International Journal of Science and Research (IJSR)
ISSN: 2319-7064
SJIF (2022): 7.942

Design and Simulation of the Control Program for the S7-300 Programmable Controller for Controlling the BINTI Hauling Robot in Lubumbashi

SANGWA Muyumbafidele[1], KAUMBA Emmanuel[2], Nyembo WA Sangwa Taty Ken[3]

[1]Electro-Energetic Engineer and Teaching Assistant at the Higher Institute of Applied Techniques in Lubumbashi

[2]Industrial Computer Engineer, Master in Criminology, Doctoral Student in Criminology and Head of Works at the Higher Institute of Applied Techniques in Lubumbashi

[3]Electrical Engineer and Teaching Assistant at the University of Likasi

Abstract: ***The increasing use of automation in urban environments has led to the deployment of robots for road traffic regulation and control, resulting in the assistance or replacement of road traffic police officers by robots. In the Democratic Republic of Congo DRC, the BINTI rolling robot, a locally developed invention, is employed for this purpose, with support from LWOMEN technologies, an association of women engineers in the DRC. Previous studies by O. BEYA, E. KAUMBA, T. KASEBA KASEBA, and G. KAMENA NKASA focused on the automation of the BINTI robots hardware, leaving the software aspect unaddressed. This study distinguishes itself by aiming to design the API program to control the BINTI rolling robot, starting from the grafcet specifications and implementing it on the S7-300 API using TIA PORTAL. The simulation of the program will be conducted using the PLCSLIM, Siemens automaton simulator.***

Keywords: Automation, BINTI rolling robot, API program, Grafcet, S7-300 automaton, TIA PORTAL

1. Introduction

Automation has led to the use of robots in urban environments to enable regulation and control of road traffic. As a result, road traffic police (PCR) officers find themselves either assisted or replaced by robots which control and regulate road traffic. This is the case of the rolling robot commonly called BINTI in the Democratic Republic of Congo (DRC), and more particularly in Lubumbashi on certain arteries. The BINTI rolling robot is a completely Congolese invention, developed by Congolese inventors with the financial support of LWOMEN technologies, an association of women engineers in the DRC.

Looking through previous studies, O. BEYA, E. KAUMBA, T. KASEBA KASEBA and G. KAMENA NKASA (2023, 5), in their study “establishment of an automated system for improving the operation of the rolling robot BINTI in Lubumbashi", proposed a grafcet for the automation of the BINTI rolling robot. Indeed, for their part, they focused on the implementation of the automated system for controlling the BINIT rolling robot without addressing the software aspect. Regarding this, our study stands out from our predecessors in that we aim to design the API program for controlling the BINTI rolling robot. Obviously, we will start from the grafcet notebook to finally program the S7-300 API under TIA PORTAL. The simulation will be carried out on the PLCSLIM, the Siemens automaton simulator.

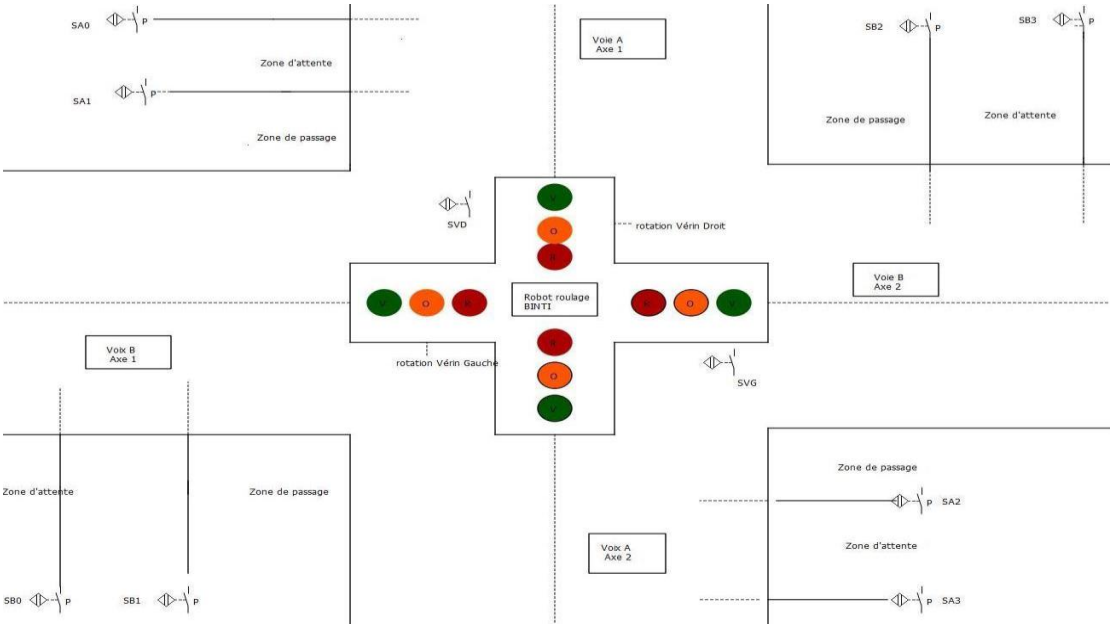

Figure 2: Block diagrams of the new model of the BINTI control system

2. Design of the S7-300 automaton program for controlling the BINTI robot

a) BINTI robot automation specifications

O. BEYA, E. KAUMBA, T. KASEBA KASEBA and G. KAMENA NKASA (2023, 5), in their study proposed specifications for the automation of the BINTI rolling robot, continuing that: "On power-up of the robot, the initialization conditions were set so that the driving robot indicates an orange light in all directions. After 5 seconds it shows a red light in all directions. After 5 seconds, we choose a situation such that the rolling robot turns on a green light on track A, and a red light on track B. The position sensor indicates the current position of the cylinder. After 5 seconds, vehicles in the waiting areas are counted. Knowing that each lane has two axes, the numbers of vehicles on the two axes of each lane will be added and then compared with the result of the perpendicular lane. The API receives these results on the basis of which it controls the robot. The counters must be reset each time the rolling robot rotates." For our part, we will therefore start from these specifications to design the control program for the S7-300 API to which we will connect the sensors and actuators necessary for controlling the BINTI robot.

b) GRAFCET of the BINTI robot

Level 1 GRAFCET is used to represent the desired operating sequence. The description of the actions and the sequence of the automatism is literal. Level 1 GRAFCET is used to identify the functions that the automation must perform. For each of these functions, it is necessary to deduce what are the actions to be carried out.

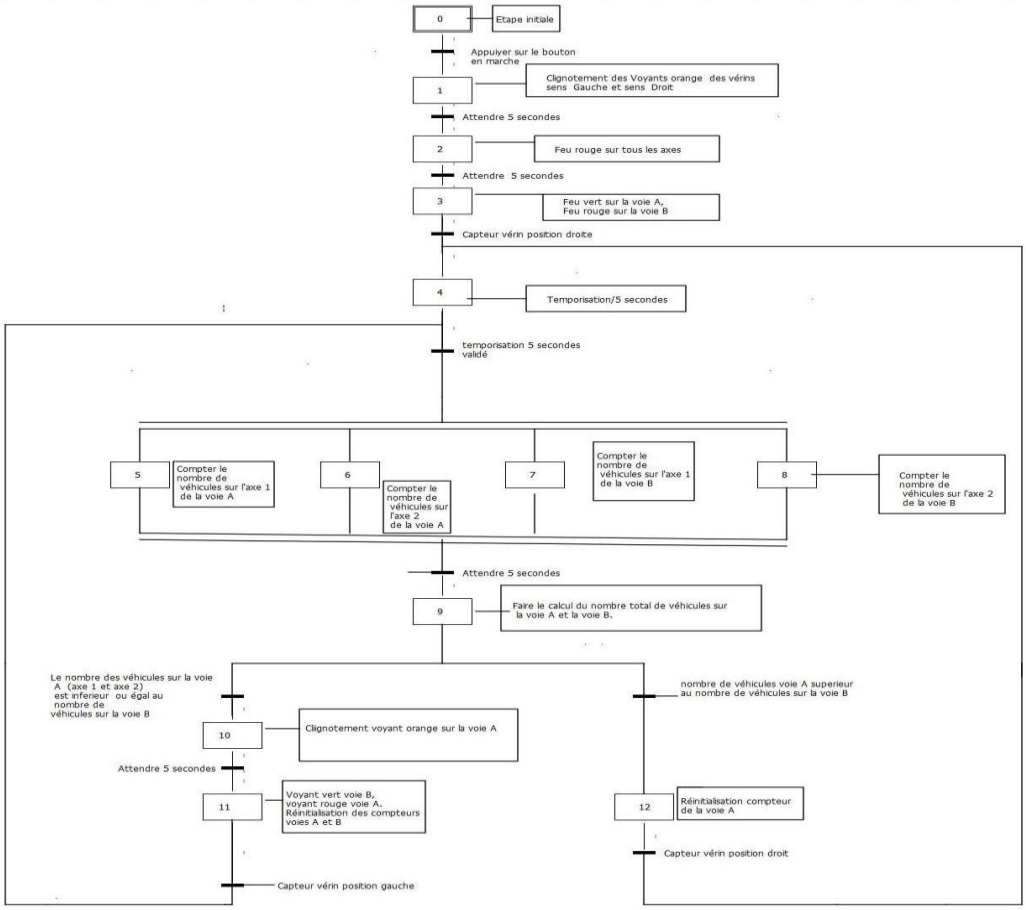

Figure 3: BINTI rolling robot control panel [O. BEYA SCORPIEN, E. KAUMBA, T. KASEBA KASEBA, G. KAMENA NKASA (2023), "Enhancing Urban Road Traffic Regulation Using Grafcet-based Command Control System for BINTI Rolling Robot in Lubumbashi", p823]

2.1 The field instruments of the BINTI rolling robot

The automation system will include a set of devices (position sensors, actuators, indicator light, etc.) which will be connected to the S7-300 programmable controller. We have chosen an appropriate nomenclature for each of these field instruments including:

1. SVG: Left position cylinder sensor
2. SVD: Right position cylinder sensor
3. C1: Vehicle counter on track A axis 1
4. C2: Vehicle counter on track A axis 2
5. C3: Vehicle counter track B axis 1
6. C4: Vehicle counter track B axis 2
7. VG: Left cylinder
8. VD: right cylinder
9. DCY: power button
10. LVA: green light channel A
11. LOA: orange indicator light track A
12. LRA: red light track A
13. LVB: green light channel B
14. LOB: orange indicator light channel B
15. LRB: red light track B
16. SMA: total vehicles track A
17. SMB: total vehicles track B
18. SA0: Vehicle sensor track A axis 1 waiting zone
19. SA1: Vehicle sensor track A axis 1 passage zone
20. SA2: Vehicle sensor track A axis 2 waiting zone
21. SA3: Vehicle sensor track A axis 2 passage zone
22. SB0: Vehicle sensor track B axis 1 waiting zone
23. SB1: Vehicle sensor track B axis 1 passage zone
24. SB2: Vehicle sensor track B axis 2 waiting zone
25. SB3: Vehicle sensor track B axis 2 passage zone

3. API programming

3.1. Creation of the project under STEP7 TIAPORTAL

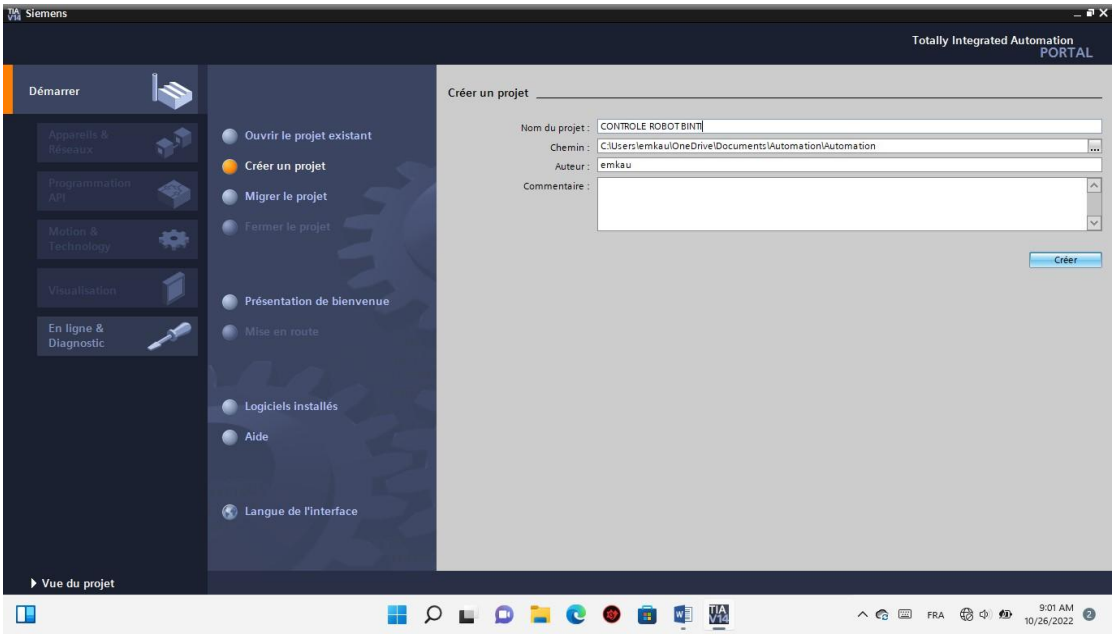

Figure 1: Creation of the project under Step7/TIA PORTAL

3.2. Hardware configuration of the S7-300 controller

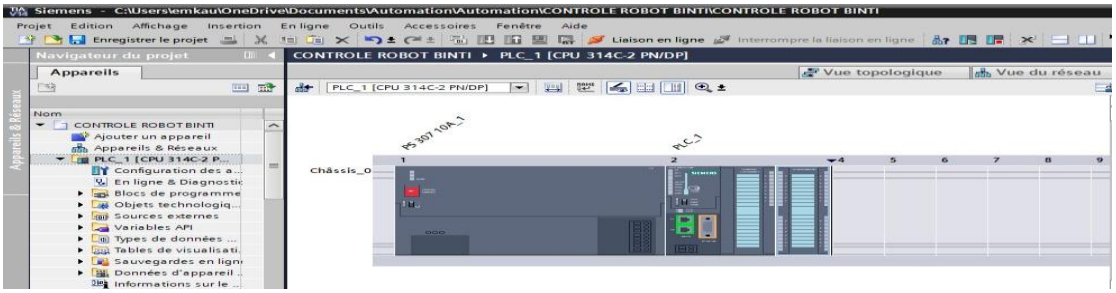

Figure 2: Hardware configuration of the s7-300 controller

3.3. Creation of mnemonics in the variable table

According to A. Andriambola (2016, 24), mnemonics are the names assigned to global API variables. The use of mnemonics instead of absolute addresses (e.g. E0.0=motor running) considerably improves the readability and clarity of a program and helps isolate possible faults. The mnemonics thus defined can be used throughout the user program of a programmable module.

a) Variables
In all programs it is necessary to define the list of variables which will be used during programming for this the variable table is created. Using the appropriate nomes makes the program more understandable and easier to handle. This type of addressing is called “relative”. After the name we define the data type of the variable, then the address. We fill in the table of variables respecting our specifications, for inputs and outputs.
Variables can be of type:

b) Entries
To know the status and progress of the process, the automaton collects information from the installation via automaton inputs which are connected to the various sensors and buttons of the installation to then process it and generate the command.

c) Exits
After processing the input data and to control the installation, the automaton must generate and send signals through these outputs. The PLC outputs are connected to the various valves and actuators of the installation.

d) Memo
Memory area in the system memory of a CPU. It is possible to access it for writing and reading (by bit, byte, word and double word). The memory area allows the user to save intermediate results.

Variables API

	Nom	Table des variables	Type de données	Adresse	Réma...	Acces...	Visibl...	Commentaire
	SA0	Table de variables s...	Bool	%I0.0		☑	☑	
	SA1	Table de variables s...	Bool	%I0.1		☑	☑	
	SA2	Table de variables s...	Bool	%I0.2		☑	☑	
	SA3	Table de variables s...	Bool	%I0.3		☑	☑	
	SB0	Table de variables s...	Bool	%I0.4		☑	☑	
	SB1	Table de variables s...	Bool	%I0.5		☑	☑	
	SB2	Table de variables s...	Bool	%I0.6		☑	☑	
	SB3	Table de variables s...	Bool	%I0.7		☑	☑	
	SVG	Table de variables s...	Bool	%I1.0		☑	☑	
0	SVD	Table de variables s...	Bool	%I1.1		☑	☑	
1	TEMPO	Table de variables s...	Bool	%M0.0		☑	☑	
2	VERIN ROTATION DROIT	Table de variables s...	Bool	%Q0.0		☑	☑	
3	VERIN ROTATION GAUCHE	Table de variables s...	Bool	%Q0.1		☑	☑	
4	LOA	Table de variables s...	Bool	%Q0.2		☑	☑	
5	LRA	Table de variables s...	Bool	%Q0.3		☑	☑	
6	LVA	Table de variables s...	Bool	%Q0.4		☑	☑	
7	LOB	Table de variables s...	Bool	%Q0.5		☑	☑	
8	LRB	Table de variables s...	Bool	%Q0.6		☑	☑	
9	LVB	Table de variables s...	Bool	%Q0.7		☑	☑	
0	DCY	Table de variables s...	Bool	%I1.2		☑	☑	
1	DELAY	Table de variables s...	Bool	%I1.3		☑	☑	
2	COMPTAGE VEHICULES VOIE A ...	Table de variables s...	Counter	%C1		☑	☑	
3	CONTROLE VOIE A	Table de variables s...	Bool	%I1.4		☑	☑	
4	REINT CPT C1	Table de variables s...	Bool	%M0.1		☑	☑	
5	TOTAL VEHICULES VOIE A AXE 1	Table de variables s...	Word	%MW2		☑	☑	
6	COMPTAGE VEHICULES VOIE A ...	Table de variabl...	Counter	%C2		☑	☑	
7	REINT CPT C2	Table de variables s...	Bool	%M0.2		☑	☑	

Figure 3: Creation of mnemonics in the variable table

3.4. Creating blocks

To carry out the automation task, the blocks containing the various programs and data must be loaded into the automaton. The existing blocks are (OB, FB, SFB, FC, SFC) which contain the programs, the instance DB and global DB data blocks which contain the program parameters.

1) Organization block (OB)
The block folder contains the blocks that must be loaded into the CPU to carry out the automation task, it includes:

- The code blocks (OB, FB, FC, SFB, SFC) which contain the programs,
- The instance DB and global DB data blocks which contain the program parameters.

We used the OB1 organization block which is called by the operating system, it calls on the other blocks which constitute the program, when we call a functional block in OB1 an associated data block will be created automatically.

2) Functional blocks (FB)
The FB is a subroutine written by the user and executed by blocks of code. It is associated with a DB instance data block relating to its memory and containing its parameters. For this program we used four blocks of this type, programmed in GRAPH language,

3) Data blocks (DB)
These data blocks are only used to store information and data but not instructions, this data will be used by other blocks.

3.5. Grafcet in the FB1 function block

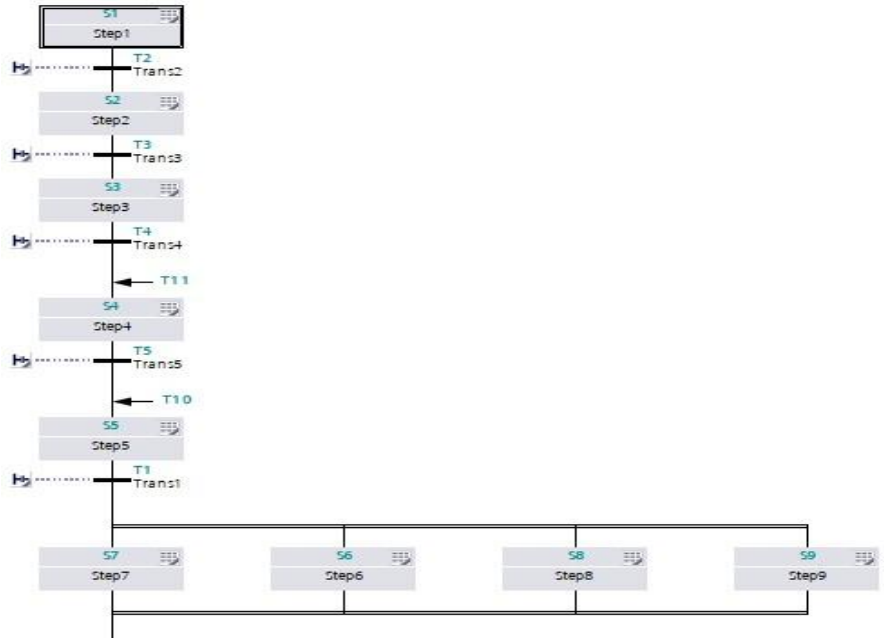

Figure 4: Creation of the grafcet of the BINTI robot program in the FB1 functional block

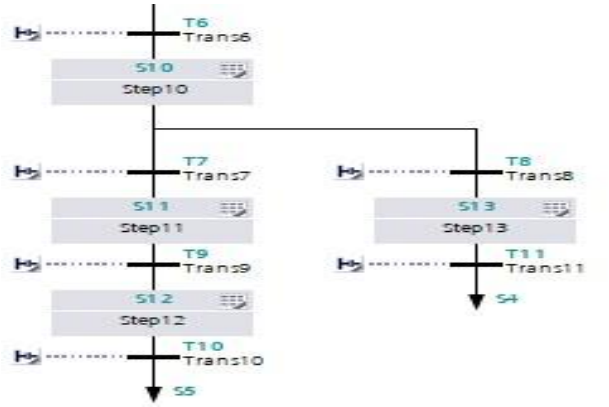

Figure 5: Creation of the grafcet of the BINTI robot program in the FB1 functional block

3.6 Function block (FC)

The FC contains routines for frequently used functions. It is memoryless and saves its temporary variables in the local data stack. However, it can use global data blocks to save its data.

3.6.1. Creation of functions FC1, FC2, FC3, FC4, FC5 and FC6, for counting vehicles and resetting counters.

1) Function FC1, FC2, FC3, FC3, FC4 for counting

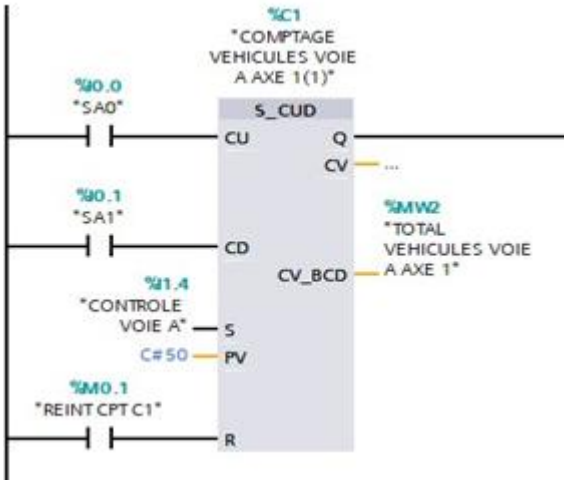

Figure 6: counter C1

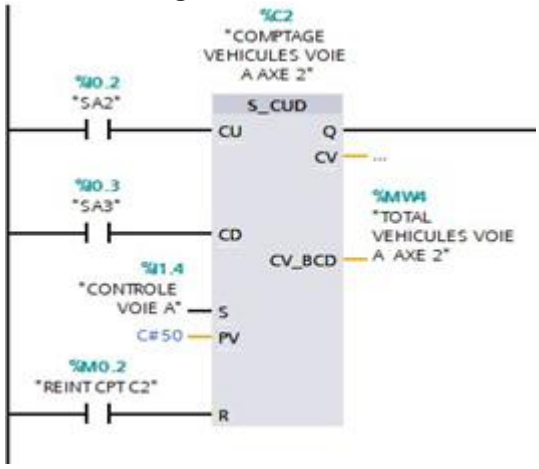

Figure 7: Counter C2

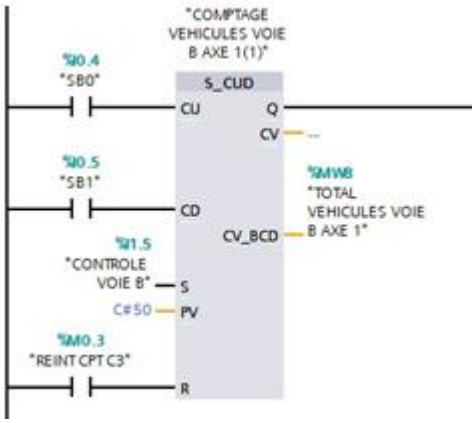

Figure 8: Counter C3

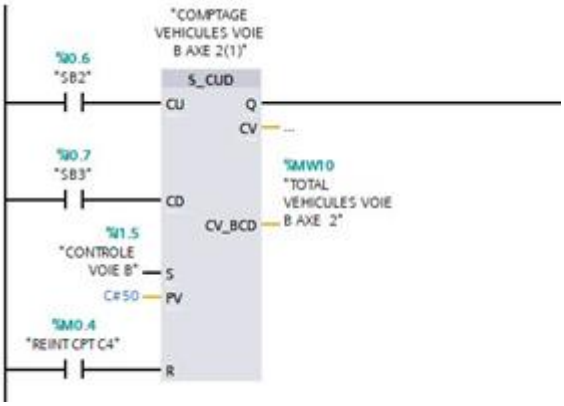

Figure 9: Counter C4

2) FC5 function for reset

The FC5 block contains two networks and is intended for calculating the number of vehicles

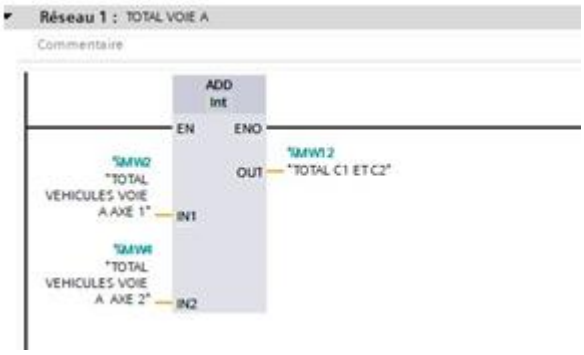

Figure 10: FC5 function for reset

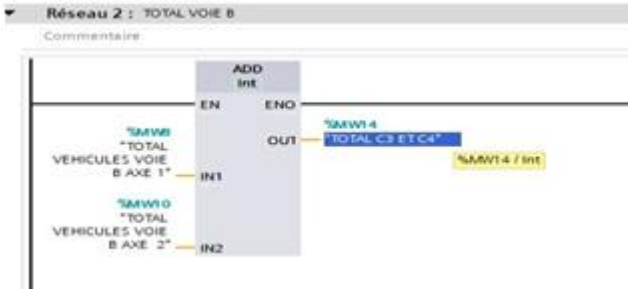

Figure 11: FC5 function for reset

1) FC6 function for reset

The FC6 block contains two networks and is intended for resetting counters

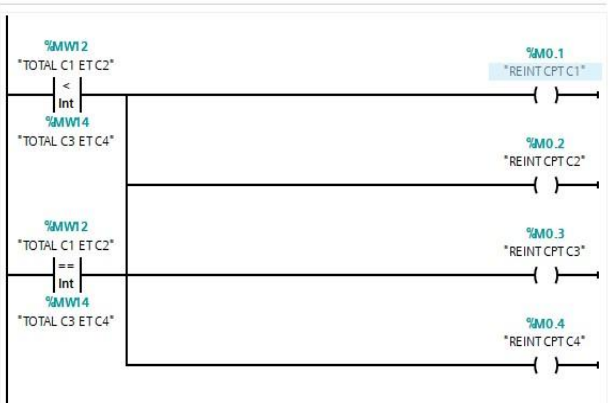

Figure 12: FC6 function for reset

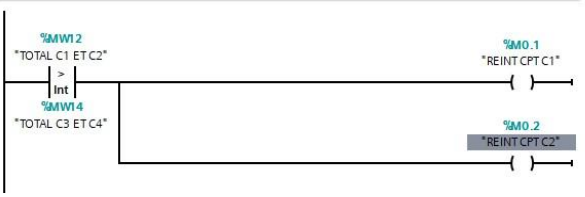

Figure 13: FC6 function for reset

3.6.2. Loading the FB1 function into the OB1 block and setting up the emergency stop in OB1

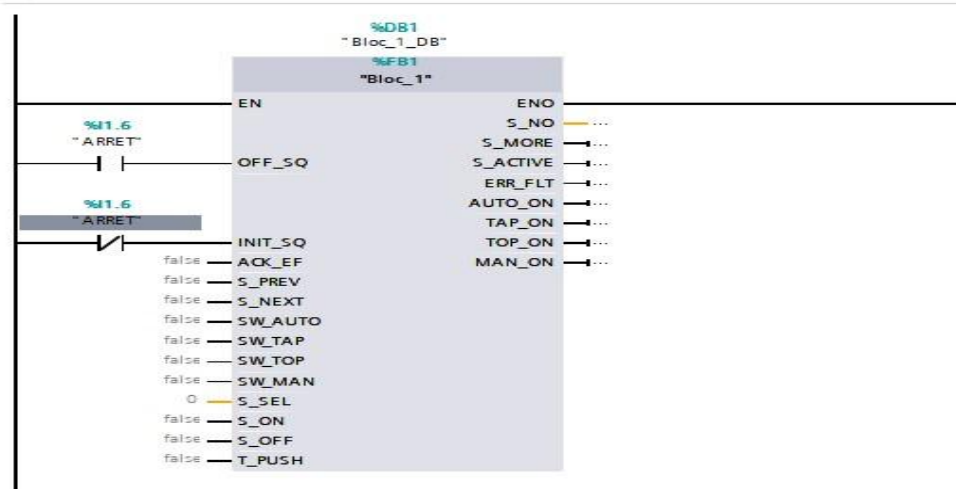

Figure 14: loading the function Fb1 in the OB1 block and setting up the emergency stop in OB1

3.6.3. Loading and testing the program in the PLCSIM

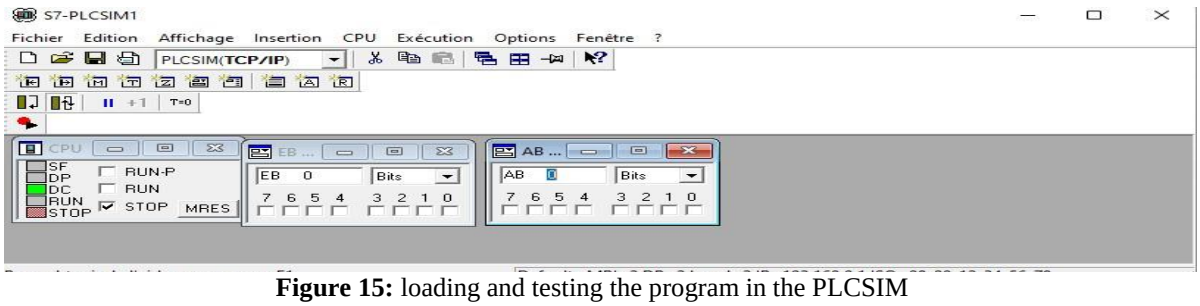

Figure 15: loading and testing the program in the PLCSIM

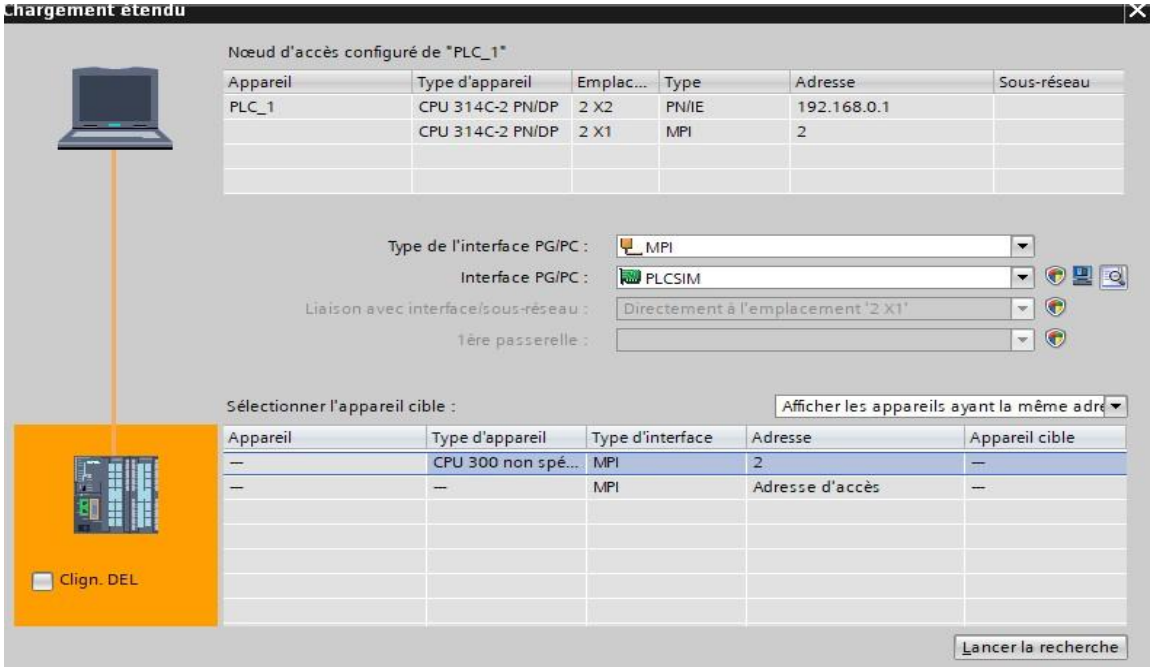

Figure 16: Connection with the API for simulation

4. Conclusion

Indeed, this study aimed to design the control program for the industrial programmable controller which will be responsible for monitoring and operating the BINTI robot. The TIA PORTAL software was the support on which the program was created and then tested on the PLCSLIM simulator. We therefore started by going through the specifications of the BINTI rolling robot which then allowed us to highlight the different input and output variables to finally carry out the programming using the TIA portal software.

References

Works

[1] BENOIT CHARROUX, Aomar OSMAN, Thierry-Mieg YANN, practice of modeling 2nd edition, Perason Education France, 2009

[2] BOUCHAHED ADEL, General structure of an automated system and automation elements; automation course, Institute of Applied Sciences and Techniques-UFM-Constantine-1.

[3] FOX-HERO M., "Automation course 4", Collège l'harmattant, Course Notes, 2022.

[4] FOX-HERO M., "Automation course 6", Collège l'harmattant, Course Notes, 2022.

[5] JEANCY DIASOLUA, study of the development of an intelligent driving robot: Counting of objects and evaluation of Traffic, University of Kinshasa, Degree in computer science, 2015.

[6] LYSEE BERNARD PALISSY, cours-i2511-IEC1131v030.odt, ed.LibreOffice, 2014

Websites

[7] https://humanoides.fr/2014/01/des-robots-mades-in-congo-font-la-circulation-akinshasa, accessed on May 12, 2023.

[8] https://www.memoireonline.com/12/21/12580/tude-de-llaboration-dun-robotroulage-intelligent-Comptage-des-objets-et-.html, accessed July 22, 2023