#include <Wire.h>

#include <LiquidCrystal\_I2C.h>

#define flowSensorPin 2 // Flow sensor signal pin

#define buzzerPin 8 // Buzzer pin

#define ledPin 9 // LED pin (optional alert)

#define timeLimit 120 // Time limit in seconds (2 minutes = 120 seconds)

#define alertDuration 15000 // Alert duration in milliseconds (15 seconds)

#define calibrationFactor 270 // Number of pulses per liter for the flow sensor

// Variables

volatile int flowPulseCount = 0;

unsigned long startTime;

unsigned long alertStartTime;

bool showerStarted = false;

bool alertActive = false;

float waterUsed = 0.0; // Total water used in liters

// Initialize the LCD with the I2C address (usually 0x27)

LiquidCrystal\_I2C lcd(0x27, 16, 2);

void setup() {

// Initialize serial communication

Serial.begin(9600);

// Initialize the LCD

lcd.init();

lcd.backlight(); // Turn on the LCD backlight

lcd.clear();

// Set pin modes

pinMode(flowSensorPin, INPUT);

pinMode(buzzerPin, OUTPUT);

pinMode(ledPin, OUTPUT);

// Attach interrupt to the flow sensor

attachInterrupt(digitalPinToInterrupt(flowSensorPin), countFlowPulse, RISING);

// Start with buzzer and LED off

digitalWrite(buzzerPin, LOW);

digitalWrite(ledPin, LOW);

}

void loop() {

// Check if water flow has started

if (flowPulseCount > 0 && !showerStarted) {

showerStarted = true;

startTime = millis(); // Start the timer

Serial.println("Water flow detected. Shower started.");

}

// Check if water flow has stopped

if (showerStarted && flowPulseCount == 0) {

resetSystem(); // Reset system if flow stops

}

// Calculate elapsed time and display if shower is ongoing

if (showerStarted && !alertActive) {

unsigned long elapsedTime = (millis() - startTime) / 1000; // Time in seconds

// Calculate water used in liters

waterUsed = (flowPulseCount / calibrationFactor);

Serial.print("Elapsed time: ");

Serial.println(elapsedTime);

Serial.print("Water used: ");

Serial.print(waterUsed);

Serial.println(" L");

// Update LCD with elapsed time and water used

lcd.setCursor(0, 0);

lcd.print("Time: ");

lcd.print(elapsedTime);

lcd.print("s");

lcd.setCursor(0, 1);

lcd.print("Liters: ");

lcd.print(waterUsed, 2); // Display water with two decimal places

// Trigger alert if time limit is exceeded

if (elapsedTime >= timeLimit) {

activateAlert(); // Start the alert

}

}

// Check if the alert is active and handle the alert duration

if (alertActive && (millis() - alertStartTime) >= alertDuration) {

deactivateAlert();

}

delay(1000); // Wait for 1 second before the next loop iteration

}

// Function to count pulses from the flow sensor (interrupt service routine)

void countFlowPulse() {

flowPulseCount++;

}

// Function to reset system if water flow stops before time limit

void resetSystem() {

Serial.println("Flow stopped. Resetting system.");

showerStarted = false;

flowPulseCount = 0;

waterUsed = 0.0;

lcd.clear();

deactivateAlert(); // Ensure alert is turned off

}

// Function to activate the buzzer and LED alert

void activateAlert() {

alertStartTime = millis(); // Record the time when the alert started

alertActive = true;

tone(buzzerPin, 1000); // Play a tone at 1000 Hz

digitalWrite(ledPin, HIGH); // Turn on LED

Serial.println("Time limit exceeded! Stop shower.");

}

// Function to deactivate the buzzer and LED alert after 15 seconds

void deactivateAlert() {

noTone(buzzerPin); // Turn off the tone

digitalWrite(ledPin, LOW); // Turn off LED

alertActive = true; // Alert stays active until reset

}