

```

#include <HX711.h>
#include <LiquidCrystal.h>
#include <Keypad.h>

// Initialize LCD
LiquidCrystal lcd(1, 0, 2, 3, 4, 5); // Creates an LC object. Parameters: (rs, enable, d4,
d5, d6, d7)

// Initialize Load Cell
const int loadCellDoutPin = A2; // Dout pin for the load cell
const int loadCellSckPin = A1; // Sck pin for the load cell
HX711 scale; // Create an HX711 object for load cell interfacing

// Initialize Keypad
const byte ROWS = 4; // Number of rows in the keypad
const byte COLS = 3; // Number of columns in the keypad
char keys[ROWS][COLS] = { // Define the layout of the keypad buttons
    {'1', '2', '3'},
    {'4', '5', '6'},
    {'7', '8', '9'},
    {'*', '0', '#'}
};
byte rowPins[ROWS] = {6, 7, 8, 9}; // Pins connected to keypad rows
byte colPins[COLS] = {10, 11, 12}; // Pins connected to keypad columns
Keypad keypad = Keypad(makeKeymap(keys), rowPins, colPins, ROWS, COLS); // Create a Keypad
object

// Initialize Relay and Push Button
const int Motor_Relay = A0; // Pin connected to the motor relay
const int Buzzer_Relay = A4; // Pin connected to the buzzer relay
const int Buzzer_LED = A5; // Pin connected to the buzzer relay
const int PushButton = A3; // Pin connected to the push button
boolean filling = false; // Flag to indicate if the filling process is active
boolean targetConfirmed = false; // Flag to indicate if the target weight is confirmed

// Variables
int targetWeight = 0; // Store the target weight entered by the user
int currentWeight = 0; // Store the current weight measured by the load cell

void setup() {
    lcd.begin(16, 2); // Initialize a 16x2 LCD display

    scale.begin(loadCellDoutPin, loadCellSckPin); // Initialize the load cell

    pinMode(Motor_Relay, OUTPUT); // Set motor relay pin as an output
    pinMode(Buzzer_Relay, OUTPUT); // Set buzzer pin as an output
    pinMode(Buzzer_LED, OUTPUT); // Set buzzer LED pin as an output
    pinMode(PushButton, INPUT_PULLUP); // Set push button pin as an input with internal pull-up

    // Display initial message

```

```

lcd.clear(); // Clear the LCD screen
lcd.print("Enter Target(g):"); // Display the initial message
lcd.setCursor(0, 1); // Move to the second line of the LCD
}

void loop() {
  char key = keypad.getKey(); // Read the current pressed key from the keypad

  // Check if target weight is not confirmed yet
  if (!targetConfirmed) {
    if (key != NO_KEY) {
      // Process numeric keys for target weight entry
      if (key >= '0' && key <= '9') {
        targetWeight = targetWeight * 10 + (key - '0');
        lcd.print(key); // Display the entered digit on the LCD
      }
      // Confirm target weight with '*' key
      else if (key == '*') {
        lcd.clear();
        lcd.print("Target:");
        lcd.print(targetWeight);
        lcd.print(" g"); // Add ' g' for grams
        lcd.setCursor(0, 1);
        lcd.print("Press Button.."); // Display instructions
        targetConfirmed = true; // Confirm the target weight
      }
      // Reset the program with '#' key
      else if (key == '#') {
        resetProgram(); // Call the resetProgram function
      }
    }
  } else { // Target weight confirmed
    if (key == '#') {
      resetProgram();
    }
  }

  // Read current weight from load cell
  if (digitalRead(PushButton) == LOW && !filling && targetConfirmed) {
    filling = true; // Start the filling process
    lcd.clear();
    lcd.print("Filling...");
    while (filling) { // Loop while filling is active
      currentWeight = scale.get_units(10); // Read current weight from load cell
      digitalWrite(Motor_Relay, HIGH); // Turn on the motor relay
      lcd.setCursor(0, 1);
      lcd.print("Current:");
      lcd.print(currentWeight);
      lcd.print(" g");
      if (currentWeight >= targetWeight) {

```

```

        stopFilling(); // Call the stopFilling function if target weight is reached
    }
}
}

// Reset the program variables and LCD display
void resetProgram() {
    filling = false; // Stop the filling process
    targetConfirmed = false; // Reset the target confirmation
    targetWeight = 0; // Reset target weight
    currentWeight = 0; // Reset current weight
    lcd.clear();
    lcd.print("Enter Target(g):");
    lcd.setCursor(0, 1);
}

// Stop the filling process
void stopFilling() {
    filling = false; // Stop the filling process
    digitalWrite(Motor_Relay, LOW); // Turn off the motor relay
    lcd.clear();
    lcd.print("Done Filling!"); // Display completion message
    digitalWrite(Buzzer_Relay, HIGH); // Turn on the buzzer
    digitalWrite(Buzzer_LED, HIGH); // Turn on the buzzer
    delay(3000); // Wait for 3 seconds
    digitalWrite(Buzzer_Relay, LOW); // Turn off the buzzer
    digitalWrite(Buzzer_LED, LOW); // Turn on the buzzer
    lcd.clear();
    lcd.print("Target:");
    lcd.print(targetWeight);
    lcd.print(" g");
    lcd.setCursor(0, 1);
    lcd.print("Press Button..");
    targetConfirmed = true; // Confirm the target weight
}

```