Problem Statement:

Design a microcontroller-based system that alerts the user when the sludge level in a water tank exceeds a certain threshold, indicating the need for cleaning.

Scope of the Solution:

Boundaries:

- The solution focuses on creating a prototype to monitor the sludge level in a water tank.
- The scope does not include the automation of the tank cleaning process itself; it only alerts the user when cleaning is required.

Objectives:

- To design a reliable and cost-effective system for monitoring the sludge level in a water tank.
- To provide a visual alert to the user when the sludge level exceeds a predefined threshold.
- To interface the system with an Arduino microcontroller.
- To enable user-friendly customization of the sludge threshold value.

Solution Aims:

- Alert the user when the sludge level in the water tank exceeds a specified threshold.
- Provide real-time feedback on the sludge level through an LED indicator.

Constraints or Limitations:

- The accuracy of the system depends on the precision of the sensor, and it may not be suitable for very large tanks or tanks with irregular shapes.
- The system is limited to a visual alert (LED) and does not provide remote notifications by default.
- It assumes that the tank's conditions, such as lighting and water quality, do not interfere with the sensor's performance.
- The prototype is limited to the capabilities of an Arduino microcontroller.
- Users should adjust the sludge threshold value manually.

Required Components:

Hardware Components:

1. Arduino board (Arduino Uno R3)
2. Infrasonic sensor
3. LED
4. Resistor
5. Breadboard and jumper wires
6. Power supply for Arduino

Software Components:

1. Arduino IDE

2. Embedded C

Information about Hardware Components and Specialized Equipment:

1. Arduino Board (e.g., Arduino Uno): The central component of the project, serving as the microcontroller. It contains a microcontroller chip, input/output pins, and a USB interface for programming. An USB cable is required to connect it to your computer.

2. Infrasonic Sensor: The sensor typically consists of two main parts: a transmitter (ultrasonic sender) and a receiver (ultrasonic receiver). It emits ultrasonic sound waves and measures the time taken for the sound to bounce back from an object, which is then used to calculate the distance. It has four pins: VCC (power supply), GND (ground), TRIG (trigger), and ECHO (echo). Ensure you understand the sensor's datasheet for proper wiring and usage.

3. LED and Resistor: LEDs have two leads, an anode (longer lead, positive) and a cathode (shorter lead, negative). A current-limiting resistor is connected in series with the LED to prevent it from drawing too much current from the Arduino pin.

4. Breadboard and Jumper Wires: Used for creating temporary circuits. Jumper wires are used to connect components on the breadboard.

5. Power Supply for Arduino: Power the Arduino through the USB connection from the computer or by using an external power source (e.g., a battery or an adapter).

Code:

Language- C++

```
const int topSensorPin = 2;      // IR sensor at the top of the tank
const int bottomSensorPin = 3;   // IR sensor at the bottom of the tank
const int LED = 4;               // LED for the alert
const int sludgeThreshold = 20;  // Threshold distance in centimeters
const int triggerDistance = 50;  // Distance to trigger the alert

void setup() {
  pinMode(topSensorPin, INPUT);
  pinMode(bottomSensorPin, INPUT);
  pinMode(LED, OUTPUT);
  Serial.begin(9600);
}

int measureSludgeDistance(int sensorPin) {
  // Variables to hold the sensor values
  int sensorSum = 0;
  int sensorCount = 10;  // Number of readings

  for (int i = 0; i < sensorCount; i++) {
    int sensorValue = digitalRead(sensorPin);
    sensorSum += sensorValue;
    delay(10);  // Delay between readings
  }

  // Calculate the average sensor value
  int averageSensorValue = sensorSum / sensorCount;
  int distance = map(averageSensorValue, 0, 1023, 0, 50);
```

```
  return distance;
}

void loop() {
  int topSensorValue = digitalRead(topSensorPin);
  int bottomSensorValue = digitalRead(bottomSensorPin);

  if (bottomSensorValue == LOW && topSensorValue == HIGH) {
    int sludgeDistance = measureSludgeDistance(bottomSensorPin);
    Serial.print("Sludge level distance: ");
    Serial.print(sludgeDistance);
    Serial.println(" cm");

    if (sludgeDistance <= triggerDistance) {
      // Sludge level has crossed the threshold
      digitalWrite(LED, HIGH); // Turn on the LED
      Serial.println("Sludge level is high. Clean the tank!");
    } else {
      digitalWrite(LED, LOW); // Turn off the LED
      Serial.println("Sludge level is normal.");
    }
  } else {
    digitalWrite(LED, LOW); // Turn off the LED
    Serial.println("Sludge level is normal.");
  }

  delay(1000); // Delay
}
```

Simulated circuit:

Components-

1. Arduino Uno R3
2. IR Sensor
3. LED
4. Resistor
5. Wires

Functionality:

The microcontroller checks for the level of sludge based on the defined threshold value. If the sludge level is below the lower IR sensor, or if it is between the lower and the upper sensors, The sludge level is normal and the same is printed. When it crosses the upper sensor, the LED glows and cleaning alert is given.