TRail
(Terminal on Rails)

Daria Chekeres, Ardi Cerkini, Tehini Bourg, Ruaeg Bürcher, Valentin Keusen, Lucas Guesry, Gandhi Gautier

MIT Project Proposal
Course : CS-358 Making Intelligent Things
Team member number : 7

October 19, 2025

Contents

1. Project Presentation

As the group project, we have chosen to build a train container terminal. The structure of the terminal will allow enough space between the holding pillars for: a train and a container. Being itself on rails, it will move on the “train axis”, to position itself properly in order to precisely grab containers (which will either be positioned on the trains or on the ground, next to the train rails). This will be done thanks to a claw which will be mounted on a structure connected to upper rails, allowing movement from train to container line. The structure will allow the claw to descend using a pulley system. Switches will be used in order to determine the correct positions of the claw (if it touches a container) and the structure (how far did we go up and on the left and right sides).

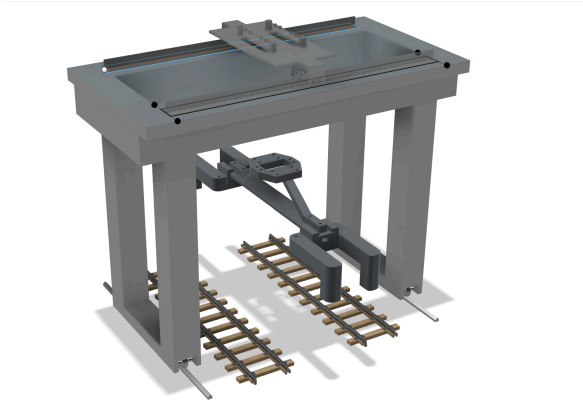
1.1. Mechanical Description

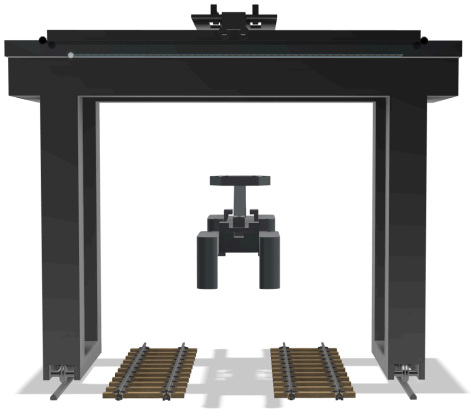
In the next description, left and right mean perpendicular to the rails of the trains going under the crane, while forward and backward mean parallel to them.

The whole structure moves forwards/backwards with one geared motor close to the ground. On top, a cart which holds the claw elevation mechanism will move left and right thanks to a geared motor. On the left end, a microswitch detects when the cart reaches its furthest allowed position and thus the origin can be detected after a reboot.

The claw is mounted on the cart and goes up/down (with extremities detected by 2 microswitches) with two more NEMA 17 steppers (attached on the cart) which (un)wind two coils of rope each, attached to a pulley. Those steppers were chosen since they can pull strongly enough to lift the containers. Finally, on the claw a servo spins to open/close the claw, the model was chosen to be strong enough to hold a loaded container.

The step design is available on the GitHub repository.





1.2. Software Description

1.2.1. User Stories

- As a user, I want to be able to have access to an app on a computer. On this app, I want to be able to see where each container is parked on which parking slot. On this app, I want to change the position of a container onto another parking slot.

- As a developer, I want my program to be able to detect the containers and the position on which they are, and act accordingly to the will of the user.

1.2.2. How The Terminal Will Be Used

The terminal will first scan its area of work (i.e. the containers under its control), send that info to a computer. Then, the user will send an order (i.e. "Move Container 1 to Parking Slot A") through an app. If time allows, we would try to give a real GUI to that app. But this app could also end up being a simple CLI program. The terminal will act on it and wait for another order after.

The terminal will have 3 main states:
- Setup: The terminal is getting ready (by scanning its area)
- Stall: The terminal is awaiting for order
- In Work: The terminal is moving a container
  1. Computing of the path needed to take the container and move to the desired parking slot
  2. Retrieval of the container
  3. Deposing of the container

1.2.3. How Will The Terminal See

The camera on the terminal will feed images to the computer, which will perform all the computations required to find the correct path for the proper position of the claw. It will then send commands to the brain micro-controller of the terminal, which will then control the motors. To do so, the computer will send the commands to the camera that will send directly to the brain micro-controller. The camera and the brain micro-controller are connected to each other, but only to allow to send commands to the brain micro-controller through camera's Wi-Fi.

The camera will be set directly on the terminal. To do so, we will use an ESP32-CAM as our camera with the given OV2640. Since the field-of-view of the OV2640 is too narrow, we will need to "hack" it by removing the original lens of the OV2640 sensor and adding to it an M12 mount, so that we can attach to it a proper M12 lens. That way, we can use a wide-angle lens.

2. Risk Assessment

Risk on the software side include:

- Getting machine learning to work
- Getting machine learning to work in different luminosity contexts
- Using a camera that sometimes has quality/fps issues
- Communication between the structure and the computer

Risk on the structural side include:

- Getting smooth movement of the structure with motors on only one side
- Steady movement of the claw
- Keeping the claw level during (de)elevation
- Being sure that the grip of the claw is strong enough but not destructive

Risk on the electronic side include:

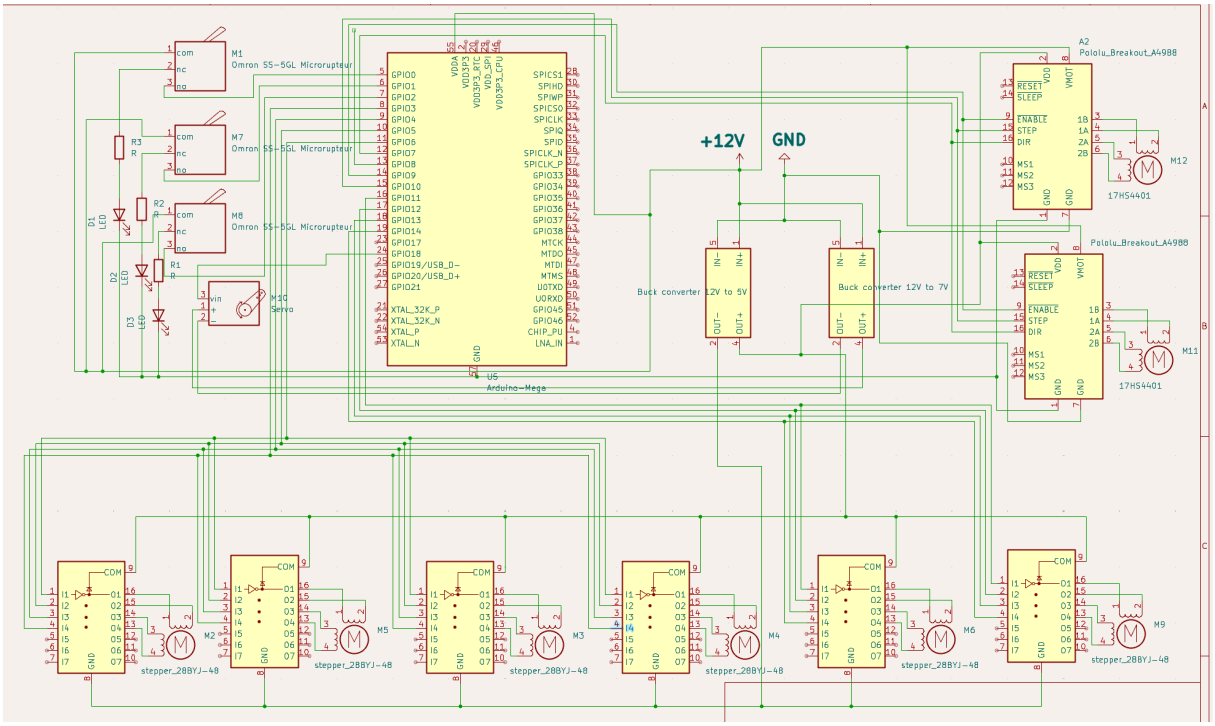
- Supplying power (voltage) correctly to all motors
- Connecting moving parts with wires

3. Electronics chart

The electrical schematic of our system is centered around an ESP32-S3 microcontroller which serves as the main control unit for all actuators and sensors of the miniature container carrier.

Two A4988 stepper motor drivers are used to control a pair of NEMA-17 (17HS4401) stepper motors, responsible for the precise lifting and positioning mechanisms. The system also includes two DC gear motors with drivers integrated, connected to the ESP32 which drives along the rails and moves the top structure to move horizontally. Three SS-5GL microswitches are integrated as limit switches to detect end-of-travel positions and ensure mechanical safety.

Additionally, a DMS15 servo motor is included for the claw. We omitted to put pull down resistors for the switches and for the servo on this schematic, since we thought that it will improve readability and that this wasn't the focus of the chart.



4. Bill of materials

PRICE	QUANTITY	TOTAL CHF
Rallonge cable usb°		
12.95	2	25.9
Power adapter (In course's stock)		
unknown	1	unknown
Switches°		
2.55	4	10.20
ESP32-CAM (In course's stock)		
4	1	4
FTDI USB-to-serial (In course's stock)		
2	1	2
Logitech Brio 105°		
25.7	1	25.7
ED-LENS-M12-230325-12°		
11.52	1	11.52
17HS4401 + A4988 (In course's stock)		
15	2	30
Steppers 28BYJ-48 + ULN2003 (In course's stock)		
2	6	12
USB isolator (In course's stock)		
unknown	1	unknown
Servo tester		
unknown	1	unknown
claw rotation transmitter°		
8.85	2	17.7
Buck converter°		
4	2	8
Arduino Mega (Provided)		
14	1	14

Servo (In course's stock)

5	1	5

GT2 timing belt (In course's stock)

1/m	4	unknown

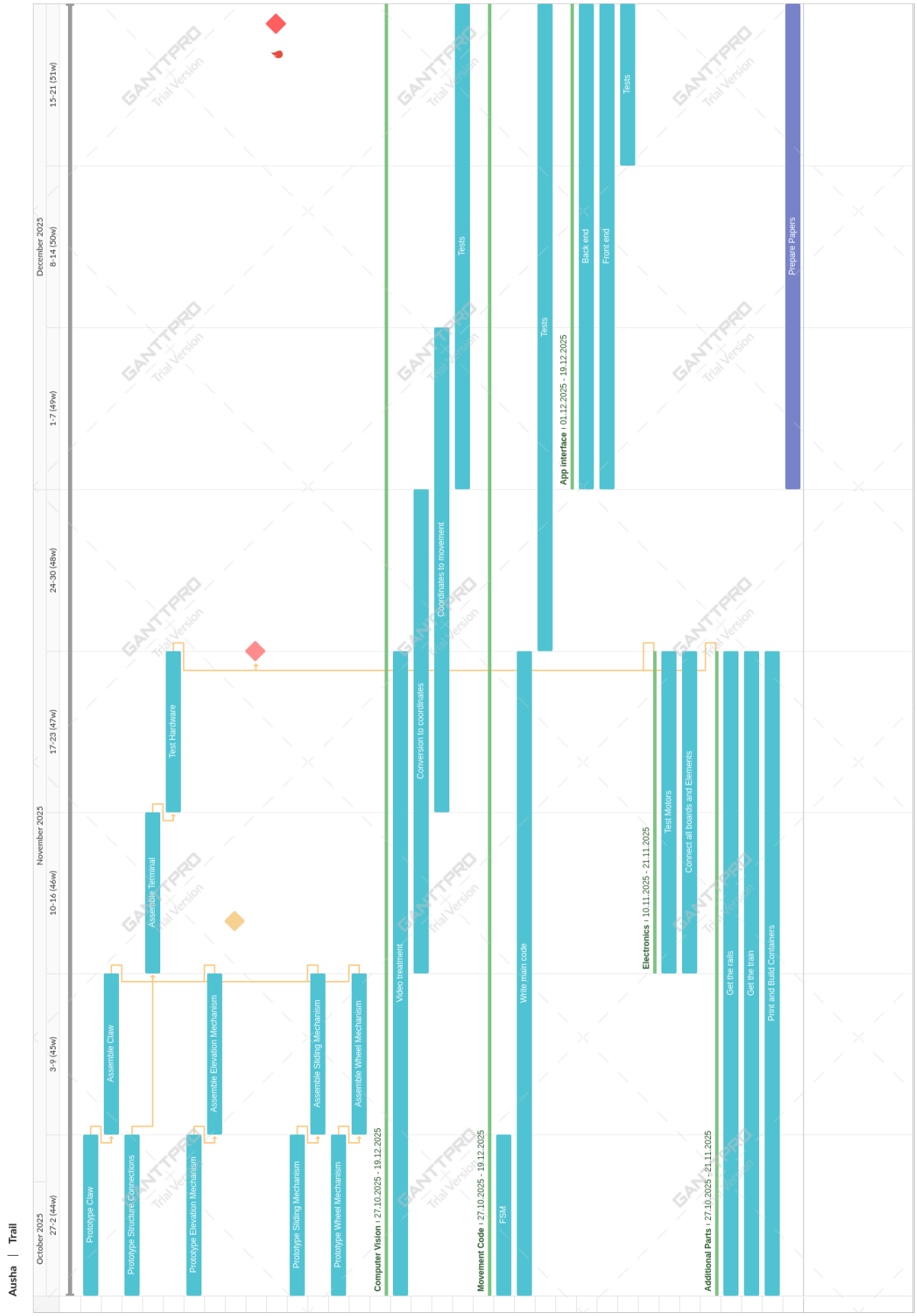
Total:

166.02 CHF

5. Work Breakdown

The following image shows all tasks we have planned to successfully finish the project in time. Some of them take more time than just one week because we deemed that it is better to start the software as early as possible. By doing so, we are planning on reducing the number of "surprises" linked to either the code or the hardware part. Moreover, we have fixed our own deadline for the mechanical part at the end of week 10. The ungraded prototyping presentation and the main presentation are also included.

In addition, an optional task was planned as to not oversimplify the software : creating application's own proper interface. In case we do not manage to stick with the tight schedule, this task will be unfortunately disregarded. With this schedule and optional constraints, we should be done by week 13, leaving one week to fix arising issues.


Ausha | Trail
October 2025
November 2025
December 2025
27-2 (44w)
3-9 (43w)
10-16 (46w)
17-23 (47w)
24-30 (48w)
1-7 (49w)
8-14 (50w)
15-21 (51w)
Prototype Claw
Assemble Claw
Prototype Structure Connections
Assemble Terminal
Test Hardware
Prototype Elevation Mechanism
Assemble Elevation Mechanism
Prototype Sliding Mechanism
Assemble Sliding Mechanism
Prototype Wheel Mechanism
Assemble Wheel Mechanism
Computer Vision : 27.10.2025 - 19.12.2025
Video treatment
Conversion to coordinates
Coordinates to movement
Tests
Movement Code : 27.10.2025 - 19.12.2025
FSM
Write main code
Tests
App interface : 01.12.2025 - 19.12.2025
Back end
Front end
Tests
Electronics : 10.11.2025 - 21.11.2025
Test Motors
Connect all boards and Elements
Additional Parts : 27.10.2025 - 21.11.2025
Get the rails
Get the train
Print and Build Containers
Prepare Papers
GANTTPRO
Trial Version