Group 4

Chilaka Anand Sunhith CH23B005

Dharmesh S CH23B008

Munagala Kedhar Reddy CH23B024

Rushikesh Sanjayrao Amale CH23B033

Velavali Shaik Mohammad Aakhib CH23B040

Feedback Control of Brine Solution Concentration using Arduino

Objective:

The objective of this project is to design and implement a **feedback control system** to achieve a desired salt concentration in a mixing beaker using Arduino. The system will use two sources: pure water and a concentrated salt solution. A conductivity/TDS sensor will provide real-time concentration feedback, which will be used by Arduino to control pumps/liquid flow, maintaining the solution at the setpoint concentration.

Experimental Plan:

- **Setup**:
  - Beaker 1: Pure water (dilution source).
  - Beaker 2: Concentrated NaCl solution (concentration source).
  - Beaker 3: Mixing tank with stirrer and TDS sensor.
  - Arduino controls pumps from Beaker 1 and Beaker 2.

➢ **Procedure**:

- User sets desired concentration (setpoint) in Arduino code.
- Arduino reads concentration in Beaker 3 via TDS sensor.
- Controller logic (On–Off or PID) adjusts pumps:
  - If concentration < setpoint → add from Beaker 2.
  - If concentration > setpoint → add from Beaker 1.
- Continuous stirring ensures homogeneity.
- System runs until concentration stabilizes around setpoint.

Equipment Required:

Hardware:

- **Microcontroller:** An Arduino board (e.g., Arduino Uno) will serve as the brain of the system. link
- **Sensor:** A **TDS (Total Dissolved Solids) / Conductivity Sensor** to measure the salt concentration in the mixing tank. link (could be available in labs also)
- **Actuators:**
  - Two **peristaltic pumps** with controllable speed or on/off functionality. These are ideal for pumping fluids without contamination. link (2 in quantity)
  - A **motor driver module** (like an L298N) or a **relay module** to allow the Arduino to control the pumps. link
- **Mixing Equipment:**
  - Three **beakers** or containers. (could be taken from lab)
  - A **magnetic stirrer and stir bar** to ensure the solution in the mixing tank is homogeneous. (available in lab)
- **Power Supply:** Separate power supplies for the Arduino and the pumps (as pumps often require more current than the Arduino can provide).

- **Miscellaneous:** Jumper wires and breadboard (link), and tubing for the pumps (link).

Chemicals:

- **Sodium Chloride (NaCl):** Common table salt.
- **Pure Water:** Distilled or deionized water for accurate sensor readings.

Process Block Diagram:

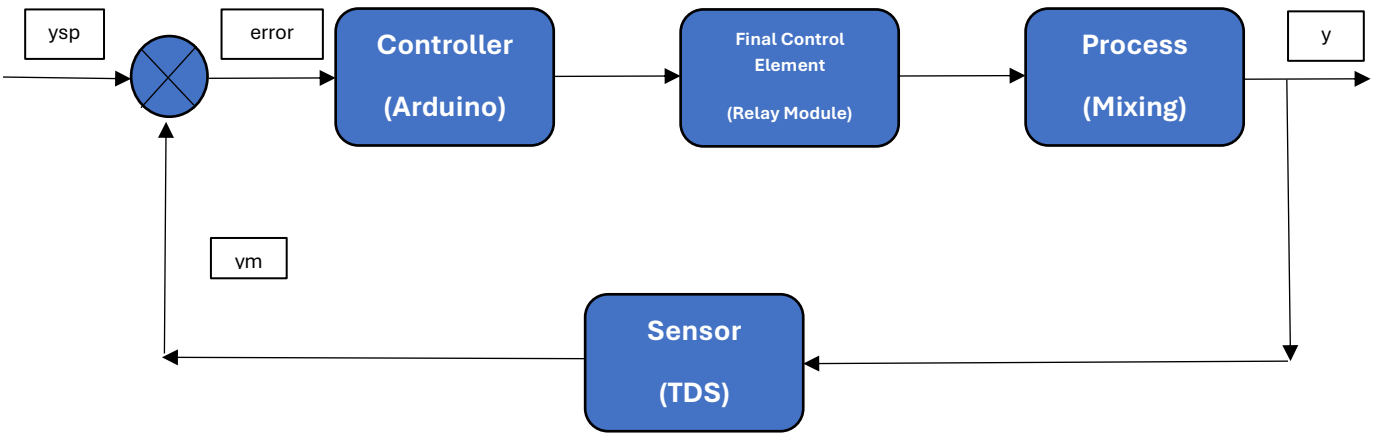
Here is a block diagram illustrating the feedback control loop for your system.

1. **Setpoint (SP):** This is the **desired concentration** you define in the Arduino code. It's the target value the system aims to achieve.
2. **Comparator:** This is a conceptual point within your Arduino's code. It continuously calculates the **Error** by subtracting the measured concentration from the setpoint (Error = SP - CV).
3. **Controller:** The **Arduino** itself. It takes the error signal and, based on its programmed logic (On-Off or PID), decides what action to take.
4. **Final Control Element (Actuators):** These are the **two pumps**. The Arduino sends a signal to the motor driver or relay, which turns the appropriate pump on or off.
5. **Process:** The **mixing beaker** where the pure water and concentrated salt solution are combined. The stirrer is a key part of this process, ensuring a uniform mixture.
6. **Controlled Variable (CV):** The **actual, real-time concentration** of the brine solution in the mixing tank.
7. **Measuring Element (Sensor):** The **TDS/Conductivity sensor**. It measures the Controlled Variable and sends this information back to the Arduino, closing the feedback loop.

How the Feedback Loop Works:

The system operates in a continuous cycle:

- The **TDS sensor** measures the current salt concentration in the mixing tank.
- The **Arduino** compares this measured value to your desired **setpoint**.
- If the measured concentration is too low (positive error), the controller activates the pump connected to the **concentrated NaCl solution**, increasing the concentration.
- If the measured concentration is too high (negative error), the controller activates the pump connected to the **pure water**, diluting the solution and decreasing its concentration.
- This process repeats, making continuous adjustments until the measured concentration stabilizes at the setpoint value.

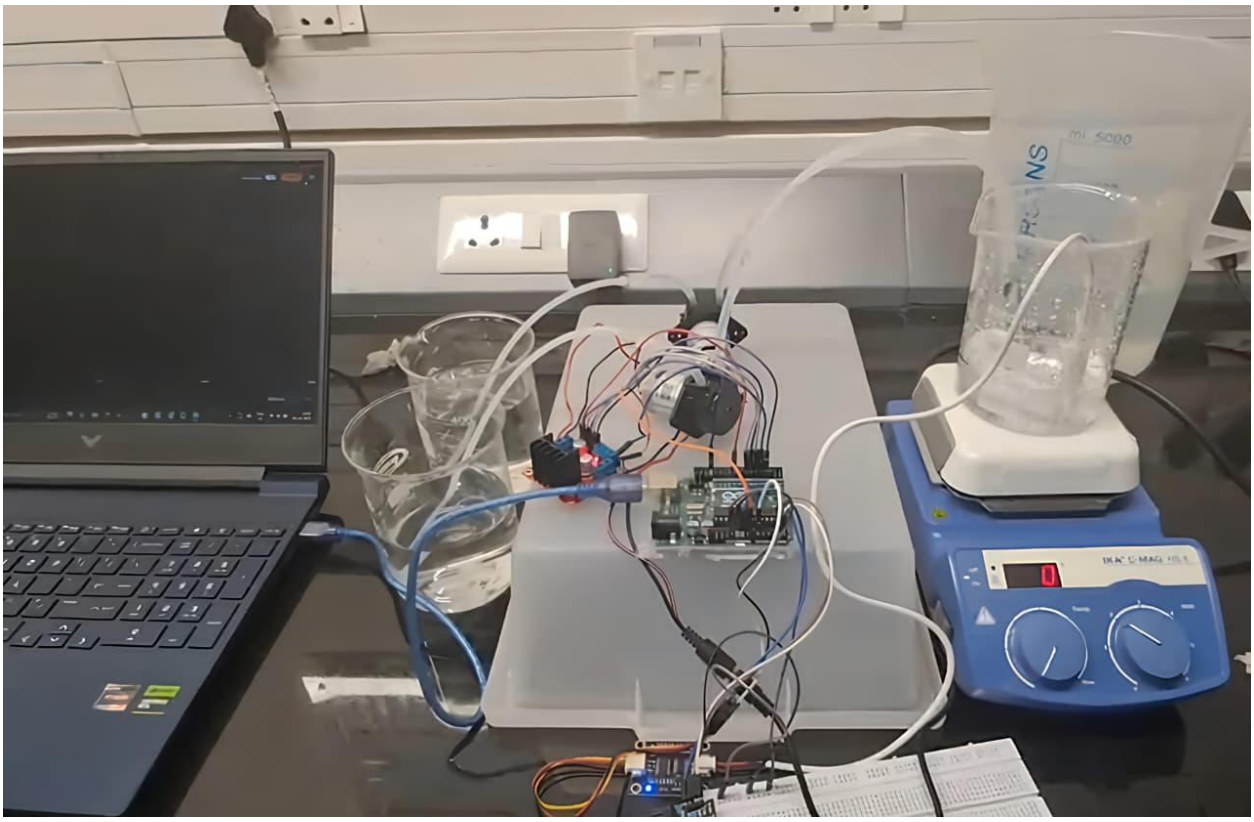


Circuit Connections and Simulation:

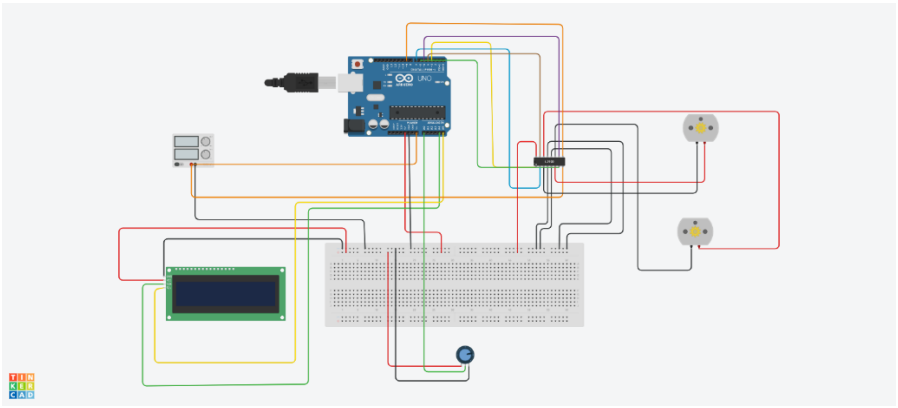
Component	Pin	Connects To
12V 2A Adapter	Positive (+) Wire	L298N +12V Terminal
	Positive (+) Wire	Arduino VIN Pin (This powers the Arduino)
	Negative (-) Wire	L298N GND Terminal (The middle screw terminal)
	Negative (-) Wire	Arduino GND Pin (This is the Common Ground)

TDS Sensor	VCC (or +)	Arduino 5V Pin
	GND (or -)	Arduino GND Pin
	Signal (or A)	Arduino A0
I2C LCD Display	VCC	Arduino 5V Pin
	GND	Arduino GND Pin
	SDA	Arduino A4
	SCL	Arduino A5
L298N Driver	ENA	Arduino D3 (This is for Pump 1 Speed)
	IN1	Arduino D7
	IN2	Arduino D6
	ENB	Arduino D9 (This is for Pump 2 Speed)
	IN3	Arduino D5
	IN4	Arduino D4
Pump 1 (Brine)	Wire 1 & 2	L298N OUT1 & OUT2 Terminals
Pump 2 (Water)	Wire 1 & 2	L298N OUT3 & OUT4 Terminals

Experimental setup:

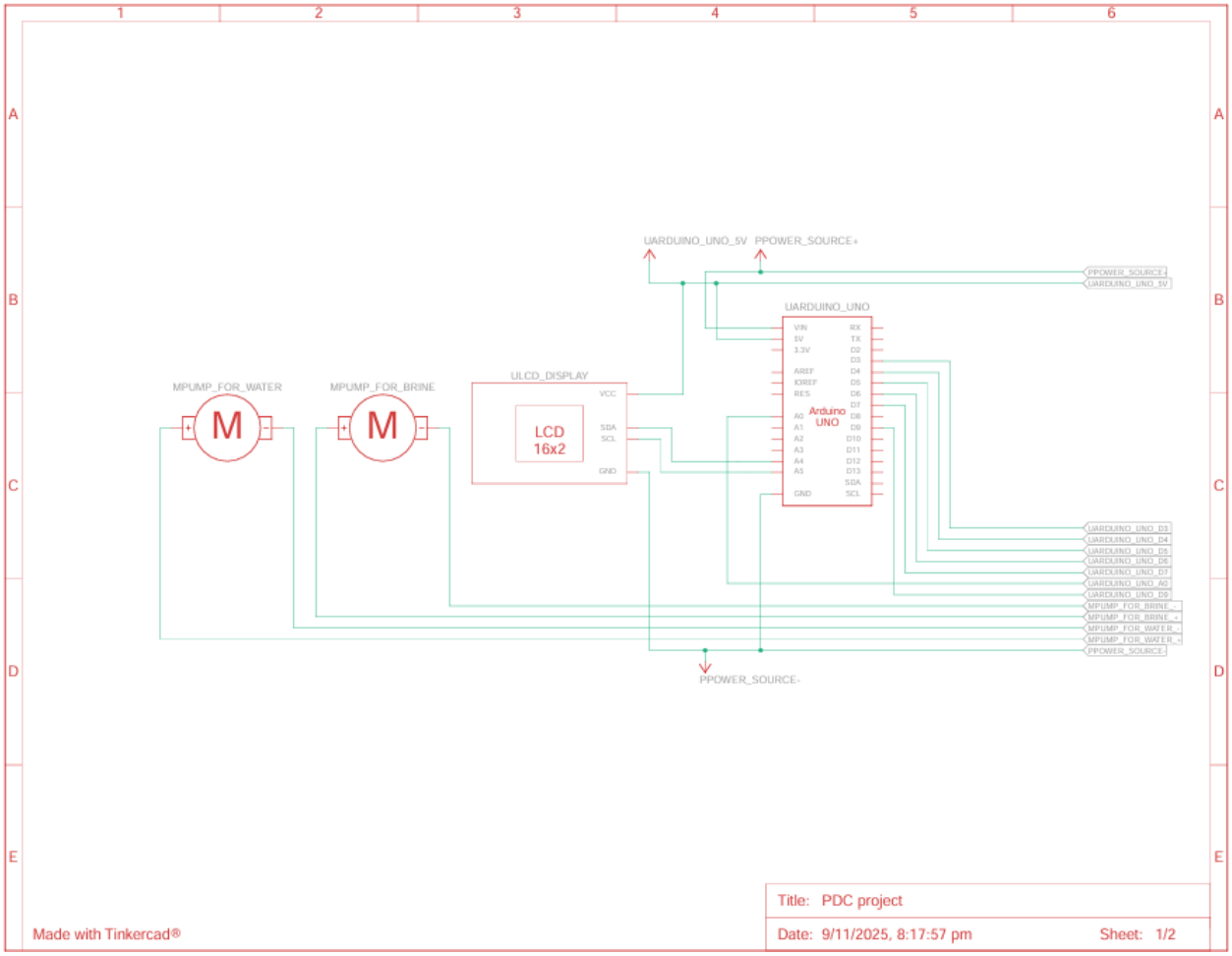


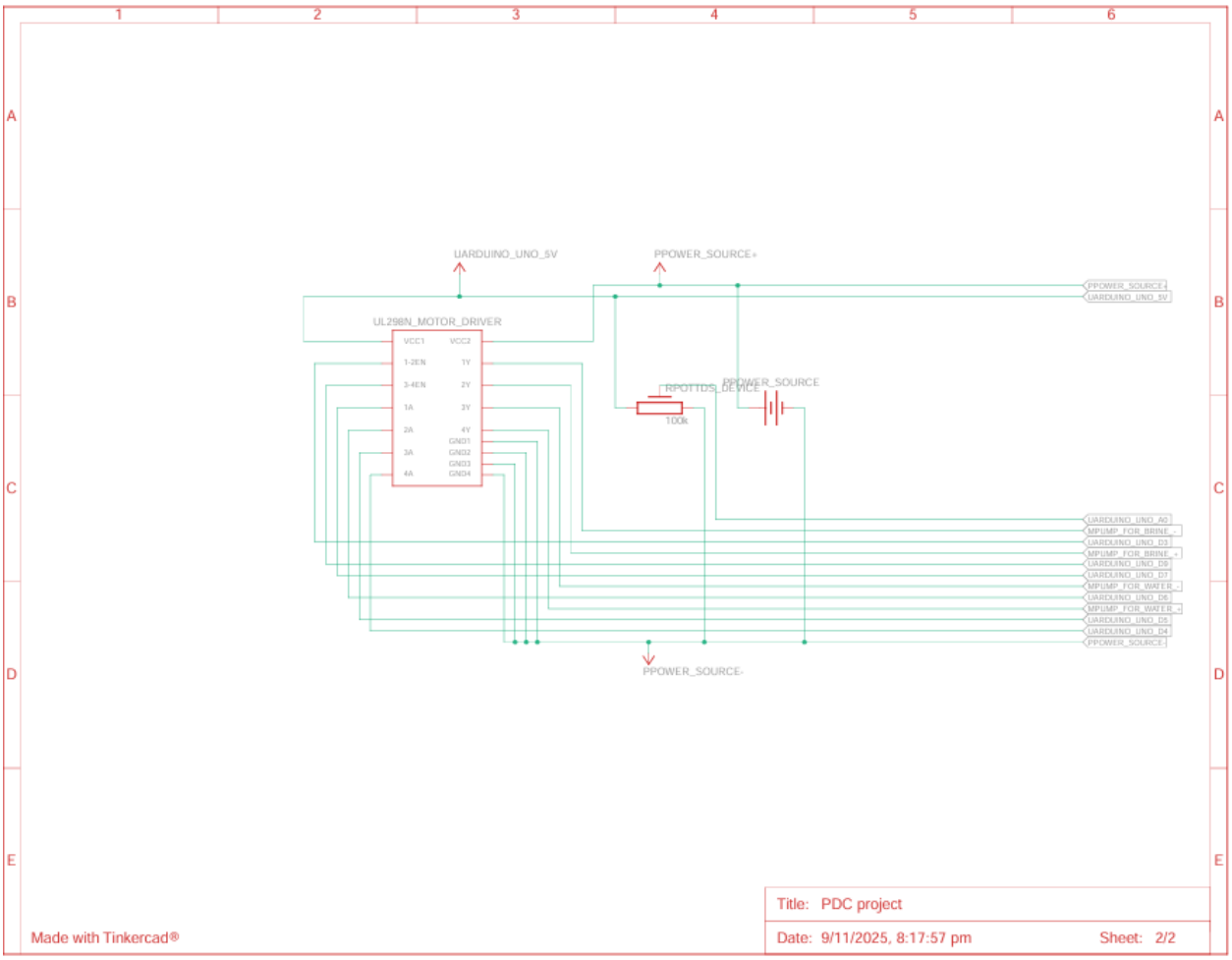
Overall Circuit design done in Tinkercad:



In the above design, as Tinkercad doesn't have components like the TDS sensor, therefore we have used a potentiometer. By varying the resistance in the potentiometer, the measured value changes, and the motor speed adjusts accordingly.

Detailed connections are shown below.





To simulate the process yourself, use this link.

Arduino Code and Things to Note:

To access the Arduino code, look into this GitHub repository.

- The maximum limit that the TDS sensor can measure is around 650 ppm; therefore, it's suggested to have your salt solution at a concentration of around 500 ppm.
- A time delay of 1.5 seconds has been given to the actuation of the motor pumps to ensure that the mixture is well mixed.
- The deadband range used is 10 ppm ((+ or –) 5ppm). If the measured concentration stays in that region for 10 seconds continuously, the motor pumps will be turned off. But the measurement will be continuously taken and if the value is not in the deadband range, the pumps will be turned on again.

Innovative Suggestions:

- Integrate a level sensor for overflow protection and level control.
- Add LED or buzzer alarms for overflow and sensor malfunction.
- Add a temperature sensor for conductivity compensation.

Guide: Dr. Nabil M