SMALL SIZE AUTOMATED RIVER LOCK MODEL SUPERVISED BY WEB APPLICATION

2024-2025 Technological School Project ESO

12 DE DICIEMBRE DE 2024

AGORA BARCELONA INTERNATIONAL SCHOOL
Nicolas de Saint-Aubert

1. Introduction

This project involves creating a model of a river lock using

- **Arduino (Bitbloq robot)**
- **robotic sensors** and **actuators,**
- a **web app (Bitbloq app)**,
- **printed mechanical component (3D Printer)**
- **recycled tubes and plastic containers and bottles**.

The goal is to demonstrate the working of a river lock and automate its operations using sensors and actuators.

2. Working of a River Lock

youtube.com/watch?v=dmZ7hBMTY8Q

A river lock is a device used for raising and lowering boats between stretches of water of different levels on river and canal waterways. It consists of a chamber with gates at each end that control the flow of water in and out of the chamber.

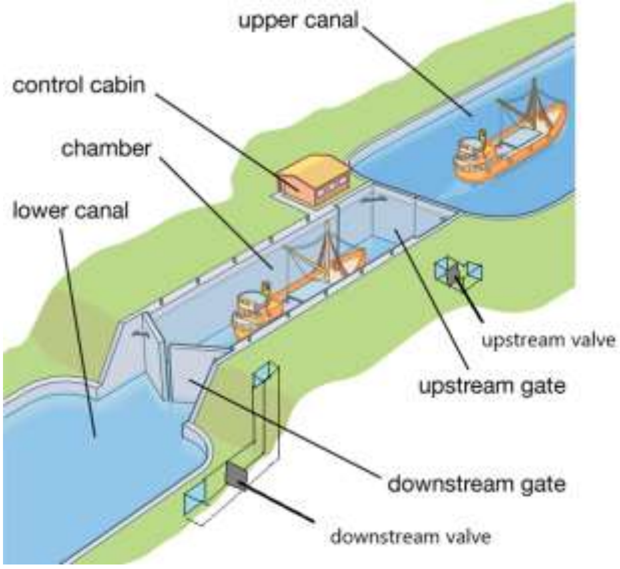

Ilustración 1 River Lock Overview

- **Chamber**: The main area where the boat is positioned. It is a watertight enclosure that can be filled with or emptied of water to change the water level.
- **Gates**: These are located at both ends of the chamber. They open to allow the boat to enter or exit and close to create a watertight seal. There are usually two sets of gates: upstream and downstream.
- **Valves**: These control the flow of water into and out of the chamber. When the valves are opened, water flows in or out, raising or lowering the water level inside the chamber.
- **Sensors**: These detect the presence of a boat and provide feedback to the control system to automate the opening and closing of gates and valves.
- **Control cabin**: Arduino!

3. Logic of the Automated River Lock

- **Detect Boat**: Sensors detect the presence of a boat:
    - **Upstream Boat Sensor**
    - **Downstream Boat Sensor**
    - **Chamber Boat Sensor**
- **Open Valves and Doors**: Automatically open the water valves and gates to allow the boat to enter or exit.
    - **Upstream Valve Motor**
    - **Downstream Valve Motor**
    - **Upstream Gate Motor**
    - **Downstream Gate Motor**
- **Fill/Free Water:** Sensors to detect when the chamber is filled or "freed" of water
    - **Water Level Sensor**

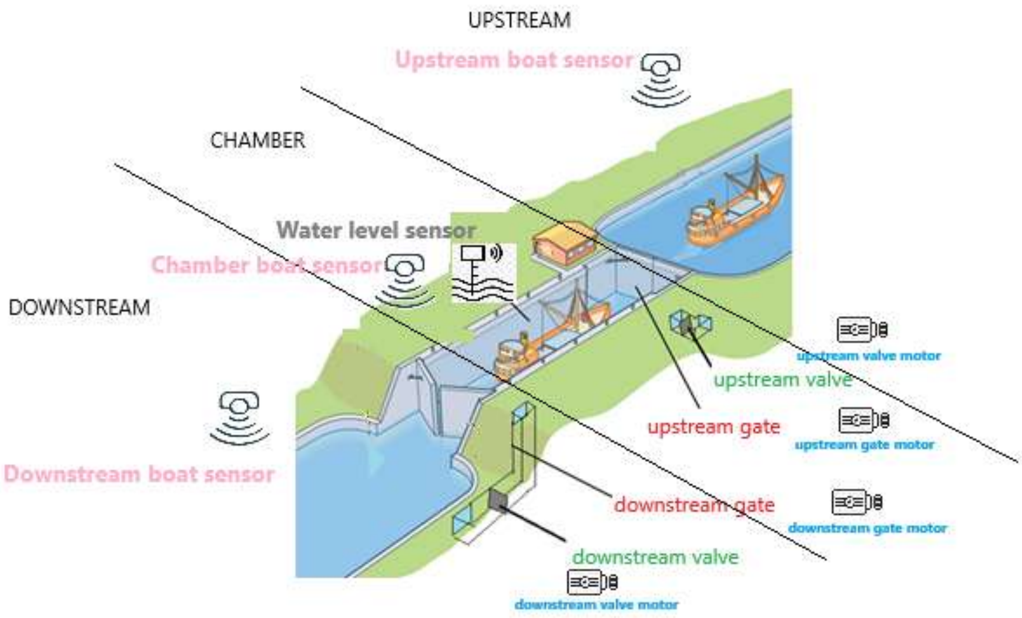

Ilustración 2 Automated River Lock

4. River Lock Model

Overview

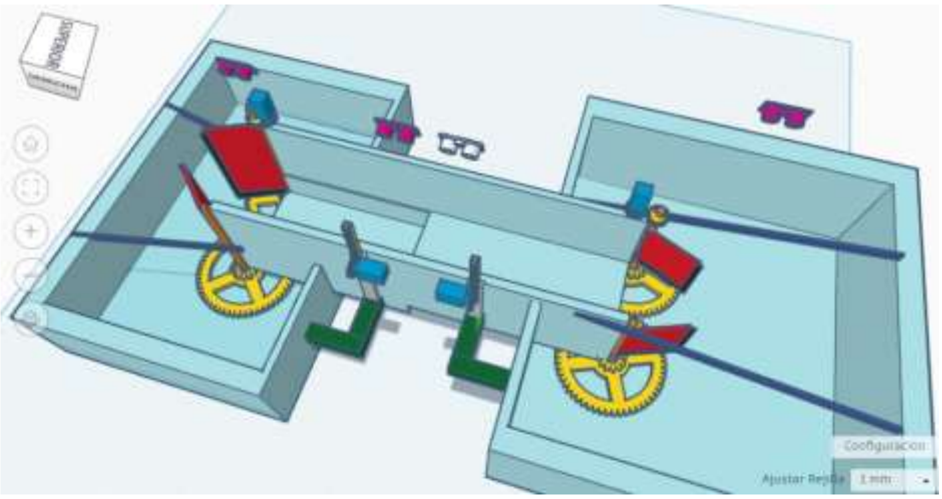

Figure 1 Full Modell View

Physical part

- **Chamber**: The main part where the boat is raised or lowered: water up, water down, water filling
- **Gates**: Control the entry and exit of the boat **downstream gate, upstream gate**).
- **2 Valves**: Control the flow of water into and out of the chamber, at **downstream valve**, and **upstream valve**

Electromechanical components

- **Controlling the doors, 2 moors/servos**: To open or close the doors
  - **Downstream Gate Motor**
  - **Upstream Gate Motor**
- **Controlling the valves, 2 motors/servos**: To open or close the valve:
  (**Downstream Valve Motor**
  - **Upstream Valve Motor**
- **3 boat sensors**: Detect the presence of the boat at
  - **Upstream Boat Sensor**
  - **Downstream Boat Sensor**
  - **Chamber Boat Sensor**
- **Dectecting the water level in the chamber**:
- Water Level Sensor

Small Scale Model (ESO 1)

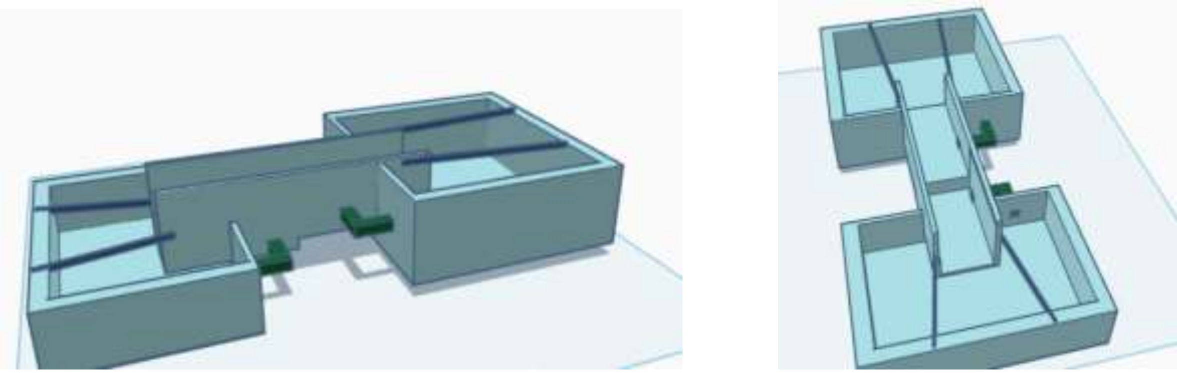

Figure 2 Model Top Views

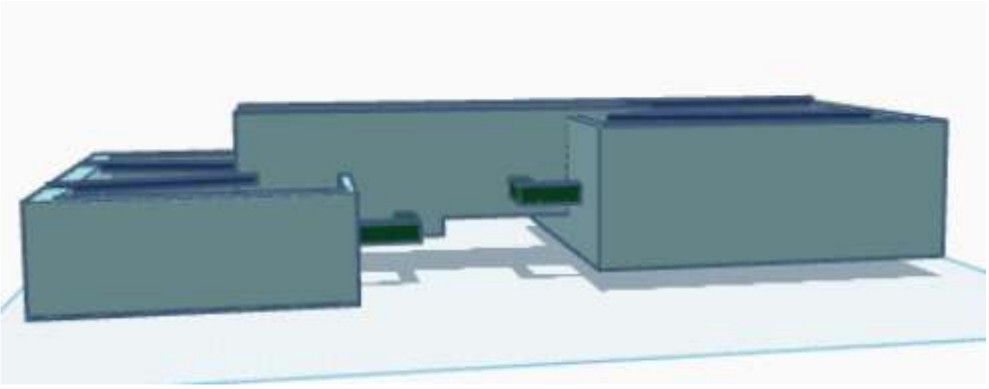

Figure 3 Model Front View

Tinkercad: river lock model.

Mechanical Components (ESO1/ESO3)

- **Opening/Closing gates' structure**

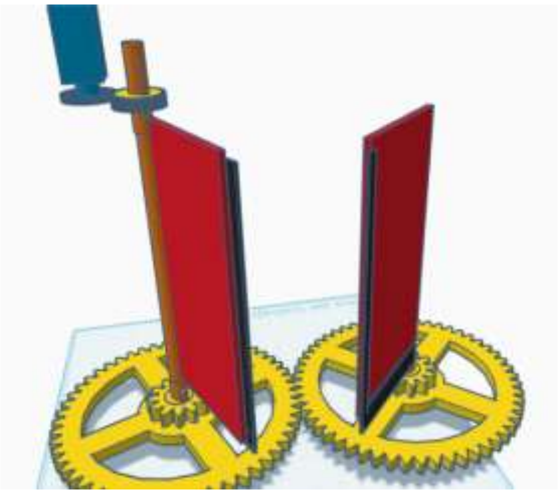

Figure 4 Gates' Mechanism

Tiinkercad: gate-mecanism

- **Opening/Closing Valves' structure**

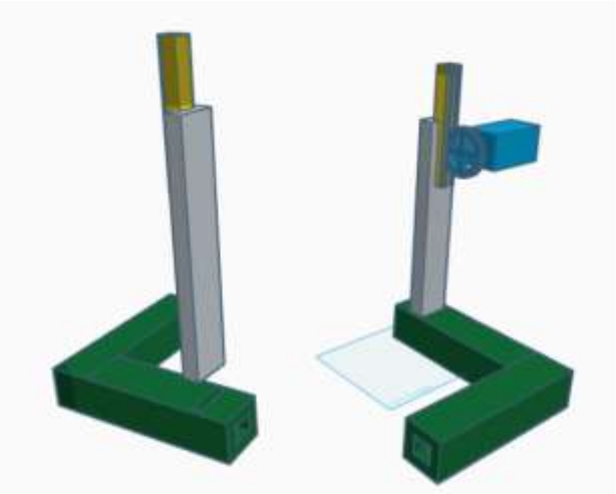

Figure 5 Valves' Mechanism

Tiinkercad: Valve Mechanism

Automated Control (ESO3)

state machine to control the Lock

2 States Machines, 1 for each direction of the boat.

Communicate with the app

Send Message to App

At each status change, the adequate message will be send to the app (the protocol is given in the following).

Respond to App Request

1. A special request may be received from the app (***get_state***) to get the actual status of the lock, the board control must respond to it adequately (see protocol).
2. There is a special message that also request the board to reset (***reset***) to a default state of the lock.

Default state of the lock

The default state of the lock must be:

- Water at low level
- Gates closed
- Upstream gate closed
- Downstream gate open

Supervision Mobile Application (ESO 2)

Grapical Representation

The application must use a **graphical representation** of the lock showing:

- Status of the **downstream and upstream gates**: closing/opening/closed/opened
- Status of the **chamber**: filling water/freeing water/filled/free
- The position of the **boat: upstream/downstream/chamber**

The application must have **2 buttons**

- Reset board
- Get state

These 2 buttons will send a message to that board that will respond with the actual state. The graphical representation must be actualized with that response. This is explained in the following.

Communicate with the board

It will receive information from the board, see:

Arduino/Web App Communication (ESO 2/ESO 3).
Board/APP Protocol Specification (ESO 2/ESO 3)

Arduino/Web App Communication (ESO 2/ESO 3)

Arduino to WebApp

The Arduino will send messages to the web app to describe its state:

Gates' status

- **Opening Downstream Gate**: {message: "opening_gate_downstream"}
- **Closing Downstream Gate**: {message: "closing_gate_downstream"}
- **Open(ed) Downstream Gate**: {message: "gate_downstream_open"}
- **Closed Downstream Gate**: {message: "gate_downstream_closed"}
- **Opening Upstream Gate**: {message: "opening_gate_upstream"}
- **Closing Upstream Gate**: {message: "closing_gate_upstream"}
- **Open(ed) Upstream Gate**: {message: "gate_upstream_open"}
- **Closed Upstream Gate**: {message: "gate_upstream_closed"}

Boat detection

- **Detecting Boat at Gate Downstream**: {message: "detect_boat_downstream"}
- **Detecting Boat at Gate Upstream**: {message: "detect_boat_upstream"}
- **Detecting Boat in chamber**: {message: "detect_boat_chamber"}

Chamber filling

- **Filling Water**: {message: "chamber_filling_water"}
- **Freeing Water**: {message: "chamber_freeing_water"}
- **Chamber filled**: {message: "chamber_level_down"}
- **Chamber freed:** {message: "chamber_level_up"}

WebApp to Arduino

Synchronization at Startup

At startup or when a button is pressed, the app will ask for the state of the lock to ensure synchronization:

- **Request State**: {request: "get state"}
- **State Response**: {response: “gate upstream: open/close/opening; gate upstream: open/close/opening; chamber: filling/freeing/filled/free; boat:none/upstream/downstream/chamber”}

Resetting the Lock

The app may send a message to reset the lock to its default position:

- **Default Position**: Gates closed, water level at the lower level.
- **Reset Message**: {message: "reset"}

Board/APP Protocol Specification (ESO 2/ESO 3)

Here’s the table based on the provided information:

Table 1 Communication Protocol

Lock Event	App Event	Message	Actions at Reception
Opening Downstream Gate		opening_gate_downstream	App **updates** status to show downstream gate opening
Closing Downstream Gate		closing_gate_downstream	App **updates** status to show downstream gate closing
Open(ed) Downstream Gate		gate_downstream_open	App **updates** status to show downstream gate open
Closed Downstream Gate		gate_downstream_closed	**App updates** status to show downstream gate closed
Opening Upstream Gate		opening_gate_upstream	**App updates** status to show upstream gate opening
Closing Upstream Gate		closing_gate_upstream	**App updates** status to show upstream gate closing
Open(ed) Upstream Gate		gate_upstream_open	**App updates** status to show upstream gate open
Closed Upstream Gate		gate_upstream_closed	**App updates** status to show upstream gate closed
Detecting Boat at Gate Downstream		detect_boat_downstream	**App updates** status to show boat detected at downstream gate
Detecting Boat at Gate Upstream		detect_boat_upstream	**App updates** status to show boat detected at upstream gate

Lock Event	App Event	Message	Actions at Reception
Detecting Boat in Chamber		detect_boat_chamber	**App updates** status to show boat detected in chamber
Filling Water		chamber_filling_water	**App updates** status to show water filling
Freeing Water		chamber_freeing_water	**App updates** status to show water freeing
Chamber Filled		chamber_level_down	**App updates** status to show chamber filled
Chamber Freed		chamber_level_up	**App updates** status to show chamber freed
	Ask for state at startup or at user click	get_state	**Lock responds** current state: {"response": "gate upstream: open/close/opening; gate downstream: open/close/opening; chamber: filling/freeing/filled/free; boat: none/upstream/downstream/chamber"} **App updates** to show the actual states
	Reset Lock Position	reset	**Lock resets** to default position **responds** current state: {"response": "gate upstream: open/close/opening; gate downstream: open/close/opening; chamber: filling/freeing/filled/free; boat: none/upstream/downstream/chamber"} **App updates** to show the actual states