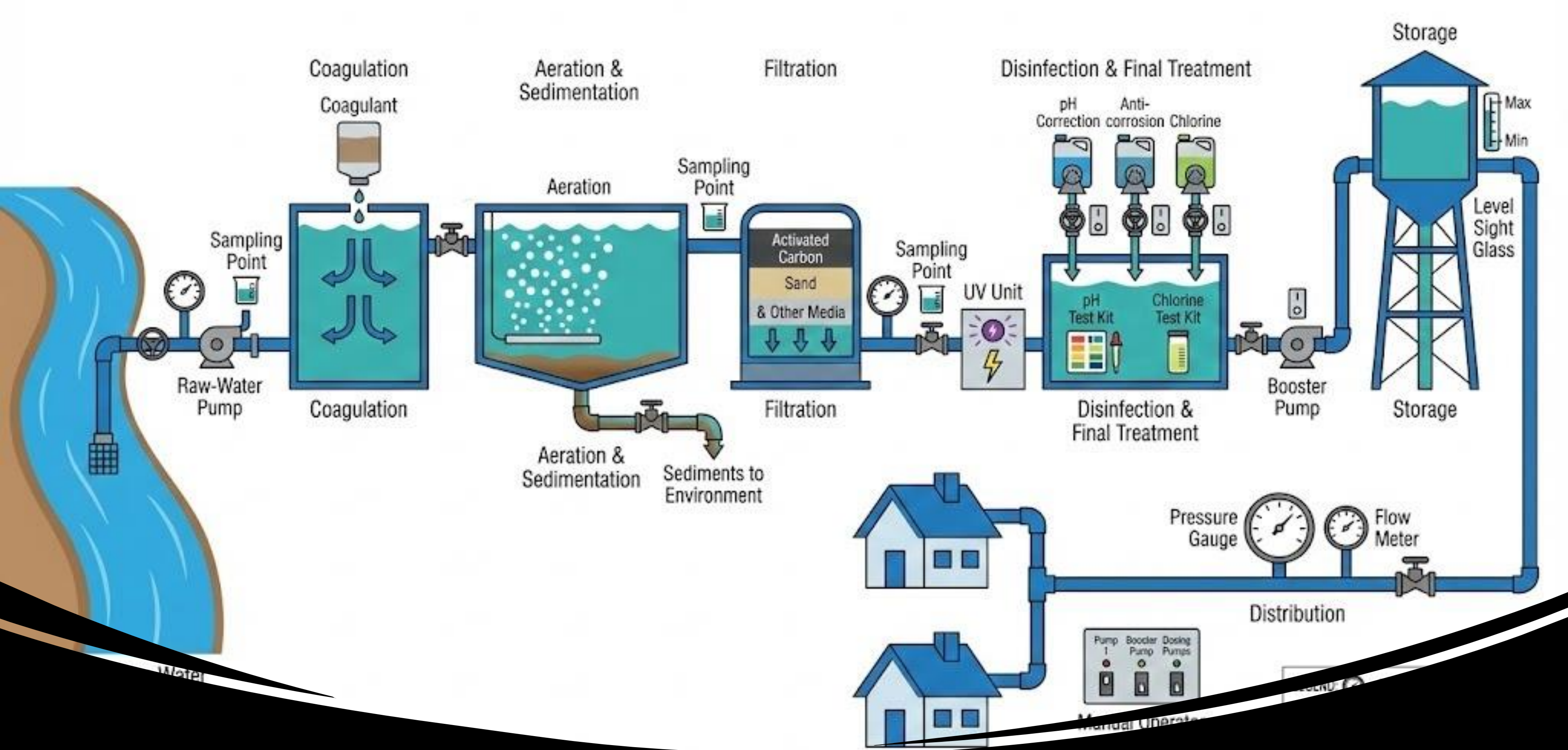
Smart WaterTreatment Control System

ILYASS NACHIT

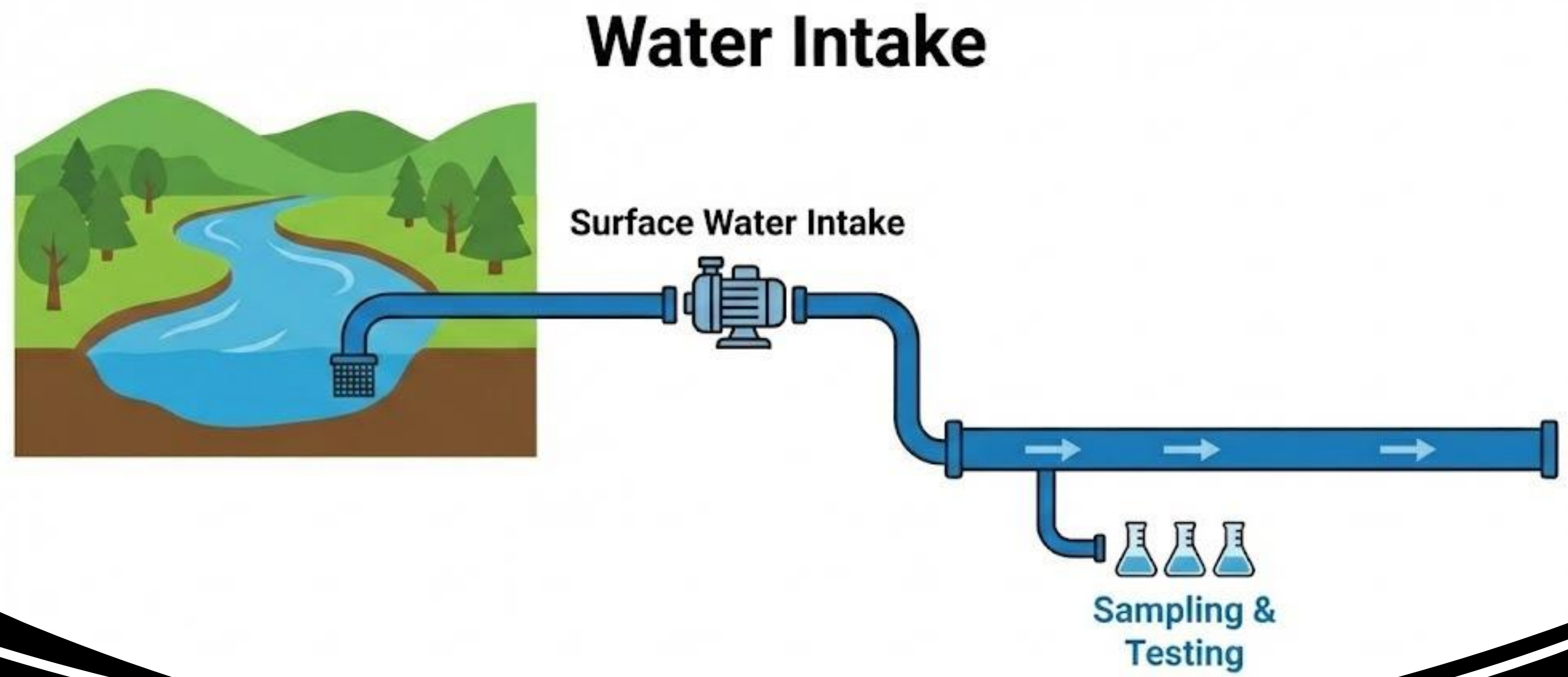
Problem and Objective

- There are many parameters that affect drinking water quality: **pH, turbidity, TDS, chlorine, level, flow rate, and pressure**.
- In many plants, measurements are still taken **manually** and **infrequently**.
- **Objective:**
- Measure these parameters **in real time**
- **Automatically control** pumps and chemical dosing using an **ESP32**
- **Simulate and test** the system using a **MATLAB/Simulink model**



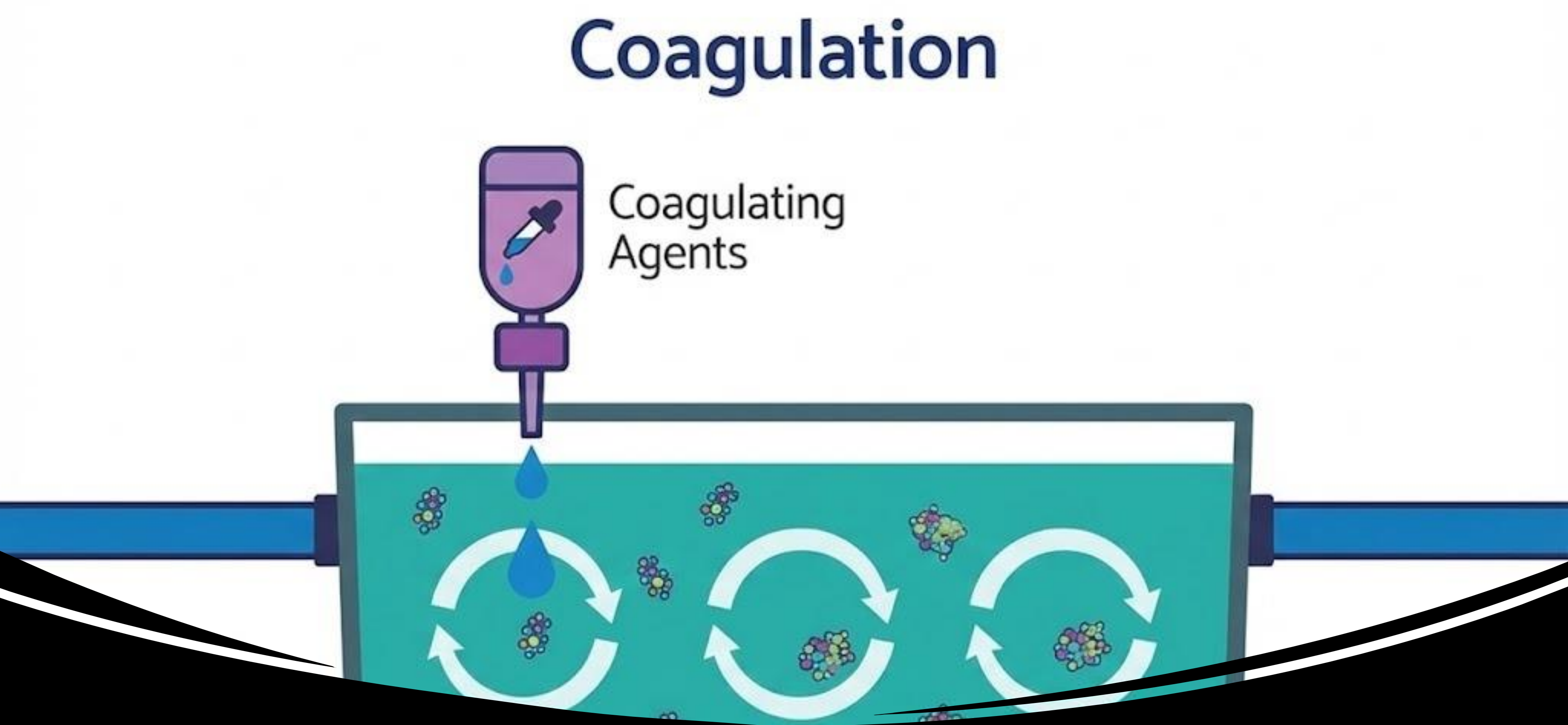
Conventional Water Treatment Process (Overview)

- **Raw Water Intake (Surface Water)**
- **Coagulation**
- **Aeration & Sedimentation**
- **Filtration**
- **Disinfection & Final Treatment**
- **Storage & Distribution**



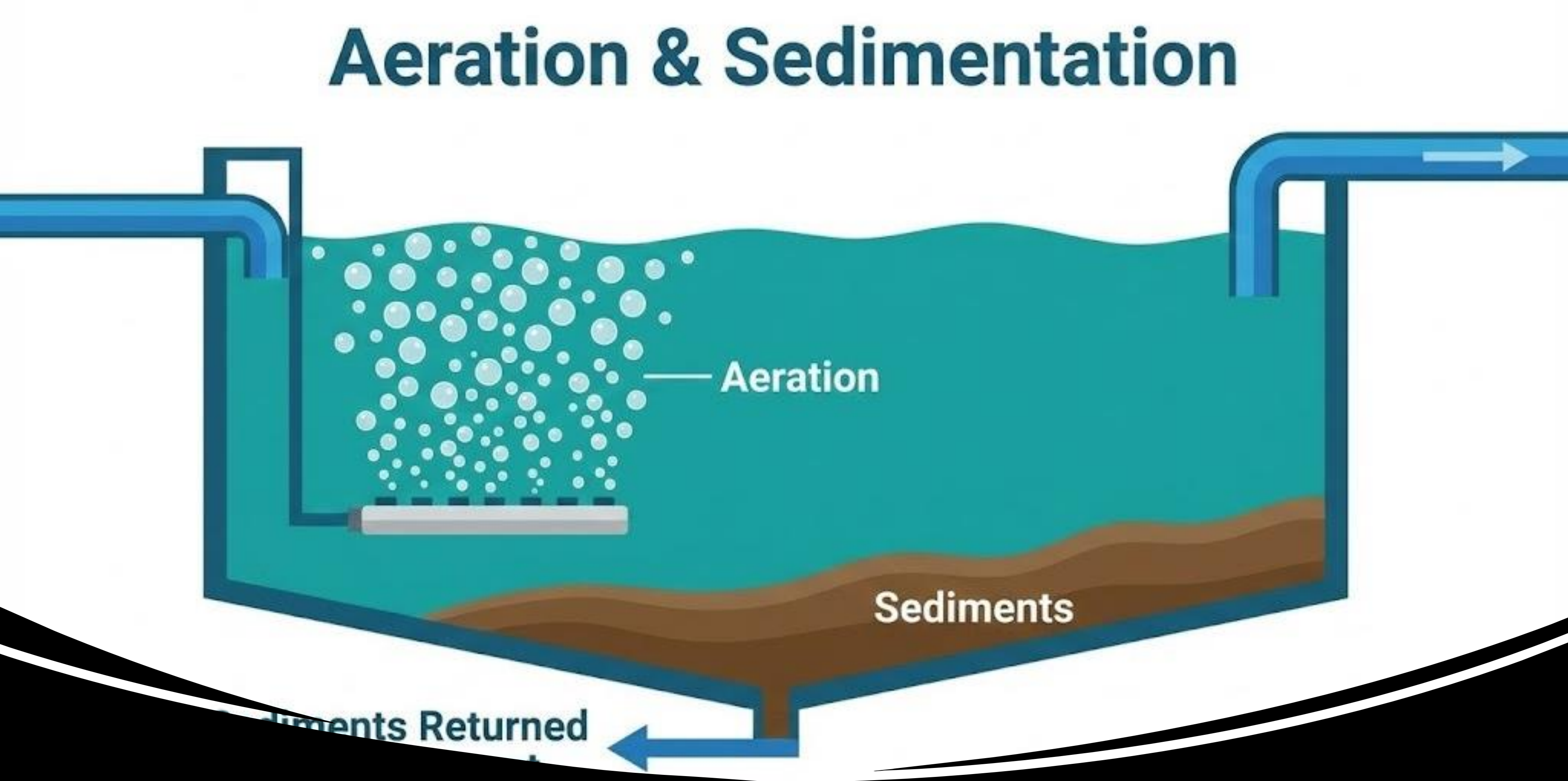
Water Intake (Raw Water Intake)

- Water is pumped from a surface water source.
- Screens (grates) remove large solids.
- First sampling point: preliminary quality check.



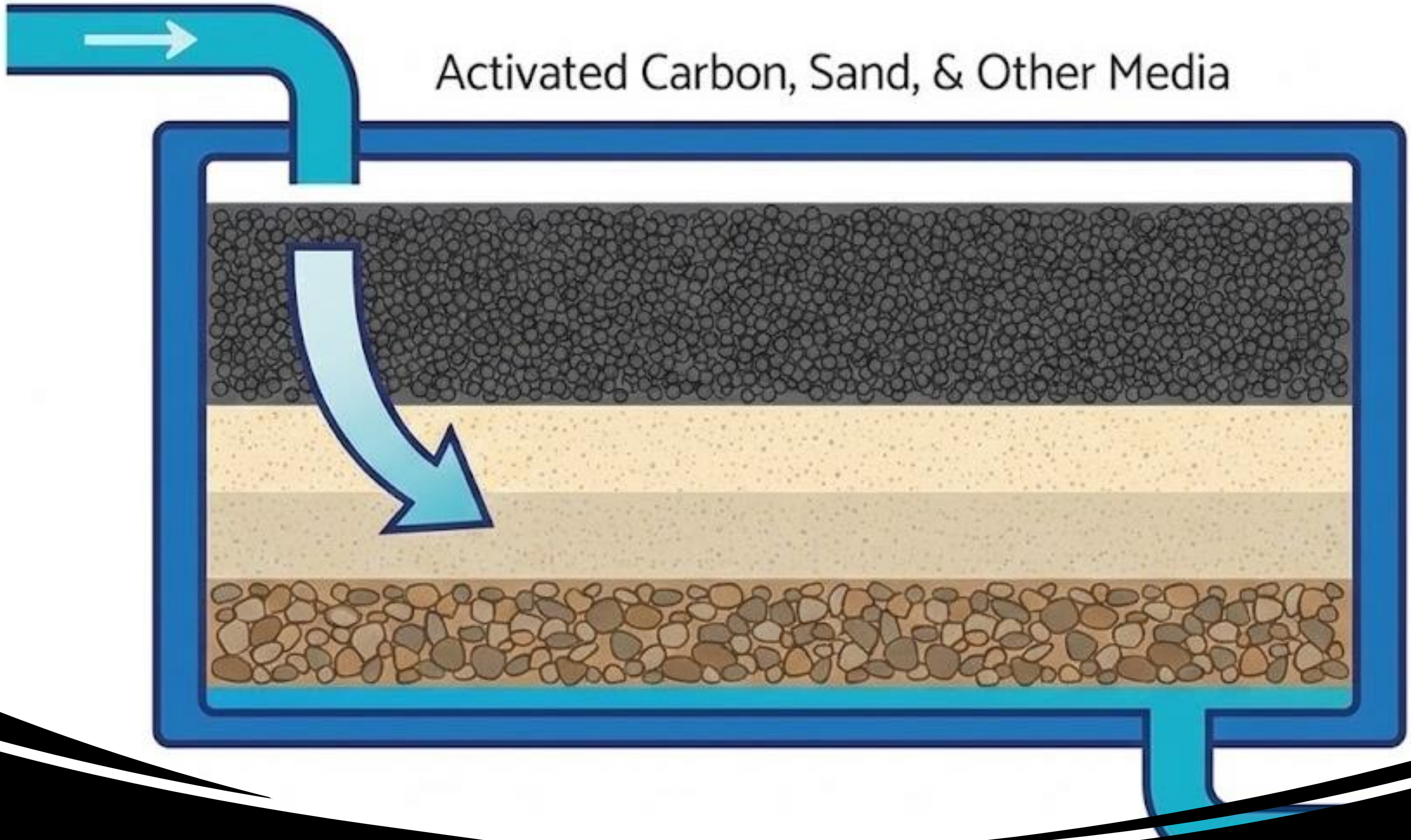
Coagulation

- A coagulant chemical is added.
- Small particles clump together and form larger flocs.
- This makes sedimentation and filtration easier.



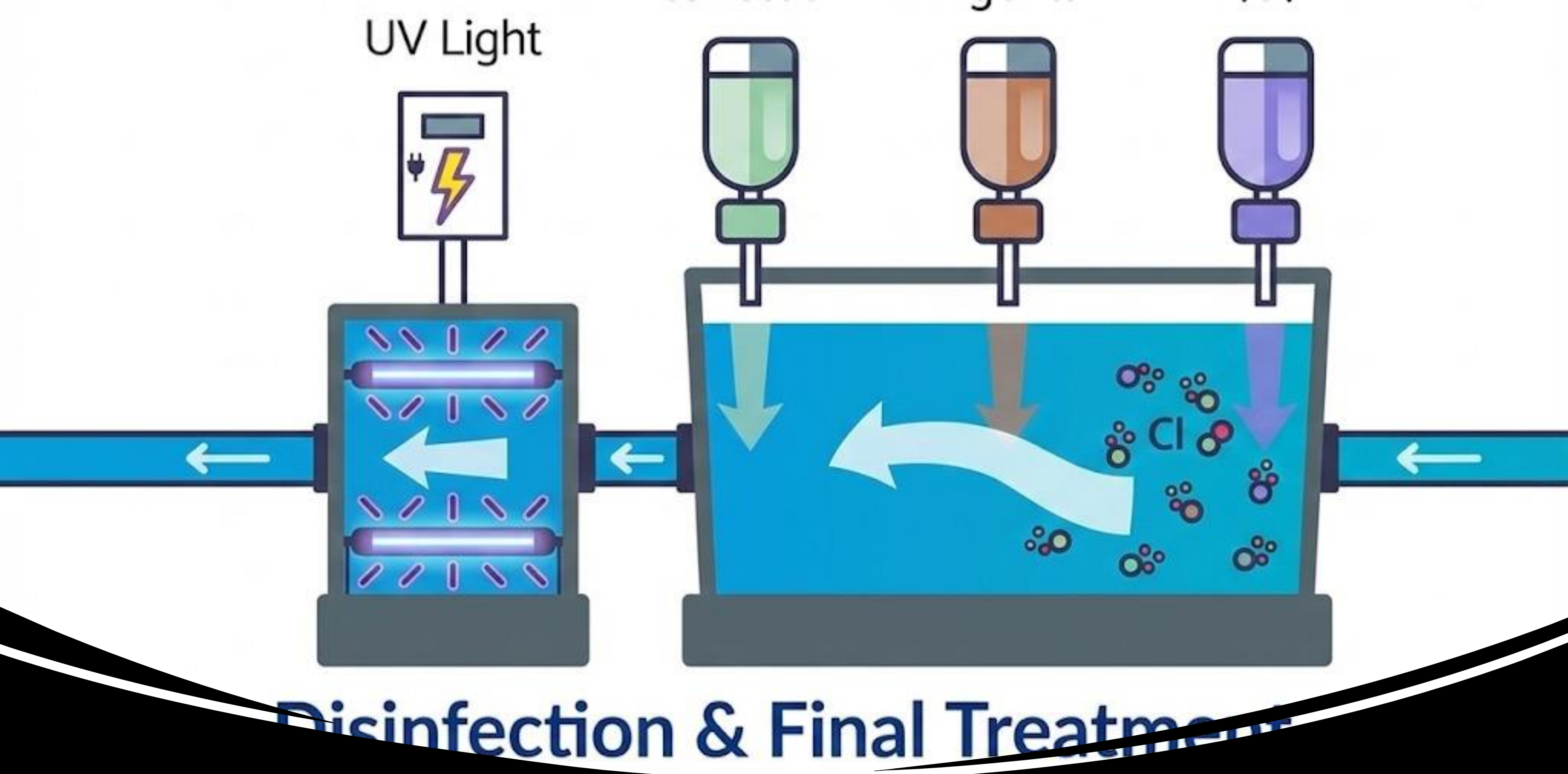
Aeration & Sedimentation

- Aeration increases dissolved oxygen and removes undesirable gases.
- Heavy flocs settle to the bottom, and sludge is discharged.



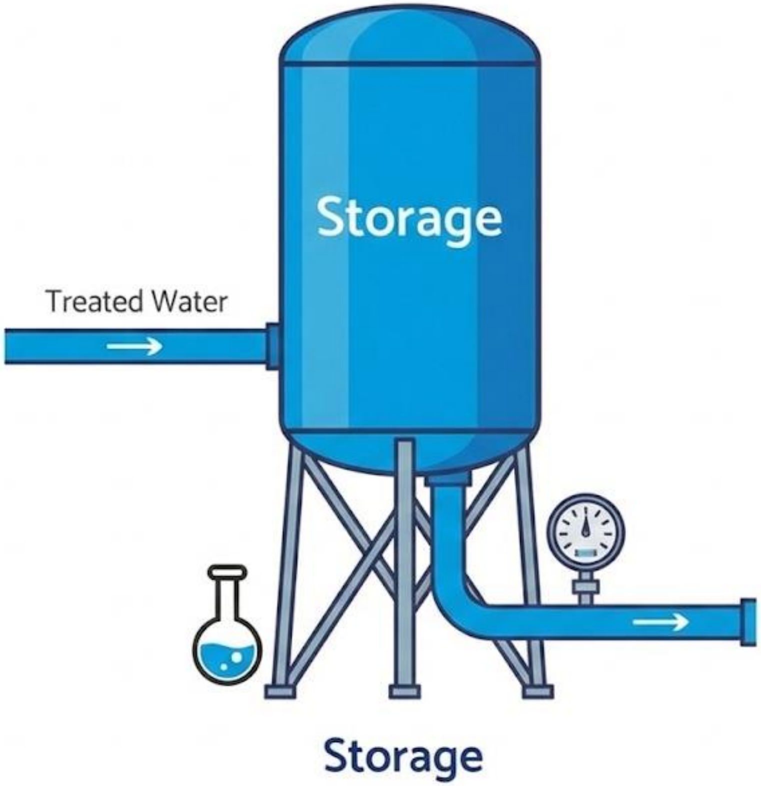
Filtration

- Layers of gravel, sand, and activated carbon.
- Fine particles and some dissolved substances are removed.
- Water clarity improves, and odor/taste are reduced.



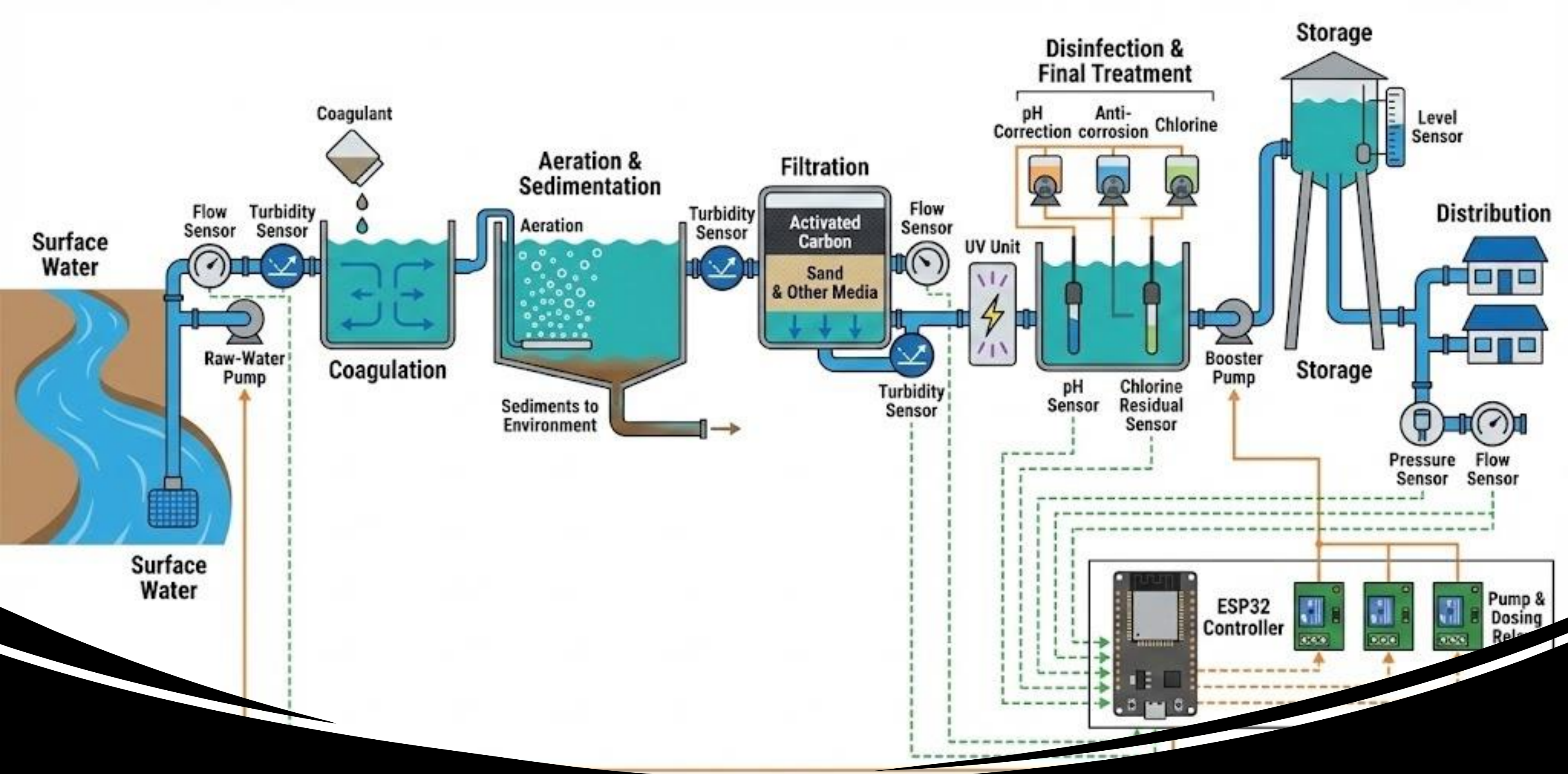
Disinfection & Final Treatment

- UV kills microorganisms.
- pH adjustment, corrosion control, and chlorination.
- The water is now ready for storage.



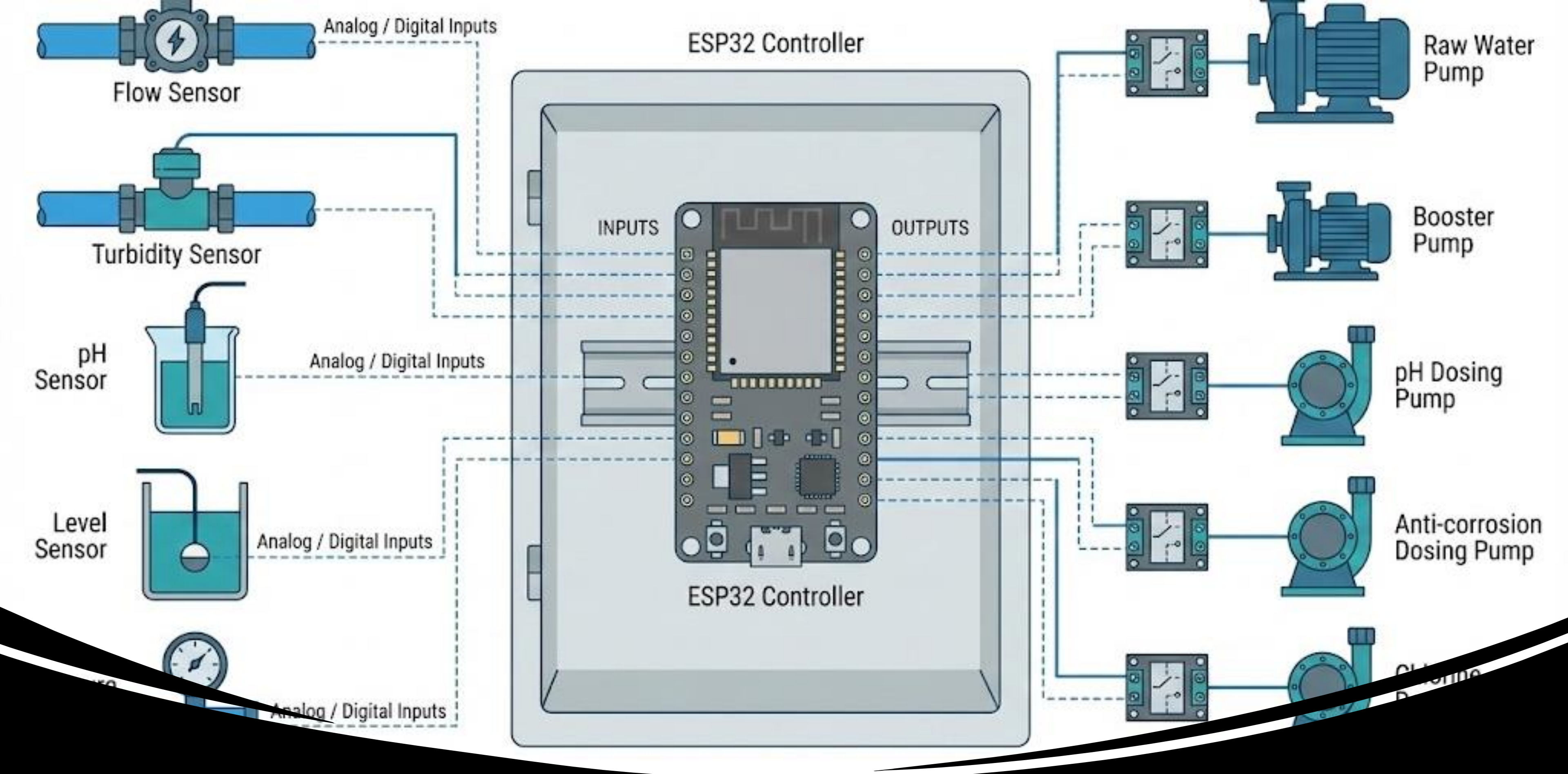
Storage & Distribution

- Water is stored in tanks equipped with level sensors.
- Distribution is managed by controlling pressure and flow rate.



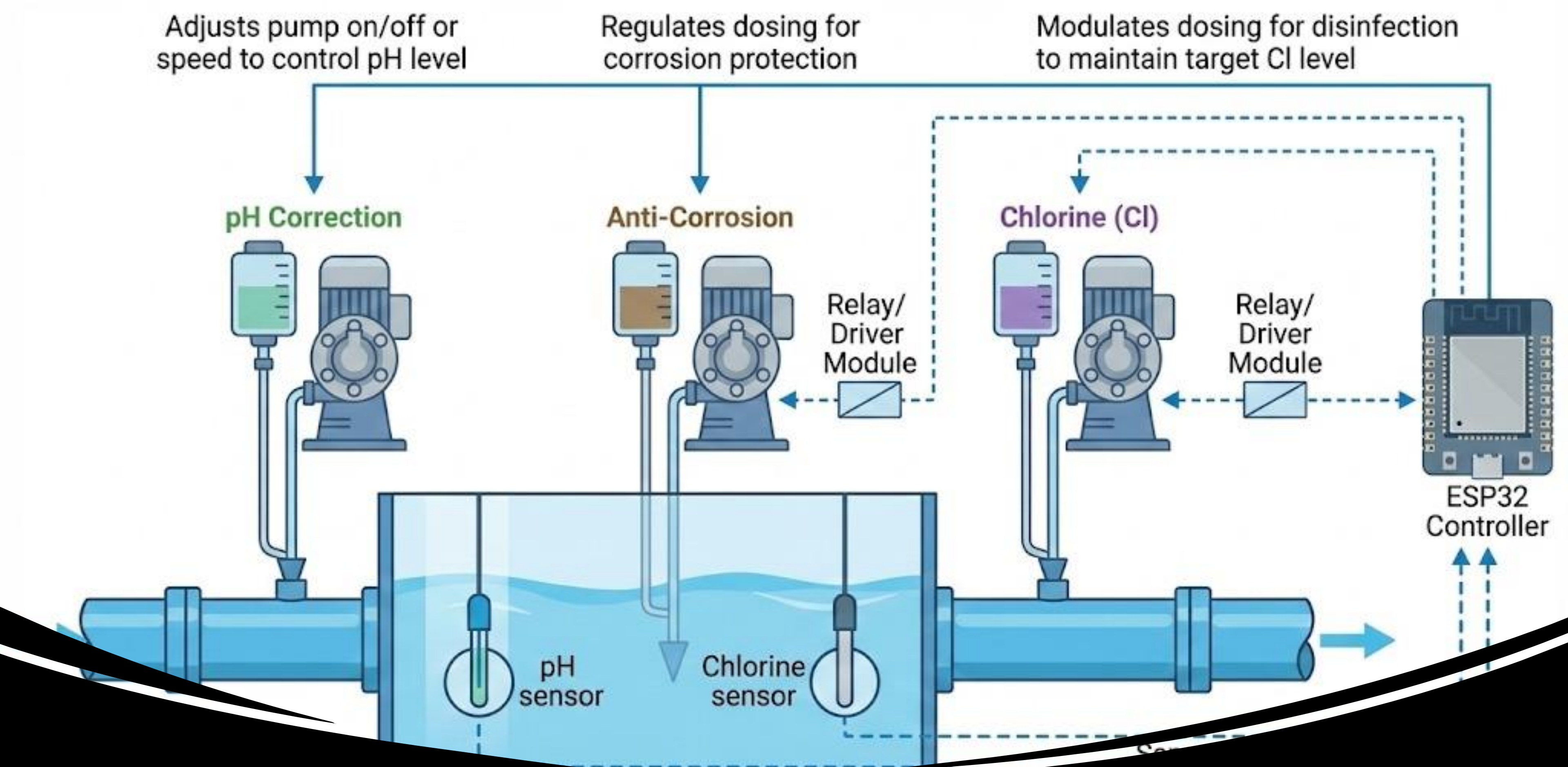
Proposed Smart System: Overall Architecture

- **Green lines:** data from sensors to the ESP32
- **Orange lines:** commands from the ESP32 to pumps and dosing pumps
- This transforms a conventional plant into a **smart IoT-based system**.



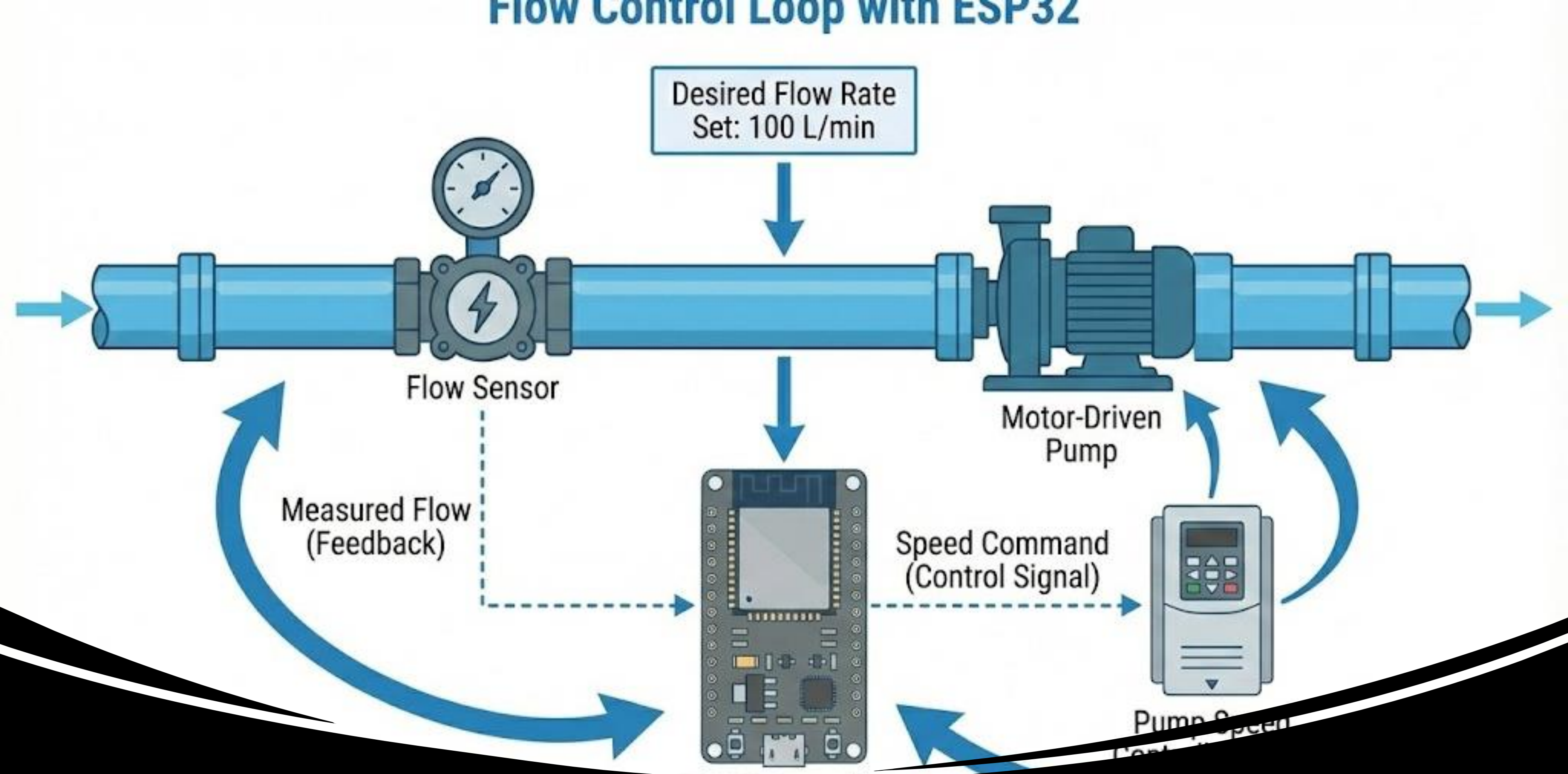
ESP32 Control Panel (Inputs/Outputs)

- **Inputs (analog/digital):**
- Flow, turbidity, pH, level, and pressure sensors
- **Outputs:**
- Raw water pump, booster pump
- pH dosing pump, anti-corrosion dosing pump, chlorine dosing pump
- The ESP32 controls the relays based on these signals



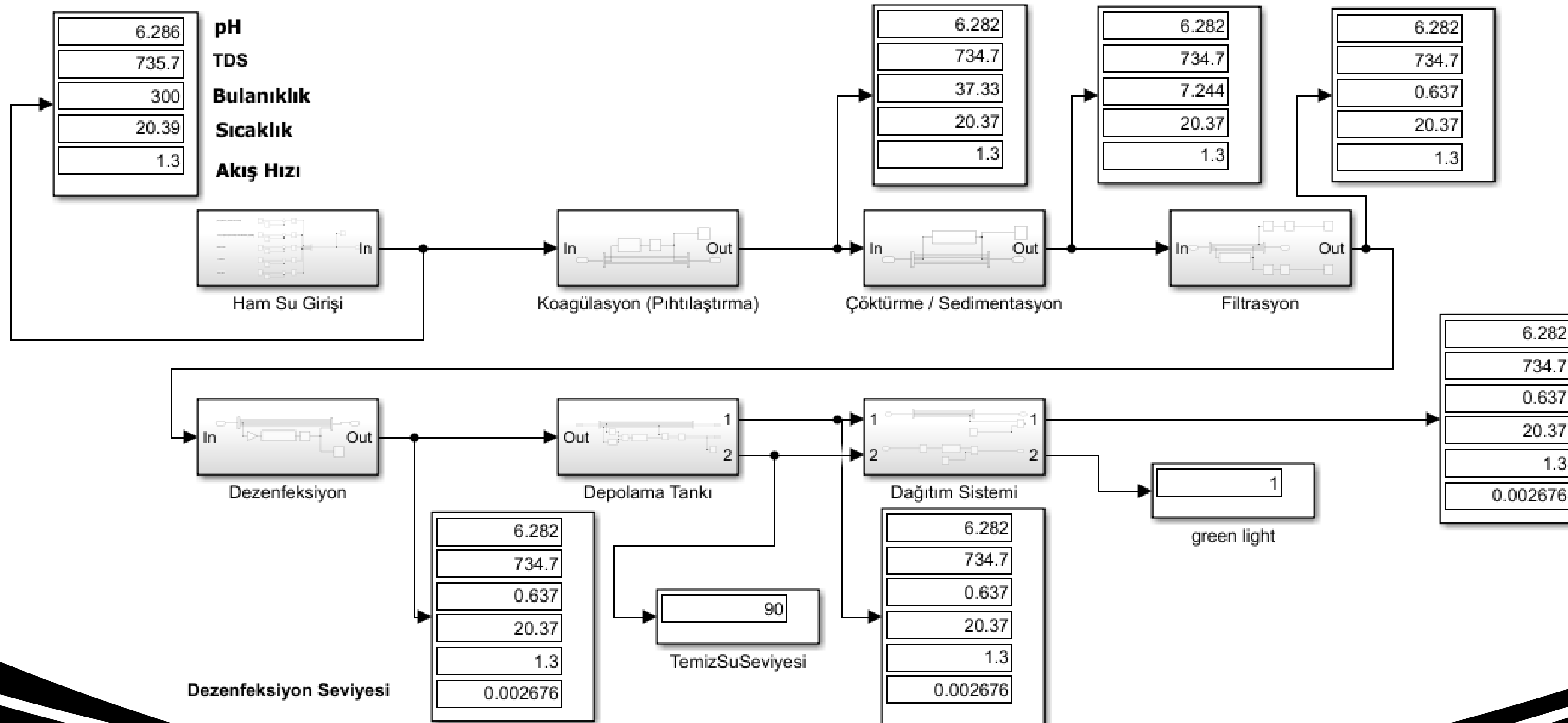
Chemical Dosing Control Loop

- There is a **pH sensor** and a **chlorine sensor** in the tank.
- The ESP32 reads the real-time values from these sensors.
- If pH is low → the **pH dosing pump** runs.
- If chlorine level is low → the **chlorine dosing pump** runs.
- Each pump stops automatically once the target range is reached.



Flow Control Loop

- Target flow rate: for example **100 L/min.**
- The measured value from the flow sensor is fed back to the ESP32.
- The ESP32 sends a speed command to the **motor driver / VFD**.
- This provides **closed-loop (feedback) control.**



Simulink Model

- Each process step (**Raw Water, Coagulation, Sedimentation, Filtration, Disinfection, Storage, Distribution**) is modeled as a separate block.
- In each block:
- Variables such as **pH, TDS, temperature, turbidity, and level**
- Subsystems can simulate **pumps, delays, losses**, etc.
- **Goal:** test the control algorithm in simulation first.

ESP32 Control Rules

Raw-Water Pump

Tank level < 70% → **ON**

Tank level > 90% → **OFF**

Booster Pump

Pressure < 2.5 bar **and** level > 20% → **ON**

Pressure > 3 bar **or** level < 10% → **OFF**

pH Correction Pump

pH < 6.8 → **ON**

pH > 7.2 → **OFF**

Chlorine Dosing Pump

Cl < 0.30 mg/L → **ON**

Cl > 0.60 mg/L → **OFF**

Anti-corrosion Pump

If any main pump is running → **ON**

These rules can later be replaced with more advanced algorithms such as **PID control**.

Example Scenarios / Operating Modes

Scenario 1 – Tank level is dropping

Level sensor reads **65%**

The system automatically turns **ON the raw-water pump**

Scenario 2 – pH is decreasing

Measured pH: **6.5**

The **pH correction pump** adds chemical

When pH reaches about **7.0**, the pump stops

Scenario 3 – Distribution pressure is low

Pressure sensor detects low pressure

The system turns **ON the booster pump**

When pressure rises to around **3 bar**, the pump turns **OFF**

Results and Future Work

We transformed a conventional water treatment plant into an **IoT-based smart system**.

The **ESP32** provides a low-cost, flexible, and internet-connected control solution.

Thanks to the **Simulink model**, testing and optimization are possible before building the real system.

Future work:

PID / fuzzy control

Web interface / mobile app

A **small-scale prototype** in a lab environment using real sensors

Sources

- **TS 266 – Regulation on Water Intended for Human Consumption**
  Republic of Türkiye Ministry of Environment, Urbanization and Climate Change
  (Drinking water quality parameters and limit values)
- **Davis, M. L., & Cornwell, D. A.** *Introduction to Environmental Engineering*
  (Water treatment processes: coagulation, sedimentation, filtration, disinfection)
- **ESP32 Technical Documentation** (Datasheet & Technical Reference Manual)
  Espressif Systems (ADC, digital I/O, Wi-Fi features)
- **Arduino IDE & ESP32 Core Documentation**
  Arduino Documentation (ESP32 programming and example codes)
- **MATLAB & Simulink Documentation** (Water Systems / Control Systems)
  MathWorks Help Center (Modeling water systems, control blocks, Simulink examples)
- **Course Notes and Instructor Presentations**
  Dicle University, Department of Electrical & Electronics Engineering
  (Materials provided within the "Engineering Design" course)