AUTOMATED HANDLING OF MATERIALS IN LOGISTICS 4.0

SCARLAT Andrei Daniel
Faculty of Industrial Engineering and Robotics, Study program: Industrial Logistics, Academic year: 4, e-mail: scarlat.andrei.upb@gmail.com
Scientific coordinator: PhD. Lecturer Constantin - Adrian POPESCU

Summary: The purpose of this research was to create an AGV (Automated Guided Vehicle) to transport pallets from the warehouse and reduce work accidents. The following components were used: an Arduino board, an L293N2 motor driver, two geared motors and two wheels, an HC-SR04 distance sensor to avoid obstacles on the way, 2 infrared sensor modules for line tracking, a holder for 2 Li-ion batteries and the batteries themselves. A pallet grabbing system was also implemented using an SG90 servo motor.

Keywords: AGV, distance sensor, Arduino, Li-ion 18650 battery, L298N driver.

1. Introduction

In the context of logistics 4.0, technology plays an increasingly important role in the process of handling and transporting products. However, handling and transport issues remain one of the main challenges of the logistics industry. In this regard, the development and implementation of automated product handling solutions represent a priority for the logistics industry. In this study, we will analyze the technologies of automated product handling used in logistics 4.0 and present the integration of an AGV project in this context.

In the logistics industry, automated product handling technologies are becoming increasingly important. In this trend, robots and AGVs (Automated Guided Vehicles) are the most used solutions. Robotic wheels and arms are capable of performing a wide range of tasks, such as lifting and placing products in specific locations, while AGVs can be used to transport products between different points within a logistics facility.

Additionally, process control systems, which integrate sensors and artificial intelligence, can be used to automate and optimize the product handling process.

2. The current stage

Human operators traditionally have several disadvantages compared to automated guided vehicles (AGVs). Logistics service providers need to pay salaries on time, provide vacation time, and sick leave in case of accidents, which in many cases are inevitable. In contrast, an AGV only needs to be refueled from one working cycle to another and can avoid accidents and obstacles using a distance sensor that helps estimate the distance from the object to the AGV.

Moreover, when using a human operator for providing logistics services in a warehouse, unintentional human errors may occur, causing products to be mixed up or not stored in the designated location. However, in the case of using an AGV, these events cannot occur because they repeat the algorithm and route for which they were assigned. Before we begin, it is worth noting that using an AGV represents a major advantage for a logistics center looking to obtain high-quality logistics services, without errors, and in the shortest possible time.

Let's start by presenting the block diagram of the AGV's component connection, designed as a prototype to make it easier to understand the process of picking up a pallet with goods and transporting it from one point to another.

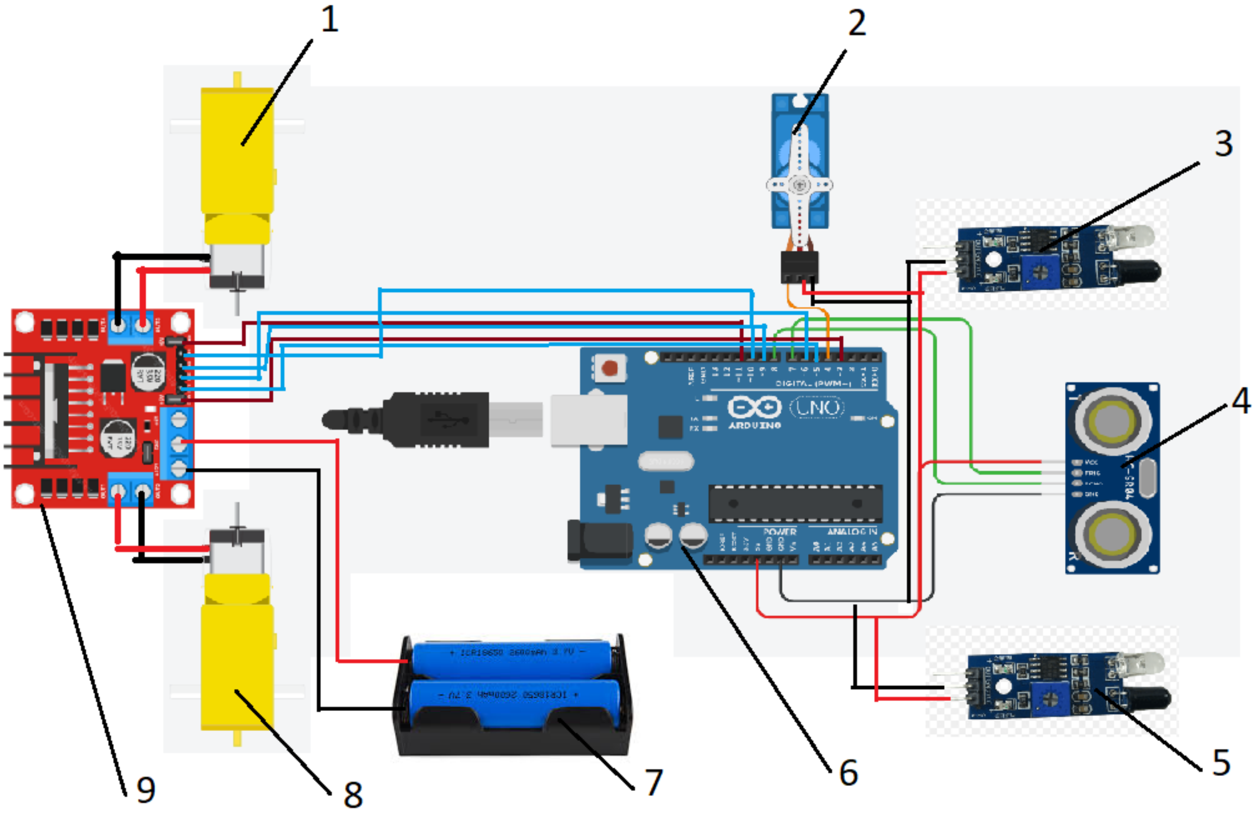


Fig. 1. Wiring diagram

Table 1. Components table

Nr.crt	
1.	Left gearbox motor
2.	SG90 Servo motor
3.	Left IR sensor
4.	HCSR-04 distance sensor
5.	Right IR sensor
6.	Arduino board
7.	Li-ION 18650 batteries
8.	Right gearbox motor
9.	L298N driver

The 3D model of the AGV that led to its physical creation from the virtual model is presented in Fig. 2.

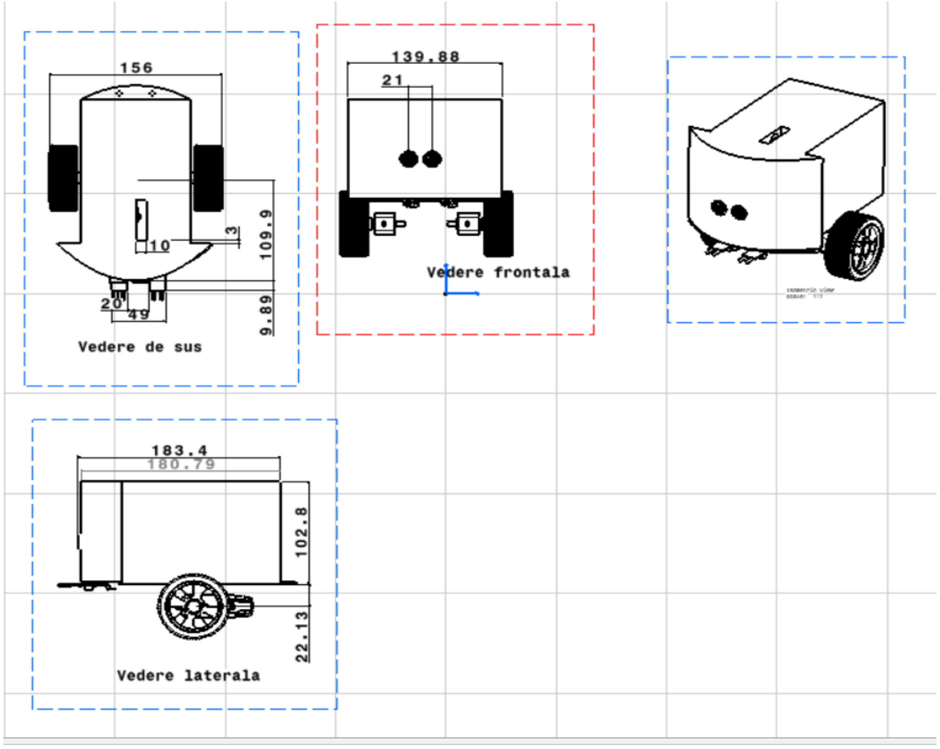
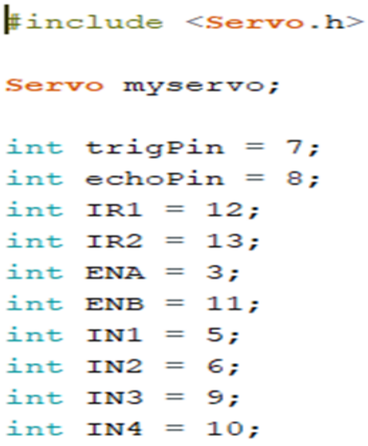


Fig. 2. 3D model of the AGV

The chassis model is made of plexiglass, measuring 20x10, on the back of which the 2 wheel motors were attached, and on the top surface the Arduino board was attached, a L293N motor driver, which was interconnected with the Arduino board, the battery holder with 2 Li-ION batteries. The HC-SR04 distance sensor was also connected to the board to avoid obstacles and the 2 IR infrared sensor modules to track the AGV guide line. The connections between the driver, board, and sensors were made through jumper wires. The casing was made of cardboard, which was later painted to give it a unique and pleasant appearance, according to the virtual model. First, the input variables must be defined (for example, pins of the distance sensor, motor pins, and IIR pins). Then, the algorithm for detecting objects is introduced.

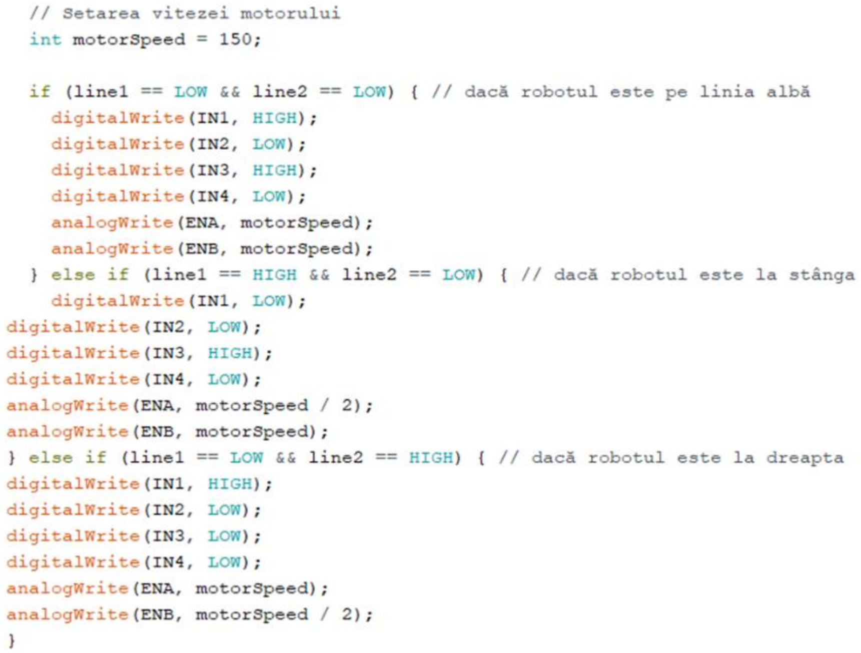


```
#include <Servo.h>

Servo myservo;

int trigPin = 7;
int echoPin = 8;
int IR1 = 12;
int IR2 = 13;
int ENA = 3;
int ENB = 11;
int IN1 = 5;
int IN2 = 6;
int IN3 = 9;
int IN4 = 10;
```

Fig. 3. The pin definitions for the AGV components

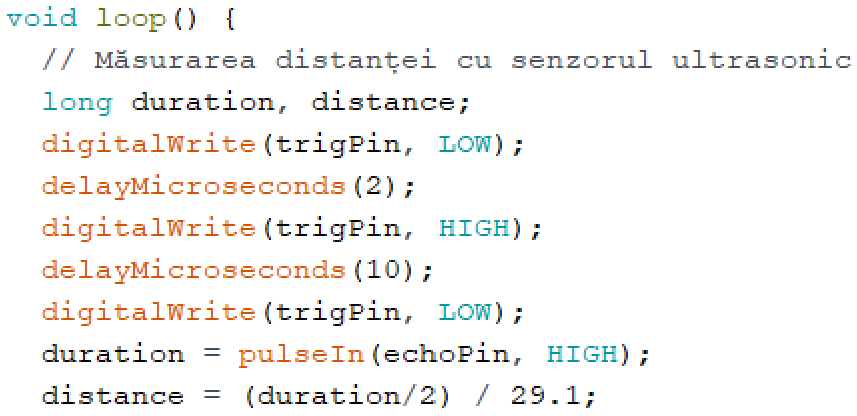


```
  // Setarea vitezei motorului
  int motorSpeed = 150;

  if (line1 == LOW && line2 == LOW) { // dacă robotul este pe linia albă
    digitalWrite(IN1, HIGH);
    digitalWrite(IN2, LOW);
    digitalWrite(IN3, HIGH);
    digitalWrite(IN4, LOW);
    analogWrite(ENA, motorSpeed);
    analogWrite(ENB, motorSpeed);
  } else if (line1 == HIGH && line2 == LOW) { // dacă robotul este la stânga
    digitalWrite(IN1, LOW);
digitalWrite(IN2, LOW);
digitalWrite(IN3, HIGH);
digitalWrite(IN4, LOW);
analogWrite(ENA, motorSpeed / 2);
analogWrite(ENB, motorSpeed);
} else if (line1 == LOW && line2 == HIGH) { // dacă robotul este la dreapta
digitalWrite(IN1, HIGH);
digitalWrite(IN2, LOW);
digitalWrite(IN3, LOW);
digitalWrite(IN4, LOW);
analogWrite(ENA, motorSpeed);
analogWrite(ENB, motorSpeed / 2);
}
```

Fig. 4. The algorithm for the AGV movement

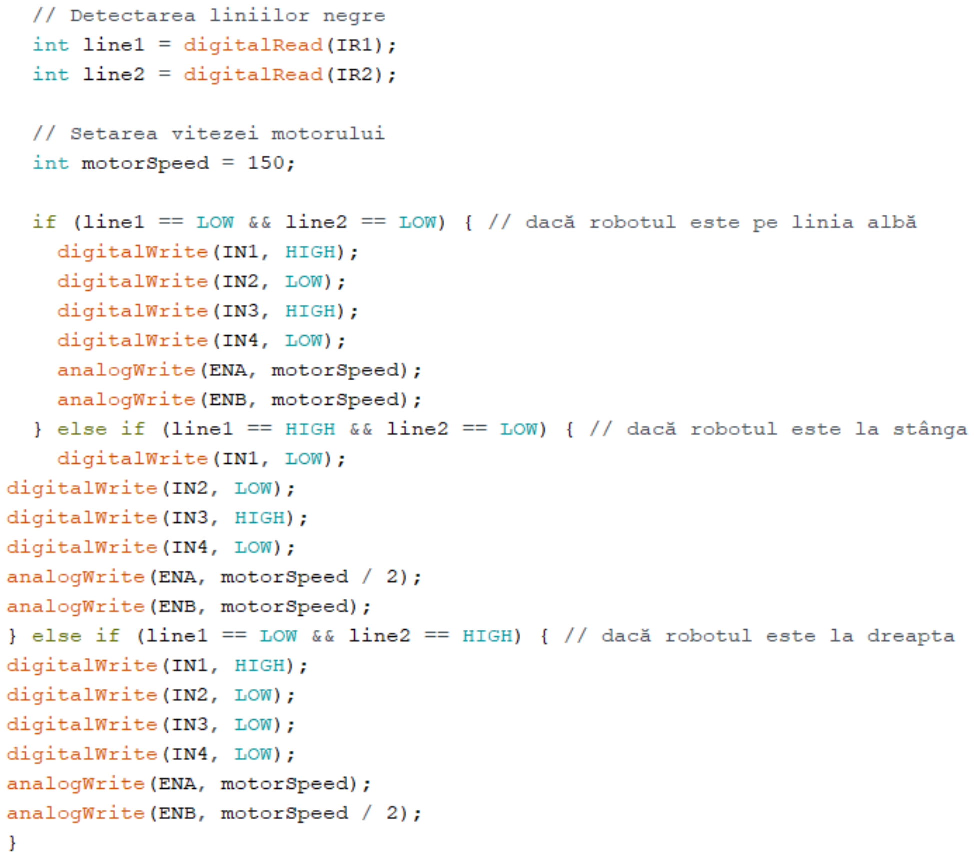
The operating principle is as follows: The L293N driver works with the 2 gear motors through the transmission and distributes the power supply received from the battery. The Arduino board orders the movement of the AGV by transmitting commands from the board's microprocessor to the driver. The HC-SR04 distance sensor detects objects appearing on the path and, if they are within 10 cm, redirects the AGV off the path to avoid collision.



```
void loop() {
  // Măsurarea distanței cu senzorul ultrasonic
  long duration, distance;
  digitalWrite(trigPin, LOW);
  delayMicroseconds(2);
  digitalWrite(trigPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigPin, LOW);
  duration = pulseIn(echoPin, HIGH);
  distance = (duration/2) / 29.1;
```

Fig. 6. The algorithm for distance calculation

Set the trigger pin of the HC-SR04 distance sens the infrared IR sensor modules detect the base color of the floor and follow the black line, and when the line is perpendicular to the direction of travel, it sends a command to the Arduino board to stop the motor reducers, as the AGV has reached its destination.



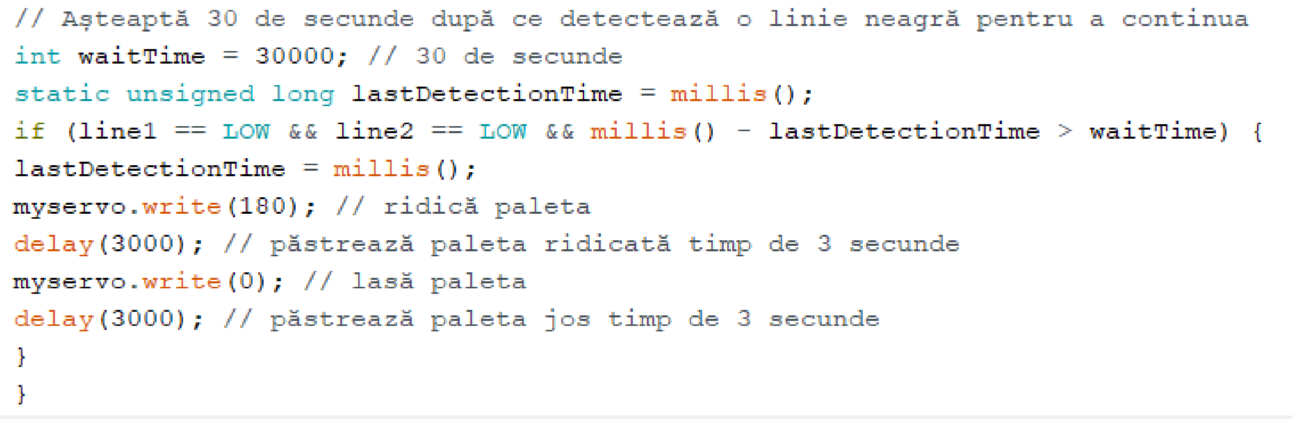
```
  // Detectarea liniilor negre
  int line1 = digitalRead(IR1);
  int line2 = digitalRead(IR2);

  // Setarea vitezei motorului
  int motorSpeed = 150;

  if (line1 == LOW && line2 == LOW) { // dacă robotul este pe linia albă
    digitalWrite(IN1, HIGH);
    digitalWrite(IN2, LOW);
    digitalWrite(IN3, HIGH);
    digitalWrite(IN4, LOW);
    analogWrite(ENA, motorSpeed);
    analogWrite(ENB, motorSpeed);
  } else if (line1 == HIGH && line2 == LOW) { // dacă robotul este la stânga
    digitalWrite(IN1, LOW);
digitalWrite(IN2, LOW);
digitalWrite(IN3, HIGH);
digitalWrite(IN4, LOW);
analogWrite(ENA, motorSpeed / 2);
analogWrite(ENB, motorSpeed);
} else if (line1 == LOW && line2 == HIGH) { // dacă robotul este la dreapta
digitalWrite(IN1, HIGH);
digitalWrite(IN2, LOW);
digitalWrite(IN3, LOW);
digitalWrite(IN4, LOW);
analogWrite(ENA, motorSpeed);
analogWrite(ENB, motorSpeed / 2);
}
```

Fig. 7. Algorithm for setting up IR sensors and motor-reducer speed

At that moment, the SG90 servo motor makes a 180-degree movement to pick up the pallet.



```
// Așteaptă 30 de secunde după ce detectează o linie neagră pentru a continua
int waitTime = 30000; // 30 de secunde
static unsigned long lastDetectionTime = millis();
if (line1 == LOW && line2 == LOW && millis() - lastDetectionTime > waitTime) {
lastDetectionTime = millis();
myservo.write(180); // ridică paleta
delay(3000); // păstrează paleta ridicată timp de 3 secunde
myservo.write(0); // lasă paleta
delay(3000); // păstrează paleta jos timp de 3 secunde
}
}
```

Fig. 8. Algorithm for pallet handling mechanism

After connecting the elements and uploading the code to the Arduino board, we obtained a prototype as shown in fig.9.

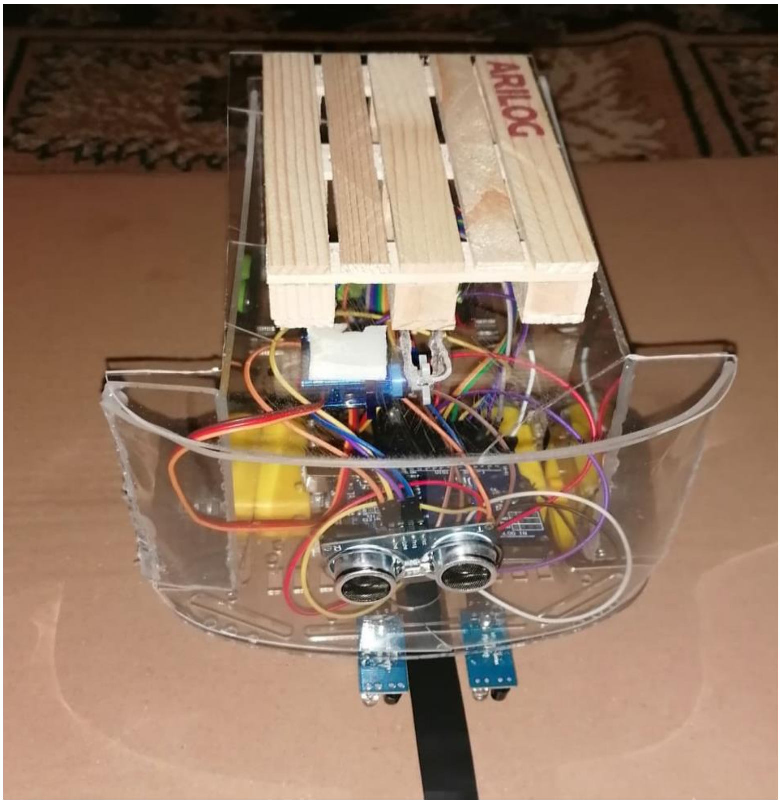


Fig. 9. Prototype of AGV

3. Conclusions

In conclusion, automated product handling technologies are playing an increasingly important role in the logistics industry, and Logistics 4.0 offers excellent opportunities for optimizing product handling and transportation processes. The AGV project presented in this study can be integrated into a Logistics 4.0 system through the use of sensors and artificial intelligence. Implementing these technologies can lead to improved efficiency and cost reduction in the logistics industry.

4. Bibliography

[1]. Traian Anghel (2020), Programming play Arduino, Editura Parallel 45, ISBN 978-973-47-3204-3
[2]. O'Reilly (2012), Environmental Monitoring with Arduino, ISBN 978-1-449-31056-1
[3]. https://logistic-specialist.ro/logistica-automatizata-ce-sunt-agv-urile/
[4].https://www.researchgate.net/publication/347805525_Development_of_Low_Cost_Pellet_Loading_and_Unloading_AGV