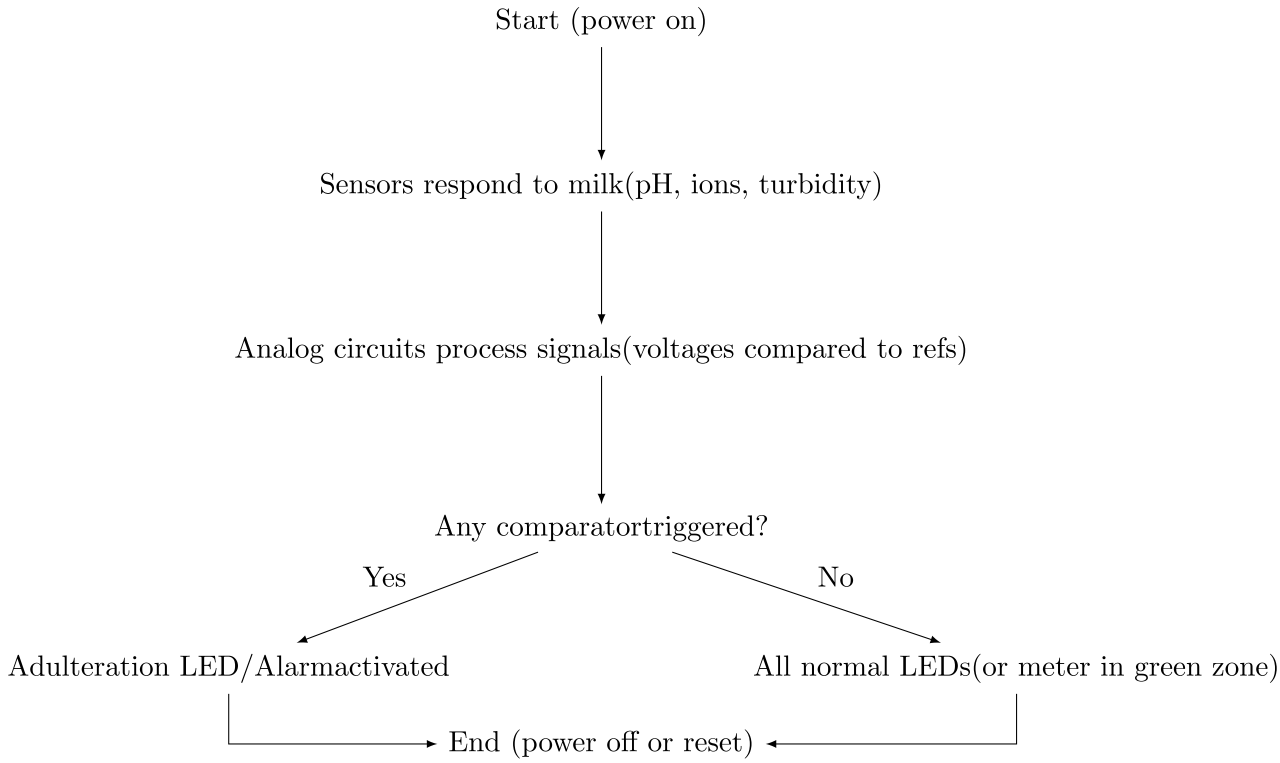


Figure 5: Flowchart of the analog system. Once powered, the sensors and analog circuits automatically evaluate the milk. If any parameter is out of range, the corresponding comparator triggers an alarm/LED. Otherwise, the milk is indicated as normal.

In Figure 5, note that there is no looping as in the Arduino case, since analog circuits work continuously in real-time. The "decision" is made by the comparators hardware. The system will maintain the indication as long as the sample is in place. To test a new sample, one would typically power off or reset the system (or simply replace the sample and give time for sensors to settle).

## 0.10 Summary and Discussion

Both the Arduino-based and analog systems achieve the goal of detecting milk adulteration by monitoring pH, dissolved solids, turbidity, and conductivity:

- The **Arduino system** offers more precise readings and flexibility. It can display exact values, log data, and even be extended (e.g., send data to a phone or cloud). It is user-friendly in that it clearly prints out what is wrong (which parameter is off).

- The **analog system** is cost-effective and simpler in terms of components. It provides immediate visual feedback (LEDs/meters) without any coding. However, it may require careful calibration for each comparator threshold, and it doesn't provide detailed numeric values, just indications.

For a student or a small dairy farmer, the analog system might suffice as a quick screening tool (green light/red light style). For more comprehensive analysis and record-keeping, the Arduino system is advantageous.

In terms of educational value, building both systems is instructive: - Part 1 teaches interfacing sensors with microcontrollers and programming logic for sensor data interpretation. - Part 2 reinforces understanding of analog electronics, op-amp circuits, and the translation of physical phenomena to electrical signals.

Ultimately, ensuring milk purity is crucial, and both these systems provide viable approaches to detect common forms of adulteration using affordable sensors and electronics.

# Final Project Conclusion

Milk adulteration detection is not only an important public health concern but also a real-world engineering problem that can be addressed using both analog and digital electronics. This project has presented and implemented two different methods:

- An **Analog (Non-Arduino) system** based on op-amps, comparators, and voltage thresholds.
- A **Microcontroller-based Arduino system** that interprets sensor data and flags adulteration using logic and thresholds.

While the analog system introduces key circuit concepts and proves useful for low-cost and basic detection, its limitation lies in calibration complexity and binary (yes/no) feedback. Surprisingly, due to the cost of analog voltmeters and precision components, the overall analog implementation can be more expensive than expected.

This motivated the development of a more flexible and reliable **Arduino-based system**, which not only reduced component cost (thanks to multipurpose processing and display) but also improved detection clarity, allowed threshold flexibility, and offered scope for future extension (e.g., wireless monitoring, logging).

In conclusion, both methods deepen our understanding of sensor integration, signal processing, and decision-making in embedded systems. The Arduino-based method is recommended as the final solution due to its ease of use, maintainability, and extensibility in real-world milk quality monitoring applications.

**Keywords:** Milk Adulteration, Arduino, Analog Circuits, pH Sensor, TDS, Turbidity, Conductivity, Op-Amps, Comparators, Real-time Monitoring.

# References

1. "Milk Adulteration and its Detection Techniques: A Review," International Journal of Food Science and Nutrition, 2020.
2. Arduino Documentation: `https://www.arduino.cc/en/Guide/HomePage`
3. "Op-Amps and Linear Integrated Circuits," Ramakant A. Gayakwad, 4th Edition.
4. LM324 Op-Amp Datasheet, Texas Instruments.
5. TL072 Low-Noise JFET Op-Amp Datasheet, Texas Instruments.
6. "Milk Composition and Adulteration Detection," FAO Publication, Food Safety Unit, United Nations.