

Cancela Automática para processos industriais

Geisiel Garcia Leal

Luiz Antônio Graciano Júnior

Michael Almeida Machado

Wellington Henrique de Souza

Resumo: Esse projeto consiste na implementação de uma cancela automática, qual possui a finalidade em ajudar no controle de veículos e pedestres que serão implantadas dentro de uma área fabril. A implementação da cancela automática tem como objetivo principal criar um sistema inteligente que automatizará o controle de acesso através de sensores e de um micro controlador, otimizando a segurança e trazendo mais praticidade em diferentes áreas de circulação dentro do ambiente fabril. O projeto de melhoria visa aumentar a segurança na da circulação de pedestre no ambiente fabril, trazendo mais proteção e redução no número de acidentes por atropelamento com o sistema que contém sinalizadores que indicarão quando o veículo rebocador estiver em rota de abastecimento para que as cancelas de pedestre sejam bloqueadas momentaneamente.

Palavras-Chave: Cancela automática, sensores, micro controlador, segurança, veículo rebocador.

Abstract: This project consists of implementing an automatic cancellation, which aims to help control vehicles and pedestrians that will be deployed within an industrial area. The main objective of implementing automatic cancellation is to create an intelligent system that automates access control through sensors and a microcontroller, optimizing security and bringing more practicality in different circulation areas within the manufacturing environment. The improvement project aims to increase safety in pedestrian circulation in the manufacturing environment, bringing more protection and reducing the number of accidents due to being run over with the system that contains flags that indicate when the towing vehicle is on the supply route so that the pedestrian gates become blocked momentarily.

Keywords: Automatic gate, sensors, micro controller, security, towing vehicle.

1 INTRODUÇÃO

O projeto da cancela automática foi escolhido pensando na segurança dos colaboradores da empresa, podendo ser aplicado em diversos ambientes do cenário fabril. Tendo em vista a melhoria no processo produtivo, deixando o processo mais eficaz e seguro para a movimentação dos veículos rebocadores e para os colaboradores que estiverem na rota de travessia. Foi escolhido esse tema para o desenvolvimento de projeto, através de experiências de atos inseguros vividas no ambiente de trabalho e relatos de companheiros de trabalho sobre atos inseguros em diversos locais, sendo de diversos riscos e gravidades (ex: desde uma escoriação até a um acidente fatal).

Em consenso com todo o grupo escolhemos este tema para desenvolvimento, visando em uma melhoria no processo produtivo deixando-o mais rápido e eficaz com a segurança em primeiro lugar, e reduzir ao máximo possível o numero de acidentes envolvendo veículos e pedestres.

E lembrando novamente que a segurança é indiscutível em qualquer cenário fabril. E podendo ser melhorado quanto a prioridade do fluxo da linha de produção e segurança de todos os envolvidos.

2 OBJETIVO

Identificando os riscos na travessia de pedestres nas empresas, esse projeto tem como foco garantir um ambiente mais seguro e aumentando a qualidade de serviços. O projeto das cancelas automáticas será implantado nos pontos estratégicos do cenário fabril para garantir a eficácia do mesmo.

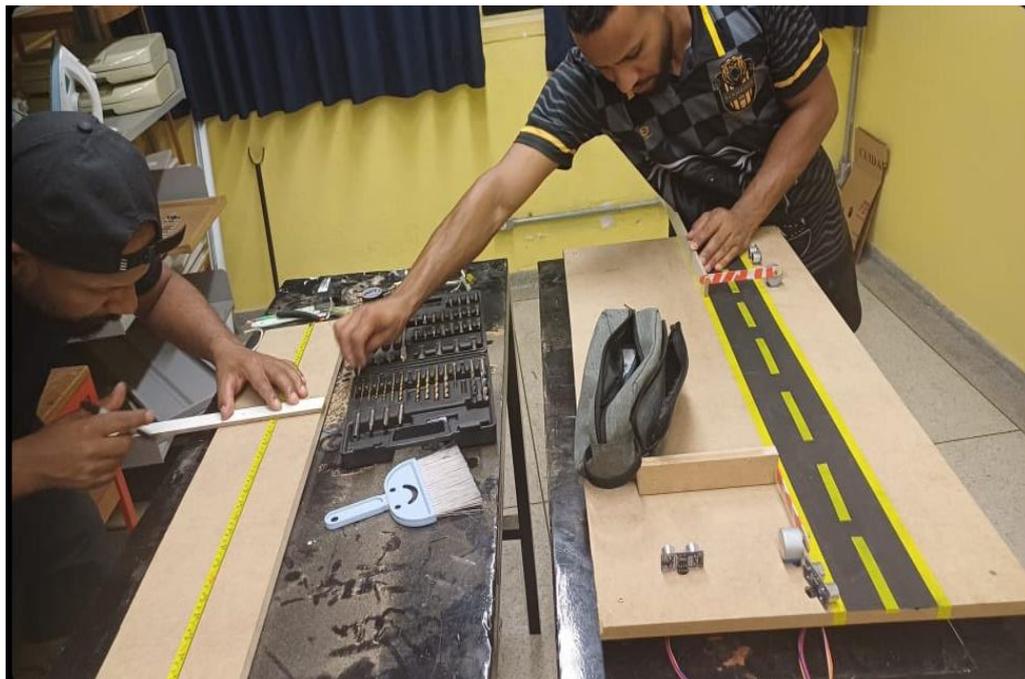
3 DESENVOLVIMENTO

O projeto de implementação das cancelas automáticas industriais foi pensado na segurança no ambiente fabril, e analisando o cenário dentro de uma fábrica encontramos este ponto de atenção, pois a movimentação de veículos rebocadores com peças e matéria prima na maioria das grandes empresas tem o fluxo elevado.

Numa análise de caso chegamos a implementação de um sistema de cancelas automáticas para pedestres e veículos rebocadores, tendo em vista a eficácia do processo produtivo e priorizando a segurança de todos.

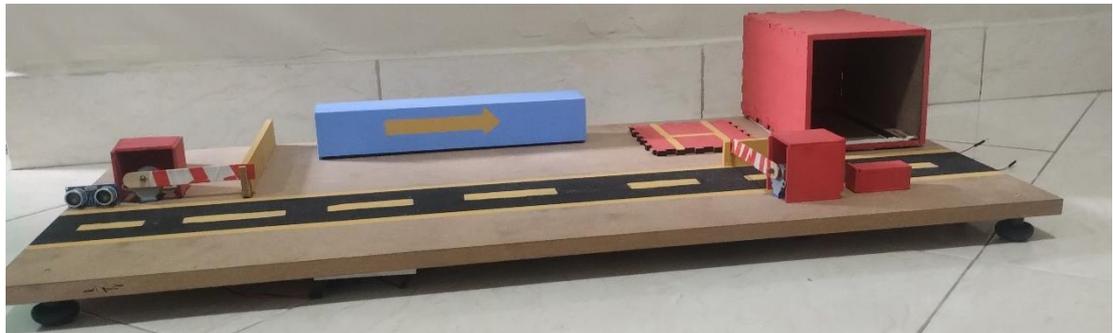
A cancela funcionará da seguinte forma, o veículo rebocador passa por um sensor que irá mandar um sinal luminoso para a cancela dos pedestres que servirá de aviso para os pedestres que o veículo está vindo, e entre o tempo desse sensor até o veículo rebocador passar no próximo sensor, será contado quantas pessoas passarão neste tempo. E quando o veículo passar no próximo sensor será bloqueado a cancela para os pedestres e abrindo a cancela principal que é a cancela do veículo rebocador.

A prioridade no fluxo produtivo será sempre do veículo rebocador, desde que o condutor do veículo tenha a visão de que ele e os demais colaboradores que estão a volta do trajeto esteja em um fluxo seguro. No término da rota de fluxo deste ponto de travessia de pedestres, terá mais um sensor que fará com que a cancela principal da rota seja fechada, e seja liberada novamente a travessia dos pedestres.

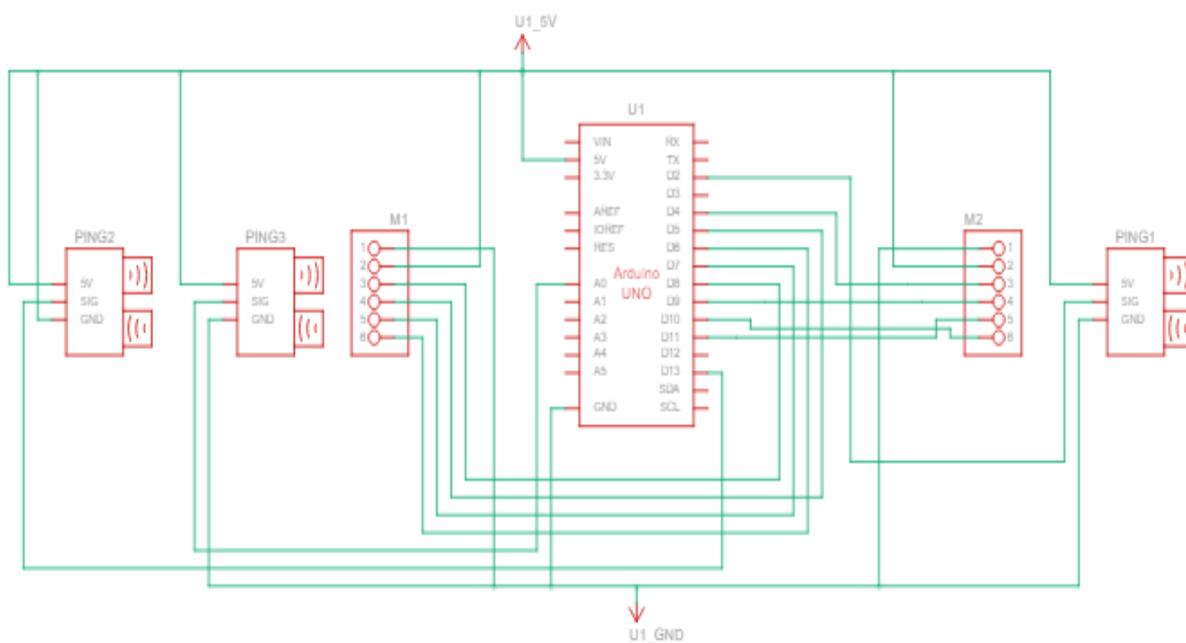


4 CONSIDERAÇÕES FINAIS

Portanto a segurança dentro de uma empresa é primordial e indiscutível, sendo assim chegamos a este projeto com o foco na segurança e qualidade no trabalho de um modo geral, para que os colaboradores sempre tenham consciência dos riscos que estão a sua volta, realizando alguma atividade ou até mesmo numa simples travessia de um ponto ao outro.



5 Esquema Elétrico



6 Programação

```
#include <Stepper.h>
```

```
const int stepsPerRevolution = 65; // Número de passos por revolução do motor
```

```
const int motorPin1 = 8; // Conecte o pino 8 do Arduino ao pino IN1 do driver do motor
```

```
const int motorPin2 = 9; // Conecte o pino 9 do Arduino ao pino IN2 do driver do motor
```

```
const int motorPin3 = 10; // Conecte o pino 10 do Arduino ao pino IN3 do driver do motor
```

```
const int motorPin4 = 11; // Conecte o pino 11 do Arduino ao pino IN4 do driver do motor
```

```
const int motorZPin1 = 4; // Conecte o pino 4 do Arduino ao pino IN1 do driver do motor
```

```
const int motorZPin2 = 5; // Conecte o pino 5 do Arduino ao pino IN2 do driver do motor
```

```
const int motorZPin3 = 6; // Conecte o pino 6 do Arduino ao pino IN3 do driver do motor
```

```
const int motorZPin4 = 7; // Conecte o pino 7 do Arduino ao pino IN4 do driver do motor
```

```
const int triggerPin1 = 2; // Conecte o pino 2 do Arduino ao pino de trigger do sensor ultrassônico 1
```

```
const int echoPin1 = 3; // Conecte o pino 3 do Arduino ao pino de echo do sensor ultrassônico 1
```

```
const int triggerPin2 = 12; // Conecte o pino 4 do Arduino ao pino de trigger do sensor ultrassônico 2
```

```
const int echoPin2 = 13; // Conecte o pino 5 do Arduino ao pino de echo do sensor ultrassônico 2
```

```
const int triggerPin3 = A0; // Conecte o pino 6 do Arduino ao pino de trigger do sensor ultrassônico 3
```

```
const int echoPin3 = A1; // Conecte o pino 7 do Arduino ao pino de echo do sensor
ultrassônico 3

long duration1, distance1;

long duration2, distance2;

long duration3, distance3;

//Stepper myStepper(stepsPerRevolution, motorPin1, motorPin2, motorPin3,
motorPin4);

Stepper motorX(stepsPerRevolution, motorPin1, motorPin3, motorPin2, motorPin4);

Stepper motorZ(stepsPerRevolution, motorZPin1, motorZPin3, motorZPin2,
motorZPin4);

void setup() {

pinMode(triggerPin1, OUTPUT);

pinMode(echoPin1, INPUT);

pinMode(triggerPin2, OUTPUT);

pinMode(echoPin2, INPUT);

pinMode(triggerPin3, OUTPUT);

pinMode(echoPin3, INPUT);

motorX.setSpeed(60); // Defina a velocidade do motor (em RPM)

motorZ.setSpeed(60);

Serial.begin(9600); // Inicializa a comunicação serial

}

void loop() {

distance1 = medeDist(triggerPin1, echoPin1);

Serial.println("Distancia medida pelo Sensor 1: " + (String)distance1);

// Verifica se a distância medida pelo Sensor 1 é menor que 10 cm

if (distance1 < 10) {
```

```
// Move o motor 1 a 90 graus no sentido horário
motorX.step(500)
Serial.println("Presença detectada pelo Sensor 1! Motor 1 girou 90 graus.");
delay(1000); // Aguarda um pouco para evitar detecções repetidas do mesmo objeto
// Move o motor 2 a -90 graus
motorZ.step(-500);
Serial.println("Motor 2 girou -90 graus.");
delay(1000);
}
distance2 = medeDist(triggerPin2, echoPin2);
Serial.println("Distancia medida pelo Sensor 2: " + (String)distance2);
// Verifica se a distância medida pelo Sensor 2 é menor que 10 cm
if (distance2 < 10) {
// Move o motor 2 a 90 graus no sentido horário
motorX.step(-500);
Serial.println("Presença detectada pelo Sensor 2! Motor 2 girou 90 graus.");
delay(1000); // Aguarda um pouco para evitar detecções repetidas do mesmo objeto
// Move o motor 1 a -90 graus
motorZ.step(500);
Serial.println("Motor 1 girou -90 graus.");
delay(1000);
}
distance3 = medeDist(triggerPin3, echoPin3);
Serial.println("Distancia medida pelo Sensor 3: " + (String)distance3);
// Verifica se a distância medida pelo Sensor 3 é menor que 10 cm
```

```
if (distance3 < 10) {  
  // Reseta os motores para a posição inicial  
  motorX.step(-500);  
  motorZ.step(500);  
  Serial.println("Presença detectada pelo Sensor 3! Motores foram resetados.");  
  delay(1000); // Aguarda um pouco para evitar detecções repetidas do mesmo objeto  
}  
delay(2000);  
}  
long medeDist(int triggerPin, int echoPin) {  
  // Gera um pulso de 10 microssegundos no pino de trigger  
  digitalWrite(triggerPin, LOW);  
  delayMicroseconds(2);  
  digitalWrite(triggerPin, HIGH);  
  delayMicroseconds(10);  
  digitalWrite(triggerPin, LOW);  
  // Mede o tempo de ida e volta do pulso  
  long duration = pulseIn(echoPin, HIGH);  
  // Calcula a distância em centímetros  
  long distance = duration * 0.034 / 2;  
  return distance;  
}
```

7 Lista de materiais

Os materiais usados foram:

- Uma placa de madeira (MDF)
- Palitos
- Pezinhos
- Três sensores ultrassônicos
- Dois motores de Passo
- Um Arduino UNO
- Protoboard
- Cabos machos e cabos fêmea
- Notebook para a leitura do programa

REFERÊNCIAS

<https://www.tecsevenautomacao.com.br/cancela-estacionamento>

<https://github.com/Power041/cancela>

<https://www.robocore.net/tutoriais/controlando-motor-de-passo>